



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

YVES EMMANUEL

MULTILAYER PERCEPTRON: RECONHECIMENTO DE DÍGITOS
MANUSCRITOS

Recife-PE
2023

YVES EMMANUEL

MULTILAYER PERCEPTRON: RECONHECIMENTO DE DÍGITOS
MANUSCRITOS

Projeto de graduação apresentado como requisito à
obtenção de aprovação na disciplina de Redes Neurais
do Curso de Graduação em Ciência da Computação da
Universidade Federal de Pernambuco.

Prof. Dr. Germano Vasconcelos.

Recife-PE
2023

Sumário

	Páginas
1 Introdução	4
2 Revisão Literária	4
3 Metodologia	5
3.1 Norma do Gradiente x Épocas	6
3.2 Perda x Épocas	7
3.3 Acurácia x Épocas	8
4 Resultados	9
4.1 Norma do Gradiente x Épocas	9
4.2 Perda x Épocas	9
4.3 Acurácia x Épocas	10
4.4 Outros gráficos	10

1 Introdução

O reconhecimento de caracteres manuscritos é um problema muito discutido em visão computacional e reconhecimento de padrões (LECUN et al., 1998). O reconhecimento preciso de dígitos manuscritos é crucial em muitas aplicações, como classificação de códigos postais, processamento de cheques bancários e verificação de assinaturas. Nos últimos anos, métodos de aprendizado profundo, como o perceptrons de multicamadas (Multi-Layer Perceptron - MLP), têm mostrado resultados promissores na resolução desse problema.

Neste projeto, nosso objetivo é desenvolver um modelo MLP para prever dígitos manuscritos. Utilizaremos o conjunto de dados popular MNIST fornecido na disciplina, que contém 50.000 imagens de treinamento, 10.000 imagens de teste e 10.000 imagens de validação de dígitos manuscritos de 0 a 9. Nosso objetivo é treinar um modelo MLP que possa classificar essas imagens com precisão em seus respectivos rótulos de dígitos.

Esse problema apresenta vários desafios. Por exemplo, a variabilidade no estilo de escrita, onde diferentes indivíduos podem escrever o mesmo dígito de maneiras ligeiramente diferentes, torna desafiador reconhecer o dígito consistentemente em diferentes amostras (LECUN et al., 1998). Além disso, a sobreposição de dígitos, onde dois ou mais dígitos podem se sobrepor na mesma imagem, torna difícil separá-los e reconhecer cada dígito individualmente (Due Trier; JAIN; TAXT, 1996). Esses desafios exigem um modelo robusto o suficiente para que possa lidar com padrões complexos e diversos de forma eficiente. O MLP é uma escolha popular para esse problema, pois pode aprender relacionamentos não-lineares complexos entre as características de entrada e os rótulos de saída.

Nosso projeto envolverá pré-processamento de dados, seleção de modelos, ajuste de hiperparâmetros e avaliação de desempenho. Com isso, esperamos obter um modelo MLP otimizado para a tarefa de reconhecimento de dígitos manuscritos.

Os resultados deste projeto poderão se estendidos para outros problemas relacionados, como reconhecimento de caracteres e reconhecimento de escrita à mão, e pode potencialmente ser aplicado em várias indústrias, incluindo bancária, logística e segurança.

2 Revisão Literária

O reconhecimento de dígitos é um problema bem estabelecido no campo da visão computacional e tem sido amplamente estudado ao longo dos anos. Muitas abordagens diferentes foram propostas para lidar com esse problema, incluindo *template matching*, árvores de decisão, máquinas de vetor de suporte (SVMs), redes neurais convolucionais (CNNs) e perceptrons multicamadas (MLPs).

Abordagens iniciais para o reconhecimento de dígitos, como *template matching* e árvores de decisão, eram relativamente simples e computacionalmente eficientes, mas muitas vezes sofriam de precisão limitada e generalização para novos dados. SVMs, que usam uma função

núcleo para mapear dados de entrada em um espaço de maior dimensão, mostraram-se eficazes em alguns casos, mas exigiram ajuste cuidadoso de seus parâmetros e sofrem de um alto custo computacional. SVMs exigem que todas as observações de treinamento sejam mantidas em memória, o que limita o tamanho do conjunto de dados que pode ser processado. Além disso, o processo de otimização dos SVMs requer a solucionar um problema com complexidade quadrática, que é computacionalmente caro em grandes conjuntos de dados (CORTES; VAPNIK, 1995). O problema com complexidade quadrática surge na busca pelos vetores de suporte e pelo hiperplano ótimo que maximiza a margem entre as classes. A solução é encontrada através de um algoritmo de otimização que itera várias vezes até convergir para a solução ótima, o que pode levar muito tempo em grandes conjuntos de dados.

