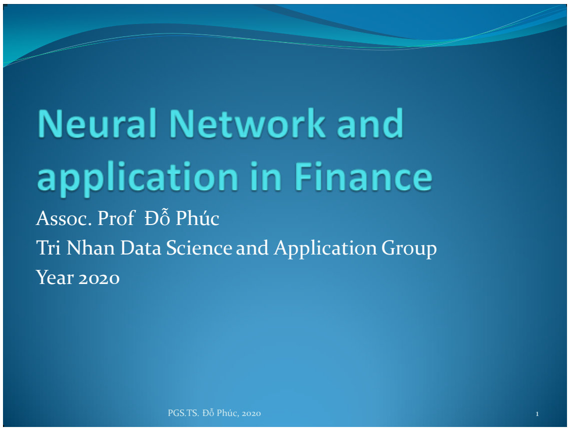
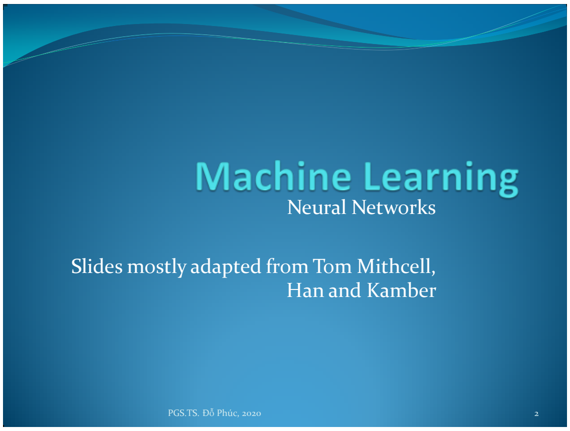


