Deep Learning Technology and Application

Ge Li

Peking University

Table of contents

带有卷积权值的网络

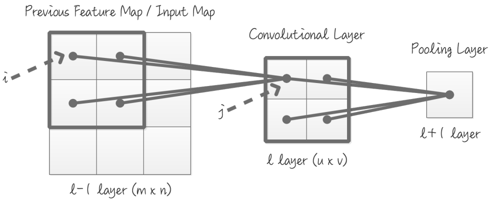
设 α_{ij} 为第 i 个输入特征图与第 j 个卷积核之间的连接强度；

$$a_j^l = f\left(\sum_{i=1}^{N_{in}} \alpha_{ij}(x_i^{l-1} * k_i^l) + b_j^l\right)$$

其中：

- N_{in} 为输入特征图的个数；
- $\sum_i \alpha_{ij} = 1$ 且 $0 \leq \alpha_{ij} \leq 1$；
- 可以设 $\alpha_{ij} = \frac{exp(c_{ij})}{\sum_k exp(c_{kj})}$；
- 接下来的运算，将针对一个确定的输出特征图 j，因此上述公式可以简化为：$\alpha_i = \frac{exp(c_i)}{\sum_k exp(c_k)}$；

带有卷积权值的网络



接下来看一下，推导 α_{ij}，即 α_i 的求解方法：

$$\frac{\partial J}{\partial \alpha_i} = \frac{\partial J}{\partial z^l}\frac{\partial z^l}{\partial \alpha_i} = \sum_{u,v}(\delta^l \circ (a_i^{l-1} * k_i^l))_{uv}$$

继续求解关于 c_i 的导数：

$$\frac{\partial J}{\partial c_i} = \sum_k \frac{\partial J}{\partial \alpha_k}\frac{\partial \alpha_k}{\partial c_i} = \alpha_i\left(\frac{\partial J}{\partial \alpha_i} - \sum_k \frac{\partial J}{\partial \alpha_k}\alpha_k\right)$$

带有卷积权值的网络

为了能够增强 α_i 的效果，我们可以想办法让它变得更稀疏：

$$\hat{J}(W,b) = J(W,b) + \Omega(\alpha) = J(W,b) + \lambda \sum_{i,j} |(\alpha)_{ij}|$$

在这个条件下，α 如何更新呢？

$$\frac{\partial \Omega}{\partial \alpha_i} = \lambda \; sign(\alpha_i)$$

将 α 的求解方式代入：

$$\frac{\partial \Omega}{\partial c_i} = \sum_k \frac{\partial \Omega}{\partial \alpha_k} \frac{\partial \alpha_k}{\partial c_i} = \lambda \left(|\alpha_i| - \alpha_i \sum_k |\alpha_k| \right)$$

因此：

$$\frac{\partial \hat{J}}{\partial c_i} = \frac{\partial J}{\partial c_i} + \frac{\partial \Omega}{\partial c_i}$$

Tiled Convolution

Tiled Convolution :

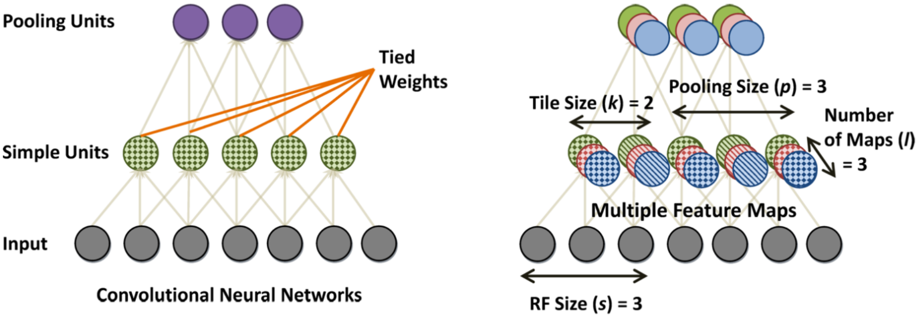
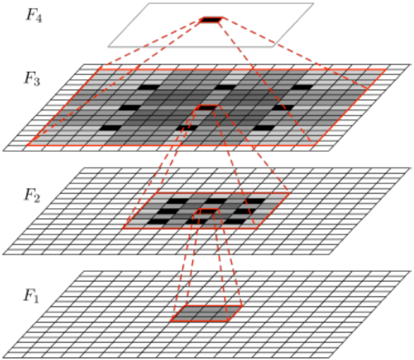