Nos últimos anos, abordagens de aprendizado profundo, como CNNs e MLPs, mostraram um sucesso notável no reconhecimento de dígitos e outras tarefas de visão computacional. CNNs, em particular, tornaram-se o método preferido para o reconhecimento de imagem devido à sua capacidade de aprender representações hierárquicas de características visuais diretamente dos dados brutos de pixels (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

No entanto, MLPs ainda são uma escolha popular para o reconhecimento de dígitos devido à sua simplicidade, escalabilidade e capacidade de lidar com mapeamentos não lineares de entrada-saída. Vários estudos mostraram que abordagens baseadas em MLP podem alcançar desempenho comparável ou até superior a modelos de aprendizado profundo mais complexos, especialmente quando o conjunto de dados é relativamente pequeno (CHEN et al., 2018).

Apesar do progresso significativo no reconhecimento de dígitos com abordagens baseadas em MLP, ainda existem vários desafios que precisam ser abordados, como lidar com dígitos ruidosos e sobrepostos e melhorar a robustez do modelo a variações nos estilos de escrita. Nas próximas seções, apresentaremos nossa abordagem ao problema de reconhecimento de dígitos usando um modelo MLP e discutiremos como avaliá-la adequadamente.

3 Metodologia

Os dados utilizados neste projeto foram coletados do conjunto de dados de dígitos manuscritos MNIST, que consiste em 70.000 imagens em escala de cinza de dígitos manuscritos, cada uma com tamanho de 28x28 pixels. O conjunto de dados foi dividido em um conjunto de treinamento de 60.000 imagens e um conjunto de validação de 10.000 imagens.

Tabela 1. *Hiperparâmetros do Grid Search*

Camadas Intermediárias	Taxa de Aprendizagem	Épocas	Tamanho do Lote
32	0.0001	100	32
64	0.001	200	64
128	0.01	300	128

Para otimizar o desempenho do modelo MLP, realizamos uma busca exaustiva pelos

melhores hiperparâmetros (tabela 1), incluindo o número de camadas intermediárias, a taxa de aprendizagem (*learning rate*), o número máximo de iterações e o tamanho do lote. Utilizamos a implementação `MLPClassifier` do `scikit-learn` com o algoritmo de aprendizagem *Stochastic Gradient Descent* (SGD), uma taxa de aprendizado constante e a função de ativação linear retificada (ReLU) para as camadas intermediárias da rede.

Para realizar essa busca, utilizou-se a validação cruzada com 5 partições (*5-fold cross-validation*), onde a métrica de acurácia foi utilizada para avaliar o desempenho de cada combinação de hiperparâmetros. A validação cruzada com *k-folds* é uma técnica que consiste em dividir o conjunto de dados em *k* partições iguais, onde o modelo é treinado *k* vezes, usando *k-1* partições para treinamento e uma partição para validação em cada iteração. Esse processo é repetido para cada partição ser utilizada como a partição de validação. Isso permite que todo o conjunto de dados seja utilizado tanto para treinamento quanto para validação, evitando a possibilidade de *overfitting* em um conjunto de validação específico.

Selecionamos o melhor conjunto de hiperparâmetros (tabela 2) com base na busca exaustiva com validação cruzada, que resultou na maior acurácia. À priori, consideramos o resultado ruim, pois o modelo ótimo da busca resultou em uma acurácia de 0,796 nos dados de validação. A partir daí, analisamos a fase de treinamento para justificar tal desempenho e procurar melhorias.

Tabela 2. Melhores hiperparâmetros encontrados

Camadas Intermediárias	Taxa de Aprendizagem	Épocas	Tamanho do Lote
64	0.01	100	128

3.1 Norma do Gradiente x Épocas

Analisar a norma do gradiente é uma ferramenta de diagnóstico importante usada no campo do aprendizado de máquina para avaliar o desempenho de um modelo durante a fase de treinamento. A norma do gradiente é uma medida da magnitude dos gradientes utilizada no processo de otimização para atualizar os parâmetros do modelo.

