

MINISTRY OF EDUCATION AND SCIENCE OF REPUBLIC OF KAZAKHSTAN
SULEYMAN DEMIREL UNIVERSITY
ENGINEERING FACULTY



Department of Computer Engineering

UDC 004.042

Manuscript Copyright

Halil Alaka

Evaluation Performance of Students by use Neural Network in Kazakhstan

6M070400– «Computing systems and software» speciality

Kaskelen, 2013

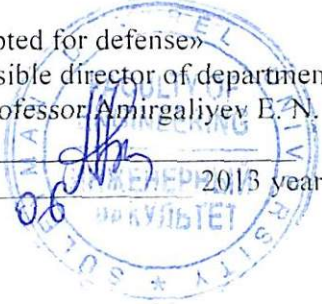
MINISTRY OF EDUCATION AND SCIENCE OF REPUBLIC OF KAZAKHSTAN
SULEYMAN DEMIREL UNIVERSITY
ENGINEERING FACULTY



Department of Computer Engineering

«Accepted for defense»
responsible director of department
d.t.s professor Amirgaliyev E. N.

« 13 » 06 2013 year



Director of department
for postgraduate education
Ph.s.c. S.Aydosgeldin

« » 2013 year



Master dissertation

Evaluation Performance of Students by use Neural Network in Kazakhstan

6M070400– «Computing systems and software» speciality

Master student

A handwritten signature in blue ink, appearing to be 'Halil Alaka'.

Halil Alaka

Supervisor

A handwritten signature in blue ink, appearing to be 'Bekir Karlık'.

Prof.Dr. Bekir Karlık

Kaskelen, 2013

Түйіндеме

Осы диссертация Жасанды Нейрон желісінің (ЖНЖ) Қазақстандағы Талдықорған қазақ-түрік жасөспірімдер лицейінің оқушыларының оқу үлгерімін болжау қабілеттілігін бағалайды.

Білім беру мекемелері - адам өмірінің маңызды бөлігі. Мектеп әкімшілігі және оқушылардың ата-аналары оқушының оқудағы үлгеріміне көп көңіл аударады. Академиялық зерттеулер оқушының үлгерімділік деңгейінің көтерілуін бақылау үшін бірнеше үлгісін өңдеп шығарды.

Жасанды Нейрон желісінің адамды шешім қабылдаумен қамтамасыз ету жүйесіндегі аналитикалық, логикалық тәсілдер арқылы шешілуі қиын проблемаларды шешу сипатымен қамтамасыз етуге шамасы бар. Жасанды Нейрон желісі үлгілерді шығару үшін және белгілі бір заңдылық жоқ немесе қисынсыз жағдайлардағы сипаттамаларды жасап шығару үшін үлкен көлемдегі мәліметтерді талдау жасай алады.

Жасанды Нейрон желісінің бұл үлгісі оқушылар туралы мәліметтерге сүйене құрылған. Жасанды жүйке желісіндегі TANI болжау бағдарламасы кіріс және шығыс параметрлер негізінде оқушының үлгерімін қадағалау қызметін атқарады.

Бұл зерттеу нәтижелері Нейрон желісі үлгісінің оқушының нағыз оқу үлгерімін ең кішкентай ақаулармен бақылай алатынын және қазіргі мен болашақтағы сипаттамалар ұқсас болған жағдайда жақсырақ бағалай алатынын көрсетеді.

Резюме

Результаты диссертации позволяют оценивать возможность Искусственных Нейронных Сетей (ИНС) к прогнозированию производительности учеников Талдыкорганского Казахско-Турецкого лицея в Казахстане.

Образовательные учреждения принимают большое участие в жизни людей. Школьная администрация и родители интересуются производительностью учеников. Исследователи из университетов разработали определенные модели для прогнозирования улучшения в производительности.

Искусственные Нейронные Сети имеют потенциал для обеспечения человеческих особенностей решения проблем, которые трудно смоделировать, используя аналитические и логические техники Системы Поддержки Принятия Решений (СППР). ИНС могут проанализировать большие объемы данных для установления шаблонов и характеристик в ситуациях, в которых правила или логика неизвестны.

Модель ИНС спроектирована и построена на доступной информации о студентах. Программа прогнозирования ИНС TANI используется для прогнозирования производительности учеников, основываясь на входных и выходных параметрах.

Результат этого исследования показывает, что модель нейронной сети может прогнозировать производительность ученика с очень малой погрешностью. При том, прогнозируя лучше, когда нынешние и будущие образцы имеют схожие характеристики.

ABSTRACT

This thesis evaluates the ability of Artificial Neural Networks (ANN) to predict the performance of Taldykorgan Kazakh Turkish High School students in Kazakhstan.

Educational institutions take a big part in people life. School administrators and student's parents pay attention for the performance of the students. Academic researchers have developed several models to predict the improve the performance.

Artificial Neural Networks have the potential to provide human characteristics of problem solving that are so difficult to simulate using the analytical, logical techniques of Decision Support System(DSS) . ANN can analyze big quantities of data to establish patterns and characteristics in situations where the rules or logic are not known(Turban,et al.2006).

ANN model is designed, built on available student's data. The TANI forecasting program of ANN is used to predict the student's performance based on inputs and output parameters.

This study's result shows that the neural network model can predict student's real performance with very small errors and predicts better when the today's and future's samples have similar characteristics

Önsöz

Bu çalışma Yapay Sinir Ağlarının (YSA) Taldykorgan Kazak Türk Lisesi öğrencilerinin performans tahminini değerlendirmektedir.

Eğitim kurumları insan hayatında önemli bir yer tutmaktadır. Okul yönetimleri ve aileler öğrencilerinin performansı adına ayrıca bir önemiyet vermektedir. Akademik çalışmalar performans gelişiminin tahmini için birçok model geliştirmiştir.

Yapay Sinir Ağları, insan karakterlerinin problemlerini çözmede Karar Destek Sisteminin mantık ve analitik tekniklerinin yanında büyük bir potansiyel sağlamaktadır. YSA, kural ve mantığın bilinmediği yerde yer alan modellerin ve karakterlerin büyük değerlerini analiz yapabilmektedir.

YSA modeli uygun öğrenci verileri tabanlı dizayn edilip kurulmuştur. Öğrencilerin girdi ve çıktı parametreleri kullanılarak öğrenci performansını değerlendirmede YSA'nın Tanı tahmin programından faydalanılmıştır.

bu çalışmanın sonucu gösteriyor ki sinir ağları modeli öğrenci performansını çok küçük hata payıyla tahmin edebilmekte ayrıca sonraki benzer karakterlere sahip örneklerde de daha iyi tahminlerde bulunabilmektedir.

CONTENTS

Contents	7
Abbreviations	10
Introduction	11
1.Researches on Evaluation Performance	12
2.What's Neural Network?	16
2.1.Historical background of ANN	16
2.2.Why Neural Network?	17
2.3.How the Human brain Learns	17
2.4.From Human Neurones to Artificial Neurones	19
2.5. A Simple Neuron	19
2.6. What Are the benefits of Artificial Neural Networks?	20
2.7. How to Use Artificial Neural Networks	21
2.8. Advantages	22
2.9. Disadvantages	22
2.10. Network Layers	23
2.10.1 Supervised Learning	24
2.10.2 Unsupervised Learning	24
3.The Backpropagation	25
3.1 Backpropagation Algorithm	27
3.2 Learning as an Optimization Problem	28

CONTENTS

3.3 An Analogy for Understanding Gradient Descent	31
3.4 Derivation	32
3.5 Models of Learning	35
3.6 Multithreaded backpropagation	36
3.7 History of backpropagation	37
4. Cascade Correlation Network	37
4.1 Cascade Correlation Network Architecture (CNN)	38
4.2 Training Algorithm for CNN	39
5. Evaluation Student's Performance	43
5.1 Methodology of Research	45
5.1.1 Data Collection	45
5.1.2 Variables of Research(inputs-outputs)	45
5.1.3 Application of Forecasting Results in KTL based on ANN	47
5.1.4 Selection of inputs and outputs for Tani Supervised Learning	47
5.1.5 Activation Function Selection	47
5.1.6 Training and Validating the Model	47
5.1.7 Trained Model's Performance Test	48
6. Application of Forecasting students' performance using ANN	49
6.1 Application Details and Discussion of obtained results	49

7. Conclusion

66

8. References

68

ABBREVIATIONS

(NN) Neural Network

(ANN) Artificial Neural Network

(BP) Backpropagation

(w) weight

$\sum w, EW, EA, EI$ sum of

(ITS) Intelligent Tutoring System

MOODLE (Modular Object-Oriented Dynamic Learning Environment)

(CMS) course management system

(ANFIS) adaptive neuro-fuzzy inference system

(AI) artificial intelligent

(CNN) cascade neural network

(RbF) radial basis function

(KTL) Kazakh Turkish Lyceum

(SMAS) Self-adaptive Monitoring and Analysis System

(MSE) Main Square Error

(AE) Absolute Error

(LMS) Least Mean Square

INTRODUCTION

Each day, life is become more complex and hurt. There are a lot of relations between events and people. If you know your way which you want to go, you start race firstly. You will come across with problems which you feel yourself very weak in front of it and you turn back or stop there. Our feelings are changeable. Today's rights can be tomorrow's wrongs. Your way can be changes unexpected time. You cannot understand what happened.

Future generation dependent on today's education. If their education is good enough so they can improve themselves and do their best. but we have a big problem. How can we give good education? Is education only teaching science or technology? Exactly we need technology and science but that's not all. We have to help them to expose their real performance which is already waiting for be discover.

Artificial Neural Network (ANN) is one of the prediction way. by this way we tried to solve this problem. If we decrease the wrong prediction we can get or solve problem easily.

1 RESEARCHES ON EVALUATION PERFORMANCE

In worldwide there are many researches about evaluation performance of firms, machines, students, academic personal and workers etc. but we just take studies about evaluation performance of human.

New technologies have provided the Education field with innovations that allow significant improvements in the teaching/learning process. Their introduction not only reduces the effective cost of the application of pedagogical theories but also opens up the possibility of exploring models from very different fields, facilitating their interaction and integration. One of the main innovations introduced since the first Computer Aided Learning programs are the so-called Intelligent Tutoring System (ITS), that, in contrast to traditional programs, have the ability to adapt to each individual learner. It is precisely this ability to adapt to each student that allows these programs to improve the teaching/learning process, as it has already been shown that the best learning method is individualized learning.

Therefore, if the key characteristic of an ITS is its ability to adapt to each student the key component of such a system is the student model, where all the information about the student is stored, including his/her cognitive state about the subject domain. The cognitive state is generated from student behavior during interaction with the system, that is, it is inferred by the system from the information available; previous data answers to questions posed by the system, so by the previous data next student information evaluate[1].

A total of 341 Latvian students and eight teachers participated in this study of student self-evaluation and teacher evaluation. Students completed a 12-lesson teacher-directed instructional program on conducting and writing a report of their own experimental research. Sixteen classes were randomly assigned to one of four treatment conditions;

1. no in-program evaluation,

2.self evaluation and revision at the research design and draft final report stages,

3.teacher evaluation and student revision at both stages,

4.self-plus-teacher evaluation and student revision at both stages. Student in the teacher-evaluation and self-plus-teacher evaluation conditions received significantly higher ratings from an independent rater on their final research reports. However, students under the self –evaluation conditions had greater confidence in their ability to conduct future experiments[2].

MOODLE (Modular Object-Oriented Dynamic Learning Environment) is defined as a course management system (CMS), a free,Open Source software package designed using pedagogical principles, to help educators by creating effective online learning communities[3]. This system has a number of interactive learning activity modules like forums, chats, quizzes and assignments that facilitate the learning from a participative position. In addition to these learning modules, it includes another one to register and track user`s accesses and the activities and resources that have been accessed. For each user, the system stores ;the hour of the access, the IP address of the user`s computer, the user identification and the kind of activity or resources that has been visited. Artificial Neural Network proposed has been developed and tested using the logs of the access from a group of 240 students of data processing of the University of Cordoba in the Methodology and Programming Technology[4].

In another study: To further clarify the salient difference between measurement and evaluation, Arreola (2000: xviii) indicated that “evaluation is the process of interpreting a measurement (or aggregate of measurements) by means of a specific value or a set of values, to determine the degree to which the measurement(s) represent a desirable condition”. Thus, the result of evaluation is a judgement of the degree to which the measurement or aggregate of measurement represents the desirable condition. Therefore, judgements may be represented as

either a word (that is, “excellent”, “good”, “poor”, or a number where the number is used simply as a label equivalent to a word (e.g. excellent = 1, good = 2, poor =3) [5].

There is one more study about evaluation student performance, according to the work focuses on a systematic approach in assessing and reasoning the student's performance and efficiency level in programming technique course. There are four criteria required to indicate the student's performance and efficiency level which are scores earned, time spent, number of attempts and help needed. A fuzzy rule base model that has been proposed in previous work is found to be insufficient in deciding all possible conditions. To deal with this problem, this work focuses on the adaptive neuro-fuzzy inference system (ANFIS) approach in determining the possible conditions in order to form a fuzzy rule based system of a student model. The back-propagation is utilized as the learning mechanism for the neural network to solve the incompleteness in the decision made by human experts. by training the neural network with 18 human decisions that are certain, the neural network has successfully derived other decisions to form a complete fuzzy rule base and able to adjust its parameter by learning mechanism. However, some of the decisions are found illogically classified[6].

On the other hand student's behaviors in classroom was also research subject. Recently, there are many problems in software laboratory course management. It is, particularly, widespread that student play video games or do other things which are unrelated to the course. However, most of present monitoring software are all based on the operation of teachers, rather than a real-time system. which is very inconvenient. In order to carry out the laboratory course management better and help teachers to monitor and analyze students' behaviors of operating computer, we design and implement a model of Self-adaptive Monitoring

And Analysis System (SMAS) for students' behaviors in laboratory course. For monitoring, we implement a module which uses the technology neural-

network-based expert system, with which we can monitor the students' behavior real-time and self-adaptively. For analysis, with the methodologies such as Factor Analysis, Cluster Analysis, and Discriminant Analysis, we propose and implement an evaluation algorithm, which can evaluate the students' performance in a course quantificationally and obviously[7].

The similar study with this study is from Arabian Gulf University. The Effectiveness of Artificial Neural Networks in Forecasting the Failure Risk:Case Study on Pre-Medical Students of Arabian Gulf University submitted by Jawaher Khaled Aref Alanzi in 2003.

In her study she researched performance of student and their failure risk at college of medicine and medical sciences. She used student's parameters like Grade Point Average, score of medical college admission test, and a structured personal interview. Artificial Neural Networks-Cascade Correlation Network Architecture which is trained using the Quick Propagation algorithm is used for prediction. ANN model is designed, based on available existing academic acceptance system results. The cascade correlation learning structure ANN is used to forecast the student's grades based on inputs and output parameters.

