Neural Networks 97 (2018) 152–161

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Margined winner-take-all: New learning rule for pattern recognition

Kunihiko Fukushima *

Fuzzy Logic Systems Institute, Iizuka, Fukuoka 820–0067, Japan

ARTICLE INFO

Article history:
Received 10 March 2017
Received in revised form 10 July 2017
Accepted 12 October 2017
Available online 7 November 2017

Keywords:
Pattern recognition
Deep CNN
Neocognitron
Learning rule
Margined WTA
Interpolating-vector

ABSTRACT

The neocognitron is a deep (multi-layered) convolutional neural network that can be trained to recognize visual patterns robustly. In the intermediate layers of the neocognitron, local features are extracted from input patterns. In the deepest layer, based on the features extracted in the intermediate layers, input patterns are classified into classes. A method called IntVec (interpolating-vector) is used for this purpose.

This paper proposes a new learning rule called *margined Winner-Take-All* (*mWTA*) for training the deepest layer. Every time when a training pattern is presented during the learning, if the result of recognition by WTA (Winner-Take-All) is an error, a new cell is generated in the deepest layer. Here we put a certain amount of margin to the WTA. In other words, only during the learning, a certain amount of handicap is given to cells of classes other than that of the training vector, and the winner is chosen under this handicap. By introducing the margin to the WTA, we can generate a compact set of cells, with which a high recognition rate can be obtained with a small computational cost. The ability of this mWTA is demonstrated by computer simulation.

1. Introduction

Deep convolutional neural networks (deep CNN) exhibit a large power for robust recognition of visual patterns. The *neocognitron* is a network classified to this category. It is a multilayered network, and has an architecture with weight-sharing convolutional neural layers (Fukushima, 1980, 2013a). It can be trained to recognize visual patterns robustly.

The neocognitron, however, uses a learning rule different from the conventional deep learning (Bengio, Courville, & Vincent, 2013; Bengio, Lamblin, Popovici, & Larochelle, 2007; Hinton, Osindero, & LarochelleTeh, 2006; Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012; Lee, Grosse, Ranganath, & Ng, 2009; Ranzato, Huang, Boureau, & LeCun, 2007; Schmidhuber, 2015; Vincent, Larochelle, Lajoie, Bengio, & Manzagol, 2010).

For training intermediate layers, we use an unsupervised learning rule, called *AiS* (*Add-if-Silent*) (Fukushima, 2013b, 2014a; Fukushima & Shouno, 2015). Under the AiS rule, a new cell is generated if all postsynaptic cells are silent in spite of non-silent presynaptic cells. The generated cell learns the activity of the presynaptic cells in one-shot, by adjusting its input connections to be proportional to the activity of the presynaptic cells. Thus, by the learning, each cell in an intermediate layer comes to work as a feature-extracting cell. The input connections of a cell can be represented by a vector, which we call the reference vector of the cell. We can interpret that the reference vector shows the cell's preferred feature, to which the cell responds the most strongly.

Once a cell is generated, its reference vector does not change any more. Thus the learning is a process of simply choosing reference vectors from the set of training vectors. It then does not require time-consuming repetitive calculation, because there is no need of tuning the reference vectors. With the AiS rule, reference vectors of generated cells in a layer can distribute uniformly in the multi-dimensional feature space.

During the recognition phase, the neocognitron classifies the input pattern based on features extracted by the intermediate layers. In the neocognitron discussed in this paper, the method of *IntVec* (*Interpolating-Vector*) is used for classification (Fukushima, 2007, 2014a). It is known that the recognition error can be greatly reduced by the use of the IntVec than by the WTA (winner-take-all) or by the SVM (support vector machine). During recognition, we measure similarities between the test vector and planes (or lines) that are spanned by every trio (or pair) of reference vectors of the same class, and choose the nearest plane (or line). The class name of the chosen plane (or line) is taken as the result of classification.

This paper proposes a new learning rule suited for training the deepest layer. One of the important roles of learning is to produce a compact set of reference vectors (namely, cells of the deepest layer) that can accurately represent the large set of training vectors. This is important especially when the IntVec, which searches the nearest plane, is used for recognition. If the number of reference vectors can be made small, we can easily search the nearest plane without spending a heavy computational cost.

* Correspondence to: 634-3, Miwa, Machida, Tokyo 195–0054, Japan.
E-mail address: fukushima@m.ieice.org.
URL: http://personalpage.flsi.or.jp/fukushima/index-e.html.

https://doi.org/10.1016/j.neunet.2017.10.005
0893-6080/

Incidentally, some ideas similar to the IntVec has previously been reported elsewhere (Chien & Wu, 2002; Li & Lu, 1999). However, they try to use all training vectors directly as reference vectors. Hence a huge computational cost is required for recognition, because of a large number of planes or lines, which are made of all possible combinations (trios or pairs) of training vectors. For using the IntVec for practical applications, it should be combined with a method of generating a compact set of reference vectors.

In the deepest layer, each feature-extracting cell (namely, its reference vector) has a label indicating the class name of the training pattern by which the cell is generated. Different from intermediate layers, uniform distribution of reference vectors in the feature space is not necessarily useful. To get a high recognition rate with a small computational cost, it is desired that the labeled reference vectors distribute more densely near class borders than near the center of a cluster. To satisfy this condition, this paper proposes the use of a new learning rule, which is named *margined Winner-Take-All* (*mWTA*).

The basic idea of the mWTA was proposed previously by the author (Fukushima, 2016). This paper proposes an improved version of the mWTA, in which a new rule for determining the margin is adopted. The condition for tuning the connections has also been modified. By the use of the improved rule, we can acquire a smaller error rate with a smaller computational cost.

