¡Les damos la bienvenida!

¿Comenzamos?

Unidad 4. NLP & Deep Learning aplicado a Ciencia de Datos

Redes Neuronales Básicas

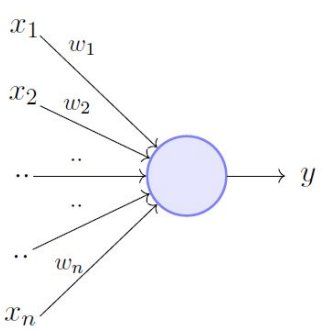
CODERHOUSE

Perceptrón y Perceptrón multi-capa

Perceptrón

Frank Rosenblatt, un psicólogo estadounidense, propuso el modelo de perceptrón clásico en 1958. Minsky y Papert (1969) lo refinaron y analizaron cuidadosamente; su modelo se conoce como modelo de perceptrón.

El modelo de perceptrón, propuesto por Minsky-Papert, es un modelo computacional más general que la neurona de McCulloch-Pitts.



$$y = 1 \quad if \sum_{i=1}^{n} w_i * x_i \geq \theta$$

$$= 0 \quad if \sum_{i=1}^{n} w_i * x_i < \theta$$

Rewriting the above,

$$y = 1 \quad if \sum_{i=1}^{n} w_i * x_i - \theta \geq 0$$

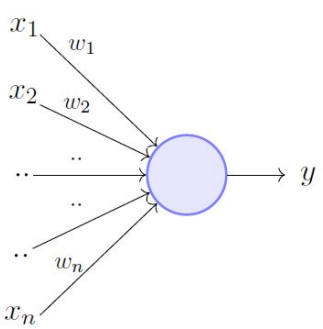
$$= 0 \quad if \sum_{i=1}^{n} w_i * x_i - \theta < 0$$

Fuente: Perceptron

Perceptrón

En este caso el output deseado se obtiene como una suma ponderada de las variables input.

A diferencia del algoritmo de neurona propuesto por McCulloch y Pitts los inputs ya no necesitan ser variables binarias lo cual lo hace un algoritmo generalizable y bastante útil en diversos contextos



$$y = 1 \quad if \sum_{i=1}^{n} w_i * x_i \geq \theta$$

$$= 0 \quad if \sum_{i=1}^{n} w_i * x_i < \theta$$

Rewriting the above,

$$y = 1 \quad if \sum_{i=1}^{n} w_i * x_i - \theta \geq 0$$

$$= 0 \quad if \sum_{i=1}^{n} w_i * x_i - \theta < 0$$

Fuente: Perceptron

Ejemplo en vivo

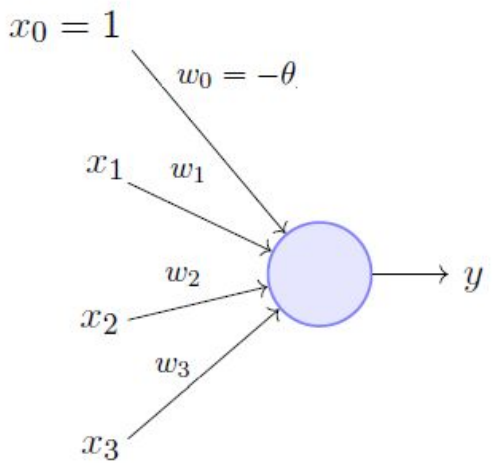
Veamos un ejemplo de Perceptrón

Duración: **10 minutos**

CODERHOUSE

Ejemplo

Consideremos la tarea de predecir si una persona verá un juego aleatorio o no. Utilizaremos 3 variables independientes binarias para simplificar el análisis

