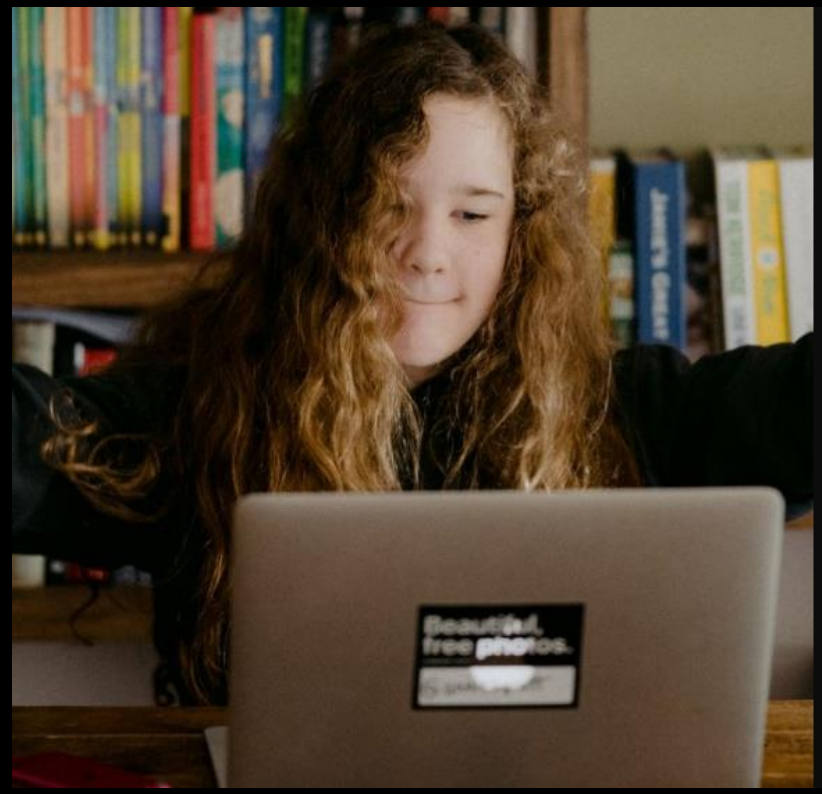
Semilleros
by DSRP

Ruta de Aprendizaje



COMPUTER VISION

- **6 lecciones** de Deep Learning
- **6 lecciones** de Computer Vision

Temario: Introduction to Deep Learning

Primera semana (Deep Learning)

- Intro to “Deep Learning”
- The Linear Unit

* Definición “Deep Learning”
* Librerías populares de Python
* Herramientas de ciencia de datos
* Nociones básicas
  - ¿Qué es una entrada?
  - ¿Qué es un peso?
  - ¿Qué es el sesgo?

