Saulius Garalevicius

Using Memory - Prediction Framework For Invariant Pattern Recognition

Final report

December 2005

Problem specification

The objective of this project is to design, implement and analyze a system for recognizing invariant visual patterns based on the memory-prediction framework. The memory-prediction framework is a large-scale theory of the functioning of natural neural networks in human neocortex. The project aims at developing a novel learning platform suitable for running experiments, investigating memory-prediction concepts in practice and possible further elaboration of the system based on the experimental results.

The memory-prediction framework can be implemented using several different approaches. One of them is modeling the framework exactly according to the biological description provided in (1). Another approach is to build a model which may not be biologically accurate in its details but still work according to the same principles as described in (2). The goal of this project is to analyze and implement the memory-prediction framework using the principles outlined in (2) and experimentally investigate its characteristics and performance. During this project I used C++ to implement and experiment with the main structures of the model and apply it to learn to recognize visual two-dimensional black-and-white patterns. This task allows us to clearly monitor the performance of the system and visualize its internal representations.

Background knowledge and existing work

The memory-prediction theory focuses on the functioning of the human neocortex. A hierarchical network structure guides the functional principles of each region in the cortex. All regions in the hierarchy perform the same basic operation. The inputs to the regions at the lowest levels of the cortical hierarchy come from our senses and are represented by spatial and temporal patterns. However, a system running the neocortical algorithm will be able to learn based on any kind of patterns that we choose to give it.

The neocortex learns sequences of patterns by storing them in an invariant form in a hierarchical neural network. It recalls the patterns auto-associatively when given only partial or distorted inputs. The structure of stored invariant representations captures the important relationships in the world, independent of the details. The primary function of the neocortex is its ability to make predictions by combining knowledge of the invariant structure with the most recent observed details.

The regions in the hierarchical network are connected by multiple feedforward and feedback connections. Prediction requires a comparison between what is happening (feedforward) and what you expect to happen (feedback). Each region is a collection of many smaller subregions that are only connected to their neighbors indirectly, through regions higher up the hierarchy. Each region learns sequences, develops “names” (invariant representations) for the sequences it knows and passes these names to the next regions higher in the cortical hierarchy. As a region of cortex learns sequences, the input to the next region changes and the higher region can now learn sequences of these higher-order objects.

These are the basics of the memory-prediction theory first published in 2004 by Jeff Hawkins (1). Since the theory is novel, very few theoretical or practical studies of the memory-prediction framework exist today. The only published articles about modeling the framework are (2) and (3) by Dileep George. Jeff Hawkins has recently founded a new company called Numenta that is developing a new type of computer memory system modeled after the human neocortex, but it has not published anything yet.

In general, invariant pattern recognition has been an area of active research for a long time. These efforts were based on artificial neural networks as well as other technologies and usually used only the spatial image information to develop invariant representations (6, 7). However, the performance of these models was limited and generalization questionable.

System design

This pattern recognition system needs to be designed to simulate biological processes in the brain that are described in the memory-prediction framework. The basic design decision is how much equivalence the system should have to the biological processes in the brain. Since this is the top-down framework, the most significant task is to accurately implement the large-scale, high-level processes of neocortex, while the details of low-level structures can be approximated and do not need to mirror the functions of individual cells or small cell groups. The memory-prediction framework describes the neocortex as the hierarchy of regions. The regions consist of many small subregions, which in turn contain a collection of cortical columns, consisting of several neurons each. This model mirrors the biological structure of regions and subregions connected in a functional hierarchy, but implements the described functions of a subregion using selected programming techniques, not as a collection of columns and neurons. This design can be elaborated quite easily to include more biological details in a subregion without violating the overall architecture of the high-level cortical hierarchy.

