Machine Learning and
Computer Vision Group

IST AUSTRIA
Institute of Science and Technology

Deep Learning with TensorFlow

http://cvml.ist.ac.at/courses/DLWT_W18

Lecture 5:
Convolutional Networks

Convolutional Neural Networks

Martin Töpfer

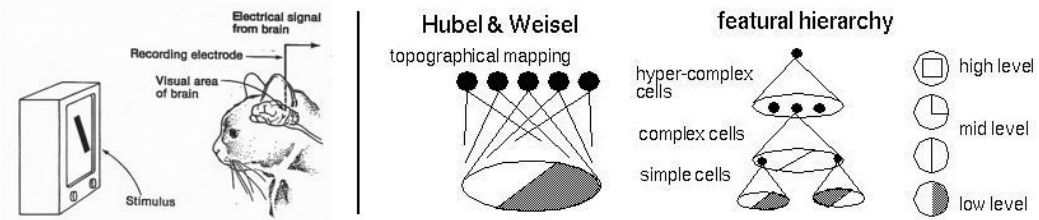
IST Austria

December 17, 2018

Outline

Biological motivation

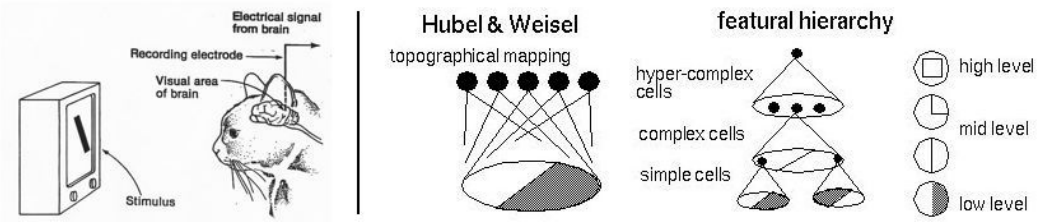
- 1958: David Hubel, Torsten Weisel: neurons in visual cortex have often small local receptive field
  - some reacts only to certain shapes



hskld

Biological motivation

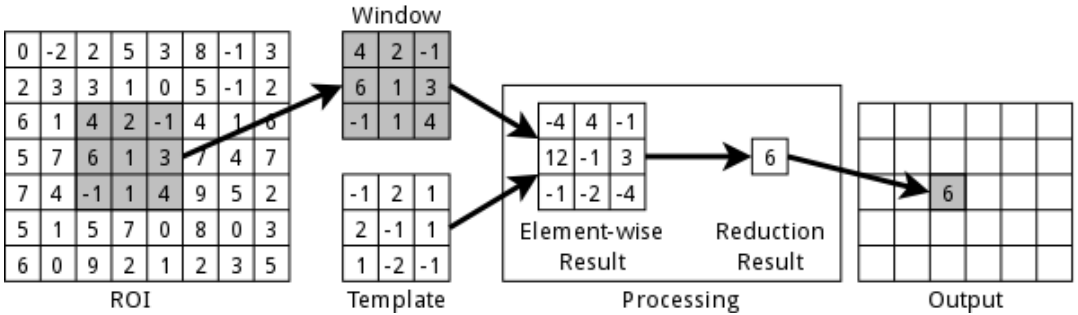
- 1958: David Hubel, Torsten Weisel: neurons in visual cortex have often small local receptive field
  - some reacts only to certain shapes
- Some neurons have larger receptive field
  - they recognize more complex patterns
  - combines outputs of lower-level neurons



hskld

Computer graphics motivation - filters

- for capturing features of an image like edge detection etc.
- multiply each pixel and its neighbours by some matrix