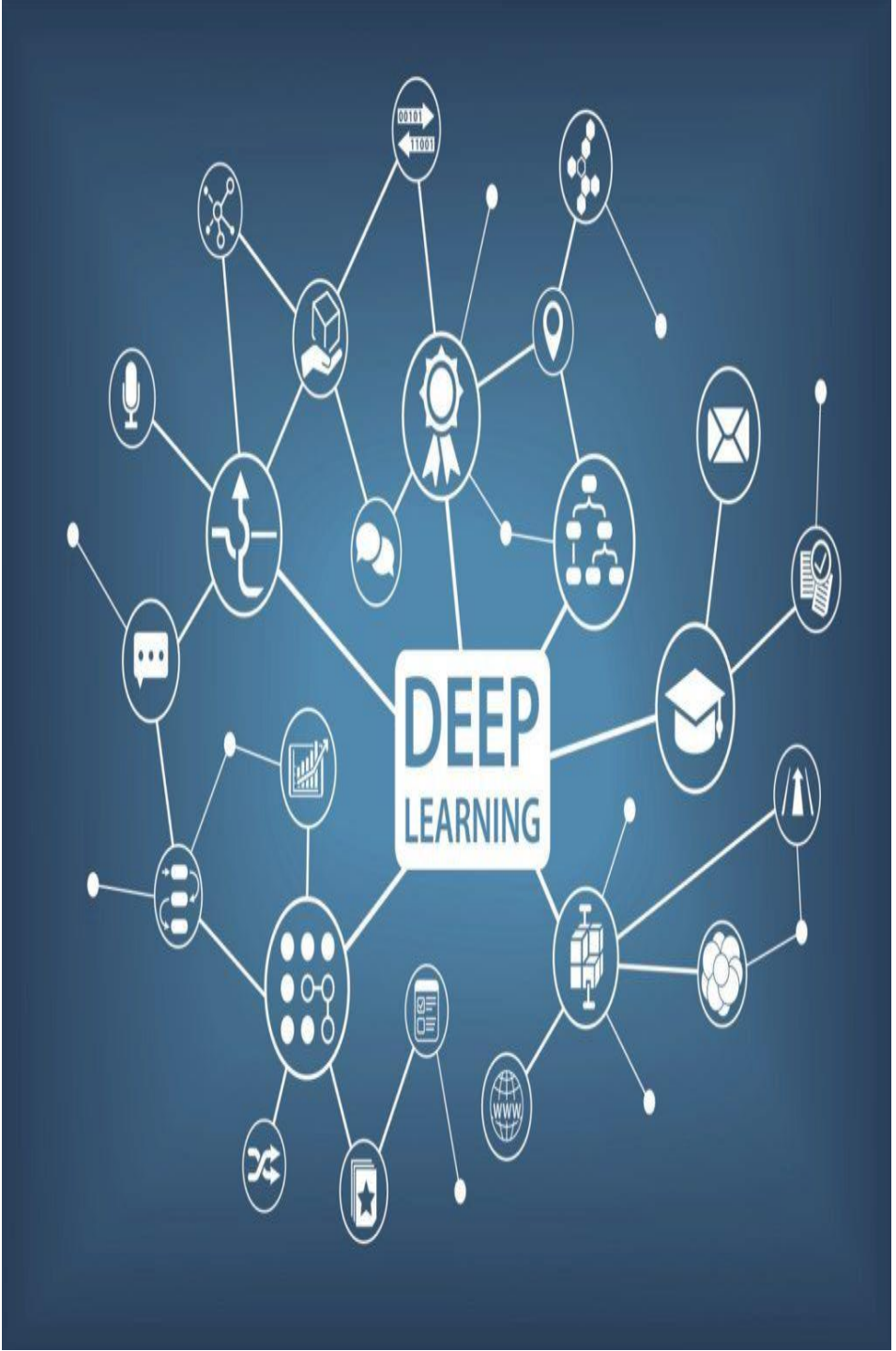
- Linear Units in Keras
- Ejercicios

- Nociones básicas
  - ¿Qué es una salida?
  - ¿Qué es “unit”?
  - ¿Qué es “input_shape”?
* Ejemplos en código

- Primer laboratorio (ejercicios básicos en códigos)

A single Neuron

Semilleros by DSRP

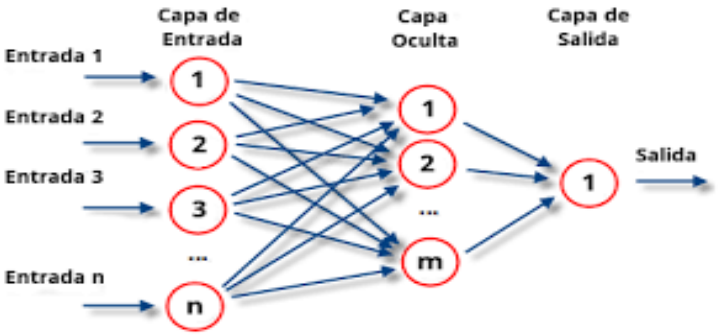


Lesson 1.1: Intro to “Deep Learning”

