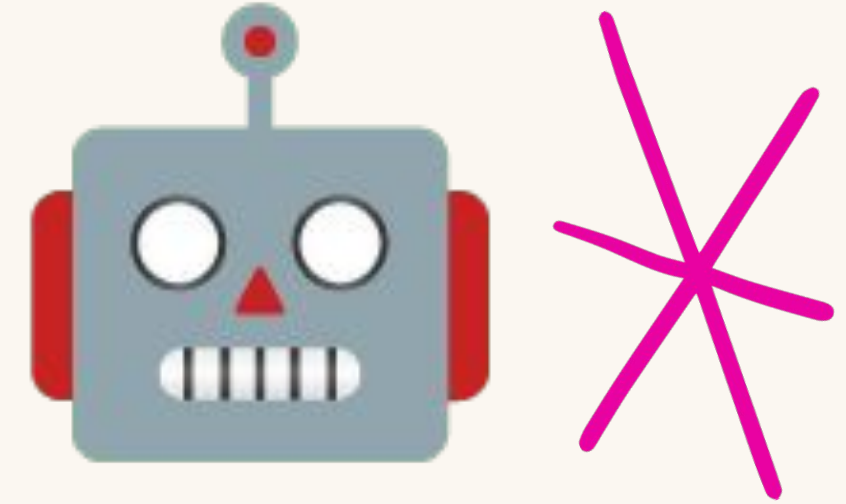


Desbloqueando el Poder de los Datos



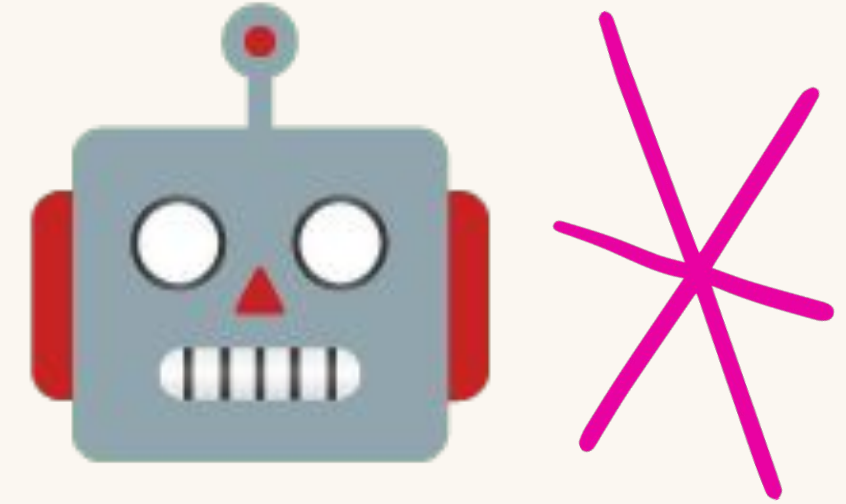
# Inteligencia Artificial & Ciencia de Datos para todos

Descanzo. Regresamos a las: 8:08 a.m.

¿Te gustaría comenzar el día con alguna canción en específico?

Coméntala en el chat 🎵💬

Desbloqueando el Poder de los Datos



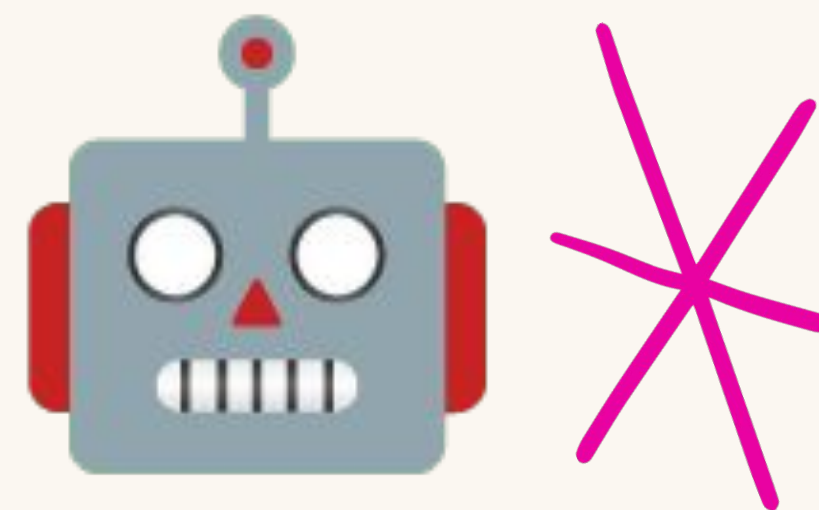
# Inteligencia Artificial & Ciencia de Datos para todos

Comenzamos a las 7:05 a.m. en punto.

¿Te gustaría comenzar el día con alguna canción en específico?

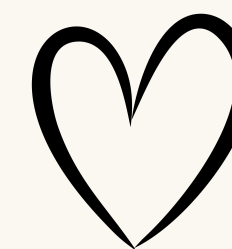
Coméntala en el chat 🎵💬

Desbloqueando el Poder de los Datos



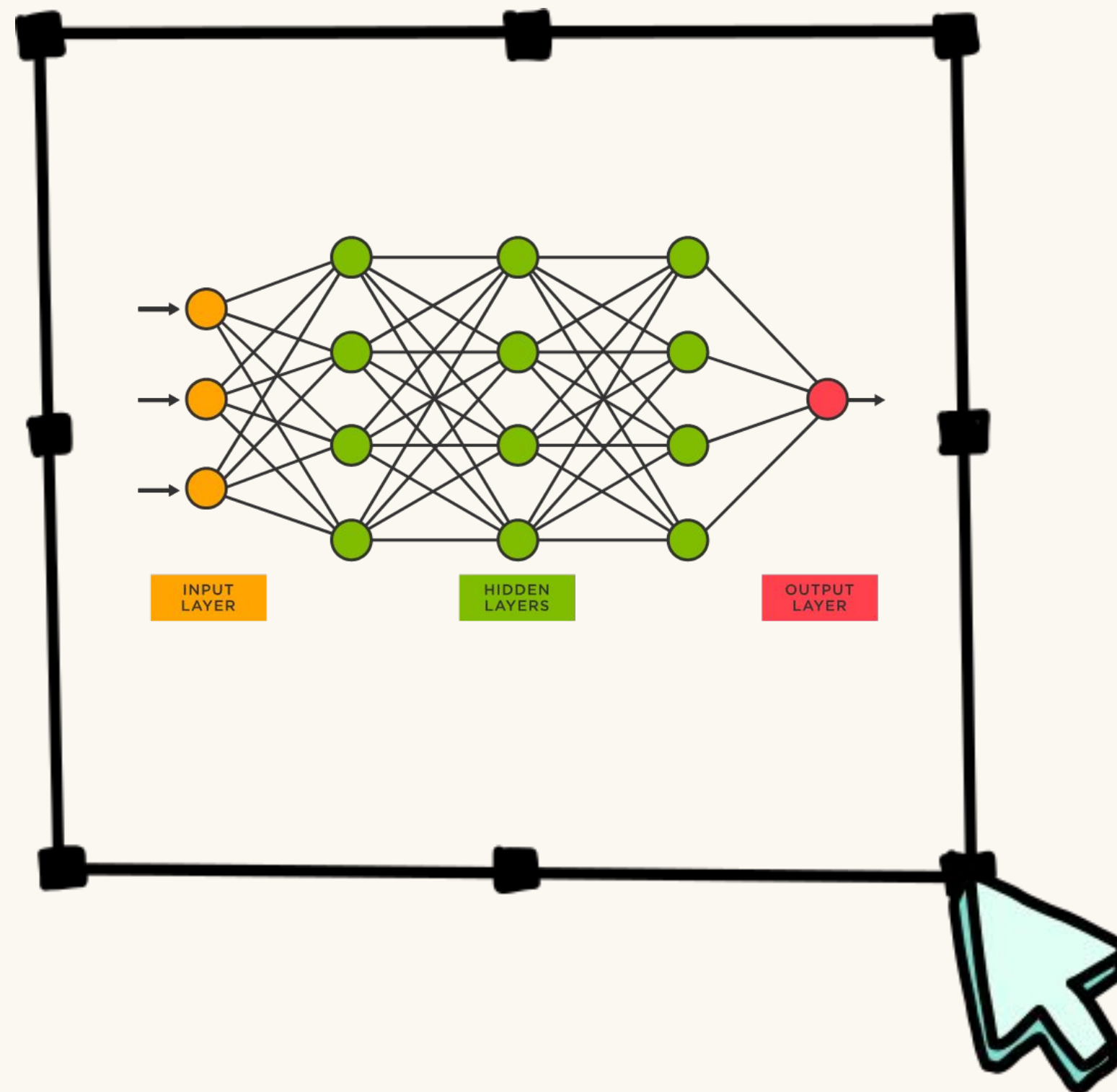
# Inteligencia Artificial & Ciencia de Datos para todos





# Introducción a redes neuronales

Noviembre 5, 2024







1. Pautas entrega final
2. Redes Neuronales
3. Código



# Temas tercer corte:

Clase #	Fecha	Tema	
Clase #11	Octubre 15, 2024	Modelos NO supervisados: K-Medias	
Clase #12	Octubre 22, 2024	Modelos de Ensemble: Random Forest y GB	
Clase #13	Octubre 29, 2024	Comunicación de negocios y desarrollo profesional	Escoger grupo/tema/datos
Clase #14	Noviembre 5, 2024	Introducción a redes neuronales	Entrega #1: Código y diapositivas
Clase #15	Noviembre 12, 2024	Presentaciones finales	Entrega #2: Mismo código y diapositivas pero incluyendo retroalimentación
Clase #16	Noviembre 19, 2024	Presentaciones finales	

Página web del curso con videos, diapositivas, código y más:  
<https://github.com/vivianamarquez/unicomfacauca-ai-2024>

# Entregas

1. Paso #1 - Octubre 28 - [Enlace](#)  
Escoger grupo, tema, tipo de modelo y conjunto de datos
2. Paso #2 - Noviembre 4  
Primera entrega del proyecto final.  
Este día deben entregar código y diapositivas.  
Les haré retroalimentación que deben incorporar en la entrega final.
3. Paso #3 - Noviembre 11  
Entrega final del proyecto final.  
Van a entregar el mismo código y diapositivas, pero incluyendo la retroalimentación dada.  
Estas serán las mismas diapositivas y código que se usará en la presentación final.
4. Paso #4 - Noviembre 12/19.  
Presentación de su proyecto final.  
Hay que asistir las dos fechas.

# Talleres tercer corte:

Taller #	Descripción	Mandatorio	Se aceptan entregas tardes	Enlace	Fecha de entrega	Porcentaje en el tercer corte	Porcentaje en el curso
Taller # 11	K-Medias	Sí	Sí	Diapositiva 59	Octubre 21, 2024	10%	4%
Taller # 12	Modelos de Ensamble	Sí	Sí	Diapositiva 31	Octubre 28, 2024	10%	4%
Encuesta	Responder a la pregunta: ¿Qué te gustaría saber/aprender en la clase de desarrollo profesional?	Sí	NO	<a href="#">Enlace</a>	Octubre 28, 2024	5%	2%
Excel Grupos	Decidir si van a trabajar individualmente o en parejas para el proyecto final, escoger proyecto y seleccionar datos	Sí	Preferiblemente no, pero se acepta hasta Nov 4	<a href="#">Enlace</a>	Octubre 28, 2024	5%	2%
Primera Entrega Proyecto Final	Entrega de diapositivas y código del proyecto final	Sí	NO	Diapositiva 9	Noviembre 4, 2024	15%	6%
Segunda Entrega Proyecto Final	Entrega de diapositivas y código del proyecto final incorporando la retroalimentación	Sí	NO		Noviembre 11, 2024	25%	10%
Presentación	Presentación final	Sí	NO (Estar presente en las dos sesiones)		Noviembre 12 y 19	30%	12%



# Primera entrega del proyecto final

## 1. Expectativas del código

Regresión/Clasificación

<https://colab.research.google.com/drive/1M0ed9qFcAJRJu5EAsXcllokvnqjDqtMp?usp=sharing>

Agrupamiento


<https://colab.research.google.com/drive/1Yo963g3o-hu8NEty8xL0almXrZQsR4K5?usp=sharing>

## 1. Diapositivas

Debe contener lo mismo que en las expectativas del código, pero adicionalmente debe definir claramente su audiencia. Por favor enviar **enlace de Google slides** para poderles dar la retroalimentación ahí mismo.

- El día de la presentación, ambas personas en el grupo deben hablar
- Sugerencia para los ingenieros: Subir el proyecto a GitHub



1.  Pautas entrega final
2. Redes Neuronales
3. Código







*Artificial Intelligence*

The diagram consists of three overlapping circles. The largest circle is pink and contains the text 'Artificial Intelligence' and 'Inteligencia Artificial'. Inside it is a blue circle containing 'Machine Learning' and 'Aprendizaje automático'. Inside the blue circle is a yellow circle containing 'Deep Learning' and 'Aprendizaje profundo'.


Inteligencia Artificial

*Machine Learning*

Aprendizaje automático

*Deep Learning*

Aprendizaje profundo



## Inteligencia Artificial

**Definición:** La inteligencia artificial (IA) se refiere al desarrollo de sistemas informáticos que pueden realizar tareas que normalmente requieren inteligencia humana.

**Alcance:** Amplio. La IA abarca todo lo que permite a las computadoras imitar la inteligencia humana, incluyendo la robótica, el procesamiento del lenguaje natural y la resolución de problemas.





*Artificial Intelligence*

The diagram consists of three overlapping circles. The largest circle is pink and contains the text 'Artificial Intelligence' and 'Inteligencia Artificial'. Inside it is a blue circle containing 'Machine Learning' and 'Aprendizaje automático'. Inside the blue circle is a yellow circle containing 'Deep Learning' and 'Aprendizaje profundo'. The blue circle is also outlined with a purple border.

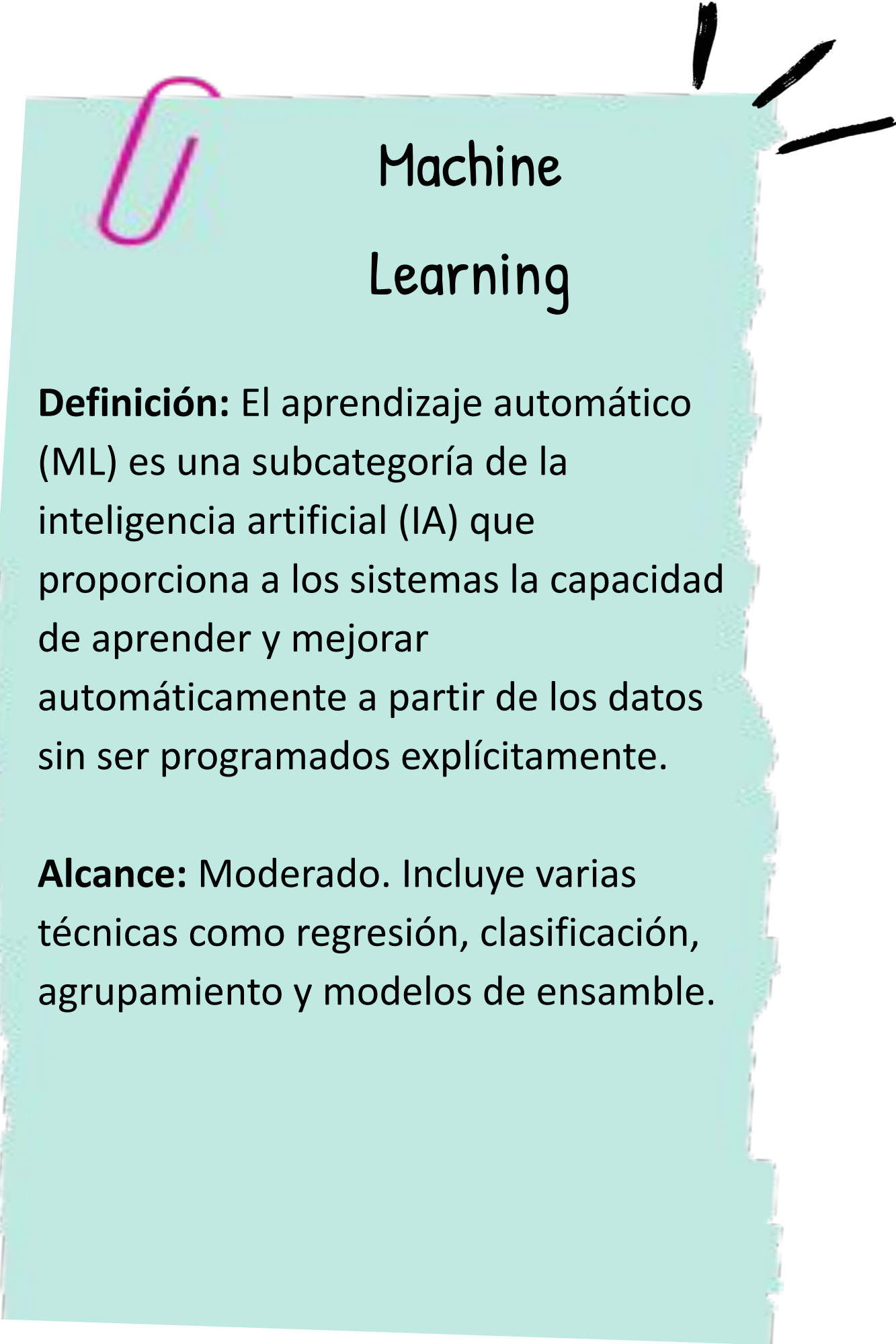
Inteligencia Artificial

*Machine Learning*

Aprendizaje automático

*Deep Learning*

Aprendizaje profundo



## Machine Learning

**Definición:** El aprendizaje automático (ML) es una subcategoría de la inteligencia artificial (IA) que proporciona a los sistemas la capacidad de aprender y mejorar automáticamente a partir de los datos sin ser programados explícitamente.

