Fundamentos de Data Science

UTEM
UNIVERSIDAD TECNOLÓGICA METROPOLITANA
del Estado de Chile

SEMANA 7

Prof. Juan Pablo Angerstein
jangerstein@utem.cl

OBJETIVO SEMANA 7

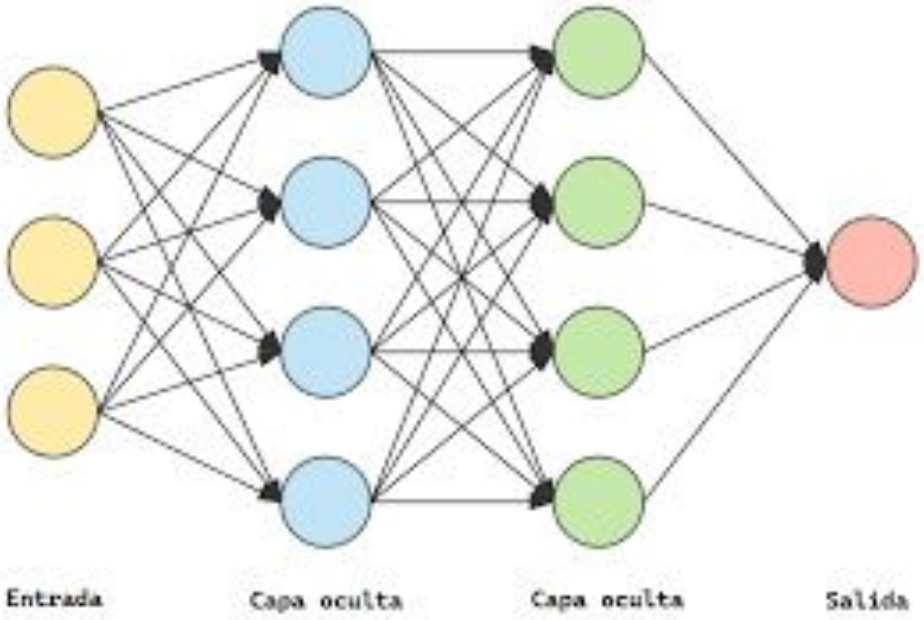
Esta semana nos enfocaremos en la importancia de:

- Introducción a Redes Neuronales

Semana VII: Introducción a Redes Neuronales

¿Qué es una Red Neuronal Artificial?

Es un modelo computacional inspirado (muy vagamente) en la estructura y funcionamiento del cerebro humano. Consiste en un conjunto de nodos de procesamiento interconectados, llamados neuronas, organizados en capas.



Analogía: Un Equipo de Especialistas

Imagina que quieres identificar un animal en una foto.

- La primera capa de especialistas solo identifica bordes y colores.
- La segunda capa, basándose en lo anterior, identifica formas (círculos, líneas).
- La tercera combina esas formas para identificar partes (ojos, orejas, cola).
- La capa final toma esas partes y concluye: "Es un gato".

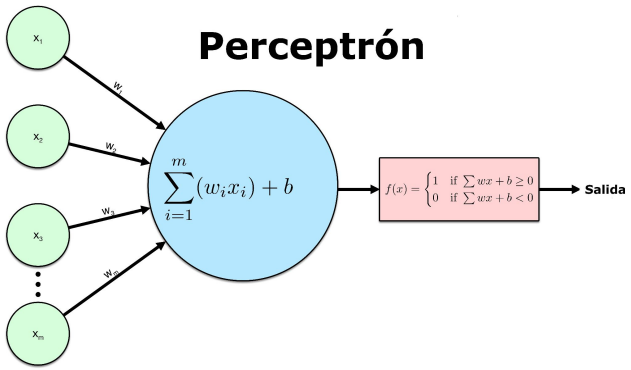
Cada capa aprende características cada vez más complejas.

El Bloque Básico: El Perceptrón

El Perceptrón es la forma más simple de una neurona artificial, creada en los años 50. Es el "átomo" del Deep Learning.

¿Cómo funciona?

1. Recibe varias entradas (features), cada una con un peso que indica su importancia.
2. Calcula la suma ponderada de las entradas (cada entrada multiplicada por su peso).
3. Pasa este resultado a una función de activación, que decide si la neurona se "activa" (ej. produce un 1) o no (ej. produce un 0).



El proceso de "aprender" consiste en encontrar los pesos correctos para tomar la decisión correcta.

