

Setting the Hidden Layer Neuron Number in Feedforward Neural Network for an Image Recognition Problem under Gaussian Noise of Distortion

Vadim Romanuke¹

¹ Applied Mathematics and Social Informatics Department, Khmelnytskyi National University, Khmelnytskyi, Ukraine

Correspondence: Vadim Romanuke, Applied Mathematics and Social Informatics Department, Khmelnytskyi National University, Institut'skaya str., 11, 29016, Khmelnytskyi, Ukraine. Tel: 380-972-895-239. E-mail: romanukevadimv@mail.ru

Received: January 18, 2013 Accepted: March 1, 2013 Online Published: March 10, 2013

doi:10.5539/cis.v6n2p38

URL: <http://dx.doi.org/10.5539/cis.v6n2p38>

Abstract

There is considered an image recognition problem, defined for the single hidden layer perceptron, fed with 5-by-7 monochrome images on its input under Gaussian noise of their distortion. In this neural network the hidden layer neuron number should be set optimally to maximize its productivity. For minimizing traintime duration and recognition error rate both simultaneously there are suggested two ways of solving the corresponding two-objective minimization problem. One of them deals with equilibrium conception, and the other takes Bernoulli criterion for getting the single minimization problem.

Keywords: image recognition, feedforward neural network, hidden layer neurons, traintime, neural network performance

1. A Problem of Setting Neural Network Architecture

Up-to-time neural networking solves a great many of practical tasks and problems, which cannot be solved with strict mathematical apparatus, because sometimes it takes much longer period to substantiate a case study mathematically, than just to train the appropriate architecture neural network and use it (Arulampalam & Bouzerdoum, 2003; Haykin, 1999; Kollias, 1996; Wöhler & Anlauf, 2001). Such training is nothing else but approximating that substantiation iteratively until the desired precision is reached. Nevertheless, the traintime may increase long as the problem dimensions for the neural network expand (Egmont-Petersen, de Ridder, & Handels, 2002; Haykin, 1999; Lo, Chan, Lin, Li, Freedman, & Mun, 1995). This can be fixed with adjusting the neural network architecture, where the adjustment is accomplished via selecting the architecture type (Aladag, 2011; Benardos & Vosniakos, 2007; Ciarelli, Oliveira, & Salles, 2012; Iannella & Back, 2001; Mahmoud & Ben-Nakhi, 2003), whereupon distributing the total of neurons over neural network layers (Han & Yin, 2008; Yuan, Xiong, & Huai, 2003; Zeng & Yeung, 2006). Up to now there have not been published any exact ways of setting the neural network architecture for solving a defined class problem. Though, the network type is determined easier than neurons over neural network layers are distributed.

2. Methods of Distributing Neurons over Neural Network Layers

It has practically been observed and grounded that a problem of the defined class is solved with the corresponding neural network architecture (Haykin, 1999; Iannella & Back, 2001; Lo et al., 1995). For instance, the image recognition problem is solved with the feedforward neural network of one or more hidden neuron layers (Goltsev & Gritsenko, 2012; Kollias, 1996; Wilson, Grother, & Barnes, 1996; Wöhler & Anlauf, 2001). This neural network, being the multilayer perceptron, is usually trained with the backpropagation algorithm, providing the shorter traintime duration (Hagan & Menhaj, 1994; J. Plaza, A. Plaza, Perez, & Martinez, 2009; Plumb, Rowe, York, & Brown, 2005; Yu, Manry, Li, & Narasimha, 2006). A hidden layer neuron has the number of inputs I_{hidden} , being equal to the number of elementary features of the image. The network output layer consists of Q_{output} neurons, where Q_{output} is the number of elements in the image alphabet. Therefore the number of neurons in the hidden layer q_{neuron} is expressed through neither image dimensions, nor the cardinality of set of all possible images at the neural network input. This number is assigned heuristically, basing

mostly on the experience (Haykin, 1999; Zeng & Yeung, 2006; Zhang, Ma, & Yang, 2003), for reaching satisfactory performance under the accepted gradient method of backpropagation algorithm implementation (Hagan & Menhaj, 1994; Plaza et al., 2009; Yu, Manry, Li, & Narasimha, 2006). Roughly, the assignment initial point q_{neuron} is selected agreeing the inequality (Kruglov, Dli, & Golunov, 2001) for number of training samples N_{train} .

$$\frac{N_{\text{train}}}{10} - I_{\text{hidden}} - Q_{\text{output}} \leq q_{\text{neuron}} \leq \frac{N_{\text{train}}}{2} - I_{\text{hidden}} - Q_{\text{output}} \quad (1)$$

Else, to take the number q_{neuron} initially there may be used the inequality (Kruglov, Dli, & Golunov, 2001) for synaptic weights number L_{weight} .

$$\frac{Q_{\text{output}} N_{\text{train}}}{1 + \log_2 N_{\text{train}}} \leq L_{\text{weight}} \leq Q_{\text{output}} \left(1 + \frac{N_{\text{train}}}{Q_{\text{output}}} \right) (1 + I_{\text{hidden}} + Q_{\text{output}}) + Q_{\text{output}} \quad (2)$$

Letting have another hidden layer neuron number assignment:

$$q_{\text{neuron}} = \frac{L_{\text{weight}}}{I_{\text{hidden}} + Q_{\text{output}}} \quad (3)$$

Wherethrough, if (3) is non-integer, q_{neuron} is taken as the nearest integer to (3). But heuristic estimations (3) by the inequality (2) may constitute very wide range for the same neural network, whereas hidden layer neuron number is the single value. The same consequence comes from the inequality (1). Besides, how-and-where-ever the number q_{neuron} is assigned, in (1) or (3), it is just only initial, and after it should be changed, decreased and increased, with observing the neural network performance (Benardos & Vosniakos, 2007; Mahmoud & Ben-Nakhi, 2003; Zeng & Yeung, 2006). Obviously, the best neural network performance indicates at the best number q_{neuron} . Contrariwise, this performance is usually reached after pretty long traintime duration that could have been shortened. And shortening the traintime duration is fulfilled by adjusting the number q_{neuron} . In fact, any attempts to improve the neural network performance and to shorten the traintime duration simultaneously are driven to appear confrontational: there is an interval of values of q_{neuron} , where the network performance improves and the traintime duration increases.

3. Objective of the Article

Let there be the feedforward neural network with the single hidden layer, destined for an image recognition problem. The objective of the present article is to determine theoretically such hidden layer neuron number that ensures both the network performance and traintime duration satisfactory. Also it should be demonstrated how to apply that theoretical method of setting the hidden layer neuron number for a practical case. To accomplish this there is going to be used the Matlab[®] environment (license 161051) with its powerful tool Neural Network Toolbox for constructing and testing neural networks (Ballabio & Vasighi, 2012; Kuzmanovski & Novič, 2008).

4. Setting Neurons into the Hidden Layer Theoretically

As the number of neurons in the neural network hidden layer depends on quality of its productivity then this number is optimal when it is the argument of the maximized productivity quality over the set of all tolerable hidden layer neuron numbers. Before maximization the explicit estimation of dependence of quality against hidden layer neuron number must be obtained. This is a hard enough task that may be fulfilled only in the process of the neural network functioning (Kordík, Koutník, Drchal, Kovářik, Čepék, & Šnorek, 2010; Plumb, Rowe, York, & Brown, 2005; Torrecilla, Otero, & Sanz, 2007), but it is a question for the next section. So now, assume that any needed estimation of neural network productivity exists and is stated as a mathematical expression. In an image recognition problem, quality of such productivity includes the traintime duration t_{train} and the neural network performance, measured as sum-squared error, mean-squared error or mean absolute error (Egmont-Petersen, de Ridder, & Handels, 2002; Haykin, 1999; Yu et al., 2006). Whatever the performance type is observed, it is actually the recognition error rate r_{er} .

Then, having nonnegative dependences $t_{\text{train}}(q_{\text{neuron}})$ and $r_{\text{er}}(q_{\text{neuron}})$ against the hidden layer neuron number, there are problems

$$\min_{q_{\text{neuron}} \in [q_{\text{min}}; q_{\text{max}}] \subset \mathbb{N}} \{t_{\text{train}}(q_{\text{neuron}})\} \quad (4)$$

And

$$\min_{q_{\text{neuron}} \in [q_{\text{min}}; q_{\text{max}}] \subset \mathbb{N}} \{r_{\text{er}}(q_{\text{neuron}})\} \quad (5)$$

for getting the optimal number

$$q_{\text{neuron}}^* \in \arg \min_{q_{\text{neuron}} \in [q_{\text{min}}; q_{\text{max}}] \subset \mathbb{N}} \{t_{\text{train}}(q_{\text{neuron}})\} \quad (6)$$

And

$$q_{\text{neuron}}^* \in \arg \min_{q_{\text{neuron}} \in [q_{\text{min}}; q_{\text{max}}] \subset \mathbb{N}} \{r_{\text{er}}(q_{\text{neuron}})\} \quad (7)$$

where q_{min} is the minimal hidden layer neuron number, and q_{max} is the maximal one. Here, if dependences $t_{\text{train}}(q_{\text{neuron}})$ and $r_{\text{er}}(q_{\text{neuron}})$ were both decreasing then there wouldn't have been any difficulties to determine q_{neuron}^* in (6) and (7) for ensuring the network performance value and traintime duration both sufficiently minimal. Only a specificity of problems (6) and (7) is that dependences $t_{\text{train}}(q_{\text{neuron}})$ and $r_{\text{er}}(q_{\text{neuron}})$ are put defined on the real range $[q_{\text{min}}; q_{\text{max}}]$, although they make sense only on the range $[q_{\text{min}}; q_{\text{max}}]$ of integers. This specificity is of fundamental importance when minimizing a function over the real range $[q_{\text{min}}; q_{\text{max}}]$ is transferred to minimizing the function over the integer range $[q_{\text{min}}; q_{\text{max}}] \subset \mathbb{N}$. In addition, it has to be accented what values of neural network productivity are between each couple of integers from the segment $[q_{\text{min}}; q_{\text{max}}]$, because the estimation is practically done for integers only. Henceforward, let every couple of integers from the segment $[q_{\text{min}}; q_{\text{max}}]$ in dependences $t_{\text{train}}(q_{\text{neuron}})$ and $r_{\text{er}}(q_{\text{neuron}})$ be connected linearly. So, these dependences are continuous piecewise linear functions, and such simplification is the best as any other interpolations are out of naturalness (Edwards, 1965).

