Data-driven Sparse Connections for Deep Convolutional Neural Network

Liu Ying
School of Computer and Control
University of Chinese Academy of Sciences
Beijing, China 100000
Email: yingliu@ucas.ac.cn

Xiang Chao
School of Computer and Control
University of Chinese Academy of Sciences
Beijing, China 100000
Email: xiangchao215@mails.ucas.ac.cn

***Abstract*—Convolutional Neural Networks(CNNs) have achieved great success in many computer vision tasks, especially in image recognition. However, as neural networks grow deeper and deeper, to some extend, we've found them becoming difficult to train, and requiring samples in large scale dramatically, even with the help of Dropout and Drop-connect methods, which do improve the accuracy a bit but burdens the training process as a sacrifice. To overcome this, we proposed a novel method to generate dynamic graphs of computation for varied inputs. As our computation graphs are determined by input samples and proved to be pretty sparse, we call them Data-driven Sparse Connections(DSCs). We've applied our DSCs to a few popular image recognition tasks, and it is shown deep CNNs with DSCs win over the state-of-the-art on many tasks, such as MNIST, FICAR-10, and FICAR-100, to name a few.**

I. Introduction

Deep convolutional neural networks (CNNs) is firstly realized by Fukusima [1] with max-pooling layers [2] trained by backprop [3] on GPUs [4] have become the state-of-the-art in object recognition [5]–[8], segmentation/detection [9], [10], and scene parsing [11], [12] (for an extensive review see [13]). These architectures consist of many stacked feedforward layers, mimicking the bottom-up path of the human visual cortex, where each layer learns progressively more abstract representations of the input data. Low-level stages tend to learn biologically plausible feature detectors, such as Gabor filters [14]. Detectors in higher layers learn to respond to concrete visual objects or their parts, e.g., [15]. Once trained, the CNN never changes its weights or filters during evaluation.

II. Nonlinear Connections with Bounded Parameters

Inspired by biological neuron synapses [16]–[18], we propose a new computation model for connections between neurons. In connection models known so far, the most popular and probably the most useful one [19] is :

$$y = \sigma\left(\sum_{i=1}^{N} w_i x_i + b\right)$$

While, the activation function $\sigma(\cdot)$ can be the sigmoid function $sigmoid(x) = \frac{1}{1+e^{-x}}$, or $tanh(x)$, or ReLU function $ReLU(x) = max(0, x)$ [20]. In addition, a different form of activation function, $maxout(x) = max_{i=1}^{N}(w_i x_i)$, is being used in CNNs recently [7]. However, these activation functions are all neuron-based, and simply suppose that each connection acts as simple as a linear function, $o(x_i) = w_i x_i$, in which, w_i is a float number without even a bound. While in biological neural system, synapses, mimicked by artificial connections, are not really behaving linearly. In synapses, signals are transformed from electrical form to chemical one, and transferred by some proteins to a connected neuron, then again, transformed back to electrical form to inhibit or activate the connected neuron [21], [22]. In this process, signals can be amplified or reduced, but not in a linear way. Futhermore, signal strengths have lower and upper bounds which differs from the denotation of connections in CNNs. Thus we propose a new model for artificial neurons and their connections. Suppose, there're 3 neurons, $\mathcal{N}_a$,$\mathcal{N}_b$, $\mathcal{N}_c$, and 2 connections, $\mathcal{C}_{a\to c}$, $\mathcal{C}_{b\to c}$, the graph is shown in figure (1).

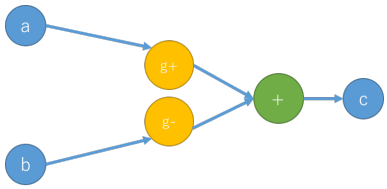

Fig. 1. Neurons and Connections

As synapses have two kinds: excitatory and inhibitory ones [23], [24], we propose two categories of connections the same way. For excitatory connections, as $\mathcal{C}_{a\to c}$ shown in figure (1), their signal transferring functions are described as equation (1), in which, $\alpha \in (0, 1)$ is guaranteed. For excitatory connections,

$$g^{+}(x) = min\left(max\left(0, \frac{x-\alpha}{1-\alpha}\right), 1\right) \tag{1}$$

For inhibitory connections,

$$g^{-}(x) = -g^{+}(x) \tag{2}$$

Activation function can be any other type that constrains the output between 0 and 1, although we set activation function as the same in equation (1) as in our model.

