Introdução à redes neurais

O que é uma Rede Neural?

Uma rede neural é um modelo computacional inspirado na estrutura e funcionamento dos neurônios biológicos. Em termos simples, uma rede neural tenta imitar a forma como o cérebro humano processa informações. Os neurônios biológicos recebem sinais através das dendrites, processam esses sinais no corpo celular e enviam o resultado através do axônio para outros neurônios. De forma análoga, uma unidade computacional ou "neurônio" em uma rede neural recebe entradas, processa essas entradas e produz uma saída.

Como Funciona um Neurônio Computacional?

Um neurônio em uma rede neural consiste em várias partes:

Camada de Entrada

- Recebe os dados de entrada (features).

Pesos

- Cada entrada é multiplicada por um peso. Os pesos são ajustáveis e determinam a importância de cada entrada.

Função Soma

- As entradas ponderadas são somadas para produzir um único valor.

Função de Ativação

- A soma ponderada é passada por uma função de ativação que decide se o neurônio deve ser ativado ou não. As funções de ativação comuns incluem ReLU (Rectified Linear Unit), sigmoid e tanh.

Exemplo de um Neurônio

Entrada (x) -> Pesos (w) -> Função Soma (Σ) -> Função de Ativação (f) -> Saída (y)

Perceptron e Limitações

O que é um Perceptron?

- Um perceptron é o tipo mais simples de rede neural, consistindo de uma única camada de neurônios.
- Representa uma função linear que divide o espaço de entrada em duas partes.

Função Soma no Espaço

- A função soma no perceptron pode ser vista como um hiperplano no espaço de entrada que separa duas classes.

Limitações de um Único Neurônio

- Um único neurônio não pode resolver problemas que não são linearmente separáveis, como o problema do XOR.

Correção do Neurônio

Predição Errada

- Quando a predição de um neurônio está errada, os pesos são ajustados para melhorar a predição futura.

Ajuste dos Pesos

- O ajuste dos pesos é feito usando um algoritmo de aprendizado, como o gradiente descendente. A fórmula básica de atualização dos pesos é:

$$w_i = w_i + \Delta w_i$$

$$\Delta w_i = \eta * (y - \hat{y}) * x_i$$

Onde:

- `w_i` é o peso ajustado.
- `Δw_i` é a mudança no peso.
- `η` é a taxa de aprendizado.
- `y` é o valor alvo.
- `ŷ` é a predição do modelo.
- `x_i` é a entrada correspondente.

Redes Neurais Simples e Deep Learning

Redes Neurais Simples

- Consistem em uma camada de entrada, uma camada oculta e uma camada de saída.
- Usadas para problemas relativamente simples.

Redes Neurais de Deep Learning

- Consistem em várias camadas ocultas (Deep Learning).
- Capazes de capturar características complexas dos dados.

Tipos de Arquiteturas de Redes Neurais

Exemplos de Arquiteturas

- **Redes Neurais Convolucionais (CNNs)**: Usadas principalmente para tarefas de visão computacional.
- **Redes Neurais Recorrentes (RNNs)**: Usadas para processamento de sequências, como em NLP.
- **Redes Adversariais Generativas (GANs)**: Usadas para gerar novos dados sintéticos.

Função de Perda (loss)

O que é Função de Perda?

- Uma função de perda mede o quão bem o modelo está se saindo. É uma medida de erro entre a predição do modelo e o valor real.

Exemplos de Funções de Perda

- **Erro Quadrático Médio (MSE)**: Média dos quadrados das diferenças entre as predições e os valores reais.
- **Erro Absoluto Médio (MAE)**: Média das diferenças absolutas entre as predições e os valores reais.
- **Erro Logarítmico (Log Loss)**: Usado principalmente em classificações binárias.

