

PROGRAMA DE PÓS-GRADUAÇÃO DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO
ASSOCIAÇÃO UFMA/UFPI
Disciplina: Redes Neurais (DCC40)
Professor: Areolino de Almeida Neto
Aluno: Raimundo Vale

Trabalho sobre Reconhecimento de padrões via RNA tipo MLP

Atividades do trabalho

01. Projetar os dados de entrada para a rede, realizando um pré-processamento ou não, além de dividir a base em dados de treino, validação e teste;

R → A base de dados indicada foi lida do arquivo indicado ("car.data") convertendo os dados da seguinte forma:

Valores dos atributos:

buying → v-high = 3, high = 2, med = 1, low = 0

maint → v-high = 3, high = 2, med = 1, low = 0

doors → 2 = 0, 3 = 1, 4 = 2, 5-more = 3

persons → 2 = 0, 4 = 1, more = 2

lug_boot → small = 0, med = 1, big = 2

safety → low = 0, med = 1, high = 3

E a classificação dos carros será da lida da seguinte forma:

unacc → 0

acc → 1

good → 2

v-good → 3

02. Definir como será a saída da rede, de modo a apresentar corretamente a classe;

R → A saída da rede será o vetor Y ao final de cada tratamento das entradas em que o erro quadrático seja menor que o erro máximo admitido (0,005)

03. Construir a RNA, definindo sua estrutura e pesos iniciais;

R → A RNA possui 6 neurônios na camada de entrada, 6 neurônios na camada escondida e 1 neurônio na cada de saída.

04. Por a rede em treinamento, ajustando a configuração da rede, conforme explanação posterior;

R → A rede foi treinada e a saída registrada no arquivo saida.mat

05. Treinar a rede sempre em uma quantidade fixa de épocas com um programa de treino de confecção própria;

R → Feito somente o treinamento inicial

06. Fazer a validação do treinamento simultaneamente;

R → Não consegui fazer uma visualização adequada da validação

07. Testar a generalização do aprendizado, com dados apropriados, no melhor treinamento obtido, salvando os valores dos pesos desse treinamento.

R → Pesos finais salvos porém não avaliados.

Desta forma, serão realizados treinamentos para cada caso a seguir:

T1. 10 treinamentos reinicializando os pesos aleatoriamente a cada treinamento, sem alterar nada mais na configuração da rede, neste caso não será necessário tirar a média dos erros em cada época; Nos treinamentos não consegui gerar gráficos que corroborassem o aprendizado, restando a saída em no arquivo saídaT1.mat.

Neste treinamento os valores atribuídos foram:

$\eta = 0,5$;

Erro máximo (E_{\max}) = 0,005;

Função de ativação: linear ($k = 1$);

Todos os bias iguais a zero;

Todos os pesos iniciais: aleatórios;

Quantidade de dados em cada treinamento: Aleatório entre 1 e 100.

Gráficos: Não consegui gerar gráficos que validassem o aprendizado.

Arquivo MathLab: bpmlpRvaleT1.m

Arquivo saída resumido: saídaTreinos1.mat

Arquivo saída analítico (com os pesos das camadas): saídaT1.mat

T2. 10 treinamentos reinicializando os pesos aleatoriamente a cada treinamento e variando a função de ativação (sigmoide e tanh) dos neurônios escondidos a cada 10 treinamentos, perfazendo um total de 20 treinamentos, sem alterar nada mais na configuração da rede;

Treinamento com sigmoide os valores atribuídos foram:

$\eta = 0,5$;

Erro máximo (E_{\max}) = 0,005;

Função de ativação: sigmoide ;

Todos os bias iguais a zero;

Todos os pesos iniciais: aleatórios;

Quantidade de dados em cada treinamento: Aleatório entre 1 e 25

Gráficos: Não consegui gerar gráficos que validassem o aprendizado.

Arquivo MathLab: bpmlpRvaleT21.m

Arquivo saída resumido: saídaTreinos21.mat

Arquivo saída analítico (com os pesos das camadas): saídaT21.mat

Aprendizado falhou no décimo treinamento quando não foi possível diminuir o erro.

Treinamento com tangente hiperbólica os valores atribuídos foram:

$\eta = 0,5$;

Erro máximo (E_{\max}) = 0,005;

Função de ativação: tangente hiperbólica;

Todos os bias iguais a zero;

Todos os pesos iniciais: aleatórios;

Quantidade de dados em cada treinamento: Aleatório entre 1 e 25

Gráficos: Não consegui gerar gráficos que validassem o aprendizado.

Arquivo MathLab: bpmlpRvaleT22.m

Arquivo saída resumido: saídaTreinos22.mat

Arquivo saída analítico (com os pesos das camadas): saídaT22.mat

Aprendizado falhou no décimo treinamento quando não foi possível diminuir o erro.

T3. 10 treinamentos reinicializando os pesos aleatoriamente a cada treinamento e variando a quantidade de neurônios escondidos a cada 10 treinamentos, em um total de 10 variações nessa quantidade, perfazendo um total de 100 treinamentos, sem alterar nada mais na configuração da rede;

Treinamento com variação aleatória da quantidade de neurônios da camada escondida:
Aprendizado não ocorreu pois não consegui como ajustar o algoritmo para quantidade aleatória de neurônios na camada escondida.
Arquivo MathLab: bpmlpRvaleT30.m

T4. 10 treinamentos reiniciando os pesos aleatoriamente a cada treinamento e variando a taxa de aprendizado a cada 10 treinamentos, em um total de cinco variações nessa taxa perfazendo um total de 50 treinamentos, sem alterar nada mais na configuração da rede.

Neste treinamento os valores atribuídos foram:
Taxa de aprendizagem (eta): Aleatória entre 0,001 e 1;
Erro máximo (E_{max}) = 0,005;
Função de ativação: linear ($k = 1$);
Todos os bias iguais a zero;
Todos os pesos iniciais: aleatórios;
Quantidade de dados em cada treinamento: Aleatório entre 1 e 50.
Gráficos: Não consegui gerar gráficos que validassem o aprendizado.
Arquivo MathLab: bpmlpRvaleT40.m
Arquivo saída resumido: saidaTreinos41.mat até o saidaTreinos45.mat
Arquivo saída analítico (com os pesos das camadas): saidaT41.mat até o saidaT45.mat

Conclusão:

O algoritmo foi implementado porém como não consegui gerar os gráficos. A falta dos gráficos tornam os resultados duvidosos ao ponto de não poder confirmar se houve aprendizado nem se o algoritmo estava corretamente implementado. A total incompletude da fase T3 é forte indício de que o algoritmo contém erros ao ponto de não ter sido possível executar nenhum treinamento.