



Universidade Federal do Rio Grande do Norte  
Centro de Tecnologia  
Departamento de Engenharia de Computação e  
Automação

## DCA0121 – Inteligência Artificial Aplicada

### Projeto de Unidade

Docente:

Marcelo Augusto Costa Fernandes

Discentes:

Alexandre Henrique Soares Dias

Gabriel Medeiros Coelho

Josué Oliveira de Araújo

MAIO/2018

## Proposta

O objetivo do nosso projeto é implementar uma rede neural que reconhece dígitos alfa numéricos manuscritos baseado no algoritmo MLP. A base de dados usada para treinamento e testes foi a MNIST, que é uma fonte confiável e amplamente utilizada para a solução de problemas como o nosso.

Os dados são dispostos aleatoriamente, e eles representam imagens de 28x28 pixels. Abaixo mostramos alguns exemplos de dados da MNIST database.



A partir das características dos nossos dados, modelamos nossa rede neural com 784 entradas, onde cada entrada representa um pixel da imagem, e uma camada oculta entre a camada de entrada e saída. Utilizamos 40000 imagens para treinamento em cada época, e 5000 imagens para validar o desempenho da rede neural. Também vale ressaltar que o treinamento ocorreu de modo iterativo. Após uma série de testes, conseguimos definir os parâmetros da rede, que foram:

Passo de aprendizagem = 0.2

Número de neurônios na camada oculta = 450

Número de neurônios na camada de saída = 10

Número de épocas para o treinamento = 7

Inicialmente, obtivemos uma convergência com em média 66% de acertos na saída da rede neural, após ajustar estes parâmetros exaustivamente, nossa taxa de acerto subiu para quase 88%. Sendo que, somos plenamente conscientes de que se aumentarmos o número de épocas no treinamento, certamente aumentaremos a

taxa de acerto, no entanto, acreditamos que os resultados obtidos foram suficientes para demonstrar que nossa rede neural funciona.

Uma saída que obtivemos do nosso algoritmo é mostrada logo a seguir

```
Taxa de acerto = 87.36%
Taxa de acerto de cada dígito:
Dígito 0: 97.6087%
Dígito 1: 97.1979%
Dígito 2: 85.8491%
Dígito 3: 85.8%
Dígito 4: 89.8%
Dígito 5: 77.4123%
Dígito 6: 90.2597%
Dígito 7: 89.6484%
Dígito 8: 76.8916%
Dígito 9: 81.9231%
```

Nela é possível visualizar a taxa de acerto global e específica para cada dígito, em destaque para o dígito 5, que durante todos os treinamentos se mostrou um dos mais difíceis de serem caracterizados.

Fotos que mostram algumas saídas obtidas durante o treinamento evidenciam a evolução do reconhecimento do dígito 5 por parte da rede neural. A saída é constituída de 10 valores, cada um representando cada dígito, onde da esquerda para a direita temos os valores correspondentes à saída do número 0 à 9, de tal modo que a saída esperada para um dígito é sempre 1 no índice correspondente a ele e zero nos demais.

Para a época 2 do treinamento, obtivemos a saída:

```
vetor de saidas: 0.000203557 6.07477e-05 7.61414e-06 0.746243 5.21176e-08 0.407169 3.32947e-06 0.00087013 0.00199591 1.79792e-05
saída esperada: 0 0 0 0 1 0 0 0 0 0
vetor de erros: -0.000203557 -6.07477e-05 -7.61414e-06 -0.746243 -5.21176e-08 0.592831 -3.32947e-06 -0.00087013 -0.00199591 -1.79792e-05
Z's da camada 2: -8.49936 -9.70872 -11.7855 1.07868 -16.7698 -0.375683 -12.6127 -7.046 -6.21466 -10.9263
Z's da camada 1: 0.56176 0.376093 0.194393 0.402744 0.281204 0.361586 0.451602 0.691442 0.760227 0.434741
label = 5
vetor de saidas: 6.10784e-05 2.3083e-05 1.06695e-05 0.999549 2.57815e-07 0.611121 3.45586e-07 0.000262086 2.38481e-05 5.50209e-06
saída esperada: 0 0 0 1 0 0 0 0 0 0
vetor de erros: -6.10784e-05 -2.3083e-05 -1.06695e-05 0.000450654 -2.57815e-07 -0.611121 -3.45586e-07 -0.000262086 -2.38481e-05 -5.50209e-06
Z's da camada 2: -9.70329 -10.6764 -11.4481 7.70436 -15.171 0.452025 -14.878 -8.24658 -10.6438 -12.1104
Z's da camada 1: 0.818847 0.513326 0.393714 0.624213 0.362148 0.592737 0.745848 0.996442 0.687243 -0.152238
label = 3
```

Se observarmos atentamente, para o label 5, a saída na segunda época apresenta um valor maior para o índice 4 (isto é, dígito 3), que corresponde a 0.746243 enquanto que a saída do próprio dígito 5 tem um valor menor, igual a 0.407169. Isto

é, na segunda época, a rede não sabe classificar claramente um dígito 5. Enquanto que para a saída do label 3, vemos que mesmo na segunda época, ela já desponta em relação aos outros. Em outras palavras, a rede consegue classificar bem o dígito 3.