Hierarchical architecture

The system is organized as a hierarchical network. The hierarchical structure is used by the learning and recognition algorithms. The hierarchy has several levels that model cortical regions. Each level consists of a number of modules called subregions. A subregion can have several child subregions in a lower level and a single parent subregion in the upper level. The 8x8 subregions in the bottom most level (level 0) accept the input black-and-white bitmap of size 32x32 pixels. Each subregion accepts a pixel patch of size 4x4 taken from the input image; hence level 0 subregions each have small receptive fields compared to the size of the whole image. These patches do not overlap in this implementation; however, it would also be possible to have partially overlapping input to adjacent subregions. There are 4x4 subregions in level 1; each is connected to 2x2 subregions in level 0. Thus a level 1 subregion receives information from a larger portion of the input image, however, only indirectly through the child subregions in level 0. The same principle applies to all hierarchy levels. Finally, in level 2 we have a single subregion that receives input from all 4x4 subregions in level 1 and thus covers the whole input image by receiving indirect information about it.

This type of hierarchical network structure is analogous to the hierarchy of the visual regions in human neocortex. The human neocortex is also organized as a hierarchy of cortical regions. The receptive field size in the cortical regions also gradually increases in the higher levels of the hierarchy. The neural structures in higher regions of the cortex represent increasingly complex structures and the structures in the top visual region represent visual objects just like in this model.

Learning algorithm

The learning process consists of two stages: feedforward and feedback. The two stages are performed for all subregions starting with the lowest region and finishing with the top region.

In the general case, every subregion learns temporal sequences of spatial patterns, where a sequence is a collection of a specified number of patterns occurring one after another. In the simplest case, the sequence length is one and we get a single pattern in a sequence.

During the feedforward stage of learning, a subregion learns the most likely sequences of its inputs. This can be done empirically by observing the inputs over all training examples and remembering in a set B all sequences with their percentage of occurrences greater than a certain threshold δ. If $\delta = 0$, all distinct observed sequences will be remembered. After the set B is formed, every sequence in B can be uniquely represented by its index k. This index constitutes the “name” of the sequence and is produced as an output to the higher level. This system stores a single shared set B for all subregions within the same region in order to save memory. This is justified by the fact that all subregions within a region in principle are able to observe and memorize the same sequences.

Once all level 0 subregions have finished their feedforward stage of learning, subregions in level 1 can begin their feedforward stage. The idea of learning is the same, but the information being remembered is different. Subregions in level 0 learned directly the sequences of their visual inputs. A subregion in level 1 receives the “names” of the memorized sequences in its child subregions as its input. Therefore a level 1 subregion learns the most frequent simultaneously occurring combinations of sequences observed in its child subregions in the level below. Once again, the index in the set of learned sequences is presented as an output. The same feedforward process can be repeated for an arbitrary number of hierarchy levels.

The second stage of learning is called a feedback (contextual embedding) stage. It starts in a child subregion after both the child and its parent subregions have completed their feedforward learning. The idea is to use feedback from the parent to the child subregion to embed the lower level sequences in the context of the higher level sequences. To achieve that, the "name" (index k) of the active sequence in the parent subregion is fed back to its child subregions. The child subregion uses this index provided by the parent to increment the count for its active sequence that just caused the parent in the higher level to produce the sequence k. Thus the child subregion forms a conditional probability distribution (CPD) matrix of its sequences given the sequences at the parent subregion. The feedback stage is also repeated for all subregions in all hierarchy levels. The CPD matrices are normalized at the end of all learning.

This model has only three hierarchy levels, and the learned most likely sequences in level 0 are predefined to simplify the learning process. These most likely sequences include simple 4x4 patterns of vertical and horizontal lines, diagonals, corners, line junctions and other frequently observed combinations. The visual input to the subregion in level 0 is compared to the patterns in memory and the pattern with the smallest Hamming distance from the input is selected. Then the index of the group the pattern belongs to is outputted as a "name" of the observed input. For level 1 subregions the threshold δ is set to 0 to maximize recognition performance at the expense of greater memory usage. The top level region has image categories as the output.

The system is able to learn invariances it is exposed to during training. The current system learns translation invariances because it is presented to various shifts of the same visual symbol during training (the whole training symbol is placed in various positions inside the 32x32 pixel input field). Thus the system learns to recognize the symbol independent of its position. The same principle could be applied to learn other invariances, such as flipped/mirrored image, etc.

Recognition algorithm

After the learning is completed, each subregion has a CPD matrix that stores the probability distribution of sequences of that subregion given the sequences of the parent subregion. Therefore, the resulting structure can be viewed as a tree structured Bayesian network. In this network the task of recognizing an image I can be described as follows. Given any image I, find the most likely set of known sequences at each subregion, the combination of which best explains the given image I in a probabilistic sense.

