Reminder of the First Paper on Transfer Learning in Neural Networks, 1976

Stevo Bozinovski
South Carolina State University, Orangeburg, SC, USA
E-mail: sbozinovski@scsu.edu

Overview paper

Keywords: transfer learning, neural networks

Received: June 10, 2019

This paper describes a work on transfer learning in neural networks carried out in 1970s and early 1980s, which produced its first publication in 1976. In the contemporary research on transfer learning there is a belief that pioneering work on transfer learning took place in early 1990s, and this paper updates that knowledge, pointing out that the transfer learning research started more than a decade earlier. This paper reviews the pioneering 1970s research, and addresses important issues relevant for the current transfer learning research. It gives a mathematical model and geometric interpretation of transfer learning, and a measure of transfer learning indicating positive, negative, and no transfer learning. It presents experimental investigation in the mentioned types of transfer learning. And it gives an application of transfer learning in pattern recognition using datasets of images.

Povzetek: Ta članek opisuje delo na področju prenosa učenja v nevronskih omrežjih, opravljeno v sedemdesetih in zgodnjih osemdesetih letih prejšnjega stoletja, ki je prvo publikacijo izdalo leta 1976. V sodobni raziskavi o transfernem učenju obstaja prepričanje, da je pionirsko delo na področju transfernega učenja potekalo v začetku devetdesetih let, in ta članek to znanje posodablja. poudarja, da so se raziskave o transfernem učenju transfernem učenju začele 15 let prej. Ta članek pregleduje raziskave in obravnava pomembna vprašanja za sedanje raziskave o transfernem učenju. Daje matematični model in geometrijsko razlago transfernega učenja. Daje merilo transfernega učenja, vključno s pozitivnim, negativnim in tabula rasa prenosnim učenjem. Predstavlja eksperimentalno raziskovanje omenjenih vrst transfernega učenja. Uporablja prenosno učenje pri prepoznavanju nabora podatkov.

1 Introduction

Transfer learning is a machine learning method where a learning model developed for a first learning task is reused as the starting point for a learning model in a second learning task (Tan et al. 2018). It is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem (Wikipedia > Transfer Learning, October 2020). Often previous learning is referred to as source and the next learning as target (Pratt 1993, Pan and Yang 2010, Weiss et al. 2016). Basically it is using a pre-trained neural network (trained for Task1) for achieving shorter training time (positive transfer learning) in learning Task2. Transfer learning is an emphasized way of learning in contemporary multistage neural networks named deep neural networks (e.g., Goodfellow et al. 2016).

According to (Wikipedia > Transfer Learning, October 2020), the earliest work on transfer in machine learning is attributed to Lorien Pratt (1993). That work points out the earlier work on the subject (Pratt et al. 1991). After 1993, as pointed in Pan and Yang (2010) the fundamental motivation for transfer learning in the field of machine learning was discussed at a NIPS-95 workshop on "Learning to Learn" (Baxter et al. 1995).

In the context described above, this paper informs on an explicit work on transfer learning which took place fifteen years before the Pratt et al. (1991) work. That research, reviewed here, started 1972 producing some unpublished reports (Bozinovski 1972, 1974) and a published report in 1976 (Bozinovski and Fulgosi, 1976) which explicitly in the title addressed the transfer learning concept. Research continued after that, and reports were given in (Bozinovski et al. 1977, Bozinovski 1978, 1981, 1985a, 1985b, 1995).

That initial research on transfer learning is important to the current effort in transfer learning, because in addition of presenting initial concept of transfer learning in neural networks, it describes an early approach of defining a measure of transfer learning which is of interest to current efforts in transfer learning (Tan et al. 2018). The review presented here, in addition to mathematical treatment of transfer learning, describes the experimental investigation on transfer learning which took place during 1976-1981. This paper also gives an application of transfer learning, in obtaining shorter training sequences in learning a dataset of images representing letters.

In the sequel the paper first reviews the neural network used in early research on transfer learning, during 1972-1981. Then it gives a mathematical model of supervised learning, in which it explicitly introduces transfer learning. Then it gives a geometrical model of transfer learning, including positive, negative, and no transfer learning. Then, in Section 5, it defines a mathematical index, a measure of transfer learning. In Section 6 the paper discusses a search for a solution of pattern classification problem in case of negative transfer learning. In Section 7 the paper discusses the multi-class multi-template problem of transfer learning. Section 8 shows results of experimental investigation in transfer learning. It first shows experiments with small set of low resolution images representing letters, demonstrating experimentally the effect of tabula rasa, positive, and negative transfer. The paper then extends to an application of transfer learning in case of learning a dataset of three sets each containing 26 images representing letters. The section 9 reviews the related work by other authors which appeared after 1986, influenced by the renewed interest in neural networks due to the book of Rumelhart, McClelland, and the PDP Group (1986), including the work of Pratt et al. (1991) and Pratt (1993). The paper ends with a discussion and conclusion section.

2 The neural network

The neural network used in our study (Bozinovski 1972, 1974, 1995) is shown in Figure 1.

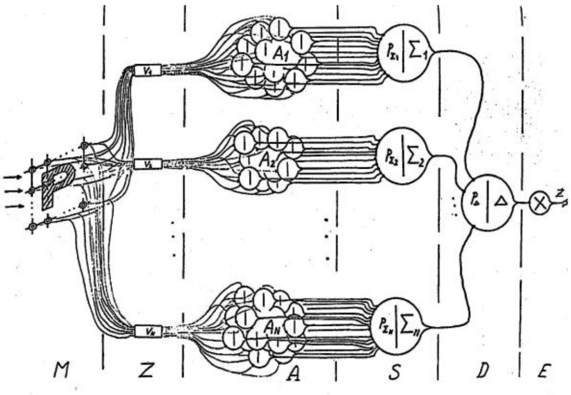


Figure 1: A 5-layer neural network used in supervised learning for pattern recognition in the research described here (Bozinovski 1974, 1995).

The network contains 5 computational stages (layers). The first one, M, is the sensor layer, with sensors arranged according to a need, for example as a matrix retina. Sensors are binary giving values 0 or 1.

Second layer, Z, is a *feature extraction layer*. Feature is a pattern which is used as input in recognition of a higher level pattern. Examples of features might be "horizontal line", "circle", "upper left corner", or a rather complex feature. Important is that the feature is a first stage in recognizing a pattern, which is a set of features. One way of defining a feature is to pre-wire all sensors in a horizontal line and to create an output from Z layer, with interpretation "horizontal line". The other way is to add a Z-element with trainable weights and produce an output with interpretation "horizontal line". The number of outputs from Z-layer is often larger than the number of input sensors. For example each sensor can be considered a feature, plus some needed features such as "middle horizontal line", "left corner" or "square".

The outputs of the layer Z are inputs to the third layer, the A-layer. It contains A-elements, or associative units, as named originally by Rosenblatt (1958, 1962), and used in early neural learning research (e.g. Glushkov, 1967). We will use that term, but we will also use the term associative weights. A weight represents the *relevance* of the feature in creating the concept of a pattern. They are divided into subsets $A_1, A_2, ..., A_n$, each subset having inputs from the feature layer Z. The subsets are associated to a cognitive concept, *a class* to which input patterns are classified. If there are n possible classes and N_s possible features, then each A element can be represented by values w_{is}, i=1,..,n; j=1,..,N_s. They are in general real numbers. Each class of A elements represent a concept, a cognitive class, that will be learned in the pattern classification process. For example, if a task is to classify images, then one set of A elements will be devoted to recognize image "E", another to recognize the image "F" etc.

Next layer, S, are elements that perform some computation over the subsets of A elements representing cognitive classes. An S element s_i computes some function $y(w_{ij})$ over the elements w_{ij} , i = 1,..,n; j =1,.., N_s. Most often these elements compute a weighted and thresholded sum $y_i = \Sigma_j w_{ij}\ x_j - p_{\Sigma i}$ where $p_{\Sigma i}$ is named threshold of the element s_i. Further in the text we will use the θ-notation for threshold , i.e. $p_{\Sigma i} = \theta_i$. There are n S-elements in this layer, $s_1, s_2, ..., s_n$. A subset of A elements and the corresponding S element is named a neuron of the neural network.

