

**UNIVERSIDADE FEDERAL DE PERNAMBUCO**

AMANDA ARRUDA MELO SILVA

LUCAS VINÍCIUS DA SILVA

LUÍS FELIPE OLIVEIRA COSTA

MICHEL LEONIDAS ALEIXO DA SILVA

SEVERINO CARLOS DA SILVA JUNIOR

**RELATÓRIO DO MINI-PROJETO 1: ANÁLISE DE DESEMPENHO DA  
APLICAÇÃO DE MLPs AO CIFAR-10**

**RECIFE**

**2025**

## INTRODUÇÃO

Este relatório tem como objetivo apresentar uma exploração da aplicação de redes neurais Perceptron Multicamadas (MLPs) na classificação de imagens do conjunto de dados CIFAR-10, que é composto por 60.000 imagens coloridas de 32x32 pixels, divididas em 10 classes distintas. Apesar de a arquitetura MLP não ser a mais indicada para tarefas de classificação de imagens devido à sua incapacidade de explorar informações espaciais, o presente mini-projeto visa investigar como diferentes variações arquiteturais e de hiperparâmetros afetam seu desempenho neste domínio.

Foram conduzidos oito experimentos distintos, cada um empregando uma configuração específica de arquitetura, função de ativação, otimizador, taxa de aprendizado, regularização (dropout e weight decay), uso ou não de data augmentation, e diferentes tamanhos de lote e número de épocas de treinamento. A escolha dessas configurações teve como finalidade explorar a capacidade de generalização das MLPs sob diferentes condições, buscando compreender quais fatores contribuem para um melhor desempenho no conjunto de teste.

A análise dos resultados considera as métricas de acurácia, precisão, recall e F1-score para avaliação do desempenho dos modelos. Além disso, são discutidos aspectos relativos à eficiência de treinamento, sinais de overfitting e impacto de cada estratégia de regularização.

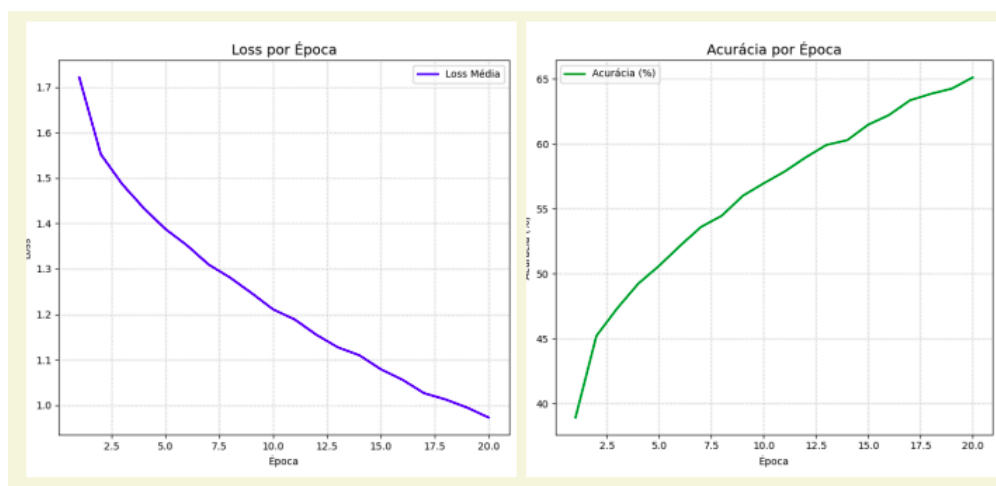
## EXPERIMENTO 1

Na primeira abordagem, foi utilizada uma arquitetura composta por três camadas lineares: a camada de entrada com 3072 neurônios, seguida por uma camada intermediária com 1024 neurônios, outra com 512 neurônios e, finalmente, uma camada de saída com 10 neurônios, correspondentes às classes do dataset. As funções de ativação utilizadas foram ReLU entre as camadas intermediárias. Para regularização, foi aplicado dropout com taxa de 30% nas duas primeiras camadas ocultas.

