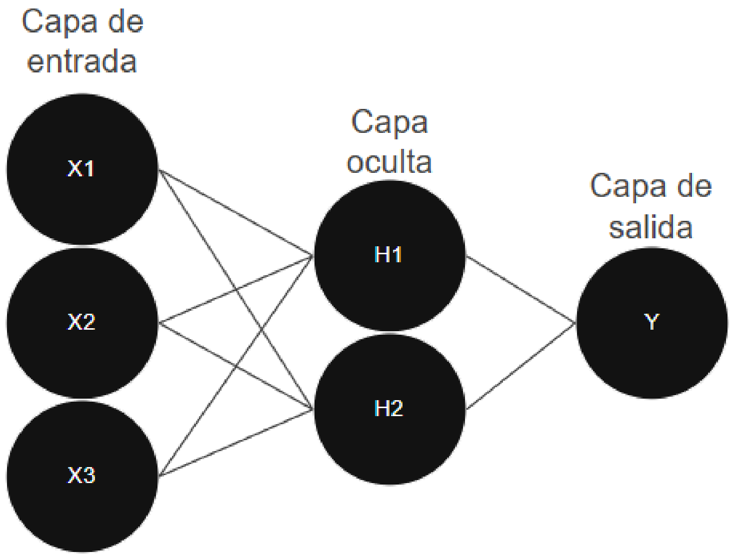
1. Ejercicio: Redes Neuronales Artificiales

1. Dibuja una red neuronal artificial con una capa de entrada, una capa oculta y una capa de salida.



1.1. Etiqueta los pesos de las conexiones con valores numéricos ficticios y asigna calores de entrada.

Entradas:
- X1 = 0.5
- X2 = 0.8
- X3 = 0.3

Pesos de entrada a Capa Oculta:
- X1 → H1: **w11 = 0.4**
- X1 → H2: **w12 = 0.7**
- X2 → H1: **w21 = 0.6**
- X2 → H2: **w22 = 0.2**
- X3 → H1: **w31 = 0.3**
- X3 → H2: **w32 = 0.9**

Pesos de Capa Oculta a Salida:
- H1 → Y: **v1 = 0.5**
- H2 → Y: **v2 = 0.8**

Bias:
- Bias H1 = 0.1
- Bias H2 = 0.2
- Bias Y = 0.15

1.2. Indica cómo calcularías la salida de la red utilizando las entradas, los pesos y una función de activación simple (por ejemplo, la función de activacion sigmoide).

Activación sigmoide → σ(z) = 1 / (1 + e^(-z))

Para calcular H1:

z1 = (X1 × w11) + (X2 × w21) + (X3 × w31) + bias_H1
z1 = (0.5 × 0.4) + (0.8 × 0.6) + (0.3 × 0.3) + 0.1
z1 = 0.2 + 0.48 + 0.09 + 0.1
z1 = 0.87

H1 = σ(z1) = 1 / (1 + e^(-0.87))
H1 = 1 / (1 + 0.4190)
H1 ≈ 0.705

Para calcular H2:
z2 = (X1 × w12) + (X2 × w22) + (X3 × w32) + bias_H2
z2 = (0.5 × 0.7) + (0.8 × 0.2) + (0.3 × 0.9) + 0.2
z2 = 0.35 + 0.16 + 0.27 + 0.2
z2 = 0.98

H2 = σ(z2) = 1 / (1 + e^(-0.98))
H2 = 1 / (1 + 0.3753)
H2 ≈ 0.727

Para calcular la Salida:
z_out = (H1 × v1) + (H2 × v2) + bias_Y
z_out = (0.705 × 0.5) + (0.727 × 0.8) + 0.15
z_out = 0.3525 + 0.5816 + 0.15
z_out = 1.0841

Y = σ(z_out) = 1 / (1 + e^(-1.0841))
Y = 1 / (1 + 0.3382)
Y ≈ 0.747

Característica	X1	X2	X3	H1	H2	Bias
H1	w11 = 0.4	w21 = 0.6	w31 = 0.3	[No weight from H1 to H1]	[No weight from H2 to H1]	0.1
H2	w12 = 0.7	w22 = 0.2	w32 = 0.9	[No weight from H1 to H2]	[No weight from H2 to H2]	0.2
Y	[No weight from X1 to Y]	[No weight from X2 to Y]	[No weight from X3 to Y]	v1 = 0.5	v2 = 0.8	0.15

2. ¿Por qué las redes neuronales artificiales se entrenan ajustando los pesos en las conexiones entre neuronas? Explica brevemente.

