



SISTEMAS INTELIGENTES

Lecture 09: Artificial Neural Networks

Dr. Edwin Valencia Castillo
Departamento de Sistemas
Facultad de Ingeniería
Universidad Nacional de Cajamarca
2024

Motivación



Podrías describir un perro?

Motivación



Podrías describir un gato?

Motivación



Puedes diferenciar un gato de un perro?

Como lo podría hacer la Inteligencia Artificial?

Motivación



**Seguiría la IA
reconociendo un perro?**

Motivación



**Seguiría la IA
reconociendo un perro?**

Motivación



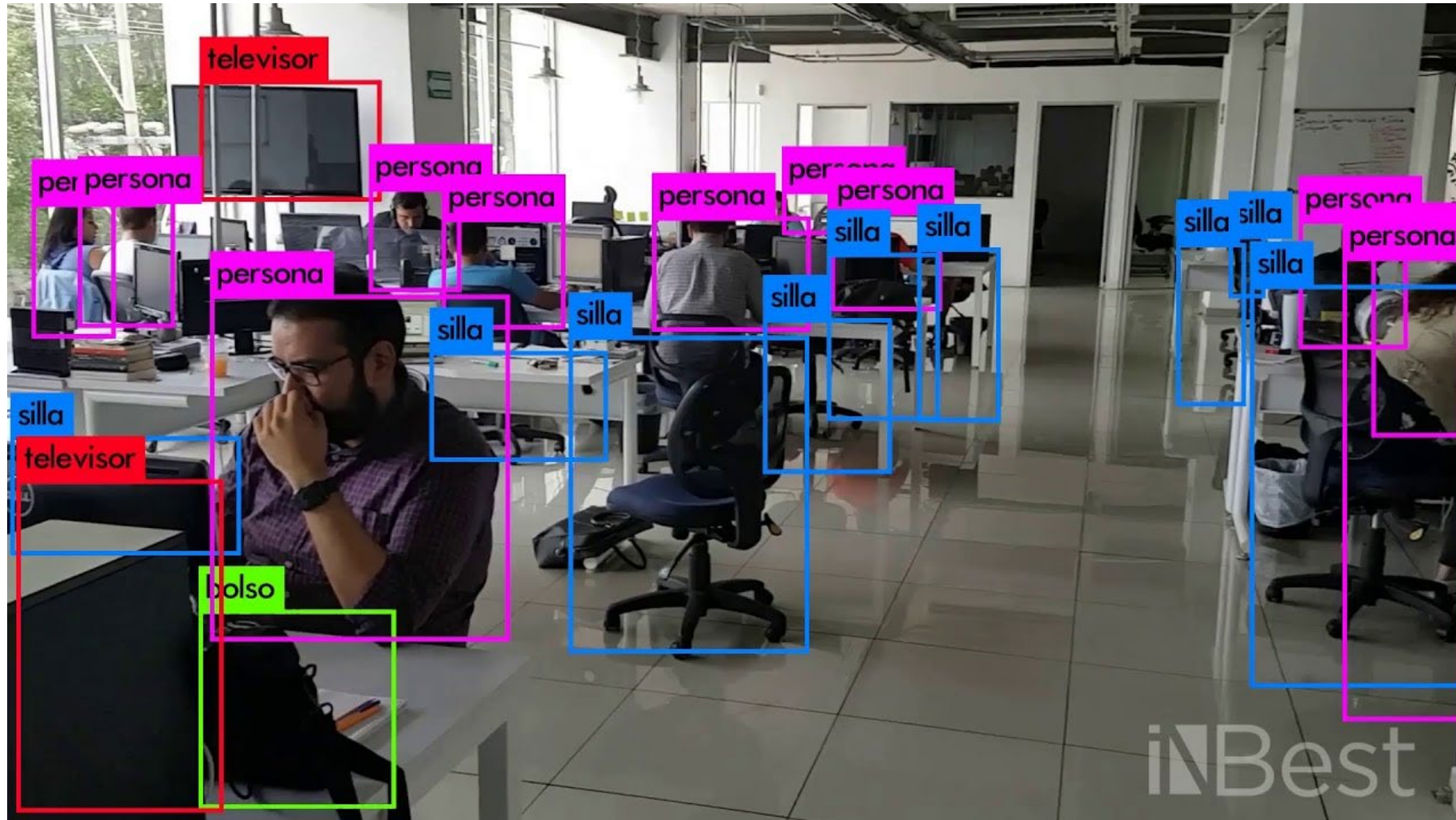
Seguiría la IA reconociendo todos como perros?

Aplicaciones

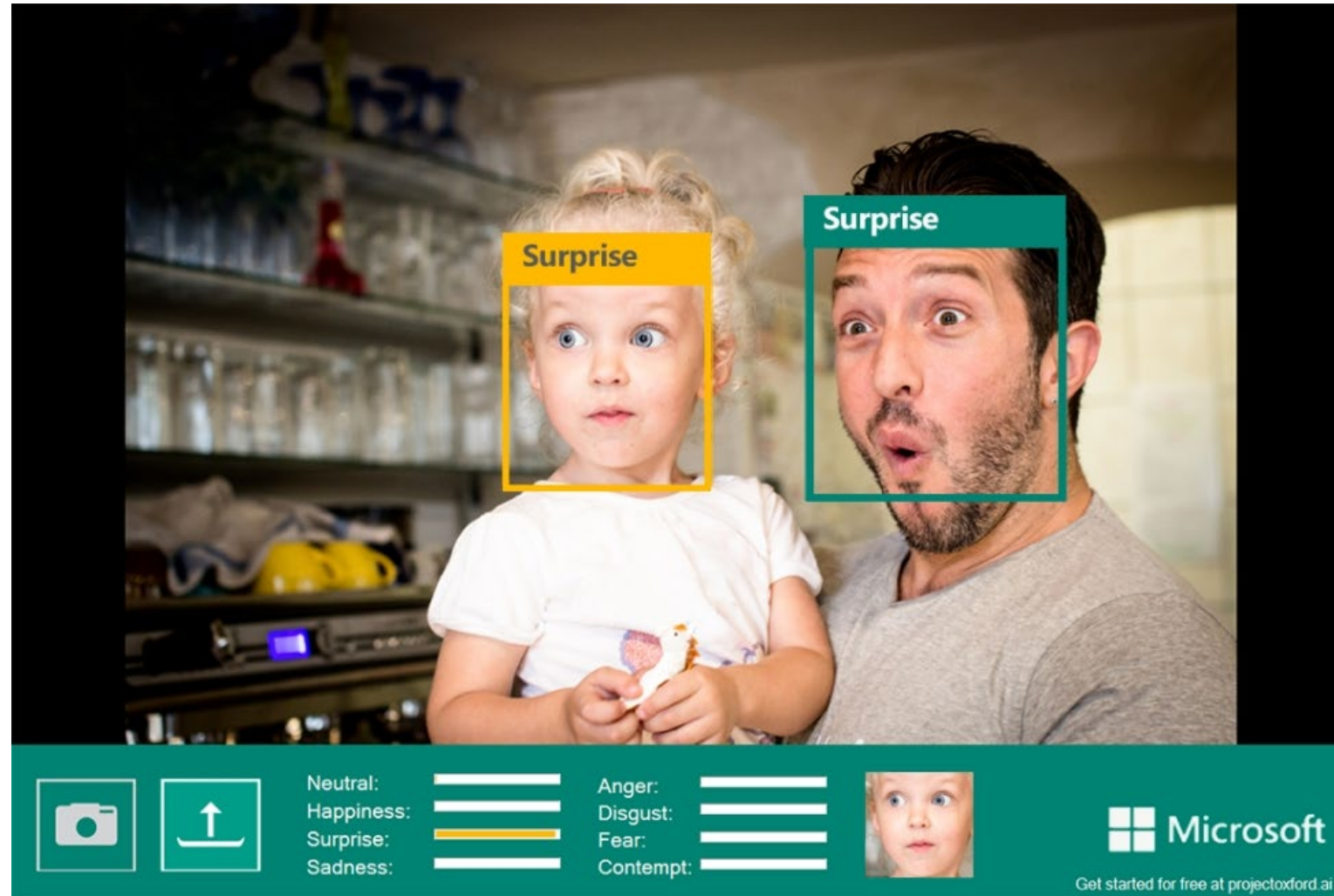


Fuente: [Link](#)

Reconocimiento de objetos



Detección de sentimientos



Segmentación semántica



Introducción

✓ **Que tienen en común todas las aplicaciones que hemos visto?**

La inteligencia artificial se fundamentaba principalmente en sistemas de reglas y conocimiento definidos por humanos.

Estos sistemas eran capaces de resolver problemas basados en reglas formales que suelen resultar difíciles para las personas.

Sin embargo, tareas sencillas e intuitivas para una persona (como reconocer un objeto) son de gran complejidad para un computador.

Introducción

La representación de los datos es crucial para el éxito de un algoritmo de machine learning

En muchas ocasiones, la definición de los datos o features que va a utilizar un sistema de machine learning es diseñada explícitamente por personas.

