Neocognitron For Rotated Pattern Recognition

Michael Tran, Siddheswar Ray, and Ronald Pose

School of Computer Science & Software Engineering

Monash University, Clayton Victoria, 3168, Australia.

vtran@cs.monash.edu.au, sid@ cs.monash.edu.au, rdp@cs.monash.edu.au

Abstract

Ideally computer pattern recognition systems should be insensitive to scaling, translation, distortion and rotation. Many neural network models have been proposed to address this purpose. The Neocognitron is a multi-layered neural network model for pattern recognition introduced by Fukushima in the early 1980s. It was considered effective and, after supervised learning, it can recognise input patterns without being affected by distortion, change in size, or shift in position but it was not designed to handle rotated patterns. This paper examines the layers of the Neocognitron to emphasize their importance in Fukushima's theory. We also provide an additional filer layer to deal with rotated patterns.

Key Words: Pattern recognition, character recognition, Neocognitron, neural network.

1. Introduction

In his studies of handwritten recognition, Fukushima proposed a multi-layered neural network model, the Neocognitron [3]. This is considered one of the most effective models. After supervised learning it can recognise input patterns without being affected by distortion, change in size, or shift in position. There have been a few attempts to augment Neocognitron to recognise rotated patterns. These have not been entirely successful, not only because of the complexity of the problem but also because of the complexity of the structure itself. Furthermore, these attempts have neglected to study the filters that are used to conduct the proper training of the Neocognitron. The choice of filter sets for the Neocognitron's filtering layers has a big impact on performance.

1.1 Review of the Neocognitron

The Neocognitron was initially designed to process both handwritten numeric and alphabetic characters [3,4,5]. In Fukushima's later publications [5,6,7], numeric characters are used to illustrate the process of learning in both supervised training and self-organised networks. A set of 40 patterns for each number from 0 to 9, and a large number of filters were used to construct Fukushima's experiment [10,12].

Fukushima's papers considered handwritten characters as sets of lines, loops, corners and curves, set down in particular spatial relationships. The Neocognitron is composed of a number of cascaded connected modular structures in pairs, US and UC, where S stands for Simple, and C stands for Complex [3,4]. These are preceded by an input layer UC0. This is depicted in Figure 1. US1 denotes layer of S-cells and UC1 denotes a layer of C-cells. Each layer comprises a number of cell-planes. Each cell-plane contains many S-cells or C-cells, and is represented by a rectangle.

The complexity of features is extracted from a number of cascaded US/UC layers. The UC4 layer at the highest module is a recognition layer, representing the result of pattern recognition. Each layer in the Neocognitron consists of a number of S-planes, which detects the location of particular features, and a number of C-planes, which amplifies the representation by blurring these features [7,9,10,11,12].

During each iteration of training, only the seed cell has its input weights reinforced and it is selected in accordance with the learning paradigm used in the Neocognitron. An S-cell must respond more strongly to the input feature within its receptive field than any of its immediate neighbours. This competitive approach to seed cell selection is applied across all S-planes in the layer that is being trained. Once a seed cell has been located within a certain S-plane and the cells within that plane have had their weights updated; the seed cell will no longer respond to any new input features [4,5,12].

It is conspicuous that other works have concentrated on the sophistication of formulae and theory, rather than an analysis of the fundamental issue of filter selection. Steuer [11] emphasises the parallel structure and new method of training. The arguments of Barnard are only for the possibility of using the Neocognitron to recognise rotated objects [1]. Banarse and Duller [2] provide their Paradise network model with rich pattern classes input for experiment. Lovell [9] focuses on algorithms and statistical formulations. He has carried out extensive testing of the Neocognitron and provided a large set of test data. However, Lovell's work only relates to the performance statistics rather than the architecture of the Neocognitron. In the literature there exists no detailed study of the use of Fukushima's filters. We believe that a review of the filters used in Fukushima's Neocognitron is crucial for the practical application of his theory in any further studies.

2. Filter Attributes

Fukushima's filters come in four sets. Each applies to an appropriate plane. They are indexed by four files NC.US1, NC.US2, NC.US3, and NC.US4. The number of entries in each are NC.US1 (12 entries), NC.US2 (123 entries), NC.US3 (73 entries), NC.US4 (25 entries) [10]. Each entry is a text file, in which the first line indicates dimensions of a matrix (two-dimensional array) and the rest is the filter matrix. One entry from NC.US1 is listed below in Figure 2. The same file structure is applied to all filters in four layers.

