Going Deeper with GoogLeNet and CaffeJS

“Deep Learning in the Browser”

Christoph Körner

Slides available on bit.ly/2cwhdyG

About me

- Visual Computing at Vienna University of Technology
- Data Scientist at T-Mobile Austria
- Author of Data Visualizations with D3 and AngularJS
- Author of Learning Responsive Data Visualization
- Contributor at n3-line-chart
- Organizer of Vienna Kaggle Meetup
- LinkedIn: at.linkedin.com/in/christophkoerner
- Twitter: @ChrisiKrnr
- Google+: +ChrisiHififm
- Github: github.com/chaosmail

Agenda

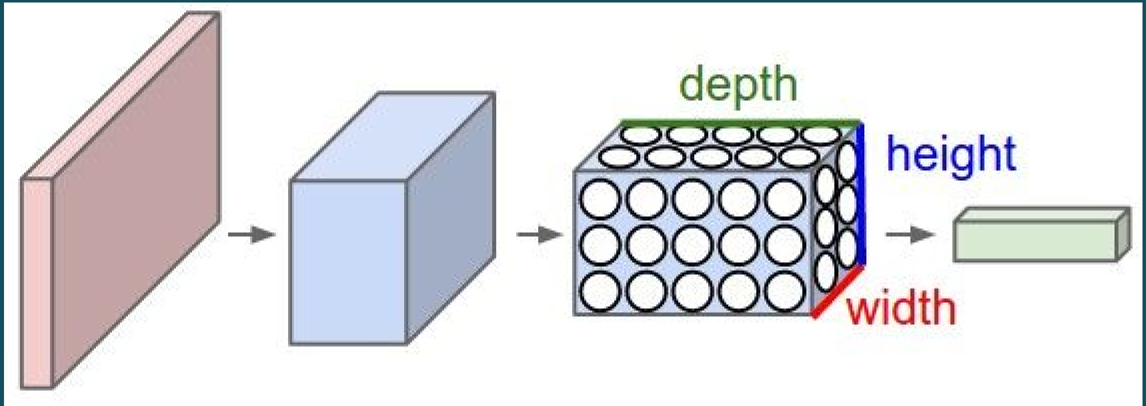
- Deep Learning Refresher
- GoogLeNet
- CaffeJS - Deep Learning in the browser
- Whats next

Deep Learning Refresher

Ingredients for Deep Classif.

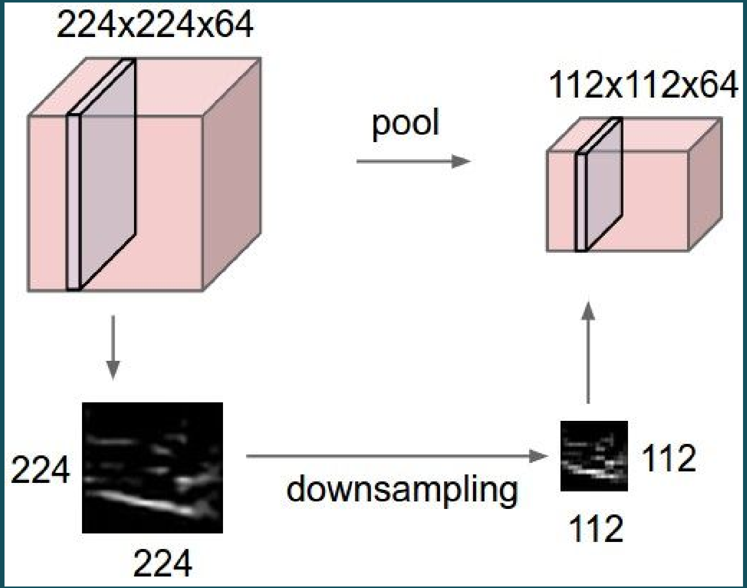
- Convolutions + Nonlinearities
- Pooling
- Fully Connected layers
- Softmax
- Many layers and parameters