**Alcance:** Moderado. Incluye varias técnicas como regresión, clasificación, agrupamiento y modelos de ensamble.





*Artificial Intelligence*

The diagram consists of three overlapping circles. The largest circle is pink and contains the text 'Artificial Intelligence' and 'Inteligencia Artificial'. Inside it is a blue circle containing 'Machine Learning' and 'Aprendizaje automático'. Inside the blue circle is a yellow circle containing 'Deep Learning' and 'Aprendizaje profundo'. The yellow circle is also outlined with a purple border.

Inteligencia Artificial

*Machine Learning*

Aprendizaje automático

*Deep Learning*

Aprendizaje profundo



## Deep Learning

**Definición:** El aprendizaje profundo (DL) es una subcategoría del aprendizaje automático (ML) que utiliza redes neuronales. Estas redes intentan simular el comportamiento del cerebro humano—aunque aún lejos de igualar su capacidad—para "aprender" a partir de grandes cantidades de datos.

**Alcance:** Reducido. DL es una forma específica, pero poderosa, de aprendizaje automático.



*Artificial Intelligence*

Inteligencia Artificial

*Machine Learning*

Aprendizaje automático

*Deep Learning*

Aprendizaje profundo



En este curso nuestro principal enfoque fue **machine learning** (aprendizaje automático)

Ventas				
Fecha	Producto	Potencia	Unidades	Ganancias
enero	Bicicletas	Eléctrica	476	\$751.604
enero	Bicicletas	Manual	302	\$581.350
enero	Motonetas	Eléctrica	387	\$427.248
enero	Motonetas	Manual	309	\$48.513
enero	Patinetas	Eléctrica	251	\$135.791
febrero	Bicicletas	Eléctrica	354	\$558.966
febrero	Bicicletas	Manual	219	\$336.165
febrero	Motonetas	Eléctrica	312	\$583.128
febrero	Motonetas	Manual	419	\$396.793



*Artificial Intelligence*

Inteligencia Artificial

*Machine Learning*

Aprendizaje automático

*Deep Learning*

Aprendizaje profundo

 PyTorch

 TensorFlow

Hoy aprenderemos un poco  
de **deep learning**  
(aprendizaje profundo)



# ¿Cuándo usar Deep Learning vs Machine Learning?

## Machine Learning

- Pocos datos
- Tenemos poco poder computacional
- La interpretación de los modelos es importante
- Queremos entrenar más rápido
- Problemas que por lo general han tenido éxito con ML

## Deep Learning

- Muchos datos
- Tenemos poder computacional
- Los patrones de nuestros datos son muy complejos
- Problemas que por lo general han tenido éxito con ML



**STILL WAITING**



**FOR MY ML MODEL TO TRAIN**

[makeameme.org](http://makeameme.org)

# GPU vs CPU

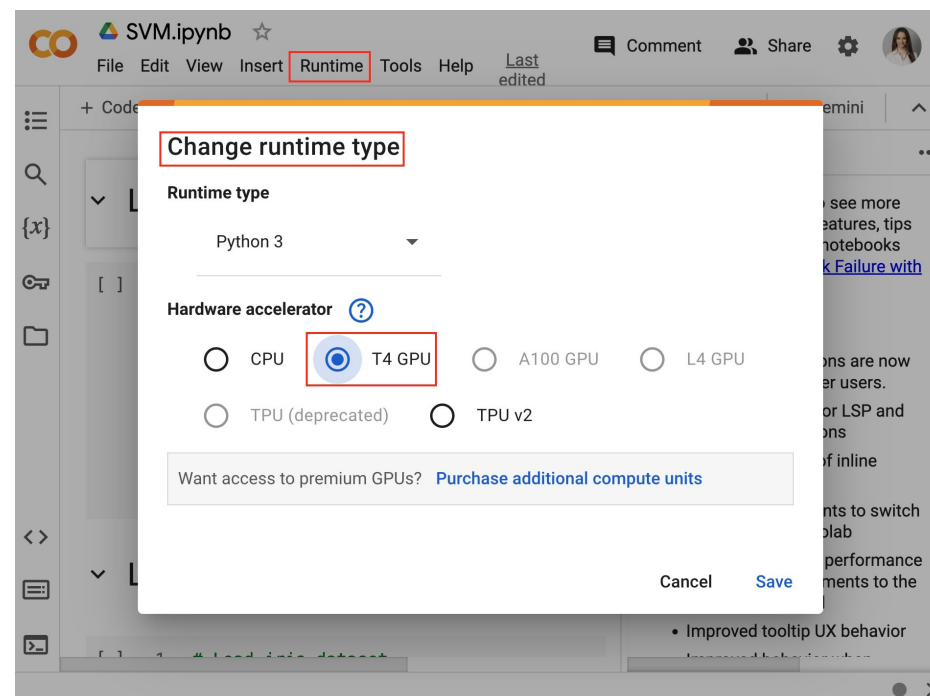
Las GPU (Unidades de Procesamiento Gráfico) y las CPU (Unidades Centrales de Procesamiento) están diseñadas con arquitecturas diferentes y para fines distintos.

## **Paralelismo:**

- GPUs: Originalmente diseñadas para renderizar gráficos, las GPU están pensadas para manejar múltiples tareas simultáneamente. Tienen miles de núcleos más pequeños diseñados para el procesamiento paralelo, lo que las hace expertas en el manejo de múltiples cálculos simultáneamente.
- CPUs: Suelen tener menos núcleos (por ejemplo, 4, 8 o 16 núcleos en las CPU modernas). Están optimizadas para tareas que requieren un procesamiento secuencial y son más adecuadas para tareas de propósito general y para gestionar las operaciones del sistema.



# What is Google Colab?



EC2

VM

GCE

**ORIGINAL**

**DEEPFAKE**

**EM HONEST**

**Y \*THIS\* ONE**

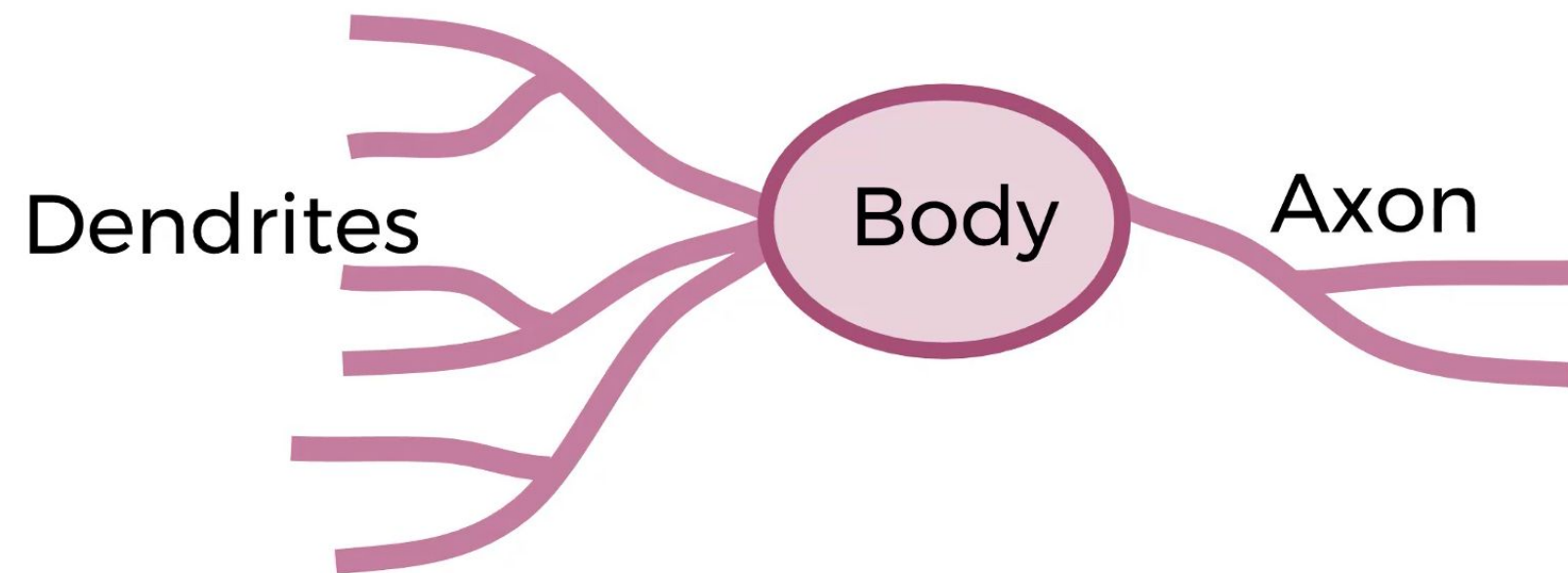
**EM HONEST**

**Y \*THIS\* ONE**

# Redes Neuronales

- Una red neuronal está compuesta de neuronas
- Las Redes Neuronales Artificiales (ANN) están inspiradas las redes neuronales biológicas

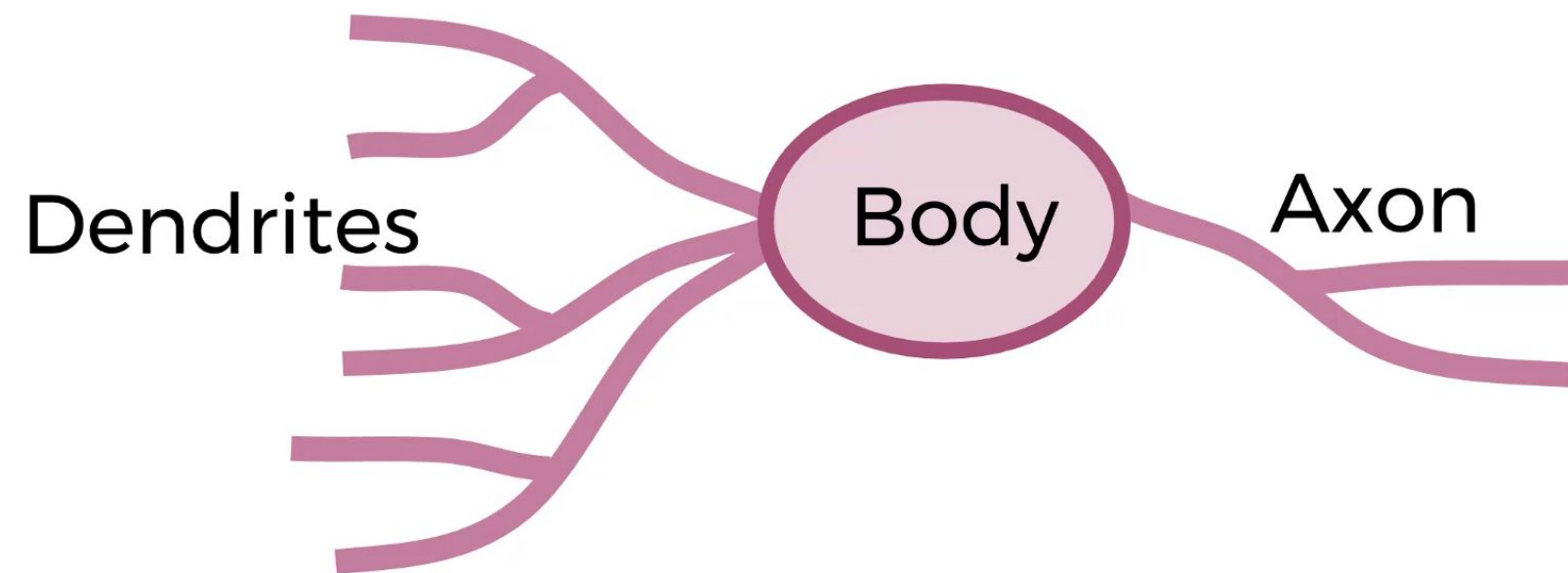
# Una neurona biológica



De una manera simplificada:

- Las dendritas alimentan el cuerpo de la célula a través de señales eléctricas
- La respuesta es después pasada a través del axón

# Una neurona biológica



Una neurona  
artificial se llama  
**Perceptrón**

De una manera simplificada:

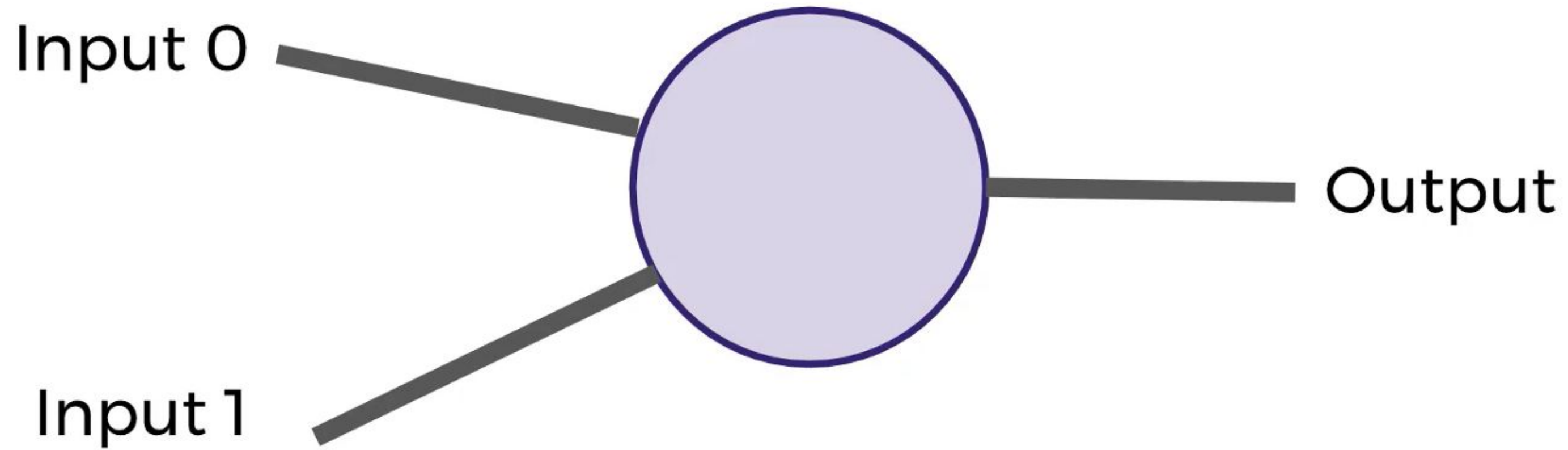
- Las dendritas alimentan el cuerpo de la célula a través de señales eléctricas
- La respuesta es después pasada a través del axón