In the acyclic tree structured network this problem can be solved by using inference with local message passing. This system uses Pearl's belief propagation algorithm described in (5) to recognize the given image by finding the most likely explanation for it at each subregion in the hierarchy. The summary of the formulas for belief propagation rules used in the recognition algorithm can be found in pages 255-257 of (5).

Training data and process

The training data consists of 91 image categories with two examples each. Subsets of the training data containing smaller number of categories were also used for experimentation. Translations of the whole input symbol are generated from each training example within the boundaries of the 32x32 image. The training process is performed for each generated translation of the training example.

The following training process is performed for each input image. The image is fed as the input to the bottom region, while the output in the top region is set to the known numerical category of the image. Then the feedforward stage of learning is performed for every subregion starting from the bottom region and finishing with the region second from the top. Then the contextual embedding stage is performed for all subregions in the same order as for the feedforward stage. The only role of the top region during the learning process is to provide the category of the observed image for lower levels during the feedback stage.

Note that the training of the system described above requires only one pass through the training images, so each image is seen by the system only once. In addition, the system is able to perform quite well when only two slightly different training examples of each category are presented (although more examples would improve recognition performance). Because of these advantages, the training time is much faster than using iterative training in traditional neural networks. The current system takes about 4 minutes (C++ program on AMD Athlon XP 2000+, 512 Mb RAM, Windows XP) to learn the 91 image categories.

Testing data and process

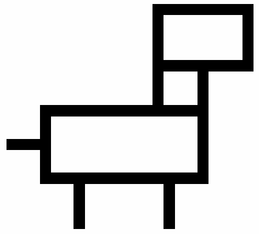
The testing data consisted of a number of user-drawn images that were reasonably similar to one of the categories training data. The testing process is performed as follows. The test image is fed as the input to the bottom region. Then the recognition is performed for all subregions using belief propagation as described above. The recognition process also starts at the subregions in the lowest level and goes up the hierarchy until it reaches the top subregion.

After the recognition procedure is completed, the top subregion outputs the learned category that best explains the given test image in a probabilistic sense. In addition, the top region stores the beliefs for all image categories, so we can extract more than one result from the output. This implementation provides ten best predicted categories for the given image together with graphical comparison of the beliefs associated with these categories. Not only the best predicted category is seen from this result, but we can also infer how confident the system was in predicting it as compared with the second-best and subsequent predictions. If the difference between the first and subsequent beliefs is large, the confidence of prediction is also large. If they have similar beliefs, the system shows that it did not definitely decide about just one category, but that the example may also be classified as one of the subsequent categories with similar beliefs.

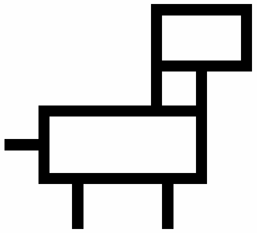
Evaluation and analysis of results

While experimenting with the system, I verified its capability to learn invariant representations from visual patterns, store these patterns in the hierarchy and recall them auto-associatively. During the experimentation I varied many internal constants affecting the learning process and also made modifications to the algorithms and data structures themselves.

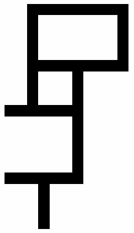
The following figures illustrate the main recognition capabilities of the system trained to recognize 91 categories of images. One of the two original training images in category "dog" is shown in figure 1. The system easily recognized the shifted version of the original image shown in figure 2. Note that the translation of images is the only invariance that the system is explicitly exposed to during the training; hence other invariances described below are discovered automatically by the system.