Os hiperparâmetros utilizados foram: batch size de 128, 20 épocas de treinamento, otimizador Adam com taxa de aprendizado de 0,001, e função de perda CrossEntropyLoss. Não foi utilizado agendamento de taxa de aprendizado, nem técnicas adicionais de regularização como L2.

O desempenho obtido nesta configuração foi: acurácia de teste de aproximadamente 55%, F1-score médio de 0,55, precisão média de 0,56 e recall médio de 0,55. O modelo apresentou aprendizado progressivo e sem sinais acentuados de overfitting. Contudo, a diferença entre treino e teste apontou para possíveis melhorias por meio de ajustes mais robustos de regularização e complexidade.

A análise dos gráficos de treino e validação indica um comportamento estável da perda durante as 20 épocas, com uma evolução gradual e paralela entre as curvas. Isso sugere que, apesar da simplicidade da arquitetura, a regularização via dropout foi suficiente para evitar overfitting severo, embora os resultados ainda se mantenham limitados por uma capacidade representacional reduzida.



Desempenho do treinamento do experimento 1.

## EXPERIMENTO 2

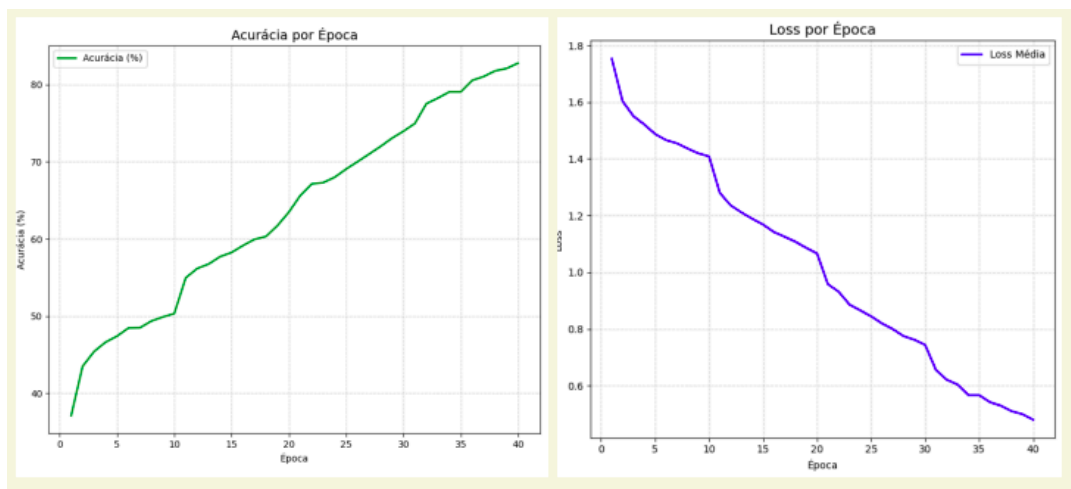
A segunda abordagem buscou expandir a capacidade representacional da MLP com uma arquitetura mais profunda: a rede foi composta por quatro camadas lineares, com dimensões  $3072 \rightarrow 2048 \rightarrow 1024 \rightarrow 512 \rightarrow 10$ . Todas as camadas ocultas utilizaram ReLU como função de ativação, e foram acompanhadas por dropout com taxa de 30%. Além disso, foi incorporada a regularização L2 com weight decay de  $1e-4$ , e um scheduler StepLR com fator de decaimento de 0,5 a cada 10 épocas.

Foram mantidos o batch size de 128, taxa de aprendizado de 0,001, função de perda CrossEntropyLoss e otimizador Adam com weight decay. O treinamento foi realizado por 40 épocas, com a expectativa de ganhos em capacidade de modelagem e estabilidade.

Em termos de desempenho, o modelo apresentou uma acurácia de teste de 58%, F1-score médio de 0,58, precisão média de 0,59 e recall de 0,58. Embora o desempenho em teste tenha sido o melhor entre todas as abordagens, a análise das curvas de treinamento revelou um considerável distanciamento entre treino e validação, indicando a presença de overfitting mesmo com as estratégias de regularização aplicadas.