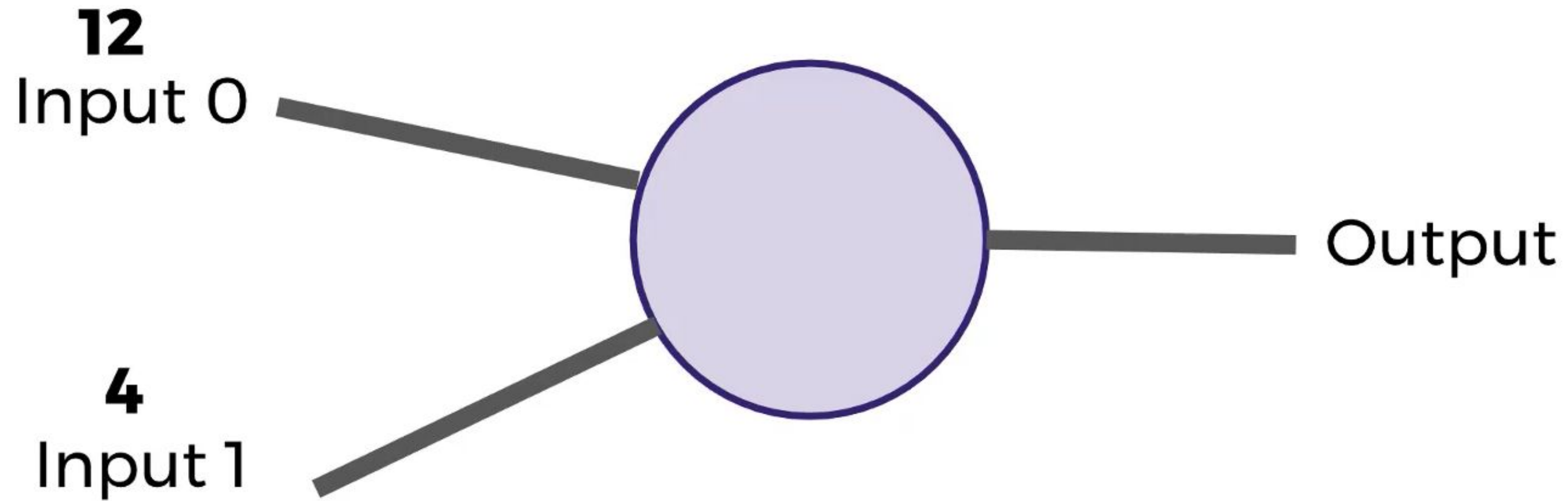
# Una neurona artificial = Perceptrón

- El perceptrón es una de las formas más sencillas de red neuronal
- Puede considerarse la base de redes más complejas

# Perceptrón

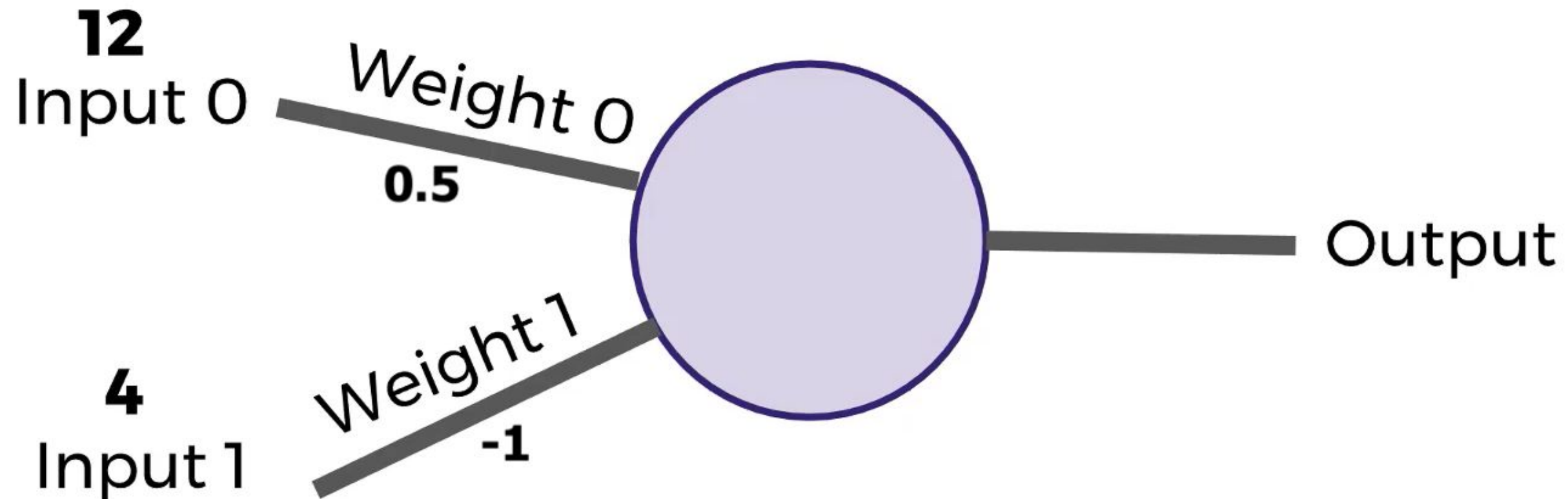


# Perceptrón



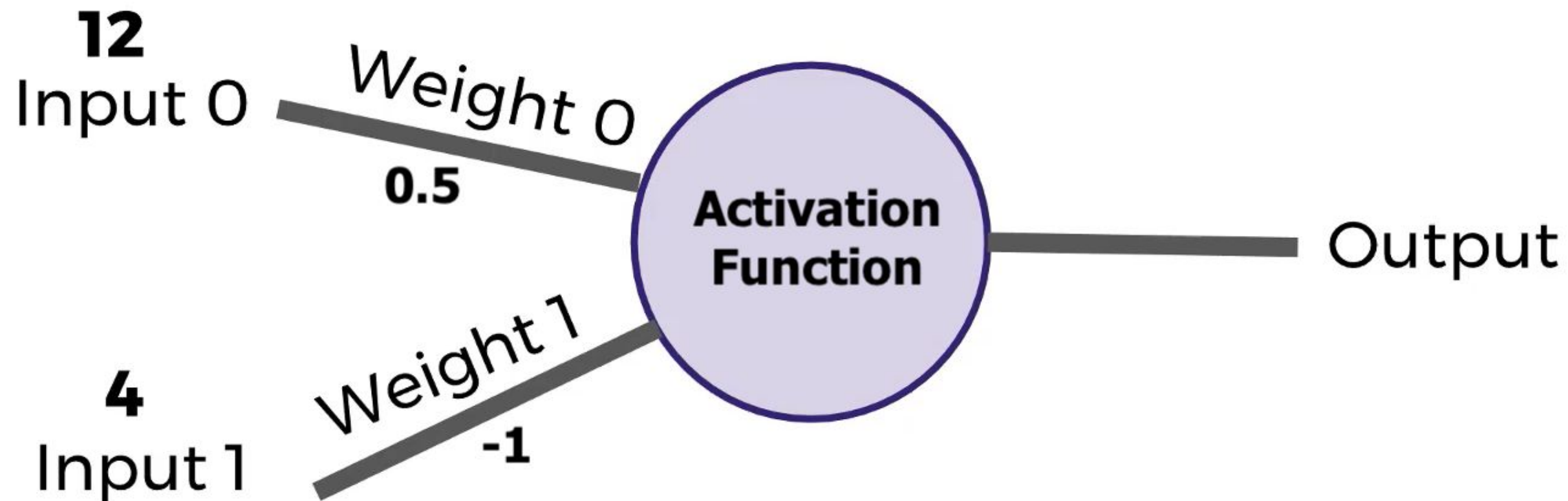
- Como todo modelo de machine learning, tenemos valores de entrada

# Perceptrón



- Después, los valores de entrada son multiplicados por pesos
- Estos pesos son inicializados de manera aleatoria

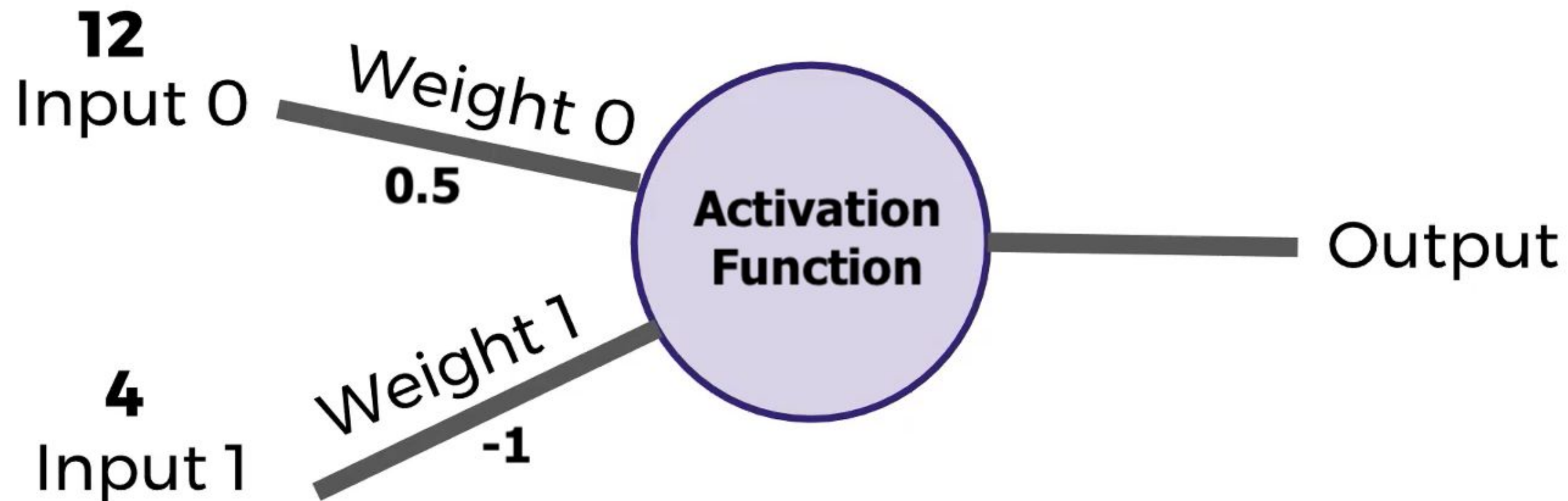
# Perceptrón



- El resultado (en este ejemplo  $12 \cdot 0.5 + 4 \cdot -1 = 2$ ) se pasa por una función de activación
- Existen muchas funciones de activación

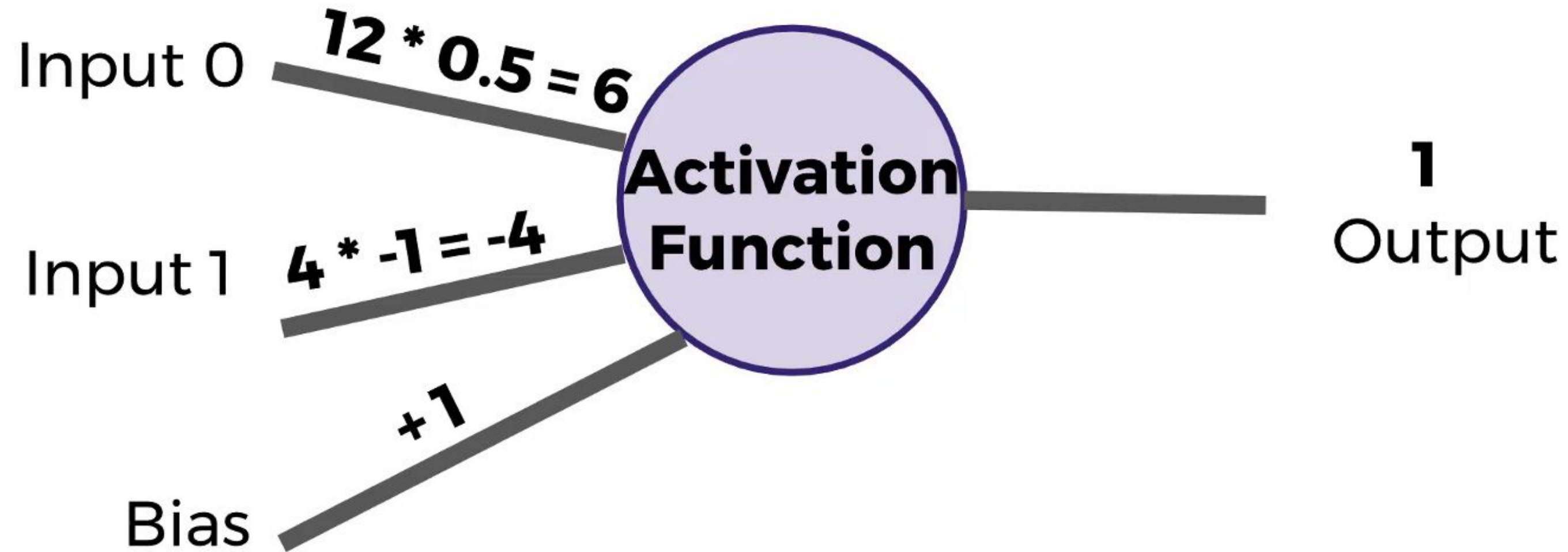


# Perceptrón



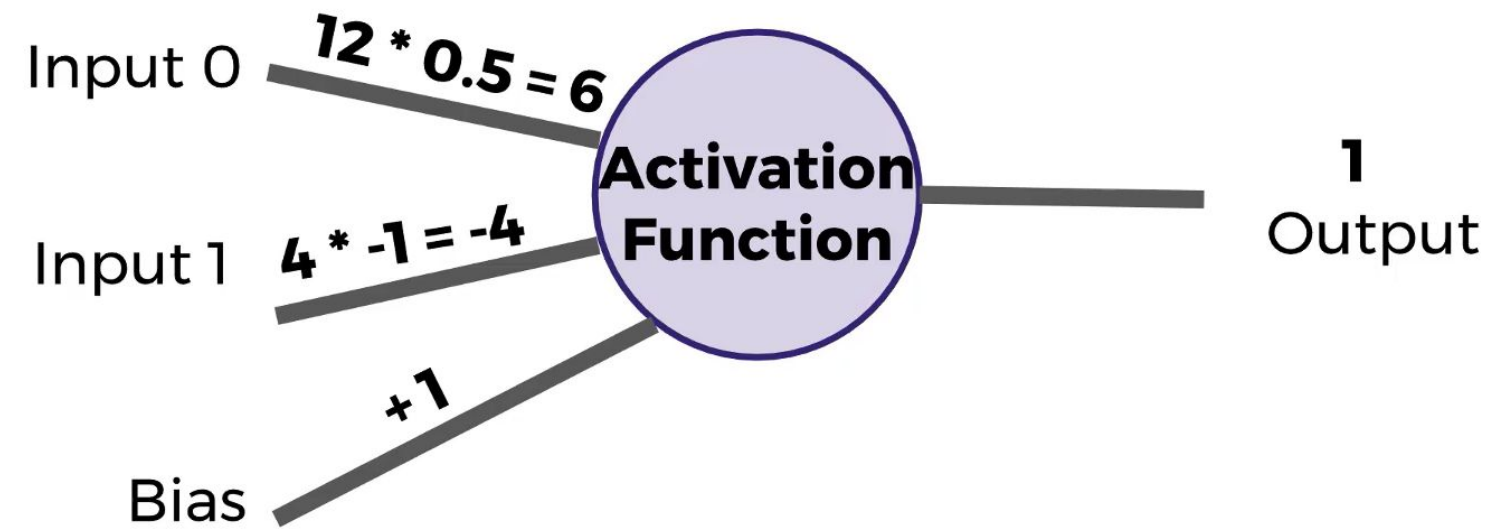
- El resultado (en este ejemplo  $12 \cdot 0.5 + 4 \cdot -1 = 2$ ) se pasa por una función de activación
- Existen muchas funciones de activación- **EJEMPLO:** Positiva=1, Negativa=0