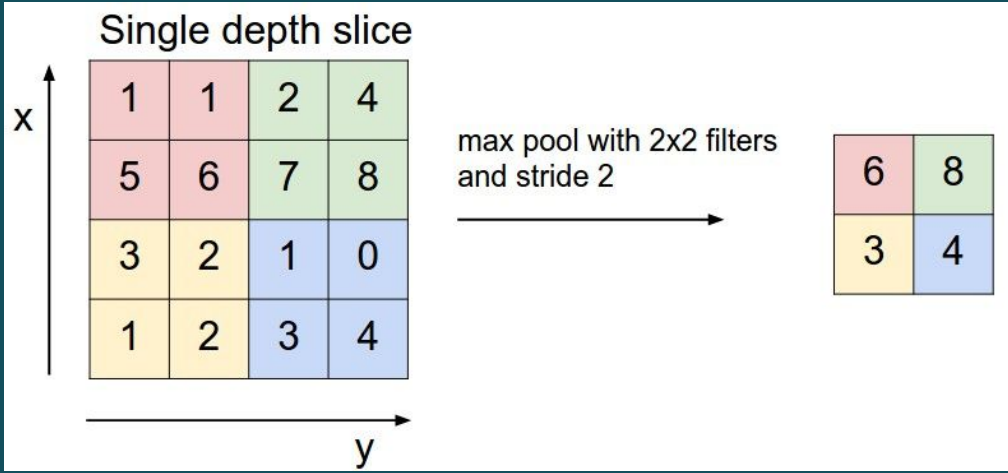
Convolutions



Source: CS231n Github

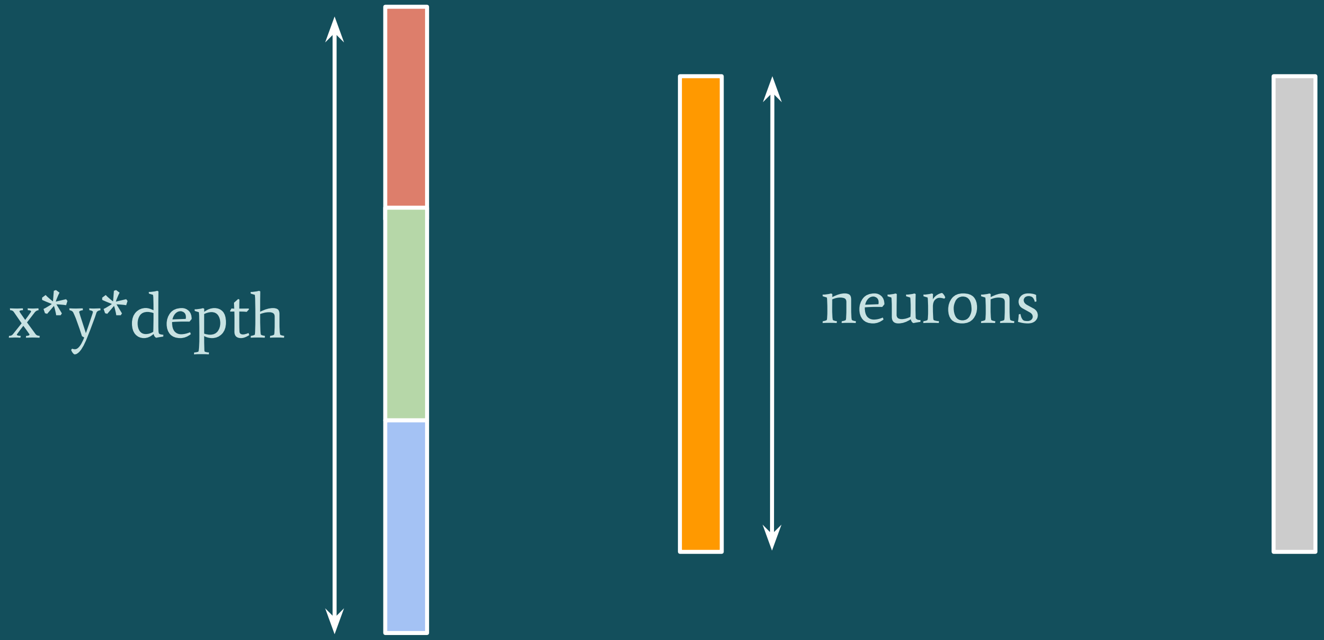
Pooling



