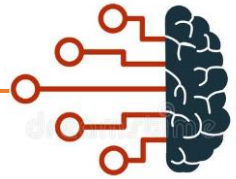
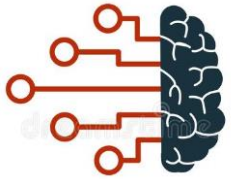


Aprendizaje Supervisado
Deep Learning

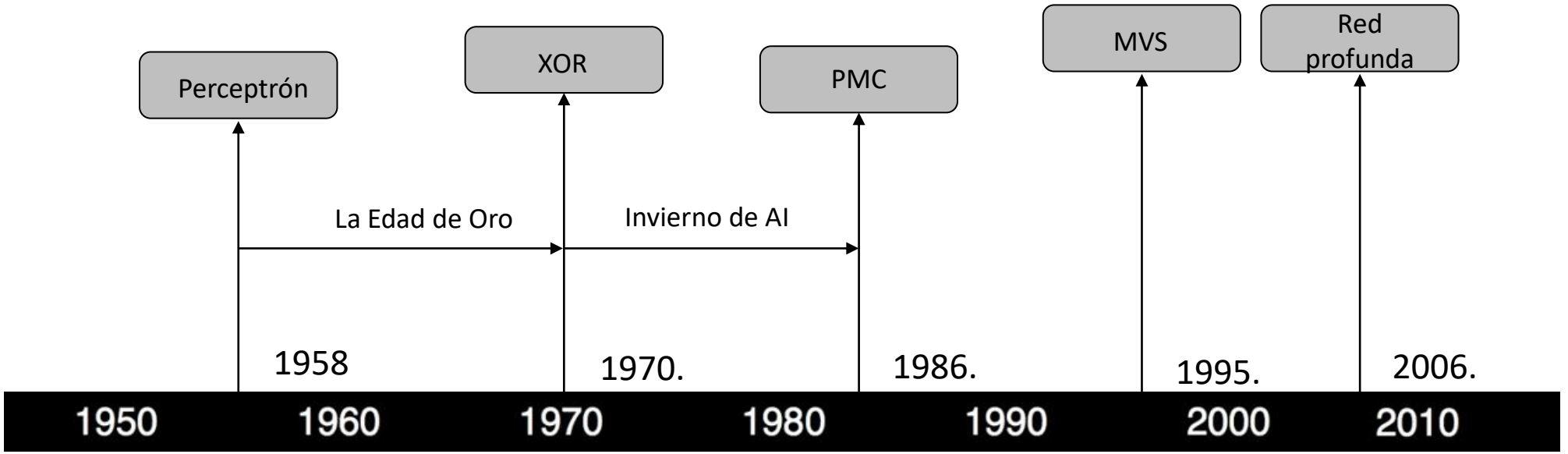
Lic. Patricia Rodríguez Bilbao

Contenido

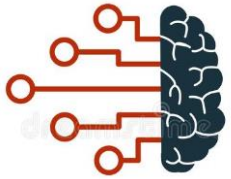


1. Definiciones
2. Función de activación
3. Red Neuronal
4. Tipos de redes neuronales
5. Problemas comunes

Historia del desarrollo de las RNA

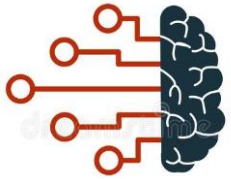


Red Neural Convolucional (CNN)



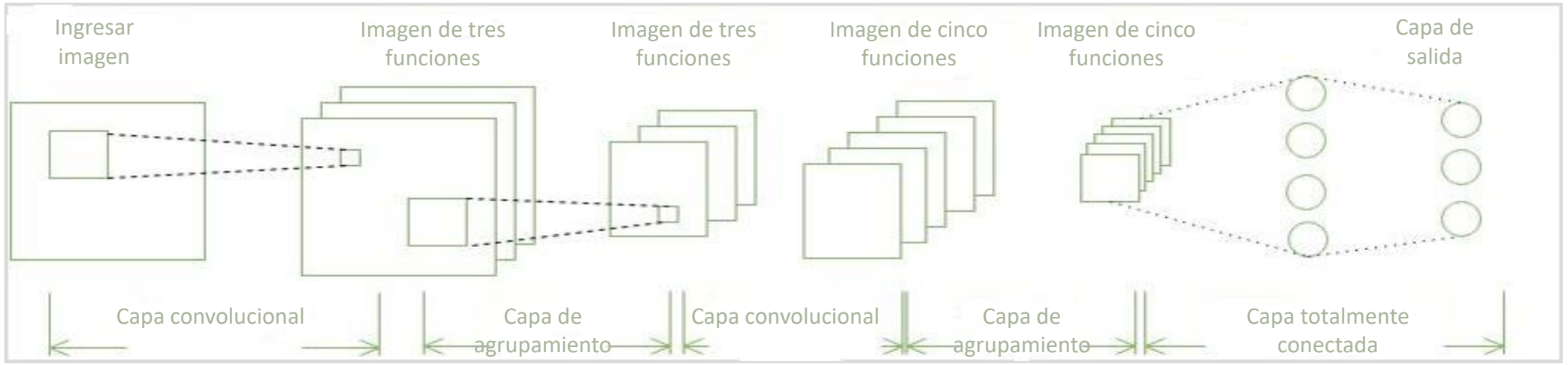
- En los años 60, Hubel y Wiesel estudiaron las neuronas del cortex de los gatos utilizadas para la sensibilidad local y la selección de dirección, y descubrieron que su estructura de red única podía simplificar las redes neuronales de retroalimentación y propusieron el CNN.
- La CNN, es una red de prealimentación (feedforward), sus neuronas responden **a las unidades circundantes dentro del rango de cobertura**, destaca en el procesamiento de imágenes, incluye **una capa convolucional, una capa de pooling y una capa totalmente conectada.**
- En la actualidad, la CNN se ha convertido en uno de los puntos críticos de la investigación en muchos campos científicos, especialmente en el campo de la clasificación de patrones. La red es ampliamente utilizada porque evita el pre-procesamiento de imágenes y permite la entrada directa de imágenes originales.