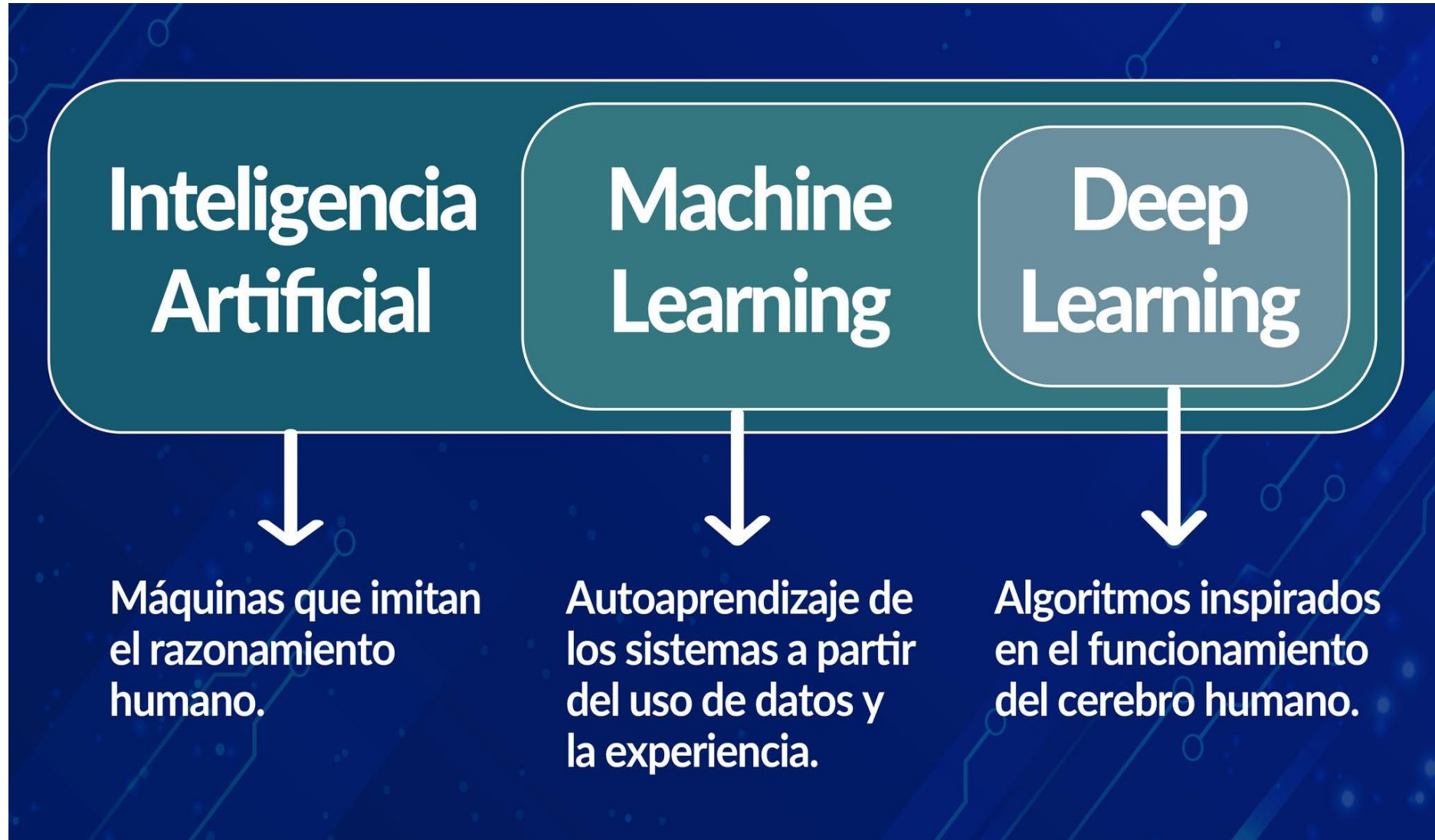
Idea del Deep Learning : obtener representaciones expresadas en términos de otras representaciones más sencillas, generadas directamente por los algoritmos.

Introducción

- ✓ Que tienen en común todas las aplicaciones que hemos visto?

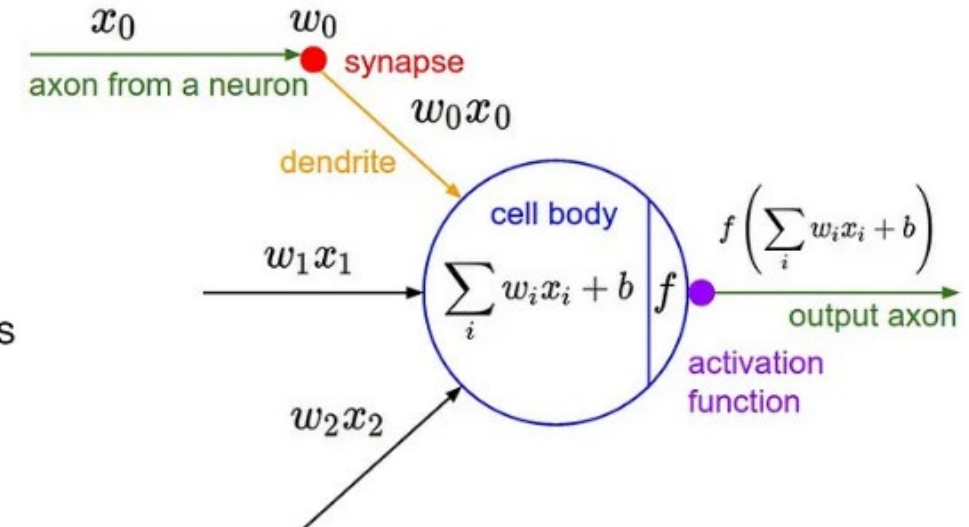
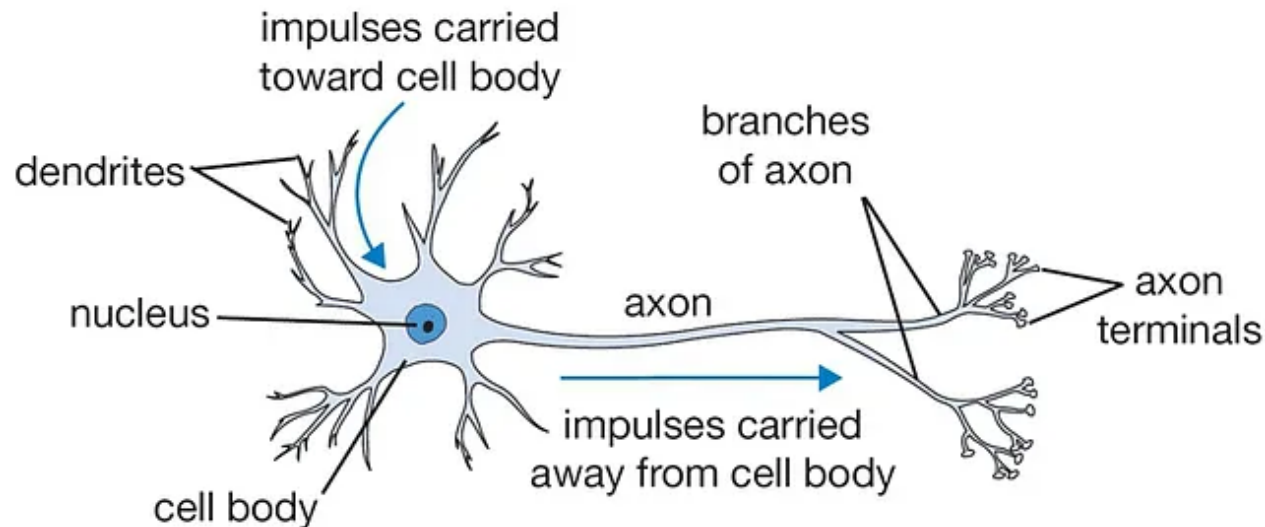
Sus features son muy difícil de describir por los humanos

Introducción



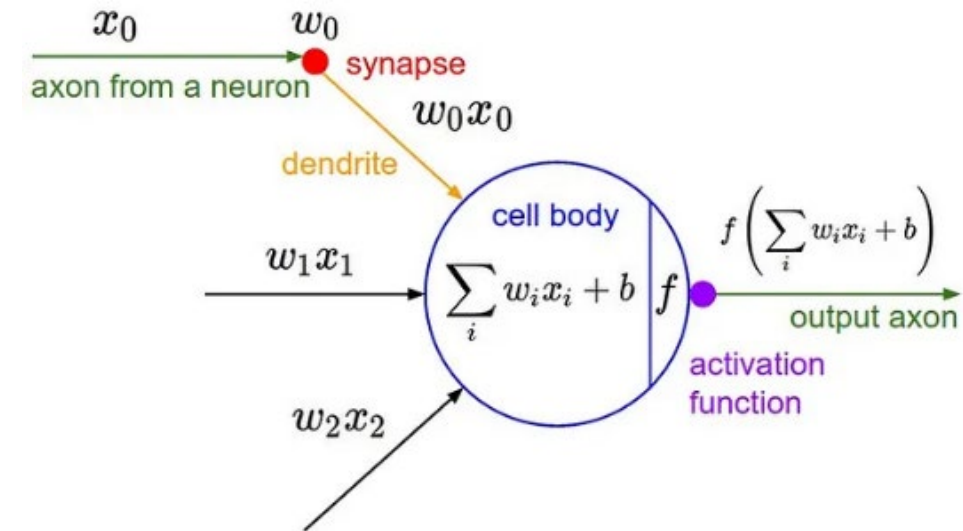
Neurona – Motivación Biológica

- ✓ La unidad básica de cálculo del cerebro es la neurona. En el sistema nervioso humano se encuentran aproximadamente 86 mil millones de neuronas, que están conectadas entre sí mediante aproximadamente 10^{14} — 10^{15} sinapsis.