# Perceptrón



- Se agrega un sesgo para evitar problemas matemáticos

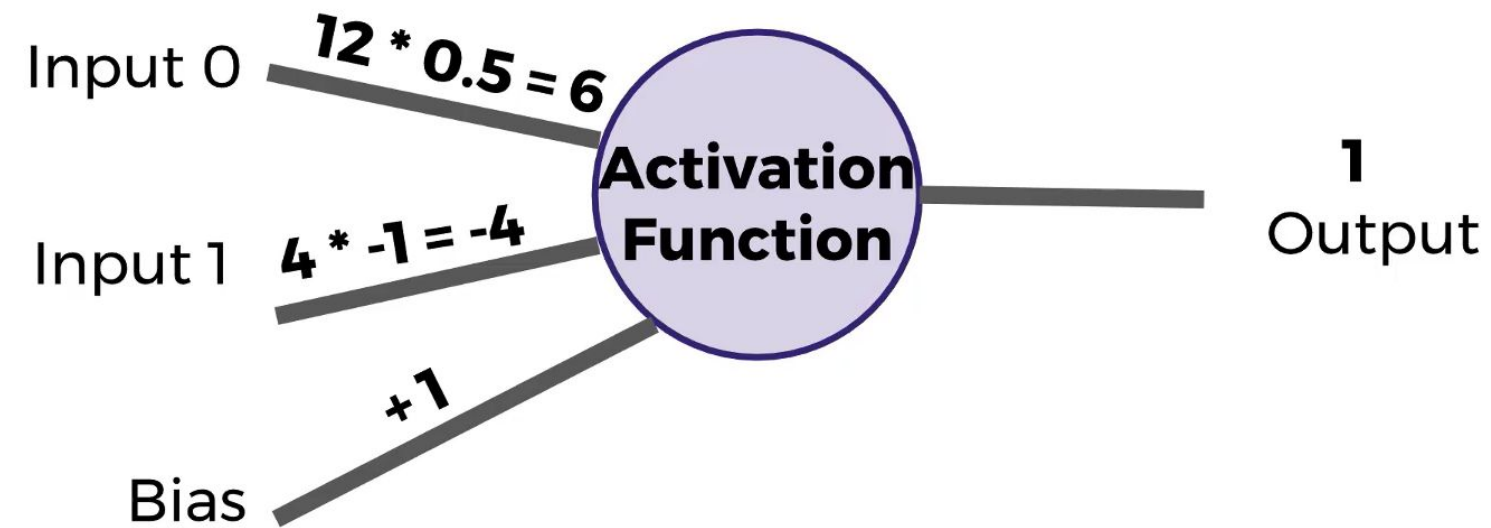
# Perceptrón



- Matemáticamente tenemos:  $\sum_{i=0}^n w_i x_i + b$

Cuando tenemos una red neuronal compuesta de varios perceptrones, se extiende a forma matricial.

# Perceptrón

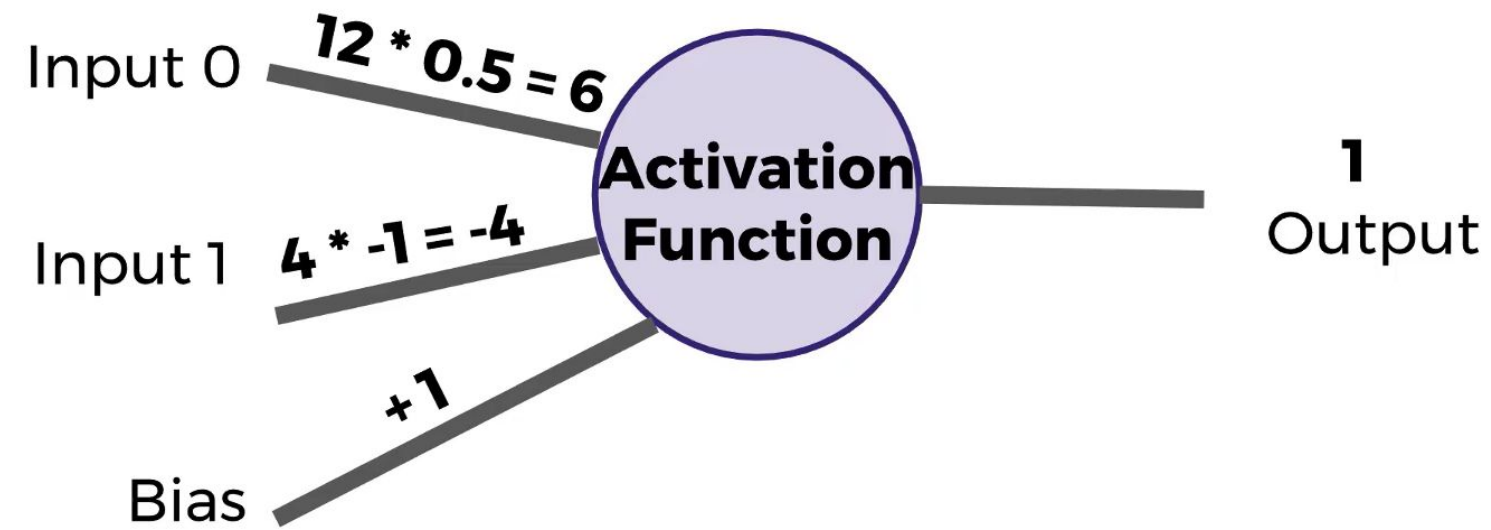


Un perceptrón es simplemente un regresor lineal con picante:  
La función de activación

- Matemáticamente tenemos:  $\sum_{i=0}^n w_i x_i + b$

Cuando tenemos una red neuronal compuesta de varios perceptrones, se extiende a forma matricial.

# Perceptrón

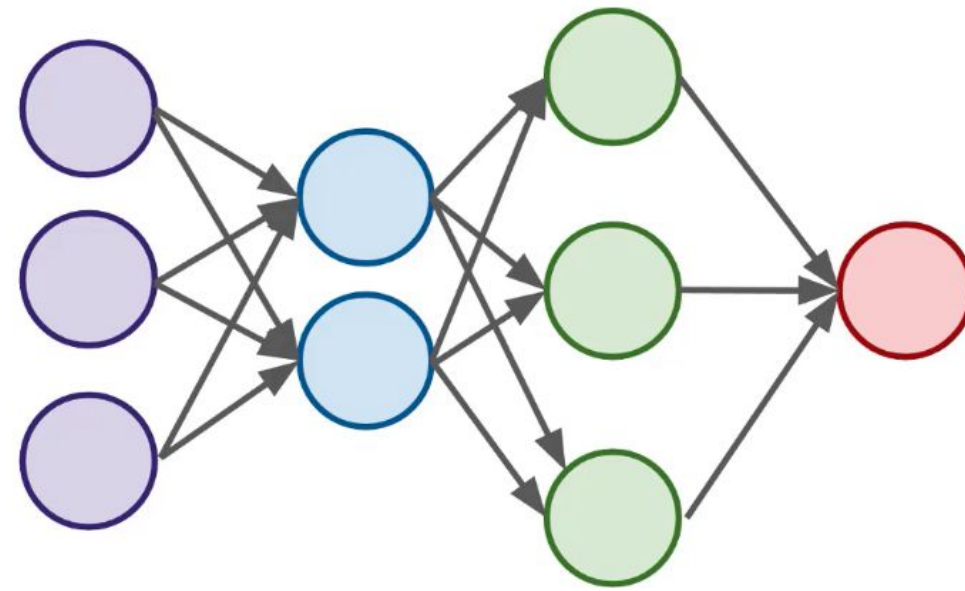


A perceptron is simply a Linear classifier with a twist: Activation Function

- In math terms, we have  $\sum_{i=0}^n w_i x_i + b$

When we have a neural network composed of several perceptrons, it is extended to matrix form

# Una red neuronal artificial



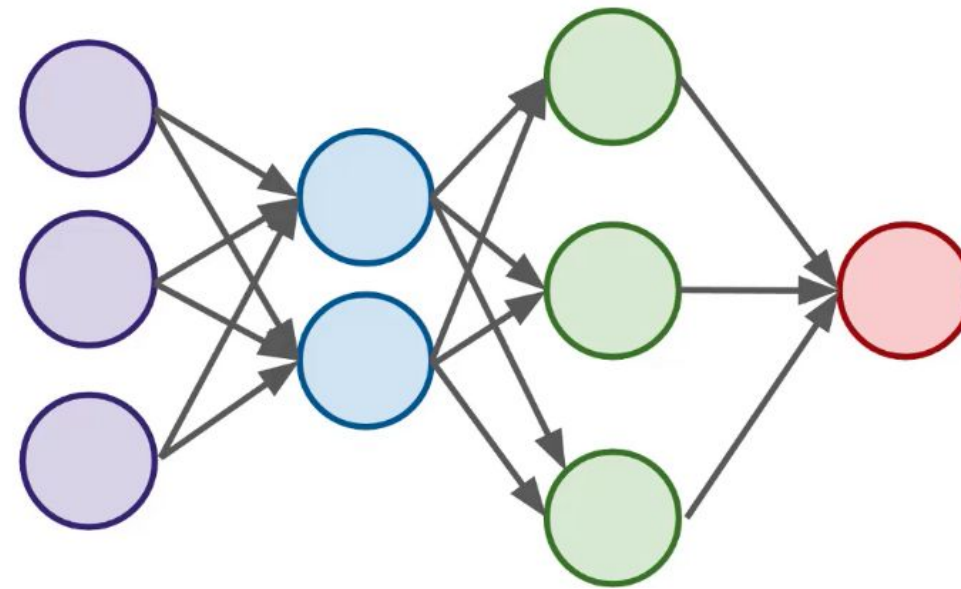
Partes:

- Capa de entrada (input): Valores reales de los datos
- Capas ocultas (2 en este caso): Deep network
- Capa de salida (output): Estimado final

A medida que crece el número de capas, crece el nivel de abstracción



# Una red neuronal artificial



Partes:

- Capa de entrada (input): Valores reales de los datos
- Capas ocultas (2 en este caso): Deep network
- Capa de salida (output): Estimado final

A medida que crece el número de capas, crece el nivel de abstracción



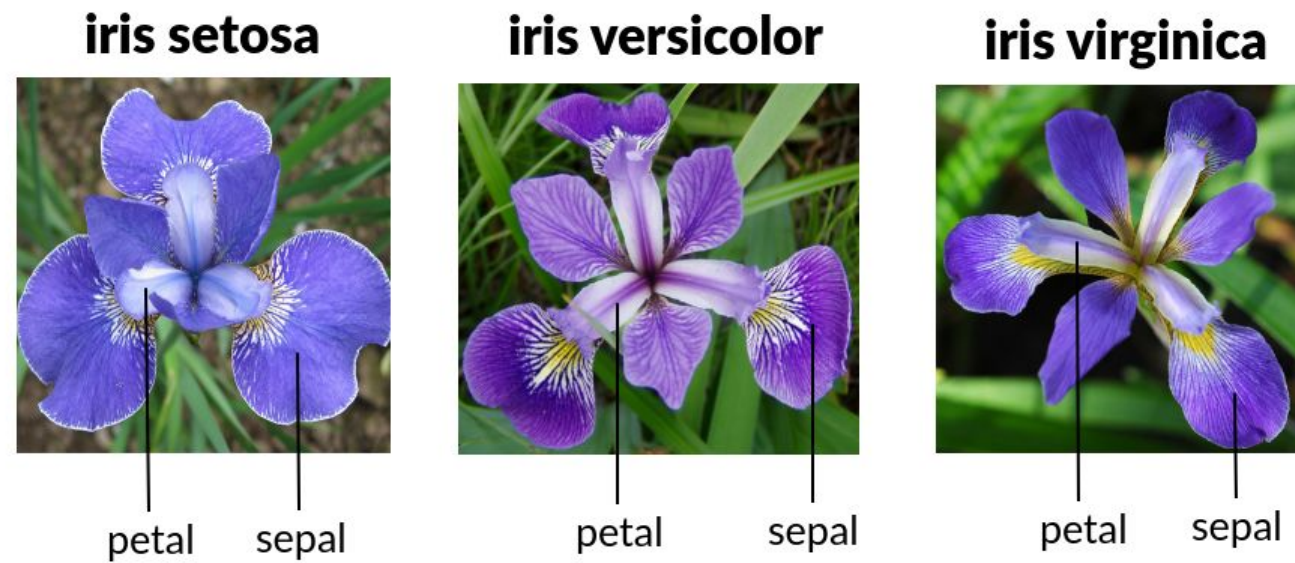
Las NN envían los datos a un espacio donde sean separables linealmente

Me using neural network for simple regression problem





# Una red neuronal artificial



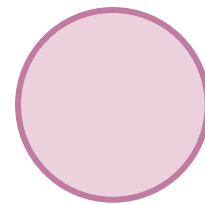
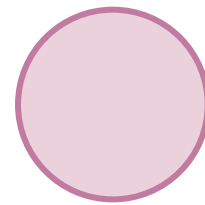
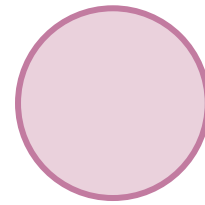
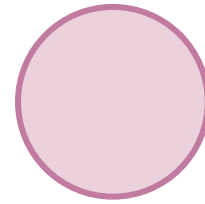
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa



# Una red neuronal artificial



	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa



**Capa de entrada:** Cuatro neuronas correspondientes a las cuatro características

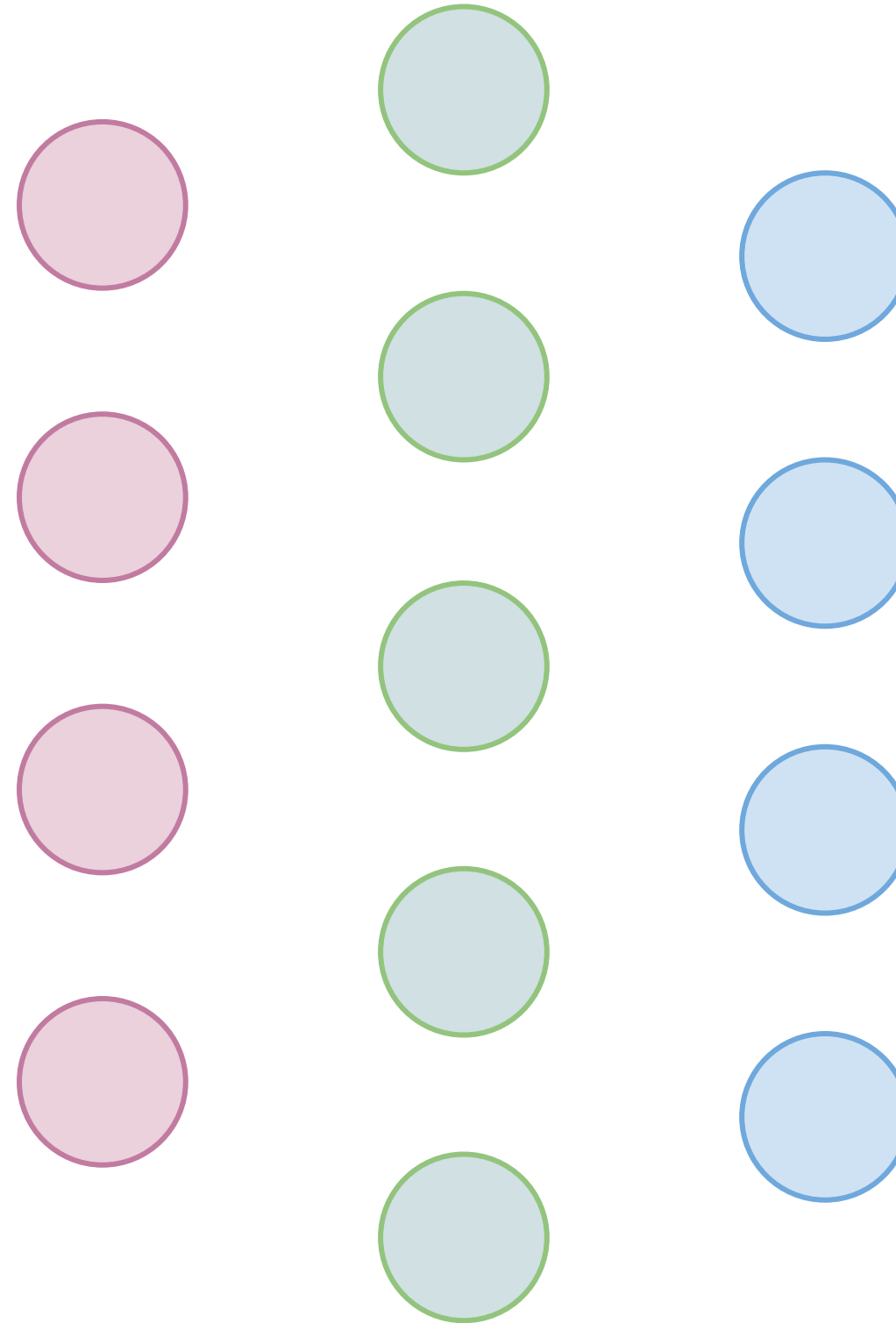




# Una red neuronal artificial



	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

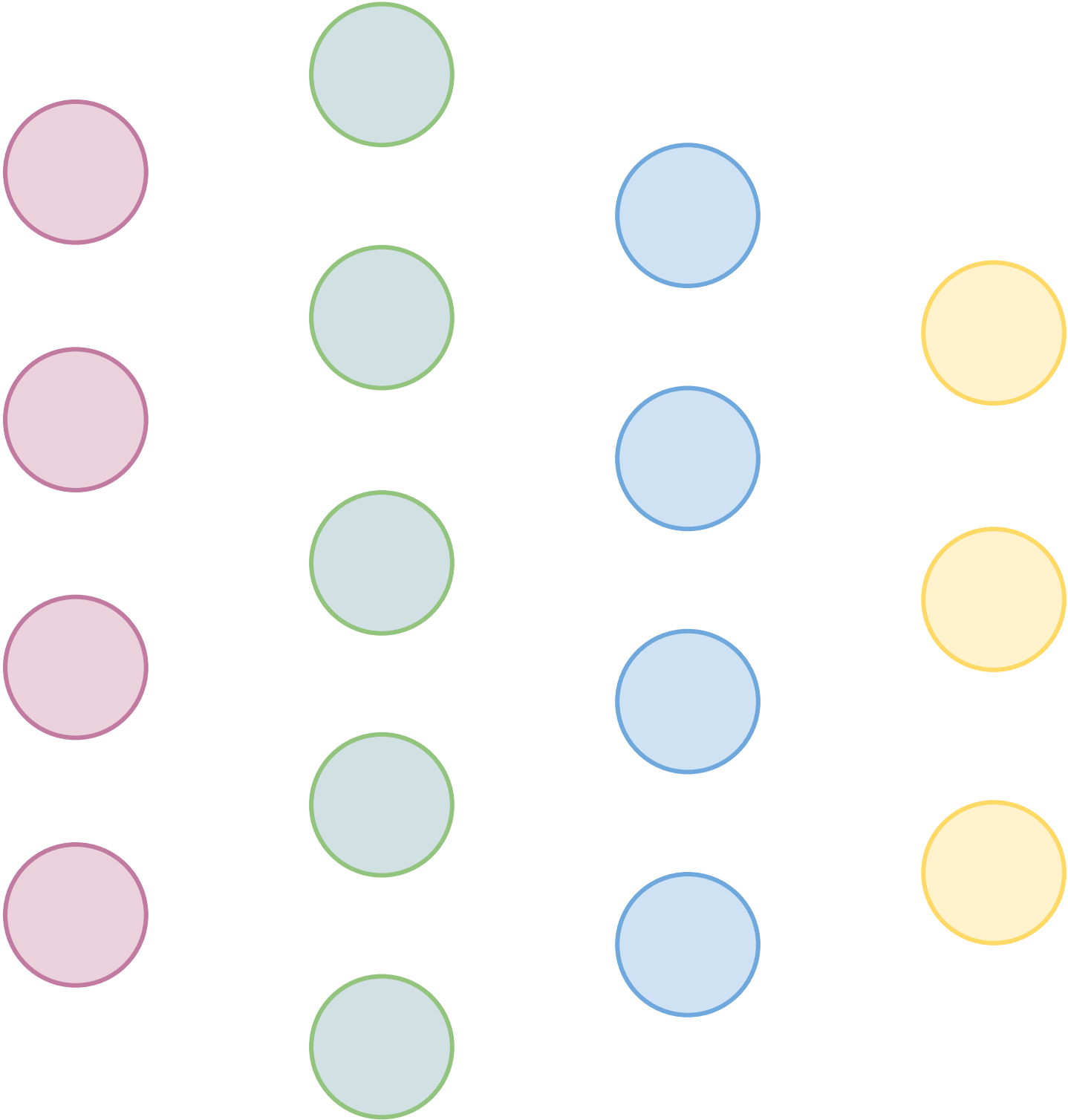


**Capas ocultas:** El número y tamaño de las capas ocultas puede variar en función de la complejidad del modelo

# An artificial neural network



	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa



**Capa de salida:** Tres, correspondientes a las tres clases de flores de Iris





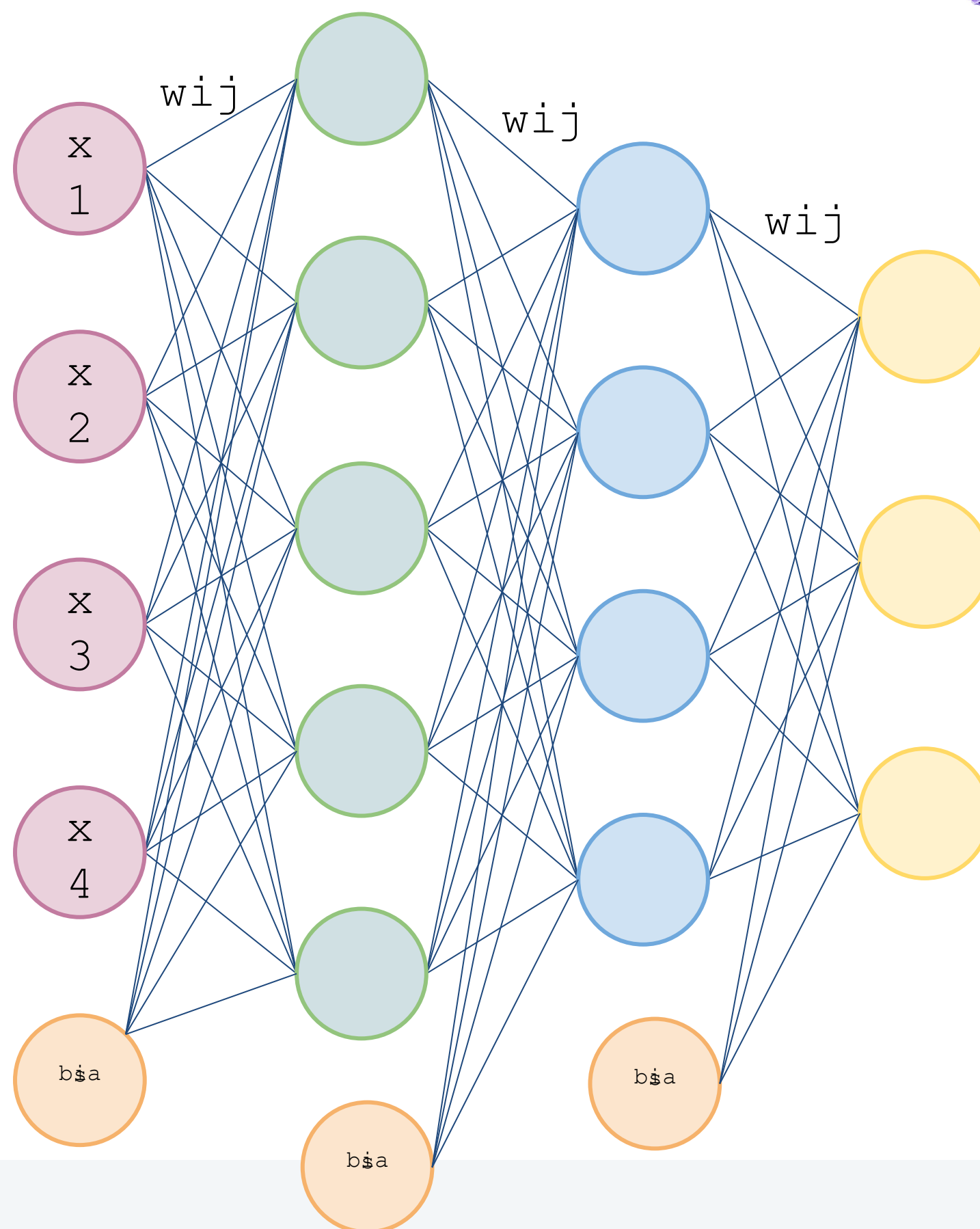
# Una red neuronal artificial



Paso 1:

## Inicialización

Los pesos y sesgos de la red suelen inicializarse, a menudo con pequeños valores aleatorios



	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

**Sesgo:** Cuando se ilustra una red neuronal, el sesgo se representa a menudo como un nodo adicional en cada capa, excepto en la capa de salida. En PyTorch, cuando usas `nn.Linear`, el sesgo se añade automáticamente por ti.

# An artificial neural network

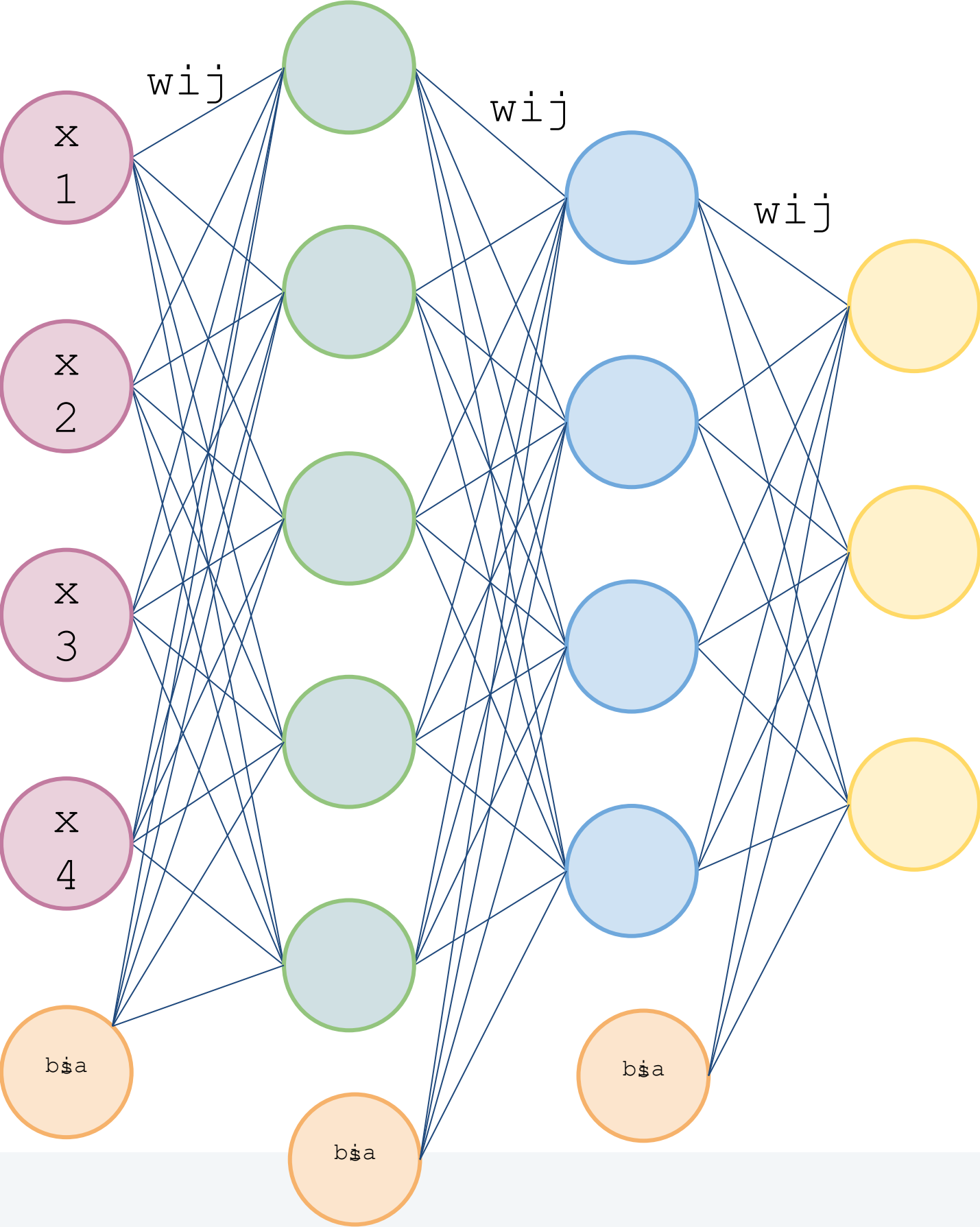


	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Paso 2:

Input

Ingresamos los  
datos



# An artificial neural network

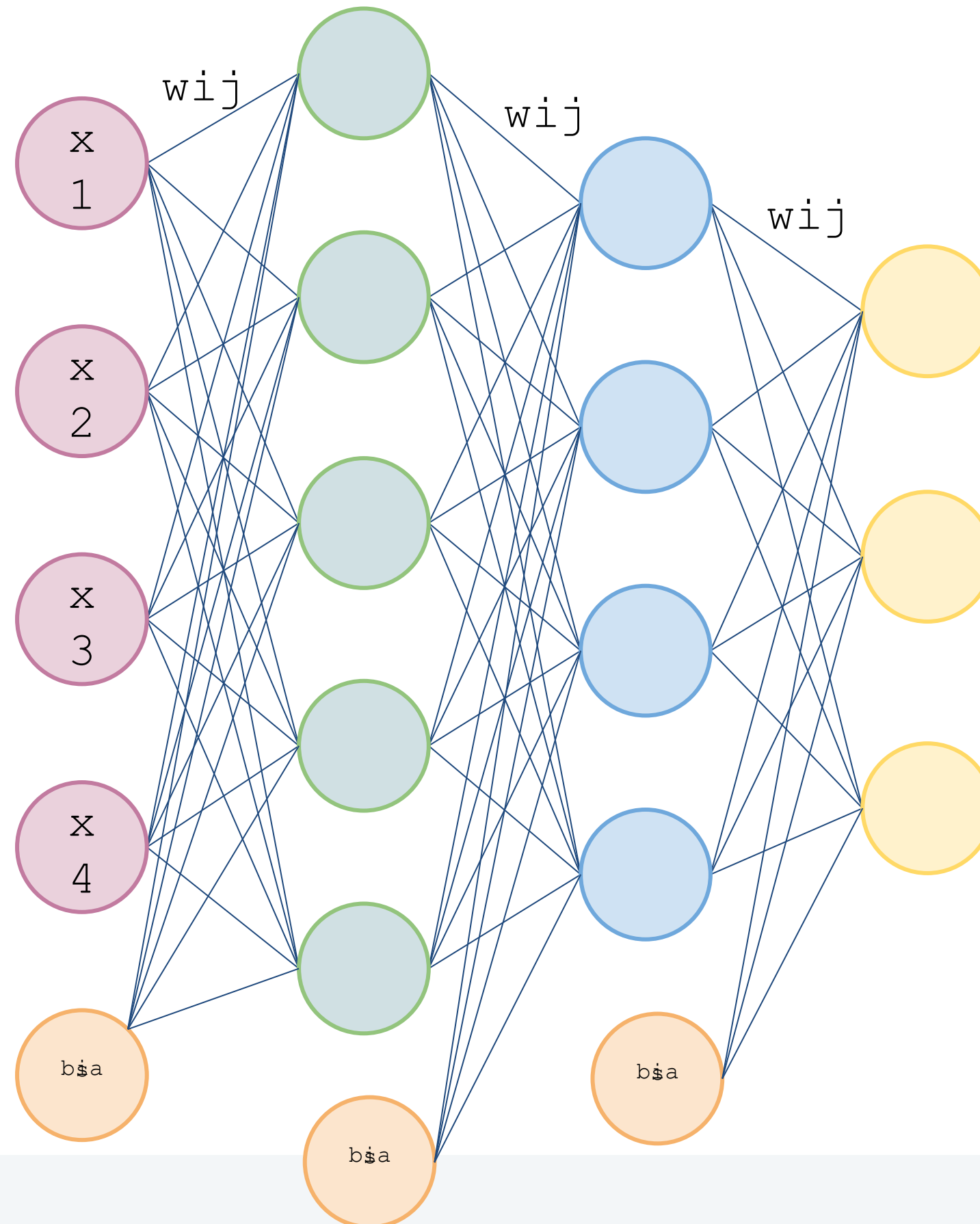


Propagación hacia delante →

Paso 3:

## Propagación hacia delante

- Partiendo de la capa de entrada, cada neurona calcula una suma ponderada de las entradas.
- Esta suma se pasa a través de una función de activación para producir la salida de la neurona.
- Esta salida se convierte en la entrada de la siguiente capa, y el proceso se repite en cada una de las capas siguientes de la red.



