IJICSE
Open Access

Contents lists available at www.ijicse.in

International Journal of Innovative Computer Science & Engineering

Volume 4 Issue 4; July-August-2017; Page No. 36-42

Neural Network for Image Classification

Rishab Sharma[1], Mohammad Zohaib[2], Akshay Kumar Sharma[3]

[1] Maharaja Agrasen Institute of Technology, Sector 22 Rohini, New Delhi

rishabsharmaddn@gmail.com

[2] BMS College of Engineering, Banglore India

zohaib27may@gmail.com

[3] IIT Roorkee

me@aksh.co

Received 04 August 2017; Accepted 28 August. 2017

ABSTRACT

As observed machine learning, computer vision techniques and other computer science algorithms cannot compete the human level of intelligence in pattern recognition such as hand written digits and traffic signs. But here we have reviewed a biologically plausible deep neural network architecture which can make it possible using a fully parameterizable GPU implementation deep neural network independent of the pre-wired feature extractors designing, which are rather learned in a supervised way. In this method tiny fields of winner neurons gives sparsely connected neural layers which leads to huge network depth as found in human like species between retina and visual cortex. The winning neurons are trained on many columns of deep neurons to attain expertise on pre-processed inputs in many different ways after which their predictions are averaged. Also GPU used, enables the models to be trained faster than usual. Upon testing the proposed method over MNIST handwriting data it achieves a near-human performance. Upon considering traffic sign recognition, our architecture has an upper hand by a factor of two. We also tried to improve the state-of-theart on a huge amount of common image classification benchmarks.

Keywords: Neural network, Machine learning, Computer vision

INTRODUCTION

A human visual system (HVS) model is used for image processing and computer vision to study various biological and psychological processes. HVS efficiently identifies objects within cluttered scenes. For computers and machine learning this is difficult due to viewpoint-dependent object variability, and the high in-class variability of many object types. Deep hierarchical neural models roughly mimic the nature of mammalian visual cortex, and are among the most promising architectures for such tasks. In this paper we review deep, hierarchical neural networks trained by simple back propagation and tested ove r MNIST [19], Latin letters [13], Chinese characters [22], traffic signs [33] and NORB (jittered, cluttered) [20] benchmarks , setting new and improved records. In the experiments we deep convey DNN have proved their efficiency on handwritten digits and other 3D objects but they prove their efficiency when they are wide and deep i.e many layers . But training such networks takes a lot of time which may range from several days to months. Also due to high data transfer latency multi-threading and multi CPU code is also not effective in this case therefore we use graphical processing units (GPUs) because of which large DNN are trained within days instead of months, thus making MCDNN feasible. In our implementation weights of the DNN updates after each image. We also show how combining several DNN columns into a Multi-column DNN (MCDNN) further decreases the error rate by 30-40%.

We start with a description of MCDNN architecture followed by creation of the training set and the data pre-processing and then we conclude by summarizing the results.

II. ARCHITECTURE

Deep Neural Network

Corresponding author: Rishab Sharma

A deep neural network (DNN) is an artificial neural network with multiple hidden layers between the input and output layers. Deep neural networks can model complex non-linear relationships similar to an artificial neural network. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives. The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network. In our case DNN has convolutional and max pooling layers, and each layer only receives incoming data from its previous layer. The neural network weights are optimized through minimization of the misclassification error over the training set.

Convolutional Neural Network and Convolutional layer

Convolutional neural network is similar to a simple neural network. Similar to an ordinary neural network they are made up of neurons that have learnable weights and biases. Some inputs is received by a neuron which performs a dot product and optionally follows it with a non-linearity. A single differentiable score function is expressed by the whole network from the raw image pixels on one end to class scores at the other. Also the convolutional neural network has a loss function on the last fully-connected layer. The convolutional layers do a convolution in 2D space on M^(n-1) input map provided with filters to give M^n active output maps by passing it through a non-linear activation function.

