PreNet: Parallel Recurrent Neural Networks for Image Classification

Junbo Wang[1,2], Wei Wang[1], Liang Wang[1,2], and Tieniu Tan[1(✉)]

[1] Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
{junbo.wang,wangwei,wangliang,tnt}@nlpr.ia.ac.cn
[2] University of Chinese Academy of Sciences (UCAS), Beijing, China

Abstract. Convolutional Neural Networks (CNNs) have made outstanding achievements in computer vision, e.g., image classification and object detection, by modelling the receptive field of visual cortex with convolution and pooling operations. However, CNNs have ignored to model the long-range spatial contextual information in images. It has long been believed that recurrent neural networks (RNNs) can model temporal sequences well by virtue of horizontal connections, and have been successfully applied in speech recognition and language modelling. In this paper, we propose a hierarchical parallel recurrent neural network (PreNet) to model spatial context for image classification. In this network, when transforming the whole image into sequences in four directions, we adopt the way of row-by-row/column-by-column scanning instead of traditional pixel-by-pixel scanning, for the convenience of fast convolution implementation. Following the recurrent network, a max-pooling operation is used to reduce the dimensionality of the obtained feature maps. The resulting PreNet can be easily paralleled on GPUs. We evaluate the proposed PreNet on two public benchmark datasets: MNIST and CIFAR-10. The proposed model can achieve the state-of-the-art classification performance, which demonstrates the advantage of PreNet over many comparative CNN structures.

Keywords: Convolutional neural networks · Image classification
Recurrent neural networks

1 Introduction

CNNs have become very popular in the field of deep learning, and have been successfully applied in computer vision, e.g., image classification [22], object detection [6], and semantic segmentation [26]. These successes are generally considered to be attributed to modelling the receptive field of visual cortex with local convolution operation and achieving translation invariance with max-pooling operation. However, several researchers argue that variant poolings may lose

J. Yang et al. (Eds.): CCCV 2017, Part II, CCIS 772, pp. 461–473, 2017.
https://doi.org/10.1007/978-981-10-7302-1_38

important location information [33], and local convolution can not capture the global spatial context in an image.

Different from the feedforward connections of convolutional neural networks, RNNs have horizontal feedback connections, which can model the long-range dependency in sequential data. With the capability of sequence modelling, RNNs with long short-term memory [12] have achieved great success in machine translation [3], question and answering [19], video super-resolution [16,17] and speech recognition [10]. Recently, more studies have been devoted to improving the memory capacity of RNNs for modelling longer-range contextual information, e.g., neural turing machine [14] and associative long short-term memory [5].

In this paper, we propose a hierarchical parallel recurrent neural network to model spatial context for image classification. Instead of using traditional recurrent neural networks in a pixel-by-pixel scanning order which leads to very long sequences (e.g., 10,000-length sequence for 100×100 pixel image), we adopt parallel recurrent neural networks to scan images in the way of row-by-row/column-by-column. Parallel recurrent neural networks not only shorten the sequence length to 100 for a 100×100 pixel image, but also use fast convolution operation to propagate information. Following parallel recurrent neural networks, a max-pooling operation is used to reduce the dimensionality of the obtained features maps. Through stacking parallel recurrent neural network layer and max-pooing layer, a hierarchical parallel recurrent neural network (PreNet) is proposed for image classification.

The most similar work to our PreNet is ReNet [35] which also replaces the convolution+pooling layer with four recurrent neural networks. However, our PreNet is much different from ReNet in many aspects. First, ReNet vertically sweeps the obtained feature maps based on the result of horizontal sweep, while PreNet recurrently handles the image sequences in four directions simultaneously. Second, ReNet sweeps the input from patch to patch in a pixel-by-pixel way, while PreNet sweeps the input in the way of row-by-row/column-by-column. Third, ReNet utilizes a fully connected operation at each recurrent time, while PreNet employs a efficient convolutional operation for fast parallel execution.

Finally, we evaluate the proposed PreNet on two widely-used benchmark datasets: MNIST and CIFAR-10. The experimental results demonstrate that the proposed model can achieve the state-of-the art classification performance on these datasets.

