

Rede Neural Simples

Implementando uma RNA Simples

O diagrama abaixo mostra uma rede simples. A combinação linear dos pesos, inputs e viés formam o input h , que então é passado pela função de ativação $f(h)$, gerando o output final do perceptron, etiquetado como y .

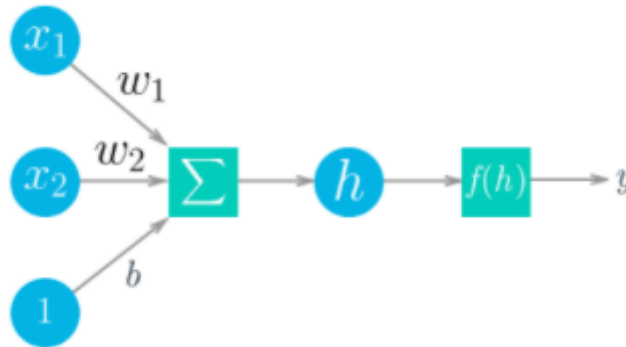


Diagrama de uma rede neural simples

Círculos são unidades, caixas são operações. O que faz as redes neurais possíveis, é que a função de ativação, $f(h)$ pode ser qualquer função, não apenas a função degrau.

Por exemplo, caso $f(h)=h$, o output será o mesmo que o input. Agora o output da rede é

$$h = \frac{1}{n} \sum_{i=1}^n (w_i * x_i) + b$$

Essa equação deveria ser familiar para você, pois é a mesma do modelo de regressão linear! Outras funções de ativação comuns são a função logística (também chamada de sigmóide), tanh e a função softmax. Nós iremos trabalhar principalmente com a função sigmóide pelo resto dessa aula:

$$f(h) = \text{sigmoid}(h) = \frac{1}{1 + e^{-h}}$$

Vamos implementar uma RNA de apenas um neurônio!

Importando a biblioteca

In [1]:

```
import numpy as np
```

Função do cálculo da sigmóide

In [2]:

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

Vetor dos valores de entrada

In [3]:

```
x = np.array([0.8,-0.3])  
b=0.1
```

Pesos das ligações sinápticas

In [4]:

```
w = np.array([0.2,-0.1])
```

Calcule a combinação linear de entradas e pesos sinápticos

In [5]:

```
h=np.dot(x,w)+b
```

Aplicado a função de ativação do neurônio

In [6]:

```
y=sigmoid(h)
```

In [7]:

```
print('A saída da rede é',y)
```

A saída da rede é 0.5719961329315186