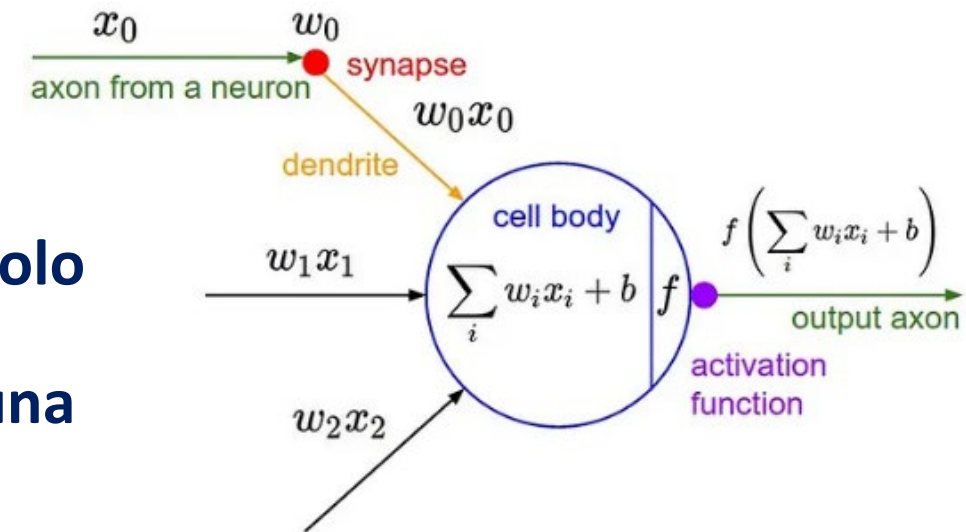
Neurona – Motivación Biológica

- ✓ La unidad básica de cálculo en una red neuronal es la neurona, a menudo denominada nodo o unidad. Recibe información de otros nodos o de una fuente externa y calcula una salida.
- ✓ Cada entrada tiene un peso asociado (w), que se asigna en función de su importancia relativa con respecto a otras entradas. El nodo aplica una función a la suma ponderada de sus entradas.



Neurona – Motivación Biológica

- ✓ La idea es que las fortalezas sinápticas (los pesos w) se pueden aprender y controlan la fuerza de la influencia y su dirección: excitatoria (peso positivo) o inhibitoria (peso negativo) de una neurona sobre otra.
- ✓ En el modelo básico, las dendritas llevan la señal al cuerpo celular donde todas se suman. Si la suma final está por encima de un cierto umbral, la neurona puede disparar, enviando un pico a lo largo de su axón.
- ✓ En el modelo computacional, asumimos que los tiempos precisos de los picos no importan, y que solo la frecuencia del disparo comunica información.
- ✓ Modelamos la tasa de disparo de la neurona con una función de activación (ej. función sigmoidea), que representa la frecuencia de los picos a lo largo del axón.

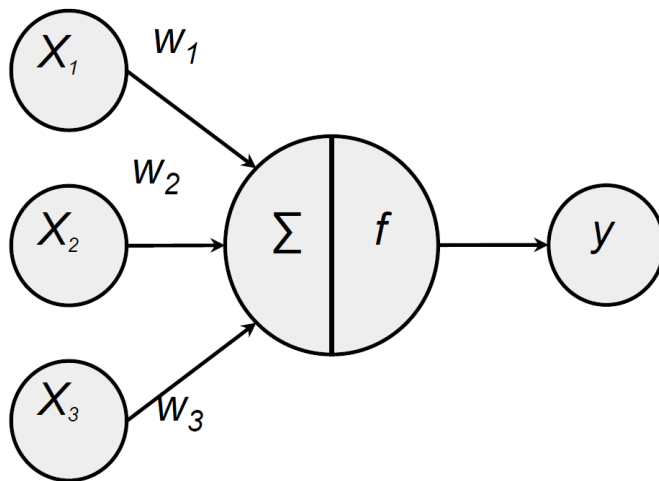


Neurona – Diferencias entre una neurona biológica y una neurona artificial

Neurona Biológica	Neurona Artificial
Señales que llegan a la sinapsis	Entradas a la neurona
Carácter excitador o inhibidor de la sinapsis de entrada	Pesos de entrada
Estimulo total de la neurona	Sumatoria de pesos por entradas
Activación o no de la neurona	Función de activación
Respuesta de la neurona	Función de salida

Perceptrón

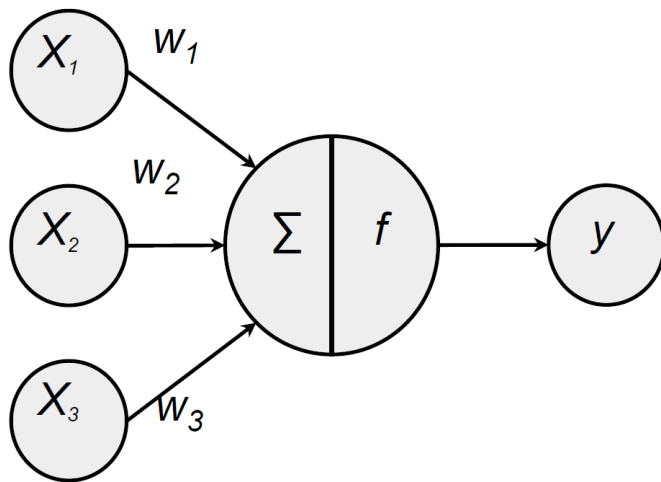
- ✓ El perceptrón es una de las arquitecturas ANN (Artificial Neural Network) más simples, inventada en 1957 por Frank Rosenblatt.
- ✓ ¿Cómo funcionan los perceptrones?
 - Un perceptrón recibe varias entradas binarias, x_1, x_2, \dots , y produce una única salida binaria:



En el ejemplo mostrado, el perceptrón tiene tres entradas, x_1, x_2, x_3 . En general, podría tener más o menos entradas. Rosenblatt propuso una regla simple para calcular la salida. Introdujo pesos, w_1, w_2, \dots , números reales que expresan la importancia de las respectivas entradas para la salida.

Perceptrón

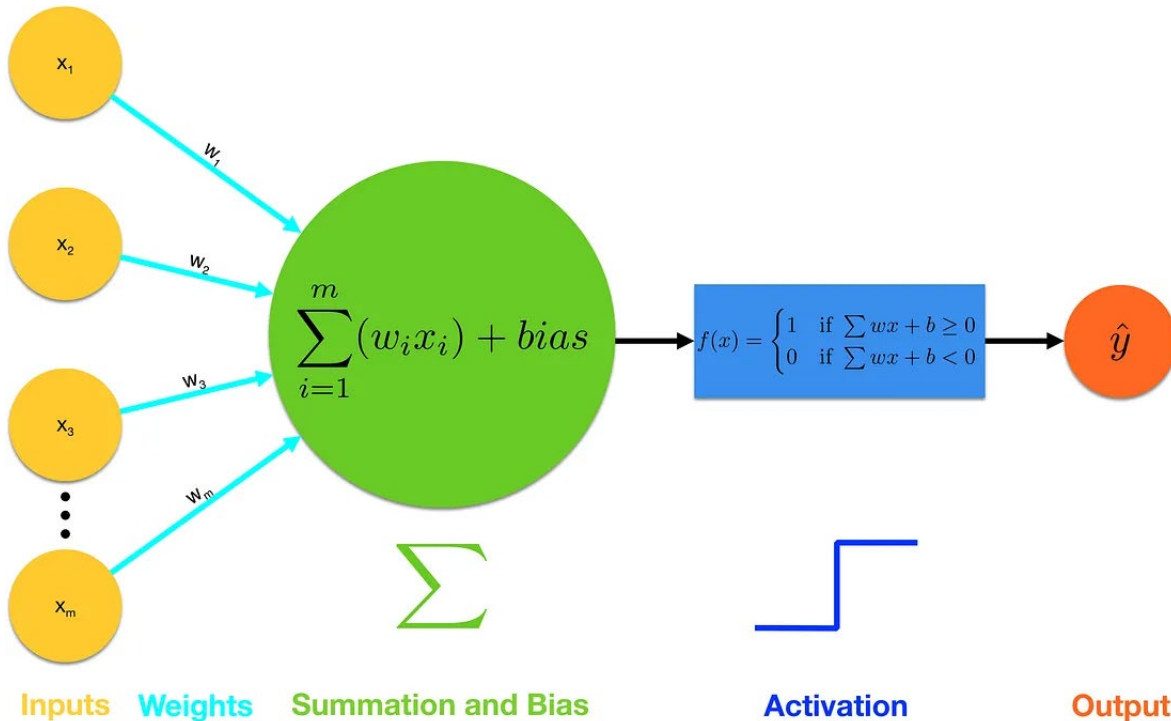
- ✓ La salida de la neurona (y) = 0 o 1, se determina por si la suma ponderada $\sum_j w_j x_j$ es menor o mayor que un valor umbral (threshold). Al igual que los pesos, el umbral es un número real que es un parámetro de la neurona. Para decirlo en términos algebraicos más precisos:



$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases}$$

Perceptrón

✓ En resumen:



1. **Inputs:** Las entradas se introducen en el perceptrón.
2. **Weights:** Los pesos se multiplican por cada entrada
3. **Summation and Bias:** Se suma y luego se agrega el sesgo
4. **Activation:** Se aplica la función de activación como la sigmoidea, la tangente hiperbólica (\tanh), la rectificadora (relu) y más.
5. **Output:** La salida se activa como 1 o como 0. Tenga en cuenta que usamos \hat{y} para etiquetar la salida producida por nuestro modelo de perceptrón.

¡Eso es todo lo que hay que saber sobre cómo funciona un perceptrón!

Perceptrón – Ejemplo

- ✓ Una forma de pensar en el perceptrón es que es un dispositivo que toma decisiones sopesando evidencias.
- ✓ Veamos un ejemplo. No es un ejemplo muy realista, pero es fácil de entender. Supongamos que se acerca el fin de semana y han oído que habrá un festival del queso en la ciudad. Les gusta el queso y están tratando de decidir si irán o no al festival. Podrían tomar su decisión sopesando tres factores:
 1. ¿Hace buen tiempo?
 2. ¿Tu novio o novia quiere acompañarte?
 3. ¿El festival está cerca del transporte público? (No tienes auto).

