IEIE Transactions on Smart Processing and Computing, vol. 4, no. 1, February 2015
http://dx.doi.org/10.5573/IEIESPC.2015.4.1.035

Deep Convolution Neural Networks in Computer Vision: a Review

Hyeon-Joong Yoo

Department of IT Engineering, Sangmyung University / Chonan, Choongnam-Do 330-720 Korea yoohj@smu.ac.kr

***** Corresponding Author:

Received April 5, 2014; Revised June 13, 2014; Accepted November 19, 2014; Published February 28, 2015

** Regular Paper*

** Review Paper: This paper reviews the recent progress possibly including previous works in a particular research topic, and has been accepted by the editorial board through the regular reviewing process.*

***Abstract*:** Over the past couple of years, tremendous progress has been made in applying *deep learning* (DL) techniques to computer vision. Especially, *deep convolutional neural networks* (DCNNs) have achieved state-of-the-art performance on standard recognition datasets and tasks such as *ImageNet Large-Scale Visual Recognition Challenge* (ILSVRC). Among them, GoogLeNet network which is a radically redesigned DCNN based on the Hebbian principle and scale invariance set the new state of the art for classification and detection in the ILSVRC 2014. Since there exist various deep learning techniques, this review paper is focusing on techniques directly related to DCNNs, especially those needed to understand the architecture and techniques employed in GoogLeNet network.

***Keywords*:** Deep learning, Convolutional neural network, ImageNet, Computer vision, GoogLeNet

1. Introduction

Over the last couple of years, deep learning techniques have made tremendous progress in computer vision, especially in the field of object recognition. Deep learning is a family of methods that uses deep architectures to learn high-level feature representation. (Several much lengthier definitions can be found in [1].) When a network has more than one hidden layer, it is of deep architecture. The essence of deep learning is to compute hierarchical features or representations of the observational data, where the higher-level features or factors are defined from lower-level ones.

The family of deep learning methods have been growing increasingly richer, encompassing those of neural networks, hierarchical probabilistic models, and a variety of unsupervised and supervised feature learning algorithms.

Active researchers in this area include those at University of Toronto, New York University, University of Montreal, Stanford University, Google, Baidu, Microsoft Research, Facebook, just to name a few. These researchers have demonstrated empirical successes of deep learning in diverse applications of computer vision, phonetic recognition, voice search, conversational speech recognition, speech and image feature coding, robotics, and so on.

In this paper we focus on the architecture and training methods of deep networks, specifically deep convolutional neural networks.

1.1 History of Neural Networks

Historically, the concept of deep learning originated from artificial neural network research. The history of artificial neural network is filled with individuals from many different fields, including psychologists and physicists. The early model of an artificial neuron is introduced by McCulloch and Pitts in 1943 [2]. Their work is often acknowledged as the origin of the neural network field. They showed that networks of artificial neurons could, in principle, compute any arithmetic or logical function.

In 1949, Hebb [3] proposed a mechanism for learning in biological neurons, and introduced the Hebb learning rule. The Hebb rule is often paraphrased as "Neurons that fire together wire together." If the set of input patterns used in training are mutually orthogonal, the association can be learned by a two-layer pattern associator using

Hebbian learning. However, if the set of input patterns are not mutually orthogonal, interference may occur and the network may not be able to learn associations, resulting in a low absolute capacity of the Hebb rule.

The first practical application of artificial neural network came with the invention of the perceptron network and associated learning rule by Rosenblatt in the late 1950s [4]. With the perceptron network, he demonstrated its ability to perform pattern recognition. At about the same time, Widrow and Hoff [5] introduced a new learning algorithm called the Widrow-Hoff learning rule, and used it to train adaptive linear neural networks, which were similar to Rosenblatt's perceptron.

Unfortunately, both of the networks suffered from the same inherent limitations, which were widely publicized in a book by Minsky and Papert [6]. They discovered two key issues with the computational machines that processed neural networks. The first issue was that single-layer neural networks were incapable of solving the exclusive-or (XOR) problem. The second significant issue was that computers were not sophisticated enough to effectively handle the long run time required by large neural networks. Many people, influenced by Minsky and Papert, believed that further research on neural networks was a dead end. The book caused many researchers to leave the field and nearly killed neural net research for more than a decade.

However, some important work continued during the 1970s. Kohonen [7] and Anderson [8] independently and separately developed new neural networks that could act as memories. Grossberg [9] was also very active during this period in the investigation of self-organizing networks.

