Universität Hamburg
Department Informatik

On the usage of Deep Learning Methods in Object Detection and Recognition

Seminar Paper

Bio-Inspired Artificial Intelligence

Ali Saleh
Matr.Nr. 6517831
3saleh@informatik.uni-hamburg.de

18.12.2013

Abstract

Machine Learning algorithms depends highly on how you present the data to the algorithm and a successful algorithm can grasp info and details from the presented data, a multi-level learning algorithm use different layers of feature detectors so that each layer can grasp several aspects and features of the input data and be used as a new data input for the next layer this method can be viewed as searching the input space from several different views and leads to grasping more info about the input than that grasped by any other non-hierarchical algorithm. this paper will give an introduction about Deep learning , previous work, and current breakthroughs.It will also give a deeper explanation for a recent attempt to use a deep hierarchical algorithms to detect and recognize objects in images and how this algorithm can be scaled to be used with real time applications.

Contents

1 Introduction

One of the earliest research fields that mathematicians and computer scientists have been working on is the idea of a machine that can show a presumably intelligent behavior. This led to many attempts to define intelligence and ways to measure it [13] and a sub-fields begun to shape many ways to simulate human intelligence using machines, and a usual way to tackle a AI problem is to decompose it to a number of sub-problems that solving each collectively simulate the original problem in a way that can be accepted as intelligence. Currently there are several ways to form a mathematical model that can learn and simulate human behavior, while supervised models have been dominant for a long time the late breakthroughs in the unsupervised learning and the development of new algorithms [4] [1] [?] for learning using multilevel learning architectures is attracting more attention to the deep learning unsupervised methods and leading to more development in this area and it's applications.

In tasks like speech recognition, object detection, or other sensory tasks brain reaches it's high performance level by working on the sensory data it acquire on different level and extract different representations and grasps different levels of abstractions that built up level by level to contribute in the final result/decision made by brain.Scientists have been working toward making a higher performance machines and algorithms to solve different tasks and problem in a way that can be interpreted as intelligent.From the different statistical models and algorithms used to achieve better performance in AI tasks Deep Learning is the algorithmic multi-level solution to simulate the way the human mind work [6].

This paper will be organized as following, in Section two a brief history of the Deep learning and several key researches will be described, then section three will discuss a recent attempt to develop new deep learning algorithms and the use of it in the areas of objects detection and classification, and there will be a comparison between the state of art algorithms in objects detection and classification and the algorithm discussed in Section three. finally section four well conclude the results out of this research.

2 Brief History of Deep Learning

One of the earliest attempts to design a multilevel computational model for recognition is attributed to Oliver Selfridge [11] in the late 1950s.Inspired by the work of Hubel and Wiesel [8] in his model the pandemonium Selfridge designed a four major recognition groups called demons where each level belongs to one of the major demons work in a specific stage of the recognition that collectively they act as a recognition system [see Figure 1].This model was able to recognize characters and had a weight adjustment method similar to the one in the back-propagation neural networks and can recognize patterns in new images exposed to it. This approach although pioneering but suffered several criticisms , for example it was argued that it needs a large number of training examples to be able to catch the different