Os gráficos de desempenho ao longo das 40 épocas de treinamento indicam um aprendizado globalmente estável e eficiente. A acurácia de treino aumentou continuamente, partindo de 37,1% na primeira época até alcançar 82,78% na

quadragésima, enquanto a perda média caiu de 1.7532 para 0.4804. Observam-se, entretanto, platôs temporários nas curvas de loss (entre as épocas 10–15 e 20–25) e de acurácia (entre as épocas 25–30), que coincidem com os momentos em que o agendador StepLR reduziu a taxa de aprendizado. Esses platôs são compatíveis com o comportamento esperado: a cada redução, o modelo passa por uma breve fase de adaptação antes de retomar a melhora. Ainda assim, o gap de 24 pontos percentuais entre acurácia de treino e teste indica alto overfitting, apesar das estratégias de regularização.



Desempenho do treinamento do experimento 2.

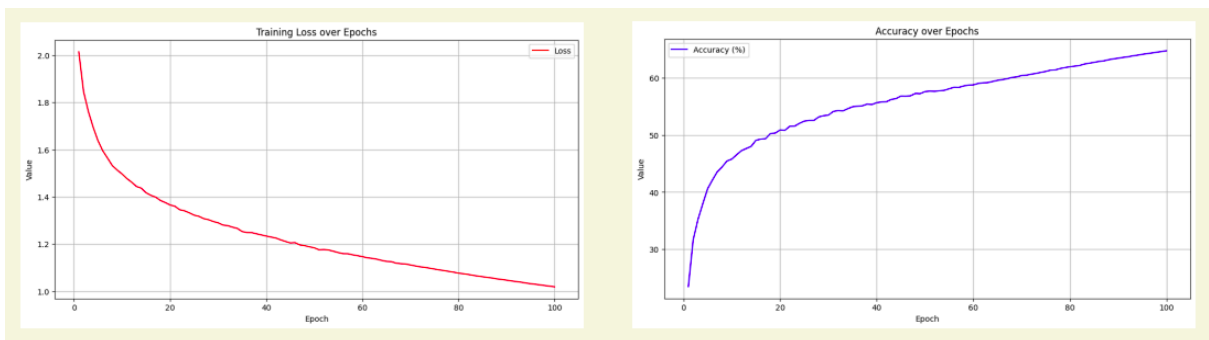
### EXPERIMENTO 3

A terceira abordagem utilizou uma arquitetura mais profunda e granular: 3072 neurônios na entrada, seguidos por quatro camadas ocultas com 1024, 512, 256 e 128 neurônios respectivamente, culminando em uma camada de saída com 10 neurônios. A função de ativação escolhida foi LeakyReLU em todas as camadas intermediárias, buscando uma maior estabilidade de gradientes e capacidade de representação.

Os hiperparâmetros definidos incluíram: batch size de 64, taxa de aprendizado de 0,0005, otimizador RMSprop, regularização L2 com weight decay de  $1e-5$ , e treino por 100 épocas com early stopping utilizando paciência de 10 épocas. Também foi aplicada data augmentation para enriquecer a variabilidade dos dados de entrada.

