

Aplicação de Redes Neurais Multicamadas (MLP) na Previsão de Desempenho Estudantil

Alunos: Rebecca Beccari e João Bender

1. Introdução

O trabalho tem como objetivo aplicar conceitos de Inteligência Artificial, especificamente Redes Neurais Artificiais, para resolver um problema de classificação binária em um contexto educacional. O tema escolhido foi a **Educação Preditiva**, buscando antecipar o resultado acadêmico de estudantes (Aprovado/Reprovado) com base em fatores socioeconômicos e comportamentais. A capacidade de prever o desempenho escolar permite que instituições de ensino atuem preventivamente, oferecendo suporte a alunos em risco de reprovação.

2. Descrição do Dataset

Utilizou-se o conjunto de dados público "**Student Performance Factors Dataset**", obtido na plataforma Kaggle:

<https://www.kaggle.com/datasets/mosapabdelghany/student-performance-factors-dataset>

- **Fonte:** Kaggle.
- **Volume de Dados:** O dataset original contém informações sobre notas, frequência, horas de estudo, envolvimento parental, entre outros.
- **Variável-Alvo (Target):** Foi criada uma variável binária `Pass_Fail` baseada na nota final (`Exam_Score`). Considerou-se **1 (Aprovado)** para notas ≥ 65 e **0 (Reprovado)** para notas inferiores.
- **Pré-processamento:**
 - **Limpeza:** Remoção de colunas identificadoras (`Student_ID`) para evitar viés.
 - **Tratamento Categórico:** Aplicação de *One-Hot Encoding* para transformar variáveis de texto (ex: Gênero, Tipo de Escola) em variáveis numéricas.

- **Normalização:** Aplicação de `StandardScaler` em todas as variáveis de entrada para garantir que os dados tivessem média 0 e desvio padrão 1, facilitando a convergência da rede neural.
- **Divisão:** Os dados foram separados em 80% para treinamento e 20% para teste.

3. Metodologia

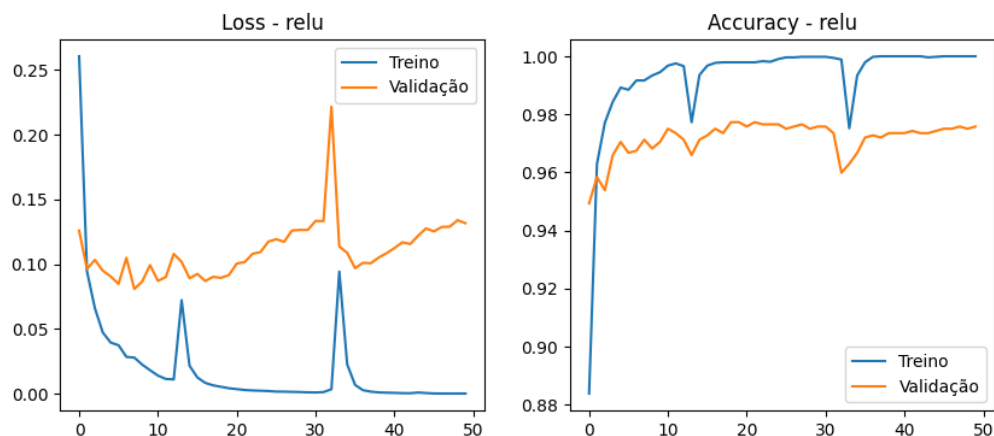
A implementação foi realizada na linguagem **Python**, utilizando a biblioteca **TensorFlow/Keras**. A arquitetura escolhida foi uma **Multilayer Perceptron (MLP)** com a seguinte estrutura:

- **Camada de Entrada:** 27 neurônios (correspondente às 27 *features* após o pré-processamento).
- **Camadas Ocultas:** 4 camadas densas com, respectivamente, **128, 64, 32 e 16 neurônios**. Essa estrutura de "funil" permite que a rede extraia características complexas nas primeiras camadas e as refine nas camadas subsequentes.
- **Camada de Saída:** 1 neurônio com função de ativação **Sigmoid** (ideal para saída binária de 0 a 1).
- **Configuração de Treinamento:**
 - **Otimizador:** Adam (Taxa de aprendizado: 0.001).
 - **Função de Perda:** Binary Crossentropy.
 - **Épocas:** 50.
 - **Batch Size:** 32.

Foram realizados dois experimentos comparativos alterando apenas a função de ativação das camadas ocultas: (1) **ReLU** e (2) **Tanh**.