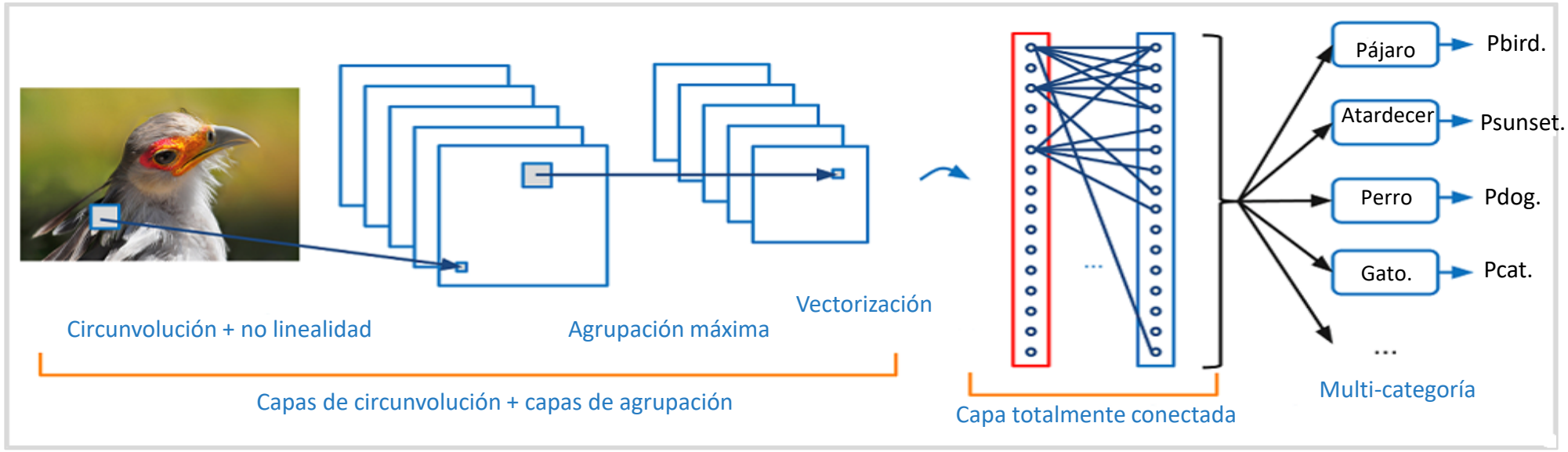
Conceptos principales de la Red CNN



- **Campo receptivo local:** Se considera generalmente que la percepción humana del mundo exterior va de local a global. **Las correlaciones espaciales entre los píxeles locales de una imagen están más cerca que entre los píxeles distantes**. Por lo tanto, cada neurona no necesita conocer la imagen global. Sólo necesita conocer la imagen local. La información local se combina a un nivel superior para generar información global.
- **Compartir parámetros**: se pueden usar uno o más filtros/kernels para escanear las imágenes de entrada. Los parámetros transportados por los filtros son pesos. En una capa escaneada por filtros, cada filtro utiliza los mismos parámetros durante el cálculo ponderado. Compartir el peso significa que cuando cada filtro escanea una imagen entera, los parámetros del filtro son fijos.

Arquitectura de la red neuronal convolucion





Cálculo de un solo filtro

- Descripción del cálculo de convolución

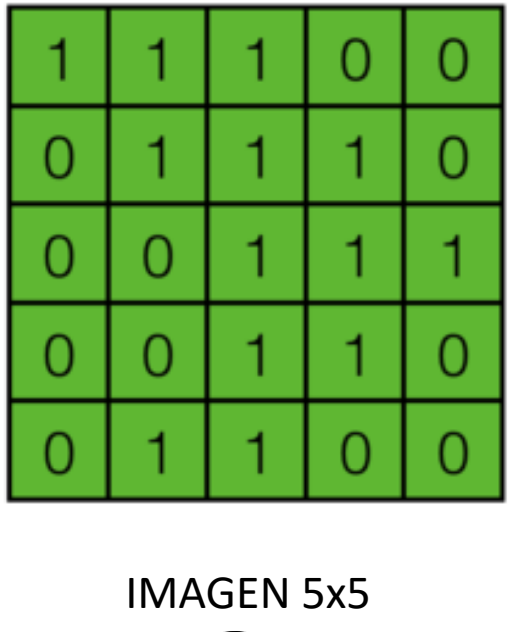
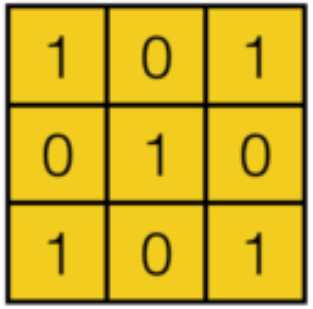
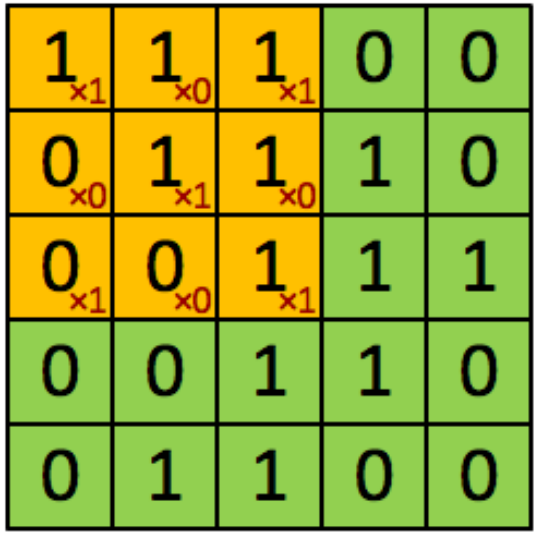


