

Optical Character Recognition

AC-2014

António Lima
David Cardoso

Objectivo

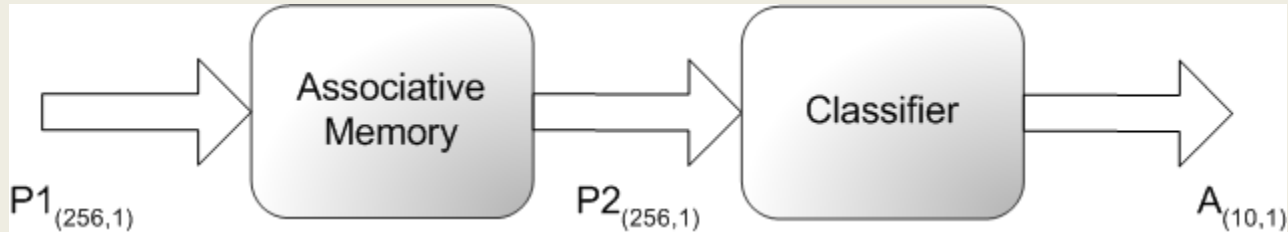
Desenvolver redes neuronais com vista a reconhecer os 10 numerais arábicos

1 2 3 4 5 6 7 8 9 0

Os dígitos a reconhecer são feitos pelo utilizador e semelhantes à fonte Arial.

Arquitetura do sistema

- Memória Associativa (opcional)
- Classificador



Input do sistema

- vector P, de tamanho 256 (matrix 16x16 que representa o desenho do numeral)

```
0000011110000000
0001100011110000
0011000000011000
0101000000001100
0110000000000110
0010000000000010
0001000000000011
0001100000000001
0000100000000001
0000110000000001
0000110000000001
0000011000000001
0000001100000001
0000000110000010
0000000011001110
0000000000111000
```

Memória Associativa

- Rede neuronal constituída por uma camada, com funções de activação lineares e sem propensão para determinados valores.
- Serve de filtro ou corrector.
- Caracterizada por:

$$P_2 = W_p \times P_1$$

em que os pesos (W_p) são calculados utilizando a transposta ou a regra de Hebb.

Classificador

- Rede neuronal constituída por uma camada
- Função de activação linear, sigmoidal ou perceptrão
- com propensão para determinados valores em cada neurónio

Produz uma classificação para o dígito de input que pode, ou não, ter passado pela memória associativa primeiro.

$$A = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Dataset

Como é que o dataset de treino influencia a performance do sistema de classificação?

- 50 exemplos, 5 para cada caracter numérico
- 100 exemplos, 10 para cada caracter numérico
- 500 exemplos, 50 para cada caracter numérico
- adicionar 10 exemplos perfeitos, 1 de cada caracter numérico em Arial

Caso de teste

Este foi o input basal utilizado para construir os restantes, sendo os seus elementos repetidos em iguais quantidades quantas vezes fosse necessário.

1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0

Resultados

Notação

Uso de Memória Associativa

1. Sim
2. Não

Modo de pesagem/ponderação (caso utilize Memória Associativa)