The process starts with the training of the weights of th deep neural network that are initially assigned randomly, training is done iteratively to minimize the error of classification on a training images. This is done by many methods: (a) our architecture is build with several maps per layer thus making it deep just like a nonlinear neurons stacked over one another which is found between the retina and visual cortex of the macaque monkeys. (b) Also to increase the efficiency of our architecture we use graphical processing units instead of noraml CPU's because of an observed increase in speed of processing. In GPU's a complex and massive code which takes several days to train in CPU , can be trained in just some hours thus speeding the process with additional factor of 60-100 compare to standard computer systems. (c) When our network is given a large amount to data to train on , it does not require any additional unsupervised pre-training. The Neural Network of this paper have 2D layers of winner-take-all neurons echoes receptive fields are overlapped to share the edges weight. Thus given some input pattern, our algorithm uses a max pooling techniques to determine the above mentioned winner neurons which are obtained by partitioning layers into quadratic regions of local inhibition which selects the most active neuron of each region and these neurons are trained further. Among the winners neuron layers some of them represent a smaller, down-sampled layer of lower resolution which then feeds the next layer in order. (d) As we filter out the winner-take-all neurons , at some point we reach a one dimensional layer from where only one trivial one dimensional winner-take-all region is feasible so the top of the order network hierarchy becomes a standard MLP (multi-layer perception) because of which the receptive fields and winner-take-all regions of our deep neural network often are minimal. So further training is given to the winner neurons but the rest of the neurons cannot forget what they learnt so far, but they are affected in the peripheral layers by weight changes. This decrease of computational changes per time interval corresponds to biologically plausible reduction of energy consumption. The weight updates occur after each gradient computation step in the training algorithm.

III. EXPERIMENT

In this experiment we used a Core i5-950 (2.3GHz) system, 18GB DDR3, and four GTX 580 graphic cards. The training images were taken in various forms and view angles which may include their translation, scaling and rotation on the other hand for validation the original pre-processed images are used. The network is trained until the error is zero which happens usually after 20 to 30 epochs. The initial weights of DNN are taken from uniform random distribution of range (−0.049, 0.049) and the activation function of each neuron is a scaled hyperbolic tangent. We have tried to bring improvement to the state-of-the-art to a huge extent on some well known image classification and computer vision benchmarks. We used a deep neural network with two input images (size = 48 x 48) along with a convolutional layer of hundred maps and filters (5x5) and non overlapping 2x2 regions of a max-pooling layer with a completely connected layer of three hundred hidden units and another with hundred hidden units which are then fully connected to six neurons output layers where one neuron is given per class.

The various image classification benchmarks on which we tested our network are:

(1)Modified National Institute of Standards and Technology database also known as MNIST is a simple but large computer vision dataset of handwritten digits that is mostly used in machine learning and computer vision for training various image processing systems. Five additional datasets were created by normalizing digit width of MNIST dataset between ten-twenty pixels which are originally normalised in a bounding box of 20 pixels, because of which we are able to have our data seems like to be collected from alternate and different angles. In the process we train a total of 40 MCDNN columns in which five columns are trained per normalization. All the deep neural networks are trained for 700 epochs with a slowly changing learning rate which is initialized with 0.002 multiplied by a factor of 0.991/epoch up to a value of 0.000029. This training of the deep neural network takes up to 15 hours and almost 500 training epochs to show an observable improvement. But during the training of the network the digit images are distorted randomly before every epoch. As later depicted in the results summarized in the table at the end of the experiment section, multi-column deep neural networks of five nets shows a better result than any other deep neural network upon being trained over the same pre-processor. We also note that the multicolumn deep neural network has an error rate of 0.23 percent which is very low as compared to other deep neural networks which when trained upon the same six datasets , yield an error of 0.55 percent thus showing that multi-column deep neural network are more efficient than DNN over the same pre-processed data . Also it marks a mile stone because it not only bring improvement to the state of the art but also shows that how an artificially developed method could challenge the human error rate of 0.2 percent on this task.

For further verification of the results we also trained five deep neural networks for each old normalization and it was seen that a sixty net multi-column deep neural network performs 0.21 percent similarly to the thirty five net multi-column deep neural networks, indicating that additional pre-processing does not further improve the efficiency of recognition.