IMAGEN 5x5



SESGO 0

FILTRO 3x3



IMAGEN

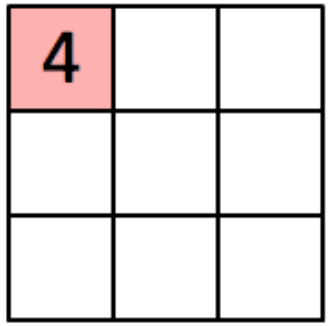
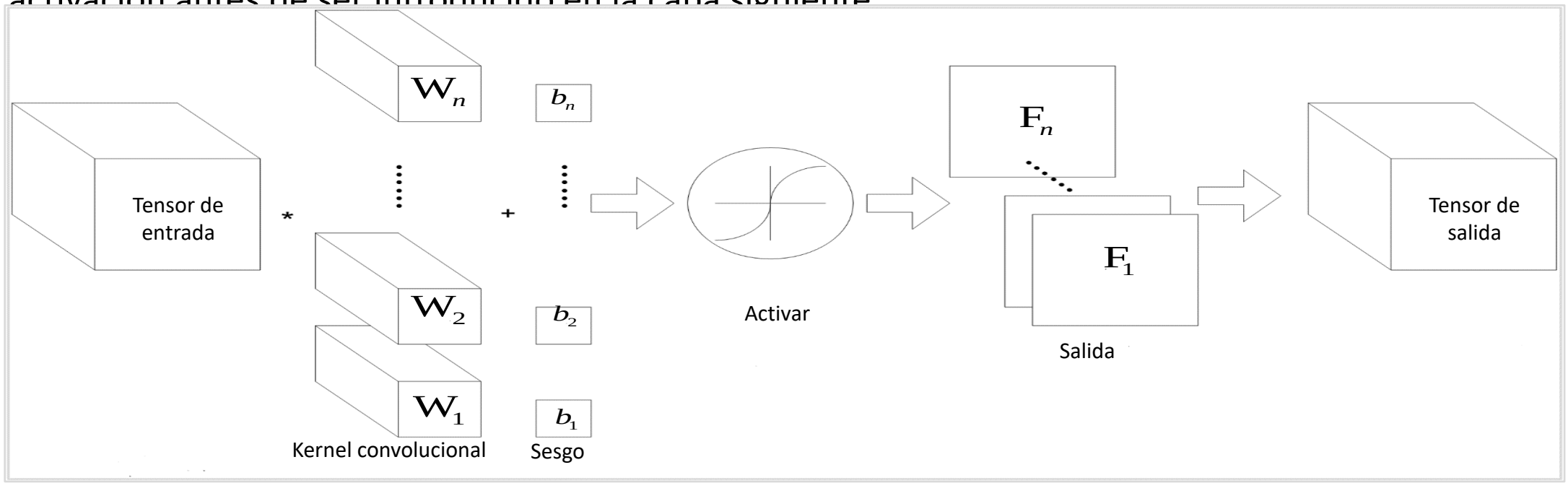


IMAGEN CONVOLUCIONADA

Han Bingtao, 2017, Red Neural Convolucional

Capa convolucional

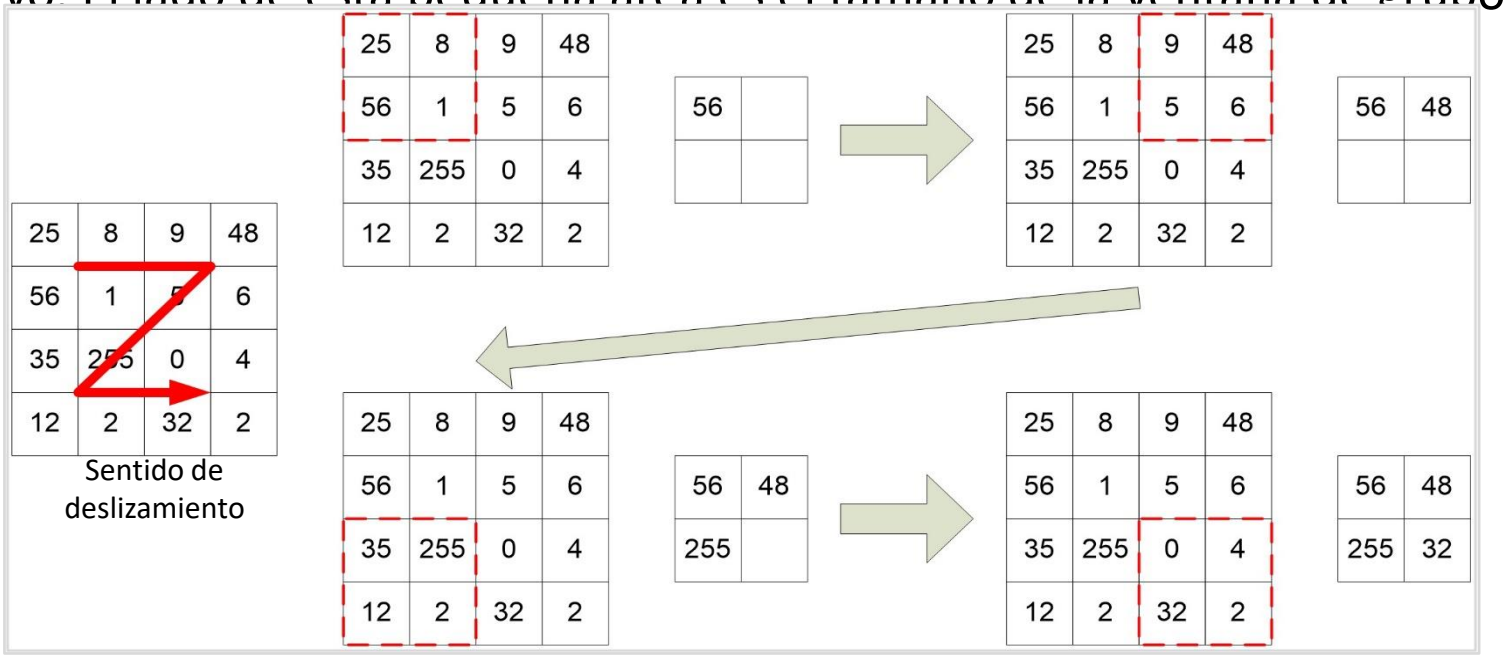
- La arquitectura básica es la convolución multicanal que consiste en múltiples convoluciones individuales.
- La salida de la capa previa (o la imagen original de la primera capa) es la entrada de la siguiente capa, El núcleo de convolución de cada capa es el peso a aprender.
- Una vez finalizada la convolución, el resultado debe ser sesgado y activado a través de funciones de activación antes de ser introducido en la capa siguiente



Capa de agrupamiento (Pooling layer)

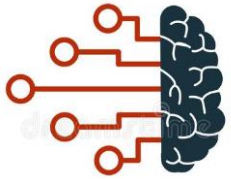
Combina unidades cercanas para reducir el tamaño de la entrada en la capa siguiente, reduciendo las dimensiones.