Las redes neuronales aprenden ajustando sus pesos, estos son los paámentros que controlan la fuerza de las conexiones. Durante el entrenamiento se van comparando las salidas con los resultados correctos, calculando el error y usan algoritmos como la retropropagación para modificar los pesos e ir poco a poco reduciendo los errores. De esta manera y tras muchas repeticiones esto va mejorando hasta que las salidas son casi correctas.

2. Ejercicio: Redes Neuronales Artificiales

1. Considera el siguiente escenario: Un médico quiere saber la probabilidad de que un paciente tenga una enfermedad E dado que presenta un síntoma S. La probabilidad inicial de que alguien tenga la enfermedad (antes de observar los síntomas) es del 1%. La probabilidad de presentar el síntoma si se tiene la enfermedad es del 80%. La probabilidad de presentar el síntoma sin tener la enfermedad es del 10%.

   Utiliza el **teorema de Bayes** para calcular la probabilidad de que el paciente tenga enfermedad si tiene el sintoma

$P(E) = 0.01$
$P(S|E) = 0.8$
$P(S|\neg E) = 0.1$
$P(\neg E) = 0.99$

1.1 Probabilidad total de presentar el síntoma.

$P(S) = P(S|E) \times P(E) + P(S|\neg E) \times P(\neg E)$
$P(S) = (0.8 \times 0.01) + (0.1 \times 0.99)$
$P(S) = 0.008 + 0.099$
$P(S) = 0.107$

1.2 Aplicar teorema de Bayes.

$P(E|S) = [P(S|E) \times P(E)] / P(S)$
$P(E|S) = (0.8 \times 0.01) / 0.107$
$P(E|S) = 0.008 / 0.107$
$P(E|S) \approx 0.0748$
$P(E|S) \approx 7.48\%$

1.2 Resultado.

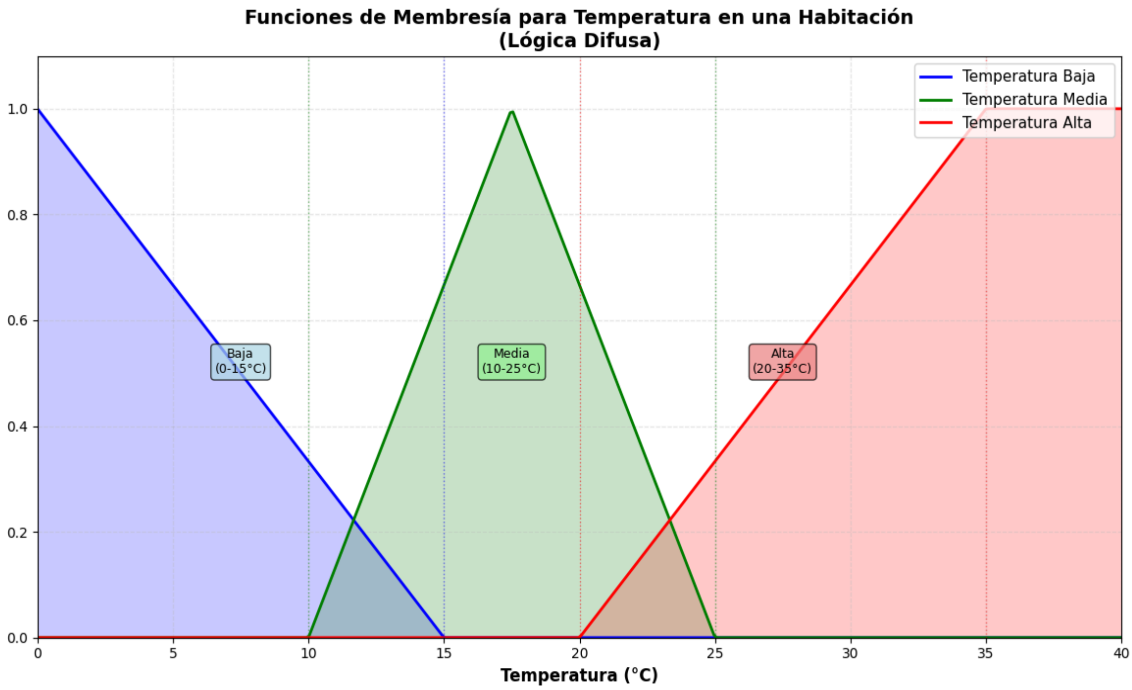
La probabilidad de que el paciente tenga la enfermedad dado que presenta el síntoma es aproximadamente del 7.48%

2. Explica en tus palabras qué significa la probabilidad condicional $P(E \mid S)$ y cómo se utiliza en la inferencia bayesiana.