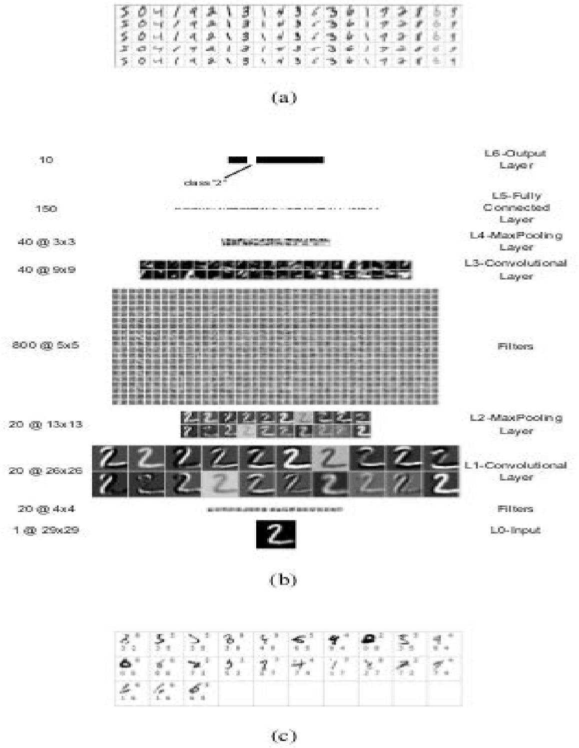

Figure 1 (a) Handwritten digits from the training set (b)DNN architecture for MNIST. weights of fully connected layers not displayed.
(c) The 23 errors of the MCDNN

(2) For our next experiment we apply our multi-column deep neural network architecture to Latin characters present in the dataset of NIST SD 19[13]. Our multi-column deep neural network proves a better recognition rate of about 2.5-5 times than other similar network architectures published. In the NIST SD 19 dataset 58000 out of 82000 characters are classified easily whereas the rest 24000 are hard to classify , which explains the minimized error rate of the sixty two class problem (3.6 % misclassification of the 58,000 digits) from the fifty two class letter problem (37% misclassification of the 24000 letter). When we classify letters there is a high amount of confusion between similar lower and upper case letters which makes letters in general , difficult (for example a,A and p,P) to classify , but in our experiment when the case insensitive data was taken, in that task error rate rates drop from 23% to 8% . Upon merging the lower case and upper case classes, 35 different classes are formed whose error rate is just bigger (8.33%) than the previous error rate. Due to smaller writer dependency on class variability classification of upper case letters is easier than the lowercase letters. The errors and confusions among different classes were analyzed in detail and in a informative way by the confusion matrix.

(3) Chinese characters are harder to recognize as

compared to Latin characters because of a larger category set and wide variability in writing styles and similar character confusion. In our experiment we use a CASIA dataset which consists of 300 samples of 3755 characters each but our system faced a computational challenge because of the resulting one million characters acquiring 3GB of data. The challenge was handled by the nets because of our fast GPU implementation else this amount of data would have taken about an year to train as only forward propagating the training set takes 30 hours on a CPU and thus the training time of a single epoch would have taken several days whereas in our case of GPU implementation it was about 4 hours making the network training feasible within some days instead of months. In our experiment we train the deep neural network on both offline as well as online data. In offline training for the character recognition task we place all the characters in a 50x50 image boundary after resizing all the characters to a 40x40 pixel size and normalizing their contrast. In online dataset training we place the resulting images in the centre of a 50x50 image after resizing them to forty cross forty pixels and smooth out the resulting images over a 4x4 pixel neighbourhood using a Gaussian blurring filter and also using a uniform variance of 0.64. But as directed by the documentation of the dataset the starting two hundred and forty writers from the CASIA database are used for training whereas the remaining sixty writers are for testing purpose. Therefore the total number of test and training characters is 234238 and 938669 respectively. Our method is based on raw pixel intensities rather than feature extraction and the extracted features are subjected to a dimensionality reduction as followed by other proposed methods on this topic. In our method we learn dimensionality reduction and feature extraction in supervised way. We obtained a error rate of 8 percent compared to 10.01 percent error rate of the best method proposed so far[22] and a recognition rate of 6.53 percent as compared to 7.6 percent rate of the best method proposed so far[22]. Thus we reach a conclusion that although this classification was very hard because of many classes and very few samples per class, our completely supervised deep neural network was able to beat the state of the art by a huge margin.

