

Universidad Nacional de Ingeniería

Facultad de Ingeniería Química y Textil



TESIS

**Modelo neuronal híbrido basado en la ecuación de estado de
Peng-Robinson para predecir el equilibrio líquido vapor en la
mezcla CO₂-H₂O**

Para obtener el Título Profesional de Ingeniero Químico

Elaborado por

Luis Gianfranco Gonzalez Alvarez

 [0009-0003-5814-1009](https://orcid.org/0009-0003-5814-1009)

Asesor

Dr. Edgar Waldo Norabuena Meza

 [0000-0003-3921-931X](https://orcid.org/0000-0003-3921-931X)

TOMO I DE I

LIMA – PERÚ

2024

Citar/How to cite	Gonzalez Alvarez [1]
Referencia/Reference	[1] L. Gonzalez Alvarez, “ <i>Modelo neuronal híbrido basado en la ecuación de estado de Peng-Robinson para el equilibrio líquido vapor de la mezcla CO₂-H₂O</i> ” [Tesis de pregrado]. Lima (Perú): Universidad Nacional de Ingeniería, 2024.
Estilo/Style: IEEE (2020)	

Citar/How to cite	(Gonzalez, 2024)
Referencia/Reference	Gonzalez, L. (2024). <i>Modelo neuronal híbrido basado en la ecuación de estado de Peng-Robinson para el equilibrio líquido vapor de la mezcla CO₂-H₂O</i> . [Tesis de pregrado, Universidad Nacional de Ingeniería]. Repositorio institucional Cybertesis UNI.
Estilo/Style: APA (7ma ed.)	

Dedicatoria

A mi prometida Kita, por su amor, apoyo constante y por siempre creer en mí.

A mis padres Elsa y Luis por motivarme a realizar este trabajo de investigación.

A mis hermanos Rodrigo y Elizabeth por siempre sacarme una sonrisa.

Agradecimientos

A la Facultad de Ingeniería Química y Textil de la Universidad Nacional de Ingeniería, y sus docentes que me brindaron el conocimiento necesario a lo largo de mis estudios de pregrado para poder desarrollar la presente tesis.

Al Dr. Edgar Norabuena por asesorarme en la elaboración y revisión de la presente tesis.

Resumen

En este trabajo, se estudió el equilibrio líquido-vapor (ELV) de la mezcla $\text{CO}_2\text{-H}_2\text{O}$, empleando la ecuación de estado de Peng-Robinson (EOS-PR). Se identificó que la EOS-PR no ajustaba adecuadamente al comportamiento experimental, especialmente en la curva de rocío. Con el objetivo de mejorar las predicciones, se desarrolló un modelo híbrido utilizando redes neuronales. Para ello, se creó un algoritmo que permite calcular el parámetro “a” de mezcla en fase líquida de la EOS-PR y el parámetro de interacción binaria “ s_{ij} ” de la regla de mezcla de Van der Waals utilizando las fracciones molares experimentales de CO_2 en fase líquida y vapor, las presiones y la temperatura de cada isoterma y aplicando el método de optimización estocástica de recocido simulado en un rango de temperaturas entre 267°C a 350°C.

A partir de este análisis, se implementaron dos redes neuronales con arquitectura [2, 20, 10, 1], una para predecir el parámetro de interacción binaria “ s_{ij} ” y otra para ajustar el parámetro “a” de mezcla en fase líquida mediante una corrección al valor determinado por la regla de mezcla de Van der Waals. Los modelos fueron entrenados utilizando datos experimentales parametrizados recopilados por Tödheide & Franck (1963), y se aplicó un aumento de datos mediante splines paramétricos. Los resultados muestran que la combinación de funciones de activación ReLU-TanH predicen las isothermas experimentales con un error medio de porcentaje absoluto de 2.09% para temperaturas entre 267°C a 350°C, y la combinación de funciones de activación ReLU-ReLU predicen las isothermas experimentales con un error medio de porcentaje absoluto de 1.81% para temperaturas entre 267°C a 300°C. Este enfoque permite obtener una mejor representación del comportamiento del ELV de la mezcla $\text{CO}_2\text{-H}_2\text{O}$, comparado con las predicciones de la EOS-PR sin corrección.

Palabras clave:

Equilibrio líquido-vapor, $\text{CO}_2\text{-H}_2\text{O}$, ecuación de estado de Peng-Robinson, red neuronal híbrida, optimización estocástica.

Abstract

In this work, the liquid-vapor equilibrium (VLE) of the CO₂-H₂O mixture was studied using the Peng-Robinson equation of state (EOS-PR). It was found that EOS-PR did not adequately fit the experimental behavior, especially in the dew curve. To improve predictions, a hybrid model using neural networks was developed. An algorithm was created to calculate the “a” mixing parameter in the liquid phase of the EOS-PR and the binary interaction parameter “s_{ij}” from the Van der Waals mixing rule, using experimental mole fractions of CO₂ in the liquid and vapor phases, pressures, and temperatures for each isotherm. The simulated annealing stochastic optimization method was applied over a temperature range between 267°C and 350°C.

From this analysis, two neural networks with architecture [2, 20, 10, 1] were implemented: one to predict the binary interaction parameter “s_{ij},” and another to adjust the “a” mixing parameter in the liquid phase by correcting the value determined by the Van der Waals mixing rule. The models were trained using parametrized experimental data collected by Tödheide & Franck (1963), and data augmentation was performed using parametric splines. The results show that the combination of ReLU-TanH activation functions predicted the experimental isotherms with a mean absolute percentage error of 2.09% for temperatures between 267°C and 350°C, and the combination of ReLU-ReLU activation functions predicted the experimental isotherms with a mean absolute percentage error of 1.81% for temperatures between 267°C and 300°C. This approach provides a better representation of the VLE behavior of the CO₂-H₂O mixture compared to EOS-PR predictions without corrections.

Keywords:

Vapor-liquid equilibrium, CO₂-H₂O, Peng-Robinson equation of state, hybrid neural network, stochastic optimization.

Tabla de Contenido

Pág.	
Resumen	v
Abstract	vii
Introducción	xvi
Capítulo I. Parte introductoria del trabajo	1
1.1 Generalidades	1
1.2 Descripción del problema de investigación	2
1.3 Objetivos del estudio	3
1.1.1 Objetivo general	3
1.1.1 Objetivos específicos	3
1.4 Antecedentes investigativos	3
Capítulo II. Marcos teórico y conceptual	7
2.1 Generalidades	7
2.2 Marco teórico	7
2.2.1 Equilibrio de fases líquido-vapor	7
2.2.2 Coeficiente de fugacidad de un componente en una mezcla	8
2.2.3 Propiedades termodinámicas de CO ₂ y H ₂ O como componentes puros.	10
2.2.4 Diagrama P _{xy}	10
2.2.5 Diagrama T _{xy}	11
2.2.6 Coeficiente de distribución	12
2.2.7 Ecuaciones de estado	13
2.2.8 Método ϕ - ϕ (fi-fi)	21
2.2.9 Algoritmo clásico para la solución del ELV con el método ϕ - ϕ (fi-fi)	22
2.3 Marco conceptual	23
2.3.1 Redes neuronales	23

2.3.2	Estructura de una red neuronal.....	24
2.3.3	Modelamiento híbrido en procesos industriales	27
2.3.4	Funciones de activación	29
2.3.5	Modelamiento del equilibrio líquido-vapor usando redes neuronales como predictor de parámetros de interacción binaria	34
2.3.6	Métodos computacionales.....	35
2.3.7	Splines paramétricos	37
Capítulo III. Desarrollo del trabajo de investigación		39
3.1	Métodos	39
3.1.1	Obtención de data experimental	39
3.1.2	Parametrización de data experimental	40
3.1.3	Determinación de parámetros “a” en fase líquida y “s _{ij} ” de la EOS – PR	40
3.1.4	Definición de red neuronal para modelamiento del equilibrio líquido – vapor	52
3.1.5	Entrenamiento de red neuronal.....	53
3.1.6	Método de validación.....	54
3.2	Resultados.....	55
3.2.1	Obtención de data experimental	55
3.2.2	Parametrización de data experimental	56
3.2.3	Solución del ELV usando PR-EOS Original	68
3.2.4	Determinación de parámetros “a” y “s _{ij} ” en la EOS – PR	74
3.2.5	Definición de red neuronal para modelamiento del equilibrio líquido-vapor	92
3.2.6	Entrenamiento de la red neuronal.....	94
3.2.7	Validación	98
Capítulo IV. Análisis y discusión de resultados		111
Conclusiones		114
5.1	Conclusión general.....	114
5.1.1	Conclusión Específica 1	114

5.1.2	Conclusión Específica 2	114
5.1.3	Conclusión Específica 3	115
	Recomendaciones	117
	Próximas Investigaciones.....	118
	Referencias bibliográficas	120
	Anexos	132

Lista de Tablas

	Pag
Tabla 1	Propiedades de las especies puras en la mezcla.10
Tabla 2	Data experimental del ELV de CO ₂ – H ₂ O55
Tabla 3	Data experimental del ELV de CO ₂ – H ₂ O a 267°C56
Tabla 4	Data experimental del ELV de CO ₂ – H ₂ O a 268°C58
Tabla 5	Data experimental del ELV de CO ₂ – H ₂ O a 270°C60
Tabla 6	Data experimental del ELV de CO ₂ – H ₂ O a 275°C62
Tabla 7	Data experimental del ELV de CO ₂ – H ₂ O a 300°C64
Tabla 8	Data experimental del ELV de CO ₂ – H ₂ O a 350°C66
Tabla 9	Errores MSE y MAPE para cada estructura neuronal.....106

Lista de Figuras

	Pag
Figura 1 Diagrama P_{xy} para la mezcla R-134a y R-245fa a 313.15K.....	11
Figura 2 Diagrama T_{xy} para la mezcla R-134a y R-245fa a 101.3kPa.	12
Figura 3 Algoritmo para solución del ELV (Diagrama $P_{xy} - T$ constante) utilizando el método fi-fi.....	23
Figura 4 Estructura de una red neuronal	24
Figura 5 Estructura de una red neuronal.	26
Figura 6 Red neuronal 2-12-9-2	27
Figura 7 Red Neuronal Híbrida en Serie Tipo I.....	28
Figura 8 Red Neuronal Híbrida en Serie Tipo II.....	28
Figura 9 Red Neuronal Híbrida en Paralelo	29
Figura 10 Función de activación ReLU	30
Figura 11 Función de activación Leaky ReLU con $\alpha=0.5$	31
Figura 12 Función de activación Sigmoide	32
Figura 13 Función de activación Tangente hiperbólica	33
Figura 14 Función de activación Softmax	34
Figura 15 Diagrama de flujo de procedimiento de aplicación de redes neuronales al ELV	35
Figura 16 Diagrama de bloques del procedimiento para el presente trabajo de investigación.....	39
Figura 17 Nuevo algoritmo para encontrar los parámetros de interacción binaria y el parámetro “a” de mezcla en fase líquida	51
Figura 18 Isoterma suavizada a 267°C.....	57
Figura 19 Datos aumentados a 267°C.....	58
Figura 20 Isoterma suavizada a 268°C.....	59
Figura 21 Datos aumentados a 268°C.....	60

Figura 22	Isoterma suavizada a 270°C.....	61
Figura 23	Datos aumentados a 270°C.....	62
Figura 24	Isoterma suavizada a 275°C.....	63
Figura 25	Datos aumentados a 275°C.....	64
Figura 26	Isoterma suavizada a 300°C.....	65
Figura 27	Datos aumentados a 300°C.....	66
Figura 28	Isoterma suavizada a 350°C.....	67
Figura 29	Datos aumentados a 350°C.....	68
Figura 30	Comparación de EOS-PR con data experimental a 267°C y $k_{ij} = 0.05$	70
Figura 31	Comparación de EOS-PR con data experimental a 267°C y $k_{ij} = 0.058$	71
Figura 32	Comparación de EOS-PR con data experimental a 268°C y $k_{ij} = 0.055$	72
Figura 33	Comparación de EOS-PR con data experimental a 270°C y $k_{ij} = 0.055$	72
Figura 34	Comparación de EOS-PR con data experimental a 275°C y $k_{ij} = 0.054$	73
Figura 35	Comparación de EOS-PR con data experimental a 300°C y $k_{ij} = 0.058$	73
Figura 36	Comparación de EOS-PR con data experimental a 350°C y $k_{ij} = 0.058$	74
Figura 37	Comportamiento de los parámetros de la EOS-PR a 270°C	80
Figura 38	Evaluación de las soluciones del ELV usando los parámetros encontrados mediante optimización estocástica a $T = 270^{\circ}\text{C}$	81
Figura 39	Comportamiento de los parámetros de la EOS-PR a 267°C	82
Figura 40	Comportamiento de los parámetros de la EOS-PR a 268°C	82
Figura 41	Comportamiento de los parámetros de la EOS-PR a 275°C	83
Figura 42	Comportamiento de los parámetros de la EOS-PR a 300°C	83
Figura 43	Comportamiento de los parámetros de la EOS-PR a 350°C	84
Figura 44	Variación entre a_m^I calculado por el nuevo algoritmo y el a_m^I calculado mediante la regla de mezcla de Van der Waals a $T = 270^{\circ}\text{C}$	85
Figura 45	Variación entre a_m^I calculado por el nuevo algoritmo y el a_m^I calculado mediante la regla de mezcla de Van der Waals a $T = 267^{\circ}\text{C}$	86

Figura 46	Variación entre a_m^I calculado por el nuevo algoritmo y el a_m^I calculado mediante la regla de mezcla de Van der Waals a $T = 268^\circ\text{C}$	87
Figura 47	Variación entre a_m^I calculado por el nuevo algoritmo y el a_m^I calculado mediante la regla de mezcla de Van der Waals a $T = 275^\circ\text{C}$	87
Figura 48	Variación entre a_m^I calculado por el nuevo algoritmo y el a_m^I calculado mediante la regla de mezcla de Van der Waals a $T = 300^\circ\text{C}$	88
Figura 49	Variación entre a_m^I calculado por el nuevo algoritmo y el a_m^I calculado mediante la regla de mezcla de Van der Waals a $T = 350^\circ\text{C}$	89
Figura 50	Comparación del ELV usando los parámetros calculados y datos experimentales a 267°C	90
Figura 51	Comparación del ELV usando los parámetros calculados y datos experimentales a 268°C	90
Figura 52	Comparación del ELV usando los parámetros calculados y datos experimentales a 275°C	91
Figura 53	Comparación del ELV usando los parámetros calculados y datos experimentales a 300°C	91
Figura 54	Comparación del ELV usando los parámetros calculados y datos experimentales a 350°C	92
Figura 55	Red neuronal [2, 20, 10, 1]	93
Figura 56	Arquitectura híbrida para la solución del ELV de la mezcla $\text{CO}_2\text{-H}_2\text{O}$	94
Figura 57	Gráfico de pérdida en entrenamiento y validación para predecir s_{ij} (ReLU)....	96
Figura 58	Gráfico de pérdida en entrenamiento y validación para predecir Δa_m^I (ReLU)	96
Figura 59	Gráfico de pérdida en entrenamiento y validación para predecir s_{ij} (TanH)	97
Figura 60	Gráfico de pérdida en entrenamiento y validación para predecir Δa_m^I (TanH).....	98
Figura 61	Comportamiento de s_{ij} a 269°C usando ReLU como función de activación en capas ocultas	99
Figura 62	Comportamiento de Δa_m^I a 269°C usando ReLU como función de activación en capas ocultas	100
Figura 63	Diagrama P_{xy} usando s_{ij} y Δa_m^I predichos a 269°C	101
Figura 64	Comportamiento de s_{ij} a 269°C usando Tangente hiperbólica como función de activación en capas ocultas	102
Figura 65	Comportamiento de Δa_m^I a 269°C usando ReLU como función de activación en capas ocultas	103

Figura 66	Diagrama P_{xy} usando s_{ij} y Δa_m^I predichos a 269°C.....	104
Figura 67	Diagrama P_{xy} usando s_{ij} y Δa_m^I predichos a 269°C.....	105
Figura 68	Diagrama P_{xy} usando s_{ij} y Δa_m^I predichos a 269°C.....	106
Figura 69	Comparación de los diagramas P_{xy} obtenidos de las diferentes combinaciones neuronales.....	107

Introducción

El concepto de operaciones unitarias establecido por Arthur D. Little en 1916 (Van Antwerpen, 1980), comprende a diferentes métodos de separación de mezclas, y representó un gran avance para la Ingeniería Química. A medida que se categorizaban las operaciones unitarias surgió la necesidad de un nuevo campo especializado en procesos de separación, en donde una de las principales ramas son análisis computacionales aplicados a problemas de equilibrio de fases (King, 2000).

La mezcla de dióxido de carbono (CO_2) y agua (H_2O) es de particular interés en aplicaciones industriales y ambientales, como la captura y almacenamiento de CO_2 (Pappa et al., 2009). Sin embargo, el modelamiento del comportamiento termodinámico de esta mezcla presenta desafíos, ya que las ecuaciones de estado tradicionales, como la ecuación de estado de Peng-Robinson (EOS-PR), no logran ajustarse adecuadamente a los datos experimentales en ciertos rangos de temperatura y presión, especialmente en la predicción de la curva de rocío (Paulus & Penoncello, 2006).

La ecuación de Peng-Robinson ampliamente utilizada por su simplicidad y aplicabilidad en una gran variedad de mezclas, presenta limitaciones cuando se trata de sistemas altamente polarizados o no ideales, como el sistema $\text{CO}_2\text{-H}_2\text{O}$ (Forero & Velásquez, 2016).

Algunas alternativas a las EOS tradicionales han sido utilizadas para predecir el ELV de $\text{CO}_2 - \text{H}_2\text{O}$, usualmente se utilizan EOS basadas en interacciones moleculares, las más conocidas son las ecuaciones de estado cubicas más asociación (Monteiro et al., 2020) o ecuaciones de estado basadas en la teoría de fluidos auto asociativos de cadenas perturbadas (Diamantonis & Economou, 2012).

Con el fin de mejorar la precisión de la EOS-PR para este sistema, se ha recurrido a técnicas avanzadas de modelización, como el uso de redes neuronales híbridas, que combinan la robustez de los modelos empíricos con la capacidad de aprendizaje automático de las redes neuronales (Psichogios & Ungar, 1992).

El uso de las redes neuronales ha sido ampliamente usado en la determinación del ELV. Vyhmeister et al. (2017) utilizó redes neuronales para predecir parámetros de interacción binaria del ELV de las mezclas Acetona-Butano, Acetona-Etanol y Acetona-Metanol. También Si-Moussa et al. (2008) utilizó redes neuronales para predecir el ELV de CO₂ con esteres.

Capítulo I. Parte introductoria del trabajo

1.1 Generalidades

El equilibrio líquido-vapor (ELV) de la mezcla CO_2 - H_2O es de gran interés industrial debido a su fuerte relación con la industria petrolera, por ejemplo, en la remoción de gases ácidos como el CO_2 y H_2S en el gas natural (Valtz et al., 2004). Además, la mezcla CO_2 - H_2O es fundamental en tecnologías emergentes como la captura y almacenamiento de carbono, el cual busca reducir la cantidad de dióxido de carbono en la atmósfera (*What is carbon sequestration?* | U.S. Geological Survey, s/f), por lo que la predicción precisa del comportamiento termodinámico de esta mezcla es esencial para optimizar el diseño y operación de equipos de separación y compresión.

Es así como se utilizan las Ecuaciones de estado (EOS) para poder modelar el ELV de la mezcla, sin embargo, las EOS convencionales tienen limitaciones y no pueden llegar a modelar mezclas a temperaturas supercríticas para uno de sus componentes. Una solución ante esta limitación es la utilización de EOS cúbicas basadas en las interacciones moleculares entre el CO_2 y H_2O , un ejemplo de ellas son las ecuaciones de estado cúbicas más asociación, basadas en la teoría de perturbaciones de Wertheim (Diamantonis & Economou, 2012).

La EOS de Peng - Robinson (EOS - PR) nace como un proyecto para mejorar la capacidad predictiva de la EOS de Soave-Redlich-Kwong para condiciones cercanas al punto crítico (Lopez-Echeverry et al., 2017). Por otro lado, la EOS-PR tiene limitaciones, una de ellas se presenta cuando tenemos agua en la mezcla ya que en la base de datos original que tomó Peng – Robinson no consideraron al agua (Peng & Robinson, 1976), otra limitación es que se recomienda utilizar la EOS – PR cuando se tenga componentes no polares o débilmente polares, ya que la EOS – PR no fue diseñada originalmente para componentes polares (Varbanov et al., 2020).

En los últimos años se ha implementado la inteligencia artificial (IA) y en específico técnicas de Aprendizaje profundo como Redes Neuronales (RN) para poder predecir

propiedades basadas en data. Se han encontrado aplicaciones en diversos campos como ingeniería, bioquímica, simulaciones físicas, optimización energética, entre otros (J.-Q. Wang & Taherdoost, 2023). Existen algunas investigaciones que logran implementar satisfactoriamente RN para predecir ELV u otros fenómenos relevantes para la industria (Vyhmeister et al., 2017), pero la mayoría de ellas ignora la parte teórica del fenómeno que están estudiando y terminan desarrollando un modelo predictivo basado en datos que no puede ser utilizado para determinar propiedades derivadas o hacer un análisis más profundo de las predicciones encontradas (Jirasek & Hasse, 2023).

Una herramienta muy utilizada en el desarrollo de modelos de aprendizaje profundo y en específico aprendizaje de maquina es Python, ya que ofrece una experiencia de “conectar y usar”, y además su extensa cantidad de librerías facilitan la implementación de modelos de aprendizaje automático para cualquier persona con un poco de nociones básicas de programación (Gao et al., 2020).

1.2 Descripción del problema de investigación

Las EOS tradicionales, como la EOS – PR, han sido fundamentales en la predicción del ELV para una variedad de sistemas. Sin embargo, estas ecuaciones presentan limitaciones significativas cuando se aplican a la mezcla $\text{CO}_2\text{-H}_2\text{O}$ debido a las características únicas de este sistema. La fuerte polaridad del agua contrasta con la no polaridad del CO_2 , lo que genera interacciones moleculares complejas que las EOS tradicionales no logran modelar adecuadamente (Varbanov et al., 2020). Esto resulta en predicciones imprecisas del ELV con una desviación de hasta en un 90% en la predicción del CO_2 en la curva de rocío en un rango de temperaturas entre 290K a 340K y 70 bar a 120 bar. Y con desviaciones, observables gráficamente en un el diagrama de presión versus composiciones molares, aún mayores a temperaturas mayores a 473K (Chiew et al., 2020).

La limitación de las EOS tradiciones para predecir el comportamiento de la mezcla $\text{CO}_2 - \text{H}_2\text{O}$ compromete la utilidad de estas ecuaciones en aplicaciones industriales críticas, como la optimización de inyección y almacenamiento de CO_2 en formaciones

geológicas. La captura y almacenamiento de CO_2 es una estrategia efectiva y novedosa para reducir las emisiones de carbono en la atmósfera, se estima que se capturan 45Mt CO_2 por año (*How carbon capture and storage can help us tackle the climate crisis* | *World Economic Forum*, s/f), por lo que es de suma relevancia determinar las interacciones complejas entre el CO_2 y el H_2O para una viabilidad técnica y económica (Mohshim et al., 2024).

Basándose en lo expuesto, este estudio se centra en la siguiente pregunta como su problema principal: ¿Podrá un modelo neuronal híbrido basado en la ecuación de estado de Peng-Robinson predecir el equilibrio líquido-vapor de la mezcla binaria $\text{CO}_2 - \text{H}_2\text{O}$, y cómo se compararán sus predicciones con los datos experimentales?

1.3 Objetivos del estudio

1.1.1 Objetivo general

- Modelar la ecuación de estado de Peng-Robinson para predecir el equilibrio líquido-vapor de la mezcla $\text{CO}_2 - \text{H}_2\text{O}$ mediante la implementación de un modelo híbrido neuronal con una precisión mayor al 97%.

1.1.1 Objetivos específicos

- Obtención de data experimental parametrizada del equilibrio líquido-vapor de la mezcla $\text{CO}_2 - \text{H}_2\text{O}$.
- Definición de red neuronal para modelamiento de equilibrio líquido-vapor.
- Entrenamiento y validación del modelo híbrido neuronal.

1.4 Antecedentes investigativos

El modelamiento ELV para la mezcla $\text{CO}_2 - \text{H}_2\text{O}$ ha sido estudiado a lo largo de la historia debido a su comportamiento no ideal por la fuerte polaridad del agua comparada con la no polaridad del CO_2 (Varbanov et al., 2020). Es por ello que muchos expertos en el tema han tratado de encontrar una EOS, un ejemplo de ello es el trabajo realizado por Gallagher et al. (1993), en donde estudiaron el equilibrio para condiciones de temperatura

entre 400K y 1000K, presiones hasta 100MPa, y fracciones molares de hasta 30% de CO₂. Se basaron en la energía libre de Helmholtz siguiendo el principio de estados correspondientes generalizados. Gallagher et al. (1993) no realizaron experimentos, solo se enfocaron en desarrollar la parte matemática y termodinámica del ELV y compararon sus resultados con los obtenidos por Tödheide & Franck (1963) para las curvas de equilibrio y Greenwood, (1973) para otras propiedades termodinámicas. Los resultados del modelo mostraron un error de 10% por encima de los datos experimentales de la curva de puntos obtenida del diagrama de presión versus fracción molar del CO₂ con respecto a los resultados obtenidos por Tödheide & Franck (1963)

Vyhmeister et al. (2017) implementó redes neuronales híbridas para predecir los parámetros de interacción binaria en la EOS - PR con la finalidad de generalizar la EOS a mezclas no consideradas en el entrenamiento de su red neuronal. Se entrenó la red neuronal con las mezclas de Acetona-Butanol, Acetona-Etanol y Acetona-Metanol a diferentes temperaturas para predecir el diagrama de presión versus composición molar de acetona. Usaron las propiedades termodinámicas de los componentes puros (Punto de ebullición, masa molar, factor acéntrico, temperatura crítica y presión crítica) y propiedades del sistema como presión y temperatura como datos de entrada, utilizaron 10 neuronas en la capa oculta y una neurona en la capa de salida para predecir la fracción molar en fase vapor de acetona. Se obtuvieron desviaciones menores al 5% al comparar el modelo obtenido con la data experimental de Amer et al. (2002)

Vaferi et al. (2018) desarrolló una red neuronal para predecir los puntos de burbuja y rocío de sistemas binarios CO₂ – Refrigerantes en un rango de temperaturas entre 263.15 K a 367.30 K y presiones entre 0.18MPa a 9.09MPa. Mediante “prueba y error” determinaron que la arquitectura óptima de la red neuronal consiste en una red neuronal densa de tres capas; En la capa de entrada se consideraron: temperaturas reducidas, presiones críticas, factores acéntricos, fracción molar del CO₂ en fase líquida y vapor; En la capa oculta usaron 14 neuronas y en la capa de salida obtuvieron la presión. Luego validaron sus predicciones con diferentes bases de data experimental de ELV de sistemas

binarios CO₂ – Refrigerantes obteniendo un error cuadrático medio máximo entre todas las mezclas analizadas de 0.0641 para la mezcla binaria CO₂ – HFC-134a.

Monteiro et al. (2020) se basaron en la ecuación de estado cubica más asociación para predecir el ELV de la mezcla CO₂ – H₂O y sus propiedades volumétricas. Ellos no realizaron experimentos, sino que utilizaron 544 puntos experimentales obtenidos de literatura. Monteiro et al. (2020) ajustaron el modelo para optimizar, simultáneamente, el ELV y las propiedades volumétricas de la mezcla. Los resultados del ajuste mostraron desviaciones de 0.49% para la fracción molar de CO₂ en fase líquida y 4.63% para la fracción molar de CO₂ en fase vapor a condiciones de temperaturas entre 273K a 647K y presiones entre 0 a 123 bar.

Otra investigación donde implementaron redes neuronales para predecir el ELV de una mezcla binaria fue la realizada por Aminian & ZareNezhad (2020), los cuales desarrollaron una red neuronal para predecir el comportamiento de fases de CO₂ supercrítico y ácidos grasos. Recolectaron 678 puntos del equilibrio líquido – vapor de ácidos grasos y esteres de metilo y etilo. Luego diseñaron una red neuronal donde utilizaron como datos de entrada: Las temperaturas críticas, presiones críticas y factor acéntrico de los ácidos grasos, la presión y temperatura del sistema. 16 neuronas en la primera capa oculta, 10 neuronas en la segunda capa oculta y 6 neuronas en la tercera capa oculta. Las predicciones las compararon con los datos experimentales obtenidos de literatura científica de diferentes autores obteniendo desviaciones máximas de 5.54% para la fracción molar líquida de CO₂ para el ELV de CO₂ – Ácido Linoleico, y 0.70% en la fracción molar de CO₂ en fase vapor para el ELV de CO₂ – Acetato de metilo.

Petroli et al. (2022) modeló el ELV de los sistemas ternario CO₂ – Biodisel – Etanol y CO₂ – Biodisel – Metanol usando redes neuronales en el siguiente rango de operación: Temperaturas entre 303.15 K a 343.15 K, presiones entre 4.30 MPa a 15.62 MPa, fracciones molares de CO₂ entre 0.4 a 0.99. Usaron la data experimental obtenida por Pinto et al. (2012) y Araújo et al. (2012) para entrenar la red neuronal, finalmente obtuvieron un

error cuadrático medio de 0.13 para la mezcla ternaria CO_2 – Biodisel – Etanol y un error cuadrático medio de 0.16 para la mezcla ternaria CO_2 – Biodisel – Metanol.

Luego existen algunas investigaciones que implementaron redes neuronales para predecir el ELV de una mezcla binaria, tenemos a Pizzano et al. (2023) en donde trataron de predecir la fracción molar de CO_2 en fase líquida de una mezcla de polidimetilsiloxano y CO_2 mediante redes neuronales artificiales con propagación hacia adelante, usando la temperatura y presión como datos de entrada y la composición de CO_2 en fase líquida como dato de salida, la arquitectura que utilizaron fue de dos neuronas en la capa de entrada, treinta neuronas en la capa oculta y una neurona en la capa de salida llegando a obtener errores cuadráticos medios de 0.02 a 30°C, 0.01% a 40°C y 0.026% a 50°C superando así la EOS de Sanchez & Lacombe (1976), la cual se usa comúnmente para predecir el equilibrio de fases en sistemas poliméricos.

Capítulo II. Marcos teórico y conceptual

2.1 Generalidades

Los modelos híbridos, que combinan redes neuronales con ecuaciones fenomenológicas como la de Peng-Robinson (PR-EOS), ofrecen una ventaja significativa. Estos modelos integran la capacidad predictiva de las redes neuronales con el rigor teórico de las ecuaciones de estado, permitiendo una mejor generalización y extrapolación de resultados. Esto no solo mejora la precisión en la predicción del equilibrio líquido-vapor, sino que también permite la estimación de otras propiedades termodinámicas relacionadas, abordando así las limitaciones inherentes a los modelos puramente neuronales (*Predicting Vapor-Liquid Equilibria Using Physic-Inspired Machine Learning Models* | AIChE, s/f).

2.2 Marco teórico

2.2.1 Equilibrio de fases líquido-vapor

El equilibrio es una condición en la cual las propiedades macroscópicas de un sistema (aislado) permanecen constantes a lo largo del tiempo y espacio en donde todos los potenciales que podrían originar cambios están perfectamente equilibrados, entonces no habrá una fuerza impulsora que motive cambios dentro del sistema (Smith et al., 2022). En el equilibrio líquido-vapor las tasas de vaporización y condensación son iguales. (Koretsky, 2013).

Para que un sistema multicomponente se encuentre en equilibrio de fases líquido vapor se deben cumplir las siguientes condiciones:

- La temperatura en la fase líquida y el vapor deben ser iguales ($T_{liq} = T_{vap}$)
- La presión en la fase líquida y el vapor deben ser iguales ($P_{liq} = P_{vap}$)

- La fugacidad de todos los componentes en la fase líquida y el vapor deben ser iguales ($\widehat{f}_{liq} = \widehat{f}_{vap}$) (Mokhatab et al., 2019)

2.2.2 Coeficiente de fugacidad de un componente en una mezcla

Para un fluido ideal podemos definir la energía molar libre de Gibbs para un gas ideal a una temperatura constante tal como se observa en la Ecuación 1.

Integrando la Ecuación 1 podemos obtener la Ecuación 2.

Para un fluido real podemos tomar la forma de la Ecuación 2 y formular la Ecuación 3, en donde se presenta una nueva propiedad " f_i " que será llamada fugacidad de la especie pura " i ".

Si analizamos el caso en el que la energía molar libre de Gibbs es ideal obtenemos la Ecuación 4.

$$dG_i^{ig} = V_i^{ig} dP = \frac{RT}{P} dP = RT d(\ln P); T \text{ constante} \quad (1)$$

$$G_i^{ig} = \Gamma_{i(T)} + RT \ln(P) \quad (2)$$

$$G_i = \Gamma_{i(T)} + RT \ln(f_i) \quad (3)$$

$$f_i^{ig} = P \quad (4)$$

Donde:

G_i^{ig} : Energía molar libre de Gibbs ideal del componente i .

G_i : Energía molar libre de Gibbs real del componente i .

V_i^{ig} : Volumen molar ideal del componente i .

f_i : Fugacidad del componente i .

f_i^{ig} : Fugacidad ideal del componente i .

Luego podemos sustraer la energía molar libre de Gibbs ideal y la energía molar libre de Gibbs real para obtener su propiedad residual según la Ecuación 5, obteniendo la Ecuación 6.

$$G_i - G_i^{ig} = G_i^R \quad (5)$$

$$G_i^R = RT \ln \left(\frac{f_i}{P} \right) = RT \ln(\phi_i) \quad (6)$$

Como vemos en la Ecuación 6 podemos definir a la relación f_i/P como una nueva propiedad, esta propiedad se llamará coeficiente de fugacidad " Φ_i ", como lo podemos ver en la Ecuación 7.

$$\frac{f_i}{P} = \phi_i \quad (7)$$

Donde:

Φ_i : Coeficiente de fugacidad del componente "i"

Además, podemos calcular el coeficiente de fugacidad con la Ecuación 8.

$$\ln(\phi_i) = \int_0^P (Z_i - 1) \frac{dP}{P}; T \text{ constante} \quad (8)$$

Luego podríamos realizar un análisis similar, considerando a $\hat{f}_i$ como la fugacidad del componente en la mezcla y a y_i como la fracción molar del componente en la mezcla, así obtendríamos las Ecuaciones 9 y 10

$$\hat{\phi}_i = \frac{\hat{f}_i}{y_i P} \quad (9)$$

$$\overline{G}_i^R = RT \ln \left(\frac{\hat{f}_i}{y_i P} \right) = RT \ln(\hat{\phi}_i) \quad (10)$$

Donde:

$\hat{\phi}_i$: Coeficiente de fugacidad del componente “i” en una mezcla

$\overline{G}_i^R$: Energía libre residual parcial molar de Gibbs del componente “i”

2.2.3 Propiedades termodinámicas de CO₂ y H₂O como componentes puros.

Las propiedades termodinámicas que se utilizan en la EOS – PR son el factor acéntrico (w), temperatura crítica (T_c) y presión crítica (P_c).

Podemos ver las propiedades en la Tabla 1, estas fueron obtenidas de Smith et al. (2022).

Tabla 1

Propiedades de las especies puras en la mezcla.

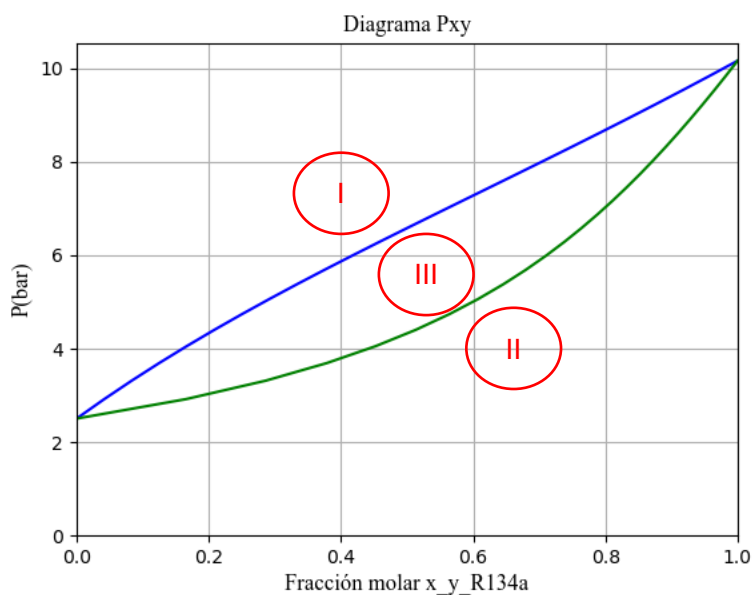
Propiedades de especies puras			
Especie	w	T _c (K)	P _c (bar)
CO ₂	0.224	304.2	73.83
H ₂ O	0.345	647.1	220.55

2.2.4 Diagrama P_{xy}

Los diagramas de P_{xy} son la representación gráfica del comportamiento de una mezcla a diferentes presiones manteniendo la temperatura constante. En la Figura 1, tenemos el diagrama P_{xy} para la mezcla R134a (1) y R245fa (2). La gráfica nos ayuda a visualizar dos líneas importantes, la línea azul es la línea de los puntos de burbuja (líquido saturado) y la línea verde es la línea de los puntos de rocío (vapor saturado) (Smith et al., 2022). En el eje X de la Figura 1 tenemos la fracción molar del componente R134a (x₁, Solo es necesario un componente, ya que la fracción molar del otro componente se puede calcular mediante la ecuación x₁+x₂=1), mientras que en el eje Y tenemos la presión.

Figura 1

Diagrama P_{xy} para la mezcla R-134a y R-245fa a 313.15K.



Nota. Gráfica propia con data obtenida del software REFPROP 9.0

Asimismo, podemos identificar algunas áreas clave en la Figura 1:

- Área I – Líquido subenfriado o Líquido: En esta área el líquido está en presiones por encima de su punto de burbuja.
- Área II – Vapor sobrecalentado o Vapor: En esta área el vapor está en presiones por debajo de su punto de rocío.
- Área III – Mezcla Líquido Vapor: En esta área encontraremos las dos fases presentes en la mezcla Líquido y Vapor.

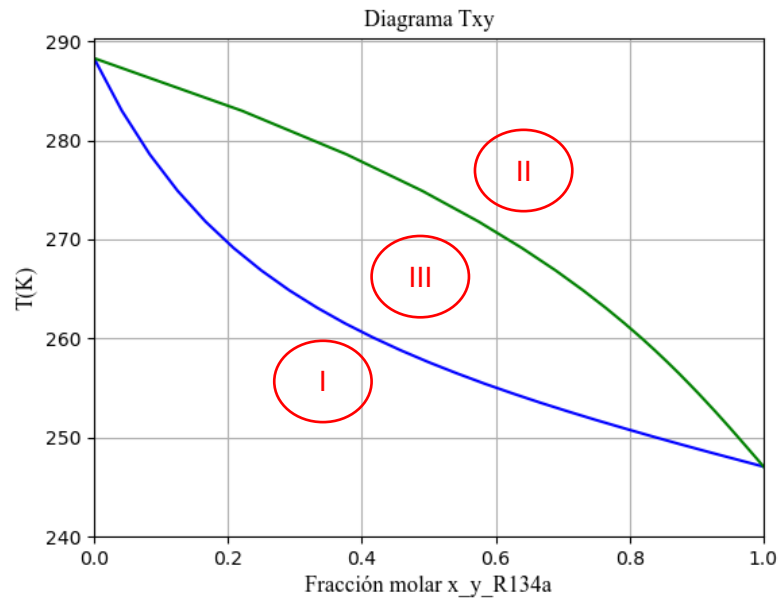
2.2.5 Diagrama T_{xy}

Similar a los diagramas P_{xy} , los diagramas de T_{xy} son la representación gráfica del comportamiento de una mezcla a diferentes temperaturas manteniendo la presión constante. En la Figura 2, tenemos el diagrama T_{xy} para la mezcla R134a (1) y R245fa (2). La gráfica nos ayuda a visualizar dos líneas importantes, la línea azul es la línea de los puntos de burbuja (líquido saturado) y la línea verde es la línea de los puntos de rocío (vapor saturado) (Smith et al., 2022). En el eje X de la Figura 2 tenemos la fracción molar

del componente R134a (x_1 , Solo es necesario un componente, ya que la fracción molar del otro componente se puede calcular mediante la ecuación $x_1+x_2=1$), mientras que en el eje Y tenemos la temperatura.

Figura 2

Diagrama T_{xy} para la mezcla R-134a y R-245fa a 101.3kPa.



Nota. Creación gráfica propia con data obtenida del software REFPROP 9.0

Asimismo, podemos identificar algunas áreas clave en la Figura 2:

- Área I – Líquido subenfriado o Líquido: En esta área el líquido está en temperaturas por debajo de su punto de burbuja.
- Área II – Vapor sobrecalentado o Vapor: En esta área el vapor está en temperaturas por encima de su punto de rocío.
- Área III – Mezcla Líquido Vapor: En esta área encontraremos las dos fases presentes en la mezcla Líquido y Vapor.

2.2.6 Coeficiente de distribución

Para poder describir la razón entre la fracción molar de un componente en una fase con respecto a otra podemos definir el “Coeficiente de distribución” o el “Factor K”, el cual

se define como la razón entre la fracción molar en la fase vapor entre la fracción molar en la fase líquida tal como se ve en la Ecuación 11.

$$K_i = \frac{y_i}{x_i} \quad (11)$$

Donde:

K_i : Coeficiente de distribución

2.2.7 Ecuaciones de estado

Las EOS tratan de modelar el comportamiento de fluidos reales. Las EOS son expresiones analíticas que relacionan las variables de estado de un sistema como la Presión, Temperatura y Volumen (Ahmed, 2016). Podemos crear EOS al resolver ecuaciones de conservación de la energía (Marmur, 1985), también podemos usar una EOS ya desarrollada y modificarla u optimizarla para componentes o mezclas específicas e incluso para condiciones fuera del límite de la EOS.

2.2.7.1 Clasificación de las ecuaciones de estado

La relación más simple entre las propiedades de estado de un sistema nos lleva a un tipo de ecuación llamada ley de gases ideales (Reinert & Kähler, 2010). Por otro lado tenemos EOS más complejas que según Carnahan & Starling (1972) se pueden clasificar en tres tipos:

- Ecuaciones cúbicas: Son EOS que son cúbicas con respecto al Volumen. Este tipo de ecuaciones ha demostrado ser capaces de modelar sistemas termodinámicos en un amplio rango de Presión y Temperatura (Smith et al., 2022).
- Ecuaciones del virial: Originalmente formulada en su forma general por H. Kamerlingh Onnes como una serie infinita utilizando potencias negativas de la densidad en su formulación (Wisniak, 2012), y luego como serie truncada desarrollada por Thiesen (1885).

- Ecuaciones basadas en simulación molecular: Son las ecuaciones derivadas de funciones de potencial. Por ejemplo, Carnahan & Starling (1972) desarrollaron una EOS para describir el comportamiento de esferas rígidas, un dónde se considera a las partículas como esferas rígidas que no pueden superponerse, ellos sustituyeron el termino de repulsión intermolecular en la ecuación de Van der Waals y de Soave-Riedlich-Kwong por una ecuación virial.

2.2.7.2 Ley de Gases Ideales y su modificación para gases reales

Boyle, en 1662, observó que el volumen de un gas disminuye a medida que se incrementa la presión, siempre que la temperatura se mantenga constante. Más adelante, Charles descubrió que el volumen de un gas aumenta proporcionalmente al incrementarse la temperatura, manteniendo la presión constante. En 1834, Clapeyron unificó estos principios en lo que conocemos como la Ley de los Gases Ideales, tal como se observa en Ecuación 12 (Mansour & Mansour, 2020).

$$PV = RT \quad (12)$$

Donde:

P: Presión absoluta, psia

V: Volumen molar, pies³/mol

R: Constante de gases ideales, 10.73159 pies³ R⁻¹ lb-mol⁻¹

T: Temperatura absoluta, Rankine (R)

La ley de los gases ideales funciona muy bien para presiones bajas, pero los compuestos o mezclas a presiones elevadas pueden llegar a desviarse hasta en un 500% del comportamiento ideal de un gas. Entonces surgió la necesidad de una modificación de la ley de los gases ideales para poder predecir el comportamiento de los gases reales. Se utiliza el factor de compresibilidad (Z) para corregir esta desviación, obteniendo la Ecuación 13 (Carroll, 2020).

$$PV = ZRT \quad (13)$$

Donde:

Z: Factor de compresibilidad

2.2.7.3 Ecuación de estado de Van der Waals

Van der der Waals propuso la primera ecuación cubica de estado (Smith et al., 2022).

Se baso en el concepto de que la presión del sistema estaba compuesta por una presión de atracción y otra de repulsión, tal como se ve en la Ecuación 14.

$$P = P^{rep} + P^{atr} \quad (14)$$

Donde:

P^{rep} : Presión de repulsión.

P^{atr} : Presión de atracción.

Van der Waals propuso las Ecuaciones 15 y 16 para las presiones de repulsión y atracción respectivamente. Se baso en que la constante “b” toma en cuenta el volumen molar finito de las moléculas lo que hace que el volumen molar real sea más grande en el modelo de gas ideal. Mientras que el termino cuadrático negativo del volumen molar en la componente atractiva es debido a las fuerzas intermoleculares de atracción.

$$P^{rep} = \frac{RT}{V - b} \quad (15)$$

$$P^{atr} = \frac{a}{V^2} \quad (16)$$

Donde:

a: Parámetro de atracción, constante, atm L²/mol.

b: Parámetro que ajusta el volumen molar ideal, constante, mol/L.

Uniendo ambos componentes obtenemos la EOS de Van der Waals como se observa en la Ecuación 17.

$$P = \frac{RT}{V - b} - \frac{a}{V^2} \quad (17)$$

2.2.7.4 Ecuación de estado de Redlich-Kwong

Después de aproximadamente 100 años, Redlich & Kwong modificaron la EOS de Van der Waals (16.2: *van der Waals and Redlich-Kwong Equations of State - Chemistry LibreTexts*, s/f). Redlich & Kwong modificaron la presión de atracción en la EOS de Van der Waals para que sea dependiente de la temperatura, también reformularon su dependencia volumétrica tal como se ve en la Ecuación 18 (Redlich & Kwong, 1949).

$$P = \frac{RT}{V - b} - \frac{a}{\sqrt{T}V(V + b)} \quad (18)$$

De donde, a diferencia de la EOS de Van der Waal, Redlich & Kwong nos muestran las Ecuaciones 19 y 20 para los parámetros “a” y “b”.

$$a = 0.42780 \frac{R^2 T_c^2}{P_c} \quad (19)$$

$$b = 0.086640 \frac{RT_c}{P_c} \quad (20)$$

Luego podemos introducir el factor de compresibilidad en la Ecuación 18 obteniendo una ecuación cúbica, tal como se muestra en la Ecuación 21.

$$Z^3 - Z^2 + (A - B - B^2)Z - AB = 0 \quad (21)$$

Donde:

$$A = \frac{aP}{R^2 T^{2.5}}$$

$$B = \frac{bP}{RT}$$

$$Z = \frac{PV}{RT}$$

2.2.7.5 Ecuación de estado de Peng – Robinson

Después del gran impacto que tuvo la EOS de Redlich-Kwong al modificar la EOS de Van der Waals, varios investigadores trataron de modificarla para que se ajuste mejor a la data experimental para sustancias puras y otros mejoraron la capacidad de la EOS para predecir correctamente el equilibrio líquido-vapor. Peng & Robinson (1976) encontraron que la EOS de Redlich-Kwong generaba satisfactoriamente las densidades molares para los gases, pero no predecía muy bien las densidades molares de los líquidos.

Peng & Robinson expresan la presión del sistema como una suma entre la presión de repulsión y atracción entre moléculas, tal como se vio en la Ecuación 14.

Ellos también mencionan, al igual que Van der Waals, que la presión de repulsión, tal como se muestra en la Ecuación 15 debe ser proporcional a la temperatura porque las moléculas se moverán más rápido y por ende colisionarán más conforme la temperatura se eleve. También comentan que debido a la propiedad de impenetrabilidad dos moléculas no pueden ocupar el mismo espacio, por lo tanto, el fluido presenta un volumen molar mínimo representado por el parámetro “b”. Cuando el volumen molar “V” se acerca al volumen molar mínimo, la presión de repulsión se incrementa indefinidamente (Dahm & Visco, 2014).

Por otro lado, la presión de atracción, tal como se muestra en la Ecuación 22, se utiliza para representar el impacto de las fuerzas que atraen a las moléculas entre sí. Se

utiliza una función dependiente del volumen molar $g(v)$ que se basa en un estudio de ecuaciones semi empíricas (Peng & Robinson, 1976).