As a result by the neural network student's performance can be predicted.

2 WHAT'S NEURAL NETWORK?

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurones. This is true of ANNs as well.

2.1 Historical background

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras.

Many important advances have been boosted by the use of inexpensive computer emulations. Following an initial period of enthusiasm, the field survived a period of frustration and disrepute. During this period when funding and professional support was minimal, important advances were made by relatively few researchers. These pioneers were able to develop convincing technology which surpassed the limitations identified by Minsky and Papert. Minsky and Papert, published a book (in 1969) in which they summed up a general feeling of frustration (against neural networks) among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding.

The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pitts. but the technology available at that time did not allow them to do too much[8].

2.2 Why Neural Network?

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
2. Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.
3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

2.3 How the Human brain Learns?

Much is still unknown about how the brain trains itself to process information, so theories abound. In the human brain, a typical neuron collects signals from others through a host of fine structures called *dendrites*. The neuron sends out spikes of electrical activity through a long, thin strand known as an *axon*,

which splits into thousands of branches. At the end of each branch, a structure called a *synapse* converts the activity from the axon into electrical effects that inhibit or excite activity from the axon into electrical effects that inhibit or excite activity in the connected neurones. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes.

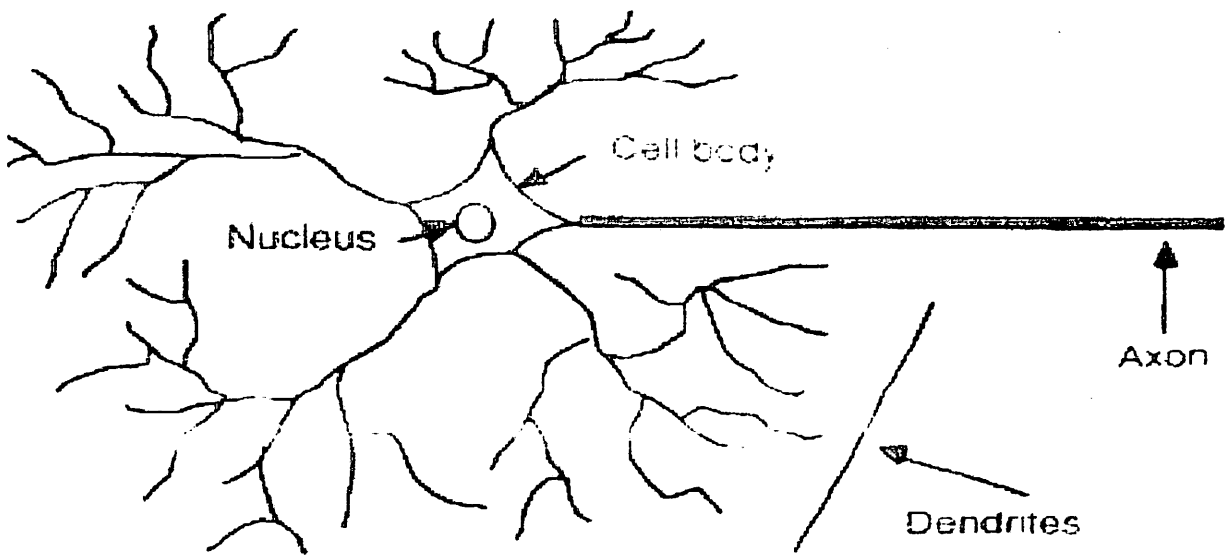


Figure 1-Human Neuron

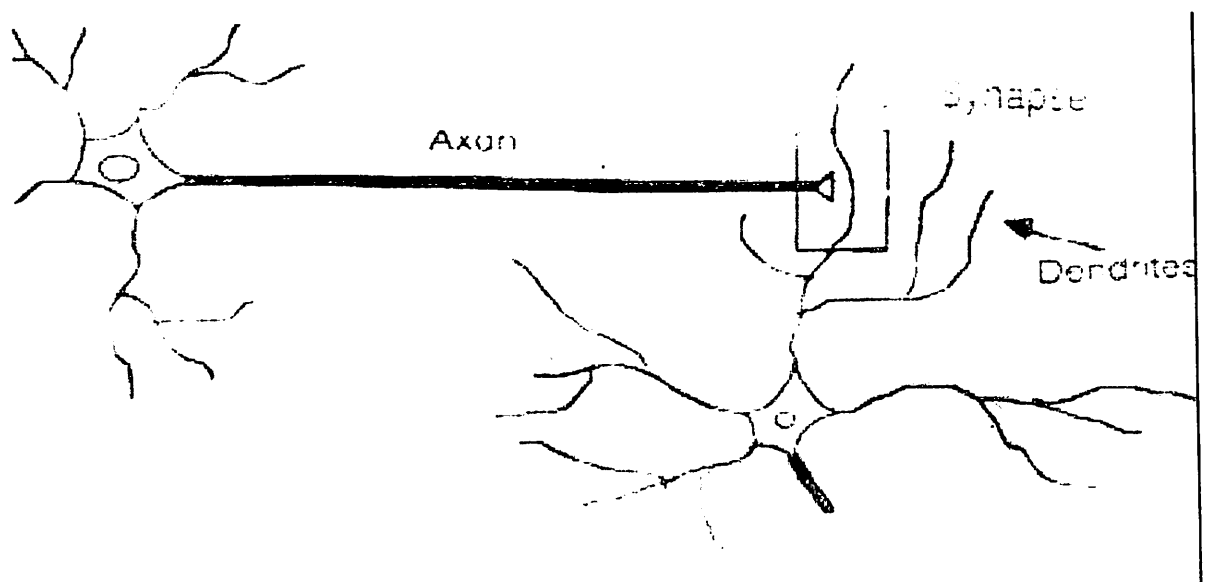


Figure 2-Synapse and Components of Neuron

2.4 From Human Neurones to Artificial Neurones

We conduct these neural networks by first trying to deduce the essential features of neurones and their interconnections. We then typically program a computer to simulate these features. However because our knowledge of neurones is incomplete and our computing power is limited, our models are necessarily gross idealisations of real networks of neurones.

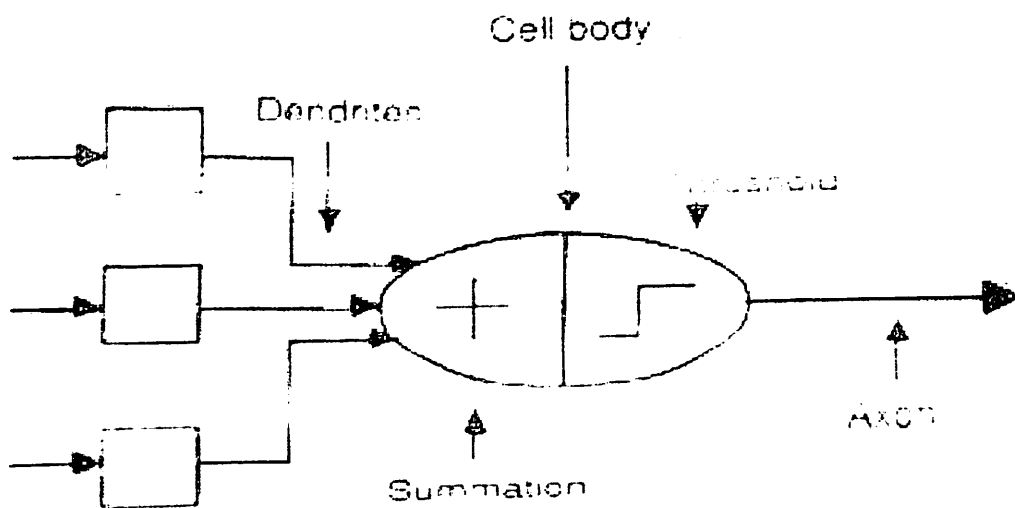


Figure 3-The neuron model

2.5 A Simple Neuron

An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not[9].

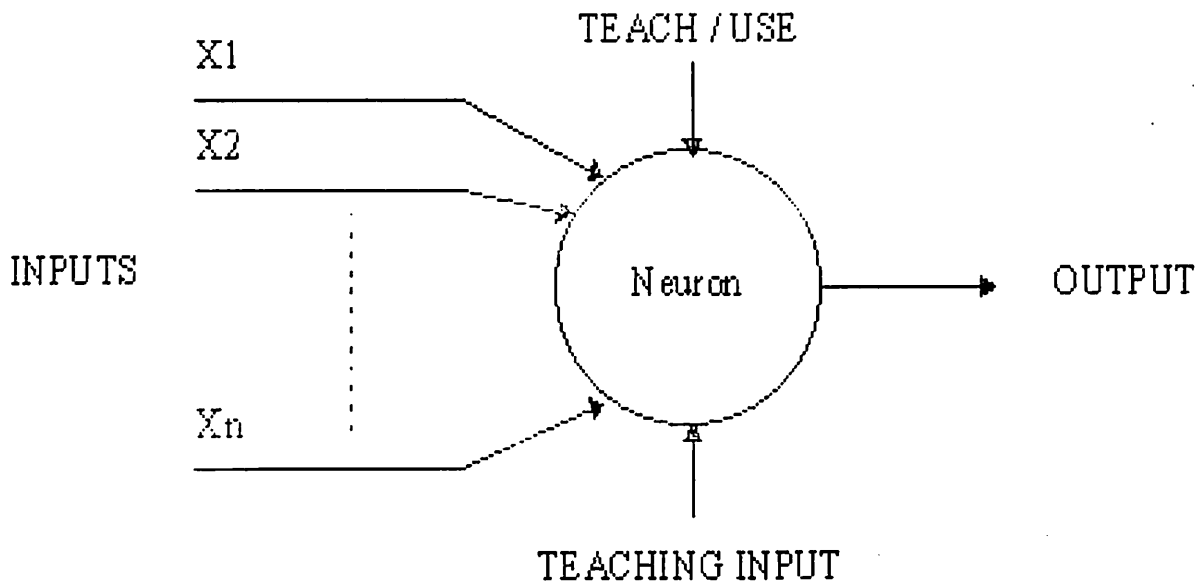


Figure 4-Simple Neuron

2.6 What Are the benefits of Artificial Neural Networks?

An artificial neural network is a programmed computational model that aims to replicate the neural structure and functioning of the human brain. It is made up of an interconnected structure of artificially produced neurons that function as pathways for data transfer. Artificial neural networks are flexible and adaptive, learning and adjusting with each different internal or external stimuli. Artificial neural networks are used in sequence and pattern recognition systems, data processing, robotics and modeling. There are different types of neural networks, including feedforward neural network, radial basis function (RbF), Kohonen self-organizing network and the recurrent neural network[10].

2.7 How to Use Artificial Neural Networks

1. Collect the examples that will be used to train the neural network. How you do this is crucial to the training process. If you do not start with good examples, the training will very likely fail. Assuming that the neural network is

going to distinguish between good and bad patterns, there should be roughly the same number of each and both the good patterns and the bad patterns should represent a statistical distribution of the possible patterns. Divide the examples into two sets: the training set and the testing set. Both sets should have an equal number of good and bad patterns and both sets should represent a statistical distribution of the possible patterns. Start the separation into training set and testing set by choosing at random. If the distribution is not statistically representative, choose again until it is.

2. Go through the training set, one example at a time. Present an example to the neural network and, if the network identifies it correctly, do nothing. If the example is misclassified, adjust the weights slightly so that the answer would be slightly better if the same pattern were presented to the net again. Continue all the way through the training set. After each time through the training set, test the net with all the examples in the testing set. Do not change the net weights during presentations of the testing set.

3. Continue this cycle until the net correctly identifies all the patterns in the testing set. If this never happens, you either need a different neural network or a different set of examples. For example, if you are using a feedforward network, you might need to change the number of elements in the hidden layer. If the problem is with the examples, you might try choosing more examples in the set[11].

2.8 Advantages

Neural networks can develop algorithms based on known inputs and results, and can eventually learn to predict events with a high degree of certainty. because neural networks are parallel systems, if one part of the system fails, other parts continue to function normally. because a neural network is a natural learning system. once created, it generally requires no programming.

2.9 Disadvantages

One disadvantage of a neural network is that it needs time to train. The network will only be as good as the initial data and corrected training data. If the neural network is given incorrect data and told the data is correct, it will apply that invalid data to future decisions and predictions. Software-based neural networks run on different architectures than most common computers. Therefore, unless a dedicated hardware and software system is used, complex translation systems are required to convert neural data to a format usable by common computing systems[12].

2.10 Network Layers

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "**input**" units is connected to a layer of "**hidden**" units, which is connected to a layer of "**output**" units.

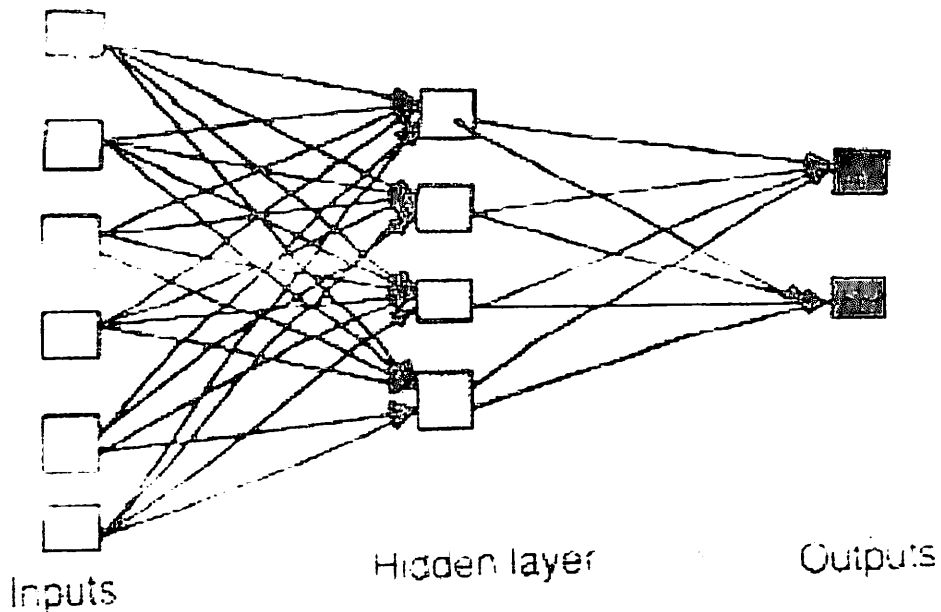


Figure 5-An example of a simple feedforward network

- The activity of the input units represents the raw information that is fed into the network.

- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.

- The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

We also distinguish single-layer and multi-layer architectures. The single-layer organisation, in which all units are connected to one another, constitutes the most general case and is of more potential computational power than hierarchically

structured multi-layer organisations. In multi-layer networks, units are often numbered by layer, instead of following a global numbering.

