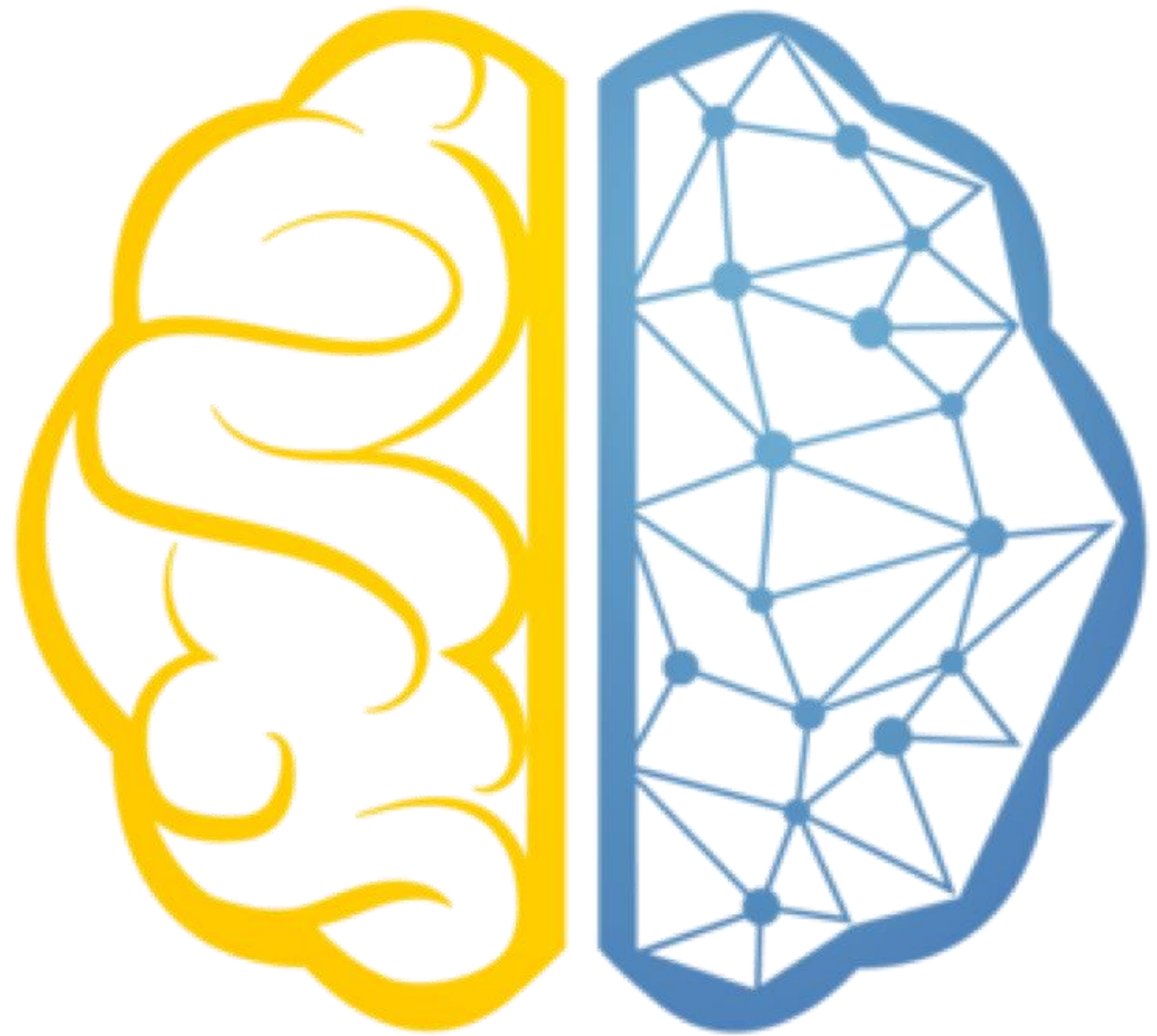


Inteligencia Artificial

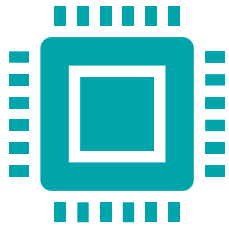
Ejercicios Practicos - Paso a Paso

Comparte con la comunidad:

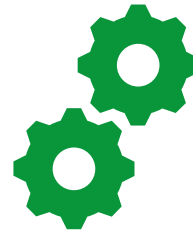
<https://datadosis.com>



Que es Inteligencia Artificial



La inteligencia artificial/aprendizaje de máquinas no sólo significa robots o películas de ciencia ficción!

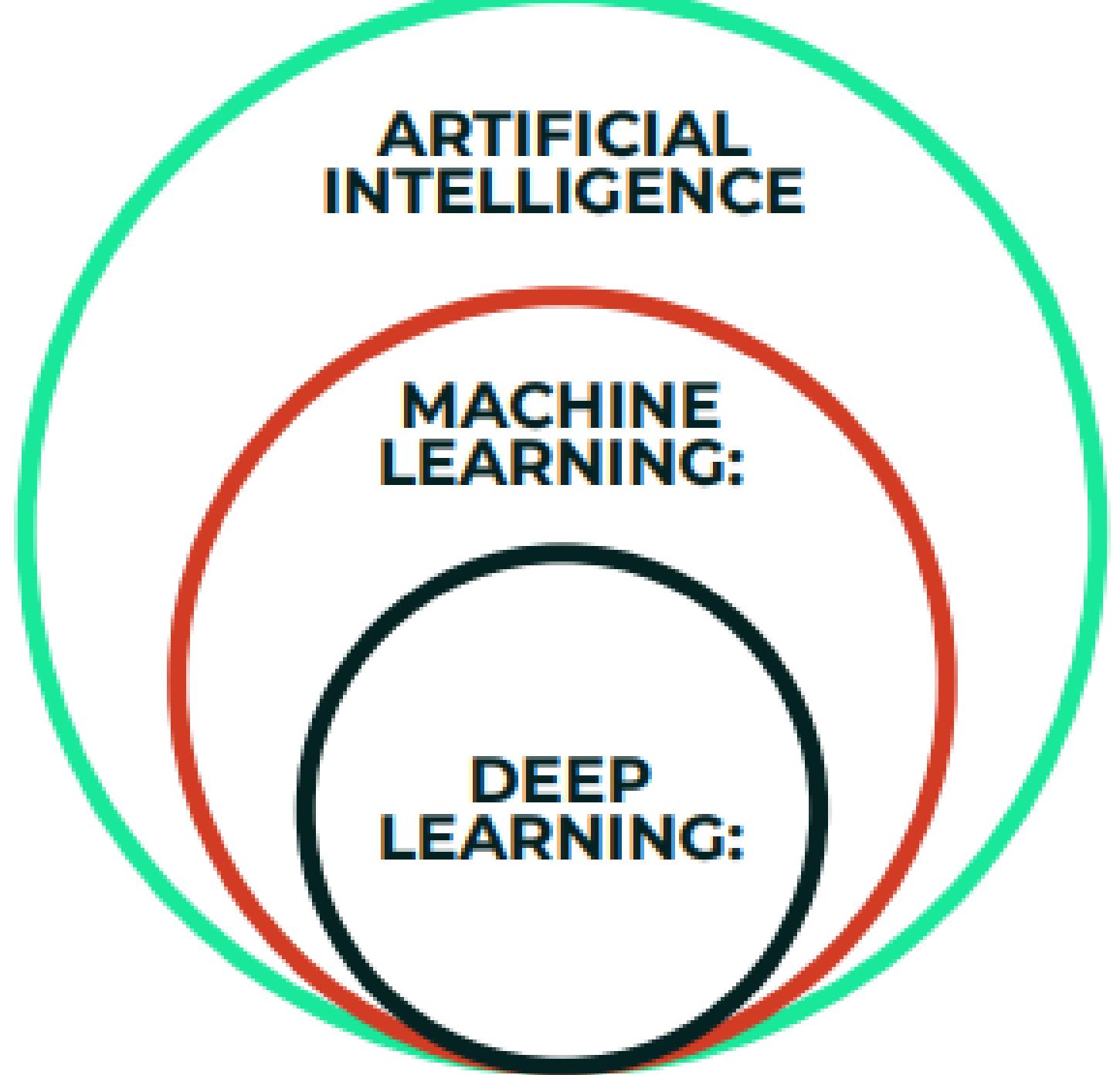


¡Las máquinas y las aplicaciones de aprendizaje profundo están en todas partes!




El motor de búsqueda de Google, los sistemas de recomendación de la Amazon, el reconocimiento facial de Facebook (etiquetado), Siri y mucho mas!

IA vs ML vs DL



Inteligencia Artificial

Ciencia que permite a las computadoras imitar la inteligencia humana, como la toma de decisiones, el procesamiento de textos y la percepción visual.



La IA es un campo más amplio (es decir: el gran paraguas) que contiene varios subcampos como el aprendizaje de la máquina, la robótica y la visión por ordenador.

Machine Learning

El aprendizaje de las máquinas es un subcampo de la Inteligencia Artificial que permite a las máquinas mejorar en una tarea determinada con la experiencia.

Es importante señalar que todas las técnicas de aprendizaje de máquinas se clasifican como de Inteligencia Artificial.

Sin embargo, no toda la Inteligencia Artificial podría contar como Aprendizaje Automático ya que algunos motores básicos basados en reglas podrían clasificarse como IA pero no aprenden de la experiencia por lo que no pertenecen a la categoría de aprendizaje automático.

Deep Learning

El Aprendizaje Profundo es un campo especializado del Aprendizaje Automático que se basa en el entrenamiento de Redes Neuronales Artificiales Profundas (RNA) utilizando grandes conjuntos de datos como imágenes.



Las RNA son modelos de procesamiento de información inspirados en el cerebro humano.



El cerebro humano está formado por miles de millones de neuronas que se comunican entre sí mediante señales eléctricas y químicas y permiten a los humanos ver, sentir y tomar decisiones.

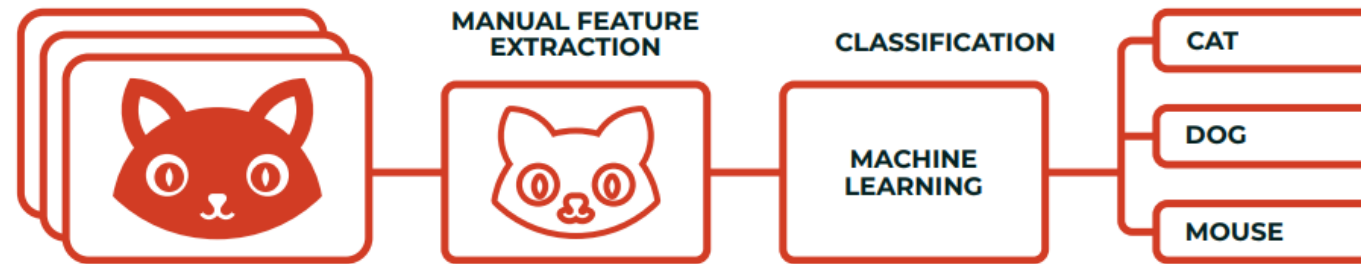


ML vs DL

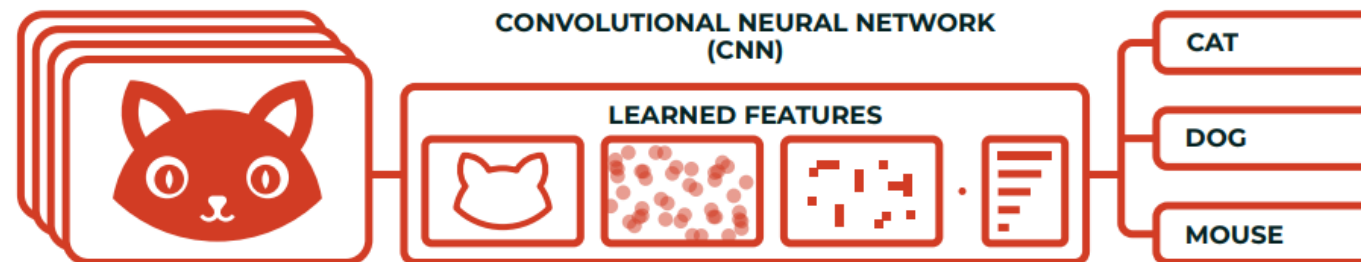
- Lo que diferencia el aprendizaje profundo de las técnicas de aprendizaje de las máquinas es su capacidad para extraer características automáticamente:
 - El proceso de aprendizaje de la máquina:
 - (1) seleccionar el modelo a entrenar,
 - (2) realizar manualmente la extracción de características.
 - Proceso de aprendizaje profundo:
 - (1) Seleccionar la arquitectura de la red,
 - (2) las características se extraen automáticamente alimentando los datos de entrenamiento (como imágenes) junto con la clase de objetivo (etiqueta).

ML vs DL

MACHINE LEARNING



DEEP LEARNING

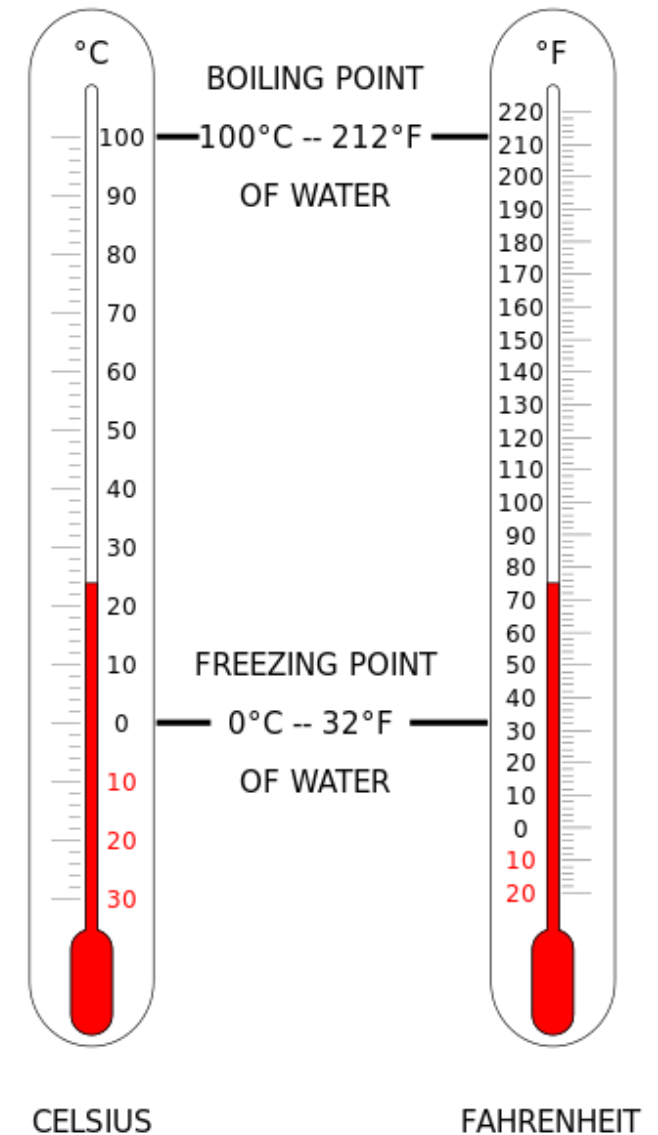


Ejercicios de Inteligencia Artificial

Proyecto 1: Convertir Celsius a Fahrenheit

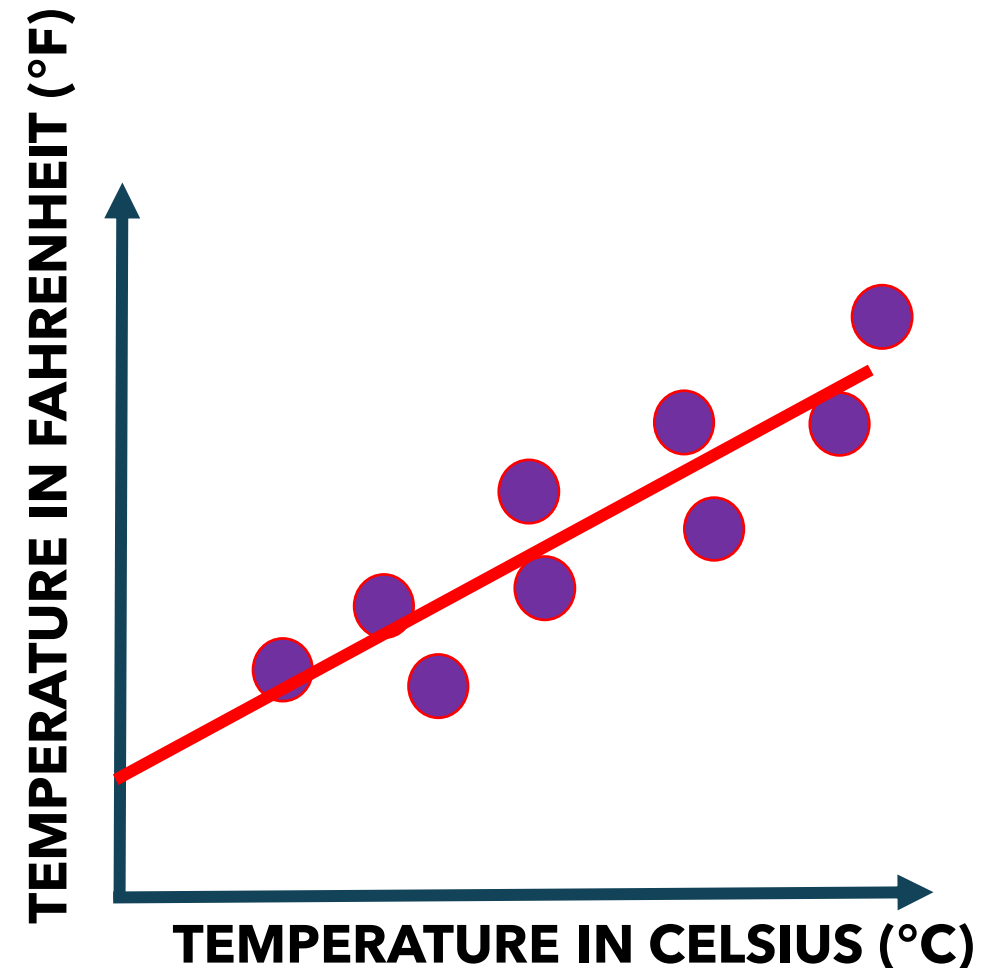
- En este proyecto, construiremos un simple modelo de aprendizaje de máquinas para convertir las temperaturas de Celsius a Fahrenheit.
- La ecuación es la siguiente:
 - $T(^{\circ}\text{F}) = T(^{\circ}\text{C}) \times \frac{9}{5} + 32$
- Por ejemplo, convirtamos la temperatura de 0°C en Fahrenheit:
 - $T(^{\circ}\text{F}) = (0^{\circ}\text{C} \times \frac{9}{5}) + 32 = 32^{\circ}\text{F}$

Fuentes: https://commons.wikimedia.org/wiki/File:Thermometer_CF.svg



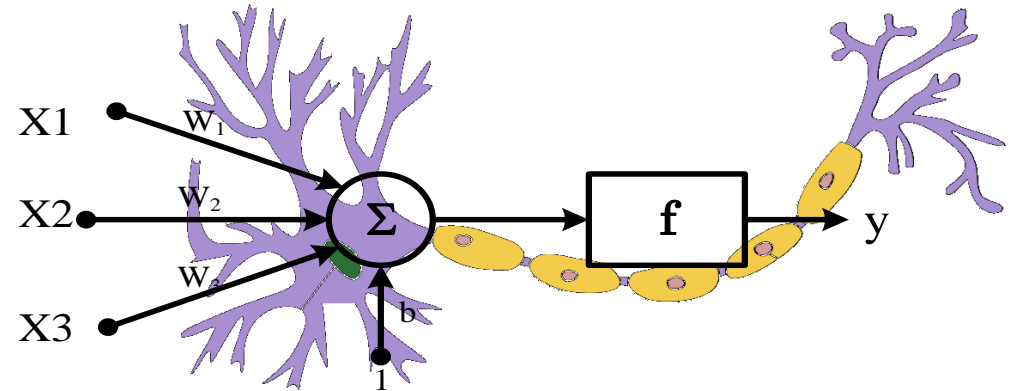
Proyecto 1: Convertir Celsius a Fahrenheit

- El objetivo es predecir el valor de una variable Y en base a otra variable X .
- X se llama la variable independiente e Y se llama la variable dependiente.
- Esto se llama "Regresión" y se tratará con mucho más detalle en secciones posteriores del curso.

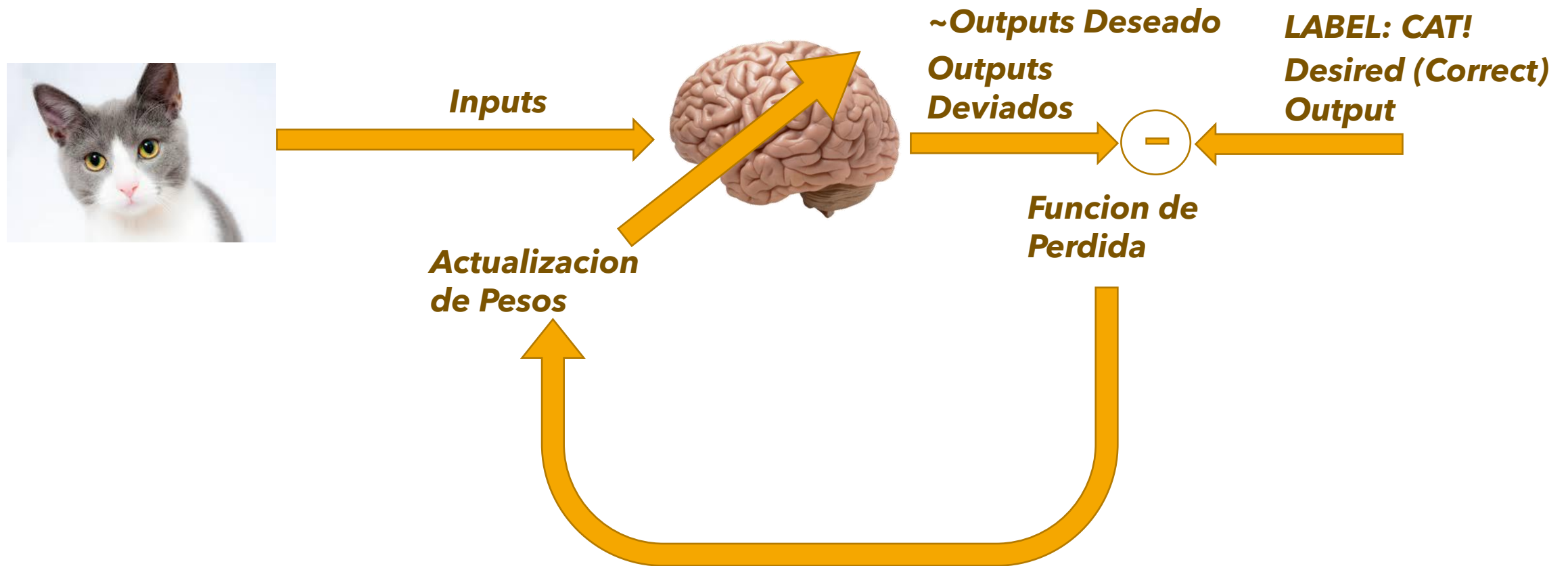


Que son las Redes Neuronales y Como Aprenden?