During the 1980s research in neural networks increased dramatically. The impediments in the 1960s were overcome, and, in addition, important new concepts were introduced and responsible for the rebirth of neural networks. Firstly, in 1982, physicist Hopfield used statistical mechanics to explain the operation of a certain class of recurrent network, which could be used as an associative memory [10]. Another key development was the backpropagation algorithm, a generalized form of the delta rule, for training multilayer perceptron networks. Although the derivation procedure had previously been published by Werbos in [11], the most influential publication of the backpropagation algorithm was by Rumelhart and McClelland [12]. They showed that this method works for the class of semilinear activation functions (non-decreasing and differentiable). It was the answer to the criticisms Minsky and Papert had made in 1969. These new development reinvigorated the field of neural networks.

One long-term goal of machine learning research is to produce methods that are applicable to highly complex tasks, such as perception (vision, audition), reasoning, intelligent control, and other artificially intelligent behaviors. In order to progress toward this goal, algorithms that can learn highly complex functions with minimal need for prior knowledge, and with minimal human intervention must be discovered. However, shallow architectures can be very inefficient in terms of required number of computational elements and examples. However, for deep architectures, there are such backpropagation-specific properties that can occasionally be a problem as slow learning speed (the further the weights are from the output layer, the slower backpropagation learns) and overfitting.

Researchers tried using stochastic gradient descent and backpropagation to train deep networks. Unfortunately, except for a few special architectures, they didn't have much luck. The networks would learn, but very slowly, and in practice often too slowly to be useful.

Building on Rumelhart et al. [12], LeCun et al. [13] showed that stochastic gradient descent via backpropagation was effective for training convolutional neural networks (CNNs). CNNs saw commercial use with [14], but then fell out of fashion with the rise of support vector machines and other, much simpler methods such as linear classifiers.

Promising new methods like Hinton et al.'s [15] have been developed that enable learning in deep neural nets, and in 2012, Krizhevsky et al. [16] rekindled interest in CNNs by showing substantially higher image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC, http://www.image-net.org/). These techniques have enabled much deeper (and larger) networks to be trained - people now routinely train networks with many hidden layers. And, it turns out that these perform far better on many problems than shallow neural networks [16-18].

The rest of this paper is organized as follows. The next section describes in chronological order important historical events and techniques needed to understand the architecture and training methods of current state of the art deep convolutional neural network, GoogLeNet. Then the GoogLeNet network is investigated deeper in Section 3. Finally, we conclude this paper in Section 4.

2. Deep Convolution Neural Networks

Shallow architectures have been shown effective in solving simple or well-constrained problems, but their limited modeling and representational power can cause difficulties when dealing with more complicated real-world applications involving natural signals such as natural image and visual scenes. Human information processing mechanisms suggest the need of deep architectures for extracting complex structure and building internal representation form rich sensory inputs. However, training deep neural networks is hard, as backpropagated gradients quickly vanish exponentially in the number of layers. A set of techniques has been developed that enable learning in deep neural networks. People now routinely train networks with many hidden layers. And, it turns out that these perform far better on many problems than shallow neural networks. Deep neural networks have finally attracted wide-spread attention. In this section, we describe some important techniques and events necessary to understand the deep convolutional neural network called GoogLeNet. Convolutional networks are an attempt to solve the dilemma between small networks that cannot learn the training set, and large networks that seem over-parameterized.

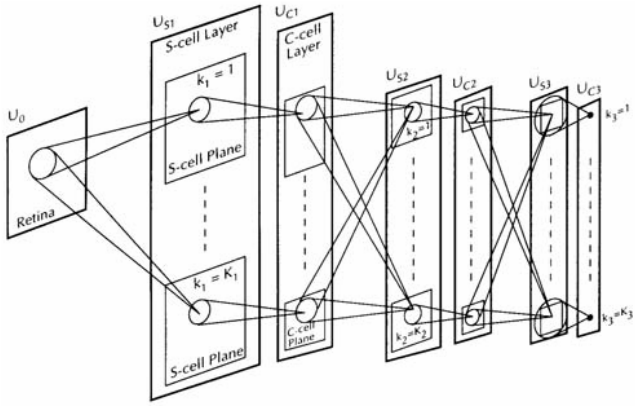

Fig. 1. Hierarchical structure of the neocognitron network (Adapted from [19]).

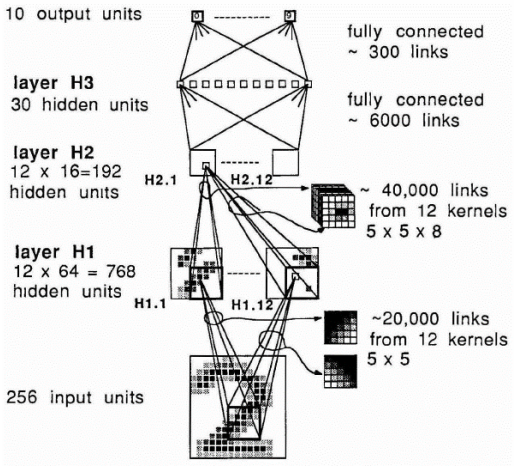

Fig. 2. Architecture of CNN in 1989 (Adapted from [13]).