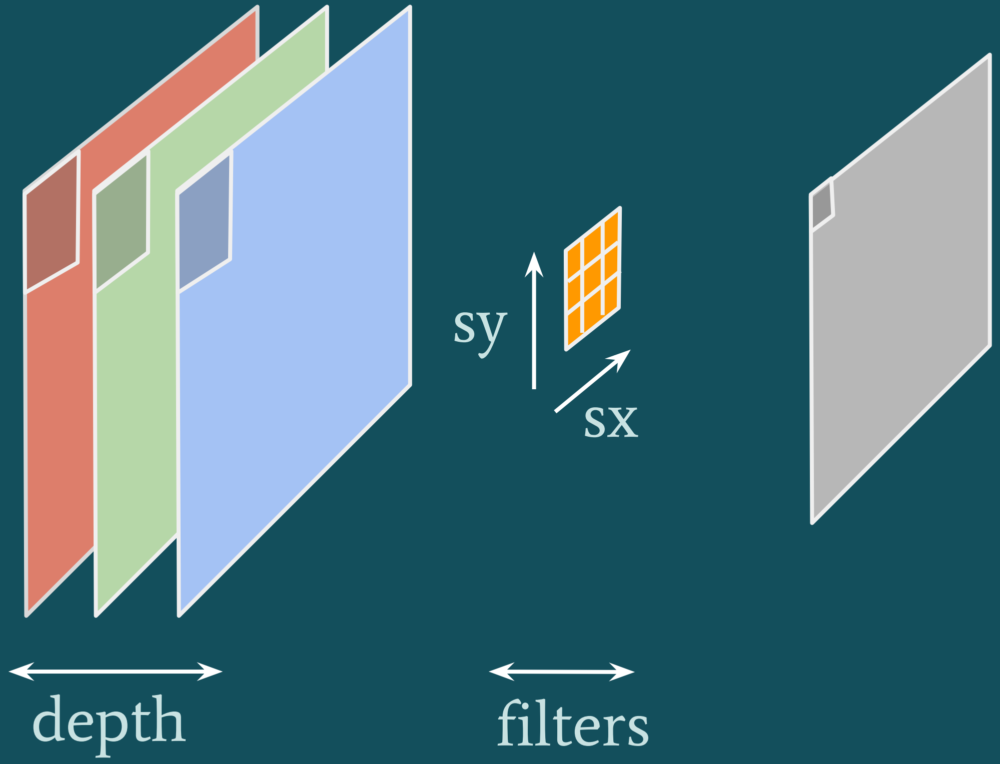


Source: CS231n Github

Parameters in Dense Layers

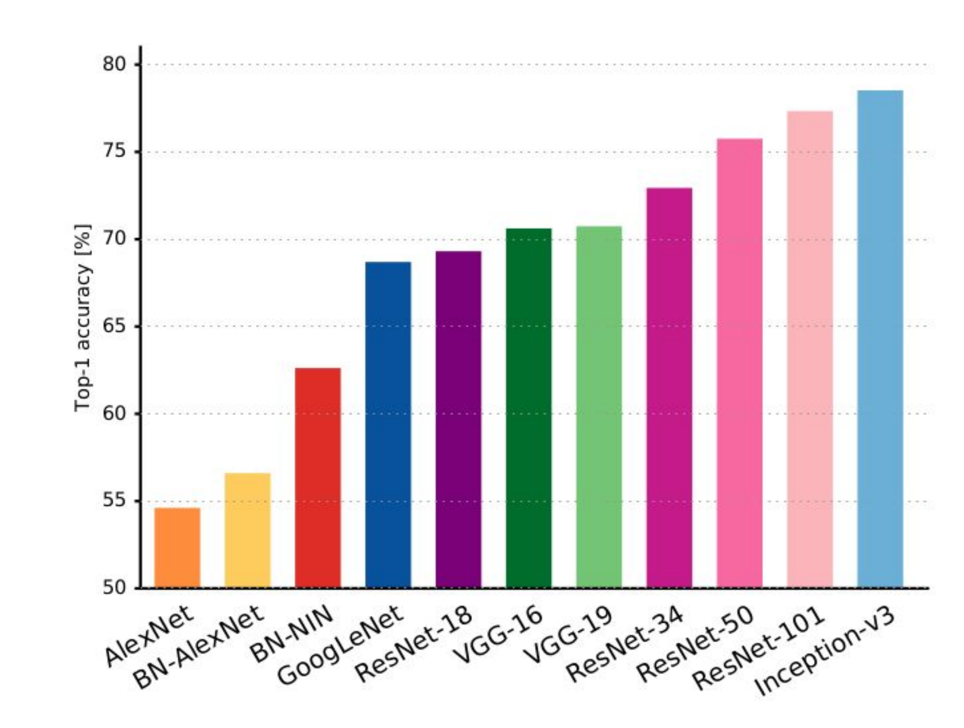


Parameters in Convolutions