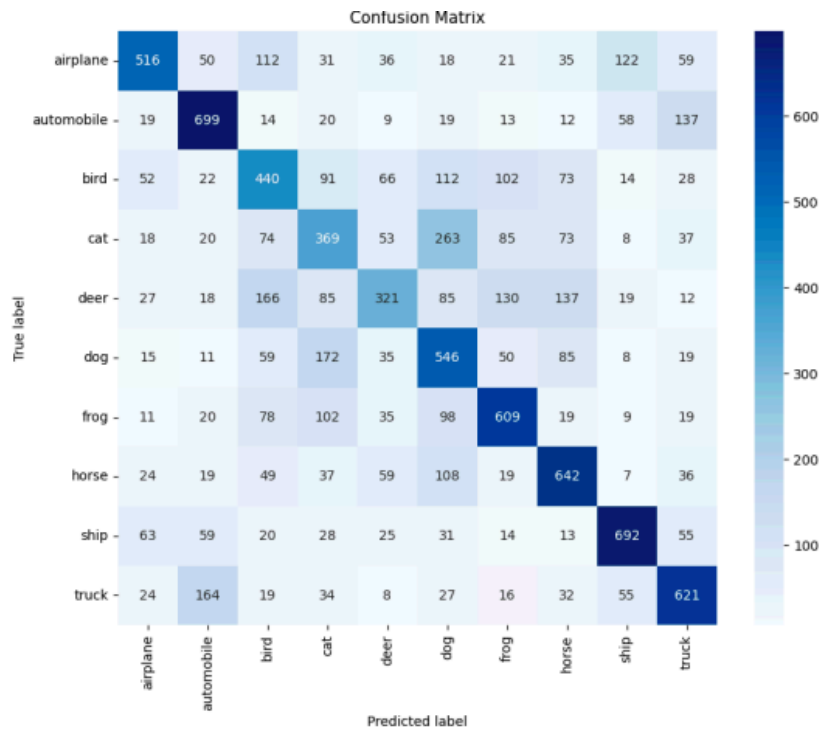
O modelo atingiu uma acurácia de teste de 54%, F1-score de 0,54, precisão de 0,54 e recall de 0,54. A uniformidade entre as métricas sugere um comportamento equilibrado entre as classes, embora sem ganhos significativos sobre abordagens mais simples. Os gráficos indicam uma curva de validação estável, com leve tendência a platô após 70 épocas, indicando um aprendizado eficiente, mas limitado por capacidade de abstração.

Os gráficos do Experimento 3 indicam um aprendizado progressivo e estável ao longo das 100 épocas. A acurácia de treino aumentou de forma constante, partindo de aproximadamente 25% e alcançando cerca de 66%, enquanto a função de perda caiu gradualmente de 2.0 para aproximadamente 1.0, sem sinais de overfitting. Esse comportamento sugere uma aprendizagem eficiente, mas com ganho incremental decrescente após as primeiras 30 épocas.



### Desempenho do treinamento do experimento 3.

A matriz de confusão revela bom desempenho em categorias visuais mais distintas (como automóveis, navios, sapos e cavalos), mas dificuldade expressiva em diferenciar classes com padrões visuais similares, como gatos, cães e veados. O modelo se mostra confiável em contextos com maior contraste entre classes, mas ainda limitado na discriminação de detalhes visuais sutis.



Matriz de confusão do experimento 3.

## EXPERIMENTO 4

Neste experimento, foi empregada uma abordagem baseada em otimização automática de hiperparâmetros utilizando a biblioteca Optuna. O objetivo foi encontrar configurações eficientes para o modelo MLP por meio de uma busca sistemática com cinco trials, limitada por restrições computacionais do Google Colab. Os hiperparâmetros ajustados incluíram: tamanho do batch, número de neurônios em três camadas ocultas, função de ativação, taxa de dropout e tipo de otimizador.

A melhor combinação encontrada apresentou taxa de aprendizado de aproximadamente 0.00034, batch size 16, arquitetura:  $3072 \rightarrow 170 \rightarrow 155 \rightarrow 155 \rightarrow 10$ , função de ativação LeakyReLU, otimizador AdamW com weight decay =  $1e-4$  e dropout de aproximadamente 15,7%.

O modelo foi treinado por 50 épocas. Os gráficos indicam que tanto o loss de treino quanto o de validação diminuíram de forma coordenada até aproximadamente a 30ª época, a partir da qual o loss de validação começa a oscilar em torno de 1.3, enquanto o de treino continua caindo. Isso indica um início de sobreajuste (overfitting leve) após esse ponto.

Já na acurácia, observa-se crescimento simultâneo entre treino e validação até a 30ª época, onde ambas atingem patamares próximos a 55–56%. Após esse ponto, a acurácia de validação estabiliza e oscila levemente, enquanto a acurácia de treino continua subindo até o final. Esse padrão reforça a observação de overfitting progressivo e controlado, típico de cenários com batch pequeno e redes profundas.

Em termos de desempenho final, o modelo alcançou acurácia de 56.07%, precisão de 55.98%, recall de 56.07% e F1-score de 55.75%. Tais métricas confirmam um desempenho equilibrado e robusto, com bom comprometimento entre sensibilidade e especificidade.

