ISSN 1407-7493 COMPUTER SCIENCE DATORZINĀTNE 2008-7493

INFORMATION TECHNOLOGY AND MANAGEMENT SCIENCE
INFORMĀCIJAS TEHNOLOĢIJA UN VADĪBAS ZINĀTNE

NEURAL NETWORKS IN FINGERPRINT CLASSIFICATION PROBLEM

Arita Takahaši, Dr.sc.ing., lecturer, Institute of Information Technology, Faculty of Computer Science and Information Technology, Riga Technical University, Meža 1/4, Riga LV-1048, Latvia, e-mail: aritatakahashi@yahoo.com.

***Keywords:** neural networks, delta learning rule, discrete perceptron learning rule, search in a multi-dimensional space*

1. Introduction

There have been several extended and complicated studies of fingerprint classification problem like that of Prabhakar [1] which is aimed at developing a sophisticated software system that could be used to identify criminals or suspects. This paper does not intend to offer new solutions to a real problem, but is focused on a pilot-study of a task that could be used as a practical exercise for students studying delta learning rule of artificial neural networks. Course books, such as the one offered by Zurada [2] offer various exercises, but they are rather algorithms with plain numbers without any description of the objects that the network is used for to classify.

When analysing the performance of the delta learning rule, it is important to examine each move separately by means of a graphical representation of the search process. The impact of various values of constant λ and the choice of the initial weight values, for example, can be studied by analysing the moves of the algorithm through a two-dimensional weight space. Unfortunately, the course books do not offer a graphical representation of moves of the algorithm through multi-dimensional weight space, consisting of more than 2 weights. This study approaches the problem of graphical representation of moves in multi-dimensional weight space and demonstrates the moves of the algorithm, applied to the finger classification problem.

2. Problem Description

Let us suppose that there are three main classes of fingerprints – loop, arch and whorl (see Fig. 1). This kind of classification was introduced by Henry in 1899 (see Prabhakar [1]).

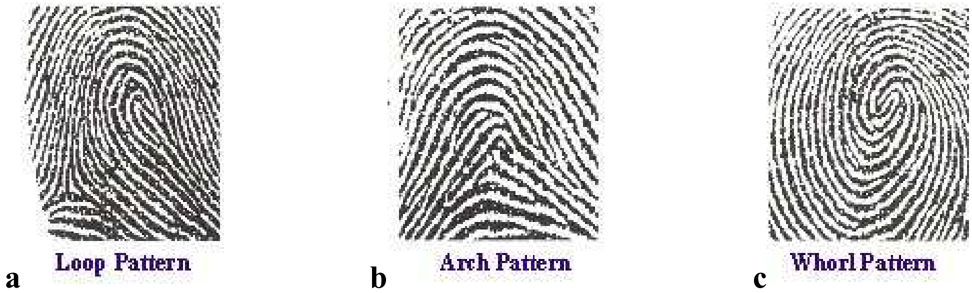

Figure 1. Classes of fingerprints according to Henry's classification

The loops can be classified further in "right loops" and "left loops", but the arch can have a subtype of a "tented arch". There can be discussions whether hybrids of these basic types should be defined as new classes or if they should be defined as hybrids. This study regards the types depicted in Figure 1 as classes, but the types carrying features from several classes as hybrids. The study also ignores the fact that the loop can go right or left. Any loop will be presumed bending the way "right loop" does.

At the beginning of this study the following hypothesis was formulated: an artificial neural network learns to classify representatives of pure classes much faster than those of hybrids.

Another assumption was that a discrete neural network works faster than a continuous. The task is to train an artificial neural network to classify the fingerprints into three classes – loop class, arch class and whorl class. Therefore, it seems logical to choose a one-layer network, consisting of three continuous perceptrons (see Fig.2).