La probabilidad condicional $P(E \mid S)$ significa la probabilidad de que un paciente tenga la enfermedad después de saber que presenta el síntoma. En la inferencia bayesiana, esto se usa para actualizar una creencia inicial (como la probabilidad de tener la enfermedad antes de ver síntomas) con nueva evidencia (el hecho de que el paciente tiene el síntoma). Así, se combina lo que ya se sabía con lo que se observa, para obtener una estimación más precisa y útil en la toma de decisiones, como en un diagnóstico médico.

3. Ejercicio: Lógica Difusa

1.1 Dibuja una función de membresía para la temperatura en una habitación, donde los valores pueden ser "baja", "media" y "alta". Asegúrate de que:



1.2. ¿En qué se diferencia la lógica difusa de la lógica clásica (verdadero/falso)? Explica por qué la lógica difusa es útil en situaciones de incertidumbre.

La lógica clásica (también llamada binaria o booleana) solo acepta dos valores: verdadero (1) o falso (0). Por ejemplo, “la temperatura es alta” sería totalmente cierto o totalmente falso, sin matices.

En cambio, la lógica difusa permite grados intermedios de verdad. Algo puede ser “un poco alto”, “muy bajo” o “medianamente cálido”, representado con valores entre 0 y 1. Esto refleja mejor cómo razonamos los humanos en la vida real.

La lógica difusa es muy útil en situaciones de imprecisión como controlar la temperatura de una habitación, ajustar la velocidad de un tren, o regular el lavado en una lavadora. En estos casos, no hay límites claros, y la lógica difusa permite tomar decisiones suaves y adaptativas basadas en grados de pertenencia, no en reglas rígidas.

4. Ejercicio: Sistemas Basados en Reglas

1.1 Escribe 5 reglas "si-entonces" (IF-THEN) para un sistema experto que diagnostique problemas en una computadora. Usa como ejemplo lo siguiente:

- Regla 1: Si el ordenador se apaga repentinamente y el ventilador está muy ruidoso, entonces el problema es el sobrecalentamiento del procesador.
- Regla 2: Si el teclado no responde y está desconectado por Bluetooth, entonces el problema es la conexión inalámbrica del teclado.
- Regla 3: Si aparecen pantallas azules frecuentes y se instalaron nuevos controladores, entonces el problema es un controlador incompatible.
- Regla 4: Si el ordenador no detecta internet y el router está apagado, entonces el problema es la falta de conexión a la red local.
- Regla 5: Si los programas se cierran solos y la memoria RAM está casi llena, entonces el problema es la falta de memoria disponible.

1.2 ¿Cuál es la ventaja de utilizar sistemas basados en reglas para diagnosticar problemas en comparación con un humano experto? ¿Qué limitaciones podrían tener estos sistemas?

- Ventaja principal:

Los sistemas basados en reglas son rápidos, consistentes y disponibles todo el tiempo. A diferencia de un humano, no se cansan, no olvidan reglas y dan el mismo diagnóstico ante los mismos síntomas. Además, pueden usarse en lugares donde no hay expertos disponibles, como en soporte técnico remoto o en software de autodiagnóstico.

- Limitaciones principales:

Estos sistemas solo saben lo que se les ha programado. Si aparece un problema nuevo o una combinación de síntomas que no está en sus reglas, no pueden razonar más allá de eso. Tampoco entienden el contexto, el “sentido común” o matices que un humano sí captaría. Por ejemplo, no sabrían interpretar una descripción vaga como “la computadora va rara”, a menos que esa frase esté explícitamente traducida a reglas.

5. Ejercicio: Sistemas Basados en Reglas

Un test de una enfermedad tiene un 95% de probabilidad de detectar correctamente a las personas enfermas (sensibilidad), pero también da un resultado positivo en un 5% de las personas sanas (falsos positivos). Sabemos que un 1% de la población está enferma.

Calcula la probabilidad de que una persona esté realmente enferma si ha dado positivo en el test. Utiliza los siguientes datos

- $P(E) = 0.01$ (Probabilidad de estar enfermo)
- $P(P \mid E) = 0.95$(Probabilidad test positivo si la persona está enferma)
- $P(P \mid \neg E) = 0.05$ (Probabilidad test positivo si la persona no está enferma)

$P(\neg E)=1-P(E)=0.99 \rightarrow$ Probabilidad de no estar enfermo.

Aplicando el teorema de Bayes

$P(P)=(0.95)(0.01)+(0.05)(0.99)=0.0095+0.0495=0.059$

$P(E \mid P)= 0.059 / 0.0095 \approx 0.161$ç

Esto significa que aunque el test dé positivo, la probabilidad real de que la persona esté enferma es aproximadamente del 16.1%