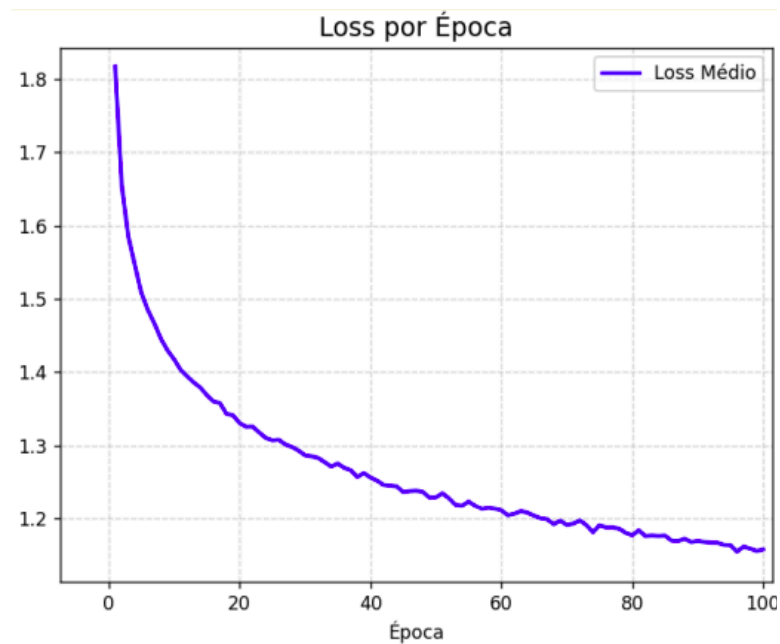
Apesar do número limitado de trials do Optuna, a combinação resultante revelou-se eficaz, especialmente pelo uso de ativação LeakyReLU e regularização leve. Esta estratégia demonstra o potencial de buscas automáticas de hiperparâmetros mesmo sob restrição computacional, e destaca a capacidade do Optuna em encontrar boas soluções com poucas amostragens.

## **EXPERIMENTO 5**

A quinta experiência retomou a arquitetura com três camadas intermediárias, agora com 128, 256 e 128 neurônios, utilizando LeakyReLU como ativação. Não houve uso de dropout. O otimizador adotado foi Adam com weight decay de  $1e-5$ , taxa de aprendizado de 0,001, batch size de 64, 100 épocas de treinamento e uso de early stopping com paciência de 5 épocas. Também foi aplicada data augmentation.

O modelo alcançou acurácia de teste de 54%, F1-score de 0,54, precisão de 0,58 e recall de 0,54. Observa-se uma maior precisão em relação à capacidade de identificar corretamente os exemplos positivos, com um leve prejuízo no recall. O gráfico de perda revela uma trajetória estável e com aprendizado. A perda média inicial, em torno de 1.82, cai progressivamente até aproximadamente 1.15 ao final das 100 épocas. O modelo apresenta uma fase de aprendizado acelerado nas primeiras 20 épocas, seguida de uma redução mais gradual e controlada. Pequenas oscilações são visíveis a partir da metade do treinamento, mas permanecem dentro de um intervalo limitado, o que indica que o modelo continuou aprendendo sem apresentar sobreajuste evidente. A curva de loss corrobora a eficácia da configuração adotada, mesmo na ausência de dropout explícito.



