Relatório sobre a atividade "12 - Prática: Redes Neurais (II)"

Lucas Gabriel Arenhardt

1. Introdução

Neste módulo, foram abordados conceitos de redes neurais artificiais e aprendizado de máquina. O foco principal foi compreender a estrutura e o funcionamento das redes neurais, bem como os processos envolvidos no treinamento desses modelos. As aulas exploraram a definição de neurônios artificiais, a importância das funções de ativação, as diferentes camadas de uma rede neural, e os métodos utilizados para minimizar o erro, como backpropagation e algoritmos de descida do gradiente. Este relatório visa consolidar os conceitos apresentados, fornecendo uma visão dos principais componentes e processos envolvidos na construção e otimização de redes neurais.

2. Conceitos

2.1. Neurônio Artificial

O neurônio artificial é a unidade básica de processamento de uma rede neural. Inspirado no funcionamento dos neurônios biológicos, ele recebe sinais de entrada, aplica pesos a esses sinais, soma os resultados e, em seguida, passa essa soma por uma função de ativação para gerar a saída.

2.2. Função de Soma

A função de soma realiza a agregação ponderada das entradas recebidas pelo neurônio, somando-as após multiplicar cada uma pelo seu respectivo peso. Essa soma ponderada é então passada à função de ativação para gerar a saída do neurônio. A equação que descreve a função de soma é:

$$soma = \sum_{i=1}^{n} xi * wi$$

2.3. Função de Ativação

A função de ativação introduz não-linearidade no modelo, permitindo que a rede neural aprenda e represente funções complexas. Entre as funções de ativação mais comuns estão: sigmóide, tangente hiperbólica e ReLU.

2.4. Camadas

As camadas são grupos de neurônios organizados de maneira sequencial ou hierárquica, e cada camada realiza uma transformação específica nos dados de entrada. As camadas principais em uma rede neural são:

2.4.1. Camada de Entrada

A camada de entrada é responsável por receber os dados brutos do problema. Não realiza processamento, apenas encaminha os dados para as camadas seguintes.

2.4.2. Camadas Ocultas

As camadas ocultas processam as informações recebidas e extraem características relevantes. Cada neurônio em uma camada oculta aplica a função de soma e a função de ativação às suas entradas, gerando saídas que são usadas como entradas para a próxima camada. Essas camadas são essenciais para a capacidade da rede neural de aprender representações complexas dos dados.

2.4.3. Camada de Saída

A camada de saída gera o resultado final da rede neural, que pode ser uma classificação, uma regressão ou outra forma de previsão, dependendo do problema em questão. A escolha da função de ativação na camada de saída depende do tipo de tarefa: por exemplo, a função softmax é frequentemente usada em problemas de classificação multiclasse. Já a função sigmóide é geralmente usada em problemas de classificação binária.

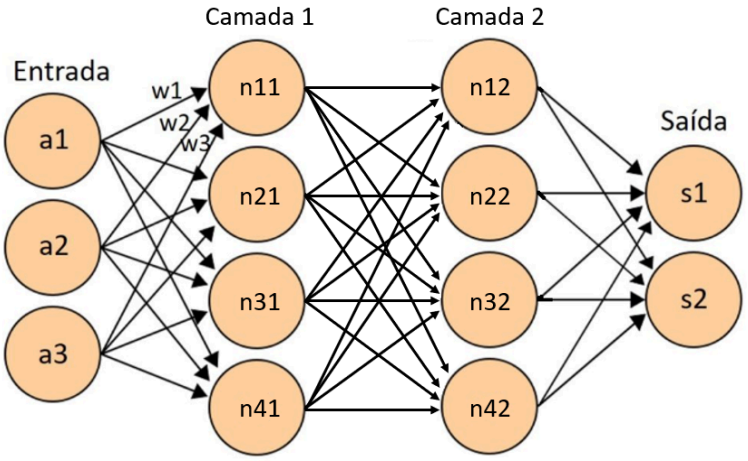

Imagem 1 - Representação de uma rede neural.
As camadas 1 e 2 correspondem às camadas ocultas.