All learning methods used for adaptive neural networks can be classified into two major categories:

2.10.1 Supervised learning

Supervised learning which incorporates an external teacher, so that each output unit is told what its desired response to input signals ought to be. During the learning process global information may be required. Paradigms of supervised learning include error-correction learning, reinforcement learning and stochastic learning.

An important issue concerning supervised learning is the problem of error convergence, ie the minimisation of error between the desired and computed unit values. The aim is to determine a set of weights which minimises the error. One well-known method, which is common to many learning paradigms is the least mean square (LMS) convergence.

2.10.2 Unsupervised learning

Unsupervised learning uses no external teacher and is based upon only local information. It is also referred to as self-organisation, in the sense that it self-organises data presented to the network and detects their emergent collective properties. Paradigms of unsupervised learning are Hebbian learning and competitive learning. From Human Neurones to Artificial Neurons the aspect of learning concerns the distinction or not of a separate phase, during which the network is trained, and a subsequent operation phase. We say that a neural network learns off-line if the learning phase and the operation phase are distinct. A neural network learns on-line if it learns and operates at the same time. Usually, supervised learning is performed off-line, whereas unsupervised learning is performed on-line.

3 THE BACKPROPAGATION

Backpropagation, an abbreviation for "backward propagation of errors", is a common method of training artificial neural networks. From a desired output, the network learns from many inputs, similar to the way a child learns to identify a dog from examples of dogs.

It is a supervised learning method, and is a generalization of the delta rule. It requires a dataset of the desired output for many inputs, making up the training set. It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections that loop). backpropagation requires that the activation function used by the artificial neurons (or "nodes") be differentiable.

The goal of any supervised learning algorithm is to find a function that best maps a set of inputs to its correct output. An example would be a simple classification task, where the input is an image of an animal, and the correct output would be the name of the animal. Some input and output patterns can be easily learned by single-layer neural networks (i.e. perceptrons). However, these single-layer perceptrons cannot learn some relatively simple patterns, such as those that are not linearly separable. To illustrate its limitation, consider the example mentioned above of ascribing labels to images. A human may classify an image of an animal by recognizing certain features such as the number of limbs, the texture of the skin (whether it is furry, feathered, scaled, etc.), the size of the animal, and the list goes on. A single-layer neural network however, must learn a function that outputs a label solely using the intensity of the pixels in the image. There is no way for it to learn any abstract features of the input since it is limited to having only one layer. A multi-layered network overcomes this limitation as it can create internal representations and learn different features in each layer. The first layer may be responsible for learning the orientations of lines using the inputs from the individual pixels in the image. The second layer may combine the features learned in the first layer and learn to identify simple shapes such as circles. Each higher layer learns more and more abstract features such as those mentioned above that

can be used to classify the image. Each layer finds patterns in the layer below it and it is this ability to create internal representations that are independent of outside input that gives multi-layered networks its power. The goal and motivation for developing the backpropagation algorithm is to find a way to train multi-layered neural networks such that it can learn the appropriate internal representations to allow it to learn any arbitrary mapping of input to output.^[1]

For better understanding, the backpropagation learning algorithm can be divided into two phases: propagation and weight update.

Phase 1: Propagation

Each propagation involves the following steps:

1. Forward propagation of a training pattern's input through the neural network in order to generate the propagation's output activations.
2. backward propagation of the propagation's output activations through the neural network using the training pattern target in order to generate the deltas of all output and hidden neurons.

Phase 2: Weight update

For each weight-synapse follow the following steps:

1. Multiply its output delta and input activation to get the gradient of the weight.
2. bring the weight in the opposite direction of the gradient by subtracting a ratio of it from the weight.

This ratio influences the speed and quality of learning; it is called the *learning rate*. The sign of the gradient of a weight indicates where the error is increasing, this is why the weight must be updated in the opposite direction.

Repeat phase 1 and 2 until the performance of the network is satisfactory.

Backpropagation Algorithm

Actual algorithm for a 3-layer network (only one hidden layer):

initialize the weights in the network (often small random values)

do

for each example e in the training set

$O = \text{neural-net-output}(\text{network}, e)$ // forward pass

$T = \text{teacher output for } e$

compute error $(T - O)$ at the output units

compute δ_{wh} for all weights from hidden layer to output layer //

backward pass

compute δ_{wi} for all weights from input layer to hidden layer //

backward pass continued

update the weights in the network

until all examples classified correctly or stopping criterion satisfied

return the network

As the algorithm's name implies, the errors propagate backwards from the output nodes to the input nodes. Technically speaking, backpropagation calculates the gradient of the error of the network regarding the network's modifiable weights.^[2] This gradient is almost always used in a simple stochastic gradient descent algorithm to find weights that minimize the error. Often the term "backpropagation" is used in a more general sense, to refer to the entire procedure encompassing both the calculation of the gradient and its use in stochastic gradient

descent. backpropagation usually allows quick convergence on satisfactory local minima for error in the kind of networks to which it is suited.

Backpropagation networks are necessarily multilayer perceptrons (usually with one input, one hidden, and one output layer). In order for the hidden layer to serve any useful function, multilayer networks must have non-linear activation functions for the multiple layers: a multilayer network using only linear activation functions is equivalent to some single layer, linear network. Non-linear activation functions that are commonly used include the logistic function, the softmax function, and the gaussian function.

The backpropagation algorithm for calculating a gradient has been rediscovered a number of times, and is a special case of a more general technique called automatic differentiation in the reverse accumulation mode.

It is also closely related to the Gauss–Newton algorithm, and is also part of continuing research in neural backpropagation.

3.2 Learning as an optimization problem

Before showing the mathematical derivation of the backpropagation algorithm, it helps to develop some intuitions about the relationship between the actual output of a neuron and the correct output for a particular training case. Consider a simple neural network with two input units, one output unit and no hidden units. Each neuron uses a linear output that is the weighted sum of its input.

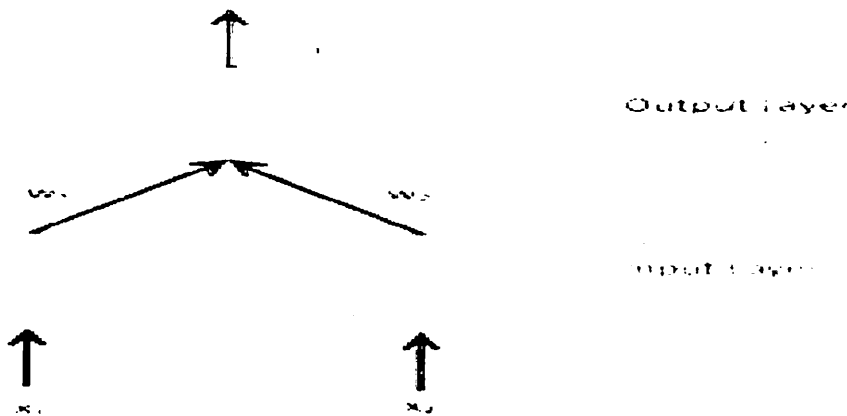


Figure 6-input output layers

A simple neural network with two input units and one output unit

Initially, before training, the weights will be set randomly. Then the neuron learns from training examples, which in this case consists of a set of tuples (x_1, x_2, t) where x_1 and x_2 are the inputs to the network and t is the correct output (the output the network should eventually produce given the identical inputs). The network given x_1 and x_2 will compute an output y which very likely differs from t (since the weights are initially random). A common method for measuring the discrepancy between the expected output t and the actual output y is using the squared error measure:

$$E = (t - y)^2,$$

where E is the discrepancy or error.

As an example, consider the network on a single training case: (1, 1, 0), thus the input x_1 and x_2 are 1 and 1 respectively and the correct output, t , is 0. Now if the actual output y is plotted on the x-axis against the error E on the y-axis, the result is a parabola. The minima of the parabola corresponds to the output y which minimizes the error E . For a single training case, the minima also touches the x-axis, which means the error will be zero and the network can produce an output y that exactly matches the expected output t . Therefore, the problem of mapping inputs to outputs can be reduced to an optimization problem of finding a function that will produce the minimal error.

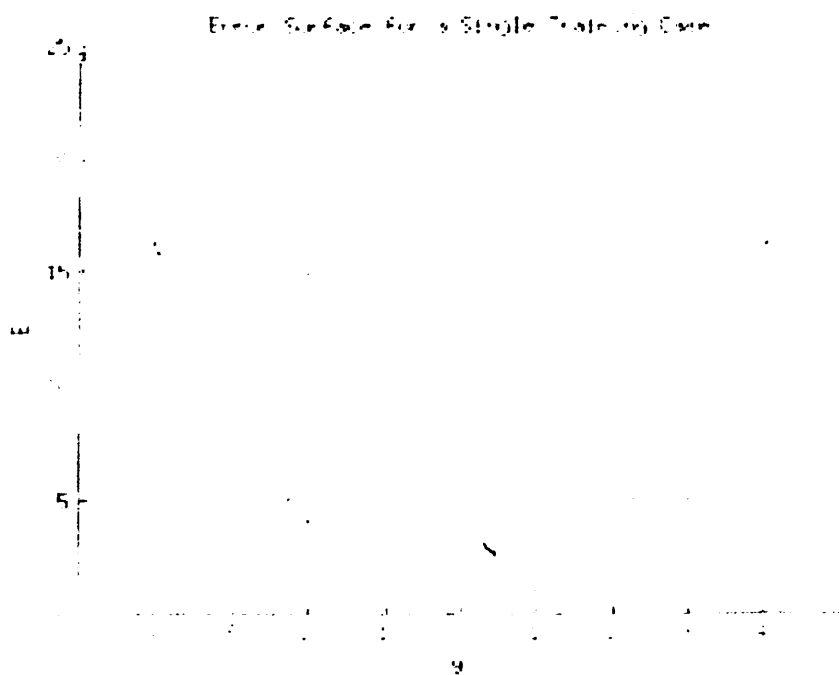


Figure 7-Error surface of a linear neuron for a single training case.

However, the output of a neuron depends on the weighted sum of all its inputs:

$$y = x_1w_1 + x_2w_2,$$

where w_1 and w_2 are the weights on the connection from the input units to the output unit. Therefore, the error also depends on the incoming weights to the neuron, which is ultimately what needs to be changed in the network to enable learning. If each weight is plotted on a separate horizontal axis and the error on the vertical axis, the result is a parabolic bowl (If a neuron has k weights, then

the dimension of the error surface would be $k+1$, thus a $k+1$ dimensional equivalent of the 2D parabola).

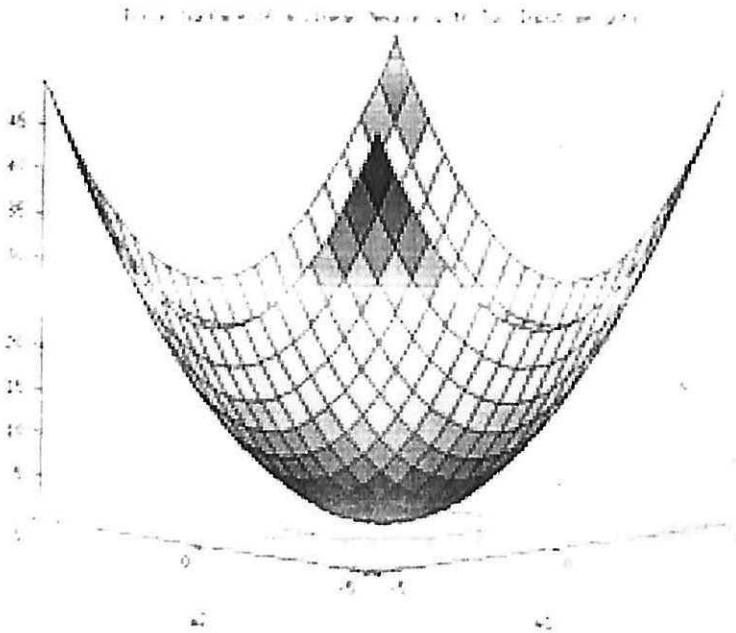


Figure 8-Error surface of a linear neuron with two input weights

The backpropagation algorithm aims to find the set of weights that minimizes the error. There are several methods for finding the minima of a parabola or any function in any dimension. One way is analytically by solving systems of equations, however this relies on the network being a linear system, and the goal is to be able to also train multi-layer, non-linear networks (since a multi-layered linear network is equivalent to a single-layer network). The method used in backpropagation is gradient descent.

3.3 An analogy for understanding gradient descent

The basic intuition behind gradient descent can be illustrated by a hypothetical scenario. A person is stuck in the mountains is trying to get down (i.e. trying to find the minima). There is heavy fog such that visibility is extremely low. Therefore, the path down the mountain is not visible, so he must use local information to find the minima. He can use the method of gradient descent, which

involves looking at the steepness of the hill at his current position, then proceeding in the direction with the most negative steepness (i.e. downhill). If he was trying to find the top of the mountain (i.e. the maxima), then he would proceed in the direction with most positive steepness (i.e. uphill). Using this method, he would eventually find his way down the mountain. However, assume also that the steepness of the hill is not immediately obvious with simple observation, but rather it requires a sophisticated instrument to measure, which the person happens to have at the moment. It takes quite some time to measure the steepness of the hill with the instrument, thus he should minimize his use of the instrument if he wanted to get down the mountain before sunset. The difficulty then is choosing the frequency at which he should measure the steepness of the hill so not to go off track.

In this analogy, the person represents the backpropagation algorithm, and the path down the mountain represents the set of weights that will minimize the error. The steepness of the hill represents the slope of the error surface at that point. The direction he must travel in corresponds to the gradient of the error surface at that point. The instrument used to measure steepness is differentiation (the slope of the error surface can be calculated by taking the derivative of the squared error function at that point). The distance he travels in between measurements (which is also proportional to the frequency as which he takes measurement) is the learning rate of the algorithm. See the limitation section for a discussion of the limitations of this type of "hill climbing" algorithm.