$$\sigma(x) = g^{+}(x) \tag{3}$$

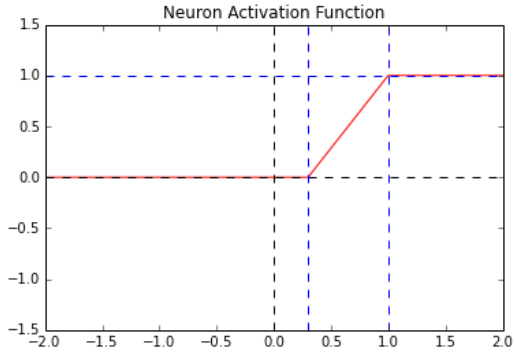

Fig. 2. Activation Function of Neuron with $\alpha = 0.3$

The α is the only parameter, and it could be varied for different neurons, with the same bounds of that in signal transferring equations. The activation function of a neuron may look like figure (2).

For a M-layer neural network, The forward-computaton equation for the kth neuron on $(p+1)$th$(p = 1, 2, \cdots, M)$ layer, $\mathcal{N}_{p_k}$, with the input layer denoted as the first layer, is

$$o_{p+1_k} = \sigma(\sum_{i=1}^{N_p} g_{i\to k}(o_{p_i})) \tag{4}$$

In which, $g_{i\to k}$ is the signal transferring function from ith neuron in pth layer to kth neuron in $(p+1)$th layer, and o_{p_i} is the activation value of the ith neuron on pth layer.

For network in figure (1), we let o_a, o_b and o_c denote the activation strengths of neuron $\mathcal{N}_a$, $\mathcal{N}_b$ and $\mathcal{N}_c$ respectively, and $\alpha_{a\to c}$, $\alpha_{b\to c}$ as the parameters of connection $\mathcal{C}_{a\to c}$, $\mathcal{C}_{b\to c}$ similarly. Then we will have:

$$o_c = \sigma(g^+(o_a)|_{\alpha=\alpha_{a\to c}} + g^-(o_b)|_{\alpha=\alpha_{b\to c}}) \tag{5}$$

While, since the equations for connections and activation function of neurons in our model are not really differentiable, the famous backprop algorithm cannot be applied in the training section. Thus, we propose a training algorithm that deals with non-differentiable situation like this.

Firstly, let's introduce two model parameters: ξ and η, they together determine the learning rate of our model. In addition, $\xi \in (0, 1)$ and $\eta \in (0, 1)$. Let $x = \sum_{i=1}^{N_p} g_{i\to k}(o_{p_i})$, $y = \sigma(x)$, and the given feedback error for current neuron $\mathcal{N}_{p_k}$ is Δy. Similarly, the feedback error for x is Δx. In case parameter α goes to 1 and fails our equations, we propose a upper-bound α_{max}, which is very close to 1 but less than 1. Thus, the parameter updatng rules are,

$$\Delta x = \eta\left((1-\alpha)(y+\Delta y) + \alpha - x\right) \tag{6}$$

$$\hat{\alpha} = \frac{x - y - \Delta y}{1 - y - \Delta y} \tag{7}$$

$$\Delta\alpha = \xi \cdot \begin{cases} \left(max\left(0, \hat{\alpha}\right) - \alpha\right), & \Delta y > 0 \\ \left(min\left(\alpha_{max}, \hat{\alpha}\right) - \alpha\right), & \Delta y < 0 \end{cases} \tag{8}$$

Let $\Delta x_{p+1_{i\to k}}$ be the correction for connection from neuron $\mathcal{N}_{p_i}$ to $\mathcal{N}_{p+1_k}$, according to the definition of Δx in equation (6),

$$\Delta x_{p+1_k} = \eta((1-\alpha_{p+1_k})(o_{p+1_k} + \Delta o_{p+1_k}) + \alpha_{p+1_k} - x_{p+1_k}) \tag{9}$$

and,

$$\Delta x_{p+1_k} = \sum_{i=1}^{N_p} \Delta x_{p+1_{i\to k}} \tag{10}$$

We assume that the correction strength for each connection is only related to the activation value of the neuron which emits the connection. Thus we will get,

$$\Delta x_{p+1_{i\to k}} = \frac{o_{p_i}}{\sum_{j=1}^{N_p} o_{p_j}} \cdot \Delta x_{p+1_k} \tag{11}$$