2.1 Neocognitron

A hierarchical multilayered artificial neural network called neocognitron was proposed by Fukushima in 1980 [19]. Local features in the input are integrated gradually and classified in the higher layers. Fig. 1 shows its architecture. Later, in 1989, the idea of local feature integration is adapted in LeCun et al.'s convolutional neural networks [13].

The neocognitron was specifically proposed to address the problem of handwritten character recognition. It is a hierarchical network with many pairs of layers corresponding to simple (S layer) and complex (C layer) cells with a very sparse and localized pattern of connectivity between layers.

The number of planes of simple cells and of complex cells within a pair of S and C layers being the same, these planes are paired, and the complex plane cells process the outputs of the simple plane cells. The simple cells are trained so that the response of a simple cell corresponds to a specific portion of the input image. If the same part of the image occurs with some distortion, in terms of scaling or rotation, a different set of simple cells responds to it. The complex cells output to indicate that some simple cell they correspond to did fire. While simple cells respond to what is in a contiguous region in the image, complex cells respond on the basis of a larger region. As the process continues to the output layer, the C-layer component of the output layer responds, corresponding to the entire image presented in the beginning at the input layer. The neocognitron, however, lacked a supervised training algorithm.

2.2 CNN

Building on Rumelhart et al. [12], LeCun et al. [13] showed that stochastic gradient descent via backpropagation was effective for training convolutional neural networks, a class of models that extend the neocognitron. LeCun et al. presented a convolutional neural network consisting of 3 hidden layers including 2 convolutional layers, and 64,660 connections. They applied it to handwritten zip code recognition. Fig. 2 shows its architecture. Unlike previous works, the network was directly trained on a low-level representation of data that had minimal preprocessing (as opposed to elaborate feature extraction), thus demonstrating the ability of backpropagation networks to deal with large amounts of low-level information.

The first hidden layer is composed of several planes called feature maps. Distinctive features of an object can appear at various locations on the input image. It seems judicious to have all units in a plane share the same set of weights, thereby detecting the same feature at different locations. Thus the detection of a particular feature at any location on the input can be easily done using the "weight sharing" technique. Since the exact position of the feature is not important, the feature maps need not have as many units as the input. However, units do not share biases. Due to the weight sharing, the network has few free parameters. For example, though layer H1 has 19,968 connections, it has only 1,068 free parameters (768 biases plus 25 times 12 feature kernels).

The connection scheme between H1 and H2 is slightly more complicated: Each unit in H2 combines local information coming from 8 of the 12 different feature maps in H1. Its receptive field is composed of eight 5 by 5 neighborhoods centered on units that are at identical positions within each of the eight maps. Once again, all units in a given map are constrained to have identical weight vectors. Layer H3 has 30 units, and is fully connected to H2. The output layer is also fully connected to H3. In summary, the network has 1,256 units, 64,660 connections, and 9,760 independent parameters.

The target values for the output units were chosen within the quasilinear range of the sigmoid. This prevents the weights from growing indefinitely and prevents the output units from operating in the flat spot of the sigmoid.

During each learning experiment, the patterns were repeatedly presented in a constant order. The weights were updated according to the stochastic gradient or "on-line" procedure. From empirical study (supported by theoretical arguments), the stochastic gradient was found to converge much faster than the true gradient, especially on large, redundant data bases. It also finds solutions that are more robust.

2.3 LeNet-5

Convolutional networks combine three architectural ideas to ensure some degree of shift, scale, and distortion invariance: local receptive fields, shared weights (or weight replication), and spatial or temporal sub-sampling. Fig. 3 shows the architecture of LeNet-5, a convolutional neural network proposed by LeCun et al. in 1998 [14], which was used commercially for reading bank checks.

A convolutional layer is composed of several feature maps (with different weight vectors), so that multiple features can be extracted at each location. The receptive fields of contiguous units in a feature map are centered on correspondingly contiguous units in the previous layer. Therefore receptive fields of neighboring units overlap. All the units in a feature map share the same set of weights and the same bias so they detect the same feature at all possible locations on the input.

A sequential implementation of a feature map would scan the input image with a single unit that has a local receptive field, and store the states of this unit at corresponding locations in the feature map. This operation is equivalent to a convolution, followed by an additive bias and squashing function, hence the name convolutional network.