Neural Network and application in Finance

Assoc. Prof Đỗ Phúc

Tri Nhan Data Science and Application Group

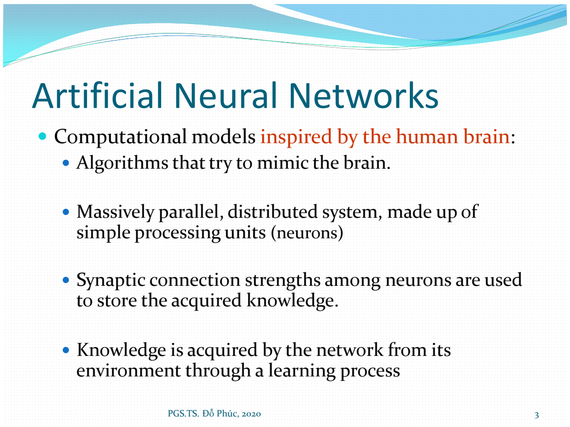
Year 2020



Machine Learning

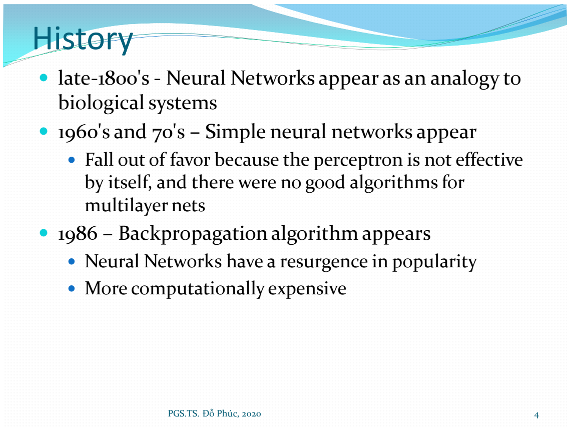
Neural Networks

Slides mostly adapted from Tom Mithcell,
Han and Kamber



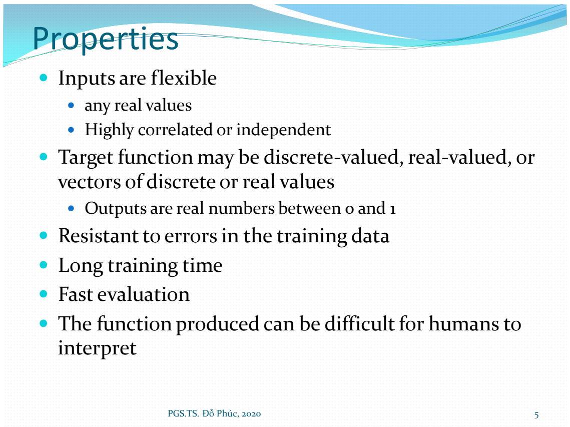
Artificial Neural Networks

- Computational models inspired by the human brain:
  - Algorithms that try to mimic the brain.
  - Massively parallel, distributed system, made up of simple processing units (neurons)
  - Synaptic connection strengths among neurons are used to store the acquired knowledge.
  - Knowledge is acquired by the network from its environment through a learning process



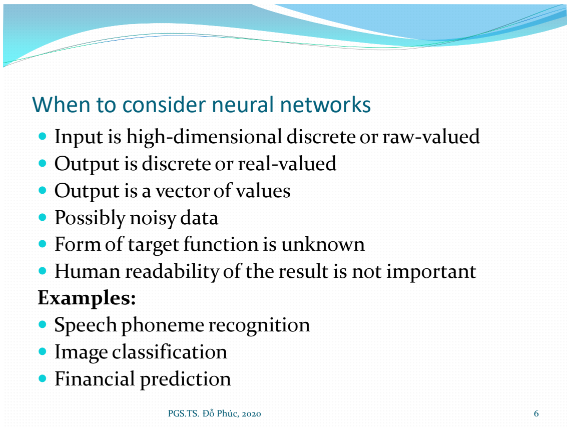
History

- late-1800's - Neural Networks appear as an analogy to biological systems
- 1960's and 70's – Simple neural networks appear
  - Fall out of favor because the perceptron is not effective by itself, and there were no good algorithms for multilayer nets
- 1986 – Backpropagation algorithm appears
  - Neural Networks have a resurgence in popularity
  - More computationally expensive



Properties

- Inputs are flexible
  - any real values
  - Highly correlated or independent
- Target function may be discrete-valued, real-valued, or vectors of discrete or real values
  - Outputs are real numbers between 0 and 1
- Resistant to errors in the training data
- Long training time
- Fast evaluation
- The function produced can be difficult for humans to interpret



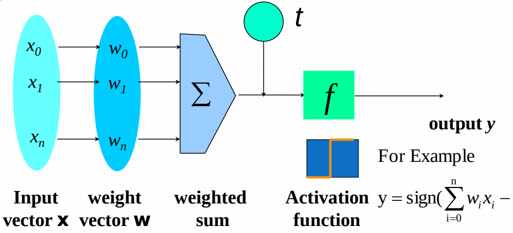
When to consider neural networks

- Input is high-dimensional discrete or raw-valued
- Output is discrete or real-valued
- Output is a vector of values
- Possibly noisy data
- Form of target function is unknown
- Human readability of the result is not important

Examples:

- Speech phoneme recognition
- Image classification
- Financial prediction

A Neuron (= a perceptron)



$y = \text{sign}(\sum_{i=0}^{n} w_i x_i - t)$

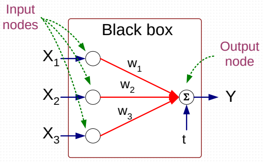
- The *n*-dimensional input vector **x** is mapped into variable y by means of the scalar product and a nonlinear function mapping

Perceptron

- Basic unit in a neural network
- Linear separator
- Parts
  - N inputs, x_1 ... x_n
  - Weights for each input, w_1 ... w_n
  - A bias input x_0 (constant) and associated weight w_0
  - Weighted sum of inputs, $y = w_0x_0 + w_1x_1 + \dots + w_nx_n$
  - A threshold function or activation function,
    - i.e 1 if y > t, -1 if y <= t