2 Model Description

In this part, we will first introduce traditional recurrent neural networks [11] and parallel recurrent neural networks, and then describe our proposed PreNet. Finally, we will compare the model structure between PreNet and the other two (ReNet and ConvNet).

2.1 Traditional Recurrent Neural Network

Different from common multilayer perception, RNN propagates information via horizontal feedback connections which can model the long-range dependency in sequential data, while multilayer perception passes on the activations of the neurons to the other neurons via feedforward connection. The input of the hidden layer in RNN includes both the input at current timestep and the hidden activations at previous timestep. Given a T length input sequence $\{x_t\}_{t=1,\cdots,T}$, where x_t is an input vector at timestep t, the hidden activations h_t and the output activations o_t at time $t(t = 1, \cdots, T)$ are computed as follows:

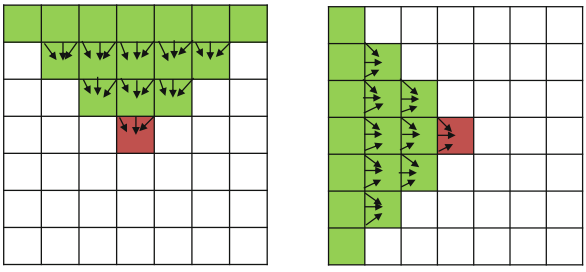
$$h_t = \sigma\left(x_t w_{xh} + h_{t-1} w_{hh} + b\right) \tag{1}$$

$$o_t = h_t w_{ho} \tag{2}$$

where w_{xh}, w_{hh}, and w_{ho} denote the network weights, b denotes the bias term, and σ denotes the activation function $\sigma(x) = \frac{1}{1+e^{-x}}$.

2.2 Parallel Recurrent Neural Network

Stollenga et al. [32] proposed a PyraMiD-LSTM model for biomedical volumetric image segmentation, which employs an elegant scanning way to considering each pixels entire spatio-temporal context. Besides, since the PyraMiD-LSTM uses a rather different scanning way, which is different from the traditional cuboid order of computations in MD-LSTM [13], it becomes very easy to parallelize.



(a)row-by-row scanning (b) column-by-column scanning

Fig. 1. The pyramidal connection topology of parallel recurrent neural network.

As we can see from Fig. 1, all inputs to all units are from four directions: left, right, up, or down. In Fig. 1, we adopt the way of row-by-row/column-by-column scanning instead of traditional pixel-by-pixel scanning, for the convenience of fast convolution implementation. For example, when the convolutional kernel size is 3, the stride is 1, and the elements in the second row (column) are the results of convolution of the first row (column) elements with the learned linear filter

at the stride. Figure 1 just shows the scanning ways of from left to right (from up to down), and the computation from other two directions can be operated in the same way. Since all the elements in a whole grid row (or column) can be computed independently, the proposed model can be easy to parallelize.

Assuming that $X = \{x_{i,j}\}$ denotes the input image or the feature map from a previous layer, where $X \in R^{c \times h \times w}$, and c denotes the number of channels or the feature dimensionality, h denotes the height, and w denotes the width. Specifically, we split X into h rows, where $x_t \in R^{c \times 1 \times w}$ $(t = 1, 2, \ldots, h)$ is the t-th row of the input image. The input sequence length is h or w, the input vector at time t is x_t, the weights are w_{xh} and w_{hh}, the activations of the hidden layer are h^i $(i \in \{lr, rl, ud, du\})$, where i denotes the sweep direction, i.e., left-to-right, right-to-left, up-to-down, down-to-up, respectively. When the recurrent model computes the hidden output of the direction of up-to-down, the hidden output of other three directions can be gained simultaneously in the same way. After the results of four directions are gained, the final output can be obtained by summing them. As we use the padding operation in the convolution operation, the output feature maps can keep the same size as the input. Now, each vector in final output feature maps can represent corresponding features in the context of the whole image.