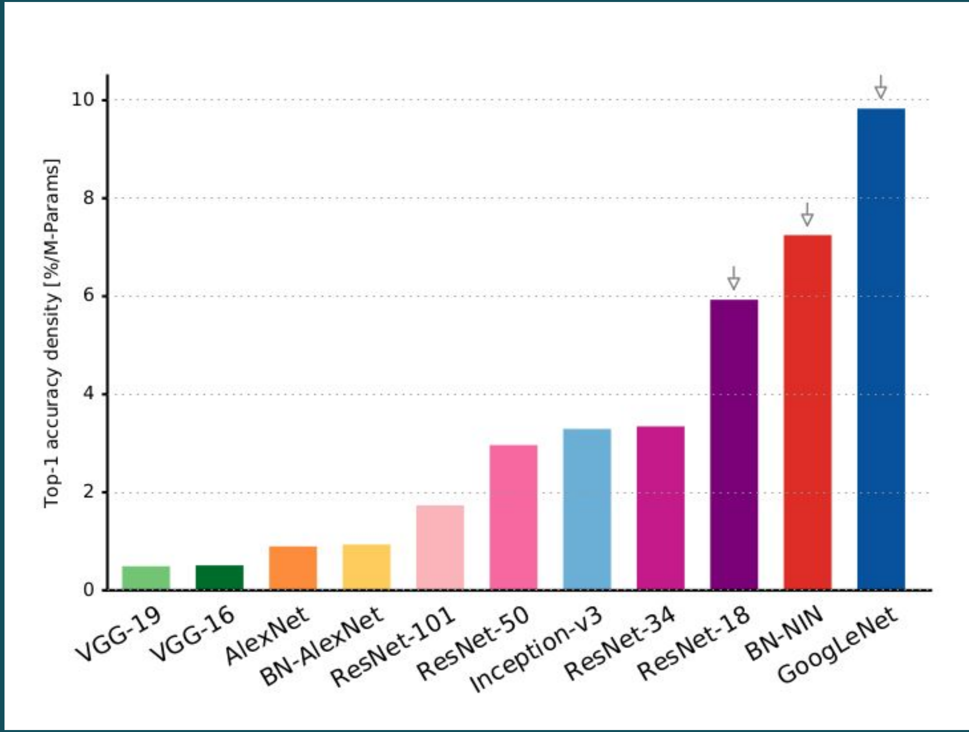
Deep Learning Architectures

Top-1 accuracy



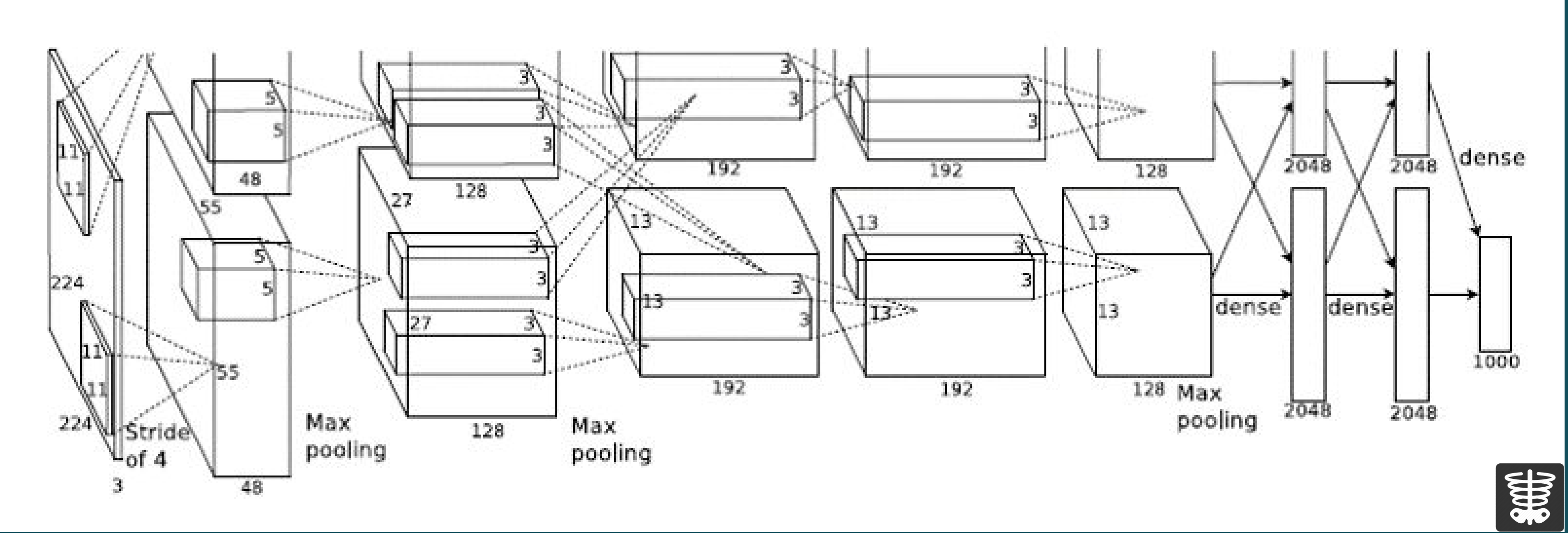