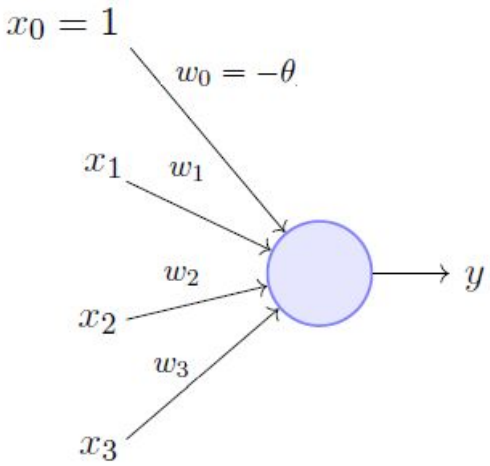


$$x_1 = isPremierLeagueOn$$
$$x_2 = isManUnitedPlaying$$
$$x_3 = isFriendlyGame$$

Fuente: Perceptron

Ejemplo

w1,w2 y w3 son los pesos por otro lado w0 se le conoce como el sesgo. Todos estos parámetros dependen de los datos. Mientras más grande (>**θ**) sea la suma ponderada mayor probabilidad tendremos de que una persona vea el partido



$$x_1 = isPremierLeagueOn$$
$$x_2 = isManUnitedPlaying$$
$$x_3 = isFriendlyGame$$

Fuente: Perceptron

Ventajas y desventajas

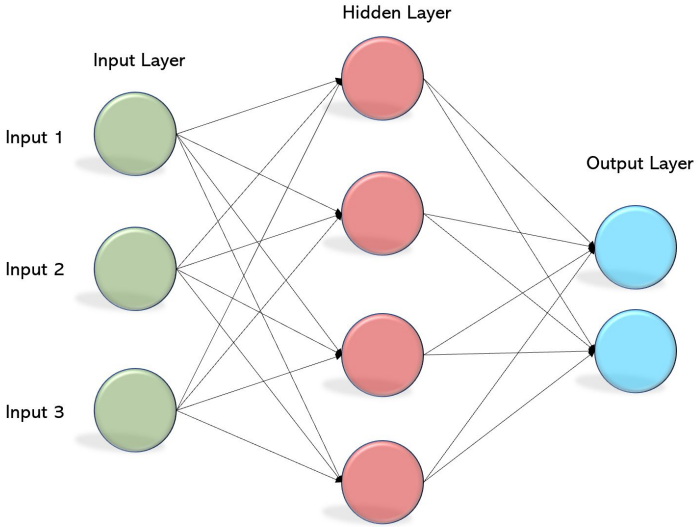
Ventajas	Desventajas
Tiene buenos fundamentos matemáticos	No se puede entender fácilmente la representación del conocimiento
Si la solución existe puede ser encontrada	Es un clasificador binario principalmente
Funciona bien con problemas bien definidos	No funciona bien con datos que no sean linealmente separables (problema XOR)
Funciona bastante bien a pesar de presentar ruido en los datos	No se puede actualizar el conocimiento del algoritmo si no es con entrenamiento

Perceptrón multicapa (MLP)

Perceptrón multicapa (MLP)

Un perceptrón multicapa (MLP) es una red neuronal artificial que genera un conjunto de salidas a partir de un conjunto de entradas.

Se caracteriza por varias capas de nodos de entrada conectados como un grafo dirigido entre las capas de entrada y salida. Utiliza backpropagation para entrenar la red ajustando los pesos correspondientes