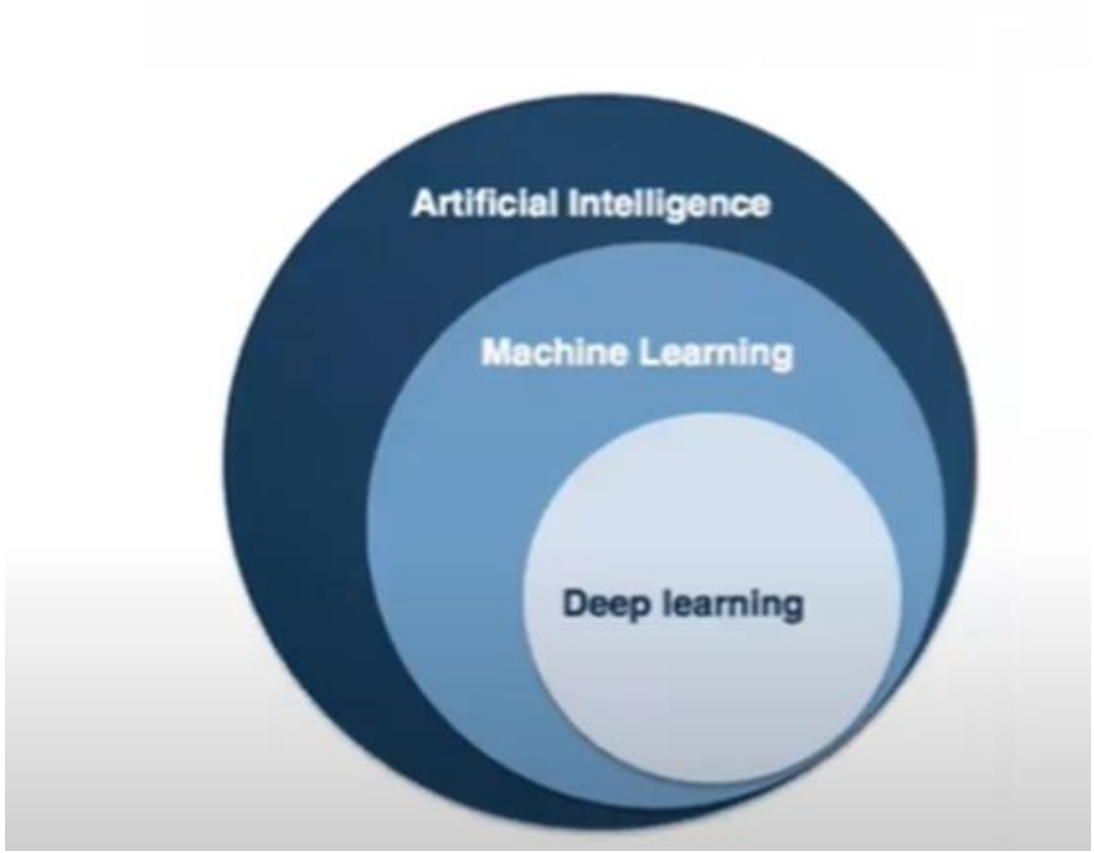
Deep Learning

El Deep learning es un tipo de Machine Learning que entrena a una computadora para que realice tareas como las hacemos los seres humanos, como el reconocimiento del habla, la identificación de imágenes o hacer predicciones.

En lugar de organizar datos para que se ejecuten a través de ecuaciones predefinidas, el deep learning configura parámetros básicos acerca de los datos y entrena a la computadora para que aprenda por cuenta propia reconociendo patrones mediante el uso de muchas capas de procesamiento.



Lesson 1.1: Intro to “Deep Learning”



La Inteligencia Artificial (IA) es la **combinación de algoritmos, la simulación de procesos de inteligencia humana por parte de máquinas**, en especial sistemas informáticos.

Inteligencia Artificial

Máquinas simulando el comportamiento y razonamiento de los humanos. Para ello, usan diferentes técnicas, entre ellas, Machine Learning.

Machine Learning

Es la capacidad de las computadoras para aprender por sí mismas a partir de datos y experiencia. En su uso más complejo utiliza Deep Learning.

Deep Learning

Algoritmos que permiten clasificar y relacionar grandes volúmenes de información imitando las redes neuronales.

Semilleros by DSRP

Lesson 1.1: Intro to “Deep Learning”

Librerías populares de Python

pandas

Contiene módulos de análisis y estructuras de datos de alto rendimiento

NumPy

Se ocupa de la computación científica y matemática con Python

learn

Biblioteca de software Machine Learning para análisis de datos, minería y Machine Learning

TensorFlow

Ayuda a crear módulos y funciones de Deep Learning, como detección / reconocimiento de imágenes

matplotlib

Matplotlib es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy.

