2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Using Recurrent Networks for Non-Temporal Classification Tasks

Saurav Biswas, Muhammad Zeshan Afzal, Thomas Breuel
saurav@iupr.com, afzal@iupr.com, tmb@iupr.com
Image Understanding and Pattern Recognition Lab,
Technical University Kaiserslautern,
Department of Computer Science,
Kaiserslautern-67663,
Germany

***Abstract*— In recent years, deep neural networks have led to considerable advances in the performance of neural network architectures. However, deep architectures tend to have a large numbers of parameters, leading to long training times and the need for huge amounts of training data and regularization. In addition, biological neural networks make extensive use of recurrent and feedback connections, which are absent for most commonly used deep architectures. In this paper, we investigate the use of recurrent neural networks as an alternative to deep architectures. The approach replaces depth with recurrent computations through time. It can also be seen as a deep architecture with parameter tying. We show that for a comparable numbers of parameters or complexity, replacing depth with recurrency can result in improved performance.**

I. Introduction

Deep learning in neural networks has, in recent years, been applied to real-world classification and recognition problems [1]. Deep neural networks have been very successful in visual recognition problems [2]. They are loosely motivated by ideas of feature hierarchies, for example the Neocognitron [3] and the HMAX model [4], [5] to extract biological features from the images, and information theoretic considerations.

While on one side, deep learning has been used to simulate the feature hierarchies, another active area of research are recurrent neural networks (RNN) which have been used to mimic the recurrent processing in visual cortex. The importance of recurrence has also been discussed in neurophysiology and there are a number of evidences supporting the presence of recurrence either within the same level of the visual cortex [6], [7] or within different levels. The communication between the different levels of visual cortex is driven by the feedforward and the feedback connections [8], [9], [10]. A good review that postulates the presence of recurrent connections is by Lamme and Roelfsema [6]. The primary reasoning behind the recurrence is the presence of the neurons which remain active after they have participated in the feedforward step. Lamme and Roelfsema [6] devised several criterias on the basis of which one could distinguish the recurrent processing from the feedforward sweep.

The first indication is the change in the tuning of neuron. While the early response of the neurons only depicts the presence of input used for stimulation, later the same neurons convey information about the properties of the object. The contextual information occuring outside the classical receptive field modulates the cell's response which is considered as the second indication for the involvment of the recurrent connections. The third indication which is devised is the processing time of a visual task. The delayed processing times in activities in visual tasks(like visual search and curve tracing) are used to implicate the presence of the recurrent connections.

The role of recurrence has also been explored in the context of specific visual tasks. One such study has been carried out by Koivisto. et. al. [11]. The causal role of recurrent processes in visual cortex has been explored. There also exist some studies which describe the amount of recurent processing required for a specific visual task. For exmaple Meuwese et. al. [12] discussed the metacognitive ability in general for object detection and categorization. They have postulated that metacognitive ability relies more on recurrent processes. The work describes that the amount of recurrent processing that takes place for object categorization is more than the recurrent processing required for detection.

In parallel with advances in deep learning, recurrent neural networks (RNN) have made significant advances in classification and recognition tasks,specially with Long Short-Term Memory (LSTM) networks. LSTM networks are recurrent neural network architectures intended to address the vanishing gradient problem [13]. They have demonstrated competitive performance on a number of sequence classification and Optical Character Recognition (OCR) tasks [14], [15]. When applied to temporal sequences, the recurrence relationship of the neural network naturally maps to the time axis of the input sequence. When applied to image classification, LSTM networks have been used by treating each dimension of the image analogous to a temporal axis and combining the outputs [16] Another way of looking at this is that multidimensional LSTM networks function like a kind of nonlinear infinite impulse response filter. The nonlinear feedback, or the "impulse reponse" that the LSTM network incorporates over time doesn't become zero at any certain

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

timestep. The network is thus able to maintain its memory state over long periods of time.

Recurrent neural networks are structurally similar to Multilayer Perceptrons (MLP) with the distinction that there are connections between hidden units, which introduce feedback in the network. Through these connections, the network is able to retain information about the previous inputs, and discover temporal correlations in data that are far away from each other. Training RNNs involves using the Backpropagation through Time (BPTT) algorithm [17]. To propagate the gradients of the error function back through the network, it is *unfolded* over time. Thus the recurrent weights, in the unfolded representation are shared between the layers of the unfolded network in the sense that each layer has the same copy of weights. Thus RNNs demonstrate the concept of *parameter tying* by virtue of sharing the recurrent weights over all timesteps of the unfolded network. The gradients from the final layer are then propagated back over all the layers of the unfolded network, the deltas being added up and averaged over all the timesteps finally. This value is then added to current value of the recurrent weight matrix. This process is repeated for all input sequences in the dataset.