Artificial Neural Networks (ANN)

- Model is an assembly of inter-connected nodes and weighted links
- Output node sums up each of its input value according to the weights of its links
- Compare output node against some threshold t



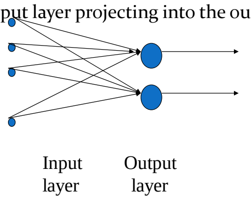
Perceptron Model

$$Y = I(\sum_i w_i x_i - t)$$ or

$$Y = sign(\sum_i w_i x_i - t)$$

Different Network Topologies

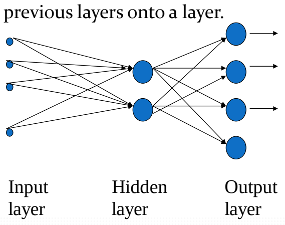
- Single layer feed-forward networks
  - Input layer projecting into the output layer



Single layer network

Different Network Topologies

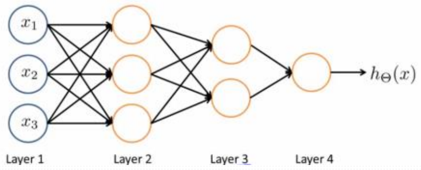
- Multi-layer feed-forward networks
  - One or more hidden layers. Input projects only from previous layers onto a layer.



2-layer or 1-hidden layer *fully connected* network

Different Network Topologies

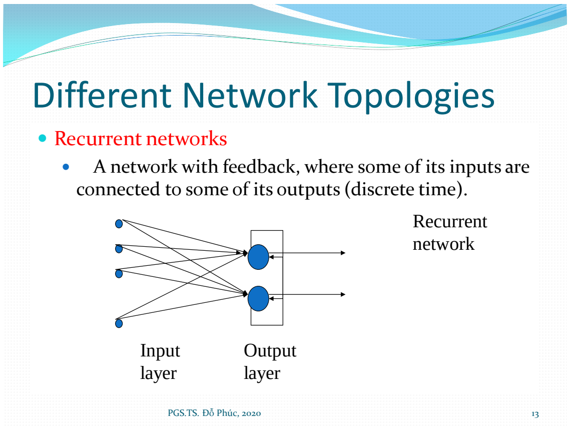
- Multi-layer feed-forward networks



Input layer

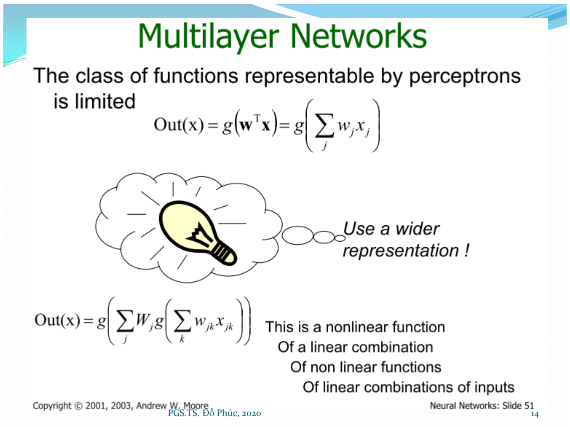
Hidden layers

Output layer



Different Network Topologies

- Recurrent networks
  - A network with feedback, where some of its inputs are connected to some of its outputs (discrete time).