$$h_t^{lr} = \sigma\left(x_t^{lr} * w_{xh}^{lr} + h_{t-1}^{lr} * w_{hh}^{lr} + b^{lr}\right) \tag{3}$$

$$h_t^{rl} = \sigma\left(x_t^{rl} * w_{xh}^{rl} + h_{t-1}^{rl} * w_{hh}^{rl} + b^{rl}\right) \tag{4}$$

$$h_t^{ud} = \sigma\left(x_t^{ud} * w_{xh}^{ud} + h_{t-1}^{ud} * w_{hh}^{ud} + b^{ud}\right) \tag{5}$$

$$h_t^{du} = \sigma\left(x_t^{du} * w_{xh}^{du} + h_{t-1}^{du} * w_{hh}^{du} + b^{du}\right) \tag{6}$$

$$h = h^{lr} + h^{rl} + h^{ud} + h^{du} \tag{7}$$

2.3 Our PreNet

Based on the fact that parallel recurrent neural network employs an elegant scanning way to consider each pixels entire spatio-temporal context and it is easy to parallelize, we propose a hierarchical parallel recurrent neural network (PreNet) to model spatial context for image classification. As we can see from the above section, the PreNet architecture can map an input image to an output feature map. Therefore, we can stack multiple PreNet layers to make our network deeper, which can extract more complex and useful features of the input image. In our experiment, we first directly stack multiple PreNet layers followed by fully connected layers to form a deep network for image classification. However, the computational cost is high because the size of the feature map in all PreNet layers is constant. To reduce the dimensionality of the obtained features maps, a max-pooling operation is added after the PreNet layers. The proposed hierarchy PreNet architecture is shown in Fig. 2, and the main process is as follows.

Given input image, it first enters one PreNet layer which comprises of four recurrent neural networks from four directions respectively, and every recurrent

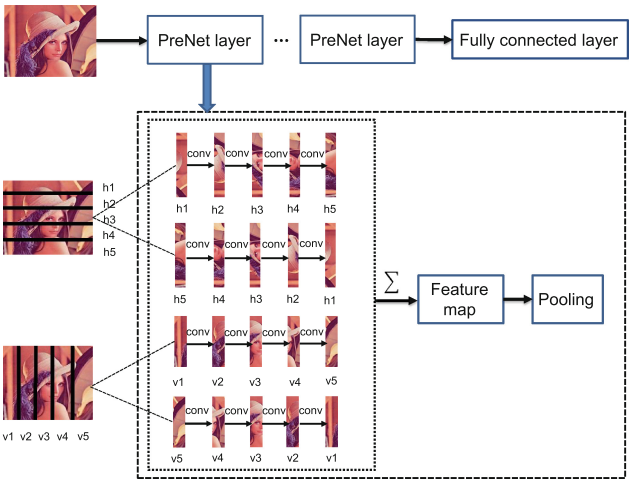


Fig. 2. The proposed Parallel Recurrent Neural Network (PreNet).

neural network handles the input image by the way of row-by-row/column-by-column scanning. Then it enters a pooling layer which maps the input feature map to a smaller size than the input. Let us denote function φ by the combination of the PreNet layer and the pooling layer, so we can stack multiple φ to form a hierarchy PreNet network. Finally, it is followed by two fully connected layers whose activation functions are rectify and softmax, respectively.

2.4 Model Compariation

In this section, we will compare the proposed PreNet with conventional convolutional neural networks (ConvNet) and ReNet. Since the comparison between ReNet and ConvNet is described in [35], we just explain the similarities and differences between PreNet and the other two here.

The comparison between PreNet and ReNet

- Both networks sweep the input image or feature map from four same directions. ReNet handles the sequences in the horizontal two directions and vertical two directions sequentially, while PreNet handles the sequences in four directions simultaneously. PreNet is obviously easier to parallelize than ReNet.
- ReNet sweeps the input from patch to patch in a pixel-by-pixel way, while PreNet handles the input in the way of row-by-row/column-by-column scanning. Both the way of scanning can ensure that each feature activation gains contextual information with respect to the whole image.
- Both networks can reduce the dimensionality of the feature map, which results in lower computational cost. However, ReNet reduces the dimension of the input feature map by dividing it into many non-overlapping patches and the

existence of learned lateral connections, but PreNet reduces the dimension of the input feature map by a max-pooling operation.