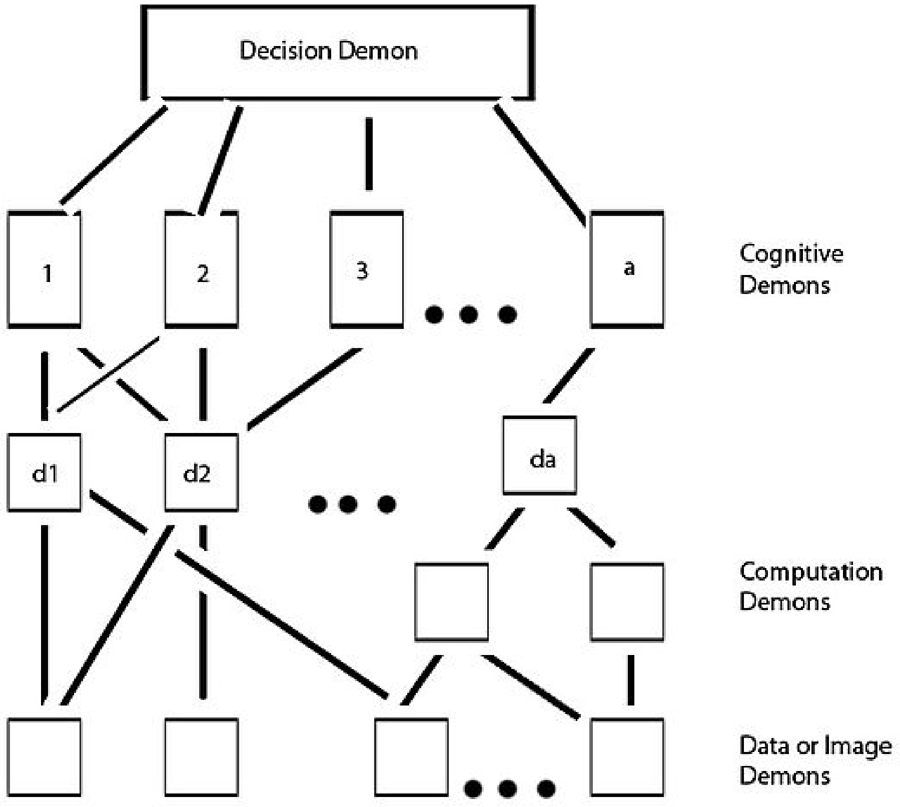


Figure 1: The concept of pandemonium architecture

possible patterns which may lead to it suffering from the same template matching problems [10], also most of the researches done building on the pandemonium architecture are towards a restricted area of recognition of simple hand drawings like written letters and generalizing those results to other areas of recognition may be misleading [9].

Years after that another Hubel and Wiesel inspired attempt have been made by Fukushima in the early 1980s [3] [2], Neocognitron is a multilevel neural network can be used for visual pattern recognition. Figure 2 shows the neocognitro architecture where is the lowest layer is used as an input layer. Each cell in a layer receives input from the layer directly before it's layer by being connected to the cells in a restricted area of this layer called it's receptive fields.The layers in this structure is divided in two groups based on the cell type of each layer. Those layers are alternatively organized so that each "C-Cells" layer is succeeded by an "S-Cells" layers plus the first layer which is the input layer and the last layer which is a "C-Cells" recognition layer.

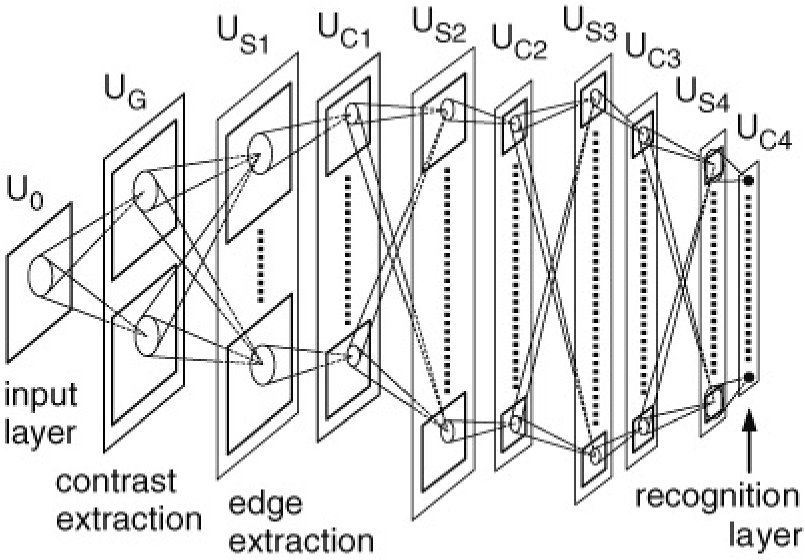


Figure 2: A typical neocognitron architecture (courtesy of Kunihiko Fukushima)

S-Cells works as features extractors , their input connections are variable and is changed by the learning process. The learning process determine the selective response of each s-cell towards the particular features present in it's receptive field.While **C-Cells** are fixed invariable cells that receive it's input from s-cells , each c-cell receive it's input from a group of s-cells that are trained to extract the same feature from different places in the image and give an output signal if at least one s-cell in it's receptive field gives an excitatory output so that if an feature changed it's position and the s-cell that responds is changed the c-cell will continue giving the same output.

During the training process the lower C-Cells extract the local features and the S-Cells tolerate the change in position of this features and gradually this local features build up into global features, and since s-cells tolerate some of the errors from the lower c-levels, s-cells at the highest levels gives robust responses to specific features even if it's slightly deformed. At the highest level C-cell works as recognition layer as it's integrate the information from all the previous levels, each cell responds only to one specific pattern even if the pattern is deformed , shifted in position , or changed in size. neocognitron is found in many different application like optical character recognition and it's advances and ingenuity is the base for many other advances in the area of deep learning.