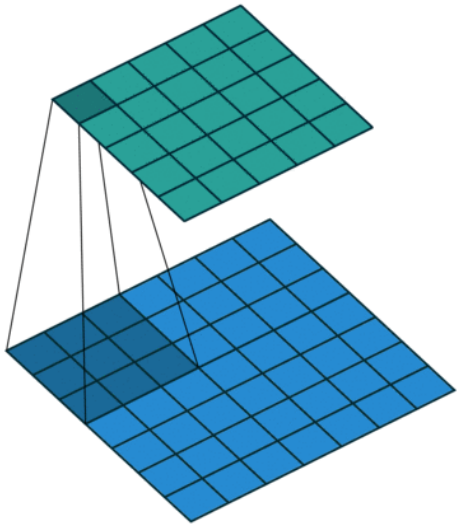


ML motivation - fully-connected networks disadvantages

- not using structure of the input data
  - 2D picture treated as 1D vector
- a lot of parameters to learn
  - especcially for larger input
- locality
  - pattern recognized on a particular place is not recognized elsewhere

Convolutional Layer

- Each neuron is connect only to its receptive field = small rectangle in previous layer

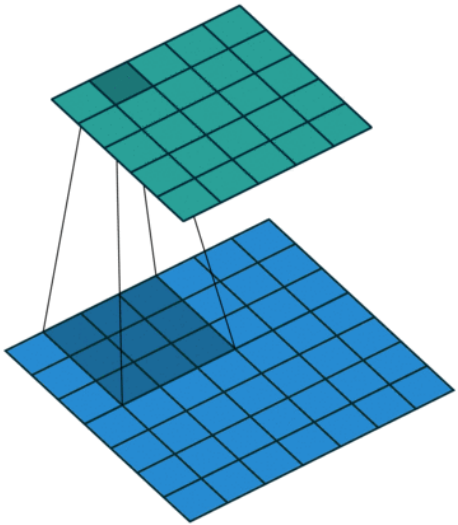


http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

http://index-of.es/Varios2/HandsonMachineLearningwithScikitLearnandTensorflow.pdf

Convolutional Layer

- Each neuron is connect only to its receptive field = small rectangle in previous layer

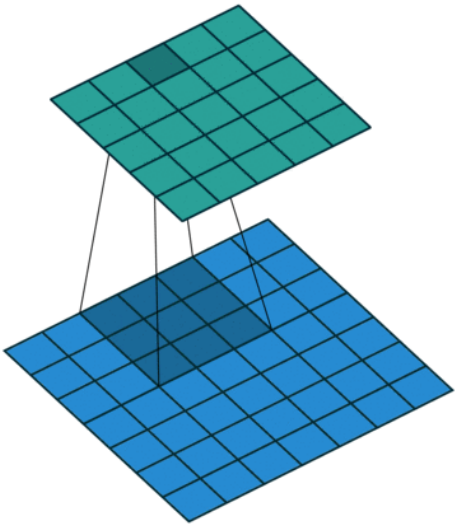


http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

http://index-of.es/Varios2/HandsonMachineLearningwithScikitLearnandTensorflow.pdf

Convolutional Layer

- Each neuron is connect only to its receptive field = small rectangle in previous layer

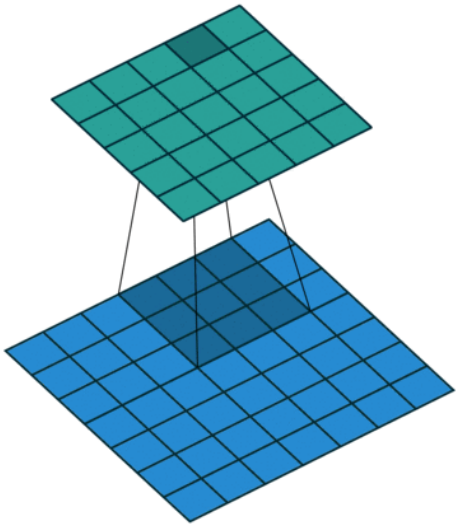


http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

http://index-of.es/Varios2/HandsonMachineLearningwithScikitLearnandTensorflow.pdf

Convolutional Layer

- Each neuron is connect only to its receptive field = small rectangle in previous layer