The comparison between PreNet and ConvNet

- Both architectures employ a set of filters in the input image or the feature map from the layer below. Nevertheless, the information of PreNet is transmitted via lateral connections which propagate across the whole image, while ConvNet only captures local information in a fixed input size. The lateral connections make the model extract more contextual information at each layer.
- They both have the max-pooling layer. For ConvNet, pooling operations are responsible for the translation invariant property. But for PreNet, pooling operations are mainly responsible for reducing dimensionality.

3 Experiments

To demonstrate the effectiveness of the proposed PreNet, we apply it to image classification on two publicly available datasets.

3.1 Datasets

The two evaluation datasets and corresponding experimental protocols are described as follows.

MNIST. The dataset [23] is composed of binary images of handwritten digits, and has been widely used for training various image processing systems. The images are all gray with a size of 28×28, each of which represents a handwritten digit from 0 to 9. In this experiment, we follow the standard split and use 50,000, 10,000 and 10,000 images for training, validation and test, respectively.

CIFAR-10. The dataset [21] consists of 60000 color images. These images have a size of 32×32, each of which belongs to one of 10 classes including airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck, totally 6000 images per class. Following the standard procedure, we split the dataset into 40,000 training, 10,000 validation and 10,000 test images. We normalize each pixel to have zero-mean and unit-variance across all training samples.

3.2 Pre-processing and Data Augmentation

It is believed that data preprocessing can usually have a significant impact on the final performance of a model [20]. To make fair comparisons, we exploit different pre-processing strategies for different datasets. For the MNIST dataset, we normalize each pixel of an image to the range of $[0, 1]$. For the CIFAR-10

dataset, we perform z-score standardization to make image pixels have zero-mean and unit-variance.

According to [22], data augmentation can always reduce overfitting and improve final performance. To extensively compare our model with others, we conduct our experiment using non-augmentation and augmentation, respectively. In the context of data augmentation, we mainly exploit three kinds of transformations [24,35] including flipping, shifting, random cropping.

3.3 Model Parameter Setup

In Table 1, we illustrate parameter settings of the proposed PreNet on the MNIST and CIFAR-10 datasets. The main parameters include the number of PreNet layers N_{pre}, their corresponding feature dimensionality d_{pre} and kernel size k_{pre}, the number of pooling layers N_{pool} and their corresponding kernel size k_{pool}, the number of fully connected layers N_{fc} and their corresponding feature dimensionality d_{fc}, and the activation function f_{fc}. All our experiments are performed on a ubuntu computer with an NVIDIA Tesla K40 GPU.

Table 1. Model parameters used in the experiments.

Parameters	MNIST	CIFAR-10
N_{pre}	2	2
d_{pre}	32	48
k_{pre}	7	5
N_{pool}	2	2
k_{pool}	2	2
N_{fc}	1	1
d_{fc}	128	256
f_{fc}	$max(0, x)$	$max(0, x)$
$Flipping$	No	Yes
$Shifting$	Yes	Yes
$RandomCropping$	Yes	Yes

3.4 Training

We apply stochastic gradient descent (SGD) with nesterov momentum [34] to train the networks. At the same time, to release the exploding gradient problem [2], we employ the gradient norm clipping strategy [28]. To avoid overfitting during training, in addition to the data augmentation, we perform dropout after all PreNet layers and fully connected layers. In addition, we use batch normalization [18] to accelerate the convergence of the proposed model. During training, we choose the best model which minimizes the classification loss function on the

validation set. We set the learning rate 0.01 at start and drop it by a factor of 10 after every 500 epoches. The momentum term is 0.9, the batch size is 100, and all weights are initialized according to a uniform distribution.