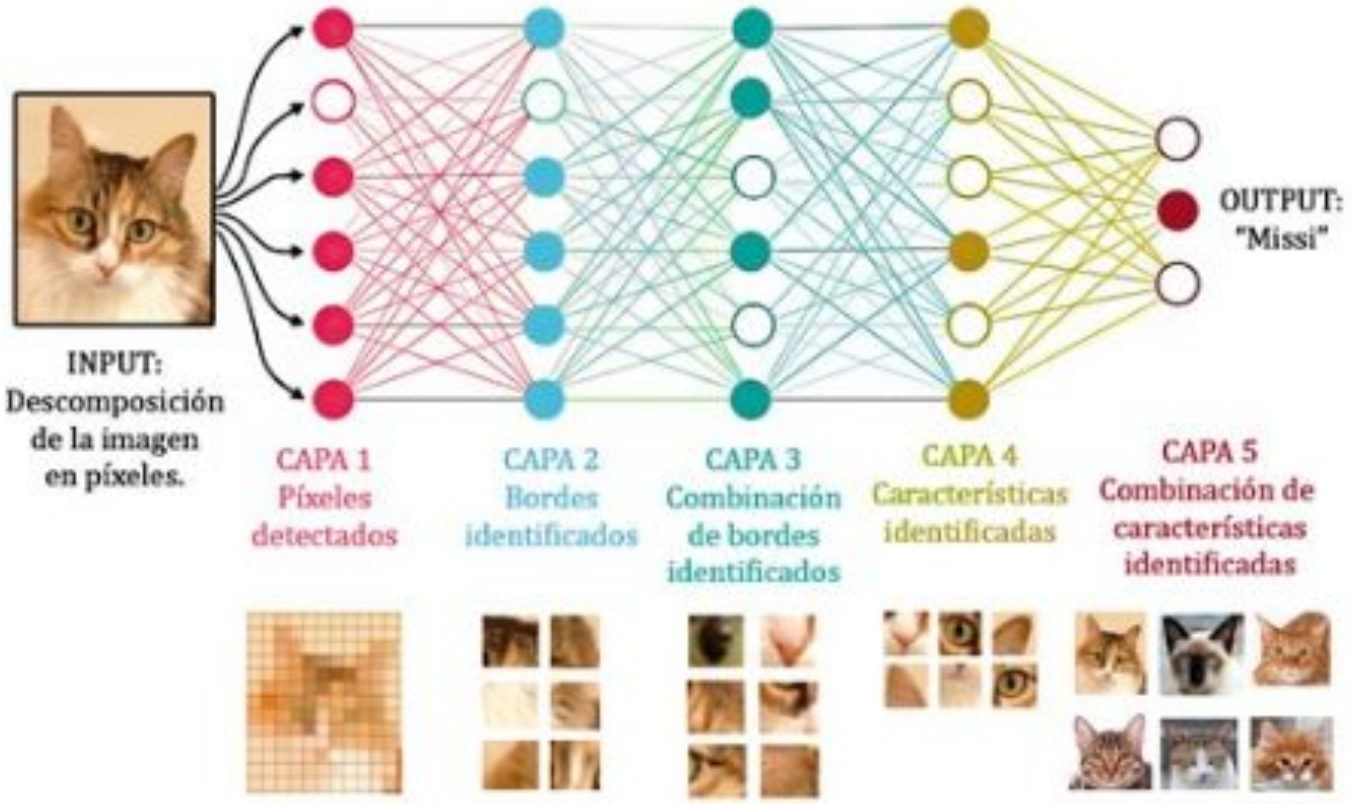
De una Neurona a una Red: Las Capas

Una sola neurona solo puede aprender patrones muy simples. El verdadero poder surge al conectar miles de ellas en capas, formando una Red Neuronal Artificial (RNA).

El tipo más común es la Red Feedforward, donde la información fluye en una sola dirección, hacia adelante.

- Capa de Entrada: Recibe los datos crudos (ej. los pixeles de una imagen). Hay una neurona por cada feature.
- Capas Ocultas: El "cerebro" de la red. Aquí es donde ocurre el aprendizaje de patrones complejos. Puede haber una o muchas.
- Capa de Salida: Produce la predicción final (ej. la probabilidad de que la imagen sea un "gato").

Red Feedforward



La Función de Activación: Dando No-Linealidad

La función de activación es un componente crucial. Si las neuronas solo sumaran sus entradas ponderadas, una red con 100 capas se comportaría igual que una simple regresión lineal. ¡No podría aprender patrones complejos!

