Effect of Synthetic Emotions on Agents' Learning Speed and Their Survivability

Šarūnas Raudys

Vilnius Gediminas Technical University, Saulėtekio 11, Vilnius, Lithuania
raudys@ktl.mii.lt

Abstract. The paper considers supervised learning algorithm of nonlinear perceptron with dynamic targets adjustment which assists in faster learning and cognition. A difference between targets of the perceptron corresponding to objects of the first and second categories is associated with stimulation strength. A feedback chain that controls the difference between targets is interpreted as synthetic emotions. In a population of artificial agents that ought to learn similar pattern classification tasks, presence of the emotions helps a larger fraction of the agents to survive. We found that optimal level of synthetic emotions depends on difficulty of the pattern recognition task and requirements to learning quality and confirm Yerkes-Dodson law found in psychology.

Keywords: synthetic emotions, bio-inspired modeling, multi-agent systems, cognition, learning, neural networks, Yerkes–Dodson law.

1 Introduction

It is commonly admitted in psychology that emotions are an evolutionary mechanism important for learning and survival, adaptation, perception, evaluation, reasoning, memory and decision making [1- 5]. Simililar conclusions have been obtained while investigating "computer emotions" [6-9]. Some computer scientists claim that feedback chains could be interpreted as synthetic emotions. So far most work was performed to analyse symbolic reasoning based algorithms. It was found that feedback chains could be useful for faster learning and cognition [8-9].

One may hope that training speed of connectionist based learning systems is also affected by synthetic emotions. There exists a large amount of neural networks literature (mostly dating from the late 80s and early 90s) which addresses rates of learning with backpropagation algorithm. The speed of the algorithm can be significantly enhanced by using adaptive learning rate, η, and momentum [10- 12]. Dynamical change of η is widely used to control back propagation training process of multilayer perceptron (MLP) (see e.g. [13]). The training speed, a type of the classification rule obtained and generalization error can be affected also by input data scaling, preliminary data transformations and a noise injection to training input data (see e.g. reviews [14, 15, Chapter 4]). For a general introduction into artificial neural networks, training speed and a generalization error problems see e.g. [15 -17].

In [18] it was demonstrated that training speed of the single layer perceptron (SLP) based classifier depends on difference $s = |t_1 - t_2|$ between desired outputs, t_1, t_2,

M. Capcarrere et al. (Eds.): ECAL 2005, LNAI 3630, pp. 1 – 10, 2005.

corresponding to two diverse pattern classes. The training speed was measured as a number of training epochs required to train SLP. It was shown that while varying difference *s*, called stimulation, from 0 to 1 (if sigmoid activation function is used) the number of training epochs required to achieve a priori fixed classification error, P_{goal}, decreases at first, saturates and then starts increasing. It is Yerkes-Dodson (YD) law found in psychology [19, 20].

Up to now in psychology there is no unique definition of emotions. Significant role of emotions in evolution and adaptation suggests that there must be more than one mechanism for generating them [4]. From a variety of possible definitions of emotions, in present paper we relate synthetic emotions with training speed of the single layer perceptron used to solve pattern recognition task. We extend dynamic parameters' adjustment to target values and *associate synthetic emotions with dynamic changes in s, the stimulation*. We suppose that organization of the feedback chain assists in an increase in the stimulation if supervised learning was successful. We assume also that the feedback decreases the stimulation if learning was unsuccessful. This process may be called self-stimulation, reinforcement [16].

Our objective is not to investigate ways how to increase learning speed of back propagation algorithm. Our aim is to reveal principal mechanisms of interpretation of self-stimulation as synthetic emotions in connectionist learning systems. Global models could help in understanding factors affecting learning process in humans, societies and machines. They could offer to cognitive psychologists, sociologists and computer scientists one more model. In future they would assist in creating a variety of imitation situations for detailed studies of human and animal behaviors, improving strategies and algorithms to train single robots and groups of them.