- La agrupación máxima (max pooling), el valor máximo en un área cuadrada pequeña se selecciona como el representante de esta área
- La agrupación promedio (average pooling), el valor medio se selecciona como el representativo. El lado de esta pequeña área es el tamaño de la ventana de grupo



Agrupación máxima cuyo tamaño de ventana es 2

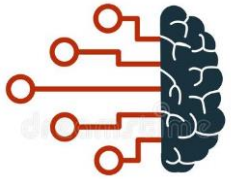
Capa totalmente conectada



- La capa completamente conectada es esencialmente un clasificador. Las características extraídas en la capa convolucional y la capa de agrupación se enderezan y colocan en la capa completamente conectada para generar y clasificar los resultados.
- Generalmente, la función Softmax se utiliza como función de activación de la capa de salida final completamente conectada para combinar todas las características locales en características globales y calcular la puntuación de cada tipo.

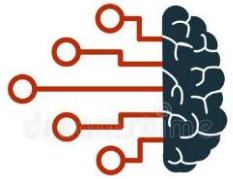
$$\sigma(\mathrm{z})_j = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

Red Neuronal Recurrente – Recurrent Neural Network RNN



- Es una red neural que captura información dinámica en datos secuenciales a través de conexiones periódicas de nodos de capa ocultos. Puede clasificar datos secuenciales.
- La RNN puede mantener un estado de contexto e incluso almacenar, aprender y expresar información relacionada en ventanas de contexto de cualquier longitud. A diferencia de las redes neuronales tradicionales, no se limita al límite del espacio, sino que también soporta secuencias de tiempo. En otras palabras, hay un lado entre la capa oculta del momento actual y la capa oculta del momento siguiente.
- La RNN es ampliamente utilizada en escenarios relacionados con secuencias, tales como videos que consisten en marcos de imágenes, audio que consiste en clips, y oraciones que consisten en palabras.

Arquitectura de Red Neuronal Recurrente (1)



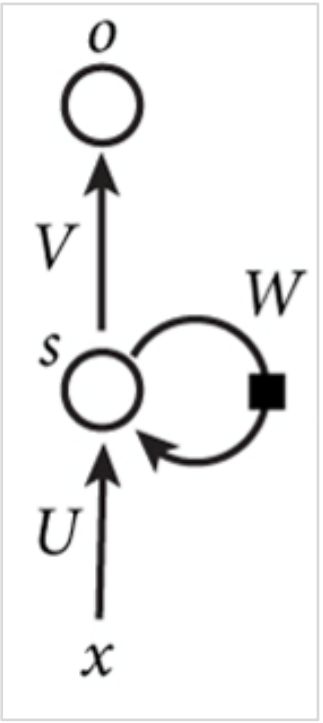
- X_t es la entrada de la secuencia de entrada en el tiempo t.
- S_t es la unidad de memoria de la secuencia en el tiempo t y almacena en caché la información anterior.

$$S_t = tanh(UX_t + WS_{t-1}).$$

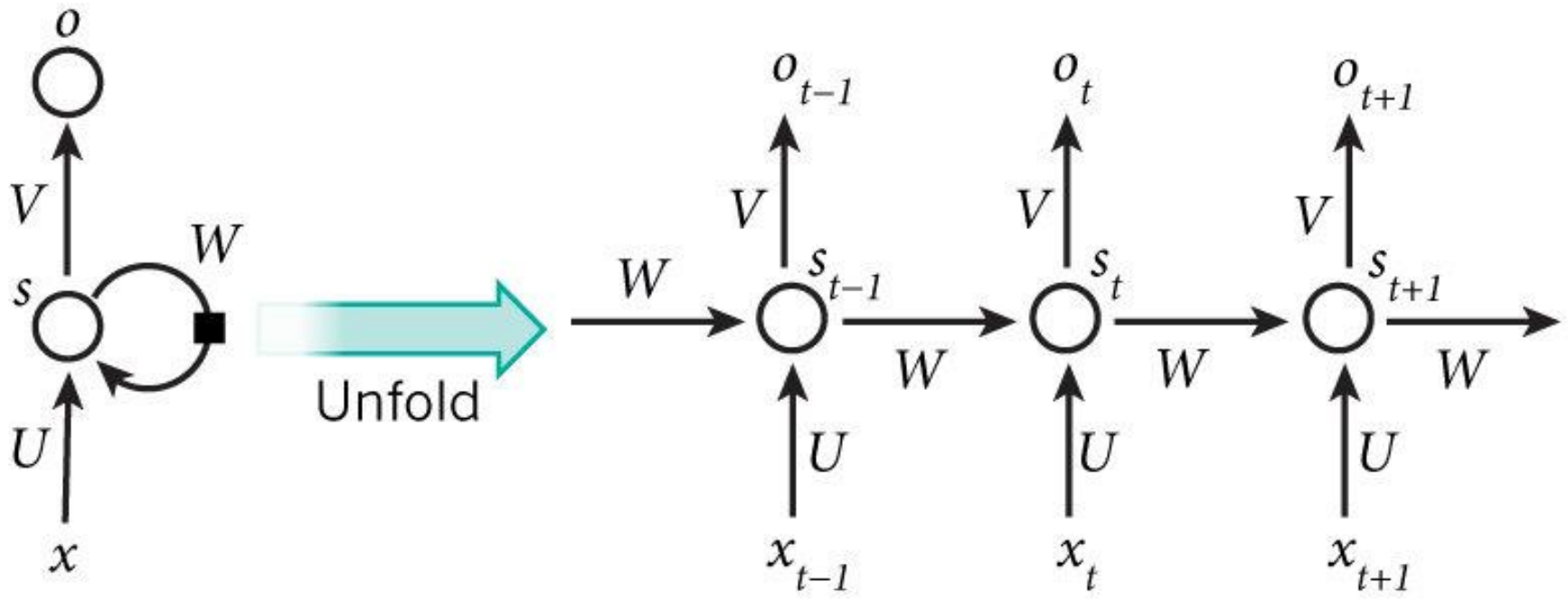
- O_t es la salida de la capa oculta de la secuencia en el tiempo t.

$$O_t = tanh(VS_t)$$

- O_t después de pasar por múltiples capas ocultas, puede obtener el resultado final de la secuencia en el tiempo t.