3.5 Results and Analysis

We compare the proposed PreNet with several state-of-the-art models on the MNIST and CIFAR-10 datasets in Tables 2 and 3, respectively. All results to be compared in Tables 2 and 3 are from [35]. On the MNIST dataset, our proposed PreNet model can outperform most state-of-the-art models. On the CIFAR-10 dataset, the PreNet model performs comparably to most state-of-the-art models. It is worth mentioning that PreNet performs 0.16% and 1.35% better than ReNet on MNIST and CIFAR-10, respectively. Although the performance of our model doesn't outperform some state-of-the-art models on CIFAR-10, there are some reasons behind it. As we all know, the digital structure on MNIST is simple, while the intra-class spatial context variation on CIFAR-10 is complicated.

To verify the advantage of PreNet further, we compare PreNet with ConvNet on the parameters of the same order of magnitude on the CIFAR-10 dataset. Specifically, we use the same number of layers for PreNet and ConvNet, and compare their results in Table 4, where the ConvNet-one means one layer convolution, and PreNet-one means one layer PreNet, and the meanings of "-two" are similar. In this experiment, we set the number of fully connected layers as 1, and the number of neurons as 128. All the results in Table 4 are computed in the same computing environment, and data augmentation is not used in these experiments. From the experimental results, we can find that PreNets can achieve

Table 2. Comparison with existing models on MNIST.

Method	Test error
DropCNN [36]	0.28%
PreNet	**0.29%**
S-SCNN [9]	0.31%
DBSN [4]	0.35%
CKN [27]	0.39%
DSN [24]	0.39%
SCNN [29]	0.40%
FMP [8]	0.44%
Maxout [7]	0.45%
ReNet [35]	0.45%
NIN [25]	0.47%
COSFIRE [1]	0.52%

Table 3. Comparison with existing models on CIFAR-10.

Method	Test error
FMP [8]	4.5%
S-SCNN [9]	6.28%
NIN [25]	8.8%
Maxout [7]	9.35%
PV-Maxout [31]	9.39%
BO [30]	9.5%
PreNet	**11%**
DCNN [22]	11%
DropCNN [36]	11.10%
ReNet	12.35%
SP-CNN [37]	15.13%
PCFD [15]	15.6%

Table 4. Comparison with ConvNet models on CIFAR-10 (without data augmentation).

Method	Test accuracy
ConvNet-one	74.66%
PreNet-one	**77.5%**
ConvNet-two	80.92%
PreNet-two	**83.57%**

Table 5. The experimental results of PreNet on the CIFAR-10 by setting different parameters.

Parameter	Test accuracy
$N_{pre} = 1$	71.39%
$N_{pre} = 2$	78.15%
$d_{pre} = 16$	69.23%
$d_{pre} = 32$	71.39%
$d_{pre} = 48$	72.73%
$d_{pre} = 64$	72.80%
$k_{pre} = 3$	70.76%
$k_{pre} = 5$	71.39%
$k_{pre} = 7$	71.41%
$d_{fc} = 32$	70.12%
$d_{fc} = 64$	71.39%
$d_{fc} = 128$	72.98%
$d_{fc} = 256$	73.79%
$d_{fc} = 512$	74.28%

about 3% better than ConvNets. These results demonstrate the advantage of PreNet compared with convolutional neural networks further.

In the following, we testify how the parameters of the proposed PreNet affect the final performance. In a nutshell, we study a parameter by changing its value while fixing other parameters. In Table 5, we present the experimental results of different parameter setups on the CIFAR-10 dataset. Similarly, all the results in Table 5 are gained in the same computing environment, and data augmentation is not used in these experiments. The meaning of these parameters are explained in the previous section. The first column in table indicates four kinds of parameters respectively: the number of PreNet layers, the feature dimensionality of each PreNet layer, the kernel size of each PreNet layer, the feature dimensionality of each fully connected layer. The second column is their corresponding results.

In our experiments, we set different values for these parameters respectively. By increasing the values of N_{pre}, d_{pre}, k_{pre} and d_{fc}, the performance of the model gradually gets better. The situation is obviously consistent with our thought. As we do not fine tuning the learning rate, the model performance does not increase obviously when changing the values of some parameters such as k_{pre}. What calls for special attention is that the model performance cannot increase too much when the value of the parameter becomes too large because of overfitting.