An interesting property of convolutional layers is that if the input image is shifted, the feature map output will be shifted by the same amount, but will be left unchanged otherwise. This property is at the basis of the robustness of convolutional networks to shifts and distortions of the input. A large degree of invariance to geometric transformations of the input can be achieved with this progressive reduction of spatial resolution compensated by a progressive increase of the richness of the representation (the number of feature maps).

The convolution/subsampling combination, inspired by Hubel and Wiesel's notions of "simple" and "complex" cells [20], was implemented in Fukushima's Neocognitron [19], though no globally supervised learning procedure such as backpropagation was available then.

Starting with LeNet-5 [14], convolutional neural networks (CNN) have typically had a standard structure – stacked convolutional layers (optionally followed by contrast normalization and maxpooling) are followed by one or more fully-connected layers. All the weights are learned with backpropagation. Variants of this basic design are prevalent in the image classification literature and have yielded the best results to-date on MNIST, CIFAR and most notably on the ImageNet classification challenge [16, 18, 21]. For larger datasets such as Imagenet, the recent trend has been to increase the number of layers [22] and layer size [23, 24], while using dropout [25] to address the problem of overfitting.

2.4 Multi-Scale ConvNet

Sermanet et al. [26] modified the traditional ConvNet (convolutional networks) architecture by feeding 1st stage features in addition to 2nd stage features to the classifier as shown in Fig. 4. Each stage is composed of a (convolutional) filter bank layer, a non-linear transform layer, and a spatial feature pooling layer. ConvNets are

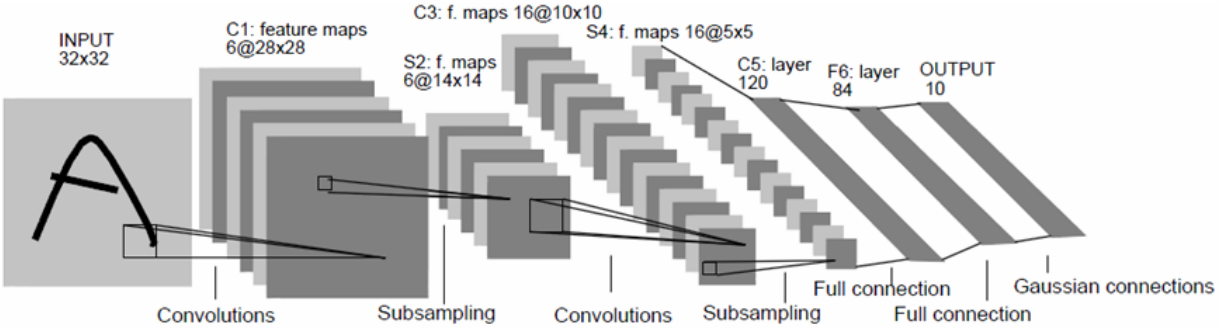

Fig. 3. Architecture of LeNet-5, a Convolutional Neural Network, here for characters recognition. Each plane is a feature map, i.e., a set of units whose weights are constrained to be identical (Adapted from [14]).

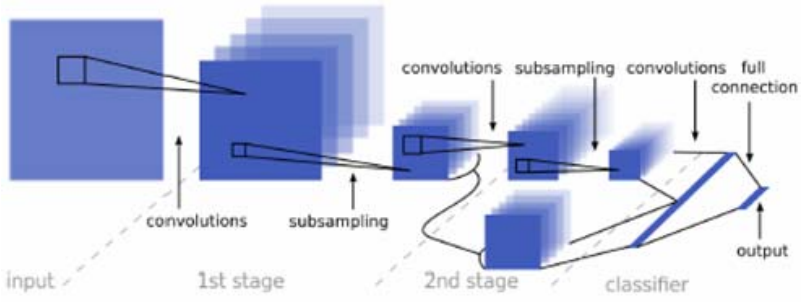

Fig. 4. A 2-stage ConvNet architecture. The input is processed in a feedforward manner through two stage of convolutions and subsampling, and finally classified with a linear classifier. The output of the 1st stage is also fed directly to the classifier as higher-resolution features. (Adapted from [26]).

generally composed of one to three stages, capped by a classifier composed of one or two additional layers.

Contrary to Fan et al.'s [27], they used the output of the first stage after pooling/subsampling rather than before. Additionally, applying a second subsampling stage on the branched output yielded higher accuracies than with just one. Feeding the outputs of all the stages to the classifier allows the classifier to use, not just high-level features, which tend to be global, invariant, but with little precise details, but also pooled low-level features, which tend to be more local, less invariant, and more accurately encode local motifs with more precise details. The motivation for combining representation from multiple stages in the classifier is to provide different scales of receptive fields to the classifier.

When applied to the task of traffic sign classification as part of the GTSRB competition, experiments produced a new record of 99.17%, above the human performance of 98.81%.