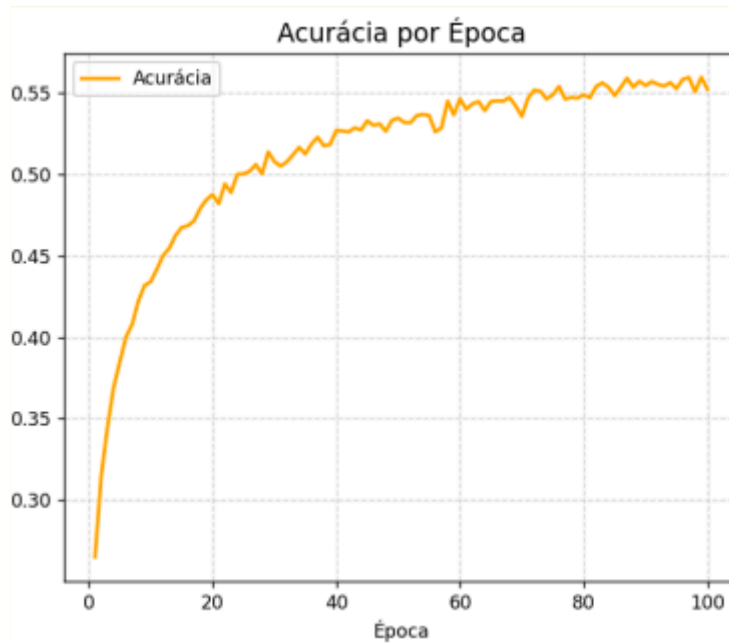


Perda conforme as épocas do experimento 5.

## EXPERIMENTO 6

Na sexta abordagem, foi mantida a mesma arquitetura da quinta experiência (3072  $\rightarrow$  128  $\rightarrow$  256  $\rightarrow$  128  $\rightarrow$  10), mas com alterações no otimizador e na regularização. O otimizador utilizado foi o SGD com momentum de 0,9, mantendo a taxa de aprendizado em 0,001. Não foi aplicado dropout e tampouco weight decay. O modelo foi treinado por 100 épocas, com early stopping de 5 épocas e uso de data augmentation.

O desempenho do modelo foi inferior em relação à configuração com Adam: acurácia de teste de 51%, F1-score de 0,50, precisão de 0,51 e recall de 0,51. O uso de SGD, apesar de mais simples e potencialmente mais robusto em longo prazo, não foi suficiente para superar as limitações de capacidade da arquitetura ou compensar a ausência de dropout. O gráfico de acurácia por época revela oscilações mais acentuadas na validação e indícios de estagnação após 60 épocas.



## EXPERIMENTO 7

Este experimento adotou uma abordagem mais simples: uma arquitetura com apenas duas camadas ocultas de 128 e 64 neurônios, seguidas pela saída de 10 neurônios. Foi utilizada a ativação ReLU, com dropout de 30% em ambas as camadas ocultas. O otimizador foi Adam, com learning rate de 0,001. O modelo foi treinado por 20 épocas, sem data augmentation nem scheduler. O desempenho ficou abaixo das abordagens anteriores, com acurácia de teste de 52%, F1-score de 0,52, precisão de 0,52 e recall de 0,52.

## EXPERIMENTO 8

Na oitava e última abordagem, buscou-se explorar uma função de ativação alternativa: Tanh. A arquitetura contou com três camadas intermediárias com 256, 128 e 64 neurônios, seguidas da camada de saída. Foi utilizado dropout de 20% em todas as camadas ocultas. O otimizador foi SGD com momentum, learning rate de 0,0005, batch size de 128, e o modelo foi treinado por 20 épocas com data augmentation.

Esta abordagem apresentou o pior desempenho entre todas as testadas: acurácia de teste de apenas 36%, F1-score de 0,34, precisão de 0,36 e recall de 0,36. A função Tanh, embora teoricamente capaz de modelar não-linearidades, não se mostrou eficaz neste contexto, especialmente com learning rate muito baixo.

## CONCLUSÃO

Os resultados indicam que, embora as MLPs possam atingir desempenho razoável na tarefa de classificação do CIFAR-10, sua capacidade de generalização é limitada em comparação com arquiteturas convolucionais mais apropriadas ao processamento de imagens. A melhor acurácia de teste foi alcançada pela segunda abordagem (58%), que combinou uma arquitetura profunda com regularização via dropout e L2, embora com sinais de overfitting evidentes.

As abordagens que incorporaram data augmentation (experimentos 3, 4, 5 e 6) mostraram benefícios moderados, especialmente quando aliadas a otimizadores como Adam ou AdamW. Por outro lado, a oitava abordagem, que utilizou Tanh e SGD com learning rate reduzido, apresentou desempenho significativamente inferior, destacando a importância da escolha adequada da função de ativação e taxa de aprendizado.