- El cerebro tiene más de 100 mil millones de neuronas que se comunican a través de señales eléctricas y químicas.
- Las neuronas se comunican entre sí y nos ayudan a ver, pensar y generar ideas. El cerebro humano aprende creando conexiones entre estas neuronas.
- Las RNA son modelos de procesamiento de información inspirados en el cerebro humano.

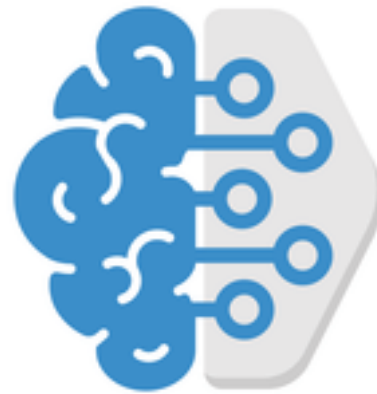


Proyecto 1: Que son las RNA y Como Aprenden



CONSTRUYAMOS LA RED MÁS SIMPLE!

"UN MODELO DE UNA SOLA NEURONA"

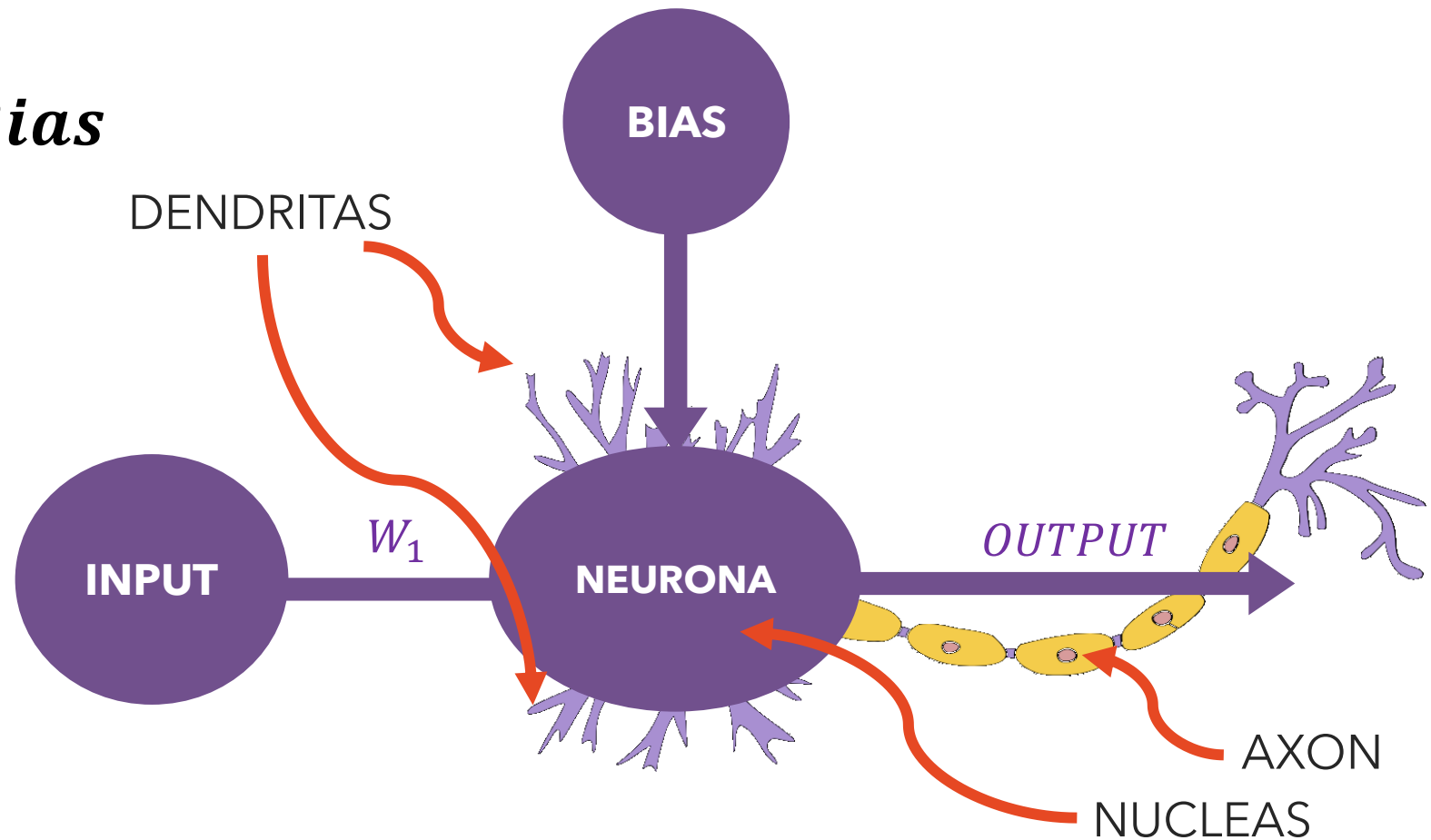


Proyecto 1: Modelo de 1 Neurona

$$Output = Input * W_1 + Bias$$

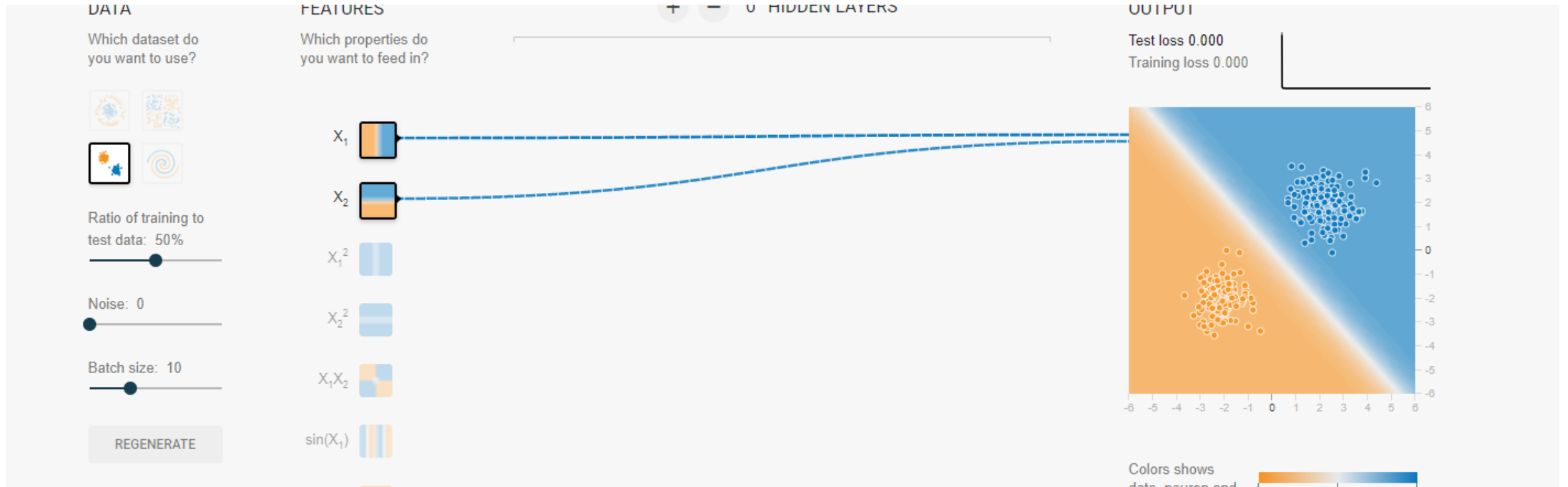
$$T(^{\circ}F) = T(^{\circ}C) * W_1 + Bias$$

$$T(^{\circ}F) = T(^{\circ}C) * 9/5 + 32$$



PLAYGROUND:

- Chequea esto: <https://playground.tensorflow.org/>



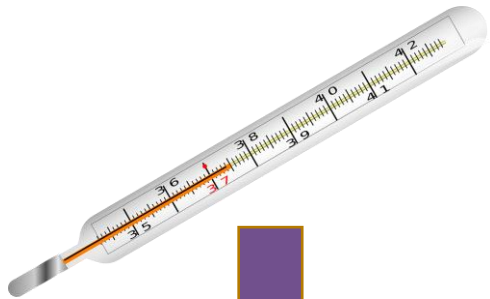
Proyecto #2: Predicciones de Ingresos

- Eres dueño de un negocio de helados y quieres crear un modelo para predecir los ingresos diarios en dólares basados en la temperatura (degC).
- Decidiste construir una simple Red Neural Artificial para resolver este problema.
- Conjunto de datos:
 - Entrada (X): Temperatura del aire exterior
 - Salida (Y): Ingresos diarios totales generados en dólares



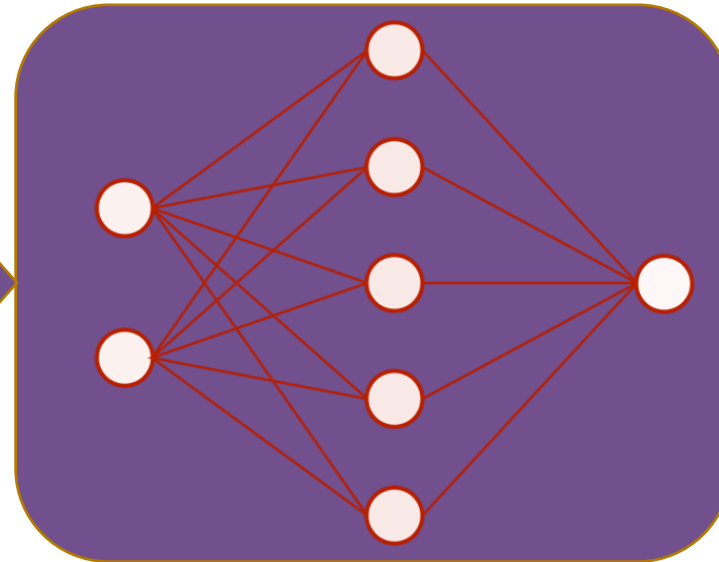
Proyecto #2: Porque es Importante?

MIDE LA TEMPERATURA



TEMPERATURA

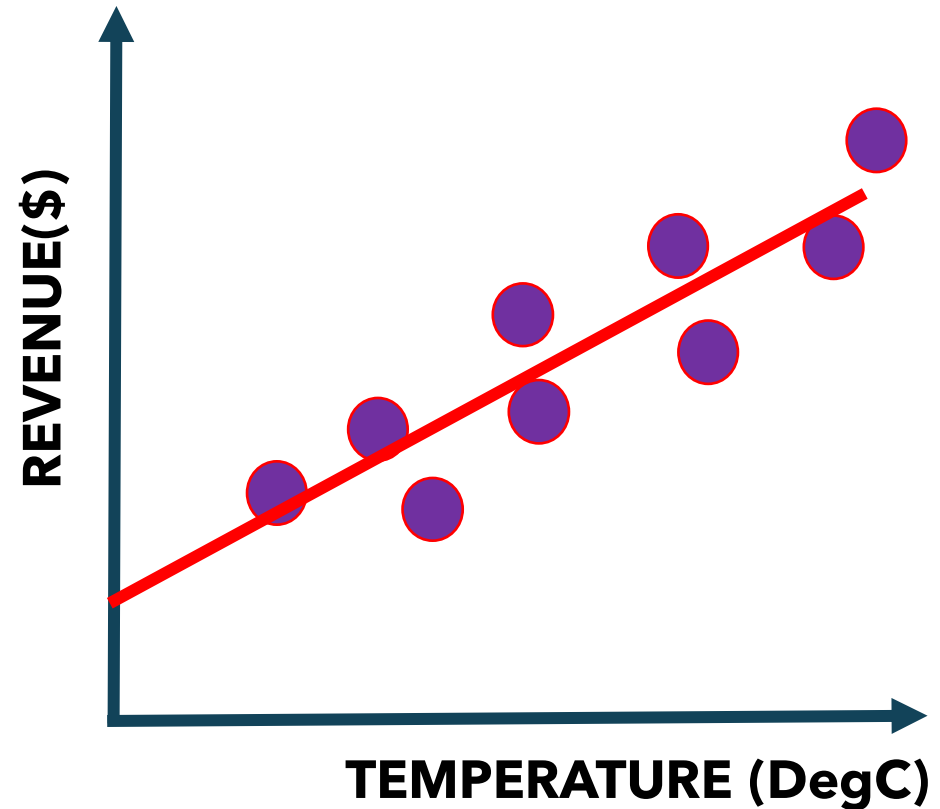
MODELO ENTRENADO



PREDICCIONES DE GANANCIAS

Proyecto #2: Predicciones de Ingresos

	Temperature	Revenue
0	24.566884	534.799028
1	26.005191	625.190122
2	27.790554	660.632289
3	20.595335	487.706960
4	11.503498	316.240194
5	14.352514	367.940744
6	13.707780	308.894518
7	30.833985	696.716640
8	0.976870	55.390338
9	31.669465	737.800824
10	11.455253	325.968408
11	3.664670	71.160153



Proyecto #3: Predecir Rentas de Bicicleta (Regresion)

- El objetivo es predecir el uso del alquiler de bicicletas basado en datos como la temperatura, la humedad, la velocidad del viento, etc.
- Este problema es mucho más avanzado comparado con proyectos anteriores, así que tenemos que construir una red de percepciones de múltiples capas.



Proyecto #3: Predecir Rentas de Bicicleta (Regresion)

- Entradas (Inputs):
 - instantánea: índice de registro
 - fecha: fecha
 - temporada: temporada (1: primavera, 2: verano, 3: otoño, 4: invierno)
 - año: año (0: 2011, 1: 2012)
 - mes: mes (1 a 12)
 - hr: hora (0 a 23)
 - día festivo: si el día es festivo o no - día de la semana: día de la semana
 - Día laborable: si el día no es ni fin de semana ni festivo es 1, de lo contrario es 0.

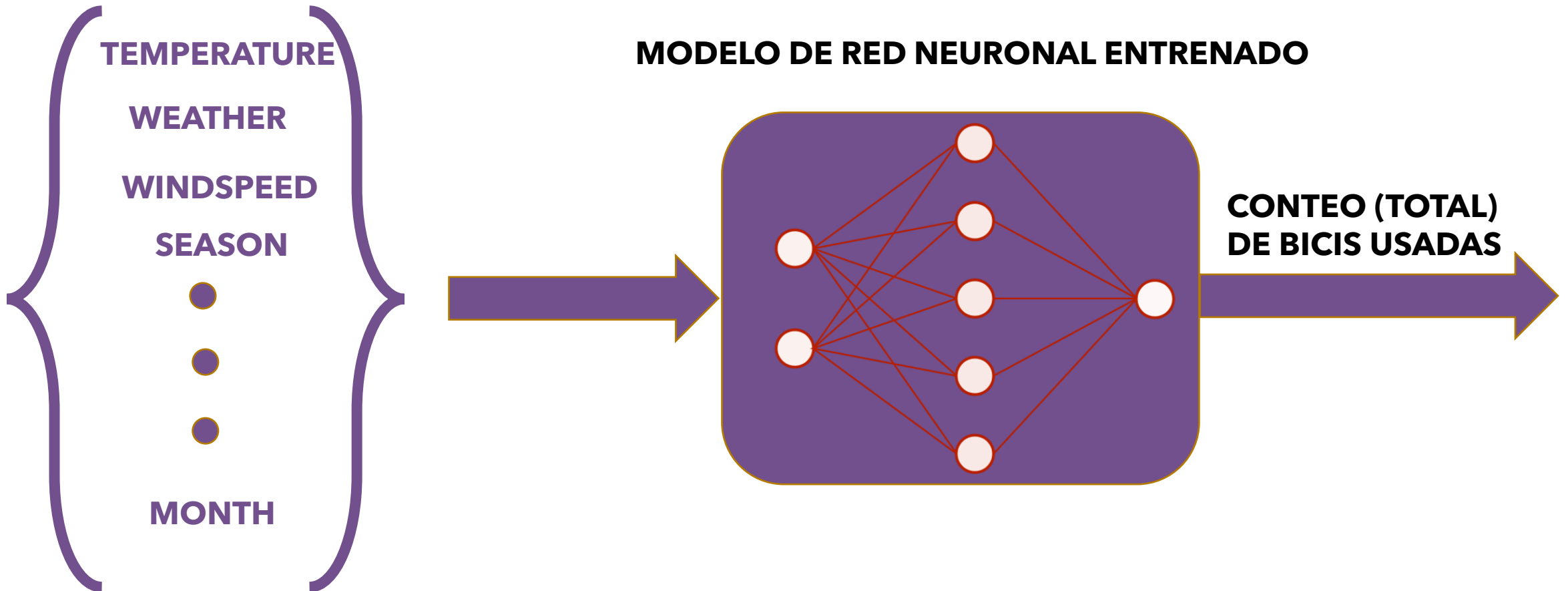
Proyecto #3: Predecir Rentas de Bicicleta (Regresion)

- Inputs (continuación):
 - weathersit :
 - 1: Claro, pocas nubes, parcialmente nublado
 - 2: Niebla + Nubes, Niebla + Nubes rotas, Niebla + Pocas nubes, Niebla
 - 3: Nieve ligera, Lluvia ligera + Tormenta eléctrica + Nubes dispersas, Lluvia ligera + Nubes dispersas
 - 4: Lluvia fuerte + paletas de hielo + tormenta eléctrica + niebla, nieve + niebla
 - temp : Temperatura normalizada en Celsius. Los valores se dividen a 41 (máximo)

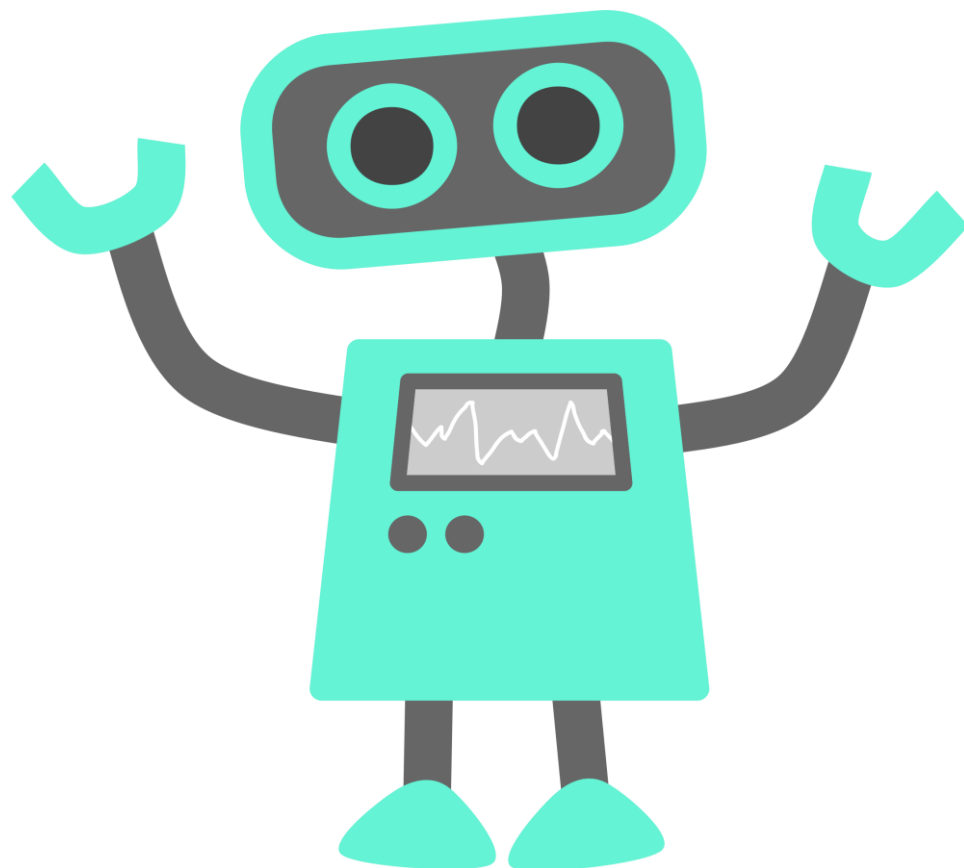
Proyecto #3: Predecir Rentas de Bicicleta (Regresion)

- Inputs (continuación):
 - Velocidad del viento: Velocidad del viento normalizada. Los valores se dividen a 67 (máximo)
- Salidas:
 - Casuales: recuento de usuarios casuales
 - registrado: recuento de usuarios registrados
 - cnt: recuento del total de bicicletas de alquiler incluyendo tanto las casuales como las registradas

Proyecto #3: Predecir Rentas de Bicicleta (Regresion)

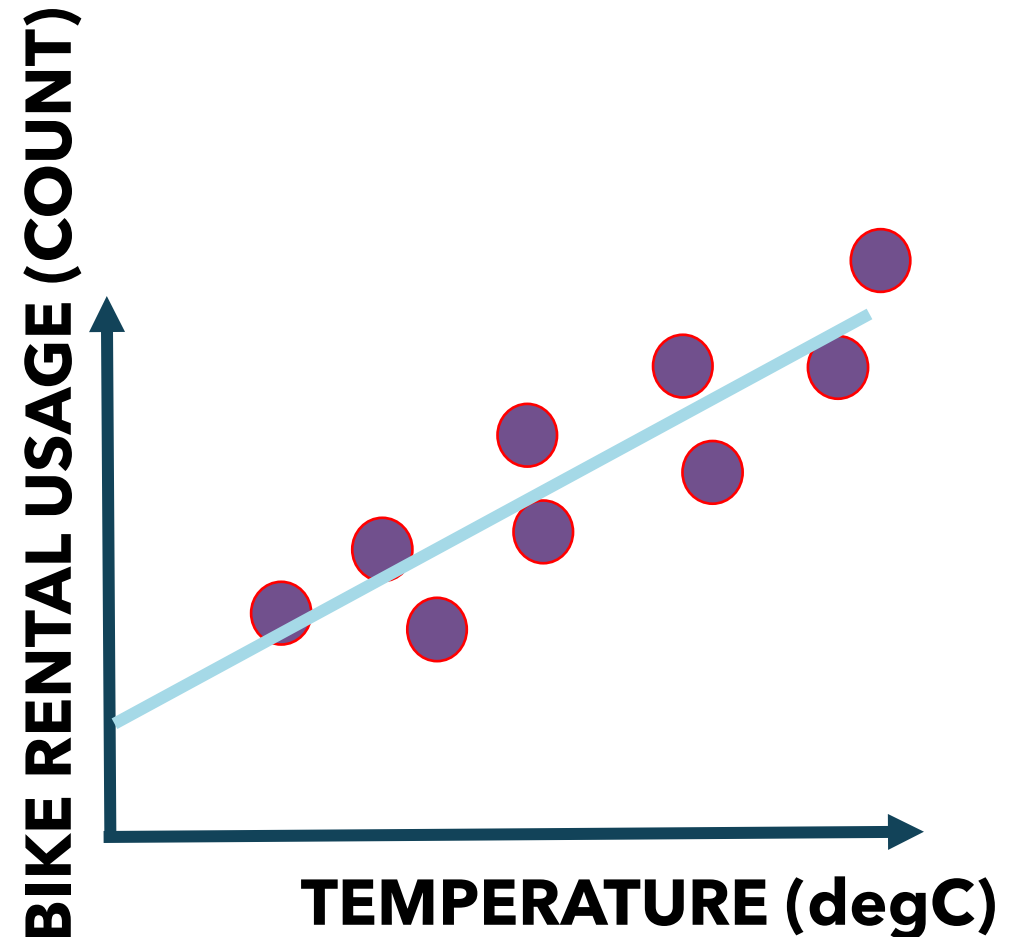


Que es Regresion?