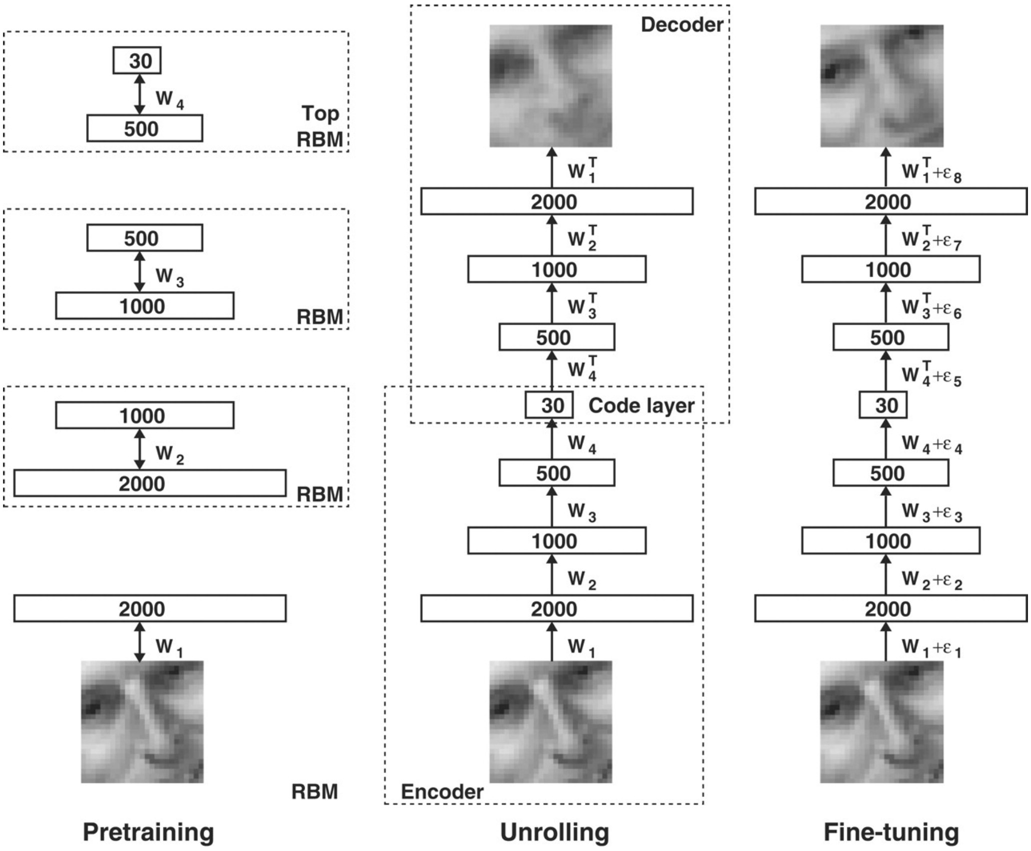
At the middle of the 1980s the Boltzmann Machine has been invented by Hinton & Sejnowski [5] [?] a stochastic neural network type that is composed of symmetrically connected units and is capable of learning internal features from the data. Typically a Boltzmann machine can be used in search and learning problems but we are here will be focusing on the learning problem. The Boltzmann machine can be trained by several data vectors and it can then it can generate those data vectors. There are two general structures of the Boltzmann machine one that have only visible units and the other can have hidden units also.The Boltzmann Machine had a major problem that prevented using it in a large scale application, as the number of neurons increase the learning time increase rapidly and the learning become very slow. In 1986 Smolensky invented the Restricted Boltzmann Machine [12] it consists as the normal Boltzmann Machine with the restriction that its neurons form a Bipartite graph and it have only connection between the hidden units and the visible units and no connections are between units from the same layer. The learning of the RBMs require taking several iterations using the training data and re-adjusting the connections weights using different learning rules. As we will see below RBMs can be used in deep learning by constructing multiple RMBs one over the other and using the results from one machine as the data for the next one.

Nowadays deep learning is getting more and more of the interest from computer science researchers and a lot of effort have been given toward solving different problems in different research areas using deep learning algorithms like speech recognition , signal processing, natural language processing and object recognition.

In 2006 Hinton [7] introduced the generative deep belief networks that uses RBMs as it's basis , the Deep belief network can be viewed as a set of RBMs stacked one over the other and each layer receives it's connection from the output units of the RBMs in the previous layer and it's learning algorithm learn layer by layer. The first layer uses the training data and the learning rule and algorithm and backpropagating the error to adjust it's it's weights to achieve better performance, and then the output of this layer is used as input for the next layer and so on. Hinton reported [7] the results of using the deep belief network for classifying the MNIST dataset. The whole training time of the network was about one week , and the error rate of the best model was 1.25%. This method while introduced a new way of training neural networks suffered from some limitations (some of them was solved and will be mentioned in the next section) it works well with binary images but for a nonbinary values in images it's treated as probabilities, it's not translation invariant [?] and scaling the algorithm up to work with real size images was a challenge at this time.