Let $\alpha_{p_{i\to k}}$ be the parameter of the connection from the ith neuron in pth layer to the kth neuron in $(p+1)$th layer. Using equation (8), we can update the parameters of connections. For excitatory connections,

$$\hat{\alpha}_{p_{i\to k}} = \frac{o_{p_i} - g^+_{p_{i\to k}}(o_{p_i}) - \Delta x_{p+1_{i\to k}}}{1 - g^+_{p_{i\to k}}(o_{p_i}) - \Delta x_{p+1_{i\to k}}} \tag{12}$$

$$\Delta\alpha_{p_{i\to k}} = \xi \cdot \begin{cases} max\left(0, \hat{\alpha}_{p_{i\to k}}\right) - \alpha_{p_{i\to k}}, \\ \quad \Delta x_{p+1_{i\to k}} > 0 \\ min\left(\alpha_{max}, \hat{\alpha}_{p_{i\to k}}\right) - \alpha_{p_{i\to k}}, \\ \quad \Delta x_{p+1_{i\to k}} < 0 \end{cases} \tag{13}$$

For inhibitory connections, we only need to replace $\Delta x_{p+1_{i\to k}}$ with $-\Delta x_{p+1_{i\to k}}$, and replace $g^+_{p_{i\to k}}(o_{p_i})$ with $-g^-_{p_{i\to k}}(o_{p_i})$ in equation (12) and (13),

$$\hat{\alpha}_{p_{i\to k}} = \frac{o_{p_i} + g^-_{p_{i\to k}}(o_{p_i}) + \Delta x_{p+1_{i\to k}}}{1 + g^-_{p_{i\to k}}(o_{p_i}) + \Delta x_{p+1_{i\to k}}} \tag{14}$$

$$\Delta\alpha_{p_{i\to k}} = \xi \cdot \begin{cases} min\left(\alpha_{max}, \hat{\alpha}_{p_{i\to k}}\right) - \alpha_{p_{i\to k}}, \\ \quad \Delta x_{p+1_{i\to k}} > 0 \\ max\left(0, \hat{\alpha}_{p_{i\to k}}\right) - \alpha_{p_{i\to k}}, \\ \quad \Delta x_{p+1_{i\to k}} < 0 \end{cases} \tag{15}$$

Biologically, A synapse can switch from excitatory to inhibitory [25]. In our model, we propose a training strategy which allows connections switch between excitatory and inhibitory during training process.

We propose a valve $\tau \in (0, \alpha_{max})$, which is very close to 0 but greater than 0, if $g_{p_{i\to k}} = g^+$, $\Delta x_{p+1_{i\to k}} < 0$ and $\alpha_{p_{i\to k}} + \Delta\alpha_{p_{i\to k}} - \alpha_{max} < \tau$, then for connection $\mathcal{C}_{p_{i\to k}}$, let $g_{p_{i\to k}} = g^-$, this update will switch connections from excitatory to inhibitory. Similarly, if $g_{p_{i\to k}} = g^-$, $\Delta x_{p+1_{i\to k}} > 0$ and $\alpha_{p_{i\to k}} + \Delta\alpha_{p_{i\to k}} - \alpha_{max} < \tau$, then for connection $\mathcal{C}_{p_{i\to k}}$, let $g_{p_{i\to k}} = g^+$, this update will switch connections from inhibitory to excitatory.

So far we've derivated the equations for updating parameters of neurons in $(p+1)$th layer and the parameters of connections from pth layer to $(p+1)$th layer. With equation (9),(11) and (6), we calculate the error of the output in the pth layer, Let $\Delta o_{p_{i\to k}}$ be the correction for $\mathcal{N}_{p_i}$ contributed by connection $\mathcal{C}_{p_{i\to k}}$, for excitatory connections,

$$\Delta o^{+}_{p_{i\to k}} = \eta((1-\alpha_{p_{i\to k}})(g^{+}_{p_{i\to k}}(o_{p_i}) + \Delta x_{p+1_{i\to k}}) + \alpha_{p_{i\to k}} - o_{p_{i\to k}}) \quad (16)$$

For inhibitory connections,

$$\Delta o^{-}_{p_{i\to k}} = -\Delta o^{+}_{p_{i\to k}} \quad (17)$$

Then the totol correction for the output of neuron $\mathcal{N}_{p_i}$ is,

$$\Delta o_{p_i} = \sum_{k=1}^{N_{p+1}} \Delta o_{p_{i\to k}} \quad (18)$$

III. Competition Between Filters: Filter Alienation within Same Layer

For filters in the same layer, to reduce replication and extract features from an image as many as possible, filter alienation is essential. Thus we propose a pretraining algorithm for filter initialization. Let $\mathcal{F}_{p_k}$ denote the kth filter in layer of depth p, and o_{p_k} be its activation value. Firstly, we initiliazed the pth layer with N_p filters, for which we set the connection parameters $\left\langle \alpha_{p_{k_1}}, \alpha_{p_{k_2}}, \cdots, \alpha_{p_{k_{N_p}}} \right\rangle$ randomized as strictly different value.