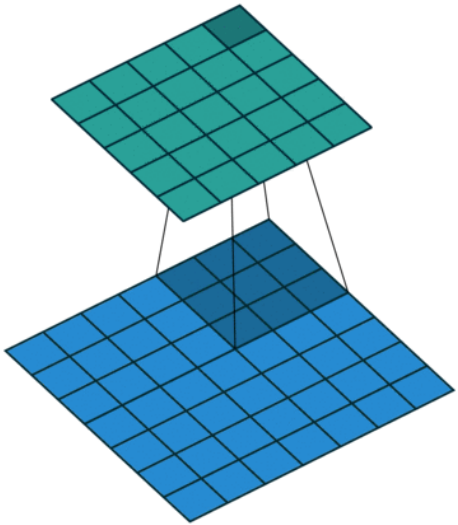


http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

http://index-of.es/Varios2/HandsonMachineLearningwithScikitLearnandTensorflow.pdf

Convolutional Layer

- Each neuron is connect only to its receptive field = small rectangle in previous layer

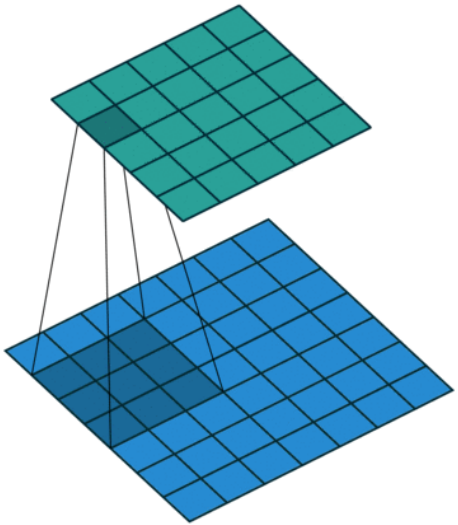


http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

http://index-of.es/Varios2/HandsonMachineLearningwithScikitLearnandTensorflow.pdf

Convolutional Layer

- Each neuron is connect only to its receptive field = small rectangle in previous layer

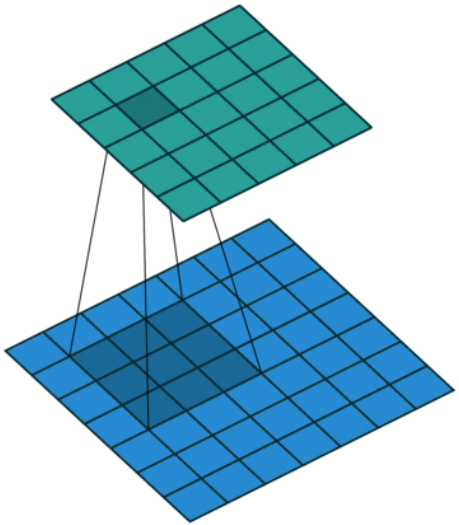


http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

http://index-of.es/Varios2/HandsonMachineLearningwithScikitLearnandTensorflow.pdf

Convolutional Layer

- Each neuron is connect only to its receptive field = small rectangle in previous layer

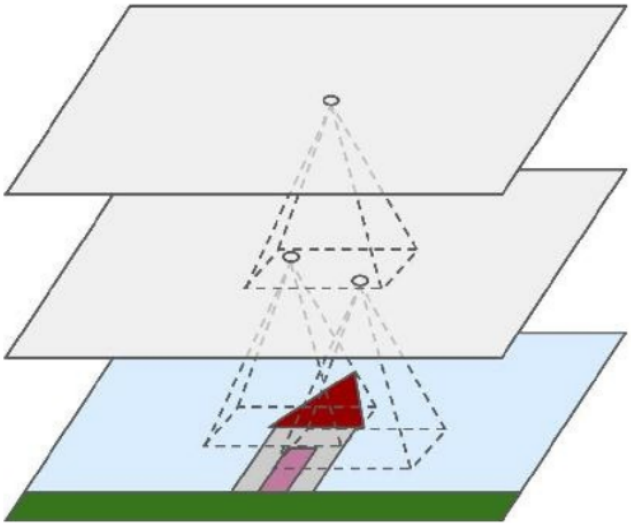


http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

http://index-of.es/Varios2/HandsonMachineLearningwithScikitLearnandTensorflow.pdf