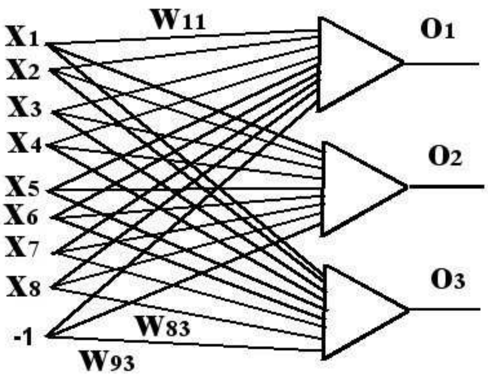

Figure 2. A one-layer perceptron network for classification into three classes

The teacher's desired responses d in case of the loop class are:

$$d_{loop} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{1}$$

The arch class is described by the following desired outputs:

$$d_{arch} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (2)$$

But the firing of the third neuron signifies features characteristic of the whorl class.

$$d_{whorl} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3)$$

The input signals x1, x2, … x8 are the length and the width of the most important pattern lines, measured in millimetres (see Table 1).

Table 1

The input signals and the corresponding features of the fingerprints

The most important pattern lines							
							
length (mm)	**width (mm)**	**length (mm)**	**width (mm)**	**length (mm)**	**width (mm)**	**Radius (Horizontal) (mm)**	**Radius (Vertical) (mm)**
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8

We can see that a whorl can be described by features x7 and x8, an arch can be described by features x1, x2, x3, x4, x5 and x6, but a loop can be described by x5 and x6.

2. Training sets

The experiments presented in this paper are run on three training sets – the one containing both pure representatives of the classes and their hybrids (see Table 2), another containing only pure representatives of the classes (see Table 3), and the third consisting only of hybrids (see Table 4).

Table 2

The training set consisting of both pure and hybrid fingerprint types

No.	Type	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	d_1	d_2	d_3
1		3,00	1,00	0,00	0,00	0,00	0,00	5,00	10,00	0,00	0,00	1,00
2		0,00	0,00	0,00	0,00	10,00	5,00	2,00	0,00	0,80	0,00	0,20
3		10,00	3,00	10,00	4,00	0,00	0,00	2,00	4,00	0,00	0,50	0,60

4		0,00	0,00	0,00	0,00	9,00	4,00	0,00	0,00	1,00	0,00	0,00
5		11,00	3,00	11,00	2,00	10,00	4,00	3,00	6,00	0,00	0,60	0,60
6		11,00	4,00	10,00	4,00	10,00	4,00	0,00	0,00	0,00	1,00	0,00
7		10,00	3,00	10,00	3,00	10,00	3,00	0,00	0,00	0,00	1,00	0,00
8		10,00	3,00	10,00	2,00	10,00	3,00	2,00	5,00	0,00	0,50	0,50

We can see from Table 2 that the pure representatives of the classes have desired responses “0” or “1” while the hybrids are described with values between 0 and 1 – in some cases with “0,5” or “0,6” – if the features of the both classes are equally well present and – “0,2” – if the features of the one class are dominated by the features of the other class. The expert’s decisions on what values to assign to d_1, d_2 and d_3 are in some degree subjective. It can slow down the classification process, but as the table does not contain any descriptions that could be regarded as contradictious, the network learns to classify the fingerprints without many difficulties.

The training set containing only pure elements of the fingerprint classes is depicted in Table 3. We can see from the table that this set can be used to train a discrete network with a similar structure that is shown in Figure 2. Therefore, this set was used to train both a continuous and discrete network. We also can see that this set is simply a subset of the one described in Table 2.

Table 3

The training set consisting of only pure types

No.	Type	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	d_1	d_2	d_3
1		3,00	1,00	0,00	0,00	0,00	0,00	5,00	10,00	0,00	0,00	1,00
4		0,00	0,00	0,00	0,00	9,00	4,00	0,00	0,00	1,00	0,00	0,00
6		11,00	4,00	10,00	4,00	10,00	4,00	0,00	0,00	0,00	1,00	0,00
7		10,00	3,00	10,00	3,00	10,00	3,00	0,00	0,00	0,00	1,00	0,00

The third training set is illustrated in Table 4. It also represents a subset of Table 2. Some of the presented fingerprint patterns are extremely wide-spread and common (for example, a loop with a whorl in the centre). Other patterns are very rare – arch patterns can be hardly found even in hybrid types. Some students can be interested in creating their own training sets by including descriptions of their own fingerprints or the fingerprints of their friends. Therefore they can learn about avoiding contradictions when providing the expert values of class membership.

