

ICMC - USP

SCC0270 - Redes Neurais

Exercício 2

Auto-associador

Docente: Roseli Aparecida Francelin Romero

Aluno: Lucas Antognoni de Castro, 8936951

São Carlos

Setembro 2017

Enunciado

Implementar e treinar uma rede neural relacionada ao problema do auto-associador (*Encoding Problem*) em que um conjunto de padrões ortogonais são mapeados num conjunto de padrões de saída também ortogonais, através de uma camada escondida com poucos neurônios.

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

O dado utilizado no programa é uma matriz identidade com dimensões 10×10 , contendo valores “0” ou “1”.

```
1000000000
```

```
0100000000
```

```
0010000000
```

```
0001000000
```

```
0000100000
0000010000
0000001000
0000000100
0000000010
0000000001
```

Cada linha da matriz representa um padrão de entrada que será mapeado na rede neural. A matriz está armazenada no arquivo *id_10.dat*.

Testes e Resultados

A rede neural utilizada apresenta a seguinte estrutura:

Entrada	Escondida	Saída
10	3	10

Para a camada escondida foi definido o número de neurônios aplicando $\log_2 N$, em que N é o número de colunas da matriz de entrada. Tanto a camada de entrada quanto a de saída apresentam N neurônios.

Foram utilizados diversos valores para a taxa de aprendizado da rede neural (*eta*) e limiares (*threshold*). Ao final da bateria de testes obtivemos bons resultados, tanto em questão de acurácia, quanto velocidade de convergência com os seguintes valores de variáveis:

Eta	Threshold	Acurácia
0.5	1e-2	100.00%

O cálculo do número de acertos é feito realizando a comparação com a saída da rede e o resultado esperado para a mesma. À saída da rede é aplicada a função `np.round()` que realiza o arredondamento dos valores. Dessa maneira podemos identificar e realizar a comparação que incrementa um contador, caso os vetores sejam iguais.

Os valores dos pesos para as camadas escondida e de saída, bem como seus respectivos biases, se encontram nos arquivos *weight_n_bias_start.txt* (iniciais) e *weight_n_bias_end.txt* (finais).

Executando

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

```
python3 encoder.py
```

O programa será executado e os resultados impressos na tela.

Por padrão, os valores do `eta` e `threshold`, respectivamente: 0.5 e 1e-2 e podem ser alterados na chamada da função `encoder(eta, threshold)`.