Proceedings of the 8th International Conference on Artificial Neural Networks (ICANN), Skövde, Sweden, pp 567-572, 1998.

Activity Driven Update in the Neural Abstraction Pyramid

Sven Behnke and Raúl Rojas
Institute of Computer Science, Free University of Berlin
14195 Berlin, Germany
{ behnke | rojas }@inf.fu-berlin.de

Abstract

The *Neural Abstraction Pyramid* is a hierarchical neural architecture for image interpretation based on image pyramids and cellular neural networks and inspired by the principles of information processing found in the visual cortex. In this paper we extend the model by describing a parallel mechanism of bottom-up attention control, the *Activity Driven Update* of processing elements. We apply this mechanism to the binarization of handwritten ZIP-codes in a real-world application. The experimental results indicate that updating only a fraction of the processing elements is sufficient for good binarization. Both speed and performance of the application were improved with the new method.

1 Introduction

The human brain manages to focus its limited resources on the relevant visual stimuli of complex scenes. This is done by a mechanism called *attention control* that is driven by salient visual stimuli and the interpretation goal. Psychophysical evidence [3] suggests that attention comprises two components: a bottom-up, fast, primitive mechanism that selects stimuli based on their saliency and a second, slower, top-down mechanism, the spotlight of attention, that is under cognitive control. Both processes compete to select visual stimuli for detailed investigation. As a result the focus of attention is moved from one location to the next either using the covert spotlight or by overt saccadic eye movements.

In this paper we focus on bottom-up attentional processes that work on a finer time scale, namely in the first few milliseconds of a fixation. In [5] it has been shown that latencies can improve image segmentation. The idea is to delay a stimulus based on its relative value. We propose a method named *Activity Driven Update* of processing nodes in the *Neural Abstraction Pyramid*. The update sequence depends on the saliency of a stimulus. This leads to short delays that can be used to improve image interpretation. Salient parts are interpreted first and provide via horizontal and vertical feedback links a larger context for the interpretation of the more ambiguous image parts.

The paper is organized as follows: In the next section we give a brief summary of the Neural Abstraction Pyramid [2] architecture and algorithms. The proposed Activity Driven Update is presented in section 3. Section 4 describes a first application, the binarization of handwriting. The paper concludes with a discussion of the experimental results and gives an outlook of future work.

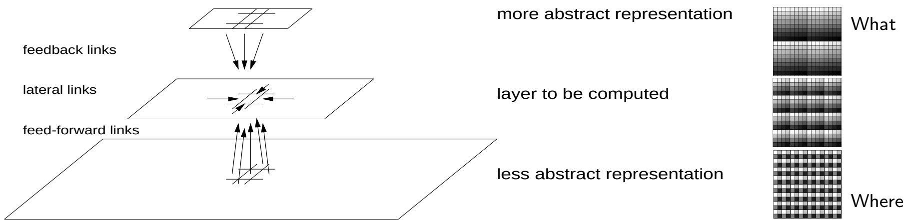

Figure 1: Sketch of the Neural Abstraction Pyramid.

2 Neural Abstraction Pyramid

The Neural Abstraction Pyramid [2] is a hierarchical neural architecture for image interpretation that is based on the ideas of image pyramids and cellular neural networks. It is inspired by the principles of information processing found in the visual cortex. Algorithms for this architecture are defined in terms of local interactions of processing elements that utilize horizontal as well as vertical feedback loops. The goal is to transform a given image into a sequence of more and more abstract representations while the level of detail decreases.
The main features of the Neural Abstraction Pyramid architecture are:

- *Pyramidal shape:* Layers of neural processing elements (*nodes*) are arranged vertically to form a pyramid (see Fig. 1).
- *Analog representation:* The nodes of each layer describe the image in a two dimensional representation. The level of abstraction of these representations increases with height, while the level of detail (spatial resolution) decreases. The bottom layer stores the given image (a signal). Subsymbolic representations are present in intermediate layers, while the highest layers contain almost symbolic descriptions. The representation consists of some *quantities* that can have *values* from a finite interval.
- *Local interaction:* Each node is connected to some nodes from its neighborhood via directed *weighted links*. The shared weights of all nodes in a layer are described by a common *template*. The types of links are:
  - *Feed-forward links:* perform feature extraction,
  - *Lateral links:* facilitate consistent image interpretation,
  - *Feedback links:* propagate interpretation hypotheses downwards.
- *Discrete time computation:* The update of a node's values for time step t depends only on the input values at $(t-1)$. The update can be done:
  - *Layer by layer:* All nodes of a layer are updated simultaneously. The layers are processed in a predetermined sequence, e.g. bottom-up.
  - *Priority driven:* The update sequence depends on a priority that can be defined e.g. in terms of the presence of reliable inputs.
- *Multiscale representation:* Quantities can be stored as image pyramid.