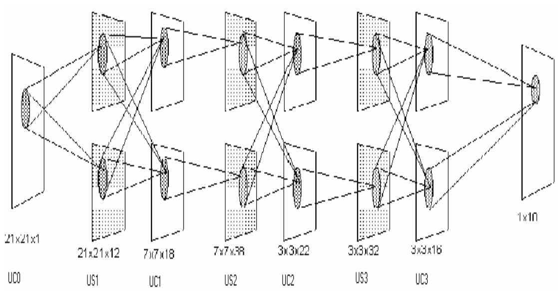

Figure 1. Layout of Neocognitron

3 3
0 0 1
0 1 0
1 0 0

Figure 2. Content of one filter in layer 1

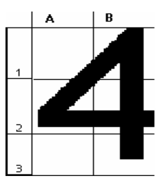

Figure 3. A numeric character 4 in 6-square block

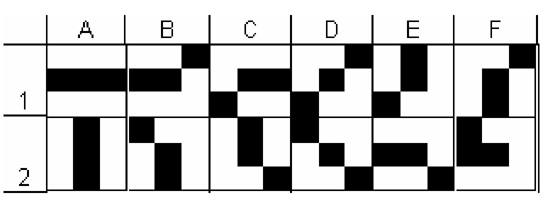

Figure 4. Filters layer 1

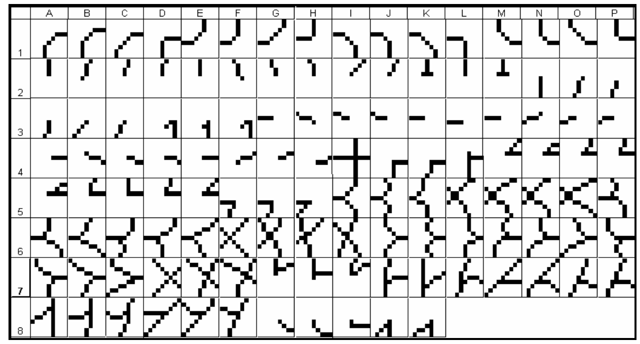

Figure 5. Filters in layer 2

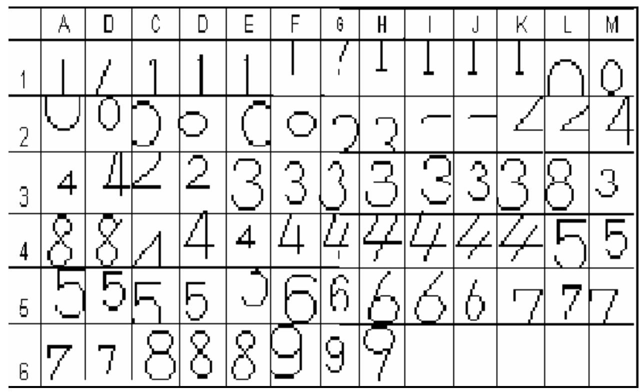

Figure 6. Filters in layer 3

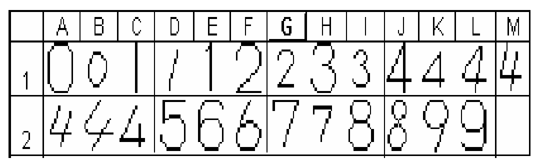

Figure 7. Filters in layer 4

3. USs and UCs

3.1 US layers (Feature extraction)

Layer 1: As seen in Figure 4, the filter set in this layer covers all possible features of any character, such as vertical and horizontal lines, curves and crosses. A pattern input through this layer will have the outcome of 12 cells. Each cell extracts only the feature that it was designed to detect.

Any noise will be removed, as it will not be matched in any of the patterns of these filters. The filter set of this layer is mainly for *noise reduction*.

Layer 2: The filter set for this layer covers all possible features of numeric characters that can be arranged in 6-square blocks as in Figure 3. This reduces the possibility of similar patterns occurring in several digits such as, for example, the top left of 8 is similar to the top left of zero.

However, by eliminating those in unexpected positions, this layer determines the shapes and locations of the features detected in the corresponding visual fields. For example, the pattern of numeric character 4 as in cell A2 of Figure 3 cannot occur in any of the patterns of characters 0, 1, 2, 3 and 5.