1. Método de pesagem transposta
2. Regra de Hebb

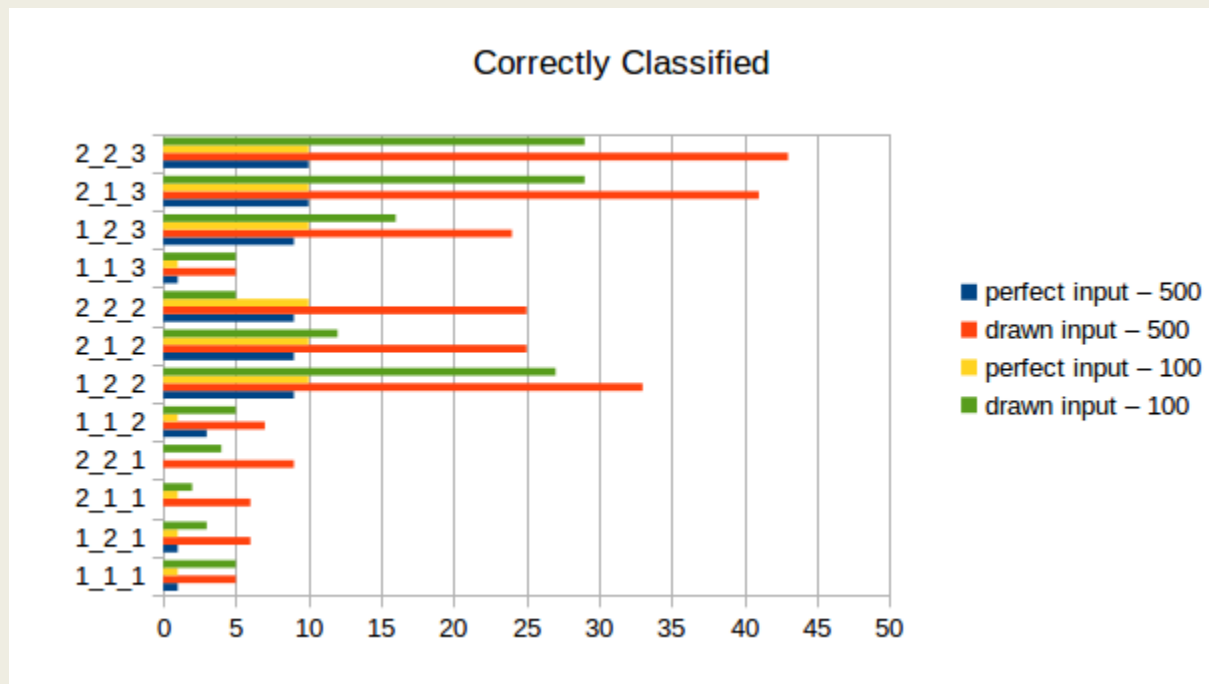
Função de activação e, função de aprendizagem do peso e da propensão