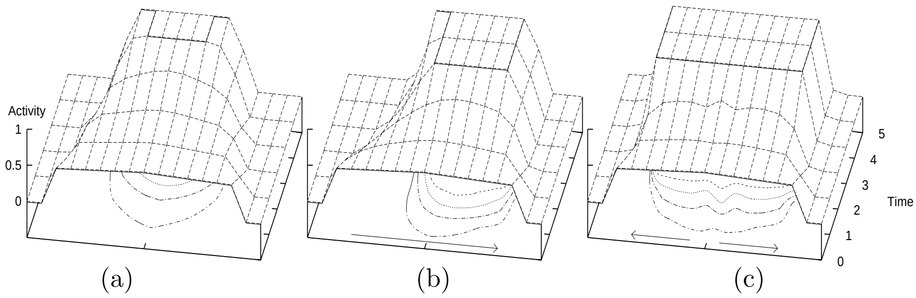

Figure 2: Update methods: (a) buffered, (b) unbuffered, (c) activity driven.

The described architecture has been designed to facilitate the development of image interpretation algorithms that utilize both horizontal and vertical feedback loops. They have to be implemented in a way that honors the principles of *Gestalt psychology*, e.g. proximity, continuity, closure, and simplicity. To make this possible, it is necessary to specify for each layer simple consistent representations that model the objects potentially present in the image. The link weights and update rules have to be selected such that they favor simple and consistent representations instead of complicated or inconsistent ones.

Image interpretation works *iteratively*. First the given image is fed into the bottom layer. In the course of computation the interpretations spread upwards via feed-forward links at locations where little ambiguities exist. These partial results provide via lateral and feedback links a larger context for the interpretation of the more ambiguous stimuli. The quality of the interpretation increases and after a few iterations the interpretation is stable.

3 Activity Driven Update

The interpretation performance of the Neural Abstraction Pyramid depends on the update sequence of the nodes as Fig. 2 illustrates using a simple example. It is shown how a one dimensional stimulus develops in three different update modes. We used a plateau stimulus (shown in the front) that increases slightly from the edges (0.5) towards the middle (0.56). The successive cuts show how the stimulus develops over time under a dynamic that is described by: $q_x^{t+1} = \mathsf{max}(0, \mathsf{min}[1, q_x + (q_{x-1}^{t(+1)} + q_{x+1}^{t(+1)})/2 - 0.5])$. Neighboring nodes have excitatory links and the activity is mapped to $[0, 1]$ using a negative bias and saturation. The update modes differ in the handling of the neighboring activities.

(a) *Buffered update* is conservative. All nodes have to be computed in time step t until the resulting activities can be used in step $(t + 1)$. This makes the result independent of the update sequence within a time step. All nodes can be computed in parallel since no dependencies exist. On the other hand, the interpretation is slow, because information travels horizontally with a speed of only one node per time step.

(b) *Unbuffered update* computes the nodes in a predetermined order, here from left to right. The resulting activity q_{x-1}^{t+1} of a node is used immediately to compute the activity q_x^{t+1} of its right neighbor. The interpretation converges much faster, since information travels the full length from left to right within the same time step. However, the information flow from right to left is still slow which results in an undesired asymmetric response of the system.

(c) *Activity Driven Update* uses the same unbuffered strategy to speed up convergence. It prevents undesired asymmetric responses by making the update sequence dependent on the input. The nodes are computed in the sequence of their activity with the most active node computed first. Fast communication occurs now from the more active to the less active image parts. Since the activities represent confidence of interpretation, the image parts that are easy to interpret are computed first which in turn bias and speed up the interpretation of the more ambiguous image parts. If multiple interpretations compete, the one that first receives support from the context is likely to win. Activity Driven Update also speeds up computation, because the nodes with activity zero will never get active and so they don't need to be updated. In most applications the vast majority of nodes will be inactive.

Ordered update does not require global communication. If integrate-and-fire nodes are used, those that receive a stimulus that fits their receptive field will fire earlier than nodes that get a suboptimal stimulus. The firing nodes trigger their neighbors via excitatory links, if they are close enough to the firing threshold. This leads to an avalanche effect that causes a fast traveling activity wave. The wave stops if it crashes with a wave from the opposite direction or enters locations that have an activity that is too low to be triggered. If the nodes need about the same refractory time, all nodes will become activated synchronously again. This synchronization could be a basis for feature binding.

4 Binarization of Handwriting

We illustrate Activity Driven Update in the Neural Abstraction Pyramid architecture using a real-world application. The task is to separate handwriting in the foreground from the background in gray level images that have been provided by Siemens AG and show scanned ZIP-codes from German letters. This task is nontrivial, since the envelopes are mostly made of dark paper.

A histogram based thresholding technique, similar to [4] has been used for binarization in the original ZIP-code recognition system. Its limitations become visible when a gray level gradient, broken lines or noise are present in the image.