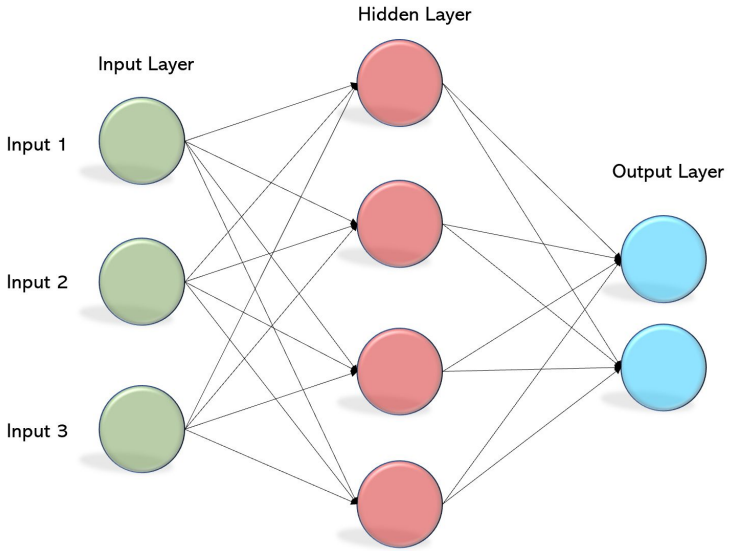


Fuente: Perceptron

Perceptrón multicapa (MLP)

Cada nodo excepto los de entrada, tiene una función de activación no lineal.

El MLP es un algoritmo que se usa ampliamente para resolver problemas que requieren aprendizaje supervisado. Tiene diversas aplicaciones que incluyen reconocimiento de voz, reconocimiento de imágenes y traducción automática.



Ejemplo en vivo

Veamos un ejemplo de Perceptrón multicapa

Duración: **10 minutos**

CODERHOUSE

Ejemplo

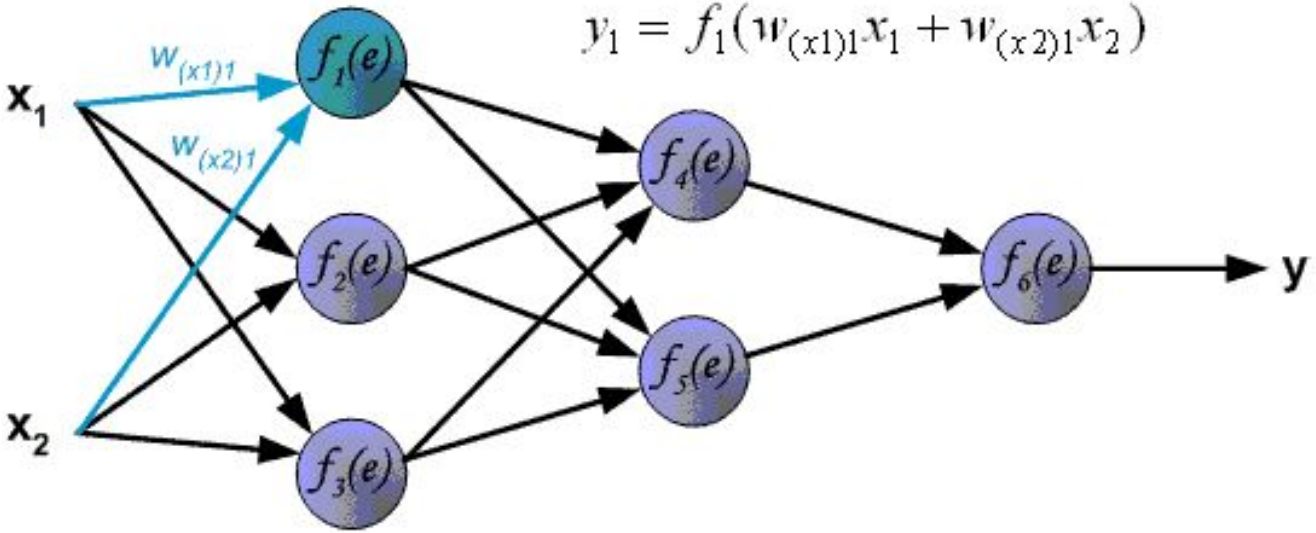
Consideremos la tarea de predecir el salario neto de una persona con base en dos variables (x1= cantidad de horas trabajadas y x2= horas extra)

En la figura de la derecha observamos la aplicación del algoritmo back-propagation con el fin de obtener los pesos correspondientes

Pasos involucrados en el algoritmo Back propagation:

1. **Propagación Forward:** en donde se propagan las funciones de activación desde el input hacia el output
2. **Propagación Backward:** cuando el error entre valores reales y los predichos para ajustar pesos y sesgo