(4) For our next experiment we tested our multi-column deep neural network with four columns on a collection of 3D model stereo images (NORB). The 3D objects are centrally placed on random backgrounds along with some cluttering from a second peripherally placed object. The database we are using , contains 55 toys belonging to five generic categories which are designed for experimenting with object recognition in three dimensional space. The objects used in the dataset were imaged under eight lighting conditions by two cameras, 18 azimuths and nine elevations. The training set has a total of 293200 images whereas 58220 images are for the testing set. In our experiment we scaled down the 110x110 pixel original images to 48x48 which was large enough to preserve the details in the image yet small enough for fast training. For our experiment no pre-processing was implemented for the dataset. We used the first two folds to perform the two rounds experiment in order to compare the results which do not use the complete training data with those which utilised the entire training data. Our network was tested over with many distortion parameters such as rotation (maximum 20 ◦) , translation (maximum 20 %) and scaling (maximum 15%) which were applied to all our NORB experiments. Our network architecture is although deep but still it has very few map layers and the setup of the learning rate is as follows: eta (start 0.003; stop 0.00000234 ; factor 0.91) .We observed that even if we trained less amount of data to the multi-column deep neural network , it still improves the state of the art from 5 percent to 3.5 percent as due to the small size of the network , our training rate grows as fast as up to 165s/epoch for 120 epochs thus giving 0.5 ms for each sample to test which makes this method to process the complete training set. When we double the maps keeping the architecture same and train the all 10 folds keeping the learning rate of setup same, we observe that the training time per epoch increases to 35 min and the testing time for a sample increases from 0.5 ms to 1.4ms which simply because now we are using a larger net and more amount of training data. But in exchange of slow training and testing rate, we acquire a low error rate of 2.5 % , which further improves the state of the art . Finally from NORB we conclude that about 85% error rates are simply associated with correct second predictions as ER drops from 2.5% to 0.45% when we consider second predictions. Also we observed that more than half the errors were because of the confusion between the trucks and cars, thus NORB classification is hard although it only has six classes, training and test instances which sometimes differ by huge margin.

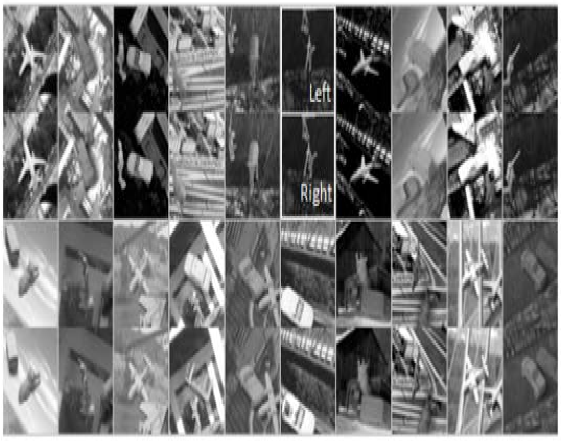

Figure 2 Twenty NORB stereo images

(5) For our last experiment we tested our network over GTSRB dataset for traffic sign recognition which is a very important application of computer vision for the automotive Industry and an effort in the field of driver's assistance. The training set has 26640 images out of which about 12500 are for testing. The images of the dataset contain one traffic sign each which vary in size from 14x14 to 251x251 pixels with a 10% border around the traffic sign and they all are not squares. For processing we crop every image to a bounding box frame because our deep neural network implementation needs all the input images to be of the same size therefore we resize all the training images to a 50x50 pixels after a visual inspection. So because of resizing, the scaling factor for any image with a rectangular bounding box is different for both the axis and some images are forced to have to have a squared bounding box. In our experiment we used standard image pre-processing methods to normalize the input traffic sign images which vary too much in contrast and illumination. We achieved an error rate of 0.6% on the test set after training five deep neural network for each dataset which resulted in a multi-column deep neural network with twenty five columns. Among the 70 errors which come up, over 82% are linked with second predictions thus we can infer that our network is unsure of the traffic symbol classification it is doing and therefore the probabilities of the predicted class are very low erroneously. But as most of its predicted class probabilities are either close to one or close to zero, therefore we can say that in general it is confident of its predictions. We get an error rate of 0.28% upon rejecting only one percent of all images whereas to get an error rate of 0.03% on a single classification, the percentage of images to be rejected increases to about 6.8% giving a confidence below 0.9. We can check about 83 images per second on one GPU with our multi-column deep neural network. As mentioned earlier, we used four GPU's for our implementation which takes about 40 hours to train our multicolumn deep neural network with twenty five columns which pays off by out performing one of the best algorithms by a factor 2.5.