1. Sigmoidal com Gradient Descent
2. Linear com Gradient Descent
3. Hard-Limit com Perceptrão

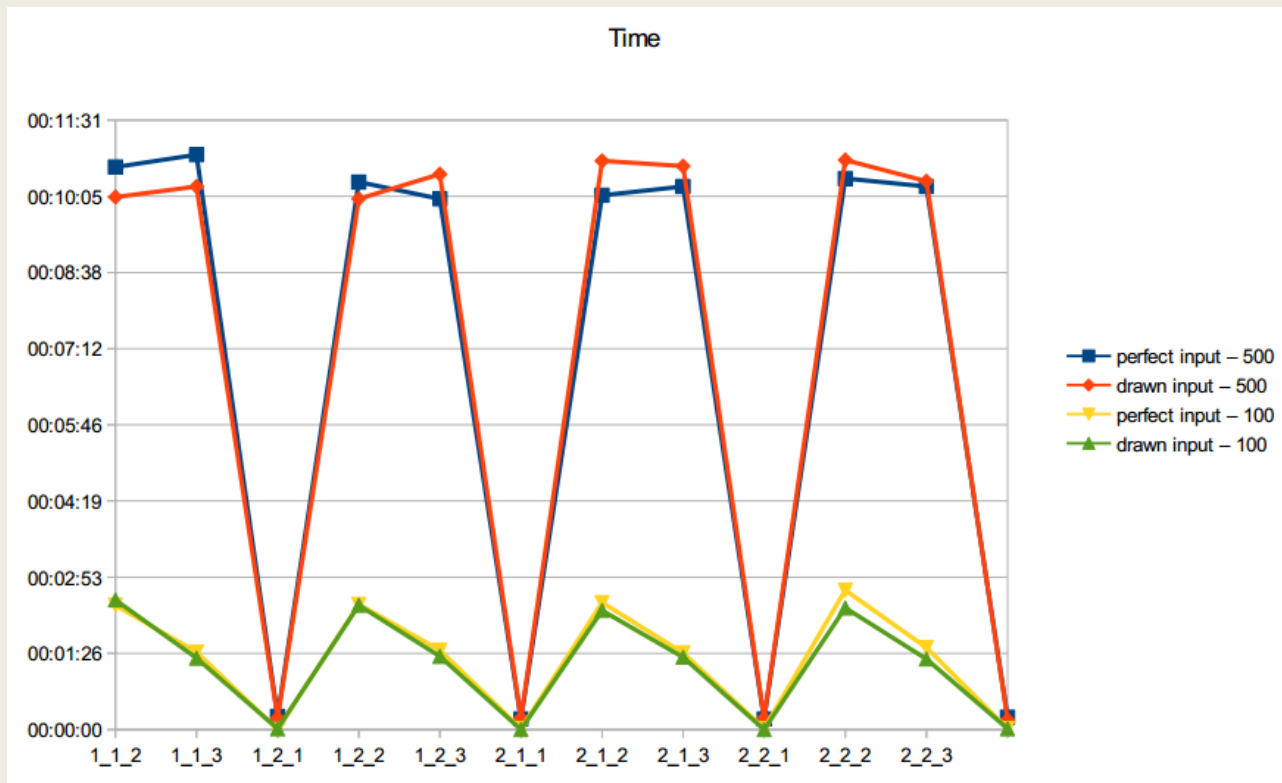
Resultados

Caso de teste	<i>perfect input</i> – 500	<i>drawn input</i> – 500	<i>perfect input</i> – 100	<i>drawn input</i> – 100
Com memória,transposta, sigmoide com gradient descent	1	5	1	5
Com memória,transposta, linear com gradient descent	3	7	1	5
Com memória,transposta, hard-limit com perceptron	1	5	1	5
Com memória,regra de Hebb, sigmoide com gradient descent	1	6	1	3
Com memória,regra de Hebb, linear com gradient descent	9	33	10	27
Com memória,regra de Hebb, hard-limit com perceptron	9	24	10	16
Sem memória, sigmoide com gradient descent	0	6	1	2
Sem memória,linear com gradient descent	9	25	10	12
Sem memória, hard-limit com perceptron	10	41	10	29
Sem memória, sigmoide com gradient descent	0	9	0	4
Sem memória,linear com gradient descent	9	25	10	5
Sem memória, hard-limit com perceptron	10	43	10	29

