

ICMC - USP

SCC0270 - Redes Neurais

# Exercício 6

## Rede de Hopfield

Docente: Roseli Aparecida Francelin Romero

Aluno: Lucas Antognoni de Castro, 8936951

São Carlos

Dezembro 2017

## Enunciado

---

Para o exercício 6 da disciplina foi implementada e testada a Rede de *Hopfield* para atuar como reconhecedor de algarismos entre 0 e 9.

## Plataforma utilizada

---

O referido trabalho foi confeccionado no sistema operacional *Linux* (*Arch*), 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 matrizes com valores binários (*0* e *1*) para representar cada um dos algarismos a serem memorizados pela rede. Cada padrão está contido em uma matriz quadrada *9x9*, que é passada como um vetor de *81* posições para a o algoritmo processar. Os dados são divididos em um conjunto de treinamento e outro para teste, sendo que este último apresenta ruídos que serão utilizados para verificar a capacidade de recuperação da rede.

## Testes e Resultados

---

Sabemos que a arquitetura da rede de Hopfield apresenta restrições em relação ao número de padrões que pode armazenar. Este número é regido pela seguinte equação:

$$M \leq 0.15 \cdot N$$

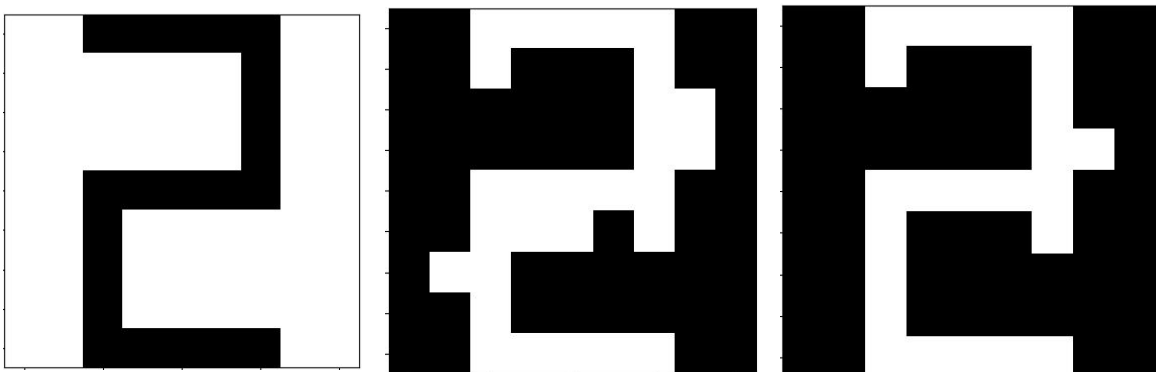
na qual  $M$  é o número de padrões e  $N$  o número de neurônios disponíveis na arquitetura da rede.

Levando em conta os 10 padrões de entrada, devemos ter no mínimo:

$$N = 10 / 0.15 = 66$$

para que a rede consiga memorizar todos os algarismos. Como  $8^2 = 64$ , devemos ter o número de neurônios igual a  $9^2 = 81$ , uma vez que a matrix  $W$  de neurônios deve ser quadrada. Representamos os algarismos com as mesmas dimensões para que tenhamos uma maior resolução dos mesmos.

Cada um dos padrões são apresentados a rede para treino e logo após os padrões com ruídos são apresentados para teste. Tomando o número **2** como exemplo, temos a seguinte entrada e saída respectiva:



Podemos verificar que a rede apresentou um bom grau de recuperação para padrão exemplo. Ao fazer a recuperação com a multiplicação do vetor de entrada com os pesos, a saída está sujeita a interferência dos outros padrões armazenados, num efeito conhecido como ***crosstalk***. Também a qualidade da recuperação está relacionada a capacidade dos dados serem linearmente separáveis, que em caso positivo, confere uma porcentagem de acurácia maior no processo.

## Executando

---

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

```
python3 hopfield.py
```

O programa será executado e os resultados impressos na tela.