3.4 Derivation:

Since backpropagation uses the gradient descent method, one needs to calculate the derivative of the squared error function with respect to the weights of the network. The squared error function is (the $\frac{1}{2}$ term is added to cancel the exponent when differentiating):

Therefore the error, E , depends on the output y . However, the output y depends on the weighted sum of all its input:

$$y = \sum_{i=1}^n w_i x_i$$

n = the number of input units to the neuron

w_i = the i^{th} weight

x_i = the i^{th} input value to the neuron

The above formula only holds true for a neuron with a linear activation function (that is the output is solely the weighted sum of the input). In general, a non-linear, differentiable activation function, φ , is used. Thus, more correctly:

$$y = \varphi(\text{net})$$

$$\text{net} = \sum_{i=1}^n w_i x_i$$

This lays the groundwork for calculating the partial derivative of the error with respect to a weight w_i using the chain rule:

Since the weighted sum net is just the sum over all products $w_i x_i$, therefore the partial derivative of the sum with respect to a weight w_i is just the corresponding input x_i . Similarly, the partial derivative of the sum with respect to an input value x_i is just the weight w_i :

$$\frac{\partial \text{net}}{\partial w_i} = x_i$$

$$\frac{\partial \text{net}}{\partial x_i} = w_i$$

The derivative of the output y with respect to the weighted sum net is simply the derivative of the activation function φ :

$$\frac{dy}{d\text{net}} = \frac{d}{d\text{net}} \varphi$$

This is the reason why backpropagation requires the activation function to be differentiable. A commonly used activation function is the logistic function:

$$y = \frac{1}{1 + e^{-z}}$$

which has a nice derivative of:

$$\frac{dy}{dz} = y(1 - y)$$

For example purposes, assume the network uses a logistic activation function, in which case the derivative of the output y with respect to the weighted sum net is the same as the derivative of the logistic function:

$$\frac{dy}{dnet} = y(1 - y)$$

Finally, the derivative of the error E with respect to the output y is:

$$\frac{dE}{dy} = \frac{d}{dy} \frac{1}{2} (t - y)^2$$

$$\frac{dE}{dy} = y - t$$

Putting it all together:

$$\frac{\partial E}{\partial w_i} = \frac{dE}{dy} \frac{dy}{dnet} \frac{\partial net}{\partial w_i}$$

$$\frac{\partial E}{\partial w_i} = (y - t)y(1 - y)x_i$$

If one were to use a different activation function, the only difference would be the $y(1 - y)$ term will be replaced by the derivative of the newly chosen activation function.

To update the weight w_i using gradient descent, one must choose a learning rate, α . The change in weight after learning then would be the product of the learning rate and the gradient:

$$\Delta w_i = -\alpha \frac{\partial E}{\partial w_i}$$

$$\Delta w_i = \alpha(t - y)\varphi'x_i$$

For a linear neuron, the derivative of the activation function φ is 1, which yields:

$$\Delta w_i = \alpha(t - y)x_i$$

This is exactly the delta rule for perceptron learning, which is why the backpropagation algorithm is a generalization of the delta rule. In backpropagation and perceptron learning, when the output y matches the desired output t , the change in weight Δw_i would be zero, which is exactly what is desired.

3.5 Modes of learning

There are three modes of learning to choose from: on-line, batch and stochastic. In on-line and stochastic learning, each propagation is followed immediately by a weight update. In batch learning, many propagations occur before updating the weights. batch learning requires more memory capacity, but on-line and stochastic learning require more updates. On-line learning is used for dynamic environments that provide a continuous stream of new patterns. Stochastic learning and batch learning both make use of a training set of static patterns. Stochastic goes through the data set in a random order in order to reduce its chances of getting stuck in local minima. Stochastic learning is also much faster than batch learning since weights are updated immediately after each propagation. Yet batch learning will yield a much more stable descent to a local minima since each update is performed based on all patterns.

3.6 Multithreaded backpropagation

Backpropagation is an iterative process that can often take a great deal of time to complete. When multicore computers are used, multithreaded techniques can greatly decrease the amount of time that backpropagation takes to converge. If batching is being used, it is relatively simple to adapt the backpropagation algorithm to operate in a multithreaded manner. The training data is broken up into equally large batches for each of the threads. Each thread executes the forward and backward propagations. The weight and threshold deltas are summed for each of the threads. At the end of each iteration all threads must pause briefly for the weight and threshold deltas to be summed and applied to the neural network. This process continues for each iteration. This multithreaded approach to backpropagation is used by the Encog Neural Network Framework.[17] The result may converge to a local minimum. The "hill climbing" strategy of gradient descent is guaranteed to work if there is only one minimum. However, oftentimes the error surface has many local minima and maxima. If the starting point of the gradient descent happens to be somewhere between a local maximum and local minimum, then going down the direction with the most negative gradient will lead to the local minimum.

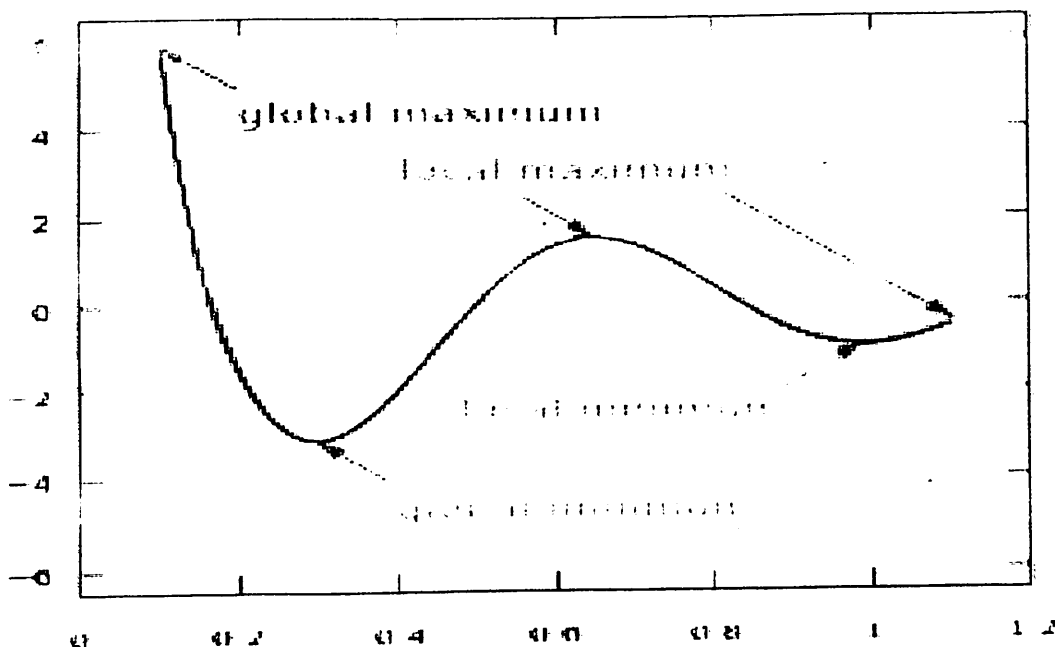


Figure 9-local,global max-min

Gradient descent can find the local minimum instead of the global minimum

- The convergence obtained from backpropagation learning is very slow.
- The convergence in backpropagation learning is not guaranteed.
- backpropagation learning does not require or normalization of input vectors; however, normalization could improve performance.^[18]

3.7 History fo backpropagation

Vapnik cites (bryson, A.E.; W.F. Denham; S.E. Dreyfus. Optimal programming problems with inequality constraints. I: Necessary conditions for extremal solutions. AIAA J. 1, 11 (1963) 2544-2550) as the first publication of the backpropagation algorithm in his book "Support Vector Machines.". Arthur E. bryson and Yu-Chi Hoddescribed it as a multi-stage dynamic system optimization method in 1969[19][20]. It wasn't until 1974 and later, when applied in the context of neural networks and through the work of Paul Werbos,[21] David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams,[22] that it gained recognition, and it led to a "renaissance" in the field of artificial neural network research. During the 2000s it fell out of favour but has returned again in the 2010s, now able to train much larger networks using huge modern computing power such as GPUs. For example, in 2013 top speech recognisers now use backpropagation-trained neural networks.

4. CASCADE CORRELATION NETWORK

Cascade correlation neural networks (Fahlman and I ibiere, 1990).are “self organizing” networks. The network begins with only input and output neurons. During the training process, neurons are selected from a pool of candidates and added to the hidden layer. Cascade correlation networks have several advantages over multilayer perceptron networks,because ;they are self organizing and grow the hidden layer during training, you do not have to be concerned with the issue of deciding how many layers and neurons to use in the network.

- Training time is very fast – often 100 times as fast as a perceptron network. This makes cascade correlation networks suitable for large training sets.
- Typically, cascade correlation networks are fairly small, often having fewer than a dozen neurons in the hidden layer. Contrast this to probabilistic neural networks which require a hidden-layer neuron for each training case.
- Cascade correlation network training is quite robust, and good results usually can be obtained with little or no adjustment of parameters.
- Cascade correlation is less likely to get trapped in local minima than multilayer perceptron networks.

As with all types of models, there are some disadvantages to cascade correlation networks:

- They have an extreme potential for overfitting the training data; this results in excellent accuracy on the training data but poor accuracy on new, unseen data. DTREG includes an overfitting control facility to prevent this.
- Cascade correlation networks usually are less accurate than probabilistic and general regression neural networks on small to medium size problems (i.e., fewer than a couple of thousand training rows). but cascade correlation scales up to handle large problems far better than probabilistic or general regression networks.

4.1 Cascade Correlation Network Architecture

A cascade correlation network consists of a cascade architecture, in which hidden neurons are added to the network one at a time and do not change after they have been added. It is called a cascade because the output from all neurons already in the network feed into new neurons. As new neurons are added to the hidden layer, the learning algorithm attempts to maximize the magnitude of the correlation between the new neuron's output and the residual error of the network which we are trying to minimize.

A cascade correlation neural network has three layers: input, hidden and output.

Input Layer: A vector of predictor variable values ($x_1 \dots x_p$) is presented to the input layer. The input neurons perform no action on the values other than distributing them to the neurons in the hidden and output layers. In addition to the predictor variables, there is a constant input of 1.0, called the *bias* that is fed to each of the hidden and output neurons; the bias is multiplied by a weight and added to the sum going into the neuron.

Hidden Layer: Arriving at a neuron in the hidden layer, the value from each input neuron is multiplied by a weight, and the resulting weighted values are added together producing a combined value. The weighted sum is fed into a transfer function, which outputs a value. The outputs from the hidden layer are distributed to the output layer.

Output Layer: For regression problems, there is only a single neuron in the output layer. For classification problems, there is a neuron for each category of the target variable. Each output neuron receives values from all of the input neurons (including the bias) and all of the hidden layer neurons. Each value presented to an output neuron is multiplied by a weight, and the resulting weighted values are added together producing a combined value. The weighted sum is fed into a

transfer function, which outputs a value. The y values are the outputs of the network. For regression problems, a linear transfer function is used in the output neurons. For classification problems, a sigmoid transfer function is used.

4.2 Training Algorithm for Cascade Correlation Networks

Initially, a cascade correlation neural network consists of only the input and output layer neurons with no hidden layer neurons. Every input is connected to every output neuron by a connection with an adjustable weight, as shown below:

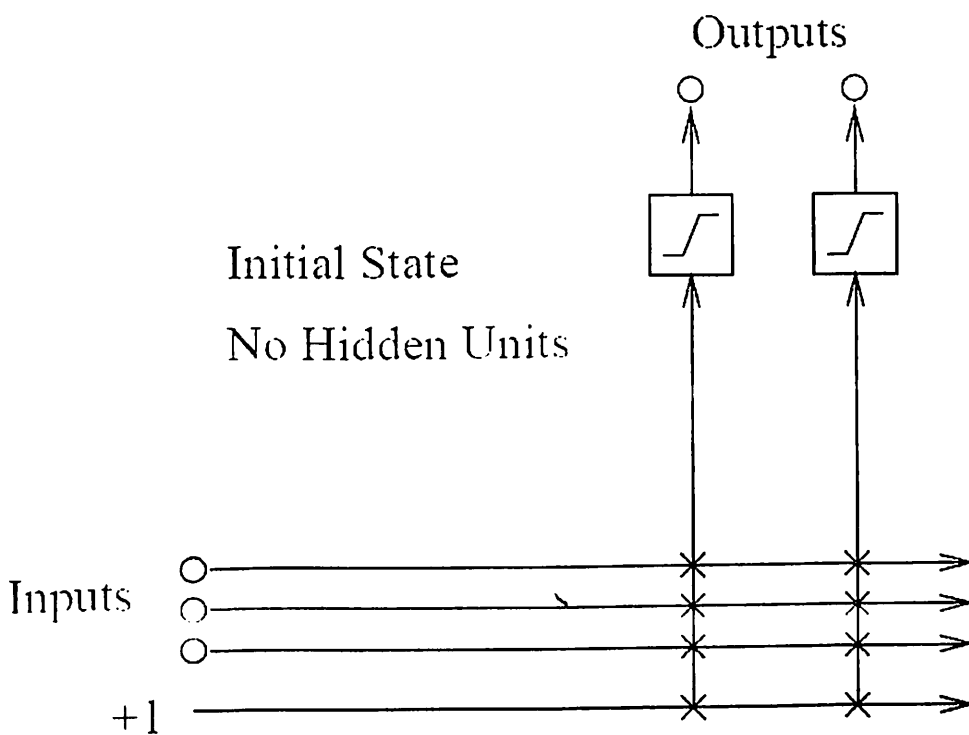


Figure 10-Every input is connected to every output neuron by a connection with an adjustable weight

Each 'x' represents a weight value between the input and the output neuron. Values on a vertical line are added together after being multiplied by their weights. So each output neuron receives a weighted sum from all of the input neurons including the bias. The output neuron sends this weighted input sum through its transfer function to produce the final output.

Even a simple cascade correlation network with no hidden neurons has considerable predictive power. For a fair number of problems, a cascade

correlation network with just input and output layers provides excellent predictions.

