



UNIVERSIDADE FEDERAL DO PIAUÍ
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS
DISCIPLINA: SISTEMAS INTELIGENTES
PROFESSORA: DEBORAH MAGALHÃES

4a PRÁTICA COMPUTACIONAL: Parte I

Redes Neurais

1. Descrição do Trabalho

O objetivo dessa prática reside em você adquirir noções básicas sobre como a Rede Neural funciona e aplicá-la para reconhecer dígitos manuscritos. O reconhecimento automatizado de dígitos manuscritos é amplamente utilizado atualmente - desde o reconhecimento de códigos postais (CEPs) em envelopes de correio até o reconhecimento de quantias escritas em cheques bancários. Para tanto, será utilizada a base de dados (**ex3data1.mat**). Tal base, contém 5000 exemplos de treinamento de dígitos de 0-9 escritos a mão, conforme ilustrado na figura abaixo:



Cada exemplo de treinamento corresponde à uma imagem do dígito em tons de cinza (20 pixels x 20 pixels), acarretando em um vetor de dimensão 400. Cada pixel é representado por um número de ponto flutuante que indica a intensidade da escala de cinza nesse local. Portanto, a base corresponde a uma matriz de 5000 x 400.

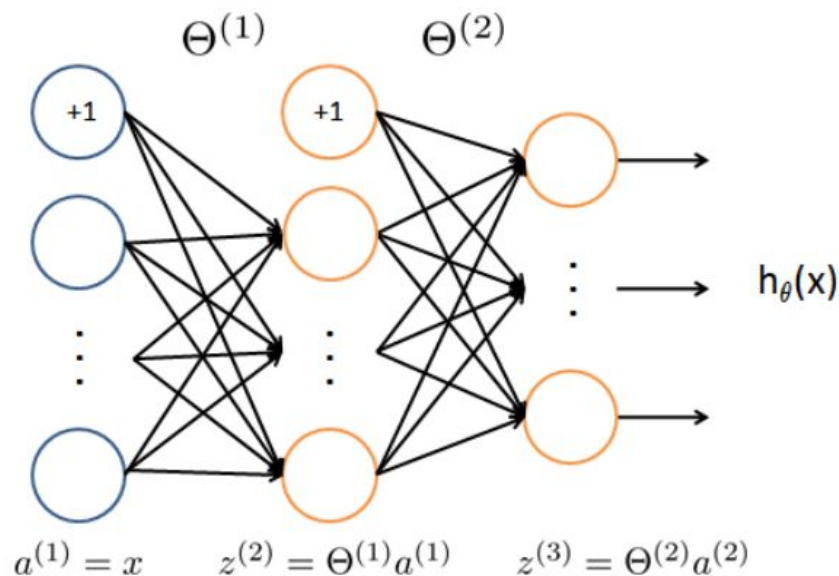
A segunda parte do conjunto de treinamento é um vetor de dimensão 5000 que contém rótulos para o conjunto de treinamento. Para facilitar a indexação, não existe um índice zero, o dígito zero é mapeado para o valor dez, portanto, um dígito “0” é rotulado como “10”, enquanto os dígitos “1” a “9” são rotulados normalmente de “1” a “9”.

Os códigos necessários sua execução são:

- ex3_nn.m - Script principal que chama as demais funções
- ex3data1.mat - Conjunto de treinamento com os dígitos
- ex3weights.mat - Pesos iniciais da rede neural
- * **predict.m - função de predição da rede neural**
- displayData.m - função para visualização dos dados
- sigmoid.m - implementação da função sigmoidal

* São as funções que você deve modificar. Todas elas estão disponíveis no sigaa.

A arquitetura projetada para este problema é ilustrada na Figura abaixo, onde temos 3 camadas: 1 camada de entrada com tamanho 400 (imagem 20 x 20); 1 camada oculta com 25 neurônios e, finalmente, 1 camada de saída com 10 neurônios (10 dígitos).