2.5. Erro

O erro, ou perda, mede a discrepância entre as previsões da rede neural e os valores reais. A função de perda é crucial para o treinamento da rede, pois orienta o processo de ajuste dos pesos. Um exemplo de função de perda é: Mean-Squared Error (MSE), calculado através da fórmula $MSE = \frac{1}{N}\sum_{i=1}^{N}(fi - yi)^2$. Possui uma utilidade muito grande, pois ao elevar o resultado ao quadrado, penaliza mais os erros.

2.6. Backpropagation

O algoritmo de backpropagation é utilizado para minimizar o erro da rede neural, ajustando os pesos de maneira eficiente. Esse processo envolve duas fases principais: a propagação direta, onde os dados de entrada são passados pela rede para calcular a saída, e a propagação para trás, onde o erro é propagado de volta através da rede para atualizar os pesos.

A atualização dos pesos é realizada utilizando métodos de descida do gradiente, que ajustam os pesos na direção oposta ao gradiente da função de perda em relação aos pesos. Existem várias variações da descida do gradiente:

2.6.1. Stochastic Gradient Descent (SGD)

No SGD, os pesos são atualizados para cada exemplo de treinamento individualmente. Isso significa que para cada amostra no conjunto de dados, a rede

neural calcula o gradiente da função de perda e atualiza os pesos. O SGD é considerado eficiente em termos de computação e ajuda a prevenir mínimos locais.

2.6.2. Batch Gradient Descent (BGD)

O BGD calcula o gradiente da função de perda utilizando todo o conjunto de dados de treinamento antes de atualizar os pesos. Esse método pode ser computacionalmente intensivo para grandes conjuntos de dados, já que cada iteração requer a passagem por todos os exemplos de treinamento.

2.6.3. Mini-Batch Gradient Descent

Este método combina os benefícios do SGD e do BGD, atualizando os pesos após calcular o erro em pequenos lotes de amostras. Isso proporciona um bom equilíbrio entre eficiência computacional e estabilidade na atualização dos pesos.

2.7. Bias

O bias é uma entrada adicional de valor “1” e sua função é ajustar a função de ativação. Isso torna a rede neural mais flexível, de forma a modelar corretamente os dados (especialmente quando as entradas são todas zero).

2.8. Underfitting

Underfitting ocorre quando o modelo não é complexo o suficiente para capturar as relações nos dados de treinamento. Isso resulta em um desempenho ruim tanto no treinamento quanto na validação. Modelos que sofrem de underfitting geralmente precisam de mais complexidade, seja na forma de mais características, mais camadas ocultas ou ajustes nos hiperparâmetros.

2.9. Overfitting

Overfitting ocorre quando o modelo é excessivamente ajustado aos dados de treinamento, capturando não apenas os padrões gerais, mas também os ruídos específicos do conjunto de treinamento. Isso resulta em um ótimo desempenho no treinamento, mas um desempenho fraco na validação e em novos dados. Técnicas como regularização, dropout e validação cruzada são usadas para combater o overfitting.

3. Conclusão

Este módulo foi essencial para aprofundar o entendimento sobre a construção e funcionamento de redes neurais artificiais. As aulas foram ilustradas com gráficos, imagens e exemplos de código, o que facilitou a compreensão dos conceitos abordados. A prática com diferentes tipos de algoritmos reforçou a aplicação dos conhecimentos teóricos. A abordagem didática do apresentador, que incluiu a utilização das bibliotecas Keras e scikit-learn, e a consulta à documentação quando necessário, tornou o processo de aprendizado mais acessível e prático. Isso permitiu uma melhor assimilação dos conteúdos e uma aplicação mais eficaz das técnicas de aprendizado de máquina em diversos problemas.