Table 4

The training set consisting of hybrids only

No.	**Type**	**x_1**	**x_2**	**x_3**	**x_4**	**x_5**	**x_6**	**x_7**	**x_8**	**d_1**	**d_2**	**d_3**
2		**0,00**	**0,00**	**0,00**	**0,00**	**10,00**	**5,00**	**2,00**	**0,00**	**0,80**	**0,00**	**0,20**
3		**10,00**	**3,00**	**10,00**	**4,00**	**0,00**	**0,00**	**2,00**	**4,00**	**0,00**	**0,50**	**0,60**
5		**11,00**	**3,00**	**11,00**	**2,00**	**10,00**	**4,00**	**3,00**	**6,00**	**0,00**	**0,60**	**0,60**
8		**10,00**	**3,00**	**10,00**	**2,00**	**10,00**	**3,00**	**2,00**	**5,00**	**0,00**	**0,50**	**0,50**

They also must be aware that subjectivity can influence the result of the experiment; on the other hand, the definitions of fingerprint classes are very vague.

3. Encoding of the problem

The problem was encoded in FoxPro 2.6 for DOS. The data and results were saved in tables that could be opened by MS Excel in order to design the charts that illustrate the search process. The initial weight vector was chosen after several unsuccessful runs of the algorithm – the first set of weights with which the algorithm worked properly and showed satisfactory results. The global solutions were not found in the first five, six runs of the algorithm, because it finished exceeding the allowed number of iterations (not reaching the desired level of the Error value E). Therefore we can say that the initial weight vector was found experimentally and its values are given in Table 5.

Table 5

The initial weight values

	w_1	**w_2**	**w_3**	**w_4**	**w_5**	**w_6**	**w_7**	**w_8**	**w_9**
Neuron1	-0.6	0.2	-0.1	0.3	2.1	3.4	1.0	-0.5	0.3
Neuron2	2.0	0.1	0.6	0.3	2.	1.4	1.0	1.0	-1.0
Neuron3	-0.5	-0.5	-0.4	-0.6	0.4	1.2	1.0	2.5	-0.5

The learning constant c of the discrete perceptron algorithm is 0,5, but the constant λ is assigned 0,08 for all three training sets of the continuous network. The maximal value of the error E is to be below 0,495 in case of four training patterns (Table 3 and Table 4) and below 0,99 in case of eight training patterns (Table 2). The error E is calculated as a sum of errors of all three neurons through the whole learning cycle (i.e., one iteration) and is evaluated as the absolute value (not squared) of the difference between the desired output and the real output of the net. Therefore, in case of eight learning patterns we can write:

$$E = \sum_{i=1}^{8} \sum_{j=1}^{3} abs(d_{ij} - o_{ij}) \tag{4}$$

The number of maximum iterations allowed for the algorithm was 700, because a very long run of the algorithm could not be used for teaching purposes successfully – the MS Excel charts would take too much memory and the visual representation of the search process would be complicated.

The output of the j'th continuous unipolar perceptron was calculated (see [2]):

$$o_j(net) = \frac{1}{e^{-\lambda \cdot net} + 1} \tag{5}$$

The graphical representation of this activation function is given in Figure 3. It was used to choose a more suitable value of the constant λ when the first arbitrarily chosen values could not maintain the search process due to being too large.

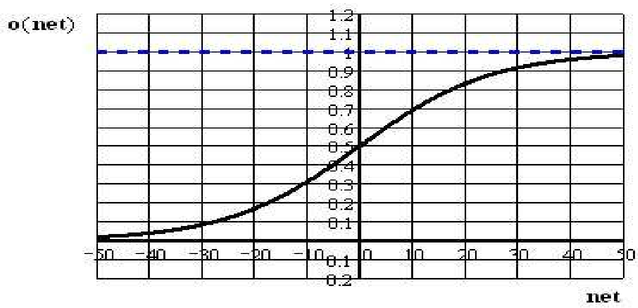

Figure 3. Unipolar continuous activation function

The output of the j'th discrete perceptron of the discrete network was calculated:

$$o_j(net) = \begin{cases} 1, & net > 0 \\ 1, & net = 0 \;\&\; d = 0 \\ 0, & net = 0 \;\&\; d = 1 \\ 0, & net < 0 \end{cases} \tag{6}$$

Such a description clarifies the output value in case of net=0. An object of the training set having net=0 is located on the boundary line between the two classes. If the object is described as belonging to one class and not belonging to another class, the adjustments of the network weights must take place, therefore the output o can be considered as "not d" or the opposite of the desired output d.

