

ICMC - USP

SCC0270 - Redes Neurais

Projeto 1

Multilayer Perceptron

Docente: Roseli Aparecida Francelin Romero

Aluno: Lucas Antognoni de Castro, 8936951

São Carlos

Setembro 2017

Enunciado

Para o primeiro projeto da disciplina foram implementadas, treinadas e testadas duas redes neurais para a resolução de problemas: um de classificação e outro de aproximação. Codificou-se o modelo de Perceptron Multicamadas (*Multilayer Perceptron*) que é uma rede neural semelhante à perceptron, mas com mais de uma camada intermediária (escondida) de neurônios e utiliza o algoritmo de aprendizado supervisionado *Backpropagation* para treinamento. Foram utilizados diferentes valores para os parâmetros da rede neural, a fim de verificar o impacto dos mesmo no aprendizado e posterior teste da mesma.

Plataforma utilizada

O referido trabalho foi confeccionado no sistema operacional **Linux** (*Ubuntu*), utilizando um ambiente de desenvolvimento com a **IDE Visual Studio Code** e terminal. O código foi desenvolvido na linguagem de programação **Python** e suas bibliotecas.

Dados

Foram utilizadas duas bases de dados para realizar a classificação e aproximação, respectivamente com a ***bluetooth.csv*** e ***test.txt***. Os dados foram

pré-processados, para que fossem normalizados entre os valores [0, 1].

Para o primeiro conjunto, cada elemento da coluna de entrada foi dividido pelo maior valor negativo de sua respectiva coluna, enquanto as classes foram divididas pelo maior valor. O segundo conjunto também teve cada elemento das colunas de entrada e classe dividido pelo maior valor das mesmas. Na implementação as funções *process_bluetooth()* e *process_test()*, são responsáveis por tal processamento dos dados.

Para cada base também foram testadas as seguintes proporções para treinamento/teste:

- 50/50
- 60/40
- 70/30

Testes e Resultados

As redes neurais utilizadas apresentam as seguintes configurações:

- Classificação (bluetooth)

Entrada	Escondida	Saída
3	20	1

- Aproximação (test)

Entrada	Escondida	Saída
14	155	1

O número de camadas escondidas e o número de neurônios na mesma foram definidos através de:

$$N_h = \frac{N_s}{(\alpha * (N_i + N_o))}$$

- N_h = número de neurônios na camada escondida
- N_s = número de amostras no conjunto de treinamento
- N_i = número de neurônios na camada de entrada
- N_o = número de neurônios na camada de saída
- α = valor arbitrário entre [2, 10]

Tal fórmula permite uma maior acurácia no treinamento da rede neural, vide explicação no link:

- <https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-netw>

Para os testes fixou-se o valor de *alfa* como 2.

Em cada treinamento e teste, variou-se a porcentagem dos dados, *eta*, e *momentum*. O *threshold* foi mantido em 0.015, pois mostrou-se o melhor valor base para ambos problemas e seus dados, nos quais as redes conseguiam atingi-lo em praticamente todos os experimentos. A seguir são exibidas tabelas comparativas com as respectivas acurácias obtidas com cada teste em ambos os conjuntos de dados. As células que apresentam o carácter '-' indicam que a rede neural, devido a sua configuração, não conseguiu reduzir o erro para menor que o *threshold*, ou o tempo do treinamento desta foi exageradamente longo, não atingindo também o limiar, e sua execução foi abortada.

- Classificação (bluetooth)

Porcentagem	<i>Eta</i>	<i>Threshold</i>	<i>Momentum</i>	Acurácia
-------------	------------	------------------	-----------------	----------

0.5	0.1	0.015	0.1	94.7368
0.5	0.5	0.015	0.1	97.3684
0.5	0.9	0.015	0.1	92.1052
0.5	0.1	0.015	0.5	-
0.5	0.1	0.015	0.9	-
0.5	0.5	0.015	0.5	97.3684
0.5	0.5	0.015	0.9	92.1052
0.5	0.9	0.015	0.5	92.1052
0.5	0.9	0.015	0.9	89.4736
0.6	0.1	0.015	0.1	100.0
0.6	0.5	0.015	0.1	98.7252
0.6	0.9	0.015	0.1	95.9824
0.6	0.1	0.015	0.5	95.8262
0.6	0.1	0.015	0.9	-
0.6	0.5	0.015	0.5	97.2725
0.6	0.5	0.015	0.9	87.2451
0.6	0.9	0.015	0.5	85.8724
0.6	0.9	0.015	0.9	-
0.7	0.1	0.015	0.1	90.9090
0.7	0.5	0.015	0.1	95.4545
0.7	0.9	0.015	0.1	97.6742
0.7	0.1	0.015	0.5	91.8252
0.7	0.1	0.015	0.9	87.6452
0.7	0.5	0.015	0.5	90.0826
0.7	0.5	0.015	0.9	84.1126
0.7	0.9	0.015	0.5	80.0976
0.7	0.9	0.015	0.9	-

- Aproximação (test)

Porcentagem	<i>Eta</i>	<i>Threshold</i>	<i>Momentum</i>	Acurácia
0.5	0.1	0.015	0.1	77.3913
0.5	0.5	0.015	0.1	78.2608
0.5	0.9	0.015	0.1	80.0
0.5	0.1	0.015	0.5	76.9565
0.5	0.1	0.015	0.9	74.5643
0.5	0.5	0.015	0.5	80.5745
0.5	0.5	0.015	0.9	75.8672
0.5	0.9	0.015	0.5	70.8262
0.5	0.9	0.015	0.9	71.0235
0.6	0.1	0.015	0.1	72.8260
0.6	0.5	0.015	0.1	70.1086
0.6	0.9	0.015	0.1	61.9565
0.6	0.1	0.015	0.5	74.2927
0.6	0.1	0.015	0.9	-
0.6	0.5	0.015	0.5	70.9872
0.6	0.5	0.015	0.9	-
0.6	0.9	0.015	0.5	68.8262
0.6	0.9	0.015	0.9	72.6245
0.7	0.1	0.015	0.1	80.8262
0.7	0.5	0.015	0.1	83.2172
0.7	0.9	0.015	0.1	77.2725
0.7	0.1	0.015	0.5	78.3825
0.7	0.1	0.015	0.9	75.0825
0.7	0.5	0.015	0.5	77.2815
0.7	0.5	0.015	0.9	71.7543

0.7	0.9	0.015	0.5	68.9826
0.7	0.9	0.015	0.9	65.0982

Executando

Extraia os arquivos da pasta e em um terminal navegue até o diretório do exercício e digite o seguinte comando:

```
python3 project.py
```

O programa será executado e os resultados serão impressos na tela.