Convolutional Layer

- Each neuron is connect only to its receptive field = small rectangle in previous layer

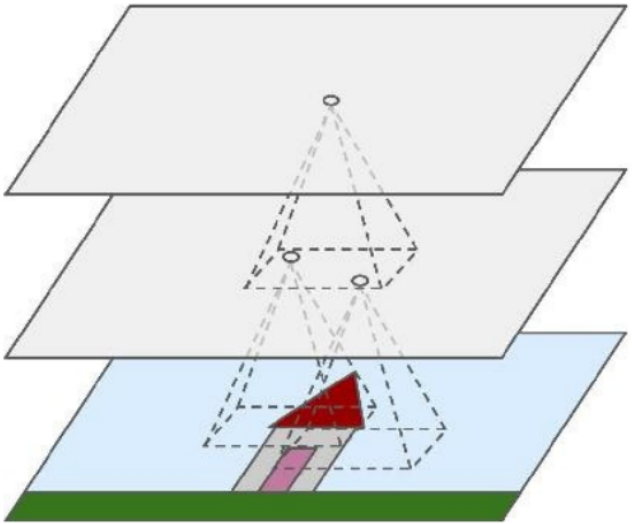


http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

http://index-of.es/Varios2/HandsonMachineLearningwithScikitLearnandTensorflow.pdf

Convolutional Layer

- Each neuron is connect only to its receptive field = small rectangle in previous layer
- 1st layer: low-level features

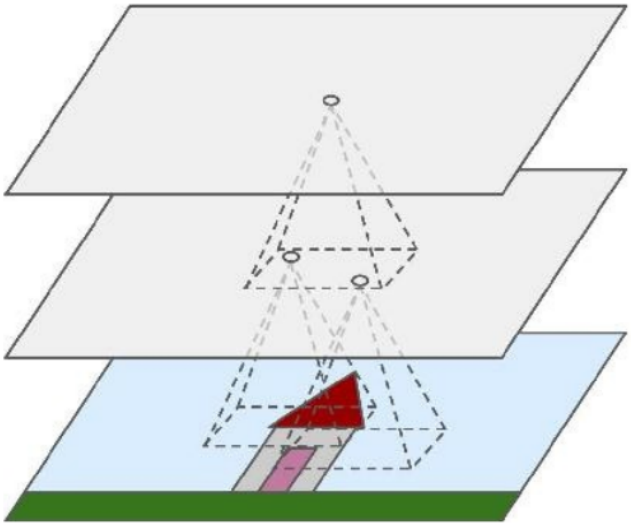


http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

http://index-of.es/Varios2/HandsonMachineLearningwithScikitLearnandTensorflow.pdf

Convolutional Layer

- Each neuron is connect only to its receptive field = small rectangle in previous layer
- 1st layer: low-level features
- 2nd layer: higher-level features
- ...

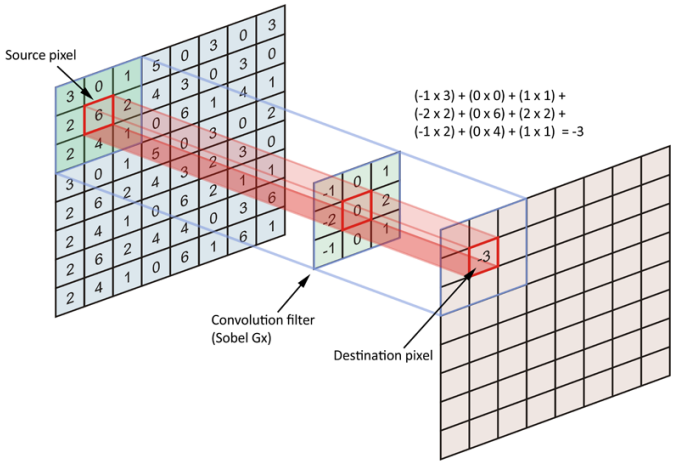


http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

http://index-of.es/Varios2/HandsonMachineLearningwithScikitLearnandTensorflow.pdf

What does neuron do?