To achieve this goal we look for the simplest model as possible. We selected a *nonlinear* SLP and a gradient descent supervised learning algorithm. In spite of simplicity of the mathematical model, in point of fact, a role of synthetic emotions on efficiency of connectionist learning systems was not considered so far.

The paper is organized as follows. Section 2 gives main terms and notations used in SLP training. In Section 3 we consider training algorithm with the feedback chain from a point of view of synthetic emotions: if classification error is decreasing, the stimulation level is increased, if classification error is increasing, stimulation is reduced. We show that such simple feedback chain, the synthetic emotions, support faster training. The dependence of training speed on level of emotions can be described by Yerkes–Dodson law [19, 20] too. Section 4 considers a situation where there are a large number of similar artificial agents that ought to learn changing pattern classification tasks. We show that presence of emotions helps larger fractions of agents to survive, i.e. to learn to solve the task in *a priori* fixed number of training iterations. Section 5 contains discussion and suggestion for future research work.

2 Training Peculiarities of the Single Layer Perceptron

Classical approach in adaptive learning is rooted in psychology, going back to early work of Thorndike [21] on animal learning and that of Pavlov [22] on conditioning. Here learning takes place through a process of punishment and reward with the goal of achieving a highly skilled behavior. In artificial intelligence, the learning is performed through continued interaction with the environment in order to minimize a scalar index of performance, a fitness (cost) function. To explain the principal trends, in present paper we start the analysis with the simplest model purposely. In order to explain a sense of our analysis, below we will present necessary definitions and show fundamental feature of gradient descent training: the weights of the single layer perceptron are increasing and gradually start slowing down the speed of training process. *The weights increase is principal importance and a novelty of our analysis.*

In our formulation, objects (situations) to be classified to one of the categories are described by input feature vectors, $\boldsymbol{x} = (x_1, x_1, \ldots x_p)$. The perceptron calculates a weighted sum of inputs, $sum = w_1{\times}x_1 + w_2{\times}x_2 + \ldots + w_p{\times}x_p + w_0$. A set of p values, w_1, $w_2, \ldots, w_p$, is called a weight vector, $\boldsymbol{w}$, and w_0 is a weight threshold value. We will use vector notation $sum = \boldsymbol{w}{\times}\boldsymbol{x} + w_0$. Very important essential of the perceptron is transfer function. Weighted sum, *sum*, is supplied to nonlinear element that calculates *output* of the perceptron as a non-linear function of *sum*. As an example one can consider sigmoid function, *output*=1/(1+exp (-*sum*)). If *sum* = 0, *output* = 0.5 (middle value). If *sum* is large negative, *output* is close to 0. If *sum* is large positive, *output* is close to 1. Note that a slope of function *output* = *f*(*sum*) is the highest where *sum*=0. If *sum* moves toward ± *infinity*, the slope diminishes and approaches zero [24, 25].

In order to use SLP practically one needs to know coefficients $w_0, w_1, \ldots, w_p$. To find the coefficients, we utilize training data called a training set, the vectors of the categories **A** and **B**: $\boldsymbol{x}_1^{(1)}, \boldsymbol{x}_2^{(1)}, \ldots, \boldsymbol{x}_{N_1}^{(1)}$ from **A** and $\boldsymbol{x}_1^{(1)}, \boldsymbol{x}_2^{(2)}, \ldots, \boldsymbol{x}_{N_2}^{(1)}$ from **B**. In perceptron training, we require that for class **A** *output* ought to be close to a priori selected target, t_1. For another class, **B**, we have to choose another value, e.g. t_2=1- t_1 [14, 15, 23]. Traditional algorithm used to train SLP is back propagation, where usually a sum of squares cost function, *cost*, is minimized,

$$cost = 1/N \sum_{i=1}^{2} \sum_{j=1}^{N_i} \left(t_j^{(i)} - f(\boldsymbol{w}{\times}\boldsymbol{x}_j^{(i)} + w_0) \right)^2, \tag{1}$$

where $\boldsymbol{w}$ is unknown p-variate weight vector, w_0 is a bias term, both to be found during training process, $t_j^{(i)}$ is a desired output (a target) of the perceptron if vector $\boldsymbol{x}_j^{(i)}$ is presented to its input.