After the addition of the first hidden neuron, the network would have this structure:

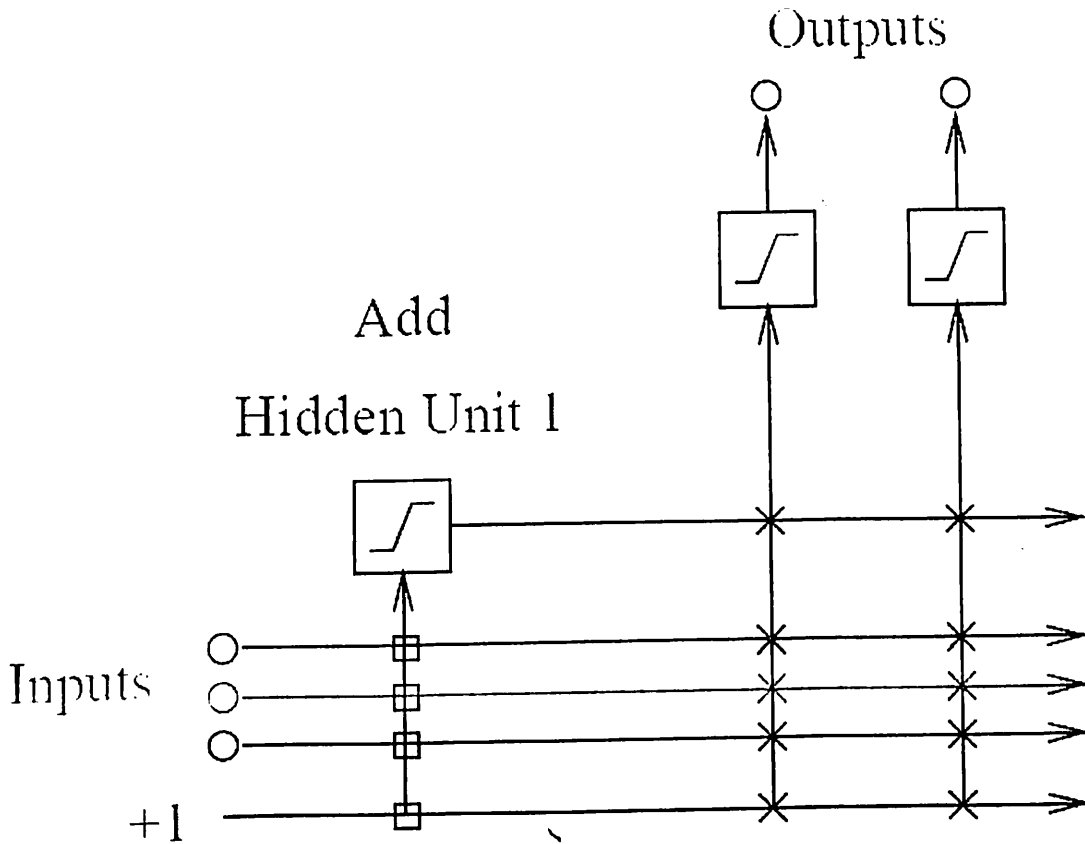


Figure 11-After the addition of the first hidden neuron, the network would have this structure

The input weights for the hidden neuron are shown as square boxes to indicate that they are fixed once the neuron has been added. Weights for the output neurons shown as 'x' continue to be adjustable.

Here is a schematic of a network with two hidden neurons. Note how the second neuron receives inputs from the external inputs and pre-existing hidden neurons.

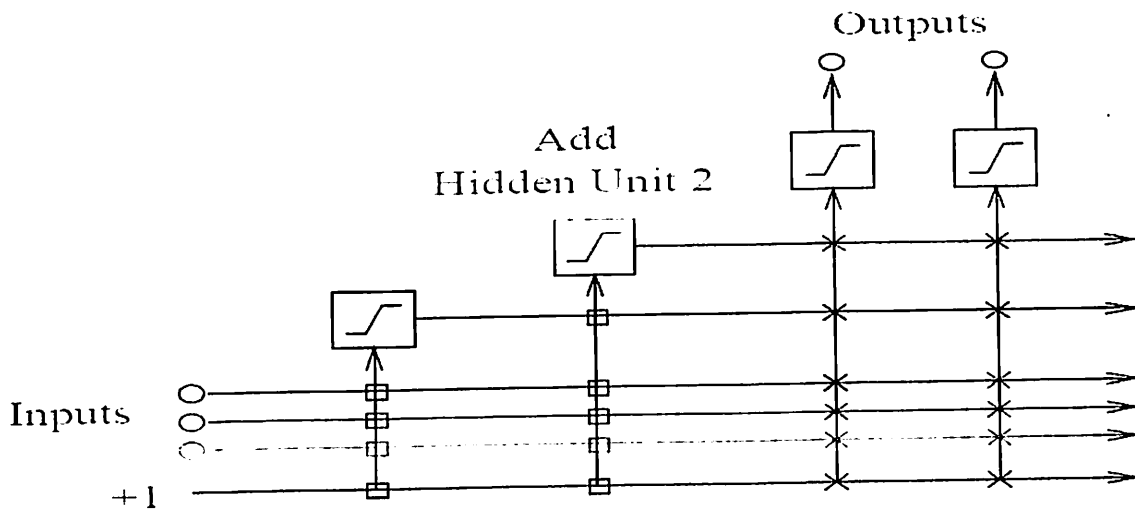
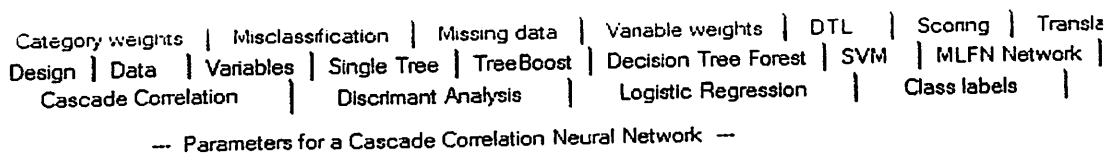


Figure 12-schematic of a network with two hidden neurons

The Cascade Correlation Network Property Page

Controls for cascade correlation analyses are provided on a screen in DTREG that has the following image:[23]



--- Parameters for a Cascade Correlation Neural Network ---

Type of model to build:

Hidden layer kernel functions:

Network training parameters

Minimum neurons:

Maximum neurons:

Candidate neurons:

Candidate epochs:

Output epochs:

Weight range:

Max. steps without improvement:

Train: Test:

Model testing and validation

No validation

Random percent:

V-fold cross-validation:

Leave-one-out validation

How to handle missing predictor variable values

Don't use rows with missing predictors

Replace missing predictors with medians

Overfitting protection control

Validate model as it grows

Hold-out sample %:

Cross validate: folds:

Prune model to optimal size

Options

Compute importance of variables

Advanced Parameters

Figure 13-Controls for cascade correlation analyses are provided on a screen in DTREG

5. EVALUATION STUDENT'S PERFORMANCE

Behavior problems in children, also known as child behavior disorders, are considered disruptive problems that have occurred for six months or more. Parents and caregivers need to know how to identify behavior problems in a child so they can seek professional treatment. Early intervention can prevent destructive habits, which may lead to failure in school, health problems or even suicide.

Now which behaviors are not behavior problems. Developmental stages, environments that are too stimulating or not stimulating enough, and new situations can all cause reactions in children that disrupt their healthy behavior. These are not to be confused with behavior problems.

Understand temperament. Sometimes temperament is confused with behavior problems in children. Temperaments include activity levels (some high, some low), approach to new stimuli (some adaptable, some withdrawn), intensity and mood (pleasant or unfriendly), attention span (concentration levels vary), and sensory threshold (some require little stimulation to elicit a response, others require more). Temperaments are neither good nor bad, but require parents and caregivers to reflect on children as individuals and respond to their needs accordingly.

If you're a teacher who is concerned about your students' poor performance on their standardized tests, you want to find creative and effective ways to help them improve. Find out what the student's weaknesses are. Discuss your concerns with the student and offer to give him extra assistance with this subject after school, during the lunch break or before school starts. It's important that you always show patience and compassion when helping your students, because this will give them the motivation and confidence which they need.

Teachers have various evaluation methods available to them for measuring students' performance in the classroom. Often the type of method used will depend on the content or task being measured, and whether it is a formal or informal evaluation. Evaluations should be fair and valid, taking into consideration the individual student's developmental level, learning disabilities and cultural context. They should also be based on clearly defined learning objectives. Effective evaluation measures the evidence a student presents to demonstrate her understanding of concepts or ability to perform tasks at a level deemed appropriate.

Write or adopt clearly defined learning objectives. Clear objectives state exactly what students will learn or be able to do, following instruction, and at what level of proficiency. Effective evaluation of students' academic ability relies on effective objectives to ensure that teachers are measuring students' learning and not something else. Researchers Grant Wiggins and Jay McTighe suggest that assessment should be based on the backward Design model, where teachers begin by establishing goals that determine the desired results of instruction, followed by establishing what criteria that will be accepted as evidence of these goals.

Write down the criteria students must demonstrate as acceptable evidence they have proficiently achieved the learning objectives. The criteria should be specific and descriptive. This step will help make designing an assessment or evaluation easier. It will also ensure that you are fairly assessing the students' knowledge based on what was taught, assuming that the instruction is also based on the learning objectives. Assess individual students' needs to ensure fair and valid evaluation. Identify learning disabilities or other disadvantages that should be accounted for in the evaluation process. Students with significant learning disabilities may need or qualify for alternate forms of assessment.

Administer regular assessment and evaluation methods. Wiggins and McTighe recommend assessing regularly throughout a unit, rather than relying on one assessment at the end of a unit. Regular assessment provides teachers a better understanding of student's performance over time and also provides students with continuous feedback and opportunity for improvement.

Score and grade assessments. Tests that have only one correct answer can be graded and scored easily from an answer sheet. Tests that ask open-ended questions or include other descriptive qualities can be scored using a rubric. A rubric establishes a measurement system that can be applied to measurements even though the answers are not criterion-referenced. Rubrics establish grading guidelines by assigning a point system for different levels of quality and demonstrated understanding[15].

5.1 Methodology of Research

This part presents the research methodology used in this research; starting from data collection and ending with the implementation of the proposed student.

5.1.1 Data collection

11th Grades students' data, including their marks at the end of the academic were collected as Excel Sheets. Data for the academic year 2012-2013 was available at the beginning of this research and had been collected; as shown in appendix.

5.1.2 Variables of Research (inputs and outputs)

Interviews with the students of Kazakh Turkish high school have been carried on to ;

Table 1- Short Notations

Variable	Notation
1-Do you live in dormitory?	Dormit.
2-Is your father alive?	Father
3-Is your mother alive?	Mother
4-Do they live together?	Together
5-During last 5 years,how many times teachers visit your home?	Visits
6-Did your teacher help you enough during your education in school?	T.help
7-Give a mark to your Class-teacher?	Std.Mark
8-Did you get your expectations in this school?	Expect.
9-What's your university exam result?(from 100)(Natianol University Exam)	N.U.E

Short notations are mentioned and will be used later.

Since there are many variables involved in the acceptance process,an approach is needed to forecast the performance.TANI is most convenient technique for this problem because of its ability in prediction problems based on its structure.

5.1.3 Application of forecasting results in Kazakh Turkish High School based on ANN

The model building process consist of four sequential steps;

1-selection of inputs and output variables for the Supervised Learning.

2-selection of activation function

3-trainin and validating the model

4-testing the goodness of the model

5.1.4 Selection of inputs and outputs for the TANI supervised learning

Program TANI runs with different set of input variables, these variables found to be the most important influencing the results at the end of the year. These inputs are ;dormitory, father,mother,together,student's mark,expectation and university exam result.

5.1.5 Activation function selection

In this step standart Sigmoid activation function is used and it's process with input variable ``x`` as follows:

$$1/(1+\exp(-x))-0,5$$

It takes the input and returns squashed output in the interval between 0 and 1. This is the most suitable to this research problem.

5.1.6 Training and Validating the model

Each supervised learning algorithm –trained using back-propagation. Each row represents a student and each column represents criteria. 20 rows are used for training, while 25 of rows are used for validating the model.

Training subset is the part of input data set used for neural network training to adjust weights. It is used to define the how many hidden units to detect the moment when predictive ability of neural network started to fail.

Test step is data set's part used to test how suitable the trained neural network forecasts on new data. We can describe this part as second step of ready (trained) network. It tests which errors will occur during future predictions. It is not used during training and thus consists of the new data entered by user for new prediction.

5.1.7 Trained Model's performance test

After creating the model through training and validating, model's goodness fitting is examined statistically. After each training, Mean Square Error (MSE) is calculated. The approximative model is identified as a best one if the MSE is enough small i.e. close to 0. The model with minimum MSE is identified as the best approximative model.

In next chapter application of Artificial Neural Networks models, results are shown.

6 APPLICATION OF FORECASTING STUDENTS PERFORMANCE USING ARTIFICIAL NEURAL NETWORK

Performance of students in Kazakh Turkish High School (KTL) after being accepted is the main research problem of this study. In previous chapters review about Artificial Neural Networks are explained and proposed solution of this problem. It is based on ANN and Artificial Intellegent (AI). Here researcether apply an ANN model to measure its effectiveness in decision making process for discovering their performance. It uses TANI to implement forecasting model. In this chapter how the chosen tool (A.N.N),is applicated forecasting model for the discovering performance are defined.

6.1 Application details and discussion of obtained results

This part contains the application of the used model in the research to predict the performance grades. Followings are a discussion for the obtained results.

Artificial Neural Networks model had been created based on student`s historical data and feelings. Variables which are input-output prepared as mother(0-1), father(0-1),together(0-1)dormitory(0-1),student`s mark(0-1),exam result(0-1) and teacher help(0-1) with their output low-middle-high performance is presented in one column.

Training Data	Load File	Reset Network	Iteration Count	Learning Rate	Train Network
			1000	0,95	Test Network
				Low	
				Middle	
				High	

Figure 14-Tani prediction program

After running program this table is appeared. At first there isn't any data on the screen. Then we add data by clicking 'load file' button. In this screen also iteration count and learning rate can be written manually. If iteration number is greater that means more training is done.

Figure show that the questions on survey which applied to the student

*In first column we want to learn where the student live during his education.

0,01 means:I don't live in dormitory

0,99 means:I live in dormitory regularly

0,5 means:sometimes I live I dormitory

*The second and third column represent the situation of the parents

0,01 means:he/she is not alive or they do not live together

0,99 means:they are alive or they live together

*In 5th column we want to learn relation between teachers and students so by visiting student's home how is it effective on performance of student. Each visit evaluated by 0.1.

* In 6th column how much teacher's help support the student, this is marked by student. They write percentage but it is normalized between (0,1).

* In 7th column Classroom Teacher was evaluated by his/her students as percentage and it is normalized between (0,1).

* in 8th column students evaluated their expectations which they got from school. Again they give points from 1 to 100 and it is normalized between (0,1).

*In last column, student's national university exam results are written. Also it was normalized between (0,1).

Table 3- input variables of students

	Do you live in dormitory?	Is your father alive?	Is your mother alive?	Do they live together?	During last 5 years, how many times teachers visit you	Did your teacher help you enough during your education?	Give a mark to your Class-teacher?	Did you get your expectations in this school?	What's your university exam result?(from 100)
Student 1	0,99	0,99	0,99	0,01	0,4	0,8	0,9	0,99	0,83
Student 2	0,99	0,99	0,99	0,99	0,2	0,7	0,8	0,7	0,76
Student 3	0,99	0,99	0,99	0,01	0,2	0,2	0,99	0,99	0,94
Student 4	0,99	0,99	0,99	0,99	0,3	0,99	0,99	0,99	0,92
Student 5	0,99	0,99	0,99	0,99	0,1	0,99	0,99	0,99	0,92
Student 6	0,99	0,99	0,99	0,01	0,2	0,6	0,99	0,99	0,92
Student 7	0,99	0,99	0,99	0,99	0,6	0,8	0,99	0,99	0,86
Student 8	0,99	0,99	0,99	0,99	0,1	0,99	0,99	0,99	0,97
Student 9	0,99	0,99	0,99	0,99	0,6	0,99	0,9	0,99	0,88
Student 10	0,99	0,99	0,99	0,99	0,3	0,8	0,99	0,99	0,86

First 10 student's data are written on excel sheet which is above. Also there is a table exist about this given data which is categorized as "about family" and "about school" at the below.

