

Professores:	António Correia	Disciplina:	Aprendizagem Computacional
Assunto:	Assignment 1	Data:	23 de outubro de 2023
Alunos:	Tomás Dias	Nº. de Estudante:	2020215701
	Ricardo Guegan		2020211358
Curso:	MEI	Turma	PL3

Relatório Projeto de Machine Learning

Todo o código desenvolvido para resolver este enunciado está presentes no ficheiro `Projeto_ML.zip` fornecido em anexo.

Introdução

Neste projeto vamos explorar reconhecimento de dígitos através do uso de redes neuronais. Vamos, a partir de um dataset de 500 dígitos desenhados por nós

Análise do DataSet

O data set utilizado é completo no sentido de ter todos os dígitos numa quantidade equilibrada, porém não é representativo do mundo real pois os dígitos foram desenhados por apenas duas pessoas e como decidimos trabalhar com uma quantidade relativamente baixa de dados preferimos focar em fazer dígitos que fossem todos semelhantes entre eles ao invés de cobrir o máximo de formas de fazer cada dígito o que teria levado a pior resultados. Como podemos ver na figura 1, aquilo que mais varia entre os mesmos dígitos são o tamanho e a precisão das linhas curvas.

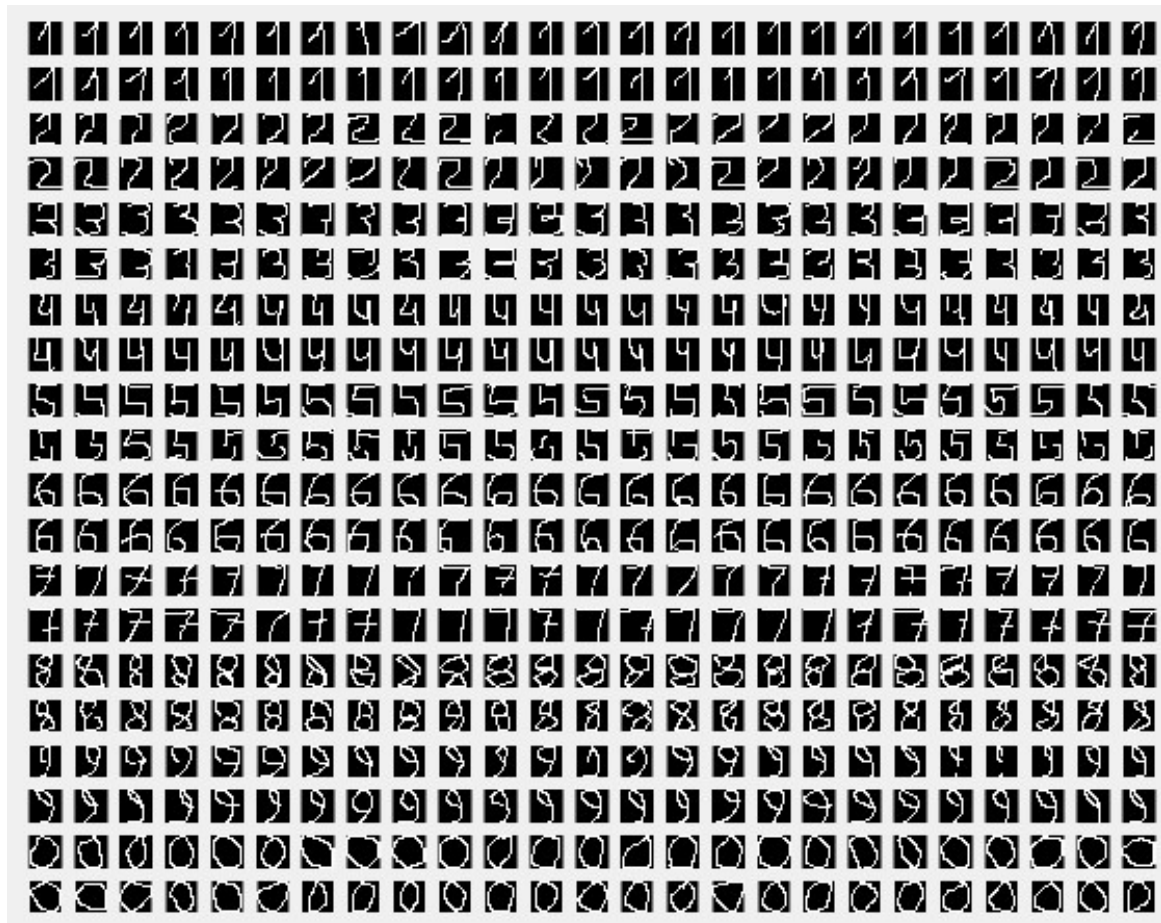


Figura 1: Dataset utilizado

Arquiterura e Implementação

Foram implementadas tres arquiteturas diferentes, Rede Neuronal de uma camada, Rede Neuronal duas camadas, Filtro + Rede Neuronal. Nestas testamos diferentes funções de ativação, treino e medidas de performance. As diferentes redes neuronais foram implementadas e configuradas com o uso das funções da deeplearning toolbox do MATLAB.