In this paper we consider symmetric targets, $t_j^{(1)}$=0.5–0.5s, $t_j^{(2)}$=0.5+0.5s, $(0 < s \leq 1)$. If parameter s is close to 0, we have similar target values. If s is close to 1, targets $t_j^{(1)}$ and $t_j^{(2)}$ are close to boundary values of sigmoid activation function $f(sum)$=1/(1+exp(-sum)), i.e. 0 and 1. Thus, parameter s is interpreted as *the strength of training signal*, *the stimulation*. During training (adaptation) process, new vector, $\boldsymbol{w}_{(t+1)}$, is equal to the previous one, $\boldsymbol{w}_{(t)}$, plus a correction term:

$$\boldsymbol{w}_{(t+1)} = \boldsymbol{w}_{(t)} + CT_{ij}^{t}, \tag{2}$$

where $CT_{ij}^{t} = -\eta \times (t_j^{(i)} - f(sum) \times (\partial f(sum)/\partial sum) \times (\partial sum/\partial \boldsymbol{w})$ is the correction term, η is called learning rate (step) parameter, $t_j^{(i)} - f(sum)$ is an error signal – a difference between the desired and actual outputs of the perceptron, $sum = \boldsymbol{w}' \boldsymbol{x}_j^{(i)} + w_0$, $\partial f(sum)/\partial sum$ is the derivative of the activation function and (p+1)-dimensional vector $(\partial f(sum)/\partial sum)\times(\partial sum/\partial \boldsymbol{w})$ is called a gradient.

If the weights are small, the gradient, $(\partial f(sum)/\partial sum)\times(\partial sum/\partial \boldsymbol{w})$, is large. Simple algebra shows that when the weights are large, the gradient becomes small. During training, the magnitudes of the weighs are increasing and affect properties of the cost function [15, 25]. Moreover, with an increase in the magnitude of the weights, the gradient is decreasing towards zero. It means that in situations when the agent (the perceptron) has learned to solve its task properly and the weights are already large, *due to the large weights the perceptron is unable to re-learn a new task quickly*.

Two parameters, the learning rate parameter, η, and a difference between desired outputs, t_1, t_2, of the perceptron, $s = |t_1 - t_2|$ can be utilized to control the training process. We are increasing η or s by multiplying/dividing these parameters by positive scalar γ if training was successful/unsuccessful during n_{inertia} training epochs. We interpret parameter γ as *self-stimulation* or *synthetic emotions*. We remind that the self stimulation model is only one definition of emotions from a variety of possible ones. We will show that parameter γ affects training speed. Training speed is measured by a number of training epochs required to achieve a goal, P_{goal}, an *a priori* defined classification performance.

3 Influence of Self-stimulation on Speed of Training Process

In order to investigate the feedback chains, we consider simple adaptation model and perform simulation studies utilizing uncomplicated data – two bi-variate Gaussian classes with mean vectors $\mu_2 = - \mu_1$, unit variances and correlation between the variables ρ=0.7. In this model, three parameters control η and s: multiplication factor γ, sensitivity parameter Δ ($\Delta > 1$) and delay (inertia) n_{inertia} after which correction of η or s is made. The parameter, γ, indicates relative increase or decrease of learning step, η, or stimulation parameter, s, if training was effective or ineffective during n_{inertia} training epochs. Let $\psi = cost(t) / cost(t\text{-}n_{\text{inertia}})$ be a ratio of current cost (1) with previous cost value calculated n_{inertia} epochs before. Parameters η or s are multiplied by factor γ, if $\psi < 1/\Delta$. The parameters η or s are divided by factor γ, if $\psi > \Delta$. Otherwise, nothing is changed. In present paper, we report results obtained when $n_{\text{inertia}} = 1$ and Δ= 1.01. In analysis of dynamic learning step change, at start, we select initial learning step value, η_0.