In practice, the image recognition problem is solved in the presence of noise, meaning that images on the neural network input generally are different from original images. So, the dependence $r_{\text{er}}(q_{\text{neuron}})$ may be treated as the greatest recognition error rate (the lowest neural network performance) $\hat{r}_{\text{er}}(q_{\text{neuron}})$ within some domain of noise or as the mean recognition error rate $\bar{r}_{\text{er}}(q_{\text{neuron}})$ over that domain. Therefore, the problem (5) is branched out into important ones

$$\min_{q_{\text{neuron}} \in [q_{\text{min}}; q_{\text{max}}] \subset \mathbb{N}} \{\hat{r}_{\text{er}}(q_{\text{neuron}})\} \quad (8)$$

$$\min_{q_{\text{neuron}} \in [q_{\text{min}}; q_{\text{max}}] \subset \mathbb{N}} \{\bar{r}_{\text{er}}(q_{\text{neuron}})\} \quad (9)$$

Traintime duration may be presented in epochs, including total epochs from training procedure with passing under noise. However, for comparing the network properties, dependences $t_{\text{train}}(q_{\text{neuron}})$ and $r_{\text{er}}(q_{\text{neuron}})$ must be normalized:

$$\tau_{\text{train}}(q_{\text{neuron}}) = \frac{t_{\text{train}}(q_{\text{neuron}})}{\max_{q_{\text{neuron}} \in [q_{\text{min}}; q_{\text{max}}]} \{t_{\text{train}}(q_{\text{neuron}})\}} \quad (10)$$

$$\hat{\rho}_{\text{er}}(q_{\text{neuron}}) = \frac{\hat{r}_{\text{er}}(q_{\text{neuron}})}{\max_{q_{\text{neuron}} \in [q_{\text{min}}; q_{\text{max}}]} \{\hat{r}_{\text{er}}(q_{\text{neuron}})\}} \quad (11)$$

$$\bar{\rho}_{er}(q_{neuron}) = \frac{\bar{r}_{er}(q_{neuron})}{\max_{q_{neuron} \in [q_{min}; q_{max}]} \{\bar{r}_{er}(q_{neuron})\}} \quad (12)$$

Consequently, instead of the problems (4), (8), (9), the problems

$$\min_{q_{neuron} \in [q_{min}; q_{max}] \subset \mathbb{N}} \{\tau_{train}(q_{neuron})\} \quad (13)$$

$$\min_{q_{neuron} \in [q_{min}; q_{max}] \subset \mathbb{N}} \{\hat{\rho}_{er}(q_{neuron})\} \quad (14)$$

$$\min_{q_{neuron} \in [q_{min}; q_{max}] \subset \mathbb{N}} \{\bar{\rho}_{er}(q_{neuron})\} \quad (15)$$

with (10) - (12) are to be solved for getting the optimal hidden layer neuron number:

$$q_{neuron}^* \in \arg \min_{q_{neuron} \in [q_{min}; q_{max}] \subset \mathbb{N}} \{\tau_{train}(q_{neuron})\} \quad (16)$$

$$q_{neuron}^* \in \arg \min_{q_{neuron} \in [q_{min}; q_{max}] \subset \mathbb{N}} \{\hat{\rho}_{er}(q_{neuron})\} \quad (17)$$

$$q_{neuron}^* \in \arg \min_{q_{neuron} \in [q_{min}; q_{max}] \subset \mathbb{N}} \{\bar{\rho}_{er}(q_{neuron})\} \quad (18)$$

Statements (16) — (18) assemble the three-objective minimization problem, which generally has no solution even over the real range $[q_{min}; q_{max}]$. Hence, it has to be figured out how to select such $\tilde{q}_{neuron} \in \mathbb{N}$ that would impart sufficiently minimal recognition error rate to the neural network of its sufficiently short traintime duration, though it is not the argument, at which minima (13) — (15) are reached (in particular, they all are not reached or at least the one of them is not reached at that point). The number \tilde{q}_{neuron} is not necessarily to be close to q_{neuron}^* from (16) — (18).

Obviously, functions $\hat{\rho}_{er}(q_{neuron})$ and $\bar{\rho}_{er}(q_{neuron})$, reflecting the same network property, are similar, although they are not identical. So, it is comfortable to assign

$$\rho_{er}(q_{neuron}) \in \{\hat{\rho}_{er}(q_{neuron}), \bar{\rho}_{er}(q_{neuron})\}$$

whereby the performance is comprehended.

If $\tau_{train}(q_{neuron})$ and $\rho_{er}(q_{neuron})$ are both strictly monotonic functions, then one of them is strictly decreasing and the other is strictly increasing, or both are strictly decreasing or strictly increasing simultaneously by $q_{neuron} \in [q_{min}; q_{max}]$. Obviously, the same assertion is true for continuous piecewise linear functions. If they both are strictly decreasing or strictly increasing simultaneously, then $q_{neuron}^* = q_{max}$ if decreasing, and $q_{neuron}^* = q_{min}$ if increasing. If one of them is strictly decreasing and the other is strictly increasing then there is a local theorem.

Theorem 1. If continuous piecewise linear functions $\tau_{train}(q_{neuron})$ and $\rho_{er}(q_{neuron})$, whose linear pieces are defined on integer-endpoint subsegments of $[q_{min}; q_{max}]$, are both strictly monotonic by $q_{neuron} \in [q_{min}; q_{max}]$, and one of them is decreasing and the other is increasing, then exists such $\tilde{q}_{neuron} \in [q_{min}; q_{max}]$, being at that single, that

$$\tau_{train}(\tilde{q}_{neuron}) = \rho_{er}(\tilde{q}_{neuron}) \quad (19)$$

If in the case of that $\tilde{q}_{neuron} \notin \mathbb{Z}$ this point belongs to the interval $(q; q+1)$ by $q \in \mathbb{N}$ then for $\tilde{q}_{neuron} = q + \frac{1}{2}$

there is the equality

$$\rho_{er}(q) - \tau_{train}(q) = \tau_{train}(q+1) - \rho_{er}(q+1) \quad (20)$$

For $\tilde{q}_{neuron} < q + \frac{1}{2}$ there is the inequality

$$|\rho_{er}(q) - \tau_{train}(q)| < |\tau_{train}(q+1) - \rho_{er}(q+1)| \quad (21)$$

and for $\tilde{q}_{neuron} > q + \frac{1}{2}$

$$|\rho_{er}(q) - \tau_{train}(q)| > |\tau_{train}(q+1) - \rho_{er}(q+1)| \quad (22)$$

Proof. Assume that

$$\tau_{train}(q_{neuron}) \neq \rho_{er}(q_{neuron}) \quad \forall q_{neuron} \in [q_{min}; q_{max}]$$

Then

$$\tau_{train}(q_{neuron}) > \rho_{er}(q_{neuron}) \quad (23)$$

or

$$\tau_{train}(q_{neuron}) < \rho_{er}(q_{neuron}) \quad (24)$$

For definiteness let it be (23). Consequently, if $\tau_{train}(q_{neuron})$ is strictly decreasing and $\rho_{er}(q_{neuron})$ is strictly increasing then

$$\tau_{train}(q_{min}) > \rho_{er}(q_{max}),$$

what is false as

$$\tau_{train}(q_{min}) = \rho_{er}(q_{max}) = 1$$

due to normalizing in (10) - (12). And if $\tau_{train}(q_{neuron})$ is strictly increasing and $\rho_{er}(q_{neuron})$ is strictly decreasing then

$$\tau_{train}(q_{max}) > \rho_{er}(q_{min}),$$

what is false also as

$$\tau_{train}(q_{max}) = \rho_{er}(q_{min}) = 1$$

due to normalizing in (10) - (12). So, there exists such $\tilde{q}_{neuron} \in [q_{min}; q_{max}]$ that (19) is true. Now, will prove the uniqueness of the point $\tilde{q}_{neuron} \in [q_{min}; q_{max}]$. Assume that exist another $\tilde{q}_{neuron}^* \in [q_{min}; q_{max}]$ such that

$$\tau_{train}(\tilde{q}_{neuron}^*) = \rho_{er}(\tilde{q}_{neuron}^*) \quad (25)$$

along with (19). For definiteness let it be $\tilde{q}_{neuron}^* > \tilde{q}_{neuron}$. Then for strictly decreasing function $\tau_{train}(q_{neuron})$

and strictly increasing function $\rho_{er}(q_{neuron})$

$$\tau_{train}(\check{q}_{neuron}) > \tau_{train}(\check{q}_{neuron}^*)$$

and

$$\rho_{er}(\check{q}_{neuron}) < \rho_{er}(\check{q}_{neuron}^*),$$

that is

$$\tau_{train}(\check{q}_{neuron}^*) < \tau_{train}(\check{q}_{neuron}) = \rho_{er}(\check{q}_{neuron}) < \rho_{er}(\check{q}_{neuron}^*) \quad (26)$$

So, (26) is contradictory to (25), and $\check{q}_{neuron} \in [q_{min}; q_{max}]$ is unique. Clearly, taking (24), or $\check{q}_{neuron}^* < \check{q}_{neuron}$, the symmetric contradiction appears. Hence, (19) is true for the single $\check{q}_{neuron} \in [q_{min}; q_{max}]$.