Proyecto #3: Que es Regresion?

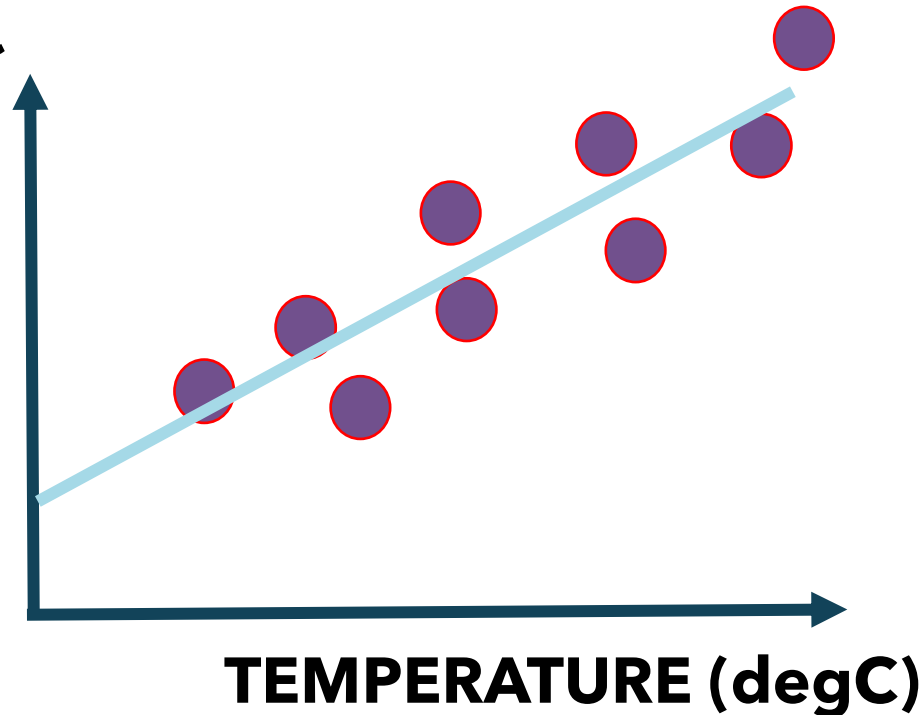
- La regresión funciona prediciendo el valor de una variable Y basada en otra variable X.
- X se llama la variable independiente e Y se llama la variable dependiente.



Proyecto #3: Que es Regresion?

	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	hum	windspeed	casual	registered	cnt
0	1/1/2011	1	0	1	0	6	0	2	0.344167	0.805833	0.160446	331	654	985
1	1/2/2011	1	0	1	0	0	0	2	0.363478	0.696087	0.248539	131	670	801
2	1/3/2011	1	0	1	0	1	1	1	0.196364	0.437273	0.248309	120	1229	1349
3	1/4/2011	1	0	1	0	2	1	1	0.200000	0.590435	0.160296	108	1454	1562
4	1/5/2011	1	0	1	0	3	1	1	0.226957	0.436957	0.186900	82	1518	1600
5	1/6/2011	1	0	1	0	4	1	1	0.204348	0.518261	0.089565	88	1518	1606
6	1/7/2011	1	0	1	0	5	1	2	0.196522	0.498696	0.168726	148	1362	1510
7	1/8/2011	1	0	1	0	6	0	2	0.165000	0.535833	0.266804	68	891	959
8	1/9/2011	1	0	1	0	0	0	1	0.138333	0.434167	0.361950	54	768	822
9	1/10/2011	1	0	1	0	1	1	1	0.150833	0.482917	0.223267	41	1280	1321
10	1/11/2011	1	0	1	0	2	1	2	0.169091	0.686364	0.122132	43	1220	1263
11	1/12/2011	1	0	1	0	3	1	1	0.172727	0.599545	0.304627	25	1137	1162
12	1/13/2011	1	0	1	0	4	1	1	0.165000	0.470417	0.301000	38	1368	1406
13	1/14/2011	1	0	1	0	5	1	1	0.160870	0.537826	0.126548	54	1367	1421
14	1/15/2011	1	0	1	0	6	0	2	0.233333	0.498750	0.157963	222	1026	1248
15	1/16/2011	1	0	1	0	0	0	1	0.231667	0.483750	0.188433	251	953	1204
16	1/17/2011	1	0	1	1	1	0	2	0.175833	0.537500	0.194017	117	883	1000
17	1/18/2011	1	0	1	0	2	1	2	0.216667	0.861667	0.146775	9	674	683

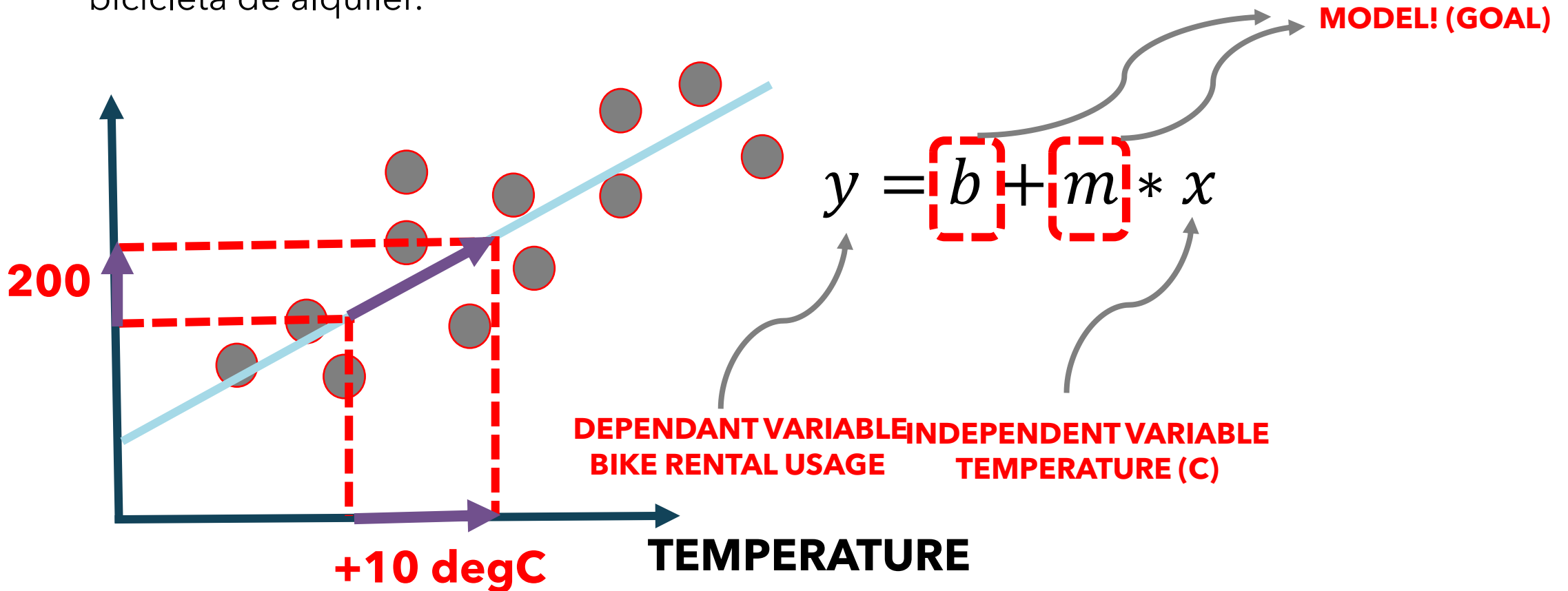
BIKE RENTAL USAGE (COUNT)



PROYECTO #3: MATEMÁTICAS DE REGRESIÓN

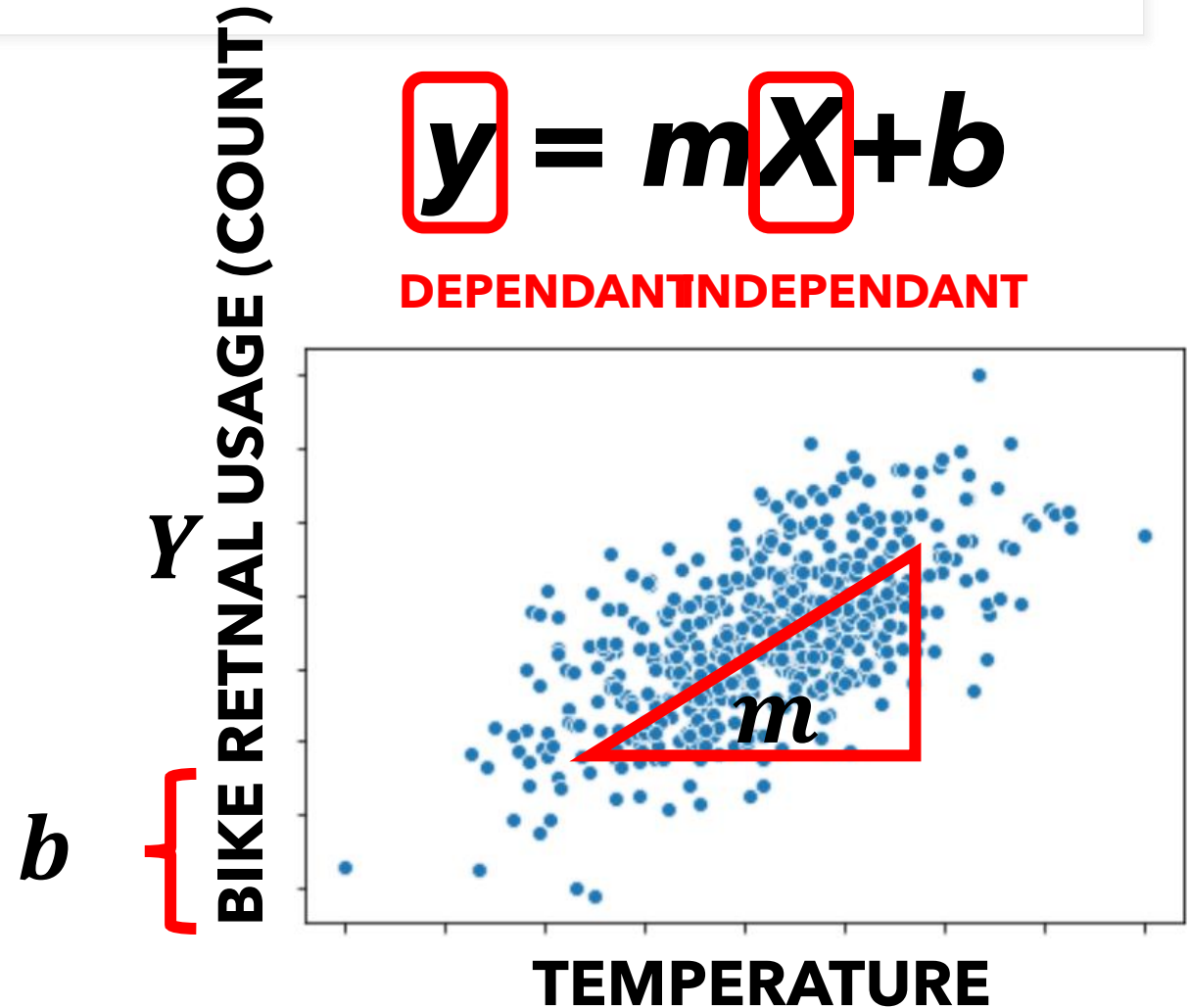
- El objetivo es obtener una relación (modelo) entre la temperatura y el uso de la bicicleta de alquiler.


BIKE RENTAL USAGE (COUNT)



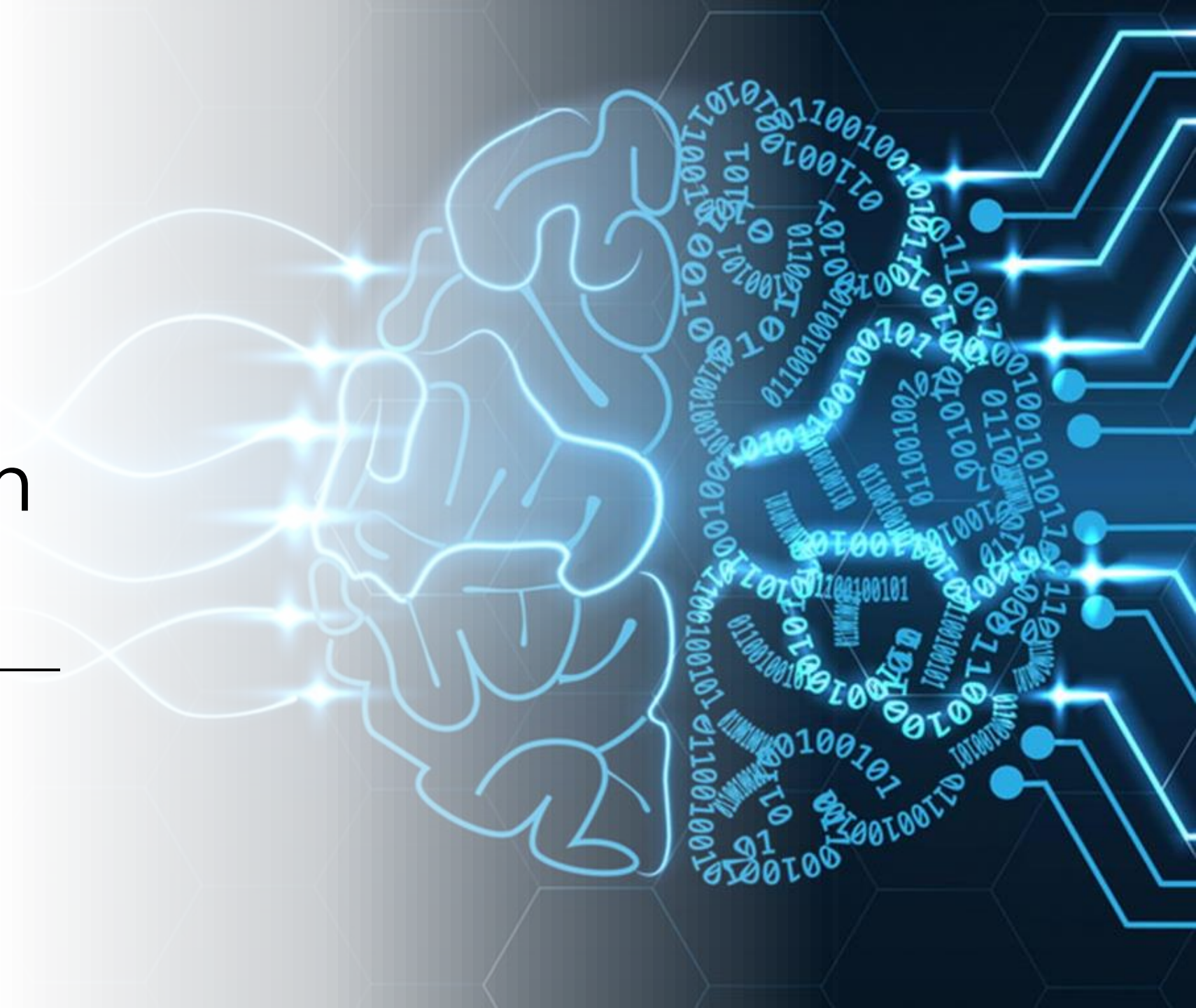
Proyecto #3: Como Usaremos el Modelo

- Una vez obtenidos los coeficientes 'm' y 'b', ¡has obtenido un modelo de regresión!
- Este modelo "entrenado" puede ser usado más tarde para predecir el monto de compra de un auto (dólares) basado en el salario anual.





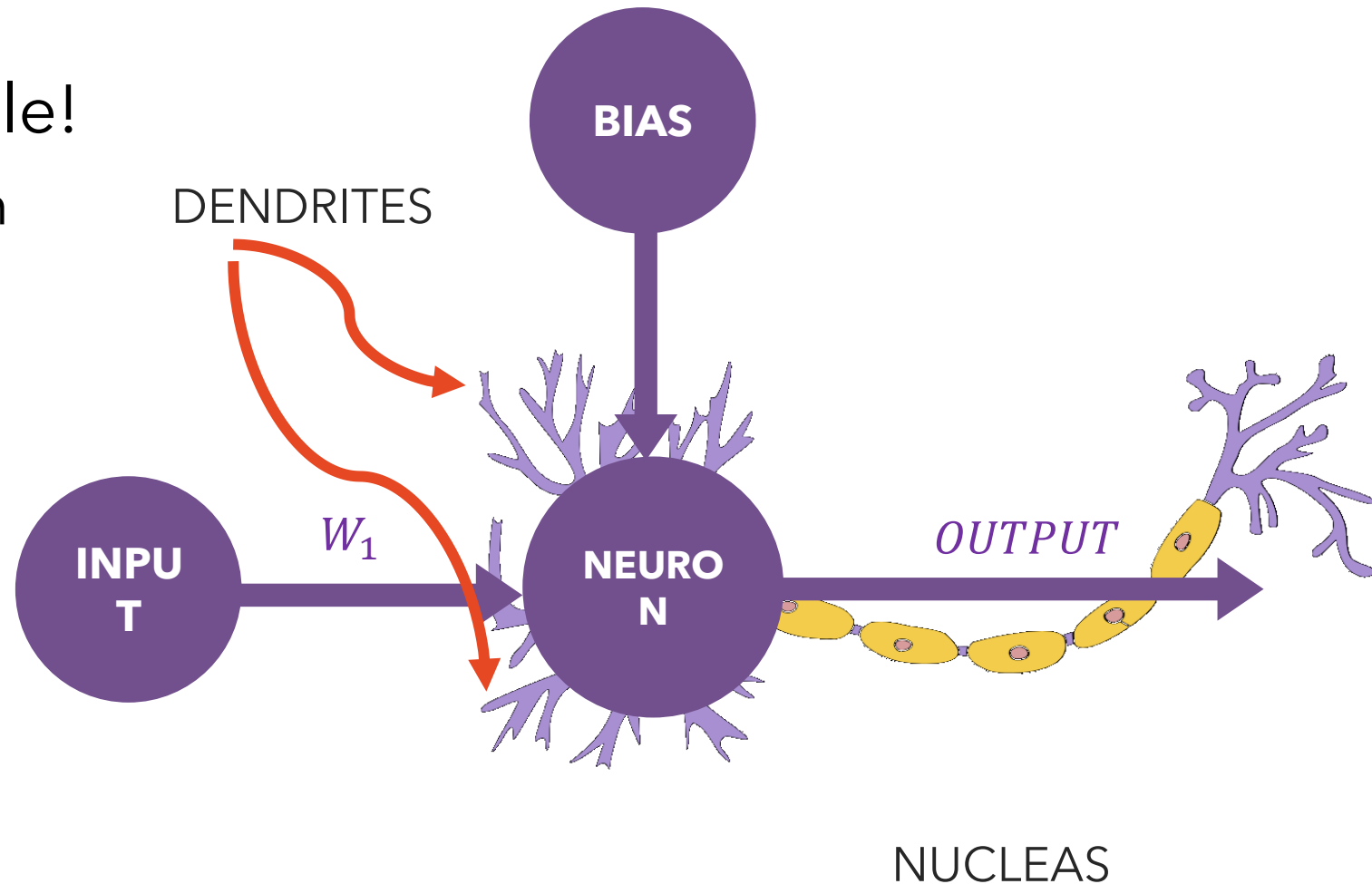
Redes Neuronales Artificiales en Accion!



Proyecto #3: Recuerdas la Primera Red?