A more powerful binarization method that is based on the Neural Abstraction Pyramid architecture was proposed in [2] (see there for more details). The idea is to detect the lines and to assign their corresponding pixels to the foreground. The fact that lines in handwriting usually exhibit good continuity is exploited. Three levels of abstraction (see Fig. 3) are used in the pyramid:

- Level 0: gray values (in), foreground (out) / background separation
- Level 1: edges in four orientations (–), (|), (/), and (\)
- Level 2: lines in four orientations

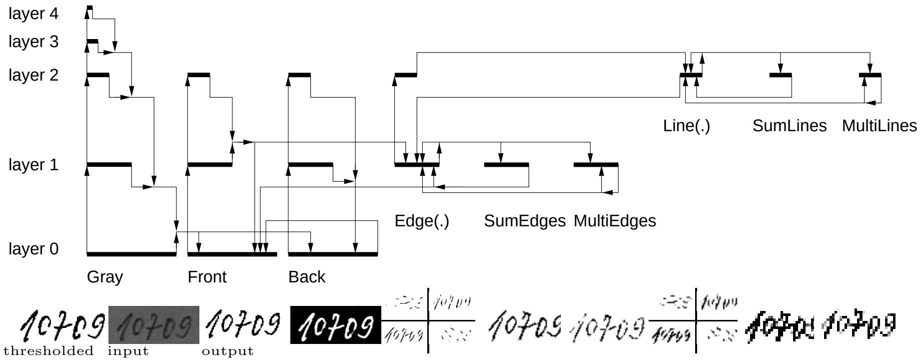

Figure 3: Sketch of the binarization application with quantities and links.

The values of the quantities are interpreted as confidences about the presence of a certain feature at the corresponding image position. The values are computed as saturated weighted sum of the input values. The interaction of the quantities is described by the following templates (see Fig. 3):

- *Gray value:* The original image is stored as a pyramid in quantity Gray.
- *Foreground and background:* Front computes the image details that are darker than their surround while Back contains the brighter background. Both templates inspect the immediate neighborhood of a node more closely and have links that facilitate a region-growing process by excitatory feedback from upper levels. In addition, Front receives inhibitory input from Back and excitatory input from SumEdges to strengthen responses that are supported by detected edges and to remove the noise.
- *Edges:* The distributed edge representation uses four quantities Edge(.) of different orientations. In addition, a quantity SumEdges represents the sum of all and MultiEdges signals the existence of more than one orientation at the same location. The Edge(.)-template functions are:
  - *Edge detection:* two levels of the Front values are inspected,
  - *Cooperation with edges of similar orientation:* via excitatory links from edge elements that would form a good continuation,
  - *Cooperation with lines of the same orientation:* via an excitatory link from the corresponding line element,
  - *Competition with edges at the same position:* via inhibitory input from MultiEdges and SumEdges.
- *Lines:* The line representation is done similarly. The edge representation is inspected to detect lines. Line elements cooperate and compete as well.

In [2] the update of the nodes was done layer by layer with buffered inputs. There line completion and noise removal could be demonstrated. It was investigated, how the performance of a ZIP-code recognition system [1] that is used in a large scale application can be improved by the pyramid binarization.

5 Results and Discussion

A set of 503 hard images that were rejected by the original recognition system was selected from 4134 handwritten five-digit ZIP-codes. These images were binarized using the Neural Abstraction Pyramid, again presented to the recognition system, and the results of both runs were combined. The original system had an acceptance rate of 84.56% with 1.17% of the accepted ZIP-codes being substitutions. Using the described modification 169 images were accepted additionally, while only one of these was substituted. Thus, the overall acceptance rate improved to 88.65% without decreasing the systems reliability.

In the new experiments we augmented the binarization pyramid by the proposed Activity Driven Update mechanism in the $\mathsf{Edge}(.)$ and in the $\mathsf{Line}(.)$ quantities. This improved both speed and performance of the system. Now 209 of the 503 selected hard images were accepted and none of these was substituted. This improved the overall acceptance rate to 89.62% and lowered the substitution rate to 1.11% of the accepted digits. The average running time for the entire recognition was 0.8 seconds per ZIP-code on a Pentium-II-266.

In this paper the extension of the Neural Abstraction Pyramid by an Activity Driven Update mechanism has been proposed. This mechanism facilitates bottom-up attention control. It accelerates convergence by recomputing only a small fraction of the nodes. Since the update sequence depends on the image content, an improved interpretation performance could be achieved. The Activity Driven Update was tested in a real-world application, the binarization of handwritten ZIP-codes. The new technique improved both speed and performance of the recognition. In the future we plan to investigate, how the Activity Driven Update can be combined with a top-down attentional process and how spontaneous synchronization can be used for segmentation and binding.

References

[1] Sven Behnke, Marcus Pfister, and Raúl Rojas. Recognition of handwritten digits using structural information. In *Proceedings of International Conference on Neural Networks, Houston*, volume 3, pages 1391–1396, 1997.

[2] Sven Behnke and Raúl Rojas. Neural abstraction pyramid: A hierarchical image understanding architecture. In *Proceedings IJCNN'98–Anchorage*, volume 2, pages 820–825, May 1998.

[3] J. Braun and D. Sagi. Vision outside the focus of attention. *Perception and Psychophysics*, 48:45–58, 1990.

[4] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern Recognition*, 19(1):41–47, 1986.

[5] R. Opara and F. Wörgötter. Using visual latencies to improve image segmentation. *Neural Computation*, 8(7):1493–1520, 1996.