The general approach of replacing deep learning architectures with recurrent architectures can be taken with any recurrent neural network architecture. We choose the LSTM architecture because of the high performance results that it has demonstrated on sequence classification tasks[18], [16]. In the sequence classification mode, the label or the class that the input sequence belongs to is output by the LSTM at the end of the sequence, that is after receiving the input of the last timestep of the sequence.

An LSTM consists of an input layer, an output layer and hidden layer(s) which are similar structurally to a hidden layer in an MLP. Additionally each node in the hidden layer (LSTM node), has three gating networks that determine how the output of the input layer is stored, retained, and output from the memory unit.

When used for just a single timestep, an LSTM network reduces effectively to the product of two MLP-like layers. So there is not effectively much of a difference in performances between a standard MLP network and an LSTM network using just one timestep input sequences. However, as the number of timesteps are increased, the LSTM network can effectively use the context from previous timesteps. It can thereby extract useful information from the entire sequence, functioning like a recursive filter. Pertaining to images, it can extract strongly correlated scale, shift and rotation invariant features from the entire image, enabling better classification performance. The rest of the paper is organized as follows. The next section motivates the use of recurrence in context of feature learning computationally followed by the experiments and results. The section that follows this discusses the findings and the paper concludes with shedding a light on the future directions.

II. MOTIVATION

Learning interesting features from images has been of primal importance in the context of visual object recognition tasks applied to neural networks architectures. Normally standard object recognition approaches using feedforward neural networks involve considering a time window that slides over the entire image and then the feature vectors that are extracted are fed into the network. However the training step in that case, takes an incredibly large amount of time. The size of the sliding window must also be prespecified or chosen via a grid search algorithm for the best performance. In such a scenario, using good feature extractors assumes primal importance.

A fundamental requirement for visual information processing is the capacity to establish visualy abstract shape properties and spatial relations [19]. The paper states that computation of spatial relations divides analysis of visual information into two stages. The first stage is the bottom up creation of certain representations of the visible environment. The second stage involves the application of processes called *visual routines* to the representations constructed in the first stage. These routines establish properties and relations that can't be represented explicitly in the initial representations. These visual routines are thus composed of sequences of elemental operations. Using a fixed set of these operations visual system in humans can assemble different routines to extract an unbounded variety of shapes properties and spatial relations. Sequential function approximators like perceptrons or multilayer perceptrons are not able to use this approach and hence perform poorly on visual object recognition tasks. The human brain, which is an integral part of the visual system, is highly recurrent in nature. This indicates that recurrence in learning architectures could prove effective in learning visual object representations.

Another important problem in the field of machine learning is the *connectedness predicate* problem, first described by Minsky in his paper [20]. The connectedness predicate states that it is necessary to classify whether an input pattern is connected or not. In other words, given any two remote pixels, one would be interested in learning if they were connected by a path of neighbouring pixels. Thus, one basically wants to learn if global connectedness exists, speaking in the context of images.

The order of the predicate is the smallest number of pixels in the image that must be sensed by some feature detector in order to compute the predicate. If the image is denoted by R, and $|R|$ is the number of pixels in the image, then the order of the predicate is unbounded, and increases atleast as fast as $\sqrt{|R|}$. To compute the predicate of an unbounded order, one requires usage of feature detectors with too large receptive fields(relative to R). Hence computationally, this becomes intractable, specifically in the case of 2 dimensional sequences, such as images.

Sequential classifiers such as perceptrons and multilayer perceptrons, have been shown to not being able to solve the connectedness problem [20]. With the recent advances in deep learning, one could try to apply deep networks to try learning this predicate. However, deep architectures are also inherently sequential in nature. Traditional architectures such as Convolutional nets have cells called receptive fields, which are sensitive to small sub regions of the input space. These fields are however, local in input space and exploit spatial pixel correlations which are local in nature. Additionally training deep networks requires tuning of several parameters.