Finalmente, Peng & Robinson nos muestran la EOS cúbica final, tal como se ve en la Ecuación 23. La cual puede ser escrita en términos del factor de compresibilidad, formando una ecuación cúbica, tal como se aprecia en la Ecuación 24.

$$p^{atr} = - \frac{a}{g(v)} \quad (22)$$

$$P = \frac{RT}{V-b} - \frac{a}{V(V+b) + b(V-b)} \quad (23)$$

$$Z^3 - (1-B)Z^2 + (A-3B^2-2B)Z - (AB-B^2-B^3) = 0 \quad (24)$$

Donde:

a: Parámetro de atracción, función dependiente de la temperatura, atm L²/mol²

b: Parámetro relacionado al volumen molar mínimo, constante, L/mol

g: Parámetro relacionado a la presión de repulsión, función dependiente de V, atm·

1

$$A = \frac{aP}{R^2T^2}$$

$$B = \frac{bP}{RT}$$

$$Z = \frac{PV}{RT}$$

Para temperaturas críticas, Peng & Robinson expresan las Ecuaciones 25, 26 y 27.

$$a_{(T_c)} = 0.45724 \frac{R^2T_c^2}{P_c} \quad (25)$$

$$b_{(T_c)} = 0.07780 \frac{RT_c}{P_c} \quad (26)$$

$$Z_c = 0.307 \quad (27)$$

Donde:

T_c : Temperatura crítica absoluta, K

P_c : Presión crítica, atm

En caso de estudiar condiciones no críticas, podemos usar las Ecuaciones 28 y 29, donde se introduce una nueva función " $\alpha_{(T_r, \omega)}$ " que depende de la temperatura reducida " T_r " y del factor acéntrico " ω " que es igual a la unidad en condiciones críticas.

$$a_{(T)} = a_{(T_c)} \alpha_{(T_r, \omega)} \quad (28)$$

$$b_{(T)} = b_{(T_c)} \quad (29)$$

Peng & Robinson determinaron la Ecuación 30 y 31 al buscar la relación entre " α ", " T_r " y " ω ".

$$\sqrt{\alpha} = 1 + \kappa(1 - \sqrt{T_r}) \quad (30)$$

$$\kappa = 0.37464 + 1.54226\omega - 0.26992\omega^2 \quad (31)$$

Luego usaron la relación termodinámica, expresada en la Ecuación 32, en la Ecuación 23, obteniendo la Ecuación 33.

$$\ln\left(\frac{f}{p}\right) = \int_0^P \left(\frac{V}{RT} - \frac{1}{P}\right) dP \quad (32)$$

$$\ln\left(\frac{f}{P}\right) = Z - 1 - \ln(Z - B) - \frac{A}{2\sqrt{2}B} \ln\left(\frac{Z + (1 + \sqrt{2})B}{Z + (1 - \sqrt{2})B}\right) \quad (33)$$

Donde:

f: Fugacidad del componente puro.

p: Presión parcial

Es necesario mencionar que la Ecuación 33 esta especificada para sustancias puras, no obstante, Peng & Robinson también nos muestran la Ecuación 34 para calcular la fugacidad del componente “k” en una mezcla, ellos señalan que debemos calcular los parámetros “a” y “b” a partir de los parámetros binarios que dependen de como interaccionan los componentes entre sí en la mezcla, esto está definido en las Ecuaciones 35, 36 y 37. En la ecuación 35 se está utilizando la regla de mezcla de Van der Waals.

$$\ln\left(\frac{f_k}{x_k P}\right) = \left(\frac{b_k}{b}\right)(Z - 1) - \ln(Z - B) \quad (34)$$

$$- \frac{A}{2\sqrt{2}B} \left(\frac{2 \sum_i x_i a_{ik}}{a} - \frac{b_k}{b} \right) \ln\left(\frac{Z + (1 + \sqrt{2})B}{Z + (1 - \sqrt{2})B}\right)$$

$$a = \sum_i \sum_j x_i x_j a_{ij} \quad (35)$$

$$b = \sum_i x_i b_i \quad (36)$$

$$a_{ij} = (1 - \delta_{ij}) a_i^{\frac{1}{2}} a_j^{\frac{1}{2}} \quad (37)$$

Donde:

k, i, j : Representan a los componentes en la mezcla.

f_k : Fugacidad del componente “k” en la mezcla.

a_i, a_j : Parámetros “a” de los componentes puros “i” y “j”

b_i : Parámetro “b” del componente “i”

a_{ij} : Parámetros binario “a” de la mezcla “i” y “j”

δ_{ij} : Coeficiente de interacción binaria entre el componente “i” y “j”

También existe la posibilidad de que se puedan usar otras reglas de mezcla como la de Panagiotopoulos & Reid (1986) para mezclas altamente polares y sistemas asimétricos tal como lo podemos apreciar en la Ecuación 38.

$$a_{ij} = \sqrt{a_i a_j} [1 - k_{ij} + (k_{ij} - k_{ji}) x_i] \quad (38)$$

En la regla de mezcla de Panagiotopoulos no se considera que el sistema es simétrico con respecto a los parámetros de interacción binaria k_{ij} , y si en caso fuera simétrico, la Ecuación 38 se reduciría a la regla de mezcla de Van Der Waals tal como la vimos en la Ecuación 35.

2.2.8 Método ϕ - ϕ (f_i - f_i)

Podemos describir las fases líquida y vapor usando una sola EOS para ambas fases, para ello se usa el método $f_i - f_i$.

Sabemos que en el equilibrio líquido vapor la fugacidad de cada especie es la misma en todas las fases, podemos expresarlo matemáticamente para una mezcla binaria como se ve en la Ecuación 39.

$$\widehat{f}_i^v = \widehat{f}_i^l, (i = 1,2) \quad (39)$$

Asimismo, podríamos usar la ecuación 9, obteniendo la Ecuación 40, que luego sería simplificada a la ecuación 41.

$$\widehat{\phi}_i^v y_i P = \widehat{\phi}_i^l x_i P, (i = 1,2) \quad (40)$$

$$\widehat{\phi}_i^v y_i = \widehat{\phi}_i^l x_i, (i = 1,2) \quad (41)$$

Donde:

$\widehat{\phi}_i^v$: Coeficiente de fugacidad en fase vapor del componente “i” en una mezcla.

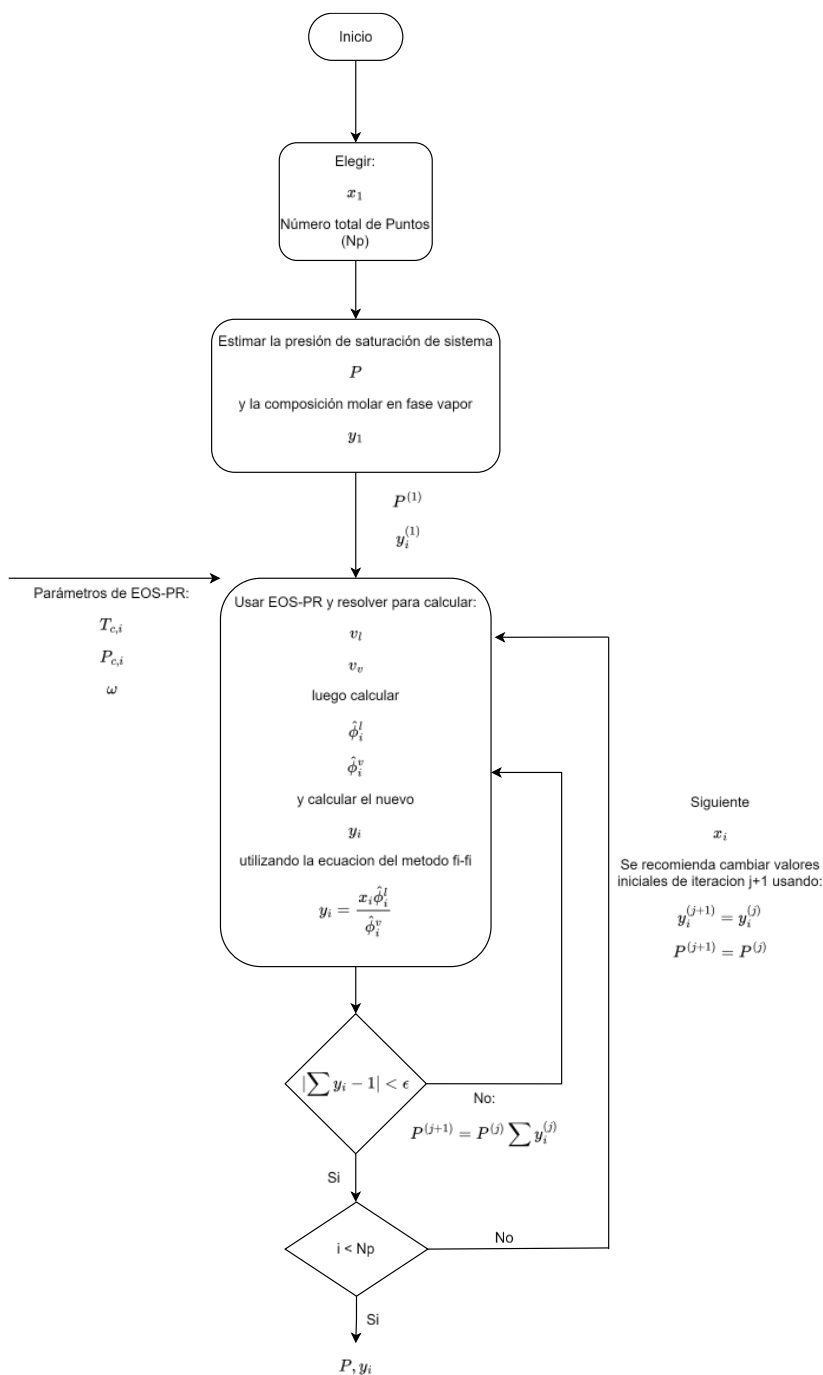
$\widehat{\phi}_i^l$: Coeficiente de fugacidad en fase líquida del componente “i” en una mezcla.

2.2.9 Algoritmo clásico para la solución del ELV con el método ϕ - ϕ (fi-fi)

En la Figura 3 podemos ver el algoritmo clásico para solucionar el ELV y crear el diagrama P_{xy} usando el método fi-fi.

Figura 3

Algoritmo para solución del ELV (Diagrama $P_{xy} - T$ constante) utilizando el método fi-fi.



Nota. Adaptado de Koretsky (2013)

2.3 Marco conceptual

2.3.1 Redes neuronales

Una red neuronal se puede definir como una serie de algoritmos y técnicas que tratan de simular cómo funcionan las neuronas del cerebro humano. Pueden encontrar

patrones en data y predecir nuevos valores basándose en los patrones encontrados (Shaik et al., 2021).

2.3.2 Estructura de una red neuronal

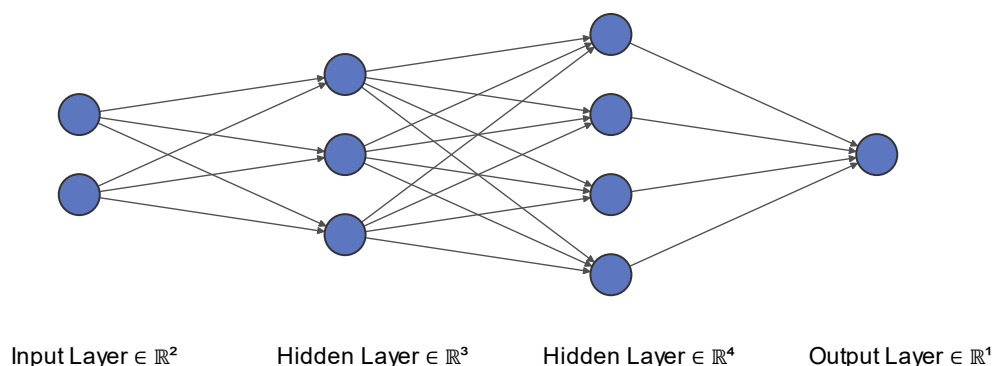
Una red neuronal tiene en su estructura; una capa de entrada (input layer) que recibe varias señales de entrada, una capa de salida (output layer) que envía varias señales de salida y una o más capas ocultas (hidden layer) que realizan transformaciones no lineales a las señales de entrada (Profillidis & Botzoris, 2019). Podemos ver una representación de una estructura de una red neuronal en la Figura 4, tenemos dos señales de entrada en la capa de entrada, dos capas ocultas con tres neuronas y cuatro neuronas respectivamente, y finalmente tenemos una señal de salida en la capa de salida.

Las señales de entrada son valores o características del sistema que estamos estudiando, las neuronas usan estos datos para encontrar patrones y finalmente envían una señal de salida que es la característica que estamos buscando obtener.

Es importante resaltar que la normalización de la data de entrada es una práctica recomendada para entrenar redes neuronales con la finalidad de tener los valores de un conjunto de datos en una misma escala (X. Wang et al., 2024).

Figura 4

Estructura de una red neuronal



Nota. Creación propia usando un graficador de red neurales en <https://alexlenail.me/NN-SVG/>

2.3.2.1 Normalización de data

La normalización de data es una metodología cuyo fin es transformar los valores de un conjunto de datos en una misma escala (X. Wang et al., 2024).

La función comúnmente utilizada para normalizar datos es la puntuación Z, el cual se puede calcular mediante la Ecuación 42 (Singh & Singh, 2020).

$$X' = \frac{X - \mu}{\sigma} \quad (42)$$

Donde:

X: Entrada del modelo no normalizado.

X': Entrada del modelo normalizado

μ : Media calculada sobre el conjunto de datos de entrada.

σ : Desviación estándar calculada sobre el conjunto de datos de entrada.

2.3.2.2 Funcionamiento de las redes neuronales

Las funciones básicas de las redes neuronales son:

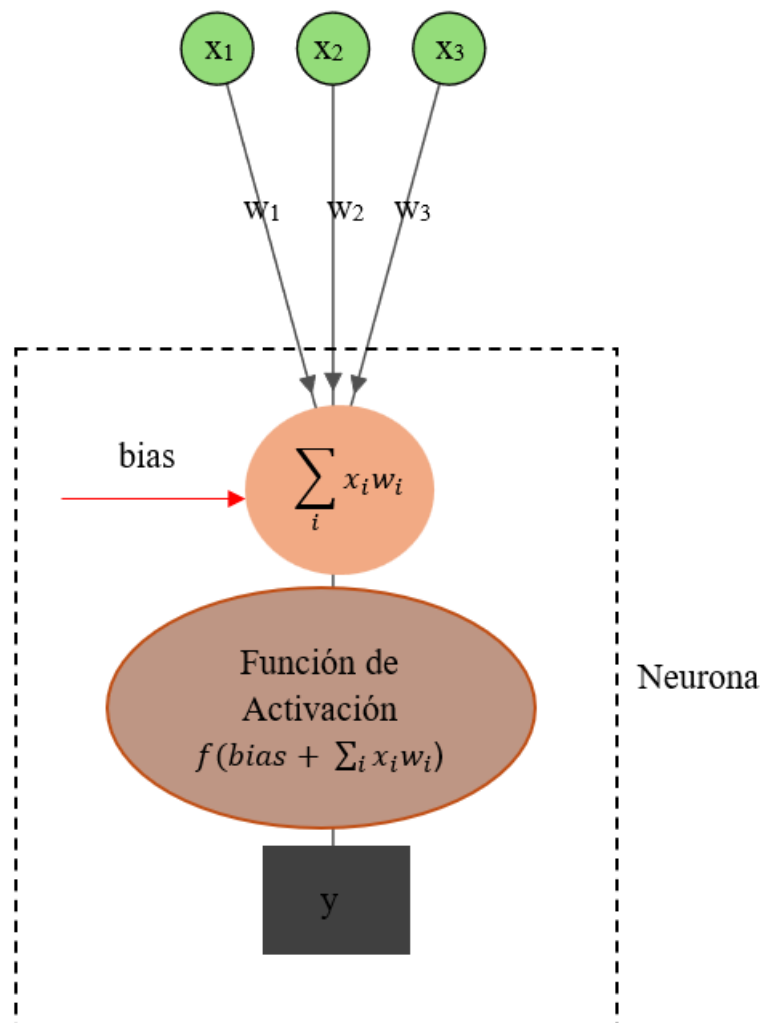
- Recibir la data.
- Usando decidir si la data activará o desactivará ciertas neuronas.
- Minimizar el error mediante iteración de la data.
- Obtención del valor de salida.

2.3.2.3 Aspecto matemático de la neurona

De acuerdo con la Figura 5, una neurona puede tener varios valores numéricos de entrada (x_i), luego estos valores se relacionan cada uno con un peso (w_i), luego hacemos una suma de los productos de x_i con sus respectivos pesos w_i , posteriormente agregaremos un bias que ajustara esta función, finalmente evaluamos nuestro resultado en una función de activación y obtenemos el valor de la neurona, que podrá ser usado, siguiendo el mismo mecanismo, junto a otras neuronas calculadas para el cálculo de valores de neuronas de la próxima capa (Calin, 2020). Podemos ver un ejemplo con pesos y bias reales para una red neuronal 2-12-9-2 (2 entradas, 12 neuronas en la capa oculta I, 9 neuronas en la capa oculta II, 2 salidas en la capa de salida) en la Figura 6.

Figura 5

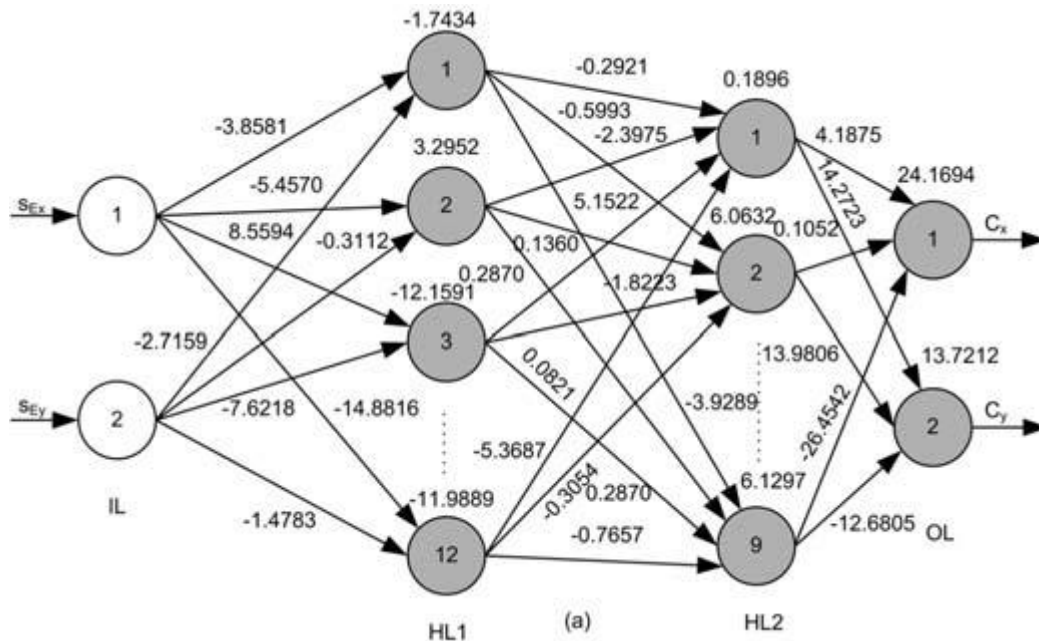
Estructura de una red neuronal.



Nota. Creación propia usando un graficador de red neurales en <https://alexlenail.me/NN-SVG/>

Figura 6

Red neuronal 2-12-9-2



Nota. Tomado de (Raj & Padmavathi, 2016).

2.3.3 Modelamiento híbrido en procesos industriales

La integración de metodologías basadas en datos y fundamentos teóricos en Ingeniería Química nos da como resultado el modelamiento híbrido. Usualmente se formulan conceptos de balances de masa y energía para la parte teórica del modelo híbrido. Mientras que en la parte basada en datos podemos utilizar diferentes técnicas, entre ellas redes neuronales (Marquardt, 2002).

2.3.3.4 Redes neuronales híbridas

Los modelos rigurosos en Ingeniería Química en conjunto con redes neuronales artificiales crean las redes neuronales híbridas (Mogk et al., 2002).

Las dos partes básicas de un arreglo neural híbrido son el modelo mecánico y la red neuronal. El modelo mecánico es un modelo que expresa leyes de conservación o fenómenos de transporte, como balances de masa, energía o conservación del momentum, mientras que la red neuronal generalmente se utiliza para modelar el fenómeno químico o físico cuando no existe un modelo teórico desarrollado o cuando es muy complejo de formular (Sansana et al., 2021).

Existen dos tipos básicos de redes neuronales híbridas:

- Redes Neuronales Híbridas en serie: En este arreglo la red neuronal alimenta al modelo mecánico (Figura 7) o el modelo mecánico alimenta a la red neuronal (Figura 8).
- Redes Neuronales Híbridas en paralelo: En este arreglo podemos tener dos variantes: La variante cooperativa es cuando el modelo mecánico puede modelar el proceso, pero existen variaciones que no las podemos explicar, entonces podemos usar la red neuronal para corregir no linealidades o efectos que no han sido considerados por el modelo mecánico; La otra variante es la competitiva, cuando funcionan tanto el modelo mecánico como la red neuronal y luego se ponderan para obtener una predicción final mediante multiplicación, adición o un filtro del Kalman (Sansana et al., 2021). Podemos apreciar el esquema de la red neuronal híbrida en paralelo en la Figura 9.

Figura 7

Red Neuronal Híbrida en Serie Tipo I.

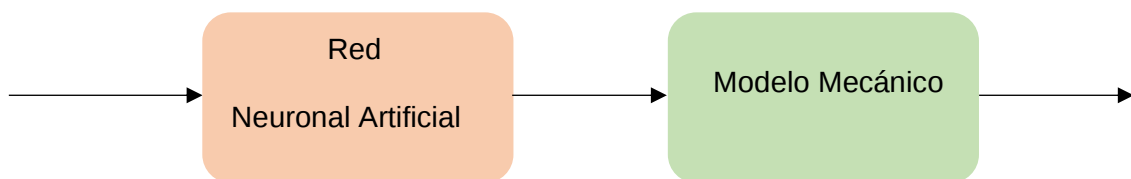


Figura 8

Red Neuronal Híbrida en Serie Tipo II.

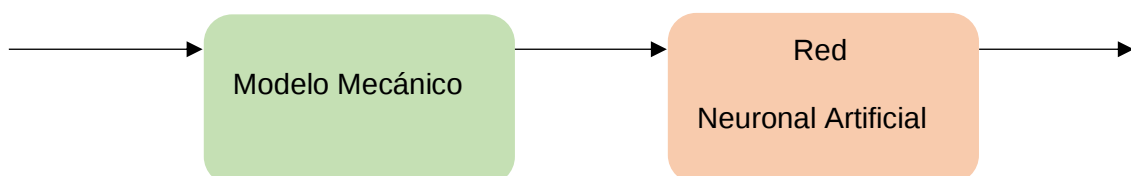
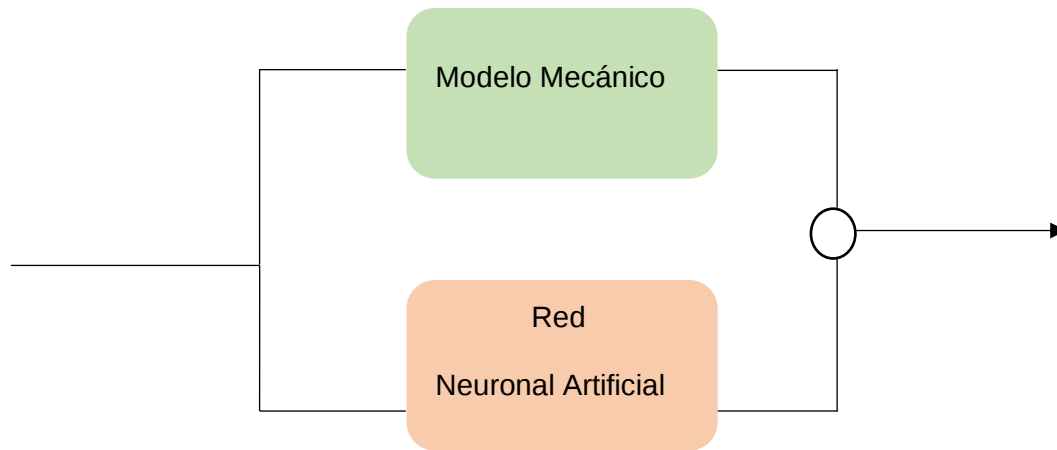


Figura 9

Red Neuronal Híbrida en Paralelo



2.3.4 Funciones de activación

Las redes neuronales necesitan de funciones de activación para poder funcionar, podemos pensar en una función de activación como una función común y corriente que se estudia en matemática, damos una entrada y recibimos una salida. Sin embargo, estas funciones tienen un rol fundamental en las redes neuronales porque introducen no linealidad en el modelo. Sin ellas, una red neuronal no podría aprender relaciones complejas entre los datos, ya que sería equivalente a una simple combinación lineal de las entradas. Esto significa que, sin una función de activación, la capacidad del modelo para representar patrones complejos sería muy limitada. Existen diversos tipos de funciones de activación, y cada una de ellas puede ser más útil que otra para un determinado problema (Montesinos López et al., 2022).

2.3.4.1 Rectified Linear Unit (ReLU)

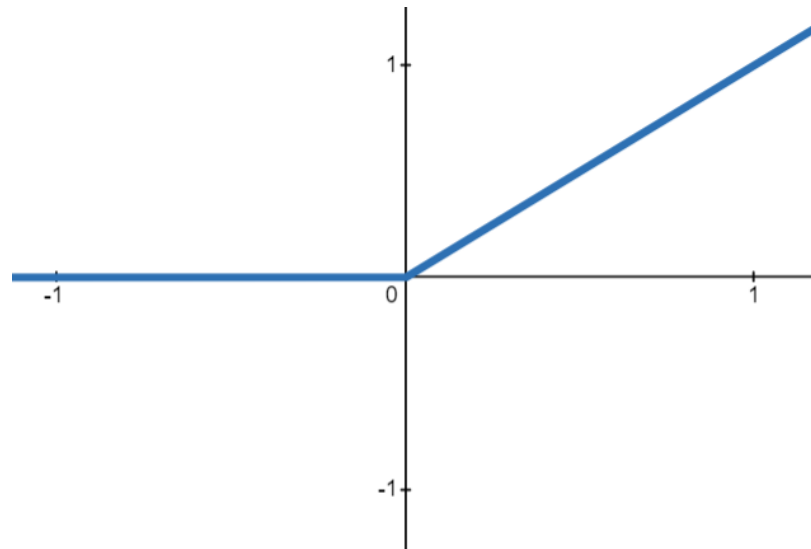
Esta función de activación es una de las más populares debido a su simplicidad y eficiencia. Activa solo las neuronas con valores positivos, dejando los negativos en cero, pero tienen un problema de causar neuronas “muertas” si varios valores quedan en cero.

Matemáticamente lo podemos expresar como la Ecuación 43 y gráficamente lo podemos ver en la Figura 10 (Sharma et al., 2020).

$$f(x) = \max(0, x) \quad (43)$$

Figura 10

Función de activación ReLU



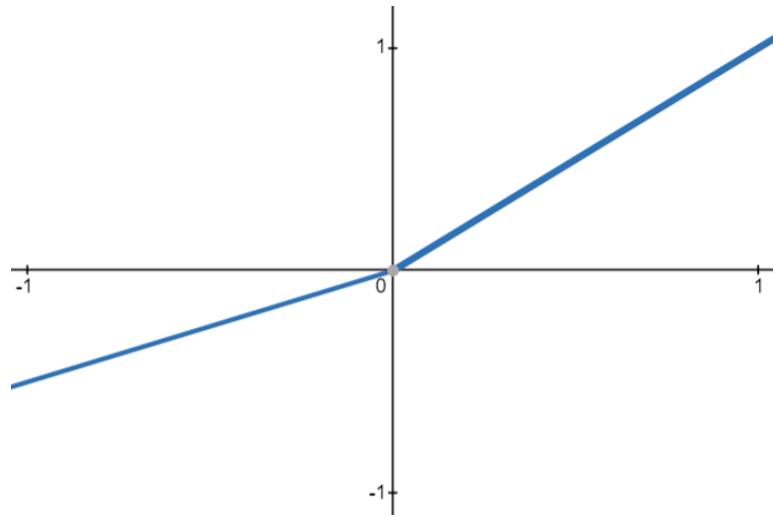
2.3.4.2 Leaky ReLU

Para solucionar el problema de neuronas muertas causadas por varios valores cero en ReLU, surgió esta función de activación que devuelve una pequeña pendiente para valores negativos. Matemáticamente lo podemos expresar como la Ecuación 44 y gráficamente lo podemos ver en la Figura 11 (Sharma et al., 2020).

$$f(x) = \begin{cases} x, & \text{si } x > 0 \\ \alpha x, & \text{si } x \leq 0 \end{cases} \quad (44)$$

Figura 11

Función de activación Leaky ReLU con $\alpha=0.5$



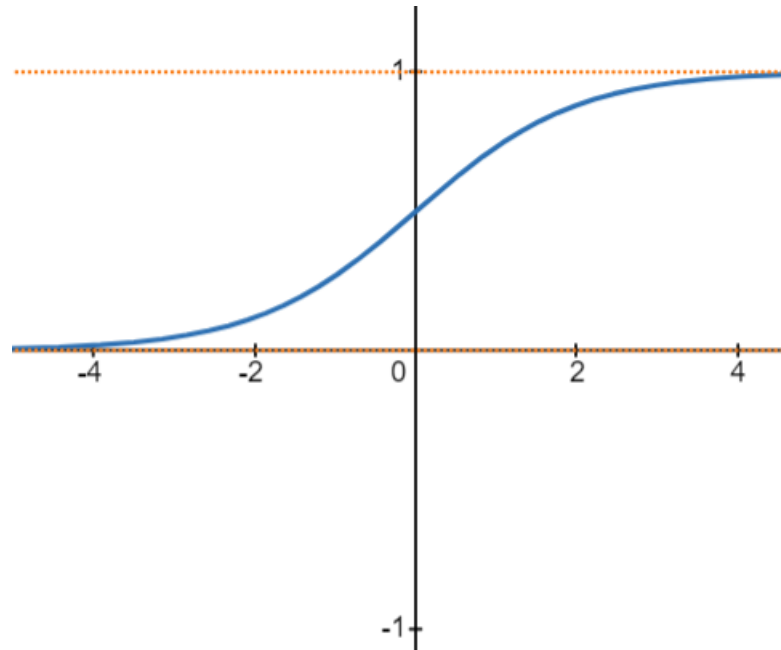
2.3.4.3 Sigmoide

Esta función convierte los valores de entrada en un valor de salida entre 0 y 1, es muy usado en problemas de clasificación binaria. Matemáticamente lo podemos expresar como la Ecuación 45 y gráficamente lo podemos ver en la Figura 12 (Sharma et al., 2020).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (45)$$

Figura 12

Función de activación Sigmoide



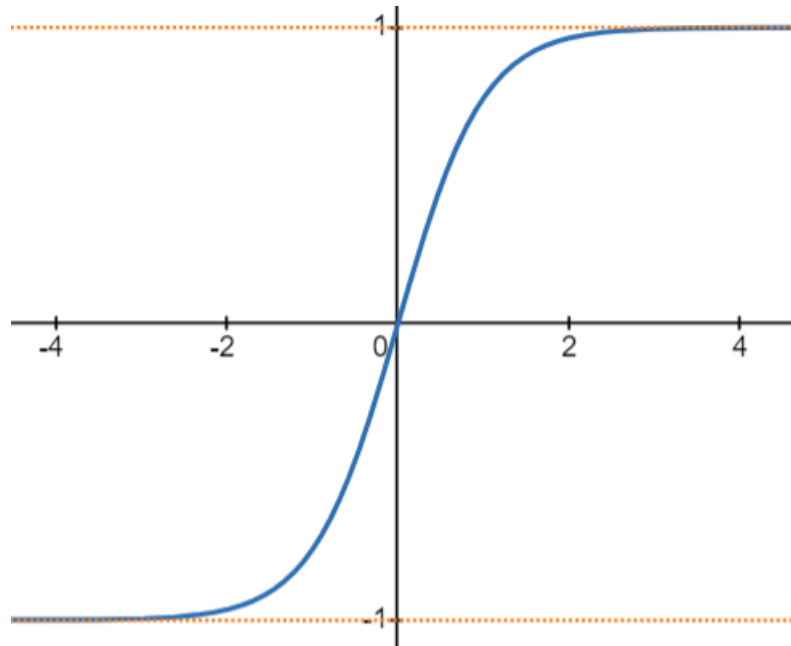
2.3.4.4 Tangente hiperbólica

Es muy similar a la sigmoide, pero con un rango de -1 a 1, por lo que es muy útil cuando se necesitan valores negativos. Matemáticamente lo podemos expresar como la Ecuación 46 y gráficamente lo podemos ver en la Figura 13 (Sharma et al., 2020).

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (46)$$

Figura 13

Función de activación Tangente hiperbólica



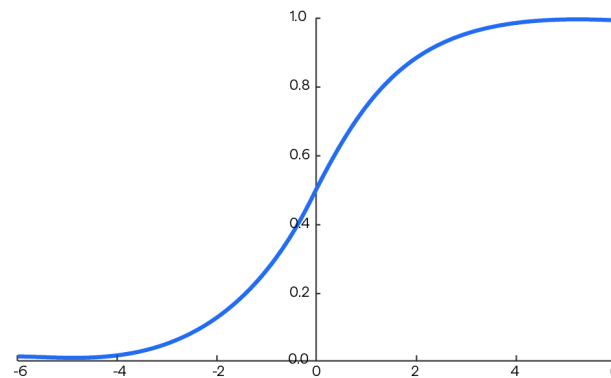
2.3.4.5 Softmax

Esta función convierte un conjunto de valores en probabilidades que suman 1, y es muy usado en la capa de salida para clasificación multiclase. Matemáticamente lo podemos expresar como la Ecuación 47 y gráficamente lo podemos ver en la Figura 14 (Sharma et al., 2020).

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (47)$$

Figura 14

Función de activación Softmax



Nota. Obtenido de (Softmax Function: Advantages and Applications | BotPenguin, s/f)

2.3.5 Modelamiento del equilibrio líquido-vapor usando redes neuronales como predictor de parámetros de interacción binaria

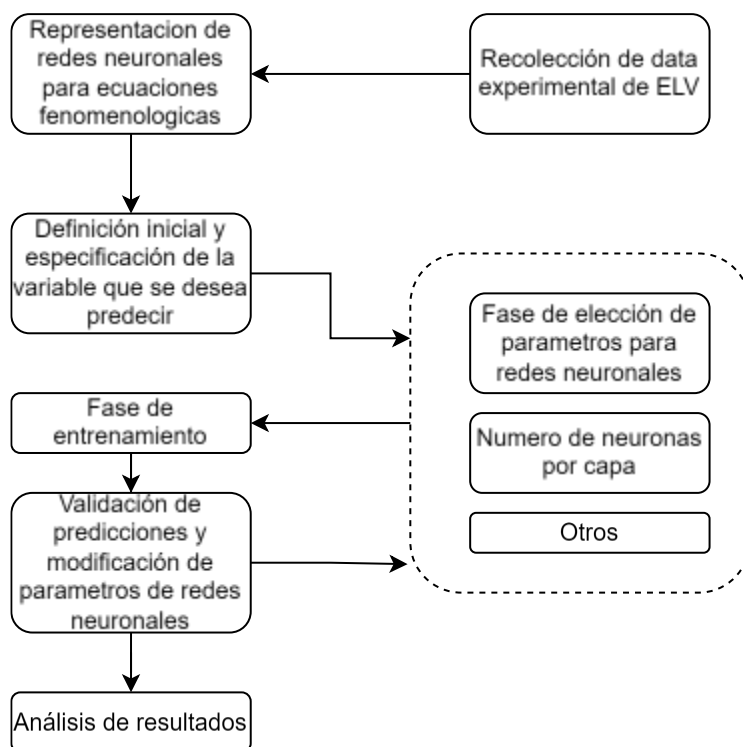
Se pueden usar redes neuronales como predictores de parámetros de interacción binaria, especialmente cuando no se tiene una expresión que nos indique como varían los coeficientes de interacción binaria con la temperatura (Vyhmeister et al., 2017).

Esto logra extender el equilibrio líquido vapor a temperaturas que no han sido consideradas en investigaciones experimentales.

Para ello podríamos aplicar un procedimiento similar al de la Figura 15.

Figura 15

Diagrama de flujo de procedimiento de aplicación de redes neuronales al ELV



Nota. Basado en el diagrama presentado por (Si-Moussa et al., 2008).

2.3.6 Métodos computacionales

2.3.6.6 Optimización por recocido simulado y búsqueda local (Dual Annealing)

El recocido simulado (Simulated Annealing) es un método de optimización estocástico para encontrar mínimos y máximos globales, por lo que suele ser usado para funciones objetivo que poseen una superficie de respuesta no lineal.

El recocido simulado tiene ese nombre por la analogía de como los metales se enfrían desde una alta temperatura, en este proceso de enfriamiento el metal alcanza progresivamente un estado de mínima energía (Suman, 2021). En cada iteración del algoritmo, se genera un nuevo punto de manera aleatoria, la distancia entre el nuevo punto y el actual se determina mediante una distribución de probabilidad cuya escala está ligada a la temperatura. Al permitir la aceptación de puntos que decrezcan la función objetivo evita quedarse atrapado en mínimos locales. El algoritmo utiliza un esquema de enfriamiento que reduce gradualmente la temperatura conforme avanza el algoritmo. A

medida que la temperatura disminuye, el algoritmo ajusta su búsqueda para finalmente converger hacia un mínimo (*What Is Simulated Annealing? - MATLAB & Simulink*, s/f).

Dual Annealing es una implementación del algoritmo clásico de recocido simulado que fue propuesto por Kirkpatrick et al. (1983) y posteriormente generalizado por Xiang et al. (1997).

El algoritmo de Dual Annealing desarrollado por la librería científica de Python Scipy (Virtanen et al., 2020) nos indica que la optimización por Dual Annealing está basada en un método para refinar la solución encontrada por un proceso generalizado de recocido. El algoritmo utiliza una distribución de Cauchy-Lorentz distorsionada, cuya forma está controlada por el parámetro q_v , tal como lo podemos ver en la Ecuación 48.

$$g_{q_v}(\Delta x(t)) \propto \frac{[T_{q_v}(t)]^{-\frac{D}{3-q_v}}}{\left[1 + (q_v - 1) \frac{(\Delta x(t))^2}{[T_{q_v}(t)]^{\frac{2}{3-q_v}}}\right]^{\frac{1}{q_v-1} + \frac{D-1}{2}}} \quad (48)$$

Donde:

q_v : Parámetro de visitas.

t : Tiempo artificial.

$\Delta x(t)$: Distancia de salto de prueba.

T : Temperatura artificial.

Desde el punto inicial, después de haber llamado a la función de distribución de visitas, podemos calcular la probabilidad de aceptación mediante la Ecuación 49.

$$p_{q_a} = \min \left\{ 1, [1 - (1 - q_a)\beta\Delta E]^{\frac{1}{1-q_a}} \right\} \quad (49)$$

Donde:

q_a : Parámetro de aceptación.

Para $q_a < 1$, la probabilidad de aceptación cero es asignada a los casos donde $[1 - (1 - q_a)^{\beta \Delta E}] < 0$.

La temperatura artificial $T_{qv}(t)$ decrece de acuerdo con la Ecuación 50.

$$T_{qv}(t) = T_{qv}(1) \frac{2^{q_v-1} - 1}{(1+t)^{q_v-1} - 1} \quad (50)$$

2.3.7 Splines paramétricos

Los splines paramétricos son curvas no necesariamente funcionales que pueden ajustarse a datos no funcionales mediante una parametrización (Dierckx, 1982).

Podemos definir un spline paramétrico como una curva dependiente de un parámetro “t”, con funciones $x(t)$ y $y(t)$ que se ajustara a data experimental (puntos de control).

Para representar la posición de un punto del spline paramétrico en un plano bidimensional podemos usar la Ecuación 51.

$$C(t) = [x(t), y(t)] \quad (51)$$

Donde:

$C(t)$: Punto (x,y)

t: Parámetro

$x(t)$: Ordenada del punto (x,y)

$y(t)$: Abscisa del punto (x,y)

Cada función $x(t)$ y $y(t)$ puede ajustarse a un conjunto de datos mediante la estimación de un polinomio de grado “n”, tal como lo vemos en la Ecuación 52 y 53.

$$x(t) = a_x + b_x t + c_x t^2 + d_x t^3 + \dots + e_x t^n \quad (52)$$

$$y(t) = a_y + b_y t + c_y t^2 + d_y t^3 + \dots + e_y t^n \quad (53)$$

Los coeficientes de los polinomios se ajustan a partir de dos condiciones: Minimizar la desviación entre el spline paramétrico y los puntos de control; Asegurar la continuidad de la curva mediante la primera y segunda derivada.

Para minimizar la desviación entra el spline paramétrico y los puntos de control, se utiliza la función de costo que vemos en la Ecuación 54.

$$Error = \sum_{i=1}^n [(x_i - Cx_{(t_i)})^2 + (y_i - Cy_{(t_i)})^2] \quad (54)$$

Donde:

$Cx_{(t_i)}$: Ordenada de spline paramétrico

$Cy_{(t_i)}$: Abscisa de spline paramétrico

x_i : Ordenada de punto de control

y_i : Abscisa de punto de control

i : Número de punto

También podemos ajustar el suavizado del spline paramétrico agregando un mayor grado de ajuste a la segunda derivada de la curva. Para ello se utiliza una nueva función de costo que incluye un término regularizador que penaliza la curvatura excesiva, tal como lo vemos en la Ecuación 55.

$$Costo\ regularizador = Error + \lambda \int (\frac{d^2C(t)}{dt^2})dt \quad (55)$$

Donde:

λ : Controla el grado de suavización

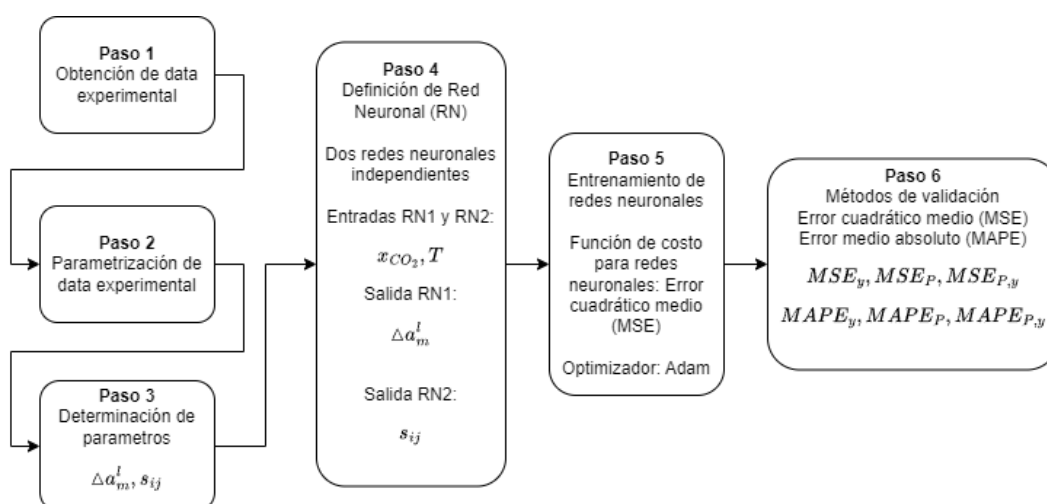
Capítulo III. Desarrollo del trabajo de investigación

3.1 Métodos

En la Figura 16 se muestra un diagrama de bloques del procedimiento realizado en la siguiente investigación.

Figura 16

Diagrama de bloques del procedimiento para el presente trabajo de investigación



3.1.1 Obtención de data experimental

Se revisaron investigaciones experimentales sobre el ELV de la mezcla $CO_2 - H_2O$. La mayoría de las bases de datos actuales como la de Song et al. (2005) estudian la solubilidad del CO_2 en H_2O por lo que las fracciones molares que consideran son muy pequeñas (0 – 0.06).

Es por ello que para la presente tesis se utilizó como base de datos, la investigación realizada por Tödheide & Franck (1963) debido a que presenta un amplio rango de temperaturas (50°C a 350°C) y presiones (200 bar a 1500 bar), además de mostrar completamente las isothermas a lo largo de la fracción molar de CO_2 tanto para la curva de rocío como para la curva de burbuja.

3.1.2 Parametrización de data experimental

Para que la red neuronal, que posteriormente implementaremos, aprenda a crear relaciones entre los datos necesitamos aumentar la data, por ello se utilizaron dos técnicas de aumentación con el fin de generar más muestras y mejorar el entrenamiento del modelo.

- Spline paramétrico
- Digitalización de datos, debido a que Tödheide & Franck (1963) muestran solamente una parte de los datos del ELV en forma tabular, se decidió realizar un procedimiento parecido al de Johnsson & Norén (2017) donde usaron un software para obtener data de un gráfico de un mapa de compresor. Las digitalizaciones en la presente tesis se realizaron mediante el software Origin 2018©. Es importante resaltar que este proceso solamente se aplicó a ciertos puntos clave de las gráficas que no especifican Tödheide & Franck (1963) de modo que sirvan como ayuda para que el spline paramétrico pueda capturar las curvas más importantes de cada isoterma.

3.1.3 Determinación de parámetros “a” en fase líquida y “s_{ij}” de la EOS – PR

La red neuronal podrá predecir la corrección del parámetro “a” en fase líquida y el parámetro de interacción binaria “s_{ij}” de la EOS – PR como veremos más adelante, es por ello que se crearán los algoritmos necesarios para resolver la EOS – PR usando el método fi-fi como se mostró en la Figura 3.

Para la presente tesis se utilizó Python como lenguaje de programación y programación orientada a objetos (POO), un enfoque que permite estructurar el código en clases y objetos, facilitando su modificación y reutilización. Uno de los beneficios clave de la POO es la capacidad de modificar y extender el código sin afectar otras partes, lo cual es particularmente útil cuando se analizan múltiples propiedades y parámetros de la EOS. Esta modularidad permitió implementar y ajustar de manera eficiente diferentes configuraciones de la EOS - PR, realizando cambios sobre las propiedades y parámetros de interacción de la mezcla CO₂-H₂O sin necesidad de reescribir grandes secciones de código.

3.1.3.1 Determinación de soluciones de una ecuación cúbica

Debido a la precisión necesaria para encontrar soluciones a la EOS – PR, y dado a su naturaleza cubica, necesitamos de un método que nos ayude a encontrar las soluciones reales. Para ello se utilizó un método basado en la formula del ángulo triple. Para profundizar más en el aspecto teórico del método, podemos revisar el trabajo de investigación de Franssen (2016) donde analiza algunos casos de “Casus Irreducibilis”

Para evitar confundir el k_{ij} con el coeficiente de distribución K , se llamarán a partir de ahora s_{ij} a los parámetros de interacción binaria que antes eran k_{ij} .

Podemos encontrar el código para encontrar las soluciones de una ecuación cúbica en las siguientes líneas:

```
import numpy as np
import math

r = 8.311451e-2

def raiz_cubica(x):
    if x>0:
        return math.pow(x, float(1)/3)
    if x<0:
        return -math.pow(abs(x), float(1)/3)
    else:
        return 0

def resolver_cubica(m,n,s,t, tipo_sol='todas'):
    n = n/m
    s = s/m
    t = t/m
    Q = (3*s-n**2)/9
    R = (9*n*s-27*t-2*n**3)/54
    D = Q**3+R**2 #Calculo de discriminante
    # Evaluación de método a seguir usando la discriminante
    # D>0, Una raíz real y dos complejas
    if D>0:
        S1 = raiz_cubica(R+np.sqrt(D))
        S2 = raiz_cubica(R-np.sqrt(D))
        sol_1 = S1+S2-n/3
        sol_2 = complex(-(S1+S2)/2-n/3, (S1-S2)*np.sqrt(3)/2)
        sol_3 = complex(-(S1+S2)/2-n/3, -(S1-S2)*np.sqrt(3)/2)
        if tipo_sol=='real':
            sol = np.array([sol_1])
        elif tipo_sol == 'compleja':
            sol = np.array([sol_2, sol_3])
        else:
            sol = np.array([sol_1, sol_2, sol_3])
    # D<0, Todas las raices son reales y distintas
    if D<=0 :
```

```

theta = np.arccos(-R/np.sqrt(-Q**3))
sol_1 = -2*np.sqrt(-Q)*np.cos(theta/3)-n/3
sol_2 = -2*np.sqrt(-Q)*np.cos((theta+2*np.pi)/3)-n/3
sol_3 = -2*np.sqrt(-Q)*np.cos((theta+4*np.pi)/3)-n/3
sol = np.array([sol_1, sol_2, sol_3])
sol = np.sort(sol)[::-1]
if tipo_sol=='real':
    sol = sol
elif tipo_sol == 'compleja':
    sol = 'No existen soluciones complejas'
else:
    sol = sol
return sol

```

Nota. Creación propia

3.1.3.2 Clase PR-EOS-Componentes

Se creó la clase PR-EOS-Componentes como un generador de objetos de componentes, el cual tiene como atributos a los parámetros a_{Tc} , b_{Tc} , α , a , b los cuales ya hemos definido en las Ecuaciones 25, 26, 30, 28, 29.

Podemos observar el código en las siguientes líneas:

```

class pr_eos_componente:
    #Esta clase definira los parametros en la EOS de Peng-Robinson
    Original
    r = 8.311451e-2
    def __init__(self, nombre, Tc, Pc, w):
        self.nombre = nombre
        self.Tc = Tc
        self.Pc = Pc
        self.w = w
        self.a_tc = self.calcular_a_tc()
        self.b_tc = self.calcular_b_tc()

        print(
            "Se creó el componente ", nombre, " con los siguientes
            parametros: ", "\n"
            "Tc = ", Tc, "K /----/ ", "Pc_1 = ", Pc, "bar /----/ ",
            "Factor Acéntrico(w) = ", w)

    def calcular_a_tc(self):
        a_tc = 0.45724*r**2*self.Tc**2/self.Pc
        return a_tc

    def calcular_b_tc(self):
        b_tc = 0.07780*r*self.Tc/self.Pc
        return b_tc

    def calcular_alfa(self, T):
        k = 0.37464 + 1.54226*self.w - 0.26992*self.w**2
        alfa = (1 + k*(1-np.sqrt(T/self.Tc)))**2
        return alfa

    def calcular_a(self, T):

```

```

        a = self.a_tc*self.calcular_alfa(T)
        return a

    def calcular_b(self):
        b = 0.07780*r*self.Tc/self.Pc
        return b

    def __str__(self):
        return f"Componente {self.nombre}: Tc={self.Tc}, Pc={self.Pc},
w={self.w}"

```

Nota. Creación propia

3.1.3.3 Clase PR-EOS-Mezcla

Se creó la clase PR-EOS-Mezcla como un generador de objetos de mezclas, el cual tiene como atributos principales a los parámetros a , b de mezcla, así como a los parámetros A y B de mezcla los cuales ya hemos definido en las Ecuación 28 y 29.

En esta clase podemos ver una gran ventaja que tiene la programación orientada a objetos pues usaremos la clase PR-EOS-Mezcla tanto para generar los parámetros para fase líquida como para fase vapor sin necesidad de duplicar el código y solo llamaremos a los parámetros cuando sea necesario evitando así cálculos intermedios innecesarios.

Es importante resaltar que para esta clase, las reglas de mezcla utilizadas fueron las de Van Der Waals, como lo vimos en la Ecuación 35, y la regla de mezcla propuesta por Panagiotopoulos & Reid (1986), como lo vimos en la Ecuación 38. También es importante resaltar que esta clase acepta como inputs opcionales a los parámetros “ a ” y “ b ” de mezcla en fase líquida, esto es muy importante ya que más adelante analizaremos el comportamiento de estos parámetros y necesitamos una manera de acceder a ellos sin modificar el código.

Podemos observar el código en las siguientes líneas:

```

class pr_eos_mezcla:
    # Esta clase definira los parametros de mezcla
    r = 8.311451e-2
    def __init__(self, componentes, fracciones_molares, T, P, sij, sji,
a_m = None, b_m = None): #Añadiendo modificador de a_m y b_m
        self.componentes = componentes # Lista de componentes
        self.fracciones_molares = fracciones_molares
        self.T = T
        self.P = P
        self.sij = sij
        self.sji = sji
        self.a_m = a_m if a_m is not None else self.calcular_a_mezcla()

```

```

        self.b_m = b_m if b_m is not None else self.calcular_b_mezcla()
#self.calcular_b_mezcla()
        self.A_m = self.calcular_A_mezcla()
        self.B_m = self.calcular_B_mezcla()

@property
def fracciones(self):
    return self.fracciones_molares[0], self.fracciones_molares[1]

def calcular_a_mezcla(self):
    x1, x2 = self.fracciones
    a1 = self.componentes[0].calcular_a(self.T)
    a2 = self.componentes[1].calcular_a(self.T)
    a11 = a1
    a22 = a2
    a12 = np.sqrt(a1 * a2) * (1 - self.sij + (self.sij - self.sji) * x1)
    a21 = np.sqrt(a2 * a1) * (1 - self.sji + (self.sji - self.sij) * x2)
    a_m = x1**2*a11 + x2**2*a22 + x1*x2*a12 + x2*x1*a21
    return a_m

def calcular_b_mezcla(self):
    b1 = self.componentes[0].calcular_b()
    b2 = self.componentes[1].calcular_b()
    x1, x2 = self.fracciones
    b_m = x1*b1 + x2*b2
    return b_m

def calcular_A_mezcla(self):
    A_m = self.a_m*self.P/r**2/self.T**2
    return A_m

def calcular_B_mezcla(self):
    B_m = self.b_m*self.P/r/self.T
    return B_m

def __str__(self):
    return f"Mezcla: a_m={self.a_m}, b_m={self.b_m}"

```

Nota. Creación propia

3.1.3.4 Clase PR-EOS-ELV

Finalmente se creó una clase relacionada a los cálculos necesarios para solucionar el ELV, esta clase tiene como input al objeto mezcla y a las fracciones molares en fase vapor, además de inputs opcionales de “a” y “b” de mezcla en fase vapor.