	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

# An artificial neural network

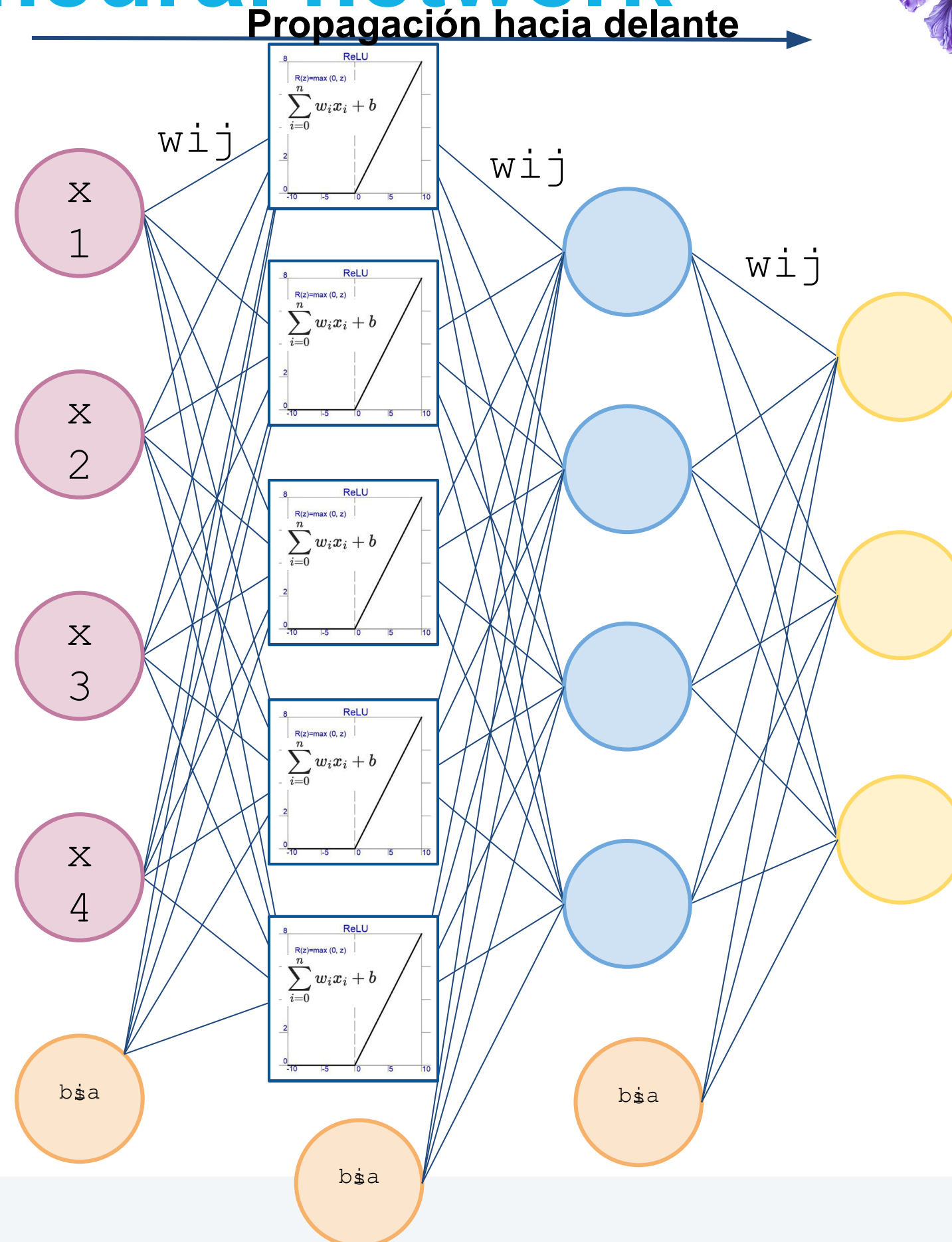


	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

## Paso 3:

### Propagación hacia delante

- Partiendo de la capa de entrada, cada neurona calcula una suma ponderada de las entradas.
- Esta suma se pasa a través de una función de activación para producir la salida de la neurona.
- Esta salida se convierte en la entrada de la siguiente capa, y el proceso se repite en cada una de las capas siguientes de la red.





# An artificial neural network

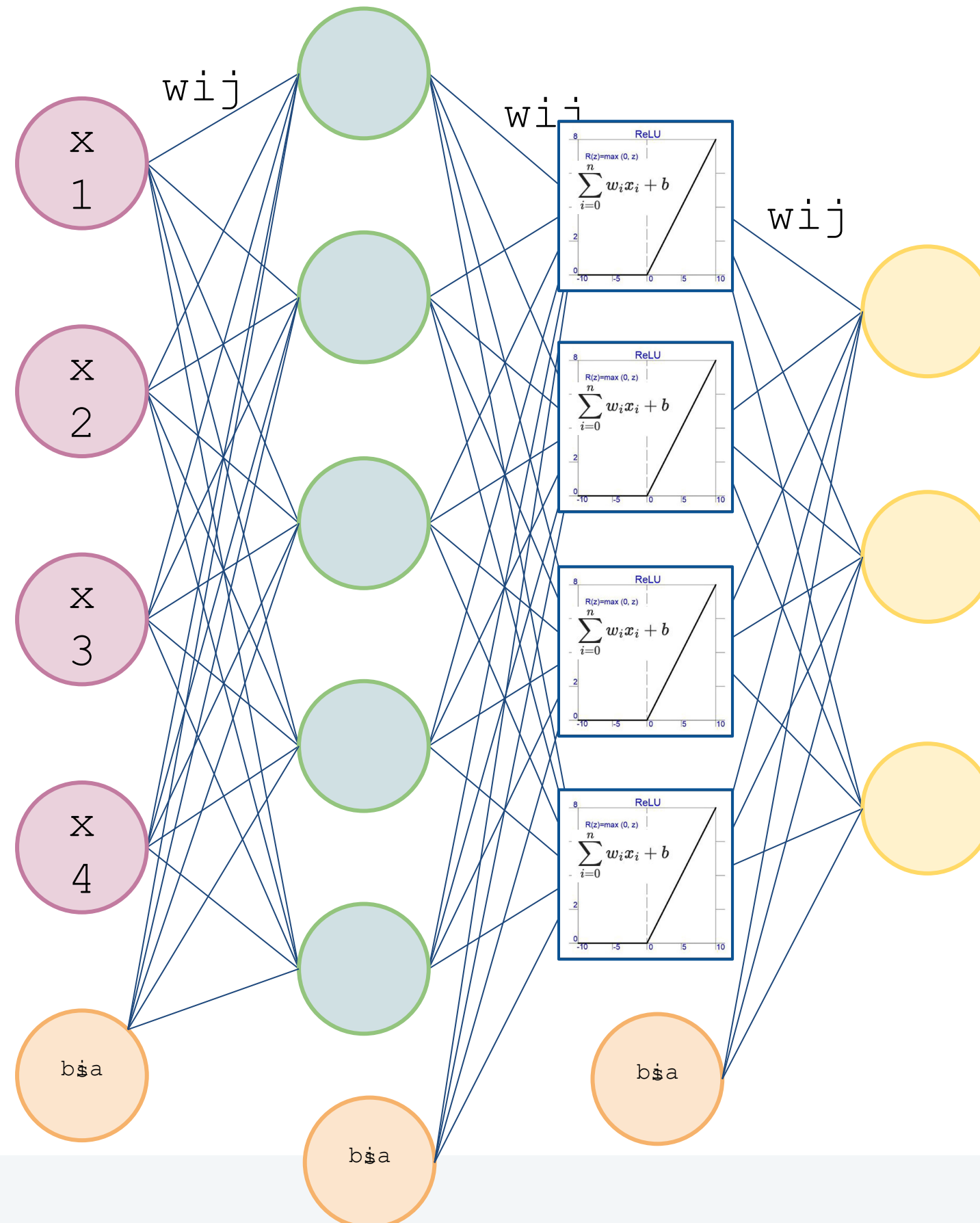


	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

## Paso 3:

### Propagación hacia delante

- Partiendo de la capa de entrada, cada neurona calcula una suma ponderada de las entradas.
- Esta suma se pasa a través de una función de activación para producir la salida de la neurona.
- Esta salida se convierte en la entrada de la siguiente capa, y el proceso se repite en cada una de las capas siguientes de la red.



# An artificial neural network

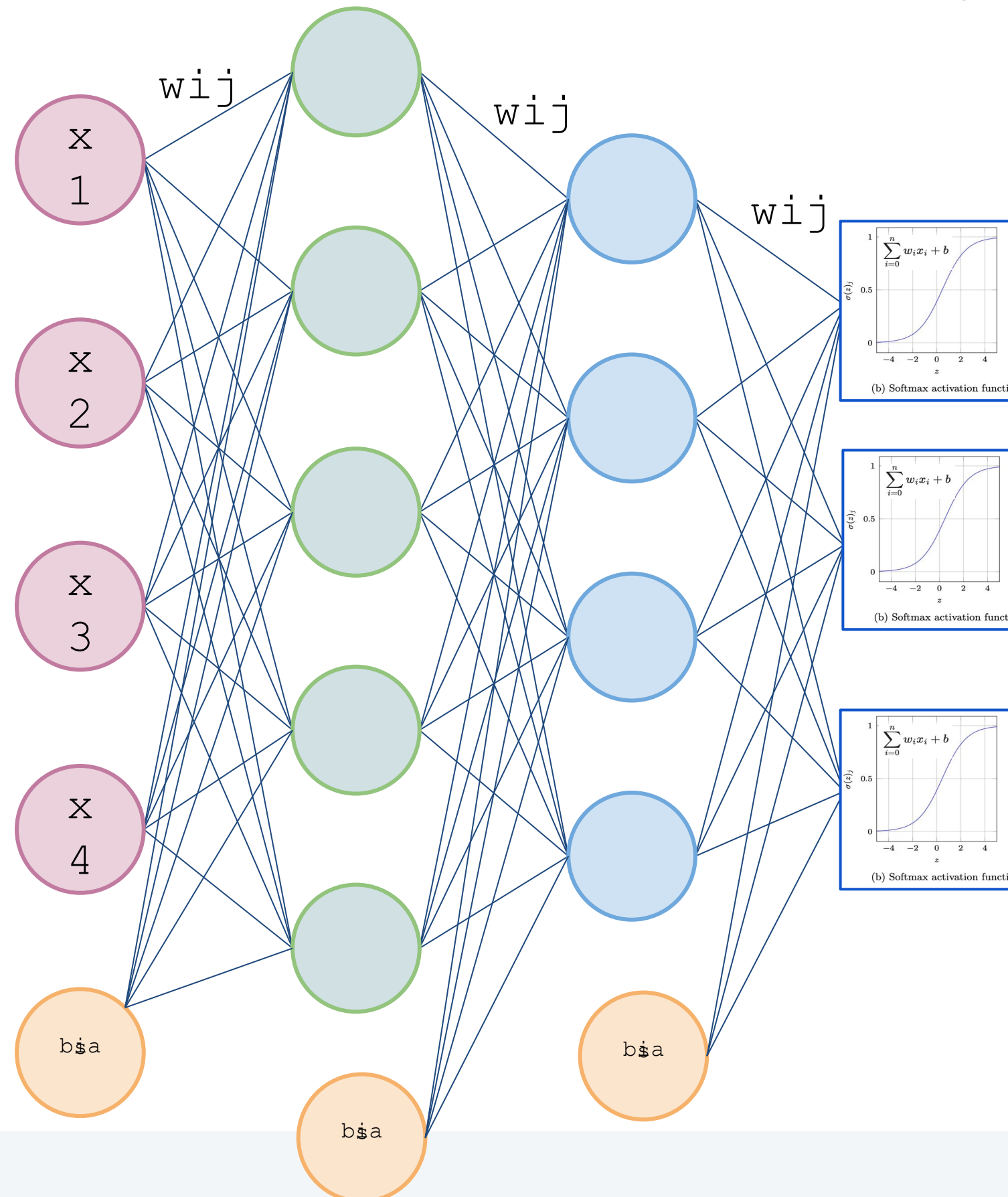


Propagación hacia delante

Paso 3:

## Propagación hacia delante

- Partiendo de la capa de entrada, cada neurona calcula una suma ponderada de las entradas.
- Esta suma se pasa a través de una función de activación para producir la salida de la neurona.
- Esta salida se convierte en la entrada de la siguiente capa, y el proceso se repite en cada una de las capas siguientes de la red.