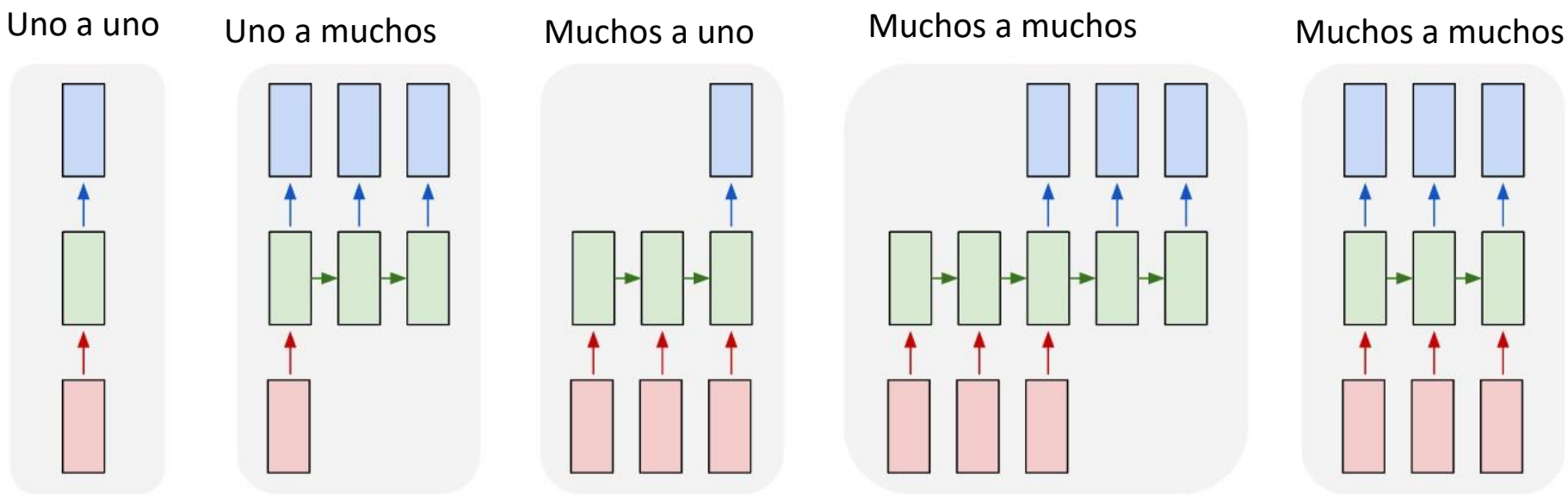


Arquitectura de Red Neuronal Recurrente (2)



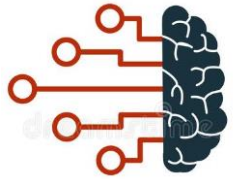
Lecond, Bengio, y G. Hinton, 2015, A Recurrent Neural Network y el despliegue en el tiempo del cómputo involucrado en su cálculo a futuro

Tipos de Redes Neuronales Recurrentes



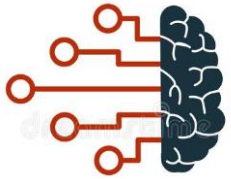
Andrej Karpathy, 2015, La eficacia irrazonable de las redes neuronales recurrentes

Retro propagación a través del tiempo - Backpropagation Through Time (BPTT)



- BPTT:
  - La retropropagación (propagación hacia atrás) tradicional es la extensión de la secuencia de tiempo.
  - Hay dos fuentes de errores en la secuencia en el momento de la unidad de memoria: la primera es del error de salida de la capa oculta en t secuencia de tiempo; el segundo es el error de la celda de memoria en la siguiente secuencia de tiempo t + 1.
  - Cuanto más larga sea la secuencia de tiempo, más probable es que la pérdida de la última secuencia de tiempo al gradiente de w en la primera secuencia de tiempo provoque el problema del gradiente de desaparición o explosión.
  - El gradiente total de peso w es la acumulación del gradiente del peso en toda la secuencia de tiempo..
- Tres pasos de BPTT:
  - Calcular el valor de salida de cada neurona mediante propagación directa.
  - Calcular el valor de error de cada neurona mediante retropropagación δ_j.
  - Calcular el gradiente de cada peso.
- Actualización de pesos mediante el algoritmo SGD.

