Rede Neural Multicamadas (MPL)

Uma rede MPL é uma classe de rede neural artificial *feedforward* (ANN). Um MLP consiste em pelo menos três camadas de nós: uma camada de entrada , uma camada oculta e uma camada de saída . Exceto para os nós de entrada, cada nó é um neurônio que usa uma função de ativação não linear . O MLP utiliza uma técnica de aprendizado supervisionado chamada *backpropagation* para treinamento.

Implementando uma RNA multicamadas

A imagem a seguir mostra a nossa rede, com as unidades de entrada marcadas como Input1, Input2 e Input3 (**Input Layer**) conectadas com os *nós* da camada oculta (**Hidden Layer**). Por sua vez as saída dos *nós* da camada oculda servem como entrada para os *nós* da camada de saída (**Output Layer**).

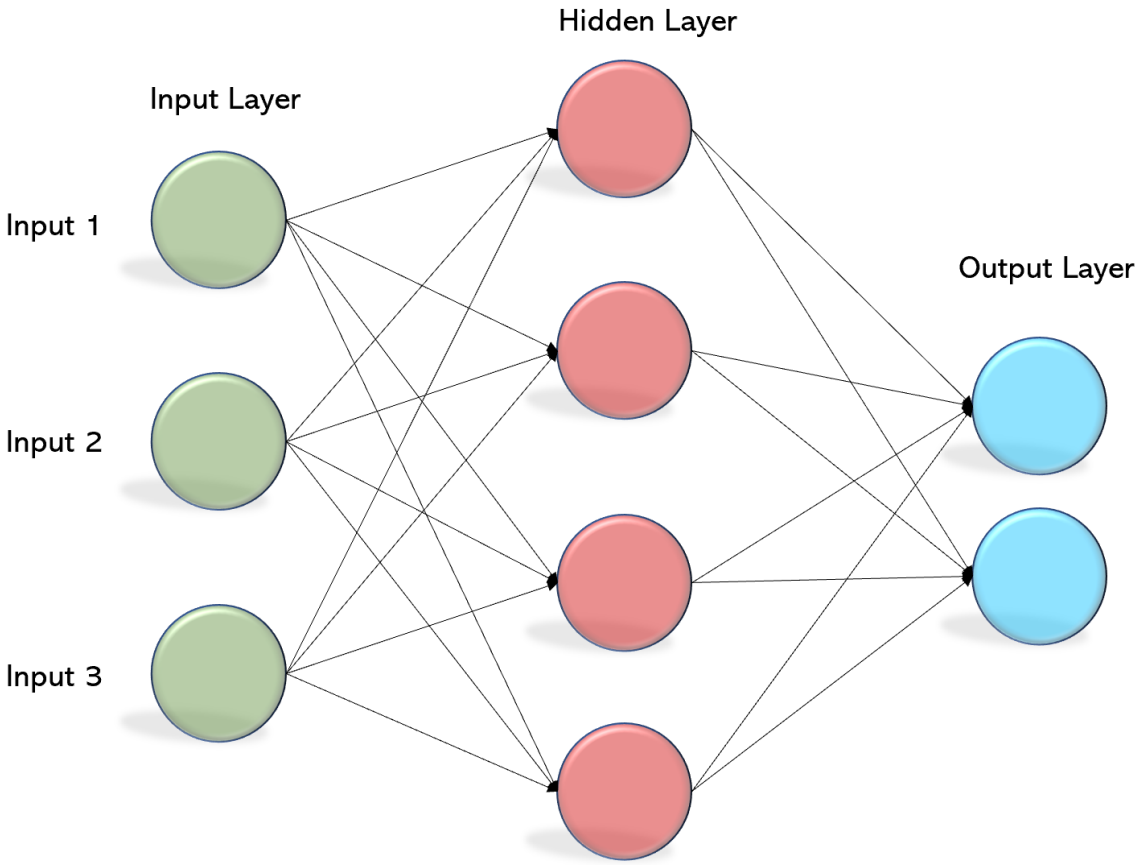

Diagrama de uma MPL

Lembrando que em cada *nó* temos:

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

onde

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

Configuração da MPL

In [1]:

```
#Importando a biblioteca
import numpy as np

#Função do cáculo da sigmóide
def sigmoid(x):
    result = (1+np.exp(-x))   # 1 + (e^ -x)
    return 1/result          # 1 / (1 + (e^ -x))

#Arquitetura da MPL
N_input = 3
N_hidden = 4
N_output = 2

#Vetor dos valores de entrada
x = np.array([0.5, 0.1, -0.2])
target =np.array([0.3, 0.8])
learnrate = 0.5

#Pesos da Camada Oculta
weights_in_hidden = np.array([[-0.07,  0.04, -0.05, 0.07],
                              [ 0.04,  0.10,  0.02, 0.01],
                              [-0.03,  0.04, -0.11, 0.06]])

#Pesos da Camada de Saída
weights_hidden_out = np.array([[-0.10,  0.09],
                               [-0.04,  0.12],
                               [-0.02,  0.04],
                               [-0.01,  0.09]])
```

Forward Pass

In [2]:

```
#Camada oculta

#Calcule a combinação linear de entradas e pesos sinápticos
hidden_layer_input = np.dot(x, weights_in_hidden)

#Aplicado a função de ativação
hidden_layer_output = sigmoid(hidden_layer_input)
```

In [3]:

```
#Camada de Saída

#Calcule a combinação linear de entradas e pesos sinápticos
output_layer_in = np.dot(hidden_layer_output, weights_hidden_out)

#Aplicado a função de ativação
output = sigmoid(output_layer_in)

print('As saídas da rede são',output)
```

```
As saídas da rede são [0.47885012 0.54255368]
```

Backward Pass

In [4]:

```
## TODO: Cálculo do Erro
error = target - output
#print('Erro da Rede: ',error)

# TODO: Calcule o termo de erro de saída (Gradiente da Camada de Saída)
output_error_term = error * output * (1 - output)

# TODO: Calcule a contribuição da camada oculta para o erro
hidden_error = np.dot(weights_hidden_out,output_error_term)

# TODO: Calcule o termo de erro da camada oculta (Gradiente da Camada Oculta)
hidden_error_term = hidden_error * hidden_layer_output * (1 - hidden_layer_output)
```

In [5]:

```
# TODO: Calcule a variação do peso da camada de saída
delta_w_h_o = learnrate * output_error_term*hidden_layer_output[:, None]
print('Variação do peso da camada de saída (delta_w_h_o): ',delta_w_h_o)
```

```
Variação do peso da camada de saída (delta_w_h_o):  [[-0.01101866  0.0
1577419]
 [-0.01128087  0.01614955]
 [-0.01115255  0.01596586]
 [-0.01129202  0.01616553]]
```

In [6]:

```
# TODO: Calcule a variação do peso da camada oculta
delta_w_i_h = learnrate * hidden_error_term * x[:, None]
print('Variação do peso da camada oculta (delta_w_i_h): ',delta_w_i_h)
```

```
Variação do peso da camada oculta (delta_w_i_h):  [[ 6.38265149e-04
5.90725285e-04  2.15529088e-04  3.87251147e-04]
 [ 1.27653030e-04  1.18145057e-04  4.31058176e-05  7.74502295e-05]
 [-2.55306060e-04 -2.36290114e-04 -8.62116351e-05 -1.54900459e-04]]
```

Atualização dos Pesos

In [7]:

```
##Atualização dos Pesos
weights_input_hidden = learnrate * delta_w_i_h
print('weights_input_hidden: ',weights_input_hidden)
weights_hidden_output = learnrate * delta_w_h_o
print('weights_hidden_output: ',weights_hidden_output)
```

```
weights_input_hidden:  [[ 3.19132575e-04  2.95362642e-04  1.07764544e-
04  1.93625574e-04]
 [ 6.38265149e-05  5.90725285e-05  2.15529088e-05  3.87251147e-05]
 [-1.27653030e-04 -1.18145057e-04 -4.31058176e-05 -7.74502295e-05]]
weights_hidden_output:  [[-0.00550933  0.00788709]
 [-0.00564043  0.00807478]
 [-0.00557628  0.00798293]
 [-0.00564601  0.00808276]]
```