Perceptrón – Ejemplo

- ✓ Supongamos que adoras el queso, tanto que estás feliz de ir al festival incluso si a tu novio o novia no le interesa y es difícil llegar al festival. Pero tal vez detestas el mal tiempo y no hay forma de que vayas al festival si el clima es malo. Puedes usar perceptrones para modelar este tipo de toma de decisiones.
- ✓ Una forma de hacerlo es elegir un peso $w_1 = 6$ para el clima, $w_2 = 2$ y $w_3 = 2$ para las demás condiciones. El valor mayor es w_1 indica que el clima es muy importante para usted, mucho más que si su novio o novia lo acompaña o la cercanía del transporte público. Por último, supongamos que elige un umbral de 5 para el perceptrón. Con estas opciones, el perceptrón implementa el modelo de toma de decisiones deseado, generando 1 siempre que el tiempo es bueno, y 0 si el tiempo es malo, no importa si tu novio o novia quiere ir o si hay transporte público cerca.

Perceptrón – Ejemplo

Entradas:

1. ¿Hace buen tiempo?
 - Representación: x_1
 - Valor: 1 (Sí) o 0 (No)
2. ¿Tu novio o novia quiere acompañarte?
 - Representación: x_2
 - Valor: 1 (Sí) o 0 (No)
3. ¿El festival está cerca del transporte público? (No tienes auto)
 - Representación: x_3
 - Valor: 1 (Sí) o 0 (No)

Salida:

- Decisión de ir al festival:
 - Representación: y
 - Valor: 1 (Sí) o 0 (No)

Parámetros del Modelo:

- Pesos:
 - $w_1 = 6$ (Clima)
 - $w_2 = 2$ (Pareja)
 - $w_3 = 2$ (Transporte)
- Umbral:
 - Umbral = 5

Función del Perceptrón:

El perceptrón tomará la decisión de ir al festival si la suma ponderada de las entradas es mayor o igual al umbral. Matemáticamente, se expresa como:

$$y = f(w_1x_1 + w_2x_2 + w_3x_3)$$

Donde $f(z) = 1$ si $z \geq 5$, de lo contrario $f(z) = 0$.

Perceptrón – Ejemplo

Tabla de Decisiones:

Caso	x_1 (Buen tiempo)	x_2 (Pareja acompaña)	x_3 (Cerca del transporte)	Suma ponderada $z = 6x_1 + 2x_2 + 2x_3$	Decisión y (Ir al festival)
1	1	1	1	$6(1) + 2(1) + 2(1) = 10$	1 (Sí)
2	1	0	0	$6(1) + 2(0) + 2(0) = 6$	1 (Sí)
3	0	1	1	$6(0) + 2(1) + 2(1) = 4$	0 (No)
4	0	0	1	$6(0) + 2(0) + 2(1) = 2$	0 (No)
5	0	0	0	$6(0) + 2(0) + 2(0) = 0$	0 (No)
6	1	1	0	$6(1) + 2(1) + 2(0) = 8$	1 (Sí)

Este modelo de perceptrón refleja que el clima es el factor decisivo más importante para ti. Si el clima es bueno, casi siempre decides ir al festival, independientemente de las otras condiciones. Pero si el clima es malo, decides no ir, sin importar si tu pareja quiere ir o si el festival está cerca del transporte.

Perceptrón – Ejemplo

- ✓ Al variar los pesos y el umbral, podemos obtener diferentes modelos de toma de decisiones. Por ejemplo, supongamos que elegimos un umbral de 3.

Función del Perceptrón con Umbral 3:

El perceptrón ahora decidirá ir al festival si la suma ponderada de las entradas es mayor o igual a 3.

$$y = f(w_1x_1 + w_2x_2 + w_3x_3)$$

Donde $f(z) = 1$ si $z \geq 3$, de lo contrario $f(z) = 0$.

Nueva Tabla de Decisiones:

Caso	x_1 (Buen tiempo)	x_2 (Pareja acompaña)	x_3 (Cerca del transporte)	Suma ponderada $z = 6x_1 + 2x_2 + 2x_3$	Decisión y (Ir al festival)
1	1	1	1	$6(1) + 2(1) + 2(1) = 10$	1 (Sí)
2	1	0	0	$6(1) + 2(0) + 2(0) = 6$	1 (Sí)
3	0	1	1	$6(0) + 2(1) + 2(1) = 4$	1 (Sí)
4	0	0	1	$6(0) + 2(0) + 2(1) = 2$	0 (No)
5	0	0	0	$6(0) + 2(0) + 2(0) = 0$	0 (No)
6	1	1	0	$6(1) + 2(1) + 2(0) = 8$	1 (Sí)

Con un umbral de 3, el perceptrón permite más flexibilidad en la decisión. Ahora, incluso si el clima es malo, podrías decidir ir al festival si las otras dos condiciones (pareja y transporte) son favorables, siempre que la suma ponderada alcance al menos 3. Este umbral reduce la importancia decisiva del clima, permitiendo que las otras condiciones influyan más en la decisión final.

Como aprende un perceptrón?

✓ Fórmula Básica del Perceptrón

➤ La salida del perceptrón (y_{salida}) se calcula como:

$$y_{\text{salida}} = f \left(\sum_{i=1}^n w_i \times x_i + b \right)$$

Donde:

- x_i son las entradas.
- w_i son los pesos asociados a cada entrada.
- b es el bias, un valor que se suma a la suma ponderada.
- f es la función de activación, que normalmente es una función escalón para un perceptrón simple.

➤ El perceptrón aprende ajustando los pesos y el bias para minimizar el error en sus predicciones. Este proceso se realiza a través de un ciclo de entrenamiento que incluye los siguientes pasos:

Como aprende un perceptrón?

- ✓ 1. Inicialización
 - Pesos y Bias: Inician con pequeños valores aleatorios. Por ejemplo, w_i y b podrían comenzar cerca de 0.
- ✓ 2. Cálculo de la Salida
 - Para una entrada $x=[x_1, x_2, \dots, x_n]$, el perceptrón calcula la salida y usando la fórmula mencionada anteriormente.
- ✓ 3. Comparación con la Salida Deseada
 - La salida calculada y_{salida} se compara con la salida esperada y_{deseada} .
- ✓ 4. Actualización de Pesos y Bias
 - Si hay un error (es decir, si y_{salida} es diferente de y_{deseada}), se ajustan los pesos y el bias. La regla de actualización para cada peso w_i y el bias b es:

$$w_i \leftarrow w_i + \eta \times (y_{\text{deseada}} - y_{\text{salida}}) \times x_i$$
$$b \leftarrow b + \eta \times (y_{\text{deseada}} - y_{\text{salida}})$$

Donde:
 η es la tasa de aprendizaje, un valor pequeño que determina cuánto se ajustan los pesos y el bias en cada iteración.
- ✓ 5. Repetición
 - Este proceso se repite para cada ejemplo de entrenamiento. Se realiza durante varias épocas (iteraciones sobre el conjunto de datos) hasta que el perceptrón clasifica correctamente las entradas (o hasta que se alcanza un criterio de parada).

Como aprende un perceptrón?

✓ Convergencia

- El perceptrón sigue ajustando sus pesos hasta que las salidas calculadas coincidan con las salidas deseadas para todos los ejemplos del conjunto de datos.
- Cuando esto sucede, se dice que el perceptrón ha "aprendido" el modelo, y los pesos se estabilizan.
- Ejemplo:

Supongamos que queremos entrenar un perceptrón para decidir si ir al festival del queso en función del clima, si tu pareja te acompaña, y la cercanía del transporte público.

Aquí, los pesos iniciales pueden ser aleatorios, pero con el tiempo, a medida que el perceptrón revisa los ejemplos y ajusta los pesos, el modelo aprenderá a tomar la decisión correcta (ir o no ir al festival) en función de las entradas.

Por ejemplo, si el clima tiene un peso alto (digamos $w_1=6$) y el transporte público tiene un peso bajo ($w_3=2$), el perceptrón aprenderá que el clima es el factor más importante en la decisión.

Perceptrón

El perceptrón no es un modelo completo de toma de decisiones humana. Pero lo que el ejemplo ilustra es cómo un perceptrón puede sopesar diferentes tipos de evidencia para tomar decisiones.

Una sola neurona o perceptrón tiene ciertas limitaciones

Mediante la combinación de neuronas se pueden construir modelos más complejos.