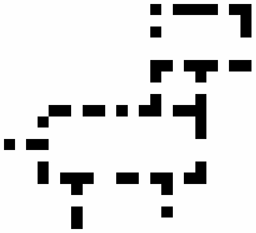
1. Original image



2. Shifted image



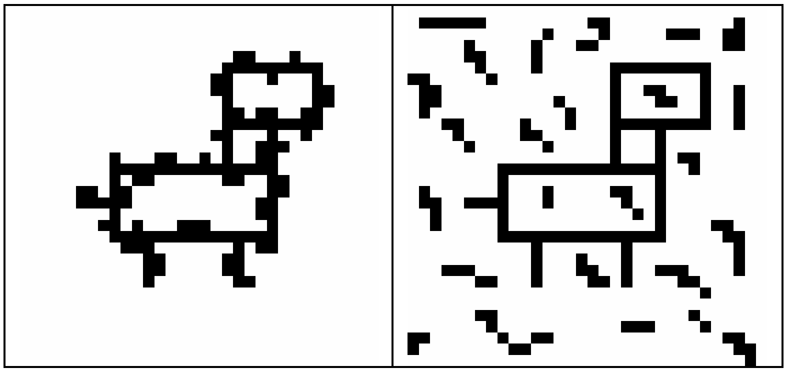
3. Half of the image



4. Part of the image

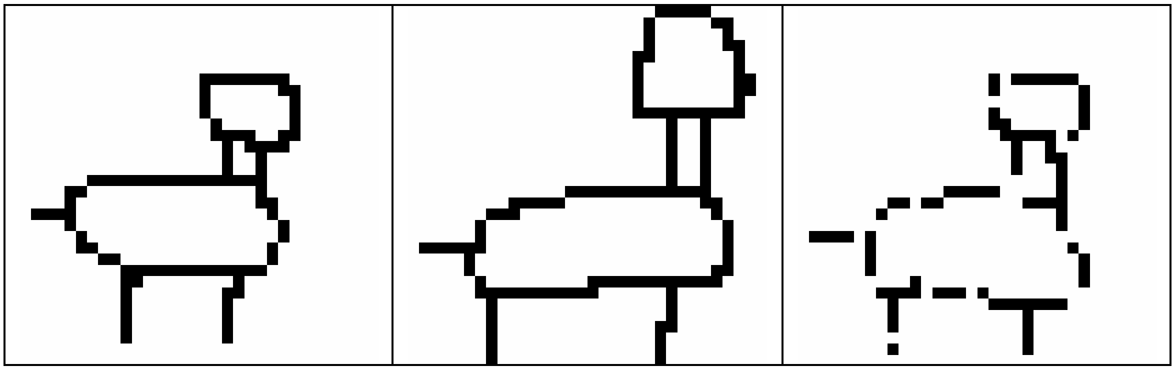
The system can function as an auto-associative memory, as demonstrated by figures 3 and 4. Given a part of the original image, the missing information is reconstructed and the category is predicted correctly. This resembles a capability of the brain to recall missing information given only partial input.

The system can also tolerate a substantial amount of noise of various types and still discern and correctly recognize the category as shown in figure 5.



5. Noisy images

Very interesting results are achieved while experimenting with arbitrary images drawn by the user that reasonably resemble one of the learned categories, but are distorted in various ways. All images shown in figure 6 were correctly recognized by the system.



6. Arbitrary hand-drawn images

Overall I observed that the system performs better while recognizing more complex images having more discernible features such as corners and line intersections, so for example images “I” and “J” are not recognized as well as “dog” or “helicopter”. The system also sometimes tends to confuse categories sharing many similar portions, such as letters “B” and “P”. I also observed that the recognition performance is slowly degrading when more and more categories are introduced in training, due to the same confusion between similar images. In some rare cases the system makes mistakes when recognizing an image although it is very similar to one of the training examples.

It is also useful to observe the relative strength of beliefs of the ten best predicted categories that is displayed by the system as a bar graph. When input image is not heavily distorted and resembles its true category much more that any other categories, we see the graph similar to the one shown in figure 7. We can judge from the graph that the winning prediction is very confident. When the input image is not readily recognizable or seems similar to several categories, the graph will look like the one in figure 8. There are several comparable predictions that correspond to the several best explanations of the given image that the system has found. Note that all ten predicted images in figure 8 have a rectangle in them which makes sense given the input image. Also, the best prediction "A" has two downward lines from the rectangle; the next three have one downward line, which is consistent with the given image in the order of decreasing similarity. This example shows that the system is able to classify similar images by taking into account their global large-scale topology (such as rectangles, line intersections at a specific position) although it is only given the input divided into many separate 4x4 pixel patches in the lowest region. Hence, the generalization and recognition of larger figures (topological parts of the object and the whole object itself) is performed by the higher hierarchy levels receiving only indirect inputs.