2.4 ReLU and Dropout

Sparsity was first introduced in computational neuroscience in the context of sparse coding in the visual system [28]. The neuroscience literature [29, 30] indicates that cortical neurons are rarely in their maximum saturation regime, and suggests that their activation function can be approximated by a rectifier.

In 2011, Glorot et al. [31] showed that using a rectifying non-linearity gives rise to real zeros of activations and thus truly sparse representations. They showed that rectifying neurons are an even better model of biological neurons and yield equal or better performance than hyperbolic tangent networks in spite of the hard non-linearity and non-differentiability at zero, creating sparse representations with true zeros, which seem remarkably suitable for naturally sparse data. Their experiments on image and text data indicated that training proceeds better when the artificial neurons are either off or operating mostly in a linear regime. Rectifying activation allowed deep networks to achieve their best performance without unsupervised pre-training on purely supervised tasks with large labeled datasets.

In 2012, Krizhevsky et al. [16] rekindled interest in CNNs by showing substantially higher image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) with their model achieving an error rate of 16.4%, compared to the 2nd place result of 26.1%. They used purely supervised learning, suggesting that classical backpropagation does well even without unsupervised pre-training. Their success resulted from training a large, deep CNN, shown in Fig. 5, on 1.2 million labeled images, together with a few twists on LeCun's CNN, i.e., max(x, 0) rectifying non-linearities and "dropout" regularization.

In terms of training time with gradient descent, the saturating nonlinearities $f(x) = \tanh(x)$ or $f(x) = (1 + e^{-x})^{-1}$ are much slower than the non-saturating nonlinearity $f(x) = \max(x, 0)$. Following Nair and Hinton [32], they refer to neurons with this nonlinearity as Rectified Linear Units (ReLUs). Deep convolutional neural networks with ReLUs train several times faster than their equivalents with tanh units. Faster learning has a great influence on the performance of large models trained on large datasets.

When a large feedforward neural network is trained on a small training set, it typically performs poorly on held-out test data due to its high capacity. To prevent this "overfitting", they employed a regularization approach called "dropout" that stochastically sets half the activations within a hidden layer to zero for each training sample during training. They used dropout in the first two fully-connected layers (See Fig. 6). Thereby a hidden unit cannot rely on other hidden units being present. This prevents complex co-adaptations on the training data. It has been shown to deliver significant gains in performance across a wide range of problems.

Another way to view the dropout procedure is as a very efficient way of performing model averaging with neural networks. A good way to reduce the error on the test set is to average the predictions produced by a very large number of different networks. The standard way to do this is to train many separate networks and then to apply each of these networks to the test data, but this is computationally expensive during both training and testing. Random dropout makes it possible to train a huge number of different networks in a reasonable time. There is almost certainly a different network for each presentation of each training case but all of these networks share the same weights for the hidden units that are present.

Dropout does not seem to have the same benefits for convolutional layers, which are common in many networks designed for vision tasks. Dropout roughly doubles the

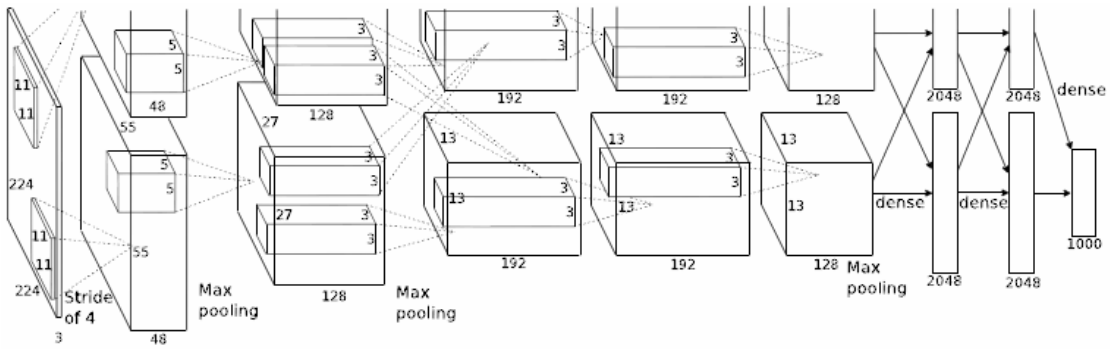

Fig. 5. Architecture of Krizhevsky et al.'s DCNN (Adapted from [16]).

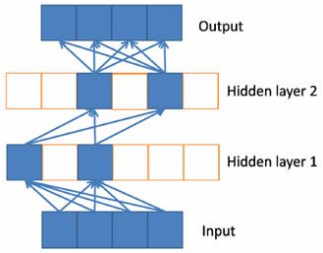

Fig. 6. Sparse propagation of activations and gradients (Adapted from [31]).

number of iterations required to converge. Without dropout, their network exhibited substantial overfitting.