Multilayer Networks

The class of functions representable by perceptrons is limited

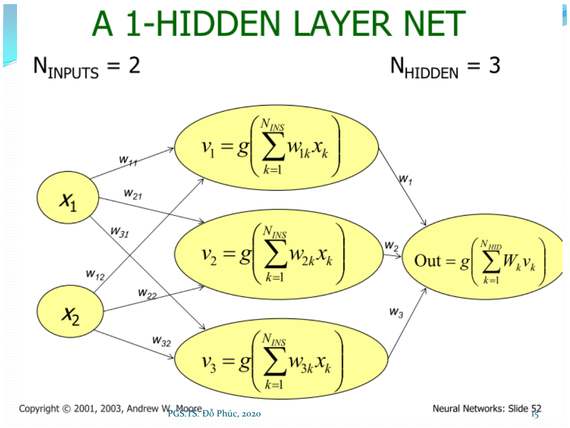
$$\text{Out}(\mathrm{x}) = g\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}\right) = g\left(\sum_j w_j x_j\right)$$

Use a wider representation !

$$\text{Out}(\mathrm{x}) = g\left(\sum_j W_j g\left(\sum_k w_{jk} x_{jk}\right)\right)$$

This is a nonlinear function
Of a linear combination
Of non linear functions
Of linear combinations of inputs

Neural Networks: Slide 51

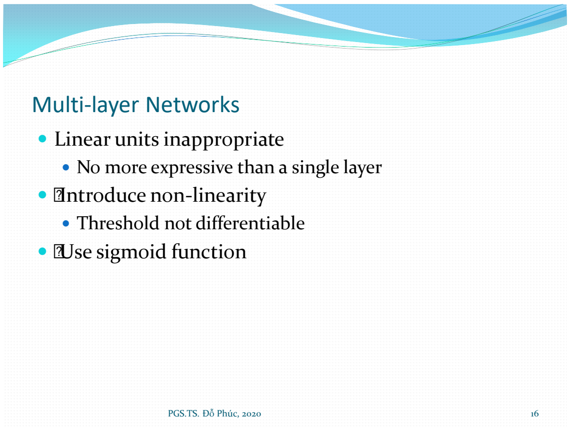


A 1-HIDDEN LAYER NET

$N_{\text{INPUTS}} = 2$ $N_{\text{HIDDEN}} = 3$

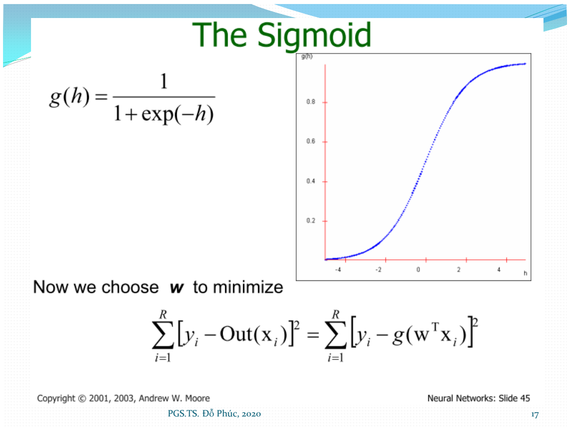
x_1
x_2
w_{11}
w_{21}
w_{31}
w_{12}
w_{22}
w_{32}
$v_1 = g\left(\sum_{k=1}^{N_{INS}} w_{1k} x_k\right)$
$v_2 = g\left(\sum_{k=1}^{N_{INS}} w_{2k} x_k\right)$
$v_3 = g\left(\sum_{k=1}^{N_{INS}} w_{3k} x_k\right)$
W_1
W_2
W_3
$\text{Out} = g\left(\sum_{k=1}^{N_{HID}} W_k v_k\right)$

Neural Networks: Slide 52



Multi-layer Networks

- Linear units inappropriate
  - No more expressive than a single layer
- Introduce non-linearity
  - Threshold not differentiable
- Use sigmoid function