Now consider how functions $\tau_{train}(q_{neuron})$ and $\rho_{er}(q_{neuron})$ differ in the integer neighborhood of point $\check{q}_{neuron} \notin \mathbb{Z}$. In the case $\check{q}_{neuron} \in (q; q+1)$ by $q \in \mathbb{N}$ these functions may be stated as $\tau_{train}(q_{neuron}) = \alpha_{\tau}q_{neuron} + \beta_{\tau}$ and $\rho_{er}(q_{neuron}) = \alpha_{\rho}q_{neuron} + \beta_{\rho}$ at $\check{q}_{neuron} \in [q; q+1]$ for some real coefficients α_{τ} , β_{τ} , α_{ρ} , β_{ρ} .

For $\check{q}_{neuron} = q + \frac{1}{2}$ there are

$$\tau_{train}(q) + \tau_{train}(q+1) = \alpha_{\tau}q + \beta_{\tau} + \alpha_{\tau}(q+1) + \beta_{\tau} = \alpha_{\tau}(2q+1) + 2\beta_{\tau} \quad (27)$$

$$\rho_{er}(q) + \rho_{er}(q+1) = \alpha_{\rho}q + \beta_{\rho} + \alpha_{\rho}(q+1) + \beta_{\rho} = \alpha_{\rho}(2q+1) + 2\beta_{\rho} \quad (28)$$

And

$$\tau_{train}\left(q + \frac{1}{2}\right) = \alpha_{\tau}\left(q + \frac{1}{2}\right) + \beta_{\tau} = \alpha_{\rho}\left(q + \frac{1}{2}\right) + \beta_{\rho} = \rho_{er}\left(q + \frac{1}{2}\right) \quad (29)$$

using (19). From (29) it follows that

$$\alpha_{\tau}(2q+1) + 2\beta_{\tau} = \alpha_{\rho}(2q+1) + 2\beta_{\rho} \quad (30)$$

where (30) means the equality of (27) and (28):

$$\rho_{er}(q) + \rho_{er}(q+1) = \tau_{train}(q) + \tau_{train}(q+1) \quad (31)$$

Clearly, the equality (31) is identical to (20).

If $\alpha_{\tau} > 0$ then $\alpha_{\rho} < 0$ and for $\check{q}_{neuron} < q + \frac{1}{2}$ there are

$$\tau_{train}(q) + \tau_{train}(q+1) = \alpha_{\tau}(2q+1) + 2\beta_{\tau} > 2\alpha_{\tau}\check{q}_{neuron} + 2\beta_{\tau} = 2\tau_{train}(\check{q}_{neuron}) \quad (32)$$

$$\rho_{er}(q) + \rho_{er}(q+1) = \alpha_{\rho}(2q+1) + 2\beta_{\rho} < 2\alpha_{\rho}\check{q}_{neuron} + 2\beta_{\rho} = 2\rho_{er}(\check{q}_{neuron}) \quad (33)$$

whence due to (19), uniting inequalities (32) and (33), the inequality

$$\tau_{\text{train}}(q) + \tau_{\text{train}}(q+1) > \rho_{\text{er}}(q) + \rho_{\text{er}}(q+1) \quad (34)$$

turns true. And due to $\alpha_{\tau} > 0$, meaning that $\tau_{\text{train}}(q_{\text{neuron}})$ increases against decreasing $\rho_{\text{er}}(q_{\text{neuron}})$ at $q_{\text{neuron}} \in [q; q+1]$, the inequality (34) may be re-stated as

$$\tau_{\text{train}}(q+1) - \rho_{\text{er}}(q+1) > \rho_{\text{er}}(q) - \tau_{\text{train}}(q) > 0 \quad (35)$$

Otherwise, if $\alpha_{\tau} < 0$ then $\alpha_{\rho} > 0$ and for $\tilde{q}_{\text{neuron}} < q + \frac{1}{2}$ there are

$$\tau_{\text{train}}(q) + \tau_{\text{train}}(q+1) = \alpha_{\tau}(2q+1) + 2\beta_{\tau} < 2\alpha_{\tau}\tilde{q}_{\text{neuron}} + 2\beta_{\tau} = 2\tau_{\text{train}}(\tilde{q}_{\text{neuron}}) \quad (36)$$

$$\rho_{\text{er}}(q) + \rho_{\text{er}}(q+1) = \alpha_{\rho}(2q+1) + 2\beta_{\rho} > 2\alpha_{\rho}\tilde{q}_{\text{neuron}} + 2\beta_{\rho} = 2\rho_{\text{er}}(\tilde{q}_{\text{neuron}}) \quad (37)$$

whence due to (19), uniting inequalities (36) and (37), the inequality

$$\tau_{\text{train}}(q) + \tau_{\text{train}}(q+1) < \rho_{\text{er}}(q) + \rho_{\text{er}}(q+1) \quad (38)$$

turns true. And due to $\alpha_{\tau} < 0$, meaning that $\tau_{\text{train}}(q_{\text{neuron}})$ decreases against increasing $\rho_{\text{er}}(q_{\text{neuron}})$ at $q_{\text{neuron}} \in [q; q+1]$, the inequality (38) may be re-stated as

$$0 < \tau_{\text{train}}(q) - \rho_{\text{er}}(q) < \rho_{\text{er}}(q+1) - \tau_{\text{train}}(q+1) \quad (39)$$

The inequality (21) is obtained with unifying the inequalities (35) and (39).

For $\tilde{q}_{\text{neuron}} > q + \frac{1}{2}$ reasonings are symmetric. If $\alpha_{\tau} > 0$ then $\alpha_{\rho} < 0$ and there are (36) and (37), and the inequality (38) turns true after having united them due to (19). Due to $\alpha_{\tau} > 0$ the inequality (38) may be re-stated as

$$0 < \tau_{\text{train}}(q+1) - \rho_{\text{er}}(q+1) < \rho_{\text{er}}(q) - \tau_{\text{train}}(q) \quad (40)$$

Otherwise, if $\alpha_{\tau} < 0$ then $\alpha_{\rho} > 0$ and there are (32) and (33), and the inequality (34) turns true after having united them due to (19). Due to $\alpha_{\tau} < 0$ the inequality (34) may be re-stated as

$$\tau_{\text{train}}(q) - \rho_{\text{er}}(q) > \rho_{\text{er}}(q+1) - \tau_{\text{train}}(q+1) > 0 \quad (41)$$

The inequality (22) is obtained with unifying the inequalities (40) and (41).

The theorem has been proved.

It is easy to see that the claim of Theorem 1 is easily narrowed (Edwards, 1965) from the segment $[q_{\min}; q_{\max}]$ to the interval $(q_{\min}; q_{\max})$. Under conditions of this theorem the hidden layer neuron number should be set at $\tilde{q}_{\text{neuron}}$ if just $\tilde{q}_{\text{neuron}} \in \mathbb{Z}$, bringing the neural network performance and the traintime duration to equilibrium (19). Of course, it occurs rarely, and so in the case of that $\tilde{q}_{\text{neuron}} \in (q; q+1)$ by $q \in \mathbb{N}$ the hidden layer neuron number should be set at q for $\tilde{q}_{\text{neuron}} < q + \frac{1}{2}$ and

$$q \in \arg \min_{k \in \{q, q+1\}} \{ \tau_{\text{train}}(k) + \rho_{\text{er}}(k) \} \quad (42)$$

and the hidden layer neuron number should be set at $q+1$ for $\tilde{q}_{\text{neuron}} > q + \frac{1}{2}$ and

$$(q+1) \in \arg \min_{k \in \{q, q+1\}} \{ \tau_{\text{train}}(k) + \rho_{\text{er}}(k) \} \quad (43)$$

Such hidden layer neuron number is optimal in the sense of that for $\tilde{q}_{\text{neuron}} < q + \frac{1}{2}$ the deviation from equilibrium (19) with q hidden layer neurons is less than with $q+1$ due to (21), and for $\tilde{q}_{\text{neuron}} > q + \frac{1}{2}$ the deviation from equilibrium (19) with $q+1$ hidden layer neurons is less than with q due to (22), where the sum of normalized traintime and performance value stays minimal. If $\tilde{q}_{\text{neuron}} < q + \frac{1}{2}$ and (43) is true then the hidden layer neuron number should be set at $q+1$, using the principle of maximizing the productivity quality due to Bernoulli criterion (Trukhayev, 1981), rather than tending to equilibrium. Similarly, for $\tilde{q}_{\text{neuron}} > q + \frac{1}{2}$ and (42) the hidden layer neuron number should be set at q , ensuring the productivity quality maximum due to (42).

The subcase of $\tilde{q}_{\text{neuron}} \notin \mathbb{Z}$ when $\tilde{q}_{\text{neuron}} = q + \frac{1}{2}$ is rare also, where instead of equilibrium (19) there is equilibrium (20), letting suggest to set the hidden layer neuron number at either q or $q+1$, if any additional preferences for one of those numbers can be provided. For instance, the less neuron number in neural network the greater possibility of saving the memory and accelerating the functionality. If nevertheless there are no any additional preferences then the hidden layer neuron number should be set randomly at either q or $q+1$, or to use the mix of two neural network architectures if it is available.

However, conditions of Theorem 1 are too weak to use them in setting neurons into the hidden layer, as usually the structure of dependences $\tau_{\text{train}}(q_{\text{neuron}})$ and $\rho_{\text{er}}(q_{\text{neuron}})$ is not so simple. Surely, monotonicity is typical for these dependences over some intervals (Haykin, 1999; Tarca, Grandjean, & Larachi, 2004), but over a wider range of hidden layer neuron number each of them may become non-monotonic. Therefore, the following theorem discloses the simplest case of non-monotonicity.

