

STAT430: Machine Learning for Financial Data

LarryHua.com/teaching

Spring 2019

Convolutional neural networks

Understanding convnet - convolution layer

- Revisit MNIST data
 - [Try R](#)
- `layer_conv_2d(filters = 32, kernel_size = c(3, 3), activation = "relu", input_shape = c(28, 28, 1))`
- Feature maps: 3D tensors (height, width, depth/channel)
 - color images: (height, width, 3)
 - black-white images: (height, width, 1)
 - eg, MNIST: (28, 28, 1)
- Size of the patches extracted from the inputs: 3×3 or 5×5
 - eg, `kernel_size = c(3, 3)`
 - use the same weights/bias for each patch
 - 10 parameters for each filter in the 3×3 case

Understanding convnet - convolution layer

- Output feature maps: 3D tensors (height', width', depth of the output feature map)
 - depth of the output feature map: number of filters computed from the convolution layer
 - filters encode specific aspects of the input data
 - eg, MNIST: **filters** = 32, the first convolution layer outputs a feature map of size (26, 26, 32)
 - Why 26?
 - A patch of size 3×3 slides over a 28×28 square, so 26×26 locations in total
- Response map: the height' \times width' grid of values for **each** filter

Understanding convnet - convolution layer

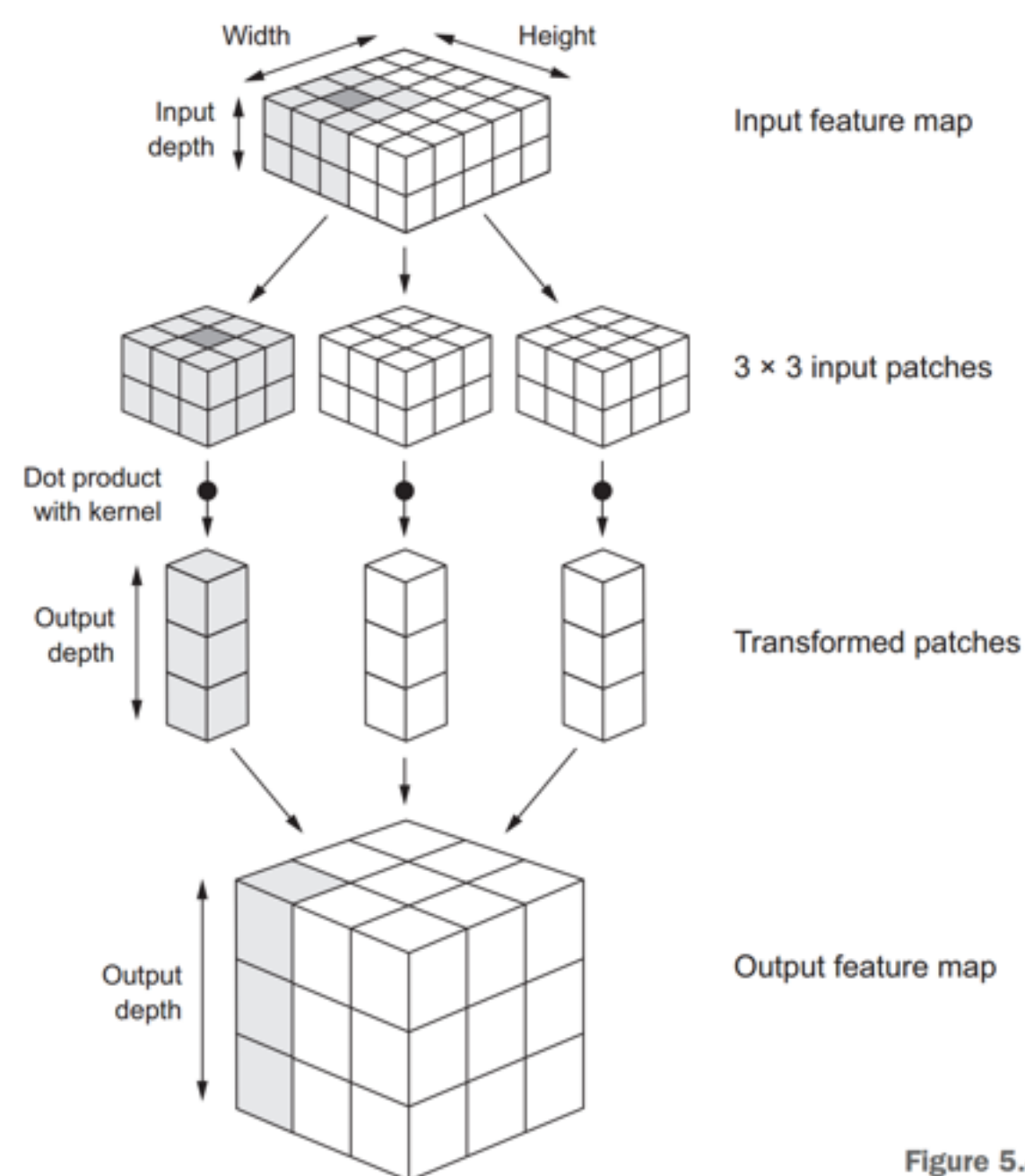


Figure 5.4

Understanding convnet - convolution layer

- Padding: adding rows and columns on each side of the input feature map to make it possible to fit center convolution windows around every input tile
 - no padding (default): `layer_conv_2d (... , padding = "valid", ...)`
 - same as input: `layer_conv_2d (... , padding = "same", ...)`
- Stride: distance between two successive windows (default = 1)
 - used to downsample feature map by a factor of the value of strides (rarely used in practice)

Understanding convnet - max-pooling

- `layer_max_pooling_2d(pool_size = c(2, 2))`
- Max pooling: extracting windows from the input feature maps, and for each channel outputting the max value within each window
 - aggressively down-sample feature maps
 - similar to convolution layer, but with patches of size 2×2
 - outputting the maximum of the 4 values for each channel
 - `pool_size = c(2, 2)` means down-sampling by a factor of 2 for both vertical and horizontal axis

Understanding convnet - max-pooling

- Why not keep fairly large feature maps all the way up?
 - Spatial information cannot be effectively mixed
 - 3×3 patches for three successive convolution layers can only be mixed to be 7×7 patches
 - i.e., each 3×3 patch in the 3rd convolution layer only contains a 7×7 patch from the image
 - too many parameters
- Why max ?
 - it's more informative to look at the maximal presence of different features
 - tends to work better than say average pooling
- [Back to Course Scheduler](#)