

Red Neuronal Multicapa.

Tinoco Sergio.

Instituto Politécnico Nacional, Escuela Superior de Cómputo.

Redes neuronales y aprendizaje profundo

15 de marzo de 2023

En el presente documento se va a presentar el desarrollo de una red neuronal multicapa desarrollada con la herramienta de Google Colab.

Para ello nos vamos a basar en el diagrama de la figura 1, en la que tenemos 4 neuronas en la capa de entrada, 3 neuronas en la capa oculta y 2 neuronas en la capa de salida.

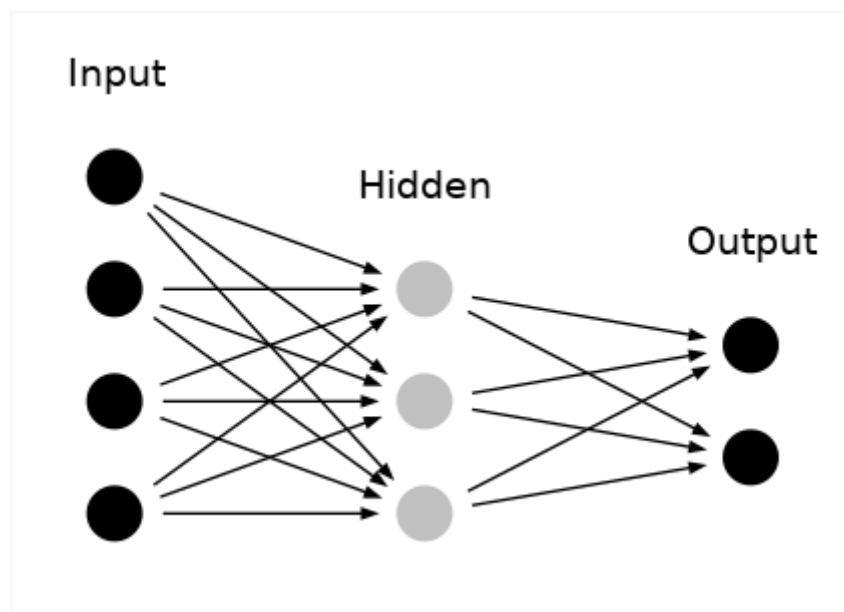


Figura 1.

Implementación.

Primeramente se definen las funciones con las cuales se va a generar el algoritmo de la red neuronal.

```
def sigmoid(x):  
    """  
    Función sigmoide  
    """  
    return 1/(1+np.exp(-x))
```

Posteriormente se definen los tamaños de las capas de la red neuronal.

```
N_input = 4  
N_hidden = 3  
N_out = 2
```

Definimos las matrices de los pesos de cada capa, así como el vector de nuestra entrada X.

Dicha matriz de pesos se va a definir de forma aleatoria siguiendo una distribución normal por medio de la función “random.normal” de Numpy.

```
w_input_to_hidden = np.random.normal(loc=0.0, scale=1.0, size=(N_input,N_hidden))  
w_hidden_out = np.random.normal(loc=0.0, scale=1.0, size=(N_hidden,N_out))  
x = [0.1,0.2,0.3,0.4]
```

Se calculan los valores de entrada para nuestra capa oculta, aplicando el producto punto y la función de activación sigmoide.

```
input_to_hidden = np.dot(X,W_input_to_hidden)
out_of_hidden = sigmoid(input_to_hidden)

print("salida de la capa oculta: ", out_of_hidden)
```

De igual forma se calculan los valores de entrada para nuestra capa de salida utilizando los valores obtenidos en la capa anterior.

```
input_to_out = np.dot(out_of_hidden,W_hidden_out)
out_of_network = sigmoid(input_to_out)

print("salida de la red: ", out_of_network)
```

Resultados

```
salida de la capa oculta: [0.38912802 0.31204542 0.44032407]
salida de la red: [0.30248918 0.50798886]
```

Como se pudo observar en los resultados, cada capa va a retornar un vector con los valores de salida de acuerdo al número de neuronas que se tengan en cada capa.

Estos valores podrían cambiar si intentamos optimizar la red neuronal a través de un método como Backpropagation.

Como conclusión puedo decir que una red neuronal multicapa funciona igual que una red neuronal simple con la pequeña diferencia de que los pesos son expresados por medio de matrices donde cada fila representa los valores de las entradas y las columnas representan el número de neuronas en la capa actual.