

# Relatório CARD 12 - Prática: Redes Neurais (II)

Willian Augusto Soder de Souza

O objetivo deste relatório é abordar os principais conhecimentos adquiridos ao assistir às seis seções do curso “Deep Learning com Python de A a Z - O Curso Completo”. Inicialmente, o apresentador mostra diversos conceitos teóricos importantes sobre redes neurais, e, posteriormente, as outras seções demonstram a implementação prática (cujos códigos estão em anexo). Neste relatório, explicarei os conceitos teóricos apresentados, seguidos de um breve resumo sobre cada um deles.

- **Neurônio Artificial:** é um componente básico de redes neurais artificiais. Ele recebe múltiplas entradas, cada uma ponderada por um valor específico, calcula uma soma ponderada dessas entradas e a passa por uma função de ativação para produzir uma saída. Essa saída pode ser usada como entrada para outros neurônios em camadas subsequentes da rede, permitindo a construção de modelos capazes de aprender e generalizar a partir de dados, sendo amplamente utilizados em tarefas de reconhecimento de padrões, classificação, e previsão.
- **Função de ativação:** são componentes cruciais em redes neurais artificiais, responsáveis por introduzir não-linearidade ao modelo, permitindo que a rede aprenda e represente relações complexas nos dados. Sem elas, uma rede neural, independentemente do número de camadas, seria equivalente a uma simples transformação linear, incapaz de resolver problemas complexos. As principais e mais usadas funções de ativação são: Sigmóide (Produce saídas no intervalo  $(0, 1)$ ), ReLU (Produce saídas no intervalo  $(0, \infty)$ ), TanH (Produce saídas no intervalo  $(-1, 1)$ ).
- **Camadas:** Em redes neurais, existem três tipos de camadas: a camada de entrada, as camadas ocultas e a camada de saída. A camada de entrada é a primeira camada que recebe os dados brutos, com cada neurônio correspondendo a uma característica do conjunto de dados. As camadas ocultas são as intermediárias entre a camada de entrada e a camada de saída, onde ocorre o processamento principal; elas podem ser uma ou várias, e é nelas que as transformações não lineares, realizadas pelas funções de ativação, permitem a rede aprender representações complexas dos dados. A camada de saída é a última camada que produz o resultado final da rede, como uma classificação ou predição, com a quantidade de neurônios geralmente correspondendo ao número de classes no problema de classificação ou ao formato desejado da predição no caso de regressão.

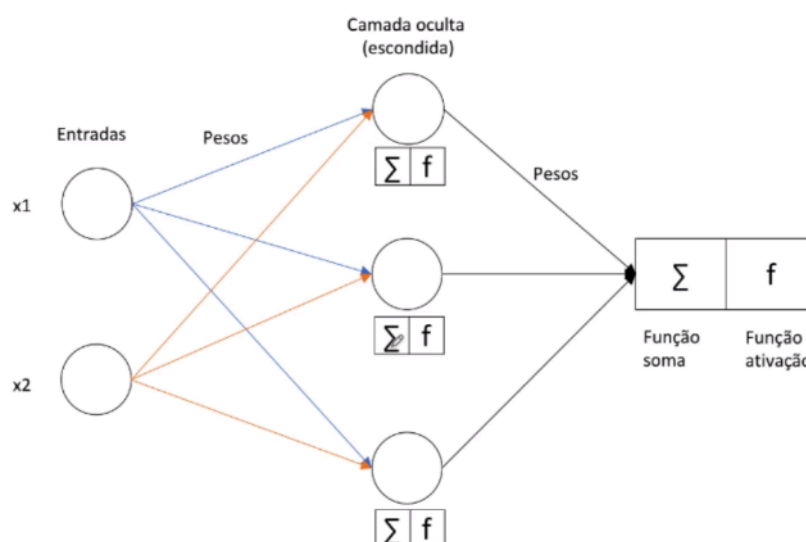


Figura 1 - Rede-Neural estrutura básica.

- **Erro:** o cálculo do erro em redes neurais artificiais é fundamental para o processo de aprendizado e é feito comparando a saída da rede com o valor real esperado, utilizando uma função de perda ou custo. Para cada exemplo de treinamento, a rede produz uma saída que é comparada com o valor alvo, e a diferença entre esses valores é calculada pela função de perda, como o erro quadrático médio (MSE) para problemas de regressão ou a entropia cruzada para problemas de classificação. O objetivo é minimizar esse erro ajustando os pesos da rede. Isso é feito utilizando o algoritmo de retropropagação (backpropagation), que calcula o gradiente do erro em relação a cada peso e usa métodos como o gradiente descendente para atualizar os pesos, reduzindo assim o erro de forma iterativa através das épocas de treinamento.
- **Descida do Gradiente:** é um algoritmo de otimização usado para ajustar os pesos da rede e minimizar a função de perda. Após calcular o erro entre a previsão da rede e o valor real, a retropropagação é usada para determinar os gradientes da função de perda em relação a cada peso. Esses gradientes indicam a direção de maior aumento do erro, e a descida do gradiente ajusta os pesos na direção oposta para reduzir o erro, com a magnitude das atualizações controlada por um hiperparâmetro chamado taxa de aprendizado. Esse processo é repetido iterativamente para muitos exemplos de treinamento, ajustando os pesos progressivamente para melhorar a performance da rede.
- **Ajustes de pesos:** é o processo de atualizar os pesos das conexões entre neurônios para minimizar o erro de previsão, com base nos gradientes calculados pela retropropagação.
- **Bias:** são termos adicionados a cada neurônio para permitir que o modelo se ajuste mesmo quando todas as entradas são zero, aumentando a flexibilidade da rede e melhorando sua capacidade de aprendizado.
- **Validação cruzada:** é uma técnica usada em redes neurais artificiais para avaliar a performance do modelo, dividindo o conjunto de dados em múltiplos subconjuntos; o modelo é treinado em alguns desses subconjuntos e testado nos restantes, rotacionando essa divisão várias vezes, para garantir que o modelo generalize bem a dados não vistos e evitar overfitting.
- **Overfitting:** ocorre quando o modelo aprende muito bem os detalhes e ruídos do conjunto de treinamento, resultando em excelente performance nos dados de treinamento, mas baixo desempenho em novos dados, devido à falta de generalização.
- **Underfitting:** ocorre quando o modelo é muito simples para capturar os padrões existentes nos dados, resultando em baixo desempenho tanto nos dados de treinamento quanto nos dados de teste, indicando que o modelo não está representando adequadamente a complexidade do problema.

## CONCLUSÃO:

Em resumo, este material foi muito importante para o entendimento de redes neurais, pois além de explicar bem os conceitos teóricos com gráficos e exemplos sobre o funcionamento das funções de ativação, pesos, erro, entre outros, também detalhou como aplicar esses conceitos no Python usando as bibliotecas Keras e scikit-learn. O apresentador destacou pontos cruciais, desde o pré-processamento dos dados com técnicas como Label Encoder (que transforma valores categóricos em números inteiros) e One Hot Encoder (que transforma valores categóricos em uma matriz binária), até a criação da rede (incluindo a quantidade de neurônios e as melhores funções de ativação), treinamento e avaliação. Os exemplos fornecidos foram muito didáticos, e o apresentador utilizou diversas bases de dados para reforçar o entendimento desses fundamentos, que são extremamente importantes para uma carreira em machine learning e data science.