Theorem 2. Whatever continuous piecewise linear functions $\tau_{\text{train}}(q_{\text{neuron}})$ and $\rho_{\text{er}}(q_{\text{neuron}})$ with single minima within $(q_{\text{min}}; q_{\text{max}})$ are, being defined with its linear pieces on integer-endpoint subsegments of $[q_{\text{min}}; q_{\text{max}}]$ and having no maxima within $(q_{\text{min}}; q_{\text{max}})$, if $\tau_{\text{train}}(q_{\text{neuron}})$ has the minimum in $q_{\text{neuron}}^{(1)} \in (q_{\text{min}}; q_{\text{max}})$ and $\rho_{\text{er}}(q_{\text{neuron}})$ has the minimum in $q_{\text{neuron}}^{(2)} \in (q_{\text{min}}; q_{\text{max}})$ by $q_{\text{neuron}}^{(1)} \neq q_{\text{neuron}}^{(2)}$, and

$$\frac{\tau_{\text{train}}(q_{\text{neuron}})}{dq_{\text{neuron}}} \neq 0 \text{ by } q_{\text{neuron}} \in (q; q+1), \quad q = \overline{q_{\text{min}}, q_{\text{max}} - 1} \quad (44)$$

$$\frac{\rho_{\text{er}}(q_{\text{neuron}})}{dq_{\text{neuron}}} \neq 0 \text{ by } q_{\text{neuron}} \in (q; q+1), \quad q = \overline{q_{\text{min}}, q_{\text{max}} - 1} \quad (45)$$

then they are both strictly monotonic within the interval between endpoints $q_{\text{neuron}}^{(1)}$ and $q_{\text{neuron}}^{(2)}$, and one of them is decreasing and the other is increasing over that interval.

Proof. For definiteness let it be $q_{\text{neuron}}^{(1)} < q_{\text{neuron}}^{(2)}$. As neither $\tau_{\text{train}}(q_{\text{neuron}})$ has maximum by $q_{\text{neuron}} \in (q_{\text{neuron}}^{(1)}; q_{\text{neuron}}^{(2)})$, nor $\rho_{\text{er}}(q_{\text{neuron}})$ has maximum by $q_{\text{neuron}} \in (q_{\text{neuron}}^{(1)}; q_{\text{neuron}}^{(2)})$ then

$$\frac{\tau_{\text{train}}(q_{\text{neuron}})}{dq_{\text{neuron}}} < 0 \text{ by } q_{\text{neuron}} \in (q_{\text{min}}; q_{\text{neuron}}^{(1)}),$$

$$\frac{\tau_{\text{train}}(q_{\text{neuron}})}{dq_{\text{neuron}}} > 0 \text{ by } q_{\text{neuron}} \in (q_{\text{neuron}}^{(1)}; q_{\text{max}}),$$

and

$$\frac{\rho_{er}(q_{neuron})}{dq_{neuron}} < 0 \text{ by } q_{neuron} \in (q_{min}; q_{neuron}^{(2)}),$$

$$\frac{\rho_{er}(q_{neuron})}{dq_{neuron}} > 0 \text{ by } q_{neuron} \in (q_{neuron}^{(2)}; q_{max})$$

with (44), (45). So,

$$\frac{\tau_{train}(q_{neuron})}{dq_{neuron}} > 0 \text{ by } q_{neuron} \in (q_{neuron}^{(1)}; q_{neuron}^{(2)}),$$

$$\frac{\rho_{er}(q_{neuron})}{dq_{neuron}} < 0 \text{ by } q_{neuron} \in (q_{neuron}^{(1)}; q_{neuron}^{(2)}),$$

that is the function $\tau_{train}(q_{neuron})$ is monotonically increasing, and the function $\rho_{er}(q_{neuron})$ is monotonically decreasing over the interval $(q_{neuron}^{(1)}; q_{neuron}^{(2)})$. Clearly, if there were taken $q_{neuron}^{(1)} > q_{neuron}^{(2)}$ primarily then the function $\tau_{train}(q_{neuron})$ would have appeared monotonically decreasing by the monotonically increasing function $\rho_{er}(q_{neuron})$ over the interval $(q_{neuron}^{(2)}; q_{neuron}^{(1)})$. The theorem has been proved.

The claim of Theorem 2 is easily propagated (Edwards, 1965) from the interval $(q_{neuron}^{(1)}; q_{neuron}^{(2)})$ to the segment $[q_{neuron}^{(1)}; q_{neuron}^{(2)}]$ if only to state that a function may have single maximum either in the point q_{min} or q_{max} (actually, it has). In the partial case, a function may have two maxima in these points, where the function value is the same (equal to 1). And normalizing again the functions $\tau_{train}(q_{neuron})$ and $\rho_{er}(q_{neuron})$ on the segment $[q_{neuron}^{(1)}; q_{neuron}^{(2)}]$ as

$$\tilde{\tau}_{train}(q_{neuron}) = \frac{\tau_{train}(q_{neuron})}{\max_{q_{neuron} \in [q_{neuron}^{(1)}; q_{neuron}^{(2)}]} \{\tau_{train}(q_{neuron})\}} \quad (46)$$

$$\tilde{\rho}_{er}(q_{neuron}) = \frac{\rho_{er}(q_{neuron})}{\max_{q_{neuron} \in [q_{neuron}^{(1)}; q_{neuron}^{(2)}]} \{\rho_{er}(q_{neuron})\}} \quad (47)$$

have conditions of Theorem 1, letting know that there exists the unique point $\tilde{q}_{neuron} \in [q_{neuron}^{(1)}; q_{neuron}^{(2)}]$ and to set neurons into the hidden layer optimally. Rarely, if $\tau_{train}(q_{neuron})$ and $\rho_{er}(q_{neuron})$ with single minima have their minima in the same point $q_{neuron}^{(0)} \in (q_{min}; q_{max})$ then $q_{neuron}^* = q_{neuron}^{(0)}$, meaning that the two-objective minimization problem (16), (17), or (16), (18) is solved exactly.

More generally, on the segment $[q_{min}; q_{max}]$ continuous piecewise linear functions $\tau_{train}(q_{neuron})$ and $\rho_{er}(q_{neuron})$ with nonzero derivatives may have up to $q_{max} - q_{min} + 1$ local extremums, including extremums in the endpoints. The practical rare case when these functions have their global minima in the same point $q_{neuron} = q_{neuron}^*$ is pretty trivial, though it gives the exact solution in setting hidden layer neuron number at q_{neuron}^* . So further there is only the case with different global minima of functions $\tau_{train}(q_{neuron})$ and $\rho_{er}(q_{neuron})$.

Theorem 3. If continuous piecewise linear functions $\tau_{train}(q_{neuron})$ and $\rho_{er}(q_{neuron})$, whose linear pieces are defined on integer-endpoint subsegments of $[q_{min}; q_{max}]$, have their global minima in the points $q_{neuron}^{(1)} \in (q_{min}; q_{max})$ and $q_{neuron}^{(2)} \in (q_{min}; q_{max})$ correspondingly by $q_{neuron}^{(1)} \neq q_{neuron}^{(2)}$, (44), (45), then in the case

of when their global maxima on the segment $[q_{\text{neuron}}^{(1)}; q_{\text{neuron}}^{(2)}]$ are in the points $q_{\text{max}}^{(\tau)} \in [q_{\text{neuron}}^{(1)}; q_{\text{neuron}}^{(2)}]$ and $q_{\text{max}}^{(\rho)} \in [q_{\text{neuron}}^{(1)}; q_{\text{neuron}}^{(2)}]$ correspondingly by $q_{\text{max}}^{(\tau)} \neq q_{\text{max}}^{(\rho)}$, there exists the nonempty set of the points of intersection of functions (46) and (47).

Proof. For definiteness let it be $q_{\text{neuron}}^{(1)} < q_{\text{neuron}}^{(2)}$. Assume that

$$\tilde{\tau}_{\text{train}}(q_{\text{neuron}}) \neq \tilde{\rho}_{\text{er}}(q_{\text{neuron}}) \quad \forall q_{\text{neuron}} \in [q_{\text{neuron}}^{(1)}; q_{\text{neuron}}^{(2)}].$$

Then

$$\tilde{\tau}_{\text{train}}(q_{\text{neuron}}) > \tilde{\rho}_{\text{er}}(q_{\text{neuron}}) \quad (48)$$

Or

$$\tilde{\tau}_{\text{train}}(q_{\text{neuron}}) < \tilde{\rho}_{\text{er}}(q_{\text{neuron}}) \quad (49)$$

For definiteness let it be (48). Consequently,

$$\tilde{\tau}_{\text{train}}(q_{\text{max}}^{(\tau)}) > \tilde{\tau}_{\text{train}}(q_{\text{neuron}}^{(1)}) > \tilde{\rho}_{\text{er}}(q_{\text{max}}^{(\rho)})$$

what is false as

$$\tilde{\tau}_{\text{train}}(q_{\text{max}}^{(\tau)}) = \tilde{\rho}_{\text{er}}(q_{\text{max}}^{(\rho)}) = 1 \quad (50)$$

due to conditions of this theorem and normalization in (46), (47). Assuming that (49) is true, have

$$\tilde{\tau}_{\text{train}}(q_{\text{max}}^{(\tau)}) < \tilde{\rho}_{\text{er}}(q_{\text{neuron}}^{(2)}) < \tilde{\rho}_{\text{er}}(q_{\text{max}}^{(\rho)})$$

what is false as (50) is fulfilled due to conditions of this theorem and normalization in (46), (47). So, there exists the nonempty set of the points of intersection of functions (46) and (47). Clearly, if it were taken $q_{\text{neuron}}^{(1)} > q_{\text{neuron}}^{(2)}$ primarily then the symmetric contradictions would have appeared. The theorem has been proved.

