



**UNIVERSIDADE FEDERAL DE SÃO PAULO
INSTITUTO DE CIÊNCIA E TECNOLOGIA**

Projeto 1

Classificação de Resultados de Partidas de Futebol com Redes
Neurais MLP

Gabriel Belchior Vieira Campanile
RA: 168853

São José dos Campos - SP

11 de maio de 2025

Conteúdo

1	Introdução	3
2	Metodologia	3
2.1	Pré-processamento	3
2.2	Estrutura do Modelo	3
2.3	Arquiteturas Testadas	3
3	Resultados	4
3.1	Acurácia Final sem Momentum	4
3.2	Acurácia com Momentum	4
3.3	Matrizes de Confusão	4
3.4	Evolução da Perda e Acurácia	5
4	Discussão	5
5	Conclusão	6

Resumo

Este relatório apresenta um experimento de classificação de resultados de partidas de futebol utilizando redes neurais MLP (Multilayer Perceptron) em PyTorch. Foram testadas cinco arquiteturas distintas, com e sem o uso de *momentum* no otimizador SGD. Os modelos foram avaliados com base em acurácia, perda e matriz de confusão. O objetivo é identificar a melhor configuração de rede para o problema e analisar os benefícios do uso de *momentum*.

1 Introdução

O problema abordado consiste na classificação do resultado final de partidas de futebol em três categorias: vitória do time da casa (classe 0), empate (classe 1) e vitória do time visitante (classe 2). A base de dados utilizada contém atributos estatísticos como *Elo Rating*, forma recente dos times e probabilidades fornecidas por casas de apostas.

2 Metodologia

2.1 Pré-processamento

Os dados foram tratados e normalizados utilizando *StandardScaler*, e os rótulos de classe foram codificados com *LabelEncoder*. Também foram removidas linhas com valores nulos e escolhidas variáveis com relevância potencial no resultado da partida.

2.2 Estrutura do Modelo

A rede neural implementada segue a estrutura MLP (Multilayer Perceptron), com diferentes combinações de camadas ocultas. Todas as arquiteturas utilizaram a função de ativação *ReLU* e a função de custo *CrossEntropyLoss*, ponderada de acordo com o desbalanceamento das classes.

2.3 Arquiteturas Testadas

Foram avaliadas 5 arquiteturas com diferentes profundidades:

1. Sem camadas ocultas
2. 1 camada oculta com 10 neurônios
3. 2 camadas ocultas com 20 e 10 neurônios
4. 3 camadas ocultas com 40, 20 e 10 neurônios
5. 4 camadas ocultas com 80, 40, 20 e 10 neurônios

Todos os modelos foram treinados com o otimizador SGD, por 500 épocas e taxa de aprendizado de 0.1. A melhor arquitetura foi reavaliada com o uso de *momentum* = 0.9.

3 Resultados

3.1 Acurácia Final sem Momentum

Arquitetura	Acurácia (%)
0 camadas ocultas	46.17
1 camada (10)	46.27
2 camadas (20, 10)	46.50
3 camadas (40, 20, 10)	46.64
4 camadas (80, 40, 20, 10)	46.72

Tabela 1: Acurácias das arquiteturas testadas sem momentum.

3.2 Acurácia com Momentum

A arquitetura de melhor desempenho (número 5) foi reavaliada com *momentum*, alcançando uma acurácia de 47.66%. Observou-se uma pequena melhora em relação ao modelo sem *momentum*.

3.3 Matrizes de Confusão

As matrizes de confusão dos modelos com e sem momentum são mostradas a seguir:

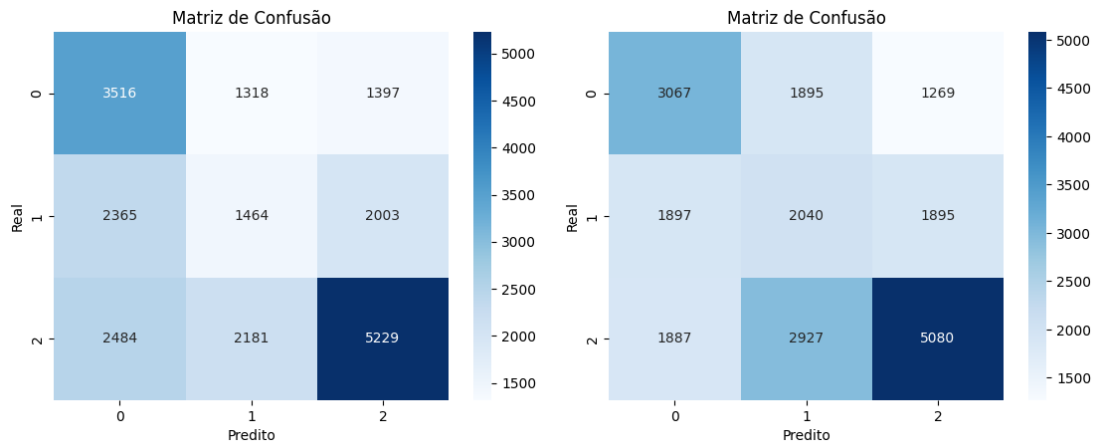


Figura 1: Matrizes de confusão dos modelos sem (esquerda) e com momentum (direita).

3.4 Evolução da Perda e Acurácia

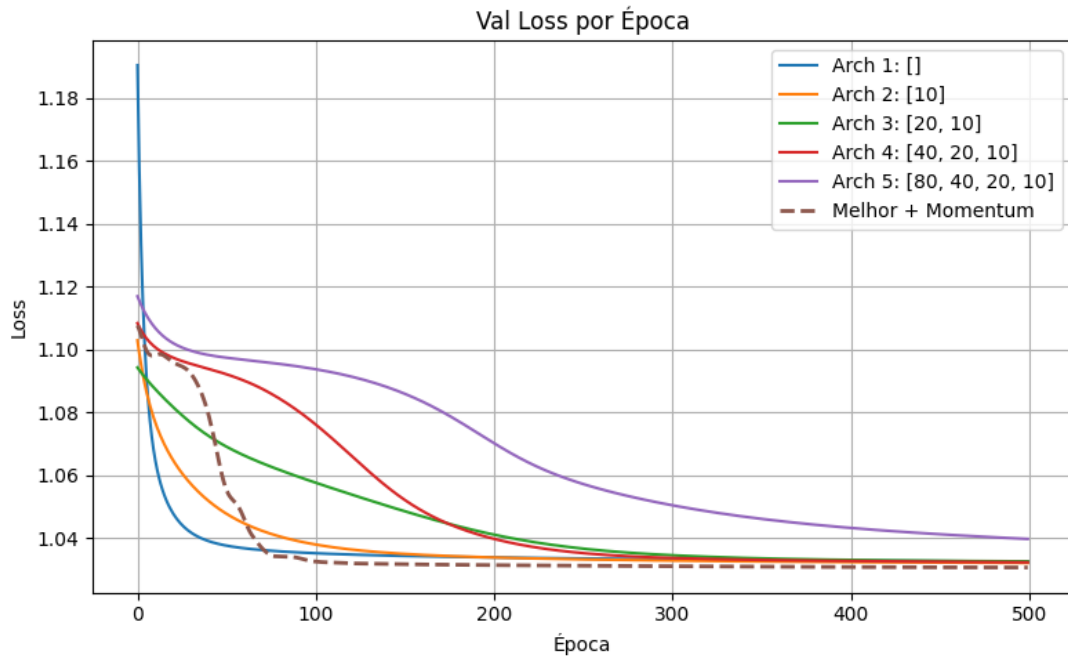


Figura 2: Curvas de perda (validação) por época para todas as arquiteturas.