Source: An Analysis of Deep Neural Network Models

Top-1 accuracy density



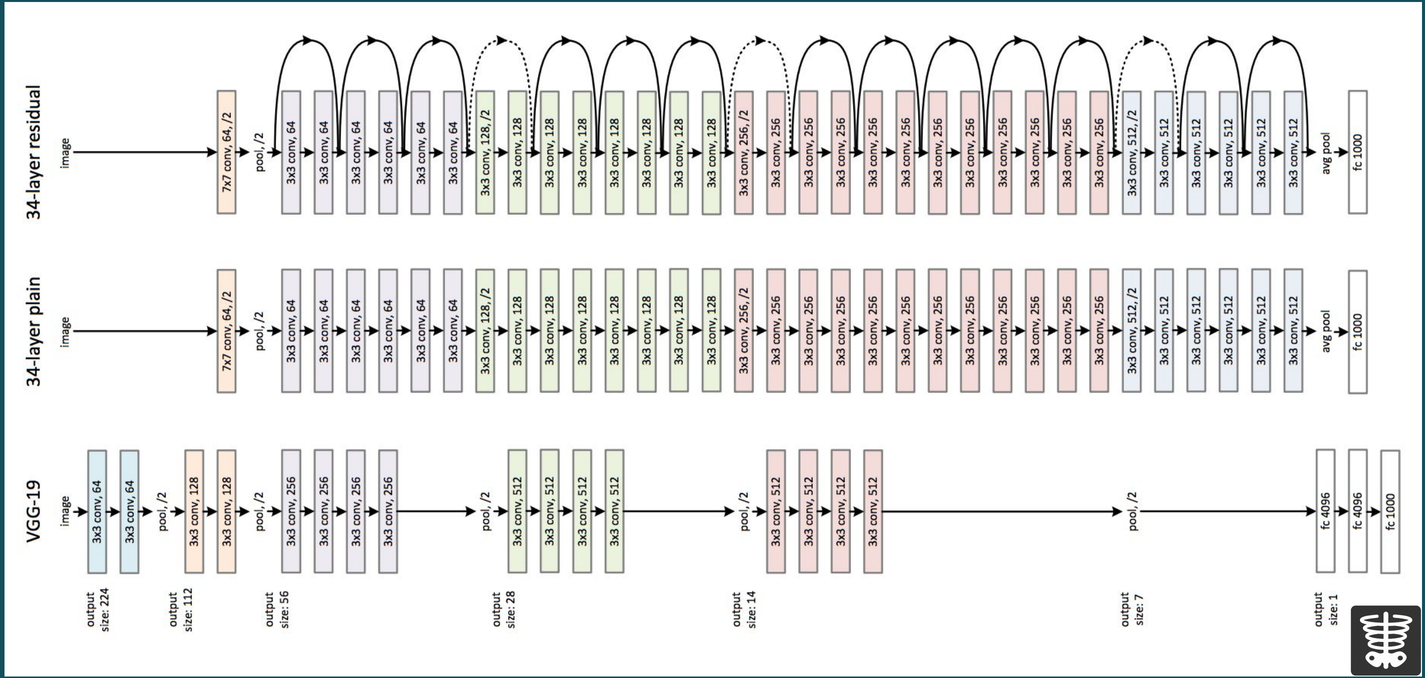
Source: [An Analysis of Deep Neural Network Models]