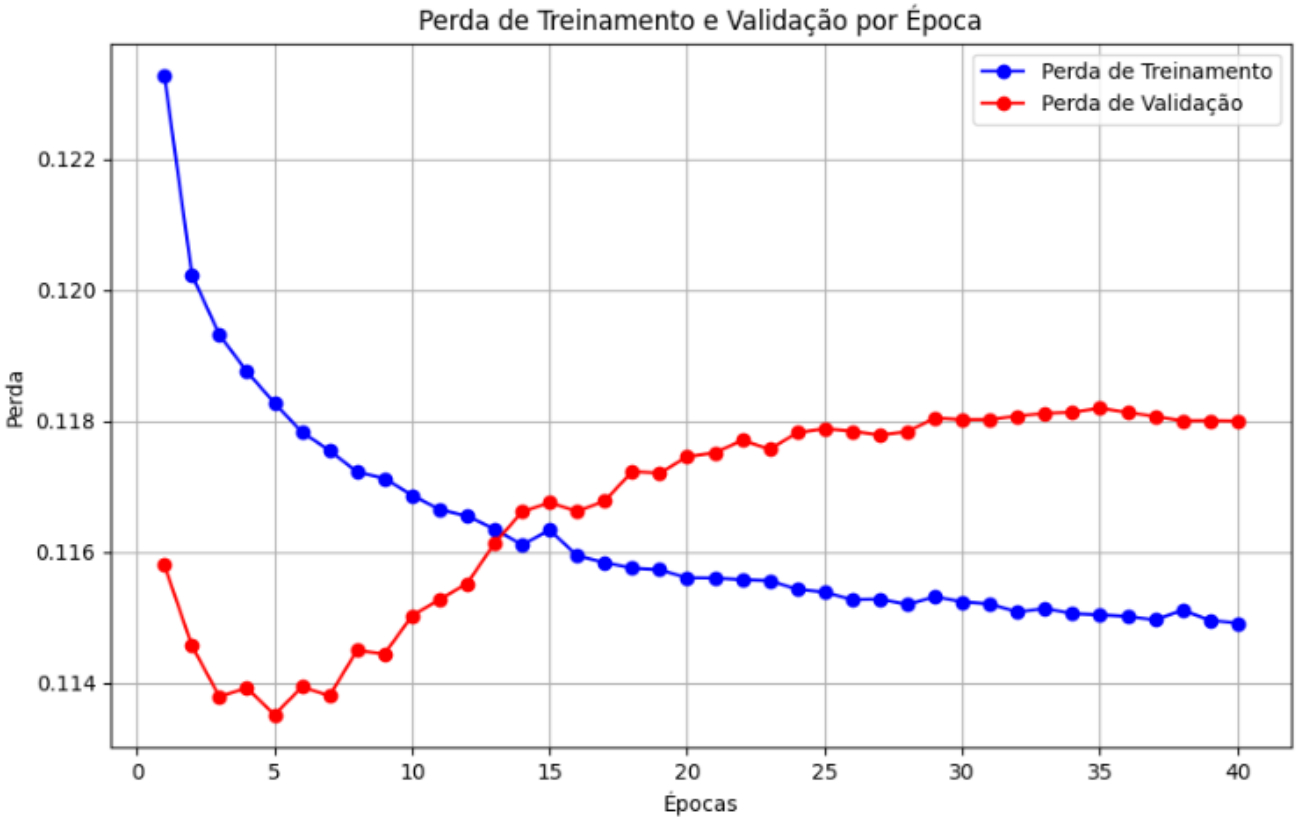
Loss de Treinamento:

- **O que é**:
  - A loss de treinamento é a medida do erro do modelo nos dados de treinamento durante cada época do treinamento.
- **Propósito**:
  - Ela é usada pelo otimizador para ajustar os pesos da rede neural durante o treinamento, com o objetivo de minimizá-la.
- **Tendência**:
  - Espera-se que a loss de treinamento diminua ao longo das épocas. Isso indica que o modelo está aprendendo com sucesso os padrões nos dados de treinamento.
- **Possíveis Problemas**:

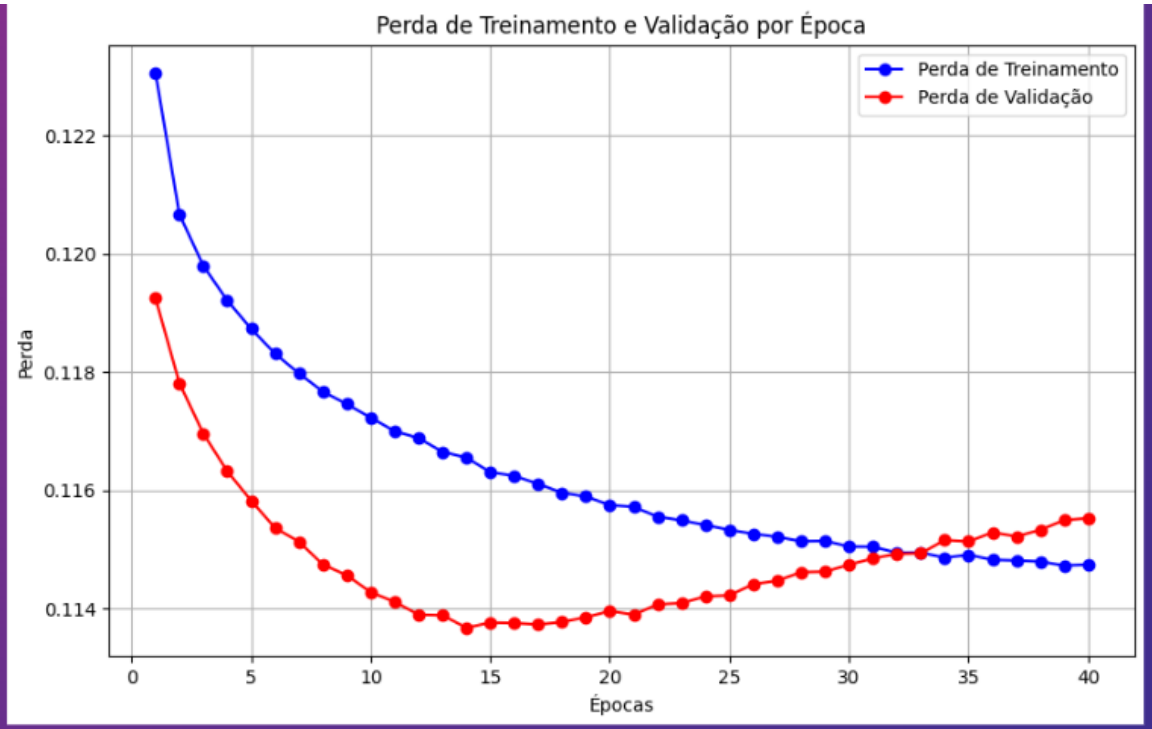
    - Se a loss de treinamento aumenta continuamente, pode indicar que o modelo não está sendo treinado corretamente ou que há problemas com o conjunto de dados ou a arquitetura do modelo.

Loss de Validação:

- **O que é**:
    - A loss de validação é a medida do erro do modelo em um conjunto de dados de validação, que são dados que o modelo não viu durante o treinamento.
- **Propósito**:
    - Ela é usada para avaliar a capacidade do modelo de generalizar para dados não vistos e detectar overfitting.
- **Tendência**:
    - Idealmente, a loss de validação deve diminuir à medida que o modelo aprende com os dados de treinamento, mas depois pode começar a aumentar. Isso é normal e esperado, pois indica que o modelo está começando a se ajustar demais aos dados de treinamento e perdendo a capacidade de generalizar para novos dados.
- **Possíveis Problemas**:
    - Se a loss de validação continuar a diminuir, enquanto a loss de treinamento diminui, isso pode indicar que o modelo não está sendo treinado por tempo suficiente para alcançar sua capacidade máxima. No entanto, se a loss de validação começar a aumentar, pode indicar overfitting, sugerindo que o treinamento deve ser interrompido para evitar ajuste excessivo aos dados de treinamento.




```
# ReLU(x) = max(0, x): Para uma entrada x, a saída será x se x
hide = tf.keras.layers.Dense(100, activation='relu')(inp)
hide = tf.keras.layers.Dense(100, activation='relu')(hide)
```




```
hide = tf.keras.layers.Dense(100, activation='relu')(inp)
```

Época

- Uma época é uma passagem completa por todo o conjunto de dados de treinamento. Durante uma época, o modelo de rede neural vê cada exemplo de treinamento exatamente uma vez e ajusta os pesos da rede com base nesses exemplos.

Interpretação:

- **Convergência**:
  - Se tanto a loss de treinamento quanto a loss de validação estiverem diminuindo, isso indica que o modelo está aprendendo efetivamente e não está sofrendo de overfitting significativo.
- **Overfitting**:
  - Se a loss de treinamento continuar a diminuir, mas a loss de validação começar a aumentar, isso indica overfitting e sugere que o treinamento deve ser interrompido para evitar piora do desempenho do modelo em dados não vistos.
- **Underfitting**:
  - Se tanto a loss de treinamento quanto a loss de validação estiverem altas e não estiverem diminuindo, isso pode indicar underfitting, onde o modelo não está sendo capaz de aprender efetivamente com os dados de treinamento.

Portanto, monitorar tanto a loss de treinamento quanto a loss de validação durante o treinamento é crucial para entender o comportamento do modelo e ajustar o treinamento conforme necessário para obter um modelo bem generalizado.
Gradiente de uma Rede Neural

O que é o Gradiente?

- O gradiente é uma medida de quanto a função de perda muda em relação a uma pequena mudança nos pesos.
- Usado no algoritmo de gradiente descendente para minimizar a função de perda ajustando os pesos.

Exemplo Prático simples com TensorFlow (TF)

Ex. Rede Neural simples com Tensor Flow

Exemplo mais complexo (Resnet50)

Ex. mais complexo (Resnet50)