Con esta clase se pretende calcular volúmenes molares en ambas fases, factores de compresibilidad en ambas fases y coeficientes de fugacidad para ambos componentes en ambas fases.

Para el coeficiente de fugacidad se utilizó la Ecuación 9, y para las soluciones de la ecuación cúbica del volumen molar se utilizó el método descrito en la sección 3.1.3.1

juntamente con la forma cúbica de la EOS-PR respecto al volumen, como lo podemos ver en la Ecuación 50.

$$PV^3 + (Pb - RT)V^2 + (-3Pb^2 - 2RTb + a)V + (Pb^3 + RTb^2 - ab) = 0 \quad (50)$$

Podemos observar el código en las siguientes líneas:

```
class pr_eos_elv:
    # Esta clase calcula el ELV de la mezcla binaria
    def __init__(self, mezcla, fracciones_molares_y, a_m_vapor = None,
b_m_vapor = None): #y1 guess
        self.mezcla = mezcla
        self.co2 = mezcla.componentes[0]
        self.h2o = mezcla.componentes[1]
        self.x1 = mezcla.fracciones_molares[0]
        self.x2 = mezcla.fracciones_molares[1]
        self.fracciones_molares_y = fracciones_molares_y
        self.T = mezcla.T
        self.P = mezcla.P
        self.sij = mezcla.sij
        self.sji = mezcla.sji
        self.a_m = mezcla.a_m
        self.b_m = mezcla.b_m
        self.A_m = mezcla.A_m
        self.B_m = mezcla.B_m
        self.mezcla_vapor = pr_eos_mezcla(
            self.mezcla.componentes,
            self.fracciones_molares_y,
            self.T,
            self.P,
            self.mezcla.sij, self.mezcla.sji,
            a_m = a_m_vapor if a_m_vapor is not None else None,
            b_m = b_m_vapor if b_m_vapor is not None else None
        )

    @property
    def fracciones(self):
        return self.fracciones_molares_y[0],
self.fracciones_molares_y[1]

    def calcular_vol(self, fase):
        m = self.P
        if fase == 'l':
            n = self.P*self.b_m - r*self.T
            s = -3*self.P*self.b_m**2 - 2*r*self.T*self.b_m + self.a_m
            t = self.P*self.b_m**3 + r*self.T*self.b_m**2 -
self.a_m*self.b_m
            sol = resolver_cubica(m, n, s, t, 'real')
            sol = sol[sol>self.b_m] # Para obtener soluciones
consistentes
```

```

        vol = np.min(sol)
        elif fase == 'v':
            n = self.mezcla_vapor.P*self.mezcla_vapor.b_m -
            r*self.mezcla_vapor.T
            s = -3*self.mezcla_vapor.P*self.mezcla_vapor.b_m**2 -
            2*r*self.mezcla_vapor.T*self.mezcla_vapor.b_m + self.mezcla_vapor.a_m
            t = self.mezcla_vapor.P*self.mezcla_vapor.b_m**3 +
            r*self.mezcla_vapor.T*self.mezcla_vapor.b_m**2 -
            self.mezcla_vapor.a_m*self.mezcla_vapor.b_m
            sol = resolver_cubica(m, n, s, t, 'real')
            sol = sol[sol>self.b_m]
            vol = np.max(sol)
        return vol

    def calcular_Z(self, fase):
        Z = self.P*self.calcular_vol(fase)/r/self.T
        return Z

    def calcular_lnphi(self, fase):
        if fase == 'l':
            mezcla_binaria = self.mezcla
        if fase == 'v':
            mezcla_binaria = self.mezcla_vapor

        z1 = mezcla_binaria.fracciones_molares[0]
        z2 = mezcla_binaria.fracciones_molares[1]

        a11 = mezcla_binaria.componentes[0].calcular_a(self.T)
        a22 = mezcla_binaria.componentes[1].calcular_a(self.T)
        a12 = np.sqrt(a11 * a22)*(1-mezcla_binaria.sij+\
            (mezcla_binaria.sij-
            mezcla_binaria.sji)*z1)
        a21 = np.sqrt(a22 * a11)*(1-mezcla_binaria.sji+\
            (mezcla_binaria.sji-
            mezcla_binaria.sij)*z2)

        sij = mezcla_binaria.sij
        sji = mezcla_binaria.sji

        lnphi_co2 =
        ((self.co2.calcular_b()/mezcla_binaria.b_m)*(self.calcular_Z(fase)-1))
        - np.log(self.calcular_Z(fase)- mezcla_binaria.B_m) -
        mezcla_binaria.A_m/(2*np.sqrt(2)*mezcla_binaria.B_m)*((2*z1*a11+z2*(a21
        +a12)+2*z1*z2**2*(sij-
        sji)*np.sqrt(self.co2.calcular_a(self.T)*self.h2o.calcular_a(self.T)))/
        mezcla_binaria.a_m -
        self.co2.calcular_b()/mezcla_binaria.b_m*np.log((self.calcular_Z(fase)
        +(1+np.sqrt(2))*mezcla_binaria.B_m)/(self.calcular_Z(fase)+(1-
        np.sqrt(2))*mezcla_binaria.B_m))

        lnphi_h2o =
        ((self.h2o.calcular_b()/mezcla_binaria.b_m)*(self.calcular_Z(fase)-1))
        - np.log(self.calcular_Z(fase)- mezcla_binaria.B_m) -
        mezcla_binaria.A_m/(2*np.sqrt(2)*mezcla_binaria.B_m)*((2*z2*a22+z1*(a12
        +a21)+2*z2*z1**2*(sji-
        sij)*np.sqrt(self.h2o.calcular_a(self.T)*self.co2.calcular_a(self.T)))/
        mezcla_binaria.a_m -

```

```

self.h2o.calcular_b()/mezcla_binaria.b_m)*np.log((self.calcular_Z(fase)
+(1+np.sqrt(2))*mezcla_binaria.B_m)/(self.calcular_Z(fase)+(1-
np.sqrt(2))*mezcla_binaria.B_m))

return [lnphi_co2, lnphi_h2o]

```

Nota. Creación propia

3.1.3.5 Determinación del equilibrio líquido vapor usando método ϕ - ϕ (fi-fi)

Se utilizó el método fi-fi propuesto por (Smith et al., 2022) tal y como lo vimos en la

Figura 3.

El código lo podemos ver líneas abajo:

```

def det_elv_PR_EOS(rango_x, y_1_vector, P_vector, T_vector, sij):
    datos = len(rango_x)

    max_iter = 50
    T_op = T_vector[0]

    x_1_datos = rango_x

    P_store = np.zeros(datos)
    x_1_store = np.zeros(datos)
    y_1_store = np.zeros(datos)
    y_1 = y_1_vector[0]
    P_ini = P_vector[0]

    for i, x_1 in enumerate(x_1_datos):

        x_2 = 1 - x_1
        P_old = 0
        iter_P = 0

        while abs(P_old - P_ini) > 1e-6 and iter_P < max_iter:
            mezcla = pr_eos_mezcla([co2, h2o], [x_1, x_2], T_op, P_ini,
            sij[i], sij[i])
            elv = pr_eos_elv(mezcla, [y_1, 1 - y_1])

            ln_phi_l1, ln_phi_l2 = elv.calcular_lnphi('l')
            y_1_old = 0
            iter_y = 0

            while abs(y_1_old - y_1) > 1e-6 and iter_y < max_iter:
                ln_phi_v1, ln_phi_v2 = elv.calcular_lnphi('v')

                K1 = np.exp(ln_phi_l1 - ln_phi_v1)
                K2 = np.exp(ln_phi_l2 - ln_phi_v2)
                suma_y = K1 * x_1 + K2 * x_2

                y_1_old = y_1
                y_1 = K1 * x_1 / suma_y
                iter_y += 1

            P_old = P_ini

```

```

        P_ini = P_old * suma_y
        if i > 0 and (P_old < P_store[i - 1]):
            print(f'Esta convergiendo a P menores, var P: {P_old -
P_store[i - 1]}')
            P_ini = P_store[i - 1]*1.01

        iter_P += 1

        if iter_P == max_iter:
            print(f"No se encontraron soluciones perfectas en iter
{iter_P}, suma y={suma_y}")

        #Más alla del punto critico se generarian valores identicos
        if abs(y_1 - x_1) < 1e-2:
            P_store[i] = np.nan
            x_1_store[i] = np.nan
            y_1_store[i] = np.nan
        else:
            P_store[i] = P_ini
            x_1_store[i] = x_1
            y_1_store[i] = y_1

        #Para evitar diferencia de presiones elevada entre puntos
consecutivos
        if i > 0 and (P_store[i] / P_store[i - 1]) > 2:
            P_store[i:] = np.nan
            x_1_store[i:] = np.nan
            y_1_store[i:] = np.nan
            break

    return np.array(P_store), np.array(x_1_store), np.array(y_1_store)

```

Nota. Creación propia

Es importante aclarar algunas partes del código, como los avisos de advertencia cuando la presión en la iteración “i-1” es mayor a la presión en la iteración “i”. Sabemos que al tener un loop anidado de fracciones molares dentro de un loop general de presiones se puede originar sobreestimaciones o subestimaciones de la presión inicial en la iteración “i” debido al valor de y_{i-1} ya que ambos están relacionados mediante la corrección de y_i que podemos ver en el algoritmo propuesto por (Koretsky, 2013) o en la Ecuación 51, una mala estimación inicial de la presión originaria que la gráfica tome soluciones no deseadas, es por ello que se condiciona al algoritmo a que la presión estimada inicial siempre suba, y el anuncio es para analizar si genera soluciones aceptables o no, que serán medidas mediante la suma de las fracciones molares de la fase vapor.

$$y_i = \frac{K_{i-1}x_{i-1}}{\sum_j y_j} \quad (51)$$

3.1.3.6 Determinación del equilibrio líquido vapor usando método ϕ - ϕ (fi-fi) con entrada manual de parámetros a_m^l y s_{ij}

El código es muy parecido al de la determinación del ELV, solo que en este nuevo código se obvia el cálculo del parámetro “a” de mezcla en fase líquida y el parámetro de interacción binaria “ s_{ij} ” y se toman valores discretos estimados por el usuario.

```
# Algoritmo de resolución del ELV
# El algoritmo nos da los valores después de resolver el ELV mediante
el método phi-phi

def det_elv_1BIP_aml(rango_x, y_1_vector, P_vector, T_vector, sij,
a_m):
    datos = len(rango_x)

    max_iter = 50
    T_op = T_vector[0]

    x_1_datos = rango_x

    P_store = np.zeros(datos)
    x_1_store = np.zeros(datos)
    y_1_store = np.zeros(datos)

    for i, x_1 in enumerate(x_1_datos):
        y_1 = y_1_vector[i]
        P_ini = P_vector[i]
        x_2 = 1 - x_1
        P_old = 0
        iter_P = 0

        while abs(P_old - P_ini) > 1e-6 and iter_P < max_iter:
            mezcla = pr_eos_mezcla([co2, h2o], [x_1, x_2], T_op, P_ini,
sij[i], sij[i], a_m[i])
            elv = pr_eos_elv(mezcla, [y_1, 1 - y_1])

            ln_phi_l1, ln_phi_l2 = elv.calcular_lnphi('l')
            y_1_old = 0
            iter_y = 0

            while abs(y_1_old - y_1) > 1e-6 and iter_y < max_iter:
                ln_phi_v1, ln_phi_v2 = elv.calcular_lnphi('v')

                K1 = np.exp(ln_phi_l1 - ln_phi_v1)
                K2 = np.exp(ln_phi_l2 - ln_phi_v2)
                suma_y = K1 * x_1 + K2 * x_2

                y_1_old = y_1
                y_1 = K1 * x_1 / suma_y
                iter_y += 1

            P_old = P_ini
```

```

        P_ini = P_old * suma_y
        if i > 0 and (P_old < P_store[i - 1]):
            print(f'Esta convergiendo a P menores, var P: {P_old -
P_store[i - 1]}')
            P_ini = P_store[i - 1]*1.01

        iter_P += 1

        if iter_P == max_iter:
            print(f"No se encontraron soluciones perfectas en iter
{iter_P}, suma y={suma_y}")

        #Más alla del punto critico se generarian valores identicos
        if abs(y_1 - x_1) < 1e-2:
            P_store[i] = np.nan
            x_1_store[i] = np.nan
            y_1_store[i] = np.nan
        else:
            P_store[i] = P_ini
            x_1_store[i] = x_1
            y_1_store[i] = y_1

        #Para evitar diferencia de presiones elevada entre puntos
consecutivos
        if i > 0 and (P_store[i] / P_store[i - 1]) > 2:
            P_store[i:] = np.nan
            x_1_store[i:] = np.nan
            y_1_store[i:] = np.nan
            break

    return np.array(P_store), np.array(x_1_store), np.array(y_1_store)

```

Nota. Creación propia

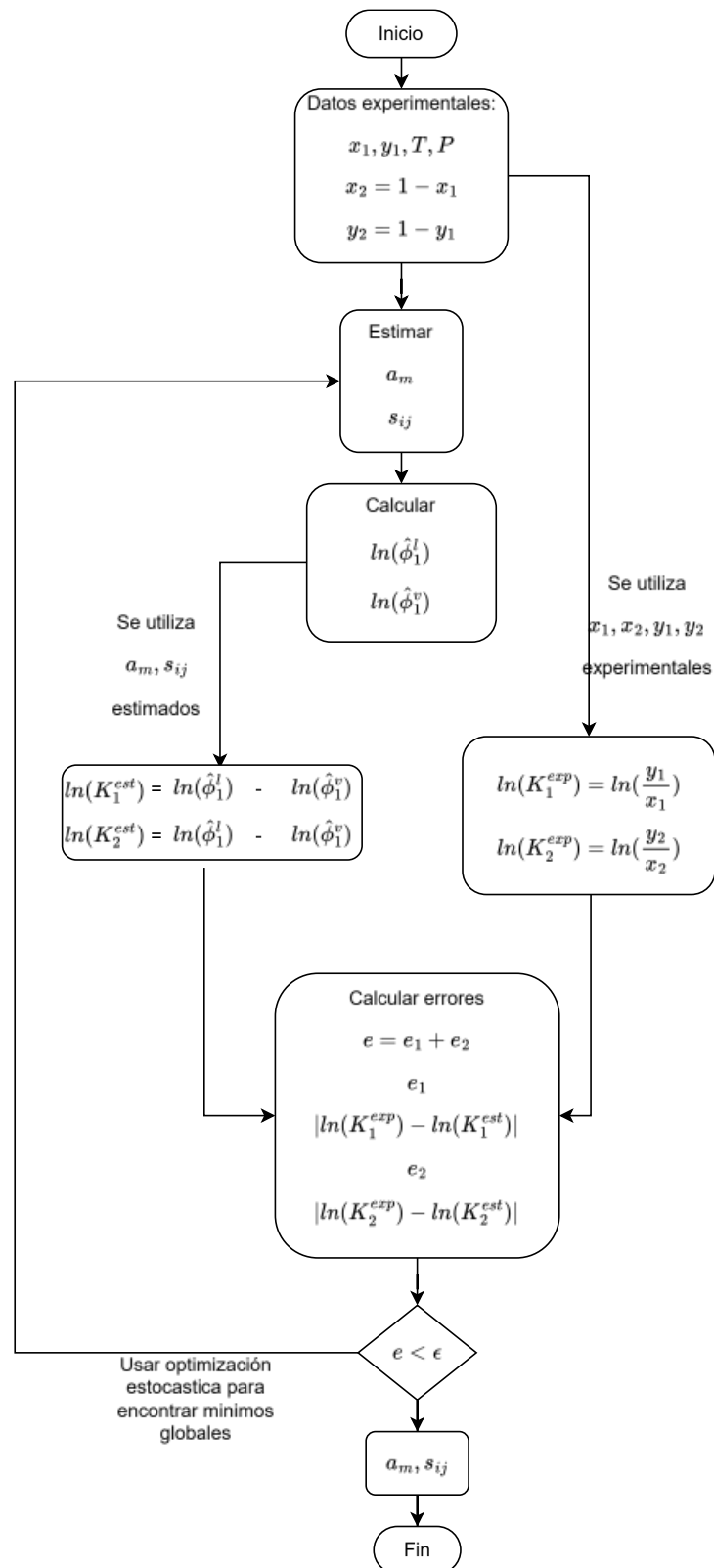
3.1.3.7 Nuevo algoritmo para encontrar los parámetros de interacción binaria y el parámetro “a” de mezcla en fase liquida utilizando data experimental de diagramas P_{xy}

Se desarrolló un nuevo algoritmo para encontrar el parámetro de interacción binaria (s_{ij}) y el parámetro “a” de mezcla en fase liquida (a_m^l) basándose en el coeficiente de distribución visto en la Ecuación 11, el método fi-fi de la Ecuación 41 y el método de optimización estocástica por recocido simulado.

El nuevo algoritmo se presenta en la Figura 17.

Figura 17

Nuevo algoritmo para encontrar los parámetros de interacción binaria y el parámetro “a” de mezcla en fase líquida



Para poder usar el nuevo algoritmo debemos definir una función de costo para minimizar el error. La función de costo según el nuevo algoritmo se observa líneas abajo.

```
# Placeholders
lnk1 = 1
lnk2 = 1
x1 = 0.5
x2 = 0.5
y1 = 0.5
y2 = 0.5
#####

# Probando con la regla de mezcla de van der Waals original
def funcion_costo_1BIP_aml(parametros):
    a_m, sij = parametros
    mezcla = pr_eos_mezcla([co2, h2o], [x1, x2], T, P, sij, sij, a_m)
    elv = pr_eos_elv(mezcla, [y1, y2])
    ln_coef_fug_1_v_exp, ln_coef_fug_2_v_exp = elv.calcular_lnphi('v')
    ln_coef_fug_1_l_exp, ln_coef_fug_2_l_exp = elv.calcular_lnphi('l')
    lnk1_exp = ln_coef_fug_1_l_exp - ln_coef_fug_1_v_exp
    lnk2_exp = ln_coef_fug_2_l_exp - ln_coef_fug_2_v_exp
    e = np.abs(lnk1_exp-lnk1)+np.abs(lnk2_exp-lnk2)
    return e
```

Nota. Creación propia

3.1.4 Definición de red neuronal para modelamiento del equilibrio líquido – vapor

Se estableció una red neuronal híbrida en serie tipo I, donde la red neuronal podrá predecir parámetros que luego serán usados por la EOS – PR.

Para la implementación de redes neuronales se plantea las siguientes consideraciones:

- Redes neuronales independientes, se plantea esta estructura para que cada red neuronal pueda capturar las características de la corrección del parámetro “ a_m ” (Δa_m^l) y de “ s_{ij} ”, así evitaremos interferencias entre los parámetros, además se lograría ajustar de manera independiente la arquitectura e hiperparámetros de la red neuronal.
- Una red neuronal predecirá la corrección del parámetro “ a_m ” con respecto al calculado mediante la regla de mezcla de Vander Waals (Δa_m^l), mientras que otra red neuronal predecirá el parámetro de interacción binaria (s_{ij}).
- Las entradas para ambas redes neuronales serán las mismas entradas que tiene un problema clásico de determinación de ELV: “ x_{CO_2} ” y “ T ”, mientras

que la salida para la primera red neuronal será el parámetro de interacción binaria s_{ij} , y en la segunda red neuronal, las entradas serán las mismas que la primera, pero la salida será Δa_m^l .

3.1.5 Entrenamiento de red neuronal

Para entrenar la red neuronal, primero se debe tomar una parte de la data para entrenar la red neuronal y otra parte que luego servirá para validar nuestras predicciones. También se necesita de una función de costo que servirá como función objetivo a minimizar, en la presente tesis se usa una función de costo que está basada en el error cuadrático medio como lo podemos ver en la Ecuación 52. También se usa la función de costo para analizar la “pérdida” en el entrenamiento y en la validación, ya que se puede evaluar el error cuadrático medio en cada actualización de hiperparámetros.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (52)$$

Donde:

y_i : Son los valores verdaderos de Δa_m^l y s_{ij}

$\hat{y}_i$: Son los valores predichos para Δa_m^l y s_{ij} por la red neuronal

n : número total de puntos analizados en el conjunto de datos.

Si tenemos dos redes neuronales distintas, en donde se predecirá la corrección del parámetro “ a_m^l ” (Δa_m^l) y el parámetro “ s_{ij} ”, tendríamos las funciones de costo de las Ecuaciones 53 y 54.

$$MSE = \frac{1}{n} \sum_{i=1}^n (\Delta a_{m,i}^l - \widehat{\Delta a}_{m,i}^l)^2 \quad (53)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (s_{ij,i} - \hat{s}_{ij,i})^2 \quad (54)$$

Al modelar una red neuronal se utiliza algún optimizador que se encargará de actualizar los hiperparámetros del modelo basándose en los gradientes calculados por la función de pérdida. Usaremos Adam, un método que combina el descenso de gradiente con una tasa de aprendizaje adaptativa, para profundizar en los aspectos teóricos de Adam podemos revisar el artículo desarrollado por Kingma & Ba (2014). El motivo principal por el que se eligió Adam fue por su flexibilidad en el ajuste de tasa de aprendizaje para cada hiperparámetro en la red neuronal, por su eficiencia computacional y porque requiere de menos ajuste manual.

3.1.6 Método de validación

Debido a la falta de datos experimentales del ELV de CO₂-H₂O en la literatura, se validarán las predicciones del modelo híbrido neuronal con los datos experimentales parametrizados obtenidos de Tödheide & Franck, (1963).

Diversas investigaciones de implementación de redes neuronales al ELV como la de Argatov & Kocherbitov (2019) han utilizado el MAPE (Error absoluto medio) como medida de error en la predicción del ELV, mientras que otros autores como Vaferi et al. (2013) han usado el MAPE y el MSE (Error cuadrático medio) como medida de error en la predicción del ELV.

Es por ello que para la presente investigación se usarán MAPE y MSE como medidas de error de las predicciones de la fracción molar de CO₂ en fase vapor y la presión de equilibrio. Las medidas de error se pueden ver en las Ecuaciones 53, 54, 55, 56, 57 y 58.

$$MSE_y = \frac{1}{n} \sum_{i=1}^n (y_{CO_2}^{exp} - y_{CO_2}^{calc})^2 \quad (53)$$

$$MSE_p = \frac{1}{n} \sum_{i=1}^n (p^{exp} - p^{calc})^2 \quad (54)$$

$$MSE_{P,y} = \frac{1}{n} \sum_{i=1}^n [(P^{exp} - P^{calc})^2 + (y_{CO_2}^{exp} - y_{CO_2}^{calc})^2] \quad (55)$$

$$MAPE_y = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_{CO_2}^{exp} - y_{CO_2}^{calc}}{y_{CO_2}^{exp}} \right| \quad (56)$$

$$MAPE_p = \frac{1}{n} \sum_{i=1}^n \left| \frac{P^{exp} - P^{calc}}{P^{exp}} \right| \quad (57)$$

$$MAPE_{P,y} = \frac{MAPE_p + MAPE_y}{2} \quad (58)$$

3.2 Resultados

3.2.1 Obtención de data experimental

La Tabla 2 muestra los resultados experimentales del ELV de la mezcla $CO_2 - H_2O$ publicados por Tödheide & Franck (1963).

Tabla 2
Data experimental del ELV de $CO_2 - H_2O$

T(°C)	267		268		270		275		300		350	
P(bar)	x CO_2	y CO_2	x CO_2	y CO_2	x CO_2	y CO_2	x CO_2	y CO_2	x CO_2	y CO_2	x CO_2	y CO_2
200	0.026	0.567	0.026	0.559	0.026	0.547	0.025	0.519	0.023	0.352	0.008	0.055
250											0.026	0.012
300			0.048	0.620	0.049	0.614	0.049	0.591	0.049	0.454	0.051	0.162
325											0.077	0.160
400			0.064	0.644	0.066	0.635	0.070	0.612	0.079	0.480		

500	0.082	0.650	0.083	0.642	0.087	0.631	0.092	0.611	0.125	0.460
600			0.095	0.638	0.102	0.628	0.114	0.595	0.225	0.335
700			0.118	0.622	0.124	0.607	0.137	0.572		
800			0.138	0.603	0.148	0.585	0.163	0.541		
900			0.160	0.586	0.170	0.565	0.195	0.507		
1000	0.170	0.575	0.179	0.569	0.190	0.545	0.241	0.455		
1100			0.197	0.556	0.207	0.525				
1200			0.211	0.547	0.226	0.503				
1300			0.228	0.530	0.250	0.466				
1400			0.248	0.518						
1500	0.240	0.526	0.278	0.497						

Nota: Adaptado de Tödheide & Franck (1963)

3.2.2 Parametrización de data experimental

Se utilizó splines paramétricos para aumentar la data y también se digitalizó algunos puntos de las gráficas experimentales que desarrolló Tödheide & Franck (1963).

3.2.2.1 Data aumentada para T = 267°C

En la Tabla 3 se tiene la data experimental y digitalizada.

Tabla 3

Data experimental del ELV de CO₂ – H₂O a 267°C

T(°C)		267		
P(bar)	x CO ₂	y CO ₂	Método de obtención	
200	0.026	0.567	Tabla 2	
500	0.082	0.650	Tabla 2	
1000	0.170	0.575	Tabla 2	
1500	0.240	0.526	Tabla 2	
1650	0.277	0.509	Digitalización	
1850	0.363	0.438	Digitalización	

Posteriormente se utilizó esta data para crear curvas suavizadas del ELV usando un spline paramétricos en Python, para la elección de los parámetros se consideró que la curva sea suave, tenga la dirección que sugiere Tödheide & Franck (1963) y que pueda pasar por la mayor cantidad de datos experimentales vistos en la Tabla 3.

Los parámetros que se seleccionaron para el spline a 267°C fueron los siguiente:

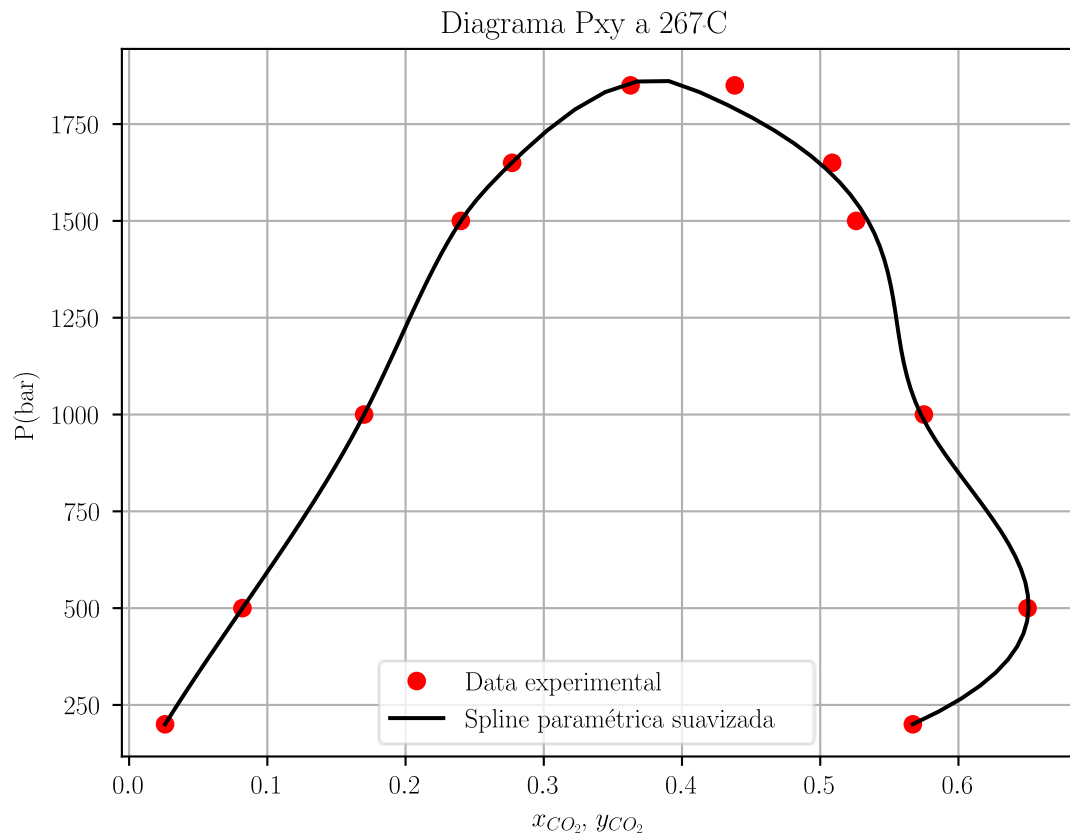
- Degree (Grado del Spline): 3

- Smoothing Condition (Suavizado): 3

La gráfica suavizada y los puntos experimentales se pueden observar en la Figura 18.

Figura 18

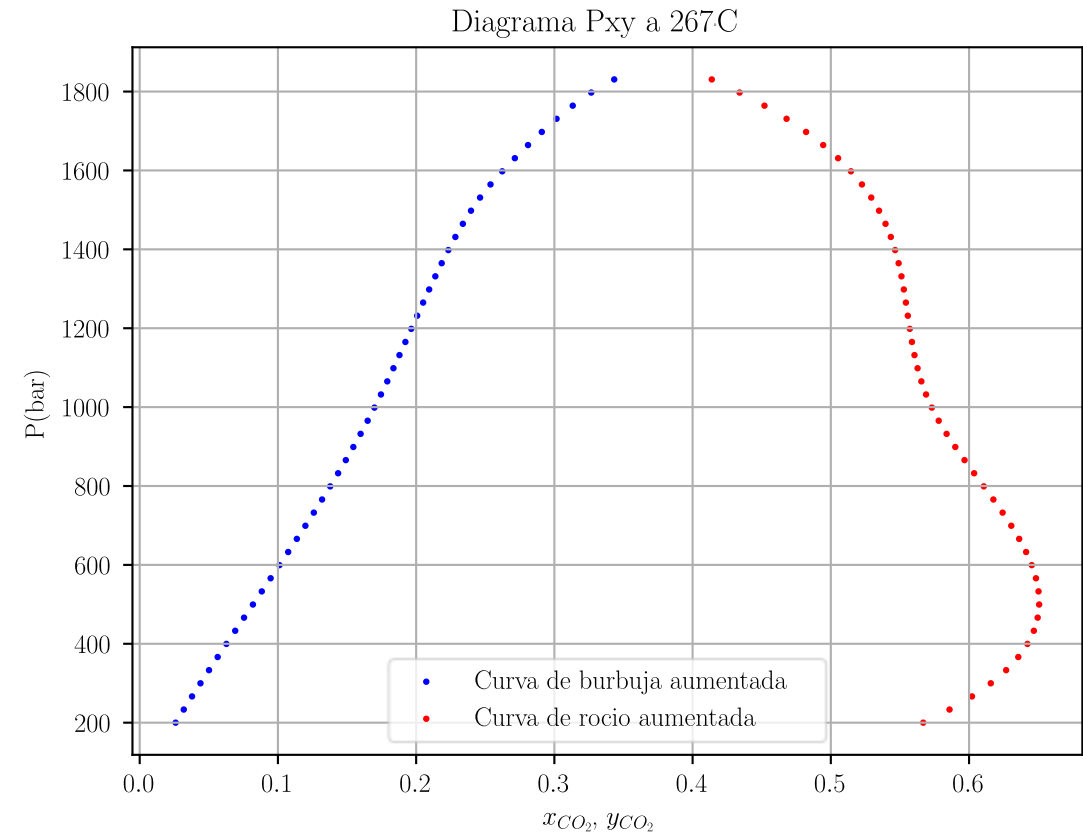
Isoterma suavizada a 267°C



De esta curva suavizada se generarán 100 puntos a la misma presión, 50 puntos en la curva de rocío y 50 puntos en la curva de burbuja, los puntos aumentados los podemos ver en el Anexo 1, y en la Figura 19 podemos ver a los 100 puntos gráficamente.

Figura 19

Datos aumentados a 267°C.



3.2.2.2 Data aumentada para T = 268°C

En la Tabla 4 se tiene la data experimental y digitalizada.

Tabla 4
Data experimental del ELV de CO₂ – H₂O a 268°C

T(°C)				268
P(bar)	x CO ₂	y CO ₂	Método de obtención	
200	0.026	0.559	Tabla 2	
300	0.048	0.620	Tabla 2	
400	0.064	0.644	Tabla 2	
500	0.083	0.642	Tabla 2	
600	0.095	0.638	Tabla 2	
700	0.118	0.622	Tabla 2	
800	0.138	0.603	Tabla 2	
900	0.160	0.586	Tabla 2	
1000	0.179	0.569	Tabla 2	
1100	0.197	0.556	Tabla 2	
1200	0.211	0.547	Tabla 2	
1300	0.228	0.530	Tabla 2	
1400	0.248	0.518	Tabla 2	
1500	0.278	0.497	Tabla 2	
1530	0.306	0.477	Digitalización	
1550	0.316	0.467	Digitalización	
1570	0.329	0.452	Digitalización	

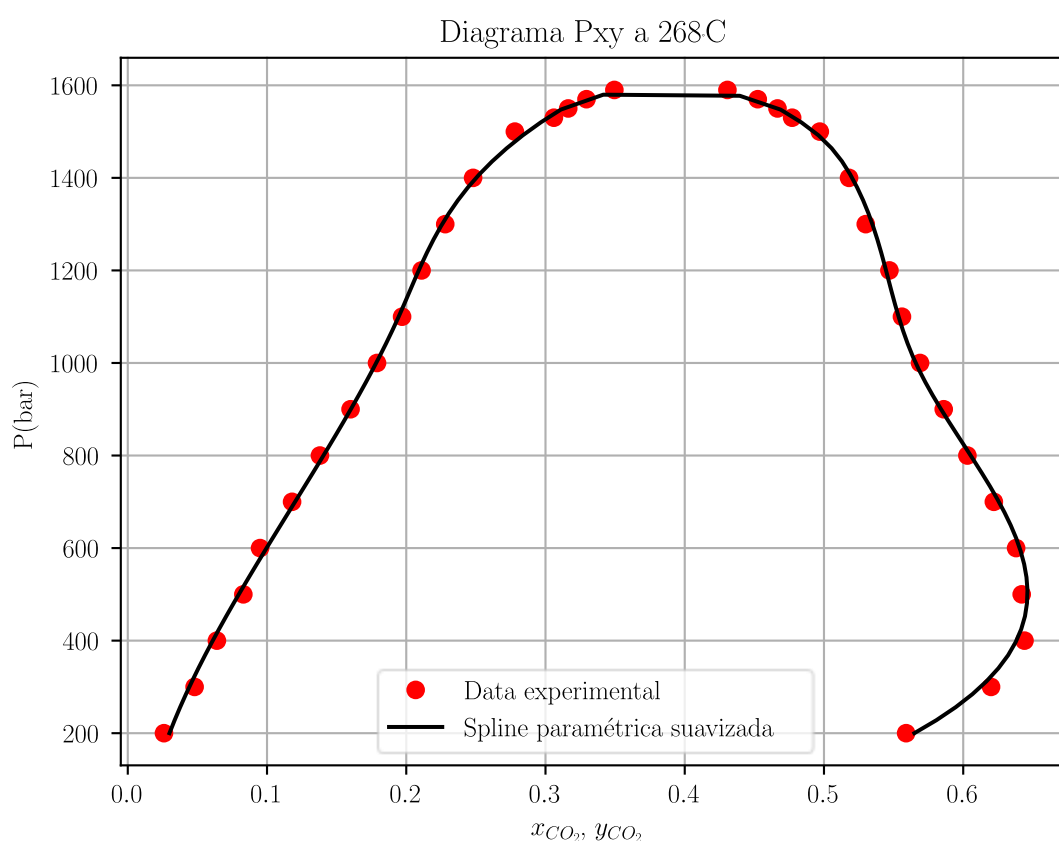
Los parámetros que se seleccionaron para el spline a 268°C fueron los siguientes:

- Degree (Grado del Spline): 3
- Smoothing Condition (Suavizado): 3

La gráfica suavizada y los puntos experimentales se pueden observar en la Figura 20.

Figura 20

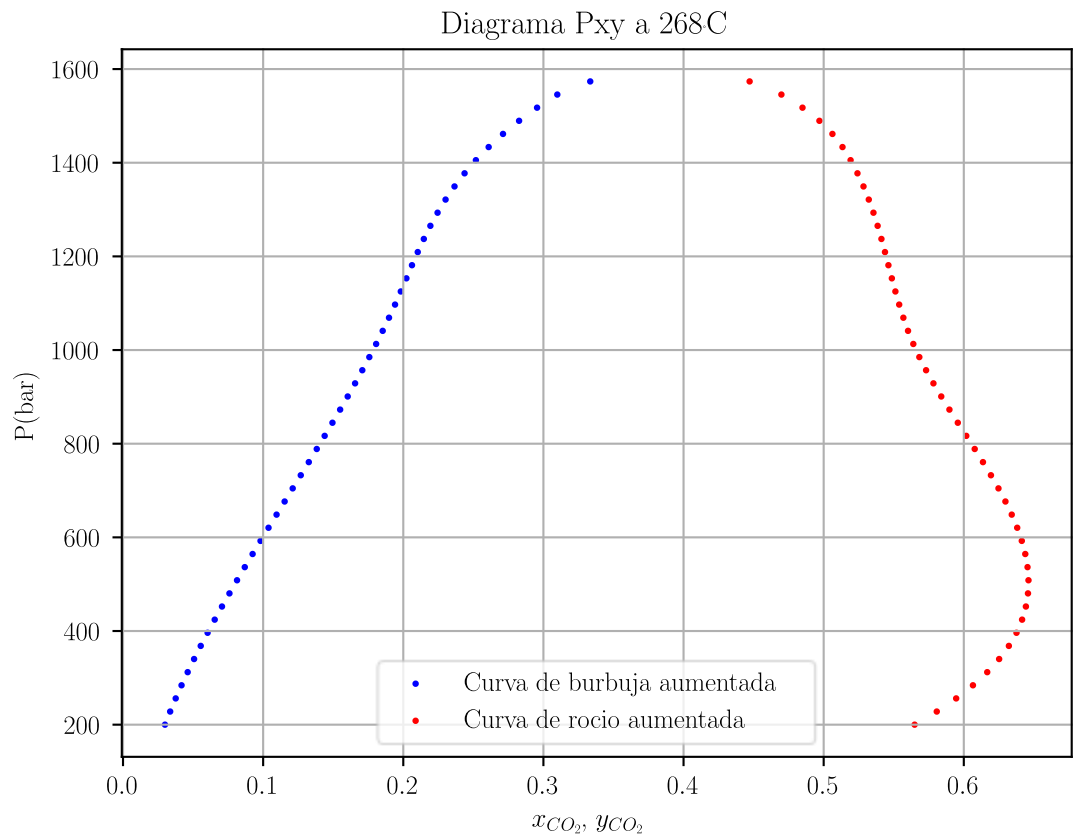
Isoterma suavizada a 268°C



De esta curva suavizada se generarán 100 puntos a la misma presión, 50 puntos en la curva de rocío y 50 puntos en la curva de burbuja, los puntos aumentados los podemos ver en el Anexo 2, y en la Figura 21 podemos ver a los 100 puntos gráficamente.

Figura 21

Datos aumentados a 268°C



3.2.2.3 Data aumentada para T = 270°C

En la Tabla 5 se tiene la data experimental y digitalizada.

Tabla 5

Data experimental del ELV de CO₂ – H₂O a 270°C

T(°C)		270		
P(bar)	x CO ₂	y CO ₂	Método de obtención	
200	0.026	0.547	Tabla 2	
300	0.049	0.614	Tabla 2	
400	0.066	0.635	Tabla 2	
500	0.087	0.631	Tabla 2	
600	0.102	0.628	Tabla 2	
700	0.124	0.607	Tabla 2	
800	0.148	0.585	Tabla 2	
900	0.170	0.565	Tabla 2	
1000	0.190	0.545	Tabla 2	
1100	0.207	0.525	Tabla 2	
1200	0.226	0.503	Tabla 2	
1300	0.250	0.466	Tabla 2	
1330	0.286	0.433	Digitalización	
1340	0.295	0.421	Digitalización	
1350	0.309	0.405	Digitalización	
1360	0.327	0.381	Digitalización	

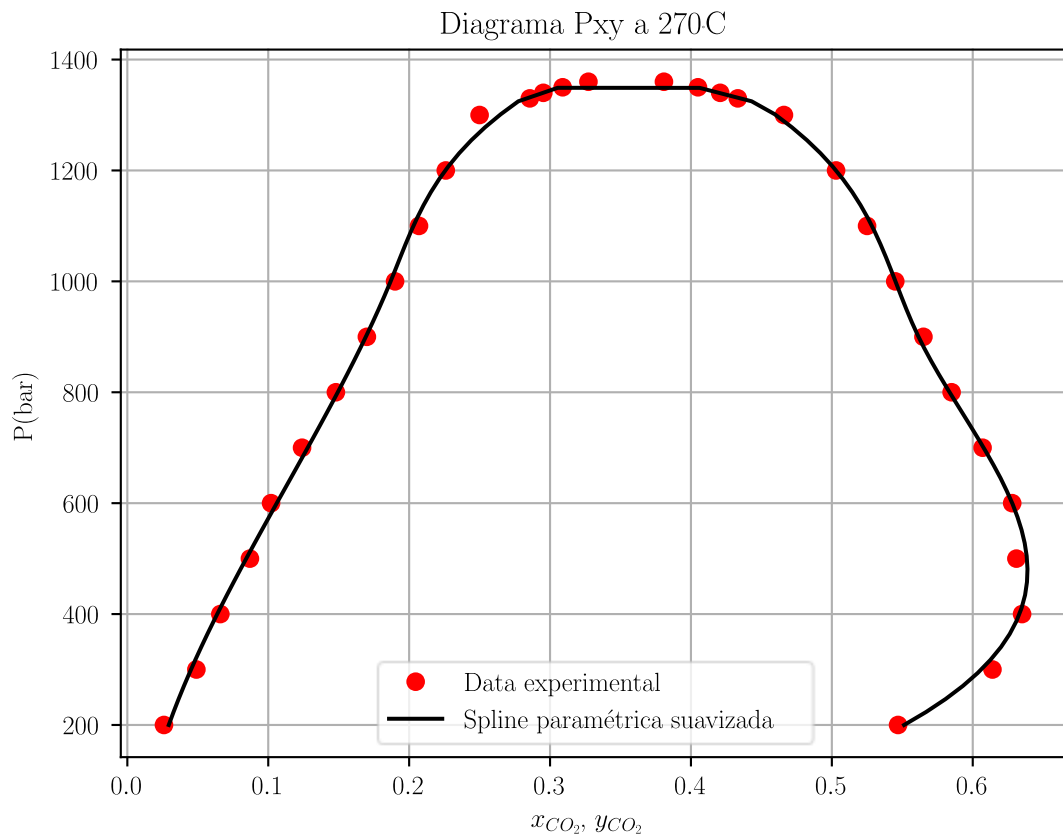
Los parámetros que se seleccionaron para el spline a 270°C fueron los siguientes:

- Degree (Grado del Spline): 3
- Smoothing Condition (Suavizado): 3

La gráfica suavizada y los puntos experimentales se pueden observar en la Figura 22.

Figura 22

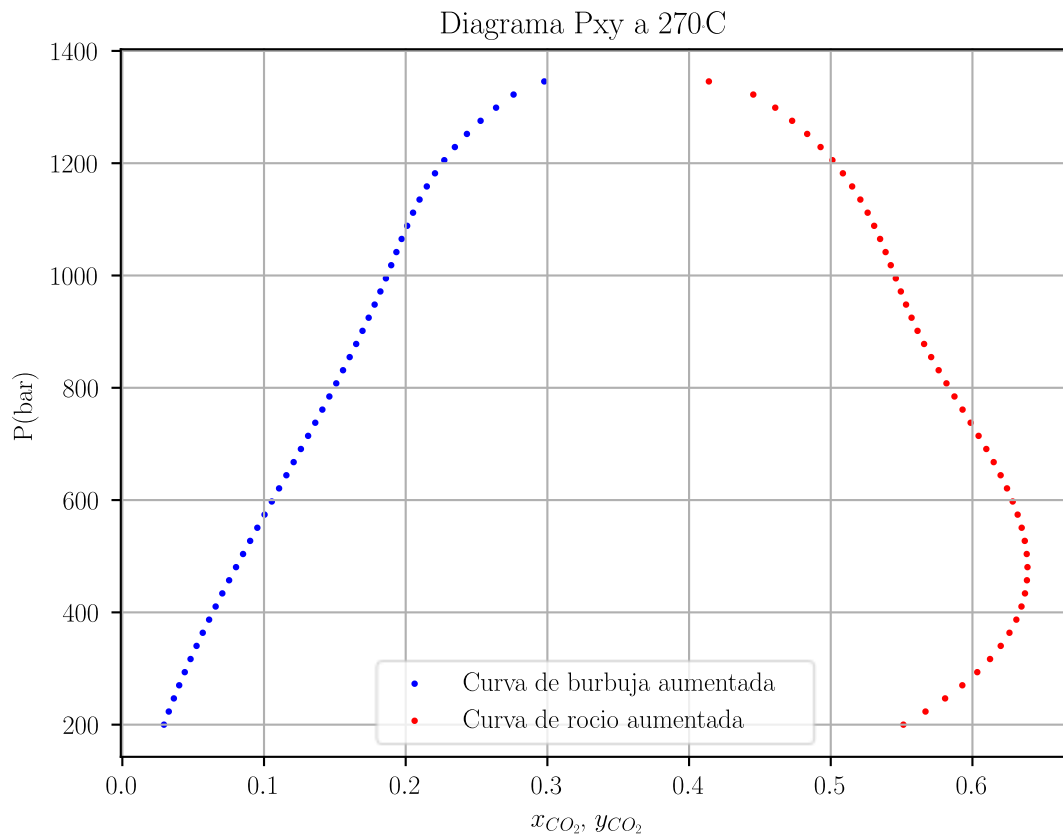
Isoterma suavizada a 270°C



De esta curva suavizada se generarán 100 puntos a la misma presión, 50 puntos en la curva de rocío y 50 puntos en la curva de burbuja, los puntos aumentados los podemos ver en el Anexo 3, y en la Figura 23 podemos ver a los 100 puntos gráficamente.

Figura 23

Datos aumentados a 270°C



3.2.2.4 Data aumentada para T = 275°C

En la Tabla 6 se tiene la data experimental y digitalizada.

Tabla 6

Data experimental del ELV de CO₂ – H₂O a 275°C

T(°C)				275
P(bar)	x CO ₂	y CO ₂	Método de obtención	
200	0.025	0.519	Tabla 2	
300	0.049	0.591	Tabla 2	
400	0.070	0.612	Tabla 2	
500	0.092	0.611	Tabla 2	
600	0.114	0.595	Tabla 2	
700	0.137	0.572	Tabla 2	
800	0.163	0.541	Tabla 2	
900	0.195	0.507	Tabla 2	
1000	0.241	0.455	Tabla 2	
1010	0.252	0.444	Digitalización	
1040	0.276	0.427	Digitalización	
1080	0.347	0.347	Digitalización	

Los parámetros que se seleccionaron para el spline a 275°C fueron los siguiente:

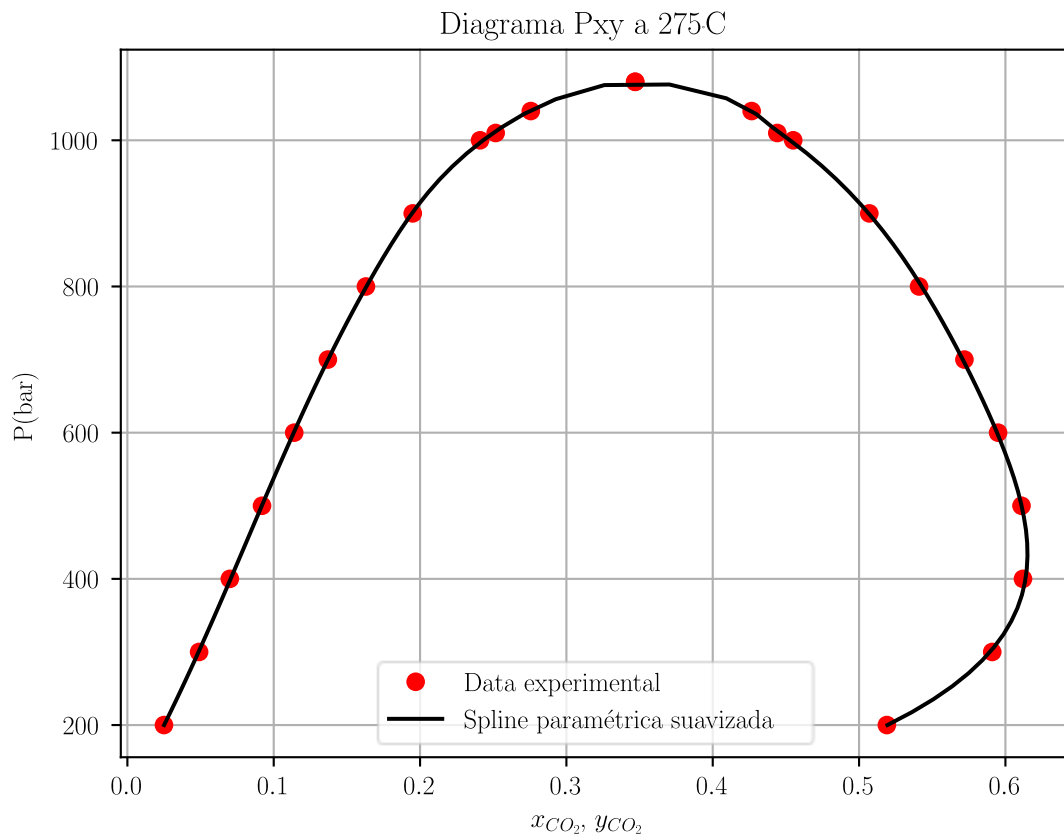
- Degree (Grado del Spline): 4

- Smoothing Condition (Suavizado): 3

La gráfica suavizada y los puntos experimentales se pueden observar en la Figura 24.