The next layer, D, is an *arbiter layer*, which chooses an S-element out of n alternative S-elements. Usual way is computing a maximum function. This layer can be composed by set of neurons which have a common threshold. Such an Isothreshold Neural Network (e.g. Bozinovski 1985a) has a common threshold value equal to maximal value of the individual neuron thresholds. Such a network provides a mechanism for computing maximal value in neural networks. In addition, the maximal value might be normalized to 1, and the maximum computing network can be viewed as computing fuzzy union if the input values are also normalized between 0 and 1. The output of this layer is an integer from 0 to n. For example, output d = 2 means that the observed pattern belongs to class 2 out of the considered n classes. The output d = 0 means that the classification is undecided, possibly there are two S-elements computing the same largest value, so there is no single maximal value.

The next layer E, is output interface layer. It activates some device that is controlled by this neural network. For example if d=2 is computed, then this layer may activate a speech device telling the sound representation of the class 2.

The neural network presented in Figure 1 was the one we started our research in neural networks with. The first task we considered was distinguishing a horizontal vs vertical line on a matrix retina (Bozinovski 1972). That is not reviewed here. This paper is focused on modeling transfer learning.

3 Mathematical modeling of transfer learning in a neural network

For purpose of presenting the concept of transfer learning, here we use a simplified version of the 5-layer network on Figure 1. Let the layer M consists of m synapses or sensors. Let layer Z does not compute any additional feature besides the sensor inputs, so it just represents connections from sensors to A-elements. Let each subset of A elements has the same connections to the sensors as the other subset of A-elements. The A-elements will be named synaptic weights, such that the weight w_{is} represents the s-th synapse element in the i-th class of A elements. Then the S-element s_i computes the function $y_i = \Sigma_s w_{is} x_s - \theta_i$. Let the layer D is represented by a maximum selector function: ($d_i = 1$ if $y_i = \max_i\{y_i\}$ otherwise $d_i = 0$). Other way of denoting a maximum selector is $d = \text{indmax}\{y_i\}$ where indmax{ } returns the index of the maximal element in the considered set. In the literature this function is usually written as d = argmax{ }, but we use our original notation (Bozinovski and Fulgosi, 1976).

3.1 An approach toward modeling supervised learning in neural networks

The principal learning concept of the neural network approach toward machine learning is the concept of (synaptic) weights (e.g. Rumelhart et al. 1986, Goodfellow et al. 2016). In pattern classification with neural networks the principal representation spaces are the *pattern feature space* and *weights space*. However, it should be noted that while in artificial neural nets synaptic weights are observable, in real biological systems they are not observable. So it is interesting to use a representation of the supervised learning problem which will not deal with synaptic weights as primary representation concept. Here we will describe such a representation which is a *weights-free* and we call it *teaching space* (Bozinovski 1981, 1985b).

Let us note that in a supervised learning there is a system named teacher who has a reference model of the knowledge to be transferred in the other system named learner or student. The teaching space approach is based on the following notation:

Let x be a body of knowledge to be learned by the student. For example x might be a visual pattern to be classified in a class. The supervised learning procedure (training) contains both teaching trials (where the teacher presents the knowledge about x), and examination trials (where the student presents its knowledge about x). After the training is completed there will be many exploitation trials, where the learner will show its knowledge in an application.

Let !(x) denotes a teaching (or advising) trial**,** representing the teacher's reference model knowledge about x. Let ?(x) denotes a test (or examination) trial**,** representing the current learner's knowledge about x. Then, the goal of the teaching process becomes

$$?(x) = !(x) \text{ for all considered x.} \qquad (1)$$

The learner we use is a maximum selector classifier (Figure 1). For each input pattern x in an test trial, the learner computes n alternatives, i.e., computes n functions $y_1(x),..,y_n(x)$, and chooses the one with maximal value. If there is no maximal value the learner gives special answer meaning "undecided", for example value 0.

Lets define a set *X* of N objects (patterns), $X=\{x_1,..,x_i,..,x_j,..,x_N\}$, to be classified into n classes, $C_1,..,C_k,..,C_q,..,C_n$, where N $\geq$ n. Let, by teachers reference model, the i-th pattern belongs to the k-th class and j-th pattern belongs to the q-th class. That can be written as

$$!(x_i)= C_k; \quad i=1, \ldots, N; k=1,...,n; \qquad (2.1)$$

$$!(x_j)= C_q; \; j = 1,\ldots, N, q=1,...,n; j \neq i, q\neq k; \qquad (2.2)$$

In an examination trial it is computed the maximum value, which means that the correct classification is achieved if the following pair of inequalities holds

$$?(x_i) = !(x_i) = C_k \;\Leftrightarrow\; y_k(x_i) > y_q(x_i) \qquad (3.1)$$

$$?(x_j) = !(x_j) = C_q \;\Leftrightarrow\; y_q(x_j) > y_k(x_j) \qquad (3.2)$$

Further, we assume that the patterns are represented as feature vectors $\mathbf{x}_1,..,\mathbf{x}_N$ and that the weight vectors are represented with $\mathbf{w}_1,..,\mathbf{w}_n$, where $\mathbf{w}_k$ is associated with the class C_k.

The learning process is governed by a consequence driven teaching process with an *error correction learning rule*

$$\text{if } ?(x_i) \text{ is different than } (!(x_i) = C_k)$$
$$\text{then correct } \mathbf{w}_k \text{ toward } \mathbf{x}_i\text{: } \mathbf{w}_k = \mathbf{w}_k + c\mathbf{x}_i \qquad (4)$$

where c is a constant. In words, if the classifier erroneously classifies the pattern $\mathbf{x}_i$ in an test trial, a teaching trial is introduced in which the pattern $\mathbf{x}_i$ is added to the weight $\mathbf{w}_k$, lecturing that $\mathbf{x}_i$ belongs to C_k.

Here c is a learning rate which is a constant and we use the value c=1.

3.2 Introducing transfer learning

Consider neural network from Figure 1 which has capability to classify N patterns into n classes, N $\geq$ n. Consider the simplest task, two patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ to be classified into two classes C_k and C_q. The problem is stated with relations (3). However let us emphasize that k and q *are arbitrary* in the set of {1,...,n| k≠q}, and also i and j *are arbitrary* in the set {1,...,N| i≠j}.

Now we *introduce transfer learning*. Let assume the considered neural network has been subject of a learning task which we call *first learning task*. After that first learning task the neural network learner is now subject to a *second learning task*. The second learning task will be carried out by a supervised learning (or teaching) process represented by a teaching sequence L. The teaching sequence contains both teaching and test (examination) trials. However, the memory of the learner is updated only during the teaching trials. The test trials demonstrate the knowledge already stored in the memory of the neural network learner.

Let $y_k(\mathbf{x}_i)$ be the output of S_k element of the neural network at the completion of the first learning task. It is the initial knowledge as demonstrated by this neural network before the second learning task. We emphasize that with notation $y_k^0(\mathbf{x}_i) := y_k(\mathbf{x}_i)$, pointing out with a superscript 0 that it is initial knowledge for the second learning task. So the output $y_k^0(\mathbf{x}_i)$ manifests *the transfer learning from the first teaching task* about the concept class k, before the second teaching task with teaching sequence L is applied.

Let $y_k(\mathbf{x}_i/L)$ be the output of S_k element representing class k when shown pattern $\mathbf{x}_i$ after the *learning in the second task* with the learning sequence L. So the second learning task will be modeled with the following outputs from elements S_k and S_q

$$y_k(\mathbf{x}_i/L) = y_k^0(\mathbf{x}_i)+ a_{ij}p_i \tag{5.1}$$

$$y_q(\mathbf{x}_j/L) = y_q^0(\mathbf{x}_j)+ a_{ji}p_j \tag{5.2}$$

where p_i is the number of appearance of pattern $\mathbf{x}$ in a teaching trial of the teaching sequence L, i.e. the number of application of the learning rule (4), and a_{ij} is the inner product between patterns, $a_{ij} = \mathbf{x}_i^T\mathbf{x}_j$.