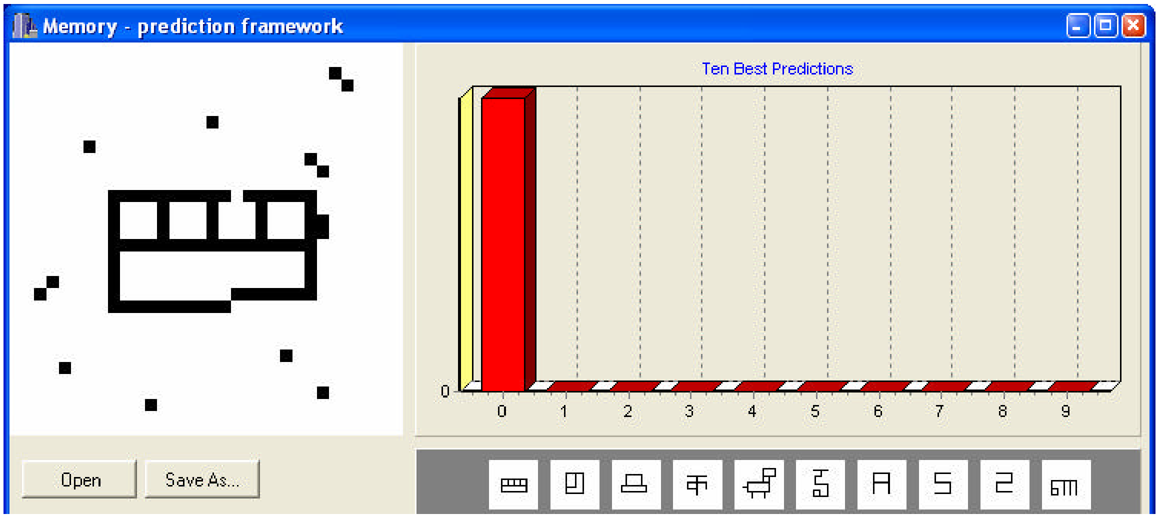


Figure 7. Confident prediction

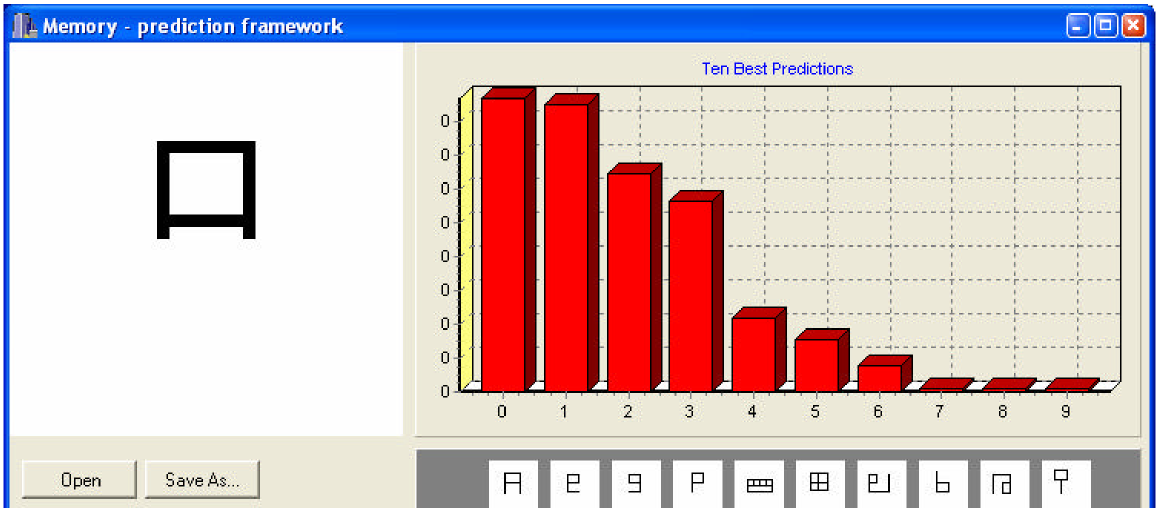


Figure 8. Uncertain prediction

Conclusions

In conclusion, we see that the system exhibits robust recognition performance that is invariant to various deformations, translations, noise and even large scale changes. Scale changes are one area where traditional neural networks have difficulty dealing with. Note that this performance is achieved by presenting only two training images of each category, each training image is seen by the system only once and the image is recognized by showing it to the system in a single position only. The recognition performance may be increased even further by adding more examples within the same category in the training set. It may also improve if the input image is shown in several different positions and inference is made based on all of them. These and other modifications are part of my future work.

This system outperforms my previous attempt to implement the memory-prediction framework. That approach used more intuitive, biology-oriented approach based exclusively on (1) and did not use inference based on belief propagation. As a result, it was able to function as an auto-associative memory and was quite robust to noise, but the performance in recognizing shifted, zoomed or transformed images was poor. This model recognizes various types of transformations much better.

This model also offers a number of advantages over traditional neural networks while delivering a comparable or better recognition performance. Inspired directly by the functioning of the natural neural networks in the human brain, the memory-prediction framework allows us to model these networks using a top-down approach. Therefore the overall large-scale structure of the model resembles that of the brain, independent of the microscopic details. On the contrary, traditional neural networks (at best) only try to model the low-level interactions between individual neurons without regard of the overall cortical structure. Thus the current model is more structure-oriented approach and can be meaningfully evaluated not just from behavioristic standpoint (percentage of correct answers), but also by interpreting its internal representations, understanding the kinds of transformations that can be recognized and so forth.