Ejemplo FP



Ventajas y desventajas

Ventajas	Desventajas
Puede ser aplicado a problemas complejos no lineales	No se conoce hasta qué grado cada variable independiente es afectada por la dependiente
Funciona bien con grandes cantidades de datos	Los cálculos son difíciles de realizar y consumen bastante tiempo
Provee predicciones rápidas luego de entrenamiento	El funcionamiento propio del modelo depende de la calidad del entrenamiento
Mismo grado de precisión se puede alcanzar con menores cantidades de datos	No se comprende del todo como opera el mecanismo de entendimiento del problema

Para pensar

¿Si aumentamos la cantidad de capas ocultas en una red neuronal de Perceptrón multicapa, el error de clasificación de los datos de prueba siempre disminuye.?
Verdadero/Falso

Contesta en el chat de Zoom

Para pensar

Falso, no siempre se disminuye el error en el entrenamiento al aumentar las capas de una red neuronal, muchas veces se puede generar overfitting al aumentar las capas

Contesta en el chat de Zoom

Hands on Lab

A continuación utilizaremos el siguiente enlace para entender el funcionamiento de las redes neuronales.

Taller
Primera Red

Práctica integradora: Deep Learning

2da PRÁCTICA HACIA EL PROYECTO FINAL

Deep Learning

Consigna

- ✓ Considerando lo visto dentro de las clases de redes neuronales seleccionar un dataset de trabajo (puede ser nuevo o de los preparados para learning estilo MNIST).
- ✓ Cargar el dataset usando la función adecuada.
- ✓ Hacer una análisis exploratorio e identificar al menos tres insights sobre el mismo.
- ✓ Entrenar una red neuronal sencilla al menos dos capas (puede ser convolucional o recurrente).
- ✓ Plantear conclusiones.

Para trabajo final si se opta por Deep Learning debe adicionarse alguna de las siguientes tareas:

- ✓ Adicionar al menos dos capas a la red para mejorar su rendimiento.
- ✓ Comparar los resultados contra el modelo sencillo. Dimensionar mejoras.

CODERHOUSE

2da PRÁCTICA HACIA EL PROYECTO FINAL

Deep Learning

Formato

- ✓ Entregar un archivo con formato .ipynb. Debe tener el nombre “Datasets+Apellido.ipynb”.

Sugerencias

- ✓ Preparar el código y probar los resultados con distintas entradas

Entrega
de tu Proyecto final

PROYECTO FINAL

Data Science

Consigna:
El proyecto final constara de dos partes. Un mínimo requerido respecto a NLP que estará cubierto por la primera actividad hacia el PF, Deberás entregar un dataset del estilo texto (libro, paper, documento, colección de tweets, etc) donde se desarrollen de mínima dos de las tareas más usuales de preprocesamiento de NLP.
También un mínimo requerido de Deep Learning donde puedas construir tu primera red neuronal sencilla, lo que cubrirá la segunda actividad hacia el PF.

Para el TP Final deberás elegir una de las dos actividades hacia el PF y profundizar en el mismo, ya sea realizando un análisis de texto sobre la actividad de NLP o mejorando la red, via adición de capas, en el de Deep Learning.

CODERHOUSE

¿Preguntas?

¿Aún quieres conocer más?
Te recomendamos el siguiente material

MATERIAL AMPLIADO

Recursos Adicionales

Título

- ✓ El Perceptron Multicapa | **Interactive Chaos**
- ✓ Perceptron Multicapa | **IBM**
- ✓ CS230 deep learning| **Stanford**

Disponible en nuestro repositorio.

MATERIAL AMPLIADO

Recursos Adicionales

Título

- ✓ Python Bootcamp | **Pierian Data**
- ✓ Perceptron Multicapa | **Hugo Larochelle**
- ✓ CS231 deep learning| **Stanford**

Disponible en nuestro repositorio.

Educación digital
para el mundo real