Finally, we show the feature maps of the convolutional layers in PreNet and ConvNet, respectively. In Figs. 3 and 4, there are some feature maps of PreNet and ConvNet corresponding to a bird and a horse. The layer1 means the first

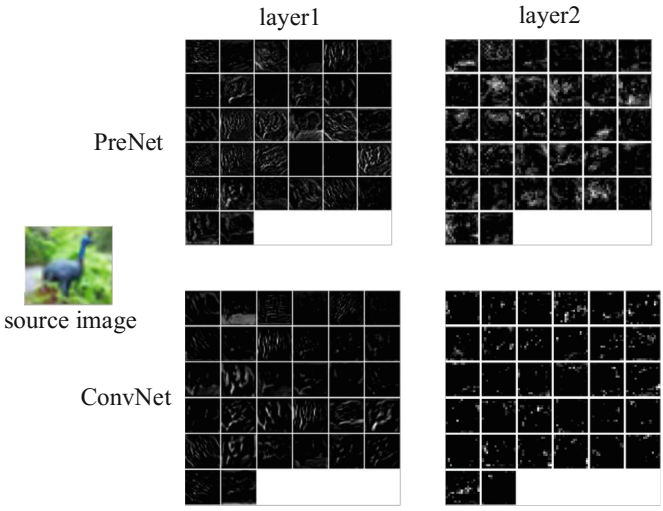


Fig. 3. The feature maps of PreNet and ConvNet on a bird image.

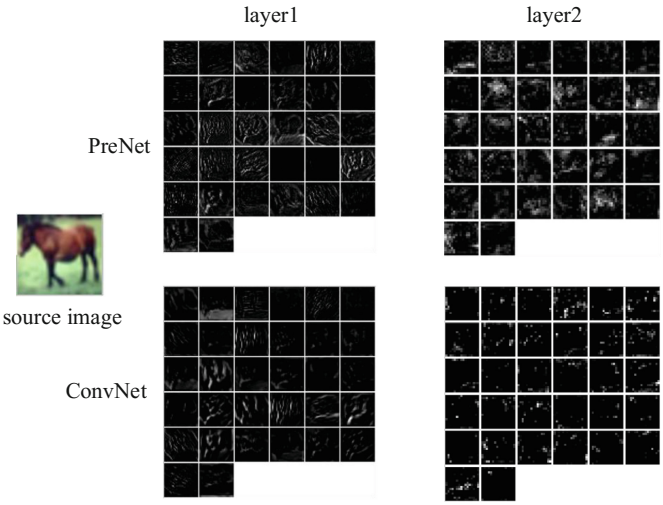


Fig. 4. The feature maps of PreNet and ConvNet on a horse image.

layer (PreNet or ConvNet) map, and the layer2 means the second layer (PreNet or ConvNet) map. As we can see from the Figs. 3 and 4, PreNet can capture richer context information compared to the little contour information of ConvNet. This is also consistent with our original motivation.

4 Conclusion

In this paper, we have proposed a hierarchical parallel recurrent neural network (PreNet) to model spatial context for image classification. The main idea is to handle the input image in the way of row-by-row/column-by-column scanning from four different directions by recurrent convolution operation. On the one hand, this structure makes the model easily capture global context information of the object in the image. On the other hand, it naturally facilitates parallel computing of the model. Our experimental results have shown the advantage of PreNet compared to common convolution neural networks and ReNet.

To further extend our work, there are several things worth exploring: First, the recurrent unit of PreNet is just conventional RNN instead of LSTM. It is a fact that LSTM has a powerful ability of modeling a long sequence, which can capture long-range memory about contextual information. It can be used in more complicated datasets if LSTM is applied to the PreNet model. Second, given that ReNet sweeps the input from patch to patch in a pixel-by-pixel way, a variant of PreNet can be explored in this respect. That is to say, a variant of PreNet can scan the input from patch to patch (e.g., slice row to slice row) in the same way as the original work principle.

Acknowledgements. This work is jointly supported by National Key Research and Development Program of China (2016YFB1001000), and National Natural Science Foundation of China (61525306, 61633021).