Durante a fase de treinamento, os gradientes são calculados usando o algoritmo de *backpropagation*, que propaga os erros de volta pela rede para calcular os gradientes em cada camada anterior. A norma do gradiente é calculada pela norma L2 dos gradientes, que é uma medida da magnitude do vetor. Se os gradientes forem muito pequenos, isso pode levar a uma convergência lenta e baixo desempenho, enquanto se forem muito grandes, isso pode causar instabilidade numérica e dificuldade em encontrar um bom conjunto de parâmetros. Esses são os problemas de *vanishing* e *exploding gradient norm*.

O aumento da norma do gradiente (figura 1) indica que os gradientes estão se tornando maiores à medida que o processo de otimização continua, o que pode levar a instabilidade

numérica e dificuldade em encontrar um bom conjunto de parâmetros do modelo. Isso pode resultar em tempos de treinamento mais lentos e desempenho geral inferior do modelo.

Para abordar esse problema, podemos ajustar a taxa de aprendizado ou pré-processar os dados para remover ruído ou melhorar sua representatividade. Seguiremos alterando a taxa de aprendizado que foi passado muito baixa para o *grid search*, mas antes pode ser útil monitorar outras métricas, como a função de perda ou precisão, para obter uma compreensão mais completa da fase de treinamento.

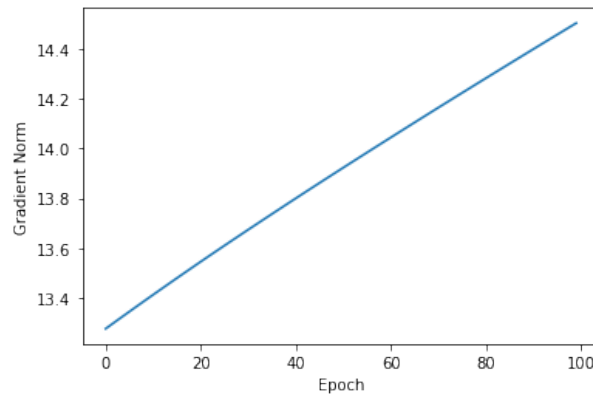


Figura 1. Plot: norma do gradiente x épocas

3.2 Perda x Épocas

Além da norma do gradiente, também monitoramos a função de perda durante o treinamento (figura 2). O gráfico da perda por épocas mostrou uma diminuição gradual na perda ao longo do treinamento, porém em um ritmo muito lento. Isso pode indicar que o modelo não está aprendendo de forma eficaz, ou que há problemas com a qualidade dos dados ou com a arquitetura do modelo.

Na fase de validação, observamos que a perda se tornou ruidosa, aumentando ligeiramente e terminando ligeiramente acima da perda na primeira época. Isso indica *overfitting*, onde o modelo aprendeu a se ajustar aos dados de treinamento com muita precisão, resultando em um desempenho inferior em dados não vistos. É possível que o modelo possa se beneficiar de técnicas adicionais de regularização ou modificações na arquitetura do modelo.

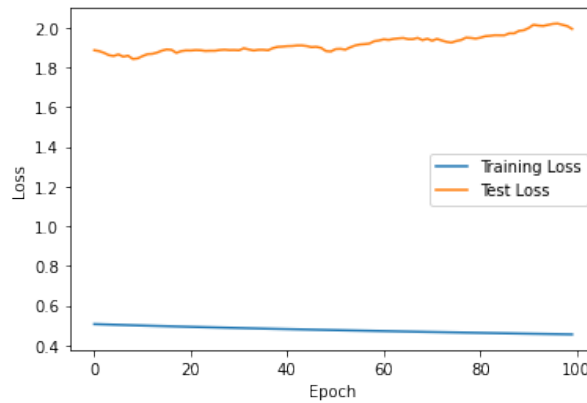


Figura 2. Plot: perda x épocas

3.3 Acurácia x Épocas

O gráfico de acurácia x épocas é uma das ferramentas de diagnóstico mais importantes que pode fornecer informações sobre o desempenho do modelo. Neste caso, o gráfico mostra que o modelo foi capaz de melhorar sua acurácia ao longo do tempo (figura 3), mas em um ritmo lento. A acurácia começou em cerca de 0,82 e ficou abaixo de 0,84 antes de começar a se estabilizar, indicando que o modelo foi capaz de aprender a partir dos dados, mas não efetivamente.