AlexNet: 62,378,344 params (250MB)



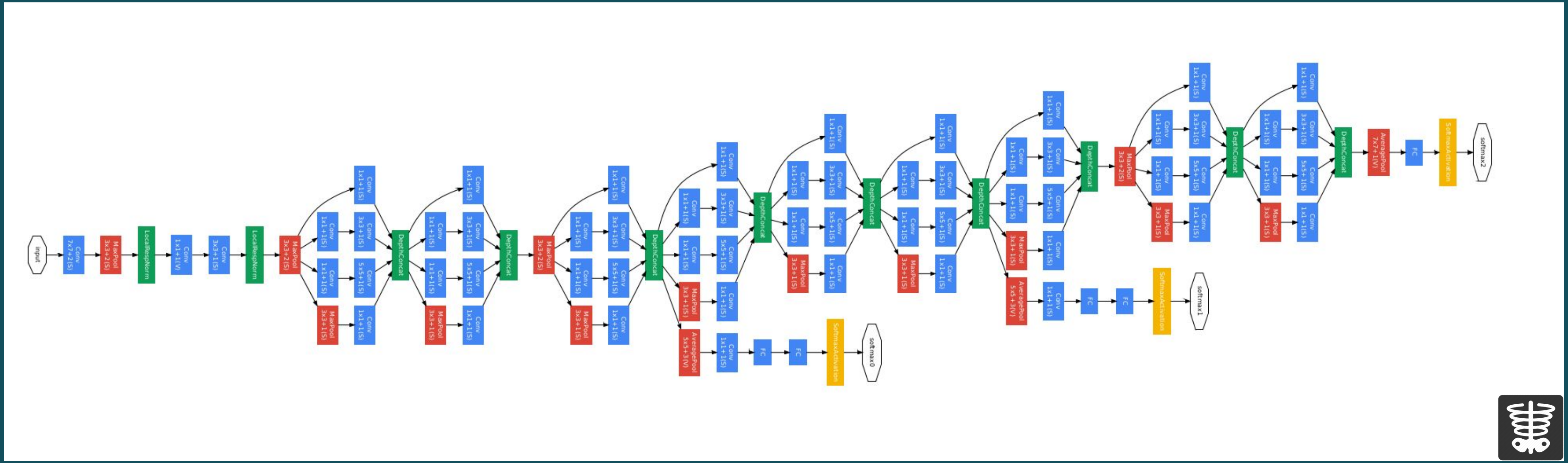
Source: [ImageNet Classification with Deep Convolutional Neural Networks]

VGG (~400MB) and ResNet (~230MB)



Source: [Deep Residual Learning for Image Recognition](), [ResNet]()

GoogLeNet: 6,998,552 params (28MB)