Figura 24

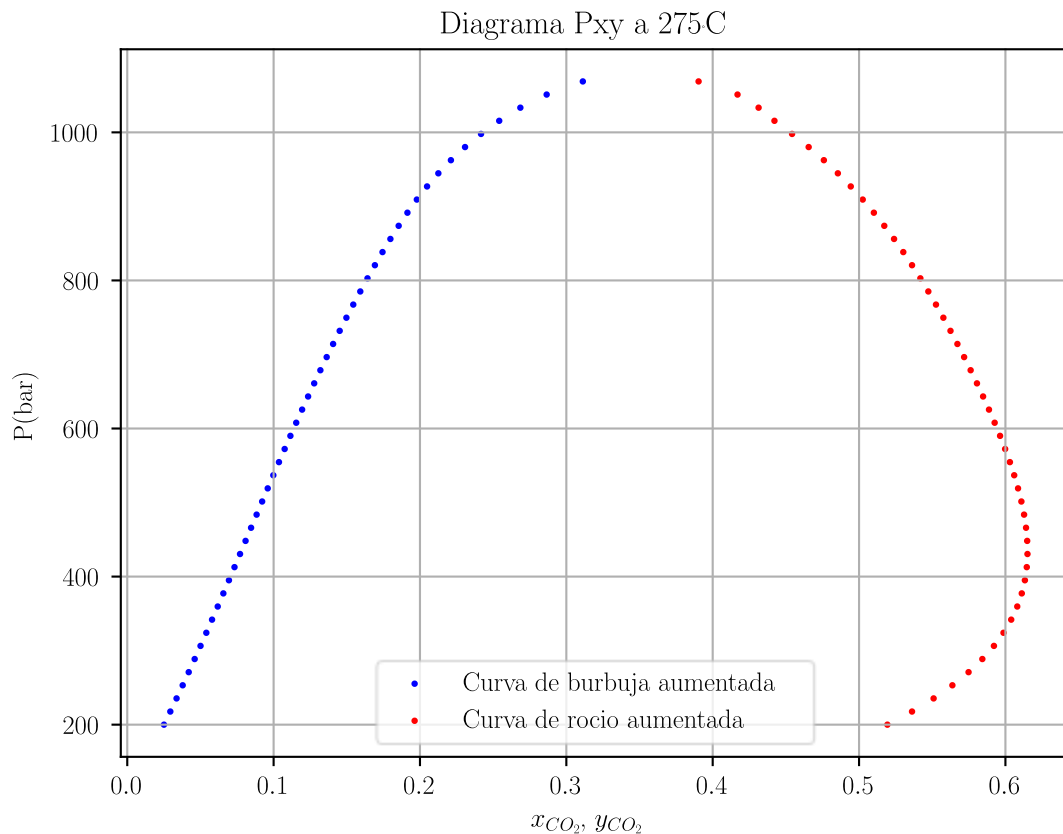
Isoterma suavizada a 275°C



De esta curva suavizada se generarán 100 puntos a la misma presión, 50 puntos en la curva de rocío y 50 puntos en la curva de burbuja, los puntos aumentados los podemos ver en el Anexo 4, y en la Figura 25 podemos ver a los 100 puntos gráficamente.

Figura 25

Datos aumentados a 275°C



3.2.2.5 Data aumentada para T = 300°C

En la Tabla 7 se tiene la data experimental y digitalizada.

Tabla 7

Data experimental del ELV de CO₂ – H₂O a 300°C

T(°C)				300
P(bar)	x CO ₂	y CO ₂	Método de obtención	
200	0.023	0.352	Tabla 2	
300	0.049	0.454	Tabla 2	
400	0.079	0.480	Tabla 2	
500	0.125	0.460	Tabla 2	
570	0.195	0.373	Digitalización	
600	0.225	0.335	Tabla 2	

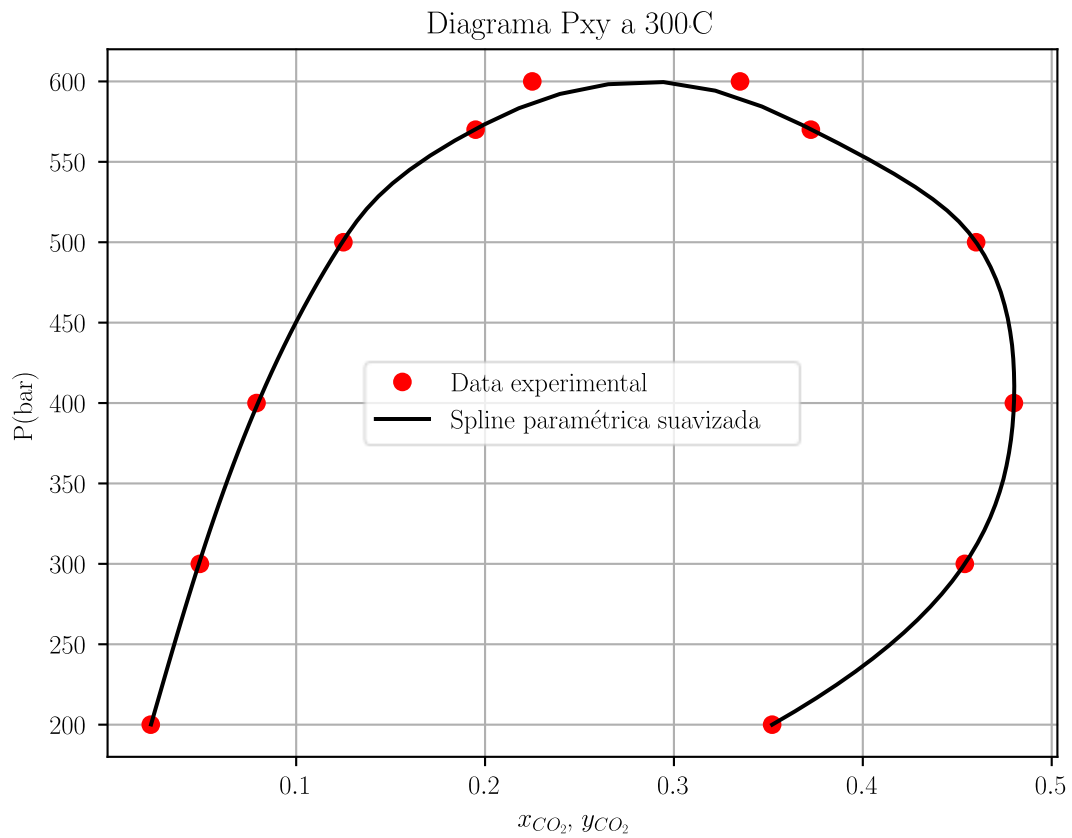
Los parámetros que se seleccionaron para el spline a 300°C fueron los siguiente:

- Degree (Grado del Spline): 3
- Smoothing Condition (Suavizado): 4

La gráfica suavizada y los puntos experimentales se pueden observar en la Figura 26.

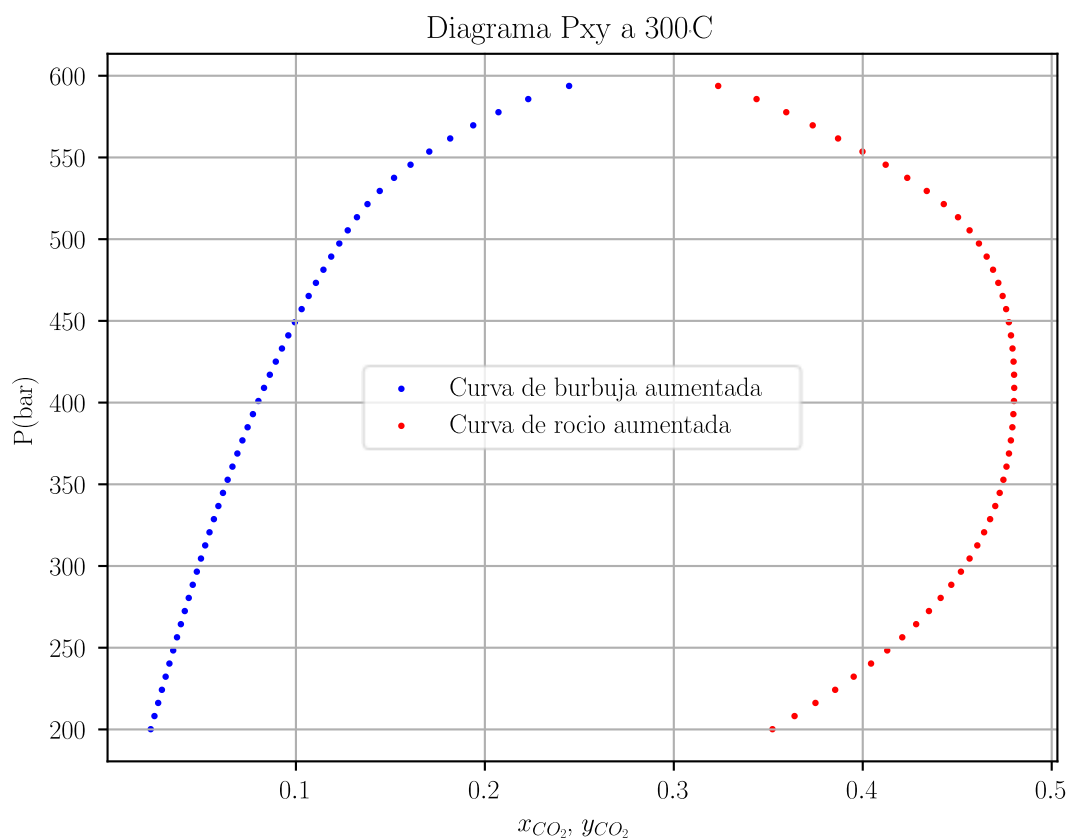
Figura 26

Isoterma suavizada a 300°C



De esta curva suavizada se generarán 100 puntos a la misma presión, 50 puntos en la curva de rocío y 50 puntos en la curva de burbuja, los puntos aumentados los podemos ver en el Anexo 5, y en la Figura 27 podemos ver a los 100 puntos gráficamente.

Figura 27 Datos aumentados a 300°C



3.2.2.6 Data aumentada para T = 350°C

En la Tabla 8 se tiene la data experimental y digitalizada.

Tabla 8

Data experimental del ELV de $CO_2 - H_2O$ a 350°C

T(°C)		350		
P(bar)	x CO ₂	y CO ₂	Método de obtención	
200	0.008	0.055	Tabla 2	
230	0.019	0.094	Digitalización	
250	0.026	0.012	Tabla 2	
270	0.033	0.146	Digitalización	
300	0.051	0.162	Tabla 2	
325	0.077	0.160	Tabla 2	
333	0.125	0.125	Tabla 2	

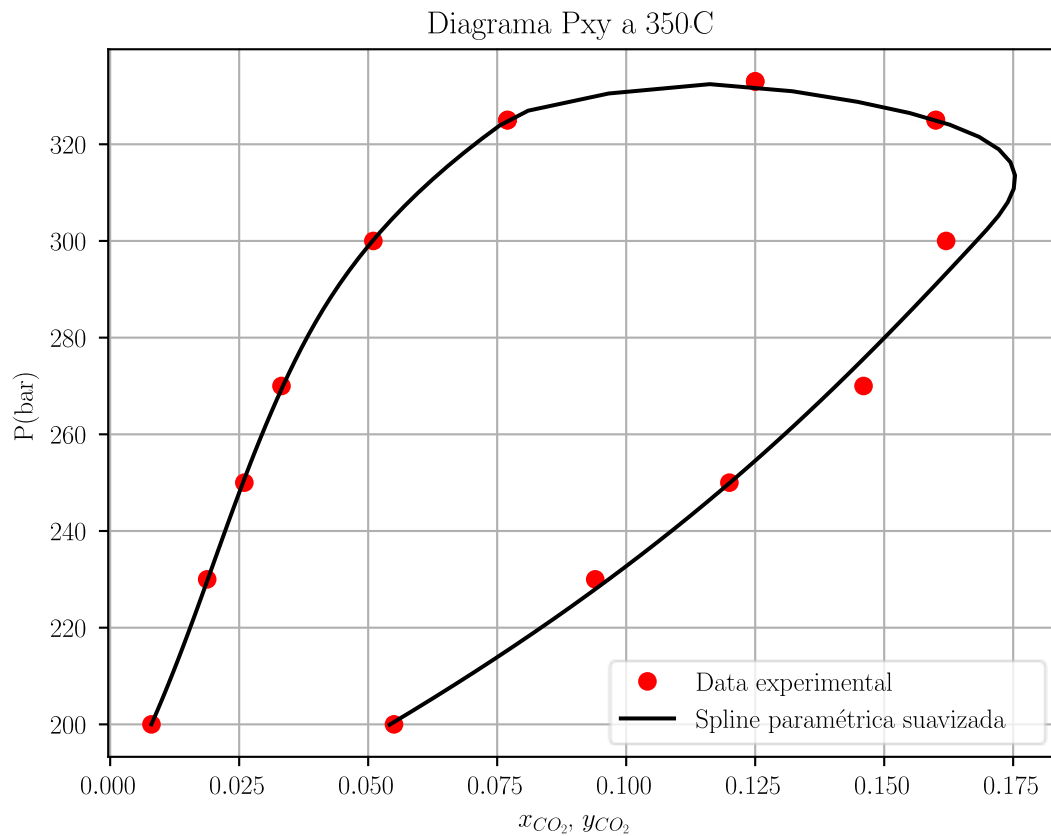
Los parámetros que se seleccionaron para el spline a 350°C fueron los siguiente:

- Degree (Grado del Spline): 3
- Smoothing Condition (Suavizado): 3

La gráfica suavizada y los puntos experimentales se pueden observar en la Figura 28.

Figura 28

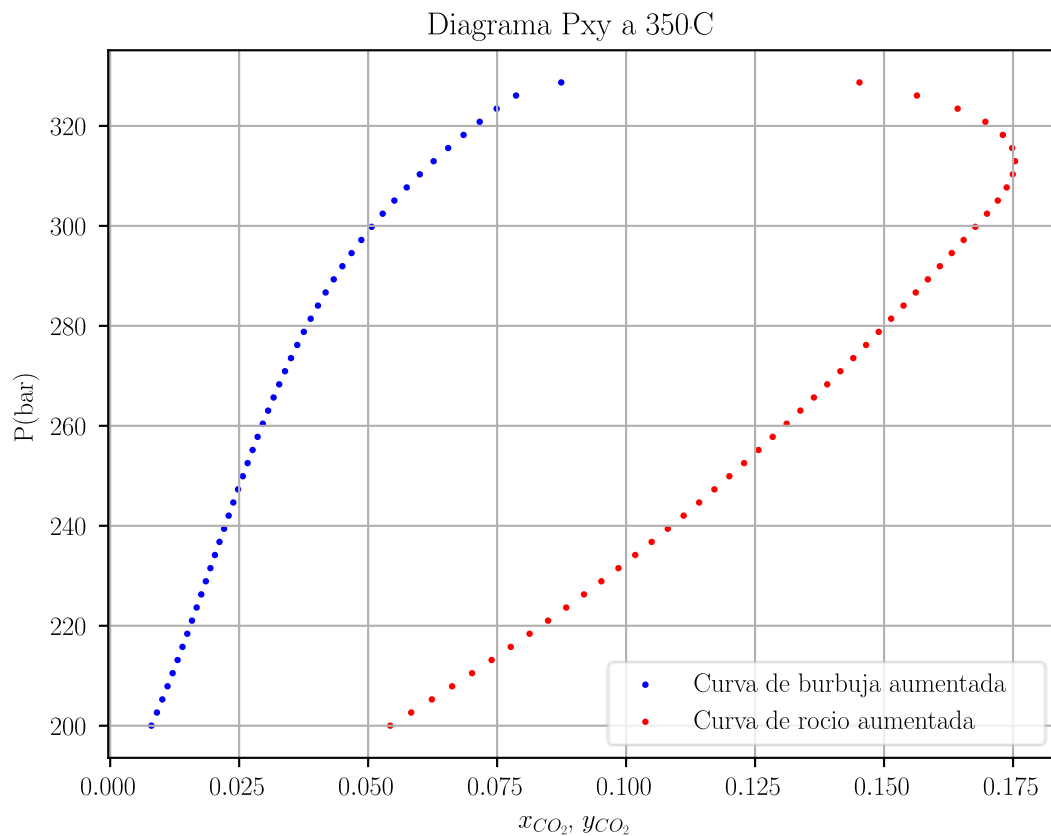
Isoterma suavizada a 350°C



De esta curva suavizada se generarán 100 puntos a la misma presión, 50 puntos en la curva de rocío y 50 puntos en la curva de burbuja, los puntos aumentados los podemos ver en el Anexo 6, y en la Figura 29 podemos ver a los 100 puntos gráficamente.

Figura 29

Datos aumentados a 350°C



3.2.3 Solución del ELV usando PR-EOS Original

Se usó la EOS – PR junto con el método fi-fi para determinar las curvas de rocío y burbuja del ELV de la mezcla $CO_2 - H_2O$. Como ejemplo aplicativo usaremos una temperatura de 267°C e iremos variando los parámetros de interacción binaria s_{ij} .

Inputs:

```
sij_values = np.full_like(x1_data_267, 0.05)
mezcla_pr_eos_267 = det_elv_PR_EOS(x1_data_267, np.array([0.57]),
np.array([200]), np.array([267+273.15]), sij_values)
```

Nota. Creación propia

Outputs:

```
No se encontraron soluciones perfectas en iter 50, suma
y=1.0000000062961978
No se encontraron soluciones perfectas en iter 50, suma
y=1.0000000331036276
```

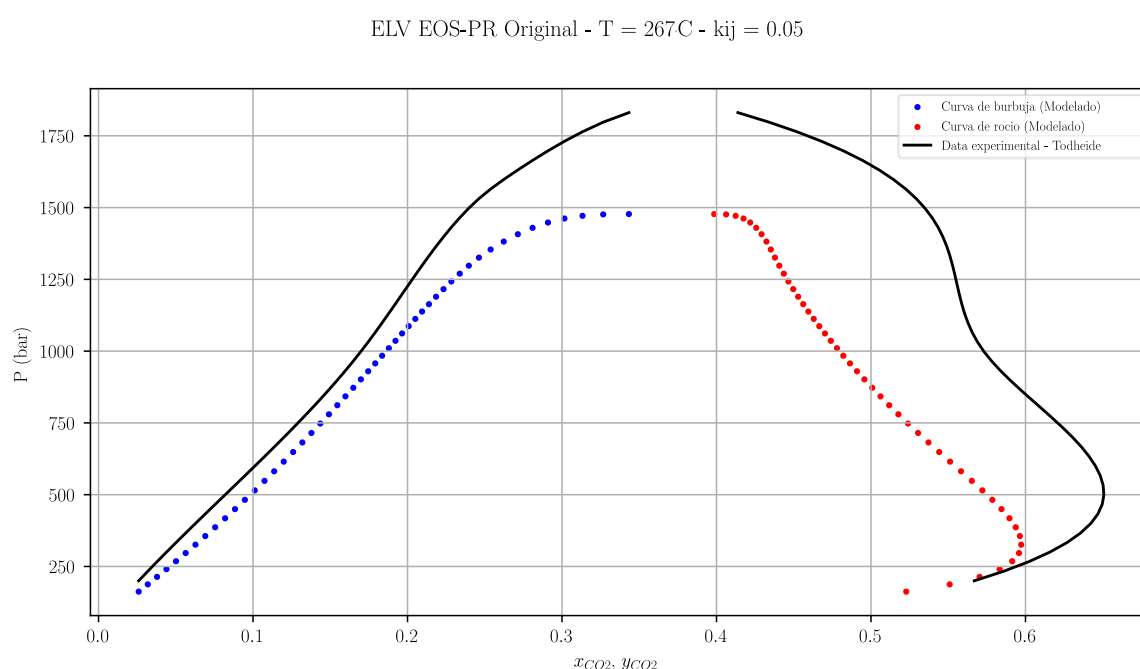
No se encontraron soluciones perfectas en iter 50, suma
y=1.0000001387367394
No se encontraron soluciones perfectas en iter 50, suma
y=1.000000475772107
No se encontraron soluciones perfectas en iter 50, suma
y=1.0000013673742383
No se encontraron soluciones perfectas en iter 50, suma
y=1.0000033655767417
No se encontraron soluciones perfectas en iter 50, suma
y=1.00000723367599
No se encontraron soluciones perfectas en iter 50, suma
y=1.0000138142846167
No se encontraron soluciones perfectas en iter 50, suma
y=1.0000238059511406
No se encontraron soluciones perfectas en iter 50, suma
y=1.0000375323735513
No se encontraron soluciones perfectas en iter 50, suma
y=1.0000548020718405
No se encontraron soluciones perfectas en iter 50, suma
y=1.0000749128865811
No se encontraron soluciones perfectas en iter 50, suma
y=1.000096787600139
No se encontraron soluciones perfectas en iter 50, suma
y=1.0001191780516994
No se encontraron soluciones perfectas en iter 50, suma
y=1.0001408662943978
No se encontraron soluciones perfectas en iter 50, suma
y=1.0001608139112566
No se encontraron soluciones perfectas en iter 50, suma
y=1.0001782425861054
No se encontraron soluciones perfectas en iter 50, suma
y=1.0001929011612642
No se encontraron soluciones perfectas en iter 50, suma
y=1.0002057521887677
No se encontraron soluciones perfectas en iter 50, suma
y=1.000217395879017
No se encontraron soluciones perfectas en iter 50, suma
y=1.000228154440892
No se encontraron soluciones perfectas en iter 50, suma
y=1.0002383060496824
No se encontraron soluciones perfectas en iter 50, suma
y=1.000248100933239
No se encontraron soluciones perfectas en iter 50, suma
y=1.0002577592046542
No se encontraron soluciones perfectas en iter 50, suma
y=1.0002674673638894
No se encontraron soluciones perfectas en iter 50, suma
y=1.0002773735951829
No se encontraron soluciones perfectas en iter 50, suma
y=1.0002875801116904
No se encontraron soluciones perfectas en iter 50, suma
y=1.0002981301294482
No se encontraron soluciones perfectas en iter 50, suma
y=1.0003089859306629
No se encontraron soluciones perfectas en iter 50, suma
y=1.0003199922668617

Nota. Creación propia

Como podemos ver, existen avisos de advertencia ya que la suma de las fracciones molares en fase vapor tiene una desviación de la unidad de 0.0003 como máximo, para la presente investigación se considerará aceptables a desviaciones menores al 1% o 0.01 para el algoritmo de solución del ELV. La solución del ELV usado EOS-PR la podemos analizar gráficamente en un diagrama P_{xy} , tal como lo vemos en la Figura 30, y compararla con la data experimental aumentada.

Figura 30

Comparación de EOS-PR con data experimental a 267°C y $k_{ij} = 0.05$

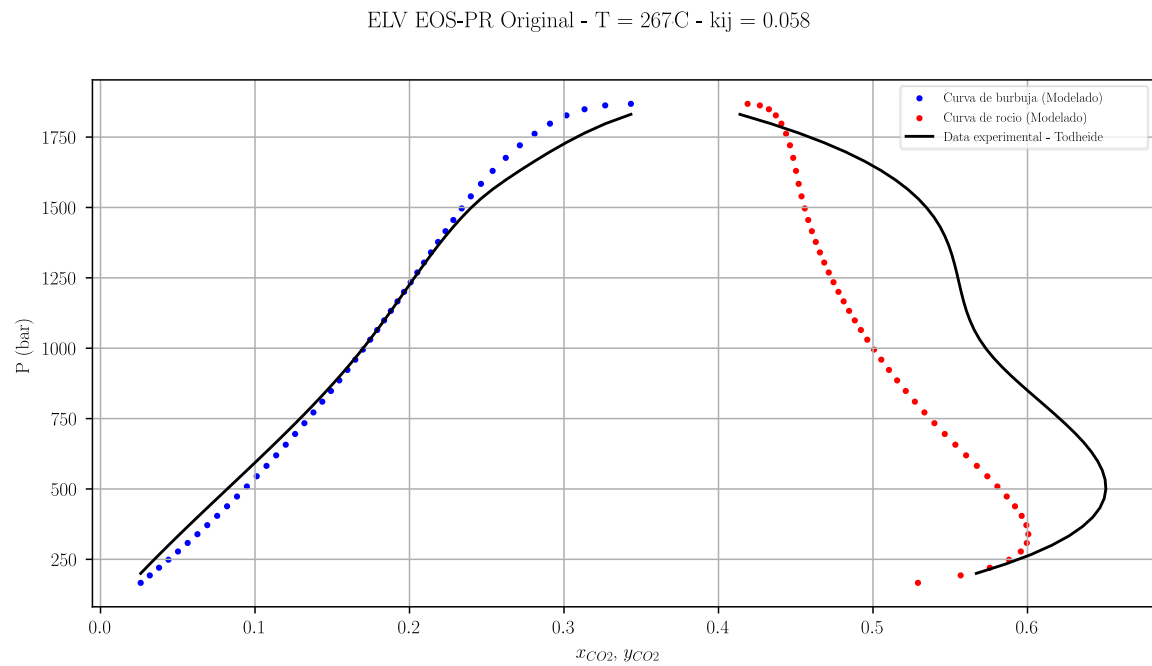


Podemos darnos cuenta a simple vista que la curva de burbuja tiene un comportamiento similar al de los datos experimentales, mientras que la curva de rocío, si bien es cierto tiene una forma similar a la data experimental, está muy alejada del comportamiento real de la mezcla.

Se realizó varias pruebas con diferentes " k_{ij} " y se obtuvo el mismo comportamiento, la curva de burbuja sube más rápido que la curva de rocío, y la curva de rocío no llega a asimilar el comportamiento real de la mezcla binaria. Podemos ver un ejemplo de ello, en la Figura 31, donde se tomó un $k_{ij} = 0.058$.

Figura 31

Comparación de EOS-PR con data experimental a 267°C y $k_{ij} = 0.058$



En las Figuras 32, 33, 34, 35 y 36 se puede observar un comportamiento similar para las temperaturas de 268°C, 270°C, 275°C, 300°C y 350°C respectivamente.

Figura 32

Comparación de EOS-PR con data experimental a 268°C y $k_{ij} = 0.055$

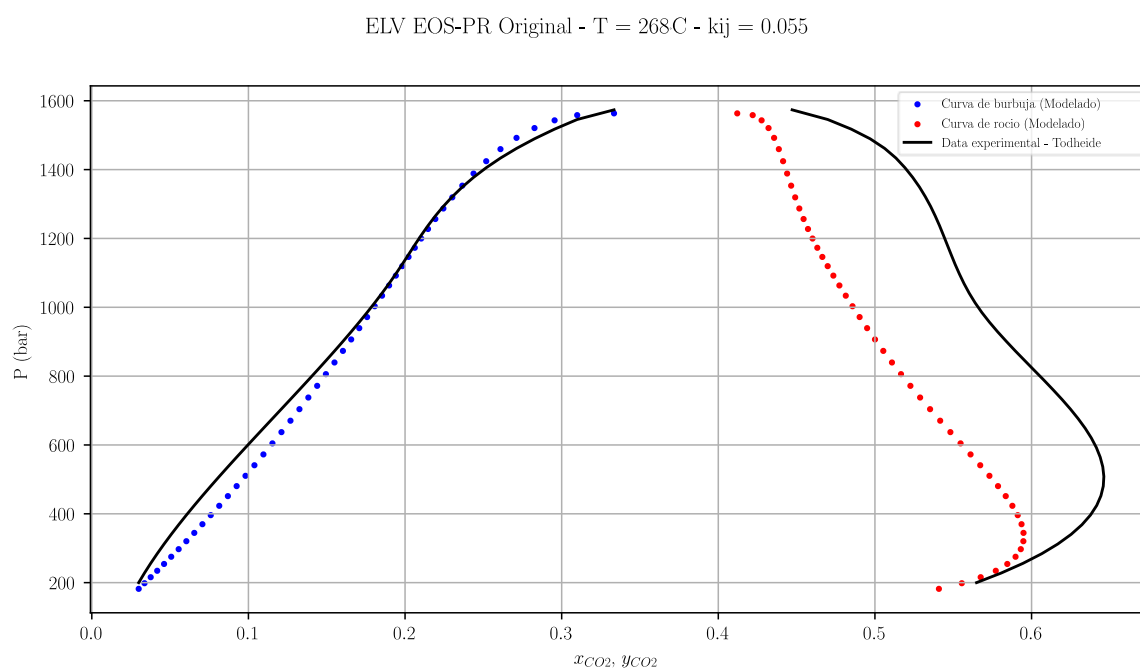


Figura 33

Comparación de EOS-PR con data experimental a 270°C y $k_{ij} = 0.055$

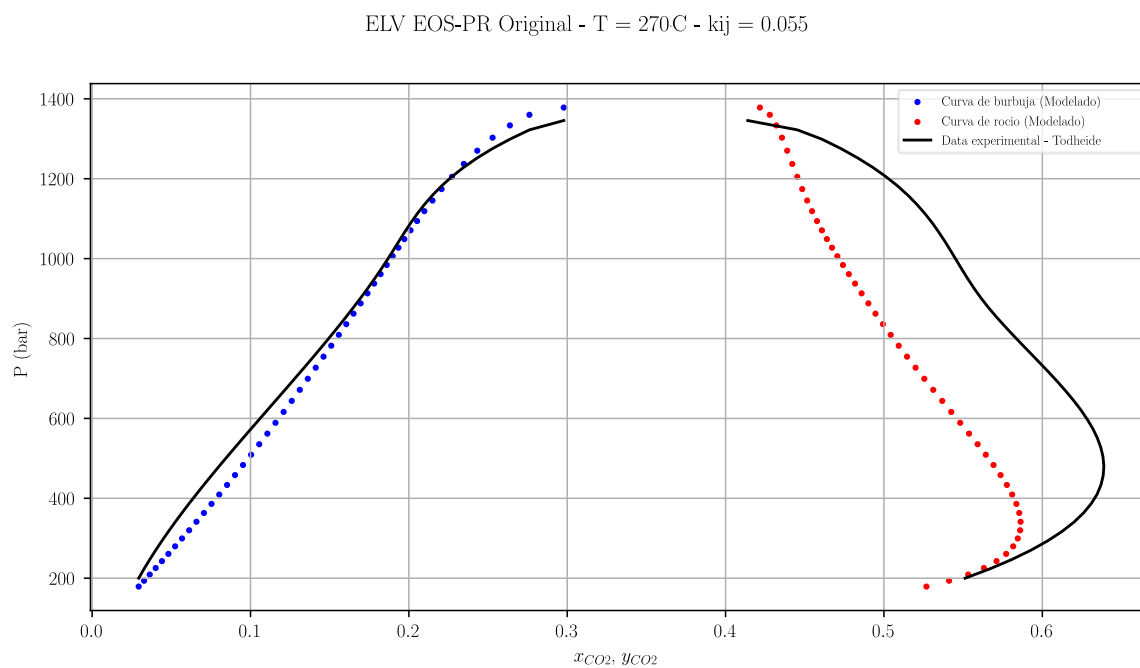


Figura 34

Comparación de EOS-PR con data experimental a 275°C y $k_{ij} = 0.054$

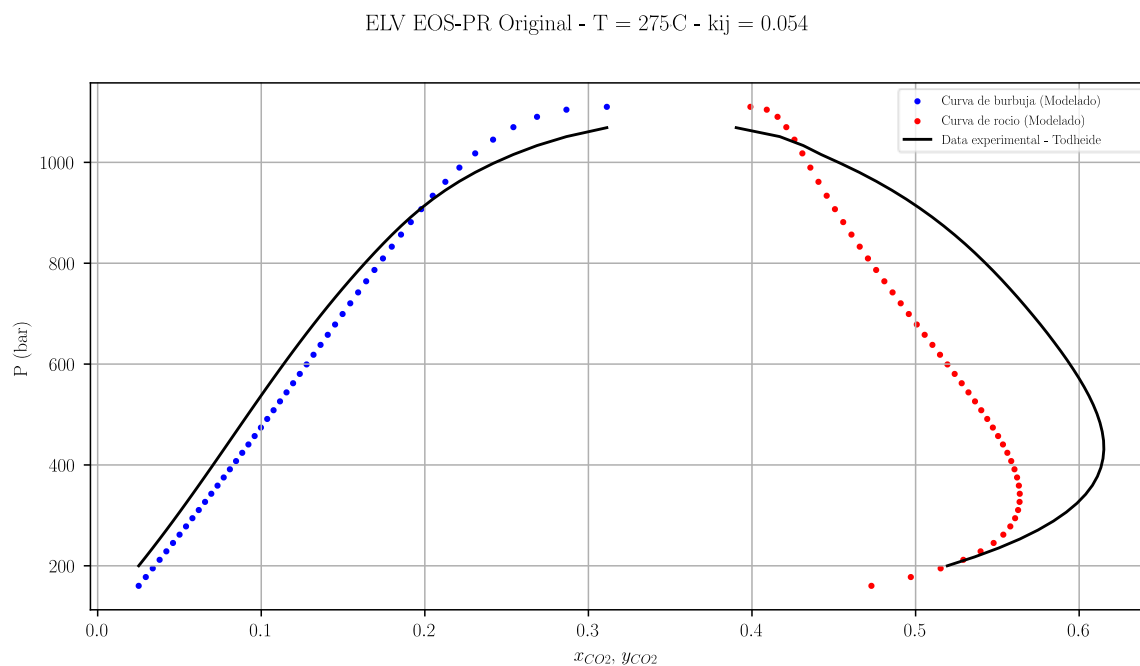


Figura 35

Comparación de EOS-PR con data experimental a 300°C y $k_{ij} = 0.058$

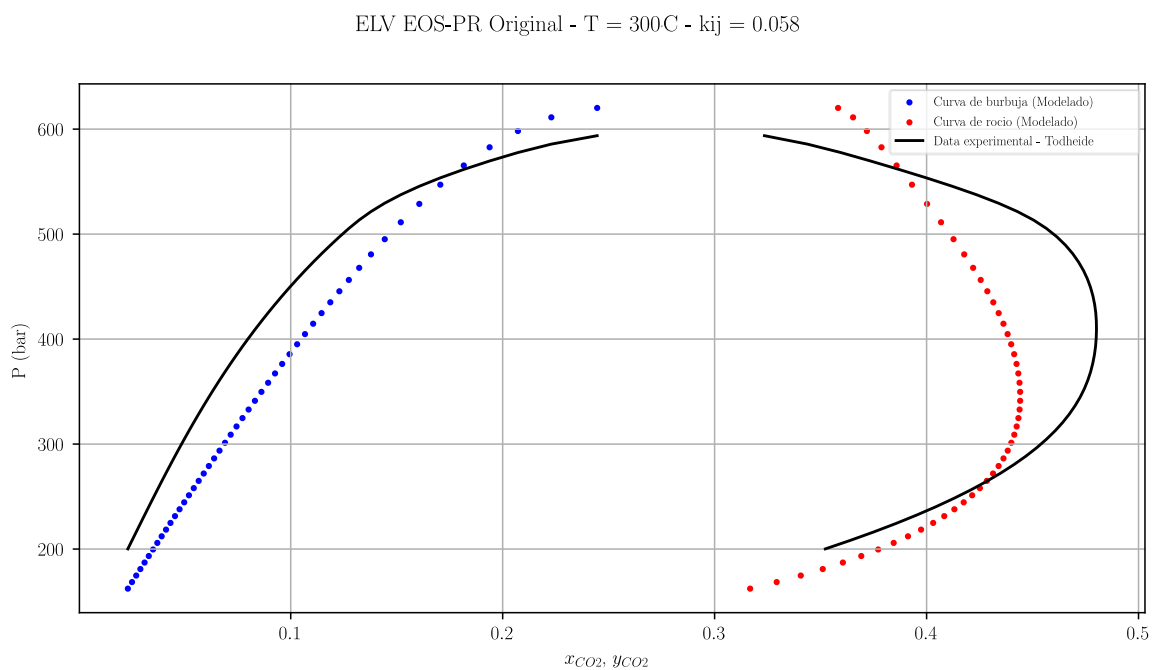
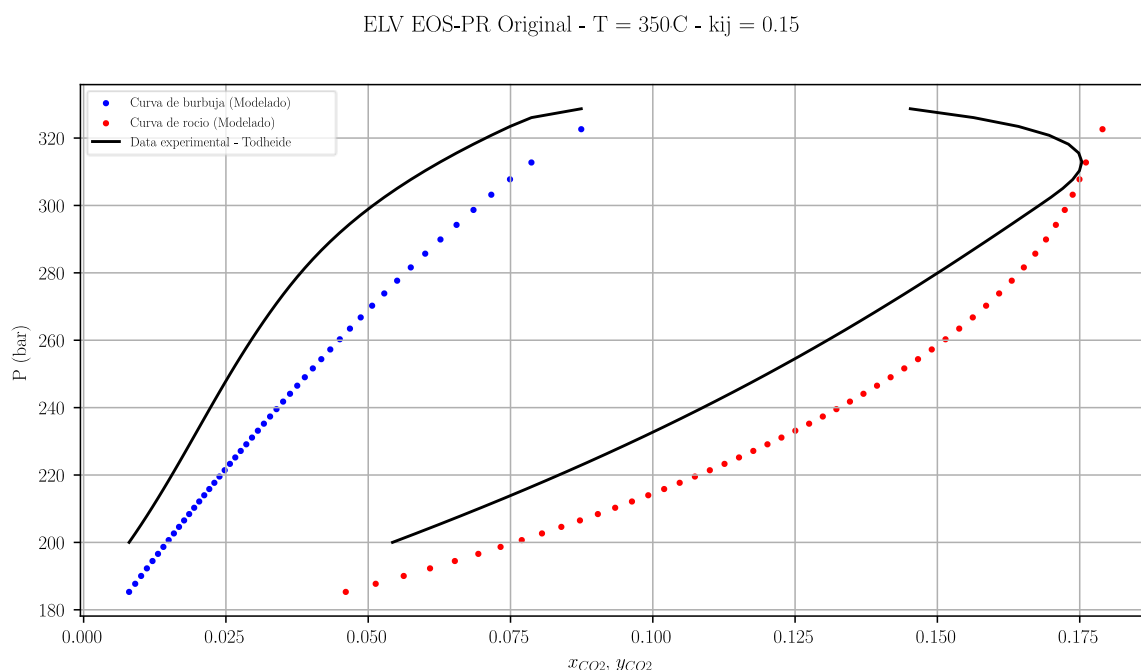


Figura 36

Comparación de EOS-PR con data experimental a 350°C y $k_{ij} = 0.058$



3.2.4 Determinación de parámetros “a” y “s_{ij}” en la EOS – PR

3.2.4.7 Cálculo de a_m^I y s_{ij}

Para fines de ejemplo se utilizar la $T = 270^\circ\text{C}$, pero el tratamiento es similar para todas las temperaturas. Se usó la librería Scipy en Python para poder usar optimización estocástica, y los valores de x_1 y y_1 fueron obtenidos de la data experimental parametrizada.

Podemos ver la aplicación del algoritmo por recocido simulado líneas abajo.

```
x1_array = x1_data_270
y1_array = y1_data_270
P_array = P_data_270
T = 270+273.15
#####
# Asumiendo una regla de mezcla como la de Van der Waals
# Definir límites para los parámetros a_m, sij
a_m_store_270_caso1 = []
sij_store_270_caso1 = []
error_algoritmo_270_caso1 = []
for idx, x in enumerate(x1_array):
    P = P_array[idx]
    x1 = x1_array[idx]
```

```

y1 = y1_array[idx]
x2 = 1-x1
y2 = 1-y1
k1_exp = y1/x1
k2_exp = y2/x2
lnk1 = np.log(k1_exp)
lnk2 = np.log(k2_exp)

if idx == 0:
    bounds = [(0, 10),      # Limites para a_m
              (-0.5, 0.5)]  # Limites para sij
else:
    bounds = [(a_m_store_270_casol[idx-1]*0.9,
a_m_store_270_casol[idx-1]*1.1),      # Limites para a_m
              (sij_store_270_casol[idx-1]*0.92, sij_store_270_casol[idx-
1]*1.08)]  # Limites para sij

# Ejecutar la optimización con recocido simulado
resultado = dual_annealing(funcion_costo_1BIP_aml, bounds,
seed=123)

# Mostrar los valores óptimos encontrados
print("Valores óptimos:")
print("a_m:", resultado.x[0])
print("sij:", resultado.x[1])
print("Mínimo error encontrado:", resultado.fun)

a_m_store_270_casol.append(resultado.x[0])
sij_store_270_casol.append(resultado.x[1])
error_algoritmo_270_casol.append(resultado.fun)

```

Nota. Creación propia

Como podemos observar se tomaron rangos de búsqueda entre 0 y 10 para a_m^I y entre -0.5 y 0.5 para s_{ij} , estos son rangos más amplios de los que comprenden los valores que se han calculado para a_m^I y s_{ij} utilizando la EOS-PR clásica.

Además, es válido resaltar que después de la primera iteración, los limites se restringen en un rango de +-10% el valor anterior obtenido, esto se hace con el fin de condicionar mucho más al algoritmo estocástico para que encuentre soluciones en una vecindad cercana a la solución hallada en la iteración anterior.

Las soluciones y sus respectivos errores se muestran en las siguientes líneas y también las podemos encontrar en el Anexo 7.

```

Valores optimos:
a_m: 6.719242859183671
sij: 0.0876140251516481
Minimo error encontrado: 1.0808842842990884e-07
Valores optimos:
a_m: 6.703471439798744

```

```

sij: 0.09139250721331413
Minimo error encontrado: 0.0006013546659423552
Valores optimos:
a_m: 6.676802624547241
sij: 0.09604424595877056
Minimo error encontrado: 0.0005514644793110923
Valores optimos:
a_m: 6.6435302462228645
sij: 0.10082498399696452
Minimo error encontrado: 1.3177597901758986e-08
Valores optimos:
a_m: 6.607970272680612
sij: 0.104872689831151
Minimo error encontrado: 2.811984924688815e-06
Valores optimos:
a_m: 6.57084925102496
sij: 0.10828966634246368
Minimo error encontrado: 1.0835369590456168e-07
Valores optimos:
a_m: 6.533260856273732
sij: 0.1109999496574255
Minimo error encontrado: 9.586178434384252e-07
Valores optimos:
a_m: 6.496041473792922
sij: 0.11294990554944277
Minimo error encontrado: 9.471896841373173e-05
Valores optimos:
a_m: 6.459189130620325
sij: 0.11428914033981949
Minimo error encontrado: 1.8309957972029167e-07
Valores optimos:
a_m: 6.423414435185654
sij: 0.11492960076836102
Minimo error encontrado: 2.269995225123722e-08
Valores optimos:
a_m: 6.388780275785594
sij: 0.11494807182931711
Minimo error encontrado: 4.310118437234056e-05
Valores optimos:
a_m: 6.355055401378499
sij: 0.1144881514287492
Minimo error encontrado: 5.681632166587569e-05
Valores optimos:
a_m: 6.3227012381215
sij: 0.11348030418610795
Minimo error encontrado: 3.841476419186307e-07
Valores optimos:
a_m: 6.291342129415877
sij: 0.11208572462363972
Minimo error encontrado: 5.6099003775678113e-08
Valores optimos:
a_m: 6.261157419051134
sij: 0.11029810424189697
Minimo error encontrado: 0.00010782319677393737
Valores optimos:
a_m: 6.231552924477482
sij: 0.10832639701474063

```

```

Minimo error encontrado: 6.715287680369997e-07
Valores optimos:
a_m: 6.203638796966516
sij: 0.10588780741427835
Minimo error encontrado: 0.00045248041580236986
Valores optimos:
a_m: 6.17501644874351
sij: 0.10369711137982246
Minimo error encontrado: 4.7379314915829696e-09
Valores optimos:
a_m: 6.147717940403282
sij: 0.10120237483318423
Minimo error encontrado: 1.8214664609939035e-07
Valores optimos:
a_m: 6.120939785836441
sij: 0.09865813393415833
Minimo error encontrado: 1.097330944110908e-06
Valores optimos:
a_m: 6.094664318528352
sij: 0.09609306784895988
Minimo error encontrado: 3.4505690353126006e-05
Valores optimos:
a_m: 6.0686190785226986
sij: 0.09360703122212094
Minimo error encontrado: 2.0415549811581002e-05
Valores optimos:
a_m: 6.0430201094461475
sij: 0.09115970832091475
Minimo error encontrado: 1.0335205147260496e-07
Valores optimos:
a_m: 6.0176550340547
sij: 0.08883193487355094
Minimo error encontrado: 1.0701811342550549e-05
Valores optimos:
a_m: 5.992590228525005
sij: 0.08662442467597982
Minimo error encontrado: 2.458595026677557e-06
Valores optimos:
a_m: 5.9677478848668
sij: 0.08457848188553596
Minimo error encontrado: 6.159977361530622e-08
Valores optimos:
a_m: 5.94313046915667
sij: 0.08271208378292455
Minimo error encontrado: 1.5826310961419843e-05
Valores optimos:
a_m: 5.918826226577858
sij: 0.08101813794604654
Minimo error encontrado: 1.795870060439242e-07
Valores optimos:
a_m: 5.894771296783024
sij: 0.07953297693565402
Minimo error encontrado: 3.890741284706678e-09
Valores optimos:
a_m: 5.871034084930888
sij: 0.07825466715212953
Minimo error encontrado: 8.078472405248505e-06

```

```

Valores optimos:
a_m: 5.847622547572437
sij: 0.07719858894101517
Minimo error encontrado: 1.1269675052716366e-05
Valores optimos:
a_m: 5.824621605947267
sij: 0.0763526949703295
Minimo error encontrado: 2.147916793393545e-05
Valores optimos:
a_m: 5.802108837034434
sij: 0.07569099478478833
Minimo error encontrado: 3.0427640124175426e-06
Valores optimos:
a_m: 5.7803577680639044
sij: 0.07512830396877947
Minimo error encontrado: 1.445746855566199e-09
Valores optimos:
a_m: 5.759388048532426
sij: 0.07465526084890081
Minimo error encontrado: 3.7102848630454766e-05
Valores optimos:
a_m: 5.7394958549079504
sij: 0.07416326539918915
Minimo error encontrado: 3.1841794863041173e-06
Valores optimos:
a_m: 5.718956439272041
sij: 0.07399439136251569
Minimo error encontrado: 0.0005616082180436299
Valores optimos:
a_m: 5.700482787101836
sij: 0.07311098244053865
Minimo error encontrado: 1.4172393814071427e-08
Valores optimos:
a_m: 5.680240058415156
sij: 0.07242521717775968
Minimo error encontrado: 3.321293800195857e-05
Valores optimos:
a_m: 5.65848620518934
sij: 0.071642771712685
Minimo error encontrado: 2.7019530943750425e-06
Valores optimos:
a_m: 5.633763847620249
sij: 0.07099149007174539
Minimo error encontrado: 0.0004889043261011583
Valores optimos:
a_m: 5.608485102440499
sij: 0.06953234631551289
Minimo error encontrado: 4.400488979922068e-05
Valores optimos:
a_m: 5.579718427105908
sij: 0.06795148751599787
Minimo error encontrado: 0.00046260410459614265
Valores optimos:
a_m: 5.544971046896393
sij: 0.06683594667190543
Minimo error encontrado: 6.975393965891197e-05
Valores optimos:

```

```
a_m: 5.506956108783725
sij: 0.06516312226394479
Minimo error encontrado: 2.6148539222781153e-08
Valores optimos:
a_m: 5.464409979419462
sij: 0.06314308043320227
Minimo error encontrado: 0.00034561118526155177
Valores optimos:
a_m: 5.414350257722903
sij: 0.06158042504097573
Minimo error encontrado: 1.8909008708267727e-06
Valores optimos:
a_m: 5.358585797642027
sij: 0.05973198366121243
Minimo error encontrado: 9.479666066680714e-06
Valores optimos:
a_m: 5.296414291605713
sij: 0.05751320133745218
Minimo error encontrado: 2.5358447974521692e-05
Valores optimos:
a_m: 5.188750924803772
sij: 0.05341871565830923
Minimo error encontrado: 2.735539970127876e-06
```

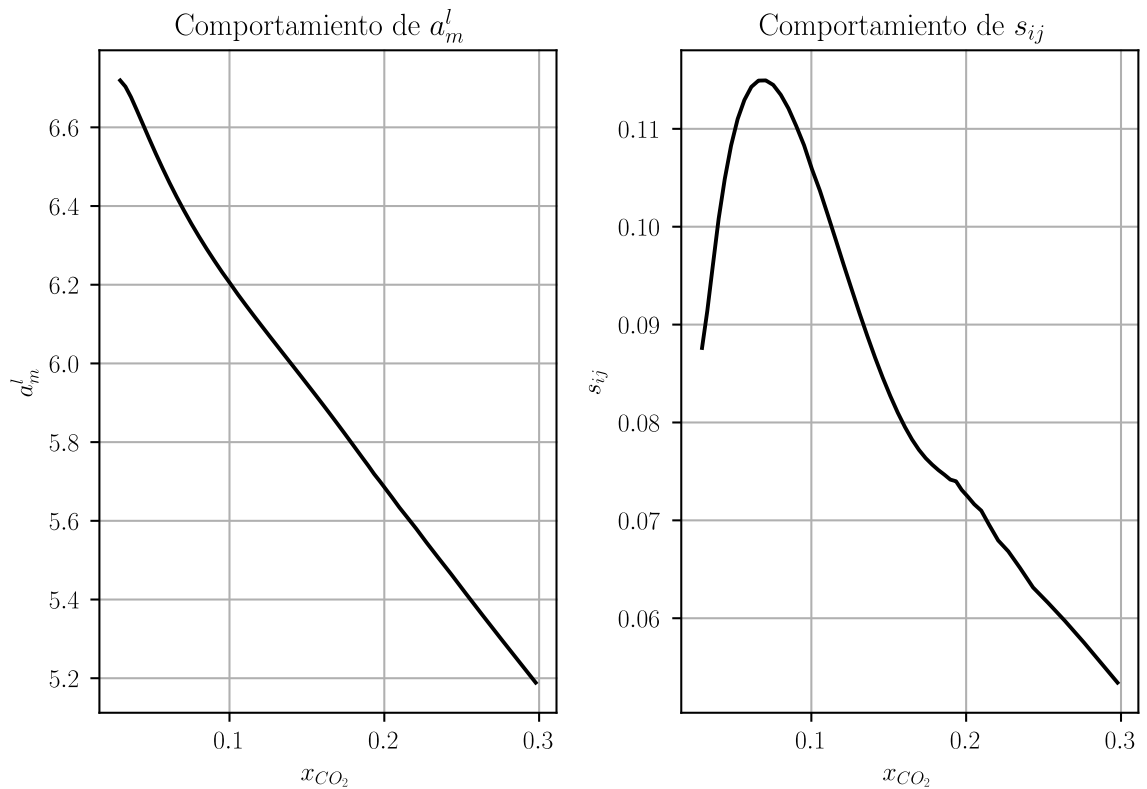
Nota. Creación propia

Como podemos ver se obtuvo un error absoluto máximo de 0.0006 con respecto a los logaritmos naturales de los coeficientes de distribución experimentales y calculados del CO_2 y H_2O . Podemos graficar el comportamiento de a_m y s_{ij} en la Figura 37.