1.1. O que deve ser feito?

Você deverá codificar o algoritmo propagação para frente (feedforward) dentro da função `predict.m` conforme ilustrado na figura abaixo. Essa função será responsável por prever as labels do conjunto de treinamento.

```
function p = predict(Theta1, Theta2, X)
% A função predict é responsável por prever uma label da imagem de entrada X,
% dada as matrizes de pesos Theta1 e Theta2.

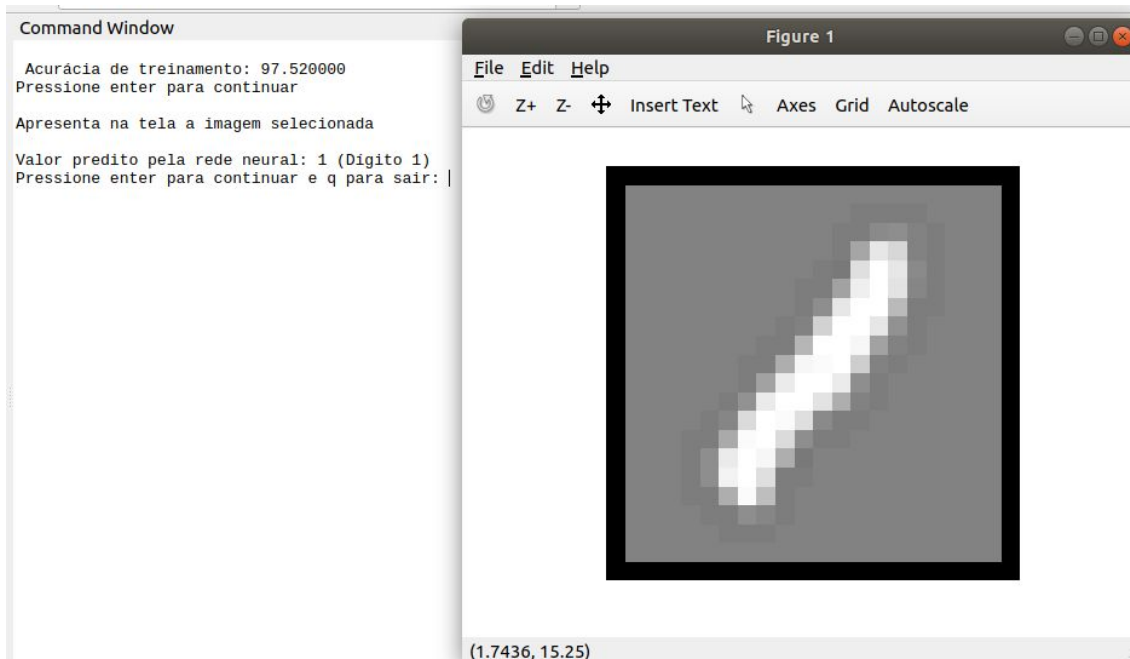
% Inicializando os valores das variáveis
m = size(X, 1);
num_labels = size(Theta2, 1);
p = zeros(size(X, 1), 1);

% não esqueça de add os bias das camadas de ativação
a1 = ??
z2 = ??
a2 = ??
z3 = ??
a3 = ??

[val, p] = max(a3, [], 2);

end
```

A função **ex3_nn.m** chamará a função **predict.m**, usando como parâmetro as matrizes de peso `Theta1` e `Theta2`. Espera-se encontrar uma acurácia em torno de 97.5%. A função **displayData.m** é responsável por renderizar na tela, o valor predito pela rede neural, como pode ser observado abaixo:



2. Avaliação

Este trabalho corresponde a uma parte da segunda avaliação parcial da disciplina e deverá ser entregue no dia **07/11**. O trabalho poderá ser apresentado de dupla e assumirá o valor de **0-2.5**.

Os seguintes critérios serão considerados na avaliação:

1. Atender ao que foi pedido na descrição deste documento;
2. Compreender os conceitos discutidos em sala;
3. Código está executando sem erros.

Atenção: se identificada a cópia de código, a nota **zero** será atribuída aos envolvidos.