Source: Going deeper with convolutions, Great Slides

GoogLeNet

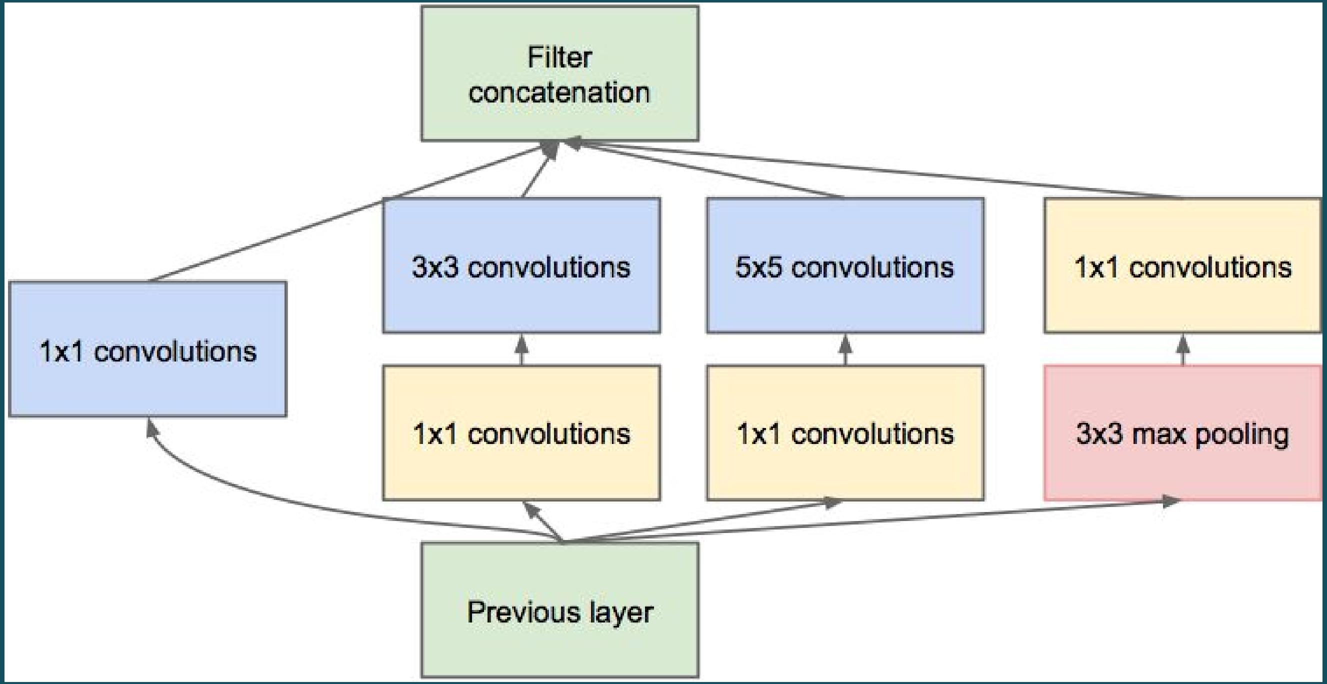
“moving from fully connected to sparsely connected architectures”

Source: Going Deeper with Convolutions, Princeton 2015

GoogLeNet

- ~10 times less parameter than AlexNet
- ~15% higher classification accuracy
- ~30MB weights in memory
- ~1.5 billion MAC operations (FP)

Inception Module (Network in a Network)



Source: [Princeton 2015]

Ingredients for Inception Module

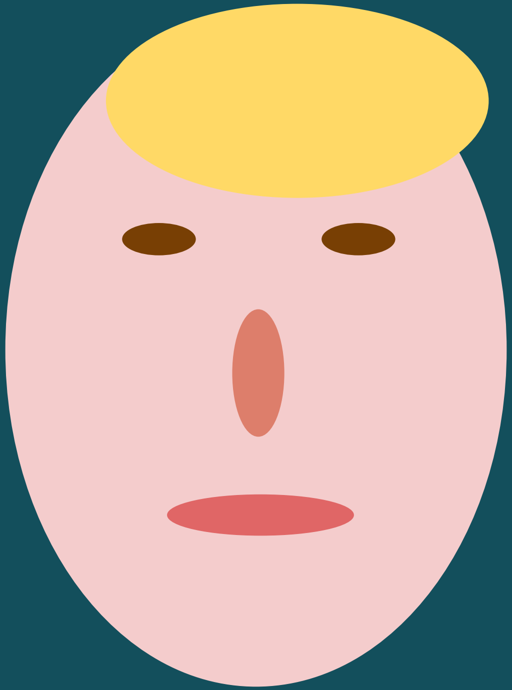
- Hebbian Principle
- Bottleneck Convolutions