Uma solução para melhorar o desempenho do MLP treinado com os parâmetros dados é tentar aumentar a taxa de aprendizado. Isso pode ajudar a acelerar a convergência do modelo e melhorar sua precisão. Além disso, podemos tentar modificar a arquitetura do modelo adicionando mais camadas ou alterando o número de neurônios em cada camada. Técnicas de regularização, como *dropout* ou regularização L1/L2, também podem ajudar a melhorar a estabilidade do processo de otimização e prevenir *overfitting*. Pré-processamento dos dados, como escalar ou normalizar, também pode ajudar a melhorar a representatividade dos dados e facilitar a aprendizagem do modelo.

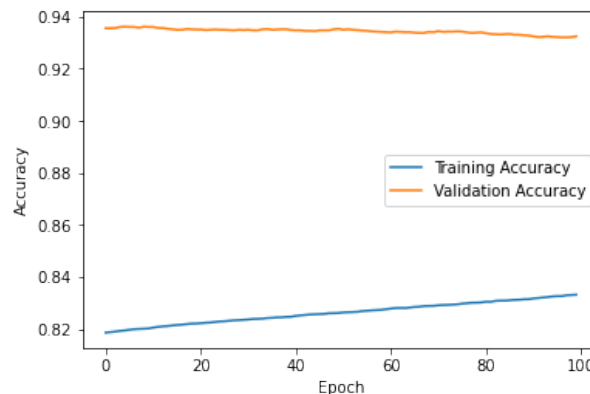


Figura 3. Plot: acurácia x épocas

4 Resultados

Utilizando o conhecimento da análise do treinamento da rede, é possível inferir que um ajuste na taxa de aprendizagem pode auxiliar numa mais rápida convergência e, consequentemente, treinamento. O modelo final atualizado teve a taxa de aprendizagem ajustada para 0,46.

4.1 Norma do Gradiente x Épocas

O gráfico da norma do gradiente (figura 4) demonstra que tivemos sucesso em controlar as normas dos vetores gradientes. Lembre-se que ao prolongar o treinamento, existirá o risco da norma diminuir o suficiente para "sumir".

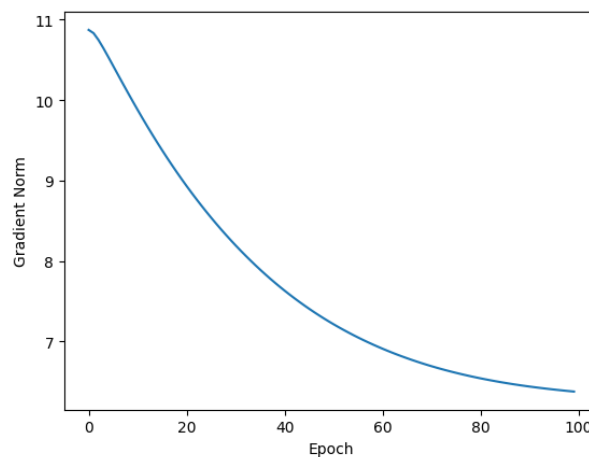


Figura 4. *Plot: norma do gradiente x épocas*

4.2 Perda x Épocas

O gráfico de perda (figura 5) evidencia que tivemos sucesso em treinar efetivamente a MLP, reduzindo sua função de perda durante a fase de treinamento.

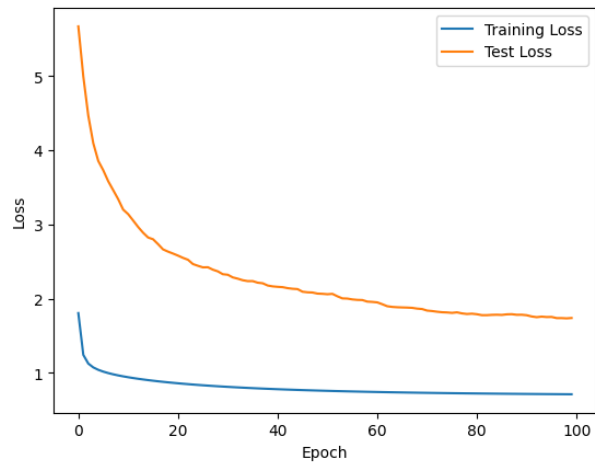


Figura 5. *Plot: perda x épocas*

4.3 Acurácia x Épocas

Além disso, obtivemos 0,93 de acurácia na validação do modelo final atualizado.