4. Results of the experiments

Some of the expectations were approved by the experiments. The hypothesis that the hybrid patterns would prolong the search process could be observed if comparing the graphical representation of the error E decreasing from iteration to iteration (see Figure 4).

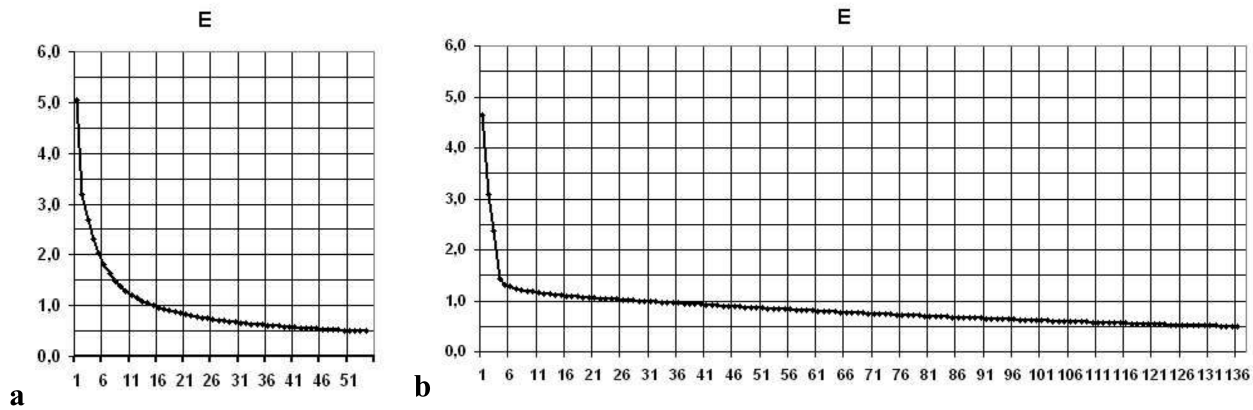

Figure 4. Charts showing the value of E over the optimization process – (a) in case of four pure patterns – (b) in case of four hybrids only

We can see from Figure 4 that in case of hybrid patterns the optimization of the weights of the neural network took 136 iterations while in case of pure patterns it took only 54 iterations to reach the same level of error E.

The values of error E in case of eight patterns (four pure and four hybrids) decrease slowly (see Figure 5). The run of the algorithm stops after 643 iterations. Obviously, the search process is much more complicated as there are both hybrids and pure examples that have to be classified correctly.

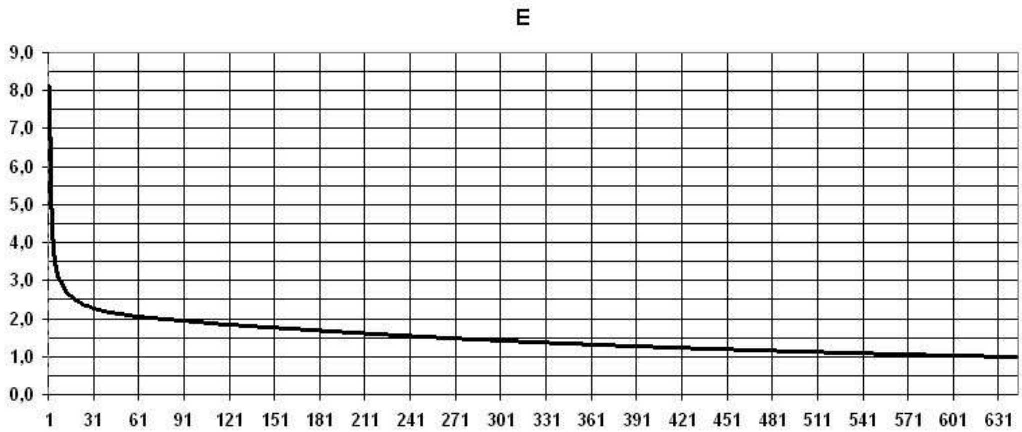

Figure 5. The value of E in case of eight mixed patterns

The assumption that a discrete neural network works faster than a continuous can be seen from Figure 6 (see also Figure 4, case (a) for comparison). We can see that the algorithm finishes its run after three iterations. A question may arise – why there are five corrections if the training set consists of only four patterns? The answer is that the corrections are summed up in all three neurons; therefore the maximum number of corrections can be 12.