So, in order a correct pattern classification to be achieved in the second task, by a neural network with maximum selector as in Figure 1, it is necessary and sufficient that the following system of inequalities holds

$$y_k(\mathbf{x}_i/L) > y_q(\mathbf{x}_i/L) \tag{6.1}$$

$$y_q(\mathbf{x}_j/L) > y_k(\mathbf{x}_j/L) \tag{6.2}$$

which leads to

$$a_{ii}\, p_i - a_{ij}\, p_j > - y_k^0(\mathbf{x}_i) + y_q^0(\mathbf{x}_i) \tag{7.1}$$

$$-a_{ji}\, p_i + a_{jj}\, p_j > y_k^0(\mathbf{x}_j) - y_q^0(\mathbf{x}_j) \tag{7.2}$$

That reasoning leads to the following Theorem:

Theorem 1. (Transfer learning in case of learning arbitrary two patterns from a set of patterns) Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be arbitrary patterns from a set $X =\{\mathbf{x}_1,..,\mathbf{x}_i,..,\mathbf{x}_j,..,\mathbf{x}_N\}$ of N patterns, which a maximum selecting neural classifier should learn to classify into given two classes C_k and C_m respectively, from a set $C=\{C_1,..,C_k,..,C_m,..,C_n\}$ of n classes. Let $a_{ij} = \mathbf{x}_i^T\mathbf{x}_j$. Let the lecture (teaching trial) $!x_i = C_k$ is presented p_i times, and let $!x_j = C_m$ is presented p_j times in the teaching sequence L. Then, the problem of correct classification learning is equivalent to the problem of finding p_i and p_j which satisfy the pair of inequalities

$$\begin{pmatrix} a_{ii} & -a_{ji} \\ -a_{ij} & a_{jj} \end{pmatrix} \begin{pmatrix} p_i \\ p_j \end{pmatrix} > \begin{pmatrix} \tau_{qk}(x_i) \\ \tau_{kq}(x_j) \end{pmatrix} \tag{8}$$

which in compact form can be written as

$$\mathbf{Ap} > \boldsymbol{\tau} \tag{9}$$

where

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_{qk}(x_i) \\ \tau_{kq}(x_j) \end{pmatrix} = \begin{pmatrix} y_q^0(\mathbf{x}_i) - y_k^0(\mathbf{x}_i) \\ y_k^0(\mathbf{x}_j) - y_q^0(\mathbf{x}_j) \end{pmatrix} \tag{10}$$

is named transfer learning vector.

Before we present the proof of the Theorem 1 we will give interpretation of the variables which appear in the theorem.

First we point out that the left side of the inequalities (8) contain a matrix of all inner products between patterns. The inner product a_{ij} between two patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ shows how much their features overlap. It can be viewed as covariance, a *manifestation of pattern similarity.* We denote that matrix $\mathbf{A} = [a_{ij}]$, and name it a *matrix of mutual similarity between patterns*. Note that this matrix is invariant to the teaching process, it simply describes relation between the given patterns.

The vector $\mathbf{p}=(p_i\, p_j)^T$ shows how many times patterns were shown in a teaching trial in the teaching sequence L. It is a *training vector of the second learning task.*

The right side of inequalities contain the variables are due to *transfer learning* from a learning task prior to this considered task of training using curriculum L. It contains differences of outputs of S-elements for each pattern shown in the teaching process, i.e. $y_{qk}^0(\mathbf{x}_i) = y_q^0(\mathbf{x}_i) - y_k^0(\mathbf{x}_i)$ for shown pattern $\mathbf{x}_i$, and $y_{kq}^0(\mathbf{x}_j) = y_k^0(\mathbf{x}_j) - y_q^0(\mathbf{x}_j)$ for shown pattern $\mathbf{x}_j$.

So the left side of matrix inequalities, $\mathbf{Ap}$, contains all controllable and observable parameters of the teaching process. If patterns are known, the matrix $\mathbf{A}$ is known. The teaching sequence L is the one it is looked for, and after it is found, the vector $\mathbf{p}$ will be known. However, the right side of inequalities, vector $\boldsymbol{\tau}$, which represents *transfer learning, is in general case not known.* Teaching of a biological brain does not assume that initial values of weights are known. Often the task is to teach a learner regardless the transfer learning. However, because of unknown transfer learning teaching process might converge in a longer time.

The proof of the Theorem can be expressed using a reasoning flow diagram as shown in Figure 2. The equations and inequalities used have been already described in the text.

Note that if all thresholds in the network are equal, then the transfer learning can be expressed as

$$\tau_{kq}(\mathbf{x}_j) = (\mathbf{w}_k^0\text{-}\mathbf{w}_q^0)\mathbf{x}_j \tag{11}$$

3.3 Modeling positive and negative transfer learning

In this section we will address formally the following questions.

Given a neural network that has been subject to a learning Task1, is it possible to find a teaching sequence L which will solve the teaching Task2 regardless the transfer learning from Task1?

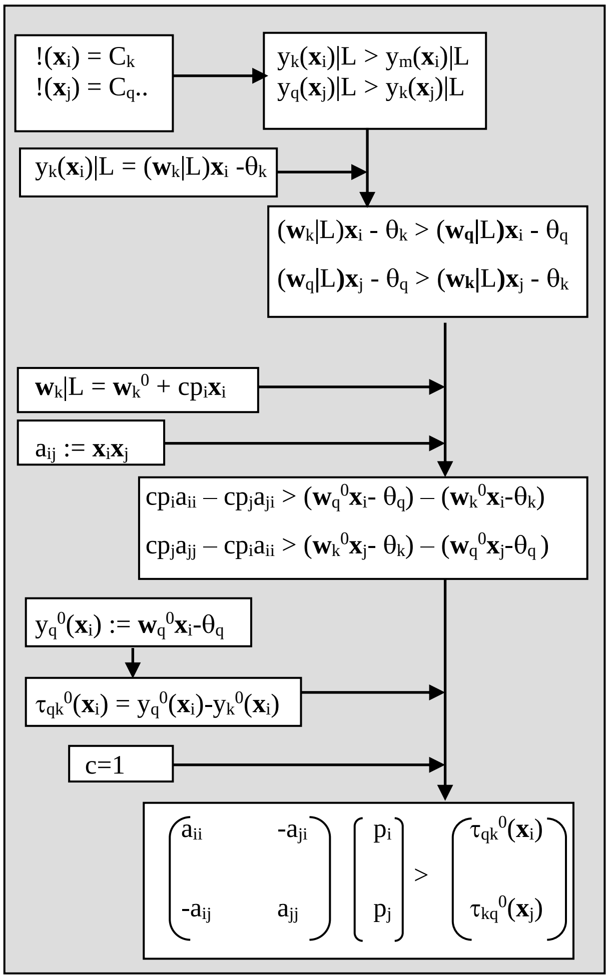

$!(\mathbf{x}_i) = C_k$
$!(\mathbf{x}_j) = C_q..$
$y_k(\mathbf{x}_i)|L > y_m(\mathbf{x}_i)|L$
$y_q(\mathbf{x}_j)|L > y_k(\mathbf{x}_j)|L$
$y_k(\mathbf{x}_i)|L = (\mathbf{w}_k|L)\mathbf{x}_i - \theta_k$
$(\mathbf{w}_k|L)\mathbf{x}_i - \theta_k > (\mathbf{w}_q|L)\mathbf{x}_i - \theta_q$
$(\mathbf{w}_q|L)\mathbf{x}_j - \theta_q > (\mathbf{w}_k|L)\mathbf{x}_j - \theta_k$
$\mathbf{w}_k|L = \mathbf{w}_k^0 + cp_i\mathbf{x}_i$
$a_{ij} := \mathbf{x}_i\mathbf{x}_j$
$cp_ia_{ii} – cp_ja_{ji} > (\mathbf{w}_q^0\mathbf{x}_i - \theta_q) – (\mathbf{w}_k^0\mathbf{x}_i - \theta_k)$
$cp_ja_{jj} – cp_ia_{ii} > (\mathbf{w}_k^0\mathbf{x}_j - \theta_k) – (\mathbf{w}_q^0\mathbf{x}_j - \theta_q)$
$y_q^0(\mathbf{x}_i) := \mathbf{w}_q^0\mathbf{x}_i - \theta_q$
$\tau_{qk}^0(\mathbf{x}_i) = y_q^0(\mathbf{x}_i) - y_k^0(\mathbf{x}_i)$
$c=1$
$\begin{pmatrix} a_{ii} & -a_{ji} \\ -a_{ij} & a_{jj} \end{pmatrix} \begin{bmatrix} p_i \\ p_j \end{bmatrix} > \begin{pmatrix} \tau_{qk}^0(\mathbf{x}_i) \\ \tau_{kq}^0(\mathbf{x}_j) \end{pmatrix}$

Figure 2: Proof of the Theorem 1 in a reasoning flow representation.

Case of positive transfer of learning. Is it possible that Task1 helps achieving shorter sequence L in Task2, than if starting from no previous transfer of learning?

Case of negative transfer of learning. Is it possible that Task1 will produce a longer sequence L in Task2, than if starting from no previous transfer of learning?

