inovex

Convolutional Neural Networks

Convolutional Neural Networks

Overview

› special architecture of deep neural networks

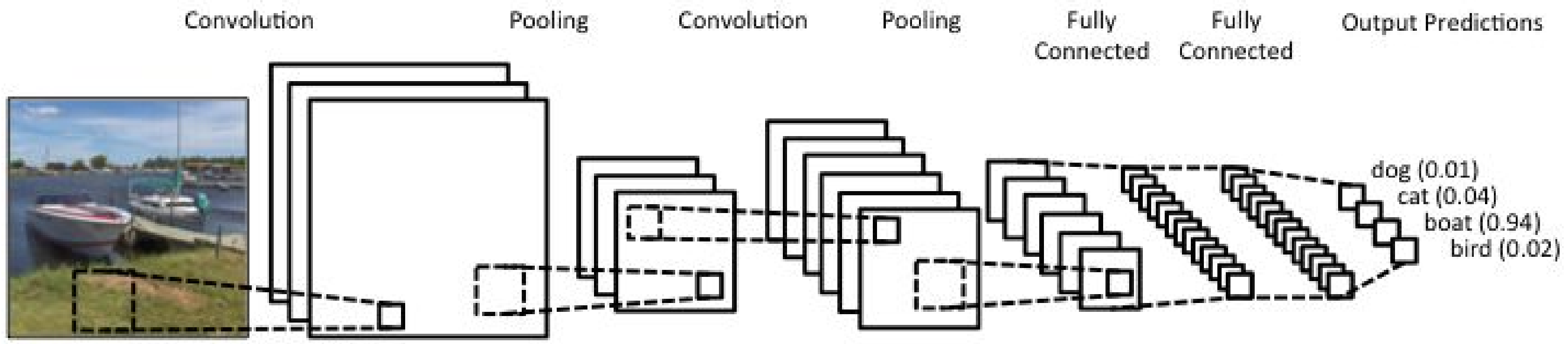
› Computer Vision

› automated feature extraction

› classification of images through automatically extracted features

Convolutional Neural Networks

Overview



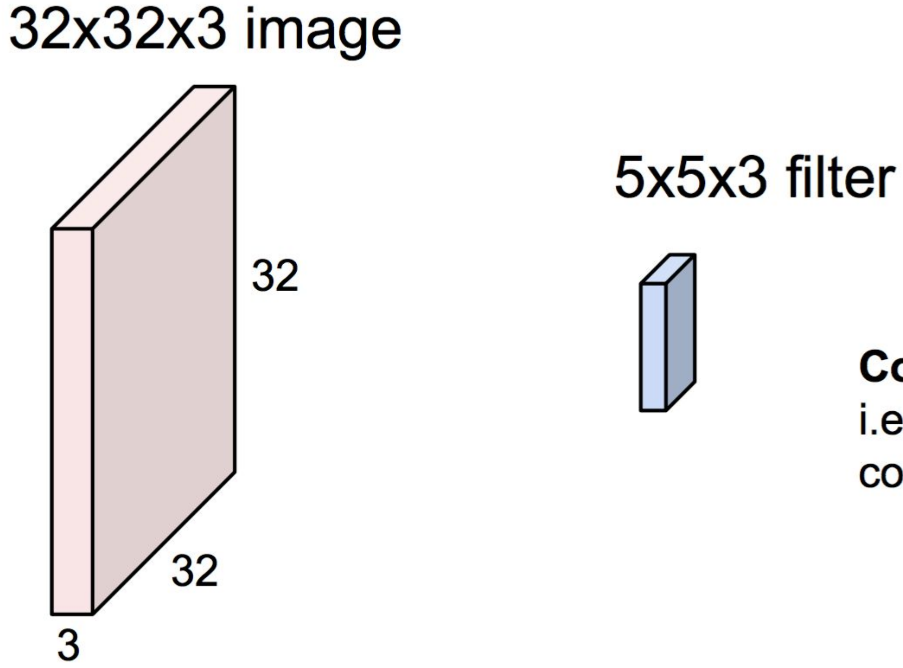
Convolutional Neural Networks

Overview

› Convolutional Layer	$\rightarrow$	Feature extraction
› Pooling Layer	$\rightarrow$	Dimensionality reduction
› Fully Connected	$\rightarrow$	Classification