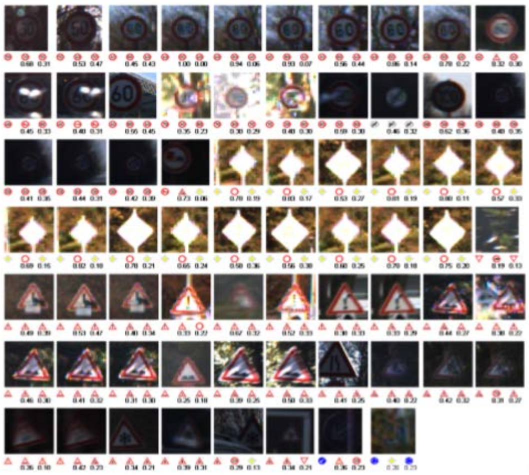

Figure 3: The 70 errors of the MCDNN, Best predictions (middle and right).

Table 1: Results on different datasets in %

Dataset	*Best result [%]*	*Multi-column Deep Neural Network [%]*
MNIST	*0.4*	*0.3*
HWDB1.0 on	*7.6*	*5.8*
NIST SD 19	*9*	*8.3*
HWDB1.0 off	*10*	*7*
Traffic signs	*1.7*	*0.6*
NORB	*5*	*3*

VII. CONCLUSION

One of the main achievements is that we are able to compete human intelligence on various computer vision benchmarks whereas we also made effort to enhance the state of the art at the same time by 20-30 %. Another conclusion is that our method being completely supervised does not require additional data to feed upon and still it improves the image recognition benchmarks like MNIST, NORB , Chinese characters and traffic signs. Therefore we conclude that combining small efficient neural networks into deep and multi layer networks not only boosts their performance but also increase their efficiency and accuracy.

VIII. Acknowledgment

This work was inspired by J¨urgen Schmidhuber work on image classification as well as from Ueda, N work on optimal linear combination of neural networks for improving classification performance.

REFERENCES

1. *S. Behnke. Hierarchical Neural Networks for Image Interpretation, volume 2766 of Lecture Notes in Computer Science. Springer, 2003. 1, 2*

2. *Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In Neural Information Processing Systems, 2007. 1, 2*

3. *N. P. Bichot, A. F. Rossi, and R. Desimone. Parallel and serial neural mechanisms for visual search in macaque area V4. Science, 308:529–534, 2005. 1*

4. *P. R. Cavalin, A. de Souza Britto Jr., F. Bortolozzi, R. Sabourin, and L. E. S. de Oliveira. An implicit segmentation-based method for recognition of handwritten strings of characters. In SAC, pages 836–840, 2006. 4*

5. *D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. Neural Computation, 22(12):3207–3220, 2010. 1, 3*

6. *D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Convolutional neural network committees for handwritten character classification. In International Conference on Document Analysis and Recognition, pages 1250–1254, 2011. 1, 3*

7. *D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In International Joint Conference on Artificial Intelligence, pages 1237–1242, 2011. 1, 2, 3, 6*

8. *A. Coates and A. Y. Ng. The importance of encoding versus training with sparse coding and vector quantization. In International Conference on Machine Learning, 2011. 5, 6*

9. *E.M.Dos Santos, L.S.Oliveira, R.Sabourin ,and P.Maupin. Overfitting in the selection of classifier ensembles: acomparative study between pso and ga. In Conference on Genetic and Evolutionary Computation, pages 1423–1424. ACM, 2008. 4 [10] A. C. P.-A. M. P. V. Dumitru Erhan, Yoshua Bengio and S. Bengio. Why does unsupervised pre-training help deep learning? Journal of Machine Learning Research, 11:625– 660, 2010. 1, 2*

10. *K. Fukushima. Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4):193–202, 1980. 1*

11. *E.Granger,P.Henniges,andR.Sabourin. Supervised Learning of Fuzzy ARTMAP Neural Networks Through Particle Swarm Optimization. PatternRecognition,1:27–60,2007. 4*

12. *P. J. Grother. NIST special database 19 - Handprinted forms and characters database. Technical report, National Institute of Standards and Technology (NIST), 1995. 1, 4*