- each neuron outputs multiplication of its reception field by a matrix = filter
  - followed by activation function (oftenly ReLU)
- the same filter for all neurons
- filters are learned
- feature map



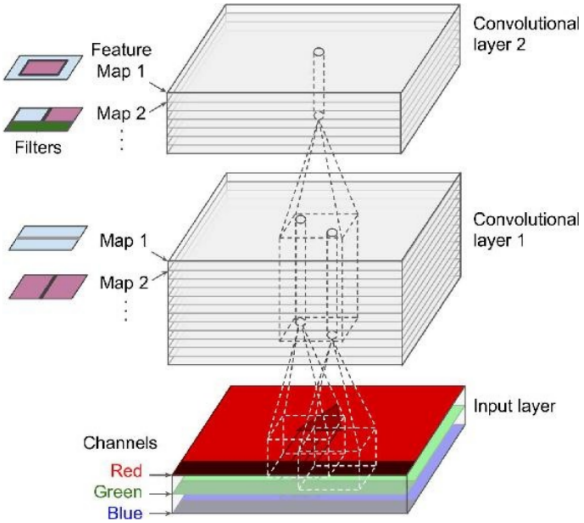
$$z_{i,j} = a\left(b_k + \sum_{u=0}^{w_k}\sum_{v=0}^{h_k} x_{i+u,j+v} \cdot w_{u,v}\right)$$

$z_{i,j}$ = output of filter in position i, j; w_k, h_k = width and height of the filter;
$w_{u,v}$ = weights of filter; $x_{i,j}$ = input from previous layer in position i, j;
b_k = bias term; a = activation function

towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Stacking feature maps

- more filters in each conv. layer
- filter uses all feature maps from previous layer



$z_{i,j,k}$ = output of a filter k in position i, j
w_k, h_k = width and height of the filter
n_{l-1} = num. of f. maps in previous layer
$w_{u,v,k',k}$ = weights of filter k
$x_{i,j,k'}$ = output of a filter k' from previous layer in position i, j
b_k = bias of filter k
a = activation function

$$z_{i,j,k} = a\left(b_k + \sum_{u=0}^{w_k}\sum_{v=0}^{h_k}\sum_{k'=0}^{n_{l-1}} x_{i+u,j+v,k'} \cdot w_{u,v,k',k}\right)$$

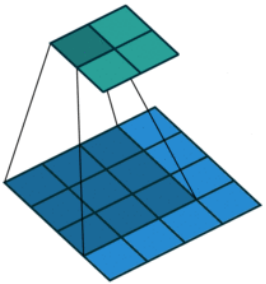
http://index-of.es/Varios2/HandsonMachineLearningwithScikitLearnandTensorflow.pdf

Padding

How to behave on the border?

