RED NEURONAL EN PYTHON

Machine learning

Javier Mauricio Gil Cardona
1088347052

Las compuertas XOR:
Para el ejemplo, utilizaremos las compuertas XOR. Si no las conoces o no las recuerdas funcionan de las siguientes manera:

Tenemos dos entradas binarias (1 ó 0) y la saluda serpa 1 sólo si una de las entradas es verdadera (1) y la otra falsa (0).

Es decir que de cuatro combinaciones posibles, solo dos tiene salida 1 y las otras dos serán 0, como vemos aquí:

- XOR(0,0) = 0
- XOR(0,1) = 1
- XOR(1,0) = 1
- XOR(1,1) = 0

Una red neuronal artificial sencilla con python y Keras

Veamos el código completo en donde creamos una red neuronal con datos de entrada las 4 combinaciones de XOR y sus 4 salidas ordenadas.

```
import numpy as np
from keras.models import Sequential
from keras.layers.core import Dense

# cargamos las 4 combinaciones de las compuertas XOR
training_data = np.array([[0,0],[0,1],[1,0],[1,1]], "float32")

# y estos son los resultados que se obtienen, en el mismo orden
target_data = np.array([[0],[1],[1],[0]], "float32")

model = Sequential()
model.add(Dense(16, input_dim=2, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='mean_squared_error',
              optimizer='adam',
              metrics=['binary_accuracy'])
```

```
model.fit(training_data, target_data, epochs=1000)

# evaluamos el modelo
scores = model.evaluate(training_data, target_data)

print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
print (model.predict(training_data).round())
```

Utilizaremos Keras que es una librería de alto nivel, para que nos sea más fácil describir las capas de la red que creamos y en background es decir, el motor que ejecutará la red neuronal y la entrenará, estará la implementación de Google llamada Tensorflow, que es la mejor que existe hoy en día.

Analicemos la red neuronal

Primero importamos las clases que utilizamos :

```
import numpy as np
from keras.models import Sequential
from keras.layers.core import Dense
```

Utilizaremos numpy para el manejo de arrays, de Jeras importamos el tipo de modelo secuencial y el tipo de capa Dense que es la normal

Después creamos los arrays de entrada y salida

```
# cargamos las 4 combinaciones de las compuertas XOR
training_data = np.array([[0,0],[0,1],[1,0],[1,1]], "float32")

# y estos son los resultados que se obtienen, en el mismo orden
target_data = np.array([[0],[1],[1],[0]], "float32")
```

Como se puede ver son las cuatro entradas posibles de la función XOR [0,0], [0,1], [1,0], [1,1] y sus cuatro salidas: 0, 1,1,0

Ahora crearemos la arquitectura de nuestra red neuronal:

```
model = Sequential()
model.add(Dense(16, input_dim=2, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

Primero creamos un modelo vacío de tipo secuencial. Este modelo se refiere a que creamos una serie de capas de neuronas secuenciales, <<una delante de otra>>.

Agregamos dos capas Dense con <<model.add()>> Realmente serán 3 capas, pues al poner input_dim=2 estámos defendiendo la capa de entrada con 2 neuronas