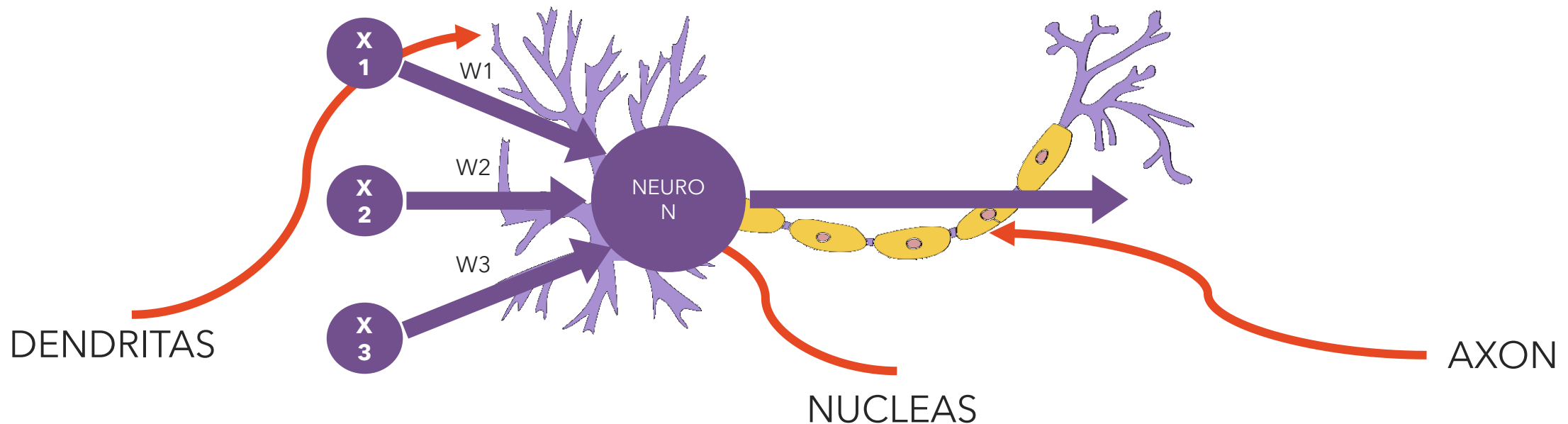
- El modelo era Muy Simple!
 - Sin function de activacion
 - Un Solo Input

$$\text{Output} = \text{Input} * W_1 + \text{Bias}$$



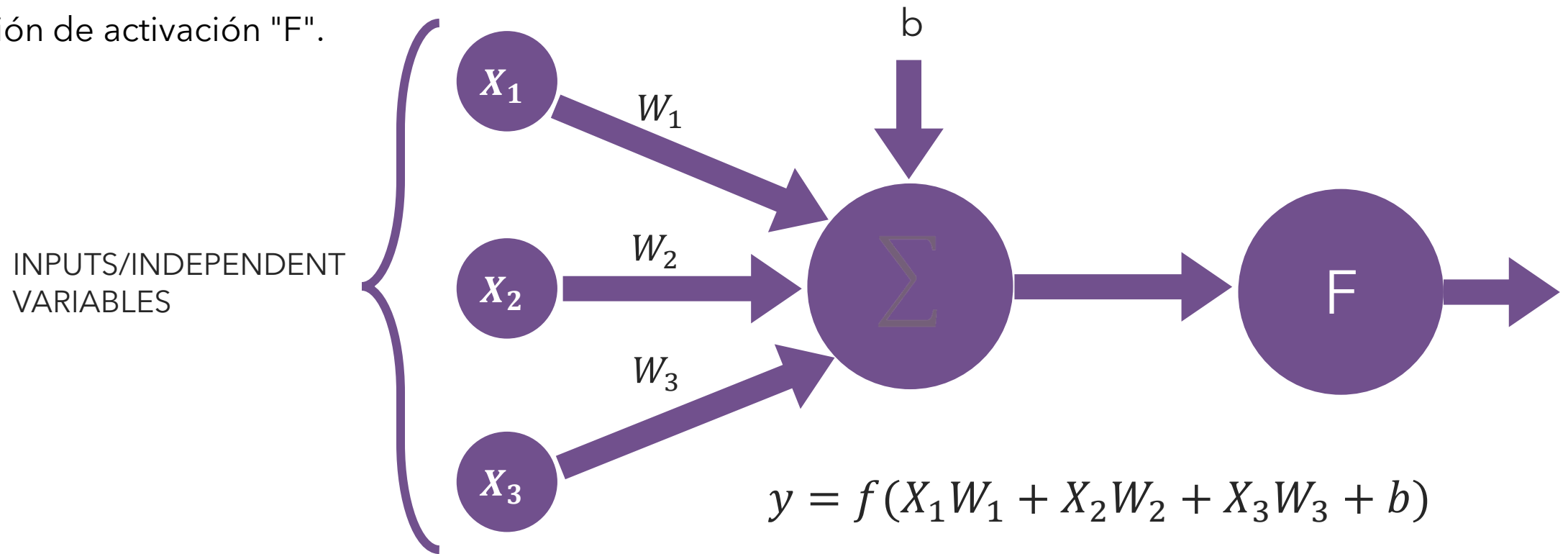
Proyecto #3: Modelo Matematico de Neurona

- La neurona recoge señales de los canales de entrada llamados dendritas, procesa la información en su núcleo y luego genera una salida en una larga y delgada rama llamada axón.



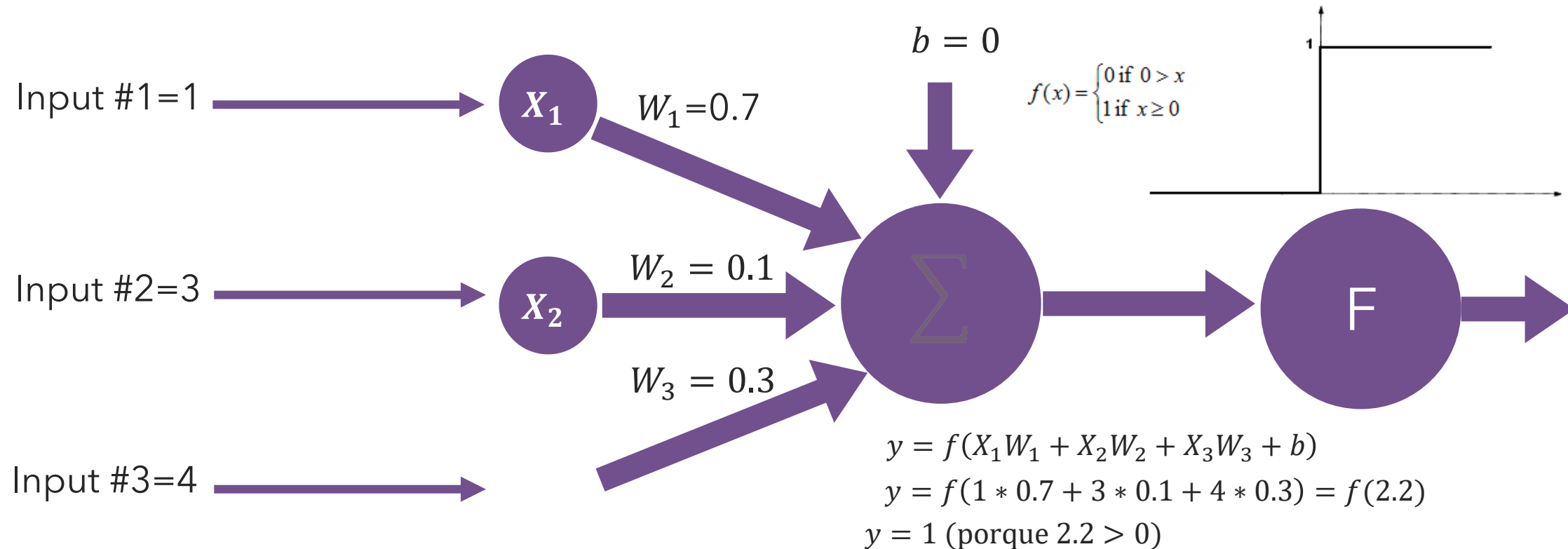
Proyecto #3: Recuerdas la Primera Red?

- El sesgo permite desplazar la curva de la función de activación hacia arriba o hacia abajo.
- Número de parámetros ajustables = 4 (3 pesos y 1 sesgo).
- Función de activación "F".



Proyecto #3: Una Neurona en Accion

- Asumamos una función de activación de la Unidad de Paso.
- La función de activación se utiliza para mapear la entrada entre (0, 1).



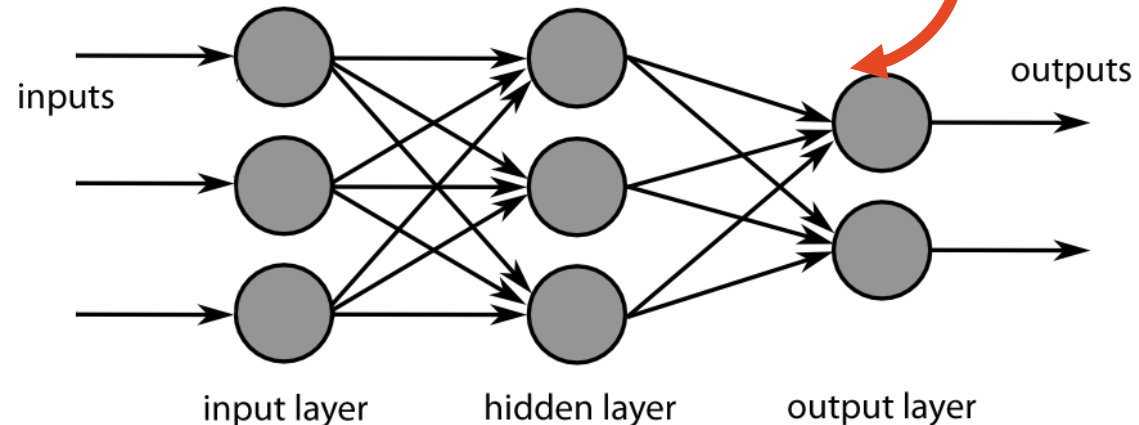
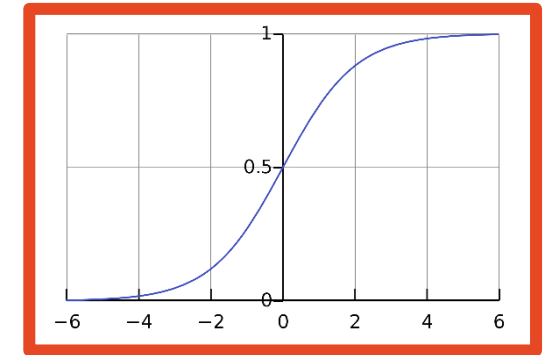
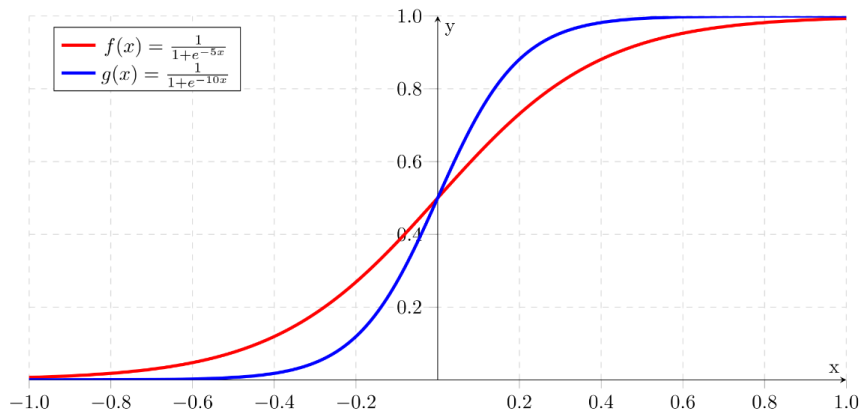


Funcion de Activacion

Proyecto #3: Funciones de Activación

- SIGMOID:

- Toma un número y lo pone entre 0 y 1
- Convierte los grandes números negativos en 0 y los grandes números positivos en 1.
- Generalmente se usa en la capa de salida.

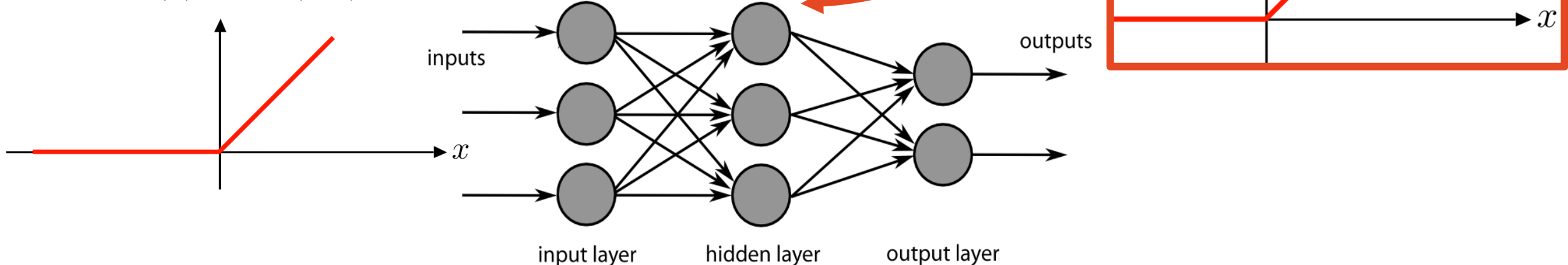


Proyecto #3: Funciones de Activación

- RELU (UNIDADES LINEALES RECTIFICADAS):

- si la entrada $x < 0$, la salida es 0 y si $x > 0$ la salida es x .
- El relu no se satura, por lo que evita el problema del gradiente de desaparición.
- Utiliza un umbral simple, por lo que es eficiente desde el punto de vista computacional.
- Generalmente se usa en capas ocultas.

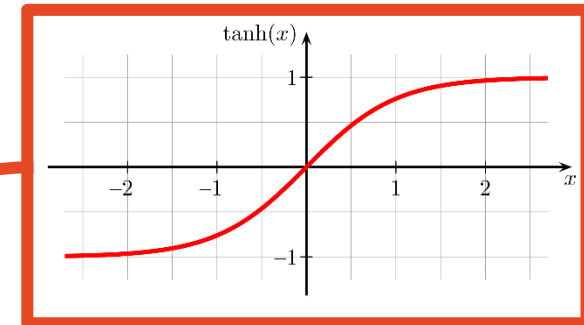
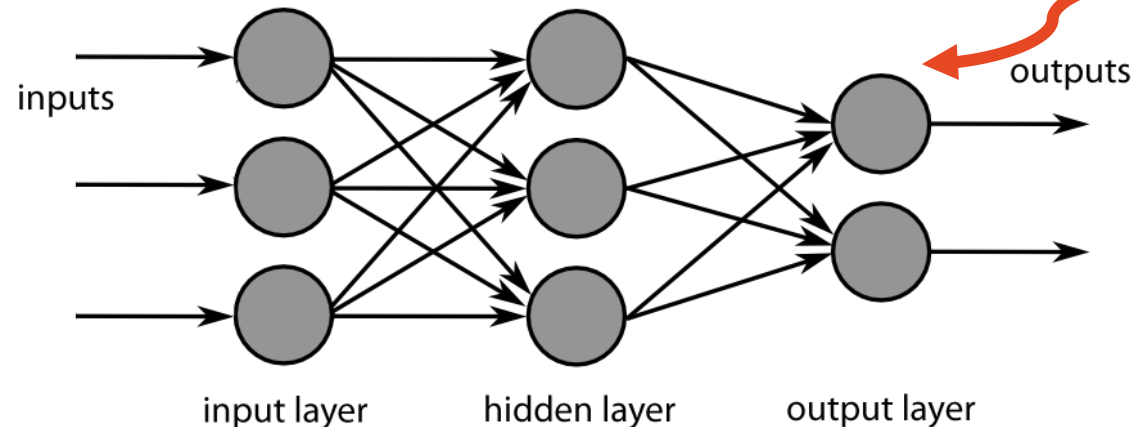
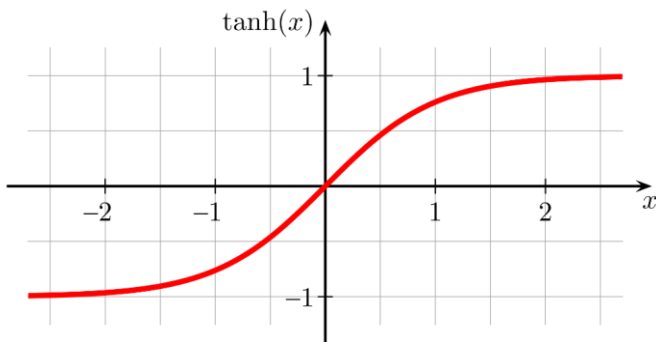
$$\text{ReLU}(x) \triangleq \max(0, x)$$



Proyecto #3: Funciones de Activación

- FUNCIÓN DE ACTIVACIÓN DE LA TANGENTE HIPERBÓLICA:

- "Tanh" es similar al sigmoide, convierte el número entre -1 y 1.
- A diferencia de la sigmoide, las salidas de tanh están centradas en cero (rango: -1 y 1).
- Tanh sufre el problema del gradiente de desaparición, por lo que mata los gradientes cuando está saturado.
- En la práctica, el tanh es preferible al sigmoide.



MODELO MULTI-NEURÓNICO (MODELO PERCEPTRÓNICO MULTICAPA)

- La red está representada por una matriz de pesos, entradas y salidas.
- Número total de parámetros ajustables = 8:
- Pesos = 6
- Sesgos = 2

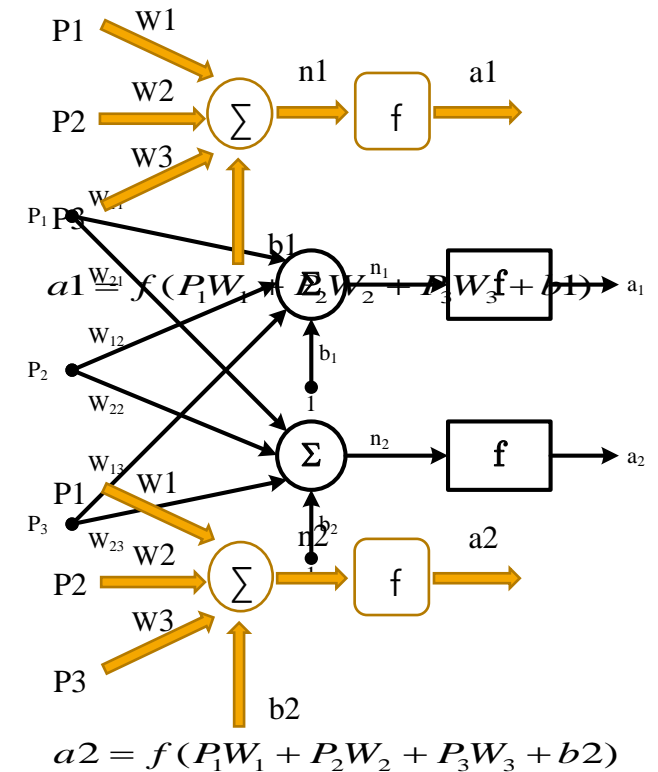
Matrix Representation

$$P = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

$$W = \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \end{bmatrix}$$

$$b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$a = f(W \times P + b)$$

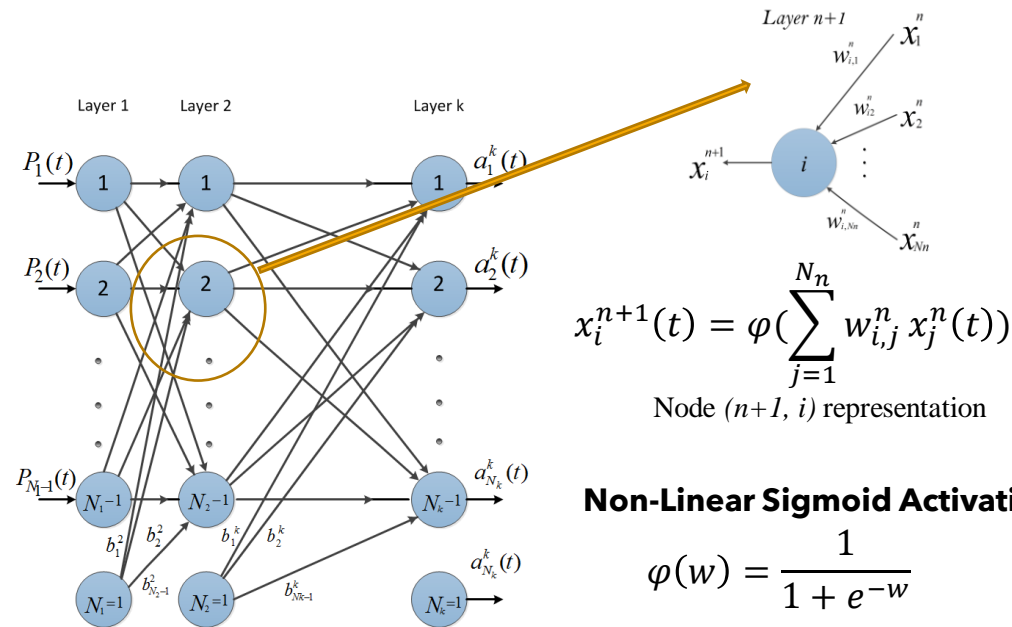


MODELO MULTI-NEURÓNICO (MODELO PERCEPTRÓNICO MULTICAPA)

- Conectemos varias de estas neuronas de forma multicapa.
- Cuantas más capas ocultas, más "profunda" será la red.

$$P = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_{N_1} \end{bmatrix}$$

$$\begin{bmatrix} W_{11} & W_{12} & \dots & W_{1,N_1} \\ W_{21} & W_{22} & \dots & W_{2,N_1} \\ \vdots & \vdots & \ddots & \vdots \\ W_{m-1,1} & W_{m-1,2} & \dots & W_{m-1,N_1} \\ W_{m,1} & W_{m,2} & \dots & W_{m,N_1} \end{bmatrix}$$



$$x_i^{n+1}(t) = \varphi\left(\sum_{j=1}^{N_n} w_{i,j}^n x_j^n(t)\right)$$

Node $(n+1, i)$ representation

Non-Linear Sigmoid Activation function

$$\varphi(w) = \frac{1}{1 + e^{-w}}$$

m : numero neuronas en capa oculta

N_1 : numero de inputs

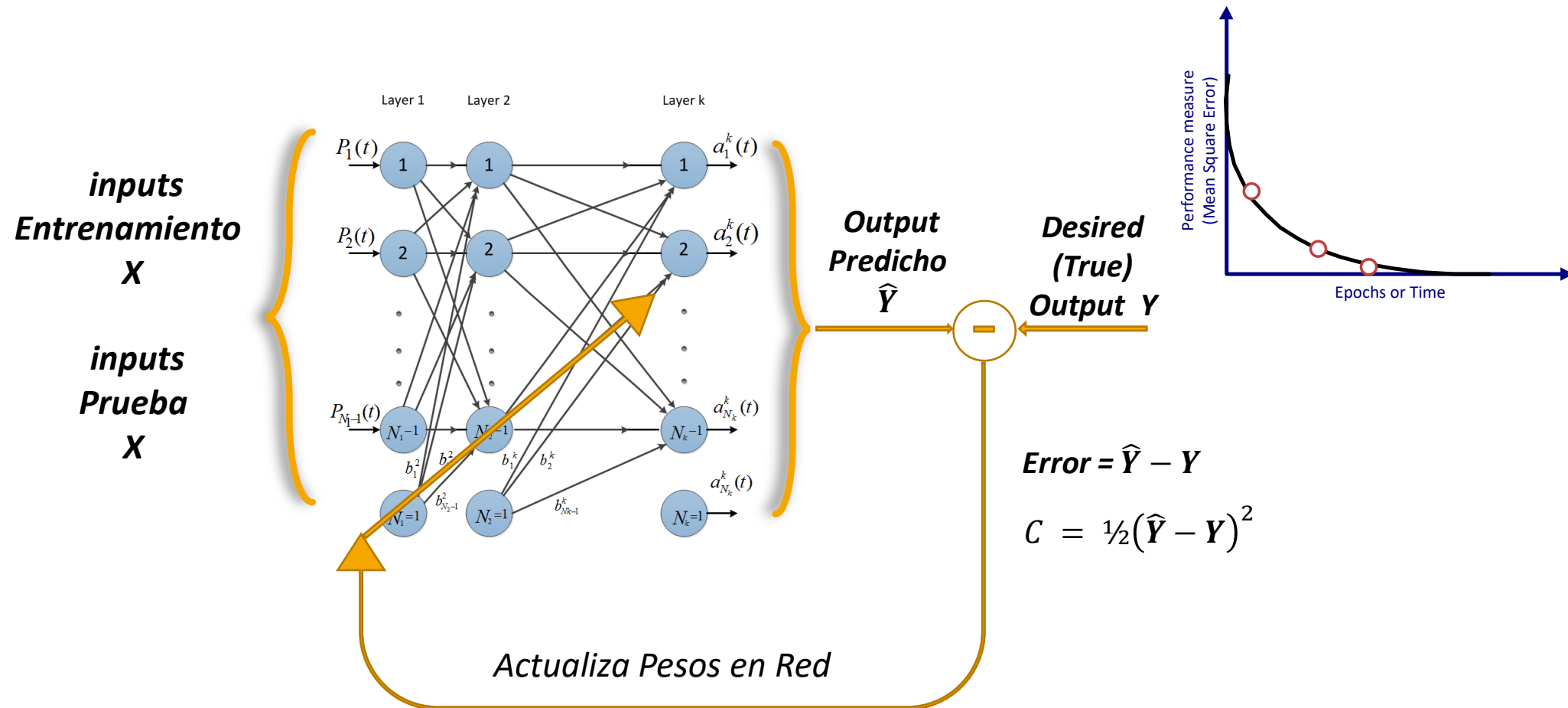


Como Entrenan las Redes Neuronales

Proyecto #3: Estrategias de Entrenamiento

- Aprendizaje supervisado
 - Se utiliza si hay un gran conjunto de datos de prueba con etiquetas conocidas (salidas).
 - El algoritmo de aprendizaje evalúa la salida (es decir, hace predicciones), compara la salida con la etiqueta y ajusta la red y la repetición.
- Aprendizaje no supervisado
 - Se utiliza con datos "no etiquetados" (no categorizados) (Ej: agrupación de k-means).
 - Dado que el algoritmo de aprendizaje funciona con datos no etiquetados, no hay forma de evaluar la exactitud de la estructura sugerida por el algoritmo
- Aprendizaje reforzado
 - El algoritmo de aprendizaje toma acciones que maximizan alguna noción de recompensa acumulativa.
 - Con el tiempo, la red aprende a preferir el tipo de acción correcta y a evitar la incorrecta.