Redes Neuronales

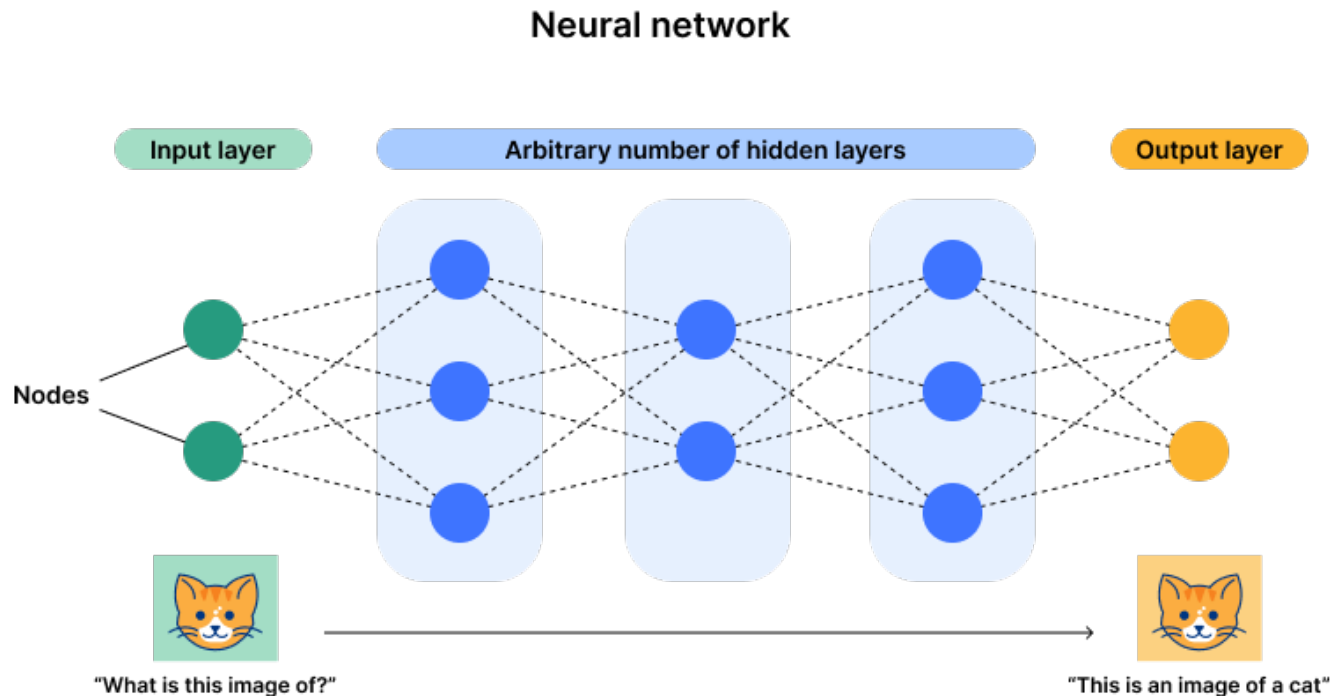
Una sola neurona representa una función lineal. Si bien los modelos lineales son bastante poderosos, tienen ciertas limitaciones, la más clara de ellas es que no permiten la interacción de distintos inputs. Como se ve en la ecuación, cada input está multiplicado por su respectivo peso, pero no interactúan entre ellos.

En la introducción histórica ya vimos que la corriente conexionista defendía que un gran número de pequeñas unidades de cómputo pueden alcanzar un comportamiento inteligente mediante su conexión en una red. De manera similar, juntando varias neuronas podemos crear una red neuronal, capaz de resolver problemas más complejos y de permitir interacciones entre los diversos inputs mediante la conexión de distintas neuronas.

Una red neuronal, también llamada multilayer perceptron (MLP), es un conjunto de neuronas simples situadas en capas.

Redes Neuronales

- ✓ Normalmente, al hablar de redes neuronales nos referimos a **fully connected networks**: redes donde todas las neuronas o nodos de una capa están conectados con todas las neuronas de la capa anterior.



Las distintas neuronas se colocan en capas y reciben como inputs la salida de todas las neuronas de la capa anterior.

Capa por capa, los outputs de una neurona son una fuente de datos para las neuronas de la siguiente capa.

Vemos aquí claramente la idea clave del **deep learning**: obtener representaciones cada vez más complejas y basadas en conceptos simples.

Como se entrena una red neuronal

- ✓ Entrenar una red neuronal significa encontrar los valores óptimos de los pesos (w) y bias (b) de todas las neuronas que mejor describen los datos en términos del output deseado.
- ✓ Por ejemplo, y como ya hemos visto: el output podría ser la probabilidad de que una imagen sea de un gato y el input, todos los píxeles que componen esa imagen.
- ✓ Otro punto a tratar aquí es que una red del tipo de la figura anterior se denomina feedforward network. Esto es debido a que no hay bucles en la red y la información siempre fluye hacia delante.
- ✓ Durante el curso, veremos otro tipo de redes que no siguen este paradigma, llamadas recurrent neural networks.

Arquitectura de una red de neuronas: capas, funciones de activación

✓ Existen numerosas variantes y modificaciones de redes de neuronas en función de:

La topología de la red (arquitectura): que describe el número de neuronas en el modelo, así como el número de capas y la forma en que están conectadas.

La función de activación de cada una de las neuronas artificiales: esta función transforma las entradas de una neurona en la señal que se propaga por la red.

El algoritmo de entrenamiento: que especifica cómo se establece la conexión de los pesos para inhibir y/o excitar neuronas en función de la señal de entrada.

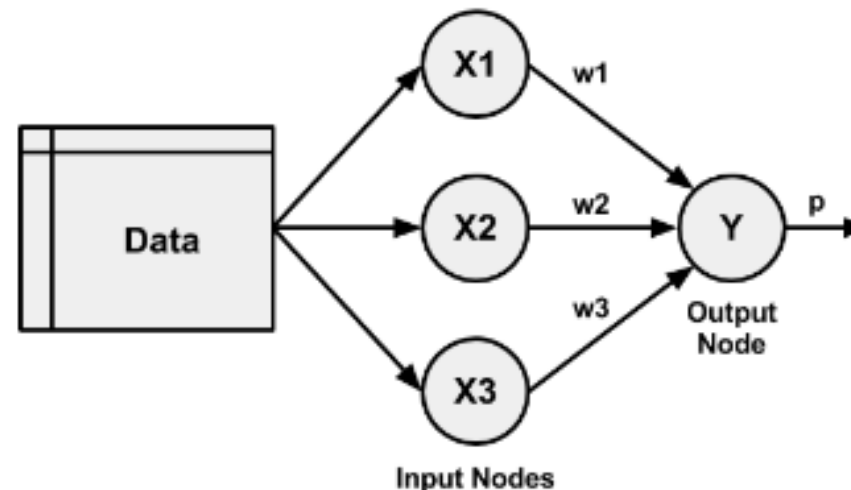
Topología de la red

- ✓ La capacidad de una red de neuronas de aprender se debe a su topología, es decir, a los patrones y estructuras de las neuronas conectadas. Existen diversas topologías de red, pero todas ellas se pueden definir en función de:
 - El número de capas.
 - Si se permite que los datos de la red viajen hacia atrás, lo que se conoce como dirección de la información.
 - El número de neuronas (nodos) en cada capa de la red.
- ✓ La topología determina la complejidad de las tareas que pueden ser aprendidas por la red.
- ✓ Generalmente, redes más grandes y complejas son capaces de identificar patrones más sutiles y fronteras de decisión más complejas.
- ✓ Sin embargo, la potencia de la red no solo esta en función de su tamaño sino también de la forma en que las neuronas se conectan.

Topología de la red

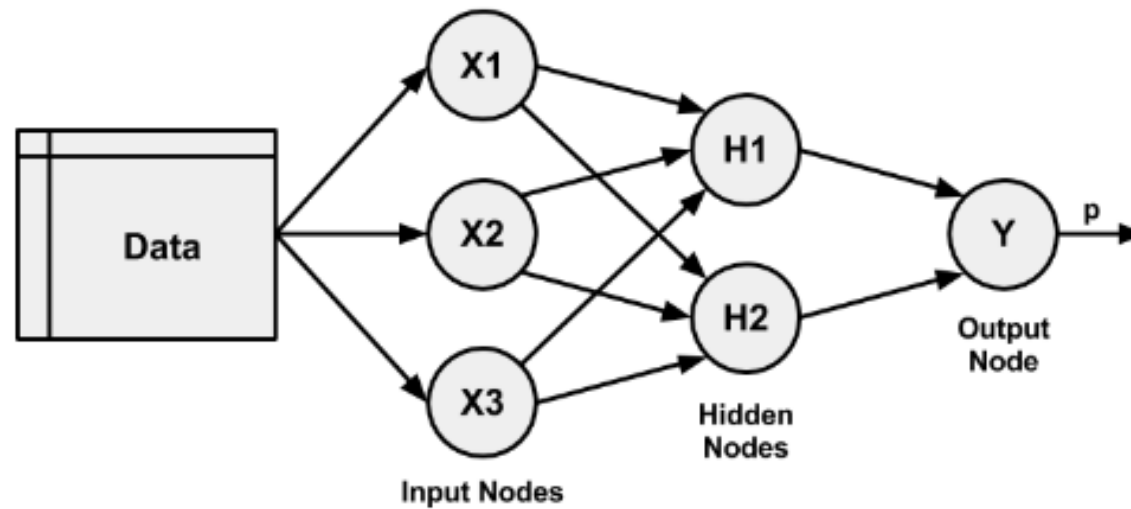
✓ Número de Capas

- Las neuronas de entrada reciben las señales sin procesar directamente de los datos de entrada. Por regla general cada una de las neuronas de entrada es capaz de procesar una variable (feature) del conjunto de datos. El valor de esta variable se transforma por medio de la función de activación del nodo. A su vez las señales de los nodos de entrada se reciben en el nodo de salida, el cual utiliza su propia función de activación (que puede ser diferente de las anteriores) para realizar la estimación.
- Los nodos de entrada y salida se agrupan en capas. Por ejemplo, en la siguiente figura se muestra una red de neuronas con 0 capas ocultas, pues la entrada está conectada directamente a la neurona de la capa de salida.