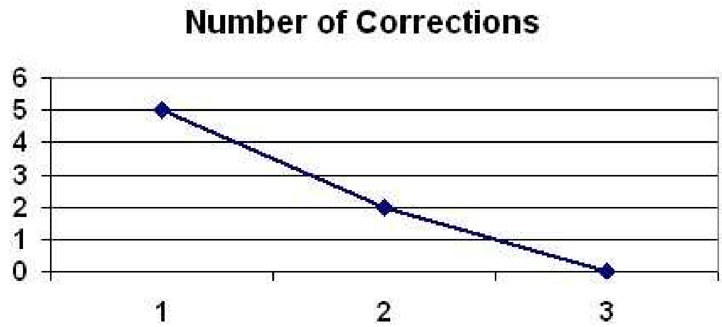

Figure 6. The number of corrections in case of discrete perceptron – four pure patterns

To analyse the search process and its movements in the space of 9 dimensions, a radar chart from MS Excel was applied to each separate step of the first two iterations (see Figure 7). This series of pictures was transformed into a cartoon, based on moving animation of “gif” images. The animation was generated with Animation Shop 2.

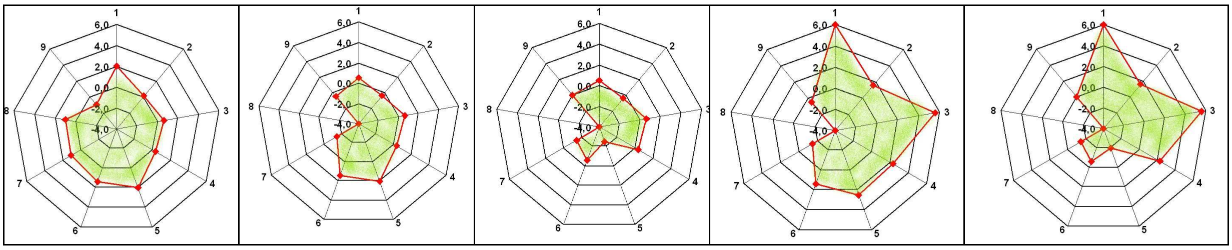

Figure 7. Dynamic changes of the weights of the second neuron of the discrete network

The main difference between the animation and Figure 7 is that the profiles not showing any weight correction have no sense of being depicted in printed form as they are simply exact copies of the charts depicting the weights one step earlier. In animation these repeated charts prolong the time of the search state by 20 time units, illustrating the dynamics of waiting/ searching situations.

A similar animation of weight changes was also created for the second perceptron of the continuous network. It analyzed the search moves of the first four iterations when the 8-pattern training set was used.

Such animations can be easily published on the Internet, shown with over-head projector and be used to illustrate the dynamics of the search process in case of multi-dimensional (more than two dimensions) search space.

5. Conclusions

The network final weights after the experiment with the four pure patterns are depicted in Table 6.

Table 6

The weight values after the training with pure class representatives

	W1	W2	W3	W4	W5	W6	W7	W8	W9
Neuron1	-4,0581	-0,9458	-2,7962	-0,6334	2,0586	3,6466	-0,0622	-2,6245	0,4871
Neuron2	3,3066	0,5280	3,2587	1,2193	-2,9204	-1,0492	-1,4562	-3,9125	0,0675
Neuron3	-0,7654	-0,5927	-0,9362	-0,7940	-3,7175	-0,5856	1,5066	3,5132	0,4502

These weights were tested with the four hybrid patterns, and the value of error E was 2,518. It is a small error if we consider that its maximal value can be 12, but its mean in case of uniform distribution would be 6. The fact that E=2,518 proves us that the training of the network with the set of pure class representatives also prepared the network for interpreting (classifying) their hybrids. We can see some similarities in the network responses with its desired outputs (see Table 7).