In order to answer those questions we will further elaborate on the inequalities (7). We repeat them here for clarity and renumber them (12) for keeping the sequence:

$$a_{ii}\, p_i - a_{ij}\, p_j > - y_k^0(\mathbf{x}_i) + y_q^0(\mathbf{x}_i) \qquad (12.1)$$

$$-a_{ji}\, p_i + a_{jj}\, p_j > y_k^0(\mathbf{x}_j) - y_q^0(\mathbf{x}_j) \qquad (12.2)$$

The inequalities (12) can be rewritten to see explicitly how p_j depends on p_i. To see that, we move terms with p_i on the right side of the inequalities (12) and we obtain the following system of inequalities:

$$- a_{ij}\, p_j > - a_{ii}\, p_i - y_k^0(\mathbf{x}_i) + y_q^0(\mathbf{x}_i) \qquad (13.1)$$

$$a_{jj}\, p_j > a_{ji}\, p_i + y_k^0(\mathbf{x}_j) - y_q^0(\mathbf{x}_j) \qquad (13.2)$$

Now we multiply equation (13.1) with -1, which changes the inequality sign from > to <. We obtain the following system of inequalities:

$$a_{ij}\, p_j < a_{ii}\, p_i + y_k^0(\mathbf{x}_i) - y_q^0(\mathbf{x}_i) \qquad (14.1)$$

$$a_{jj}\, p_j > a_{ji}\, p_i + y_k^0(\mathbf{x}_j) - y_q^0(\mathbf{x}_j)\, . \qquad (14.2)$$

where from

$$p_j < (a_{ii} / a_{ij})\, p_i + (y_k^0(\mathbf{x}_i) - y_q^0(\mathbf{x}_i))/a_{ij} \qquad (15.1)$$

$$p_j > (a_{ji} / a_{jj})\, p_i + (y_k^0(\mathbf{x}_j) - y_q^0(\mathbf{x}_j))/a_{jj} \qquad (15.2)$$

and finally

$$p_j < (a_{ii} / a_{ij})\, p_i + \tau_{kq}(\mathbf{x}_i) / a_{ij} \qquad (16.1)$$

$$p_j > (a_{ji} / a_{jj})\, p_i + \tau_{kq}(\mathbf{x}_j) / a_{jj} \qquad (16.2)$$

These inequalities can be observed geometrically as in Figure 3.

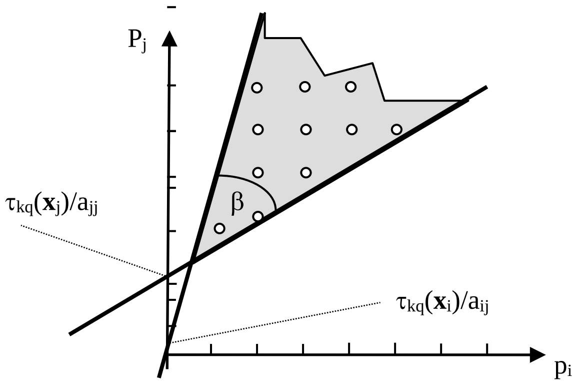

P_j
β
$\tau_{kq}(\mathbf{x}_j)/a_{jj}$
$\tau_{kq}(\mathbf{x}_i)/a_{ij}$
p_i

Figure 3: Geometric interpretation of Theorem 1.

Note that because $a_{ij} = \mathbf{x}_i^T\mathbf{x}_j$, the coefficient $a_{ii}/a_{ij} \geq 1$, and the coefficient $a_{ji}/a_{jj} \leq 1$. Because $\mathbf{x}_i \neq \mathbf{x}_j$ those coefficients are never at the same time equal 1. Because coefficients $a_{ii}/a_{ij} \geq 1$ and $a_{ji}/a_{jj} \leq 1$ are slopes of the boundaries of the solution region, it means because patterns are different, $\mathbf{x}_i \neq \mathbf{x}_j$, the angle β on Figure 3 always exists, and the solution points for (p_i, p_j) inside the shaded region defined by the angle β always exist.

So we can formulate the following statement.

Theorem 2. It is always possible to chose a teaching sequence L which will contains patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ ($\mathbf{x}_i \neq \mathbf{x}_j$), such that after training with L the learner is able to correctly classify the patterns regardless transfer of learning from a previous learning task.

The proof is given in the previous reasoning using equations (12)-(16).

The teaching space in which we observe transfer training is an *integer space*. The components p_i and p_j are non-negative integers. In Figure 3 it is shown that only the integer points are solutions for correct classification of $\mathbf{x}_i$ and $\mathbf{x}_j$.

4 Geometric interpretation of transfer learning: positive, negative, and tabula rasa

From Figure 3 we can see that the solution region of the teaching process is a convex cone defined by two parameters: 1) the position of the coordinate origin relative to the vertex of the cone, and 2) the angle of the convex cone. The orientation of the cone is always such that most of it lies within the first quadrant, although the vertex may be in any quadrant. We call this a positive convex cone.

The angle of the convex cone is determined solely by inner products between patterns. The angle represents the similarity between patterns in a sense of overlapping features.

Transfer learning is geometrically represented by the position of origin of the coordinate space (p_i, p_j) relative to the convex cone. That is illustrated in Figure 4.

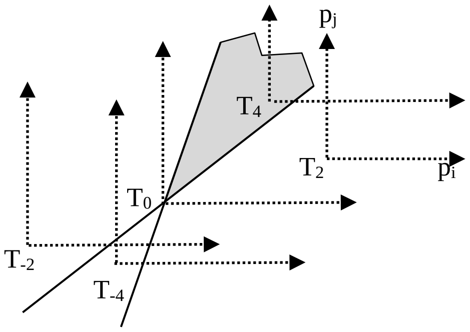


Figure 4: A geometric interpretation of transfer learning. The plane (p_i, p_j), the convex cone, and various coordinate origins representing transfer learning from a previous learning task.

As Figure 4 shows, if the peak of the convex cone is in the coordinate origin (coordinate system T_0), then there is no transfer learning. The learner starts from tabula rasa initial conditions. It means that the memory values are all equal, for example all zero. However they are not necessary zero, they need only to be all equal (homogenous initial conditions). In this condition a learning process must take place for both patterns (or lessons) x_i and x_j in order the learner to correctly recognize those patterns.

If the coordinate origin is inside the solution region (coordinate system T_4) the learner has positive transfer learning. There is no need of additional learning. The previous learning is enough for the correct recognition of the patterns.

If the coordinate origin is in region symmetrically opposite the solution region, (negative convex cone), it is an example of negative transfer learning. Coordinate system T_{-4} is such a case. Both patterns x_i and x_j have been previously, in Task1, classified into classes which are incorrect according to the new Task2. So the new learning process must include both patterns. The learning process will be longer than in case of tabula rasa condition.

If the coordinate system is in the area outside the positive and negative convex cones (examples T_2 and T_{-2} coordinate systems), then there are situations in which for one pattern there is positive transfer learning and for the other is negative.

5 Defining an index of transfer learning in a neural network

Based on the geometrical interpretation of transfer learning in Figure 4 we will now define an index of transfer learning, a numerical representation of transfer learning. Measure of negative transfer as well as transferability measure are emphasized in contemporary transfer learning research (Tan et al. 2018). The index which we will discuss here is proposed in (Bozinovski and Fulgosi 1976).

The mathematical measure of transfer learning was introduced using the following reasoning. Observe the segments the lines in Figure 4 define intercepting with ordinate p_j. For T_0 coordinate system both lines have intercept 0. For coordinate system T_4, one intercept is positive (for the boundary line $p_j > p_i$) and the other is negative (for the boundary line $p_j < p_i$). For coordinate system T_2 one intercept is positive and the other is negative. For T_1 both intercepts are negative. For T_3 both intercepts are positive. Note that also in Figure 3 above, it is shown a case of both positive intercepts. So we will only observe the sign of the intercepts, positive, negative, or zero, and we will define index of transfer learning.

Note that the intercepts are defined as $\tau_{kq}(\mathbf{x}_i)/a_{ij}$ and $\tau_{kq}(\mathbf{x}_j)/a_{jj}$ and consequently their signs are defined as $\mathrm{sign}(\tau_{kq}(\mathbf{x}_i)/a_{ij})$ and $\mathrm{sign}(\tau_{kq}(\mathbf{x}_j)/a_{jj})$ where sign( ) is a function that gives 1 for positive, 0 for zero, and -1 for negative argument. Now we can define an index, a measure of transfer learning on the basis of signs of intercepts of the boundary lines for patterns $\mathbf{x}_i$ and $\mathbf{x}_j$.

$$TL(\tau_{kq}(\mathbf{x}_i), \tau_{kq}(\mathbf{x}_j)) = 3\mathrm{sign}\ (\tau_{kq}(\mathbf{x}_i) - \mathrm{sign}(\tau_{kq}(\mathbf{x}_j) \qquad (17)$$

According to this index, if both signs are positive then TL =2. That corresponds to a coordinate system T_2 in Fig. 4. If both are negative, then TL = -2 and that corresponds to the coordinate system T_{-2} is Fig 4. If both are zero, then TL=0, which corresponds to coordinate system T_0 in Fig. 4. If sign ($\tau_{kq}(\mathbf{x}_i)$ =1 and $\mathrm{sign}(\tau_{kq}(\mathbf{x}_j)$ = -1, then TL = 4 which corresponds to coordinate system T_4 in Fig. 4.