Table4- categorized variables

Student Name	About Family				About School				
	Mother	Father	Together	Visit	Exam	Stud.Mark	Dormitory	Expectation	Teac Help
Student 1	0,01	0,99	0,01	0,1	0,70	0,75	0,99	0,8	0,75

5 model student selected from list and their information are written in text document which is called ``test``. Attention was paid on there must be only one space between two data. The this file was loaded to Tanı forecasting program.

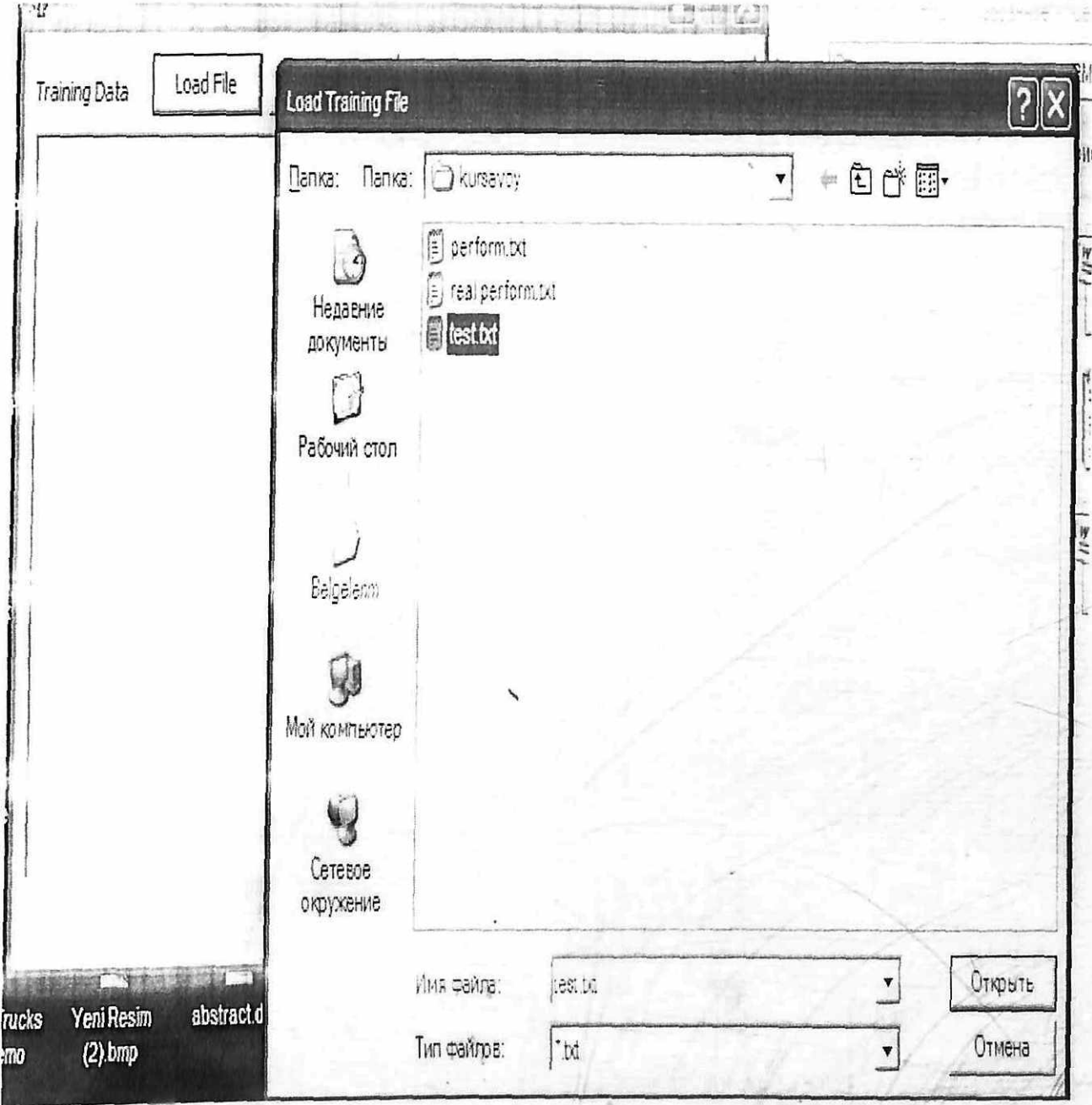


Figure 15-Addition of test file

TANI network forecasting model is illustrated in figure 2 shows the network model with 9 input variables, 5 model student's data, 1000 iteration and 0,95 learning rate.

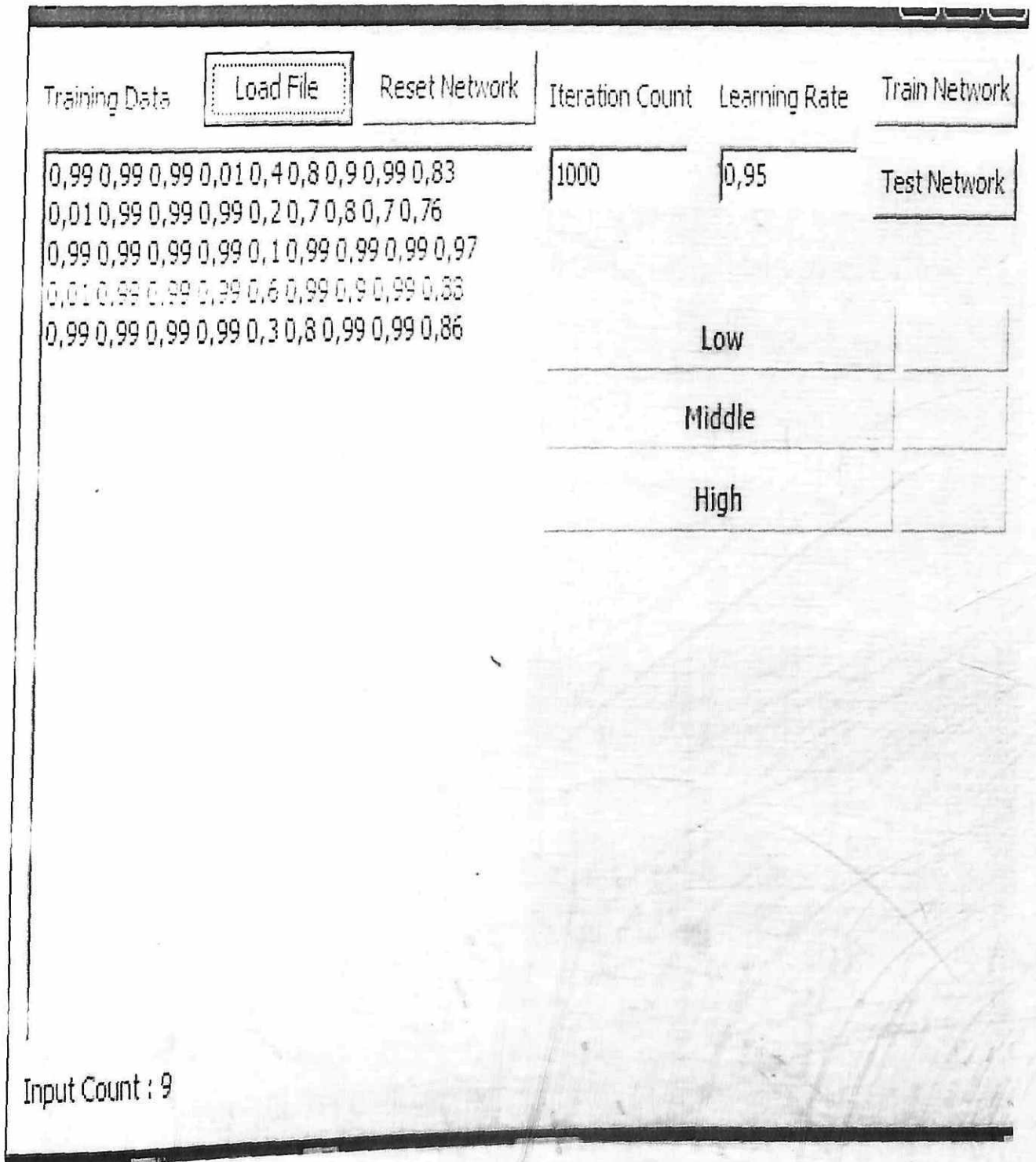


Figure 16-analizing model student

Artificial Neural Networks model had been created and data had been prepared for training the network in order to achieve the minimal possible error(0,000584452).

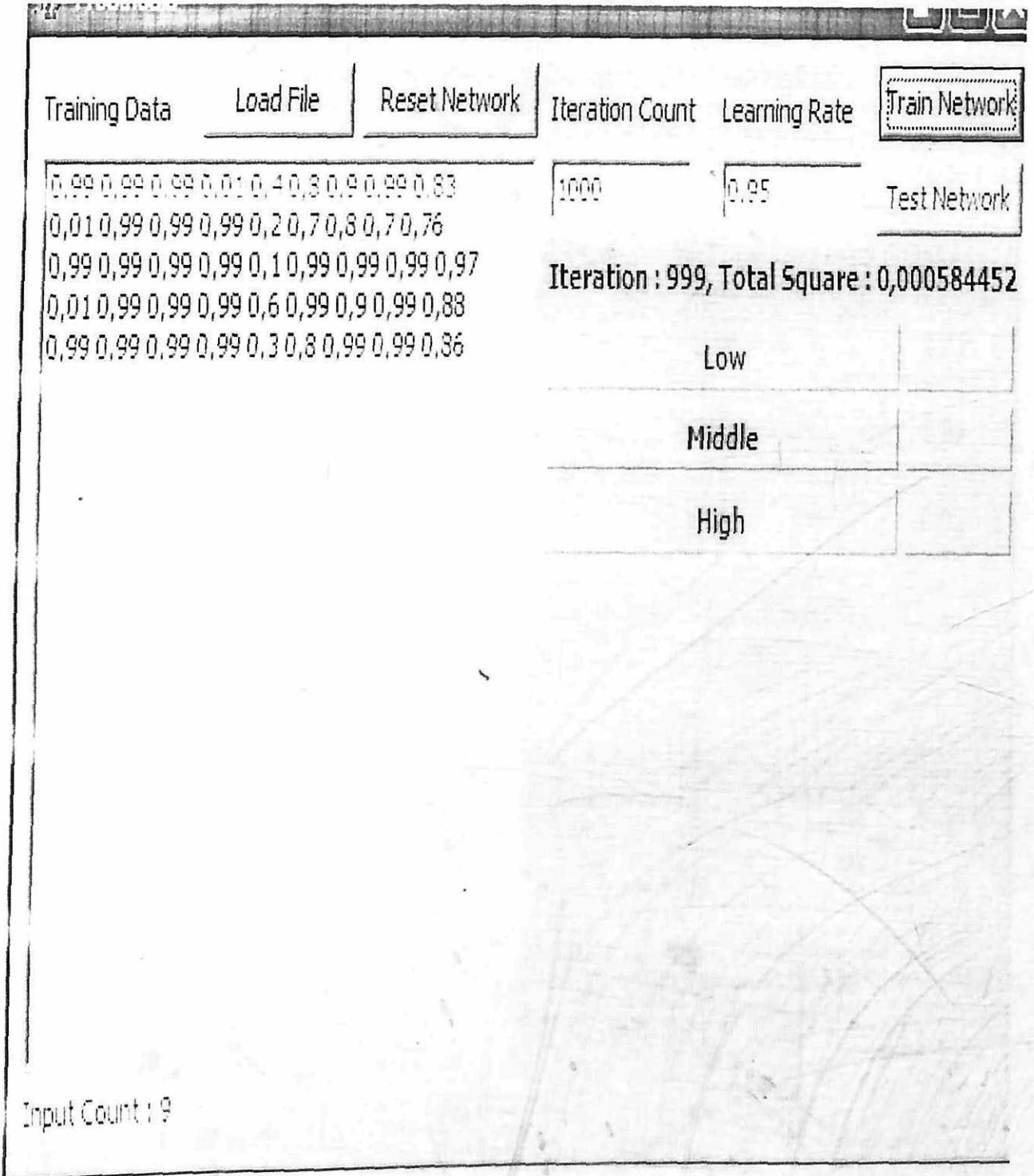


Figure 17-calculating total square error

One by one 5 student are trained and they are evaluated as high, middle and low performance. The scond student(row) is evaluated as middle(%97). In his first input he described he does not live in dormitory so this can be negative effect to his performance.

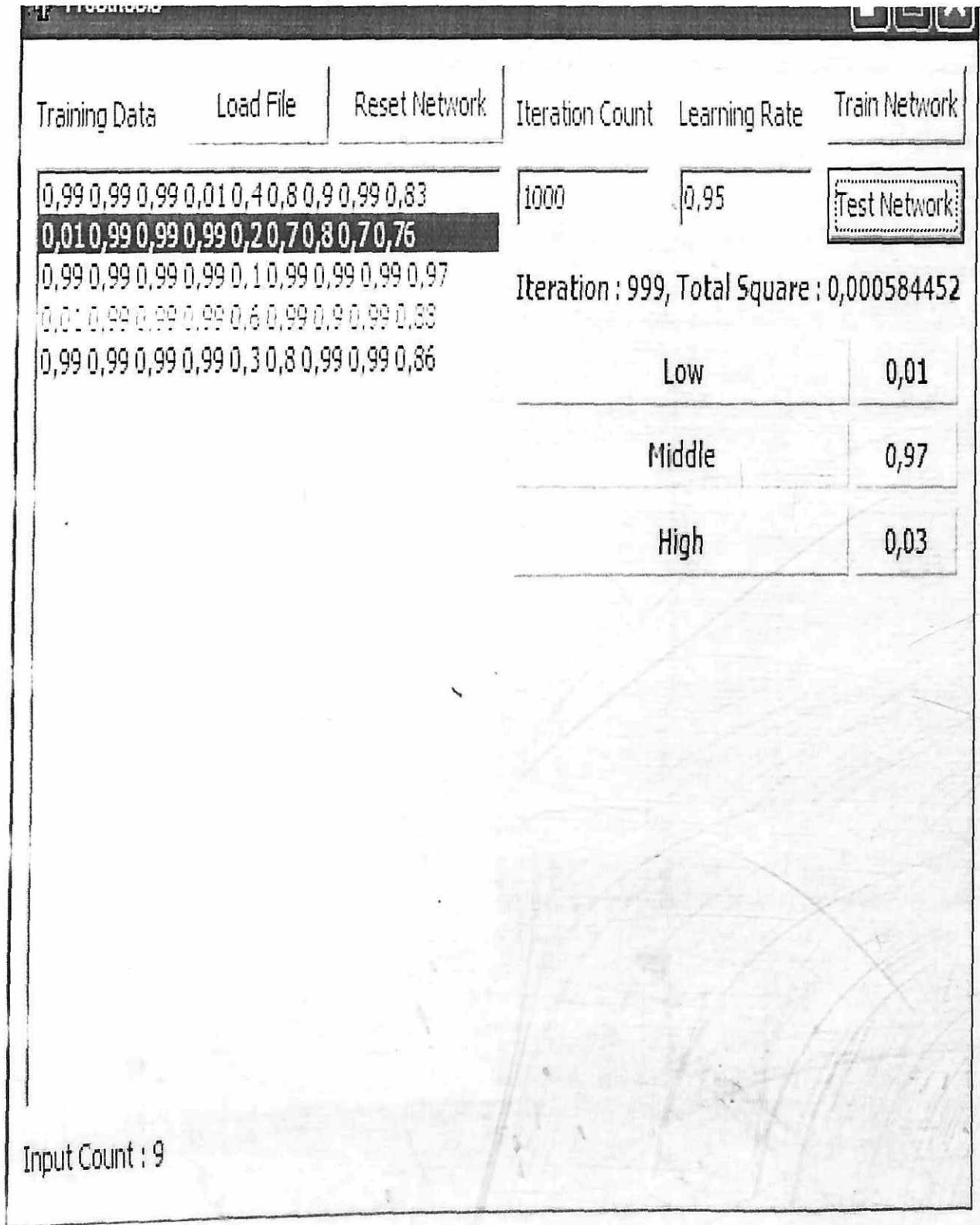


Figure 18-training model students

The last student(row) in the list show that high performance(%93). He lives in dormitory, his parent is alive, there are 3 home visit, %80 he took help from teachers, he gave %99 points to his teacher and he got %90 expectation from this school. As a result this student will be one of high performance model student in our forecasting network.