Figura 37

Comportamiento de los parámetros de la EOS-PR a 270°C

Análisis del comportamiento de parámetros de la EOS de Peng-Robinson a 270°C

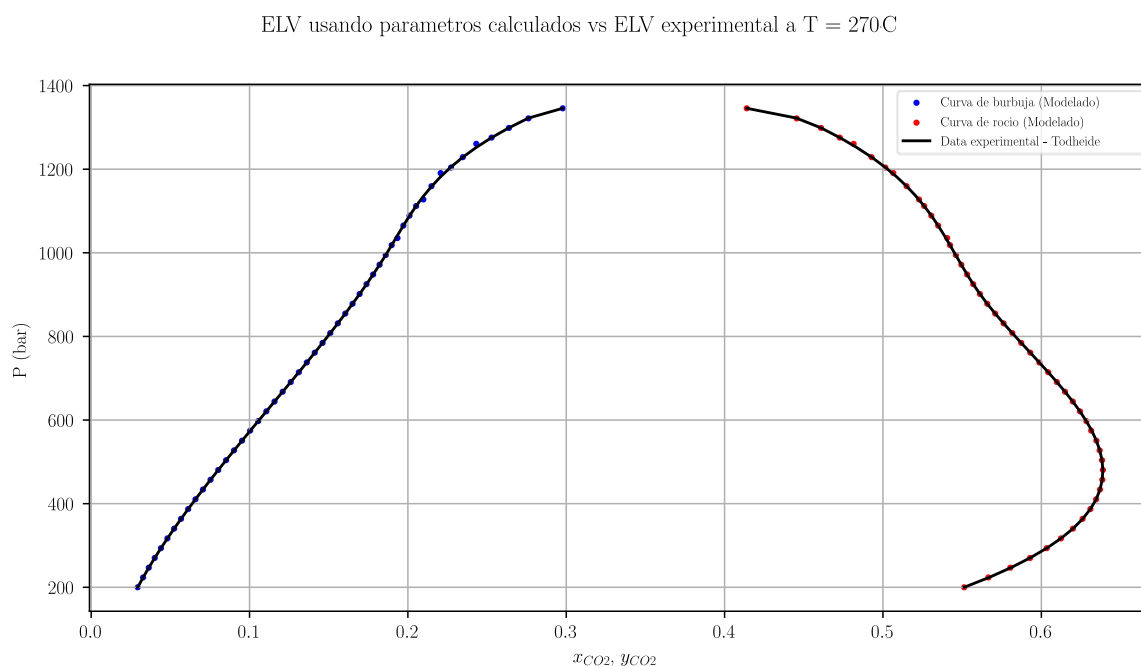


A simple vista podemos ver un comportamiento muy interesante de ambas gráficas, por un lado, el parámetro “a” de mezcla en fase líquida tiene un comportamiento cuasi-lineal lo cual explicaría su predicción medianamente buena usando la regla de mezcla de Van der Waals, mientras que por el otro lado el parámetro de interacción binaria s_{ij} tiene un comportamiento no lineal con variaciones amplias.

Podemos usar los parámetros “ a_m^l ” y “ s_{ij} ” para resolver el ELV a 270°C, tal como podemos observar en la Figura 38.

Figura 38

Evaluación de las soluciones del ELV usando los parámetros encontrados mediante optimización estocástica a $T = 270^\circ\text{C}$



Se realizó el mismo procedimiento para todas las isotermas y en cada uno de ellos podemos encontrar que las soluciones encontradas por optimización estocástica nos brindan un comportamiento gráficamente similar a los valores experimentales.

A continuación, se muestra el comportamiento de a_m^I y s_{ij} para cada isoterma. En las Figuras 39, 40, 41, 42 y 43 se observa el comportamiento de los parámetros de la EOS-PR. Los valores calculados para a_m^I y s_{ij} para cada isoterma se pueden encontrar en los Anexos 7, 8, 9, 10, 11 y 12.

Figura 39

Comportamiento de los parámetros de la EOS-PR a 267°C

Análisis del comportamiento de parámetros de la EOS de Peng-Robinson a 267°C

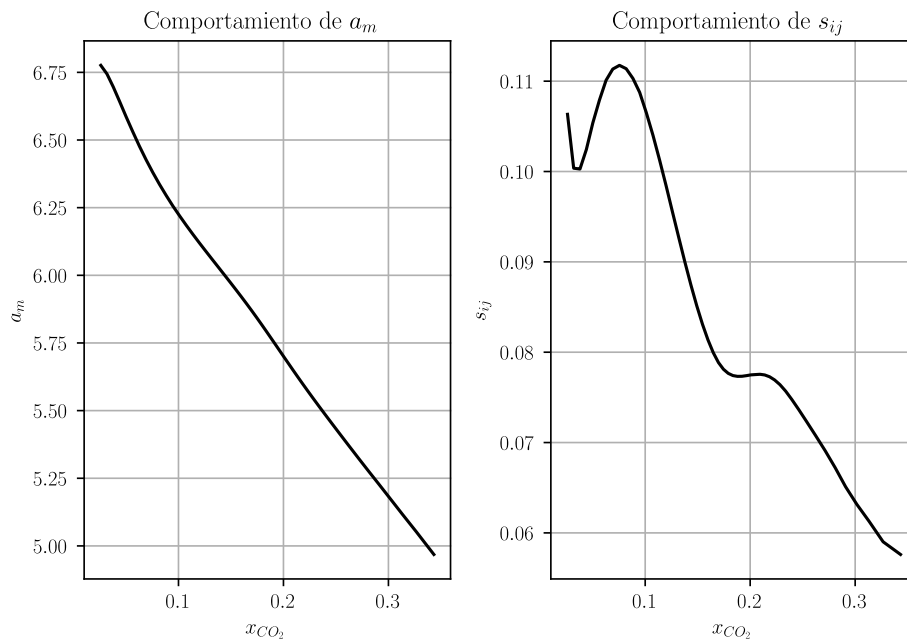


Figura 40

Comportamiento de los parámetros de la EOS-PR a 268°C

Análisis del comportamiento de parámetros de la EOS de Peng-Robinson a 268°C

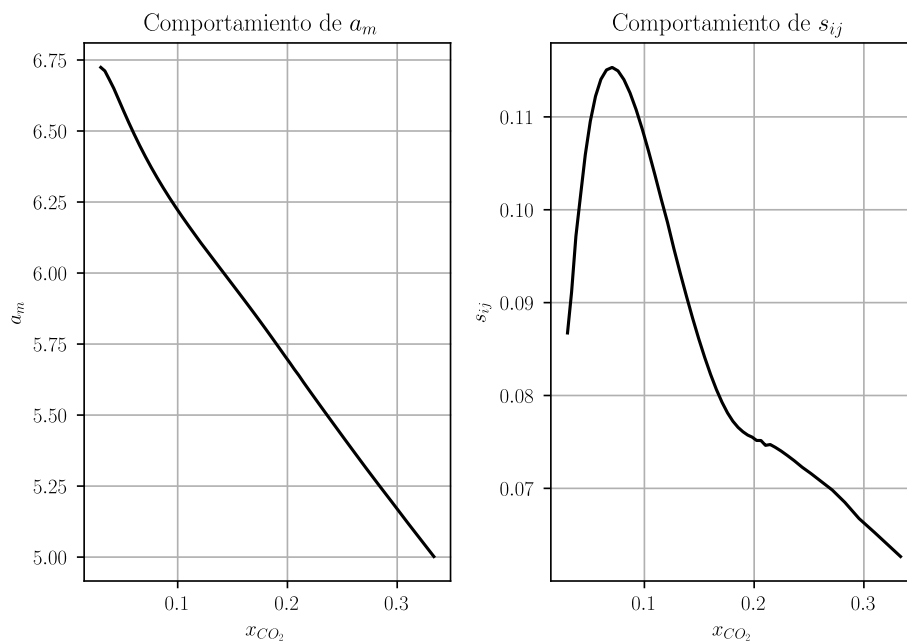


Figura 41

Comportamiento de los parámetros de la EOS-PR a 275°C

Análisis del comportamiento de parámetros de la EOS de Peng-Robinson a 275°C

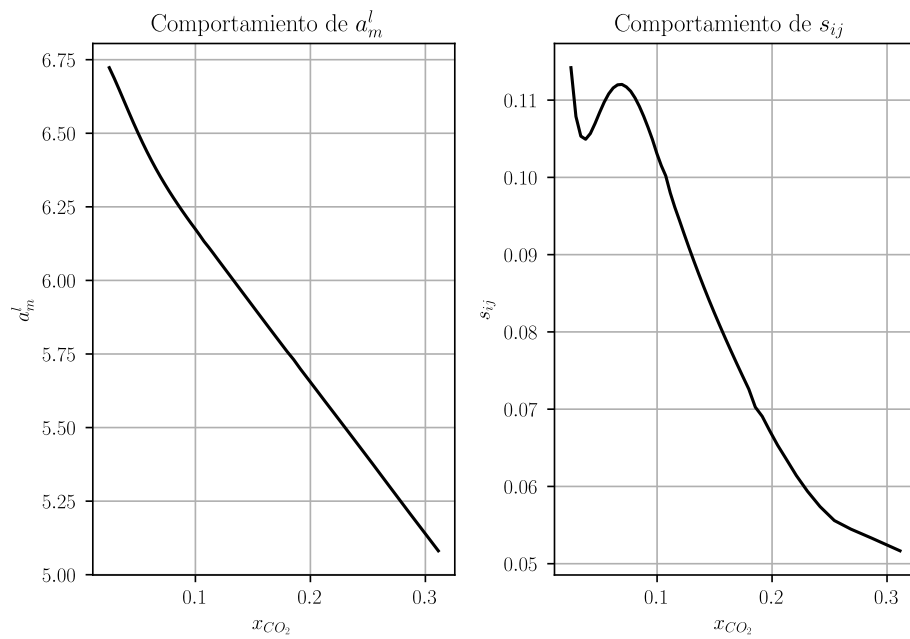


Figura 42

Comportamiento de los parámetros de la EOS-PR a 300°C

Análisis del comportamiento de parámetros de la EOS de Peng-Robinson a 300°C

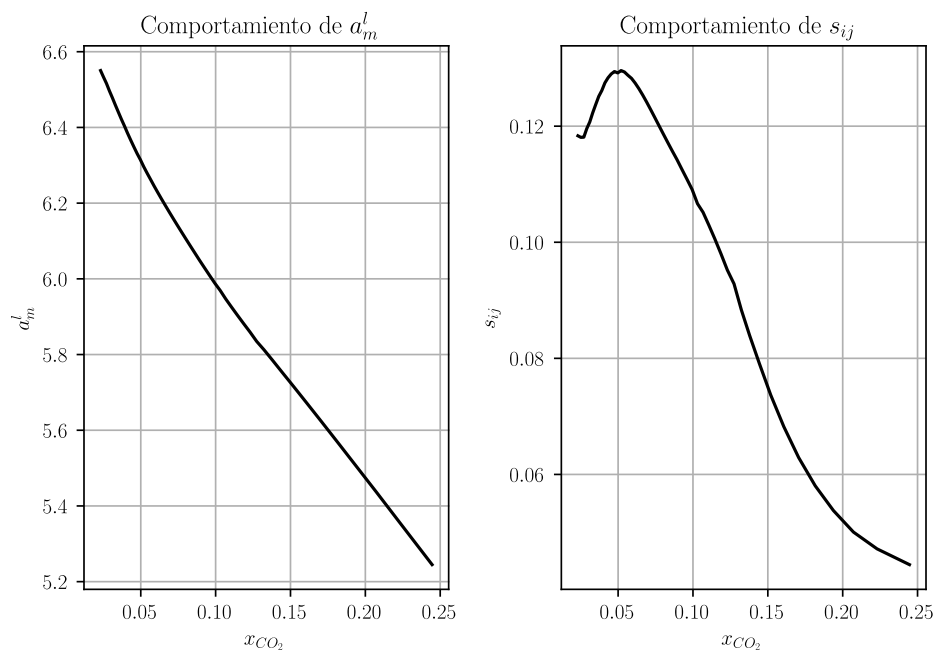
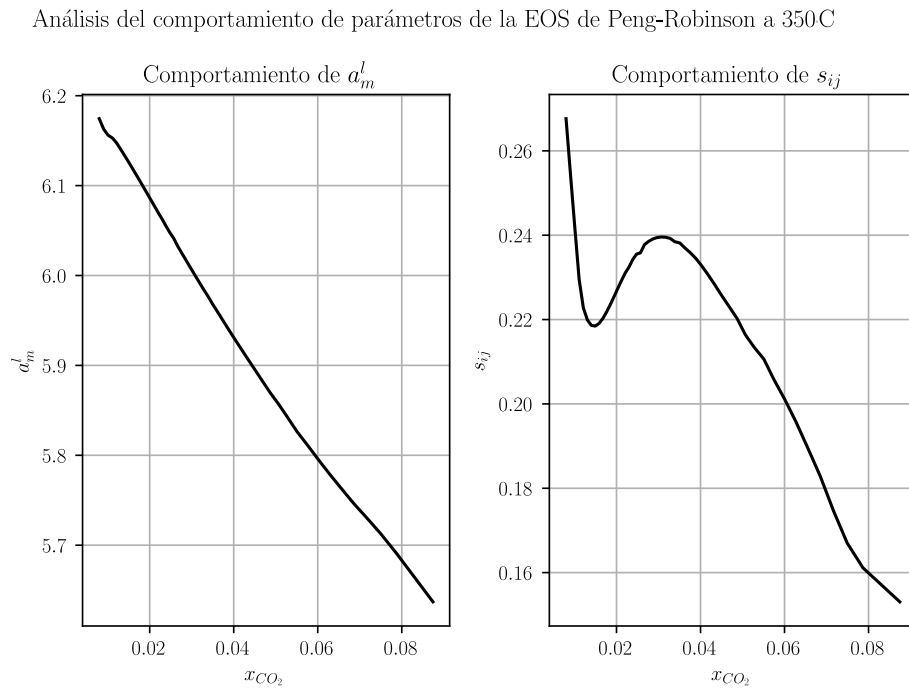


Figura 43

Comportamiento de los parámetros de la EOS-PR a 350°C

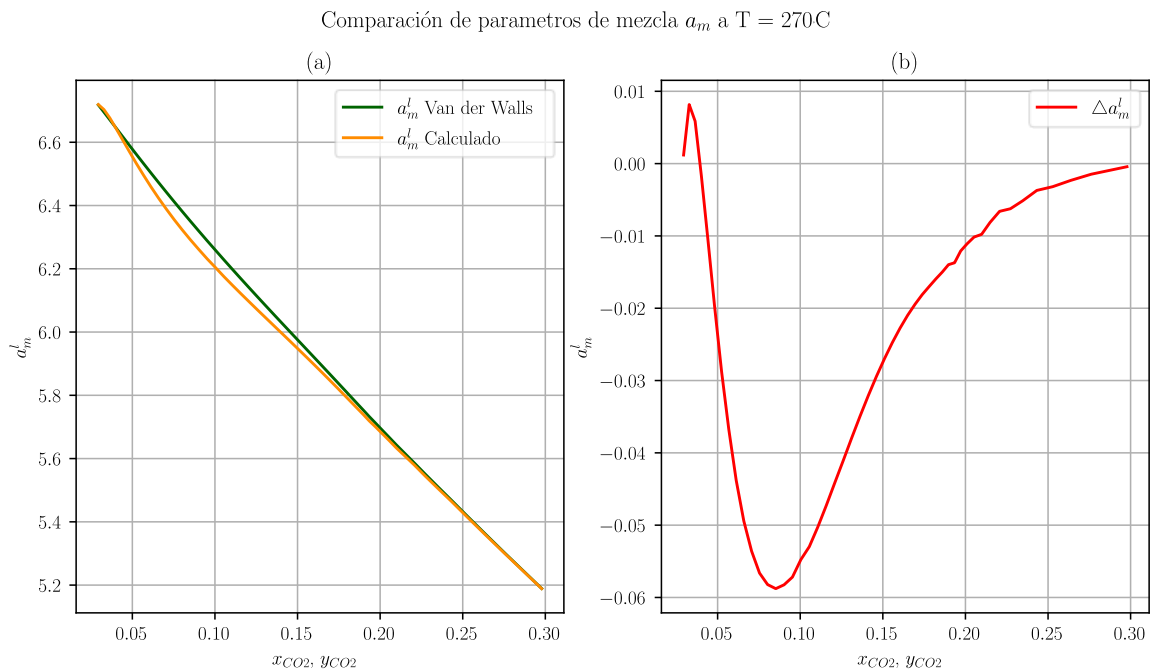


Veamos que sucede si comparamos el a_m^l calculado por la regla de mezcla de Van der Waals con el a_m^l calculado por el algoritmo propuesto.

En la Figura 44a podemos observar la comparación entre ambos parámetros a $T=270^\circ\text{C}$, vemos que los parámetros a_m^l son muy cercanos tanto el calculado mediante el algoritmo como el propuesto por Van der Waals, y en la Figura 44b podemos observar la diferencia entre ambos, podemos deducir que un pequeño cambio en el parámetro a_m^l puede causar estimaciones erróneas en la determinación del ELV.

Figura 44

Variación entre a_m^I calculado por el nuevo algoritmo y el a_m^I calculado mediante la regla de mezcla de Van der Waals a $T = 270^\circ\text{C}$



Podemos extender el análisis a las demás isothermas para ver su comportamiento, tal como se ve en las Figuras 45, 46, 47, 48 y 49.

Figura 45

Variación entre a_m^I calculado por el nuevo algoritmo y el a_m^I calculado mediante la regla de mezcla de Van der Waals a $T = 267^\circ\text{C}$

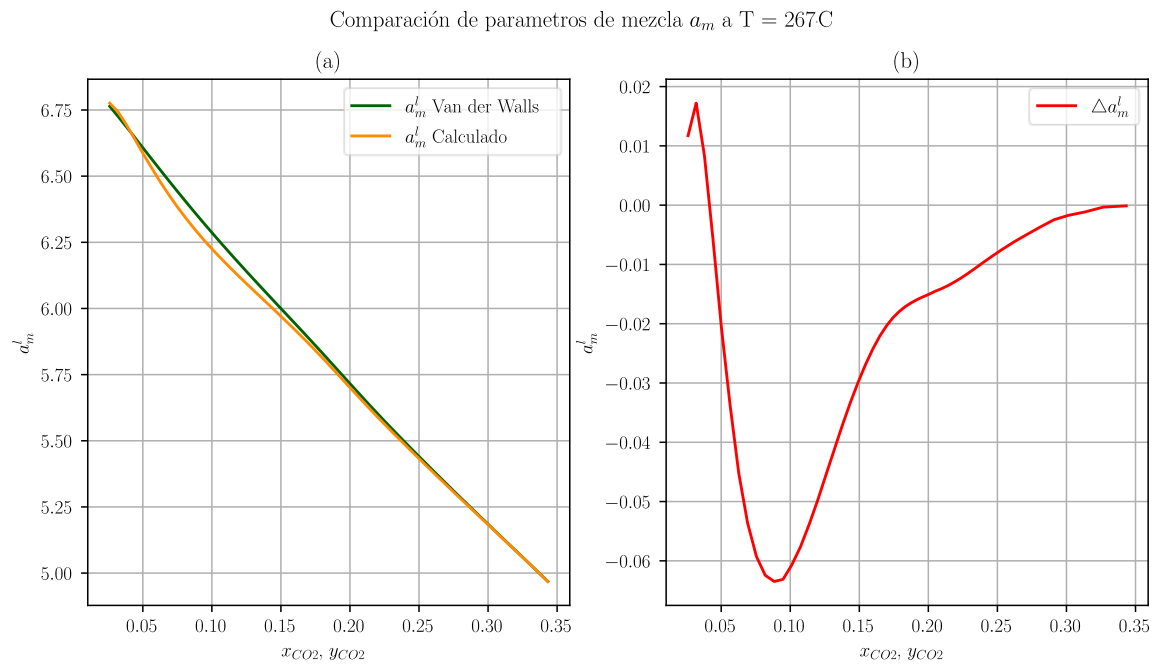


Figura 46

Variación entre a_m^l calculado por el nuevo algoritmo y el a_m^l calculado mediante la regla de mezcla de Van der Waals a $T = 268^\circ\text{C}$

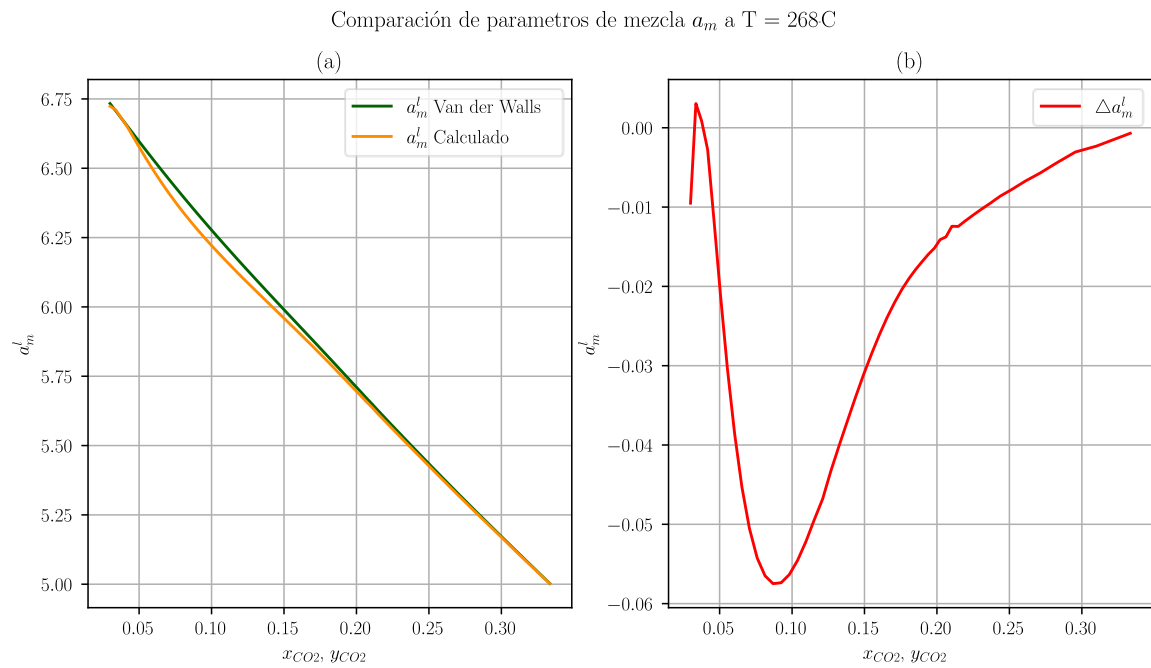


Figura 47 Variación entre a_m^l calculado por el nuevo algoritmo y el a_m^l calculado mediante la regla de mezcla de Van der Waals a $T = 275^\circ\text{C}$

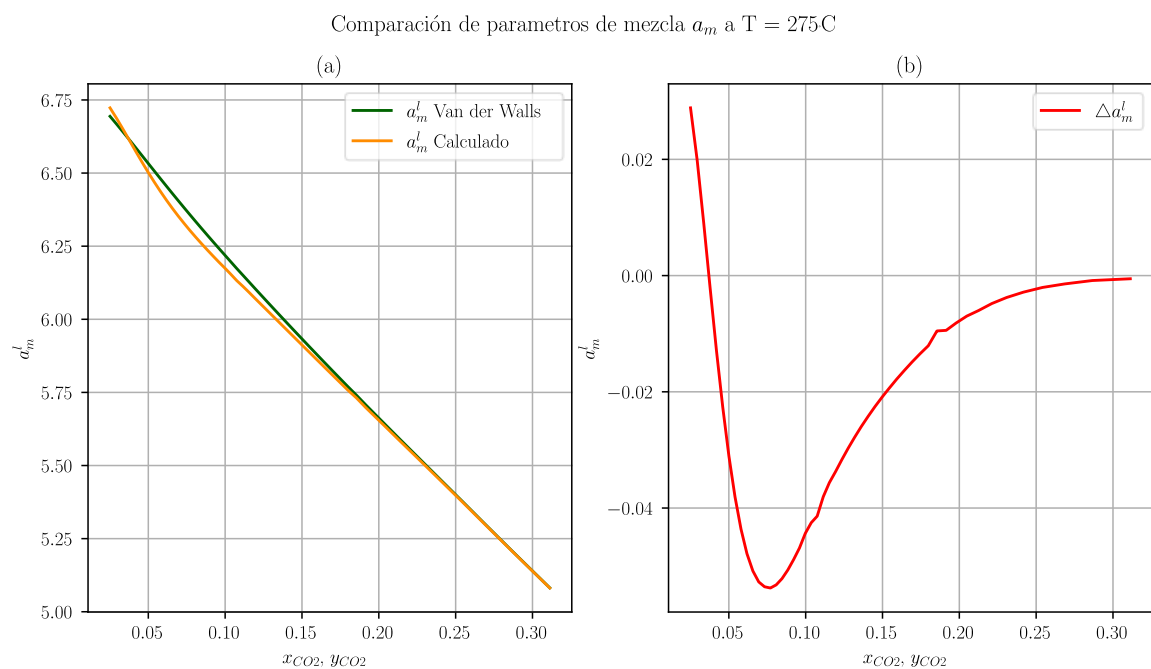


Figura 48

Variación entre a_m^I calculado por el nuevo algoritmo y el a_m^I calculado mediante la regla de mezcla de Van der Waals a $T = 300^\circ\text{C}$

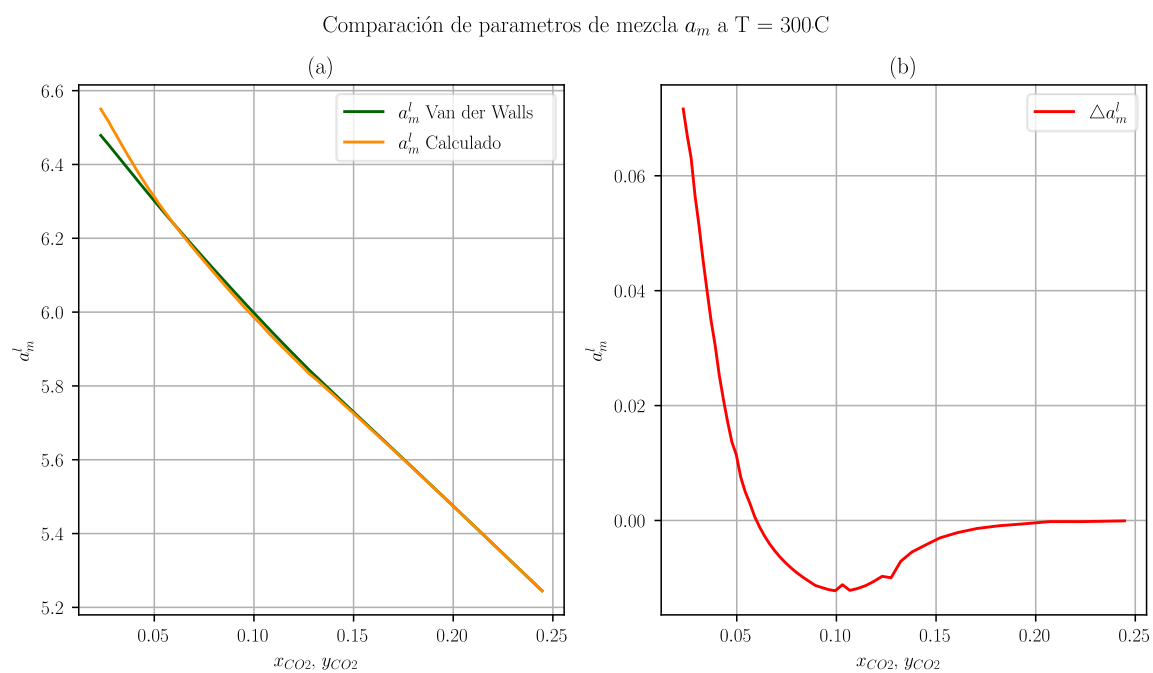
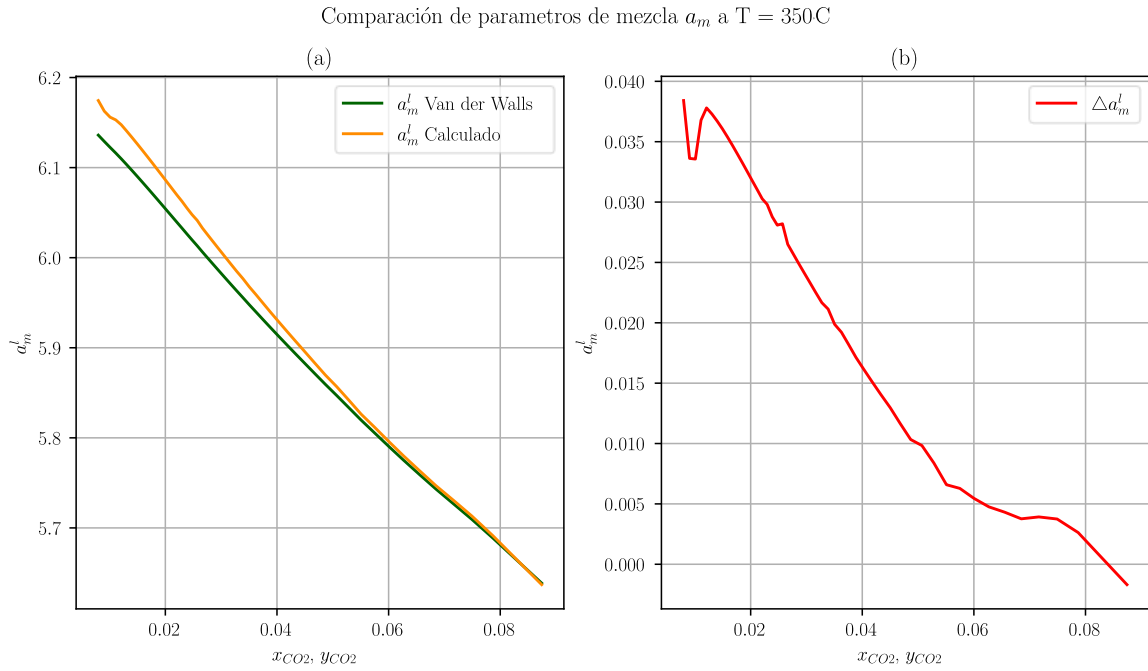


Figura 49

Variación entre a_m^I calculado por el nuevo algoritmo y el a_m^I calculado mediante la regla de mezcla de Van der Waals a $T = 350^\circ\text{C}$



Esto confirma la decisión de usar la corrección del parámetro a_m^I (Δa_m^I) obtenido por la ecuación de Van der Waals como entrada de la red neuronal, entonces la siguiente tarea sería modelar una red neuronal que pueda predecir una corrección en el parámetro a_m^I y modelar otra red neuronal para que pueda modelar el parámetro de interacción binaria S_{ij} .

Finalmente podemos observar el diagrama P_{xy} utilizando los valores calculados para " a_m^I " y " s_{ij} " calculados por el algoritmo de optimización estocástica en las Figuras 50, 51, 52, 53 y 54.

Figura 50

Comparación del ELV usando los parámetros calculados y datos experimentales a 267°C

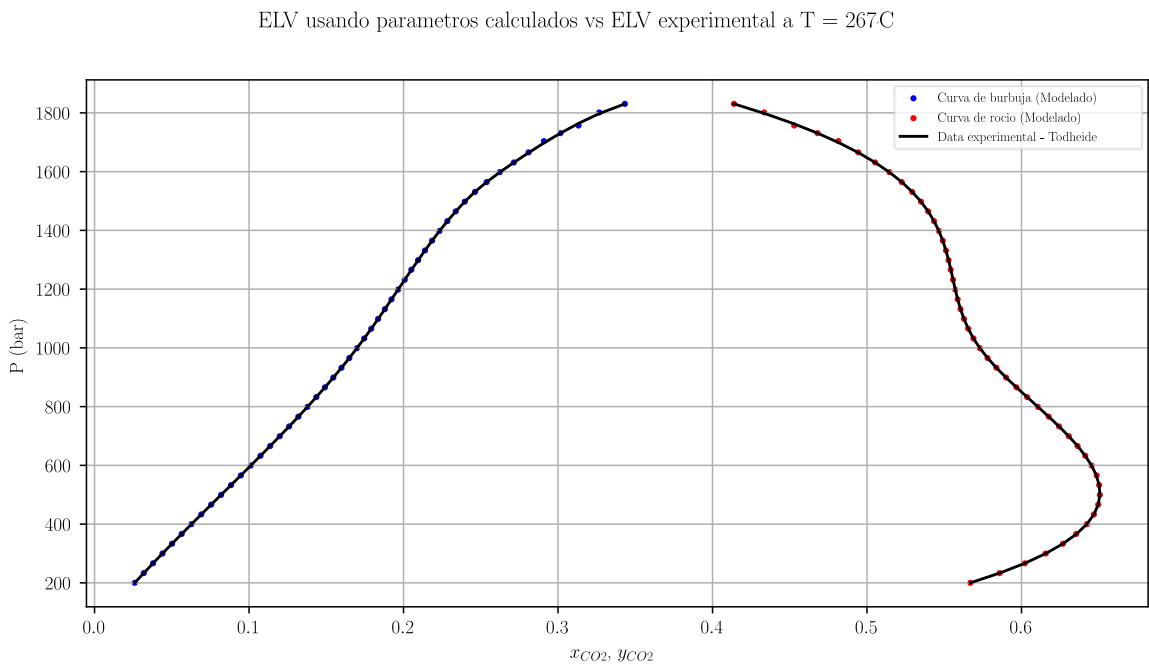


Figura 51

Comparación del ELV usando los parámetros calculados y datos experimentales a 268°C

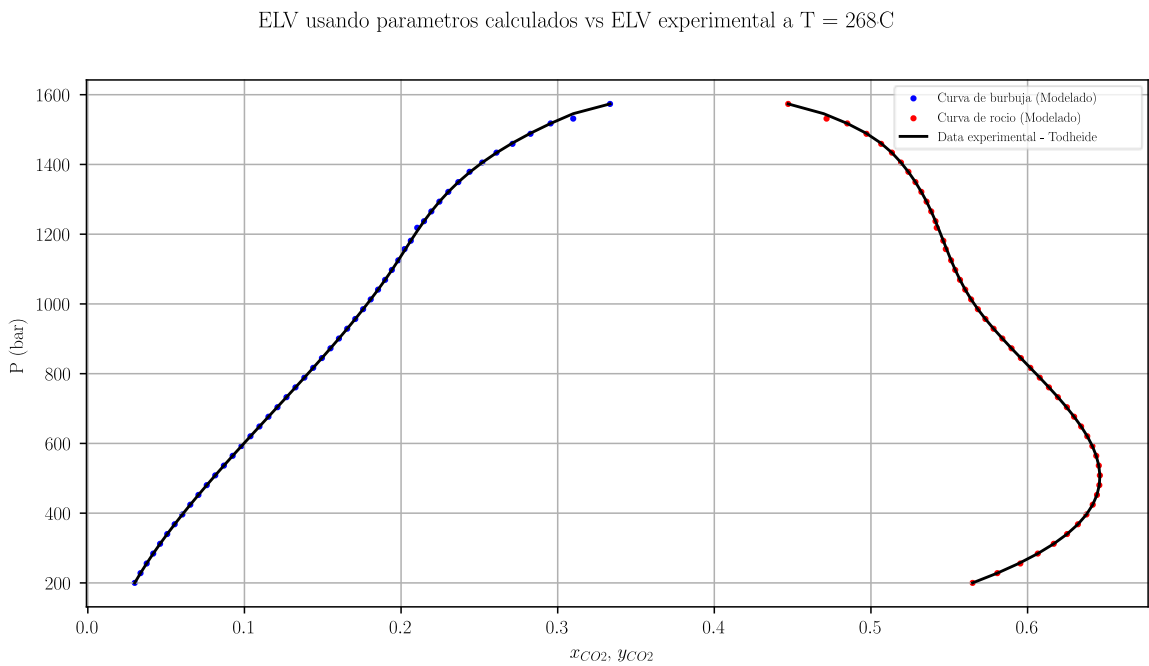


Figura 52

Comparación del ELV usando los parámetros calculados y datos experimentales a 275°C

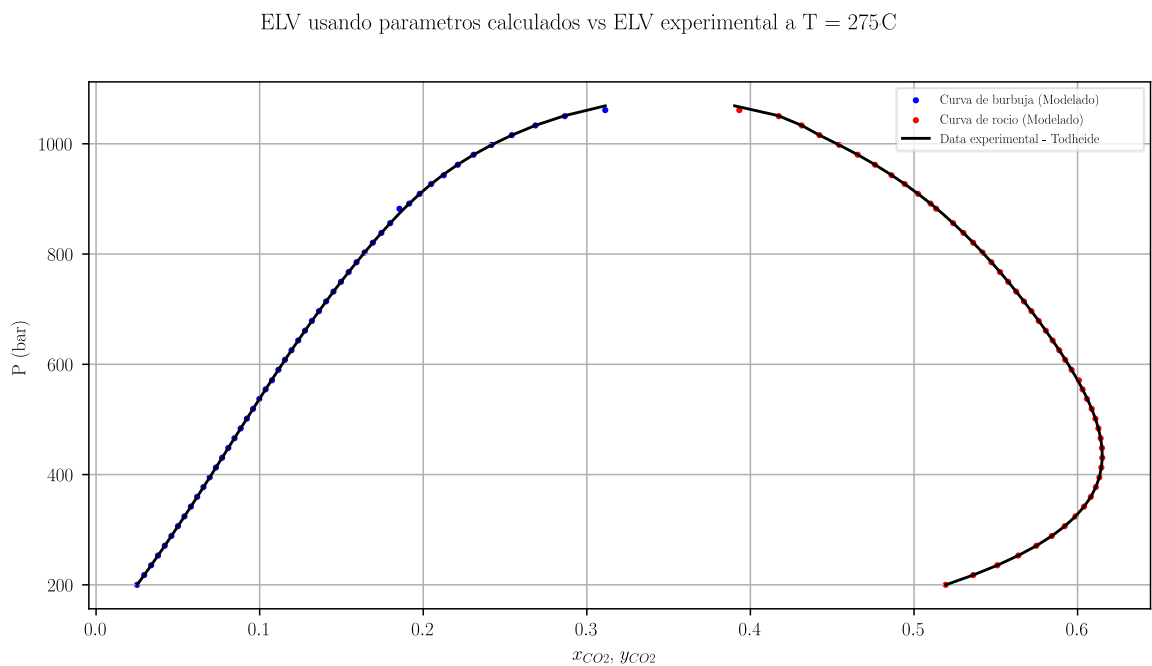


Figura 53

Comparación del ELV usando los parámetros calculados y datos experimentales a 300°C

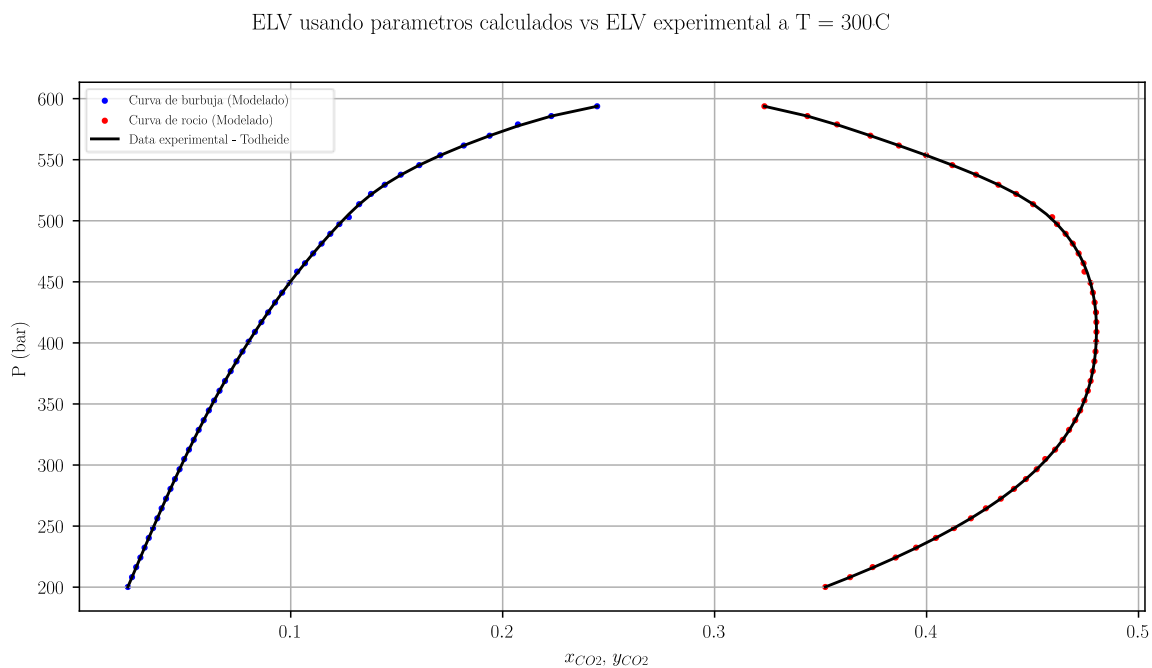
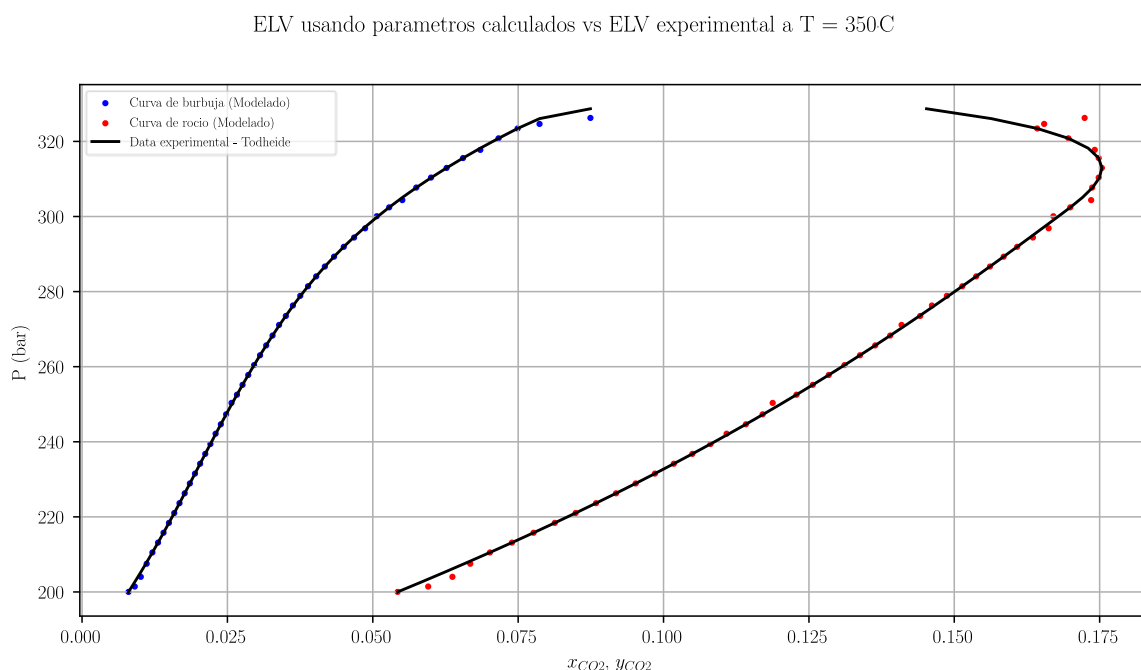


Figura 54

Comparación del ELV usando los parámetros calculados y datos experimentales a 350°C



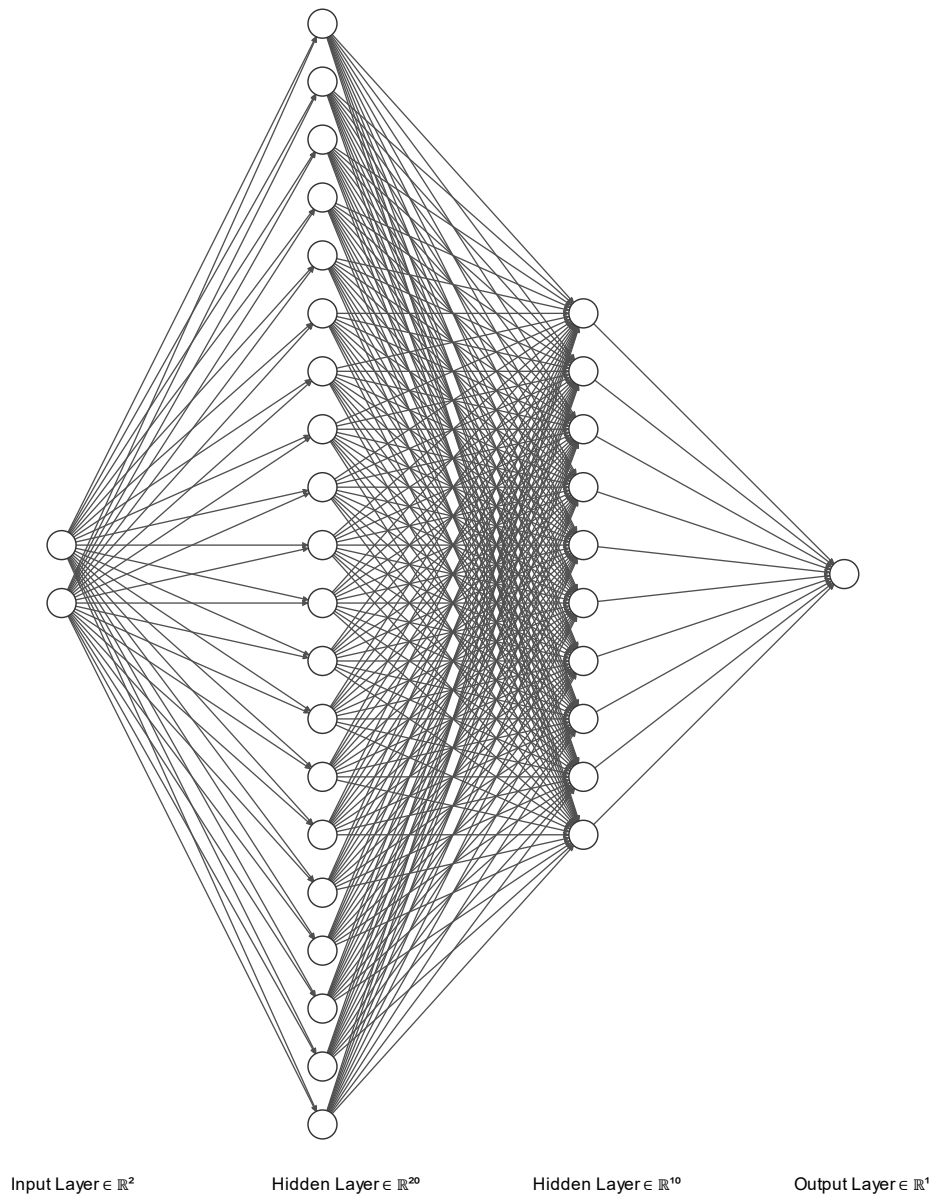
3.2.5 Definición de red neuronal para modelamiento del equilibrio líquido-vapor

Se decidió utilizar dos redes neuronales con 2 entradas en la capa de entrada (fracción molar de CO_2 en fase líquida y temperatura), 20 neuronas en la primera capa oculta, 10 neuronas en la segunda capa oculta y 1 salida en la capa de salida (corrección del parámetro " a_m^l " o parámetro de interacción binaria " s_{ij} "), lo cual puede ser representado como dos redes neuronales [2, 20, 10, 1]). Una representación del diagrama de la red neuronal se muestra en la Figura 55.

Para la presente tesis se decidió estudiar la red neuronal con 2 funciones de activación: ReLu y Tangente hiperbólica.

Figura 55

Red neuronal [2, 20, 10, 1]

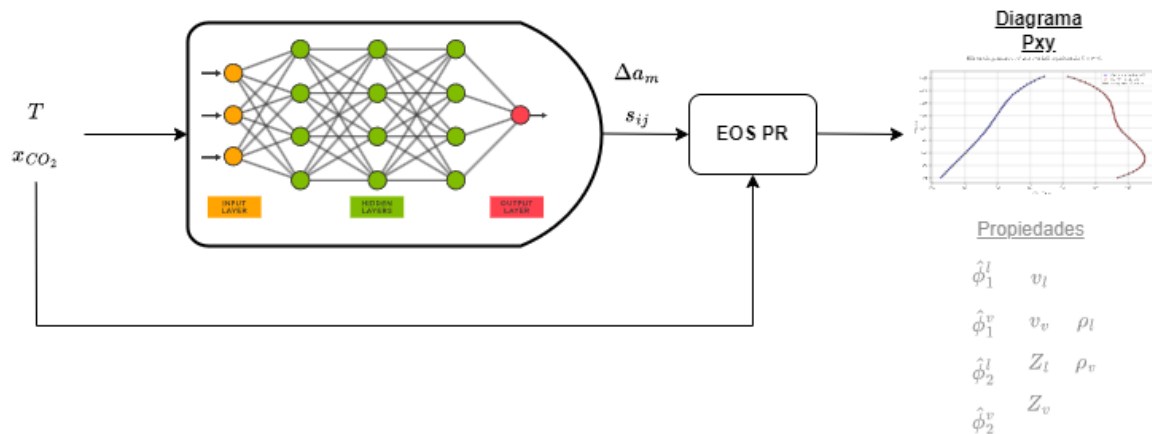


En la Figura 56 podemos ver un resumen de la arquitectura neuronal, las entradas para entrenar las redes neuronales son la temperatura y la fracción molar de CO_2 en fase líquida y la salida de las redes neuronales serán la corrección del parámetro “ a_m^l ” y el

parámetro de interacción binaria “ s_{ij} ”, luego estos parámetros serán utilizados como valores discretos en la EOS – PR para determinar el diagrama P_{xy} utilizando el método fi-fi.

Figura 56

Arquitectura híbrida para la solución del ELV de la mezcla CO_2-H_2O



Nota. Gráfico de redes neuronales basado en (Spotfire | *Neural Networks Explained: A Look at Functions Applications*, slf)

Podemos ver en la Figura 55 que se podrían estimar más propiedades usando el sistema propuesto, pero la presente investigación se centrará en la predicción del diagrama P_{xy} a temperaturas no abarcadas por los datos experimentales.

3.2.6 Entrenamiento de la red neuronal

3.2.6.1 Normalización

Para normalizar la data se toma en cuenta la Ecuación 42, la cual ya está integrada en ScikitLearn usando el objeto StandardScaler, es importante normalizar la data después de separar el conjunto de datos en entrenamiento y validación, para evitar que la información de un subconjunto influya en el otro. Si normalizamos la data antes de separar los datos, podríamos introducir sesgo al proceso, ya que los valores de las medias y desviaciones estándar se calcularían usando la información de todo el conjunto de datos, incluyendo los que deberían estar reservados para prueba o validación. Esto comprometería la evaluación del modelo y podría sobreestimar su rendimiento real.