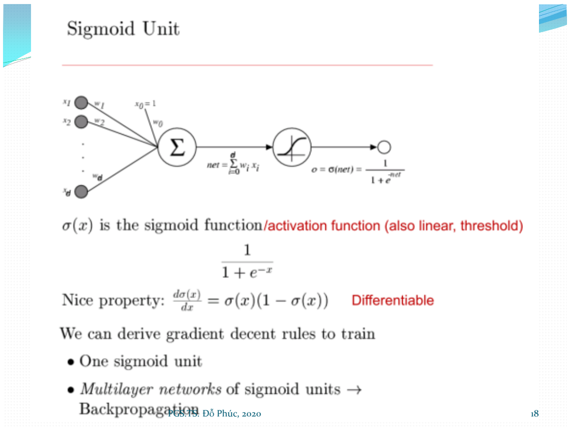
The Sigmoid

$$g(h) = \frac{1}{1+\exp(-h)}$$

Now we choose **w** to minimize

$$\sum_{i=1}^{R}\left[y_i - \text{Out}(\mathrm{x}_i)\right]^2 = \sum_{i=1}^{R}\left[y_i - g(\mathrm{w}^{\mathrm{T}}\mathrm{x}_i)\right]^2$$

Neural Networks: Slide 45



Sigmoid Unit

x_1
x_2
x_d
w_1
w_2
w_d
$x_0 = 1$
w_0
Σ
$net = \sum_{i=0}^{d} w_i x_i$
$o = \sigma(net) = \frac{1}{1+e^{-net}}$

$\sigma(x)$ is the sigmoid function/activation function (also linear, threshold)

$$\frac{1}{1+e^{-x}}$$

Nice property: $\frac{d\sigma(x)}{dx} = \sigma(x)(1-\sigma(x))$ Differentiable

We can derive gradient decent rules to train

- One sigmoid unit
- *Multilayer networks* of sigmoid units → Backpropagation

Backpropagation

$$\text{Out}(x) = g\left(\sum_j W_j g\left(\sum_k w_{jk} x_k\right)\right)$$

Find a set of weights $\{W_j\},\{w_{jk}\}$

to minimize

$$\sum_i \left(y_i - \text{Out}(x_i)\right)^2$$

by gradient descent.

That's it!
That's the backpropagation algorithm.

Backpropagation

- Iteratively process a set of training tuples & compare the network's prediction with the actual known target value
- For each training tuple, the weights are modified to **minimize the mean squared error** between the network's prediction and the actual target value
- Modifications are made in the "**backwards**" direction: from the output layer, through each hidden layer down to the first hidden layer, hence "**backpropagation**"
- Steps
    - Initialize weights (to small random #s) and biases in the network
    - Propagate the inputs forward (by applying activation function)
    - Backpropagate the error (by updating weights and biases)
    - Terminating condition (when error is very small, etc.)

How A Multi-Layer Neural Network Works?

- The **inputs** to the network correspond to the attributes measured for each training tuple
- Inputs are fed simultaneously into the units making up the **input layer**
- They are then weighted and fed simultaneously to a **hidden layer**
- The number of hidden layers is arbitrary, although usually only one
- The weighted outputs of the last hidden layer are input to units making up the **output layer**, which emits the network's prediction
- The network is **feed-forward** in that none of the weights cycles back to an input unit or to an output unit of a previous layer
- From a statistical point of view, networks perform **nonlinear regression**: Given enough hidden units and enough training samples, they can closely approximate any function

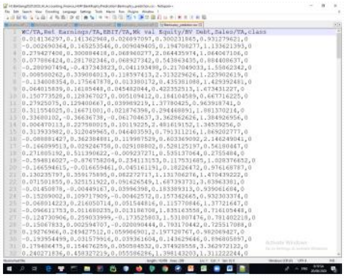
Neural Network as a Classifier

- Weakness
    - Long training time
    - Require a number of parameters typically best determined empirically, e.g., the network topology or "structure."
    - Poor interpretability: Difficult to interpret the symbolic meaning behind the learned weights and of "hidden units" in the network
- Strength
    - High tolerance to noisy data
    - Ability to classify untrained patterns
    - Well-suited for continuous-valued inputs and outputs
    - Successful on a wide array of real-world data
    - Algorithms are inherently parallel
    - Techniques have recently been developed for the extraction of rules from trained neural networks