Convolutional Neural Networks

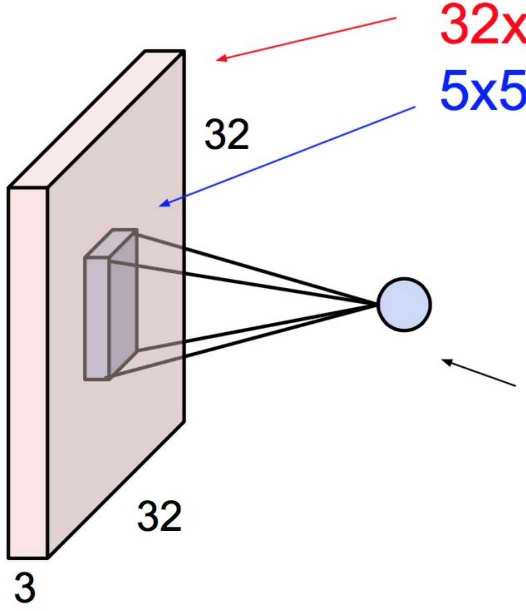
Convolutional Layer



Convolve the filter with the image i.e. "slide over the image spatially, computing dot products"

Convolutional Neural Networks

Convolutional Layer



32x32x3 image

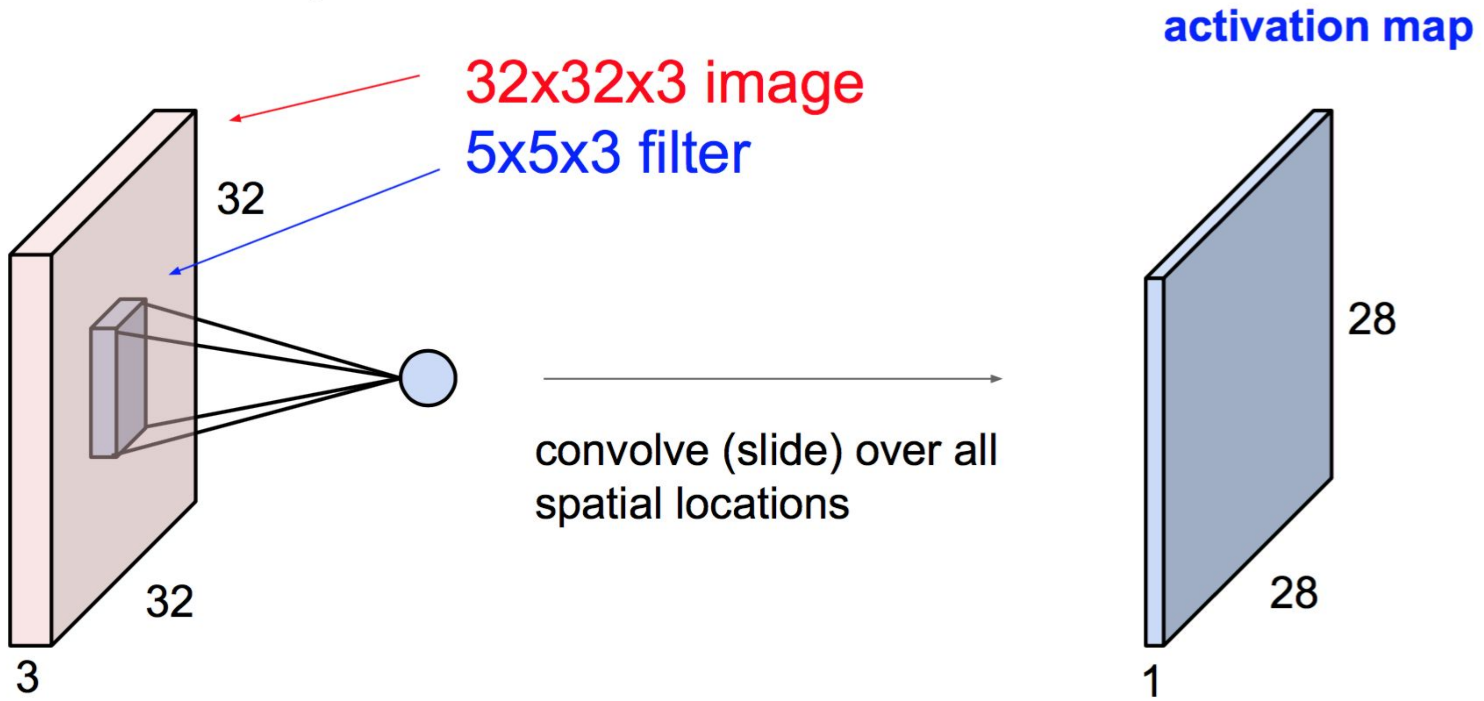
5x5x3 filter w

1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

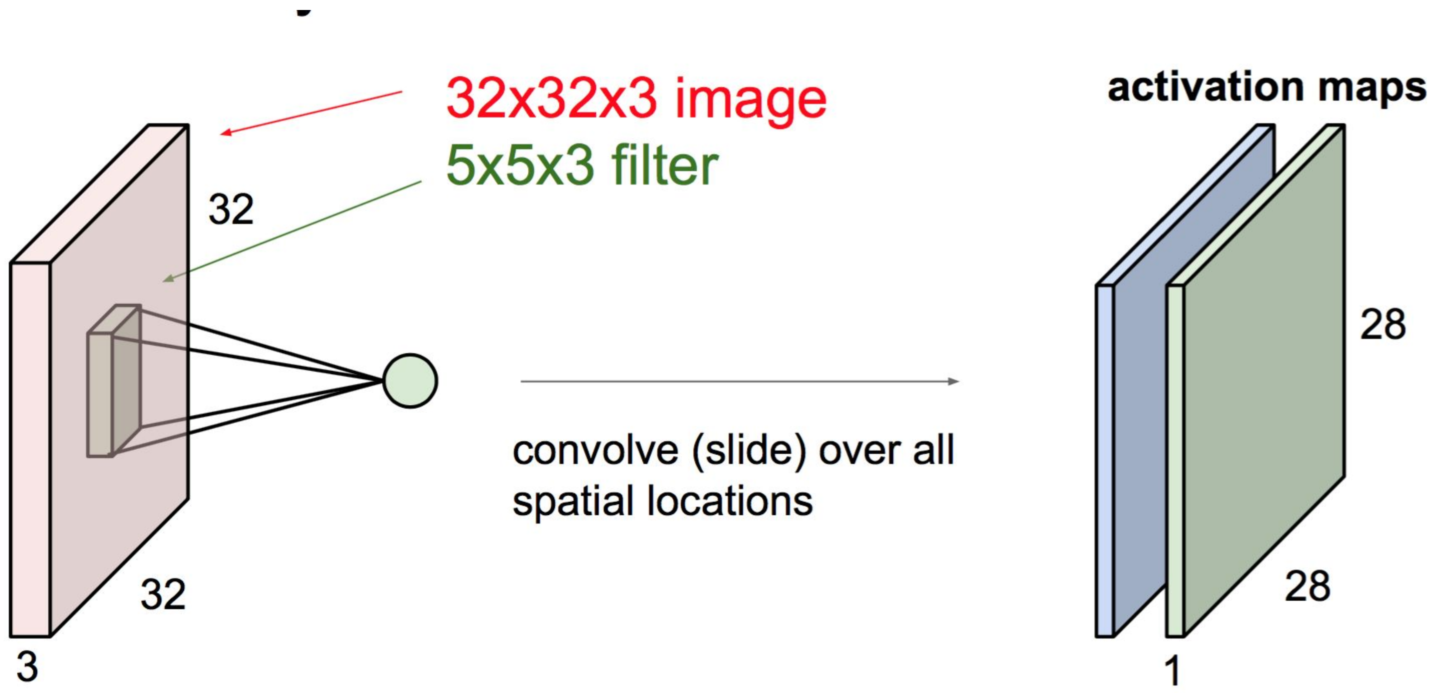
Convolutional Neural Networks

Convolutional Layer