References

1. Azzopardi, G., Petkov, N.: Trainable cosfire filters for keypoint detection and pattern recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(2), 490–503 (2013)
2. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157–166 (1994)
3. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
4. Ciresan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Deep big simple neural nets excel on handwritten digit recognition. CoRR abs/1003.0358 (2010). http://arxiv.org/abs/1003.0358
5. Danihelka, I., Wayne, G., Uria, B., Kalchbrenner, N., Graves, A.: Associative long short-term memory. arXiv preprint arXiv:1602.03032 (2016)
6. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)

7. Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. arXiv preprint arXiv:1302.4389 (2013)
8. Graham, B.: Fractional max-pooling. arXiv preprint arXiv:1412.6071 (2014)
9. Graham, B.: Spatially-sparse convolutional neural networks. arXiv preprint arXiv:1409.6070 (2014)
10. Graves, A., Mohamed, A.R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645–6649. IEEE (2013)
11. Graves, A.: Supervised sequence labelling. In: Graves, A. (ed.) Supervised Sequence Labelling with Recurrent Neural Networks. SCI, vol. 385, pp. 5–13. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24797-2_2
12. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. **18**(5), 602–610 (2005)
13. Graves, A., Schmidhuber, J.: Offline handwriting recognition with multidimensional recurrent neural networks. In: Advances in Neural Information Processing Systems, pp. 545–552 (2009)
14. Graves, A., Wayne, G., Danihelka, I.: Neural turing machines. arXiv preprint arXiv:1410.5401 (2014)
15. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
16. Huang, Y., Wang, W., Wang, L.: Bidirectional recurrent convolutional networks for multi-frame super-resolution. In: Advances in Neural Information Processing Systems, pp. 235–243 (2015)
17. Huang, Y., Wang, W., Wang, L.: Video super-resolution via bidirectional recurrent convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. (2017)
18. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
19. Iyyer, M., Boyd-Graber, J.L., Claudino, L.M.B., Socher, R., Daume III, H.: A neural network for factoid question answering over paragraphs. In: EMNLP, pp. 633–644 (2014)
20. Kotsiantis, S., Kanellopoulos, D., Pintelas, P.: Data preprocessing for supervised leaning. Int. J. Comput. Sci. **1**(2), 111–117 (2006)
21. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
23. LeCun, Y., Bottou, L., Bengio, Y., Haner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
24. Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. arXiv preprint arXiv:1409.5185 (2014)
25. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)
26. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
27. Mairal, J., Koniusz, P., Harchaoui, Z., Schmid, C.: Convolutional kernel networks. In: Advances in Neural Information Processing Systems, pp. 2627–2635 (2014)
28. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. arXiv preprint arXiv:1211.5063 (2012)

29. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In: Document Analysis and Recognition, p. 958. IEEE (2003)
30. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. In: Advances in Neural Information Processing Systems, pp. 2951–2959 (2012)
31. Springenberg, J.T., Riedmiller, M.: Improving deep neural networks with probabilistic maxout units. arXiv preprint arXiv:1312.6116 (2013)
32. Stollenga, M.F., Byeon, W., Liwicki, M., Schmidhuber, J.: Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation. In: Advances in Neural Information Processing Systems, pp. 2980–2988 (2015)
33. Su, H., Liu, F., Xie, Y., Xing, F., Meyyappan, S., Yang, L.: Region segmentation in histopathological breast cancer images using deep convolutional neural network. In: 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI), pp. 55–58. IEEE (2015)
34. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: Proceedings of the 30th International Conference on Machine Learning (ICML 2013), pp. 1139–1147 (2013)
35. Visin, F., Kastner, K., Cho, K., Matteucci, M., Courville, A., Bengio, Y.: ReNet: a recurrent neural network based alternative to convolutional networks. arXiv preprint arXiv:1505.00393 (2015)
36. Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R.: Regularization of neural networks using dropconnect. In: Proceedings of the 30th International Conference on Machine Learning (ICML 2013), pp. 1058–1066 (2013)
37. Zeiler, M.D., Fergus, R.: Stochastic pooling for regularization of deep convolutional neural networks. arXiv preprint arXiv:1301.3557 (2013)