Although this paper shows that the mWTA and IntVec are powerful methods for the neocognitron, the use of these methods is not limited to the neocognitron. They can be introduced effectively into any other neural networks for pattern recognition.

In this paper, we focus our discussion on the new learning method, mWTA, for the deepest layer. The outline of the neocognitron, including the method of learning for intermediate layers, is explained in Appendix A. Section 2 discusses the method of IntVec, which is used in the deepest layer for classifying patterns. Section 3 gives a detailed discussion on the mWTA. The results of computer simulation of the proposed method, mWTA, appear in Section 4.

2. Interpolating-vector

2.1. S-cell in the deepest layer

Test patterns presented to the input layer, U_0, are classified in the deepest layer, U_{SL}, based on the features extracted by the intermediate layers (Fig. 1). The input signals to an S-cell of U_{SL} is the response of C-cells of U_{CL-1}, which is represented by $\boldsymbol{x}$. In the deepest layer, threshold θ of S-cells is set to $\theta = 0$. Hence, from (A.2), the response of an S-cell is given by

$$u = sv \tag{1}$$

where $s = (\boldsymbol{X}, \boldsymbol{x})/\{\|\boldsymbol{X}\| \cdot \|\boldsymbol{x}\|\}$ is the similarity between $\boldsymbol{x}$ and reference vector $\boldsymbol{X}$, and $v = \|\boldsymbol{x}\|$ is the response of the V-cell.

For the economy of computational cost, analysis of the response of S-cells of U_{SL} is actually performed, not at all retinotopic locations, but only at the retinotopic location where the response of the V-cell is the largest. This means that the value of $v = \|\boldsymbol{x}\|$ is the same for all S-cells to be analyzed.

Training S-cells of U_{SL} is done by a supervised learning. The reference vectors of S-cells are created in such a way that a large number of training vectors of each class can be represented by a small number of reference vectors. Each reference vector is made of a weighted sum of training vectors of the same class and has a label of the class name. The concrete method of learning is discussed later in Section 3.

We use the method of *Interpolating-Vector* (*IntVec*) for U_{SL} to classify input patterns based on the response of S-cells of U_{SL}. Here we first discuss the method of recognition by the IntVec, before discussing the detailed method of learning.

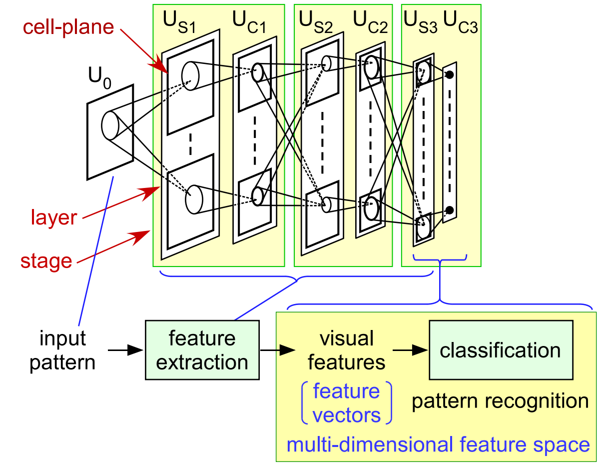

Fig. 1. The architecture of the neocognitron (Fukushima, 2014b; Fukushima & Shouno, 2015). See Appendix A for detailed explanation.

2.2. Recognition by interpolating-vector

After having finished learning, by which all reference vectors have been produced, we classify input patterns. Here, each input pattern is the response of U_{CL-1} at the retinotopic location where the response of the V-cell is the largest. Each pattern that is to be classified is called a test vector in the feature space, and is represented by $\boldsymbol{x}$.

One of the simplest methods for classifying a test vector is the *winner-take-all* (*WTA*). In the WTA, the label of the reference vector that has the largest similarity to the test vector (namely, the largest-output S-cell) shows the result of recognition. The WTA is used in many neural networks, including the original neocognitron.

It has been shown, however, that the recognition error can be largely reduced by the *IntVec* (*Interpolating-Vector*). There are several versions of the IntVec, for example, Int-2 (Fukushima, 2007, 2013a, b) and Int-3 (Fukushima & Shouno, 2015).

2.2.1. Int-2

Although we actually use Int-3 in the neocognitron discussed in this paper, we start explaining the principle of the IntVec, taking Int-2 as an example, because it is simpler than Int-3. In the multidimensional feature space, we assume lines connecting every pair of reference vectors of the same label (Fig. 2). Every line is assigned the same label as the reference vectors that span the line. We then measure distances (based on similarity) to these lines from test vector $\boldsymbol{x}$. The label of the nearest line (namely, the line that has the largest similarity to $\boldsymbol{x}$), instead of the nearest reference vector, shows the result of pattern recognition. As can be seen by the computer simulation shown in Section 4, we can classify test vectors correctly, even with a smaller number of reference vectors than by the WTA.

The mathematical process of searching the nearest line by the Int-2 can be expressed as follows (Fukushima, 2007). Let $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ be two reference vectors of the same label (Fig. 3(a)). Let $\boldsymbol{\xi}$ be a vector that is given by a linear combination of this pair of vectors. It represents a point on the line and is named an *interpolating vector*.

$$\boldsymbol{\xi} = p_i \frac{\boldsymbol{X}_i}{\|\boldsymbol{X}_i\|} + p_j \frac{\boldsymbol{X}_j}{\|\boldsymbol{X}_j\|}, \qquad (p_i + p_j = 1). \tag{2}$$

Under possible combinations of p_i and p_j, the similarity between $\boldsymbol{\xi}$ and test vector $\boldsymbol{x}$ takes a maximum value

$$s_{\text{line}} = \sqrt{(s_i^2 - 2s_i s_j s_{ij} + s_j^2)/(1 - s_{ij}^2)} \tag{3}$$