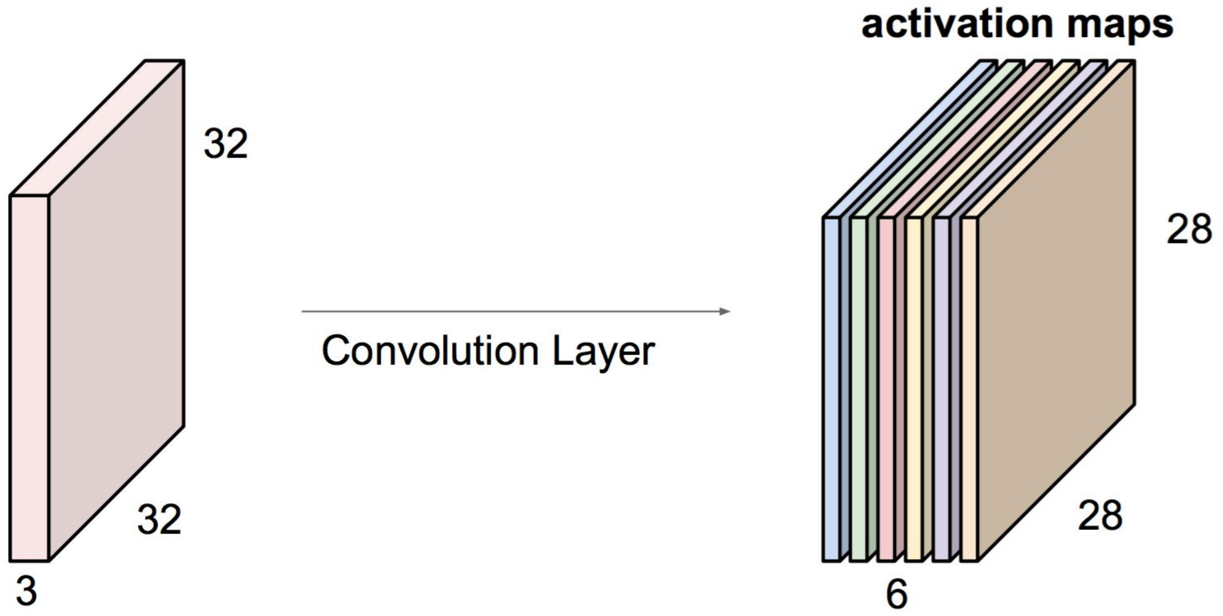
Convolutional Neural Networks

Convolutional Layer



Convolutional Neural Networks

Convolutional Layer

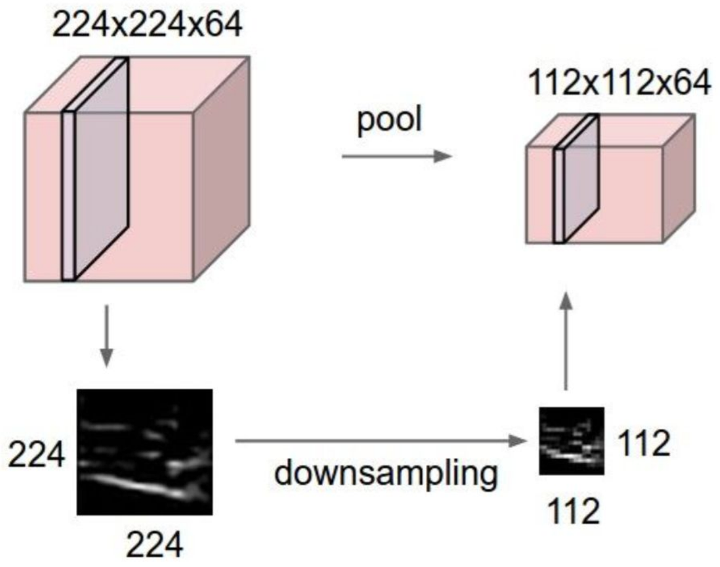


We stack these up to get a “new image” of size 28x28x6!

