

# Análisis Red Neuronal

Tipo:

Árbol de decisiones

La red neuronal busca los posibles resultados, costos y consecuencias de una decisión determinada. Es un modelo matemático, que lleva a cabo tareas muy complejas como el reconocimiento y la clasificación de patrones. El algoritmo deberá medir de alguna manera las predicciones logradas y valorarlas para comparar entre todas y obtener la mejor opción. para que el proceso pueda continuar al siguiente nodo debe tener una verificación mayor al 65% para que sea válido.

El árbol de decisiones está compuesto por tres diferentes capas, la primera es la capa de entrada la cual recibe las muestras para su entrenamiento , capas ocultas esta tiene como función determinar los patrones de las muestras e identificar patrones y por ultima la capa de salida la cual determina en campo de destino como se debe modificar la red neuronal para los mejores resultados posibles.

Los árboles de decisión están formados por nodos y sinapsis, estos son:

1. Nodo: Es en donde el árbol de decisión toma las decisiones y evalúa.
2. Sinapsis: Es el medio por el cual un nodo pasa información a otro modo según el criterio especificado ( Neurona presináptica -> Neurona postsináptica)

Para reducir el consumo de memoria, la complejidad y el tamaño de los árboles deben controlarse configurando esos valores para los parámetros.

$$\sum_{i=1}^m w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$

## Regresión Lineal:

La regresión lineal es un método para aproximar una regresión lineal entre dos variables, a groso modo es determinar cómo se relaciona una variable con otra variable.

Se utiliza porque al repetir iterativamente la regresión lineal se reduce el error al máximo por medio del árbol de decisiones.

# Análisis Red Neuronal

Tipo:  
Árbol de decisiones

## Librerías:

```
import numpy as np
```

[NumPy](#) es una librería de Python especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos.

Incorpora una nueva clase de objetos llamados arrays que permite representar colecciones de datos de un mismo tipo en varias dimensiones, y funciones muy eficientes para su manipulación.

La ventaja de Numpy frente a las listas predefinidas en Python es que el procesamiento de los arrays se realiza mucho más rápido (hasta 50 veces más) que las listas, lo cual la hace ideal para el procesamiento de vectores y matrices de grandes dimensiones.

```
import matplotlib.pyplot as plt
```

[Matplotlib](#) es una librería de Python especializada en la creación de gráficos en dos dimensiones.

Permite crear y personalizar los tipos de gráficos más comunes, entre ellos:

### Diagramas de líneas

- `plot(x, y)`: Dibuja un polígono con los vértices dados por las coordenadas de la lista `x` en el eje X y las coordenadas de la lista `y` en el eje Y.

# Análisis Red Neuronal

Tipo:  
Árbol de decisiones

```
from sklearn.tree import DecisionTreeRegressor
```

Sklearn.tree es una librería la cual contiene los métodos necesarios para desarrollar con facilidad y con precisión modelos de regresión de árboles de decisión.

```
# Create a random dataset

rng = np.random.RandomState(1)
X = np.sort(5 * rng.rand(80, 1), axis=0)
y = np.sin(X).ravel()
y[::5] += 3 * (0.5 - rng.rand(16))
```

RandomState: Controla la aleatoriedad de la estimadora.

Sort: Devuelve una copia ordenada de una matriz.

Rand: Valores aleatorios en una forma dada.

Sin: Es la ubicación en la que se almacena el resultado.

Ravel: Devuelve una matriz aplanada contigua.

```
# Fit regression model

regr_1 = DecisionTreeRegressor(max_depth=2)

regr_2 = DecisionTreeRegressor(max_depth=5)

regr_1.fit(X, y)

regr_2.fit(X, y)
```

Arange: Devuelve valores espaciados uniformemente dentro de un intervalo dado

Max\_depth: se refiere a la profundidad máxima del árbol.

Fit: Construya un regresor de árbol de decisión a partir del conjunto de entrenamiento (X,y).

# Análisis Red Neuronal

Tipo:  
Árbol de decisiones

```
# Predict  
  
X_test = np.arange(0.0, 5.0, 0.01)[: , np.newaxis]  
  
y_1 = regr_1.predict(X_test)  
  
y_2 = regr_2.predict(X_test)
```

Arange: Devuelve valores espaciados uniformemente dentro de un intervalo dado.

newaxis: su función es agregar una nueva dimensión a un array

Predict: Prediga el valor de clase o regresión para X.

# Análisis Red Neuronal

Tipo:  
Árbol de decisiones

```
# Plot the results

plt.figure()

plt.scatter(X, y, s=20, edgecolor="black", c="darkorange",
            label="data")

plt.plot(X_test, y_1, color="cornflowerblue", label="max_depth=2",
         linewidth=2)

plt.plot(X_test, y_2, color="yellowgreen", label="max_depth=5",
         linewidth=2)

plt.xlabel("data")

plt.ylabel("target")

plt.title("Decision Tree Regression")

plt.legend()

plt.show()
```

Figure: Cree una nueva figura o active una figura existente.

scatter: Un gráfico de dispersión de y frente a x con diferentes tamaños y/o colores de marcador.

plot: Trace y contra x como líneas y/o marcadores.

xlabel: Establezca la etiqueta para el eje x.

ylabel: Establezca la etiqueta para el eje y.

title: Establezca un título para los ejes

legend: Coloque una leyenda en los ejes.

Show: Mostrar todas las figuras abiertas.

# Análisis Red Neuronal

Tipo:

Árbol de decisiones