IV. Competition Between Receptive Fields: Highlighting Local Features

some text here...

V. Overall Neural Network Model

some text here...

VI. Image Compression based on Color Clustering

Using basic pixel color filters

$$\{\delta_i(r,g,b) = \vec{w_i} \cdot (r,g,b) | i \in [1, M]\}$$

M is the number of color filters for images.

VII. Train Filters Layer By Layer

Object Detection Filters

$$\{\xi_i(\mathcal{F}_k) | i \in [1, N_k]\}$$

$\mathcal{F}_k$ is the kth receptive field on last CNN pooling layer. N_k is the number of filters for Object Detecton Filters, which refers to layers from the second in CNNs.

Due to huge computation duplicate, applying sampling methods: Randomly select an image from database, and randomly select a field, with coordinates as (x, y) while keeping width and height as constant. The width and height is determined by filters. Use this piece as input to train fileters layer by layer unsupervisedly.

VIII. Data-driven sparse Connections

As showed in the previous chapter, we train filters layer by layer, Generally, if we are now at pth$(p \geq 2)$ layers, we have trained all qth$(q < p)$ layers.

Inspired by biological brain, we apply Lateral Depression model to the unsupervised training.

Given a coefficient β as minimum gap between activation of nodes. If there exists one node $\mathcal{N}_{s_0}$ ($\mathcal{N}_{s_i}$ is the ith node in sth layer), whose activation for current input is higher than its neighbors $\mathcal{U}_{s_0}$ (the neighbors, determined by some neighbor definition) nodes by at least β.

$$\mathcal{N}_{s_i} = \begin{cases} 1, & \text{if } \sigma\left(\vec{w_{s_i}} \cdot \vec{x}\right) - \sigma\left(\vec{w_{s_j}} \cdot \vec{x}\right) \geq \beta (\text{for any } j \in \mathcal{U}_{s_i}) \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

IX. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] K. Fukushima, "Neural network model for a mechanism of pattern recognition unaffected by shift in position- neocognitron," *ELECTRON. & COMMUN. JAPAN*, vol. 62, no. 10, pp. 11–18, 1979.

[2] J. Weng, N. Ahuja, and T. S. Huang, "Cresceptron: a self-organizing neural network which grows adaptively," in *Neural Networks, 1992. IJCNN., International Joint Conference on*, vol. 1. IEEE, 1992, pp. 576–581.

[3] 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, no. 4, pp. 541–551, 1989.

[4] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 1. Barcelona, Spain, 2011, p. 1237.

[5] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3642–3649.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[7] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio, "Maxout networks." *ICML (3)*, vol. 28, pp. 1319–1327, 2013.

[8] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[9] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2013, pp. 411–418.

[10] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in neural information processing systems*, 2012, pp. 2843–2851.

[11] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[12] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3626–3633.

[13] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[14] D. Gabor, "Theory of communication. part 1: The analysis of information," *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, 1946.
[15] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.
[16] W. S. Hall, *A Textbook of physiology*. Lea, 1905.
[17] W. Bayliss, "On reciprocal innervation in vaso-motor reflexes and the action of strychnine and of chloroform thereon," *Proceedings of the Royal Society of London. Series B, Containing Papers of a Biological Character*, vol. 80, no. 541, pp. 339–375, 1908.
[18] R. Gerard, "The interaction of neurones," *Ohio J. Sci*, vol. 41, pp. 160–172, 1941.
[19] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.
[20] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.
[21] E. G. Gray, "Axo-somatic and axo-dendritic synapses of the cerebral cortex: an electron microscope study," *Journal of anatomy*, vol. 93, no. Pt 4, p. 420, 1959.
[22] C. D. Harvey and K. Svoboda, "Locally dynamic synaptic learning rules in pyramidal neuron dendrites," *Nature*, vol. 450, no. 7173, pp. 1195–1200, 2007.
[23] K. Uchizono, "Characteristics of excitatory and inhibitory synapses in the central nervous system of the cat," *Nature*, vol. 207, no. 4997, pp. 642–643, 1965.
[24] H. R. Wilson and J. D. Cowan, "Excitatory and inhibitory interactions in localized populations of model neurons," *Biophysical journal*, vol. 12, no. 1, pp. 1–24, 1972.
[25] K. Ganguly, A. F. Schinder, S. T. Wong, and M.-m. Poo, "Gaba itself promotes the developmental switch of neuronal gabaergic responses from excitation to inhibition," *Cell*, vol. 105, no. 4, pp. 521–532, 2001.