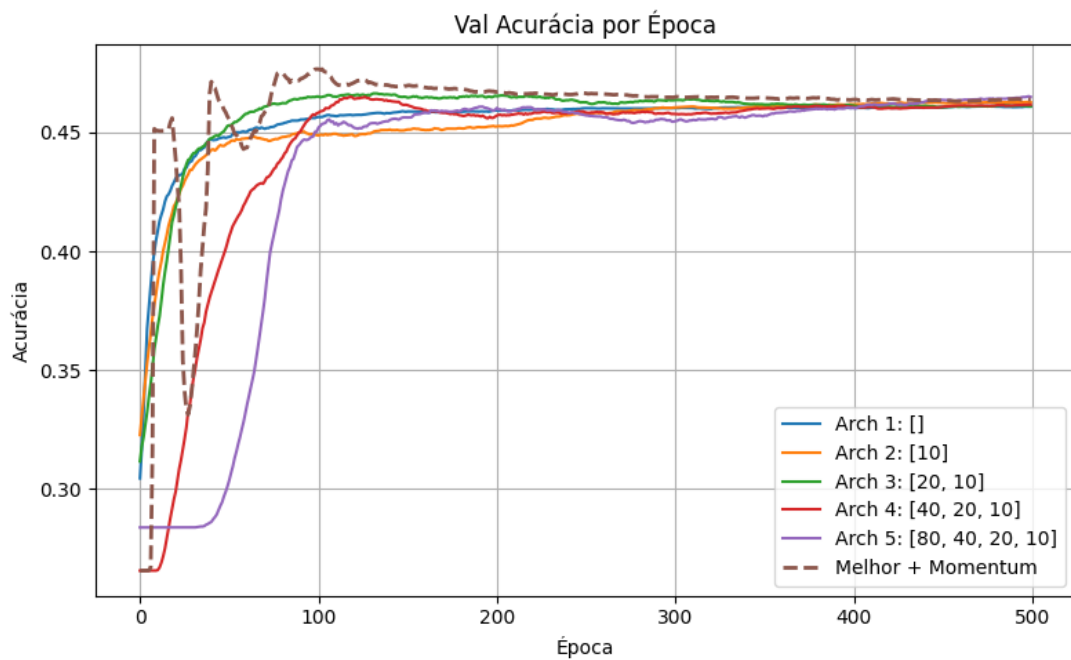


Figura 3: Curvas de acurácia (validação) por época para todas as arquiteturas.

4 Discussão

Durante os testes, observou-se que os modelos com mais camadas ocultas apresentaram desempenho ligeiramente superior, porém a melhora não foi significativa nem linear. Uma

difficuldade importante enfrentada foi o fato de que, mesmo após o tratamento do desbalanceamento das classes — atribuindo pesos inversamente proporcionais à frequência de cada classe na função de perda — o modelo não conseguia alcançar alta acurácia.

Esse tratamento corrigiu o comportamento inicial do modelo de convergir apenas para uma única classe majoritária. No entanto, como agora o modelo tentava distribuir suas previsões entre as três classes, o problema se tornou substancialmente mais difícil e a acurácia global caiu de 50% para 46% em média.

Diversas estratégias foram testadas para melhorar esse desempenho: ajuste da taxa de aprendizado, aumento do número de épocas de treinamento, variação na profundidade da rede e no número de neurônios, e uso de momentum. Apesar dessas tentativas, não foi possível superar consistentemente a barreira dos 50% de acerto.

Isso evidencia que o problema de predição de resultados de partidas de futebol, baseado apenas em dados tabulares estatísticos pré-jogo, pode não ser adequadamente resolvido por uma MLP simples. O modelo tem dificuldades em capturar as complexidades e variabilidades do futebol, como fatores contextuais, táticos e subjetivos não representados nos dados.

5 Conclusão

O experimento demonstrou que, embora redes neurais MLP possam ser aplicadas ao problema de classificação de partidas de futebol, seu desempenho é limitado quando baseadas apenas em atributos tabulares pré-jogo. Mesmo com ajustes no modelo, balanceamento das classes e técnicas como momentum, a acurácia manteve-se relativamente baixa, abaixo de 51%.

Conclui-se que o uso de MLP para este tipo de predição, com os dados disponíveis, não é o método mais eficaz. A complexidade e imprevisibilidade do futebol requerem modelos mais robustos ou dados mais ricos e contextuais para que uma IA consiga generalizar corretamente os padrões de vitória, empate ou derrota.

Próximos passos

Como trabalhos futuros, sugere-se:

- Explorar regularização como **Dropout** e penalizações L2;
- Testar outros otimizadores como **Adam** e **RMSprop**;
- Ampliar o conjunto de dados com informações contextuais (lesões, escalações, mando de campo, histórico entre os times);
- Realizar **validação cruzada** para avaliar a generalização do modelo;
- Explorar outras arquiteturas de rede como **redes recorrentes (RNN)** ou modelos baseados em atenção.