La función de activación no-lineal permite a la red aprender relaciones increíblemente complejas entre las variables.

Funciones Populares:

- ReLU (Rectified Linear Unit): Es la más utilizada hoy en día. Es muy simple y computacionalmente eficiente.
- Sigmoide: Transforma cualquier número a un valor entre 0 y 1, lo que es muy útil para representar probabilidades en la capa de salida.

El Proceso de Aprendizaje (Entrenamiento)

El objetivo del entrenamiento de una red neuronal es encontrar el conjunto de pesos óptimo para todas las conexiones que minimice el error de sus predicciones.

El proceso es un ciclo:

1. Se inicializan todos los pesos con valores aleatorios pequeños.
2. Se le muestra a la red un lote de datos de entrenamiento y se le deja hacer una predicción (forward pass).
3. Se calcula qué tan errónea fue la predicción usando una función de pérdida (loss function).
4. Se usa un algoritmo optimizador (como el Descenso del Gradiente) para ajustar ligeramente cada peso de la red en la dirección que reduzca el error.
5. Se repite este ciclo miles de veces.

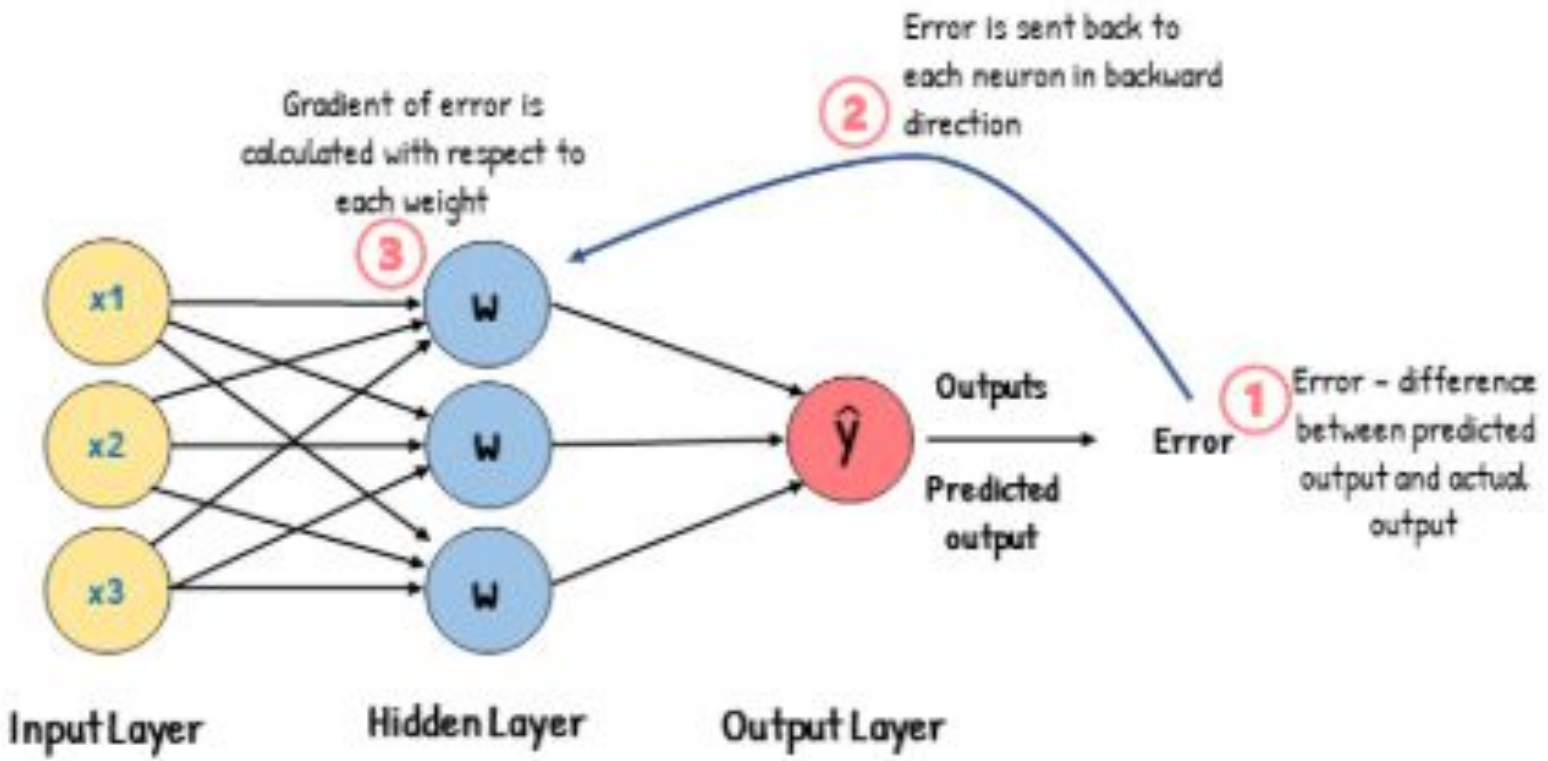
Backpropagation: El Algoritmo que Aprende (Conceptual)