Proyecto #3: Epochs



Proyecto #3: Divide Datos en Sets de Entrenamiento y Prueba

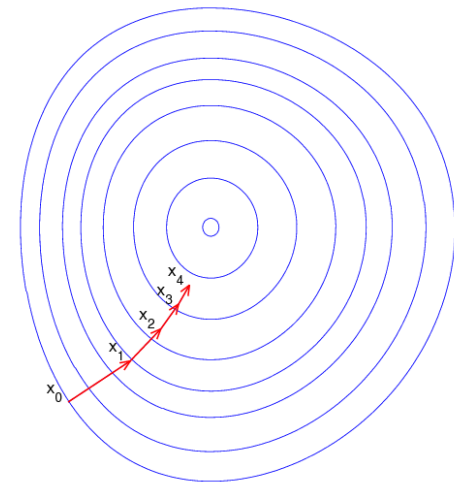
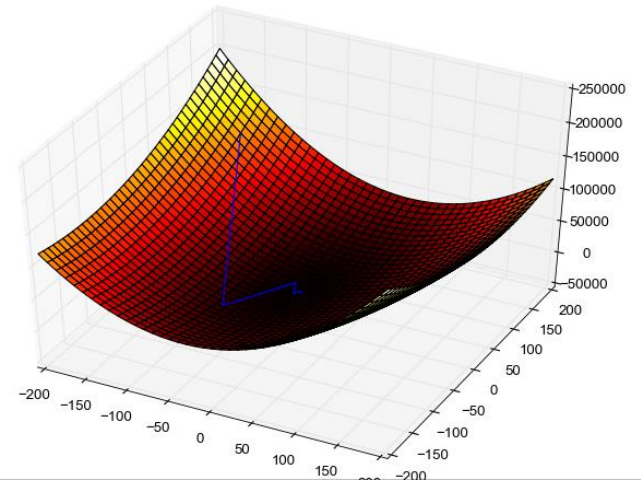
- El conjunto de datos se divide generalmente en segmentos de 50%, 25%, 25% para entrenamiento, validación y pruebas, respectivamente.
 - Conjunto de entrenamiento: se utiliza para el cálculo de gradientes y la actualización del peso.
 - Conjunto de validación:
 - utilizado para la validación cruzada que se realiza para evaluar la calidad del entrenamiento a medida que éste avanza.
 - La validación cruzada se implementa para superar el sobreajuste (sobreentrenamiento). El sobreajuste se produce cuando el algoritmo se centra en los detalles del conjunto de entrenamiento a costa de perder la capacidad de generalización.
 - La red entrenada MSE puede ser pequeña durante el entrenamiento, pero durante las pruebas, la red puede mostrar un rendimiento de generalización deficiente.
 - Conjunto de pruebas: se utiliza para probar la red entrenada.

The background is a dark blue field with a faint hexagonal grid. On the left, there are glowing, wavy blue lines. In the center, a white outline of a human brain is visible. To the right of the brain, there are glowing blue lines that resemble a circuit board, with small blue dots at the junctions. Scattered throughout the background are strings of white binary code (0s and 1s).

Descenso de Gradiente

Proyecto #3: Descenso de Gradiente

- El descenso de gradiente es un algoritmo de optimización que se utiliza para obtener los valores optimizados de peso y sesgo de la red
- Funciona tratando iterativamente de minimizar la función de costo
- Funciona calculando el gradiente de la función de costo y moviéndose en dirección negativa hasta que se alcanza el mínimo local/global
- Si se toma el positivo del gradiente, se alcanza el máximo local/global



Proyecto #3: Rango de Aprendizaje

El descenso gradual funciona de la siguiente manera:

1. Calcular la derivada (gradiente) de la función de pérdida
2. Escoge valores aleatorios para los parámetros m , b y sustitúto
3. Calcular el tamaño del paso (¿cuánto vamos a actualizar los parámetros?)

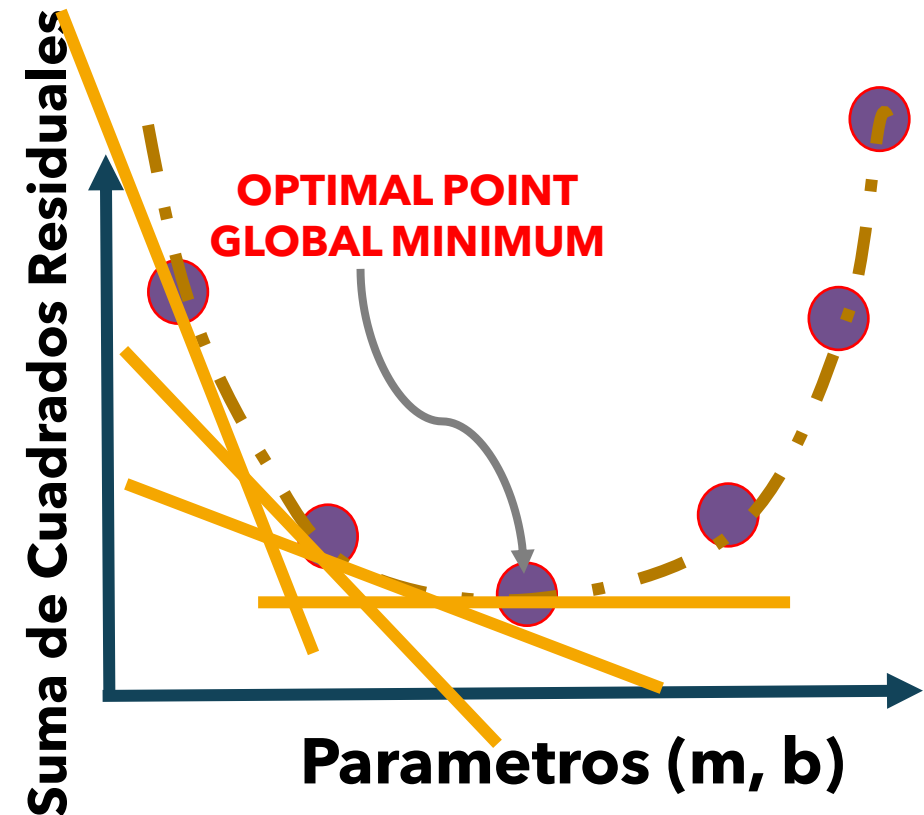
Step size = Slope * learning rate

4. Actualice los parámetros y repita

$$y = \boxed{b} + \boxed{m} * x$$

**GOAL IS TO FIND
BEST PARAMETERS**

*Nota: en realidad, este gráfico es 3D y tiene tres ejes, uno para m , b y la suma de los residuos cuadrados



Proyecto #3: Descenso de Gradiente con Matematicas

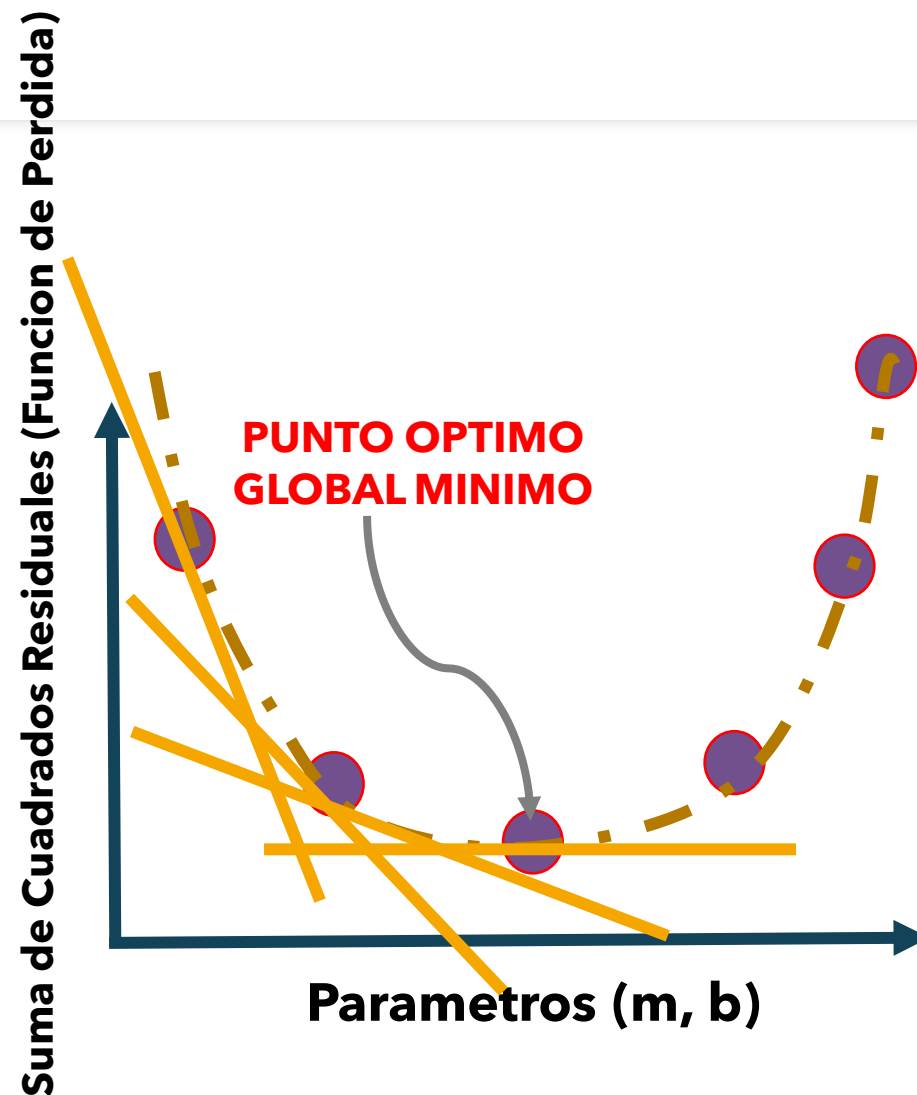
$$y = \boxed{b} + \boxed{m} * x$$

LA META ES ENCONTRAR
LOS MEJORES PARAMETROS

$$\text{Funcion de Perdida } f(m, b) = \frac{1}{N} \sum_{i=1}^n (y_i - (b + m * x_i))^2$$

$$\text{gradiente } f'(m, b) = \begin{bmatrix} \frac{df}{dm} \\ \frac{df}{db} \end{bmatrix} = \begin{bmatrix} \frac{1}{N} \sum_{i=1}^n -2x_i(y_i - (b + m * x_i))^2 \\ \frac{1}{N} \sum_{i=1}^n -2(y_i - (b + m * x_i))^2 \end{bmatrix}$$

*Nota: en realidad, este gráfico es 3D y tiene tres ejes, uno para m, b y la suma de los residuos cuadrados

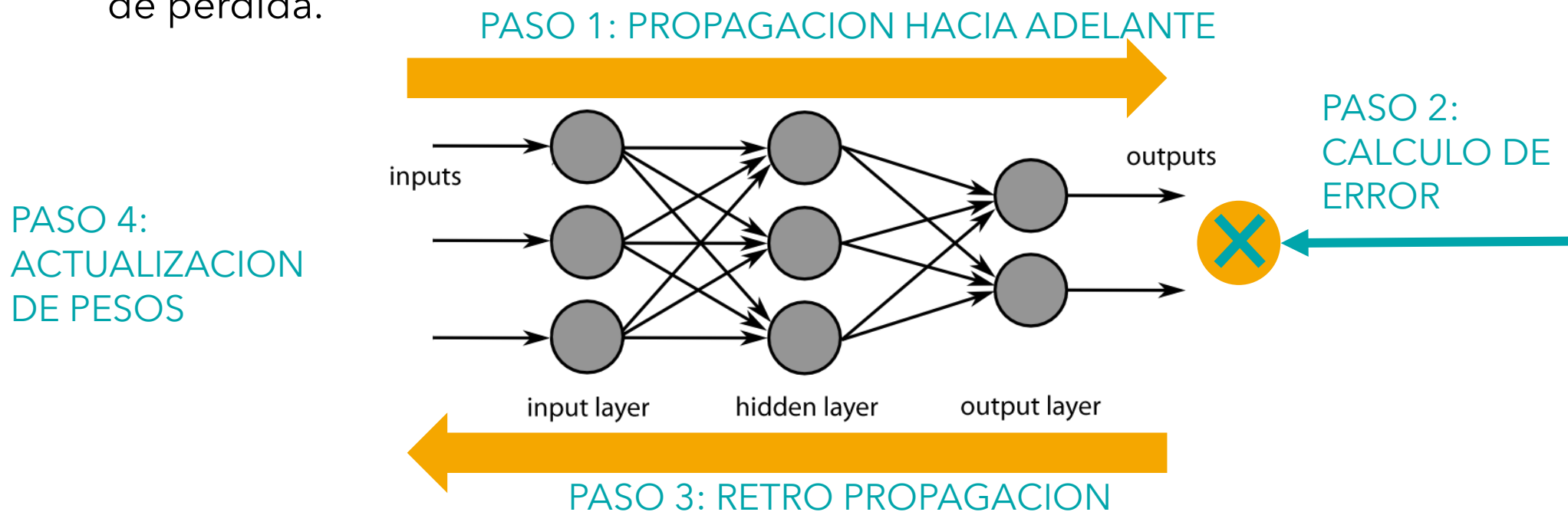


The background is a dark blue field with a faint hexagonal grid. On the left, there are glowing, wavy blue lines. In the center, a white outline of a hand is shown with fingers slightly curled. To the right of the hand, there is a cluster of glowing blue binary digits (0s and 1s) arranged in a circular pattern. Further to the right, there are glowing blue circuit-like lines with small circular nodes at various points.

Retropropagacion

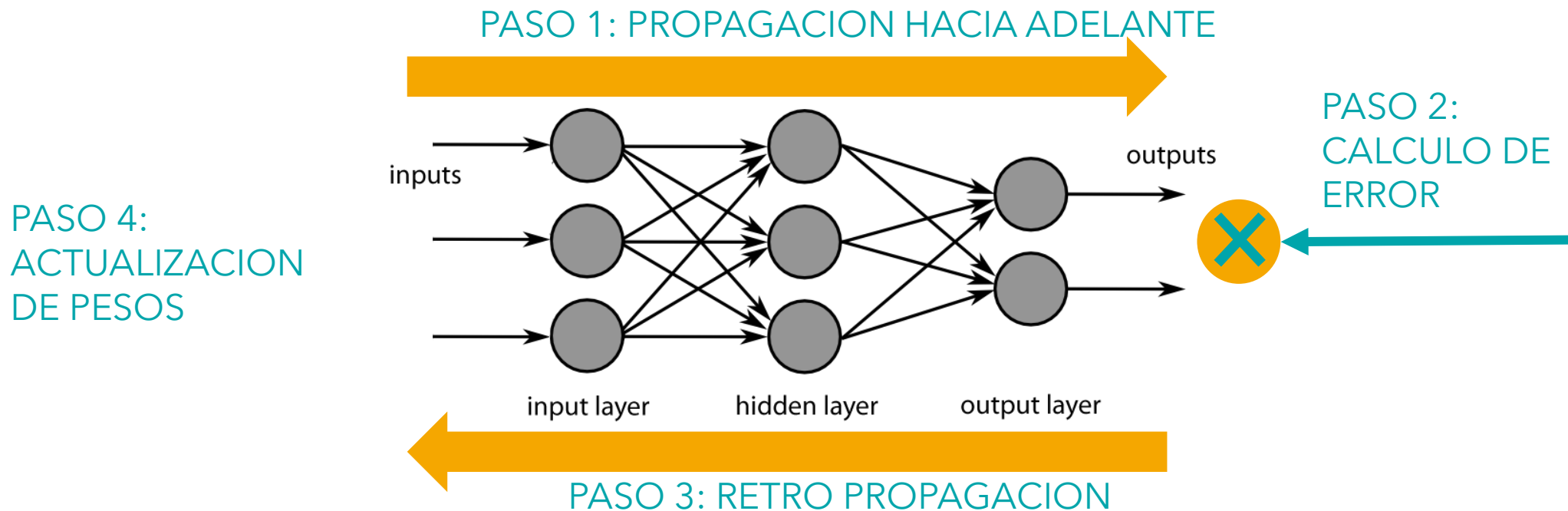
Proyecto #3: Retro-Propagacion

- La retro-propagación es un método utilizado para entrenar a las RNA mediante el cálculo del gradiente necesario para actualizar los pesos de la red.
- Se utiliza comúnmente por el algoritmo de optimización de descenso de gradiente para ajustar el peso de las neuronas mediante el cálculo del gradiente de la función de pérdida.



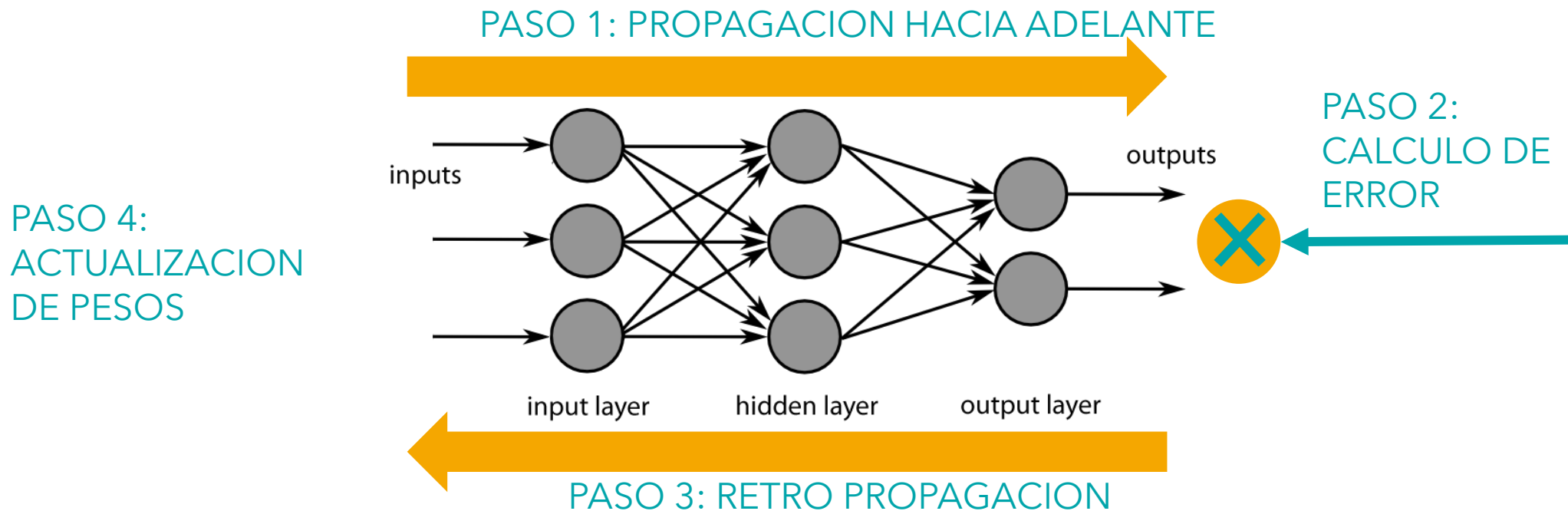
Proyecto #3: Retro-Propagacion

- Fase 1 de retropropagación: propagación
 - Propagación hacia adelante a través de la red para generar el valor o valores de salida
 - Cálculo del costo (término de error)
 - Propagación de las activaciones de salida de vuelta a través de la red utilizando el patrón de entrenamiento objetivo para generar los deltas (diferencia entre los valores de salida objetivo y los reales)



Proyecto #3: Retro-Propagacion

- Fase 2: actualización del peso
 - Calcula el gradiente de peso.
 - Se resta del peso una proporción (porcentaje) del gradiente del peso.
 - Esta proporción influye en la velocidad y la calidad del aprendizaje y se denomina tasa de aprendizaje. Cuanto mayor es la proporción, más rápido es el entrenamiento de las neuronas, pero menor es la proporción, más preciso es el entrenamiento.