This layer is, in fact, the core of the supervised training process of the Neocognitron. Practically, we know that the cross in cell B2 of Figure 3 is the central feature for the numeric character 4. We can tell from the filter set 2 in Figure 5 that the cell I4 (Figure 5) contains a copy of cell B2 (Figure 3), which is one of the specific features used to identify this number.

Similarly, the cells F6, G6, H6, I6, K6 are for the number 8; the cells K7, L7, M7 are for the number 6; and so on. Without these identified features that are unique to the individual number, the characteristic calculation will be biased and consequently produce meaningless results. The filter set of this layer is for *feature extraction*.

Layer 3: Figure 6 contains patterns of actual characters in shifted positions. For example, this filter set has several patterns of character 0 in cells (A2..F2) that were in various shifted (up/down/left/right) positions; several of 1, 2 shifted up/down (C1..K1, C3..D3); several of 3 shifted left/right (C3..K3)... It also has these numbers scaled in size variation and in shape variation.

Fukushima has stated in his papers [3,7] that the corresponding UC3 will act as a compression layer. However, some doubts have been raised regarding to the efficiency of the compression feature of UC3 [2,12]. The filter set of this layer is for *distortion adjustment*.

Layer 4: Figure 7 contains patterns of actual characters in different shapes. These patterns are close to the patterns used for training inputs. The filter set of this layer is for *feature recognition*.

3.2 UC layers (Feature amplification)

Fukushima's theory [3] has clearly defined the role of C-cells as important in handling stimulus input to allow for positional errors of the features extracted by the preceding S-cells. The functionality of each layer is listed below:

Layer 1: The functionality of this layer is to blur all features before they are merged into new pattern. The result from this layer is a reconstruction of the whole pattern even if some features were missing.

Layer 2 This layer blurs all features before they are merged to reconstruct missing features.

Layer 3: This layer blurs all features to connect them to the whole image of the pattern. Many empty cells from the preceding process can be ignored giving fewer features in a more recognizable layer.

Layer 4: This final layer merges all features to construct the final desired figure for trained pattern.

3.3 Discussion

Based on cascaded layer architectures with feed-forward input units, the Neocognitron may be considered efficient for handwritten digit recognition. The main concern of this paper is that it may not be applicable for inputs with high degree of geometric transformation such as rotated objects. From the present study, the four US layers summarized as (1) noise removal, (2) feature extraction, (3) distortion adjustment, and (4) feature recognition, that link together as a chain process so that, when a supervised pattern input passes through these four filters, it can be recorded as a known character. However, when a rotated digit is input into this structure, it will not go any further than layer 2, as its rotated patterns are not in the set.

3.4 Extra layer for rotated patterns

An additional layer called ***Rotation adjustment*** was inserted before layer 2. This layer contains a set of non-duplicated patterns derived from rotated characters in $\boldsymbol{n}$ degrees within the range of 0 to 360 degrees, with the exception of 180 degrees to prevent the confusion of case of 6 and 9. This is shown in Figure 8. To simplify the process, we choose $\boldsymbol{n} = 15$ and group these patterns in twenty-four blocks. Each holds a maximum of 6 patterns that form a digit, as in Figure 3, and with memory of its different degrees of rotation. Any input pattern passed through these blocks will be rotated back to 0 degrees. Following the normal blurring and merging, the restored digit from its rotated form will be ready to go for *feature extraction* as in Figure 9.

Figure 8. Character 4 rotated by 15 degrees from 0 to 90

This layer has not caused any deficiency in the recognition of non-rotated patterns since it will become transparent with the zero degree blocks and pertinent filter patterns.

Compared to the previous US2 which is shifted right to become US3 and had 123 entries, the new US2 (called US2r for rotation) has many more entries generated by the following algorithms. There are 40 samples for each of 10 input digit characters. Each sample has (360 div $\boldsymbol{n}$ rotated degrees = 360 / 15 = 24) rotated angles patterns and 6 blocks (sub-patterns) for each. Note that $\boldsymbol{n}$ was chosen as 15 for simplicity. In many trials, we found that the best accuracy is obtained within the range of 8 to 12 degrees. This leaves us with a slight dilemma since the best result based on size of the structure is with the 15-degree step. With n = 15 we have initially 57600 sub-patterns used to construct US2r in addition to the existing 4 layers of the original Neocognitron. Since preparing so many rotated patterns is difficult we have two strategies; one is making all the rotated patterns, the other is to use a rotation translation matrix as in [2].

