

# **Aprendizagem Computacional - Trabalho Prático 2**

**João Tiago Márcia do Nascimento Fernandes - 2011162899**  
**Joaquim Pedro Bento Gonçalves Pratas Leitão - 2011150072**

6 de Outubro de 2014

# Índice

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Aplicação Desenvolvida</b>	<b>4</b>
2.1	Memória Associativa + Classificador . . . . .	5
2.2	Classificador . . . . .	6
2.3	Implementação em Matlab . . . . .	7
2.3.1	associativeMemory.m . . . . .	7
2.3.2	createNetwork.m . . . . .	7
2.3.3	myclassify.m . . . . .	7
2.3.4	saveNetwork.m . . . . .	7
2.4	Execução . . . . .	7
<b>3</b>	<b>Testes e Resultados</b>	<b>8</b>
<b>4</b>	<b>Conclusões</b>	<b>9</b>

# 1 Introdução

Este trabalho foca-se no reconhecimento de caracteres da numeração árabe, ou seja, os caracteres 0 a 9.

Pretende-se que este reconhecimento seja realizado por uma aplicação desenvolvida em *Matlab*, que faz uso de redes neuronais na sua arquitetura interna, disponíveis na *Neural Networks Toolbox* do próprio *Matlab*.

A aplicação desenvolvida visa o estudo de duas arquiteturas distintas no reconhecimento dos caracteres:

- Na primeira arquitetura a aplicação será constituída por uma *memória associativa* e um *classificador*
- Na segunda arquitetura a aplicação apenas recorre ao *classificador*

As duas arquiteturas apresentadas estão presentes nas figuras que se seguem:

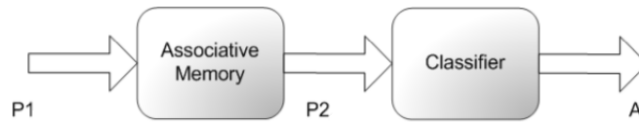


Figura 1: Arquitetura da aplicação com *memória associativa* + *classificador*



Figura 2: Arquitetura da aplicação apenas com o *classificador*

Através da análise destas figuras podemos determinar um comportamento padrão para a aplicação:

- Numa fase inicial, os caracteres a identificar poderão, ou não, ser fornecidos à *memória associativa*, que está encarregue da sua "filtragem" ou "correção": Se os caracteres fornecidos não forem perfeitos, a memória associativa aproxima-os dos respetivos caracteres perfeitos.
- De seguida os dados, corrigidos ou não, serão fornecidos ao *classificador*, que se encarregará de proceder à identificação dos mesmos.

No presente documento iremos proceder à apresentação em maior detalhe destas duas arquiteturas e das suas implementações, bem como da aplicação *Matlab* desenvolvida, e de como poderá ser utilizada. Pretendemos também fazer uma análise crítica da performance da aplicação, nomeadamente da sua capacidade de classificar corretamente novos caracteres fornecidos.

## 2 Aplicação Desenvolvida

## **2.1 Memória Associativa + Classificador**

## 2.2 Classificador

## 2.3 Implementação em Matlab

Indicar ficheiros criados e alterações ao código fonte (só dizer o que é que comentamos no mpaper.m)

### 2.3.1 associativeMemory.m

### 2.3.2 createNetwork.m

### 2.3.3 myclassify.m

### 2.3.4 saveNetwork.m

Este ficheiro não estamos a usar neste momento. Mantê-lo na aplicação e no relatório??

## 2.4 Execução

Explicar como executar a aplicação. Não esquecer que quando usamos a memória associativa assumimos que o utilizador desenha os caracteres 1;2;3;4;5;6;7;8;9;0 por esta ordem, em cada linha, preenchendo todas as linhas.

### **3 Testes e Resultados**

Descrição de como fizemos os casos de teste, dimensões, etc



## 4 Conclusões

Conclusões, lolol

**FIXME:** Testar classificação de dígitos perfeitos e de dígitos não perfeitos (alguns não perfeitos são corretamente classificados, e todos os perfeitos são corretamente classificados)

**FIXME:** Dizer que memória associativa só funciona se preencher-mos a tabela toda pela ordem: 1,2,3,4,5,6,7,8,9,0 linha-a-linha

**FIXME:** Perguntas relatório:

- How does the data set influence the performance of the classification system?
- Which architecture provides better results: only the classifier or the associative memory+classifier?
- Which is the best activation function: hardlim, linear or logsig?
- Does the Hebb rule perform well?
- Is the classification system able to achieve the main objectives (classification of digits)?
- Which is the percentage of well classified digits?
- How is the generalization capacity?
- Is the classification system robust enough to give correct outputs when new inputs are not perfect?
- Which is the percentage of well classified new inputs?