Convolutional Neural Networks

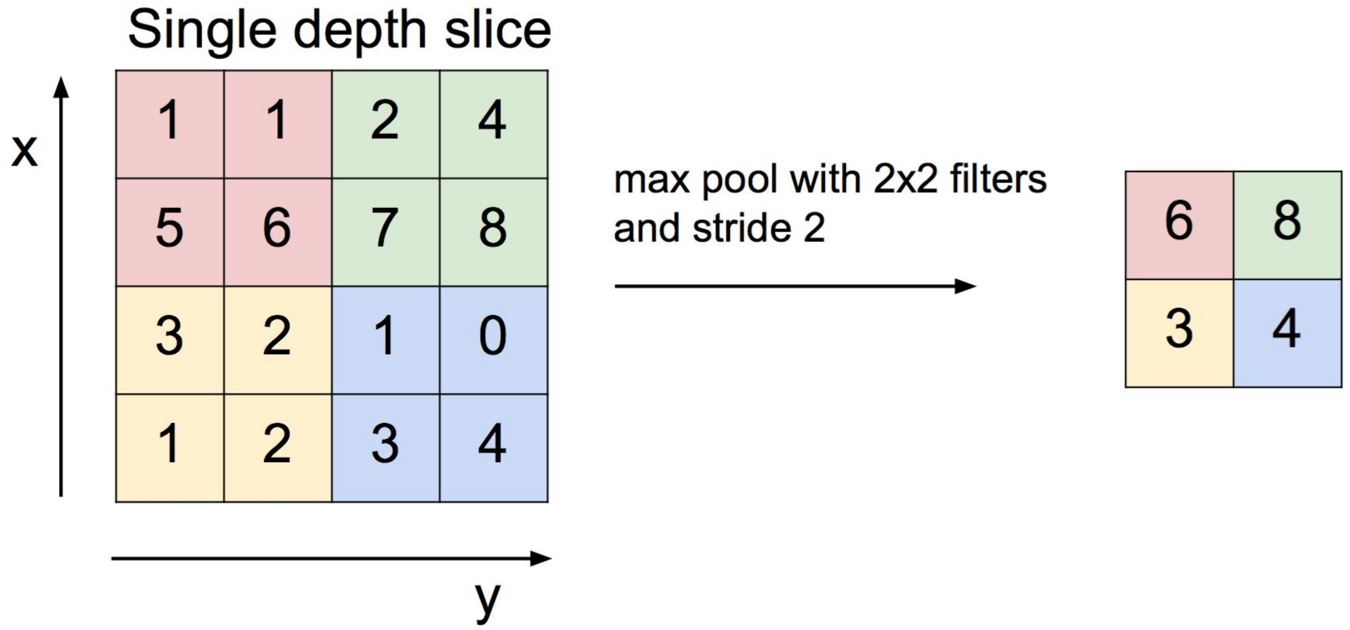
Pooling Layer

- makes the representations smaller and more manageable
- operates over each activation map independently:



Convolutional Neural Networks

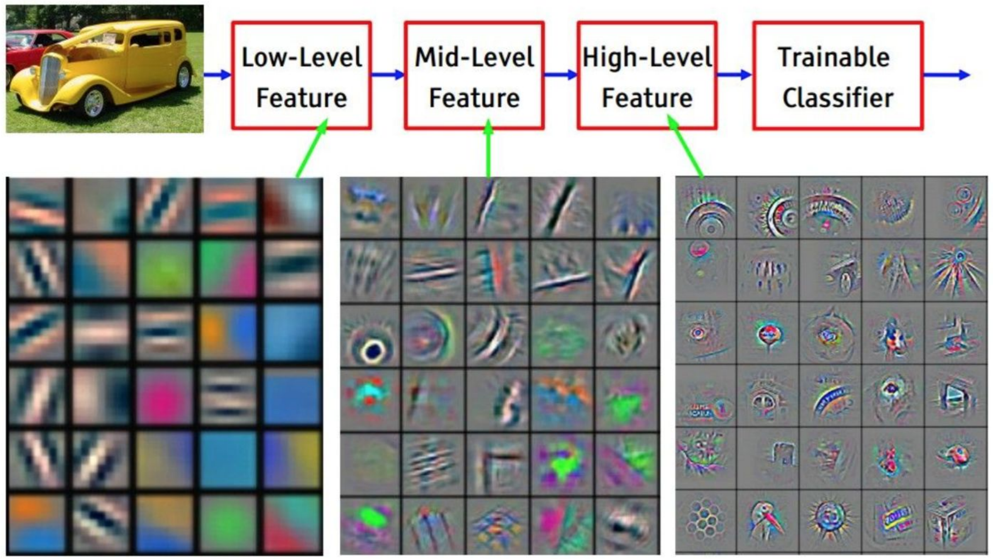
Pooling Layer



Convolutional Neural Networks

Feature extraction

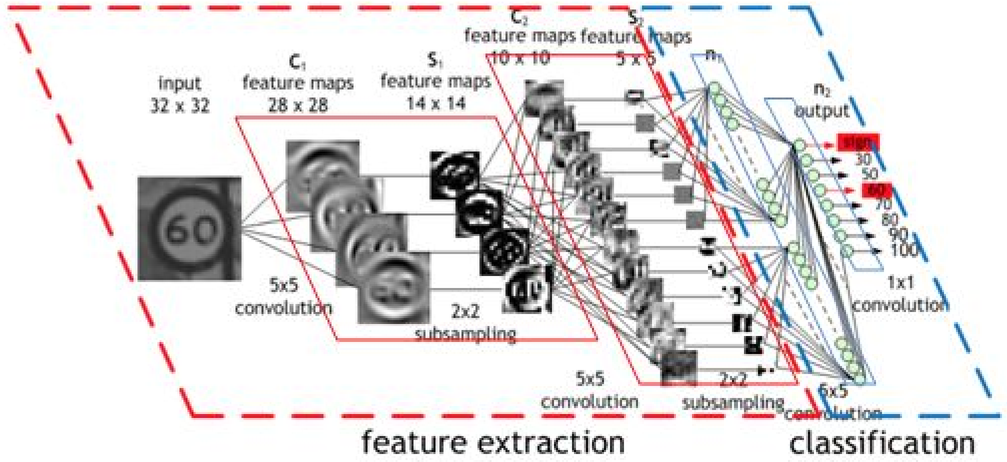
› stack multiple Convolutional and Pooling Layer together



Convolutional Neural Networks

Fully Connected

- normal neural network
- classification through extracted features



Convolutional Neural Networks

Fully Connected

- AlexNet
  - 224x224x3 images as input
  - 150.528 input values
  - feature extraction → 4096 important values

Convolutional Neural Networks

CNN in TensorFlow

```
# convolutional layer

conv = tf.layers.conv2d(input,32,[3,3],padding="same",activation=tf.nn.relu)


# max-pooling layer

max_pool = tf.layers.max_pooling2d(input,[2,2],[2,2])


#  fully connected

fc = tf.layers.dense(input,1024,activation=tf.nn.relu)
```

Convolutional Neural Networks

CNN in TensorFlow

let's recognize some handwritten digits…

03_cnn_mnist.ipynb

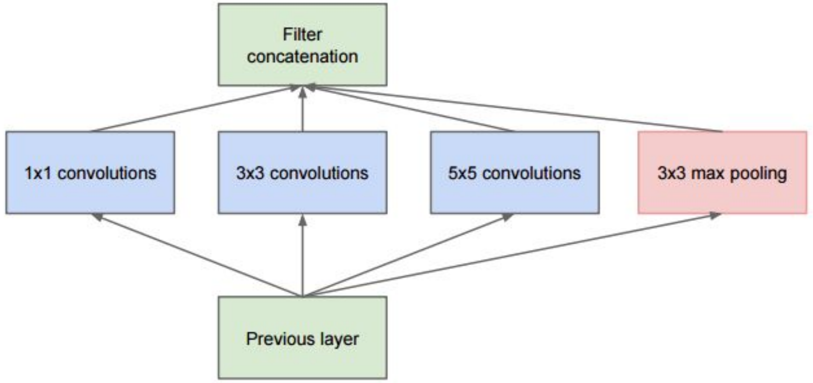
Convolutional Neural Networks

Inception

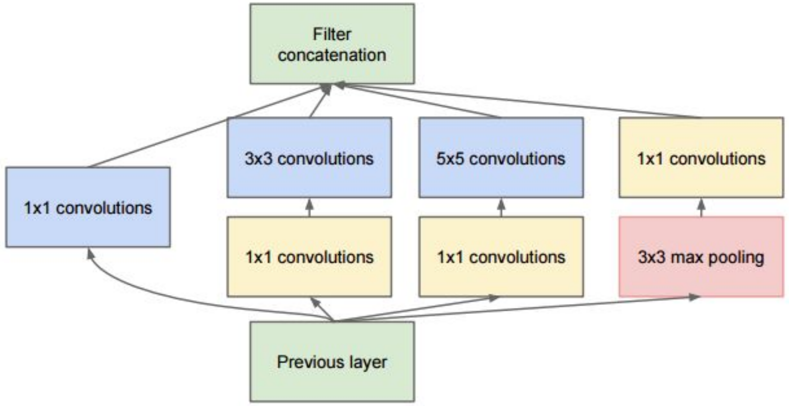
- special architecture of very deep CNN
  - very efficient training
  - high performance
- inception modules $\rightarrow$ parallel convolutions
- ILSVRC 2012: Inception-ResNet-V2 $\rightarrow$ 95,3 % Accuracy