They achieved record-breaking results on a highly challenging dataset using purely supervised learning. They found that the depth (five convolutional and three fully-connected layers) really was important for achieving their results by observing that removing convolutional layer (each of which contains no more than 1% of the model's parameters) resulted in inferior performance (a loss of about 2% for the top-1 performance of the network).

2.5 ConvNet Visualization

Though large ConvNets (convolutional networks) have recently demonstrated impressive classification performance, there is no clear understanding of why they perform so well, or how they might be improved, which is deeply unsatisfactory from a scientific standpoint. Without clear understanding of how and why they work, the development of better models is reduced to trial-and-error.

In 2013, Zeiler et al. addressed both issues [24]. They introduced a visualization technique that gives insight into the function of intermediate feature layers and the operation of the classifier. The visualization technique uses a multi-layered deconvolutional network (deConvNet), as proposed by Zeiler et al. [33], to project the feature activations back to the input pixel space. DeConvNets are not used in any learning capacity, just as a probe of an already trained ConvNet. They also performed an ablation study to discover the performance contribution from different model layers. This enabled them to find model architectures that outperformed Krizhevsky et al. [16] on the ImageNet classification benchmark.

2.6 Sparsely Connected Architecture

Bigger size typically means two drawbacks: a larger number of parameters, which makes the enlarged network more prone to overfitting, especially if the number of labeled examples in the training set is limited; and the dramatically increased use of computational resources, which is always finite, in a deep vision network where convolutional layers are chained.

The fundamental way of solving both issues would be by ultimately moving from fully connected to sparsely connected architectures, even inside the convolutions. Besides mimicking biological systems, this would also have the advantage of firmer theoretical underpinnings due to the work of Arora et al. [34]: If the probability distribution of the dataset is representable by a large, very sparse deep neural network, then the optimal network structure can be learned layerwise by analyzing the correlation statistics of the activations of the last layer and clustering neurons with highly correlated outputs. Szegedy et al. [18] took inspiration and guidance from this theoretical work, and their GoogLeNet DCNN significantly outperformed the ILSVRC 2014 classification and detection challenges.

3. A Deeper and Wider CNN

In this section the important factors in architecture and training methods of GoogLeNet's classification submission and detection submission are described. Most of the content in this section is from Szegedy et al.'s [18].

Szegedy et al.'s GoogLeNet won the classification and object recognition challenges in the ILSVRC 2014 by setting the new state of the art. GoogLeNet, a radically redesigned DCNN, used a new variant of convolutional neural network called "Inception" for classification, and the R-CNN [17] for detection. GoogLeNet submission to ILSVRC 2014 actually uses 12x fewer parameters than the winning architecture of Krizhevsky et al. [16] from two years ago, while being significantly more accurate. Compared to the 2013 result, the detection accuracy has almost doubled from 22.6% to 43.9%. The ILSVRC detection task is to produce bounding boxes around objects in images among 200 possible classes.

GoogLeNet uses a significantly deeper and wider convolutional neural network architecture than traditional DCNNs at the cost of a modest growth in evaluation time. Fig. 7 shows a schematic view of GoogLeNet network which includes 9 repeated layers of so-called Inception module. Fig. 8 shows Inception module with dimension reduction. Inspired by a neuroscience model of primate visual cortex, the Inception model uses a series of trainable filters of different sizes in order to handle multiple scales. The main idea of the Inception architecture is based on finding out how an optimal local sparse structure in a convolutional vision network can be approximated and covered by readily available dense components. An alternative parallel pooling path in the module is added in each stage since pooling operations have been essential for the success in current state of the art convolutional networks.

Since even a modest number of 5x5 convolutions can be prohibitively expensive on top of a convolutional layer with a large number of filters, 1x1 convolutions are placed before the expensive 3x3 and 5x5 convolutions as dimension reduction module. This allows for not just increasing the depth, but also the width of the networks without significant performance penalty. The resultant architecture leads to over 10x reduction in the number of parameters compared to most state of the art vision networks, which reduces overfitting during training and allows the system to perform inference with low memory footprint.