	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

# An artificial neural network

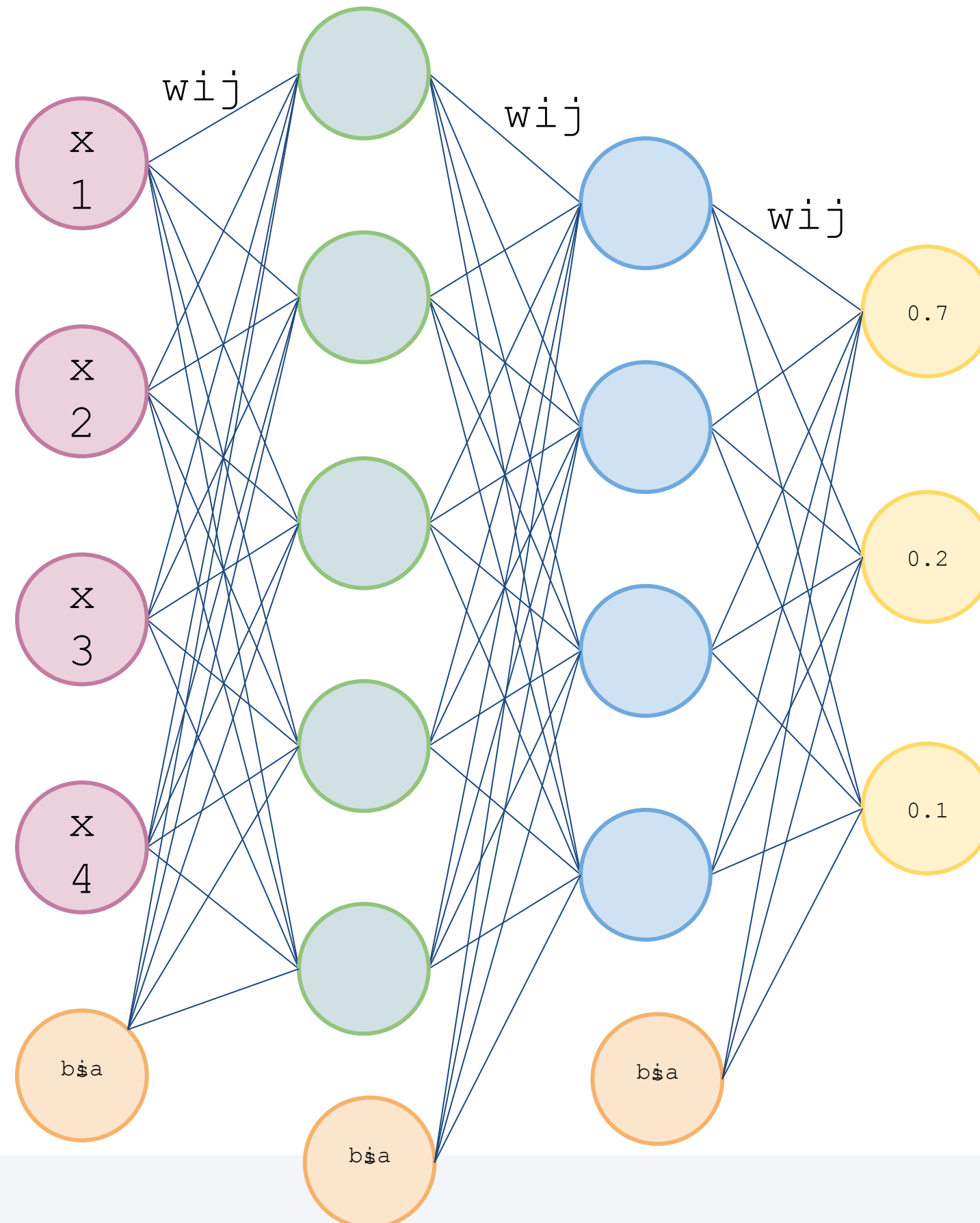


Propagación hacia delante →

Paso 3:

## Propagación hacia delante

- Partiendo de la capa de entrada, cada neurona calcula una suma ponderada de las entradas.
- Esta suma se pasa a través de una función de activación para producir la salida de la neurona.
- Esta salida se convierte en la entrada de la siguiente capa, y el proceso se repite en cada una de las capas siguientes de la red.



	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa



# An artificial neural network

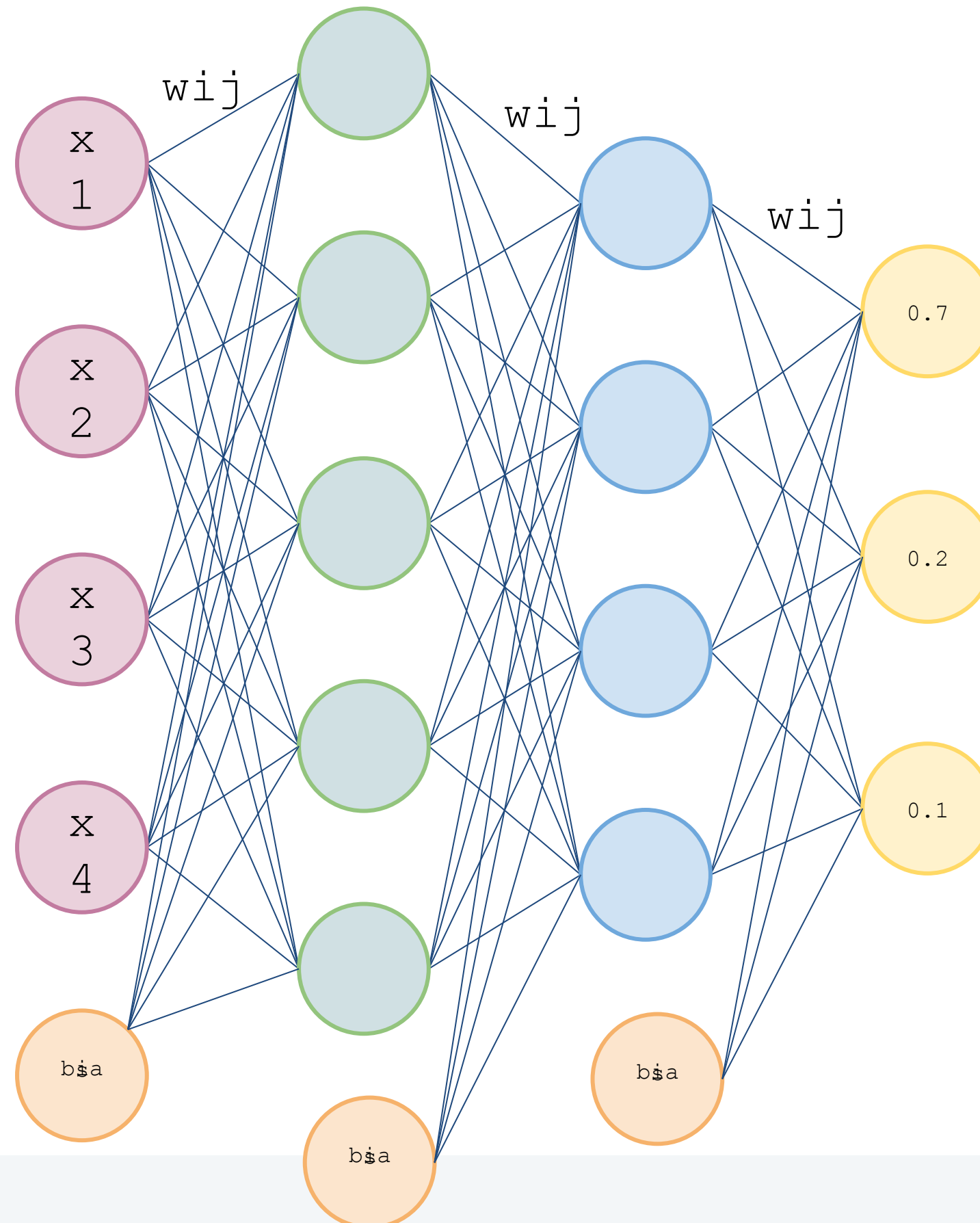


	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

## Paso 4:

### Calcular el error

- Una vez obtenida la predicción de la red (salida de la última capa), puede calcular la pérdida (o el error)
- La función de pérdida mide la diferencia entre la predicción y el objetivo.
- Regresión:
  - MSE
- Clasificación:
  - Entropía cruzada



$$L_{CE} = - \sum_{i=1}^n t_i \log(p_i), \text{ for } n \text{ classes,}$$

where  $t_i$  is the truth label and  $p_i$  is the Softmax probability for the  $i^{th}$  class.

# An artificial neural network

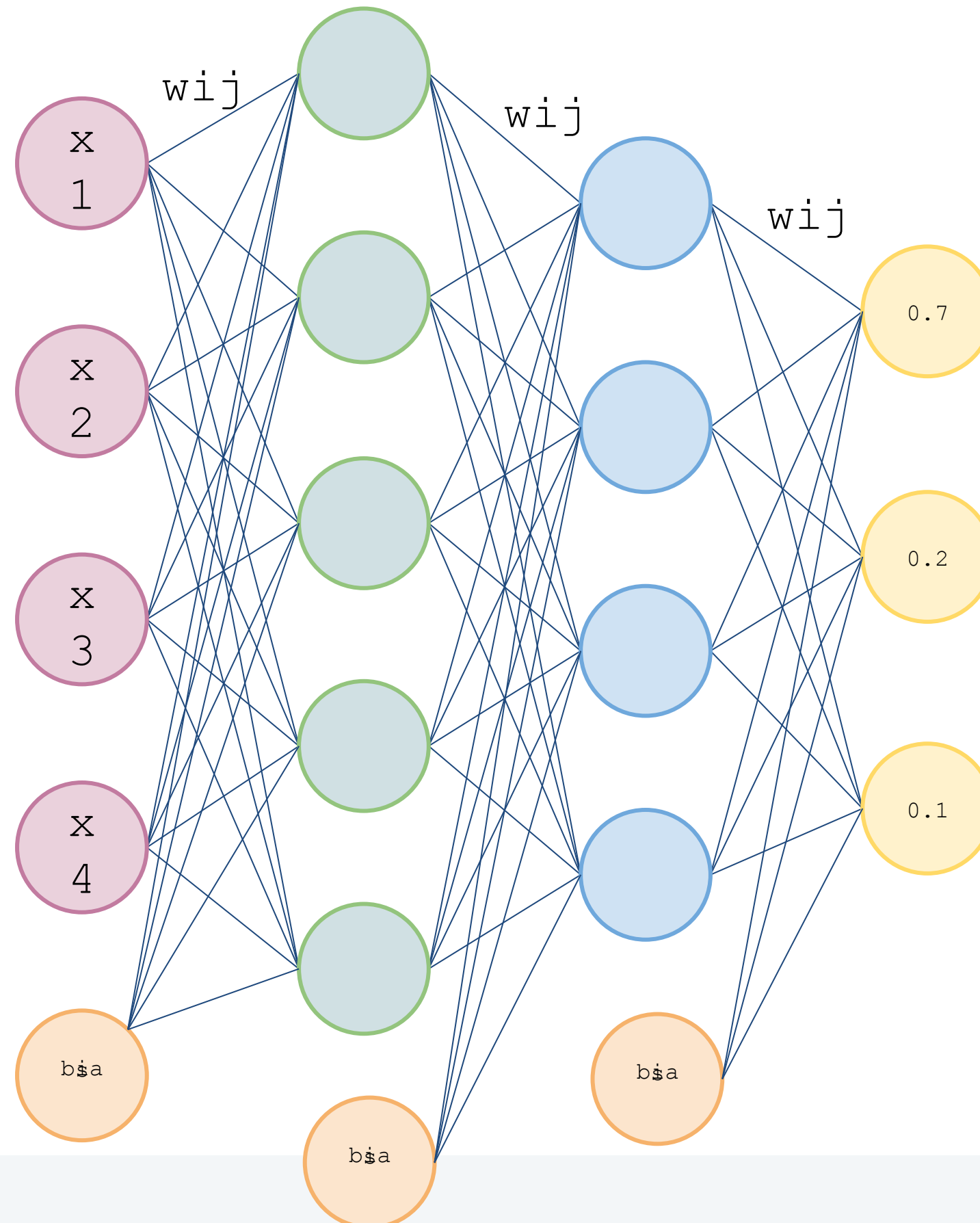


	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

## Paso 4:

### Calcular el error

- Una vez obtenida la predicción de la red (salida de la última capa), puede calcular la pérdida (o el error)
- La función de pérdida mide la diferencia entre la predicción y el objetivo.
- Regresión:
  - MSE
- Clasificación:
  - Entropía cruzada



$$L_{CE} = - \sum_{i=1}^n t_i \log(p_i), \text{ for } n \text{ classes,}$$

where  $t_i$  is the truth label and  $p_i$  is the Softmax probability for the  $i^{th}$  class.



**Objetivo de una red neuronal**  
Encontrar pesos y sesgos que minimicen la función de pérdida

# An artificial neural network

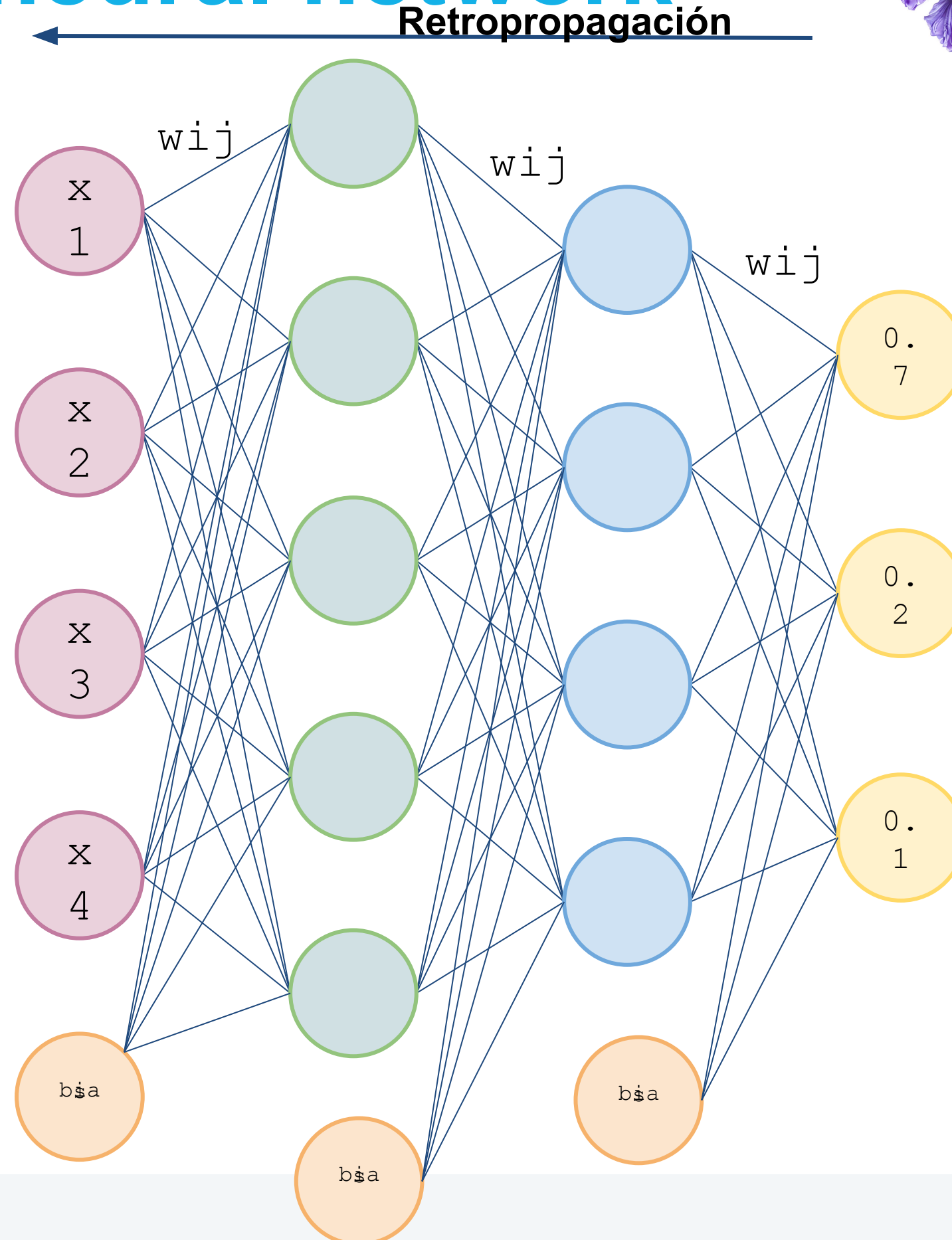


	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

## Paso 5:

### Retropropagación

- Calcular el gradiente de la función de pérdida con respecto a cada peso y sesgo de la red, es decir, la contribución de cada peso y sesgo al error.
- El cálculo comienza en la capa de salida y se desplaza hacia atrás por la red (utilizando la regla de la cadena para obtener el gradiente de cada parámetro).
- Este proceso nos da una dirección en la que ajustar cada peso y sesgo para minimizar las pérdidas.