Backpropagation (o retropropagación del error) es el algoritmo que hace posible el paso 4 del entrenamiento. Es el motor que impulsa el aprendizaje en casi todas las redes neuronales.

Analogía: "Repartiendo la Culpa"
Después de que la red hace una predicción y vemos el error final, Backpropagation funciona así:

Empieza en la capa de salida y calcula cuánta "culpa" tuvo cada neurona de esa capa en el error final.
Luego, va hacia atrás, a la capa anterior, y distribuye esa "culpa" entre las neuronas de esa capa, según qué tanto contribuyó cada una.
Repite este proceso hacia atrás hasta llegar a la primera capa.
Una vez que cada conexión de la red sabe su parte de "culpa", se ajusta su peso para reducir el error futuro.

Backpropagation: El Algoritmo que Aprende (Conceptual)



Aplicaciones del Deep Learning

Las redes neuronales profundas (Deep Learning) han revolucionado prácticamente todos los campos de la tecnología:

- Visión por Computador: Reconocimiento facial en tu celular, vehículos autónomos, diagnóstico de enfermedades a partir de imágenes médicas.
- Procesamiento de Lenguaje Natural (NLP): Traductores como Google Translate, asistentes de voz como Siri y Alexa, y los grandes modelos de lenguaje como GPT.
- Sistemas de Recomendación: Los algoritmos de Netflix, Spotify y Amazon que aprenden tus gustos y te sugieren contenido.
- Generación de Contenido: Creación de imágenes, música y texto con IA Generativa (ej. Midjourney, DALL-E).

Ventajas y Desventajas

Ventajas

- Rendimiento de Vanguardia: Son los mejores modelos para problemas complejos con datos no estructurados (imágenes, audio, texto).
- Aprendizaje de Características: No requieren que un humano diseñe las features; las capas profundas las aprenden automáticamente.

Desventajas

- Son una "Caja Negra" (Black Box): Es extremadamente difícil interpretar por qué una red tomó una decisión específica.
- Requieren Enormes Cantidades de Datos: Necesitan muchos ejemplos para aprender correctamente.
- Costo Computacional: Entrenar modelos grandes requiere hardware especializado (GPUs) y mucho tiempo.

Ejercicio 1

Un perceptrón debe decidir si un jugador es un "Objetivo Prioritario" (output=1). Tienes los siguientes datos y una función de activación de umbral simple. Calcula la salida.

- Entradas: `K/D Ratio = 2.5`, `SPM = 850`, `Es Lider de Escuadra = 1 (Sí)`
- Pesos: `w_kd = 0.5`, `w_spm = 0.002`, `w_lider = 0.2`
- Función de Activación: Si la suma ponderada es > 2.0, la salida es 1. Si no, es 0.

Proceso:

1. Multiplicar cada valor de entrada por su peso correspondiente.
2. Sumar los tres resultados para obtener la suma ponderada total.
3. Comparar la suma total con el umbral de 2.0.
4. Determinar la salida (0 o 1) basándose en la comparación.

Ejercicio 1

Un perceptrón debe decidir si un jugador es un "Objetivo Prioritario" (output=1). Tienes los siguientes datos y una función de activación de umbral simple. Calcula la salida.

- Entradas: `K/D Ratio = 2.5`, `SPM = 850`, `Es Lider de Escuadra = 1 (Sí)`
- Pesos: `w_kd = 0.5`, `w_spm = 0.002`, `w_lider = 0.2`
- Función de Activación: Si la suma ponderada es > 2.0, la salida es 1. Si no, es 0.

Proceso:

1. Multiplicar cada valor de entrada por su peso correspondiente.
2. Sumar los tres resultados para obtener la suma ponderada total.
3. Comparar la suma total con el umbral de 2.0.
4. Determinar la salida (0 o 1) basándose en la comparación.

UTEM

UNIVERSIDAD TECNOLÓGICA METROPOLITANA

del Estado de Chile