Lectura Extra sobre Retro-Propagacion

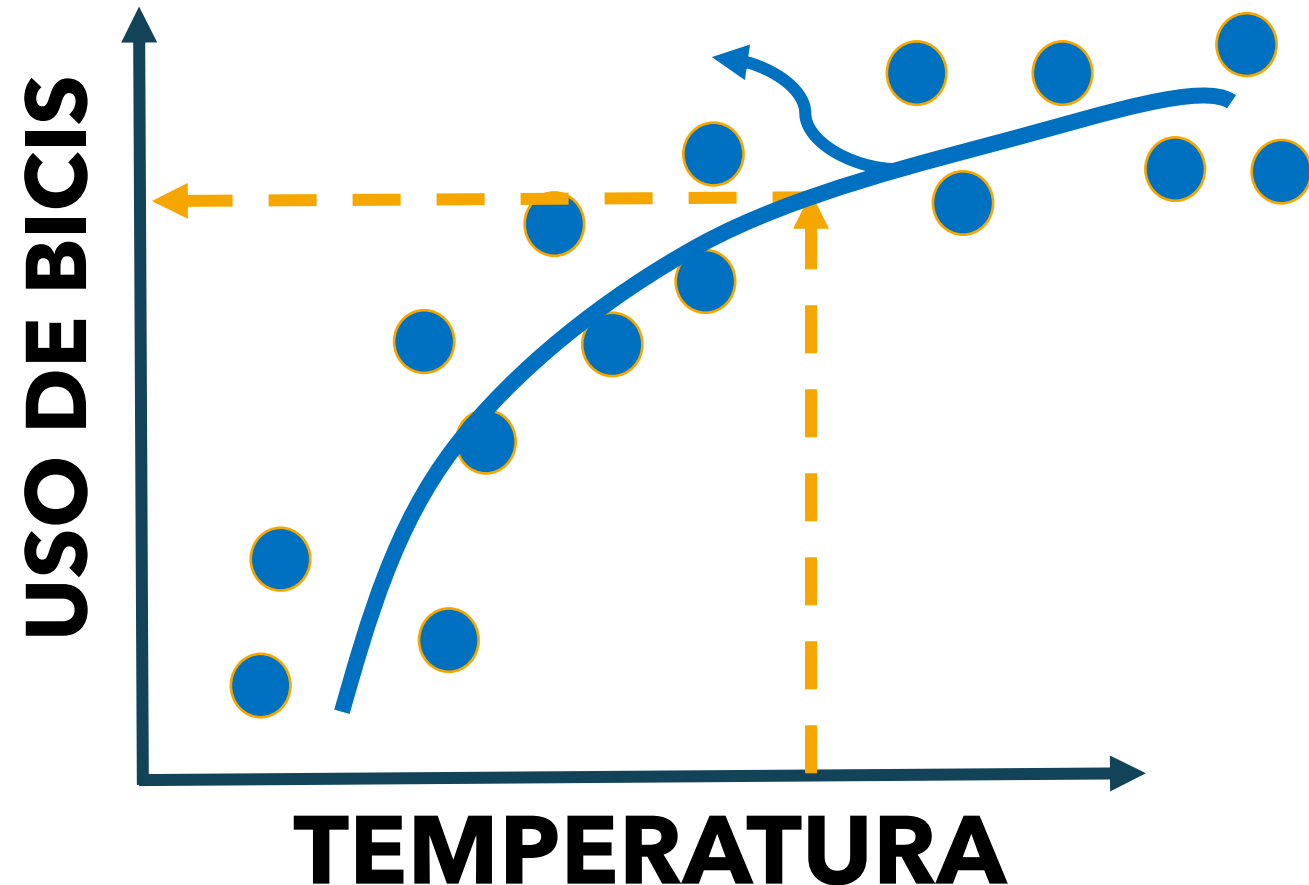
- "Backpropagation neural networks: A tutorial" by Barry J.Wythoff
- "Improved backpropagation learning in neural networks with windowed momentum", International Journal of Neural Systems, vol. 12, no.3&4, pp. 303-318.



COMPENSACIÓN DE LA VARIACIÓN DEL SESGO

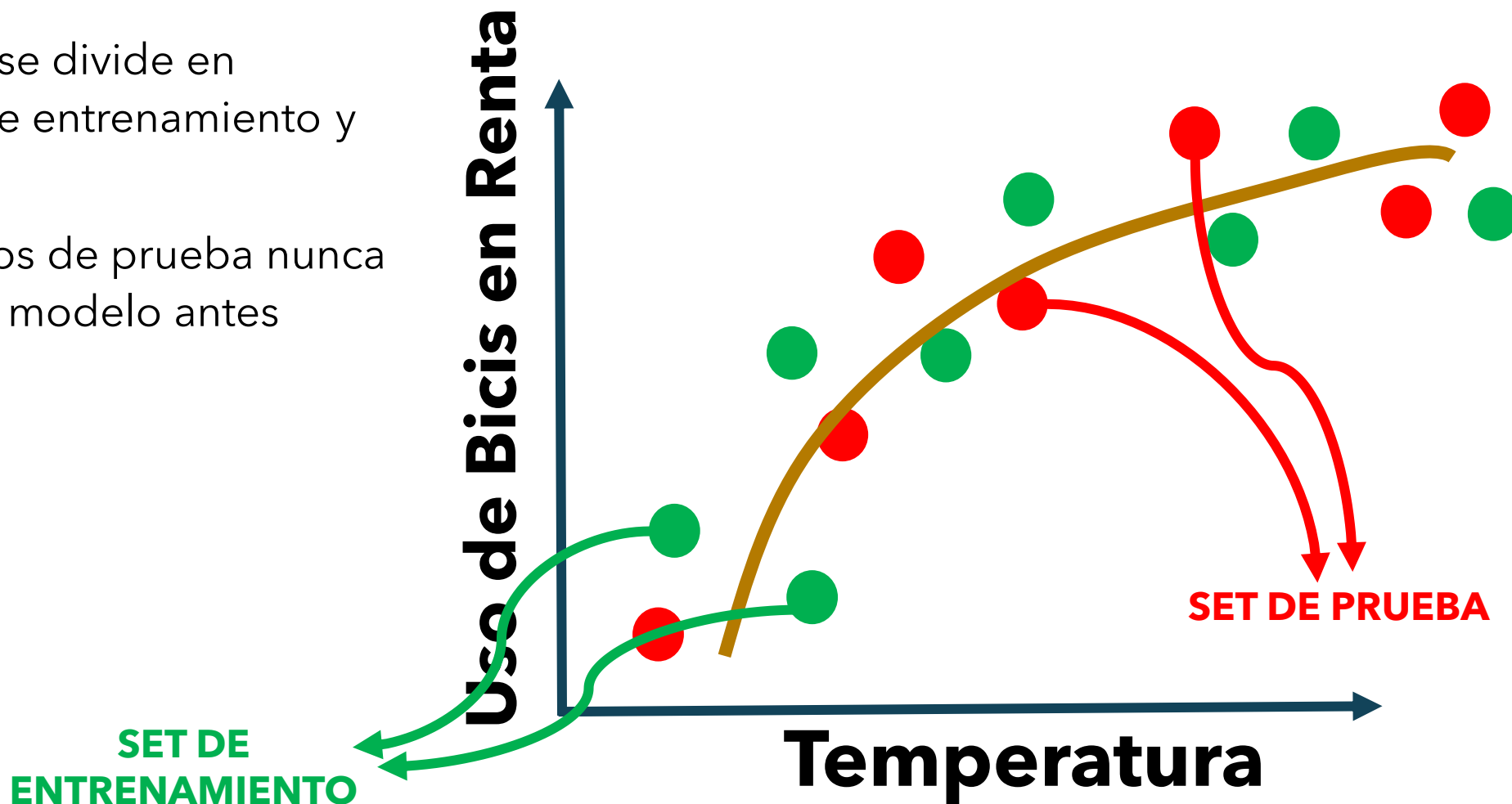
Proyecto #3: Intuición de la Variación del Sesgo

- Asumamos que queremos obtener la relación entre la temperatura y el uso de la bicicleta de alquiler.
- A medida que la experiencia de la temperatura aumenta, el uso del alquiler de bicicletas tiende a aumentar también.
- A medida que la temperatura supera un cierto límite, el uso tiende a estancarse y no aumenta más.



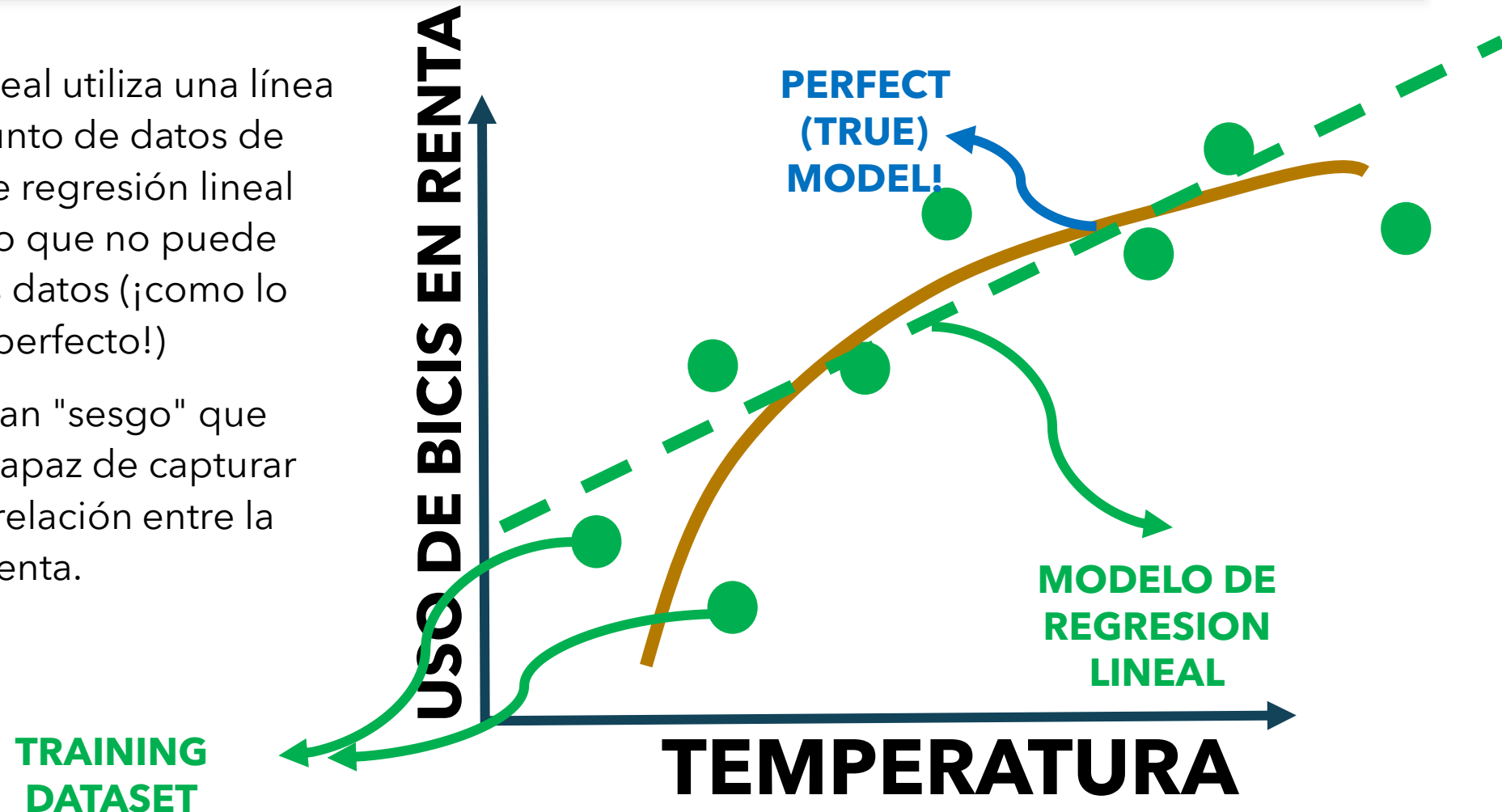
Proyecto #3: Varianza de Sesgo en Sets Entrenamiento y Prueba

- El conjunto de datos se divide en conjuntos de datos de entrenamiento y pruebas
- Los conjuntos de datos de prueba nunca han sido vistos por el modelo antes



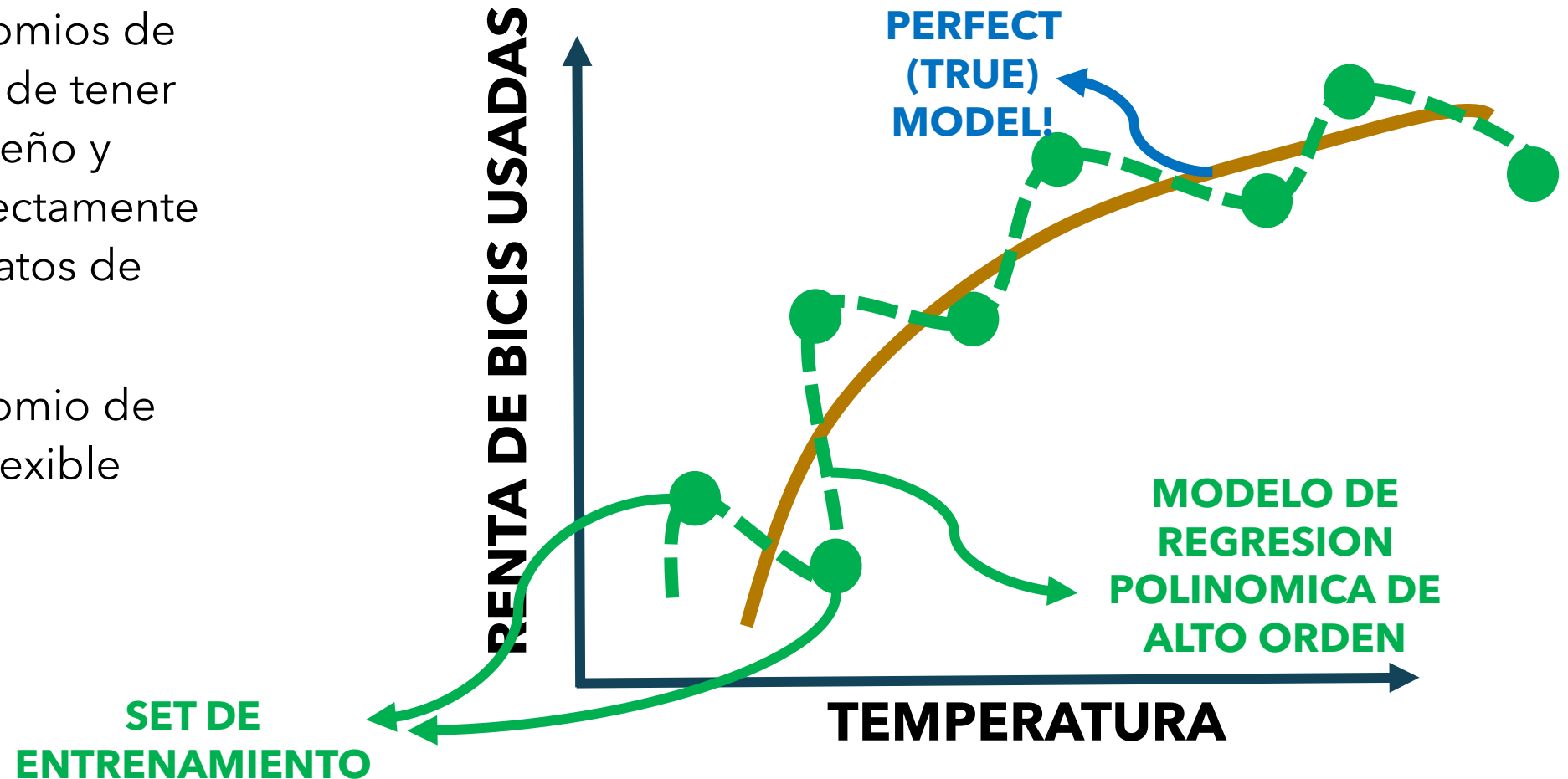
Proyecto #3: Varianza de Sesgos: Modelo #1 - Regresion Lineal Simple

- El modelo de Regresión Lineal utiliza una línea recta para ajustarse al conjunto de datos de entrenamiento. El modelo de regresión lineal carece de flexibilidad, por lo que no puede encajar adecuadamente los datos (¡como lo hace el verdadero modelo perfecto!).
- El modelo lineal tiene un gran "sesgo" que indica que el modelo es incapaz de capturar con precisión la verdadera relación entre la temperatura y el uso de la renta.