Dynamic change of learning step, η. Results of the experiments with three P_{goal} values (0.003, 0.01, 0.03) and two starting learning step values, 0.1, and 125, are presented in Fig. 1 (stimulation s=1). Three graphs obtained for η_0=0.1 indicate that dynamic change of learning step, η, speeds up training process. In spite of the fact that we started training from small initial η value, feedback chain results that parameter η grows very quickly and compensates exponential decrease of the gradient caused by the gradual increase of the weights magnitudes.. If starting learning step is small (e.g. η_0=0.1), at the very beginning we do not have notable decrease of the cost (1). Due to high value of sensitivity parameter (Δ=1.01) parameter η remains unchanged for a long time. Training process remains very slow. If starting η value is too large (e.g. η_0=125), immediately after the first iteration we have very large change of the weight vector and a saturation of the cost function. Learning becomes slow and unstable. In certain cases, learning even stops (graphs 1*b* and 2*b* in Fig. 3).

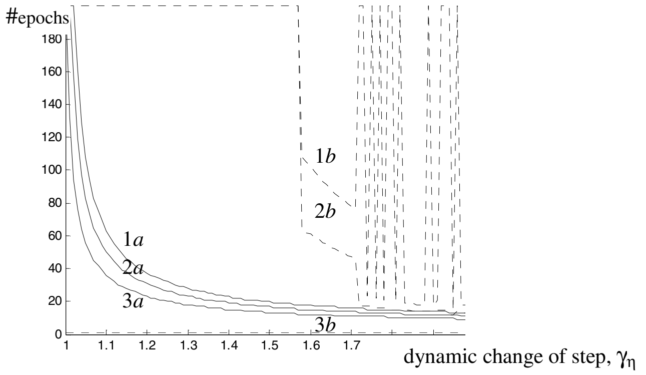

Fig. 1. A number of training epochs required to achieve P_{goal} as function of dynamic change of learning step: 1 - P_{goal} = 0.004; 2 - P_{goal} = 0.01; 3 - P_{goal} = 0.03. Almost non-overlapping classes, μ_1 = (0.6 2.4); $\boldsymbol{w}_{\text{start}}$ = [0.08 1 2]. Curves marked by "*a*" start training from η_0=0.1 (stable training) and that marked by "*b*" – start from η_0=125 (non-stable training).

We see that dynamic η change speeds up training process. In MLP training, it was used primary to help climb out from false local minimum [12, 13, 15, 17]. Dynamic η change is useful if *a priori* we do not know proper value of learning step parameter. Moderate dynamic η change assist in overcoming large weights effects if nonlinear soft-limiting activation function is used. We see also that sometimes dynamic η change could become dangerous. Therefore, we conclude that dynamic η change is suitable, however, sometimes imperfect model of synthetic emotions.

Dynamic change of stimulation, *s*. For the start, relatively small stimulation value (s_{start} = 0.002) was selected, i.e. t_1 = 0.499, t_2 = 0.501. In Fig. 2 we have a number of training epochs required to achieve P_{goal} as a function of parameter γ_s, dynamic change of stimulation strength (η=2.0; three values of P_{goal} and two values of starting weight vector, $\boldsymbol{w}_{\text{start}}$). In both experiments with different initial weights, position of starting decision boundary was the same, only magnitudes of the weights differed. To examine situations where large perceptron's weights start slowing down the training speed, almost non-overlapping pattern classes with small classification error were considered.

In second experiment, components of the initial weight vector, $\boldsymbol{w}_{\text{start}}$, w_0, were 1.5 times larger as in the first one. For that reason, at the very start we had larger sums, *sum*= $\boldsymbol{w}_{\text{start}} \times \boldsymbol{x}_j^{(i)} + w_0$. Consequently, the gradients turned out to be smaller at the very start of training. Consequently the training process became slower. Graphs in Fig 2*ab* indicate that in training with larger initial weights, we need higher number of training epochs (the learning task becomes more difficult).

