

Neural Network

Lucas Ribeiro Neis
lucasrneis@gmail.com

Vinícius Couto Biermann
viniciusbiermann@hotmail.com

Novembro 2016

1 Introdução

Foi proposto um trabalho para reconhecimento de dígitos manuscritos utilizando o conhecimento adquirido de redes neurais. Foi disponibilizado o arquivo `exdata.m` contendo dados obtidos a partir de um subconjunto da base de dados MNIST. Foi escolhida a ferramenta `MATLAB` para a realização do trabalho.

2 Desenvolvimento

A normalização dos dados foi feita usando o comando `mapminmax`.

```
[inTrain_N, PS] = mapminmax(inTrain);
```

Inicialmente, separamos o conjunto de entradas em dois conjuntos, um para testes e outro para treinamento. Mas, após alguns testes, alteramos para apenas um conjunto separando-o através do uso de parâmetros:

```
net.divideParam.trainRatio=0.7;  
net.divideParam.valRatio=0.0;  
net.divideParam.testRatio=0.3;
```

A ausência de validação deve-se que conseguimos melhores resultados sem ela.

Além disso, testamos diversos algoritmos e combinações de números de camadas com diversas funções. Conseguimos resultados satisfatórios com **Conjugate Gradient with Beale-Powell Restarts** (ou `traincgb` no `MATLAB`) como mostrado na figura 1. Infelizmente, não conseguimos usar **Levenberg-Marquardt** devido a problemas com a execução.

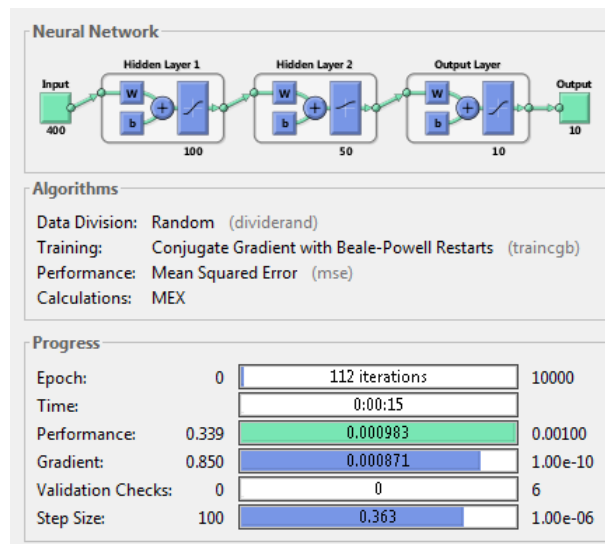


Figura 1: Exemplo de uma tela de testes.

Como mostra a figura 2, usamos duas camadas escondidas:

- Hidden Layer 1: São cem neurônios usando `tansig`.
- Hidden Layer 2: São cinquenta neurônios usando `logsig`.

Além disso, a camada de saída tem os 10 neurônios por padrão e usa `tansig` como funções de saída.

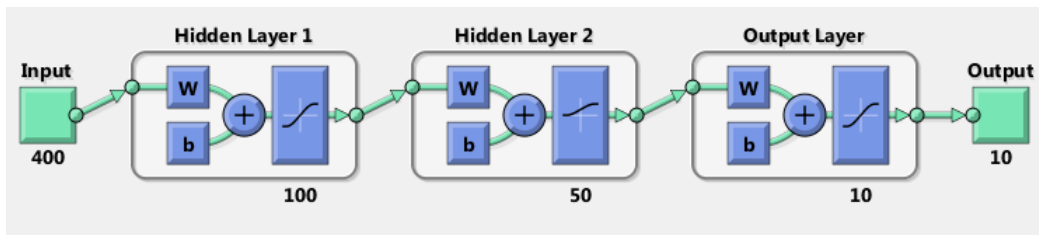


Figura 2: Arquitetura da rede utilizada.

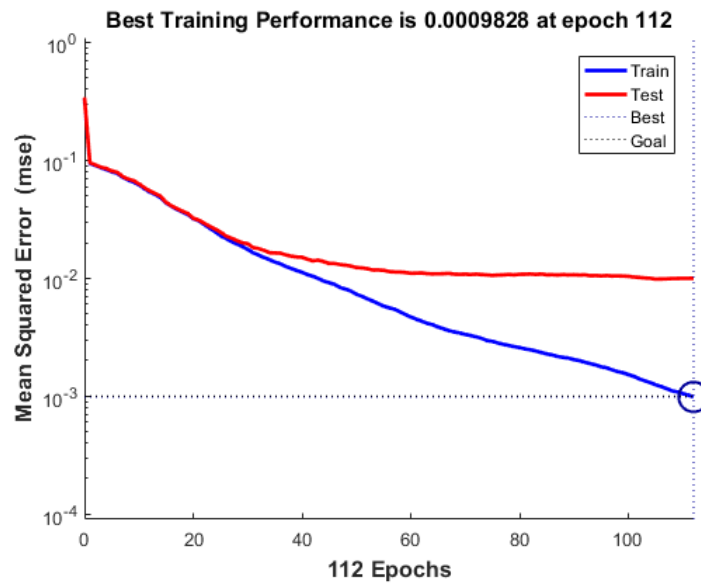


Figura 3: Gráfico de performance no treinamento e teste.

Confusion Matrix

		1	2	3	4	5	6	7	8	9	10	
1	490	1	2	0	3	0	0	2	1	0	98.2%	
	9.8%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	1.8%	
2	2	490	5	1	2	0	2	0	2	0	97.2%	
	0.0%	9.8%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.8%	
3	2	0	479	0	4	0	1	1	5	1	97.2%	
	0.0%	0.0%	9.6%	0.0%	0.1%	0.0%	0.0%	0.0%	0.1%	0.0%	2.8%	
4	0	0	1	494	0	0	2	3	1	0	98.6%	
	0.0%	0.0%	0.0%	9.9%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	1.4%	
5	1	1	6	0	484	3	2	3	0	0	96.8%	
	0.0%	0.0%	0.1%	0.0%	9.7%	0.1%	0.0%	0.1%	0.0%	0.0%	3.2%	
6	0	1	0	0	0	492	0	2	0	1	99.2%	
	0.0%	0.0%	0.0%	0.0%	0.0%	9.8%	0.0%	0.0%	0.0%	0.0%	0.8%	
7	1	4	2	1	0	0	488	0	2	0	98.0%	
	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	9.8%	0.0%	0.0%	0.0%	2.0%	
8	3	3	0	0	4	1	0	486	1	0	97.6%	
	0.1%	0.1%	0.0%	0.0%	0.1%	0.0%	0.0%	9.7%	0.0%	0.0%	2.4%	
9	1	0	2	4	2	1	4	3	485	0	96.6%	
	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	0.1%	9.7%	0.0%	3.4%	
10	0	0	3	0	1	3	1	0	3	498	97.8%	
	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	10.0%	2.2%	
	98.0%	98.0%	95.8%	98.8%	96.8%	98.4%	97.6%	97.2%	97.0%	99.6%	97.7%	
	2.0%	2.0%	4.2%	1.2%	3.2%	1.6%	2.4%	2.8%	3.0%	0.4%	2.3%	
	1	2	3	4	5	6	7	8	9	10		
	Target Class											

Figura 4: Matriz de confusão obtida.