Table 7

The comparison of outputs and the desired values

No.	**Type**	**o_1**	**o_2**	**o_3**	**d_1**	**d_2**	**d_3**
2		**0,96**	**0,05**	**0,05**	**0,80**	**0,00**	**0,20**
3		**0,00**	**0,99**	**0,39**	**0,00**	**0,50**	**0,60**
5		**0,01**	**0,77**	**0,04**	**0,00**	**0,60**	**0,60**
8		**0,01**	**0,77**	**0,04**	**0,00**	**0,50**	**0,50**

The network final weights after the training with the four hybrid patterns are depicted in Table 8.

Table 8

The weight values after the training with the four hybrid patterns

	W1	W2	W3	W4	W5	W6	W7	W8	W9
Neuron1	-2,9716	-0,4807	-2,4716	-0,2622	0,3122	2,7489	0,4315	-1,6829	0,5330
Neuron2	1,3749	-0,8898	-0,0251	-4,6304	-1,2743	-1,1738	-1,2513	4,1598	1,4627
Neuron3	0,0403	-0,6355	0,1403	-2,0903	-1,4222	-0,0285	0,2278	3,9662	0,9822

These weights were tested with the four pure class representatives, and the value of error E was 5,465. This shows that training with hybrids does not prepare the network for classifying the pure class patterns. There are many reasons for that – the idea of what is a hybrid is very subjective, and the hybrids do not define the features of the pure classes they are derived from. The expert evaluations of the hybrid patterns are very subjective, and there is almost no explanation why one hybrid would have the desired output values (0; 0,6; 0,6), but another similar hybrid would have (0; 0,5; 0,5).

The conclusion is that for training a continuous perceptron most of the training patterns should be pure class representatives, yet some hybrids should be added if their presence in the real data that should be classified is rather frequent.

If we compare the two animations – the one of the discrete perceptron and the other with the continuous one – we can see that the weight changes are very gradual in case of the continuous perceptron, but they are radical in case of the discrete perceptron. On the other hand, the discrete perceptron does not provide a high precision for the classification – the boundary between the classes can lie very close to pure representatives of any class, while the continuous perceptron would set the boundaries in the area of the hybrids rather than in the area of typical class patterns.

6. Future work

This study can be developed in a larger scale of work by modifying the network structure – it can consist of two perceptrons showing a binary code of the class number rather than a class itself. Such a study could make students think about different network structures and their impact on the classification process itself.

References

1. Prabhakar S. Fingerprint Classification and Matching Using a Filterbank [dissertation]. Biometrics Research Homepage at Michigan State University, 2001 [updated 2008 Feb 25]. URL: http://biometrics.cse.msu.edu/Publications/Thesis/SalilPrabhakar_FpMatchingFilterbank_PhD01.pdf – Visit date September 2008.
2. Zurada J.M. Introduction to Artificial Neural Systems. // Pws Pub Co. United States of America, 1992.

Takahaši Arita. Neironu tīklu pielietojums pirkstu nospiedumu klasifikācijas uzdevumā