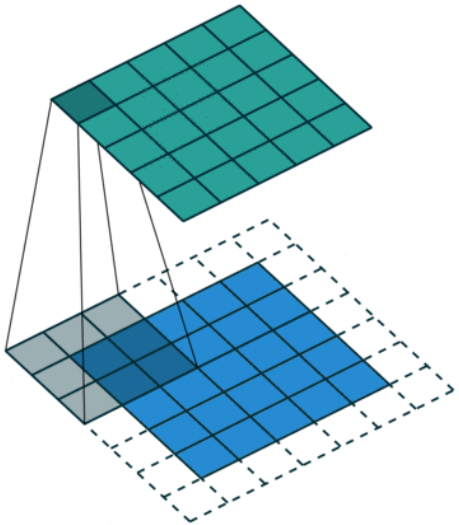
VALID

- new layer slightly smaller
- less common



SAME

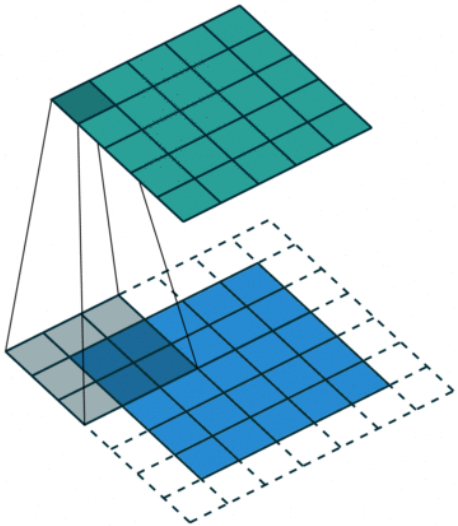
- new layer of the same size
- add zeros around



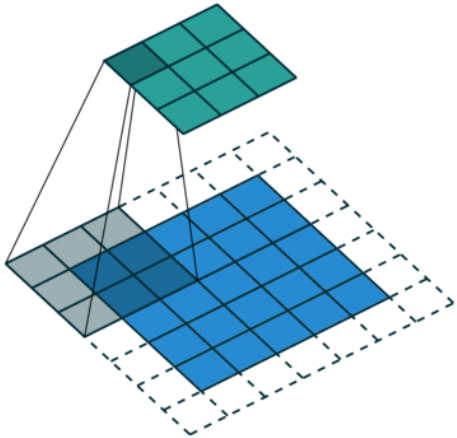
http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

Strides

strides = [1, 1, 1, 1]



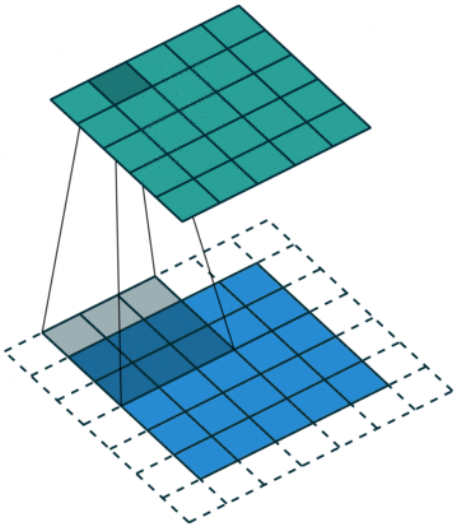
strides = [1, 2, 2, 1]



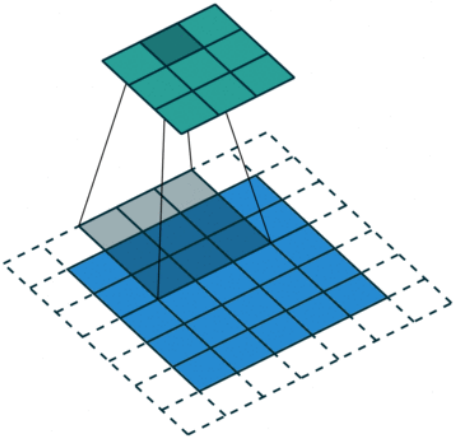
http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

Strides

strides = [1, 1, 1, 1]



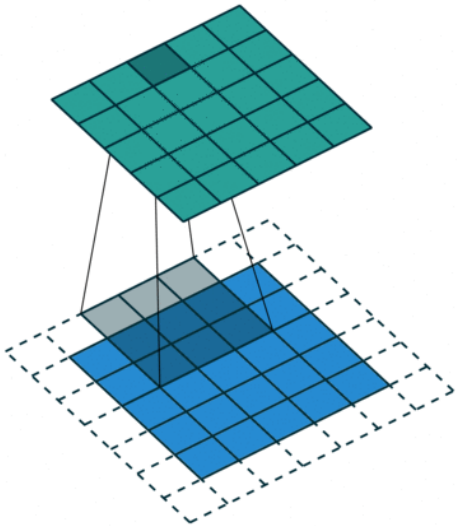
strides = [1, 2, 2, 1]