Ứng dụng NN
Dự báo bankruptcy

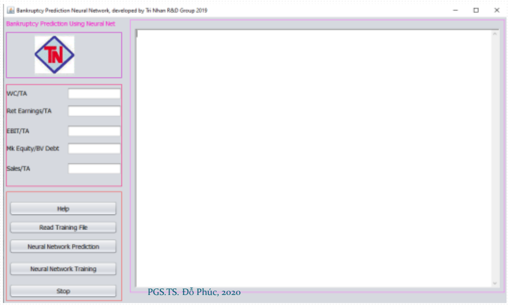
PGS.TS. Đỗ Phúc
Trí Nhân Data Science and Application Group
Year 2020

Dữ liệu: bankruptcy

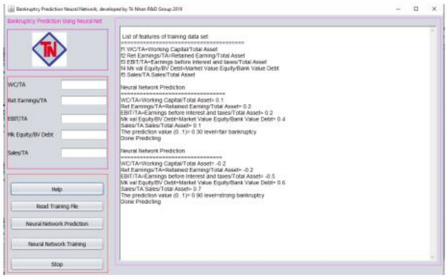


```
WC/TA,Ret Earnings/TA,EBIT/TA,Mk Val Equity/BV Debt,Sales/TA,class
0.014136297,0.141362969,0.026097097,0.300231865,0.931279621,0
-0.002690364,0.165253546,0.009049405,0.194708277,1.133621393,0
0.279427408,0.300084418,0.068960277,2.064435974,1.064047106,0
0.077886424,0.281782346,0.068927342,0.543863435,0.884408637,0
-0.280907494,-0.437343823,0.041193498,0.217049033,1.55862342,0
0.008500262,0.339084013,0.118597413,2.313229626,1.223902619,0
-0.134008354,0.175647878,0.013380172,0.435381088,1.429392481,0
0.064015839,0.16185448,0.045482044,0.422352513,1.673431227,0
0.150773528,0.128367027,0.005109412,0.184104589,0.667716225,0
0.27925075,0.129400667,0.039989219,1.37780425,0.963918741,0
0.311554025,0.16671001,0.021876398,0.294468891,1.881370214,0
0.33680102,-0.36636738,-0.062704637,3.362862626,1.384926956,0
0.006470113,0.227580015,0.10119225,2.481619152,1.34539256,0
0.313933982,0.312049965,0.046403593,0.791311216,1.869202777,0
-0.088881427,0.362384881,0.119987529,0.603369092,2.146249041,0
-0.166099513,0.029246758,0.029108802,0.528125197,0.56180447,0
0.271805192,0.511390622,-0.009237271,0.535137064,0.2755484,0
-0.594816027,-0.876758204,0.234113153,0.117531685,1.028376652,0
-0.166594615,-0.016659461,0.045161191,0.18226472,0.976168787,0
0.130235797,0.359175895,0.082272717,1.131706276,1.470439222,0
0.071501855,0.325151922,0.091626549,1.687393731,3.83963381,0
-0.01450878,-0.00449167,0.03996398,0.183389313,0.939061684,0
-0.15209002,0.109717909,-0.00462572,0.157342665,0.932303374,0
-0.068014223,0.216050714,0.051544816,0.115770846,1.37721647,0
-0.096611753,0.011680235,0.013186708,1.835163558,0.716105448,0
-0.124730906,0.259033999,-0.173525803,1.531807476,0.781402218,0
-0.15067833,0.002594707,-0.020090444,0.793170442,0.725517088,0
0.19276966,0.249427512,0.059906901,2.197720767,0.982069427,0
-0.193954499,0.031579916,0.039361604,0.143629646,0.896805897,0
0.179406475,0.154676259,0.050504532,0.374928558,3.362972122,0
0.240271836,0.458327219,0.055586286,1.398143203,1.311222244,0
```