May the conditions of Theorem 3 are true. Let $\{q_{\text{neuron}}^{(k+2)}\}_{k=1}^K$ be the set of $K \in \mathbb{N}$ points of intersection of functions (46) and (47) on the segment $[q_{\text{neuron}}^{(1)}; q_{\text{neuron}}^{(2)}]$, having integer points in the set $T = \{q_{\text{neuron}}^{(k+2)}\}_{k=1}^K \cap \mathbb{N}$. Obviously, the hidden layer neuron number is selected from the set of points

$$q_{\text{neuron}}^* \in \arg \min_{q \in \{q_{\text{neuron}}^{(k+2)}\}_{k=1}^K} \{\tau_{\text{train}}(q) + \rho_{\text{er}}(q)\} \quad (51)$$

where the point $q_{\text{neuron}}^* \in T \subset \mathbb{Z}$ is preferred to the point $q_{\text{neuron}}^* \notin T \subset \{q_{\text{neuron}}^{(k+2)}\}_{k=1}^K$. If $q_{\text{neuron}}^* \in (q; q+1)$ by $q \in \mathbb{N}$ then the hidden layer neuron number should be set in the way, described before Theorem 2. Here, however, the applied Bernoulli criterion may be extended from minimal-integer vicinity to the whole integer range $\overline{q_{\text{neuron}}^{(1)}, q_{\text{neuron}}^{(2)}}$, letting come from (51) to the problem

$$q_{\text{neuron}}^* \in \arg \min_{q \in \overline{q_{\text{neuron}}^{(1)}, q_{\text{neuron}}^{(2)}}} \{\tau_{\text{train}}(q) + \rho_{\text{er}}(q)\} \quad (52)$$

where the point $q_{\text{neuron}}^* \in T$ is preferred to the point $q_{\text{neuron}}^* \notin T \subset \overline{q_{\text{neuron}}^{(1)}, q_{\text{neuron}}^{(2)}}$ as in accordance with Theorem 3 a point from the set $T = \{q_{\text{neuron}}^{(k+2)}\}_{k=1}^K \cap \mathbb{N}$ brings the neural network performance and the traintime duration to equilibrium:

$$\tilde{\tau}_{\text{train}}(q) = \tilde{\rho}_{\text{er}}(q) \quad \forall q \in T = \left\{ q_{\text{neuron}}^{(k+2)} \right\}_{k=1}^K \cap \mathbb{N}.$$

If the problem (52) solution set is empty, what may happen for continuous piecewise nonlinear functions, then the hidden layer neuron number is selected through solving the extended problem

$$q_{\text{neuron}}^* \in \arg \min_{q \in [q_{\text{neuron}}^{(1)}; q_{\text{neuron}}^{(2)}]} \{ \tau_{\text{train}}(q) + \rho_{\text{er}}(q) \} \quad (53)$$

where the point $q_{\text{neuron}}^* \in \left\{ q_{\text{neuron}}^{(k+2)} \right\}_{k=1}^K$ is preferred to the point $q_{\text{neuron}}^* \notin \left\{ q_{\text{neuron}}^{(k+2)} \right\}_{k=1}^K$ as in accordance with Theorem 3 a point from the set of $K \in \mathbb{N}$ points of intersection of functions (46) and (47) on the segment $[q_{\text{neuron}}^{(1)}; q_{\text{neuron}}^{(2)}]$ brings the neural network performance and the traintime duration to equilibrium:

$$\tilde{\tau}_{\text{train}}(q) = \tilde{\rho}_{\text{er}}(q) \quad \forall q \in \left\{ q_{\text{neuron}}^{(k+2)} \right\}_{k=1}^K \quad (54)$$

After (53) is solved, for $q_{\text{neuron}}^* \in \left\{ q_{\text{neuron}}^{(k+2)} \right\}_{k=1}^K$ and $q_{\text{neuron}}^* \in (q; q+1)$ by $q \in \mathbb{N}$ the hidden layer neuron number should be set at q or $q+1$ due to Theorem 1. In the case of that $q_{\text{neuron}}^* \notin \left\{ q_{\text{neuron}}^{(k+2)} \right\}_{k=1}^K$, where $q_{\text{neuron}}^* \in (q; q+1)$ by $q \in \mathbb{N}$, the hidden layer neuron number should be set at

$$\arg \min_{k \in \{q, q+1\}} \{ \tau_{\text{train}}(k) + \rho_{\text{er}}(k) \}.$$

Actually, the problem (53) generalizes the procedure of searching the optimal hidden layer neuron number, but before it there must be tried (51) and (52). It remains only to say that the case, when (44) or (45) is untrue, occurs rarely, and it may lead to a continuum of the optimal hidden layer neuron numbers. Its practical rarity, as well as that $q_{\text{max}}^{(\tau)} = q_{\text{max}}^{(\rho)}$ under conditions of Theorem 3, lets omit its consideration. As for the generalization, then claims of Theorems 1 - 3 may be propagated from the continuous piecewise linear function to a polynomial, any part of which on the segment $[q; q+1]$ by $q = \overline{q_{\text{min}}, q_{\text{max}} - 1}$ has its derivative of the constant sign (Edwards, 1965).

5. Setting Neurons into the Hidden Layer for a Practical Case

For demonstrating how to apply conditions and claims of Theorems 1 - 3 for setting the hidden layer neuron number there is going to be constructed the feedforward neural network with the single hidden layer, using the Matlab® environment (license 161051) with its powerful tool Neural Network Toolbox for designing and testing neural networks. This network is destined for solving the problem of 5-by-7 monochrome image (symbol) recognition under noise of distortion, $Q_{\text{output}} = 26$. A good practical model for distortion noise is Gaussian noise. For the said image recognition problem its expectation is zero and standard deviation is set at 0.5. Due to (1) and (2) by $N_{\text{train}} = 1040$ the hidden layer neuron number may be varied as

$$43 \leq q_{\text{neuron}} \leq 459$$

or

$$40.2163 \leq q_{\text{neuron}} \leq 1083.9$$

correspondingly. The network is trained with Matlab-function `traingda` of the backpropagation algorithm, whereby 2 replicas of the pure symbol alphabet and 4 noised symbol alphabets are passed for 10 times. Practicing, it is sufficient to vary that number from 10 up to 120, as with $q_{\text{neuron}} < 10$ the neural network is trained too slow (Figure 1), and with $q_{\text{neuron}} > 120$ the neural network mean recognition error rate (Figure 2) becomes greater than the error rate for the network, trained without noise (Hagiwara, Hayasaka, Toda, Usui, & Kuno, 2001). It is seen from Figures 1 and 2 that for searching the optimal hidden layer neuron number it is sufficient to explore the segment $[20; 120]$. Dependences $\tau_{\text{train}}(q_{\text{neuron}})$ and $\bar{\rho}_{\text{er}}(q_{\text{neuron}})$ as continuous piecewise linear functions on that segment, and their normalized sum

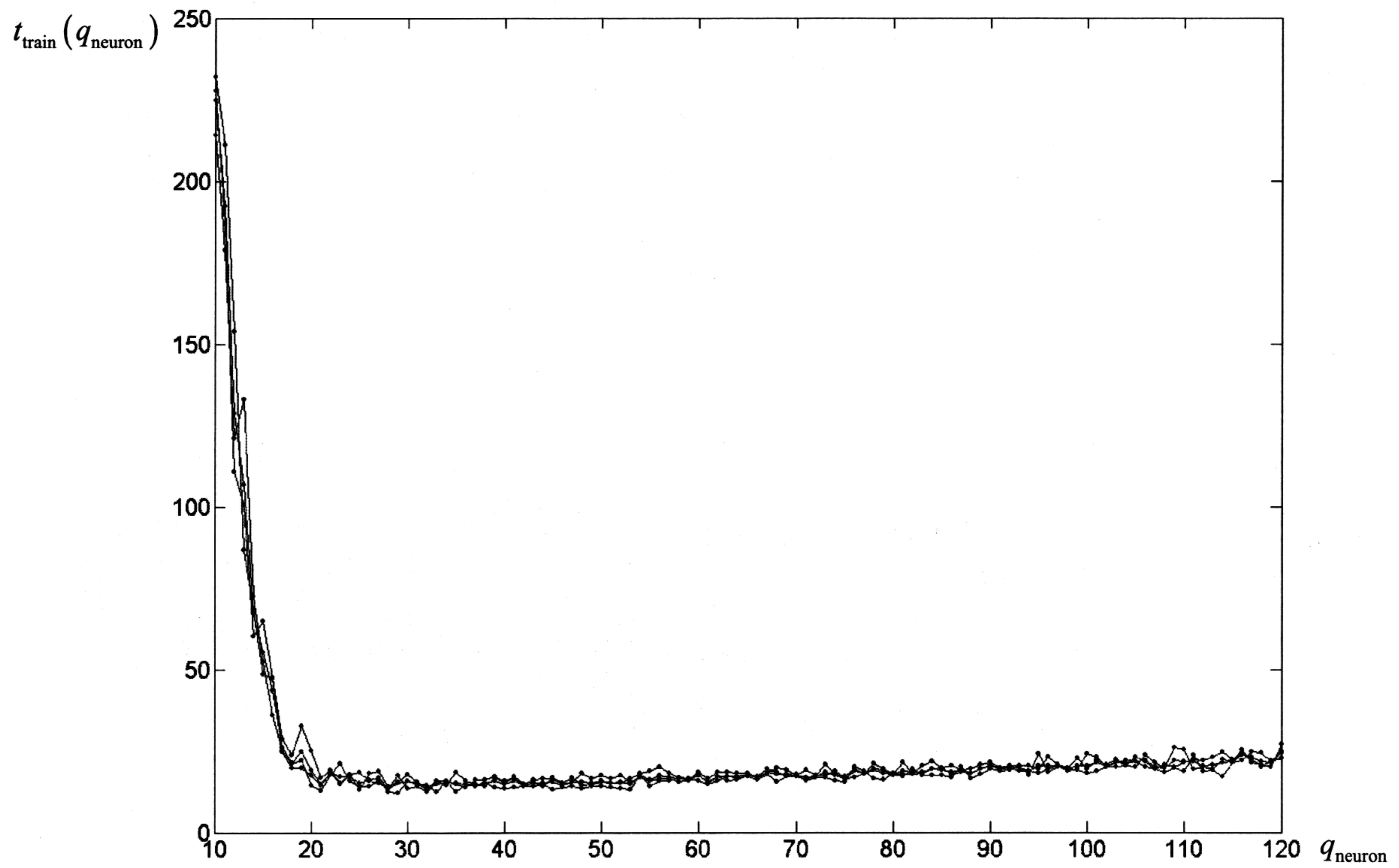


Figure 1. Dependences $t_{\text{train}}(q_{\text{neuron}})$ of the traintime duration in seconds against the hidden layer neuron number, drawn from averaging over the series of 1000 tests of the neural network