Agora, vamos analisar uma saída da época 4 para os dígitos 3 e 5:

```
vetor de saidas: 9.17498e-06 1.31803e-09 8.17518e-11 0.916175 5.50377e-13 0.963286 6.48076e-08 5.91119e-05 2.98629e-07 4.47136e-09
Saída esperada: 0 0 0 0 1 0 0 0 0
vetor de erros: -9.17498e-06 -1.31803e-09 -8.17518e-11 -0.916175 -5.50377e-13 0.0367144 -6.48076e-08 -5.91119e-05 -2.98629e-07 -4.47136e-09
Z's da camada 2: -11.599 -20.4471 -23.2273 2.39148 -28.2282 3.26718 -16.5518 -9.73602 -15.0241 -19.2256
Z's da camada 1: 0.851004 0.593384 0.418423 0.541495 0.573808 0.846844 0.70939 0.971496 1.08287 0.680899
Label = 5
vetor de saidas: 6.91333e-07 2.09975e-07 2.81965e-08 0.999993 2.01079e-10 0.00129851 2.22015e-10 1.00269e-06 1.49414e-08 8.80637e-13
Saída esperada: 0 0 0 1 0 0 0 0 0
vetor de erros: -6.91333e-07 -2.09975e-07 -2.81965e-08 7.22076e-06 -2.01079e-10 -0.00129851 -2.22015e-10 -1.00269e-06 -1.49414e-08 -8.80637e-13
Z's da camada 2: -14.1846 -15.3763 -17.3841 11.8385 -22.3273 -6.64524 -22.2283 -13.8128 -18.0191 -27.7581
Z's da camada 1: 1.29665 0.993151 0.620683 0.676773 0.63488 1.1287 1.17594 1.3244 0.863022 -0.00196835
Label = 3
```

Neste caso, é possível perceber que o número 5 obteve uma saída melhor da rede, igual à 0.963286, contudo, o valor de saída para o índice de label 3 (0.916175) ainda está próximo do valor do índice 5, e próximo de 1. Portanto, a rede reconhece razoavelmente bem o número 5, enquanto que ainda o confunde um pouco com o número 3.

Em contrapartida, para a saída do label 3, já temos um valor expressivo que o caracteriza dentre os demais. Dessa maneira, entendemos que a rede neural apresenta um ótimo reconhecimento deste dígito.

É válido observar que o vetor de erros apresenta valores menores em relação às saídas mostradas na época 2.

Por último, mostramos uma saída da época 7, novamente com ênfase nos dígitos 3 e 5:

```
vetor de saidas: 2.99943e-12 1.75035e-14 2.44855e-13 0.000815089 4.14333e-18 0.901031 2.18641e-12 8.70743e-06 1.37507e-10 6.8395e-14
Saída esperada: 0 0 0 0 1 0 0 0 0
vetor de erros: -2.99943e-12 -1.75035e-14 -2.44855e-13 -0.000815089 -4.14333e-18 0.0989693 -2.18641e-12 -8.70743e-06 -1.37507e-10 -6.8395e-14
Z's da camada 2: -26.5326 -31.6764 -29.0381 -7.1114 -40.025 2.20873 -26.8488 -11.6513 -22.7073 -30.3135
Z's da camada 1: 1.13464 0.846794 0.466802 0.392373 0.642542 1.08807 0.829749 1.08972 1.29572 0.976038
Label = 5
vetor de saidas: 8.12831e-09 1.96775e-12 8.71525e-13 1 6.76302e-13 1.25404e-06 8.84786e-15 1.33895e-09 4.09189e-11 8.93621e-18
Saída esperada: 0 0 0 1 0 0 0 0 0
vetor de erros: -8.12831e-09 -1.96775e-12 -8.71525e-13 6.13664e-09 -6.76302e-13 -1.25404e-06 -8.84786e-15 -1.33895e-09 -4.09189e-11 -8.93621e-18
Z's da camada 2: -18.6279 -26.9541 -27.7685 18.909 -28.0221 -13.5891 -32.3586 -20.4314 -23.9194 -39.2564
Z's da camada 1: 1.65704 1.33546 0.822493 0.817831 0.952743 1.33773 1.2381 1.41134 0.926539 0.0778239
Label = 3
```

Finalmente, neste caso, claramente a rede melhorou a classificação para o dígito 5, ao passo que os erros todos tendem a zero, e ainda, o valor de saída no índice do dígito 3 também está próximo de ótimo. Disso decorre a conclusão de que a rede já classifica bem o dígito 5.

E por último, vemos que para a saída do Label 3, não há mais dúvidas de que a rede aprendeu a reconhecê-lo de forma ótima.

O link para o vídeo com a explicação da implementação da rede neural é o seguinte:

<https://drive.google.com/open?id=1fQOkFLdTe4qnScrQv8Rb3quGSoHI9fgz>