



- UNIVERSIDADE DO VALE DO RIO DOS SINOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS – Centro 6 – Curso: Informática

REDES NEURAIS ARTIFICIAIS

Disciplina: Redes Neurais Artificiais

Professor responsável: *Fernando Santos Osório*

Semestre: 2003/2

Horário: 33

E-mail: *osorio@exatas.unisinos.br*

Web:

<http://www.inf.unisinos.br/~osorio/neural.html>

TRABALHO PRÁTICO 2003/2 – GRAU B

Especificado em 28.10.2003 – Versão 1.0

Faça um programa para simulação e aprendizado de uma rede neural MLP com Back-Propagation que funcione de acordo com a especificação fornecida logo abaixo.

- Especificação básica da rede: rede com três camadas do tipo X-Y-1: X entradas, Y neurônios na camada intermediária, 1 única saída. A rede é do tipo totalmente conectada (*fully connected*), ou seja, os X neurônios da entrada se conectam com todos os neurônios da camada intermediária, e estes por sua vez se conectam com o único neurônio da camada de saída. A rede **não** precisa ter atalhos (conexões tipo *short-cut* entre camadas). Os números de neurônios da camada de entrada e da camada intermediária, a serem configurados pelo usuário, devem ficar no intervalo: $0 \leq X, Y \leq 50$ (intervalo mínimo!).
- Em relação ao algoritmo de aprendizado: implementar o Back-Propagation padrão c/momentum, onde os parâmetros de taxa de aprendizado e de momentum precisam ser implementados. O aprendizado pode ser do tipo por épocas (default) ou por exemplos (implementação opcional). A função de transferência usada pode ser tanto a sigmoide convencional (0 à 1), como a tangente hiperbólica (-1 a 1), você pode optar pela que melhor lhe convier (implementação de ambas é opcional). O sigmoïd prime offset (deslocamento da sigmoide) deve ser implementado.
- Ler um arquivo texto (*.drn – definição da rede neural) com a configuração dos parâmetros referentes a simulação:
 - Número de neurônios na camada de entrada (NE) : de 1 à 50.
 - Número de neurônios na camada intermediária (NI): de 0 à 50.
 - Nome do arquivo contendo os dados de aprendizado (DA - arquivo do tipo *.apr).
 - Numero de dados (exemplos) contidos no arquivo de dados de aprendizado (NA).
 - Nome do arquivo contendo os dados de validação (DV - arquivo do tipo *.val).
(a validação é executada a cada nova época de treinamento, calculando o erro)
 - Número de dados (exemplos) contidos no arquivo de dados de validação (NV).
 - Taxa de aprendizado (LR – Learning Rate): valor real entre 0.0 e 1.0.
 - Inércia (MU – Momentum): valor real entre 0.0 e 1.0.
 - Deslocamento da Sigmoïde (SO – Sigmoid Prime Offset): valor real entre 0.0 e 0.5
 - Número máximo de épocas de aprendizado a serem simuladas (MX).
 - Erro máximo aceito em cada saída da rede (ST - Score Threshold): valor real entre 0.0 e 0.5.
 - Valor da semente de números aleatórios (SA - se for zero, usar o relógio como semente).

Exemplo de arquivo de configuração: xor.drn

```
$NE 2
$NI 2
$DA xor.apr
$NA 4
$DV xor.val
$NV 4
$LR 0.05
$MU 0.8
$SO 0.1
$MX 3000
$ST 0.1
$SA 0
```

- Os arquivos de dados são uma coleção de X+1 números reais por linha em formato texto, indicando as X entradas, seguidas da saída desejada. Exemplo: (xor.apr / xor.val)
0.0 0.0 0.0
1.0 0.0 1.0
0.0 1.0 1.0
1.0 1.0 0.0
- A simulação deve resultar em 3 arquivos gravados em disco:
 - Arquivo texto contendo a evolução do erro (sugestão: SSE – Sum of Squared Errors, ou MSE ou NMSE, ou ...) da base de aprendizado. Gravar um valor de erro (uma época) por linha do arquivo texto, salvando com o nome: erro-apr.txt.
 - Arquivo texto contendo a evolução do erro da base de validação (arquivo: erro-val.txt).
 - Ao final da simulação:
 - Gerar um arquivo texto contendo a resposta da rede para a ativação dos exemplos da base de validação. Cada exemplo deve gerar uma saída (linha) gravada no arquivo resp.txt.
 - Exibir na tela o percentual de acerto sobre a base de validação (respostas corretas), considerando o erro máximo aceito, definido no arquivo de parâmetros.
- O critério de parada da simulação é quando o simulador atingir o número máximo de épocas que foi especificado no arquivo de parâmetros.
- Não é necessário salvar os pesos finais da rede ou carregar uma rede previamente treinada. Estas funcionalidades do simulador são opcionais.
- A linguagem / ambiente de desenvolvimento a ser adotado na implementação é livre.

OBSERVAÇÕES FINAIS:

- O programa deve ser entregue até o dia da prova do grau B. O aluno que entregar o trabalho no dia do grau B, terá seu trabalho avaliado e poderá entregar uma nova versão no dia do grau C, corrigindo os problemas da versão inicial.
- O trabalho não é recuperado na prova do Grau C, sendo que o aluno que não entregar o trabalho no Grau B deverá entregá-lo no máximo até a data do Grau C.
- O trabalho prático vale 3.0 pontos na nota do Grau B
- **Entregar o programa fonte do simulador e também o executável zipados por e-mail** para o professor, **juntamente com uma pequena documentação** explicando como se usa este programa, além de um conjunto de arquivos simples (pode ser o problema do Xor) que possam ser usados para testar o simulador.
- **O trabalho é individual.**

BOM TRABALHO!