Note that the index TL considers all the integer values in the interval [-4,+4]. Figure 5 shows the TL values and their geometric interpretation.

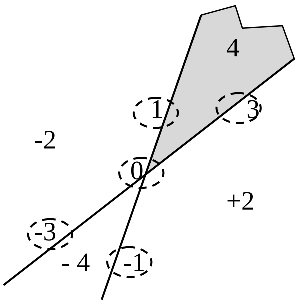


Figure 5: A geometric interpretation of index TL, a numerical index of transfer learning. It shows the values of the regions of the (p_i, p_j) plane where a learner finds itself after the first learning task Task1, and facing the second learning task Task2.

The introduced index of transfer learning shows position of the coordinate origin in the plane (p_i, p_j) relative to the peak of the vertex inside which is a solution of the pattern recognition problem. It shows where in the (p_i, p_j) plane is the starting point to learn Task2 by a learner with transfer learning from previous Task1. From Fig. 5 we can give following interpretations for transfer learning index TL:

If TL = 4 the learner correctly classifies both patterns, without need for additional learning. It is a positive transfer learning from a previous Task1.

If TL = ±1 or ±3 the learner recognizes one pattern but is *undecided* about the other. The coordinate origin lies on a boundary line of inequalities. For example, if TL=3 the coordinate origin lies on the right boundary line of the positive convex cone. In such a case, if the convex cine angle is not too small. then only one presentation of the pattern $\mathbf{x}_j$ in a teaching trial is enough that the learner correctly classify both patterns.

If TL = 0 the learner is *undecided about both patterns*. There is no transfer of a previous learning, the learner is in tabula rasa condition..

If TL= - 4 the leaner incorrectly classifies both patterns. It is example of negative transfer learning.

If TL= ±2 the learner correctly classifies one pattern but incorrectly the other one. In this case there is a transfer learning, positive for one pattern but negative for other one.

The considered index of transfer learning (17) can be normalized for value between -1 and 1 if the right side of equation (17) is divided by 4.

6 Search for a learning solution in case of negative transfer learning

To illustrate further the learning process including transfer learning, we will consider the search for a learning solution in case of negative transfer learning. Figure 6 shows such an illustration.

First let us note that the orientation of the solution convex cone in space is *regardless* of the transfer learning. The solution cone orientation depends solely on the considered patterns and their mutual position on a medium they are shown. If the patterns are digital images on a binary retina, then their mutual overlapping $a_{ij} = a_{ji}$ and self overlapping a_{ii} and a_{jj} will define the solution region. As example, imagine image patterns E, T, and F on the retina of 7x5 binary sensors.

The considered learning Task2 in Figure 6 can have different coordinate origins, due to a transfer learning. Consequently, a learning process will have different trajectory in the Task2 teaching space, depending on transfer learning from Task1.

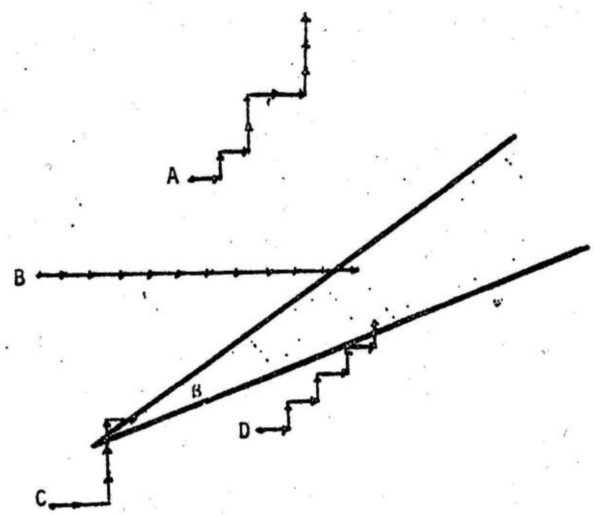


Figure 6: Some learning trajectories in teaching space of Task2, due to transfer learning from a Task1 (Bozinovski 1981).

It can be seen from Figure 6 that due to a negative transfer learning it is possible that a teaching sequence L never finds a pattern classification solution, as is the case with learning trajectory starting with initial condition A. The other cases of negative transfer learning can be compensated with carefully chosen teaching sequence L, as shown with teaching sequences B, C, and D. In case of initial condition B, it is enough that only the pattern x_i is shown several times until a solution point is found. On case of initial condition C both patterns must be shown for correct classification. In case of initial condition D, it is shown that a teaching sequence containing equal number of x_i and x_j will eventually reach a solution region. However, one can observe that also a sequence containing only x_j will eventually reach the solution region.

7 Multi-class, multi-template task

Pattern classification usually assumes several template patterns for each class to be included in the teaching process. In the test task (or in exploitation task) there might be patterns that are not shown as template patterns.

In this section we will discuss two topics. First is how the model given by Theorem 1 applies in case of several templates for a class, and second is how transfer learning is represented in the synaptic weights in an artificial neural network. As opposite to natural neural networks where weights are not observable, in artificial neural networks usually it is assumed that the synaptic weights are observable.

Consider a task in which three patterns are to be classified into two classes: $\mathbf{x}_1$, $\mathbf{x}_2 \in C_1$, $\mathbf{x}_3 \in C_2$. The two neurons associated with the two classes have weight vectors $\mathbf{w}_1$ and $\mathbf{w}_2$, and thresholds θ_1 and θ_2 respectively. The maximum selector layer for each presented pattern computes the following inequalities:

$$\begin{aligned}&(\mathbf{x}_1 \in C_1)\text{: } \mathbf{w}_1\mathbf{x}_1 - \theta_1 > \mathbf{w}_2\mathbf{x}_1 - \theta_2\\&(\mathbf{x}_2 \in C_1)\text{: } \mathbf{w}_1\mathbf{x}_2 - \theta_1 > \mathbf{w}_2\mathbf{x}_2 - \theta_2\\&(\mathbf{x}_3 \in C_2)\text{: } \mathbf{w}_2\mathbf{x}_3 - \theta_2 > \mathbf{w}_1\mathbf{x}_3 - \theta_1\end{aligned} \qquad (18)$$

In case of transfer learning, where weights have initial values $\mathbf{w}^0_i$ (i=1,2) we have

$$\begin{aligned}&(\mathbf{x}_1/L)\text{: } (\mathbf{w}^0_1+p_1\mathbf{x}_1+p_2\mathbf{x}_2)\mathbf{x}_1 - \theta_1 > (\mathbf{w}^0_2 + p_3\mathbf{x}_3)\mathbf{x}_1 - \theta_2\\&(\mathbf{x}_2/L)\text{: } (\mathbf{w}^0_1+p_1\mathbf{x}_1+p_2\mathbf{x}_2)\mathbf{x}_2 - \theta_1 > (\mathbf{w}^0_2+p_3\mathbf{x}_3)\mathbf{x}_2 - \theta_2 \qquad (19)\\&(\mathbf{x}_3/L)\text{: } (\mathbf{w}^0_2+p_3\mathbf{x}_3)\mathbf{x}_3 \quad - \theta_2 > (\mathbf{w}^0_1+p_1\mathbf{x}_1+p_2\mathbf{x}_2)\mathbf{x}_3 - \theta_1 .\end{aligned}$$

After rearrangement, and introducing $\mathbf{w}^0_{kq} = \mathbf{w}^0_k\text{-}\mathbf{w}^0_q$ and $\theta_{kq} = \theta_k - \theta_q$, where k, q $\in$ {1, 2} and k≠q, we obtain matrix representation of the classification problem which includes transfer weights

$$\begin{array}{l}(\mathbf{x}_1/L)\text{:}\\(\mathbf{x}_2/L)\text{:}\\(\mathbf{x}_3/L)\text{:}\end{array}\begin{pmatrix} a_{11} & a_{21} & -a_{31}\\ a_{12} & a_{22} & -a_{32}\\ -a_{13} & -a_{23} & a_{33}\end{pmatrix}\begin{pmatrix}p_1\\p_2\\p_3\end{pmatrix} > \begin{pmatrix}\mathbf{w}^0_{21} & 0 & 0\\ 0 & \mathbf{w}^0_{21} & 0\\ 0 & 0 & \mathbf{w}^0_{12}\end{pmatrix}\begin{pmatrix}\mathbf{x}_1\\\mathbf{x}_2\\\mathbf{x}_3\end{pmatrix} + \begin{pmatrix}\theta_{12}\\\theta_{12}\\\theta_{21}\end{pmatrix} \qquad (20)$$

The shaded areas are diagonal sub-matrices of classes. Each class sub-matrix has number of rows (and columns) equal to number of templates for that class. In case of inequalities (20), the first class contains two templates and the second contains one template pattern. From this case study we can generalize the transfer learning model for a multi-class and multi-template per class case as

$$(X/L): \mathbf{Ap} > \mathbf{W}^0\mathbf{X} + \boldsymbol{\theta} \qquad (21)$$

where $\mathbf{X} = \{\mathbf{x}_1,..,\mathbf{x}_N\}$is the set of patterns which should be learned in the second task with the curriculum sequence L.

