Recurrent Convolutional Neural Network for Object Recognition

Ming Liang, Xiaolin Hu
Department of Computer Science, Tsinghua University.

In recent years, the convolutional neural network (CNN) [5] has achieved great success in many computer vision tasks [2, 4]. Partially inspired by neuroscience, CNN shares many properties with the visual system of the brain. A prominent difference is that CNN is typically a feed-forward architecture while in the visual system recurrent connections are abundant [3]. It is generally believed that recurrent synapses contribute to context modulation [1], which is important to the processing visual signals.

Inspired by this fact, we propose a recurrent CNN (RCNN) for object recognition by incorporating recurrent connections into each convolutional layer. The key module of RCNN is the recurrent convolutional layer (RCL) (Figure 1, left). The states of RCL units evolve over discrete time steps. For a unit located at (i, j) on the kth feature map in an RCL, its net input $z_{ijk}(t)$ at time step t is given by:

$$z_{ijk}(t) = (\mathbf{w}_k^f)^T \mathbf{u}^{(i,j)}(t) + (\mathbf{w}_k^r)^T \mathbf{x}^{(i,j)}(t-1) + b_k. \qquad (1)$$

In the equation $\mathbf{u}^{(i,j)}(t)$ and $\mathbf{x}^{(i,j)}(t-1)$ denote the feed-forward and recurrent input, respectively, which are the vectorized patches centered at (i, j) of the feature maps in the previous and current layer, $\mathbf{w}_k^f$ and $\mathbf{w}_k^r$ denote the vectorized feed-forward weights and recurrent weights, respectively, and b_k is the bias. The first term in 1 is used in standard CNN and the second term is induced by the recurrent connections. The activity or state of this unit is a function of its net input $x_{ijk}(t) = g(f(z_{ijk}(t)))$, where f is the rectified linear activation function and g is the local response normalization (LRN) function [4]. Unfolding this layer for T time steps results in a feed-forward subnetwork of depth $T+1$. While the feed-forward input remains the same, the recurrent input evolves over iterations. The effective RF of an RCL unit in the feature maps of the previous layer expands when the iteration number increases.

RCNN contains a stack of RCLs, optionally interleaved with max pooling layers. Training is performed by minimizing the cross-entropy loss function using the backpropagation throught time (BPTT) algorithm [8]. This is equivalent to using the standard BP algorithm on the time-unfolded network. If we unfold the recurrent connections through time, the model becomes a very deep feed-forward network. In addition, there are many shorter paths with different lengths.

From the computational perspective, the recurrent connections in RCNN offer several advantages. First, they enable every unit to incorporate context information in an arbitrarily large region in the current layer. As the time steps increase, the state of every unit is influenced by other units in a larger and larger neighborhood in the current layer (equation (1)); as a consequence, the size of regions that the unit can "watch" in the input space also increases. In CNN the RF size is fixed, and "watching" a larger region is only possible for units in higher layers. But unfortunately the context seen by higher-level units cannot influence the states of the units in the current layer. Second, the recurrent connections increase the network depth while keep the number of adjustable parameters constant by weight sharing. This is consistent with the trend of modern CNN architecture: going deeper with relatively small number of parameters [6, 7]. Note that simply increasing the depth of CNN by sharing weights between layers can result in the same depth and the same number parameters as RCNN, but such a model may not compete with RCNN in performance, as verified in our experiments. We attribute this fact to the difficulty in learning such a deep model. Then here comes the third advantage of RCNN — the time-unfolded RCNN is actually a CNN with multiple paths between the input layer to the output layer (Figure 1), which may facilitate the learning. On one hand, the existence of longer paths makes it possible for the model to learn highly complex features. On the other hand, the existence of shorter paths may help gradient backpropagation during training.

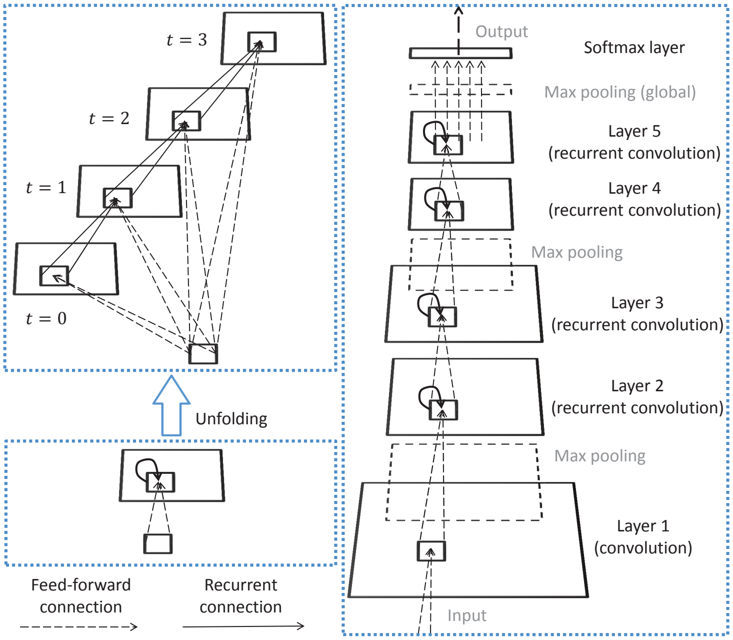

Figure 1: The overall architecture of RCNN. Left: An RCL is unfolded for $T = 3$ time steps, leading to a feed-forward subnetwork with the largest depth of 4 and the smallest depth of 1. At $t = 0$ only feed-forward computation takes place. Right: The RCNN used in this paper contains one convolutional layer, four RCLs, three max pooling layers and one softmax layer.

The model is tested on CIFAR-10, CIFAR-100, MNIST and SVHN. With fewer trainable parameters, RCNN outperforms the state-of-the-art models on all of these datasets. Increasing the number of parameters leads to even better performance. Detailed results are described in the paper. These results demonstrate the advantage of the recurrent structure over purely feed-forward structure for object recognition.

[1] Thomas D Albright and Gene R Stoner. Contextual influences on visual processing. *Annual review of neuroscience*, 25(1):339–379, 2002.

[2] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[3] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience*. Cambridge, MA: MIT Press, 2001.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[5] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[8] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.