Bankruptcy Prediction using NN

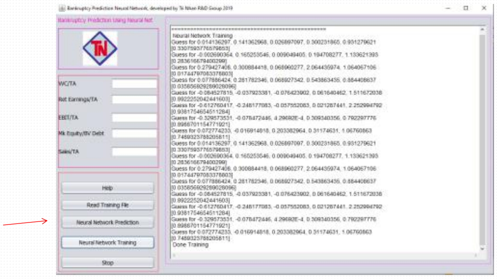

Bankruptcy Prediction Neural Network, developed by Tri Nhan R&D Group 2019
Bankruptcy Prediction Using Neural Net
WC/TA
Ret Earnings/TA
EBIT/TA
Mk Equity/BV Debt
Sales/TA
Help
Read Training File
Neural Network Prediction
Neural Network Training
Stop

Bankruptcy Prediction using NN


Bankruptcy Prediction Neural Network, developed by Tri Nhan R&D Group 2019
Bankruptcy Prediction Using Neural Net
WC/TA
Ret Earnings/TA
EBIT/TA
Mk Equity/BV Debt
Sales/TA
Help
Read Training File
Neural Network Prediction
Neural Network Training
Stop

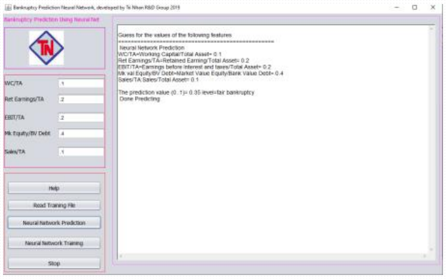
```
List of features of training data set
==========================================
f1 WC/TA=Working Capital/Total Asset
f2 Ret Earnings/TA=Retained Earning/Total Asset
f3 EBIT/TA=Earnings before interest and taxes/Total Asset
f4 Mk val Equity/BV Debt=Market Value Equity/Bank Value Debt
f5 Sales/TA Sales/Total Asset

Neural Network Prediction
==================================
WC/TA=Working Capital/Total Asset= 0.1
Ret Earnings/TA=Retained Earning/Total Asset= 0.2
EBIT/TA=Earnings before interest and taxes/Total Asset= 0.2
Mk val Equity/BV Debt=Market Value Equity/Bank Value Debt= 0.4
Sales/TA Sales/Total Asset= 0.1
The prediction value (0..1)= 0.30 level=fair bankruptcy
Done Predicting

Neural Network Prediction
==================================
WC/TA=Working Capital/Total Asset= -0.2
Ret Earnings/TA=Retained Earning/Total Asset= -0.2
EBIT/TA=Earnings before interest and taxes/Total Asset= -0.5
Mk val Equity/BV Debt=Market Value Equity/Bank Value Debt= 0.6
Sales/TA Sales/Total Asset= 0.7
The prediction value (0..1)= 0.90 level=strong bankruptcy
Done Predicting
```