Raksts ir veltīts pirkstu nospiedumu klasifikācijas uzdevumam, pieņemot, ka visi pirkstu nospiedumi pieder trijām klasēm – cilpas klasei, loka klasei un spirālveida vijuma klasei. Paraugi, kuri satur līnijas no vairākām klasēm, tiek uzskatīti par hibrīdiem. Eksperimenti parāda, ka tīri klašu pārstāvji (ne hibrīdi) var tikt izmantoti neironu tīkla (sastāvoša no trijiem nepārtrauktiem perceptroniem) apmācībā tā, ka tas ir spējīgs klasificēt šo klašu hibrīdus. Pretējs eksperiments neapstiprinājās un parādīja, ka neironu tīkls nevarēja tikt apmācīts ar hibrīdiem tā, lai tas iemācītos klasificēt tīros klašu pārstāvjus. Apmācība ar tīrajiem klašu pārstāvjiem paņēma mazāk laika kā apmācība ar hibrīdiem – tā sastāvēja no 54 iterācijām (kamēr otra vilkās 136 iterācijas). Vislaikietilpīgākais bija eksperiments, kur apmācības kopā tika iekļauti gan tīrie klašu pārstāvji, gan hibrīdi (643 iterācijas) – tomēr tā priekšrocība bija augstā precizitāte, ar kādu tīkls klasificēja gan tīros klašu pārstāvjus, gan hibrīdus. Vismazāk laiku patērējošs bija eksperiments ar diskrētu tīklu (sastāvošu no trijiem unipolāriem perceptroniem) – visai apmācībai bija nepieciešami tikai trīs cikli, lai optimizētu tīkla svarus. Tā trūkums ir tāds, ka tikai tīri klašu pārstāvji var tikt lietoti. Šī pētījuma novitāte ir animācijas izmantošana pārvietojumu daudzdimensiju svaru telpā ilustrēšanā (deviņi svari). Tika izveidotas divas animācijas (no kurām viena tika parādīta freimu veidā šajā rakstā). Animācijas ļāva salīdzināt svaru izmaiņu pakāpeniskumu dažādiem eksperimentiem, jaunu profilu parādīšanos un veco profilu atkārtotu parādīšanos.

Takahashi Arita. Neural networks in fingerprint classification problem

The paper focuses on fingerprint classification problem. It assumes all fingerprints belong to three classes – a loop class, an arch class and a whorl class. Patterns containing lines from several classes are regarded as hybrids. The experiments show that pure class representatives (not hybrids) can be used to train a neural network (consisting of three continuous perceptrons) so that it is also able to classify the hybrids of these classes. A reversed experiment failed and showed that the neural network could not be trained with hybrid patterns in order to learn how to classify the pure ones. The training with the pure patterns also took less time than the one with the hybrid – it performed 54 iterations (while the other training took 136 iterations). The most time-consuming was the experiment where both pure and hybrid patterns were included in the training set (643 iterations) – but its advantage was high precision when classifying pure and hybrid patterns. The least time consuming was the experiment with the discrete network (consisting of three unipolar perceptrons) – the whole training needed only three cycles to optimize the network weights. Its drawback is that only pure class representatives can be used. The novelty of this study is application of an animation in order to illustrate the moves through a multi-dimensional space (nine weights). Two animations were created (one of which was presented as frames in this paper).The animations made it possible to compare the gradualness of the weight changes of different experiments, emergence of new profiles and recurrence of old ones.

Такахаши Арита. Нейронные сети в задаче классификации отпечатков пальцев

В статье рассматривается задача классификации отпечатков пальцев. Автор предполагает, что все отпечатки принадлежат одному из трёх классов – классу «петля», классу «арка» или классу «спиральный виток». Образцы, содержащие линии нескольких классов, рассматриваются как гибриды. Эксперименты показывают, что чистые представители классов (не гибриды) могут использоваться

для обучения нейронной сети (состоящей из трёх непрерывных перцептронов) так, что она обучается классифицировать также и гибриды этих классов. Обратный эксперимент не подтвердился и показал, что нейронная сеть не могла обучиться на гибридах так, чтобы уметь классифицировать чистые представители классов. Обучение чистыми представителями классов заняло меньше времени, чем обучение гибридами – оно состояло из 54 итераций (другое длилось 136 итераций). Самым длительным экспериментом оказался тот, в котором в обучающей выборке использовались как чистые представители классов, так и гибриды (643 итераций) – но его преимуществом явилась высокая точность при классификации чистых представителей и гибридов. Наименьшая длительность была у эксперимента с дискретной сетью (состоящей из трёх униполярных перцептронов) – для всего обучения потребовалось только три цикла, чтобы оптимизировать веса сети. Его недостатком является то, что могут быть использованы только чистые представители классов. Новизна данного научного исследования состоит в применении анимаций для иллюстраций перемещений по многомерному пространству (девять весов). Были созданы две анимации (одна из которых была показана в виде фреймов в этой статье). Анимации позволили сравнить постепенность изменений весов в разных экспериментах, появление новых профилей и возвращение старых.