

LAB4 – Laboratório 4 de Inteligência Artificial
Assunto: Perceptron e Comitês de Perceptrons

I - Observações:

- 1. O trabalho é em Dupla:** É permitido discutir com outras duplas os problemas e as estratégias para solucioná-los, mas as implementações das soluções, bem como as resoluções dos exercícios propostos devem ser feitas somente pelos componentes da dupla. Trabalhos plagiados da Internet ou de outras duplas não serão corrigidos e receberão nota 0 (zero).
- 2. O laboratório deve ser feito durante o período de aula e apresentado para o professor ao fim da aula.**

II – Tarefas

1. Crie um projeto no Eclipse e adicione o arquivo *jar* disponível no SINEF compactado no arquivo LAB4_recursos.zip. Para isto, utilize os procedimentos indicados no laboratório 3. Execute e estude os exemplos que estão dentro do *package ia.examples*. Atenção: Para o exemplo da XOR, utilizou-se o valor *-1.0* no lugar de 0. Para este exemplo é preciso, também fornecer o cominho para o arquivo *xor.data* fornecido com os recursos do Lab4.
2. O arquivo *heart.data* fornecido no arquivo ZIP do laboratório contém as informações de uma base de dados de doenças cardíacas. Trata-se de um arquivo do tipo CSV (separado por vírgulas) com 14 colunas. As 13 primeiras são as entradas, com dados diversos (descrição no arquivo *heart-disease.description*), enquanto a coluna 14 é o valor do grau de comprometimento cardíaco. O valor da coluna 14 varia de 0 a 4, sendo que 0 é a ausência de comprometimento cardíaco.
 - a) Use a classe *CsvDBRead*, fornecida no arquivo *jar* para ler este arquivo, da mesma forma realizada no exemplo *PerceptronEnsembleXOR* para ler o arquivo *xor.data*;
 - b) Depois de lido o arquivo, você terá acesso aos dados por meio das funções *getInputs()* e *getOutputs()*. Crie uma variável *inputs[][]*, do tipo *double*, e uma variável *outputs[]*, também do tipo *double*. Use as funções *getInputs* e *getOutputs* para preencher estas variáveis. Modifique o conteúdo do array *outputs[]*, trocando todos os valores 0 para -1, e todos os valores entre 1 e 4 para 1. Neste caso, treinaremos a rede somente para identificar a presença ou ausência de comprometimento cardíaco;
 - c) Crie um perceptron simples, da classe *Perceptron*, e treine com os valores lidos do arquivo, utilizando as variáveis *inputs* e *outputs*. Execute o treinamento 10 vezes, em cada uma das vezes, anote o número de erros obtidos para o conjunto de treinamento (retorno da função *train*). Calcule a média de erros e verifique a porcentagem de erros em relação ao tamanho do conjunto de treinamento;
 - d) Crie um perceptron ensemble, da classe *PerceptronEnsemble*, e treine com os valores lidos do arquivo, utilizando as variáveis *inputs* e *outputs*. Execute o treinamento 10 vezes, em cada uma das vezes, anote o número de erros obtidos para o conjunto de treinamento (retorno da função *train*). Calcule a média de erros e verifique a porcentagem de erros em relação ao tamanho do conjunto de treinamento. Cuidado, pois o treinamento do modelo ensemble é mais demorado do que o do perceptron simples. Comece com poucos nós e um baixo valor de iterações máximas, e vá aumentando aos poucos para verificar o comportamento do treinamento.

BOM TRABALHO!