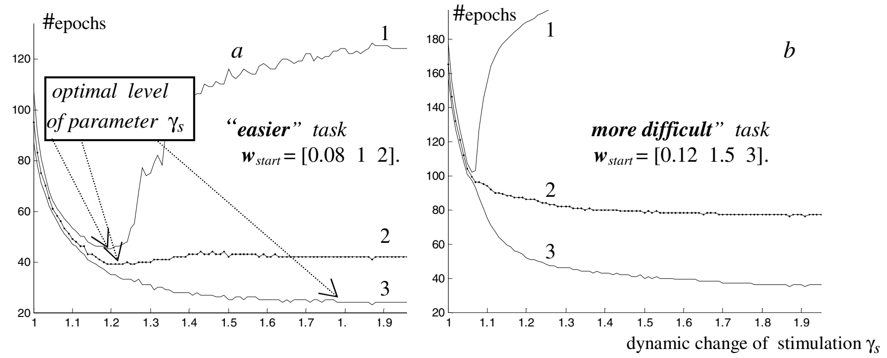

Fig. 2. A number of training epochs required to achieve P_{goal} as a function of dynamic change of stimulation strength, γ_s: 1 - P_{goal} = 0.004; 2 – P_{goal} = 0.01; 3 – P_{goal} = 0.03; Almost non-overlapping pattern classes, μ_1 = (0.6 2.4); *a*) – w_{start}= [0.08 1 2], *b*) – w_{start} = [0.12 1.5 3].

Graphs in Fig. 2 demonstrate that higher requirements to *learning quality* (smaller values of P_{goal}) necessitate higher number of training epochs. Both families of the curves indicate that *for each requirement for learning quality there exists an optimal level of parameter γ_s where training is fastest*. Both larger initial weights and smaller P_{goal} increase the difficulty of the task. We pay readers attention that *in difficult tasks, optimal values of parameter γ_s are smaller. In easier tasks optimal level of parameter* γ_s is *higher*. If the classes overlap notably (the classification task is more difficult), classification errors prevent excessive growth of the weights. The weights are smaller and the minima of the curves $\#_{epochs} = f(\gamma_s)$ are less expressed. (Fig. 3).

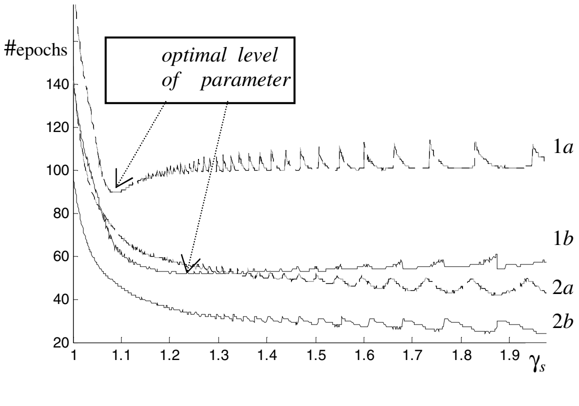

Fig. 3. A number of training epochs required to achieve P_{goal} as a function of dynamic change of stimulation, γ_s: 1 - P_{goal} = 0.08; 2 - P_{goal} = 0.16; two notably overlapping pattern classes: μ_1 = [0.3 1.2]: *a* - w_{start} = [0.12 1.5 3], *b* – w_{start} =[0.08 1 2].

4 Survivability of Population of Intellectual Agents

A natural question that arises while analyzing biological populations, social collectives and multi-agent systems in context of their evolution, is an influence of self-stimulation on survivability of the population. Consequently, we have to consider a situation where there are a large number of similar artificial agents that ought to learn new pattern classification tasks. *We assume that the agent passes away if it fails to learn fast enough to satisfy the a priori fixed condition* $P_{\text{classif}} < P_{\text{goal}}$ after the $t_{\max}$ *training epochs*. We show that presence of synthetic emotions helps larger fractions of agents to survive, i.e. to learn to solve the task rapidly. As a first step in the population analysis we will investigate many populations of agents having *different* self-stimulation parameter, γ_s. To have larger diversity of agents we assume that each population of agents is composed of r_f sub-families, f_r agents in each sub-family. Thus, in each population we have $r = f_r \times r_f$ agents. All agents in one sub-family possess *similar* characteristics, however, the families are to some extent *different*.