Para la presente investigación se usará el 80% de los datos como data de entrenamiento y el 20% como data de validación, Se puede ver el código usado para la normalización de data líneas abajo.

```
scaler_X_sij = StandardScaler()
X_train_scaled = scaler_X_sij.fit_transform(X_train)

Y_scaler_sij = StandardScaler()
Y_train_scaled = Y_scaler_sij.fit_transform(Y_train)

X_test_scaled = scaler_X_sij.transform(X_test)
Y_test_scaled = Y_scaler_sij.transform(Y_test)
```

3.2.6.2 Optimización de hiperparámetros

Para la presente tesis se utilizó la misma función de activación en todas las neuronas para cada red neuronal, por lo que se tuvo 4 combinaciones para las redes neuronales para predecir s_{ij} y Δa_m^l respectivamente: ReLu-ReLU, TanH-TanH, ReLU-TanH, TanH-ReLU. Además, se estableció 150 iteraciones (epochs) para el entrenamiento de la red neuronal para predecir s_{ij} y 200 iteraciones para predecir Δa_m^l .

3.2.6.2.1 ReLU-ReLU

En la Figura 57 se puede observar la perdida versus el número de iteraciones para la red neuronal que predecirá s_{ij} . Los valores de pesos y biases obtenidos de la red neuronal se pueden ver en el Anexo 13.

En la Figura 58 se puede observar la perdida versus el número de iteraciones para la red neuronal que predecirá Δa_m^l . Los valores de pesos y biases obtenidos de la red neuronal se pueden ver en el Anexo 14.

Figura 57

Gráfico de pérdida en entrenamiento y validación para predecir s_{ij} (ReLU)

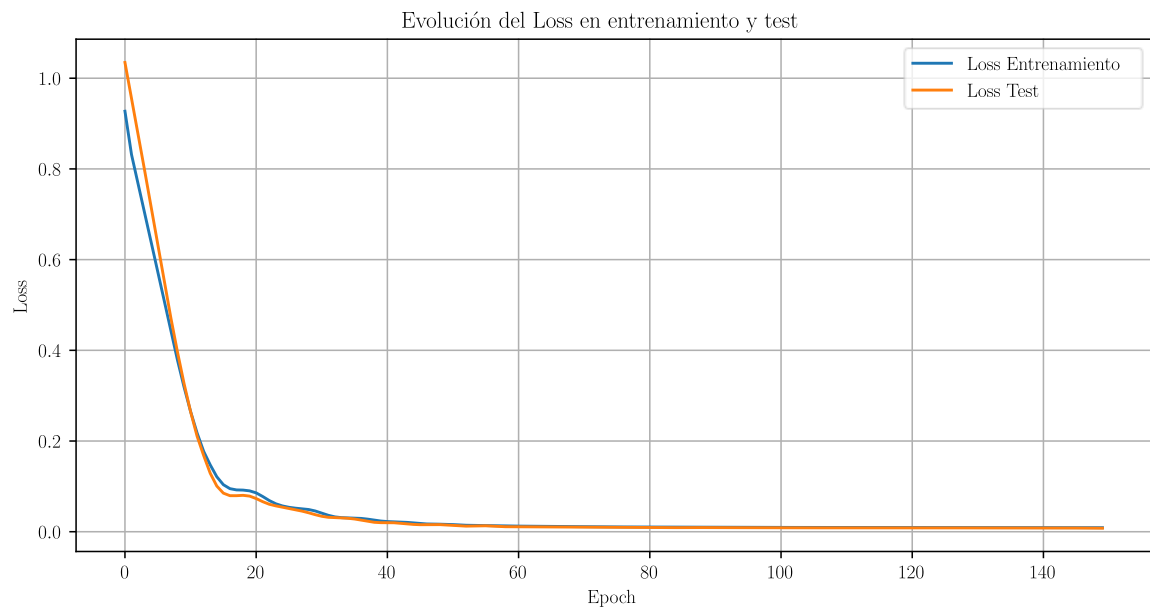
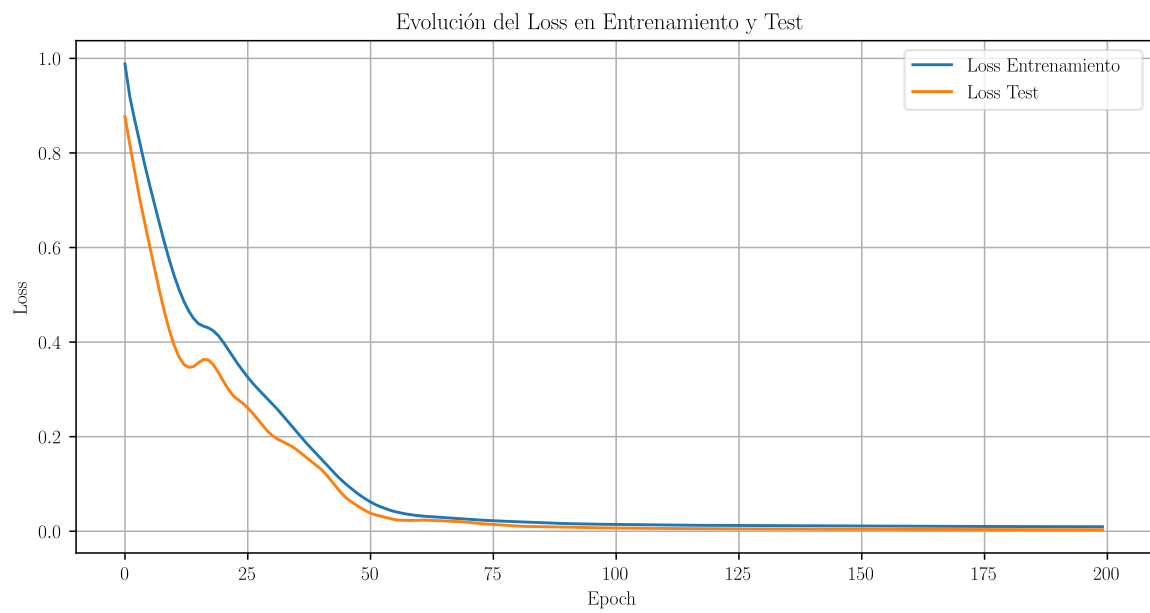


Figura 58

Gráfico de pérdida en entrenamiento y validación para predecir Δa_m^l (ReLU)



3.2.6.2.2 Tan-TanH

En la Figura 59 se puede observar la perdida versus el número de iteraciones para la red neuronal que predecirá s_{ij} . Los valores de pesos y biases obtenidos de la red neuronal se pueden ver en el Anexo 15.

En la Figura 60 se puede observar la perdida versus el número de iteraciones para la red neuronal que predecirá Δa_m^l . Los valores de pesos y biases obtenidos de la red neuronal se pueden ver en el Anexo 16.

Figura 59

Gráfico de perdida en entrenamiento y validación para predecir s_{ij} (TanH)

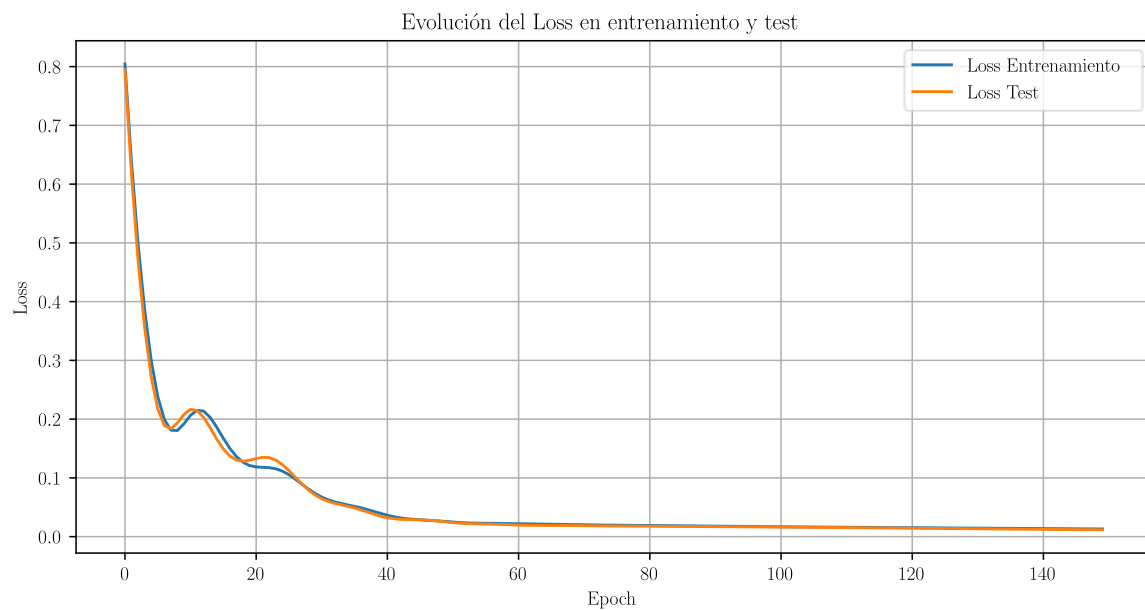
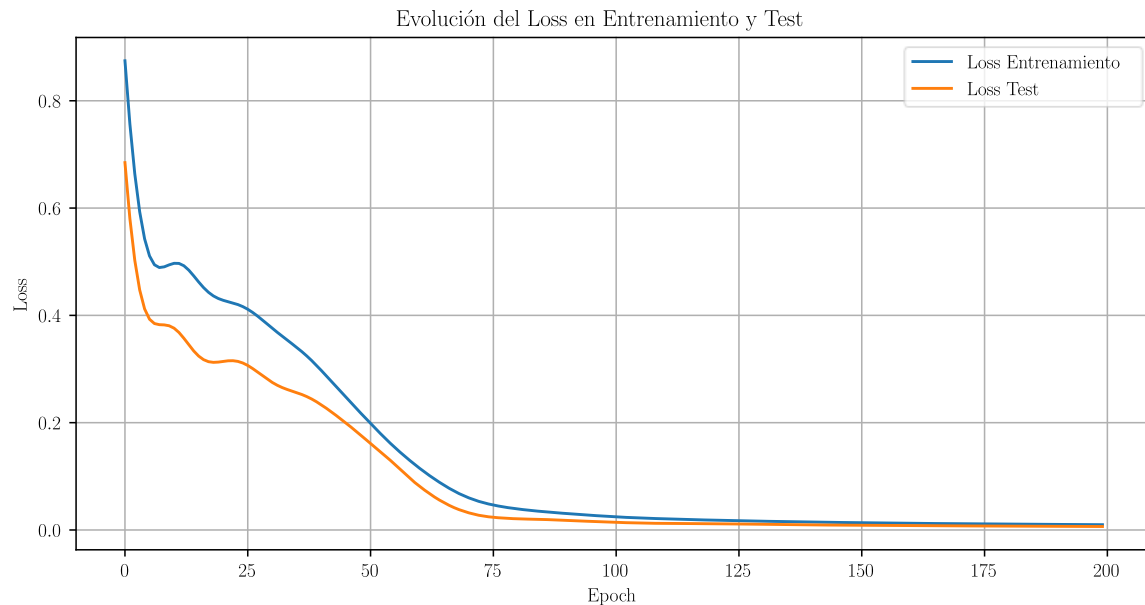


Figura 60

Gráfico de pérdida en entrenamiento y validación para predecir Δa_m^I (TanH)



3.2.6.2.3 ReLU-TanH

Se obtuvo las mismas pérdidas para predecir s_{ij} como se vio en la Figura 56, y las mismas pérdidas para predecir Δa_m^I como se vio en la Figura 59.

3.2.6.2.4 TanH-ReLU

Se obtuvo las mismas pérdidas para predecir s_{ij} como se vio en la Figura 57, y las mismas pérdidas para predecir Δa_m^I como se vio en la Figura 58.

3.2.7 Validación

Finalmente se realizó la validación cualitativa y cuantitativa de la red neuronal.

3.2.7.1 Validación cualitativa

Para la validación cualitativa lo que se busca es que exista concordancia en una isoterma de prueba, es decir si sabemos la posición y forma, de por ejemplo, isothermas a 268°C y 270°C, se esperaría que la isoterma a 269°C este ubicada en la zona comprendida entre ambas isoterma.

Como ejemplo aplicativo analizaremos cómo se comportan s_{ij} y Δa_m^I a 269°C usando las diferentes combinaciones de funciones de activación

3.2.7.1.1 ReLU-ReLU

En la Figura 61 podemos ver el comportamiento de la corrección de s_{ij} a lo largo de la fracción molar de CO_2 .

En la Figura 62 podemos ver el comportamiento de la corrección de Δa_m^I a lo largo de la fracción molar de CO_2 .

Figura 61

Comportamiento de s_{ij} a 269°C usando ReLU como función de activación en capas ocultas

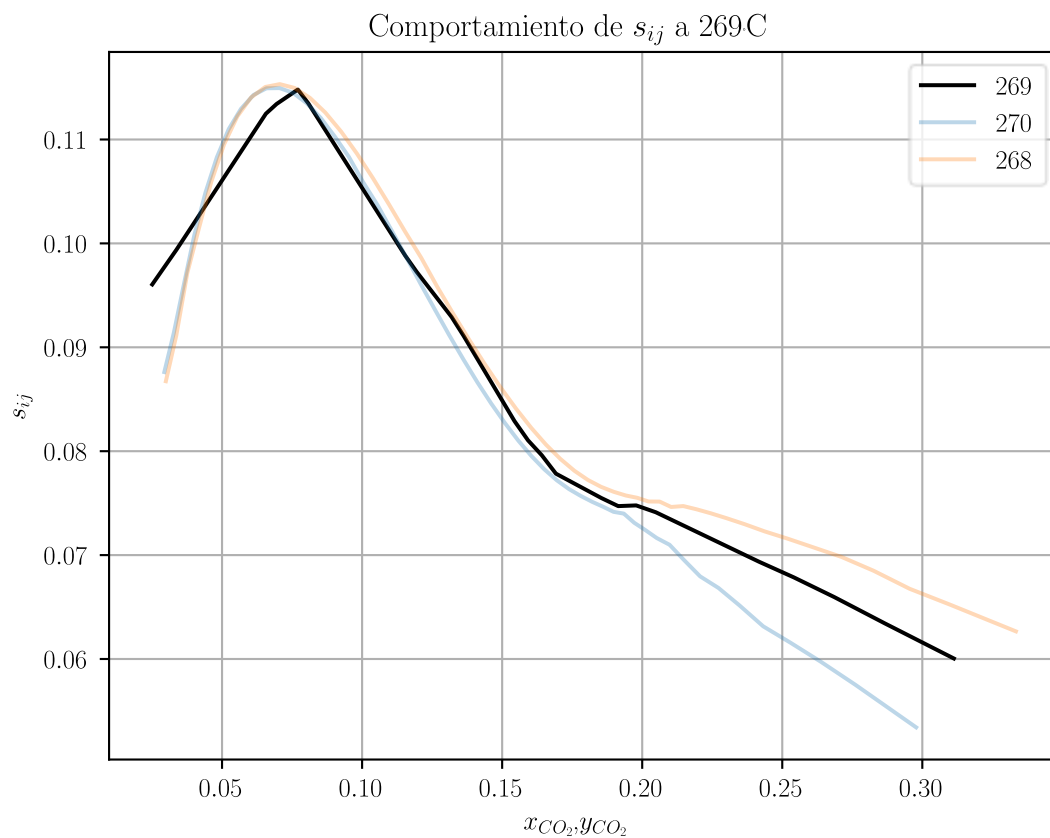
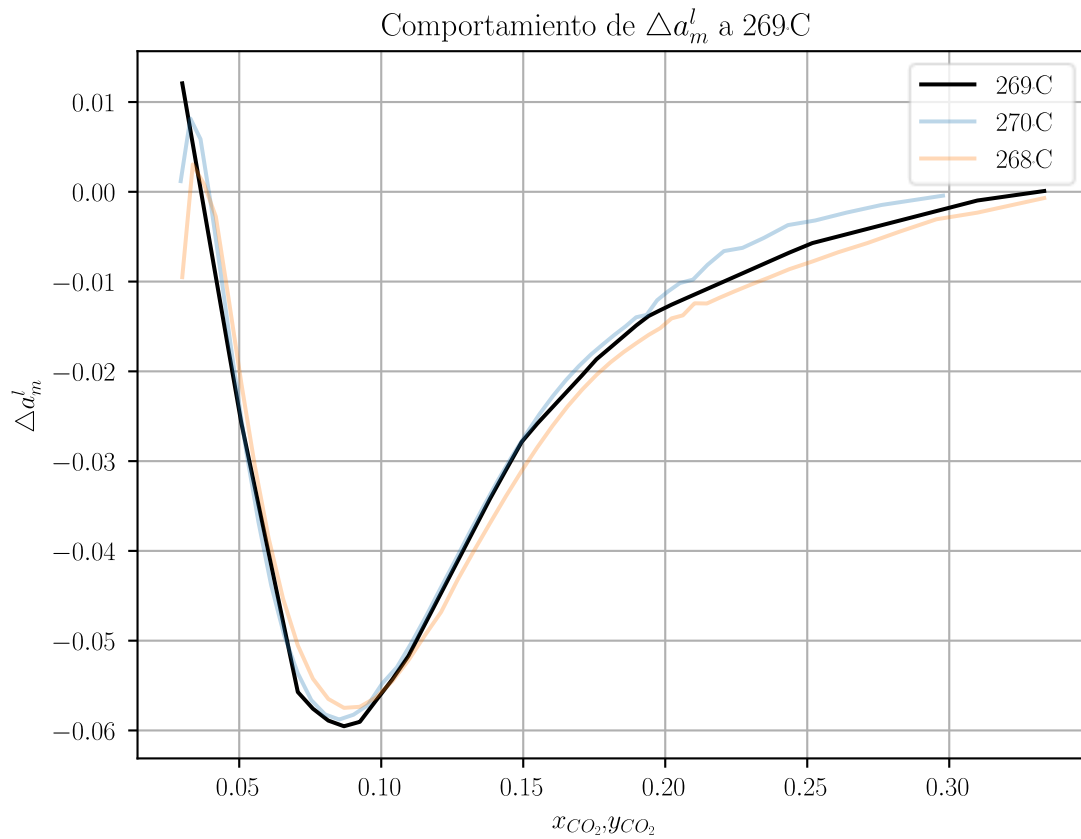


Figura 62

Comportamiento de Δa_m^I a 269°C usando ReLU como función de activación en capas ocultas

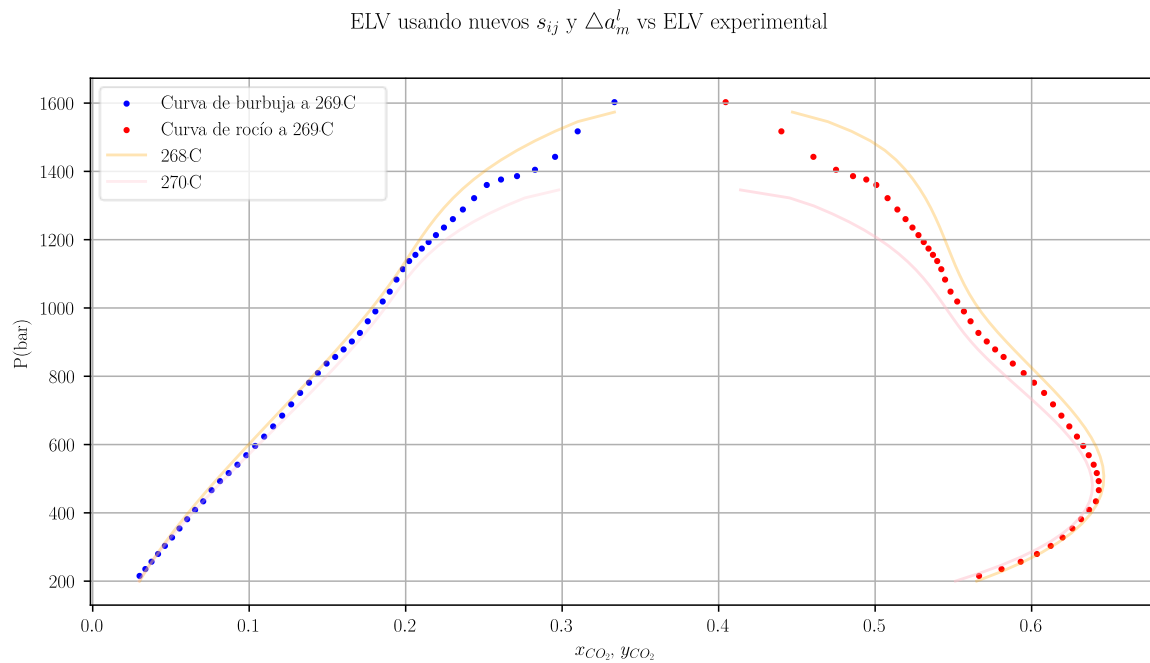


Podemos ver que la red neuronal logra captar el comportamiento del parámetro s_{ij} y Δa_m^I a lo largo de la fracción molar de CO_2 .

Finalmente podemos usar los valores predichos para s_{ij} y Δa_m^I en la solución del ELV, obtenido así el diagrama P_{xy} en la Figura 63.

Figura 63

Diagrama P_{xy} usando s_{ij} y Δa_m^l predichos a 269°C



Podemos ver que la isoterma a 269°C se encuentra entre las isotermas a 268°C y 270°.

3.2.7.1.2 TanH-TanH

En la Figura 64 podemos ver el comportamiento de la corrección de s_{ij} a lo largo de la fracción molar de CO_2 .

En la Figura 65 podemos ver el comportamiento de la corrección de Δa_m^l a lo largo de la fracción molar de CO_2 .

Figura 64

Comportamiento de s_{ij} a 269°C usando Tangente hiperbólica como función de activación en capas ocultas

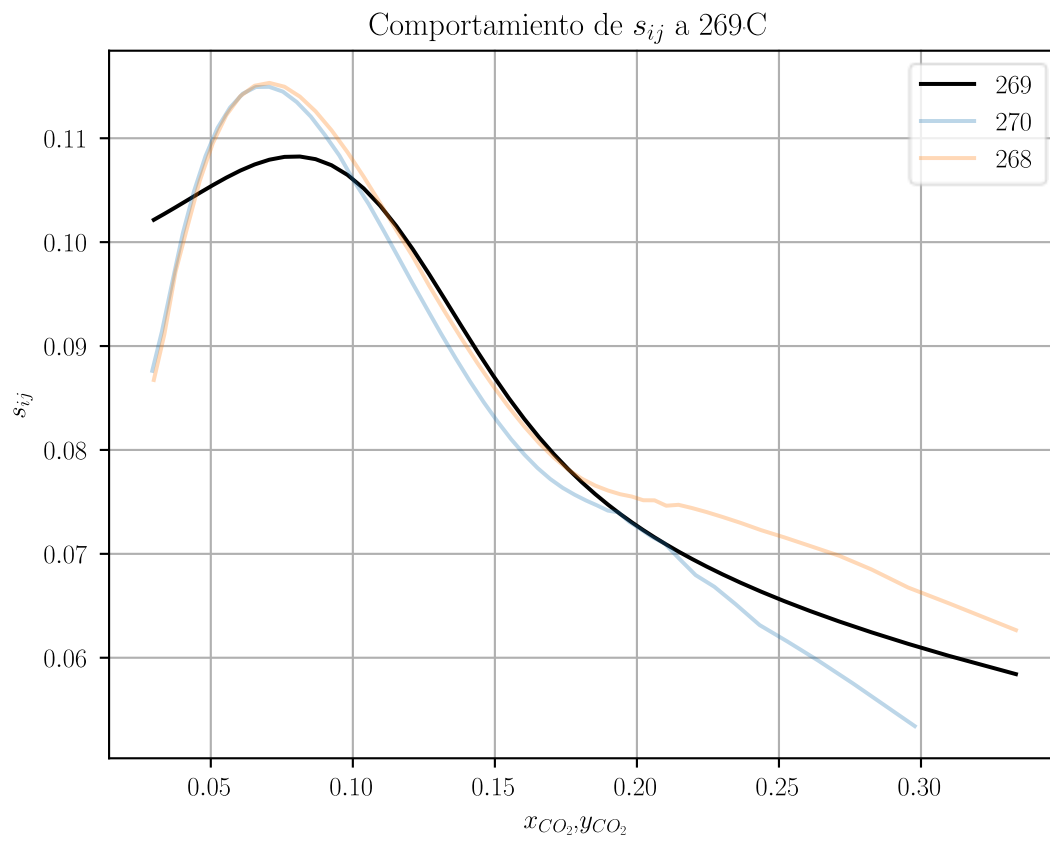
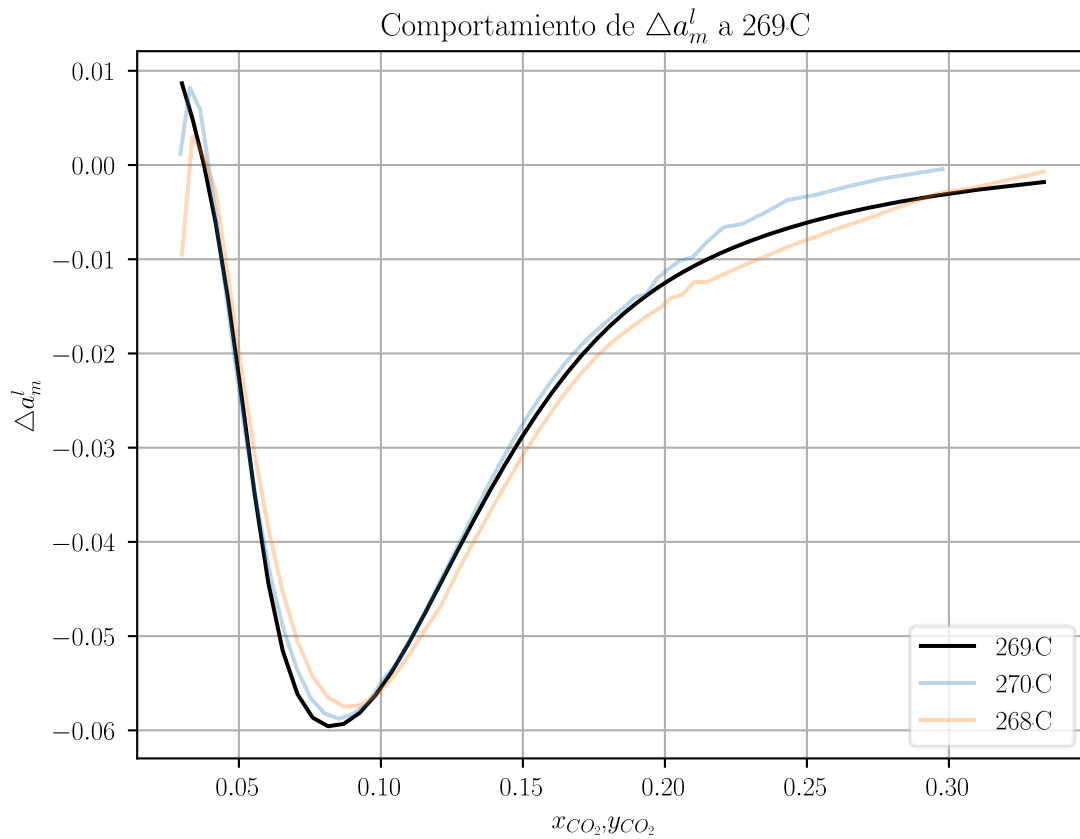


Figura 65

Comportamiento de Δa_m^I a 269°C usando ReLU como función de activación en capas ocultas

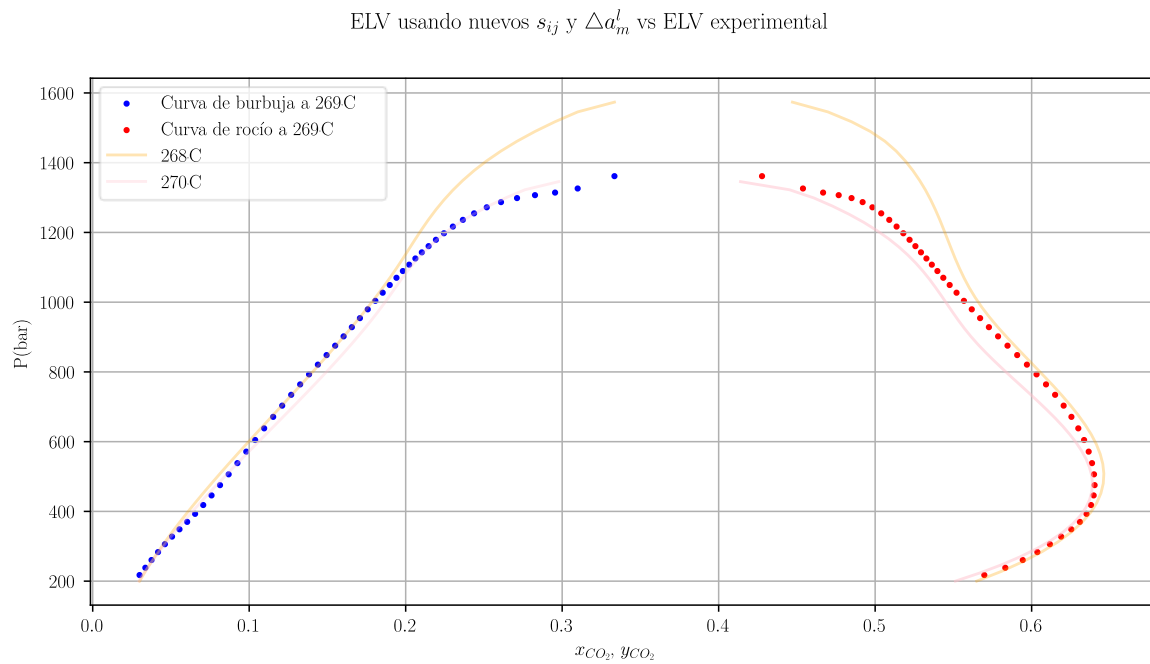


Podemos ver que la red neuronal logra captar el comportamiento del parámetro s_{ij} y Δa_m^I a lo largo de la fracción molar de CO_2 .

Finalmente podemos usar los valores predichos para s_{ij} y Δa_m^I en la solución del ELV, obtenido así el diagrama P_{xy} en la Figura 66.

Figura 66

Diagrama P_{xy} usando s_{ij} y Δa_m^l predichos a 269°C

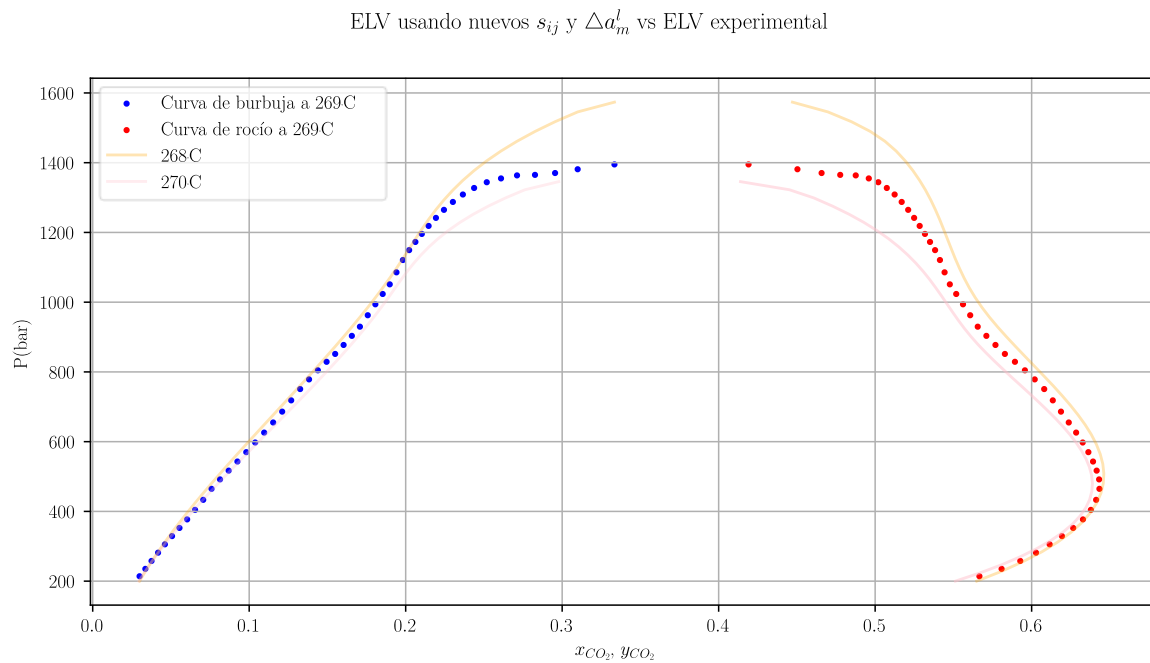


3.2.7.1.3 ReLU-TanH

Se obtuvo los valores predichos por la red neuronal para s_{ij} tal como lo vimos en la Figura 61 y los valores predichos por la red neuronal para Δa_m^l tal como lo vimos en la Figura 65. Luego podemos usar estas predicciones en la solución del ELV, obtenido así el diagrama P_{xy} en la Figura 67.

Figura 67

Diagrama P_{xy} usando s_{ij} y Δa_m^l predichos a 269°C

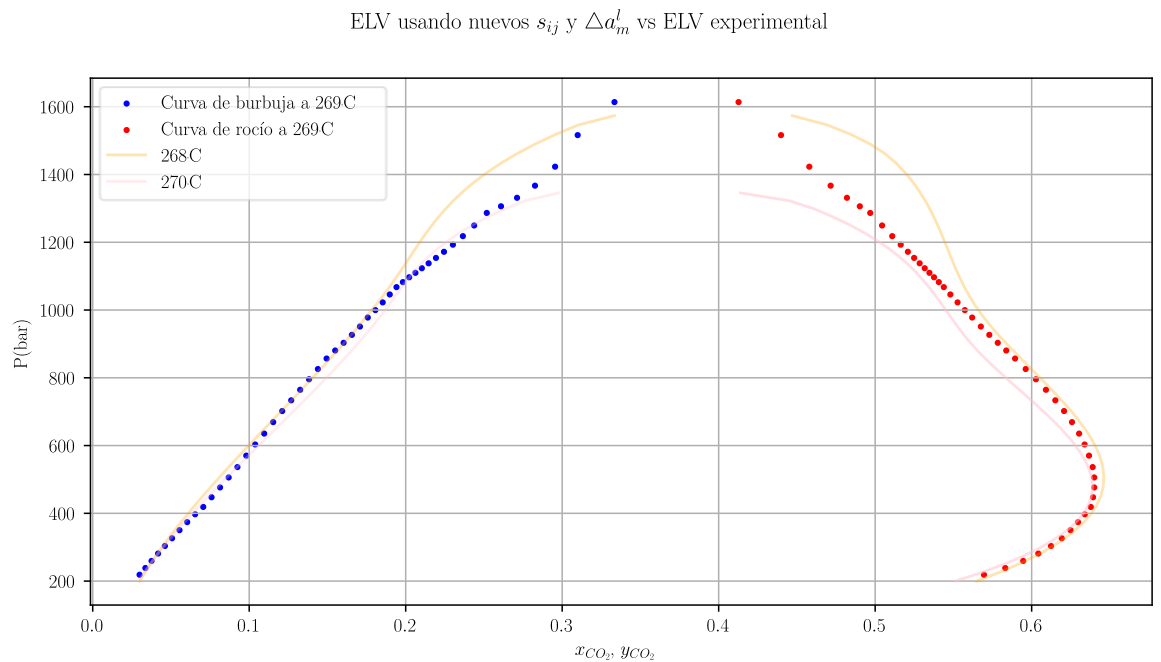


3.2.7.1.4 TanH-ReLU

Se obtuvo los valores predichos por la red neuronal para s_{ij} tal como lo vimos en la Figura 62 y los valores predichos por la red neuronal para Δa_m^l tal como lo vimos en la Figura 64. Luego podemos usar estas predicciones en la solución del ELV, obtenido así el diagrama P_{xy} en la Figura 68.

Figura 68

Diagrama P_{xy} usando s_{ij} y Δa_m^I predichos a 269°C



3.2.7.2 Validación cuantitativa

Para la validación cuantitativa, se usarán los errores MSE y MAPE.

En la Tabla 9 se pueden revisar los errores obtenidos al predecir la fracción molar en fase vapor de CO₂ y la presión de equilibrio, para cada combinación neuronal en la predicción de las isotermas experimentales de Tödheide & Franck (1963).

Tabla 9
Errores MSE y MAPE para cada estructura neuronal.

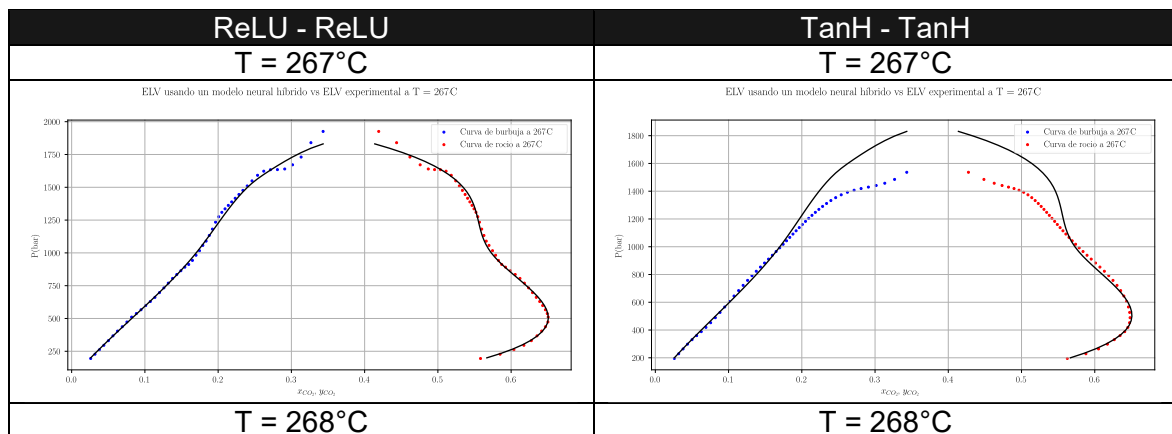
T	Error	Tipo de combinación			
		ReLU-ReLU	TanH-TanH	ReLU-TanH	TanH-ReLU
267°C	MSE _y	1.65E-05	6.43E-05	2.52E-05	4.94E-05
268°C		3.13E-05	4.48E-05	2.48E-05	6.14E-05
270°C		8.85E-05	1.07E-04	9.97E-05	8.71E-05
275°C		1.98E-05	6.65E-05	3.55E-05	3.72E-05
300°C		1.18E-05	1.07E-04	1.35E-04	4.69E-05
350°C		9.85E-04	1.06E-04	9.06E-05	7.60E-04
267°C	MSE _P	6.53E+02	1.60E+04	2.77E+03	8.47E+03
268°C		6.11E+02	4.47E+03	1.33E+03	2.09E+03
270°C		2.08E+02	9.30E+02	3.09E+02	7.66E+02
275°C		2.58E+02	3.57E+02	4.41E+02	8.55E+02
300°C		1.44E+01	3.34E+02	7.82E+01	1.10E+02
350°C		9.96E+05	2.03E+01	1.83E+01	5.56E+05
267°C	MSE _{P,y}	6.53E+02	1.60E+04	2.77E+03	8.47E+03

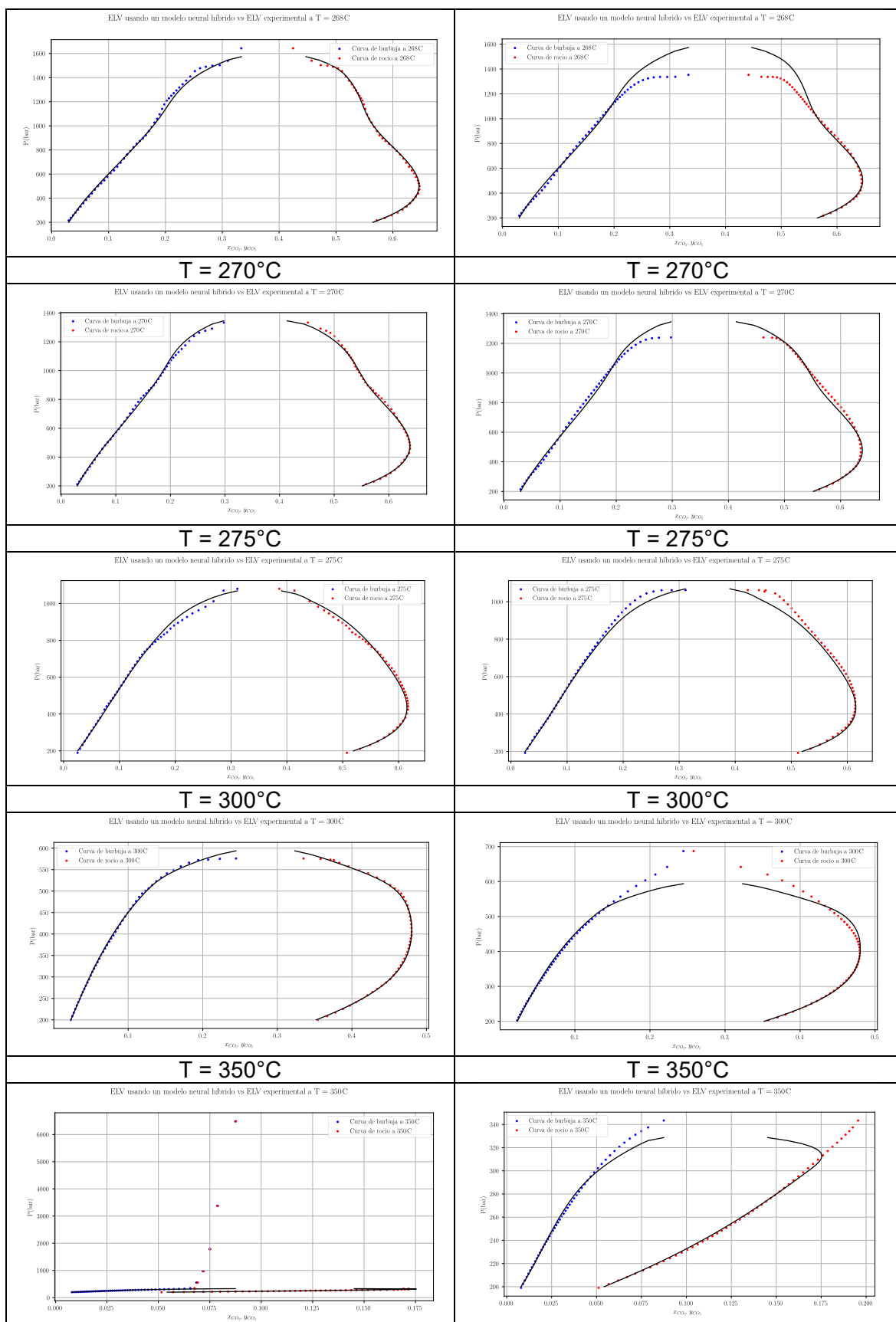
268°C		6.11E+02	4.47E+03	1.33E+03	2.09E+03
270°C		2.08E+02	9.30E+02	3.09E+02	7.66E+02
275°C		2.58E+02	3.57E+02	4.41E+02	8.55E+02
300°C		1.44E+01	3.34E+02	7.82E+01	1.10E+02
350°C		9.96E+05	2.03E+01	1.83E+01	5.56E+05
267°C	MAPE _y	0.75%	1.39%	0.91%	1.22%
268°C		0.91%	1.14%	0.90%	1.34%
270°C		1.24%	1.40%	1.25%	1.39%
275°C		0.75%	1.23%	1.00%	1.00%
300°C		0.49%	1.18%	0.97%	0.94%
350°C		3.25%	1.21%	1.16%	2.77%
267°C	MAPE _p	3.75%	16.16%	6.42%	12.81%
268°C		4.46%	8.88%	5.79%	7.61%
270°C		2.61%	5.46%	2.25%	5.40%
275°C		2.56%	2.82%	2.81%	3.58%
300°C		0.57%	1.90%	0.99%	1.65%
350°C		39.89%	0.62%	0.62%	29.71%
267°C	MAPE _{P,y}	2.25%	8.78%	3.66%	7.01%
268°C		2.68%	5.01%	3.34%	4.47%
270°C		1.92%	3.43%	1.75%	3.39%
275°C		1.65%	2.03%	1.90%	2.29%
300°C		0.53%	1.54%	0.98%	1.29%
350°C		21.57%	0.92%	0.89%	16.24%

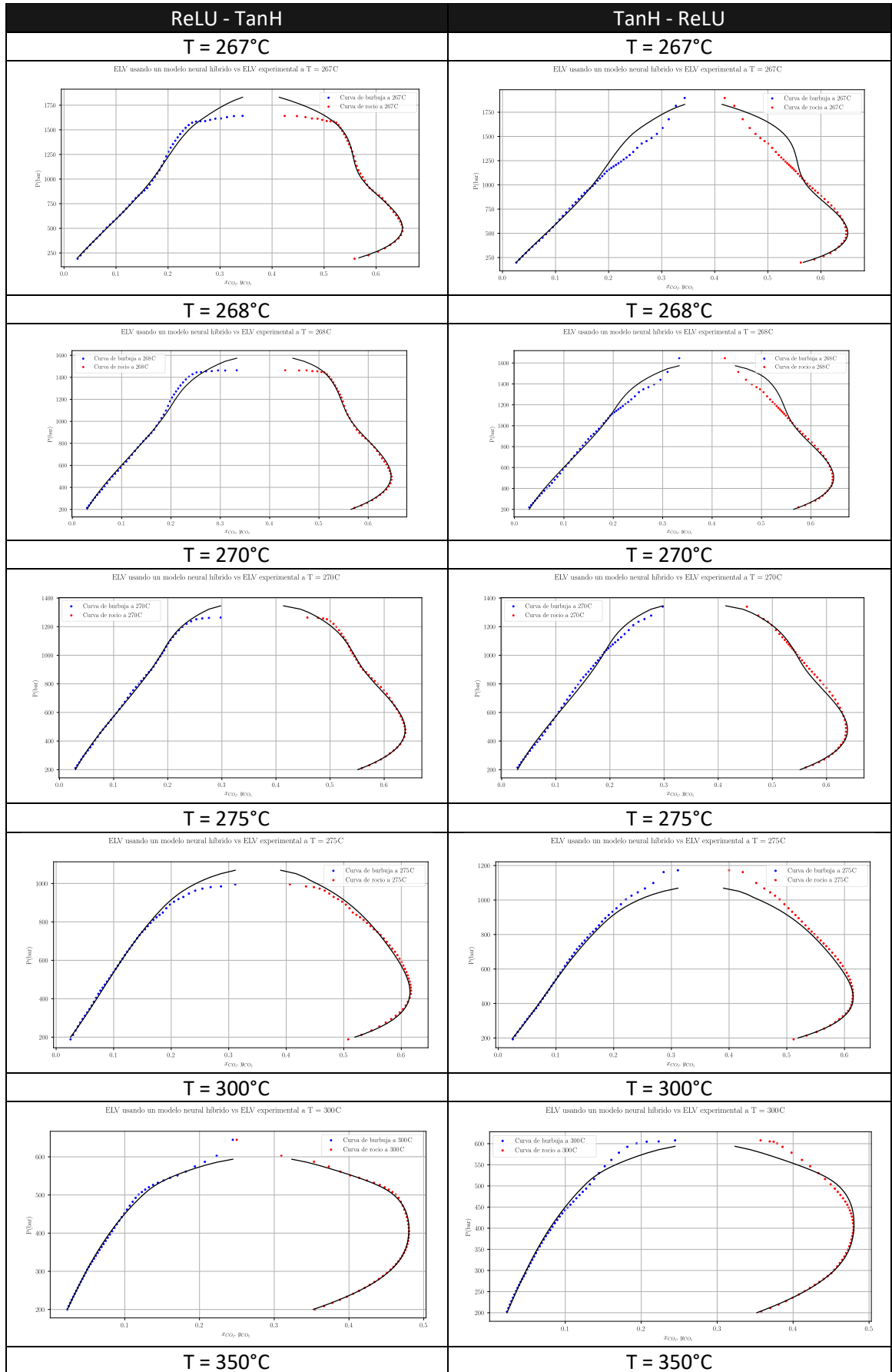
Gráficamente podemos ver el ELV usando las estimaciones de s_{ij} y a_m^l predichas por las redes neuronales y en cada combinación neuronal en la Figura 69.

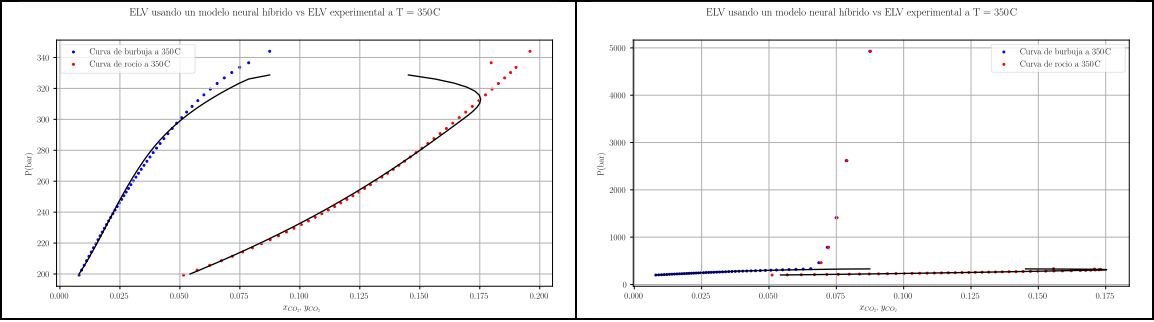
Figura 69

Comparación de los diagramas P_{xy} obtenidos de las diferentes combinaciones neuronales









Capítulo IV. Análisis y discusión de resultados

La poca data experimental del equilibrio líquido vapor de la mezcla $\text{CO}_2\text{-H}_2\text{O}$ incentiva a buscar métodos de aumentación de data, se encontró que una técnica efectiva para aumentar y parametrizar datos experimentales son las curvas suavizadas mediante splines paramétricos, que luego se utilizaron para entrenar redes neuronales. Esta metodología ha demostrado ser superior en la representación de la continuidad y suavidad de los datos en comparación con técnicas más simples, como la interpolación lineal. Esta observación es consistente con lo reportado por Carranza-Abaid et al. (2020), quienes compararon interpolaciones lineales y basadas en splines para el equilibrio líquido-vapor (ELV). En su estudio, los autores concluyeron que las interpolaciones mediante splines ofrecían curvas significativamente más suaves, lo que permite captar mejor las transiciones continuas en el comportamiento del sistema. Esto es particularmente importante en la modelización del ELV, donde la precisión en la representación de los datos es crítica para lograr predicciones confiables mediante modelos basados en data como las redes neuronales.

En la presente investigación no se calculó en número optimo de capas ocultas y neuronas sino se buscó investigaciones donde mencionen el numero optimo de capas y neuronas. Las arquitecturas neuronales propuestas [2, 20, 10, 1] concuerdan con las arquitecturas de dos capas ocultas realizadas por Nguyen et al. (2007) donde muestra que las redes neuronales con dos capas ocultas pueden capturar las relaciones no lineales complejas presentes en sistemas termodinámicos. Esto se debe a que las dos capas ocultas permiten una mayor flexibilidad en el ajuste de los hiperparámetros del modelo, logrando aproximar funciones más complejas. Además, el número total de neuronas concuerda con el trabajo realizado por Carranza-Abaid et al. (2020) donde determinaron que el número optimo de neuronas en la capa oculta para modelar el ELV es 30, lo cual brinda una red neuronal capaz de capturar el comportamiento de la variable de salida sin

llegar a sobreajustar el modelo. Por otro lado, la selección de las funciones de activación concuerda con Lopez-Ramirez et al. (2023) donde demostró que las funciones de activación ReLU generan predicciones razonables del equilibrio de fases sin la necesidad de un costo computacional elevado, mientras que la función de activación TanH concuerda con Narendranath (2020) donde obtienen errores menores al 5% en la estimación de propiedades termodinámicas. La arquitectura neuronal propuesta confirma esta tendencia, donde dos redes neuronales densas con 2 capas ocultas, 30 neuronas, y funciones de activación TanH (Para la predicción de s_{ij}) y ReLU (Para la predicción de a_{mI}) muestran una precisión superior al 97% en la predicción de las isothermas del sistema $\text{CO}_2\text{-H}_2\text{O}$.