Note that the mathematical model of transfer learning (21) divides the left side of relation to be a *teacher's side*, and right side a *learner's side*.

At the teacher side are similarity matrix A and distribution vector p showing how many times each pattern appeared in a teaching trial of the curriculum L. Matrix A shows that what matters in the teaching process are not the patterns themselves but rather their correlations, inner products, which can be interpreted as similarities.

At the learner side, $\mathbf{W}^0$ represents difference of initial conditions of the memory due to transfer learning, $\mathbf{X}$ is the vector of template vectors, a matrix containing patterns to be classified, and $\boldsymbol{\theta}$ represents difference between thresholds of neurons representing classes. Note that the matrix $\mathbf{W}^0$ contains blocks showing which template is assigned to which class.

As pointed before, the space $p = (p_1,..,p_N)$ is an integer space. Dealing with neural network learning is actually an *integer programming problem*. We are interested in the most efficient training, and we are looking for a training sequence L of the minimal length. So we look for a criterion

$$p_1+p_2+...+p_N = \min \qquad (22)$$

Such a criterion will observe the appearance of patterns only in a teaching trial. If we are interested in minimal sequence that includes test trials, then the optimality criterion is

$$(p_1+q_1) + (p_2+q_2) +...+(p_N+q_N) = \min \qquad (23)$$

where q_i is number of appearances of the pattern $\mathbf{x}_i$ in a test trial, which does not change the memory of the learner, but affects the length of the training sequence L.

8 Experimental investigation on transfer learning

Experimental investigation on transfer learning was carried out in the period 1976-1981. Initial experiments was with a dataset containing images of letters A, B, E, F, and T taken from the terminal IBM29 card puncher. Those experiments were carried out on the computer IBM 1130. Later experiments were carried out with two datasets. One dataset contained 40 images, consisting of 26 letters, 10 numbers, and 4 special symbols from the terminal IBM29. The other dataset can be described as Computer Terminals dataset, consisting of 3x26= 78 images, taken from three computer terminals: IBM29 card puncher, VR14 video screen, and VT50 video screen. The experiments were carried out on a computer VAX/VMS. Figure 7 shows the Computer Terminals dataset. As can be seen, the letters of the three terminals are mostly identical on an image with resolution 7x5, with differences in letters A, B, D, G, J, M, N, O, V, and W.

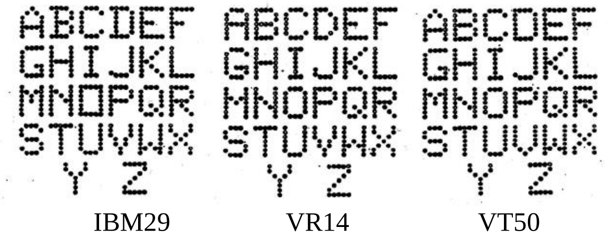


Figure 7: The dataset Computer Terminals used in experimental investigation.

8.1 An experiment in tabula rasa condition, showing influence of pattern similarity

Here we will show an experiment in tabula rasa learning , to see the influence of similarity (overlapping pattern features) on the learning process. Consider the patterns E, T, F, shown in Fig. 7. They are the same for all considered terminals. Figure 8 shows the search through the (p_E, p_T, p_F) space that the learning process performs.

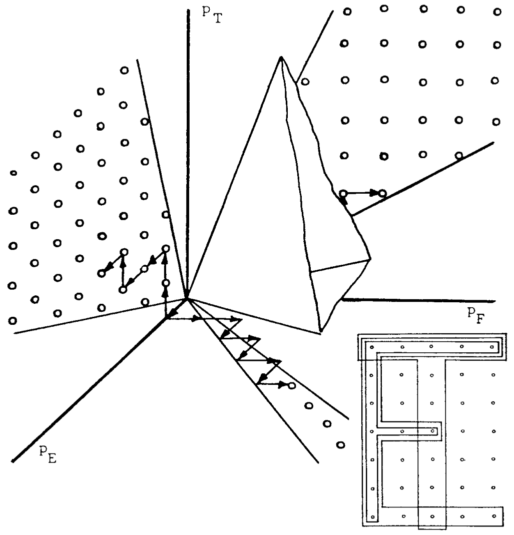


Figure 8: Learning trajectory in case of tabula rasa learner, learning similar patterns E and F, together with the pattern T (Bozinovski 1981, 1985b).

As Fig. 8 shows, the problem is the distinction between the patterns F and E. The convex cone angle is narrow, and it is possible that in some search steps the cone does not contain an integer point. The search for an integer solution is what makes necessary to repeat images E and F several times until they are distinguished by the learner.

This experiment emphasizes the problem of feature overlapping and the problem of one image included in another image. To emphasize the image-subimage relation, a measure of similarity between patterns is introduced in (Bozinovski and Fulgosi 1976). The following index

$$SL(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j/\min\{\mathbf{x}_i^T\mathbf{x}_i, \mathbf{x}_j^T\mathbf{x}_j\} \qquad (24)$$

has values between 0 and 1. If SL = 0 the solution convex cone covers the entire first quadrant of the teaching space (p_i, p_j). If 0 < SL < 1, the convex cone includes the line p_j=p_i. If SL = 1, one of the cone boundaries is the line $p_j = p_i$.

Such a measure is used to predict the length of the teaching sequence L. and with that the efficiency of the training.

8.2 Experimental investigation in positive and negative transfer learning

Experiments shown here are carried out during 1976-1978 on a IBM1130 computer. Table I shows the results of the transfer learning experiments which show both positive and negative transfer. (Bozinovski et al. 1977, Bozinovski 1978).

Task 1 Images	Task 2 Images	Task 2 Teaching sequence
No transfer learning, tabula rasa		
	A, B	AB.
	A, B, T	ABT.
	E, F	EFFEFEFEF.
	E, F, T	EFFEFEFEFTTT.
Negative transfer learning		
E, F	A, B	ABABABAB.
Positive transfer learning		
A, B, T	E, F, T	EFT.

Table 1: Experiments in transfer learning. Cases of tabula rasa, negative, and positive transfer learning.

In presenting the results of the experiments with transfer learning here we introduce the notation $L_{D2/D1}$, meaning training sequence of Task2, trained with a set of patterns D2, after the Task1 in which the learner is trained with a set of patterns D1. For a tabula rasa training, we use notation $L_{D2/\varnothing}$.

Experiment with no transfer learning. As can be seen from the presented experiments, learning the patterns E,T, and F with no transfer learning needs the teaching sequence $L_{ETF/\varnothing}$ = EFFEFEFEFTTT. The length of the sequence is due to similarity between E and F.

Experiment showing positive transfer learning. If the neural network is previously exposed to the Task1 where it learned to recognize A and B, and after that is exposed to Task2 to learn E, T, and F, then the teaching sequence for Task2 is $L_{EFT/ABT}$ = EFT. The teaching sequence for learning E, T, F in this case is *shorter than in case of tabula rasa*. That is experimental evidence of positive transfer learning.

Experiment showing negative transfer learning. If the neural network is previously exposed to a Task1 to learn E and F, and after that in Task2 to learn A and B, the teaching sequence for learning A and B is $L_{AB/EF}$ = ABABABAB. *It is longer than in case of learning A and B in tabula rasa condition*, $L_{AB/\varnothing}$ = AB. That is an experimental evidence of negative transfer learning.