In experiments reported below, f_r =200*;* r_f = 6, r = 1200, the starting stimulation, s_{start} = 0.002, Δ = 1.01. We used two-dimensional Gaussian data with fixed mean vectors; $\mu_1 = -\mu_2 = [0.6\ 2.4]^T$; correlation ρ=0.7. Initial weights, learning step η, variances, σ_1, σ_2, of the single data components, however, were random variables: $\boldsymbol{w}_{start} = \tau \times [0.08\ 1\ 2]$, $\tau = (0.999+0.3\zeta)$, $\eta = 0.5+0.05\zeta$, σ_1, σ_2 =1+0.1ζ, where all ζ were independent random variables composed of sum of two random variables, ζ_A and ζ_B, distributed uniformly in interval [-0.15 0.15]. Variable ζ_A was individual for each single agent, while variable ζ_B was common to r_f agents in single sub-family.

In the Fig. 4*a* we have curves: a mean number of epochs required to achieve goal P_{goal} during $it_{\max}$ = 200 epochs versus self-stimulation parameter, γ_s. Curves 1 and 2 remind curves 1 and 3 in Fig. 2*a* obtained for one single agent. A main difference is a scale for the number of training epochs allowed to reach the goal. In Fig. 3, learning step was approximately four times smaller. Therefore, we needed approximately four times more epochs to achieve the goals.

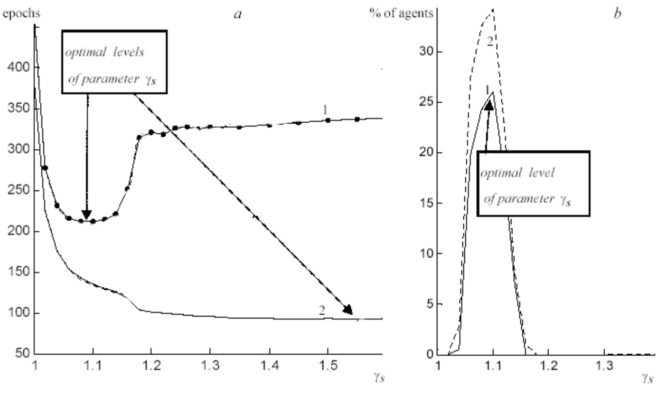

Fig. 4. *a*) Mean number of training epochs required to achieve P_{goal} as a function of dynamic change of self-stimulation, γ_s: 1 - P_{goal} = 0.004, 2 - P_{goal} = 0.03 during 200 training epochs. *b*) An effect of emotions ES on populations survivability (a fraction of agents which achieved desired classification error level, P_{goal}=0.004): 1 $it_{\max}$ = 200 epochs, 2 - $it_{\max}$ = 600 epochs.

In Fig. 4*b* we have “survivability graphs”, a fraction of agents (in percents) that succeeded achieve the goal, $P_{\mathrm{goal}} = 0.004$ during $it_{\max}$ training epochs. The graphs in 4*b* also demonstrate the Y-D Law and indicate an existence of optimal values of self-stimulation parameter, γ_s.

The graphs in Fig. 4*b* show that survival ratio crucially depends on a number of epochs allowed to achieve the goal, P_{goal}. If short training is available, in order to survive, optimal amount of emotions depends on P_{goal} notably. The stricter are requirements for learning quality (lower goal value), the smaller amount of emotions ES is necessary. For easier training tasks (high goal value, small initial weights), higher level of emotions is allowed.