4. Resultados Obtidos Abaixo, apresentamos os gráficos de desempenho (Acurácia e Perda) para os dois cenários testados.

Cenário A: Função de Ativação ReLU

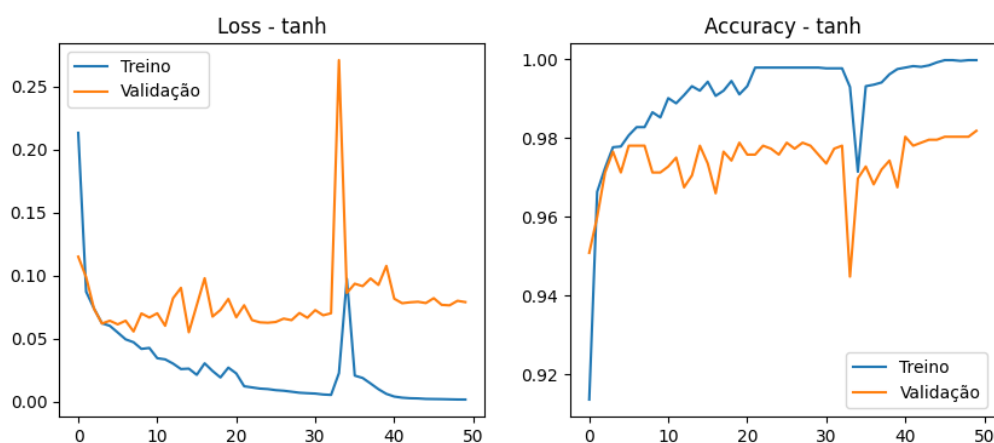


Acurácia Final no Teste:

```

166/166 ————— 1s 3ms/step - accuracy: 1.0000 - loss: 4.6082e-04 - val_accuracy: 0.9743 - val_loss: 0.1170
Epoch 43/50
166/166 ————— 1s 3ms/step - accuracy: 1.0000 - loss: 4.1514e-04 - val_accuracy: 0.9735 - val_loss: 0.1157
Epoch 44/50
166/166 ————— 1s 5ms/step - accuracy: 0.9996 - loss: 9.6726e-04 - val_accuracy: 0.9735 - val_loss: 0.1221
Epoch 45/50
166/166 ————— 1s 4ms/step - accuracy: 0.9998 - loss: 5.3576e-04 - val_accuracy: 0.9743 - val_loss: 0.1277
Epoch 46/50
166/166 ————— 1s 4ms/step - accuracy: 1.0000 - loss: 2.2725e-04 - val_accuracy: 0.9750 - val_loss: 0.1254
Epoch 47/50
166/166 ————— 1s 4ms/step - accuracy: 1.0000 - loss: 1.7612e-04 - val_accuracy: 0.9750 - val_loss: 0.1289
Epoch 48/50
166/166 ————— 1s 4ms/step - accuracy: 1.0000 - loss: 1.6840e-04 - val_accuracy: 0.9758 - val_loss: 0.1291
Epoch 49/50
166/166 ————— 1s 4ms/step - accuracy: 1.0000 - loss: 1.7016e-04 - val_accuracy: 0.9750 - val_loss: 0.1341
Epoch 50/50
166/166 ————— 1s 4ms/step - accuracy: 1.0000 - loss: 1.7544e-04 - val_accuracy: 0.9758 - val_loss: 0.1318
Acurácia final (relu): 97.58%
  
```

Cenário B: Função de Ativação Tanh



Acurácia Final no Teste: 98.18%

```
Epoch 48/50  
166/166 ██████████ 1s 30ms  
Epoch 49/50  
166/166 ██████████ 1s 30ms  
Epoch 50/50  
166/166 ██████████ 1s 30ms  
Acurácia final (tanh): 98.18%
```

5. Discussão

Os resultados demonstraram uma eficácia excepcional da rede neural na tarefa proposta. O modelo utilizando a função de ativação **Tanh** obteve um desempenho superior, alcançando **98.18% de acurácia** nos dados de teste. Os gráficos de perda (*Loss*) mostram que o modelo convergiu rapidamente, indicando que as variáveis selecionadas (como horas de estudo e frequência) possuem fortíssima correlação com o resultado final do aluno. O fato de a acurácia de validação acompanhar a de treino sugere que o modelo generalizou bem e não sofreu de *overfitting* severo.

6. Conclusão

O projeto cumpriu com êxito o objetivo de aplicar redes neurais em dados reais. A arquitetura MLP se mostrou robusta para prever o desempenho estudantil com uma alta eficácia e acurácia.