8.3 Application of transfer learning

Here we show results of experiments carried out during 1980-1981 on a VAX/VMS computer (Bozinovski 1981). The experiments consider real application, learning to recognize letters from computer terminals.

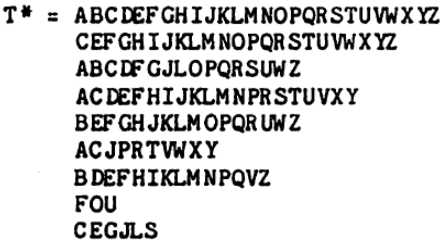
Consider the dataset Computer Terminals from Figure 7. The question we would like to answer experimentally is: If in the Task1 we teach a learner to recognize the letters from the terminal VR14, how faster the learner will be able to learn in Task2 to recognize the letters from the terminal IBM29, comparatively to learning from tabula rasa condition.

In these experiments we used the following teaching strategy (Bozinovski 1981) named perceptron teaching strategy:

```
Procedure PerceptronTeachingStrategy
iteration: teachflag = 0;
    i:=0; n=26;
    while i < n do
            i:=i+1
            grade = test($\mathbf{x}_i$);
            if grade = 'incorrect"
                    then teach($\mathbf{x}_i$), teachflag=1;
    endwhile;
    if teachflag = 1 goto iteration;
end.
```

This strategy performs test trials on all n=26 images, and only when needed, a teaching trial is applied for a particular image. After such an iteration (or epoch), another iteration takes place, and so on, until no teaching trial appeared in an iteration (teachflag=0). That means there were only test trials in the last iteration and the learner now recognizes all the patterns correctly.

Using this strategy applied to the set of letters IBM29, in case of tabula rasa, it gives the 9 iterations as shown in Figure 9.



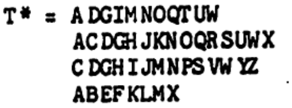
```
T* = ABCDEFGHIJKLMNOPQRSTUVWXYZ
     CEFGHIJKLMNOPQRSTUVWXYZ
     ABCDFGJLOPQRSUWZ
     ACDEFHIJKLMNPRSTUVXY
     BEFGHJKLMOPQRUWZ
     ACJPRTVWXY
     BDEFHIKLMNPQVZ
     FOU
     CEGJLS
```

Figure 9: Teaching sequence of learning the set of letters IBM29 with no transfer learning.

With T* we denote the solution teaching sequence in which only the teaching trials appear. With |T*| we denote its length, in trials. With C* we denote teaching sequence containing both teaching and test trials, and with |C*| its length. For the experiment on Fig. 9 we obtained

$|T^*|_{IBM29/\varnothing} = 135$ and $|C^*|_{IBM29/\varnothing} = 395$.

If before learning the set IBM29 in Task2, the set VR14 was learned in Task1, then in Task2 the teaching process completes in 4 iterations, with the teaching sequence shown in Fig. 10.



```
T* = A DGIMNOQTUW
     ACDGH JKNOQRSUWX
     C DGHIJMNPSVW YZ
     ABEFKLMX
```

Figure 10: The teaching sequence in case when set IBM29 is leaned, providing that previously was learned the set VR14.

In the experiment shown in Fig. 10 we obtained

$|T^*|_{IBM29/VR14} = 48$ and $|C^*|_{IBM29/VR14} = 178$.

The experiment shown in Figure 9 and 10 shows an *application* of positive transfer learning. We obtained shorter training sequence

$|C^*|_{IBM29/VR14} = 178 < |C^*|_{IBM29/\varnothing} = 395$.

The teaching time is 178/395 = 0.45 of the tabula rasa teaching time, and the speed of learning increases 1/0.45 = 2.2 times.

When we carried out an experiment of learning the set VT50 if previously learned the set VR14, the result was

$|T^*|_{VT50/\varnothing} = 207$, $|T^*|_{VT50/VR14} = 43$,

$|C^*|_{VT50/VR14} = 199 < |C^*|_{VT50/\varnothing} = 545$

The transfer learning teaching time is 199/545 = 0.36 of the tabula rasa teaching time, and the speed of learning increases 1/0.36 = 2.8 times.

This application shows the reason of use of transfer learning. If you have a knowledge of a dataset classification stored in a neural network in Task1, then transfer that knowledge to a different task which learns classification of a similar dataset. The training time will be shorter.

Here in this application subsection we give also the result of learning a dataset IBM29(40) of 40 images, defined as

$IBM29(40) = IBM29 \cup \{+, -, =, /\} \cup \{0,1,...,9\}$

starting with tabula rasa condition.

The result we obtained is: 10 iterations,

$|T^*|_{IBM29(40)/\varnothing} = 204$ and $|C^*|_{IBM29(40)/\varnothing} = 604$.

This is an example of a 1981 machine learning experiment with 40 patterns (Bozinovski 1981).

9 Transfer learning research after 1986

The main focus of this paper is to give a review of the initial work on transfer learning in neural networks which took place between 1972 (Bozinovski 1972) and 1985 (Bozinovski 1985a, 1985b). To the best of our knowledge during that time period, there was no other work on transfer learning in neural networks. That was the period when neural networks were not the main topic in Artificial Intelligence, due to the book of Minsky and Papert (1969) which pointed out some limitations of perceptron type neural networks. Although during 1970's and 1980's there were works on multilayered neural networks (e.g. Fukushima, 1975, 1980), the interest in multilayered neural networks significantly increased after 1986, due to appearance of the book by the Parallel Distributed Processing (PDP) Group (Rumelhart et al. 1986). That book reignited the interest in neural networks, and after some time, the interest in transfer learning in neural networks. Here we will give a short review on the works on transfer learning after 1986.

Early works after 1986 used other terms to describe transfer learning. One such term was "sequential learning", where negative transfer learning was covered with the term "interference" (McCloskey and Cohen, 1989). Other terms used were "adaptive generalization" (Sharkey and Sharkey, 1992), 'learning by learning" (Naik and Mammone, 1993), and "lifelong learning" (Thrun and Mitchell, 1993).

In 1991 the term transfer learning related to neural networks reappeared in literature. That was the work of Pratt. Mostow, and Kamm (1991). That paper introduced a framework of transfer learning, pointing out various types of transfer learning. That framework was also described in the work of Pratt (1993). The framework is shown in Figure 11.

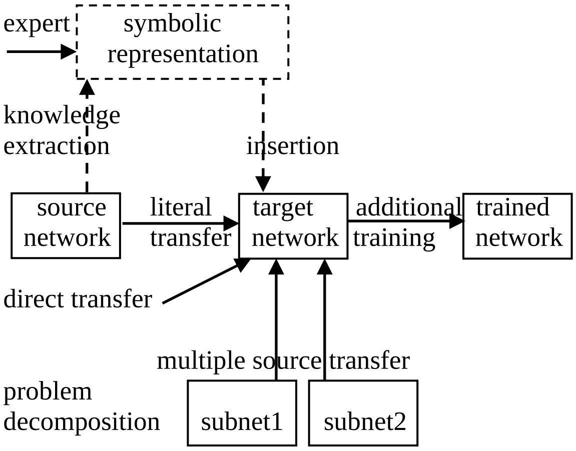


Figure 11. A general framework for transfer learning (adopted from Pratt et al., 1991).

As can be seen from Fig. 11, the general framework for transfer learning proposed in 1991 includes four types of transfer.

One is named literal transfer learning, and it is the transfer learning we used in our work (Bozinovski and Fulgosi, 1976), and is reviewed in this paper.

The second type is a transfer learning which uses direct intervention in the weights of a neural network. We call this direct memory access (DMA) type transfer of knowledge. It is an intervention in a neural network knowledge without a process of incremental learning. The weights change is named weights perturbation. An example of direct weight change described in (Pratt at al. 1991) is w = w+rw, where r is a random number between (-0.6, 0.6). Weights perturbation method was also used in the work (Agarwal et al. 1992)

The third type uses problem decomposition into subproblems, represented by subnetworks, and training the subnetworks for the subproblems, and then insert the subproblem knowledge into the target network.

The fourth type of transfer is indirect transfer, where the weight-based knowledge is extracted, then it is represented as a rule-based knowledge, then it is updated using rule based representation, and then it is inserted in a target neural network as weights-based knowledge.

A review of transfer learning in the early 1990's is given by Pratt and Jennings (1996). A review by Pan and Yang (2010) covers the period after that. Tan et al. (2018) review the deep transfer learning.