5 Discussion and Suggestions for Future Research

One may assume that in process of species evolution, nature selected only these populations of individuals which are controlling the stimulation. This “self-stimulation phenomenon” we interpret as emotions in our study. To be more precise, it is only one aspect of emotions in this paper called “emotions ES” (emotions for stimulation). To imitate synthetic emotions we restricted ourselves with simplest model: the single layer perceptron trained by back propagation algorithm and feed-back chain utilized to modify stimulation values in dependence on a success in training. The new modification of the perceptron training algorithm adapts its target values during iterative training process: if classification error decreases for some time, the stimulation level is increased, if training becomes unsuccessful, the stimulation is reduced. Simulation studies demonstrate that values of feedback control of the targets affect the speed and performance of training process. Training scenario with emotions supports faster training. The dependence of training speed on level of emotions can be described by the Yerkes–Dodson law and has inverted letter “U” shape.

In analysis of situation where there is a large number of artificial agents that ought to learn different, subsequently changing pattern classification tasks, it was found that if short training is allowed, presence of certain amount of emotions helps larger fractions of agents to survive. This peculiarity of adaptive agent could be used in multi agent systems research.

Simulation analysis of the optimal level of self-stimulation on requirements to learning quality exhibit several features that match experimental findings in humans [1, 4, 5]. Often the people are changing their motivation in dependence on the success. Children are more emotional as the grown up persons. Populations which settle in warm, easy to live climate raise lower requirements to learning quality. So, they are more emotional than ones which live in severe northern climate. In this way, very simple model of dynamic target value control is one more possibility to elucidate theoretically these important observations. Possibly, our definition of synthetic emotions could indicate new ways to train robots, intellectual agents that have to adapt and to operate in unproblematic or more difficult environmental conditions. We have seen that dynamic change of learning step allows speed up training process notably. We investigated both, dynamic change of η and s. Dynamic target value control, however, is easier to interpret from a point of view of the emotions.

Striking conclusion derived from above analysis is that such simple element as single layer perceptron equipped with dynamic change of the difference between desired outputs could be interpreted as synthetic emotions which help the agents overcome difficulties that arise during training process. In the output layer of multilayer perceptrons, we also have SLPs trained by the same type of learning algorithm. One may hope that conclusions obtained for SLP classifier most likely are valid for MLP based learning systems. In principle, significantly larger number of means can be utilized to control perceptron's training process. These means include a noise injection to components of training vectors, a noise injection to desired targets, use of weight decay term for regularization, etc. [14, 15]. Obviously, these factors could be included into the emotions model to make it more suitable to analyze real world problems. Useful extension of the model is analysis of learning process utilized to solve not a single, but a variety of varying pattern recognition tasks, like it was done in a recent research papers on aging [25] and criminality [26]. In analysis of populations of intelligent agents, one can consider situations where successful members of the populations can produce offspring, a strategy widely widespread in Nature [26, 27]. Including of the factors just mentioned, organizing of the populations into sub-groups with mechanisms of self-support of the agents inside the sub-group and restricted beneficial cooperation between them, and incorporation of the mechanisms of synthetic emotions into learning rule is a topic for future research in order to crate more realistic multi agent systems capable to survive in permanently changing environments.

The investigation performed shows that connectionist approach to examine synthetic emotions can be effective enough to explain numerous phenomena observed in real life. Modifications of single layer perceptron based training model could be useful both for cognitive psychologists, sociologists, economists as a mean to investigate learning processes and to computer scientists which develop the algorithms capable adapt rapidly in unknown and everlastingly changing environments.

Acknowledgments

The author thanks Prof. Viktoras Justickis and Vanda Dovlias for useful and challenging discussions.

References

[1] Ortony, A., Clore, G., and Collins, A. (1988). *The Cogitive Structure of the Emotions*. New York: Cambridge University Press.

[2] Griffiths, P. (1997). *What Emotions Really Are: The Problem of Psychological Categories*. Chicago: Chicago University Press.