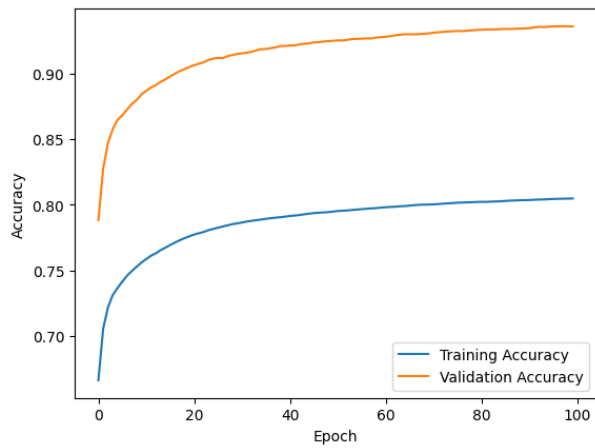


Figura 6. *Plot: acurácia x épocas*

4.4 Outros gráficos

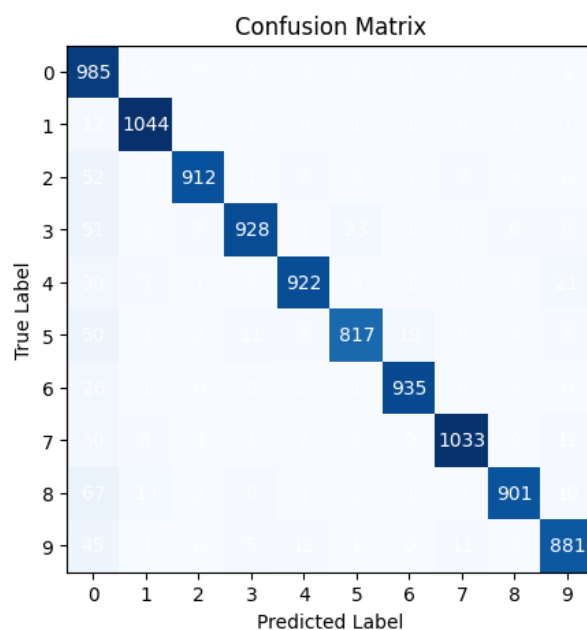


Figura 7. *Confusion Matrix*

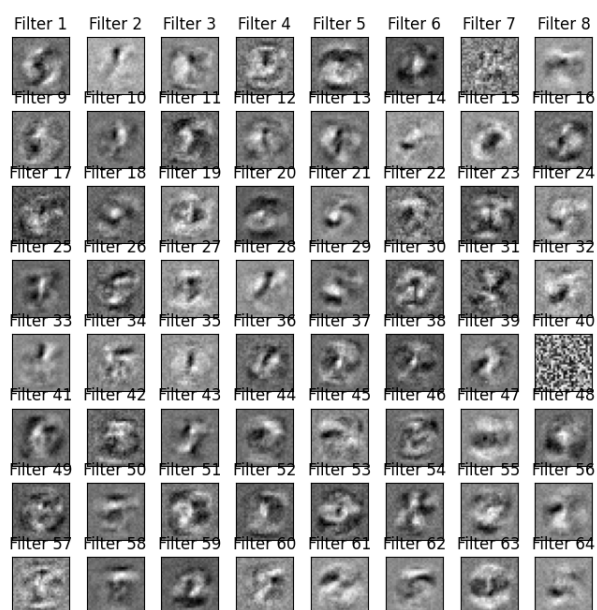


Figura 8. *Filtros calculados*

Referências

- CHEN, Y. et al. Handwritten digit recognition: Benchmarking of state-of-the-art techniques. **Computers & Electrical Engineering**, Elsevier, v. 71, p. 322–332, 2018.
- CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, Springer, v. 20, n. 3, p. 273–297, 1995.
- Due Trier Øivind; JAIN, A. K.; TAXT, T. Feature extraction methods for character recognition-a survey. **Pattern Recognition**, v. 29, n. 4, p. 641–662, 1996. ISSN 0031-3203. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0031320395001182>>.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F. et al. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2012. v. 25. Disponível em: <<https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>>.
- LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998.