Topología de la red

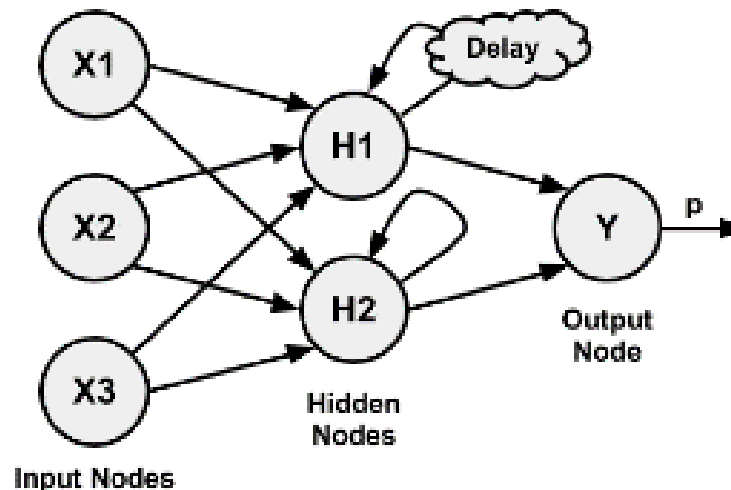
- ✓ Una red de neuronas multicapa añade una o más capas ocultas que procesan las señales de entrada antes de alcanzar el nodo de salida.
- ✓ La mayoría de las redes multicapa son fully connected lo cual implica que cada nodo en una capa se conecta a todos los nodos en la capa siguiente como se muestra en la siguiente figura.



Topología de la red

✓ Dirección de información

- Las redes en las que los datos viajan de una conexión a otra hasta alcanzar las capas de salida se conocen con el nombre de feedforward networks. A pesar de la limitación del flujo de información las redes feedforward proporcionan mucha flexibilidad.
- Por otro lado, las redes de neuronas recurrentes, recurrent network, proporcionan un feedback que permiten que las señales viajen en los dos sentidos por medio de bucles. Esta propiedad les permite aprender patrones más complejos.
- La adición de una memoria a corto plazo (con un delay) permite la capacidad de entender secuencias de eventos en el tiempo. En la siguiente figura se muestra un ejemplo de una red recurrente con memoria.



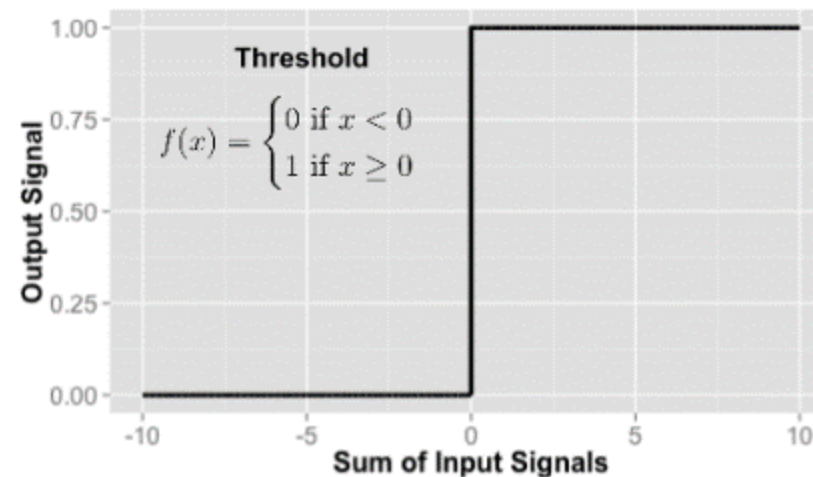
Topología de la red

✓ Número de neuronas (nodos)

- El número de neuronas de la capa de entrada viene determinado por el número de variables de los datos de entrada. El número de neuronas de la capa de salida viene determinado por el número de salidas a modelar o el número de niveles de la clase.
- No hay una regla para determinar el número de neuronas de cada capa. El número apropiado depende del número de variable de entrada, la cantidad de datos de entrenamiento, la cantidad de datos con ruido y la complejidad de la tarea de aprendizaje.
- En general, redes más complejas con un mayor número de conexiones permiten aprender problemas más complejos.
- La mejor práctica es empezar con pocas neuronas e ir incrementando el número de forma gradual.

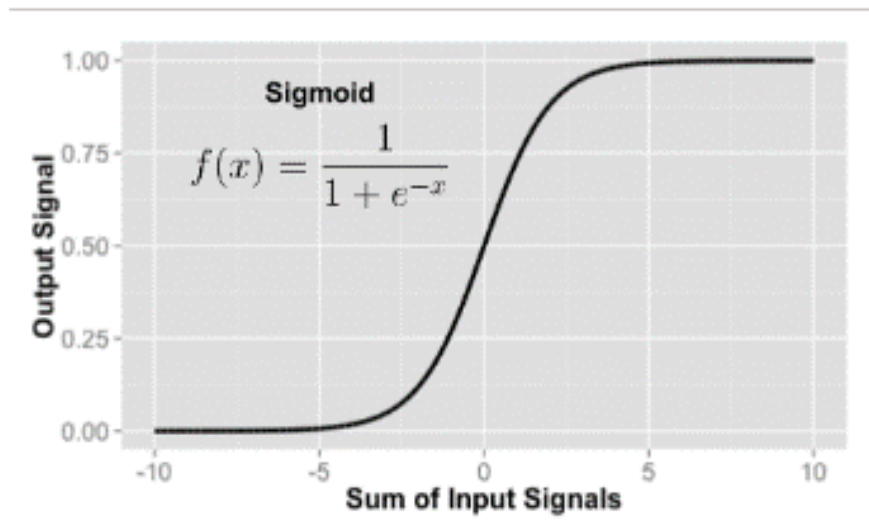
Funciones de activación

- ✓ La función de activación es el mecanismo por el cual las neuronas artificiales procesan la información y esta se propaga por la red.
- ✓ La señal de entrada se agrega y si supera un umbral determinado la señal atraviesa la neurona. Este umbral se conoce con el nombre de threshold activation function. Por ejemplo, la función de activación unitaria se activa cuando la suma de la señal de entrada es > 0 .



Funciones de activación

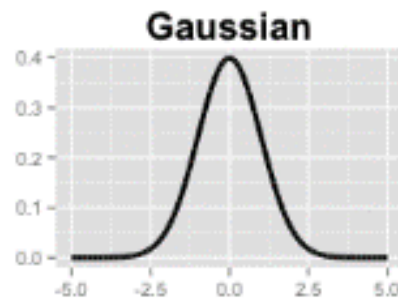
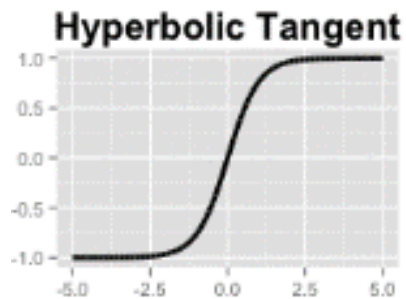
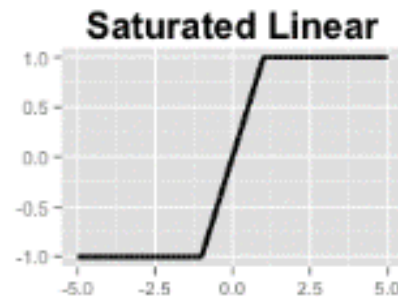
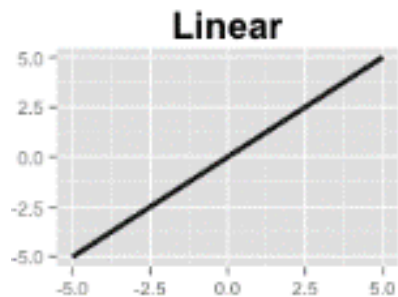
- ✓ Una de las funciones de activación más utilizada es la función sigmoide. Esto se debe a que la función es diferenciable lo que implica que se puede calcular la derivada a lo largo de todo el rango de entrada y simplifica el proceso de entrenar la red.



Para muchas funciones de activación el rango de los valores de entrada es muy estrecho. Por ejemplo, en el caso de la *sigmoide* la señal de salida siempre es 0/1 para valores por debajo o encima de -5/+5. Por este motivo es necesario normalizar los datos de entrada antes de entrenar y a la hora de predecir con una red de neuronas.

Funciones de activación

- ✓ Existen muchas otras funciones de activaciones disponibles como pueden ser la función de activación lineal, lineal saturada, tangente hiperbólica y Gaussiana.



La principal diferencia entre las funciones de activación es el rango de la señal de salida, el cual puede variar entre (0, 1), (-1, +1) y $(-\infty, +\infty)$.

La elección de la **función de activación** sesga la red de neuronas haciendo que trate mejor ciertos tipos de datos.

Por ejemplo, una función de activación lineal da lugar a una red de neuronas similar a un modelo de regresión lineal, mientras que una **función de activación gaussiana** da lugar a un **modelo de red de base radial (Radial basis function)**.

Algoritmo de entrenamiento: backpropagation

- ✓ El proceso de entrenamiento de una red de neuronas consiste en encontrar el valor óptimo de los pesos. Se trata de una tarea costosa computacionalmente. El algoritmo utilizado para el entrenamiento de la red de neuronas se conoce con el nombre de backpropagation.
- ✓ Este algoritmo proporciona una estrategia eficiente para entrenar las redes. El algoritmo itera en ciclos, llamados epochs, utilizando dos fases en cada ciclo:
 1. **Forward:** las neuronas se activan en secuencia desde la capa de entrada a la capa de salida aplicando los pesos de cada neurona y la función de activación.
 2. **Backward:** la señal de salida de la red se compara con el valor real. El error se propaga hacia atrás en la red para modificar los pesos entre las neuronas y reducir errores futuros.

Algoritmo de entrenamiento: backpropagation

Para determinar en cuanto se deben de modificar los pesos de una red se utiliza una técnica llamada descenso del gradiente (gradient descent).