Huấn luyện mạng NN


Bankruptcy Prediction Neural Network, developed by Tri Nhan R&D Group 2019
Bankruptcy Prediction Using Neural Net
WC/TA
Ret Earnings/TA
EBIT/TA
Mk Equity/BV Debt
Sales/TA
Help
Read Training File
Neural Network Prediction
Neural Network Training
Stop

```
==================================================
 Neural Network Training
Guess for 0.014136297, 0.141362968, 0.026897097, 0.300231865, 0.931279621
[0.3307593776579853]
Guess for -0.002690364, 0.165253546, 0.009049405, 0.194708277, 1.133621393
[0.2836166794002991]
Guess for 0.279427406, 0.300884418, 0.068960277, 2.064435974, 1.064067106
[0.0174479708337880З]
Guess for 0.077886424, 0.281782346, 0.068927342, 0.543863435, 0.884408637
[0.0356569292690280096]
Guess for -0.084527815, -0.037923381, -0.076423902, 0.061640462, 1.511672038
[0.8922252042441603]
Guess for -0.612760417, -0.248177083, -0.057552083, 0.021287441, 2.252994792
[0.9381754654511284]
Guess for -0.329573531, -0.078472446, 4.29692E-4, 0.309340356, 0.792297776
[0.8986701154771921]
Guess for 0.072774233, -0.016914818, 0.203382964, 0.31174631, 1.06760863
[0.7489323788205811]
Guess for 0.014136297, 0.141362968, 0.026897097, 0.300231865, 0.931279621
[0.3307593776579853]
Guess for -0.002690364, 0.165253546, 0.009049405, 0.194708277, 1.133621393
[0.2836166794002991]
Guess for 0.279427406, 0.300884418, 0.068960277, 2.064435974, 1.064067106
[0.0174479708337880З]
Guess for 0.077886424, 0.281782346, 0.068927342, 0.543863435, 0.884408637
[0.0356569292690280096]
Guess for -0.084527815, -0.037923381, -0.076423902, 0.061640462, 1.511672038
[0.8922252042441603]
Guess for -0.612760417, -0.248177083, -0.057552083, 0.021287441, 2.252994792
[0.9381754654511284]
Guess for -0.329573531, -0.078472446, 4.29692E-4, 0.309340356, 0.792297776
[0.8986701154771921]
Guess for 0.072774233, -0.016914818, 0.203382964, 0.31174631, 1.06760863
[0.7489323788205811]
 Done Training
```

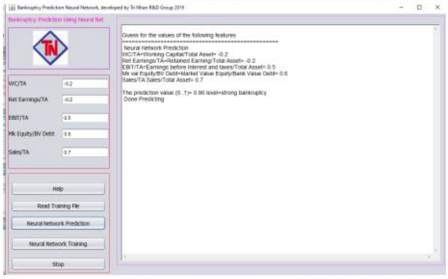
Vận hành mạng NN-fair bankruptcy


Bankruptcy Prediction Neural Network, developed by Tri Nhan R&D Group 2019
Bankruptcy Prediction Using Neural Net
Help
Read Training File
Neural Network Prediction
Neural Network Training
Stop

WC/TA .1
Ret Earnings/TA .2
EBIT/TA .2
Mk Equity/BV Debt .4
Sales/TA .1

```
Guess for the values of the following features
==========================================================
 Neural Network Prediction
WC/TA=Working Capital/Total Asset= 0.1
Ret Earnings/TA=Retained Earning/Total Asset= 0.2
EBIT/TA=Earnings before interest and taxes/Total Asset= 0.2
Mk val Equity/BV Debt=Market Value Equity/Bank Value Debt= 0.4
Sales/TA Sales/Total Asset= 0.1

The prediction value (0..1)= 0.35 level=fair bankruptcy
 Done Predicting
```

Vận hành mạng NN-strong bankruptcy


Bankruptcy Prediction Neural Network, developed by Tri Nhan R&D Group 2019
Bankruptcy Prediction Using Neural Net
Help
Read Training File
Neural Network Prediction
Neural Network Training
Stop

WC/TA -0.2
Ret Earnings/TA -0.2
EBIT/TA 0.5
Mk Equity/BV Debt 0.6
Sales/TA 0.7

```
Guess for the values of the following features
==========================================================
 Neural Network Prediction
WC/TA=Working Capital/Total Asset= -0.2
Ret Earnings/TA=Retained Earning/Total Asset= -0.2
EBIT/TA=Earnings before interest and taxes/Total Asset= 0.5
Mk val Equity/BV Debt=Market Value Equity/Bank Value Debt= 0.6
Sales/TA Sales/Total Asset= 0.7

The prediction value (0..1)= 0.86 level=strong bankruptcy
 Done Predicting
```