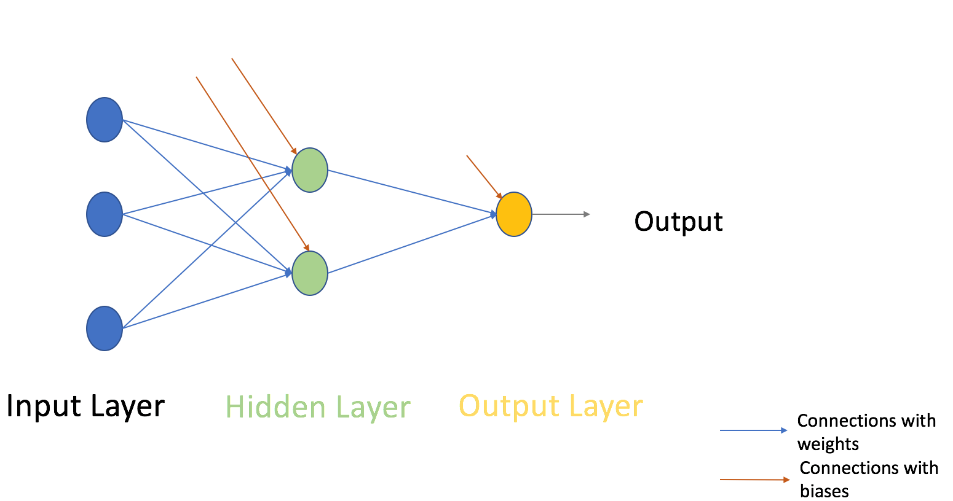
**Passos em uma rede neural em Numpy**



1. Teremos como entrada e saída:
   1. X como matriz de entrada
   2. Y como matriz de saída
2. Iniciaremos os pesos e os bias com valores aleatórios (Somente no início, na próxima interação serão atualizados):
   1. Wh como a matriz de peso na camada escondida
   2. Bh como a matriz de bias na camada escondida
   3. Wout como matriz de peso na camada de saída
   4. Bout como a matriz de bias na camada de saída
3. Nós teremos o produto da matriz como entrada e os pesos associados entre a entrada e a camada escondida, então adicionaremos os bias nos neurônios da camada escondida na respectiva entrada, isso é conhecido como **transformação linear**:
   1. (hidden\_layer\_input= matrix\_dot\_product(X,wh) + bh)
4. Realizar a transformação não-linear usando a função de ativação (Sigmoid). Sigmoid vai retornar a saída como 1/(1 + exp(-x)).
   1. (hiddenlayer\_activations = sigmoid(hidden\_layer\_input)