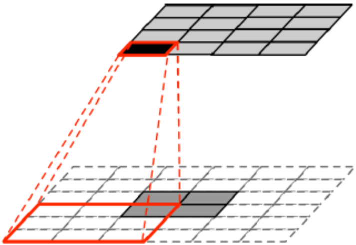
Figure 1: Left: Convolutional Neural Networks with local receptive fields and tied weights. Right: Partially untied local receptive field networks – Tiled CNNs. Units with the same color belong to the same map; within each map, units with the same fill texture have tied weights. (Network diagrams in the paper are shown in 1D for clarity.)

[1] J. Ngiam, Z. Chen, D. Chia, P. W. Koh, Q. V. Le, A. Y. Ng, Tiled convolutional neural networks, in: NIPS, 2010.

Transposed Convolution, Dilated Convolution



(c) Dilated Convolution

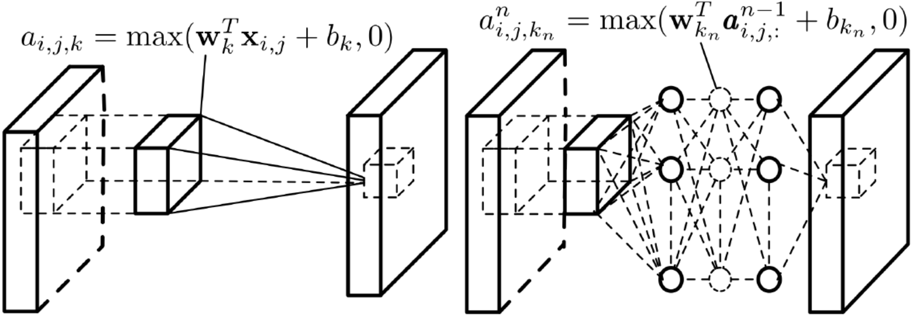


(d) Deconvolution

[d] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: ECCV, 2014.
[c] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: ICLR, 2016.

Network in Network

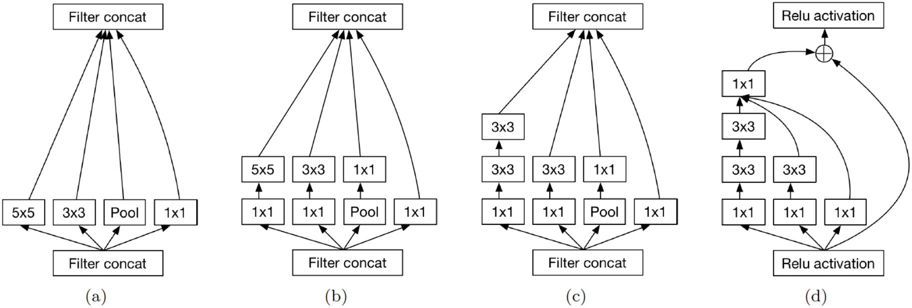
Network in Network:



(a) Linear convolution layer (b) Mlpconv layer

[1] M. Lin, Q. Chen, S. Yan, Network in network, in: ICLR, 2014.

Inception Module



[b] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: CVPR, 2015.
[c] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: arXiv preprint arXiv:1512.00567, 2015.
[d] C. Szegedy, S. Ioffe, V. Vanhoucke, Inception-v4, inception-resnet and the impact of residual connections on learning, in: arXiv preprint arXiv:1602.07261, 2016.