13. *S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In S. C. Kremer and J. F. Kolen, editors, A Field Guide to Dynamical Recurrent Neural Networks. IEEE Press, 2001. 1*

14. *D.H.Hubel and T.Wiesel. Receptive fields ,binocular interaction, and functional architecture in the cat’s visual cortex. Journal of Physiology (London), 160:106–154, 1962. 2*

15. *A. L. Koerich and P. R. Kalva. Unconstrained handwritten character recognition using metaclasses of characters. In Intl. Conf. on Image Processing, pages 542–545, 2005. 4*

16. *A. Krizhevsky. Learning multiple layers of features from tiny images. Master’sthesis, Computer Science Department, University of Toronto, 2009. 1, 6*

17. *Y.LeCun. Uneproc ´edured’ apprentis sagepourr´eseauaseuil asymmetrique (a learning scheme for asymmetric threshold networks). In Proceedings of Cognitiva 85, pages 599–604, Paris, France, 1985. 1, 2*

18. *Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradientbased learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, November 1998. 1, 2, 3*

19. *Y. LeCun, F.-J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. InComputerVisionandPatternRecognition,2004 . 1, 5*

20. *Y. LeCun, L. D. Jackel, L. Bottou, C. Cortes, J.S.Denker, H.Drucker ,I.Guyon, U.A.Muller, E.Sackinger, P.Simard, and V. Vapnik. Learning algorithms for classification: A comparison on handwritten digit recognition. In J.H.Oh, C.Kwon, and S.Cho, editors ,Neural Networks: The Statistical Mechanics Perspective, pages261–276. WorldScientific, 1995. 3*

21. *C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang. Chinese Handwriting Recognition Contest. In Chinese Conference on Pattern Recognition, 2010. 1, 4, 5*

22. *J. Milgram, M. Cheriet, and R. Sabourin. Estimating accurate multi-class probabilities with support vector machines.*

23. *In Int. Joint Conf. on Neural Networks, pages 1906–1911, 2005. 4*

24. *M.Ranzato,Y.-L.B.FuJieHuang,and Y.LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In Proc. of Computer Vision and Pattern Recognition Conference, 2007. 2*

25. *M. Ranzato, F. Huang, Y. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In Proc. Computer Vision and Pattern Recognition Conference (CVPR'07). IEEE Press, 2007. 3*

26. *M. A. Ranzato, C. Poultney, S. Chopra, and Y. Lecun. Efficient learning of sparse representations with an energy-based model. In Advances in Neural Information Processing Systems (NIPS 2006), 2006. 3*

27. *M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. Nat. Neurosci., 2(11):1019–1025, 1999. 2 [28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations, pages 318–362. MIT Press, Cambridge, MA, USA, 1986. 1, 2*

28. *R. Salakhutdinov and G. Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In Proc. Of the International Conference on Artificial Intelligence and Statistics, volume 11, 2007. 2*

29. *D.Scherer,A.M¨uller,andS.Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In International Conference on Artificial Neural Networks, 2010. 5*

30. *T. Serre, L. Wolf, S. M. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. IEEE Trans. Pattern Anal. Mach. Intell., 29(3):411–426, 2007. 2*

31. *P.Y.Simard,D.Steinkraus,andJ.C.Platt. Best practices for convolutional neural networks applied to visual document analysis. In Seventh International Conference on Document Analysis and Recognition, pages 958–963, 2003. 1, 2, 3*

32. *J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The German Traffic Sign Recognition Benchmark: Amulti-class classification competition. InInternational Joint Conference on Neural Networks, 2011. 1, 5*

33. *D.Strigl,K.Kofler,andS.Podlipnig. Performanceandscalability of gpu-based convolutional neural networks. Parallel, Distributed, and Network-Based Processing, Euromicro Conference on, 0:317–324, 2010. 1*

34. *R. Uetz and S. Behnke. Large-scale object recognition with CUDA-accelerated hierarchical neural networks. In IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), 2009. 1*

35. *P. J. Werbos. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD thesis, Harvard University, 1974. 1, 2*

36. *D. H. Wiesel and T. N. Hubel. Receptive fields of single euronesinthe cat'sstriate cortex. J.Physiol.,148:574–591, 1959. 2*

37. *Multi-column Deep Neural Networks for Image Classification Dan Cires¸an, Ueli Meier and J¨urgen Schmidhuber.*