In he same year Hinton [4] showed how a high-dimensional data can be reduced to lower dimensions data using pre-trained autoencoders, and showed that this yields a better results than the most common method Principal Component Analysis (PCA).The idea behind this research was using two sets of neural networks (encoder & decoder) one to produce lower dimensional data from the input and the other can reconstruct the original data from the lower dimensional results (see figure 3). Using several layers of RBMs stacked over each other the decoder and the encoder were built.The learning process begins with pretraining where each RBMs is learned alone and the activated feature resulting from each layer are presented to the next layer as the input the layers after then are unrolled together to create the autoencoder. After that comer the global fine tuning stage where backpropagation is used through the whole system to fine-tune the weights for optimal reconstruction. The system was tested against the sate-of-art algorithms PCA and latent semantic analysis (LSA) it was showen that the autoencoders gave much better reconstruction of the original data than both of those algorithms the tasks where among image reduction and reconstruction from the MNIST dataset and Olivetti face dataset and in both the Autoencoders gave better performance and the usage of pretraining stage gave better performance than without using it. ??

Figure 3: Autoencoders used in diminsionality reduction (Courtesy of George Hinton)



3 Deep Learning for detection and classification

3.1 Convolutional Deep Belief Networks

This paper [?] builds upon the previously introduced concept of the deep belief networks building a generative model of images with unsupervised learning.This paper addresses some of he issues mentioned before and other issues in dealing with larger images and make the model translation invariant to deal with having the objects in different places in image.The model have feature detectors that learns features in different location in the image, because feature detector that is able to learn information from one part of the image can be used to learn the same info from other parts though building a model that can learn large images using relatively small number of detectors.

This paper uses Convolution RBM rather than normal RBM the difference is that the CRBM share the weights of the hidden and visible layers are shared between all the locations in the image. the CRBM consists of two layers Visible layer V and hidden layer H. The visible layer contains $N \times N$ binary input units, while the hidden unit contains K groups of units, each group contains $M \times M$ binary units (a total of $M^2 \times K$ units) and each of those K groups is associated with a $J \times J$ filter , this filter weights are shared with all the units in the hidden group. Plus each hidden group K_i has a bias b_k and all the visible layer units have a shared bias c.A layer of probabilistic max-pooling is then added as the final layer to allow for covering larger areas of the input in the higher layers in a probabilistically sound way see Figure 4.

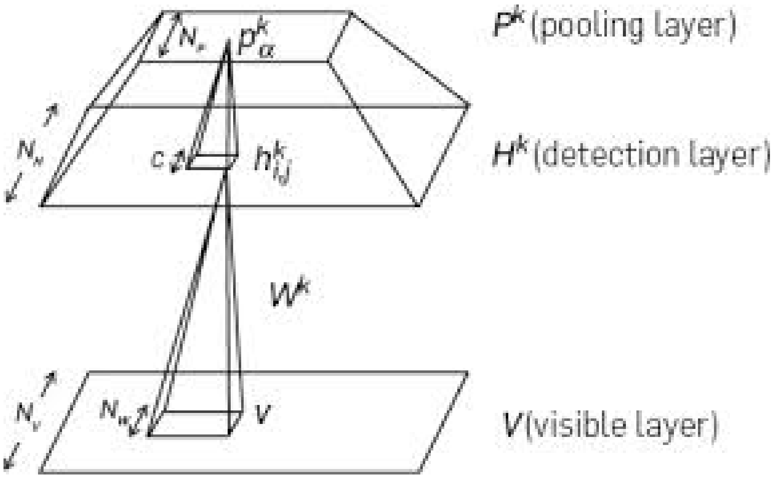


Figure 4: max-pooling CRBM (Coutresy of Honglak Lee)

The max pooling layer is intended to " layers, which shrink the representation of the detection layers by a constant factor." [?] which is computing the maximum activation of a small group of units in the detection layer (Hidden layer) this enables toleration the change in the features places and allow for the CRBM to be translation invariant. max pooling also helps speeding up the algorithm it was found that the algorithm is 10 times slower without the max pooling layer.

The CRBM model is overcomplete that the size of the input is much smaller than the network representation of it since there are K hidden groups each roughly of the size of an image, This may lead to the problem of overcomplete models which have the risk of detecting trivial features of the image and having a feature detector that represent single pixel of the input. While a common approach is to force sparse representation where a tiny fraction of the units are active given a specific stimulus, In this paper regularization was used to make the mean activation for each unit close some small constant.