Problema de Red Neuronal Recurrente

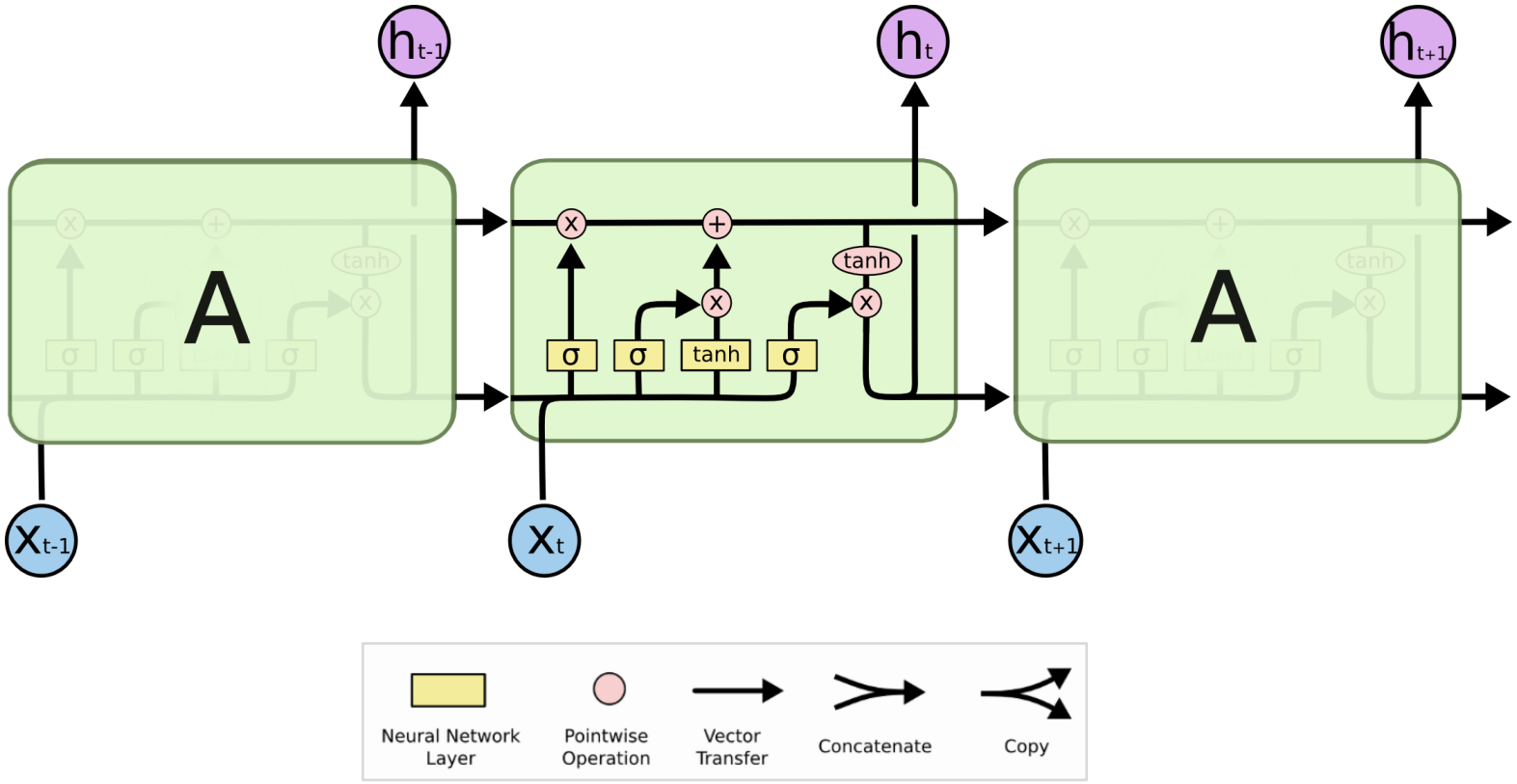


$S_t = \sigma(UX_t + WS_{t-1})$ se exitende en la secuencia de tiempo.

$$S_t = \sigma\left(UX_t + W\left(\sigma\left(UX_{t-1} + W\left(\sigma\left(UX_{t-2} + W(\dots)\right)\right)\right)\right)\right)$$

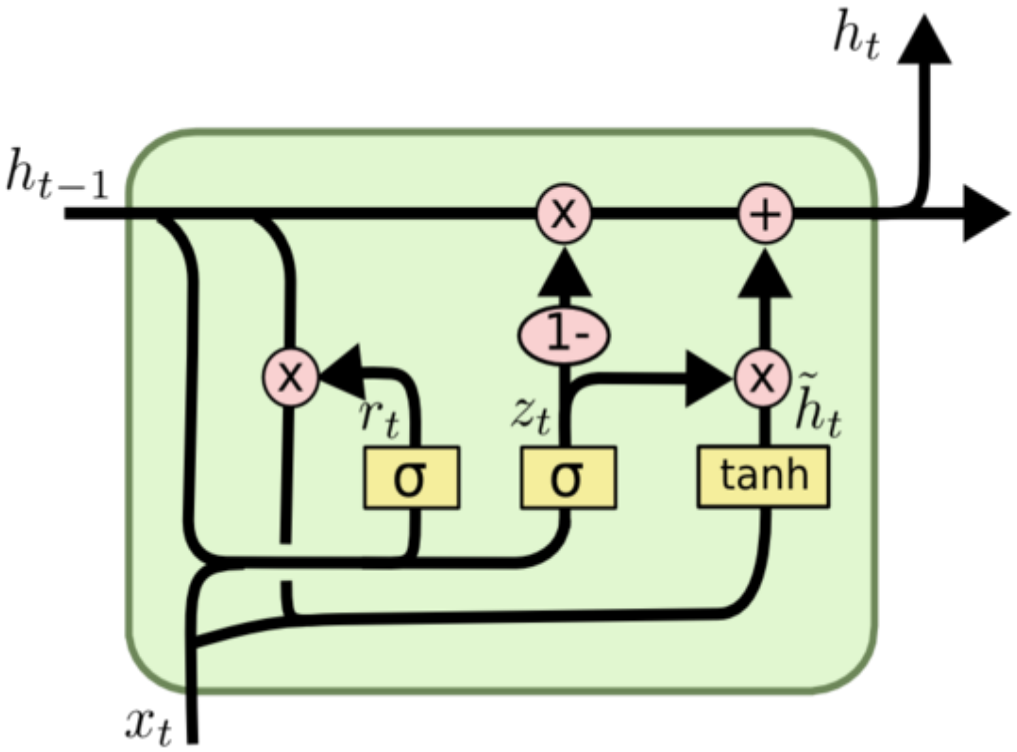
- A pesar de que la estructura estándar de RNN resuelve el problema de la memoria de información, la información se atenúa durante la memoria a largo plazo..
- La información debe guardarse durante mucho tiempo en muchas tareas. Por ejemplo, una pista al comienzo de una ficción especulativa puede no ser respondida hasta el final..
- Es posible que el RNN no pueda guardar información durante mucho tiempo debido a la capacidad limitada de la unidad de memoria.
- Esperamos que las unidades de memoria puedan recordar información clave

Red de Memoria a Corto Plazo (Long short-term Memory Network LSTM)



Colah, 2015, Descripción de las redes de LSTM

Unidad Puerteada Recurrente
Gated Recurrent Unit (GRU)