10 Discussion and conclusion

The contribution of this paper is a review of an early period of transfer learning research, a period which was not known to the current researchers in transfer learning. In current history part of transfer learning, as covered by Wikipedia >Transfer Leaning >History (2020) there is information which suggests that the beginning of transfer learning research is in 1993. This paper gives information on the transfer learning research during 1970's and early 1980's.

In this discussion let us mention that the original 1976 paper was published in the Proceedings of the symposium Informatica 1976, which took place in Bled, Slovenia, one year before appearance of the first issue of the journal Informatica, in 1977. The paper was published in Croatian, not in English, which is the main reason why the paper was not known for a rather long time.

In the review of the period 1990 - 2000 given in this paper, we can notice that the research during that period was focused on forms that transfer learning can take, and directions it can go. The fundamental concepts like a measure of transfer learning was not covered. The interest of fundamental notions was pointed out again in 2000s (Tan et al, 2018). That relates the research in 1970's to the contemporary research in transfer learning.

Let us mention that the application of transfer learning with real datasets of images described here, such as Computer Terminals dataset containing 3x26 letters and the IBM29(40) containing 40 characters on a matrix 7x5 is an early use of datasets of characters in machine learning. An example of a character dataset used in contemporary research (e.g. Wang et al. 2019) contains 9 characters (digits 0 to 9) on a matrix 28x28, with variety of templates.

In conclusion, this paper extends the knowledge in transfer learning with a relation between the pioneering work (in 1970's and early 1980's) and the current research on transfer learning, giving also a review of the period in early 1990's. Important part of that relation is the reminder of the theoretical 1976 paper, which presented the first mathematical and geometrical modeling, and a measure of transfer learning. The experimental work during 1976-1981 with datasets representing images of characters also relates to the contemporary research in machine learning.

11 Acknowledgement

The author wishes to thank professor Gjorgi Cupona from Mathematical Institute with Numeric Center of the University of Skopje, who in 1968 encouraged the author to prepare a matural thesis from the Glushkov's book. He also wishes to thank professor Ante Santic from Electrical Engineering Department of University of Zagreb, who in 1972 allowed author's work on perceptrons. Professor Ante Fulgosi from Psychology department of University of Zagreb was coauthor of the first published paper on transfer learning in 1976. The author also wishes to thank professors Michael Arbib and Nico Spinelli who enabled him to work on the research related to this paper at the Computer and Information Sciences (COINS) Department of the University of Massachusetts at Amherst for the period 1980-81, and to thank professor Andrew Barto for the period 1995-96. The work at University of Massachusetts in both periods was supported by Fulbright grants. Part of the research reported here was supported by a Sigma Xi grant-in-aid in 1981. The author also wishes to thank the reviewers of this paper for their valuable comments.

12 References

[1] A. Agarwal, R. Mammone, and D. Naik (1992) An on-line training algorithm to overcome catastrophic forgetting. In Intelligence Engineering Systems through Artificial Neural Networks. volume 2, pages 239-244. The American Society of Mechanical Engineers, AS~IE Press.

[2] J. Baxter, R. Caruana, T. Mitchell, L. Pratt, D. Silver, S. Thrun (organizers) Learning to Learn: Knowledge Consolidation and Transfer in Inductive Systems, NIPS*95 Post-conference workshop, Vail, Colorado http://socrates.acadiau.ca/courses/comp/dsilver/NIPS95ltl.nips95.workshop.pdf

[3] S. Bozinovski (1972) Perceptrons: Training in pattern recognition. (original in Croatian: Perceptroni i obucavanje u prepoznavanju oblika) unpublished student scientific competition paper, University of Zagreb

[4] S. Bozinovski (1974). Perceptrons and possibility of simulation of a teaching process (original in Croatian: Perceptroni i mogucnost simuliranja procesa obucavanja), unpublished M.Sc. thesis, Electrical Engineering Department, University of Zagreb

[5] S. Bozinovski, A. Fulgosi (1976). The influence of pattern similarity and transfer of learning upon training of a base perceptron B2. (original in Croatian: Utjecaj slicnosti likova i transfera ucenja na obucavanje baznog perceptrona B2), Proc. Symp. Informatica 3-121-5, Bled.

[6] S. Bozinovski, A. Santic, A. Fulgosi (1977). Normal teaching strategy in pair-association in the case teacher:human-learner:machine. (original in Croatian: Normalna strategija obicavanja u obucanju asocojacije parova u slucaju ucitelj:covjek-ucenik:masina), Proc. Conf. ETAN, 21:IV-341-346, Banja Luka, [available online].

[7] S. Bozinovski (1978). Experiments with non-biological systems teaching. (original in Macedonian: Eksperimenti na obucuvanje na nebioloski sistemi) Proc. Conf ETAN, 22:IV-371-379, Zadar [available online].

[8] S. Bozinovski (1981). Teaching space: A representation concept for adaptive pattern

classification. COINS Technical Report, University of Massachusetts at Amherst, No 81-28 [available online: UM-CS-1981-028.pdf]

[9] S. Bozinovski (1985a). Adaptation and training: A viewpoint. Automatika 26 (3-4) 137-144

[10] S. Bozinovski (1985b). A representation theorem for linear pattern classifier training. IEEE Transactions on Systems, Man, and Cybernetics 15(1): 159-161

[11] S. Bozinovski (1995). Neuro-genetic agents and a structural theory of self-reinforcement learning systems. CMPSCI Technical Report 95-107, University of Massachusetts at Amherst [available online: UM-CS-1995-107.pdf].

[12] K. Fukushima (1975) Cognitron: A self organizing multilayered neural network. Biological Cybernetics 20: 121-136 https://doi.org/10.1007/BF00342633

[13] K. Fukushima (1980) Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics 36: 193-202, doi: 10.1007/BF00344251.

[14] V. Glushkov (1967) Introduction to Cybernetics (original in Serbian: Uvod u Kibernetiku, translated from Russian, published by Zavod za izdavanje udzbenika Srbije)

[15] I. Goodfellow, Y. Bengio, A. Courville (2016) Deep Learning, MIT Press, DOI:10.1007/s10710-017-9314-z

[16] M. McCloskey, N. Cohen (1989) Catastrophic interference in connectionist networks: the sequential learning problem. The Psychology of Learning and Motivation, 24 DOI:10.1016/S0079-7421(08)60536-8

[17] M. Minsky, S. Papert (1969) Perceptrons. The MIT Press, 1969

[18] D. Naik, R. Mammone (1993) Learning by learning in neural networks, In R. Mammone (ed.) Artificial Neural Networks for Speech and Vision, Chapman and Hall, London.

[19] S. Pan, Q. Yang (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345– 1359 DOI:10.1109/TKDE.2009.191

[20] L. Pratt, J. Mostow, C. Kamm (1991). Direct transfer of learned information among neural networks. In Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), p. 584-589, Anaheim, CA.

[21] L. Pratt (1993). Discriminability-based transfer between neural networks. In NIPS Conference: Advances in Neural Information Processing Systems 5 Morgan Kaufmann Publishers. pp. 204-211

[22] L. Pratt, B. Jennings (1996) A survey of transfer between connectionist networks, Connection Science 8(2) 163-184. https://doi.org/10.1080/095400996116866

[23] F. Rosenblatt (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review 65: 386-408. DOI:10.1037/h0042519

[24] F. Rosenblatt (1962). Principles of Neurodynamics. Spartan Books. DOI:10.2307/1419730

[25] D. Rumelhart, J. McClelland, and the PDP Group (1986). Parallel Distributed Processing. MIT Press.

[26] N. Sharkey and A. Sharkey (1992) Adaptive generalisation and the transfer of knowledge, Proceedings of the Second Irish Neural Networks Conference, Belfast.

[27] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu (2018). A Survey on Deep Transfer Learning, arXiv:1808.01974v1 [cs.LG] 6 Aug 2018. DOI:10.1007/978-3-030-01424-7_27

[28] S. Thrun, T. Mitchell (1993) Lifelong robot learning, Technical Report IAI-TR-93-7, Institute for Informatics III, University of Bonn. https://doi.org/10.1016/0921-8890(95)00004-Y

[29] H. Wang, C. Li, X. Zhen, W. Yang, B. Zhang (2019) Gaussian Transfer Convolutional Neural Networks, IEEE Transactions on Emerging Topics in Computational Intelligence 3 (5) 360-368. DOI:10.1109/TETCI.2018.2881225

[30] K. Weiss, T. Khoshgoftaar, D. Wang (2016) A survey of transfer learning. Journal of Big Data 3:9. DOI:10.1186/s40537-016-0043-6

[31] Wikipedia > Transfer Learning (October 2020) https://en.wikipedia.org/wiki/Transfer_learning