The final network consists of several max-pooling CRBMs stacked one over the other and the final network energy function that is the sum of the energy functions for all of the individual layers. The training for this network is using the same greedy methods described in section 2 [4].

3.2 Experimental results for Classification and recognition

To prove the validity of the approach the research team experimented the model against the state-of-art algorithms in well known tasks.The results then reported compared to the other algorithms.

At first the team approached the Caltech-101 object classification task. The model was two layers of CRBMs the first layer was 24 groups of 10×10 pixels each, and the second layer was 100 groups of 10×10 pixels each. The CDBN gave test accuracy of $57.7 \pm 1.5\%$ when trained with 15 image per class and gave $65.4 \pm 0.5\%$ when trained with 30 image per class,while the Scale-invariant feature transform (SIFT) gave test accuracy of 54.0% when trained with 15 image per class and gave 64.6% when trained with 30 image per class, and shape-context gave $59.0 \pm 0.56\%$ with 15 image and $66.2 \pm 0.5\%$ with 30. It worth mentioning that the training for he CDBN was from natural scenes which are not related to the task and this means that CDBN acquired a general representation of the images that enabled it to recognize objects in new images with reasonable success rate compared to the state-of-art algorithms.

The CDBN was also used for classifying the MNIST dataset with a model of two layers the first layer was 40 groups of 12×12 pixels each, and the second layer was 40 groups of 6×6 pixels each, this structure allowed the learning of digits characteristics in the two layers where the first layer learned the strokes of the digits and the second layer combined those strokes and learned the bigger parts of the digits. A feature vector was then constructed from those features and using a SVM those features were classified. This method made it possible to achieve 0.82% test error which is much more the reported results of 1.2% by Hinton and Salakhutdinov descried in section 2.

Then a 3 layers CDBN was trained on unlabeled images randomly selected from four different object categories (cars, faces, motorbikes, and airplanes) and the results of the classification in the form of *AUC-PR* is reported per layer to show the ability of the CDBN to combine the features layer by layer to allow for better performance see [Table 1].

Layer	Faces	Motorbikes	Cars
First	0.39 ± 0.17	0.44 ± 0.21	0.43 ± 0.19
Second	0.86 ± 0.13	0.69 ± 0.22	0.72 ± 0.23
Third	0.95 ± 0.03	0.81 ± 0.13	0.87 ± 0.15

Table 1: Average AUC-PR for each classification class per layer

Those results shows that the Convolution deep believe network has the ability to learn underlying statistics of the input data in several levels and have a promising performance compared to the state-of-art algorithms in different AI tasks it was exploited to.

4 Conclusion

Deep learning architecture us a multilevel feature detectors where lower levels detect simple features and the higher levels detect more complex global features. There have been a lot of interest and development in the area of deep learning and it's ability to learn without a supervisor makes it reasonable to use with inf several applications that have a great amount of data and will be impractical to label all of this data.There are an increasing number of new research in the deep learning area, new algorithms are being developed for different applications. Finally deep learning is a promising learning field and can be used and gives results that are comparable to the state-of-art algorithms in most of the cases it was used in.

References

[1] Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009. Also published as a book. Now Publishers, 2009.

[2] Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1(2):119 – 130, 1988.

[3] Kunihiko Fukushima and Sei Miyake. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, 15(6):455–469, 1982.

[4] G E Hinton and R R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.

[5] G. E. Hinton and T. J. Sejnowski. Optimal perceptual inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, 1983.

[6] Geoffrey E. Hinton. Learning multiple layers of representation. *Trends in Cognitive Sciences*, 11:428–434, 2007.

[7] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006.

[8] D. H. HUBEL and T. N. WIESEL. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148:574–591, October 1959.

[9] Stphane Dufau Jonathan Grainger, Arnaud Rey. Letter perception: from pixels to pandemonium. *Trends in cognitive sciences*, 12, 2008.

[10] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry.* MIT Press, Cambridge, MA, USA, 1969.

[11] O. G. Selfridge. Pandemonium: a paradigm for learning in Mechanisation of Thought Processes. In *Proceedings of a Symposium Held at the National Physical Laboratory*, pages 513–526, London, November 1958. HMSO.

[12] P Smolensky. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1: Foundations. chapter Chapter 6: Information Processing in Dynamical Systems: Foundations of Harmony Theory, pages 194–281. MIT Press, 1986.

[13] Alan M Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.