Recurrence, on the other hand, has been used widely to describe certain properties which are exhibited in the visual cortex while processing a visual stimulus. While it has not been exactly determined what the exact kind of the features extracted and processed in visual cortex are (Edges [21], Gabor Filter [22]), recurrence has been shown to impact the way how the visual information has been processed regardless of the type of features. Recurrent neural networks, have been shown to be more powerful function approximators compared to multilayer perceptrons and other sequential networks [23]. The complexity of MLPs would be prohibitively large for some problem which RNNs could realize within acceptable computational constraints. Since output is fed back into the hidden layers in a recurrent architecture, RNNs can learn to identify connectedness in arbitrarily long sequences, and hence hold a clear advantage over other architectures in solving the connectedness predicate.

Recurrent neural networks can thus prove to be successful substitutes for feature extraction tasks for the early stages leading up to visual object recognition. Given that the human visual cortex is also recurrent in nature, one is tempted to design feature extractors which can learn global connectedness and remote pixel correlations in images, without having to worry about tuning a large number of parameters, as one has to in the case of deep networks.

The idea that a recurrent architecture can be trained to learn image transformations is quite interesting to investigate. LSTM networks have already been successfully used for digit recognition tasks, on the MNIST dataset [16]. Datasets of real world images consist of objects under a variety of transformations. For example in case of a dataset of handwritten digits, the same digit can be have various degrees of skew, have different amounts of blurring. Some images of the same digit can be sheared or scaled. Deep learning architectures have already yielded high performance on learning to predict the correct digit class in datasets that consist of transformed representations of each digit [24]. Recurrent neural networks, by virtue of their ability to provide deep learning via weight sharing over time can prove to be useful to learn the degree and extent of various transformations that an image of an object can be made to undergo. For example, reshaping a 2D grayscale image of an object into a 1D vector and stacking up these images column-wise would give us a sequence of images. The images in one sequence could be progressively transformed. For example the sequence could consist of decreasingly skewed 1D image representations of one particular image. Each timestep would thus have a transformed image representation as its input. The target would identify the correct object class at the end of each sequence of images. We are interested in investigating the fact that given a sequence of such successively transformed images, a recurrent network is able to learn the transformations. Also we would like to know if the learning is better, i.e better accuracy on the test set if the input sequence lengths are increased. It is interesting to see if increasing the extent of transformation or *context* in each input sequence gives us better results in the sequence classification task.

Deep networks have performed the best on object recognition tasks, particularly on the MNIST dataset [25]. MNIST Images have been mostly used in experiments where the task is to label the images with their corresponding digit classes [26]. So far on this task convolutional neural networks have achieved the best results. Experiments have also been carried out using Multidimensional LSTMs [16], where they labelled each pixel in each image with the class the digit belonged to, in addition to having an extra class for background pixels.

We are primarily interested in investigating that, given a non temporal classification task, a recurrent architecture will perform better than its non temporal or standard feedforward counterpart. Well trained deep learning architectures with more parameters and different structure are known to perform best on these problems. In the next section, the experiment that we did is described in greater detail.

III. Methods and Experiments

We performed experiments on the MNIST dataset, which consists of isolated handwritten digits. The individual images are size normalized, centered and consist of single handwritten characters that are 28 pixels by width as well as by height. The data comes divided into 60,000 training and 10,000 test images.

Since we are aiming to replace a deep architecture with a recurrent neural network architecture, each layer in the deep architecture corresponds to a timestep in the recurrent architecture. What this means is that the equivalent of a deep architecture is a recurrent architecture that receives input only at time t=0, and receives zero input in subsequent timesteps. However it is more natural to provide inputs to the RNN at each timestep. We provide three forms of inputs in our experiments. For one set of experiments, we repeatedly input the original input at each timestep. In the second set, we provide successively less blurry versions of the input in each timestep. This is motivated by the idea that early processing stages in the recurrent neural network may perform coarse classification tasks, while in the later stages(timesteps) they make finer distinctions. In the final set, we provide successively sharpened versions of the original image per timestep to the network. Note that input in each timestep is the MNIST image reshaped to 784 pixels in one dimension and applied to the network. The number of timesteps are varied from n=1 to n=12, with the idea that 1 timestep sequence should be akin to the idea of feeding input to an MLP. This process is repeated for all the images in the dataset. For the set of experiments with blurred inputs, we convolved the image with a standard Gaussian kernel, with values of sigma (σ =0 to 11). One set used increasing values of σ and the other used decreasing values, as per the requirements for our experiments. The values for σ were chosen empirically. We limited the range of sigmas to 11 because we found that for values greater than 11, there is no significant decrease in test error. Also for σ values greater than 11, the visual input loses important information, hence is not able to provide a good basis for comparing the test error decrease.