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

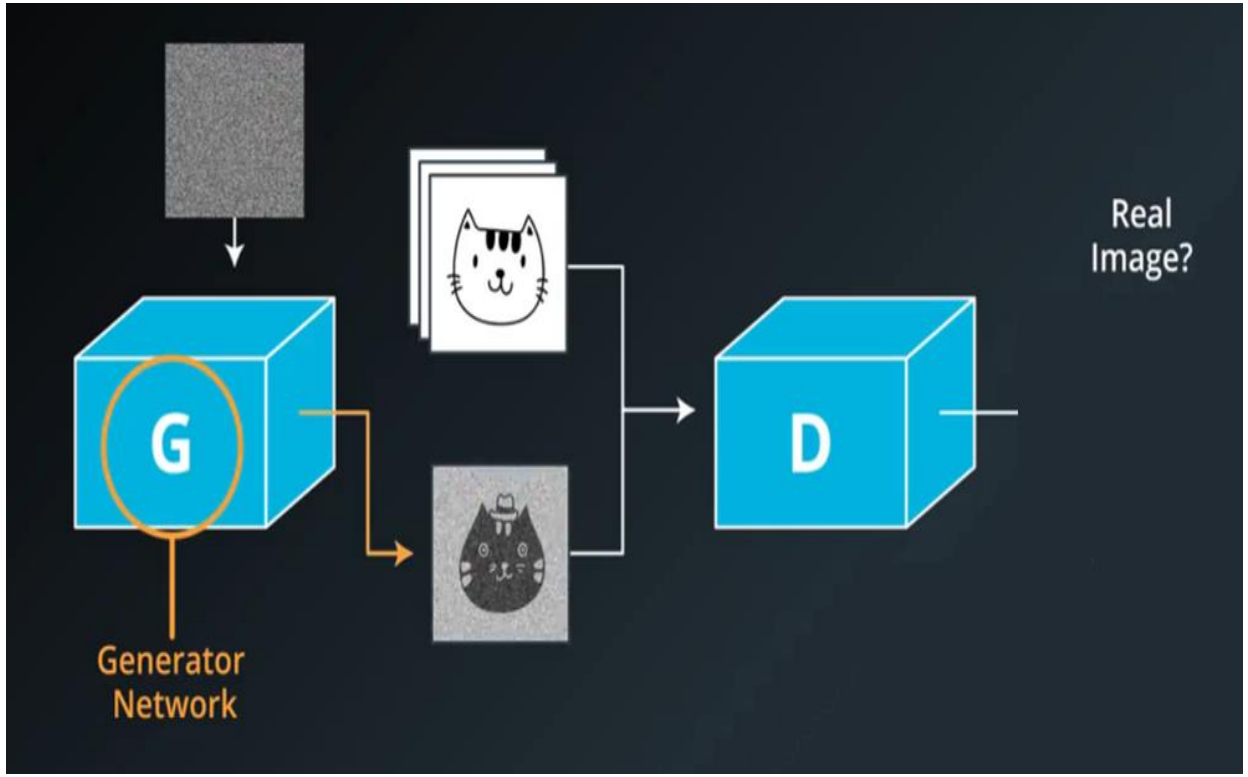
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Red Generativa Adversarial
Generative Adversarial Network (GAN)

Arquitectura

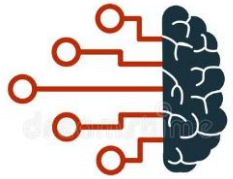


Es un marco que forma al generador G y al discriminador D a través del proceso adversarial.

A través del proceso adversarial, el discriminador puede decir si la muestra del generador es falsa o real.

La GAN adopta un algoritmo BP maduro

Red Generativa Adversarial – Generative Adversarial Network (GAN)



(1) Generador G: La entrada es ruido z, que cumple con la distribución de probabilidad previamente seleccionada manualmente, como distribución par y distribución gaussiana. El generador adopta la estructura de red del perceptrón multicapa (MLP), utiliza los parámetros de estimación de máxima verosimilitud (MLE) para representar el mapeo derivable G(z), y asigna el espacio de entrada al espacio de muestra.

(2) Discriminador D: La entrada es la muestra real x y la muestra falsa G(z), que se etiquetan como real y falso respectivamente. La red del discriminador puede utilizar los parámetros de transporte de MLP. La salida es la probabilidad D(G(z)) que determina si la muestra es real o falsa.

La GAN se puede aplicar a escenarios como la generación de imágenes, la generación de texto, la mejora de voz, la superresolución de imágenes.

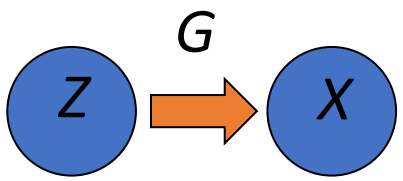
Modelo Generador y Modelo Discriminador

- Red generadora

  Genera datos de muestra

  - Entrada: vector de ruido blanco gaussiano z
  - Salida: vector de datos de muestra x

$$x = G(z; \theta^G)$$

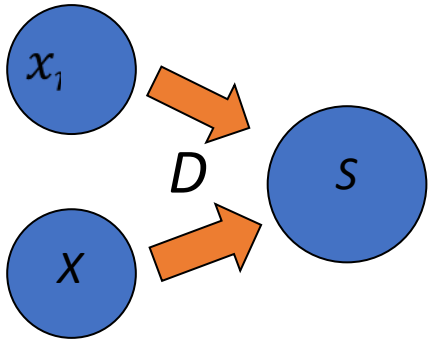


- Red discriminadora

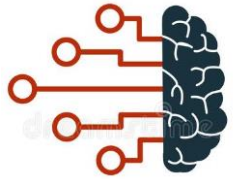
  Determina si los datos de muestra son reales

  - Input: datos de muestra real x_{real} y datos de muestra generada $x = G(z)$
  - Output: probabilidad que determina si la muestra es real o no

$$y = D(x; \theta^D)$$



Normas de entrenamiento de la GAN



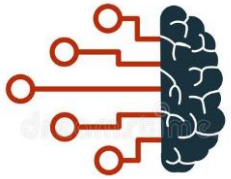
Objetivo de optimización:

función de valor

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_{z(z)}}[log(1-D(G(z)))]$$

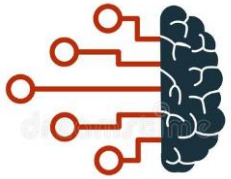
En la fase inicial de formación, cuando el resultado de G es muy pobre, D determina que la muestra generada es falsa con alta confianza, porque la muestra es obviamente diferente de los datos de entrenamiento. En este caso, log(1-D (G (z))) está saturado (donde el gradiente es 0 y no se puede realizar la iteración). Por lo tanto, elegimos entrenar a G solamente minimizando [-log(D(G(z))}].

Problemas Comunes construcción de modelos



- Desequilibrio de datos
- Problema de desvanecimiento de gradiente y de explosión de gradiente
- Sobreajuste

Desequilibrio de datos



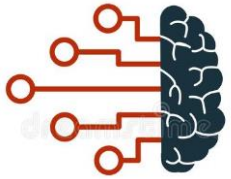
Descripción del problema: En el conjunto de datos que consta de varias categorías de tareas, el número de muestras varía mucho de una categoría a otra. Una o más categorías de las categorías predichas contienen muy pocas muestras.