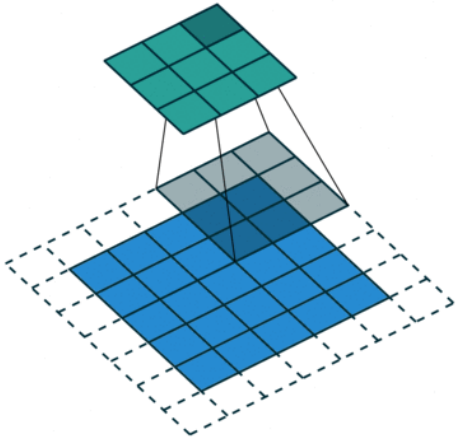
http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

Strides

strides = [1, 1, 1, 1]



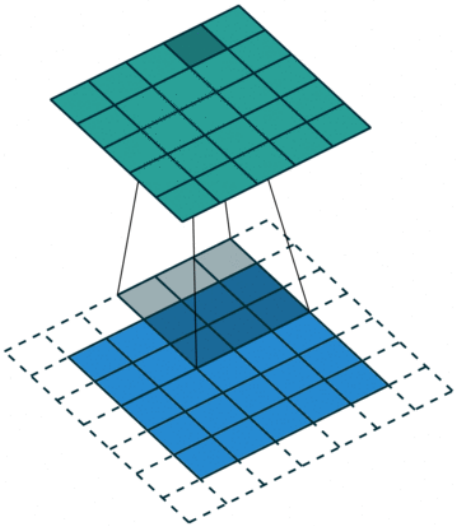
strides = [1, 2, 2, 1]



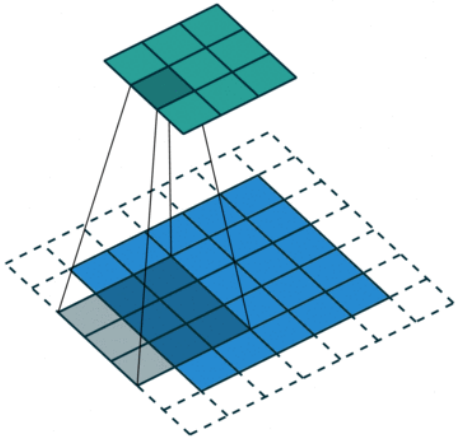
http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

Strides

strides = [1, 1, 1, 1]



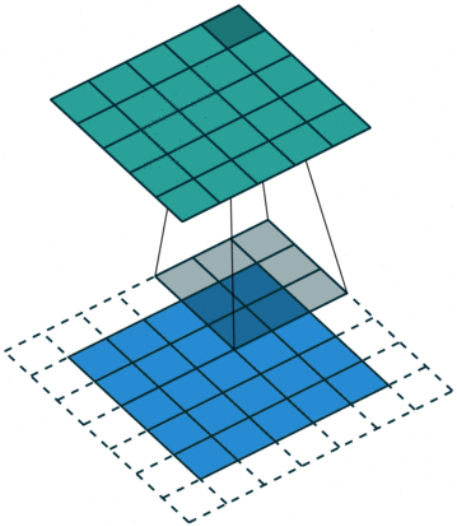
strides = [1, 2, 2, 1]



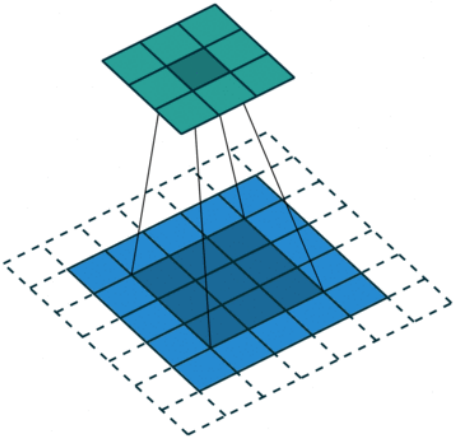
http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

