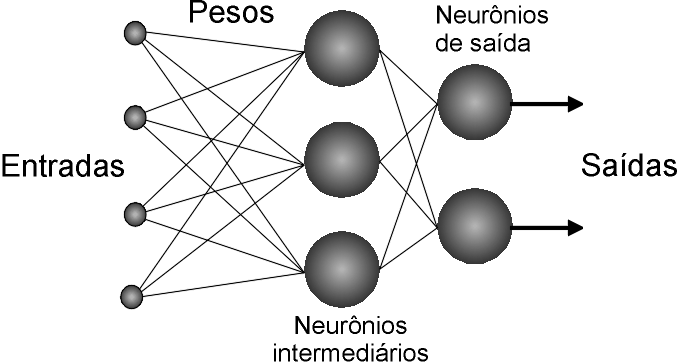
# Cálculo de Pesos e Erros

1 – O Algoritmo inicializa os pesos com valores aleatórios. Veja a figura para ver o papel dos Pesos dentro de uma Rede Neural.

Quanto maior a quantidade de entradas, maior o tempo de processamento de entradas e saídas, por conta disto, segundo o professor do curso da *Udemy*, estes sistemas são executados em servidores especializados para isto. Por exemplo, dada 50 entradas, será avaliada todas suas saídas e valores da camada oculta.



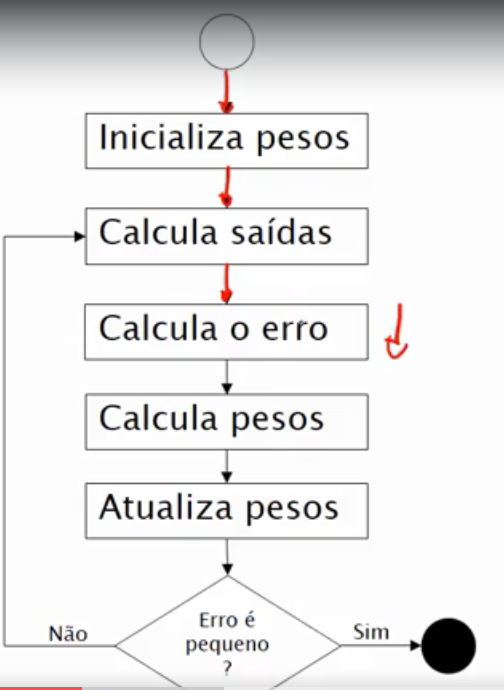
2 – Baseando se nos dados (aprendizagem supervisionada, pois já sabemos qual é a saída correta), realiza os cálculos com os pesos e calcula o erro.

Ex: Resposta Correta = 1, Resultado = 0,5. Eu queria chegar no 1 e eu adquiri 0,5 então o cáculo do erro é ( 1 – 0,5 ) = 0,5. Se o que eu adquiri for negativo, ele é colocado como positivo na hora da subtração (Abs).

3 – Calcula as mudanças nos pesos e os atualiza (Processo de *Backpropagation*).

4 – O algoritmo termina quando a taxa de erro for pequena, ou seja, o algoritmo não termina quando necessariamente a taxa de erro for ZERO.

Resumindo como é feito o Cálculo de erros e Pesos, aqui está o diagrama de blocos:

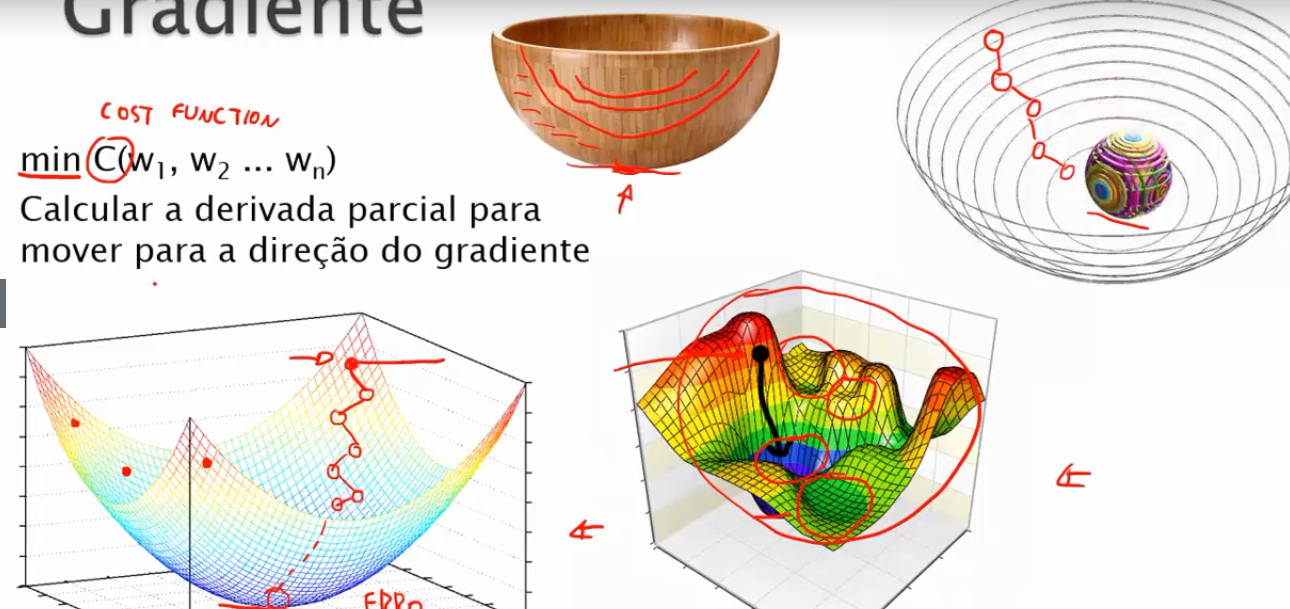


*Cost Function* – É a função de minimizar o Erro, ocorre na hora de calcular o erro.

* Para a atualização de pesos, é utilizada:
* Gradiente (*gradient descent*)
* Derivada
* Cálculo do Delta
* *BackPropagation*
* *Learning Rate* (Taxa de Aprendizagem)
* *Momentum* (Momento)

Gradiente

A finalidade é encontrar a combinação de pesos que o erro é o menor possível. Gradiente é calculado para saber **quanto ajustar os pesos.**



Força Bruta

- Fazer todas as combinações possíveis. Porém nada viável atualmente, por conta do grande tempo que vai ser levado para encontrar uma combinação de peso com o menor erro possível. Apesar dele trazer resultados mais corretos.

*Simmulated Anealing*

Algoritmos genéticos

- Algoritmos de otimização para encontrar valores. Muito conhecido na área de inteligência artificial.