This model shares many common ideas with traditional neural networks. The hierarchy consists of many relatively simple units (subregions) that do the same basic operation and could be made to run in parallel. It solves problems by using cooperation between subregions without a centralized algorithm. The knowledge and beliefs in the system are distributed between the subregions in various hierarchy levels. It learns its skills by training and is able to generalize. The design could be applied to solving various problems, not just recognizing visual patterns. However, one of the main differences is that the memory-prediction framework is an inferential system that uses beliefs for learning and recognition.

Due to the above similarities, this model shares a number of advantages with neural networks. It clearly can function as an associative memory, can tolerate noise and generalize training images to similar ones. It could in principle be made fault tolerant, especially regarding faults in subregions in lower levels. In addition, the model eliminates or reduces some of the disadvantages of neural networks. The internal knowledge and learned representations can be interpreted with little difficulty by a human being; we can analyze and explain the answer given by the system more easily. It has potentially much shorter training time, uses "one-shot" (not iterative) learning and in principle could accept additional training categories anytime, even after the initial learning is completed. It does not treat a learning problem as a discovery of a mapping of an unknown function and therefore it can provide several reasonable outputs for a single ambiguous input. Finally, it offers a greater promise of understanding what intelligence is by closely modeling the overall structure of the human neocortex.

Overall, this practical implementation confirmed the main theoretical principles of the memory-prediction framework and exhibited many promising qualities. However, a lot of research remains to be done to extend the model and address the mentioned shortcomings of the current system. This implementation of the memory-prediction framework offers many possibilities for future research and refinement of the current model.

References

1. Jeff Hawkins, Sandra Blakeslee „On Intelligence“, 2004
2. Dileep George, Jeff Hawkins „A Hierarchical Bayesian Model of Invariant Pattern Recognition in the Visual Cortex“, 2004
3. Dileep George, Jeff Hawkins „Invariant Pattern Recognition using Bayesian Inference on Hierarchical Sequences“, 2004
4. Discussion group on implementing cortexlike learning systems: http://www.onintelligence.org/forum/viewforum.php?f=3
5. Judea Pearl „Probabilistic Reasoning in Intelligent Systems“, 1988
6. Kunihiko Fukushima „Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position“. Biological Cybernetics, 36(4): 193-202, 1980
7. Maximilian Riesenhuber, Tomaso Poggio „Hierarchical models of object recognition in cortex”. Nature Neuroscience, 2(11): 1019-1025, November 1999.