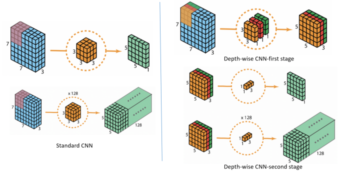
Deep Learning

Separable Convolutions and Flattening



Learning goals

- Separable Convolutions
- Flattening

Separable Convolutions

SEPARABLE CONVOLUTIONS

- Separable Convolutions are used in some neural net architectures, such as the MobileNet.
- Motivation: make convolution computationally more efficient.
- One can perform:
  - spatially separable convolution
  - depthwise separable convolution.

The **spatially separable convolution** operates on the 2D spatial dimensions of images, i.e. height and width. Conceptually, spatially separable convolution decomposes a convolution into two separate operations.

- Consider the sobel kernel from the previous lecture:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}$$

SEPARABLE CONVOLUTIONS

- this 3x3 dimensional kernel can be replaced by the outer product of two 3x1 and 1x3 dimensional kernels:

$$\begin{bmatrix} +1 \\ +2 \\ +1 \end{bmatrix} * \begin{bmatrix} +1 & 0 & -1 \end{bmatrix}$$

- Convolving with both filters subsequently has a similar effect, reduces the amount of parameters to be stored and thus improves speed:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

Figure: In convolution, the 3x3 kernel directly convolves with the image. In spatially separable convolution, the 3x1 kernel first convolves with the image. Then the 1x3 kernel is applied. This would require 6 instead of 9 parameters while doing the same operations.

SPATIALLY SEPARABLE CONVOLUTION

Example 1: A convolution on a 5×5 image with a 3×3 kernel (stride=1, padding=0) requires scanning the kernel at 3 positions horizontally and 3 vertically. That is 9 positions in total, indicated as the dots in the image below. At each position, 9 element-wise multiplications are applied. Overall, that is 9 x 9 = 81 multiplications.

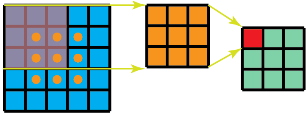


Figure: Standard convolution with 1 channel.

SPATIALLY SEPARABLE CONVOLUTION

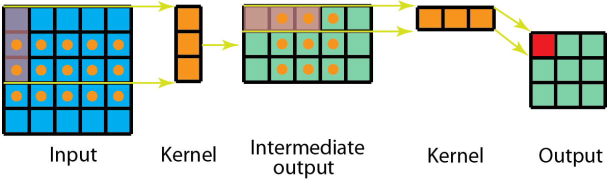


Figure: Spatially separable convolution with 1 channel. Overall, the spatially separable convolution takes 45 + 27 = 72 multiplications. (Image source: Bai (2019))

Note: However, despite their advantages, spatial separable convolutions are seldom applied in deep learning. This is mainly due to not all kernels being able to get divided into two smaller ones. Replacing all standard convolutions by spatial separable would also introduce a limit in searching for all possible kernels in the training process, implying worse training results.

DEPTHWISE SEPARABLE CONVOLUTION

- The depthwise separable convolutions, which is much more commonly used in deep learning (e.g. in MobileNet and Xception).
- This convolution separates convolutional process into two stages of depthwise and pointwise.

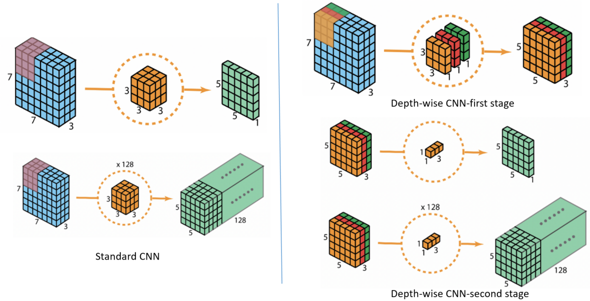


Figure: Comparison between standard cnn and separable depthwise cnn

DEPTHWISE SEPARABLE CONVOLUTION

Standard convolution:

3 x 3 x 3 kernel size

5 x 5 times move

128 kernels

3 x 3 x 3 x 5 x 5 x 128 = 86.400

Depthwise separable convolution:

Depthwise convolution

3 x 3 x 1 kernel size

5 x 5 times move

3 kernels

3 x 3 x 1 x 5 x 5 x 3 = 675

Pointwise convolution

1 x 1 x 3 kernel size

5 x 5 times move

128 kernels

1 x 1 x 3 x 5 x 5 x 128 = 9.600

675 + 9.600 = 10.275

Figure: Comparision of number of multiplications in Depthwise separable cnn and standard cnn

Therefore, fewer computations leads faster network.