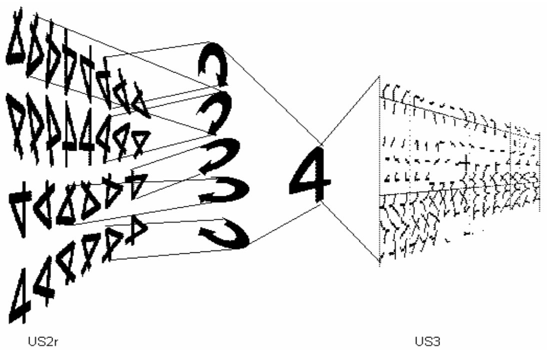

Figure 9. New US layer for rotation (US2r)

3.5 Building the new filter set

The following two-pass process is applied to pick important common cells for each set of 40 patterns and store them in a repository. Then cells with similarities greater 90 percent are removed.

1) Pick important cells of 40 patterns in each of 10 character sets

for i := 0 to 9 do // for 10 numeric char
for f := 0 to 39 do // 40 patterns for each char
{ pick_pattern = f // pick the pattern
for g := 1 to 24 do // 24 angles each 15 degrees different in rotation
for row = 1 to 3, for col = 1 to 2 do // (2x3)
{for n = f + 1 to 39 do // take next pattern
{for m = g + 1 to 24 do // take next pattern
{ next_cell = n // pick a seed cell.
if next_cell[row,col] is not blank and similar_of (pick_cell[row,col], next_cell[row,col]) > 90% then mask blank next_cell[row,col] } save pick_cell into repository. }}

2) Remove all similar cells (>90%)

For i = 1 to number of cells in repository,
for f = i + 1 to number of cells in repository do
if similar_of (cell[i], cell[f]) > 90% then
{ delete cell[f]; inc(f) }

4. Conclusion

In this paper we reviewed the capacities of Fukushima's Neocognitron. We found that the Neocognitron is highly dependent on the selection of the relevant filter sets. Failure to select good filters can lead to biased results. We also applied this understanding to provide an additional layer in the structure to handle rotated digits with various angular rotations as confirmation of our study.

In conclusion, we emphasise that proper selection of filters is the key to the success of the Neocognitron. This selection is as important as the theory itself.

5. References

[1] E. Barnard and D. Casasent, "Shift invariance and the Neocognitron," Neural Networks, vol. 3, pp. 403-410, 1990.

[2] D.S. Banarse and A.W.G. Duller, "Application of neural deformable templates to hand written digit classification," Neural Processing Letters, vol. 5, pp. 111-119, 1997.

[3] K. Fukushima, "Analysis of the Process of Visual Pattern Recognition by the Neocognitron," Neural Networks, vol. 2, pp. 410-412, 1989.

[4] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," Biological Cybernetics, vol. 36, pp. 193-202, 1980.

[5] K. Fukushima and S. Miyake, "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shift in position," Pattern Recognition, vol. 15, pp. 455-469, 1982.

[6] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: a neural network model for a mechanism of visual pattern recognition," IEEE Transactions on Systems, Man, and Cybernetics, pp. 826-834, 1983.

[7] K. Fukushima and N. Wake, "Handwritten alphanumeric character recognition by the neocognitron," IEEE Transactions on Neural Networks, vol. 2, pp. 355-365, 1991.

[8] G. Wallis, "Learning invariant responses to the natural transformations of objects," Proc. International Joint Conference on Neural Networks, vol. 2, pp. 1087-100, 1993.

[9] D. Lovell, "The Neocognitron as a System for Hand-written Character Recognition: Limitations and Improvements," Doctoral thesis, Department of Electrical Engineering, University of Queensland, July 1994.

[10] F. Schnorrenberg, "Fukushima's Neocognitron: An Implementation," Tech Report, Department of Computer Science, Texas A&M University, 1992.

[11] M. Steuer, "Parallel implementation of the Neocognitron," Proc. 1995 European Simulation Multiconference, pp. 826-830, 1995.

[12] S.D. Yeung, H. Y. Chan, and K.F. Cheung, "Incorporating production rules with spatial information onto a neocognitron neural network," International Journal of Neural Systems, vol. 5, pp. 131-142, 1994.