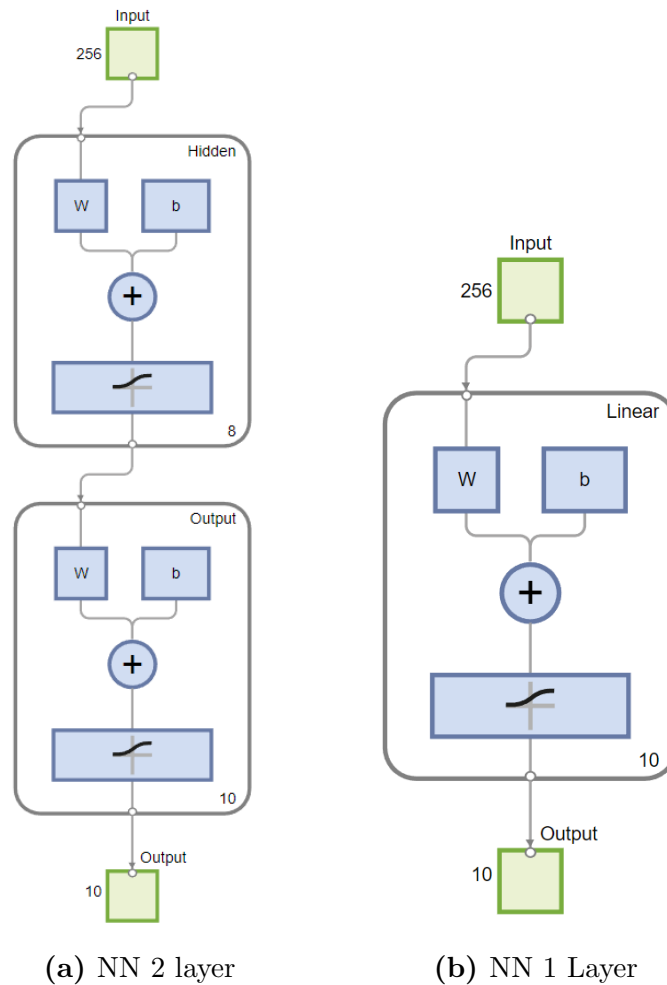


Figura 2: Arquiteturas das Redes Neuronais

Usamos também dois filtros para tentar reconstruir os dígitos em outros mais reconhecíveis para as redes neurais, o primeiro filtro é feito utilizando o metodo da pseudo-inversa, o segundo é um binary perceptron.

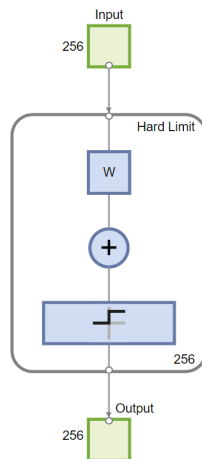


Figura 3: Binary Perceptron

Após a aplicação dos filtros ao dataset original, os resultados podem ser vistos nestas imagens, como podemos observar o metodo da pseudo inversa é menos eficaz e junta vários dígitos no mesmo quadrado, pelo outro lado o metodo do perceptron devido á sua natureza do seu algoritmo todos os dígitos são transformados em uma versão do PerfectArial, mesmo que em alguns casos não seja o correto.