Keras

A deep learning library

Keras es una biblioteca de Redes Neuronales de Código Abierto escrita en Python. Es capaz de ejecutarse sobre TensorFlow.

Herramientas de Ciencia de Datos

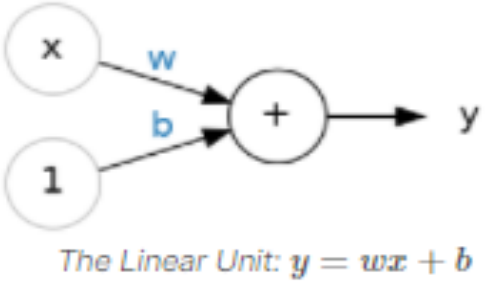
jupyter

SPYDER

colab

Semilleros by DSRP

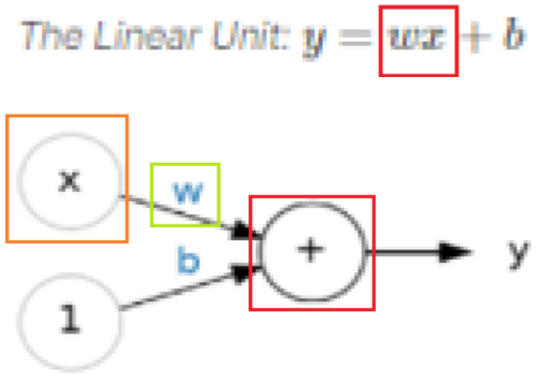
Lesson 1.2: The linear unit



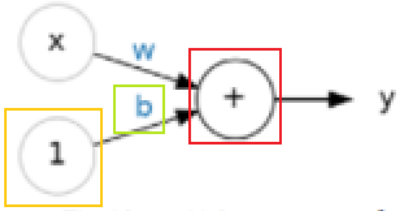
The Linear Unit: $y = wx + b$

x = entrada
w = weights (peso)
y = salida
b = bias (sesgo)

La entrada es **x**. Su conexión con la neurona tiene un peso que es **w**. Siempre que un valor fluya a través de una conexión, multiplique el valor por el peso de la conexión. Para la entrada **x**, lo que llega a la neurona es **w * x.** Una red neuronal "aprende" modificando sus pesos.



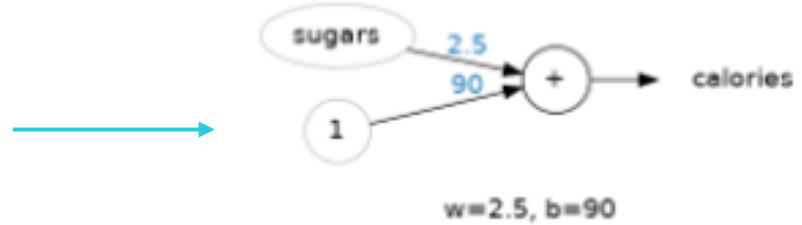
La **b** es un tipo especial de ponderación que llamamos **bias** (sesgo). El sesgo no tiene ningún dato de entrada asociado; en cambio ponemos un **1** en el diagrama para que el valor que llegue a la neurona sea simplemente b **(1 * b = b).**



La **y** es el valor que finalmente produce la neurona. Para obtener la salida, la neurona suma todos los valores que recibe a través de sus conexiones. La activación de esta neurona es **y = w * x + b**, o como fórmula **y = wx + b.**

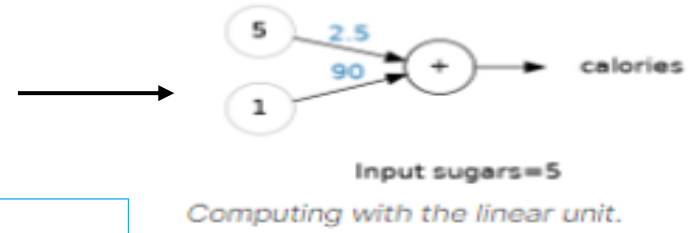
Lesson 1.3: Example – The linear unit as a model

Pensemos en cómo podría funcionar eso en un conjunto de datos de 80 cereales. Al entrenar un modelo con '**sugars**' (gramos de azúcares por porción) como entrada y '**calories**' (calorías por porción) como salida,



podríamos encontrar que el sesgo es **b = 90** y el peso es **w = 2.5**.

