Relatório -Trabalho Prático 2

OCR – Optical Character Recognition

*Realizado por:*

2009109673 – Daniel Frutuoso

2009111924 – Igor Cruz

Grupo 4

Faculdade de Ciências e Tecnologias da UC

Departamento de Engenharia Informática

Coimbra, 09 de Outubro de 2012

**Introdução**

É-nos proposto, neste trabalho, desenvolver uma aplicação Optical Character Recognition que identifique os números ( 1, 2, 3, 4, 5, 6, 7, 8, 9, 0) recorrendo a duas redes neuronais:

* Classificador
* Classificador + Memória Associativa

Para além disto, temos também que testar as redes com diferentes conjuntos de funções de transferência e funções de aprendizagem e descobrir qual a melhor combinação. Mais à frente serão apresentados os resultados.

**Comparador**

**Memória Associativa + Comparador**

**Resultados**

**Questões**

**Como é que *data set* influencia a performance do sistema classificador?**

Para um desempenho bom do sistema classificador, o *data set* deve ter um grande número de casos de treino e esses casos devem ser diversificados, possibilitando que a rede neuronal faça generalizações e seja assim capaz de classificar diferentes tipos de caracteres desenhados manualmente. É importante variar o tipo de escrita e o tamanho dos caracteres.

Em termos de treino é importante variar o tamanho e tipo de caligrafia facilitando a generalização. No entanto se o treino tiver um número elevado de casos em que a caligrafia é a mesma, bem como os tamanhos dos caracteres, então a aprendizagem é mais específica e mais perspicaz para essa caligrafia.

**Qual a estrutura que demonstra melhores resultados: apenas o classificador ou o classificador+memória associativa?**

Após implementação e teste de resultados a implementação que demonstrou maior taxa de sucesso foi a utilização de apenas um classificador para identificar os algarismos.

**Qual a melhor função de activação: *hardlim*, *linear* or *logsig*?**

Para a activação da rede neuronal responsável pela memória associativa a função que funciona melhor é a *hardlim*, que possibilita a eliminação de ruído de caracteres que não correspondem ao caracter perfeito.

Já para o classificador, após diversos teste, constatamos que a melhor função é a *logsig*, no entanto a *hardlim* também se revelou eficiente.

**O desempenho da regra de Hebb é bom?**

Não. A regra de Hebb, aplicada à memória associativa revela-se extremamente ineficiente. Isto ocorre pois os padrões de *input* e treino não são ortogonais levando à ocorrência de interferência e sobreposição entre as associações e consecutivo ruído nas saídas da memória associativa.

**O sistema classificador é capaz de atingir os objectivos principais (classificação de dígitos)?**

Sim , sem sobra de dúvidas. O nosso classificador, apresentou taxas de sucesso na ordem dos 95%, quando treinado com 350 casos de treino.

**Qual a percentagem de dígitos bem classificados?**

Tal como dito acima, a taxa de sucesso de dígitos bem classificadores recorrendo apenas ao classificador foi de cerca de 95%.

Quanto à arquitectura que recorre à memória associativa e ao classificador, a taxa de sucesso foi de cerca de 70%.

**Como é a capacidade de generalização? O sistema classificador é suficientemente robusto (para dar saídas correctas quando as entradas não são perfeitas)? Qual a percentagem de novos dígitos bem classificados?**

O sistema classificador é capaz de generalizar, no entanto os *inputs* a classificar devem ser razoavelmente legíveis. No caso de se inventarem novos tipos de letra e representações estranhas para os dígitos a percentagem de números bem classificados cai para os 40%.

**Conclusão**