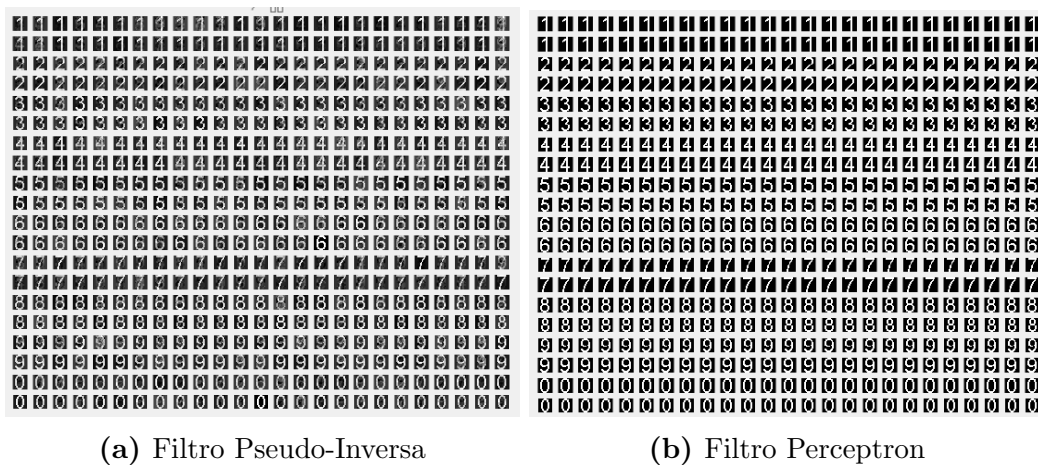


Figura 4: Resultado da Aplicação dos Filtros

Tabela 1: Tabela Das Arquiteturas

Arquitetura	ActivFunc	TrainAcc	TrainMSRE	ValAcc	ValMSRE
Associative Memory + NN	HardLim	10.0	0.9487	10.0	1.062
	purelin	100.0	0.1110	86.0	0.2220
	logsig	100.0	0.0017	90.0	0.0068
Perceptron + NN	HardLim	18.0	0.9487	10.0	1.074
	purelin	100.0	1.2182e-08	96.0	2.0433e-03
	logsig	100.0	6.8747e-04	98.0	0.2518
One Layer Neural Network	HardLim	18.0	0.9487	11.0	1.018
	purelin	82.0	0.1110	74.0	0.2220
	logsig	93.5	0.0245	88.0	0.098
Two Layer Neural Network	HardLim	40.25	0.5166	7.0	0.6943
	purelin	90.0	0.1959	12.0	0.2919
	logsig	95.0	0.1621	53.0	0.1935
Pattern Network		100.0	0.1287	88.0	0.1764

Tabela 2: Tabela das Epochs

Epochs	TrainAcc	TrainMSRE	ValAcc	ValMSRE
100	94.5000	0.0314	90	0.1378
500	99	0.0104	100	0.1444
1000	99.7500	0.0175	87	0.1415

Tabela 3: Divisao Train Validation

Ratio	TrainAcc	TrainMSRE	ValAcc	ValMSRE
50/50	100.0	0.0025	79.2000	0.1793
75/25	100.0	0.0156	84.0	0.1614
80/20	100.0	0.0091	90.0	0.1309
90/10	100.0	5.3162e-04	78.0	0.1740

Avaliação dos Resultados

Para avaliar as diferentes arquiteturas utilizamos como métricas a accuracy, que é obtida ao aplicar softmax em pós-processamento, e a MSRE que representa o desvio entra o input e a target. Todos os resultados foram obtidos com uma função de treino "trainsgc" e com um metodo de avaliação de performance que é o MSE.

Com base na tabela podemos concluir que para todas as arquiteturas, a função de ativação que obteve melhores resultados foi a sigmoide e que a pior é a hardlim, isto segue os resultados esperados pois a sigmoide é aquela que tem uma curva mais suave e que se mantém entre 0 e 1, contrariamente á purelin que não tem limites.

Consideramos que em quase todos os casos a Rede Neuronal conseguiu aprender, pois a sua accuracy é superior a 10 e o seu MSRE é inferior a 0.5.

Aprendizagens

Com este trabalho de reconhecimento de dígitos aprendemos fundamentos importante ao trabalhar com redes neuronais:

1. Classificadores: Aprendemos que na concepção de sistemas classificadores come é o caso deste queremos obter uma resposta booleana sobre se é ou não um determinado dígito, para tal temos de definir um critério de seleção e no nosso caso foi aquele que no output tinha o maior valor era aquele que considerávamos selecionado. Com tal critério de seleção utilizar um soft max para limitar o valor entre 0 e 1 não teve utilidade e logo não melhorou neu piorou os resultados.
2. Epochs: Percebemos que apesar de o número de epochs ter um impacto importante na performance da rede e que esta tem de ser suficientemente grande, no nosso caso 500, depois deste valor a rede já não melhora muito. A aprendizagem evolui de forma logaritmica com o progresso das epochs.
3. Training vs Validation: Como podemos ver pelos resultados, muitas vezes os valores dos resultados nos sets de treino são muito maiores do que nos de validação. Este problema é principalmente notável na Two Layer Network, onde devido á criação de novas features pelo facto de ter duas camadas faz com que esta encontre features especiais para o training set o que leva a problemas de overffiting e daí a grande perda de performance e o aumento do MSRE no validation Set.
4. Uso de Filtros: Percebemos a ideia de usar um agente para tentar corrigir a Data e depois treinar um segundo agente com a data corrigida. Este encadeamento de filtro mais classificador foi aquele que obteve melhores resultados.
5. Complexidade da rede: Percebemos que uma rede com mais camadas e com mais neurónios nem sempre é a melhor solução pois esta entra em problemas de overfitting. Isto é visível no facto de a one layer Network obter resultados superiores á two layer network no validation set. Também foi com apenas 8 neurónios na Hidden Layer da Two Layer Network que obtivemos os melhores resultados no validation Set.
6. Ratio Train Validation: como podemos ver na tabela da divisao Train Validation, o optimal é uma proporção de 80/20, pois este tem exemplos suficientes no train porém, mantém uma parte da população escondida oque faz com que o valor do validation set seja importante, abaixo deste valor temos falta de exmplos, acima deste valor temos overfitting.

Conclusão

Este projeto foi para nós a primeira interação com Redes Neuronais e com a Deeplearning Toolbox do MATLAB, através da qual explorámos a construção de redes neuronais e a sua parametrização. Descobrimos pelo metodo da experimentação vários conceitos fundamentais em Aprendizagem Computacional que foram mencionados na secção das aprendizagens e finalmente conseguimos construir vários classificadores capazes de reconhecer números com uma performance bastante satisfatória, considerando que a performance é medida num dataset feito por apenas duas pessoas.