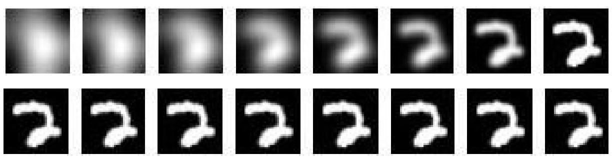

Fig. 1. The input to the LSTM is a sequence of MNIST images which have been reshaped to 1D vectors. In one setup the images are decreasingly blurred and in another no blurring is used. These sequences of images are input to the LSTM for training. The corresponding MNIST digit class serves as the target.

We used a one dimensional bidirectional LSTM for our experiments from the **RNNLIB** library written by Alex Graves [27]. The mode used was the sequence classification mode in rnnlib, which as the name suggests, runs over an entire sequence of timesteps and gives the label at the end for the whole sequence. The experiment was repeated with different configuration parameters.The validation set, from the MNIST dataset was used as an early stopping criterion, with a maximum of 30 allowed epochs for the experiment to run, after which it was stopped, provided there was no improvement in the validation set error. The input size was 784 since each timestep is a vertical column of 784 pixels. The output size was fixed at 10 which represent the 10 classes of the MNIST digits. We used just one hidden layer in all our experiments where we varied the number of LSTM memory blocks for different experiments. We used hidden blocks in the number of 10, 25, and 100.

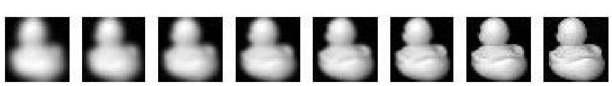

Fig. 2. The experimental setup where the COIL-20 images are stacked one after another, one image per timestep. The images are reshaped to 1D vectors and stacked over all the timesteps. In this setup, the images are blurred and the blur is decreased progressively. Each sequence of blurred COIL-20 images is an input to the LSTM

We also repeated the experiments on the COIL-20 dataset [28]. This dataset contains grayscale images of objects from 20 object classes. Each object has been placed on a turntable and the objects have been rotated through 5 degrees for each snapshot, thereby yielding 72 images of each of the 20 objects. Thus the dataset has 1440 grayscale images in total. We divided the whole dataset such that 75% is used as the training set and the remaining 25% as the test set. Thus the training set had 1080 images and the test set had 360 images.

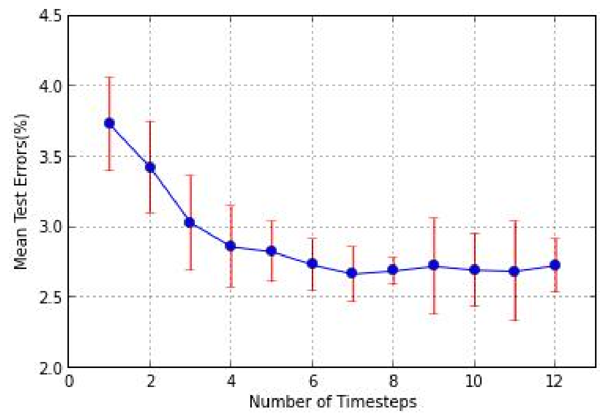

Fig. 3. Test errors for decreasing blur per timestep with 10 hidden neurons (MNIST). The experiments for each timestep have been repeated 10 times to flush out random weight initializations.

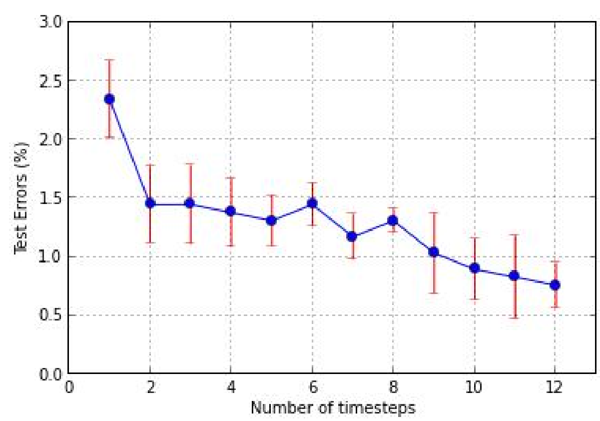

Fig. 4. Test Error decrease using 10 hidden neurons for COIL dataset. Experiments repeated 10 times for each timestep to flush out random weight initializations.