Por ejemplo, en un experimento de reconocimiento de imágenes, más de 2.000 categorías de un total de 4251 imágenes de entrenamiento contienen sólo una imagen cada una. Algunos de los otros tienen 2-5 imágenes.

Impactos:

- Debido al número desequilibrado de muestras, no podemos obtener el resultado óptimo en tiempo real porque el modelo/algoritmo nunca examina adecuadamente las categorías con muy pocas muestras.
- Dado que pocos objetos de observación pueden no ser representativos de una clase, es posible que no obtengamos muestras adecuadas para la verificación y el ensayo.

Desequilibrio de datos



Submuestreo aleatorio

- Eliminación de muestras redundantes en una categoría

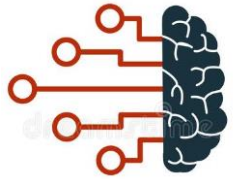
Sobremuestreo aleatorio

- Copiar muestras

Técnica de sobremuestreo de la minoría sintética

- Muestreo
- Combinación de muestras

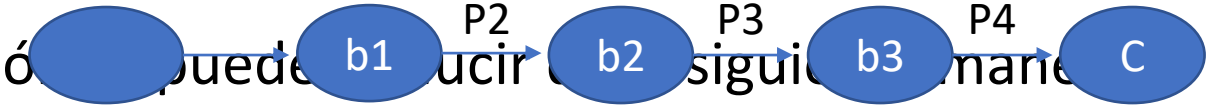
Problema de desvanecimiento o explosión de gradiente



- Desvanecimiento de gradiente: A medida que las capas de red aumentan, el valor derivado de la retropropagación disminuye, lo que causa un problema de desaparición de gradiente.
- Explosión de gradiente: A medida que aumentan las capas de red, aumenta el valor derivado de la retropropagación, lo que causa un problema de explosión de gradiente.
- Causa:

$$y_i = \sigma_{(z_i)} = \sigma(w_i x_i + b_i) \qquad \text{Where } \sigma \text{ is sigmoid function.}$$

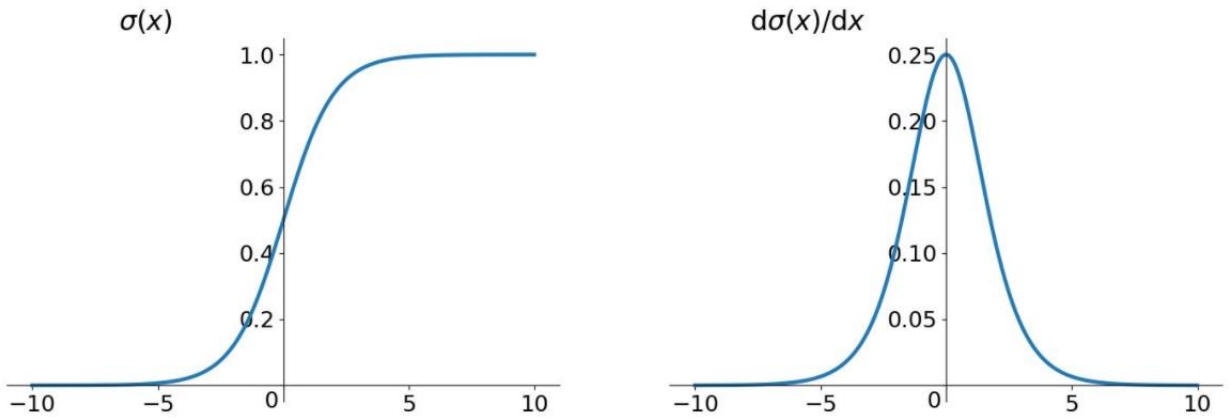
- La retropropagació[illegible]puede[illegible]ucir [illegible]siguie[illegible]mane[illegible]



$$\frac{\partial C}{\partial b_1} = \frac{\partial C}{\partial y_4}\frac{\partial y_4}{\partial z_4}\frac{\partial z_4}{\partial x_4}\frac{\partial x_4}{\partial z_3}\frac{\partial z_3}{\partial x_3}\frac{\partial x_3}{\partial z_2}\frac{\partial z_2}{\partial x_2}\frac{\partial x_2}{\partial z_1}\frac{\partial z_1}{\partial b_1}$$
$$= \frac{\partial C}{\partial y_4}\sigma'(z_4)w_4\sigma'(z_3)w_3\sigma'(z_2)w_2\sigma'(z_1)x$$

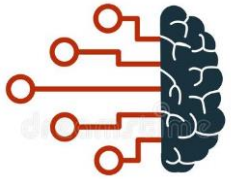
Problema de desvanecimiento o explosión de gradiente

- El máximo valor de $\sigma'(x)$ is $\frac{1}{4}$:

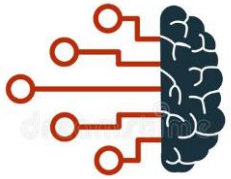


- Sin embargo, el peso de la red | w | suele ser menor que 1. Por lo tanto, | $\sigma'(z)w$ | ≤1 / 4. Según la regla de la cadena, a medida que aumentan las capas, el resultado de la derivación ∂C / (∂b_1) disminuye, lo que da como resultado el problema del gradiente desvaneciente.
- Cuando el peso de la red | w | es grande, resultando en $|\sigma'(z)w|$> 1, se produce el problema del gradiente explosivo.
- Solución: Por ejemplo, el recorte de gradiente se usa para aliviar el problema del gradiente explosivo, la función de activación de ReLU y LSTM se usan para aliviar el problema del gradiente desvaneciente.

Sobreajuste



- Descripción del problema: El modelo funciona bien en el conjunto de entrenamiento, pero mal en el conjunto de pruebas.
- Causa: Hay demasiadas dimensiones de características, supuestos de modelos y parámetros, demasiado ruido, pero muy pocos datos de entrenamiento. Como resultado, la función de ajuste predice perfectamente el conjunto de entrenamiento, mientras que el resultado de predicción del conjunto de pruebas de nuevos datos es pobre. Los datos de entrenamiento están sobreajustados sin considerar las capacidades de generalización.
- Solución: por ejemplo, aumento de datos, regularización,



Gracias……