Para el entrenamiento de la red neuronal se utilizó el optimizador Adam y esto concuerda con Lopez-Ramirez et al. (2023) donde mencionan que el optimizador Adam logra establecer los gradientes de descenso lo que permite evitar mínimos locales y calcular el mínimo global. Los errores MAPE y MSE se utilizaron para predecir las isothermas de Tödheide & Franck (1963) debido a que diversos autores como Argatov & Kocherbitov (2019) o Vaferi et al. (2018) usan estas medidas de error para validar los resultados de predicciones del ELV usando redes neuronales. Se encontró que el error MSE es muy grande cuando se mide para las presiones de equilibrio (entre $1.44\text{E}+01$ a $9.96\text{E}+05$) y muy pequeño para las fracciones molares de CO_2 en fase vapor (entre $1.18\text{E}-05$ a $9.85\text{E}-04$) por lo que al calcular el MSE de ambos mediante la Ecuación 55 se obtuvo un MSE combinado que prácticamente ignoraba el error de la fracción molar de CO_2 en fase vapor (entre $1.44\text{E}+01$ a $9.96\text{E}+05$), esto es debido a que el MSE no es un error normalizado por lo que se prefiere hacer un análisis con un error normalizado como MAPE donde se obtuvo un MAPE combinado promedio mínimo de 2.09% usando la estructura neuronal TanH-ReLU (Función de activación tangente hiperbólica para la red neuronal que estima s_{ij} y función de activación ReLU para la red neuronal que estima Δa_m^I) para un rango de temperaturas entre 267°C a 350°C . Por otro lado si se quiere tener una estimación en

un rango más restringido de temperaturas entre 267°C a 300°C se puede utilizar la estructura neuronal ReLU-ReLU (Función de activación ReLU para la red neuronal que estima s_{ij} y función de activación ReLU para la red neuronal que estima Δa_m^l) dado que se obtuvo un MAPE combinado promedio mínimo de 1.81%. Estos resultados muestran una mejor predicción del ELV con respecto a modelamiento del ELV de la mezcla CO₂-H₂O utilizando ecuaciones de estado cubicas más asociación desarrollado por Monteiro et al. (2020), donde obtuvieron un error MAPE de 4.63% para la predicción de la fracción molar de CO₂ en fase vapor.

Conclusiones

5.1 Conclusión general

Se realizó un modelamiento híbrido neuronal basado en la ecuación de estado de Peng-Robinson para predecir el equilibrio líquido-vapor de la mezcla $\text{CO}_2\text{--H}_2\text{O}$. Este modelamiento híbrido representa una nueva manera de integrar enfoques computacionales avanzados en el análisis termodinámico, combinando la ecuación de estado con la capacidad de aprendizaje de las redes neuronales. Se alcanzó una precisión superior al 97% en las predicciones de la fracción molar de CO_2 en fase vapor y presión de equilibrio, este enfoque no solo mejora la exactitud de las predicciones, sino que también proporciona una herramienta versátil para la simulación de sistemas complejos en ingeniería química.

5.1.1 Conclusión Específica 1

La parametrización de datos experimentales del equilibrio líquido vapor de la mezcla $\text{CO}_2\text{--H}_2\text{O}$ mediante el uso de splines paramétricos facilitó a las redes neuronales la identificación de tendencias y comportamientos en regiones donde los datos experimentales fueron escasos. Se logró aumentar la data de 96 puntos experimentales en el diagrama P_{xy} a 600 puntos usando splines paramétricos.

5.1.2 Conclusión Específica 2

Las redes neuronales propuestas para la predicción del parámetro de interacción binaria s_{ij} de la regla de mezcla de Van der Waals y la corrección del parámetro “a” de mezcla en fase líquida en un rango de temperaturas entre 267°C a 350°C consisten en:

- Una red neuronal con datos de entrada de fracción molar de CO_2 en fase líquida y temperatura de equilibrio; 20 neuronas en la primera capa oculta y 10 neuronas en la segunda capa oculta, todas las neuronas de las capas

ocultas con función de activación tangente hiperbólico; Una salida para el parámetro de interacción binaria s_{ij} de la regla de mezcla de Van der Waals.

- Una red neuronal con datos de entrada de fracción molar de CO_2 en fase líquida y temperatura de equilibrio; 20 neuronas en la primera capa oculta y 10 neuronas en la segunda capa oculta, todas las neuronas de las capas ocultas con función de activación ReLU; Una salida para la corrección del parámetro “a” de mezcla en fase líquida.

Si se desea resolver el equilibrio líquido-vapor en un rango más restringido de temperaturas entre 267°C a 300°C las redes neuronales propuestas para la predicción del parámetro de interacción binaria s_{ij} de la regla de mezcla de Van der Waals y la corrección del parámetro “a” de mezcla en fase líquida consisten en:

- Una red neuronal con datos de entrada de fracción molar de CO_2 en fase líquida y temperatura de equilibrio; 20 neuronas en la primera capa oculta y 10 neuronas en la segunda capa oculta, todas las neuronas de las capas ocultas con función de activación ReLU; Una salida para el parámetro de interacción binaria s_{ij} de la regla de mezcla de Van der Waals.
- Una red neuronal con datos de entrada de fracción molar de CO_2 en fase líquida y temperatura de equilibrio; 20 neuronas en la primera capa oculta y 10 neuronas en la segunda capa oculta, todas las neuronas de las capas ocultas con función de activación ReLU; Una salida para la corrección del parámetro “a” de mezcla en fase líquida.

5.1.3 Conclusión Específica 3

Los errores MAPE obtenidos en la predicción de isothermas entre 267°C a 350°C usando las combinaciones neuronales ReLU-ReLU, TanH-TanH, ReLU-TanH, TanH-ReLU fueron de 5.10%, 3.62%, 2.09% y 5.78% respectivamente, por lo que la combinación optima de funciones de activación es ReLU-TanH, mientras que si se analizan las predicciones de las isothermas en un rango más restringido de temperatura entre 267°C a 300°C, se obtiene errores MAPE de 1.81%, 4.16%, 2.33% y 3.69% respectivamente, por

lo que la combinación óptima de funciones de activación para un rango más restringido de temperaturas es ReLU-ReLU.

Recomendaciones

Luego de haber desarrollado la presente investigación se sugiere las siguientes recomendaciones.

- Se sugiere usar el nuevo algoritmo descrito en la sección 3.1.3.7 para determinar los parámetros s_{ij} y a_m^l de la ecuación de estado de Peng-Robinson para cualquier mezcla que no se logre ajustar correctamente a la ecuación de estado, y así poder analizar el comportamiento de estos parámetros para posteriormente estimar regresiones para crear términos correctivos o nuevas reglas de mezcla en la ecuación de estado de Peng-Robinson.
- Se recomienda usar el nuevo algoritmo descrito en la sección 3.1.3.7 con otras ecuaciones de estado más orientadas al modelamiento de mezclas a temperaturas supercríticas para uno de sus componentes para así evaluar su aplicabilidad.
- En la presente investigación se restringió la temperatura de estudio ya que, a temperaturas menores, las curvas de burbuja tenían un comportamiento no funcional lo que dificultaba la implementación de red neuronal densas, por lo que se sugiere que para futuras investigaciones se use una metodología basada en densidades de probabilidad como Mixture Density Networks.

Próximas Investigaciones

En el desarrollo de la presente investigación se identificaron algunas ideas de investigación que se podrían desarrollar basándose en la presente tesis:

- Data experimental del ELV de la mezcla $\text{CO}_2 - \text{H}_2\text{O}$ para temperaturas entre 267°C a 350°C : Se podrían desarrollar investigaciones experimentales del ELV para isothermas dentro del rango de temperaturas estudiado, con la finalidad de realizar una validación de la red neuronal propuesta con isothermas diferentes a las estudiadas por Tödheide & Franck (1963).
- Optimización de arquitecturas neuronales para el modelado del ELV de la mezcla $\text{CO}_2 - \text{H}_2\text{O}$: Se podría optimizar el número de capas ocultas, número de neuronas, diferentes funciones de activación para cada neurona o incluso el uso de redes neuronales dispersas para conseguir una mejor precisión en la solución del ELV de la mezcla $\text{CO}_2 - \text{H}_2\text{O}$.
- Modelamiento neuronal híbrido para el ELV basado en la EOS-PR para mezclas binarias a temperaturas supercríticas para uno de sus componentes: Esto no restringiría el uso del modelo híbrido a una sola mezcla, sino que podría ser aplicable a una variedad de mezclas. Se podría realizar un análisis similar al desarrollado, pero con más mezclas con condiciones similares parecidas a este trabajo, a temperaturas supercríticas o cercanas a las supercríticas para uno de sus componentes como: $\text{CO}_2 - \text{CH}_4$, $\text{CO}_2 - \text{N}_2$, $\text{CH}_4 - \text{H}_2\text{O}$, entre otras.
- Determinación de una nueva regla de mezcla en la EOS – PR para la mezcla $\text{CO}_2 - \text{H}_2\text{O}$: El algoritmo desarrollado en la sección 3.1.3.7 puede ser muy útil en la búsqueda de soluciones de parámetros de la EOS – PR, por lo que al tener el comportamiento de las soluciones de los parámetros basándose en datos experimentales se podría desarrollar una nueva regla de mezcla.
- Predicción de propiedades volumétricas y termodinámicas de la mezcla $\text{CO}_2 - \text{H}_2\text{O}$ mediante modelamiento neuronal híbrido basado en la EOS – PR: Al determinar

los parámetros s_{ij} y la corrección del parámetro “a” de mezcla en fase líquida se tuvieron que realizar pasos intermedios como el cálculo del factor de compresibilidad, y la densidad molar. Una próxima investigación podría enfocarse en las predicciones de estas propiedades o incluso predecir otras propiedades termodinámicas como la energía libre de Gibbs, entalpía molar u otras.

- Nuevo modelo neuronal híbrido con consistencia termodinámica para el ELV de la mezcla $\text{CO}_2 - \text{H}_2\text{O}$: Se podría realizar un análisis de consistencia termodinámica del modelo neuronal híbrido. Debido a que la EOS-PR, por la naturaleza experimental de la función alfa, no es termodinámicamente consistente, se podría explorar otras EOS, modificar la función alfa o agregar restricciones al modelo neuronal para que haga predicciones termodinámicamente consistentes.

Referencias bibliográficas

16.2: van der Waals and Redlich-Kwong Equations of State - Chemistry LibreTexts. (s/f).

Recuperado el 16 de agosto de 2024, de [https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Physical_Chemistry_\(LibreTexts\)/16%3A_The_Properties_of_Gases/16.02%3A_van_der_Waals_and_Redlich-Kwong_Equations_of_State](https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Physical_Chemistry_(LibreTexts)/16%3A_The_Properties_of_Gases/16.02%3A_van_der_Waals_and_Redlich-Kwong_Equations_of_State)

Ahmed, T. (2016). Equations of State and Phase Equilibria. *Equations of State and PVT Analysis*, 467–597. <https://doi.org/10.1016/B978-0-12-801570-4.00005-2>

Amer, H. H., Paxton, R. R., & Winkle, M. Van. (2002). Methanol-Ethanol-Acetone. *Industrial & Engineering Chemistry*, 48(1), 142–146. <https://doi.org/10.1021/IE50553A041>

Aminian, A., & ZareNezhad, B. (2020). A generalized neural network model for the VLE of supercritical carbon dioxide fluid extraction of fatty oils. *Fuel*, 282, 118823. <https://doi.org/10.1016/J.FUEL.2020.118823>

Araújo, O. A. S., Silva, F. R., Ramos, L. P., Lenzi, M. K., Ndiaye, P. M., & Corazza, M. L. (2012). Phase behaviour measurements for the system (carbon dioxide + biodiesel + ethanol) at high pressures. *The Journal of Chemical Thermodynamics*, 47, 412–419. <https://doi.org/10.1016/J.JCT.2011.11.029>

Argatov, I., & Kocherbitov, V. (2019). A note on artificial neural network modeling of vapor-liquid equilibrium in multicomponent mixtures. *Fluid Phase Equilibria*, 502, 112282. <https://doi.org/10.1016/J.FLUID.2019.112282>

Calin, O. (2020). *Deep Learning Architectures*. <https://doi.org/10.1007/978-3-030-36721-3>

- Carnahan, N. F., & Starling, K. E. (1972). Intermolecular repulsions and the equation of state for fluids. *AIChE Journal*, 18(6), 1184–1189. <https://doi.org/10.1002/AIC.690180615>
- Carranza-Abaid, A., Svendsen, H. F., & Jakobsen, J. P. (2020). Surrogate modelling of VLE: Integrating machine learning with thermodynamic constraints. *Chemical Engineering Science: X*, 8, 100080. <https://doi.org/10.1016/J.CESX.2020.100080>
- Carroll, J. (2020). Additional topics. *Natural Gas Hydrates*, 347–360. <https://doi.org/10.1016/B978-0-12-821771-9.00011-2>
- Chiew, Y. C., Jeung, W., Weathers, T., & Plant, S. (2020). Supercritical CO₂ with Water Contamination : Mixture Thermodynamic Properties at Compressor Conditions. *The 7th International Supercritical CO₂ Power Cycles Symposium*, 37.
- Dahm, K. D. ., & Visco, D. P. . (2014). *Chemical engineering thermodynamics*. https://books.google.com/books/about/Fundamentals_of_Chemical_Engineering_The.html?id=QIfAAgAAQBAJ
- Diamantonis, N. I., & Economou, I. G. (2012). Modeling the phase equilibria of a H₂O–CO₂ mixture with PC-SAFT and tPC-PSAFT equations of state. *Molecular Physics*, 110(11–12), 1205–1212. <https://doi.org/10.1080/00268976.2012.656721>
- Dierckx, P. (1982). Algorithms for smoothing data with periodic and parametric splines. *Computer Graphics and Image Processing*, 20(2), 171–184. [https://doi.org/10.1016/0146-664X\(82\)90043-0](https://doi.org/10.1016/0146-664X(82)90043-0)

- Forero, G. L. A., & Velásquez, J. J. A. (2016). A generalized cubic equation of state for non-polar and polar substances. *Fluid Phase Equilibria*, 418, 74–87.
<https://doi.org/10.1016/J.FLUID.2015.09.045>
- Franssen, J. (2016). *Generalizations of the Casus Irreducibilis*.
<https://studenttheses.uu.nl/handle/20.500.12932/23959>
- Gallagher, J. S., Crovetto, R., & Sengers, J. M. H. L. (1993). The Thermodynamic Behavior of the CO₂-H₂O System from 400 to 1000 K, up to 100 Mpa and 30% Mole Fraction of CO₂. *Journal of Physical and Chemical Reference Data*, 22(2), 431–513.
<https://doi.org/10.1063/1.555938>
- Gao, K., Mei, G., Piccialli, F., Cuomo, S., Tu, J., & Huo, Z. (2020). Julia language in machine learning: Algorithms, applications, and open issues. *Computer Science Review*, 37, 100254. <https://doi.org/10.1016/J.COSREV.2020.100254>
- Greenwood, H. J. (1973). Thermodynamic properties of gaseous mixtures of H₂O and CO₂ between 450 degrees and 800 degrees C and 0 to 500 bars. *American Journal of Science*, 273(7), 561–571. <https://doi.org/10.2475/AJS.273.7.561>
- How carbon capture and storage can help us tackle the climate crisis | World Economic Forum*. (s/f). Recuperado el 20 de octubre de 2024, de <https://www.weforum.org/agenda/2024/10/carbon-capture-storage-climate-crisis/>
- Jirasek, F., & Hasse, H. (2023). Combining Machine Learning with Physical Knowledge in Thermodynamic Modeling of Fluid Mixtures. *Annual Review of Chemical and Biomolecular Engineering*, 14(Volume 14, 2023), 31–51.
<https://doi.org/10.1146/ANNUREV-CHEMBIOENG-092220-025342/CITE/REFWORKS>

- Johnsson, R., & Norén, S. (2017). Transient model of a driven compressor. (2017). <http://lup.lub.lu.se/student-papers/record/8917120>
- King, C. J. (2000). FROM UNIT OPERATIONS TO SEPARATION PROCESSES 1. *Separation and Purification Methods*, 29(2), 233–245. <https://doi.org/10.1081/SPM-100100010>
- Kingma, D. P., & Ba, J. L. (2014). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. <https://arxiv.org/abs/1412.6980v9>
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598), 671–680. <https://doi.org/10.1126/SCIENCE.220.4598.671>
- Koretsky, M. D. . (2013). *Engineering and chemical thermodynamics*. 690. <https://www.wiley.com/en->
[ie/Engineering+and+Chemical+Thermodynamics%2C+2nd+Edition-p-](https://www.wiley.com/en-)
[9780470259610](https://www.wiley.com/en-)
- Lopez-Echeverry, J. S., Reif-Acherman, S., & Araujo-Lopez, E. (2017). Peng-Robinson equation of state: 40 years through cubics. *Fluid Phase Equilibria*, 447, 39–71. <https://doi.org/10.1016/J.FLUID.2017.05.007>
- Lopez-Ramirez, E., Lopez-Zamora, S., Escobedo, S., & de Lasa, H. (2023). Artificial Neural Networks (ANNs) for Vapour-Liquid-Liquid Equilibrium (VLLE) Predictions in N-Octane/Water Blends. *Processes* 2023, Vol. 11, Page 2026, 11(7), 2026. <https://doi.org/10.3390/PR11072026>

- Mansour, E. M., & Mansour, E. M. (2020). Equation of State. *Inverse Heat Conduction and Heat Exchangers*. <https://doi.org/10.5772/INTECHOPEN.89919>
- Marmur, A. (1985). Mathematical properties of equations of state. *Chemical Engineering Science*, 40(10), 1881–1884. [https://doi.org/10.1016/0009-2509\(85\)80124-X](https://doi.org/10.1016/0009-2509(85)80124-X)
- Marquardt, W. (2002). Adaptivity in Process Systems Modeling. *Computer Aided Chemical Engineering*, 10(C), 42–56. [https://doi.org/10.1016/S1570-7946\(02\)80038-4](https://doi.org/10.1016/S1570-7946(02)80038-4)
- Mogk, G., Mrziglod, T., & Schuppert, A. (2002). Application of Hybrid Models in Chemical Industry. *Computer Aided Chemical Engineering*, 10(C), 931–936. [https://doi.org/10.1016/S1570-7946\(02\)80183-3](https://doi.org/10.1016/S1570-7946(02)80183-3)
- Mohshim, D. F., Nasir, R., Suleman, H., Sharif, R., & Khairuddin, M. Z. F. (2024). Safety and Leakage of CO₂ Storages. *Encyclopedia of Renewable Energy, Sustainability and the Environment*, 1001–1016. <https://doi.org/10.1016/B978-0-323-93940-9.00078-5>
- Mokhatab, S., Poe, W. A., & Mak, J. Y. (2019). Phase Behavior of Natural Gas Systems. *Handbook of Natural Gas Transmission and Processing*, 37–101. <https://doi.org/10.1016/B978-0-12-815817-3.00002-2>
- Monteiro, M. F., Moura-Neto, M. H., Pereira, C. G., & Chiavone-Filho, O. (2020). Description of phase equilibrium and volumetric properties for CO₂+water and CO₂+ethanol using the CPA equation of state. *The Journal of Supercritical Fluids*, 161, 104841. <https://doi.org/10.1016/J.SUPFLU.2020.104841>
- Montesinos López, O. A., Montesinos López, A., & Crossa, J. (2022). Fundamentals of Artificial Neural Networks and Deep Learning. *Multivariate Statistical Machine*

Learning Methods for Genomic Prediction, 379–425. https://doi.org/10.1007/978-3-030-89010-0_10

Narendranath, A. D. (2020). Introducing Artificial Neural Networks as a Specific Enthalpy approximator for a course on introductory Thermodynamics. *Proceedings - Frontiers in Education Conference, FIE, 2020-October*. <https://doi.org/10.1109/FIE44824.2020.9274222>

Nguyen, V. D., Tan, R. R., Brondial, Y., & Fuchino, T. (2007). Prediction of vapor–liquid equilibrium data for ternary systems using artificial neural networks. *Fluid Phase Equilibria*, 254(1–2), 188–197. <https://doi.org/10.1016/J.FLUID.2007.03.014>

Panagiotopoulos, A. Z., & Reid, R. C. (1986). NEW MIXING RULE FOR CUBIC EQUATIONS OF STATE FOR HIGHLY POLAR, ASYMMETRIC SYSTEMS. *ACS Symposium Series*, 571–582. <https://doi.org/10.1021/BK-1986-0300.CH028>

Pappa, G. D., Perakis, C., Tsimpanogiannis, I. N., & Voutsas, E. C. (2009). Thermodynamic modeling of the vapor–liquid equilibrium of the CO₂/H₂O mixture. *Fluid Phase Equilibria*, 284(1), 56–63. <https://doi.org/10.1016/J.FLUID.2009.06.011>

Paulus, M. E., & Penoncello, S. G. (2006). Correlation for the carbon dioxide and water mixture based on the Lemmon-Jacobsen mixture model and the Peng-Robinson equation of state. *International Journal of Thermophysics*, 27(5), 1373–1386. <https://doi.org/10.1007/S10765-006-0123-5/METRICS>

Peng, D. Y., & Robinson, D. B. (1976). A New Two-Constant Equation of State. *Industrial and Engineering Chemistry Fundamentals*, 15(1), 59–64. https://doi.org/10.1021/I160057A011/ASSET/I160057A011.FP.PNG_V03

Petroli, G., Dalmolin, I., & Brusamarello, C. Z. (2022). Prediction of phase equilibrium between soybean biodiesel, alcohols and supercritical CO₂ using artificial neural networks. *Chemical Thermodynamics and Thermal Analysis*, 6, 100048. <https://doi.org/10.1016/J.CTTA.2022.100048>

Pinto, L. F., Da Silva, D. I. S., Rosa Da Silva, F., Ramos, L. P., Ndiaye, P. M., & Corazza, M. L. (2012). Phase equilibrium data and thermodynamic modeling of the system (CO₂ + biodiesel + methanol) at high pressures. *The Journal of Chemical Thermodynamics*, 44(1), 57–65. <https://doi.org/10.1016/J.JCT.2011.07.019>

Pizzano, A., Reartes, S. B. R., Hegel, P. E., & Brignole, N. B. (2023). Predicción del equilibrio líquido-vapor mediante redes neuronales: sistema PDMS + CO₂. *Memorias de las JAIIO*, 9(15), 95–108. <https://publicaciones.sadio.org.ar/index.php/JAIIO/article/view/758>

Predicting Vapor-Liquid Equilibria Using Physic-Inspired Machine Learning Models | AIChE. (s/f). Recuperado el 14 de agosto de 2024, de <https://www.aiche.org/conferences/aiche-annual-meeting/2023/proceeding/paper/predicting-vapor-liquid-equilibria-using-physic-inspired-machine-learning-models>

Profillidis, V. A., & Botzoris, G. N. (2019). Artificial Intelligence—Neural Network Methods. *Modeling of Transport Demand*, 353–382. <https://doi.org/10.1016/B978-0-12-811513-8.00008-X>

Psichogios, D. C., & Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10), 1499–1511. <https://doi.org/10.1002/AIC.690381003>

- Raj, A. A. B., & Padmavathi, S. (2016). Quality Metrics and Reliability Analysis of Laser Communication System. *Defence Science Journal*, 66(2), 175–185. <https://doi.org/10.14429/DSJ.66.9707>
- Redlich, O., & Kwong, J. N. S. (1949). On the thermodynamics of solutions. V. An equation of state. Fugacities of gaseous solutions. *Chemical Reviews*, 44(1), 233–244. https://doi.org/10.1021/CR60137A013/ASSET/CR60137A013.FP.PNG_V03
- Reinert, W., & Kähler, D. (2010). Hermeticity Tests. *Handbook of Silicon Based MEMS Materials and Technologies*, 607–616. <https://doi.org/10.1016/B978-0-8155-1594-4.00042-5>
- Sanchez, I. C., & Lacombe, R. H. (1976). An elementary molecular theory of classical fluids. Pure fluids. *Journal of Physical Chemistry*, 80(21), 2352–2362. https://doi.org/10.1021/J100562A008/ASSET/J100562A008.FP.PNG_V03
- Sansana, J., Joswiak, M. N., Castillo, I., Wang, Z., Rendall, R., Chiang, L. H., & Reis, M. S. (2021). Recent trends on hybrid modeling for Industry 4.0. *Computers & Chemical Engineering*, 151, 107365. <https://doi.org/10.1016/J.COMPCHEMENG.2021.107365>
- Shaik, M. E., Islam, M. M., & Hossain, Q. S. (2021). A review on neural network techniques for the prediction of road traffic accident severity. *Asian Transport Studies*, 7, 100040. <https://doi.org/10.1016/J.EASTSJ.2021.100040>
- Sharma, S., Sharma, S., & Athaiya, A. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International Journal of Engineering Applied Sciences and Technology*, 04(12), 310–316. <https://doi.org/10.33564/IJEAST.2020.V04I12.054>

- Si-Moussa, C., Hanini, S., Derriche, R., Bouhedda, M., & Bouzidi, A. (2008). Prediction of high-pressure vapor liquid equilibrium of six binary systems, carbon dioxide with six esters, using an artificial neural network model. *Brazilian Journal of Chemical Engineering*, 25(1), 183–199. <https://doi.org/10.1590/S0104-66322008000100019>
- Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97, 105524. <https://doi.org/10.1016/J.ASOC.2019.105524>
- Smith, J. M. ., Van Ness, H. C. ., Abbott, M. M. ., & Swihart, M. T. . (2022). *Introduction to chemical engineering thermodynamics*. 762.
- Softmax Function: Advantages and Applications | BotPenguin*. (s/f). Recuperado el 15 de septiembre de 2024, de <https://botpenguin.com/glossary/softmax-function>
- Song, Y., Chen, B., Nishio, M., & Akai, M. (2005). The study on density change of carbon dioxide seawater solution at high pressure and low temperature. *Energy*, 30(11–12), 2298–2307. <https://doi.org/10.1016/J.ENERGY.2003.10.022>
- Spotfire | Neural Networks Explained: A Look at Functions Applications*. (s/f). Recuperado el 16 de septiembre de 2024, de <https://www.spotfire.com/glossary/what-is-a-neural-network>
- Suman, S. (2021). Artificial intelligence in nuclear industry: Chimera or solution? *Journal of Cleaner Production*, 278, 124022. <https://doi.org/10.1016/J.JCLEPRO.2020.124022>
- Thiesen, M. (1885). Untersuchungen über die Zustandsgleichung. *Annalen der Physik*, 260(3), 467–492. <https://doi.org/10.1002/ANDP.18852600308>

- Tödheide, K., & Franck, E. U. (1963). Das zweiphasengebiet und die kritische kurve im system kohlendioxid-wasser bis zu drucken von 3500 bar. *Zeitschrift für Physikalische Chemie*, 37(5–6), 387–401.
https://doi.org/10.1524/ZPCH.1963.37.5_6.387/MACHINEREADABLECITATION/RIS
- Vaferi, B., Lashkarbolooki, M., Esmaeili, H., & Shariati, A. (2018). Toward artificial intelligence-based modeling of vapor liquid equilibria of carbon dioxide and refrigerant binary systems. *Journal of the Serbian Chemical Society*, 83(2), 199–211.
<https://doi.org/10.2298/JSC170519088V>
- Vaferi, B., Rahn timer, Y., Darvishi, P., Toorani, A., & Lashkarbolooki, M. (2013). Phase equilibria modeling of binary systems containing ethanol using optimal feedforward neural network. *The Journal of Supercritical Fluids*, 84, 80–88.
<https://doi.org/10.1016/J.SUPFLU.2013.09.013>
- Valtz, A., Chapoy, A., Coquelet, C., Paricaud, P., & Richon, D. (2004). Vapour–liquid equilibria in the carbon dioxide–water system, measurement and modelling from 278.2 to 318.2 K. *Fluid Phase Equilibria*, 226(1–2), 333–344.
<https://doi.org/10.1016/J.FLUID.2004.10.013>
- VAN ANTWERPEN, F. J. (1980). *The Origins of Chemical Engineering*. 1–14.
<https://doi.org/10.1021/BA-1980-0190.CH001>
- Varbanov, P. S., Wang, Q., Zeng, M., Seferlis, P., Ma, T., Klemeš, J. J., Zhao, W., Xia, L., Cao, X., Bi, R., & Xiang, S. (2020). A Modified Alpha Function for Peng-Robinson Cubic Equation of State. *Chemical Engineering Transactions*, 81, 547–552.
<https://doi.org/10.3303/CET2081092>

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... Vázquez-Baeza, Y. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* 2020 17:3, 17(3), 261–272. <https://doi.org/10.1038/s41592-019-0686-2>

Vyhmeister, E., Rodríguez-Pino, J., Reyes-Bozo, L., Galleguillos-Pozo, R., Valdés-González, H., & Rodríguez-Maecker, R. N. (2017). Vapor-Liquid equilibria modeling using gray-box neural networks as binary interaction parameters predictor. *DYNA*, 84(203), 226–232. <https://doi.org/10.15446/dyna.v84n203.56364>

Wang, J.-Q., & Taherdoost, H. (2023). Deep Learning and Neural Networks: Decision-Making Implications. *Symmetry* 2023, Vol. 15, Page 1723, 15(9), 1723. <https://doi.org/10.3390/SYM15091723>

Wang, X., Liesaputra, V., Liu, Z., Wang, Y., & Huang, Z. (2024). An in-depth survey on Deep Learning-based Motor Imagery Electroencephalogram (EEG) classification. *Artificial Intelligence in Medicine*, 147, 102738. <https://doi.org/10.1016/J.ARTMED.2023.102738>

What is carbon sequestration? | U.S. Geological Survey. (s/f). Recuperado el 15 de octubre de 2024, de <https://www.usgs.gov/faqs/what-carbon-sequestration>

What Is Simulated Annealing? - MATLAB & Simulink. (s/f). Recuperado el 14 de septiembre de 2024, de <https://www.mathworks.com/help/gads/what-is-simulated-annealing.html>

Wisniak, J. (2012). *Kamerlingh-- The Virial Equation of State.*

Xiang, Y., Sun, D. Y., Fan, W., & Gong, X. G. (1997). Generalized simulated annealing algorithm and its application to the Thomson model. *Physics Letters A*, 233(3), 216–220. [https://doi.org/10.1016/S0375-9601\(97\)00474-X](https://doi.org/10.1016/S0375-9601(97)00474-X)

Anexos

	Pag
Anexo 1: Data aumentada para $T = 267^{\circ}\text{C}$	1
Anexo 2: Data aumentada para $T = 268^{\circ}\text{C}$	2
Anexo 3: Data aumentada para $T = 270^{\circ}\text{C}$	3
Anexo 4: Data aumentada para $T = 275^{\circ}\text{C}$	4
Anexo 5: Data aumentada para $T = 300^{\circ}\text{C}$	5
Anexo 6: Data aumentada para $T = 350^{\circ}\text{C}$	6
Anexo 7: Parámetros obtenidos para $T = 267^{\circ}\text{C}$	7
Anexo 8: Parámetros obtenidos para $T = 268^{\circ}\text{C}$	8
Anexo 9: Parámetros obtenidos para $T = 270^{\circ}\text{C}$	9
Anexo 10: Parámetros obtenidos para $T = 275^{\circ}\text{C}$	10
Anexo 11: Parámetros obtenidos para $T = 300^{\circ}\text{C}$	11
Anexo 12: Parámetros obtenidos para $T = 350^{\circ}\text{C}$	12
Anexo 13: Hiperparámetros para estimar s_{ij} con función de activación ReLU en capas ocultas	13
Anexo 14: Hiperparámetros para estimar Δa_m^l con función de activación ReLU en capas ocultas	16
Anexo 15: Hiperparámetros para estimar s_{ij} con función de activación Tangente hiperbólico en capas ocultas	19
Anexo 16: Hiperparámetros para estimar Δa_m^l con función de activación Tangente hiperbólico en capas ocultas	22

Anexo 1: Data aumentada para T = 267°C

T(°C)		267		
x CO ₂	y CO ₂	x H ₂ O	y H ₂ O	P
0.026	0.567	0.974	0.433	200.001
0.032	0.586	0.968	0.414	233.282
0.038	0.602	0.962	0.398	266.563
0.044	0.616	0.956	0.384	299.845
0.050	0.627	0.950	0.373	333.126
0.056	0.636	0.944	0.364	366.407
0.063	0.642	0.937	0.358	399.688
0.069	0.647	0.931	0.353	432.970
0.075	0.650	0.925	0.350	466.251
0.082	0.651	0.918	0.349	499.532
0.088	0.650	0.912	0.350	532.813
0.095	0.648	0.905	0.352	566.095
0.101	0.645	0.899	0.355	599.376
0.107	0.641	0.893	0.359	632.657
0.114	0.636	0.886	0.364	665.938
0.120	0.631	0.880	0.369	699.220
0.126	0.624	0.874	0.376	732.501
0.132	0.618	0.868	0.382	765.782
0.138	0.611	0.862	0.389	799.064
0.144	0.604	0.856	0.396	832.345
0.149	0.597	0.851	0.403	865.626
0.155	0.590	0.845	0.410	898.907
0.160	0.584	0.840	0.416	932.189
0.165	0.578	0.835	0.422	965.470
0.170	0.573	0.830	0.427	998.751
0.175	0.569	0.825	0.431	1032.032
0.179	0.566	0.821	0.434	1065.314
0.184	0.563	0.816	0.437	1098.595
0.188	0.561	0.812	0.439	1131.876
0.192	0.559	0.808	0.441	1165.157
0.196	0.557	0.804	0.443	1198.439
0.201	0.556	0.799	0.444	1231.720
0.205	0.554	0.795	0.446	1265.001
0.209	0.553	0.791	0.447	1298.282
0.214	0.551	0.786	0.449	1331.564
0.218	0.549	0.782	0.451	1364.845
0.223	0.547	0.777	0.453	1398.126
0.228	0.543	0.772	0.457	1431.407
0.234	0.540	0.766	0.460	1464.689
0.240	0.535	0.760	0.465	1497.970
0.246	0.529	0.754	0.471	1531.251
0.254	0.523	0.746	0.477	1564.532
0.262	0.515	0.738	0.485	1597.814
0.271	0.505	0.729	0.495	1631.095
0.281	0.494	0.719	0.506	1664.376
0.291	0.482	0.709	0.518	1697.657
0.302	0.468	0.698	0.532	1730.939
0.313	0.452	0.687	0.548	1764.220
0.327	0.434	0.673	0.566	1797.501
0.343	0.414	0.657	0.586	1830.782

Anexo 2: Data aumentada para T = 268°C

T(°C)		268		
x CO ₂	y CO ₂	x H ₂ O	y H ₂ O	P
0.030	0.565	0.970	0.435	200.017
0.034	0.581	0.966	0.419	228.049
0.038	0.595	0.962	0.405	256.082
0.042	0.607	0.958	0.393	284.115
0.046	0.617	0.954	0.383	312.148
0.051	0.625	0.949	0.375	340.180
0.055	0.632	0.945	0.368	368.213
0.060	0.638	0.940	0.362	396.246
0.065	0.642	0.935	0.358	424.278
0.071	0.644	0.929	0.356	452.311
0.076	0.646	0.924	0.354	480.344
0.081	0.646	0.919	0.354	508.377
0.087	0.645	0.913	0.355	536.409
0.092	0.644	0.908	0.356	564.442
0.098	0.641	0.902	0.359	592.475
0.104	0.638	0.896	0.362	620.508
0.110	0.634	0.890	0.366	648.540
0.115	0.630	0.885	0.370	676.573
0.121	0.625	0.879	0.375	704.606
0.127	0.619	0.873	0.381	732.639
0.133	0.614	0.867	0.386	760.671
0.138	0.608	0.862	0.392	788.704
0.144	0.602	0.856	0.398	816.737
0.149	0.596	0.851	0.404	844.769
0.155	0.590	0.845	0.410	872.802
0.160	0.584	0.840	0.416	900.835
0.166	0.578	0.834	0.422	928.868
0.171	0.573	0.829	0.427	956.900
0.176	0.568	0.824	0.432	984.933
0.181	0.564	0.819	0.436	1012.966
0.185	0.560	0.815	0.440	1040.999
0.190	0.557	0.810	0.443	1069.031
0.194	0.554	0.806	0.446	1097.064
0.198	0.551	0.802	0.449	1125.097
0.202	0.549	0.798	0.451	1153.130
0.206	0.546	0.794	0.454	1181.162
0.210	0.544	0.790	0.456	1209.195
0.215	0.541	0.785	0.459	1237.228
0.219	0.539	0.781	0.461	1265.260
0.224	0.536	0.776	0.464	1293.293
0.230	0.532	0.770	0.468	1321.326
0.237	0.528	0.763	0.472	1349.359
0.244	0.524	0.756	0.476	1377.391
0.252	0.519	0.748	0.481	1405.424
0.261	0.513	0.739	0.487	1433.457
0.271	0.506	0.729	0.494	1461.490
0.283	0.497	0.717	0.503	1489.522
0.295	0.485	0.705	0.515	1517.555
0.310	0.470	0.690	0.530	1545.588
0.333	0.447	0.667	0.553	1573.620

Anexo 3: Data aumentada para T = 270°C

T(°C)		270		
x CO ₂	y CO ₂	x H ₂ O	y H ₂ O	P
0.029	0.551	0.971	0.449	199.986
0.033	0.567	0.967	0.433	223.368
0.036	0.581	0.964	0.419	246.749
0.040	0.593	0.960	0.407	270.130
0.044	0.603	0.956	0.397	293.512
0.048	0.612	0.952	0.388	316.893
0.052	0.620	0.948	0.380	340.274
0.057	0.626	0.943	0.374	363.656
0.061	0.631	0.939	0.369	387.037
0.066	0.635	0.934	0.365	410.418
0.071	0.637	0.929	0.363	433.800
0.075	0.638	0.925	0.362	457.181
0.080	0.639	0.920	0.361	480.562
0.085	0.638	0.915	0.362	503.944
0.090	0.637	0.910	0.363	527.325
0.095	0.635	0.905	0.365	550.706
0.100	0.632	0.900	0.368	574.088
0.105	0.628	0.895	0.372	597.469
0.111	0.624	0.889	0.376	620.850
0.116	0.620	0.884	0.380	644.232
0.121	0.615	0.879	0.385	667.613
0.126	0.610	0.874	0.390	690.994
0.131	0.604	0.869	0.396	714.376
0.136	0.599	0.864	0.401	737.757
0.141	0.593	0.859	0.407	761.138
0.146	0.587	0.854	0.413	784.519
0.151	0.582	0.849	0.418	807.901
0.156	0.576	0.844	0.424	831.282
0.160	0.571	0.840	0.429	854.663
0.165	0.566	0.835	0.434	878.045
0.170	0.561	0.830	0.439	901.426
0.174	0.557	0.826	0.443	924.807
0.178	0.553	0.822	0.447	948.189
0.182	0.549	0.818	0.451	971.570
0.186	0.546	0.814	0.454	994.951
0.190	0.542	0.810	0.458	1018.333
0.193	0.539	0.807	0.461	1041.714
0.197	0.535	0.803	0.465	1065.095
0.201	0.531	0.799	0.469	1088.477
0.205	0.526	0.795	0.474	1111.858
0.210	0.521	0.790	0.479	1135.239
0.215	0.515	0.785	0.485	1158.621
0.221	0.509	0.779	0.491	1182.002
0.227	0.501	0.773	0.499	1205.383
0.235	0.493	0.765	0.507	1228.765
0.243	0.483	0.757	0.517	1252.146
0.253	0.473	0.747	0.527	1275.527
0.264	0.461	0.736	0.539	1298.909
0.276	0.445	0.724	0.555	1322.290
0.298	0.414	0.702	0.586	1345.671

Anexo 4: Data aumentada para T = 275°C

T(°C)		275		
x CO ₂	y CO ₂	x H ₂ O	y H ₂ O	P
0.025	0.519	0.975	0.481	199.989
0.029	0.536	0.971	0.464	217.720
0.034	0.551	0.966	0.449	235.451
0.038	0.564	0.962	0.436	253.182
0.042	0.575	0.958	0.425	270.913
0.046	0.584	0.954	0.416	288.644
0.050	0.592	0.950	0.408	306.375
0.054	0.599	0.946	0.401	324.106
0.058	0.604	0.942	0.396	341.837
0.062	0.608	0.938	0.392	359.568
0.066	0.611	0.934	0.389	377.299
0.069	0.613	0.931	0.387	395.030
0.073	0.615	0.927	0.385	412.761
0.077	0.615	0.923	0.385	430.492
0.081	0.615	0.919	0.385	448.223
0.085	0.614	0.915	0.386	465.954
0.088	0.613	0.912	0.387	483.685
0.092	0.611	0.908	0.389	501.416
0.096	0.609	0.904	0.391	519.147
0.100	0.606	0.900	0.394	536.878
0.104	0.603	0.896	0.397	554.609
0.108	0.600	0.892	0.400	572.340
0.111	0.596	0.889	0.404	590.071
0.115	0.593	0.885	0.407	607.802
0.119	0.589	0.881	0.411	625.533
0.124	0.585	0.876	0.415	643.263
0.128	0.581	0.872	0.419	660.994
0.132	0.576	0.868	0.424	678.725
0.136	0.572	0.864	0.428	696.456
0.141	0.567	0.859	0.433	714.187
0.145	0.562	0.855	0.438	731.918
0.150	0.558	0.850	0.442	749.649
0.154	0.553	0.846	0.447	767.380
0.159	0.547	0.841	0.453	785.111
0.164	0.542	0.836	0.458	802.842
0.169	0.536	0.831	0.464	820.573
0.174	0.530	0.826	0.470	838.304
0.180	0.524	0.820	0.476	856.035
0.185	0.517	0.815	0.483	873.766
0.191	0.510	0.809	0.490	891.497
0.198	0.503	0.802	0.497	909.228
0.205	0.494	0.795	0.506	926.959
0.213	0.486	0.787	0.514	944.690
0.221	0.476	0.779	0.524	962.421
0.231	0.466	0.769	0.534	980.152
0.242	0.454	0.758	0.546	997.883
0.254	0.442	0.746	0.558	1015.614
0.269	0.431	0.731	0.569	1033.345
0.287	0.417	0.713	0.583	1051.076
0.311	0.390	0.689	0.610	1068.807

Anexo 5: Data aumentada para T = 300°C

T(°C)		300		
x CO ₂	y CO ₂	x H ₂ O	y H ₂ O	P
0.023	0.352	0.977	0.648	200.145
0.025	0.364	0.975	0.636	208.178
0.027	0.375	0.973	0.625	216.210
0.029	0.385	0.971	0.615	224.243
0.031	0.395	0.969	0.605	232.276
0.033	0.404	0.967	0.596	240.309
0.035	0.413	0.965	0.587	248.342
0.037	0.421	0.963	0.579	256.374
0.039	0.428	0.961	0.572	264.407
0.041	0.435	0.959	0.565	272.440
0.043	0.441	0.957	0.559	280.473
0.045	0.447	0.955	0.553	288.506
0.048	0.452	0.952	0.548	296.539
0.050	0.457	0.950	0.543	304.571
0.052	0.461	0.948	0.539	312.604
0.054	0.464	0.946	0.536	320.637
0.057	0.467	0.943	0.533	328.670
0.059	0.470	0.941	0.530	336.703
0.061	0.472	0.939	0.528	344.735
0.064	0.474	0.936	0.526	352.768
0.066	0.476	0.934	0.524	360.801
0.069	0.477	0.931	0.523	368.834
0.072	0.478	0.928	0.522	376.867
0.074	0.479	0.926	0.521	384.899
0.077	0.480	0.923	0.520	392.932
0.080	0.480	0.920	0.520	400.965
0.083	0.480	0.917	0.520	408.998
0.086	0.480	0.914	0.520	417.031
0.089	0.480	0.911	0.520	425.064
0.093	0.479	0.907	0.521	433.096
0.096	0.478	0.904	0.522	441.129
0.099	0.477	0.901	0.523	449.162
0.103	0.476	0.897	0.524	457.195
0.107	0.474	0.893	0.526	465.228
0.111	0.472	0.889	0.528	473.260
0.115	0.469	0.885	0.531	481.293
0.119	0.466	0.881	0.534	489.326
0.123	0.461	0.877	0.539	497.359
0.127	0.457	0.873	0.543	505.392
0.132	0.450	0.868	0.550	513.424
0.138	0.443	0.862	0.557	521.457
0.144	0.434	0.856	0.566	529.490
0.152	0.424	0.848	0.576	537.523
0.161	0.412	0.839	0.588	545.556
0.171	0.400	0.829	0.600	553.589
0.182	0.387	0.818	0.613	561.621
0.194	0.374	0.806	0.626	569.654
0.207	0.360	0.793	0.640	577.687
0.223	0.344	0.777	0.656	585.720
0.245	0.323	0.755	0.677	593.753

Anexo 6: Data aumentada para T = 350°C

T(°C)		350		
x CO ₂	y CO ₂	x H ₂ O	y H ₂ O	P
0.008	0.054	0.992	0.946	200.013
0.009	0.058	0.991	0.942	202.639
0.010	0.062	0.990	0.938	205.265
0.011	0.066	0.989	0.934	207.891
0.012	0.070	0.988	0.930	210.518
0.013	0.074	0.987	0.926	213.144
0.014	0.078	0.986	0.922	215.770
0.015	0.081	0.985	0.919	218.396
0.016	0.085	0.984	0.915	221.022
0.017	0.088	0.983	0.912	223.649
0.018	0.092	0.982	0.908	226.275
0.019	0.095	0.981	0.905	228.901
0.019	0.099	0.981	0.901	231.527
0.020	0.102	0.980	0.898	234.153
0.021	0.105	0.979	0.895	236.780
0.022	0.108	0.978	0.892	239.406
0.023	0.111	0.977	0.889	242.032
0.024	0.114	0.976	0.886	244.658
0.025	0.117	0.975	0.883	247.284
0.026	0.120	0.974	0.880	249.911
0.027	0.123	0.973	0.877	252.537
0.028	0.126	0.972	0.874	255.163
0.029	0.128	0.971	0.872	257.789
0.030	0.131	0.970	0.869	260.415
0.031	0.134	0.969	0.866	263.042
0.032	0.136	0.968	0.864	265.668
0.033	0.139	0.967	0.861	268.294
0.034	0.142	0.966	0.858	270.920
0.035	0.144	0.965	0.856	273.547
0.036	0.147	0.964	0.853	276.173
0.038	0.149	0.962	0.851	278.799
0.039	0.151	0.961	0.849	281.425
0.040	0.154	0.960	0.846	284.051
0.042	0.156	0.958	0.844	286.678
0.043	0.158	0.957	0.842	289.304
0.045	0.161	0.955	0.839	291.930
0.047	0.163	0.953	0.837	294.556
0.049	0.165	0.951	0.835	297.182
0.051	0.168	0.949	0.832	299.809
0.053	0.170	0.947	0.830	302.435
0.055	0.172	0.945	0.828	305.061
0.057	0.174	0.943	0.826	307.687
0.060	0.175	0.940	0.825	310.313
0.063	0.175	0.937	0.825	312.940
0.066	0.175	0.934	0.825	315.566
0.068	0.173	0.932	0.827	318.192
0.072	0.170	0.928	0.830	320.818
0.075	0.164	0.925	0.836	323.444
0.079	0.156	0.921	0.844	326.071
0.087	0.145	0.913	0.855	328.697

Anexo 7: Parámetros obtenidos para T = 267°C

T(°C)				267			
x CO ₂	y CO ₂	x H ₂ O	y H ₂ O	P	sij	am_Calc	am_VW
0.026	0.567	0.974	0.433	200.001	0.1063305	6.7759138	6.7641895
0.032	0.586	0.968	0.414	233.282	0.1003662	6.7441476	6.7269379
0.038	0.602	0.962	0.398	266.563	0.100275	6.69622	6.6879073
0.044	0.616	0.956	0.384	299.845	0.102439	6.6420723	6.6476134
0.050	0.627	0.950	0.373	333.126	0.1053923	6.585757	6.6064001
0.056	0.636	0.944	0.364	366.407	0.1079	6.5311024	6.5647803
0.063	0.642	0.937	0.358	399.688	0.1100955	6.4775929	6.5228479
0.069	0.647	0.931	0.353	432.970	0.1113382	6.4274674	6.4810755
0.075	0.650	0.925	0.350	466.251	0.1117512	6.3803659	6.4396231
0.082	0.651	0.918	0.349	499.532	0.1113775	6.3362396	6.3986766
0.088	0.650	0.912	0.350	532.813	0.1103026	6.2949017	6.3583847
0.095	0.648	0.905	0.352	566.095	0.1087587	6.2556473	6.3187767
0.101	0.645	0.899	0.355	599.376	0.1064968	6.2194922	6.2801893
0.107	0.641	0.893	0.359	632.657	0.1040065	6.184809	6.2424253
0.114	0.636	0.886	0.364	665.938	0.1012795	6.1517474	6.2056057
0.120	0.631	0.880	0.369	699.220	0.098445	6.1199598	6.1697298
0.126	0.624	0.874	0.376	732.501	0.095536	6.08944	6.134856
0.132	0.618	0.868	0.382	765.782	0.0927078	6.0597352	6.1009189
0.138	0.611	0.862	0.389	799.064	0.0899832	6.0308645	6.0679517
0.144	0.604	0.856	0.396	832.345	0.0874481	6.0026121	6.0359162
0.149	0.597	0.851	0.403	865.626	0.0851411	5.9749265	6.0048058
0.155	0.590	0.845	0.410	898.907	0.0831004	5.9477499	5.9746046
0.160	0.584	0.840	0.416	932.189	0.0813597	5.9210513	5.9452925
0.165	0.578	0.835	0.422	965.470	0.0799328	5.8947938	5.9168634
0.170	0.573	0.830	0.427	998.751	0.0788513	5.8689778	5.8892864
0.175	0.569	0.825	0.431	1032.032	0.0781141	5.8435809	5.8625236
0.179	0.566	0.821	0.434	1065.314	0.0776581	5.8185588	5.836449
0.184	0.563	0.816	0.437	1098.595	0.0774146	5.7938496	5.8109223
0.188	0.561	0.812	0.439	1131.876	0.0773263	5.7693834	5.7858126
0.192	0.559	0.808	0.441	1165.157	0.0773348	5.7451154	5.7610048
0.196	0.557	0.804	0.443	1198.439	0.0774047	5.7209342	5.7363645
0.201	0.556	0.799	0.444	1231.720	0.0774902	5.6967701	5.7117726
0.205	0.554	0.795	0.446	1265.001	0.0775244	5.672625	5.6871347
0.209	0.553	0.791	0.447	1298.282	0.0775546	5.648096	5.6621979
0.214	0.551	0.786	0.449	1331.564	0.0774819	5.6232721	5.6368821
0.218	0.549	0.782	0.451	1364.845	0.0772803	5.5979616	5.6109856
0.223	0.547	0.777	0.453	1398.126	0.0769305	5.5719098	5.5842553
0.228	0.543	0.772	0.457	1431.407	0.0764081	5.5448192	5.5563828
0.234	0.540	0.766	0.460	1464.689	0.0756931	5.5162857	5.5269589
0.240	0.535	0.760	0.465	1497.970	0.0747721	5.4857391	5.4954164
0.246	0.529	0.754	0.471	1531.251	0.0736513	5.4520127	5.4606172
0.254	0.523	0.746	0.477	1564.532	0.0723097	5.4134141	5.4208252
0.262	0.515	0.738	0.485	1597.814	0.0707852	5.3703665	5.3765311
0.271	0.505	0.729	0.495	1631.095	0.0691069	5.3244327	5.3293745
0.281	0.494	0.719	0.506	1664.376	0.0672223	5.277037	5.2807436
0.291	0.482	0.709	0.518	1697.657	0.0651078	5.2281375	5.2306036
0.302	0.468	0.698	0.532	1730.939	0.0631648	5.1752627	5.1769722
0.313	0.452	0.687	0.548	1764.220	0.061279	5.1171122	5.1182669
0.327	0.434	0.673	0.566	1797.501	0.0590146	5.051781	5.0521129
0.343	0.414	0.657	0.586	1830.782	0.0576095	4.968325	4.9684453