DEPTHWISE SEPARABLE CONVOLUTION

Original convolution:

5 x 5 x 3 kernel size

8 x 8 times move

256 kernels

5 x 5 x 3 x 8 x 8 x 256 = 1.228.800

Depthwise separable convolution:

Depthwise convolution

5 x 5 x 1 kernel size

8 x 8 times move

3 kernels

5 x 5 x 1 x 8 x 8 x 3 = 4.800

Pointwise convolution

1 x 1 x 3 kernel size

8 x 8 times move

256 kernels

1 x 1 x 3 x 8 x 8 x 256 = 49.152

4.800 + 49.152 = 53.952

Figure: Comparision of number of multiplications in Depthwise separable cnn and standard cnn

DEPTHWISE CONVOLUTION

As the name suggests, we perform kernel on depth of the input volume (on the input channels). The steps followed in this convolution are:

- Take number of kernels equal to the number of input channels, each kernel having depth 1. Example, if we have a kernel of size 3×3 and an input of size 6×6 with 16 channels, then there will be $16 \times 3 \times 3$ kernels.
- Every channel thus has 1 kernel associated with it. This kernel is convolved over the associated channel separately resulting in 16 feature maps.
- Stack all these feature maps to get the output volume with 4×4 output size and 16 channels.

POINTWISE CONVOLUTION

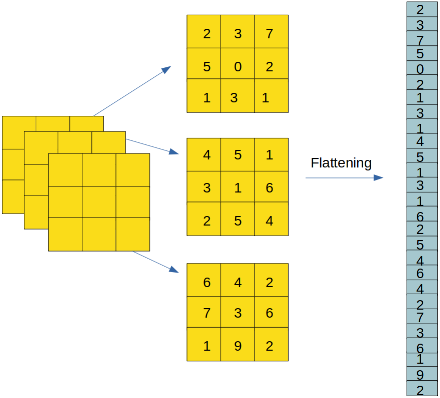
As the name suggests, this type of convolution is applied to every single point in the convolution separately (remember 1×1 convs?). So how does this work?

- Take a 1×1 conv with number of filters equal to number of channels you want as output.
- Perform basic convolution applied in 1×1 conv to the output of the Depth-wise convolution.

Flattening

FLATTENING

Flattening is converting the data into a 1-dimensional array for inputting it to the next layer. We flatten the output of the convolutional layers to create a single long feature vector. And it is connected to the final classification model, which is called a fully-connected layer.



REFERENCES

- Dumoulin, Vincent and Visin, Francesco (2016)
  A guide to convolution arithmetic for deep learning
  https://arxiv.org/abs/1603.07285v1
- Van den Oord, Aaron, Sander Dielman, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, and Koray Kavukocuoglu (2016)
  WaveNet: A Generative Model for Raw Audio
  https://arxiv.org/abs/1609.03499
- Benoit A., Gennart, Bernard Krummenacher, Roger D. Hersch, Bernard Saugy, J.C. Hadorn and D. Mueller (1996)
  The Giga View Multiprocessor Multidisk Image Server
  https://www.researchgate.net/publication/220060811_The_Giga_View_Multiprocessor_Multidisk_Image_Server
- Tran, Du, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani and Paluri Manohar (2015)
  Learning Spatiotemporal Features with 3D Convolutional Networks
  https://arxiv.org/pdf/1412.0767.pdf

REFERENCES

Milletari, Fausto, Nassir Navab and Seyed-Ahmad Ahmadi (2016)
V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation
https://arxiv.org/pdf/1606.04797.pdf

Zhang, Xiang, Junbo Zhao and Yann LeCun (2015)
Character-level Convolutional Networks for Text Classification
http://arxiv.org/abs/1509.01626

Wang, Zhiguang, Weizhong Yan and Tim Oates (2017)
Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline
http://arxiv.org/abs/1509.01626

Fisher Yu and Vladlen Koltun (2015)
Multi-Scale Context Aggregation by Dilated Convolutions
https://arxiv.org/abs/1511.07122

REFERENCES

Bai, Shaojie, Zico J. Kolter and Vladlen Koltun (2018)
An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling
http://arxiv.org/abs/1509.01626

Augustus Odena, Vincent Dumoulin and Chris Olah (2016)
Deconvolution and Checkerboard Artifacts
https://distill.pub/2016/deconv-checkerboard/*https://distill.pub/2016/deconv-checkerboard/*

Andre Araujo, Wade Norris and Jack Sim (2019)
Computing Receptive Fields of Convolutional Neural Networks
https://distill.pub/2019/computing-receptive-fields/

Zhiguang Wang, Yan, Weizhong and Tim Oates (2017)
Time series classification from scratch with deep neural networks: A strong baseline
https://arxiv.org/1611.06455

REFERENCES

Lin, Haoning and Shi, Zhenwei and Zou, Zhengxia (2017)
Maritime Semantic Labeling of Optical Remote Sensing Images with Multi-Scale Fully Convolutional Network