IV. Results

Our experiments showed us that the 1D bidirectional LSTM learns well with the addition of increasing number of timesteps. The LSTM network was tried on with different kinds of image representations in each timestep, for example we tried first with each timestep having the same image representation, or in other words the same image was repeated for all the timesteps. The number of timesteps were varied between n=1 to n=12. We repeated the experiments with each timestep having images with subsequently increasing blur and decreasing blur respectively. Each image was convolved with a gaussian kernel. The three kinds of experiments, were performed with one hidden layer having 100 LSTM blocks. We find that the setup with decreasing blur performs comparatively better than the other two setups. The decrease in test error over all the timesteps is not huge, but more gradual in the case of timesteps having images with decreasing blur, compared to the other two configurations. Now using the concept of decreasing blur per timestep, we repeated the experiment for all 12 timesteps, using different number of hidden LSTM blocks, namely 10 and 25.

We observed a tradeoff between the number of hidden LSTM blocks used and the number of timesteps in each sequence. The LSTM with 10 hidden LSTM blocks learned fastest and the test error showed the steepest decrease starting from 3.8% in the case of using just one timestep to 2.69% for 12 timesteps. In contrast the learning was slower with 25 hidden neurons and the decrease in test error was more flat, ranging from 2.80% to 2.23%. The final configuration with 100 hidden neurons learnt the slowest, and produced the flattest change in test error, ranging from 2.25% to 1.94%.

The experiments on the COIL-20 dataset yielded the same kind of pattern in the plot for the test errors per timestep as was evident in the experiment on the MNIST dataset. The decrease was more profound for the COIL-20 dataset, decreasing from 2.34% to 0.76%.

V. Discussion

The results of our classification tasks show that representations of an image (blurred representations in our case), can be successfully used to recognize the correct class label. The results from our experiment show that the test errors on the MNIST benchmarking dataset and thr COIL-20 dataset decrease significantly as we provide decreasingly blurred images per timestep, for timesteps ranging from n=1 to 12. This an indicator of the fact that recurrence can be used to learn image transformations without the use of a large number of sequentially deep layers and globally tunable parameters, the latter being the case with deep neural nets. However, the goal of the paper is not to compete with the state of the art results. Deep learning networks, with considerably large number of layers and significant number of parameters to be tuned, have provided the best results for classification tasks. In fact, Dan Ciresan [26] has shown that deep big simple neural nets give the best results (0.35% test error) for classifying the MNIST handwritten digits dataset. What we want to show is that one can improve upon classification results as one progessively increases the number of timesteps that one provides visual input for, to the recurrent architecture, thereby learning transformed representations of the original image.

Studies have shown that scale space is the first stage before more complicated feature detection steps which ultimately lead to object recognition in the mammalian cortex [29]. The main type of scale space is the Gaussian scale space. The Gaussian scale space constitutes the canonical way of generating a linear scale space, and also doesn't generate new spurious structures while going from a finer to coarser scale. In the real world, objects are composed of different structures at different scales. Thus real world objects may appear in different ways, depending on the scale of the observation. Unlike the mammalian visual cortex where scale space is inherent, for an artificial visual recognition system, there is no way to determine *a priori*, what scales are appropriate for describing the interesting structures in the image data. Hence the system must consider descriptions at multiple scales in order to capture unknown scale variations that may occur. This was one of the main motivations for us to use progessively increasing or decreasingly blurred images over increasing number of timesteps as input to the recurrent architecture.

As mentioned earlier, visual object recognition tasks require solving the *connectedness predicate* in images. This is where perceptrons and MLPs fail and deep networks require tuning of a large number of parameters for convergence. One also has to take care of learning *visual routines*, where the recognition system learns abstract represenations of individual parts [19], that facilitates the final recognition of the whole object we desire. As our experiments show, recurrent networks coupled with the idea of applying the same image, (or its abstract representations over a variety of scales), over different number of timesteps, can prove to be useful in learning useful features over time. This can prove useful in learning a variety of image transformations.

VI. Conclusion and Future Work

The results from our experiments suggest that the LSTM network is indeed able to learn the hidden representations of an object, as it progressively scans over each timestep. This kind of an approach seems quite promising and we intend to extend this idea further by trying to train LSTM networks to see if they can learn morphological transformations of an object. For example, one might try to train the 1D bidirectional LSTM to learn to correctly label objects, given that over each timestep, we pass skewed versions of the image. This might prove quite useful when we want to do multi object recognition in real life images.

References

[1] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks - a review," *Pattern Recognition*, vol. 35, no. 10, pp. 2279–2301, 2002.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1106–1114.

