



Universidade do Porto
Faculdade de Engenharia
FEUP

Redes Neurais para a predição de bancarrota em companhias polacas
Relatório Intercalar

Representantes:

Luís Leão Aguiar Braga da Cruz -up201303248@fe.up.pt
Ricardo Filipe Amaro Saleiro Abreu -up201304450@fe.up.pt
Tiago André Pérola Filipe - up201610655@fe.up.pt

2 Abril de 2017

Inteligência Artificial
3º ano do Mestrado Integrado em Engenharia Informática e Computação

Objectivo:

O objetivo deste trabalho consiste na aplicação de Redes Neurais artificiais na predição de bancarrota em companhias polacas.

Descrição:

Especificação:

O programa treina apropriadamente uma Rede Neuronal Artificial, usando o algoritmo "Back-Propagation". O modelo obtido deve poder depois ser utilizado no diagnóstico de novos casos.

O dataset utilizado divide-se em cinco ficheiros de texto, cada um representando um ano fiscal das empresas polacas.

Cada linha do ficheiro de texto representa uma empresa, contendo cada uma destas, 64 atributos. Atributos estes que são representados por valores reais.

O ficheiro de texto do primeiro ano fiscal contém taxas financeiras do 1º ano do período de previsão da classe correspondente que indica o status de falência após 5 anos. Este ficheiro contém 7027 instâncias, ou seja empresas, das quais 271 representam empresas falidas, 6756 empresas que não faliram no período de previsão.

O ficheiro de texto do segundo ano fiscal contém as taxas financeiras a partir do 2º ano do período de previsão da classe correspondente que indica o status de falência após 4 anos. Os dados contêm 10173 instâncias, 400 das quais representam empresas falidas e 9773 empresas que não faliram no período de previsão.

O ficheiro de texto do terceiro ano fiscal contém as taxas financeiras a partir do 3º ano do período de previsão da classe correspondente que indica o status de falência após 3 anos. Os dados contêm 10503 instâncias, 495 das quais representam empresas falidas, 1.0008 empresas que não faliram no período de previsão.

O ficheiro de texto do quarto ano fiscal contém as taxas financeiras a partir de 4º ano do período de previsão da classe correspondente que indica o status de

falência após 2 anos. Os dados contêm 9792 instâncias, 515 das quais representam empresas falidas, 9277 empresas que não faliram no período de previsão.

O ficheiro de texto do quinto ano fiscal contém as taxas financeiras a partir do 5º ano do período de previsão da classe correspondente que indica o status de falência após 1 ano. Os dados contêm 5910 instâncias, 410 das quais representam empresas falidas, 5500 empresas que não faliram no período de previsão.

Este *data set* caracteriza-se ainda pela falta de alguns atributos de empresas. Estes valores em falta estão representados por “?”.

Para tratar da falta de valores decidiu-se gerar um valor aleatório entre -1 e 1 e substituir aquando da leitura dos valores para uma matriz bidimensional.

Normalização:

Para a normalização dos dados foi decidido fazer uma normalização Gaussiana. Este algoritmo vai tornar o processo de aprendizagem e treino mais rápido.

Modelo de aprendizagem:

O modelo de aprendizagem a aplicar é o que utiliza *Back-propagation*. Durante o processo de treino dividimos os dados em 80% para treino e 20% para teste.

Percorremos os dados para aprendizagem de maneira aleatória, utilizando o algoritmo de Fisher-Yates.

Nesta fase inicial estamos a utilizar um ritmo de aprendizagem de 0.05 visto que de modo geral é um bom ritmo de aprendizagem, e um número máximo de 1000 ciclos de treino.

Estes valores podem vir a ser alterados posteriormente para melhorar a precisão dos resultados esperados.

Arquitetura da rede:

A rede neuronal que deve prever bancarrota de empresas polacas é constituída por três camadas: a de inputs, a camada “escondida” e a camada de output, que determina se a empresa irá falir ou não.

A camada de entrada tem cerca de 64 preceptores que representam os atributos de cada uma das empresas.

Os pesos dados a esses preceptores são gerados por um número aleatório entre -0.01 e +0.01.

Para a camada “escondida” decidimos manipular o número de preceptores para testar os seus efeitos nos resultados. Assim sendo, temos três cenários:

- Uma camada escondida com 2 preceptores. Trata-se de um número muito pequeno, que origina resultados pouco precisos da fase de teste.
- Uma camada escondida com 44 ou 45 preceptores. Este número é ótimo quanto ao nível de precisão da fase de teste.
- Uma camada escondida com 60 preceptores. Trata-se de um valor limite para que não haja *overffiting*.

A camada de saída tem dois preceptor com que prevêem se uma empresa vai falir ou não.

De seguida, a utilização destes *inputs* e pesos irá dar resultado para a camada de resultado (*output*).

Como funções ativação estamos a usar, do input para o hidden a tangente hiperbólica e do hidden para output a softmax.

A função de ativação utilizada entre a camada de *input* e camada escondida é a função de tangente hiperbólica. A função de ativação utilizada entre a camada escondida e a camada de *output* é a de *Softmax* que retorna um valor representativo de uma probabilidade, ou seja, de uma empresa falir ou não. Mais uma vez, estas funções de ativação podem vir a ser substituídas caso se justifique.

Como critério de paragem do processo de treino utilizamos o *Cross entropy error*.

Trabalho Efectuado:

Até ao momento foram feitos os seguintes passos:

- Tratamento dos dados a utilizar pela rede neuronal
- Normalização dos valores importados.
- Construção das três camadas e os seus preceptores.
- Algoritmo de *Back-propagation*.
- Processo de treino.
- Processo de teste.

Resultados esperados:

Como forma de testar o bom funcionamento do trabalho realizado, podemos testar os resultados retornados na camada de saída. E fazer a sua interpretação.

Conclusões:

Concluindo, a rede neuronal está construída pelo que agora temos de investigar a manipulação de variáveis de treino e os seus efeitos nos resultados devolvidos pela rede.

Recursos:

No desenvolvimento do trabalho feito até ao momento utilizámos o Visual Studio 2015 para programar na linguagem C#.

Baseamo-nos no livro *Neural Networks Using C#* de James McCaffrey. Para a realização deste trabalho.