Convolutional Neural Networks

Inception



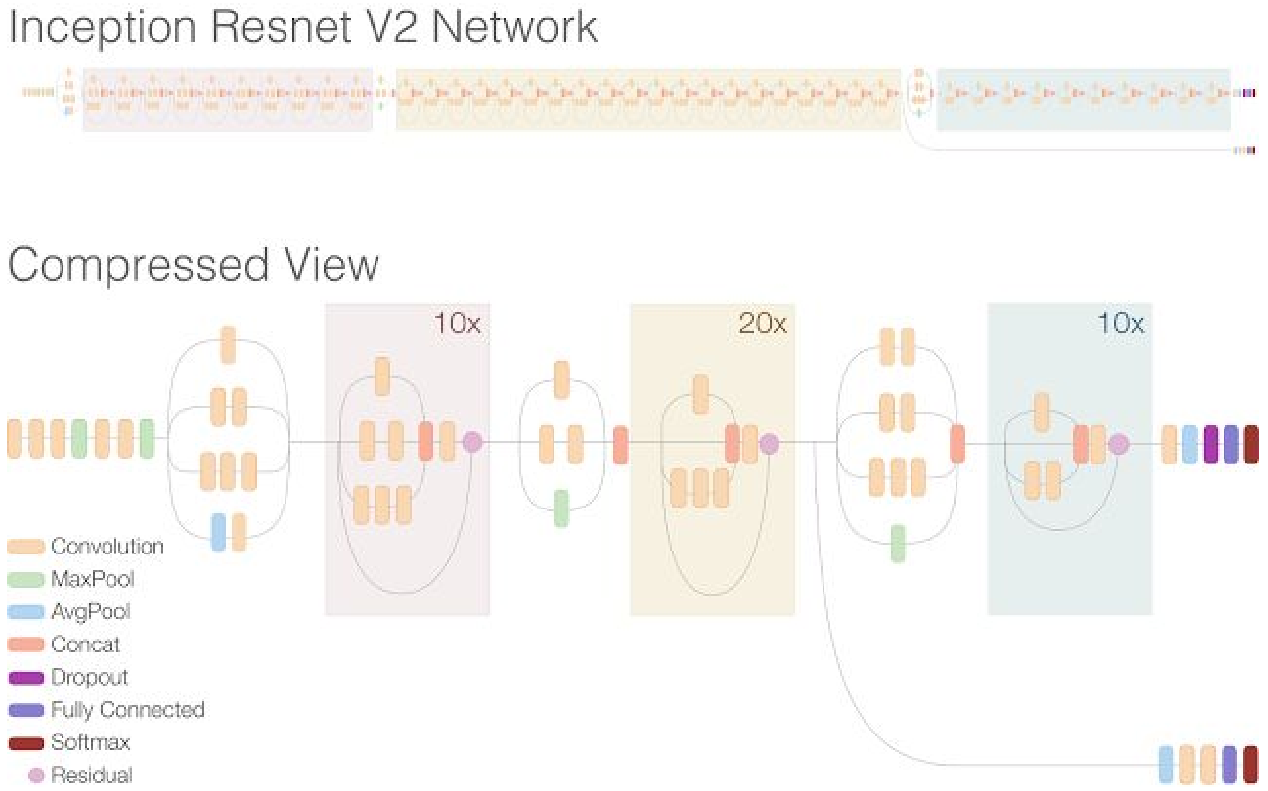
(a) Inception module, naïve version



(b) Inception module with dimension reductions

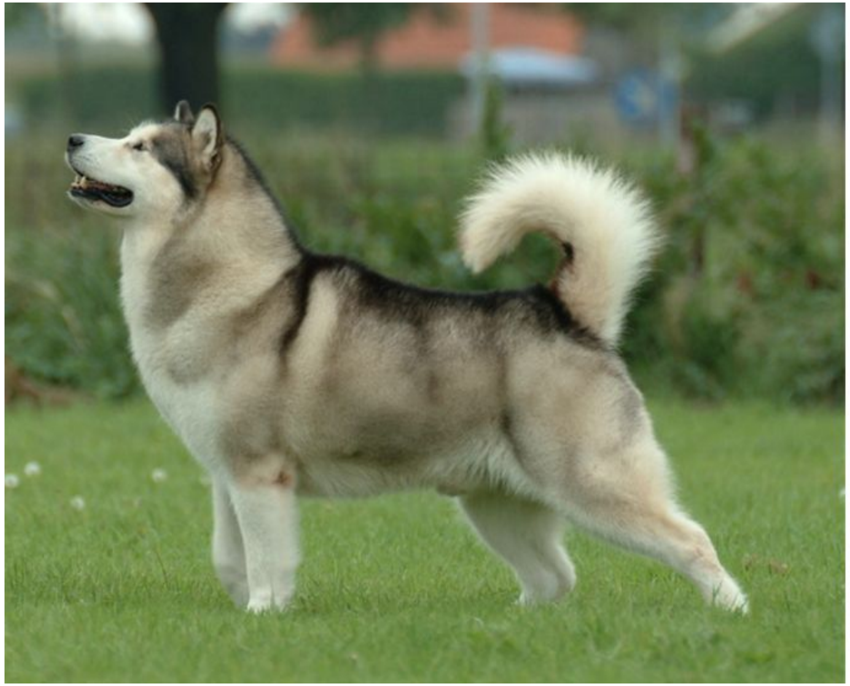
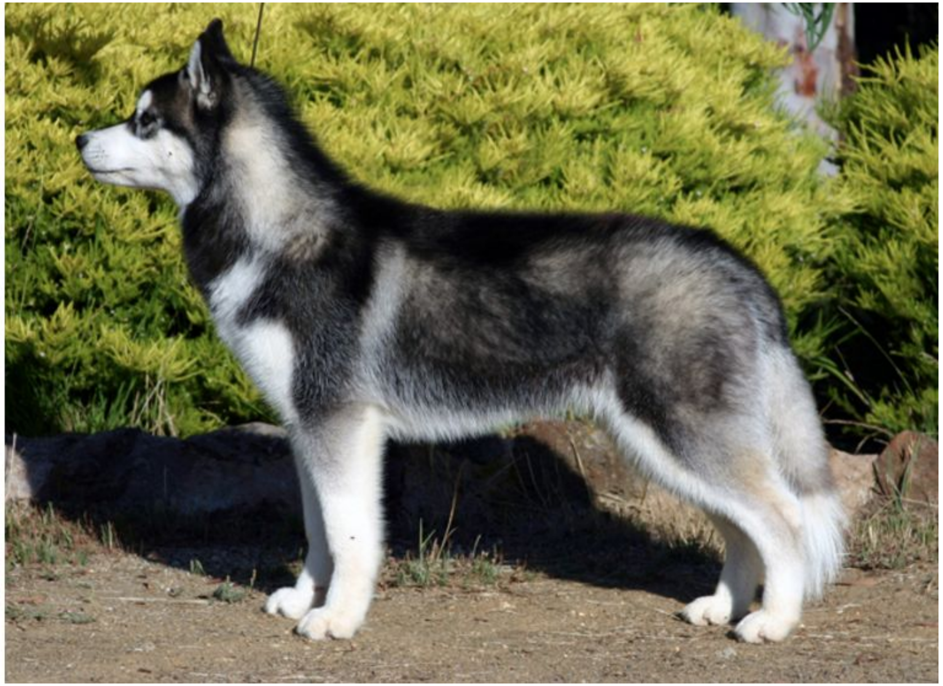
Convolutional Neural Networks

Inception-ResNet-v2



Convolutional Neural Networks

Inception-ResNet-v2

Convolutional Neural Networks

CNN Model Zoo

- state-of-the-art CNN
- implemented in TF-slim $\rightarrow$ lightweight high-level API of TensorFlow
- simple training and retraining
- https://github.com/tensorflow/models/tree/master/slim