

 **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:** 2004/2

**Horário:** 43

**E-mail:** [osorio@exatas.unisinos.br](mailto:osorio@exatas.unisinos.br)

**Web:**

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

**TRABALHO PRÁTICO 2004/2 – GRAU B**

*Especificado em 03.11.2004 – 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 serão de 2 ou 3 (você pode optar entre 2 ou 3 entradas, indicando na documentação do programa) e da camada intermediária o número de neurônios será configurado pelo usuário, devendo ficar no intervalo:  $1 \leq Y \leq 30$  (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 ou por exemplos (opção do aluno: indicar na documentação). 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 sigmoid prime offset (deslocamento da sigmoide) deve ser implementado.
- Ler um arquivo texto (\*.cfg – configuração da rede neural) com a configuração dos parâmetros referentes a simulação:
  - Número de neurônios na camada intermediária (NI): de 0 à 30.
  - Nome do arquivo contendo os dados de aprendizado (DA - arquivo do tipo \*.da).
  - Numero de dados (exemplos) contidos no arquivo de dados de aprendizado (NA).
  - Nome do arquivo contendo os dados de validação (DV - arquivo do tipo \*.dv).  
(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 Sigmoide (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 na saída da rede (ST - Score Threshold), usado para contar o nro. de acertos e de erros: valor real entre 0.0 e 0.5.
  - Valor da semente de números aleatórios (SA - se for zero, usar o relógio como semente).
  - Intervalo entre “relatórios” (Log) salvos em disco (NE – Nro. de épocas): 1 a MX.
  - Gerar (=1) ou não (=0) arquivo de pesos (SP – Salvar pesos): 0 ou 1

Exemplo de arquivo de configuração: xor.drn

```
$NI 2
$DA xor.da
$NA 4
$DV xor.dv
$NV 4
$LR 0.05
$MU 0.8
$SO 0.1
$MX 3000
$ST 0.1
$SA 0
$NE 1
$SP 1
```

- Os arquivos de dados são uma coleção de  $X+1$  números reais por linha em formato texto, indicando as  $X$  entradas da rede ( $X = 2$  ou  $3$  entradas), seguidas da saída desejada (1 único valor). Exemplo: ( xor.da / xor.dv )
 

```
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 4 arquivos gravados em disco:
  - Arquivo texto contendo a evolução do erro (sugestão: SSE – Sum of Squared Errors, ou MSE ou NMSE, ou ... indicar na documentação o tipo de medida de erro) da base de aprendizado. Gravar um valor de erro (para cada NE épocas, onde se NE for igual a 1, teremos o erro salvo em cada uma das épocas simuladas), gravando um valor por linha do arquivo texto, e salvando com o nome: erro-apr.txt.
  - Arquivo texto contendo a evolução do erro da base de validação (arquivo: erro-val.txt).
  - Arquivo texto contendo a evolução dos pesos da rede a cada NE épocas do aprendizado, onde cada neurônio terá todos os seus pesos salvos em uma linha do arquivo, separados por espaço em branco (para redes com 2 entradas, serão conjuntos de 3 valores – 2 pesos e bias, para redes de 3 entradas, serão salvos 4 valores – 3 pesos e bias por último). Serão salvos  $NI+1$  linhas, contendo os pesos dos  $NI$  neurônios, mais uma linha contendo os pesos do neurônio de saída (separados por espaço) e seu bias. Após salvar as  $NI+1$  linhas com os pesos, pular uma linha em branco no arquivo e listar novamente os pesos após NE épocas. Prosseguir com este processo até o final da simulação. Salvar este arquivo com o nome: pesos.txt, somente se SP for igual a 1 (se  $SP = 0$  este arquivo não deve ser gerado).
  - 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 (ST), 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.
- 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 (substituição da nota do trabalho prático).
- 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 4.0 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!