[3] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Networks*, vol. 1, no. 2, pp. 119–130, 1988.

[4] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," 1999.

[5] T. Serre and M. Riesenhuber, "Realistic modeling of simple and complex cell tuning in the hmax model, and implications for invariant object recognition in cortex," 2004.

[6] V. A. Lamme and P. R. Roelfsema, "The distinct modes of vision offered by feedforward and recurrent processing," *Trends in Neurosciences*, vol. 23, no. 11, pp. 571 – 579, 2000.
[7] J. M. Hupè, A. C. James, B. R. Payne, S. G. Lomber, P. Girard, and J. Bullier, "Cortical feedback improves discrimination between figure and background by v1, v2 and v3 neurons," *Nature*, vol. 394, no. 6695, pp. 784–787, 1998.
[8] J. Cudeiro and A. M. Sillito, "Looking back: corticothalamic feedback and early visual processing," *Trends in Neurosciences*, vol. 29, no. 6, pp. 298 – 306, 2006, neural substrates of cognition.
[9] C. Guo and L. Zhang, "An attention selection model with visual memory and online learning," in *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, Aug 2007, pp. 1295–1301.
[10] L. Huang, A. Treisman, and H. Pashler, "Characterizing the Limits of Human Visual Awareness," *Science*, vol. 317, no. 5839, pp. 823–825, Aug. 2007.
[11] M. Koivisto, H. Railo, A. Revonsuo, S. Vanni, and N. Salminen-Vaparanta, "Recurrent processing in v1/v2 contributes to categorization of natural scenes," *The Journal of Neuroscience*, vol. 31, no. 7, pp. 2488–2492, 2011.
[12] J. Meuwese, A. Loon, V. Lamme, and J. Fahrenfort, "The subjective experience of object recognition: comparing metacognition for object detection and object categorization," *Attention, Perception, & Psychophysics*, pp. 1–12, 2014.
[13] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, Apr. 1998.
[14] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks."
[15] T. M. Breuel, A. Ul-Hasan, M. A. Al-Azawi, and F. Shafait, "High Performance OCR for English and Fraktur using LSTM Networks," in *Int. Conf. on Document Analysis and Recognition*, Aug. 2013.
[16] A. Graves, S. Fernández, and J. Schmidhuber, "Multi-dimensional recurrent neural networks," in *Artificial Neural Networks ICANN 2007*, ser. Lecture Notes in Computer Science, J. S, L. Alexandre, W. Duch, and D. Mandic, Eds. Springer Berlin Heidelberg, 2007, vol. 4668, pp. 549–558.
[17] P. Werbos, "Back propagation through time: What it does and how to do it." in *Proceedings of the IEEE*, vol. 78, 1990, pp. 1550–1560.
[18] A. Graves, "Supervised Sequence Labelling with Recurrent Neural Networks," Ph.D. dissertation.
[19] S. Ullman, "Visual routines." *Cognition*, vol. 18, no. 1-3, pp. 97–159, Dec. 1984.
[20] M. Minsky and S. Papert, *Perceptrons - an introduction to computational geometry*. MIT Press, 1987.
[21] J. H. Elder and A. J. Sachs, "Psychophysical receptive fields of edge detection mechanisms," *Vision Research*, vol. 44, no. 8, pp. 795 – 813, 2004.
[22] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Am. A*, vol. 2, no. 7, pp. 1160–1169, Jul 1985.
[23] R. Ilin, R. Kozma, and P. Werbos, "Beyond feedforward models trained by backpropagation: A practical training tool for a more efficient universal approximator," *Neural Networks, IEEE Transactions on*, vol. 19, no. 6, pp. 929–937, June 2008.
[24] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "High-performance neural networks for visual object classification," *CoRR*, vol. abs/1102.0183, 2011.
[25] Y. Lecun and C. Cortes, "The MNIST database of handwritten digits."
[26] D. C. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *CoRR*, vol. abs/1202.2745, 2012.
[27] A. Graves, "Rnnlib: A recurrent neural network library for sequence learning problems," http://sourceforge.net/projects/rnnl/.
[28] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia Object Image Library (COIL-20)," Tech. Rep. CUCS-005-96, 1996.
[29] L. M. J. Florack, B. M. T. H. Romeny, J. J. Koenderink, and M. A. Viergever, "Scale and the differential structure of images," *Image and Vision Computing*, vol. 10, pp. 376–388, 1992.