

Universidade Federal do Rio de Janeiro  
Instituto Alberto Luiz Coimbra de  
Pós-Graduação e Pesquisa de Engenharia



Programa de Engenharia de Sistemas e  
Computação

CPS769 - Introdução à Inteligência Artificial e Aprendizagem Generativa

Prof. Dr. Edmundo de Souza e Silva (PESC/COPPE/UFRJ)

Profa. Dra. Rosa M. Leão (PESC/COPPE/UFRJ)

Participação Especial: Gaspare Bruno (Diretor Inovação, ANLIX)

*Lista de Exercícios 1b*

Luiz Henrique Souza Caldas

email: lhscaldas@cos.ufrj.br

10 de julho de 2024

## Questão 1

O objetivo deste trabalho é entender como um perceptron com duas entradas e uma entrada de bias classifica pontos em um espaço 2-D. Você usará duas funções de ativação diferentes: ReLU e Sigmoid.

1. Implemente um perceptron com duas entradas e uma entrada de bias.
2. Gere um conjunto de dados de pontos em um espaço 2D. Os pontos devem ser classificados em duas classes com base em suas coordenadas.
3. Treine o perceptron em um conjunto de dados de pontos em um espaço 2-D (escolha).
4. Use duas funções de ativação diferentes (Rectified Linear Unit (ReLU) e Sigmoid) para classificar os pontos.
5. Visualize os limites de decisão para ambas as funções de ativação.

Responda às seguintes perguntas com base no programa Python que você deverá fazer, e em suas observações:

1. Explique o processo de geração de dados no programa. Como os pontos são classificados em duas classes?

### Resposta:

No programa, os dados de treinamento são gerados usando a classe `DataGenerator` (código no final deste relatório). Essa classe gera um conjunto de pontos aleatórios em um espaço 2D com coordenadas entre 0 e 1. Uma linha aleatória é gerada com uma inclinação (slope) e uma interceptação (intercept) aleatórias.

Cada ponto é então classificado com base na posição relativa à linha. Especificamente, os pontos que estão acima da linha (onde a coordenada  $y$  é maior que a soma da inclinação vezes a coordenada  $x$  mais a interceptação) são classificados como pertencentes à classe 1, enquanto os pontos abaixo da linha são classificados como pertencentes à classe 0. Isso resulta em um conjunto de dados com pontos claramente divididos em duas classes.

2. Qual é o papel da função de ativação no perceptron? Compare as funções de ativação ReLU e Sigmoid.

### Resposta:

A função de ativação no perceptron determina a saída do neurônio com base na soma ponderada de suas entradas. Ela introduz não-linearidade no modelo, permitindo que ele resolva problemas mais complexos.

Comparação entre ReLU e Sigmoid:

- ReLU (Rectified Linear Unit): Retorna a entrada diretamente se for positiva; caso contrário, retorna zero. É computacionalmente eficiente e ajuda a resolver o problema do desvanecimento do gradiente, comum em redes profundas.

- Sigmoid: Retorna um valor entre 0 e 1, mapeando a soma ponderada de entradas para uma curva em forma de "S". É útil para problemas onde a saída precisa ser interpretada como uma probabilidade, mas pode sofrer com o desvanecimento do gradiente em redes profundas.
3. Treine o perceptron com funções de ativação ReLU e Sigmoid. Mostre os pesos finais para ambos os casos.  
**Resposta:**
  4. Trace os limites de decisão para ambas as funções de ativação. Descreva quaisquer diferenças que você observar.  
**Resposta:**
  5. Como as funções de ativação ReLU e Sigmoid afetam a capacidade do perceptron de classificar os pontos?  
**Resposta:**
  6. Como o número de iterações para a aprendizagem afeta o desempenho do perceptron e o limite de decisão?  
**Resposta:**
  7. Quais são algumas limitações potenciais do uso de um perceptron de camada única para tarefas de classificação? Sugira possíveis melhorias.  
**Resposta:**
  8. Seria possível fazer o treinamento da lista anterior apenas aumentando o número de neurônios de 1 para N? Explique de acordo com os artigos que você leu.  
**Resposta:**

## Código

Os códigos abaixo encontram-se no repositório <https://github.com/lhscaldas/cps769-ai-gen>, bem como o arquivo LaTeX com o relatório.