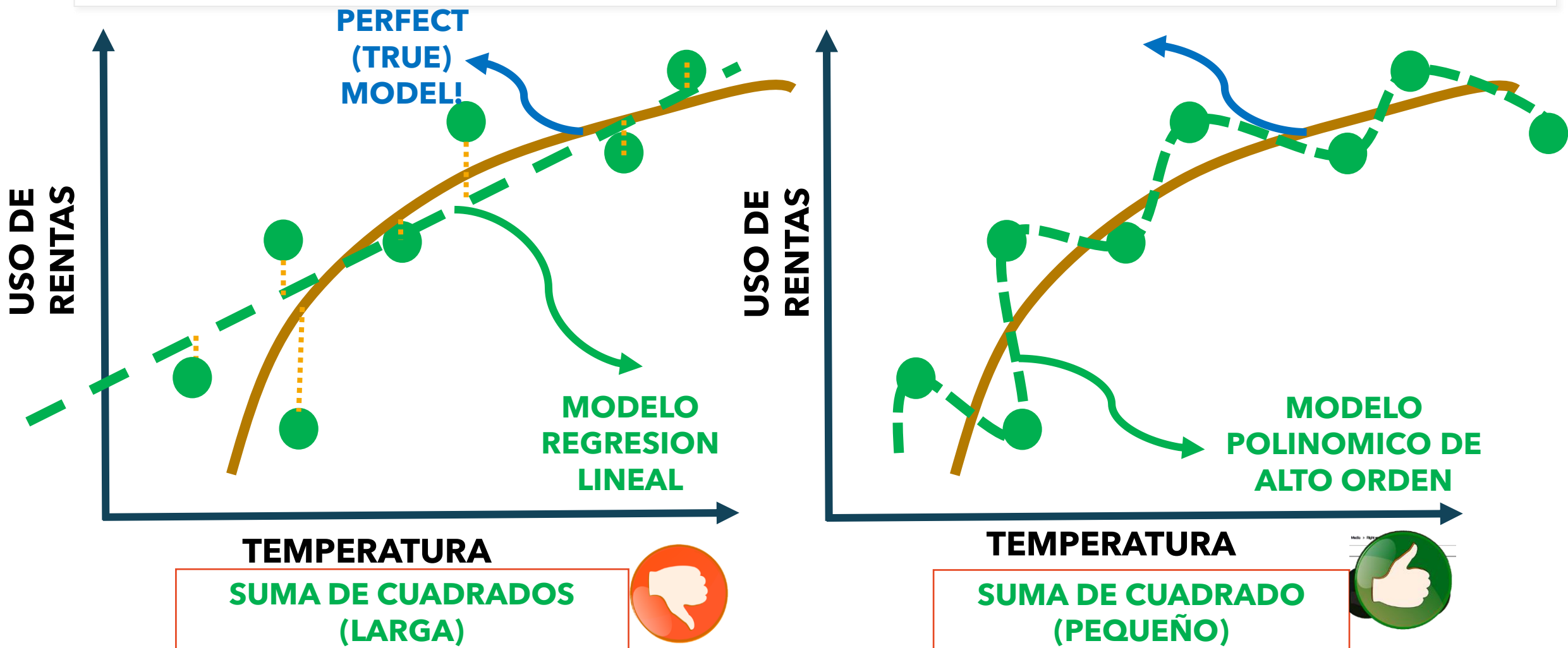


Proyecto #3: Sesgo y Varianza: Modelo #2 - Regresion Polinomial de Alto Orden

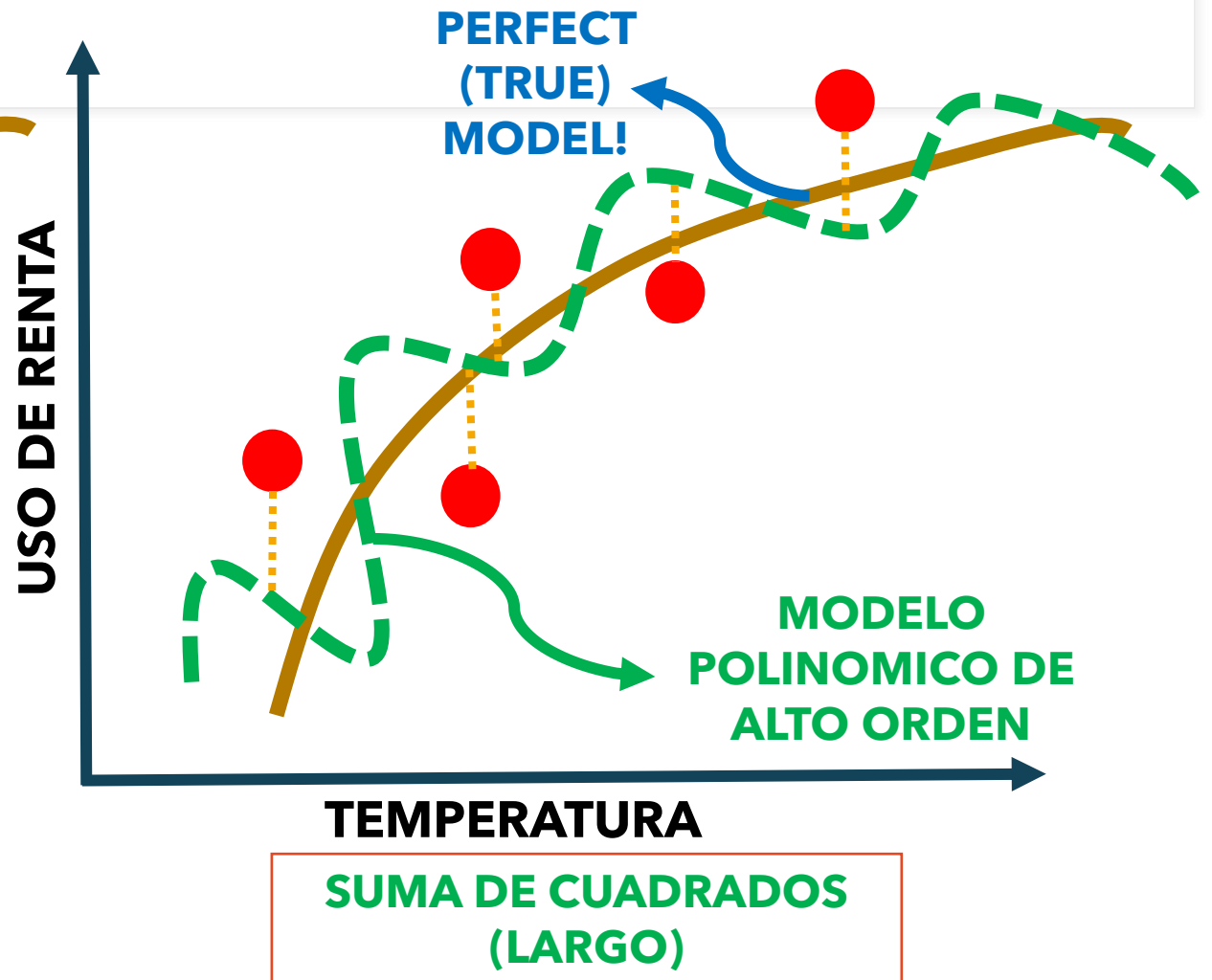
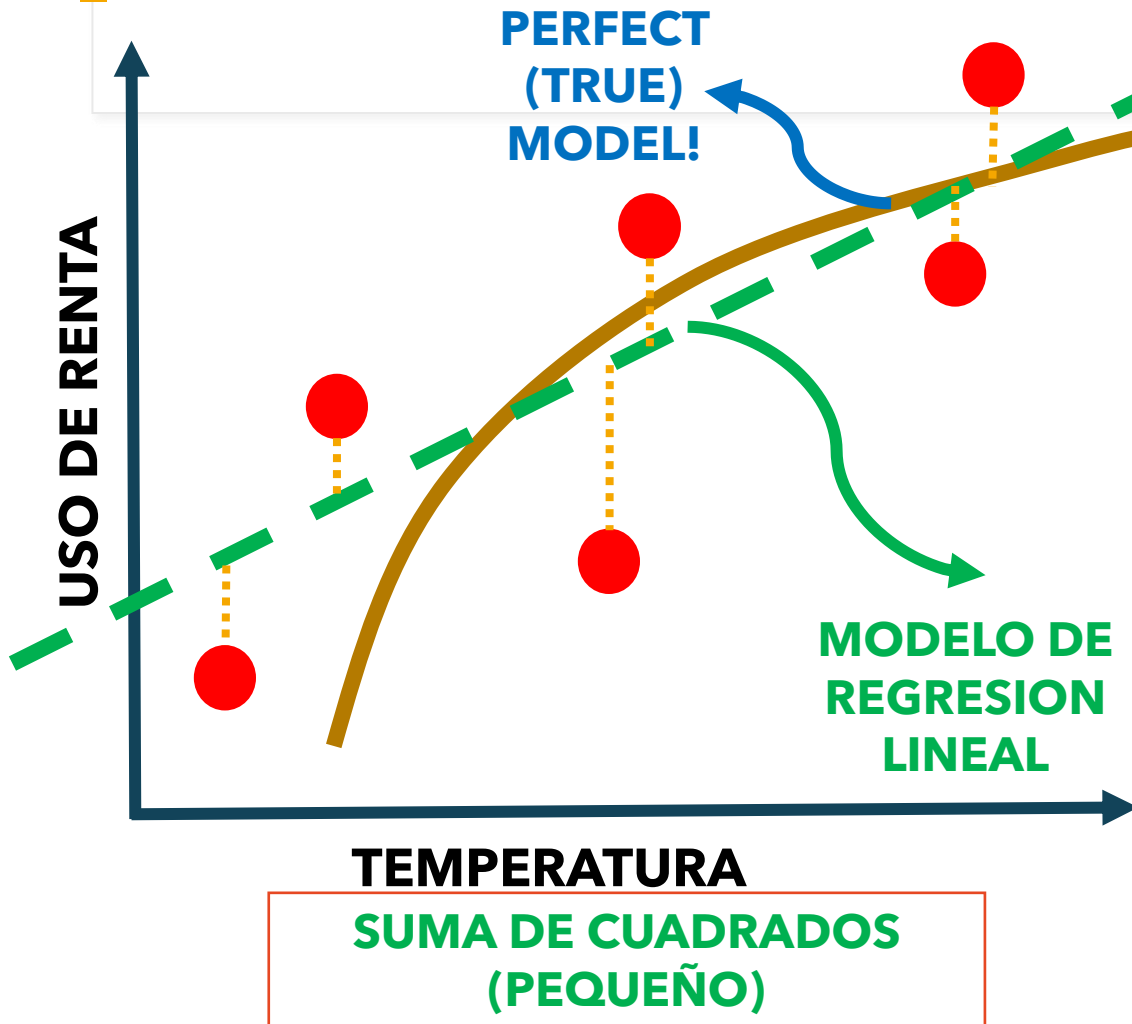
- El modelo de polinomios de alto orden es capaz de tener un sesgo muy pequeño y puede encajar perfectamente en el conjunto de datos de entrenamiento.
- El modelo de polinomio de alto orden es muy flexible



Proyecto #3: Sesgo y Varianza: Modelo #1 vs Modelo #2 Durante Entrenamiento



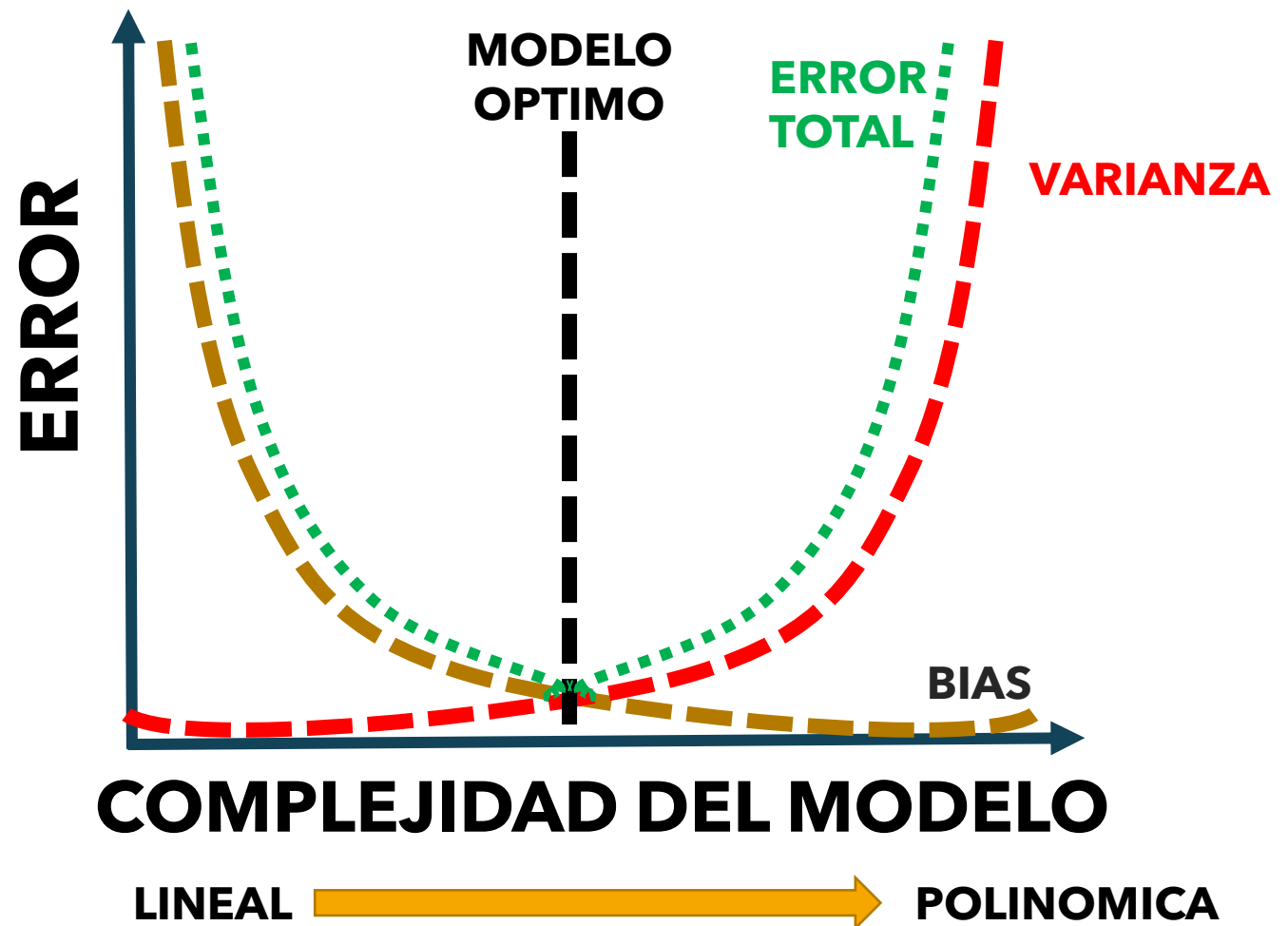
Proyecto #3: Sesgo y Varianza: Modelo #1 vs Modelo #2 Durante Prueba



El modelo polinómico funciona mal en el conjunto de datos de prueba y por lo tanto tiene una gran variación

Proyecto #3: Complejidad vs Error del Modelo

- La regularización funciona reduciendo la variación a costa de añadir algún sesgo al modelo.
- Se produce una compensación entre la varianza y el sesgo



Proyecto #3: Complejidad vs Error del Modelo

MODELO #1 (REGRESION LINEAL) (SIMPLE)	MODELO #2 (POLINOMICO DE ALTO ORDEN) (COMPLEJO)
El modelo tiene un alto sesgo porque es muy rígido (no flexible) y no puede encajar bien en el conjunto de datos de entrenamiento.	El modelo tiene un pequeño sesgo porque es flexible y puede encajar muy bien en el conjunto de datos de entrenamiento.
Tiene una pequeña variación (variabilidad) porque puede ajustarse a los datos de capacitación y a los datos de las pruebas con un nivel similar (el modelo es capaz de generalizar mejor) y evita el sobreajuste	Tiene una gran variación (variabilidad) porque el modelo se ajusta en exceso al conjunto de datos de entrenamiento y tiene un mal desempeño en el conjunto de datos de prueba
El rendimiento es consistente entre el conjunto de datos de entrenamiento y el conjunto de datos de prueba	El rendimiento varía enormemente entre el conjunto de datos de entrenamiento y el conjunto de datos de prueba (alta variabilidad)
Buena Generalizacion	Sobre Ajustado (Overfitting)

Proyecto #3: Complejidad vs Error del Modelo

- La varianza mide la diferencia en los ajustes entre el conjunto de datos de entrenamiento y el conjunto de datos de prueba
- Si el modelo se generaliza mejor, el modelo tiene una pequeña variación, lo que significa que el rendimiento del modelo es consistente entre los conjuntos de datos de entrenamiento y prueba
- Si el modelo se ajusta al conjunto de datos de entrenamiento, el modelo tiene una gran variación

EL MODELO DE REGRESIÓN PERFECTA TENDRÁ UN PEQUEÑO SESGO Y UNA PEQUEÑA VARIABILIDAD!

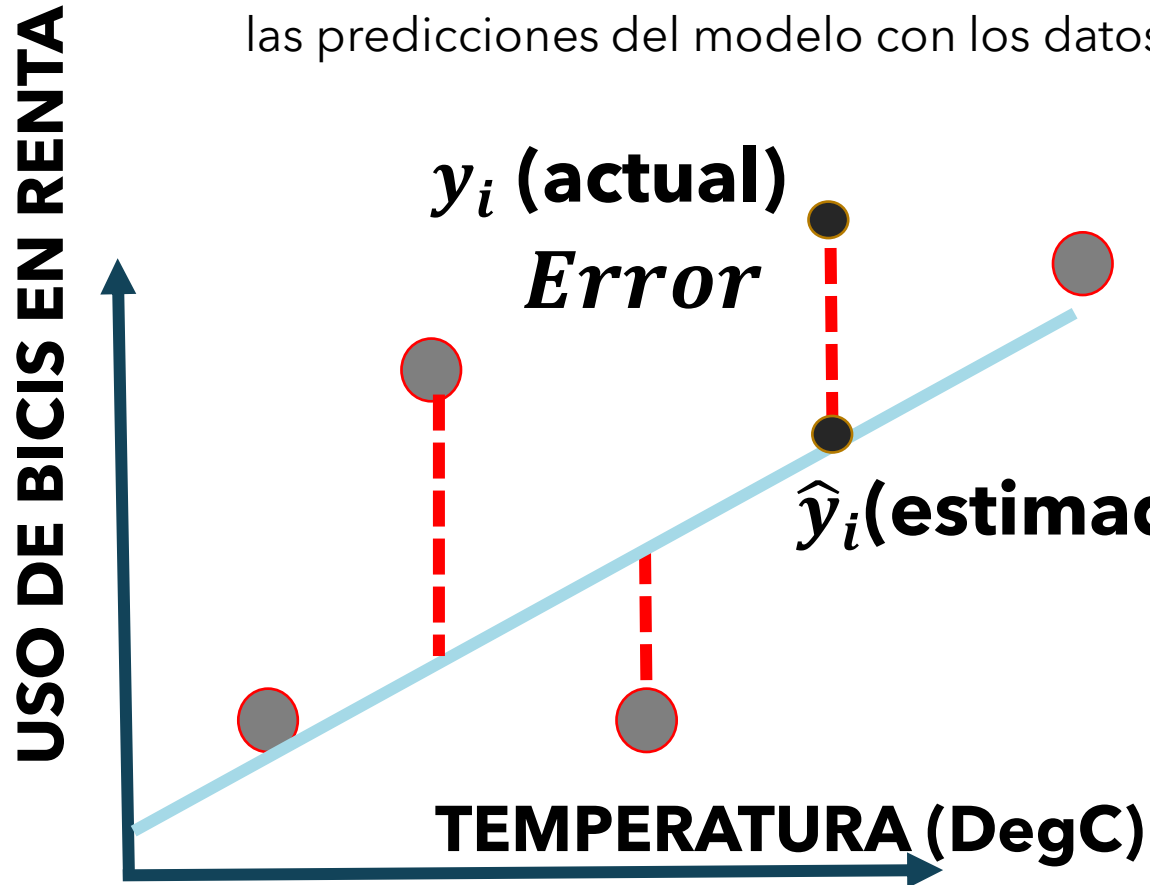
SE REALIZARÁ UN BALANCE ENTRE EL SESGO Y LA VARIANZA PARA LOS RESULTADOS FINALES

The background is a dark blue field with a subtle hexagonal grid. On the left, there are glowing, wavy blue lines. In the center, a glowing blue outline of a human brain is visible. To the right of the brain, there is a cluster of glowing blue binary code (0s and 1s). Further to the right, there are glowing blue circuit-like lines with circular nodes. The overall aesthetic is high-tech and digital.

Modelo de Evaluación del Rendimiento - Metrica

Proyecto #3: Metricas de Regresion: ¿Como Evaluar el Rendimiento de un Modelo?

- Después del ajuste del modelo, nos gustaría evaluar el rendimiento del mismo comparando las predicciones del modelo con los datos reales (Verdaderos)



$$\text{Residuales (Error)} = \hat{y}_i - y_i$$



Proyecto #3: Metricas de Regresion: Error Absoluto Medio (MAE)

- El error absoluto medio (MAE) se obtiene calculando la diferencia absoluta entre las predicciones del modelo y los valores verdaderos (reales)
- El MAE es una medida de la magnitud promedio del error generado por el modelo de regresión
- El error absoluto medio (MAE) se calcula de la siguiente manera:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

El MAE se calcula siguiendo estos pasos:

1. Calcular el residuo de cada punto de datos
2. Calcular el valor absoluto (para deshacerse del signo)
3. Calcular el promedio de todos los residuos

Si el MAE es cero, esto indica que las predicciones del modelo son perfectas.

Proyecto #3: Metricas de Regresion: Error del Cuadrado Medio (MSE)

- El Error Medio Cuadrado (MSE) es muy similar al Error Medio Absoluto (MAE), pero en lugar de utilizar valores absolutos, se calculan los cuadrados de la diferencia entre las predicciones del modelo y el conjunto de datos de entrenamiento (valores reales).
- Los valores del MSE son generalmente más grandes comparados con el MAE, ya que los residuos se están cuadrando.
- En el caso de los datos atípicos, la MSE será mucho mayor en comparación con el MAE
- En MSE, el error aumenta de forma cuadrática mientras que el error aumenta de forma proporcional en MAE
- En MSE, como el error se está cuadrando, cualquier error de predicción está siendo fuertemente penalizado
- El MSE se calcula de la siguiente manera:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- El MSE se calcula siguiendo estos pasos:
 1. Calcular el residuo para cada punto de datos
 2. Calcular el valor cuadrado de los residuos
 3. Calcular el promedio de los resultados del paso #2

Proyecto #3: Metricas de Regresion: Error Cuadratico Medio de la Raiz (RMSE)

- El Root Mean Square Error (RMSE) representa la desviación estándar de los residuos (es decir, las diferencias entre las predicciones del modelo y los valores reales (datos de entrenamiento)).
- El RMSE puede interpretarse fácilmente en comparación con el MSE porque las unidades del RMSE coinciden con las unidades de la salida.
- La RMSE proporciona una estimación del tamaño de la dispersión de los residuos.
- La RMSE se calcula de la siguiente manera:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

La RMSE se calcula siguiendo estos pasos:

1. Calcular el residuo para cada punto de datos
2. Calcular el valor cuadrado de los residuos
3. Calcular el promedio de los residuos cuadrados
4. Obtener la raíz cuadrada del resultado

Proyecto #3: Metricas de Regresion: Porcentaje de Error Absoluto Medio (MAPE)

- Los valores de MAE pueden ir de 0 a infinito, lo que dificulta la interpretación del resultado en comparación con los datos de entrenamiento.
- El porcentaje de error absoluto medio (MAPE) es el equivalente al MAE pero proporciona el error en forma de porcentaje y por lo tanto supera las limitaciones del MAE.
- MAPE puede presentar algunas limitaciones si el valor del punto de datos es cero (ya que hay una operación de división implicada)
- El MAPE se calcula de la siguiente manera:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n |(y_i - \hat{y}_i) / y_i|$$

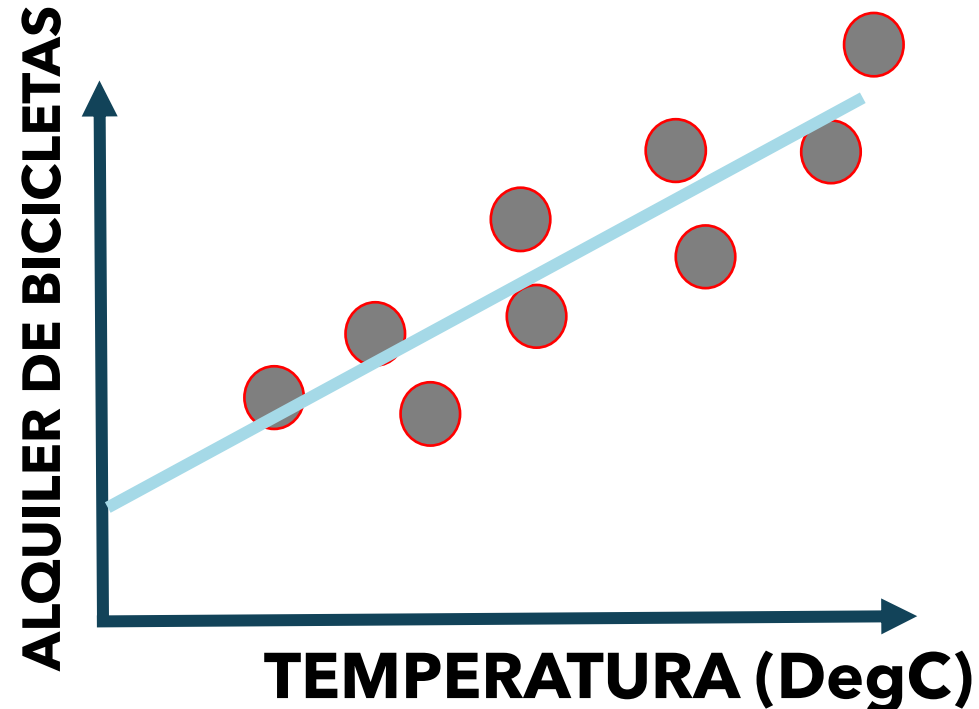
Proyecto #3: Metricas de Regresion: Porcentaje Medio de Error (MPE)

- El MPE es similar al MAPE pero sin la operación absoluta
- El MPE es útil para proporcionar una idea de cuántos errores positivos en comparación con los negativos
- El MPE se calcula de la siguiente manera:

$$MPE = \frac{100\%}{n} \sum_{i=1}^n (y_i - \hat{y}_i) / y_i$$

Proyecto #3: Metricas de Regresion: R Cuadrado (R^2)-Coeficiente de Determinacion

- R-cuadrado o el coeficiente de determinación representa la proporción de varianza (de y) que ha sido explicada por las variables independientes del modelo.
- Si $R^2=80$, esto significa que el 80% del incremento en el uso de la bicicleta se debe al aumento de la temperatura.