# An artificial neural network

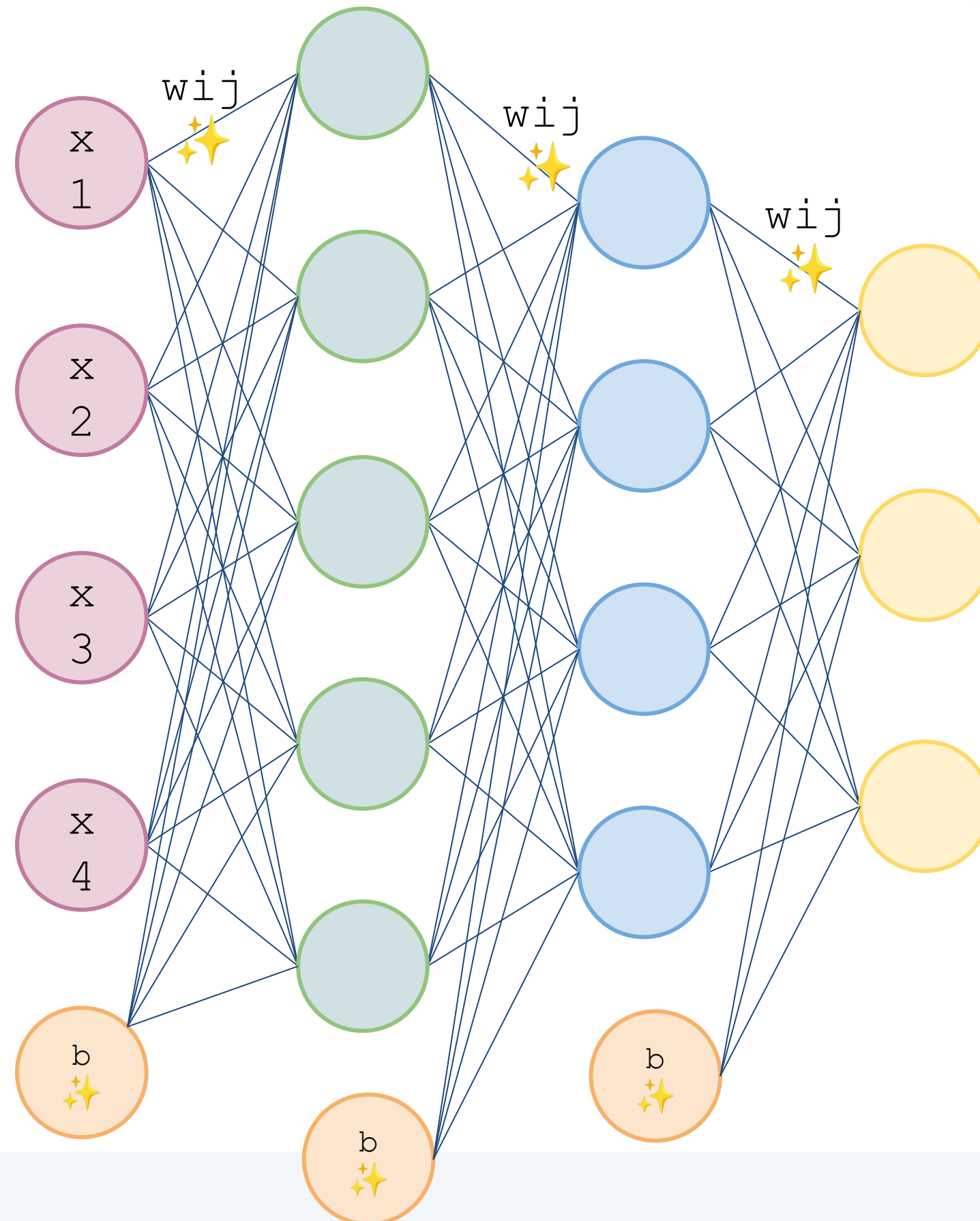


	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Paso 6:

## Actualizar pesos y sesgos

- Utilizando el descenso de gradiente o variantes (descenso de gradiente estocástico, Adam, etc.) actualice el valor de los pesos y sesgos



# An artificial neural network

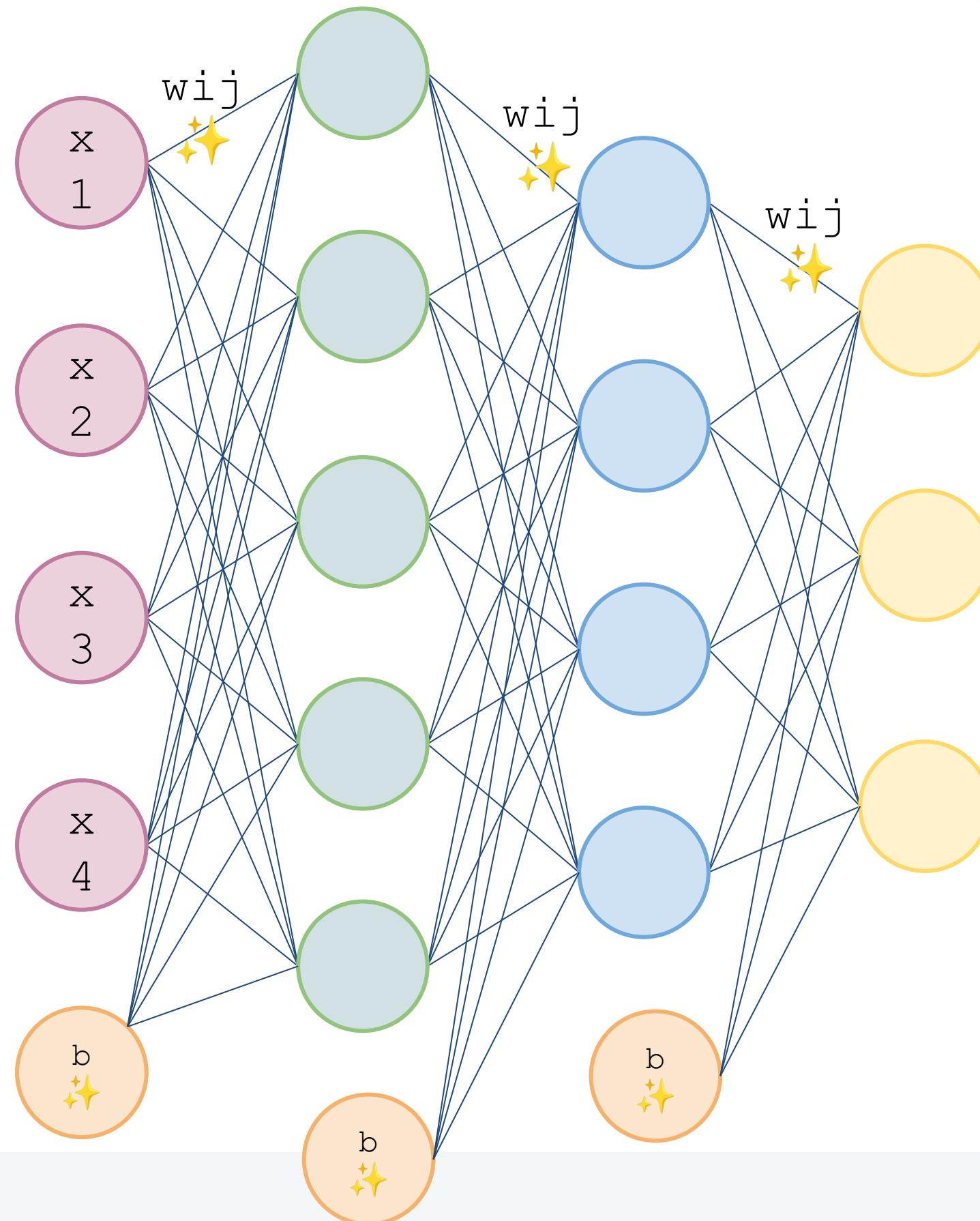


	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Paso 7:

## Iteración

- Repita los pasos 3-6 (propagación hacia delante, cálculo de la pérdida, retropropagación, actualización de pesos y sesgos) hasta que la pérdida en la validación deje de mejorar/aumentar.



## Vocabulario

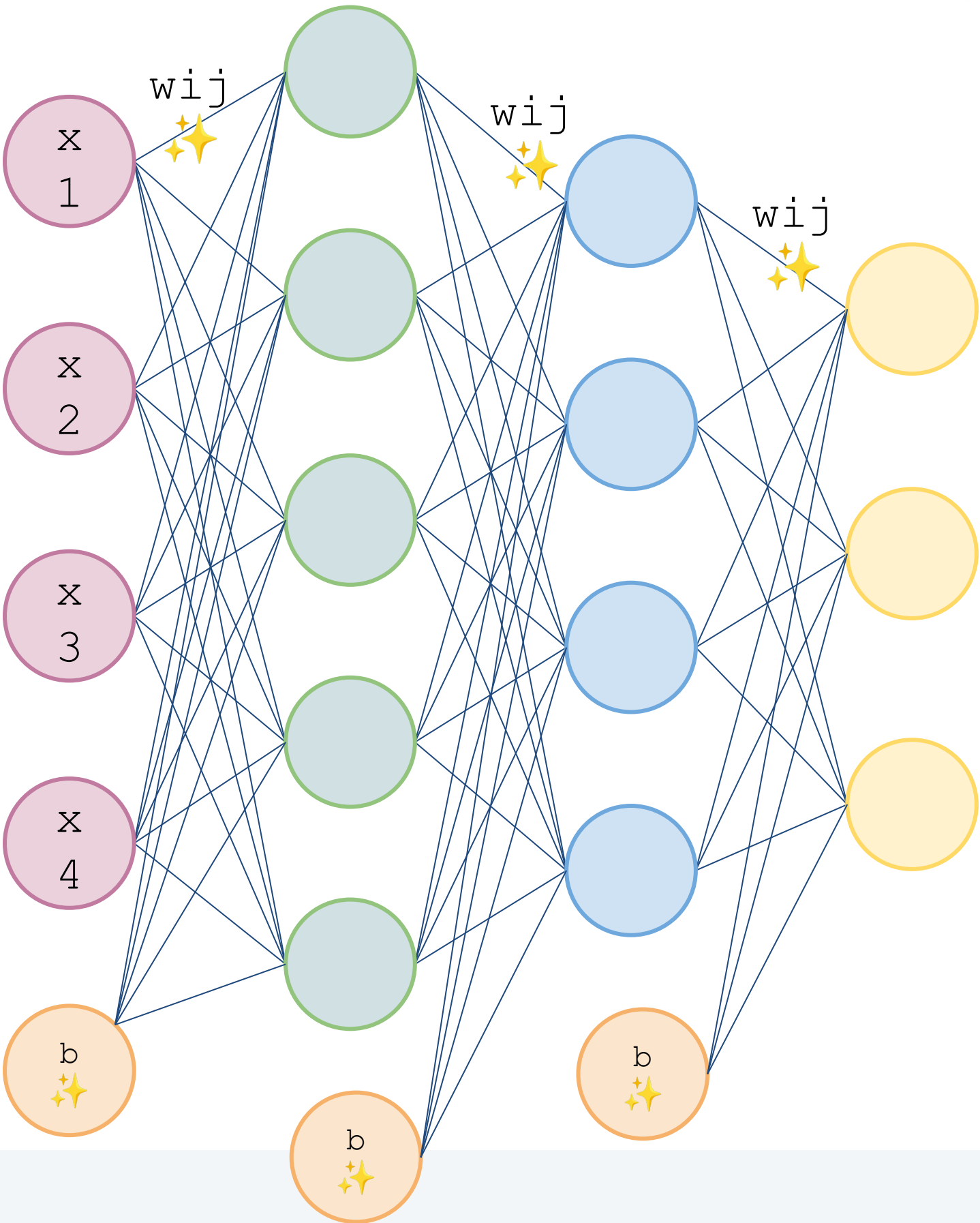
Una **época** se refiere a una pasada completa hacia delante y hacia atrás de todos los ejemplos de entrenamiento

# An artificial neural network



	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

- Paso 8:
- Evaluación
- Paso 9:
- Usar el modelo





# An artificial neural network

Código:

- <https://drive.google.com/file/d/1pyl30rnZp1ITOBSU5TkbrunhB39jVnEK/view?usp=sharing>

# Recursos adicionales

- [Neural Networks by StatQuest](#)
- [PyTorch documentation](#)

# Tipos de redes neuronales

- **Redes neuronales totalmente conectadas (FCNN)**

También conocidas como perceptrones multicapa (MLP), son la forma más sencilla de redes neuronales, en las que cada neurona de una capa está conectada a todas las neuronas de la capa siguiente.

- **Redes neuronales convolucionales (CNN)**

Son muy eficaces para procesar datos con una topología cuadriculada, como las imágenes. Las CNN han revolucionado el campo de la visión por ordenador, aportando mejoras sustanciales en el reconocimiento de imágenes, la clasificación de imágenes, la detección de objetos y muchas otras áreas.

- **Redes neuronales recurrentes (RNN)**

Diseñadas para manejar datos secuenciales como series temporales, voz o texto. Pueden mantener la información de estado devolviendo la salida de una neurona a la red como entrada para el siguiente paso, «recordando» esencialmente la información anterior.

- Las variantes incluyen: La memoria a largo plazo (LSTM) y las unidades recurrentes controladas (GRU), diseñadas para captar las dependencias a largo plazo y resolver el problema de gradiente de fuga común en las RNN básicas.



# Tipos de redes neuronales

- **Autocodificadores (AE)**

Utilizados para tareas de aprendizaje no supervisado, principalmente reducción de la dimensionalidad y aprendizaje de características.

- **Redes generativas adversariales (GAN)**

Consisten en dos redes, una generadora y otra discriminadora, que se entrenan simultáneamente en un marco de juego de suma cero. El generador aprende a crear datos similares a los de entrenamiento, y el discriminador aprende a diferenciar entre los datos generados y los reales. Se utilizan mucho para generar imágenes realistas, transferir estilos, etc.

- **Redes de transformadores**

Presentados en el artículo «Attention is All You Need», los transformadores están diseñados para manejar datos secuenciales sin necesidad de recurrencia. Se basan en un mecanismo llamado «atención» para sopesar la influencia de las distintas partes de los datos de entrada. Fundamento de modelos como BERT, GPT, etc.

- **y más...**

# Código - RNN

Generador de texto

<https://drive.google.com/file/d/15fs6JYtA-WvHAV4k-e65AAvfsr6v-nPL/view?usp=sharing>

# Código - CNN

- Introducción a la visión por computador  
[https://drive.google.com/file/d/1OjGTc\\_qJu4TKMfernC72pE14FAzr0F6p/view?usp=sharing](https://drive.google.com/file/d/1OjGTc_qJu4TKMfernC72pE14FAzr0F6p/view?usp=sharing)
- LeNet5  
<https://drive.google.com/file/d/1Rnc9uNCxLrupZB52-aBgMxO9qIFkwsr2/view?usp=sharing>
- ResNet preentrenada (aprendizaje por transferencia)  
<https://drive.google.com/file/d/1c2v3zScmOCEswbHFWOoPJa3aXI29gUrM/view?usp=sharing>
- GradCAM  
<https://drive.google.com/file/d/1nHDSDtcDodCJ6IN-BHMeRXaH-uC2GGPf/view?usp=sharing>
- ResNet con ajuste de hiperparámetros  
<https://drive.google.com/file/d/1leJH1xNlfRsuNnYuPsFsiKqjMdo0Je1w/view?usp=sharing>



# Código - Hugging Face

<https://drive.google.com/file/d/1AxQ46lxCkhTzbdobGFpqtyImWurCR93n/view?usp=sharing>

# Entrega del proyecto final

## 1. Expectativas del código

Regresión/Clasificación

<https://colab.research.google.com/drive/1M0ed9qFcAJRJu5EAsXcllokvnqjDqtMp?usp=sharing>

Agrupamiento

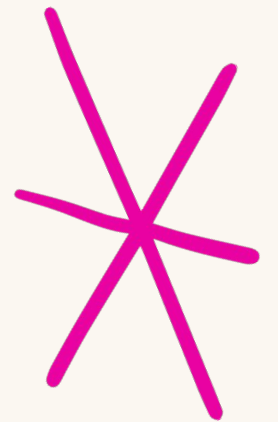
<https://colab.research.google.com/drive/1Yo963g3o-hu8NEty8xL0almXrZQsR4K5?usp=sharing>

## 1. Diapositivas

Debe contener lo mismo que en las expectativas del código, pero adicionalmente debe definir claramente su audiencia.

- El día de la presentación, ambas personas en el grupo deben hablar
- Sugerencia para los ingenieros: Subir el proyecto a GitHub

# ¡Gracias!



¿Dudas? Email de la profe:

[vroberta@unicomfauca.edu.co](mailto:vroberta@unicomfauca.edu.co)

Página web del curso con toda la info:

<https://github.com/vivianamarquez/unicomfauca-ai-2024>