L_pPooling

L_pPooling[1] 是一种泛化能力较强的 Pooling 方法；

$$y_{i,j,k} = \left[\sum_{(m,n)\in R_{ij}} (a_{m,n,k})^p \right]^{\frac{1}{p}}$$

- 当 $p = 1$ 时，L_pPooling 相当于"Average Pooling";
- 当 $p = \infty$ 时，L_pPooling 相当于"Max Pooling";

[1]A. Hyvarinen, U. Koster, Complex cell pooling and the statistics of natural images, in: NCNS, 2007.

Mixed Pooling

Mix Pooling[1] 是一种 Max Pooling 与 Mean Pooling 混合的方式：

$$y_{i,j,k} = \lambda \ max_{(m,n)\in R_{ij}} a_{m,n,k} + (1-\lambda)\frac{1}{|R_{ij}|}\sum_{(m,n)\in R_{ij}} a_{m,n,k}$$

- Inspired by random Dropout and DropConnect.
- Experiments in show that mixed pooling can better address the overfitting problems and it performs better than max pooling and average pooling.

[1]D. Yu, H. Wang, P. Chen, Z. Wei, Mixed pooling for convolutional neural networks, in: Rough Sets and Knowledge Technology, 2014.

Stochastic Pooling

- Stochastic pooling[1] randomly picks the activations according to a multinomial distribution.
- Stochastic pooling first computes the probabilities p for each region R_j by normalizing the activations within the region:

$$p_i = \frac{a_i}{\sum_{k \in R_j}}(a_k)$$

- Compared with max pooling, stochastic pooling can avoid overfitting due to the stochastic component.

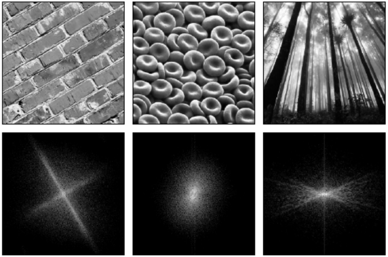
[1]M. D. Zeiler, R. Fergus, Stochastic pooling for regularization of deep convolutional neural networks, in: ICLR, 2013.

Spectral Pooling

通过离散傅立叶变换，将特征图像转换到频域再进行 Pooling[1]：
对于输入 $x \in C^{M \times N}$，其对应的傅立叶转换为：

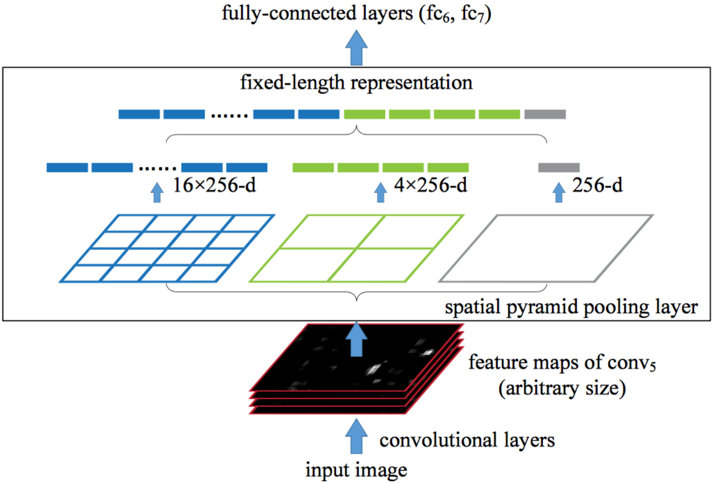
$$F(x)_{hw} = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{mn} e^{-1\pi i(\frac{mh}{M})+\frac{nw}{N}}$$

$$\forall h \in 0, \ldots, M-1, \forall w \in 0, \ldots, N-1.$$



[1] O. Rippel, J. Snoek, R. P. Adams, Spectral representations for convolutional neural networks, in: NIPS, 2015.

Spatial Pyramid Pooling



[1] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: ECCV, 2014.

Thanks.