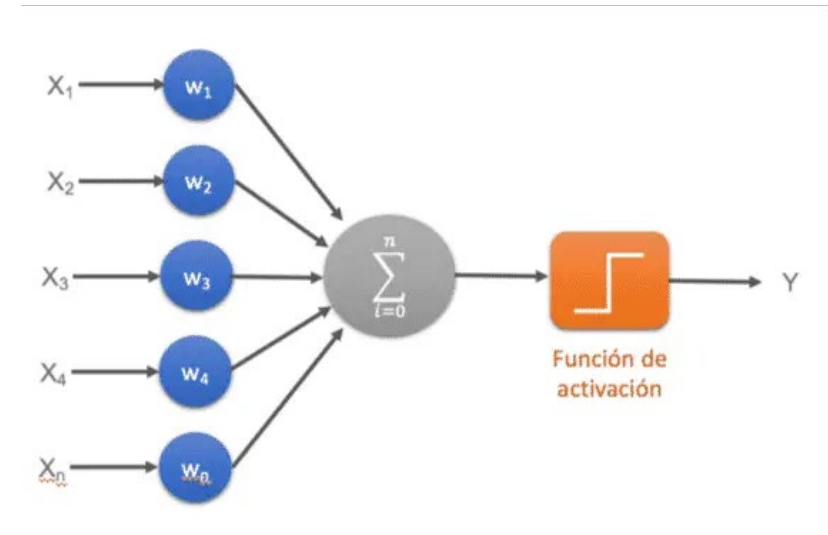
Los pesos se modifican siguiendo la dirección que produce una mayor reducción del error, utilizando para ello la derivada de la activación de cada neurona para identificar el gradiente de la dirección de los pesos futuros.

El algoritmo intenta modificar aquellos pesos que proporcionan una mayor reducción del error utilizando un parámetro conocido como learning rate. Cuanto mayor es el learning rate más rápido el algoritmo desciende por los gradientes.

Tipos de redes neuronales

✓ Red neuronal Monocapa – Perceptrón simple

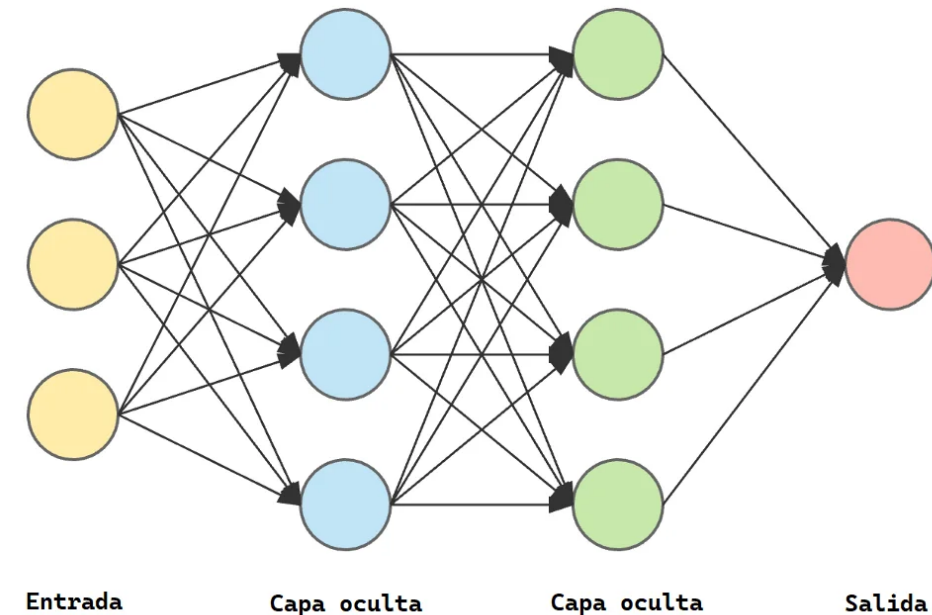
- La red neuronal monocapa se corresponde con la red neuronal más simple, está compuesta por una capa de neuronas que proyectan las entradas a una capa de neuronas de salida donde se realizan los diferentes cálculos.



Tipos de redes neuronales

✓ Redes Neuronales Feedforward - Multicapa – Perceptrón multicapa

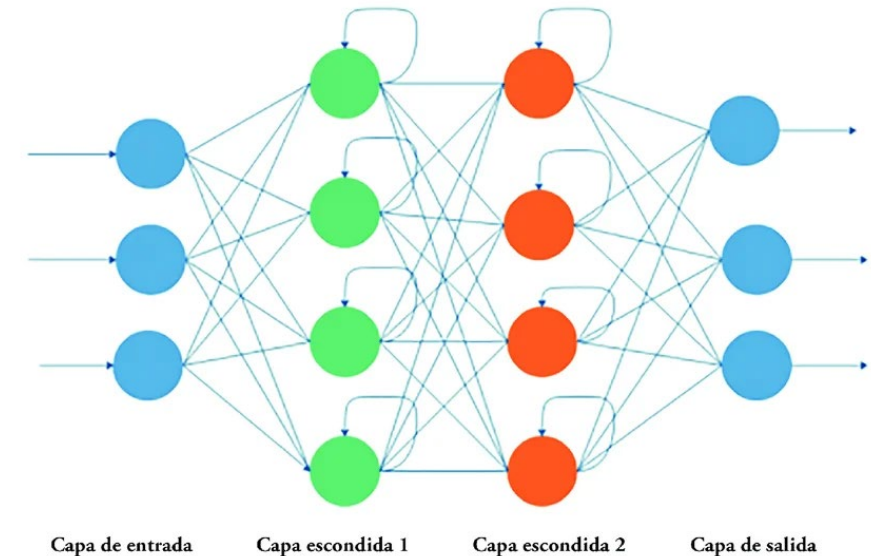
- Poseen un flujo de información unidireccional desde la entrada hasta la salida, sin bucles.
- Una red neuronal prealimentada es una red neuronal artificial donde las conexiones entre las unidades no forman un ciclo. Estas son diferentes de las redes neuronales recurrentes. La red neuronal prealimentada fue la primera y más sencilla forma de red neuronal artificial ideada.
- Como aplicaciones principales, se pueden destacar:
 - Clasificación de imágenes: Reconocimiento de productos, detección de defectos en imágenes.
 - Predicción de series temporales: Pronósticos de ventas, análisis del mercado bursátil, gestión energética.
 - Detección de fraudes: Identificación de transacciones fraudulentas en tarjetas de crédito.



Tipos de redes neuronales

✓ Redes Neuronales Recurrentes (RNN)

- Permiten procesar secuencias de datos, ya que la información se retroalimenta a través de bucles en la red. Una red neuronal recurrente (RNN) es un tipo de red neuronal artificial que utiliza datos secuenciales o datos de series temporales.
- Estos algoritmos de deep learning se usan comúnmente para problemas ordinarios o temporales como, por ejemplo, la traducción de idiomas, el procesamiento del lenguaje natural (NLP), el reconocimiento de voz y los subtítulos de imágenes; se incorporan a aplicaciones populares como Siri y Google Translate.
- Como las redes neuronales convolucionales (CNN) y las feedforward, las redes neuronales recurrentes utilizan datos de entrenamiento para aprender. Se distinguen por su “memoria”, ya que obtienen información de entradas anteriores para influir en la entrada y salida actuales.
- En contraposición, las redes neuronales profundas tradicionales asumen que las entradas y salida son independiente entre sí.
- Algunas aplicaciones típicas:
 - Traducción automática: Traducción de textos de un idioma a otro.
 - Reconocimiento de voz: Conversión de voz a texto.
 - Análisis de sentimientos: Detección de emociones en textos o publicaciones en redes sociales.

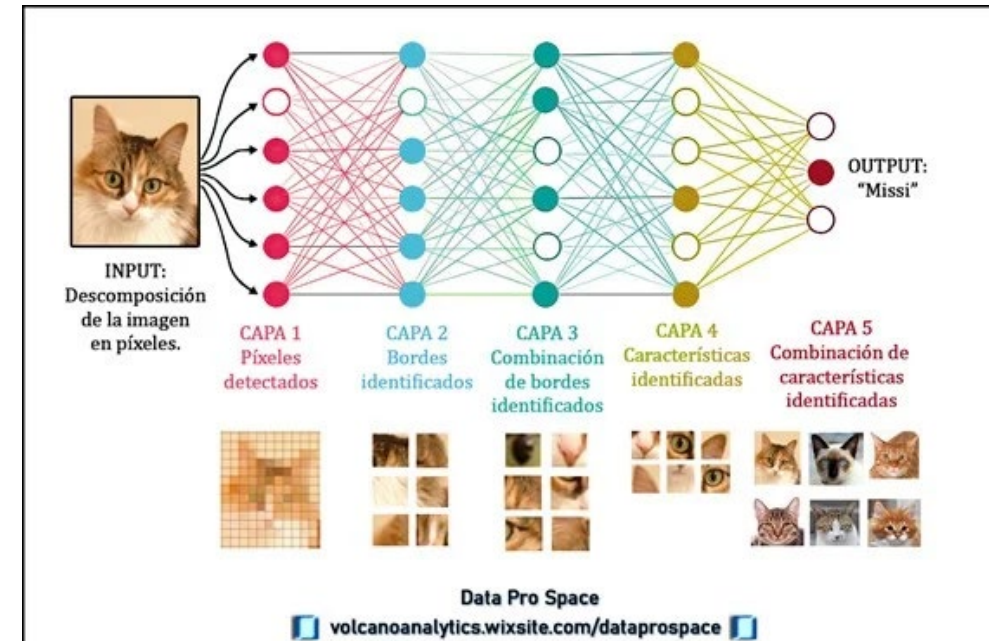


Tipos de redes neuronales

✓ Redes Neuronales Convolucionales (CNN)

➤ Se especializan en el procesamiento de imágenes, detectando patrones y características visuales. Una red neuronal convolucional es un tipo de red neuronal artificial donde las neuronas artificiales, corresponden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria de un cerebro biológico.