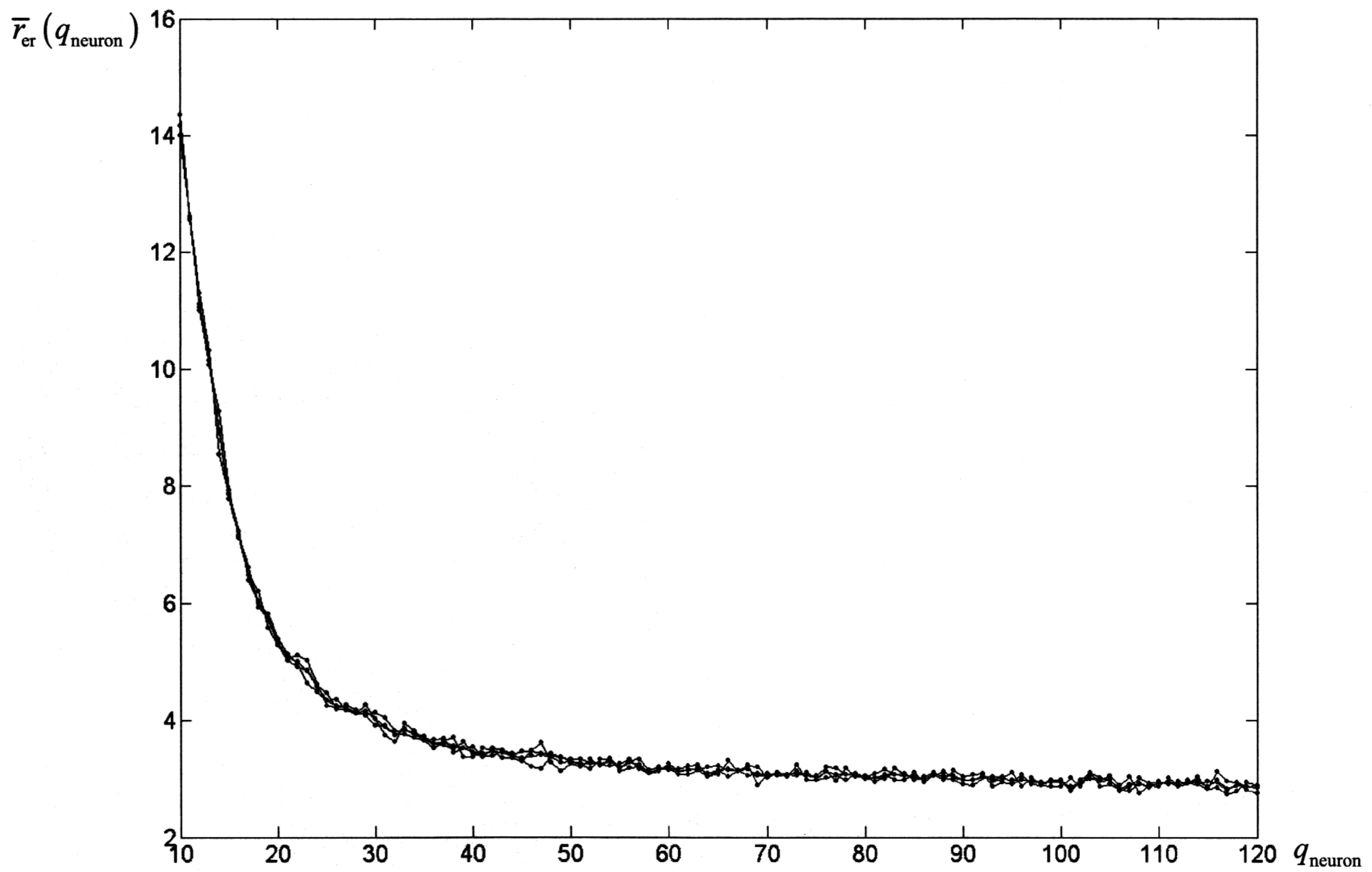


Figure 2. Dependences $\bar{r}_{\text{er}}(q_{\text{neuron}})$ of the mean recognition error rate (percentage wise) against the hidden layer neuron number, drawn from averaging over the series of 1000 tests of the neural network

$$\frac{\tau_{\text{train}}(q_{\text{neuron}}) + \bar{\rho}_{\text{er}}(q_{\text{neuron}})}{\max_{q_{\text{neuron}} \in [20; 120]} \{\tau_{\text{train}}(q_{\text{neuron}}) + \bar{\rho}_{\text{er}}(q_{\text{neuron}})\}} \tag{55}$$

as well, are shown on Figure 3, where polynomials, fitting these dependences are figured also.

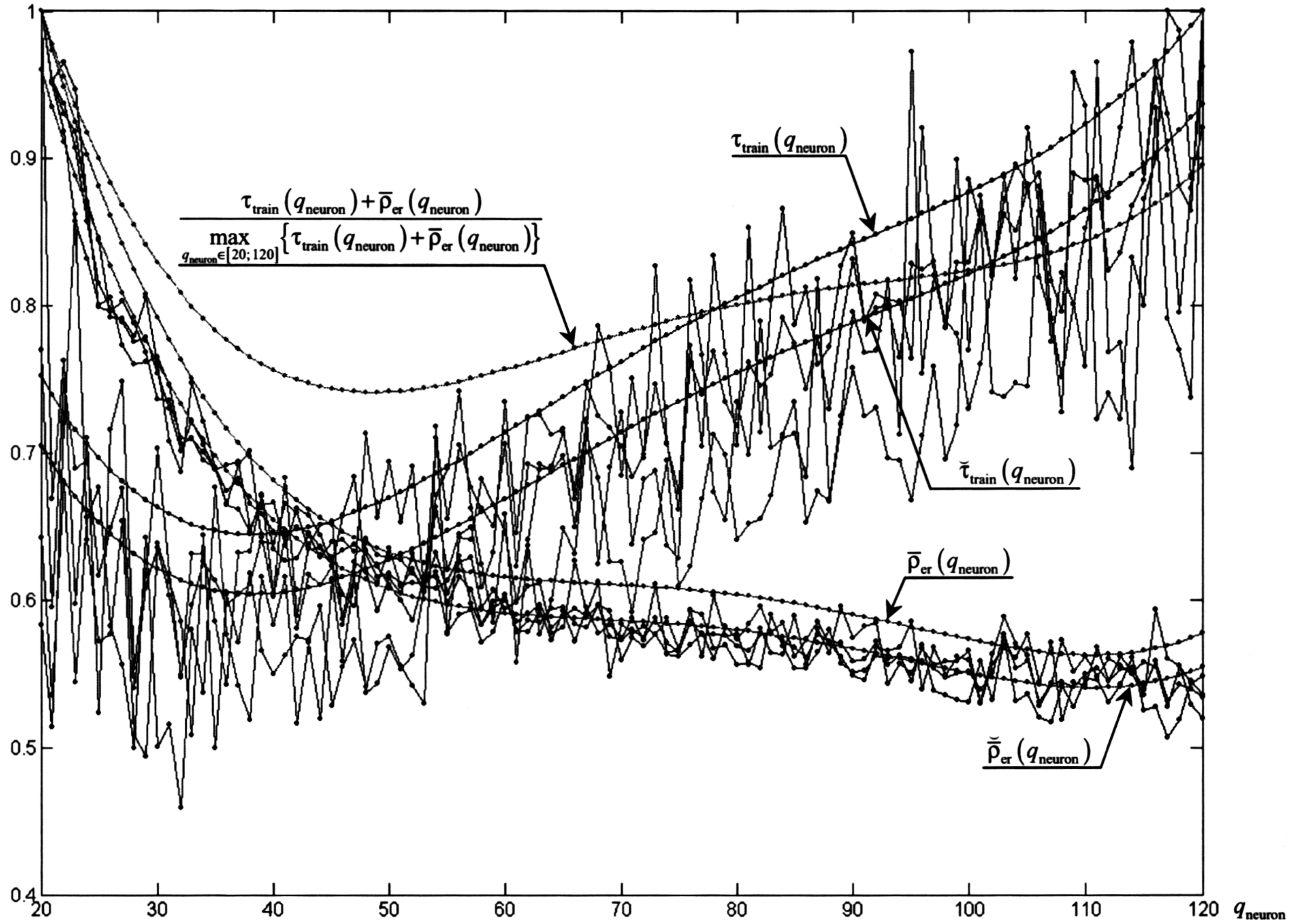


Figure 3. Dependences $\tau_{\text{train}}(q_{\text{neuron}})$ and $\bar{\rho}_{\text{er}}(q_{\text{neuron}})$ as continuous piecewise linear functions on the background of polynomials $\tilde{\tau}_{\text{train}}(q_{\text{neuron}})$ and $\tilde{\bar{\rho}}_{\text{er}}(q_{\text{neuron}})$, fitting these dependences over the segment $[20; 120]$