Strides

strides = [1, 1, 1, 1]



strides = [1, 2, 2, 1]



http://www.cvc.uab.es/people/joans/slides_tensorflow/tensorflow_html/layers.html

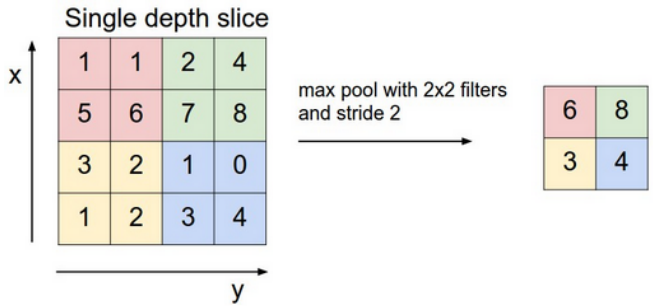
Convolutional Layer - Tensorflow overview

```
c_filters = 32   # number of filters
c_kernel_size = 3   # for square kernel 3x3
c_strides = [1,2,2,1]   # [mini-batch size, height,
                          width, depth in feature maps]
c_pad = "SAME"   # SAME or VALID
c_act = tf.nn.relu   # activation function
```

```
conv_layer = tf.layers.conv2d(previous_layer,
        kernel_size = c_kernel_size, strides = c_strides,
        padding = c_pad, activation = c_act)
```

Pooling layer

- shrinking image - either taking maximum or average
- reduction of computational load (memory)
- reduction of parameters (overfitting)

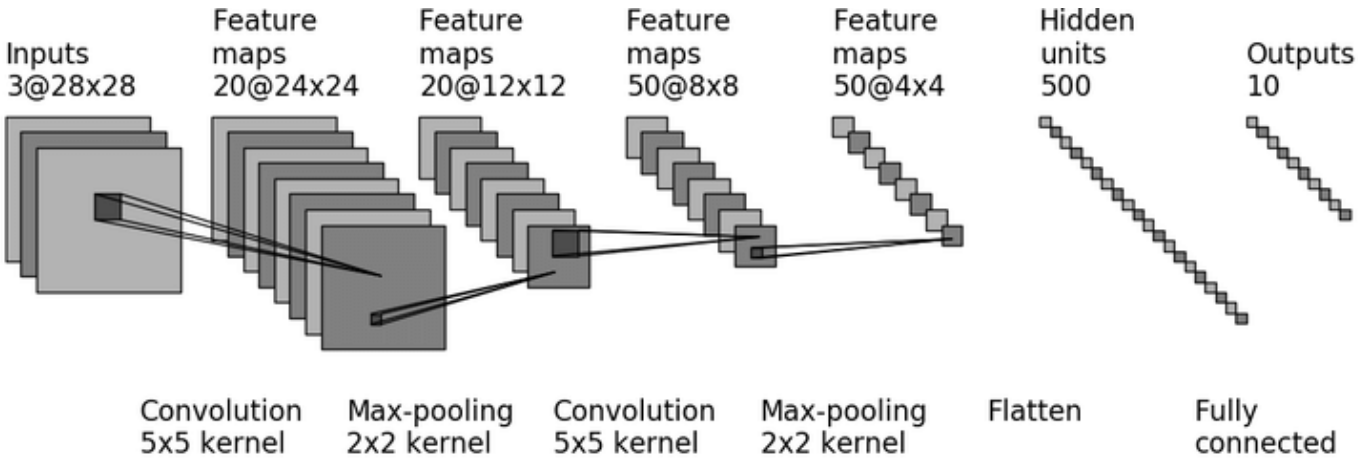



```
pool_layer = tf.nn.max_pool(input_layer,
                            ksize=[1, 2, 2, 1],
                            strides=[1, 2, 2, 1],
                            padding="VALID")
```

https://skymind.ai/wiki/convolutional-network

CNN Architecture

Le-Net5 (1998):



https://www.researchgate.net/publication/312170477_On_Classification_of_Distorted_Images_with_Deep_Convolutional_Neural_Networks

CNN compared to fully-connected networks

- smaller number of parameters (filters need a lot of memory but they share parameters)
- better results than fully-connected networks
- much more hyperparameters (different architectures + hyperparameters for each convolutional layer)