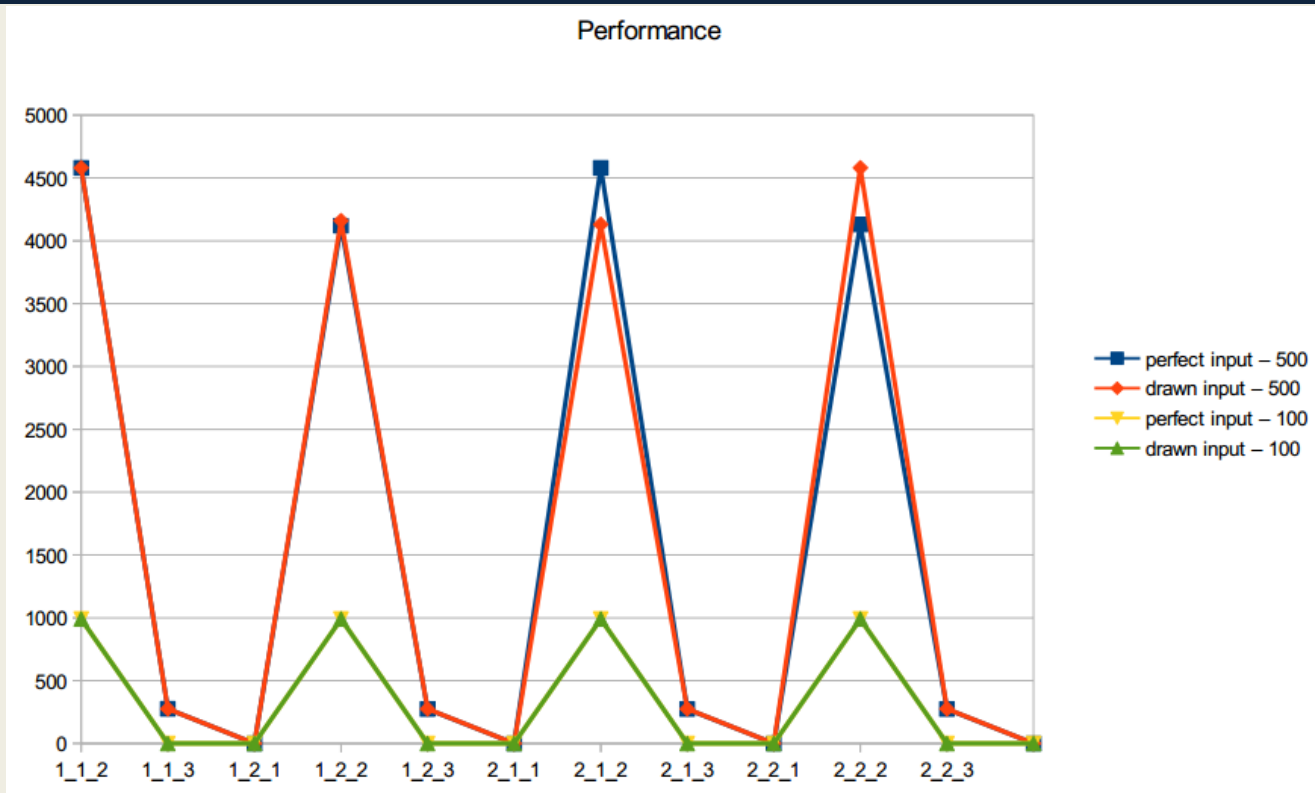
Resultados



Tempo de treino



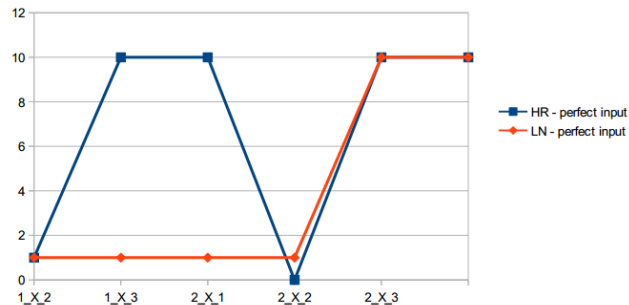
Performance



Hebb VS Pesagem Transposta

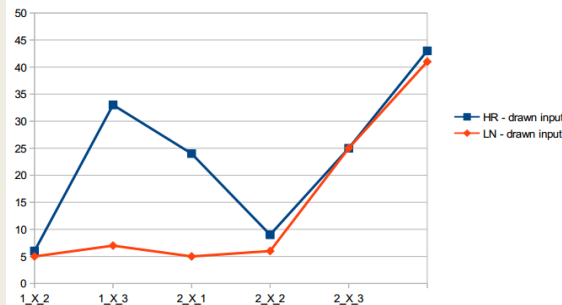
Perfect Input with 100 training cases

Hebb's Rule vs Linear Neuron



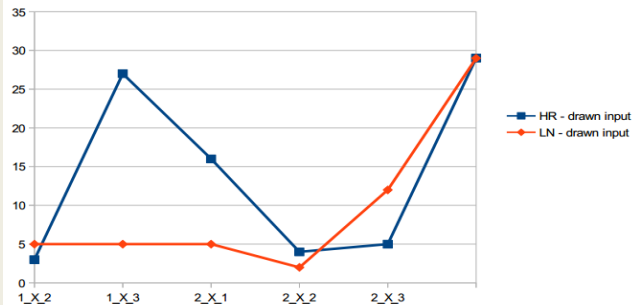
Drawn Input with 500 training cases

Hebb's Rule vs Linear Neuron



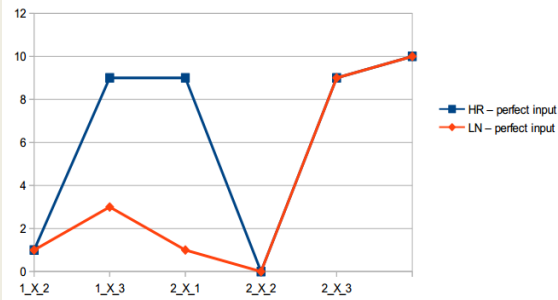
Drawn Input with 100 training cases

Hebb's Rule vs Linear Neuron



Perfect Input with 500 training cases

Hebb's Rule vs Linear Neuron



Análise dos resultados

Quando é utilizada memória associativa, a regra de Hebb apresenta resultados substancialmente melhores na maior parte dos casos, sendo equiparável a um Neurónio Linear nos restantes.

No geral, com memória associativa, a função de activação Linear com Gradient Descent é a que apresenta os melhores resultados. Já na ausência de uma memória associativa é a Hard-Limit com Perceptrão que tem melhor prestação.

De acordo com os resultados, podemos concluir que o classificador, com os inputs originais, tem mais sucesso que a inclusão do pré-processamento realizado pela memória associativa. (Porquê? Poucos inputs? N° de camadas reduzido? Ruído ou qualidade dos inputs?)

Demo

Antes de mais, um voluntário?

Vamos testar para o caso que apresentou melhores resultados:
Rede neuronal que **não usa memória associativa**, com função de activação **Hard-Limit** e, função de aprendizagem do peso e da propensão com **Perceptrão**.

Análise dos resultados da demo e explicação do código utilizado.

Fim

Questões?