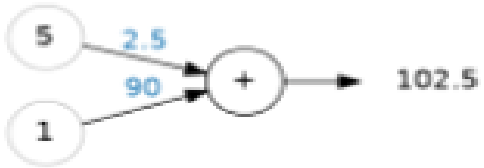
Podríamos estimar el contenido calórico de un cereal con 5 gramos de azúcar por ración así:



Computing with the linear unit.

Y, comparando nuestra fórmula, tenemos
calories = 2.5 × 5 + 90 = 102.5, tal como esperamos.

The Linear Unit: $y = wx + b$



Multiply inputs by weights and sum.

x = 5
w = 2.5
y = calories
b = 90

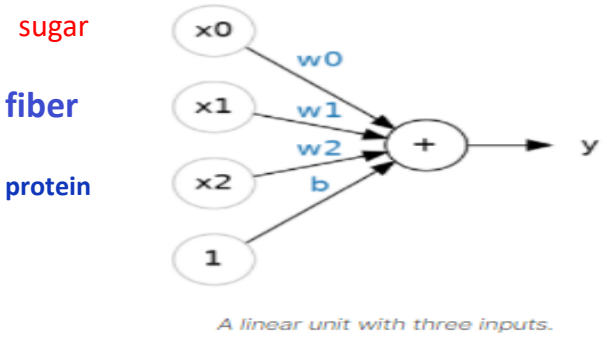
Semilleros by DSRP

Lesson 1.4: Multiple inputs

El conjunto de datos de 80 cereales tiene muchas más características que solo ‘sugars'. ¿Qué pasaría si quisiéramos expandir nuestro modelo para incluir elementos como el contenido de ‘fiber’ o ‘protein’?

Podemos simplemente agregar más conexiones de entrada a la neurona, una para cada característica adicional.

Para encontrar la salida, multiplicaríamos cada entrada por su peso de conexión y luego las sumaríamos todas.



A linear unit with three inputs.

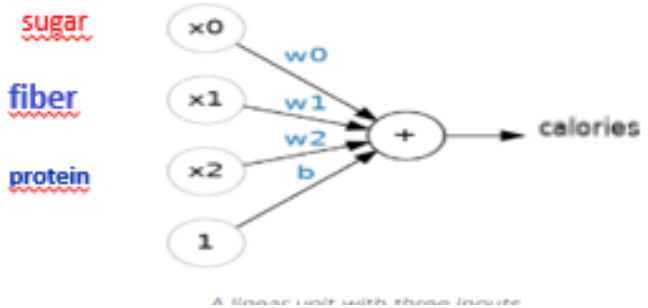
La fórmula para esta neurona sería **y = w0x0 + w1x1 + w2x2 + b**.

Una unidad lineal con dos entradas encajará en un plano, y una unidad con más entradas encajará en un hiperplano

Lesson 1.5: Linear units in Keras

La forma más sencilla de crear un modelo en Keras es a través de **keras.Sequential**, que crea una red neuronal como una pila de capas.

Podríamos definir un modelo lineal aceptando tres características de entrada (‘sugars', ‘fiber' y 'protein’) y produciendo una única salida ('caloríes') así:




```
from tensorflow import keras
from tensorflow.keras import layers

# Create a network with 1 linear unit
model = keras.Sequential([
    layers.Dense(units=1, input_shape=[3])
])
```

input_shape = representa las entradas ‘sugars’, ‘fiber’ y ‘protein’
units = representa la salida ‘calories’

Ejercicio:

Si tienes 6 entradas: ‘dolor de cabeza’, ‘fiebre’, ‘tos’, ‘erupción cutánea’, ‘conjuntivitis’ y ‘náuseas’
1 salida: ‘diagnóstico’

input_shape=[**?**]
units=**?**

Semilleros by DSRP

LABORATORIO 1

Semilleros
by DSRP