The dependence $\tau_{\text{train}}(q_{\text{neuron}})$ polynomial is

$$\begin{aligned} \tau_{\text{train}}(q_{\text{neuron}}) = & 2.9 \cdot 10^{-8} \cdot q_{\text{neuron}}^4 - 8.99835 \cdot 10^{-6} \cdot q_{\text{neuron}}^3 + \\ & + 0.001008915 \cdot q_{\text{neuron}}^2 - 0.044171 \cdot q_{\text{neuron}} + 1.2997346 \end{aligned} \tag{56}$$

and the dependence $\bar{\rho}_{\text{er}}(q_{\text{neuron}})$ polynomial is

$$\begin{aligned} \bar{\rho}_{\text{er}}(q_{\text{neuron}}) = & 3.267 \cdot 10^{-8} \cdot q_{\text{neuron}}^4 - 1.06582 \cdot 10^{-5} \cdot q_{\text{neuron}}^3 + \\ & + 0.0012673 \cdot q_{\text{neuron}}^2 - 0.0660244 \cdot q_{\text{neuron}} + 1.893601 \end{aligned} \tag{57}$$

Their normalized sum (55) is applied under Bernoulli criterion for (53). After normalization there can be seen (Figure 3) some shift upwards of polynomials, representing dependences $\tau_{\text{train}}(q_{\text{neuron}})$ and $\bar{\rho}_{\text{er}}(q_{\text{neuron}})$. For further, clearly, the continuous piecewise linear function, whose linear pieces are defined on integer-endpoint subsegments of $[q_{\text{min}}; q_{\text{max}}]$, having the same values as polynomials (56) and (57) have at points $\{\overline{q_{\text{min}}, q_{\text{max}}}\}$, can be substituted with these polynomials. At least, deductions of Theorems 1 - 3 with such substitution are fully applicable, as any linear piece on an integer-endpoint subsegment of $[q_{\text{min}}; q_{\text{max}}]$ approximates to any nonlinear piece of polynomials (56), (57), defined on that subsegment. From Figure 3 it is seen that on the segment $[20; 120]$ each of dependences $\tau_{\text{train}}(q_{\text{neuron}})$ and $\bar{\rho}_{\text{er}}(q_{\text{neuron}})$ has the single minimum: for $\tau_{\text{train}}(q_{\text{neuron}})$ this is $q_{\text{neuron}}^{(1)} = 38$, for $\bar{\rho}_{\text{er}}(q_{\text{neuron}})$ this is $q_{\text{neuron}}^{(2)} = 111$. Also they do not have any maxima within $(20; 120)$ by that the statements (44) and (45) are true. So, here conditions of Theorem 2 come into force: dependence $\tau_{\text{train}}(q_{\text{neuron}})$ is strictly increasing, and dependence $\bar{\rho}_{\text{er}}(q_{\text{neuron}})$ is strictly decreasing within the interval $(38; 111)$. At that by Theorem 1 there is the single point $\check{q}_{\text{neuron}} \approx 44.643$, giving the equilibrium (19). As here

$$\check{q}_{\text{neuron}} \in (44; 45) \text{ at } q = 44,$$

and for $\check{q}_{\text{neuron}} > 44.5$ here is the inequality (22)

$$|\bar{\rho}_{\text{er}}(44) - \tau_{\text{train}}(44)| > |\tau_{\text{train}}(45) - \bar{\rho}_{\text{er}}(45)|$$

then in the hidden layer there should be set 45 neurons due to (43).

The point of intersection of normalized (Figure 4) dependences $\tau_{\text{train}}(q_{\text{neuron}})$ and $\bar{\rho}_{\text{er}}(q_{\text{neuron}})$ on the segment $[38; 111]$ can be seen also by Theorem 3. The normalization (46), (47) shifted the intersection point: functions $\tau_{\text{train}}(q_{\text{neuron}})$ and $\bar{\rho}_{\text{er}}(q_{\text{neuron}})$ on the segment $[20; 120]$ have the intersection point $\check{q}_{\text{neuron}} \approx 44.643$, whereas functions $\check{\tau}_{\text{train}}(q_{\text{neuron}})$ and $\check{\bar{\rho}}_{\text{er}}(q_{\text{neuron}})$ on the segment $[38; 111]$ have the intersection point $q_{\text{neuron}}^{(3)} \approx 79.728$. The normalized sum of these functions reaches its minimum in the same point $q_{\text{neuron}}^* = 49$ as on the segment $[20; 120]$, as well as on the segment $[38; 111]$. So,

$$K = 1, \quad q_{\text{neuron}}^{(3)} \approx 79.728, \quad T = \emptyset,$$

and the problem (52) solution is $q_{\text{neuron}}^* = 49$, leading to 49 neurons in the hidden layer. The problem (53) solution here is the same due to the mentioned above substitution.

Obviously, that the result of solving the two-objective minimization problem (16), (18) depends on method to reduce it to the single objective minimization. The equilibrium way with (19) is suitable when it is unknown relationship of priority of the traintime duration and the network performance value, where neural network is probably to be retrained frequently. The way of Bernoulli criterion in (53) is suitable when influences of the traintime duration and the network performance value are roughly similar. However, as it is seen from Figure 4, equilibria (19) and (54) are quite different. On the other hand, the problem (52) solution always exists for continuous piecewise linear functions $\tau_{\text{train}}(q_{\text{neuron}})$ and $\rho_{\text{er}}(q_{\text{neuron}})$, whose linear pieces are defined on integer-endpoint subsegments of $[q_{\text{min}}; q_{\text{max}}]$. This may be called a Bernoulli criterion advantage (Trukhayev, 1981). Well, reasoning from Figure 4, in 5-by-7 monochrome image recognition problem the feedforward neural network should have 45 or 49 hidden layer neurons, allowing to have almost minimized the traintime duration and the network mean recognition error rate simultaneously. The number 45 is suitable for the case of total uncertainty of their influences. Nevertheless, 49 neurons in the hidden layer seem to be more adequate as Bernoulli criterion deals also with uncertainty, assuming equiprobability (Trukhayev, 1981).

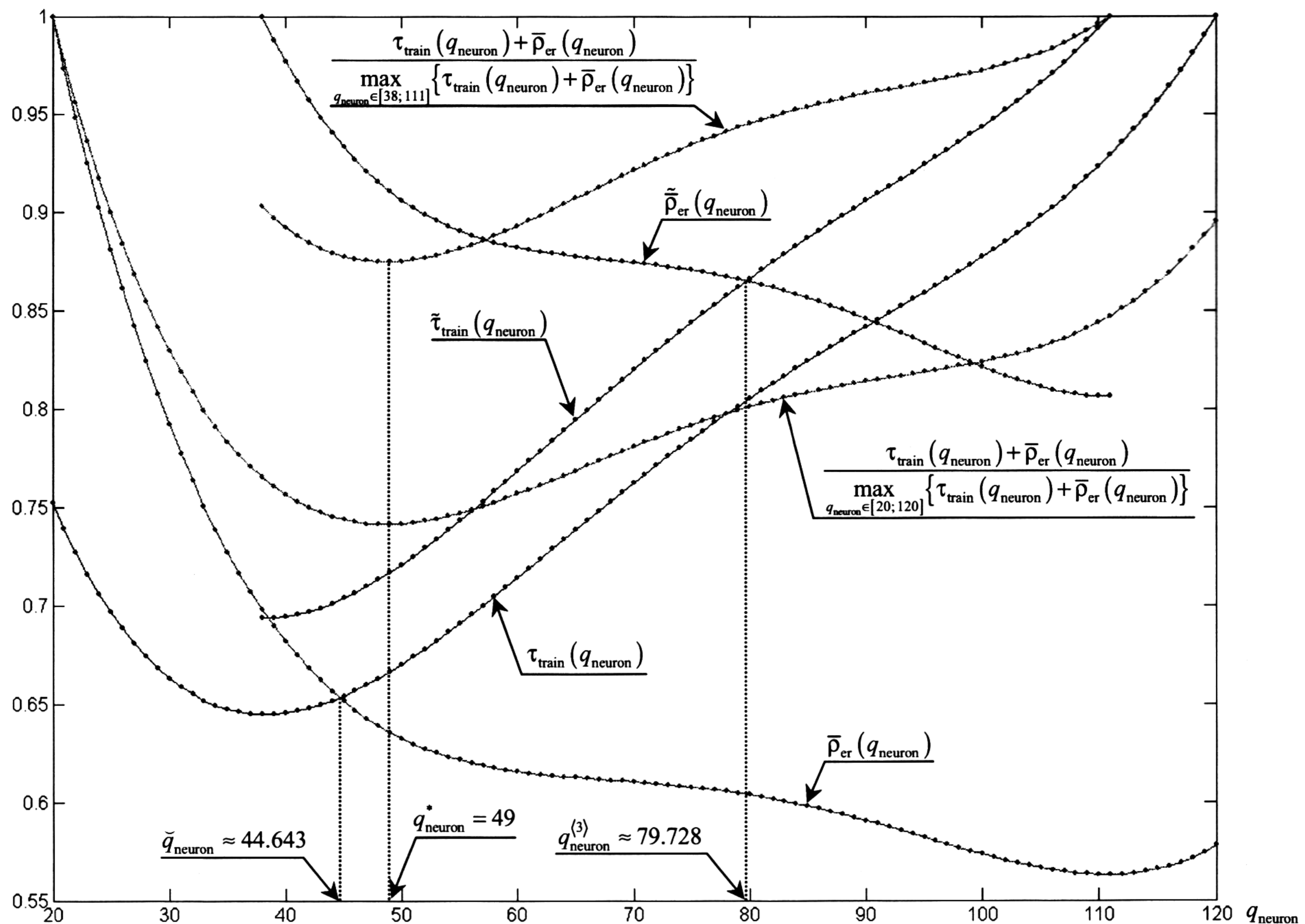


Figure 4. Dependences $\tau_{\text{train}}(q_{\text{neuron}})$, $\bar{\rho}_{\text{er}}(q_{\text{neuron}})$, and their normalized sum (55)