Anexo 8: Parámetros obtenidos para T = 268°C

T(°C)		268					
x CO ₂	y CO ₂	x H ₂ O	y H ₂ O	P	sij	am	am_VW
0.030	0.565	0.970	0.435	200.017	0.0867336	6.7239063	6.7334117
0.034	0.581	0.966	0.419	228.049	0.0911248	6.7115083	6.7084525
0.038	0.595	0.962	0.405	256.082	0.0971532	6.6821602	6.6812976
0.042	0.607	0.958	0.393	284.115	0.1014756	6.6501863	6.6529525
0.046	0.617	0.954	0.383	312.148	0.1059225	6.6113779	6.6230177
0.051	0.625	0.949	0.375	340.180	0.1095213	6.5706802	6.5919458
0.055	0.632	0.945	0.368	368.213	0.1122209	6.5294702	6.5599367
0.060	0.638	0.940	0.362	396.246	0.1140454	6.4885764	6.5271689
0.065	0.642	0.935	0.358	424.278	0.1150556	6.4484853	6.4938043
0.071	0.644	0.929	0.356	452.311	0.1153292	6.4094656	6.4599903
0.076	0.646	0.924	0.354	480.344	0.1149575	6.3716198	6.4258572
0.081	0.646	0.919	0.354	508.377	0.1140087	6.3350458	6.3915363
0.087	0.645	0.913	0.355	536.409	0.1125891	6.2996399	6.3571243
0.092	0.644	0.908	0.356	564.442	0.1107751	6.2653554	6.3227205
0.098	0.641	0.902	0.359	592.475	0.1086519	6.232078	6.2884054
0.104	0.638	0.896	0.362	620.508	0.1062916	6.1997117	6.2542537
0.110	0.634	0.890	0.366	648.540	0.103766	6.1681406	6.2203278
0.115	0.630	0.885	0.370	676.573	0.1011416	6.1372513	6.1866807
0.121	0.625	0.879	0.375	704.606	0.0986214	6.1064305	6.1532376
0.127	0.619	0.873	0.381	732.639	0.0958194	6.0771644	6.1204148
0.133	0.614	0.867	0.386	760.671	0.0932095	6.047854	6.0878876
0.138	0.608	0.862	0.392	788.704	0.0907169	6.0188603	6.0557886
0.144	0.602	0.856	0.398	816.737	0.0883434	5.9902652	6.024183
0.149	0.596	0.851	0.404	844.769	0.0861284	5.9620144	5.9930965
0.155	0.590	0.845	0.410	872.802	0.0840986	5.9340966	5.9625618
0.160	0.584	0.840	0.416	900.835	0.0822621	5.9065652	5.9326282
0.166	0.578	0.834	0.422	928.868	0.0806414	5.8794207	5.9033271
0.171	0.573	0.829	0.427	956.900	0.0792623	5.8526531	5.8746843
0.176	0.568	0.824	0.432	984.933	0.0781204	5.8263576	5.8467578
0.181	0.564	0.819	0.436	1012.966	0.0772292	5.8005649	5.8195853
0.185	0.560	0.815	0.440	1040.999	0.0765694	5.7753841	5.7932439
0.190	0.557	0.810	0.443	1069.031	0.0760926	5.7509865	5.767849
0.194	0.554	0.806	0.446	1097.064	0.0757381	5.7275897	5.743537
0.198	0.551	0.802	0.449	1125.097	0.075514	5.7050858	5.7202745
0.202	0.549	0.798	0.451	1153.130	0.0751596	5.6837957	5.6978953
0.206	0.546	0.794	0.454	1181.162	0.0751508	5.6613686	5.6751306
0.210	0.544	0.790	0.456	1209.195	0.0746307	5.6400352	5.6524541
0.215	0.541	0.785	0.459	1237.228	0.0747195	5.6153054	5.6277462
0.219	0.539	0.781	0.461	1265.260	0.0744132	5.5899443	5.601716
0.224	0.536	0.776	0.464	1293.293	0.074025	5.5622565	5.5733327
0.230	0.532	0.770	0.468	1321.326	0.0735454	5.5316721	5.5420085
0.237	0.528	0.763	0.472	1349.359	0.0729762	5.4976094	5.5071553
0.244	0.524	0.756	0.476	1377.391	0.0722785	5.4596481	5.4682628
0.252	0.519	0.748	0.481	1405.424	0.071591	5.4168183	5.4245797
0.261	0.513	0.739	0.487	1433.457	0.0707608	5.3690794	5.3758104
0.271	0.506	0.729	0.494	1461.490	0.0698212	5.3157047	5.3214153
0.283	0.497	0.717	0.503	1489.522	0.0684898	5.2569709	5.261395
0.295	0.485	0.705	0.515	1517.555	0.0667564	5.1923208	5.1953825
0.310	0.470	0.690	0.530	1545.588	0.0652285	5.118965	5.1212862
0.333	0.447	0.667	0.553	1573.620	0.0626572	5.0019303	5.0026239

Anexo 9: Parámetros obtenidos para T = 270°C

T(°C)		270					
x CO ₂	y CO ₂	x H ₂ O	y H ₂ O	P	sij	am	am_VW
0.029	0.551	0.971	0.449	199.986	0.087614	6.7192429	6.7180537
0.033	0.567	0.967	0.433	223.368	0.0913925	6.7034714	6.6953023
0.036	0.581	0.964	0.419	246.749	0.0960442	6.6768026	6.670936
0.040	0.593	0.960	0.407	270.130	0.100825	6.6435302	6.6451755
0.044	0.603	0.956	0.397	293.512	0.1048727	6.6079703	6.6183631
0.048	0.612	0.952	0.388	316.893	0.1082897	6.5708493	6.5905913
0.052	0.620	0.948	0.380	340.274	0.1109999	6.5332609	6.5620162
0.057	0.626	0.943	0.374	363.656	0.1129499	6.4960415	6.5327975
0.061	0.631	0.939	0.369	387.037	0.1142891	6.4591891	6.5030118
0.066	0.635	0.934	0.365	410.418	0.1149296	6.4234144	6.4728357
0.071	0.637	0.929	0.363	433.800	0.1149481	6.3887803	6.4423726
0.075	0.638	0.925	0.362	457.181	0.1144882	6.3550554	6.4116794
0.080	0.639	0.920	0.361	480.562	0.1134803	6.3227012	6.38092
0.085	0.638	0.915	0.362	503.944	0.1120857	6.2913421	6.3501233
0.090	0.637	0.910	0.363	527.325	0.1102981	6.2611574	6.3194055
0.095	0.635	0.905	0.365	550.706	0.1083264	6.2315529	6.2887344
0.100	0.632	0.900	0.368	574.088	0.1058878	6.2036388	6.2584028
0.105	0.628	0.895	0.372	597.469	0.1036971	6.1750164	6.2279876
0.111	0.624	0.889	0.376	620.850	0.1012024	6.1477179	6.1979655
0.116	0.620	0.884	0.380	644.232	0.0986581	6.1209398	6.1682237
0.121	0.615	0.879	0.385	667.613	0.0960931	6.0946643	6.1388068
0.126	0.610	0.874	0.390	690.994	0.093607	6.0686191	6.1096915
0.131	0.604	0.869	0.396	714.376	0.0911597	6.0430201	6.0809683
0.136	0.599	0.864	0.401	737.757	0.0888319	6.017655	6.0526177
0.141	0.593	0.859	0.407	761.138	0.0866244	5.9925902	6.0246868
0.146	0.587	0.854	0.413	784.519	0.0845785	5.9677479	5.9971819
0.151	0.582	0.849	0.418	807.901	0.0827121	5.9431305	5.9701274
0.156	0.576	0.844	0.424	831.282	0.0810181	5.9188262	5.9435723
0.160	0.571	0.840	0.429	854.663	0.079533	5.8947713	5.9175192
0.165	0.566	0.835	0.434	878.045	0.0782547	5.8710341	5.8920091
0.170	0.561	0.830	0.439	901.426	0.0771986	5.8476225	5.8670641
0.174	0.557	0.826	0.443	924.807	0.0763527	5.8246216	5.8427369
0.178	0.553	0.822	0.447	948.189	0.075691	5.8021088	5.8190979
0.182	0.549	0.818	0.451	971.570	0.0751283	5.7803578	5.7962907
0.186	0.546	0.814	0.454	994.951	0.0746553	5.759388	5.7743748
0.190	0.542	0.810	0.458	1018.333	0.0741633	5.7394959	5.7534708
0.193	0.539	0.807	0.461	1041.714	0.0739944	5.7189564	5.7326702
0.197	0.535	0.803	0.465	1065.095	0.073111	5.7004828	5.7125637
0.201	0.531	0.799	0.469	1088.477	0.0724252	5.6802401	5.6913273
0.205	0.526	0.795	0.474	1111.858	0.0716428	5.6584862	5.6686457
0.210	0.521	0.790	0.479	1135.239	0.0709915	5.6337638	5.6435463
0.215	0.515	0.785	0.485	1158.621	0.0695323	5.6084851	5.616616
0.221	0.509	0.779	0.491	1182.002	0.0679515	5.5797184	5.5863204
0.227	0.501	0.773	0.499	1205.383	0.0668359	5.544971	5.5512126
0.235	0.493	0.765	0.507	1228.765	0.0651631	5.5069561	5.5120957
0.243	0.483	0.757	0.517	1252.146	0.0631431	5.46441	5.4681191
0.253	0.473	0.747	0.527	1275.527	0.0615804	5.4143503	5.4175391
0.264	0.461	0.736	0.539	1298.909	0.059732	5.3585858	5.3609072
0.276	0.445	0.724	0.555	1322.290	0.0575132	5.2964143	5.2978817
0.298	0.414	0.702	0.586	1345.671	0.0534187	5.1887509	5.1891798

Anexo 10: Parámetros obtenidos para T = 275°C

T(°C)				275			
x CO ₂	y CO ₂	x H ₂ O	y H ₂ O	P	sij	am	am_VW
0.025	0.519	0.975	0.481	199.989	0.1141996	6.7229095	6.6940495
0.029	0.536	0.971	0.464	217.720	0.1078865	6.6866641	6.6668271
0.034	0.551	0.966	0.449	235.451	0.1053295	6.6486423	6.6396235
0.038	0.564	0.962	0.436	253.182	0.1049331	6.6102744	6.6124964
0.042	0.575	0.958	0.425	270.913	0.1056801	6.572604	6.5855298
0.046	0.584	0.954	0.416	288.644	0.1069777	6.5361296	6.5587888
0.050	0.592	0.950	0.408	306.375	0.1084094	6.5012729	6.532345
0.054	0.599	0.946	0.401	324.106	0.1097361	6.4681846	6.5062469
0.058	0.604	0.942	0.396	341.837	0.1108165	6.4368956	6.4805272
0.062	0.608	0.938	0.392	359.568	0.1115748	6.4073521	6.4552035
0.066	0.611	0.934	0.389	377.299	0.1119788	6.3794488	6.4302794
0.069	0.613	0.931	0.387	395.030	0.1120294	6.3530311	6.4057441
0.073	0.615	0.927	0.385	412.761	0.1117091	6.3280487	6.3815958
0.077	0.615	0.923	0.385	430.492	0.1111487	6.3039881	6.3577504
0.081	0.615	0.919	0.385	448.223	0.1102841	6.281004	6.3342248
0.085	0.614	0.915	0.386	465.954	0.1091883	6.2588083	6.3109581
0.088	0.613	0.912	0.387	483.685	0.1079003	6.2372354	6.2879035
0.092	0.611	0.908	0.389	501.416	0.1064574	6.2161308	6.265012
0.096	0.609	0.904	0.391	519.147	0.1048959	6.1953474	6.2422315
0.100	0.606	0.900	0.394	536.878	0.1030967	6.1752628	6.2196169
0.104	0.603	0.896	0.397	554.609	0.1015253	6.154284	6.1968063
0.108	0.600	0.892	0.400	572.340	0.1001964	6.1323345	6.1737343
0.111	0.596	0.889	0.404	590.071	0.0979925	6.1131708	6.1512208
0.115	0.593	0.885	0.407	607.802	0.0961229	6.0926785	6.1283172
0.119	0.589	0.881	0.411	625.533	0.0944296	6.0713651	6.1050942
0.124	0.585	0.876	0.415	643.263	0.0926637	6.0500449	6.0817162
0.128	0.581	0.872	0.419	660.994	0.0909165	6.0283811	6.0580725
0.132	0.576	0.868	0.424	678.725	0.0891906	6.0063323	6.0341245
0.136	0.572	0.864	0.428	696.456	0.087487	5.9838637	6.009835
0.141	0.567	0.859	0.433	714.187	0.0858045	5.9609452	5.9851689
0.145	0.562	0.855	0.438	731.918	0.0841405	5.9375499	5.9600933
0.150	0.558	0.850	0.442	749.649	0.0824904	5.9136576	5.9345778
0.154	0.553	0.846	0.447	767.380	0.0808557	5.8892252	5.9085865
0.159	0.547	0.841	0.453	785.111	0.0792155	5.8642833	5.8821073
0.164	0.542	0.836	0.458	802.842	0.0775812	5.8387542	5.8550956
0.169	0.536	0.831	0.464	820.573	0.0759368	5.8126399	5.8275272
0.174	0.530	0.826	0.470	838.304	0.0742752	5.7858326	5.7992973
0.180	0.524	0.820	0.476	856.035	0.0725852	5.7581149	5.7701913
0.185	0.517	0.815	0.483	873.766	0.0702621	5.7310623	5.7405886
0.191	0.510	0.809	0.490	891.497	0.069074	5.6985022	5.7079261
0.198	0.503	0.802	0.497	909.228	0.0672348	5.6656054	5.6737767
0.205	0.494	0.795	0.506	926.959	0.0653345	5.6298002	5.6367748
0.213	0.486	0.787	0.514	944.690	0.0634782	5.5900106	5.596024
0.221	0.476	0.779	0.524	962.421	0.061407	5.546207	5.5510194
0.231	0.466	0.769	0.534	980.152	0.0593719	5.4967301	5.500503
0.242	0.454	0.758	0.546	997.883	0.0573915	5.4405939	5.4434453
0.254	0.442	0.746	0.558	1015.614	0.0555863	5.3765586	5.3785991
0.269	0.431	0.731	0.569	1033.345	0.0544726	5.3011112	5.3025463
0.287	0.417	0.713	0.583	1051.076	0.0533126	5.2078334	5.2086839
0.311	0.390	0.689	0.610	1068.807	0.0516661	5.0811649	5.0817102

Anexo 11: Parámetros obtenidos para T = 300°C

T(°C)		300					
x CO ₂	y CO ₂	x H ₂ O	y H ₂ O	P	sij	am	am_VW
0.023	0.352	0.977	0.648	200.145	0.1183403	6.5502619	6.4786412
0.025	0.364	0.975	0.636	208.178	0.1180685	6.5329324	6.4658899
0.027	0.375	0.973	0.625	216.210	0.1181006	6.516197	6.4531017
0.029	0.385	0.971	0.615	224.243	0.1195788	6.4965147	6.4399962
0.031	0.395	0.969	0.605	232.276	0.1207112	6.4782738	6.4268964
0.033	0.404	0.967	0.596	240.309	0.122301	6.4590699	6.4136006
0.035	0.413	0.965	0.587	248.342	0.1237307	6.4403505	6.4002256
0.037	0.421	0.963	0.579	256.374	0.1251287	6.4216282	6.3867223
0.039	0.428	0.961	0.572	264.407	0.1261287	6.4037765	6.3731751
0.041	0.435	0.959	0.565	272.440	0.1274587	6.3847645	6.3593639
0.043	0.441	0.957	0.559	280.473	0.1283343	6.3666379	6.3454991
0.045	0.447	0.955	0.553	288.506	0.1289883	6.3487326	6.3315046
0.048	0.452	0.952	0.548	296.539	0.1294151	6.3310345	6.3173723
0.050	0.457	0.950	0.543	304.571	0.1291714	6.3147236	6.303251
0.052	0.461	0.948	0.539	312.604	0.129581	6.2962104	6.2886591
0.054	0.464	0.946	0.536	320.637	0.129329	6.2790435	6.2740567
0.057	0.467	0.943	0.533	328.670	0.1287595	6.2622925	6.2593163
0.059	0.470	0.941	0.530	336.703	0.1282362	6.244972	6.2442813
0.061	0.472	0.939	0.528	344.735	0.1274263	6.2279885	6.2290746
0.064	0.474	0.936	0.526	352.768	0.1264659	6.2109776	6.2136291
0.066	0.476	0.934	0.524	360.801	0.1253753	6.1938896	6.197923
0.069	0.477	0.931	0.523	368.834	0.1241759	6.1766713	6.1819322
0.072	0.478	0.928	0.522	376.867	0.1228869	6.1592752	6.1656321
0.074	0.479	0.926	0.521	384.899	0.1215289	6.1416494	6.1489958
0.077	0.480	0.923	0.520	392.932	0.1201207	6.1237451	6.1319952
0.080	0.480	0.920	0.520	400.965	0.118681	6.1055124	6.1146003
0.083	0.480	0.917	0.520	408.998	0.1172119	6.0869374	6.0967887
0.086	0.480	0.914	0.520	417.031	0.1157227	6.0679637	6.0785326
0.089	0.480	0.911	0.520	425.064	0.1142333	6.048501	6.0597959
0.093	0.479	0.907	0.521	433.096	0.1125663	6.0289987	6.0406648
0.096	0.478	0.904	0.522	441.129	0.1108727	6.0090041	6.0210239
0.099	0.477	0.901	0.523	449.162	0.1090889	5.98865	6.0008892
0.103	0.476	0.897	0.524	457.195	0.1066421	5.9694593	5.9806355
0.107	0.474	0.893	0.526	465.228	0.1051582	5.9469312	5.9591045
0.111	0.472	0.889	0.528	473.260	0.1029618	5.925596	5.9374412
0.115	0.469	0.885	0.531	481.293	0.1006032	5.9038838	5.915237
0.119	0.466	0.881	0.534	489.326	0.0980253	5.8818761	5.8925097
0.123	0.461	0.877	0.539	497.359	0.095215	5.8595259	5.869245
0.127	0.457	0.873	0.543	505.392	0.0928394	5.8347248	5.8446983
0.132	0.450	0.868	0.550	513.424	0.088356	5.8126395	5.8197456
0.138	0.443	0.862	0.557	521.457	0.0839193	5.785549	5.7910429
0.144	0.434	0.856	0.566	529.490	0.0790949	5.7534285	5.7577618
0.152	0.424	0.848	0.576	537.523	0.073685	5.7163337	5.7193665
0.161	0.412	0.839	0.588	545.556	0.0682396	5.672935	5.6750569
0.171	0.400	0.829	0.600	553.589	0.0629142	5.6233413	5.6247343
0.182	0.387	0.818	0.613	561.621	0.0580341	5.5674244	5.5683417
0.194	0.374	0.806	0.626	569.654	0.053774	5.5054936	5.5060836
0.207	0.360	0.793	0.640	577.687	0.0500668	5.4375058	5.4377016
0.223	0.344	0.777	0.656	585.720	0.0471809	5.3562085	5.3564078
0.245	0.323	0.755	0.677	593.753	0.0444593	5.2449463	5.2450156

Anexo 12: Parámetros obtenidos para T = 350°C

T(°C)				350			
x CO ₂	y CO ₂	x H ₂ O	y H ₂ O	P	sij	am	am_VW
0.008	0.054	0.992	0.946	200.013	0.2676327	6.174463	6.1360536
0.009	0.058	0.991	0.942	202.639	0.2542511	6.1626825	6.1290651
0.010	0.062	0.990	0.938	205.265	0.2415385	6.156013	6.122451
0.011	0.066	0.989	0.934	207.891	0.2294616	6.1529537	6.1161698
0.012	0.070	0.988	0.930	210.518	0.2228117	6.147586	6.1097862
0.013	0.074	0.987	0.926	213.144	0.2199178	6.1406091	6.1033169
0.014	0.078	0.986	0.922	215.770	0.2186183	6.1335799	6.0968736
0.015	0.081	0.985	0.919	218.396	0.2184483	6.1265254	6.0904538
0.016	0.085	0.984	0.915	221.022	0.2190898	6.1194396	6.084052
0.017	0.088	0.983	0.912	223.649	0.2202754	6.1123492	6.0776658
0.018	0.092	0.982	0.908	226.275	0.2218227	6.1052527	6.0712895
0.019	0.095	0.981	0.905	228.901	0.2235895	6.0981507	6.0649174
0.019	0.099	0.981	0.901	231.527	0.2254666	6.0910418	6.0585425
0.020	0.102	0.980	0.898	234.153	0.2273771	6.0839143	6.0521566
0.021	0.105	0.979	0.895	236.780	0.2292507	6.0767686	6.0457518
0.022	0.108	0.978	0.892	239.406	0.2310422	6.069592	6.0393185
0.023	0.111	0.977	0.889	242.032	0.2324302	6.0626958	6.0328905
0.024	0.114	0.976	0.886	244.658	0.2342371	6.055106	6.0263259
0.025	0.117	0.975	0.883	247.284	0.2355274	6.0478454	6.0197553
0.026	0.120	0.974	0.880	249.911	0.2358082	6.0414568	6.0132555
0.027	0.123	0.973	0.877	252.537	0.2377357	6.0328559	6.0063529
0.028	0.126	0.972	0.874	255.163	0.2385074	6.025245	5.9995172
0.029	0.128	0.971	0.872	257.789	0.2390719	6.0175125	5.9925705
0.030	0.131	0.970	0.869	260.415	0.239427	6.0096434	5.985499
0.031	0.134	0.969	0.866	263.042	0.239572	6.0016229	5.9782884
0.032	0.136	0.968	0.864	265.668	0.2395139	5.9934292	5.9709227
0.033	0.139	0.967	0.861	268.294	0.2392503	5.9850523	5.9633875
0.034	0.142	0.966	0.858	270.920	0.2384468	5.976879	5.9557431
0.035	0.144	0.965	0.856	273.547	0.2381733	5.9675928	5.9477185
0.036	0.147	0.964	0.853	276.173	0.236997	5.9588099	5.9395904
0.038	0.149	0.962	0.851	278.799	0.235882	5.9493002	5.9310831
0.039	0.151	0.961	0.849	281.425	0.2345262	5.939356	5.9222169
0.040	0.154	0.960	0.846	284.051	0.2327525	5.9291356	5.9129907
0.042	0.156	0.958	0.844	286.678	0.2306873	5.9184461	5.9033292
0.043	0.158	0.957	0.842	289.304	0.2283443	5.907242	5.8931875
0.045	0.161	0.955	0.839	291.930	0.225739	5.8954878	5.8825234
0.047	0.163	0.953	0.837	294.556	0.223107	5.882894	5.8712305
0.049	0.165	0.951	0.835	297.182	0.2202926	5.8696536	5.8593202
0.051	0.168	0.949	0.832	299.809	0.216362	5.8569048	5.8470734
0.053	0.170	0.947	0.830	302.435	0.213366	5.8421866	5.8337963
0.055	0.172	0.945	0.828	305.061	0.2106165	5.8262591	5.8196716
0.057	0.174	0.943	0.826	307.687	0.2057754	5.8118226	5.8055377
0.060	0.175	0.940	0.825	310.313	0.2011401	5.796038	5.7905794
0.063	0.175	0.937	0.825	312.940	0.1958309	5.7798873	5.7751319
0.066	0.175	0.934	0.825	315.566	0.1895822	5.7636467	5.7593344
0.068	0.173	0.932	0.827	318.192	0.1828542	5.7467966	5.743033
0.072	0.170	0.928	0.830	320.818	0.1746928	5.7306164	5.7266953
0.075	0.164	0.925	0.836	323.444	0.1670126	5.7132465	5.7094994
0.079	0.156	0.921	0.844	326.071	0.1611275	5.6913909	5.6887642
0.087	0.145	0.913	0.855	328.697	0.1530711	5.6370088	5.6386978

Anexo 13: Hiperparámetros para estimar s_{ij} con función de activación ReLU en capas ocultas

```
#-----#
## Weights and biases for NN using ReLU - author: Luis Gonzalez
#-----#
Layer: fc1.weight
Weights: tensor([[ -0.2912,  0.6121],
                 [ -0.8111, -0.2416],
                 [ -0.5174,  0.3688],
                 [ -0.2583,  0.8486],
                 [ -0.0628,  0.1871],
                 [ -0.2137, -0.1390],
                 [ -0.7840, -0.3141],
                 [ -0.2897, -0.0457],
                 [  0.4400,  0.2366],
                 [ -0.6720, -0.0606],
                 [  0.0320,  0.8422],
                 [ -0.3163,  0.7480],
                 [  0.0279, -0.0113],
                 [  0.8176, -0.4554],
                 [ -0.3471, -0.1797],
                 [ -0.1013,  0.5940],
                 [ -0.3315, -0.3645],
                 [ -0.3197, -0.6308],
                 [ -0.2290,  0.5503],
                 [  0.0328,  0.5402]])
Layer: fc1.bias
Biases: tensor([ 0.0706, -0.5915,  0.2004, -0.2518, -0.5109, -0.3645,
                0.5530,  0.2373,
                -0.5097,  0.2495,  0.6416,  1.0116,  0.2748,  0.2675,  0.6171,
                -0.4613,
                0.3854, -0.6049, -0.6092, -0.3487])
Layer: fc2.weight
Weights: tensor([[ 9.7425e-02,  4.8356e-01, -1.3768e-01, -8.0806e-02,
                  1.2275e-01,
                  -2.8223e-02,  1.1364e-01,  8.0900e-02,  8.0420e-02,  2.7697e-
01,
                  4.5810e-02,  3.4825e-02,  7.1263e-02,  2.3494e-01,  2.0323e-
01,
                  1.0687e-01,  6.3582e-02, -9.3535e-02, -6.4669e-02,  1.2250e-
02],
                 [-1.3226e-01,  7.1643e-01,  1.4246e-01, -3.5894e-01,  4.1853e-
02,
                 -3.7669e-02,  1.2062e-01,  7.9084e-02,  1.8900e-01, -1.7583e-
01,
                 4.5356e-01,  3.7521e-01,  3.5208e-01,  1.3686e-01, -4.4526e-
02,
                 -2.7709e-01, -5.7646e-02,  1.8453e-01, -2.9944e-01, -5.9065e-
02],
                 [ 1.8221e-01,  5.4633e-01, -1.5758e-01, -2.8439e-01, -2.1301e-
01,
                 -4.0191e-03, -7.9553e-03, -4.0671e-02, -2.5315e-02, -1.6871e-
02,
```

```

01,      2.8075e-01,  3.4287e-01,  1.8557e-02,  2.7106e-01,  3.3014e-
02],
[ 3.2895e-01, -4.2446e-01,  1.3242e-01,  3.8800e-01, -2.1449e-
01,
-1.3253e-01,  1.7352e-01,  1.1641e-01,  4.6655e-02,  2.8659e-
02,
1.2484e-01,  2.4531e-01,  6.1761e-02, -4.9094e-02,  3.9399e-
01,
2.2852e-01,  1.7070e-01, -1.0971e-01,  2.9519e-01,  4.0580e-
01],
[ 1.6740e-01,  6.1752e-01,  1.0863e-01, -1.4411e-01, -2.0087e-
01,
-1.0601e-01,  2.5763e-01,  1.5818e-01,  4.5686e-02, -3.0420e-
01,
6.7887e-02,  4.0549e-02,  1.2074e-01,  1.7882e-01,  2.6324e-
01,
-4.0987e-01, -2.8389e-02, -5.2056e-02, -2.4290e-01, -4.0598e-
02],
[-4.3667e-02, -4.3904e-02, -2.4331e-01, -2.4772e-01, -3.4990e-
02,
2.8917e-03, -1.5754e-01,  4.8852e-02, -2.3787e-01,  6.3106e-
02,
3.3700e-01, -3.7659e-02,  1.8443e-01,  1.5557e-01, -2.3848e-
01,
8.2471e-02, -2.7592e-01, -2.6484e-01,  1.1958e-01,  1.2620e-
01],
[ 2.5778e-01, -5.0032e-01,  3.9300e-01,  3.2851e-01,  1.0744e-
01,
7.8969e-02,  1.7449e-01,  1.6222e-01, -8.5262e-02,  3.3568e-
01,
3.6642e-01,  3.5537e-01, -5.4051e-03, -2.2564e-01,  2.7326e-
02,
1.0861e-01,  9.4003e-02, -1.0290e-02,  9.7492e-02,  2.5790e-
01],
[-7.5050e-02,  1.4414e-01, -2.5349e-01, -2.3804e-01, -1.2984e-
01,
1.5651e-01, -1.6281e-01,  1.0549e-01, -1.1563e-02,  8.8354e-
02,
-8.7688e-02, -1.2791e-01, -6.3583e-02, -3.7553e-02, -8.6771e-
02,
-5.5586e-02, -1.1429e-01, -1.1720e-01,  9.6430e-02, -2.6502e-
01],
[-7.8177e-03,  2.8134e-01,  2.5101e-03, -1.2255e-01, -1.9378e-
01,
1.1077e-01, -4.4752e-01, -1.4188e-01, -1.5690e-01, -3.9473e-
01,
2.1021e-01,  2.7559e-01, -2.1416e-02,  2.2416e-01, -2.6887e-
03,
-2.4355e-01, -4.2641e-01, -4.8237e-01, -1.9266e-01,  1.4451e-
01],
[ 7.3833e-02,  1.0770e-01, -1.3203e-01,  3.5797e-04,  1.1512e-
01,
-2.1625e-01, -3.8921e-01, -4.5868e-01,  1.3723e-02, -2.9141e-
01,

```

```
        -1.1424e+00,  2.4514e-01,  2.2658e-01,  2.4234e-01,  5.8064e-  
02,  
        -8.2698e-02, -4.2376e-01,  4.0884e-02, -1.0739e-01, -1.4922e-  
01]])  
Layer: fc2.bias  
Biases: tensor([ 0.0166,  0.3893,  0.3570,  0.3629,  0.2918,  0.0687,  
0.0665, -0.0346,  
0.3376,  0.0642])  
Layer: fc3.weight  
Weights: tensor([[ -0.1193, -0.4351, -0.4928,  0.2876, -0.2935, -0.3424,  
0.4916, -0.1356,  
-0.1824,  0.5836]])  
Layer: fc3.bias  
Biases: tensor([0.2240])
```

Anexo 14: Hiperparámetros para estimar Δa_m^l con función de activación ReLU en capas ocultas

```
#-----#
## Weights and biases for NN using ReLU - author: Luis Gonzalez
#-----#
Layer: fc1.weight
Weights: tensor([[ -0.4513,  0.2065],
                 [ -0.3020, -0.7874],
                 [ -0.6743,  0.2757],
                 [ -0.1033,  0.5540],
                 [ -0.0628,  0.1871],
                 [ -0.2137, -0.1390],
                 [ -0.9810, -0.4855],
                 [ -0.1901, -0.2010],
                 [  0.5295,  0.2251],
                 [ -0.9643, -0.4838],
                 [  0.5206,  0.6639],
                 [ -0.5564,  0.5822],
                 [ -0.3194, -0.0813],
                 [  0.2427, -0.8200],
                 [ -0.2906, -0.3733],
                 [ -0.6274,  0.5615],
                 [ -0.0866, -0.5184],
                 [ -1.1547, -0.3072],
                 [ -0.6504,  0.3907],
                 [  0.5967,  0.2666]])
Layer: fc1.bias
Biases: tensor([ 0.0145, -0.1721, -0.4158, -0.6668, -0.5109, -0.3645,
                 0.3456,  0.3403,
                 -0.3700,  0.3033,  0.6663,  0.5186,  0.0141,  0.5582,  0.7432,
                 -0.3770,
                 0.4041, -0.9092, -0.6867,  0.2696])
Layer: fc2.weight
Weights: tensor([[ 2.0933e-02, -1.1703e-01, -5.1709e-01,  7.4982e-03,
                  1.2275e-01,
                  -2.8223e-02,  1.8143e-01,  1.3518e-01, -1.0101e-01,  2.7268e-
01,
                  -5.4617e-01, -1.0818e-01,  1.3422e-01, -1.4074e-01,  3.5251e-
01,
                  -7.7768e-02,  1.7859e-01, -8.5956e-01, -2.3943e-01, -5.2601e-
01],
                 [-1.3780e-01,  2.6676e-02, -1.5705e-01, -2.2305e-01,  4.1853e-
02,
                  -3.7669e-02,  1.7276e-01,  1.6608e-01, -2.0072e-01, -1.1624e-
01,
                  -2.2112e-01,  1.5586e-01,  4.1224e-01, -1.9271e-01,  5.3448e-
02,
                  -6.4950e-01,  3.0117e-01, -6.7429e-01, -8.2688e-01, -5.2637e-
01],
                 [ 1.6691e-01,  8.4639e-02, -4.5917e-01, -1.4103e-01, -2.1301e-
01,
                  -4.0191e-03, -7.9590e-03,  1.0319e-02,  4.7535e-02,  1.3493e-
02,
```

```

01,      -4.1516e-01,  1.7573e-01,  4.6045e-02,  1.4840e-01,  3.6144e-
01],
[ 1.3287e-01, -4.4731e-01,  6.6287e-03,  7.1226e-02, -2.1449e-
01,
-1.3253e-01, -1.3440e-01,  1.0635e-03, -1.9893e-02, -2.4646e-
01,
-2.2150e-04,  7.7882e-02,  1.0183e-01,  1.3708e-01,  2.2288e-
02],
[ 1.4795e-01, -2.4929e-03, -1.6425e-01, -1.6172e-02, -2.0087e-
01,
-1.0601e-01,  3.9948e-01,  2.0667e-01, -4.4213e-01,  3.8944e-
02,
-6.3341e-01, -2.3162e-01,  1.5304e-01, -1.6184e-01,  3.4742e-
01,
-6.1487e-01,  3.3259e-01, -8.0003e-01, -4.8423e-01, -4.7125e-
01],
[-1.0373e-01, -4.3904e-02, -2.6075e-01, -2.5313e-01, -3.4990e-
02,
2.8917e-03, -1.6190e-01,  2.2723e-02, -2.7075e-01, -1.5091e-
02,
1.1723e-01, -1.5497e-01,  1.3517e-01, -5.5436e-02, -2.6343e-
01,
8.1420e-02, -8.5855e-02, -9.9820e-02,  1.1875e-01, -2.8025e-
01],
[ 6.6913e-02,  1.8227e-01,  3.4716e-01,  5.9951e-03,  1.0744e-
01,
7.8969e-02,  4.1253e-01,  1.7042e-01, -1.2421e-01,  8.1499e-
01,
2.3222e-01,  2.1681e-01,  7.4833e-02,  1.5164e-01,  1.8380e-
01,
-4.3971e-02,  5.1030e-02,  7.1810e-01, -4.8329e-02, -4.3765e-
02],
[-7.4805e-02,  1.4414e-01, -2.5337e-01, -2.3793e-01, -1.2984e-
01,
1.5651e-01, -1.6418e-01,  1.0452e-01, -2.1648e-03,  8.8366e-
02,
-8.5726e-02, -1.2847e-01, -5.5830e-02, -1.2973e-01, -8.6131e-
02,
-5.5412e-02, -1.1544e-01, -1.1720e-01,  9.6601e-02, -2.6517e-
01],
[ 1.0623e-01, -4.0404e-02, -1.3622e-01, -1.0038e-01, -1.9378e-
01,
1.1077e-01,  3.2483e-02, -6.2996e-02,  3.7269e-02, -7.7084e-
02,
-3.9330e-01,  2.1000e-01,  1.9808e-02,  7.3777e-02,  2.8330e-
01,
-4.0925e-01,  3.5734e-01, -9.5516e-01, -4.2346e-01, -2.7900e-
01],
[ 1.8979e-01,  3.3177e-01,  4.7827e-01, -3.1724e-01,  1.1512e-
01,
-2.1625e-01,  6.3224e-01, -5.4443e-02,  3.0819e-03,  5.6023e-
01,

```

```
          -5.0768e-01,  7.0303e-02,  3.6514e-01, -6.0733e-01, -1.7076e-
01,
          2.6974e-01, -2.9945e-01,  1.0053e+00,  1.6220e-01,  1.3694e-
01]])
Layer: fc2.bias
Biases: tensor([-0.0363,  0.4397,  0.3586,  0.2442,  0.2212, -0.1722,
 0.0529, -0.0183,
 0.2397, -0.3822])
Layer: fc3.weight
Weights: tensor([[ -0.5713, -0.6070, -0.7462,  0.0778, -0.5585, -0.1957,
 0.4455, -0.1388,
 -0.4071,  0.8175]])
Layer: fc3.bias
Biases: tensor([0.1135])
```

Anexo 15: Hiperparámetros para estimar s_{ij} con función de activación Tangente hiperbólico en capas ocultas

```
#-----#
## Weights and biases for NN using tanh - author: Luis Gonzalez
#-----#
Layer: fc1.weight
Weights: tensor([[ 0.1960,  0.1186],
                 [-0.6596, -0.4652],
                 [-0.3931,  0.0691],
                 [-0.1545,  0.6343],
                 [-0.2137,  0.3310],
                 [-0.3811, -0.0283],
                 [-0.9325, -0.3713],
                 [-0.2276, -0.0546],
                 [ 0.1060,  0.5565],
                 [-0.7528, -0.1501],
                 [ 0.4198,  0.4077],
                 [-0.0795,  0.3111],
                 [ 0.0950, -0.0773],
                 [ 0.6378, -0.4112],
                 [-0.4511, -0.3144],
                 [-0.1700,  0.5401],
                 [-0.2701, -0.6402],
                 [-0.7281, -0.4919],
                 [-0.1832,  0.5443],
                 [ 0.1910,  0.4482]])

Layer: fc1.bias
Biases: tensor([ 0.0734, -0.3883,  0.0458, -0.9359, -0.6560, -0.5662,
                0.2804,  0.1136,
                -0.6001, -0.1094,  0.3398,  0.7651, -0.0439,  0.5067,  0.4171,
                -0.7995,
                0.2825, -0.7834, -0.8719, -0.4804])

Layer: fc2.weight
Weights: tensor([[ 8.9911e-03,  2.9571e-01, -1.7475e-01,  1.2291e-02,
                  1.3075e-01,
                  2.5820e-01, -1.4832e-01, -6.8943e-02,  1.7682e-01,  2.0913e-
01,
                  -2.4606e-01, -1.1456e-01, -7.3460e-02,  2.1688e-01,  4.2331e-
02,
                  1.3946e-01, -2.4135e-01,  2.5856e-02, -4.7245e-02, -4.9193e-
02],
                 [-3.1286e-01,  1.8335e-01,  9.2387e-02, -5.0258e-01, -2.6809e-
01,
                  -1.9079e-01, -9.2043e-02, -1.2145e-01, -2.5522e-01, -2.2044e-
01,
                  1.0241e-01,  1.7126e-01,  1.2585e-01,  1.7677e-01, -2.0376e-
01,
                  -3.5076e-01,  1.9157e-01,  4.3153e-02, -3.9716e-01, -4.5306e-
01],
                 [ 1.1188e-01,  1.8790e-01, -1.5897e-01, -2.6001e-01, -4.2021e-
01,
                  -9.8285e-02, -2.8321e-01, -2.2765e-01, -2.2939e-01, -3.5761e-
02,
```

```

01,      -9.5272e-02,  1.6927e-01, -2.1704e-01,  2.6312e-01,  1.0980e-
01],
[ 2.0319e-01, -5.5354e-01, -1.6078e-01,  1.6061e-01, -2.4179e-
01,      -6.0858e-01,  4.0352e-02, -6.4371e-02, -8.5486e-02, -3.4703e-
01,      7.0887e-02, -2.4512e-02,  7.1905e-02,  5.4893e-02,  2.3325e-
01,      1.1681e-02, -2.9713e-04, -5.1767e-01,  9.3760e-02,  1.4000e-
01],
[ 1.0379e-01,  1.0235e-01,  1.1699e-01, -3.0096e-01, -5.4342e-
01,      -2.9583e-01,  1.0515e-01, -3.5272e-02, -4.6731e-01, -2.9937e-
01,      -1.9532e-01, -1.4468e-01, -8.6038e-02,  2.4636e-01,  1.2769e-
01,      -4.3304e-01,  2.4312e-01, -2.0823e-01, -2.7816e-01, -4.8803e-
01],
[-9.2805e-02, -2.7986e-01, -3.2983e-01, -3.3318e-01, -1.8446e-
01,      -1.5547e-01, -5.0539e-01, -8.6857e-02, -2.1766e-01, -4.5939e-
01,      2.4168e-01, -1.1314e-01,  9.4827e-02,  1.9817e-01, -4.2361e-
01,      1.3794e-02, -3.6594e-01, -1.4086e-01,  3.7398e-02, -1.1704e-
01],
[ 1.0569e-01, -9.4809e-02,  2.0359e-01,  1.4249e-01,  1.3590e-
01,      -8.9165e-02,  1.1272e-01,  6.7013e-02, -2.4748e-01,  2.0958e-
01,      1.8726e-01,  1.4147e-01, -2.4128e-03, -3.9609e-02,  2.0830e-
03,      -4.6993e-02,  3.4544e-01, -5.4752e-02, -4.3574e-02, -1.4705e-
01],
[-6.8332e-02,  1.7146e-01, -2.0278e-01, -2.4444e-01, -1.6983e-
01,      2.9710e-01, -2.6019e-01,  8.5942e-02,  1.6009e-01, -7.3186e-
02,      4.7053e-03, -5.4158e-03, -1.2514e-01, -3.8050e-03, -1.5966e-
01,      -5.2502e-02, -6.8297e-01,  1.5883e-02,  8.3371e-02, -3.0153e-
02],
[-1.3868e-01,  2.5977e-01, -1.1224e-02, -3.3259e-01, -5.6631e-
01,      1.8377e-03, -2.0355e-01, -7.2429e-02, -4.8049e-01, -1.2138e-
01,      5.1714e-03,  2.4133e-01,  8.3809e-03,  3.1462e-01,  1.8619e-
01,      -3.9709e-01,  2.8794e-01, -1.3718e-01, -3.7454e-01, -3.2235e-
01],
[ 1.6952e-01,  3.1358e-02, -2.6226e-02,  8.2995e-02,  1.4137e-
01,      -4.3745e-01,  3.0156e-01, -7.8619e-02, -5.6426e-02,  2.0412e-
02,

```

```
        -6.8407e-02,  3.0207e-02,  3.4478e-01, -1.2855e-01,  1.4464e-  
02,  
        -1.2843e-03,  2.1935e-01, -7.2456e-02, -1.5156e-02, -3.1103e-  
01]])  
Layer: fc2.bias  
Biases: tensor([-0.3422,  0.2683,  0.1755,  0.3877,  0.2402, -0.2090,  
0.0988, -0.0726,  
0.3163,  0.2130])  
Layer: fc3.weight  
Weights: tensor([[ -0.1539, -0.3371, -0.4447,  0.3736, -0.3582, -0.4576,  
0.2441, -0.2715,  
-0.2363,  0.1108]])  
Layer: fc3.bias  
Biases: tensor([0.3047])
```

Anexo 16: Hiperparámetros para estimar Δa_m^l con función de activación Tangente hiperbólico en capas ocultas

```
#-----#
## Weights and biases for NN using tanh - author: Luis Gonzalez
#-----#
Layer: fc1.weight
Weights: tensor([[ 0.3048,  0.5785],
                 [-0.2757, -0.8571],
                 [-0.6696,  0.1839],
                 [-0.5213,  0.7336],
                 [-0.6856,  0.0905],
                 [-0.1111, -0.1923],
                 [-1.4447,  0.0392],
                 [-0.6947, -0.4491],
                 [ 0.6817,  0.1199],
                 [-1.1326, -0.1881],
                 [ 0.1444,  0.8578],
                 [ 0.1659,  0.7145],
                 [-0.7420, -0.3483],
                 [ 1.0528, -0.9080],
                 [-1.2630, -0.0284],
                 [-1.0993, -0.0919],
                 [-0.6508, -0.3551],
                 [-1.2262, -0.5485],
                 [-1.0252,  0.1909],
                 [ 0.4928,  0.4624]])

Layer: fc1.bias
Biases: tensor([-0.0126, -0.4224, -0.3634, -0.6163, -0.6144, -0.9755,
                0.0698, -0.3582,
                -0.5380,  0.0467,  0.9014,  1.1008, -0.2549,  1.0823, -0.1846,
                -0.8934,
                -0.0892, -1.2571, -1.0705, -0.0722])

Layer: fc2.weight
Weights: tensor([[ 2.1321e-02,  3.1063e-01, -2.1255e-01,  1.2574e-02,
                  3.4415e-01,
                  3.4112e-01,  5.5819e-01, -8.1621e-02,  1.0359e-01,  4.9327e-
01,
                  -3.9961e-01, -3.7723e-01, -9.3840e-02,  5.5859e-01,  6.1876e-
01,
                  -8.8004e-03,  8.4690e-02, -1.4674e-01, -1.5637e-01, -8.4515e-
02],
                 [-2.2494e-01,  1.6366e-01, -1.2814e-01, -5.1897e-01, -5.2144e-
01,
                  -6.3805e-02,  5.6391e-01, -3.9041e-01,  6.2259e-02, -6.8850e-
02,
                  1.1278e-01,  1.3123e-01, -1.7019e-01,  5.8992e-01, -1.6571e-
01,
                  -6.0414e-01, -6.0540e-02, -1.5286e-01, -5.9581e-01,  8.8969e-
02],
                 [ 2.5621e-01, -2.7955e-02, -3.6901e-01, -1.9521e-01, -7.6251e-
01,
                  -4.8793e-02,  2.7039e-01, -6.4988e-01,  1.4623e-02,  4.2396e-
02,
```

```

01,      4.8946e-02,  2.4074e-01, -5.6464e-01,  6.3298e-01,  1.5306e-
01],
[  4.0496e-01, -4.1669e-01,  1.9225e-01,  5.3519e-01,  3.7909e-
02,
-4.2277e-01, -4.9979e-01, -2.8318e-01,  2.7477e-02, -3.6895e-
01,
  3.1819e-01,  3.9895e-01, -1.3806e-01, -6.2526e-01, -1.4953e-
01,
  3.2054e-01, -1.8792e-01, -1.7545e-01,  4.1326e-01,  3.8282e-
01],
[  3.6383e-01,  2.7906e-01,  2.5850e-01,  3.6278e-01,  4.3010e-
01,
  3.5281e-01,  6.8476e-01, -4.0217e-01,  6.9925e-02, -1.5627e-
02,
-4.4288e-01, -5.1012e-01, -4.5264e-01,  3.3668e-01,  1.3347e-
01,
-3.2724e-01, -6.0936e-02, -1.4954e-01, -6.1025e-02,  4.2318e-
01],
[-1.3722e-01,  8.8231e-02, -2.7984e-01, -2.3819e-01,  2.9771e-
02,
  1.9509e-01,  1.7084e-01,  1.3026e-02, -3.3031e-01,  6.4824e-
02,
  5.8976e-02, -2.4979e-01,  8.3719e-02,  3.2701e-01, -2.5248e-
01,
-2.6663e-02,  3.3116e-02, -6.1939e-02,  2.5129e-02, -2.4275e-
01],
[  1.0053e-01, -2.4692e-01,  2.1230e-01,  1.1061e-01, -8.9447e-
02,
-1.8534e-01, -4.6472e-01,  5.3521e-02, -1.1330e-01, -6.8954e-
02,
  3.7912e-01,  3.6945e-01,  2.9324e-04, -2.6386e-01, -3.8431e-
01,
  3.1039e-02, -1.7747e-01,  6.6067e-03,  8.9685e-03, -2.8991e-
02],
[-1.2322e-01,  3.3045e-01, -2.4562e-01, -2.4603e-01, -7.6836e-
02,
  3.3845e-01,  2.1464e-01,  2.0684e-01, -7.3018e-02,  1.6204e-
01,
-1.6729e-01, -2.0756e-01, -3.8129e-02,  1.9862e-01,  8.0211e-
02,
-1.2886e-01, -4.8474e-02, -3.8896e-02,  2.5907e-02, -3.2769e-
01],
[  4.4968e-01, -3.6831e-01, -3.8595e-01, -1.3273e-01, -9.2647e-
01,
-3.2011e-01,  5.8663e-01, -9.9963e-01,  7.3165e-02, -1.4872e-
01,
  5.8630e-01,  7.0786e-01, -9.1844e-01,  8.9671e-01,  1.7028e-
02,
-9.7023e-01, -6.0264e-01, -8.8625e-01, -8.3985e-01,  8.7449e-
01],
[  1.5810e-01,  8.6690e-02, -1.0998e-02,  3.0734e-02,  6.1960e-
02,
-3.3878e-01,  2.2735e-01, -1.1681e-01,  8.9024e-02, -2.0266e-
02,

```

```

-3.3265e-02,  1.1875e-01,  2.9316e-01, -9.2727e-02,  2.3496e-
01,
-1.8104e-02, -2.9725e-01, -4.4901e-04, -4.1237e-02, -6.0636e-
02]])
Layer: fc2.bias
Biases: tensor([-0.5841,  0.0789,  0.0889,  0.5130, -0.4151, -0.3283,
 0.2523, -0.1578,
 0.4742,  0.1804])
Layer: fc3.weight
Weights: tensor([[ -0.4891, -0.3989, -0.6850,  0.3972, -0.5351, -0.2998,
 0.4003, -0.2431,
 -0.7193,  0.0895]])
Layer: fc3.bias
Biases: tensor([0.1932])

```