Proyecto #3: Metricas de Regresion: R Cuadrado (R^2)-Coeficiente de Determinacion

- R-cuadrado o el coeficiente de determinación representa la proporción de varianza (y) que ha sido explicada por las variables independientes (X) en el modelo.
- Proporciona una indicación de la bondad del ajuste y , por lo tanto, una medida de cuán bien es probable que el modelo prediga las muestras no vistas, a través de la proporción de la varianza explicada.
- La mejor puntuación posible es 1,0
- Un modelo constante que siempre predice el valor esperado de y , sin tener en cuenta las características de entrada, obtendría una puntuación R^2 de 0,0.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

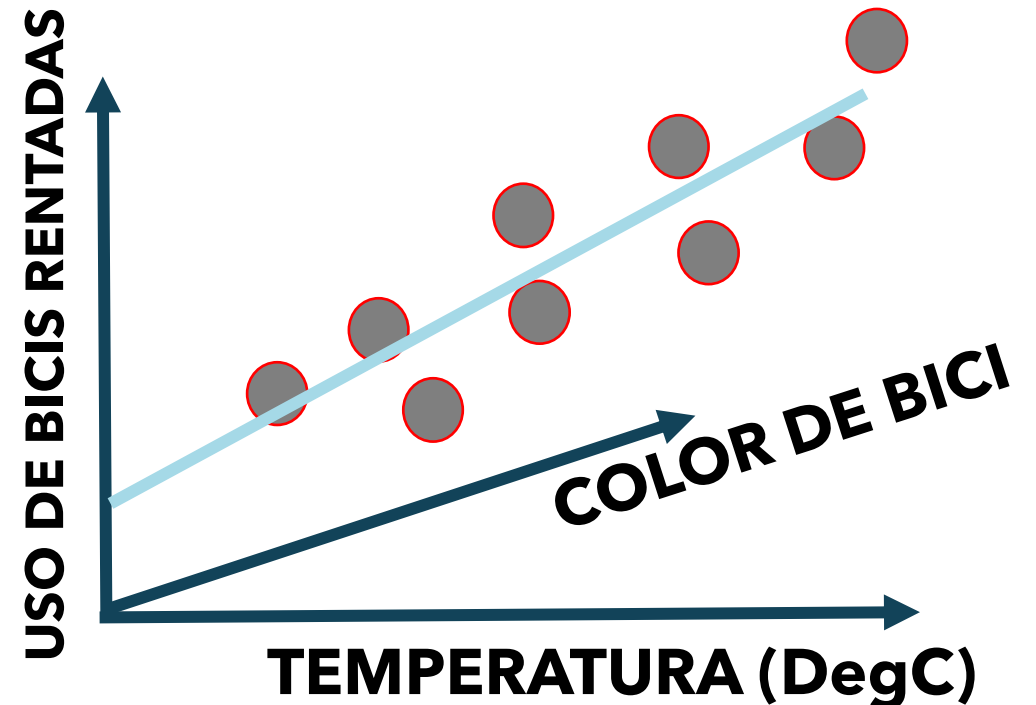
Proyecto #3: Metricas de Regresion: R Cuadrado (R^2)-Coeficiente de Determinacion

- El cuadrado R representa la proporción de la varianza de la variable dependiente (y) que ha sido explicada por las variables independientes.
- R-cuadrado proporciona una visión de la bondad del ajuste.
- Da una medida de lo bien que las muestras no vistas pueden ser predichas por el modelo, a través de la proporción de la varianza explicada.
- El valor máximo de R^2 es 1
- Un modelo constante que siempre predice el valor esperado de y, sin tener en cuenta las características de entrada, tendrá una puntuación R^2 de 0,0.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Proyecto #3: Metricas de Regresion: R Cuadrado Ajustado (R^2)

- Si $R^2=80$, esto significa que el 80% del incremento en el uso de la bicicleta de alquiler se debe al aumento de la temperatura.
- Añadamos otra variable independiente "inútil", digamos el color de la bicicleta en el eje Z.
- Ahora R^2 aumenta y se convierte en: $R^2=85\%$



Proyecto #3: Metricas de Regresion: R Cuadrado Ajustado (R^2)

- Una limitación de R^2 es que aumenta al añadir variables independientes al modelo, lo cual es engañoso ya que algunas variables añadidas podrían ser inútiles con un significado mínimo.
- Ajustado R^2 supera este problema añadiendo una penalización si hacemos un intento de añadir una variable independiente que no mejore el modelo.
- Adjusted R^2 es una versión modificada de la R^2 y tiene en cuenta el número de predictores en el modelo.
- Si se añaden predictores inútiles al modelo, la versión ajustada de R^2 disminuirá
- Si se añaden predictores útiles al modelo, Ajustado R^2 aumentará
- K es el número de variables independientes y n es el número de muestras

$$R_{adjusted}^2 = 1 - \left[\frac{(1 - R^2)(n - 1)}{n - k - 1} \right]$$

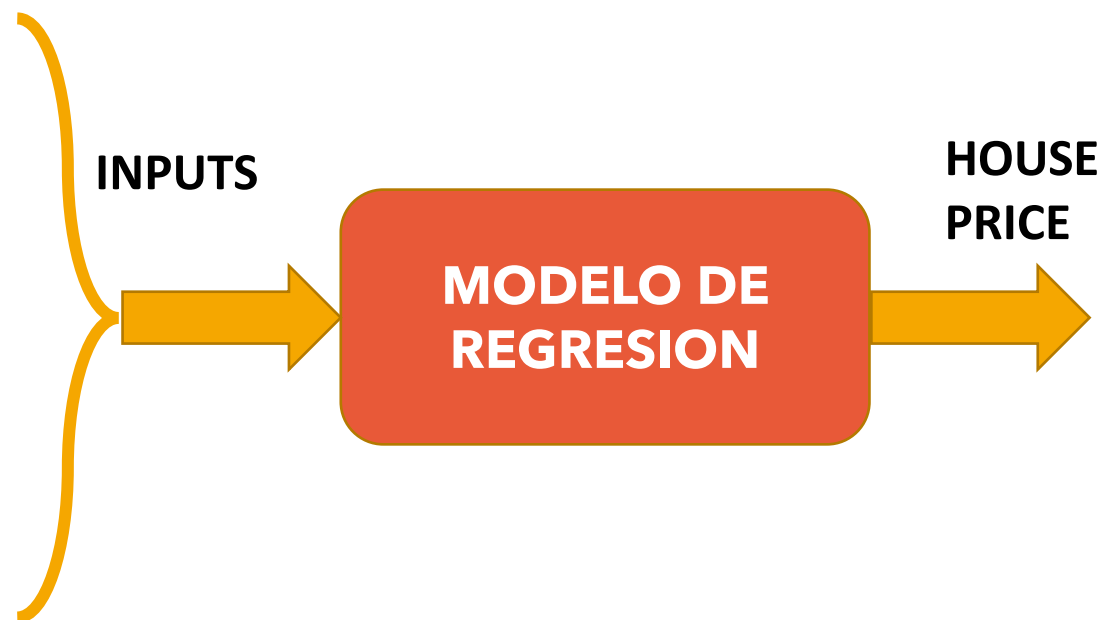
Proyecto #4: Prediccion de Precios en Bienes Raices



Proyecto #4: Predicción Precios de Bienes Raíces

- El conjunto de datos incluye los precios de venta de las casas del condado de King en Washington, EE.UU.
- Las casas que se venden en el período de tiempo: mayo de 2014 y mayo de 2015.
- Fuente de datos: <https://www.kaggle.com/harlfoxem/housesalesprediction>
- Modelo de entradas:

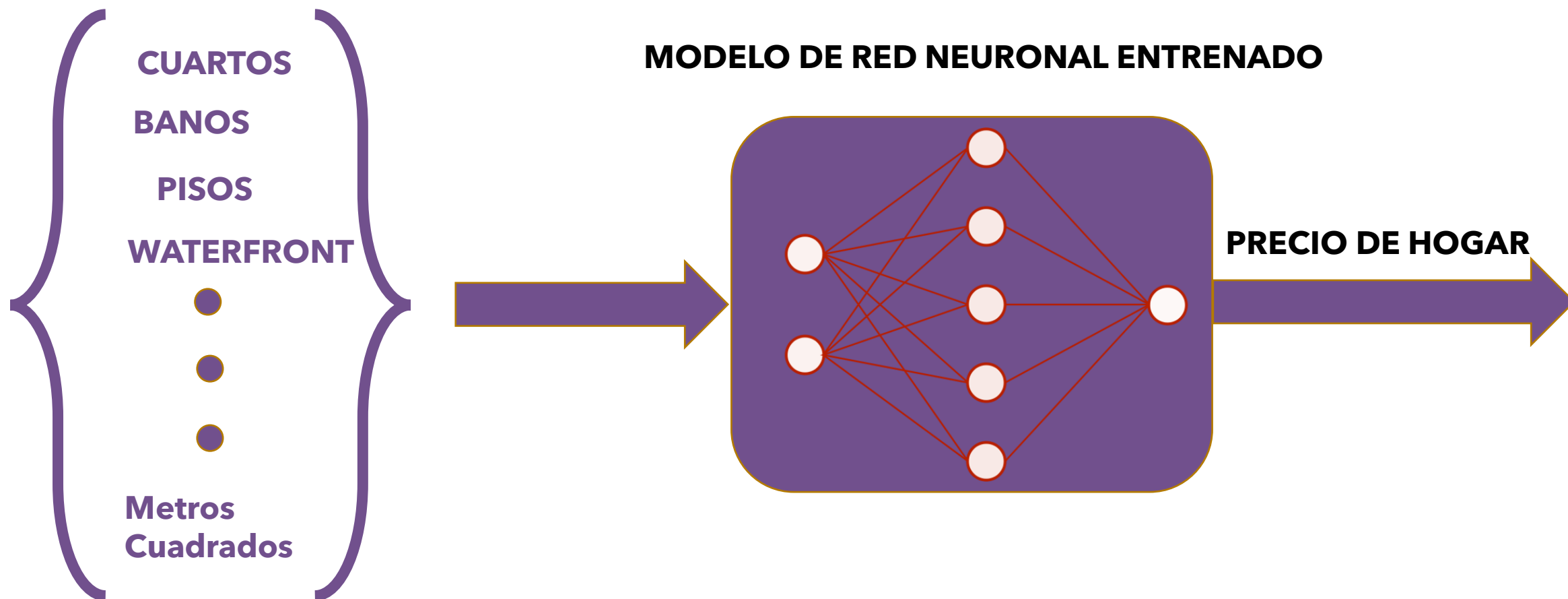
- ida: notación para una casa
- fecha: Fecha en que se vendió la casa
- dormitorios: Número de dormitorios
- baños: Número de baños
- sqft_living: home square footage
- sqft_lot: metros cuadrados del lote
- pisos: Total de pisos (niveles) en la casa
- frente al mar: propiedad frente al mar



Proyecto #4: Predicción Precios de Bienes Raíces

- Modelo de entradas:
 - condición: Qué tan buena es la condición (En general)
 - calificación: calificación general dada a la unidad de vivienda, basada en el sistema de calificación del Condado de King
 - sqft_abovesquare: imágenes de la casa aparte del sótano
 - sqft_basement: pies cuadrados del sótano
 - yr_built: Año de construcción
 - yr_renovado: Año en que la casa fue renovada
 - código postal: zip
 - lat: Coordenadas de latitud
 - largo: Coordenadas de longitud
 - sqft_living15: Superficie de la sala de estar en 2015
 - sqft_lot15: área de lotSize en 2015(implica... algunas renovaciones)
- El modelo debería predecir:
 - El precio de la casa

Proyecto #4: Prediccion Precios de Bienes Raices

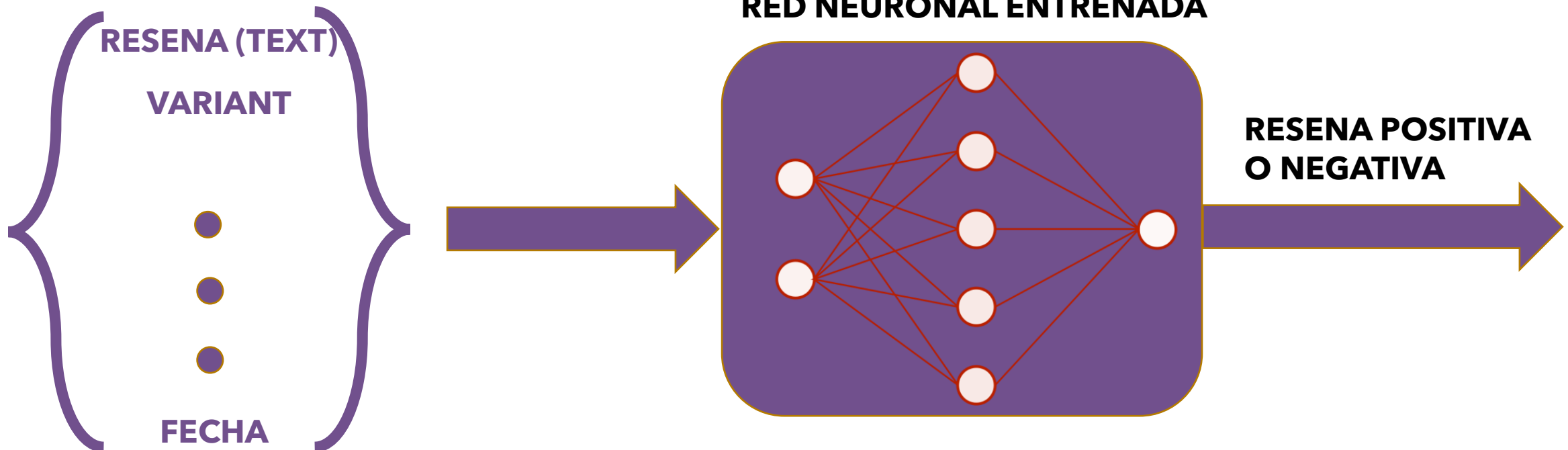



Proyecto #5: Clasificación de Texto y Análisis de Sentimiento



Proyecto #5: Clasificación de Texto y Analisis de Sentimiento

- El conjunto de datos consiste en 3000 comentarios de clientes de Amazon, calificaciones de estrellas, fecha de revisión, variante y retroalimentación de varios productos de la amazona Alexa como Alexa Echo, Echo dots.
- El objetivo es descubrir las opiniones de los consumidores y realizar un análisis de los datos.
- Conjunto de datos: www.kaggle.com/sid321axn/amazon-alexa-reviews





Tokenizacion (Vectorizador de Conteo)

Proyecto #5: Tokenizacion (Vector de Conteo)

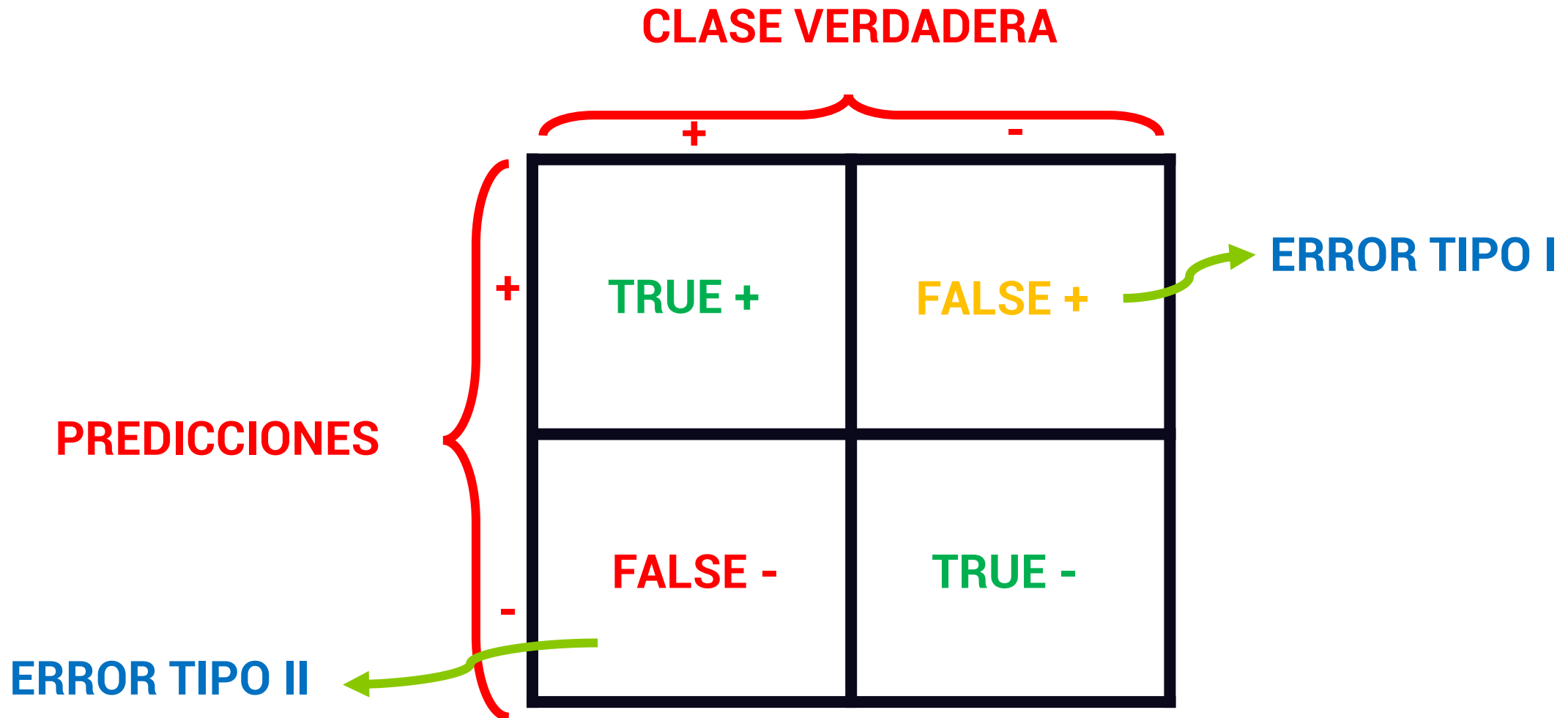
Este es el primer documento.
Este es el segundo documento.
Y este es el tercero.
¿Este es el primer documento?



```
[[0 1 1 1 0 0 1 0 1]  
[0 2 0 1 0 1 1 0 1]  
[1 0 0 1 1 0 1 1 1]  
[0 1 1 1 0 0 1 0 1]]
```

	'and'	'document'	'first'	'is'	'one'	'second'	'the'	'third'	'this'
Training Sample #1	0	1	1	1	0	0	1	0	1
Training Sample #2	0	2	0	1	0	1	1	0	1
Training Sample #3	1	0	0	1	1	0	1	1	1
Training Sample #4	0	1	1	1	0	0	1	0	1

Proyecto #5: Matriz de Confusion



Proyecto #5: Matriz de Confusion

- Se utiliza una matriz de confusión para describir el rendimiento de un modelo classification:
 - Verdaderos positivos (TP): casos en los que classifier predijo VERDADERO (tienen la enfermedad), y la clase correcta fue VERDADERA (el paciente tiene la enfermedad).
 - Negativos verdaderos (TN): casos en los que el modelo predijo FALSO (no tienen la enfermedad), y la clase correcta fue FALSA (el paciente no tiene la enfermedad).
 - Falsos positivos (FP) (Error de tipo I): classifier predijo VERDADERO, pero la clase correcta fue FALSA (el paciente no tiene la enfermedad).
 - Falsos negativos (FN) (Error de tipo II): classifier predijo FALSO (el paciente no tiene la enfermedad), pero en realidad tienen la enfermedad

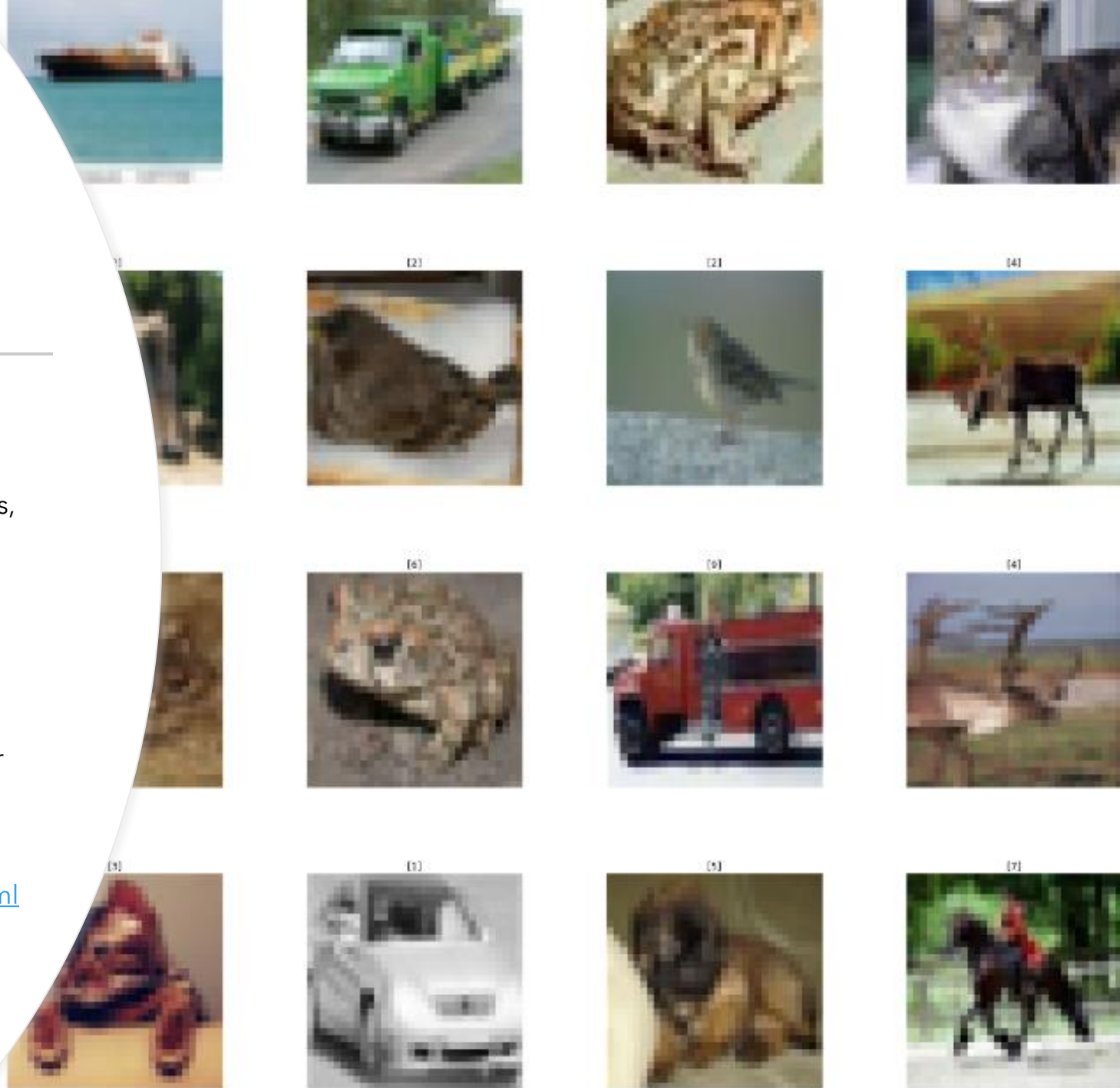
Proyecto #5: Indicadores de Rendimiento (KPI)

- Classification Precisión = $(TP+TN) / (TP + TN + FP + FN)$
- Misclassification tasa (Tasa de error) = $(FP + FN) / (TP + TN + FP + FN)$
- Precisión = $TP / \text{Total Predicciones VERDADERAS} = TP / (TP+FP)$ (Cuando el modelo predijo la clase VERDADERA, ¿con qué frecuencia fue correcta?)
- Recordatorio = $TP / \text{Real TRUE} = TP / (TP+FN)$ (Cuando la clase era realmente VERDADERA, ¿con qué frecuencia acertó el classifier?)

Proyecto #6:

Clasificación Cifar-10

- CIFAR-10 es un conjunto de datos que consiste en varias imágenes divididas en las siguientes 10 clases:
 - Aviones, coches, pájaros, gatos, ciervos, perros, ranas, caballos, barcos, camiones...
- El conjunto de datos es el Instituto Canadiense de Investigación Avanzada (CIFAR)
- CIFAR-10 se utiliza ampliamente para el aprendizaje de la máquina y las aplicaciones de visión por computador.
- El conjunto de datos consiste en 60.000 imágenes en color de 32x32, 6.000 imágenes de cada clase.
- Las imágenes tienen una baja resolución (32x32).
- Fuente de datos: <https://www.cs.toronto.edu/~kriz/cifar.html>



Proyecto #6: Clasificación Cifar-10

- El conjunto de datos consiste en 60.000 imágenes en color de 32x32, 6.000 imágenes de cada clase.