6. Conclusion

The stated two ways of setting the hidden layer neuron number in feedforward neural network with the single hidden layer are questioned for 5-by-7 monochrome image recognition problem. And the other question is of possibility of whether the stated result could be propagated to large-scale images recognition problem. For instance, there is the task to express the map of dependences $\{\tau_{\text{train}}(q_{\text{neuron}}), \rho_{\text{er}}(q_{\text{neuron}})\}$ and the segment $[q_{\text{min}}; q_{\text{max}}]$ for 5-by-7 monochrome image recognition problem into 10-by-14, 50-by-70, 600-by-800 and many others problem formats (scales). It is expectable that for problems, which scales are closed to the considered 5-by-7 format, this task is realizable, as the result for the 5-by-7 problem must be similar to 10-by-14 problem or something like that. Especially, if width and height of image in a new problem are in the same ratio 5-to-7. The most expected change after the map is that the segment $[q_{\text{min}}; q_{\text{max}}]$ is shifted right, as for larger-scale images there is need to set greater number of neurons into the hidden layer (Yuan, Xiong, & Huai, 2003), while shape of dependences $\{\tau_{\text{train}}(q_{\text{neuron}}), \rho_{\text{er}}(q_{\text{neuron}})\}$ might be maintained.

References

- Aladag, C. H. (2011). A new architecture selection method based on tabu search for artificial neural networks. *Expert Systems with Applications*, 38(4), 3287-3293. <http://dx.doi.org/10.1016/j.eswa.2010.08.114>
- Arulampalam, G., & Bouzerdoum, A. (2003). A generalized feedforward neural network architecture for classification and regression. *Neural Networks*, 16(5-6), 561-568. [http://dx.doi.org/10.1016/S0893-6080\(03\)00116-3](http://dx.doi.org/10.1016/S0893-6080(03)00116-3)
- Ballabio, D., & Vasighi, M. (2012). A MATLAB toolbox for Self Organizing Maps and supervised neural network learning strategies. *Chemometrics and Intelligent Laboratory Systems*, 118, 24-32. <http://dx.doi.org/10.1016/j.chemolab.2012.07.005>
- Benardos, P. G., & Vosniakos, G. C. (2007). Optimizing feedforward artificial neural network architecture.

- Engineering Applications of Artificial Intelligence*, 20(3), 365-382.
<http://dx.doi.org/10.1016/j.engappai.2006.06.005>
- Ciarelli, P. M., Oliveira, E., & Salles, E. O. T. (2012). An incremental neural network with a reduced architecture. *Neural Networks*, 35, 70-81. <http://dx.doi.org/10.1016/j.neunet.2012.08.003>
- Edwards, R. E. (1965). *Functional analysis: theory and applications*. Holt, Rinehart & Winston, Inc.
- Egmont-Petersen, M., de Ridder, D., & Handels, H. (2002). Image processing with neural networks - a review. *Pattern Recognition*, 35(10), 2279-2301. [http://dx.doi.org/10.1016/S0031-3203\(01\)00178-9](http://dx.doi.org/10.1016/S0031-3203(01)00178-9)
- Goltsev, A., & Gritsenko, V. (2012). Investigation of efficient features for image recognition by neural networks. *Neural Networks*, 28, 15-23. <http://dx.doi.org/10.1016/j.neunet.2011.12.002>
- Hagan, M. T., & Menhaj, M. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6), 989-993. <http://dx.doi.org/10.1109/72.329697>
- Hagiwara, K., Hayasaka, T., Toda, N., Usui, S., & Kuno, K. (2001). Upper bound of the expected training error of neural network regression for a Gaussian noise sequence. *Neural Networks*, 14(10), 1419-1429. [http://dx.doi.org/10.1016/S0893-6080\(01\)00122-8](http://dx.doi.org/10.1016/S0893-6080(01)00122-8)
- Han, M., & Yin, J. (2008). The hidden neurons selection of the wavelet networks using support vector machines and ridge regression. *Neurocomputing*, 72(1-3), 471-479. <http://dx.doi.org/10.1016/j.neucom.2007.12.009>
- Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. New Jersey: Prentice Hall, Inc.
- Iannella, N., & Back, A. D. (2001). A spiking neural network architecture for nonlinear function approximation. *Neural Networks*, 14(6-7), 933-939. [http://dx.doi.org/10.1016/S0893-6080\(01\)00080-6](http://dx.doi.org/10.1016/S0893-6080(01)00080-6)
- Kollias, S. D. (1996). A multiresolution neural network approach to invariant image recognition. *Neurocomputing*, 12(1), 35-57. [http://dx.doi.org/10.1016/0925-2312\(96\)00041-0](http://dx.doi.org/10.1016/0925-2312(96)00041-0)
- Kordík, P., Koutník, J., Drchal, J., Kovářík, O., Čepěk, M., & Šnorek, M. (2010). Meta-learning approach to neural network optimization. *Neural Networks*, 23(4), 568-582. <http://dx.doi.org/10.1016/j.neunet.2010.02.003>
- Kruglov, V. V., Dli, M. I., & Golunov, R. Y. (2001). *Fuzzy logics and artificial neural networks*. Moscow: PhysMathLit (in Russian).
- Kuzmanovski, I., & Novič, M. (2008). Counter-propagation neural networks in Matlab. *Chemometrics and Intelligent Laboratory Systems*, 90(1), 84-91. <http://dx.doi.org/10.1016/j.chemolab.2007.07.003>
- Lo, S. C. B., Chan, H. P., Lin, J. S., Li, H., Freedman, M. T., & Mun, S. K. (1995). Artificial convolution neural network for medical image pattern recognition. *Neural Networks*, 8(7-8), 1201-1214. [http://dx.doi.org/10.1016/0893-6080\(95\)00061-5](http://dx.doi.org/10.1016/0893-6080(95)00061-5)
- Mahmoud, M. A., & Ben-Nakhi, A. E. (2003). Architecture and performance of neural networks for efficient A/C control in buildings. *Energy Conversion and Management*, 44(20), 3207-3226. [http://dx.doi.org/10.1016/S0196-8904\(03\)00105-5](http://dx.doi.org/10.1016/S0196-8904(03)00105-5)
- Plaza, J., Plaza, A., Perez, R., & Martinez, P. (2009). On the use of small training sets for neural network-based characterization of mixed pixels in remotely sensed hyperspectral images. *Pattern Recognition*, 42(11), 3032-3045. <http://dx.doi.org/10.1016/j.patcog.2009.04.008>
- Plumb, A. P., Rowe, R. C., York, P., & Brown, M. (2005). Optimisation of the predictive ability of artificial neural network (ANN) models: A comparison of three ANN programs and four classes of training algorithm. *European Journal of Pharmaceutical Sciences*, 25(4-5), 395-405. <http://dx.doi.org/10.1016/j.ejps.2005.04.010>
- Tarca, L. A., Grandjean, B. P. A., & Larachi, F. (2004). Embedding monotonicity and concavity in the training of neural networks by means of genetic algorithms: Application to multiphase flow. *Computers & Chemical Engineering*, 28(9), 1701-1713. <http://dx.doi.org/10.1016/j.compchemeng.2004.01.003>
- Torrecilla, J. S., Otero, L., & Sanz, P. D. (2007). Optimization of an artificial neural network for thermal/pressure food processing: Evaluation of training algorithms. *Computers and Electronics in Agriculture*, 56(2), 101-110. <http://dx.doi.org/10.1016/j.compag.2007.01.005>
- Trukhayev, R. I. (1981). *Models of decision making under uncertainties*. Moscow: Nauka (in Russian).
- Wilson, C. L., Grother, P. J., & Barnes, C. S. (1996). Binary decision clustering for neural-network-based optical

character recognition. *Pattern Recognition*, 29(3), 425-437.
[http://dx.doi.org/10.1016/0031-3203\(95\)00105-0](http://dx.doi.org/10.1016/0031-3203(95)00105-0)

Wöhler, C., & Anlauf, J. K. (2001). Real-time object recognition on image sequences with the adaptable time delay neural network algorithm - applications for autonomous vehicles. *Image and Vision Computing*, 19(9-10), 593-618. [http://dx.doi.org/10.1016/S0262-8856\(01\)00040-3](http://dx.doi.org/10.1016/S0262-8856(01)00040-3)

Yuan, H. C., Xiong, F. L., & Huai, X. Y. (2003). A method for estimating the number of hidden neurons in feed-forward neural networks based on information entropy. *Computers and Electronics in Agriculture*, 40(1-3), 57-64. [http://dx.doi.org/10.1016/S0168-1699\(03\)00011-5](http://dx.doi.org/10.1016/S0168-1699(03)00011-5)

Yu, C., Manry, M. T., Li, J., & Narasimha, P. L. (2006). An efficient hidden layer training method for the multilayer perceptron. *Neurocomputing*, 70(1-3), 525-535. <http://dx.doi.org/10.1016/j.neucom.2005.11.008>

Zeng, X., & Yeung, D. S. (2006). Hidden neuron pruning of multilayer perceptrons using a quantified sensitivity measure. *Neurocomputing*, 69(7-9), 825-837. <http://dx.doi.org/10.1016/j.neucom.2005.04.010>

Zhang, Z., Ma, X., & Yang, Y. (2003). Bounds on the number of hidden neurons in three-layer binary neural networks. *Neural Networks*, 16(7), 995-1002. [http://dx.doi.org/10.1016/S0893-6080\(03\)00006-6](http://dx.doi.org/10.1016/S0893-6080(03)00006-6)