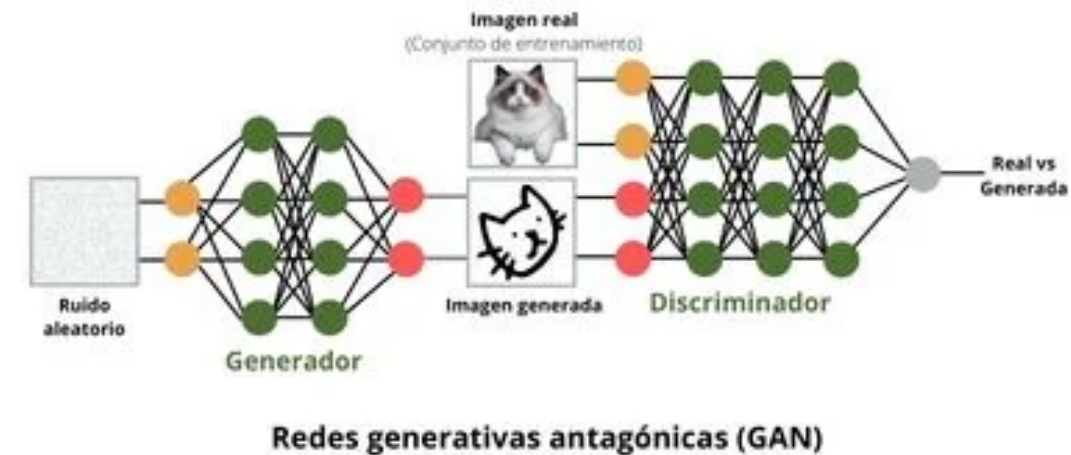
- Reconocimiento facial: Identificación de personas en imágenes o videos.
- Análisis de imágenes médicas: Detección de enfermedades en imágenes de rayos X o resonancia magnética.
- Visión artificial: Inspección de productos en líneas de producción.



Tipos de redes neuronales

✓ Redes Neuronales Generativas (GAN)

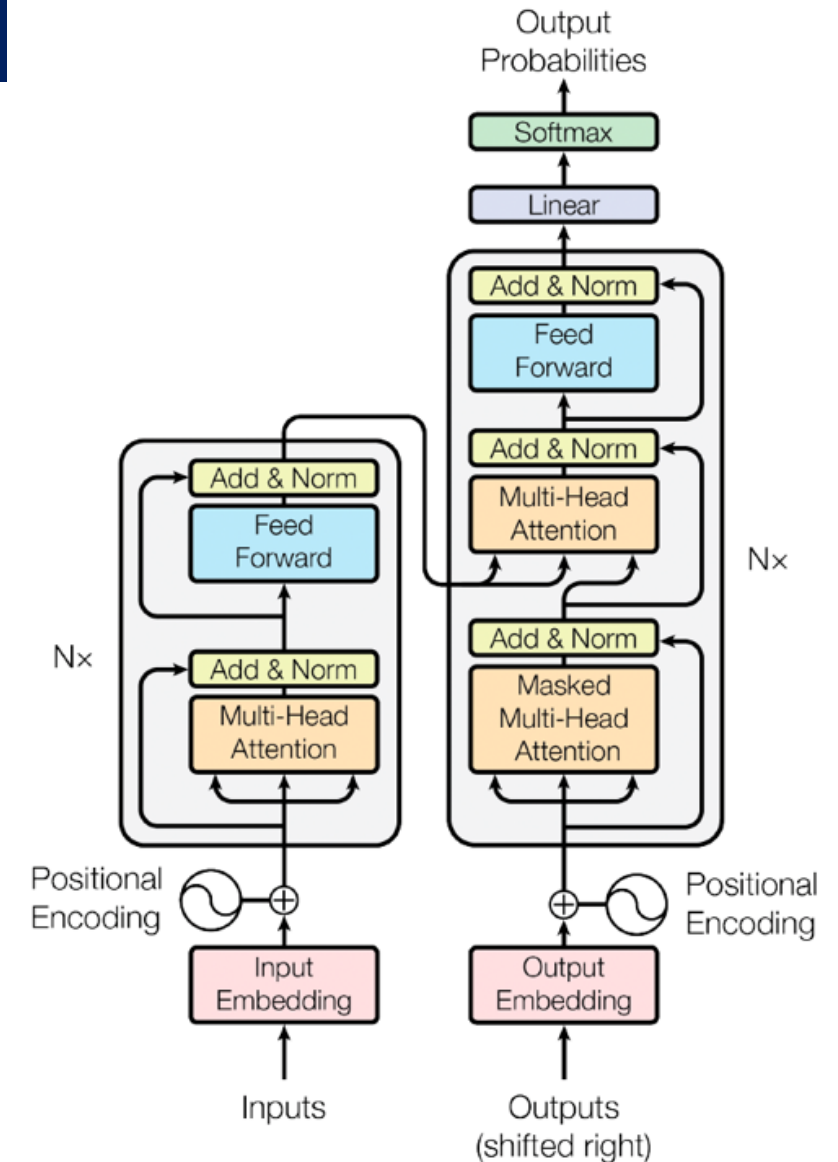
- Se utilizan para generar datos sintéticos realistas, como imágenes, textos o música.
- Arquitectura: Compuestas por dos redes neuronales profundas compitiendo entre sí: un generador y un discriminador. El Generador crea datos nuevos a partir de una distribución de probabilidad aprendida, mientras que el discriminador distingue entre datos reales y datos generados por el generador.
- Entrenamiento: Se realiza a través de un proceso de aprendizaje supervisado no competitivo. El generador busca crear datos que engañen al discriminador haciéndole creer que son reales. El discriminador busca mejorar su capacidad para distinguir entre datos reales y datos falsos.
- Usos:
 - Creación de imágenes: Generación de imágenes de personas o productos que no existen.
 - Traducción automática: Mejora de la calidad de las traducciones automáticas.
 - Desarrollo de videojuegos: Creación de personajes y escenarios realistas.



Tipos de redes neuronales

✓ Transformers

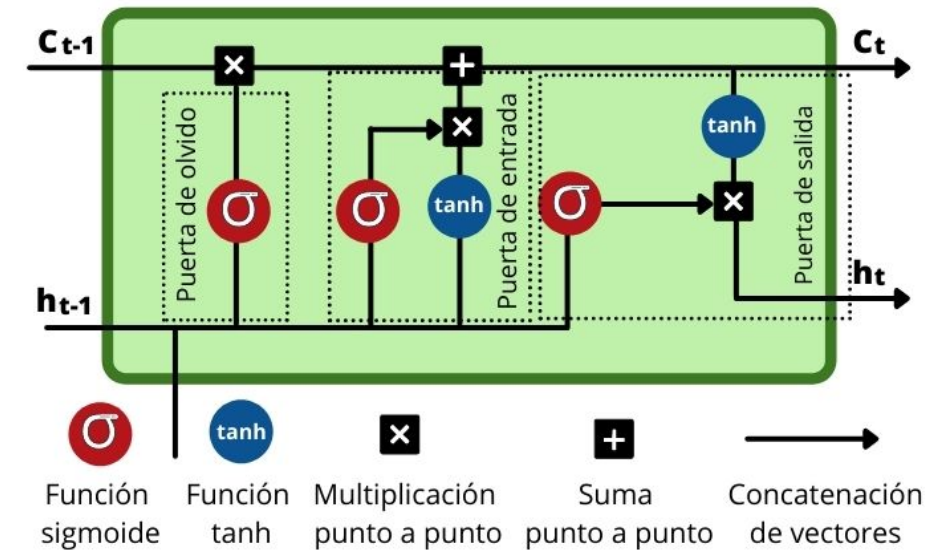
- Los transformadores son un tipo de arquitectura de red neuronal que transforma o cambia una secuencia de entrada en una secuencia de salida. Para ello, aprenden el contexto y rastrean las relaciones entre los componentes de la secuencia. Por ejemplo, considere esta secuencia de entrada: “¿De qué color es el cielo?”. El modelo transformador usa una representación matemática interna que identifica la relevancia y la relación entre las palabras color, cielo y azul. Usa esa información para generar el resultado: “El cielo es azul”.
- Las organizaciones usan modelos de transformadores para todo tipo de conversiones de secuencias, desde el reconocimiento de voz hasta la traducción automática y el análisis de secuencias de proteínas.



Tipos de redes neuronales

✓ LSTM: Long short-term memory

- LSTM (Long short-term memory) es un tipo de red neuronal recurrente (RNN) que se utiliza en el aprendizaje profundo para procesar y predecir secuencias de datos.
- La LSTM fue diseñada para abordar el problema de la desaparición del gradiente en las redes neuronales recurrentes tradicionales, que se produce cuando se retropropaga el error a través de múltiples capas y se pierde información importante en el proceso.
- La LSTM utiliza una estructura de celda con puertas que permite a la red controlar la cantidad de información que se almacena y se olvida en cada paso de tiempo, lo que la hace especialmente adecuada para el procesamiento de secuencias de datos a largo plazo.
- Las LSTMs se han utilizado con éxito en una amplia variedad de aplicaciones de aprendizaje profundo, como el procesamiento del lenguaje natural, el reconocimiento de voz, la generación de texto y la predicción de series temporales.



Una clasificación mas detallada: <https://www.asimovinstitute.org/neural-network-zoo/>

PREGUNTAS??

Dr. Edwin Valencia Castillo
Departamento de Sistemas
Facultad de Ingeniería
Universidad Nacional de Cajamarca
2024