All the convolutions, including those inside the Inception modules, use rectified linear activation. However,

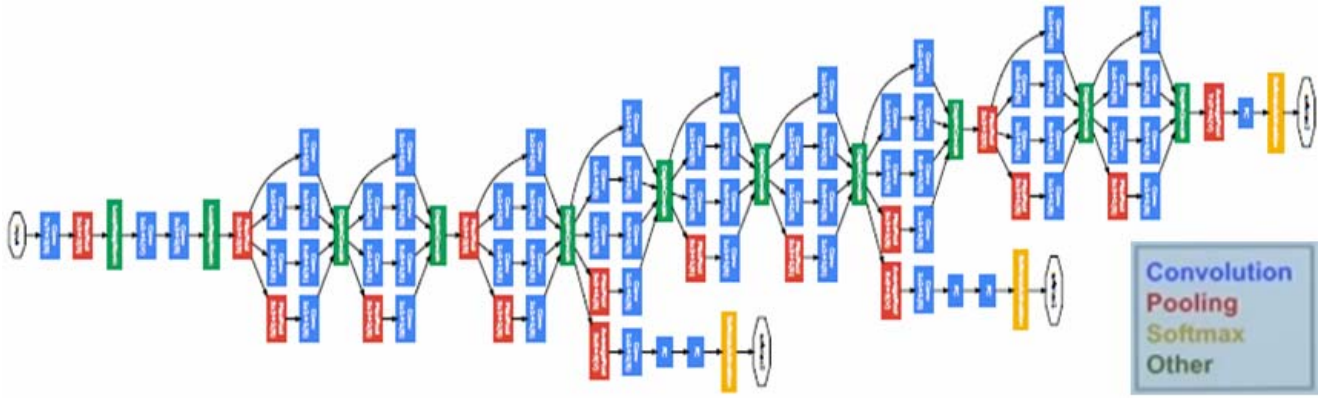

Fig. 7. A schematic view of GoogLeNet network (Adapted from [18]).

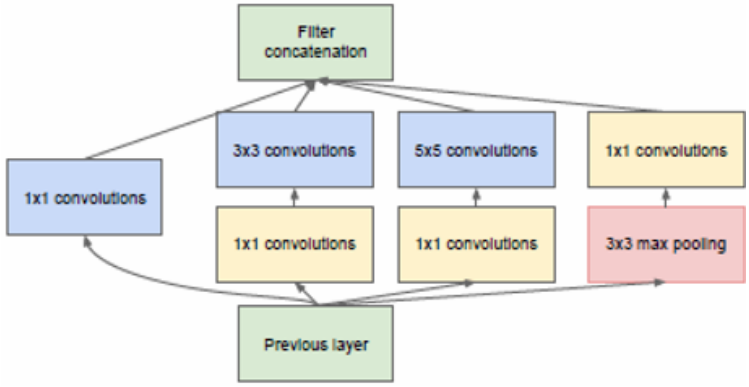

Fig. 8. Inception module with dimension reduction (Adapted from [18]).

given the relatively large depth of the network, the ability to propagate gradients back through all the layers in an effective manner was a concern. To solve this issue, inspired by [34] in which a layer-by-layer construction is suggested, they add auxiliary classifiers connected to the layers in the middle of the network during training, expecting to encourage discrimination in the lower stages in the classifier, increase the gradient signal that gets propagated back, and provide additional regularization.

However, the biggest gains in object-detection have come from the synergy of deep architectures and classical computer vision, like the R-CNN algorithm by Girshick et al. [17]. For classification challenge entry, several ideas from the work of [35] were incorporated and evaluated, specifically as they relate to image sampling during training and evaluation. Their approach yielded solid evidence that moving to sparser architectures is feasible and useful idea in general.

4. Conclusions

Deep learning techniques have made tremendous progress in computer vision, and are significantly outperforming other techniques, and even humans in certain limited recognition tests. Most of this progress is not just the result of more powerful hardware, larger datasets and bigger models, but mainly a consequence of new ideas, algorithms and improved network architectures.

Although the tremendous progress of this new technology is very impressive and encouraging, it is still difficult to predict the future success of deep neural networks. Remembering that we still know very little of the brain mechanism and have many orders of magnitude to go in order to match the infero-temporal pathway of the primate visual system, the most important advances in deep neural networks certainly lie in the future.

Acknowledgment

This material is based upon work supported by Sangmyung University, Korea. Comments by Prof. Dongsun Park at Chonbuk National University, Korea, greatly helped to improve an earlier version of this manuscript.

References

[1] L. Deng and D. Yu, *Deep Learning Methods and Applications*, now Publishers Inc., 2014.

[2] W. McCulloch and W. Pitts, “A Logical calculus of the ideas immanent in nervous activity”, *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133, 1943.

[3] Donald O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*, Wiley, June 1949.