[3] Oatley, K., and Jenkins, J. (1996). *Understanding Emotions*. Oxford: Blackwell.

[4] Izard C.E. (1993) Four systems for emotion activation – cognitive and noncognitive processes. *Psychological Review*, 100 (1): 68-90.

[5] Lazarus, R.S. (1991). *Emotions and adaptation.* New York: Oxford University Press

[6] Goleman, D. *Emotional Intelligence: Why It Can Matter More than IQ*. London: Bloomsbury Publishing, 1996.

[7] Pfeifer, R. The "fungus eater approach” to emotion: a view from artificial intelligence. *Cognitive Studies, The Japanese Society for Cognitive Sci.,* 1:42-57, 1994.

[8] Chwelos G. and Oatley K. Appraisal, computational models, and Scherer expert system, *Cognition and Emotion*, 8(3): 245-257, 1994.

[9] Picard, R. W. *Affective Computing*. Cambridge: The MIT Press, 1997.

[10] Jacobs R. A. Increased rates of convergence through learning rate adaptation, *Neural Networks*, vol. 1(3): 295–307, 1988.

[11] Radi A. and Poli R. Genetic programming discovers efficient learning rules for the hidden and output layers of feedforward neural networks", In*: Genetic Programming, Proceedings of EuroGP'99"*, (Poli R and Nordin P. Eds.), Springer-Verlag, Vol. 1598 (26-27): 120-134, 1999.

[12] Yao X. Evolving artificial neural networks, *Proc. of IEEE*, 87(1): 1423-1447, 1999.

[13] [The MathWorks, *Matlab: The language of technical computing* (www.mathworks.com, 1998)

[14] Raudys S. Classifier's complexity control while training multilayer perceptrons. *Lecture Notes in Computer Science*. Springer-Verlag, 1876, 32-44, 2000.

[15] Raudys S. *Statistical and Neural Classifiers: An integrated approach to design*. Springer-Verlag. NY, 2001.

[16] Sutton R.S. and Barto A.G. *Reinforcement Learning: An introduction*. MIT Press,: Bradford Book, Cambridge, MA, 1998.

[17] Haykin S. *Neural Networks: A comprehensive foundation*. 2nd edition. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[18] Raudys S. and Justickis V. Yerkes-Dodson law in agents' training. *Lecture Notes in Artificial Intelligence,* Springer-Verlag, Vol. 2902, pp. 54-58, 2003.

[19] Yerkes, R. M. and Dodson, J.D. The relation of strength of stimulus to rapidity of habit-formation. *Journal of Comparative Neurology and Psychology*, 18:459-482, 1908.

[20] Teigen K.H. Yerkes-Dodson – a law for all seasons. *Theory and Psychology*, vol. 4 (4): 525-547. 1994.

[21] Thorndike, E. L. *Animal Intelligence*. Darien, Connecticut: Hafner, 1911.

[22] Pavlov I. P. New researches on conditioned reflexes, *Science*, 58:359-361, 1923.

[23] Rumelhart D.E., Hinton G.E. and Williams R.J. Learning internal representations by error propagation. In: D.E. Rumelhart, J.L. McClelland (editors), *Parallel Distributed Processing: Explorations in the microstructure of cognition* vol. I, pp. 318–62, Cambridge, MA: Bradford Books, 1986.

[24] Raudys S. Evolution and generalization of a single neurone. I. SLP as seven statistical classifiers. *Neural Networks*, 11(2): 283-296, 1998.

[25] Raudys S. An adaptation model for simulation of aging process. *Int. J. of Modern Physiscs, C.* 13(8): 1075-1086, 2002.

[26] Raudys S., Hussain A., Justickis V., Pumputis A. and Augustinaitis A. Functional model of criminality: simulation study. Proc. CONTEXT'2005 (Turner R., ed.), *Lecture Notes in Computer Science,* Springer-Verlag, 2005 (in Press).

[27] Raudys S. Survival of intelligent agents in changing environments. *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 3070, 109-117, 2004.