Training Data	Load File	Reset Network	Iteration Count	Learning Rate	Train Network
0,99 0,99 0,99 0,01 0,4 0,8 0,9 0,99 0,83			1000	0,95	<input type="button" value="Test Network"/>
0,01 0,99 0,99 0,99 0,2 0,7 0,8 0,7 0,76			Iteration : 999, Total Square : 0,000584452		
0,99 0,99 0,99 0,99 0,1 0,99 0,99 0,99 0,97					
0,01 0,99 0,99 0,99 0,6 0,99 0,9 0,99 0,88					
0,99 0,99 0,99 0,99 0,3 0,8 0,99 0,99 0,86					
				Low	0,05
				Middle	0,04
				High	0,93

Input Count : 9

Figure 19-high performance student

The first student(row)'s performance is presented "low"(%97). This student's data are seem normal but 4th column(parent live together) is 0,01. Parent don't live together so this criteria is negative effect to the student's performance. This student is one of low model student in our forecasting network.

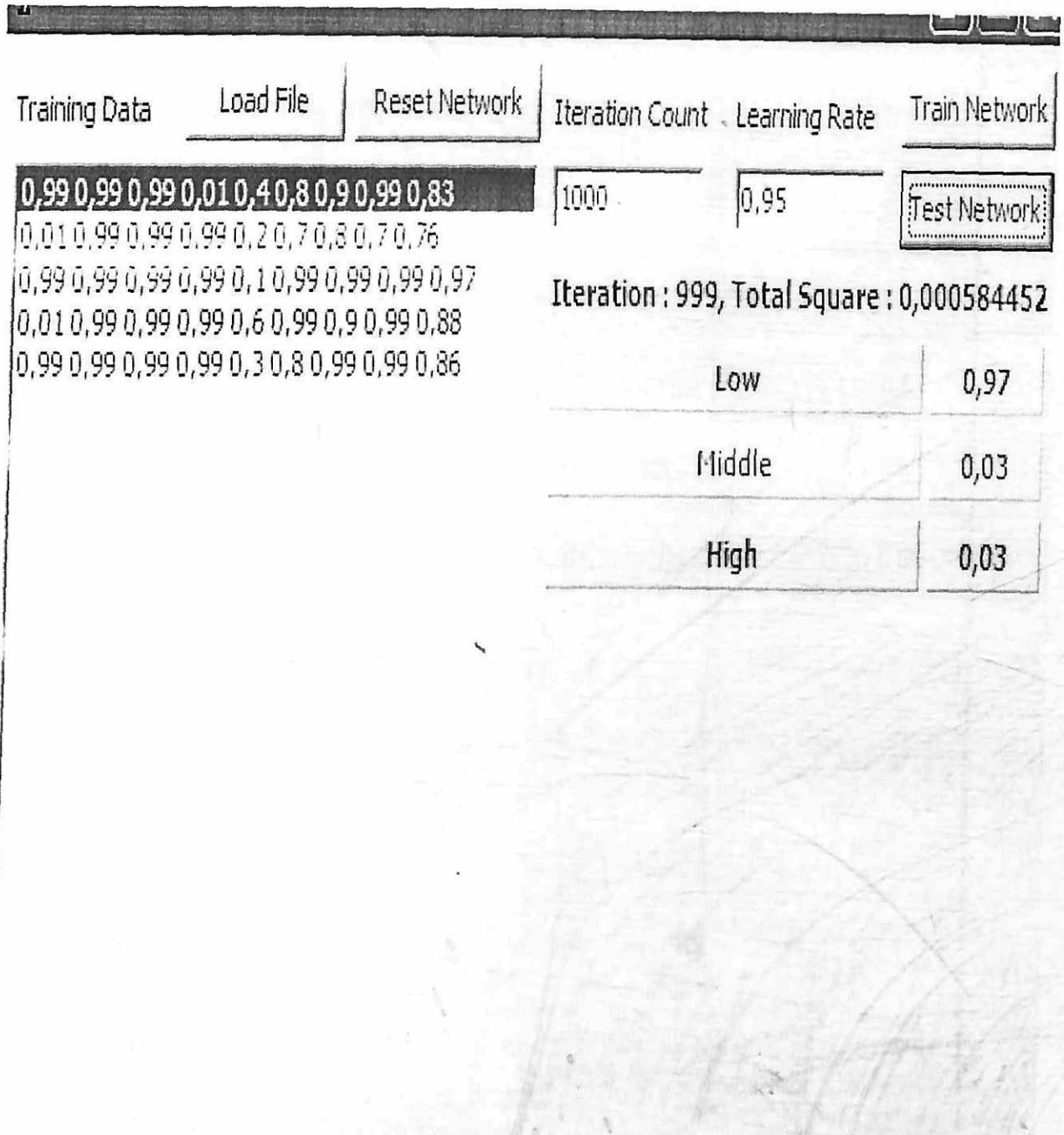


Figure 20-Low performance model

Predict on third student(row) is shown as high performance(%96). This one is also high performance model student for us. All of his entries are high value except 5th column(home visit). That means, if other inputs are high enough home visit does not effect performance very much.

Training Data	Load File	Reset Network	Iteration Count	Learning Rate	Train Network
0,99 0,99 0,99 0,01 0,40 0,80 0,90 0,99 0,83			1000	0,95	<input type="button" value="Test Network"/>
0,01 0,99 0,99 0,99 0,20 0,70 0,80 0,70 0,76					
0,99 0,99 0,99 0,99 0,10 0,99 0,99 0,99 0,97			Iteration : 999, Total Square : 0,000584452		
0,01 0,99 0,99 0,99 0,60 0,99 0,90 0,99 0,88					
0,99 0,99 0,99 0,99 0,30 0,80 0,99 0,99 0,86					
				Low	0,03
				Middle	0,03
				High	0,96

Input Count : 9

Figure 21-high performance model 2

At the end of the training of "test" models, without reset training second file was loaded to Tani. In second text file there are students which we want to evaluate their real performance. by use the previous total error square, iteration and learning rate other 20 student's performance will be predict.

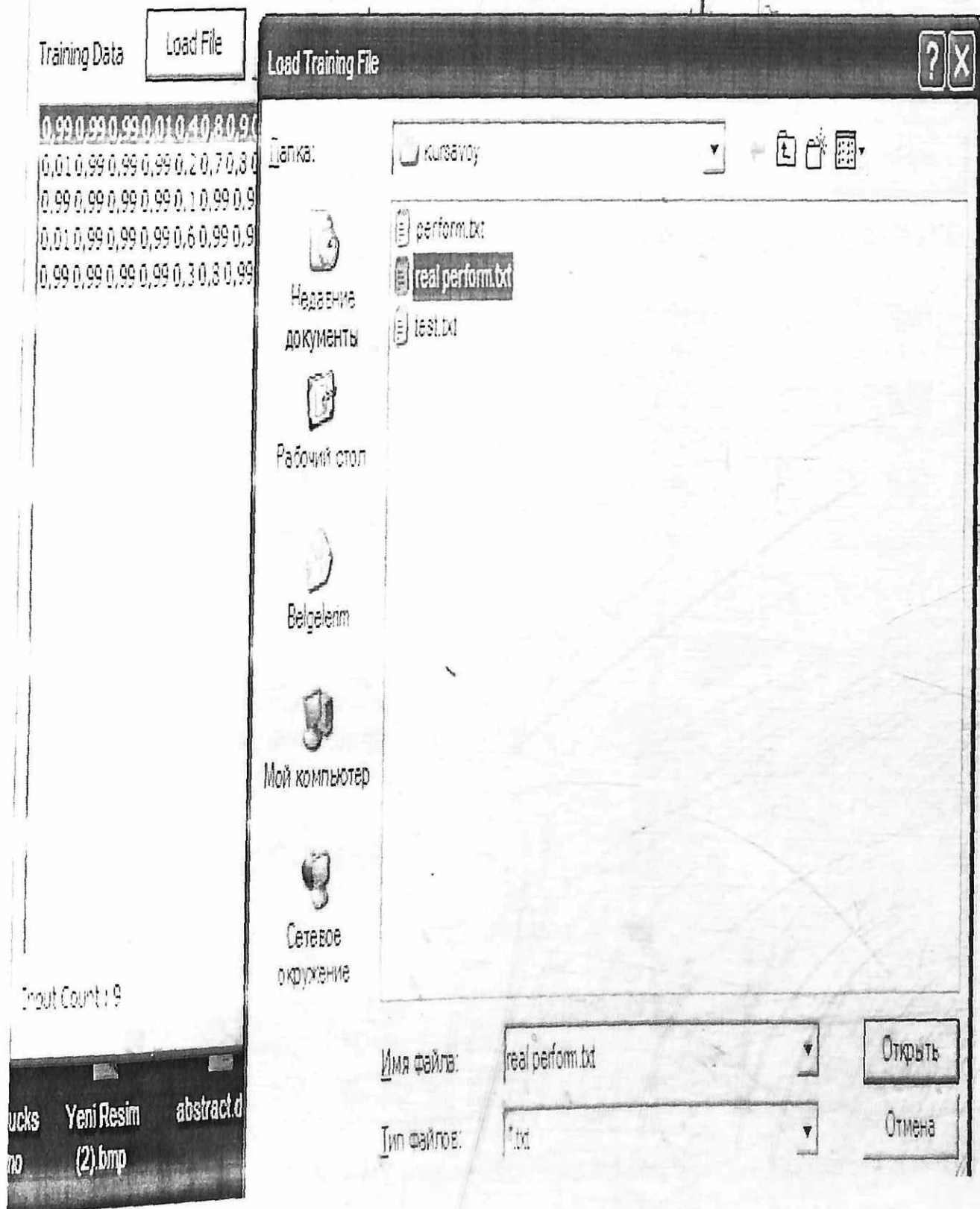


Figure 21-loading real performance file

The first student(row)'s result and previous student `s result who is trained before are close to each other (%96 and %95 high performance). Comparing their inputs show that the same column(home visit). This the proof of real prediction.

The screenshot shows a software interface for training a neural network. It includes several control buttons and a data display area.

Control Buttons: Training Data, Load File, Reset Network, Iteration Count, Learning Rate, Train Network.

Current Settings: Iteration Count: 1000, Learning Rate: 0,95. A "Test Network" button is also visible.

Training Data (Input Count: 9):

```

0,99 0,99 0,99 0,99 0,1 0,8 0,9 0,99 0,9
0,99 0,99 0,99 0,99 0,3 0,99 0,99 0,99 0,8
0,99 0,99 0,99 0,99 0,5 0,99 0,99 0,99 0,52
0,99 0,99 0,99 0,99 0,5 0,99 0,99 0,99 0,76
0,99 0,01 0,99 0,01 0,1 0,9 0,99 0,9 0,78
0,99 0,01 0,99 0,01 0,2 0,8 0,9 0,8 0,78
0,99 0,99 0,99 0,99 0,2 0,9 0,9 0,7 0,55
0,99 0,99 0,99 0,99 0,1 0,99 0,99 0,99 0,92
0,99 0,01 0,99 0,01 0,3 0,99 0,99 0,99 0,88
0,99 0,01 0,99 0,01 0,3 0,9 0,9 0,8 0,72
0,5 0,99 0,99 0,01 0,1 0,8 0,8 0,8 0,71
0,99 0,99 0,99 0,01 0,2 0,99 0,99 0,9 0,79
0,99 0,99 0,99 0,99 0,2 0,8 0,9 0,8 0,7
0,5 0,99 0,99 0,99 0,8 0,99 0,99 0,99 0,54
0,99 0,01 0,99 0,01 0,2 0,9 0,9 0,9 0,86
0,99 0,01 0,99 0,01 0,3 0,99 0,99 0,99 0,82
0,99 0,99 0,99 0,01 0,1 0,9 0,9 0,9 0,69
0,99 0,99 0,99 0,01 0,2 0,8 0,9 0,8 0,7
0,99 0,99 0,99 0,99 0,3 0,9 0,99 0,9 0,95
0,99 0,99 0,99 0,99 0,4 0,9 0,9 0,9 0,96
  
```

Performance Metrics Table:

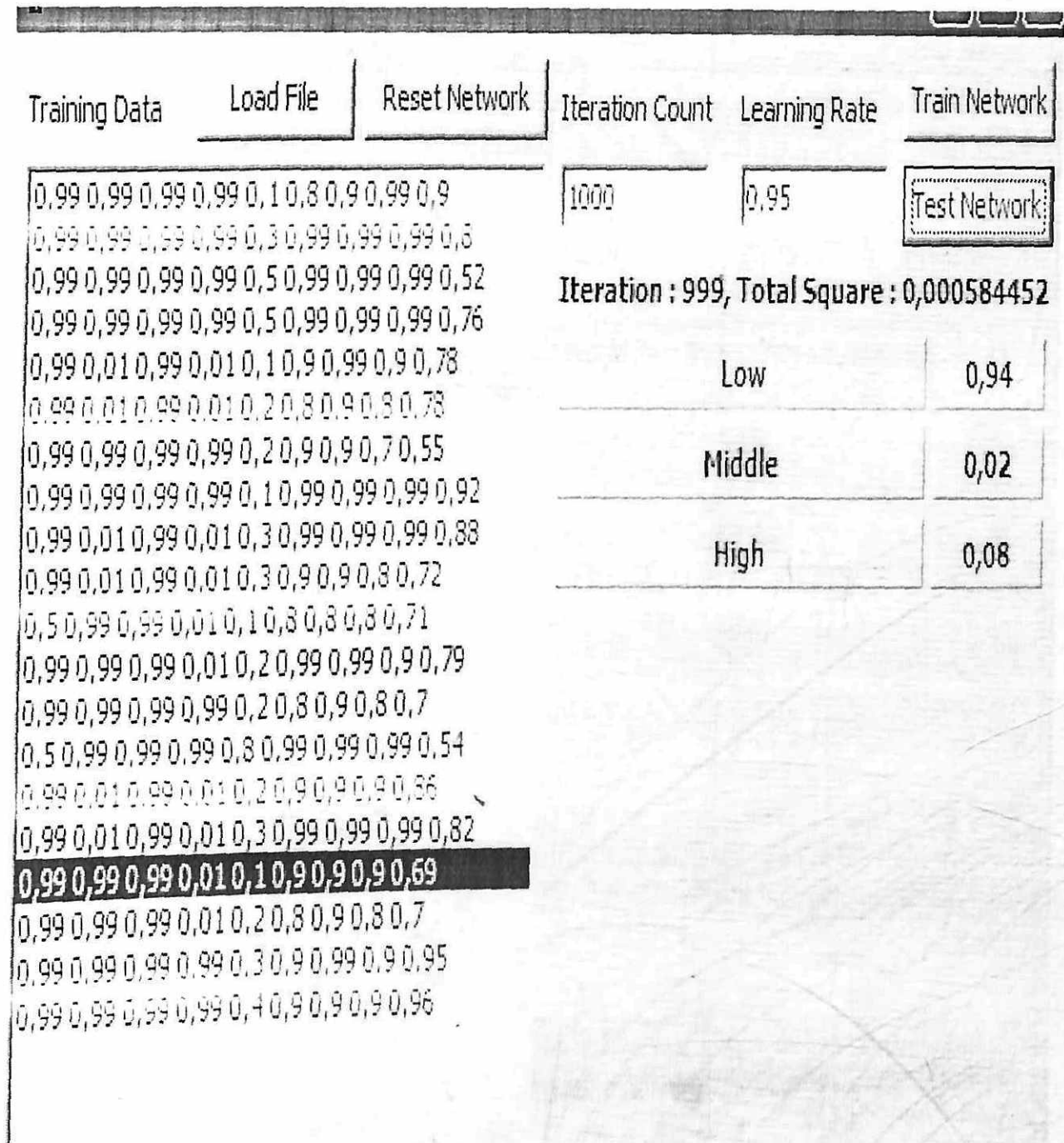
Low	0,03
Middle	0,04
High	0,95

Iteration Summary: Iteration : 999, Total Square : 0,000584452

Input Count: 9

Figure 22-prediction of high performance student

If you pay attention on 16th student values (dormitory 0,99-father 0,99-mother 0,99-together 0,01-visit 0,1-mark 0,9-expectation 0,9-and exam 0,69). Performance is low(%94). Cause of low performance can be divorced parent and home visit.



Input Count : 9

Figure 22-prediction of low performance student

By the same method other students performance can check one by one. Generally you get approximately real performance of students.

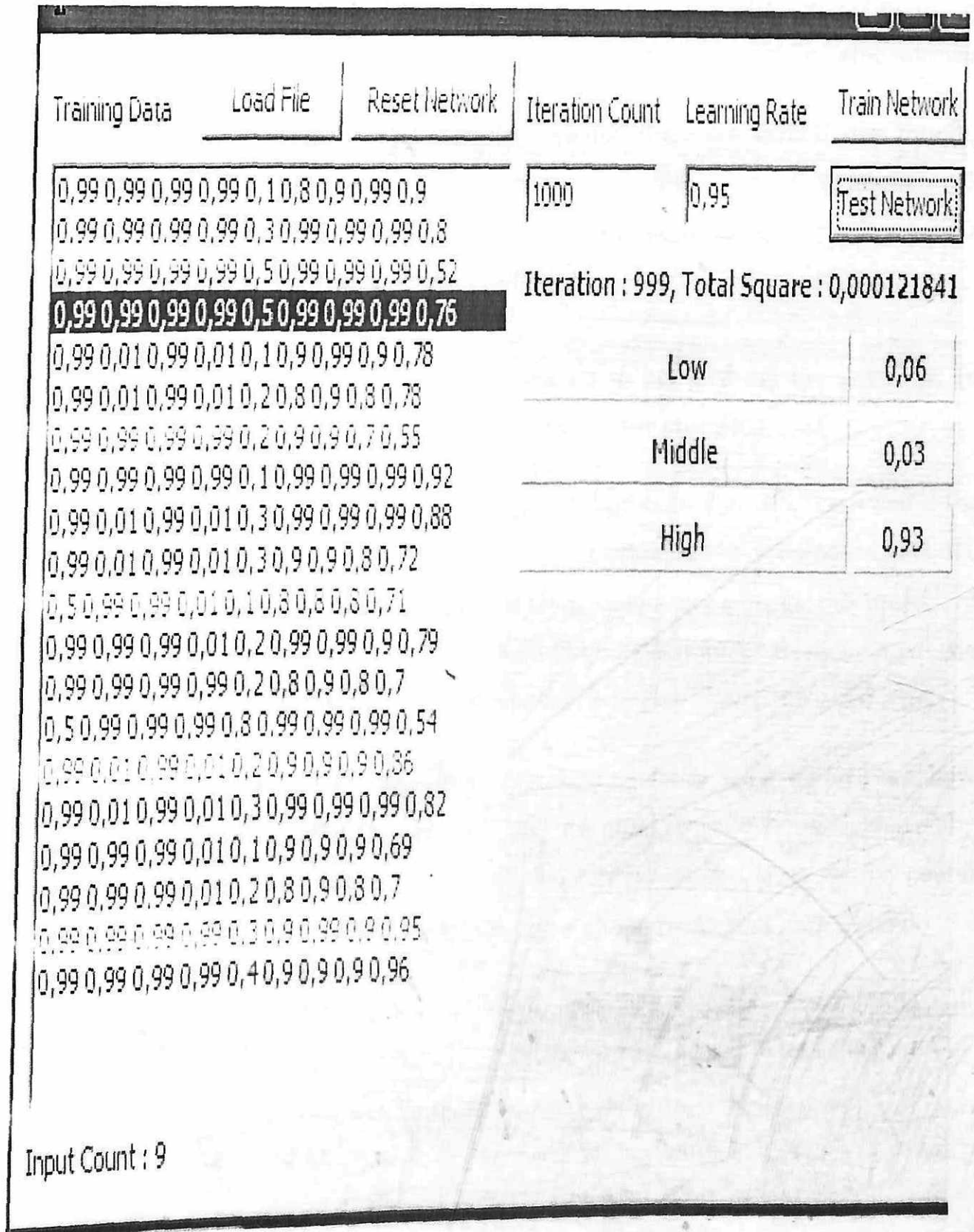


Figure 23-another prediction with smaller total square

Forecasting performance evaluation is a regression problem. A regression task is a task to predict the value of a continuous output parameter on basis of a certain input data set. Always the regression problem arises when predicting one or more than one values, which can continuously change within certain value interval.

While the network should stop the training, there are several stop training conditions to determine. If one of the next stop conditions are met, the training will be stopped. Four stopping conditions that could be set manually; error level, error change, generalization loss, number of iterations.

Error Level; describe the error level desired to achieve for the network. In this kind of problem, two parameters can be set for stopping aim;

(MSE)Mean Squared error: it is an absolute error measure that squares the difference between values from target output and actual network output to keep the positive and negative deviation from canceling each other out. by the square of each record and then added together and divide by the number of records we can calculate the MSE. It gives a number that summarizes the overall network error.

(AE)Absolute Error: Absolute error is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. Calculated by subtracting the current output values and the target output values of the neural network. The smaller network error is ,the better the network had been trained.

- i. **Error Change;** minimum approach in error. The training will be stopped when the network error change is less than the one specified during several last iterations.

- ii. **Number of iterations:** a single complete presentation of the training set to the neural network is called iteration.

- iii. **Generalization Loss:** Sometimes there will be over-training , training is unavailing and forecasting error definitely will be bigger. by stopping a network this parameter allows saving time and stopping network training when the validation error starts to rise. by use lower values there can be risk of the stopping too early when over-training only started and can be suppressed during subsequent iterations. On the other hand with higher values there is a risk of stopping after the network has already been over-trained to some extent and thus has worse forecasting error. For the best results experimenting with various datasets and networks can do this.

In the program the network will be trained automatically until stopping conditions satisfied. Minimum possible error is achieved, error reduction percentage becomes too low and generalization loss stays within acceptable limit.

The correlation coefficient is r^2 ; it measures the linear relationship between both of the desired output of the neural network and the actual output. The positively linear relationship exists when this value is close to '1'; if it is close to '-1', a negatively linear relationship exists.

The R^2 value is the coefficient of multiple determinations. The closer this value is to 1, the more the desired output from the neural network matches the actual output (Wang and Luxhoj 1999)

CONCLUSION

This research's main objective is impressiveness of Artificial Neural Networks in evaluating the performance of students. A study applied on Taldykorgan Kazakh Turkish High School students. Several studies and works about this subject analyzed and explained in this thesis. These work used Artificial Neural Networks in different real life forecasting problems.

By use Artificial Neural Network and survey on students, keeping in mind the particular problem in this study (evaluate performance of student), the model applied to forecast used the Tani which is trained using back-propagation algorithm. The aim of choosing it is that this network is a facility tool to find problem in prediction area. Our input variables are ;dormitory, parent's life, exam mark, student's mark to his teacher, expectation of student and teacher's support. The variables of output are high-middle-low performance grades. With different input variables different model had been created. The best accuracy is got with the five students' input variables(total square=0,000584452).

As an educator, you have a significant amount of influence over a student's disposition toward school. Students are naturally motivated in two ways, intrinsic and extrinsic motivation. Intrinsic motivation comes from an inner desire to do well or know more while extrinsic motivation stems from positive benefits that result from learning such as good grades or admiration. As you get to know your students, you can use a variety of methods to motivate them.

Create a supportive atmosphere that encourages struggling students. This increases student enjoyment and performance. Positive reactions to student questions, attentive listening and encouragement are ways to show support. Also recognize students who are performing below expectations and refer them for tutoring or remedial work.

Provide options to encourage students to take responsibility for learning. A student who has decision-making power feels an increased sense of motivation.

Provide tangible incentives that helps students answer the question: "what's in it for me?" Material items such as certificates, awards, prizes and cash provide extrinsic motivation for students. Tell your students that you have faith in them. Try to give your students challenging and achievable assignments to complete and provide them with the emotional support that they need to do well in class. And don't be afraid to set high expectations for your class. Constantly remind your students that you believe in them.

Encourage your students to participate in class. Some students enjoy participating in class activities and discussions, while others may shy away from the attention. However, it is important that all students have an opportunity to participate in some kind of way. So try going around the classroom and calling on students one by one to answer questions and to complete various tasks.

Reward students for doing a good job in class. Students love to get rewarded for their good work and behavior. You can start out by rewarding your students with simple verbal praise such as "great job" or "keep up the good work." There are also other ways to reward students, like giving them stickers, pencils, notepads, a set of crayons or a piece of gum (for small children). And for older students you may choose to give them gift certificates, books or exemptions from certain homework assignments[16].

All of this means that student cannot show his real performance, there are some criterions which are bad effect. For example his father dead this is negative effect. There have to be someone instead of father who takes his father role etc. Hold a meeting with the student's parents. Also answer any questions the parents may have and suggest practical methods they can use to help the student do better.

This is the most important point during our life. by the correction of this problem one person will be gained to the life with his real high performance.

REFERENCES

1. Eva Millan and Jose Luis Perez –De-La-Cruz ,A bayesian Diagnostic Algorithm for Student Modelling and its Evaluation
2. Zane Olina ,Howard J.Sullivan, Student Self-Evaluation ,Teacher-Evaluation and Learner Performance
- 3.<<http://moodle.org/>>
4. M.Delgado Calvo-Flores, E.Gibaja,M.C.Pegalajar Jimenez, and O.Perez Pineiro Predicting Student's marks from moodle logs using neural network models
5. Gabedi Nicholas Molefe,Performance measurement model and academic staff: A survey at selected Universities in South Africa and abroad pg 5251
- 6.Yusof,Nurazah , Evaluation of Student's Performance and Learning Efficiency based on ANFIS
7. Zhang, Yujia, A Self-Adaptive Monitoring and Analysis System for Students' behaviors in Laboratory Course
8. Christos Stergiou and Dimitrios Siganos, Neural Networks
- 9.http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Introduction to neural networks
10. Natasha Gilani, eHow Contributor, What Are the benefits of Artificial Neural Networks?
11. Carlos Mano, eHow Contributor, How to Use Artificial Neural Networks
12. Neural Networks Tutorial
eHow.com http://www.ehow.com/way_5817329_neural-networks-tutorial.html#ixzz2OrbUr8lv
- 13.Aleksander, I. and Morton. H. 2nd edition,An introduction to neural computing.

14. Neural Networks at Pacific Northwest National Laboratory

15. How to Measure Student Performance in a Classroom | eHow.com
http://www.ehow.com/how_10055960_measure-student-performance-classroom.html#ixzz2OtM7Dok3

16. http://www.ehow.com/how_4466239_motivate-students.html

17. Applying Multithreading to Resilient Propagation and Backpropagation

18. ISBN: 1-931841-08-X,

19. Stuart Russell and Peter Norvig. *Artificial Intelligence A Modern Approach*. p. 578. "The most popular method for learning in multilayer networks is called Back-propagation. It was first invented in 1969 by Bryson and Ho, but was largely ignored until the mid-1980s."

20. Arthur Earl Bryson, Yu-Chi Ho (1969). *Applied optimal control: optimization, estimation, and control*. Blaisdell Publishing Company or Xerox College Publishing. p. 481.

21. Paul J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974

22. Alpaydın, Ethem (2010). *Introduction to machine learning* (2nd ed. ed.). Cambridge, Mass.: MIT Press. p. 250. ISBN 978-0-262-01243-0. "...and hence the name *backpropagation* was coined (Rumelhart, Hinton, and Williams 1986a)."