[4] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychological Review*, vol. 65, pp. 386-408, 1958. Article (CrossRef Link).
[5] B. Widrow and M.E. Hoff, Jr., “Adaptive Switching Circuits,” *IRE WESCON Convention Record*, Part 4, pp. 96-104, August 1960.
[6] M. Minsky and Seymour Papert, *Percetrons*, Cambridge, MIT Press, 1969.
[7] T. Kohonen, "Correlation Matrix Memories," *IEEE Transactions on Computers*, vol. 21, pp. 353-359, April 1972. Article (CrossRef Link)
[8] James A. Anderson, “A simple neural network generating an interactive memory,” *Mathematical Biosciences*, vol. 14, pp. 197–220, 1972. Article (CrossRef Link)
[9] S. Grossberg, "Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors," *Biological Cybernetics*, vol. 23, pp. 121-134, 1976. Article (CrossRef Link)
[10] J. J. Hopfield, “Neural Networks and Physical Systems with Emergent Collective Computational Abilities,” *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, vol. 79, pp. 2554–2558, 1982. Article (CrossRef Link)
[11] P. J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, PhD thesis, Harvard University, 1974.
[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, D. E. Rumelhart and J. L McClelland, eds, vol. I, pp 318-362, MIT, Cambridge, 1986.
[13] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural Computation*, vol. 1, pp. 541–551, 1989. Article (CrossRef Link)
[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 1998. Article (CrossRef Link)
[15] Geoffrey E. Hinton and Simon Osindero, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, pp. 1527-1554, 2006. Article (CrossRef Link)
[16] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems* 25, pp. 1106–1114, 2012.
[17] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *CVPR 2014, IEEE Conference on*, 2014.
[18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, “Going Deeper with Convolutions,” *CoRR*, 2014.
[19] K. Fukushima, “Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position,” *Biol. Cybernetics*, vol. 36, pp. 193-202, 1980. Article (CrossRef Link)
[20] D. H. Hubel and T. N. Wiesel, “Receptive fields, binocular interaction, and functional architecture in the cats visual cortex,” *Journal of Physiology* (London), vol. 160, pp. 106-154, 1962. Article (CrossRef Link)
[21] M.D. Zeiler, R. Fergus, “Visualizing and Understanding Convolutional Networks,” *ECCV 2014* (Honorable Mention for Best Paper Award), Arxiv 1311.2901, Nov 28, 2013.
[22] Min Lin, Qiang Chen, and Shuicheng Yan, “Network in network,” *CoRR*, abs/1312.4400, 2013.
[23] Pierre Sermanet, David Eigen, Xiang Zhang, Micha¨el Mathieu, Rob Fergus, and Yann LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” *CoRR*, abs/1312.6229, 2013.
[24] Matthew D. Zeiler, *Hierarchical Convolutional Deep Learning in Computer Vision*, Ph.D. Thesis, Nov. 8, 2013.
[25] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *CoRR*, abs/1207.0580, 2012.
[26] Pierre Sermanet and Yann LeCun, “Traffic Sign Recognition with Multi-Scale Convolutional Networks,” *IJCNN*, 2011.
[27] J. Fan, W. Xu, Y. Wu, and Y. Gong, “Human tracking using convolutional neural networks,” *Neural Networks, IEEE Transactions on*, vol. 21, pp. 1610 –1623, 2010.
[28] B. A. Olshausen and D. J. Field, Sparse coding with an overcomplete basis set: a strategy employed by V1?, *Vision Research*, vol. 37, pp. 3311-3325, 1997.
[29] P. C. Bush and T. J. Sejnowski, *The cortical neuron*, Oxford University Press, 1995.
[30] R. Douglas and K. Martin, “Recurrent excitation in neocortical circuits,” *Science*, vol. 269, pp. 981-985, 1995.
[31] Xavier Glorot, Antoine Bordes, and Yoshua Bengio, “Deep Sparse Rectifier Neural Networks”, *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS) 2011*, Fort Lauderdale, FL, USA, vol. 15 of JMLR:W&CP 15, pp. 315-323, 2011.
[32] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” *Proc. 27th International Conference on Machine Learning*, 2010.
[33] M. Zeiler, G. Taylor, and R. Fergus, “Adaptive deconvolutional networks for mid and high level feature learning,” *ICCV*, 2011.
[34] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma, “Provable bounds for learning some deep representations,” *CoRR*, abs/1310.6343, 2013.
[35] Andrew G. Howard, “Some improvements on deep convolutional neural network based image classification,” *CoRR*, abs/1312.5402, 2013.

Hyeon-Joong Yoo is Professor of IT Engineering Department at Sangmyung University, Chonan, Korea. He received his B.S. degree in Electronics Engineering from Sogang University, Korea, in 1982 and his M.S. and Ph.D. in Electrical and Computer Engineering from the University of Missouri in 1991 and 1996, respectively. His current research interests include computer vision, pattern recognition, artificial neural network, and image processing.