Ingredient 1: Hebbian Principle

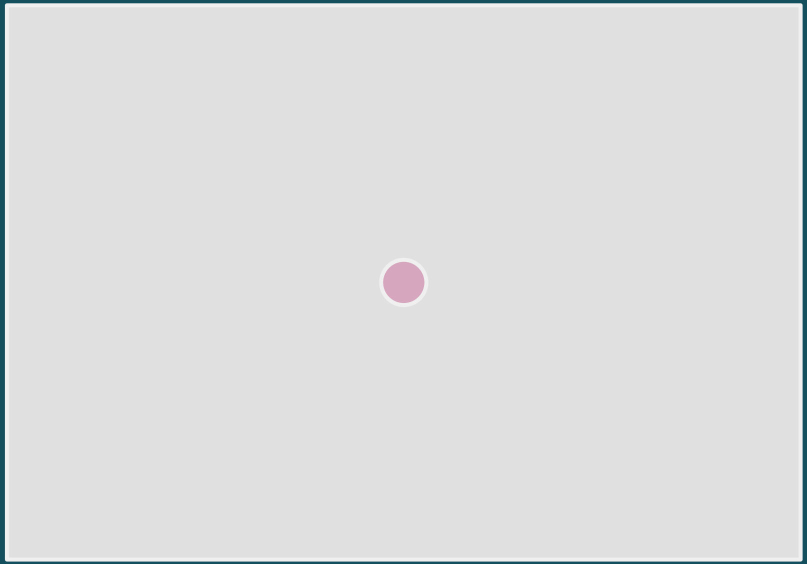
“Cells that fire together, wire together”

Source: Siegrid Löwel about Hebb’s Principle

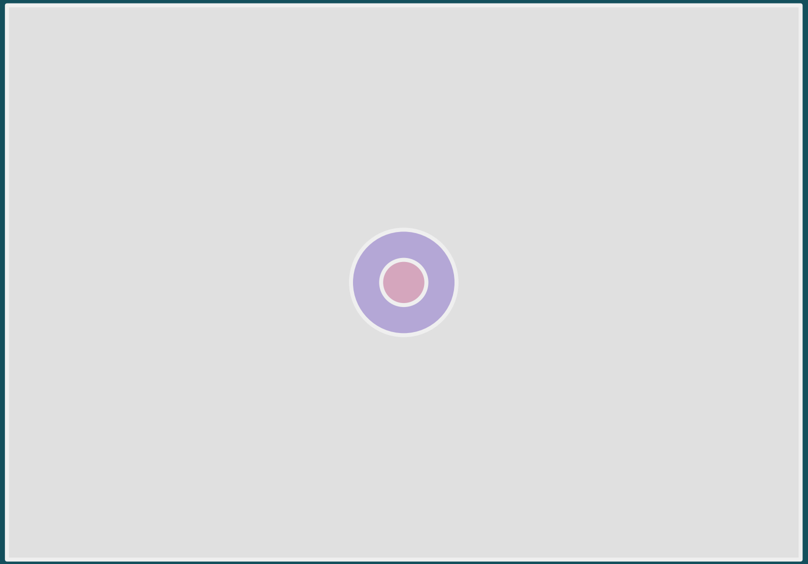
Idea of Local Neighborhood



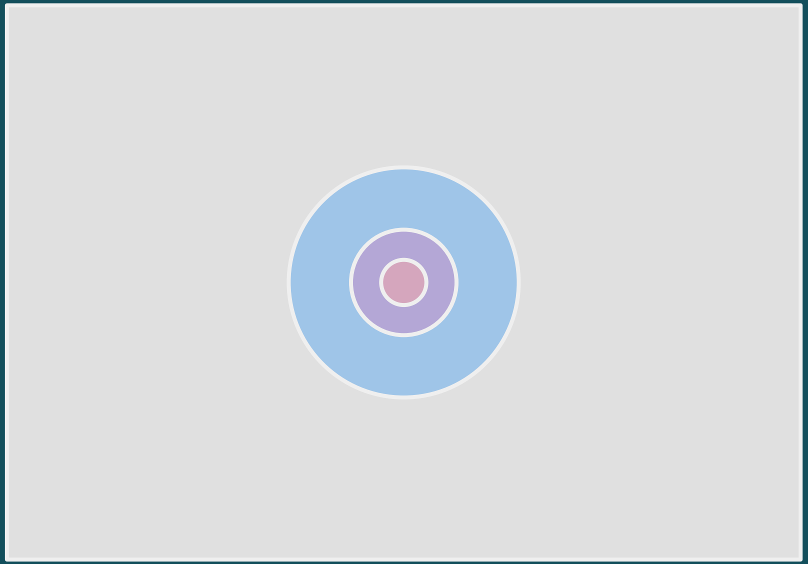
Idea of Local Neighborhood



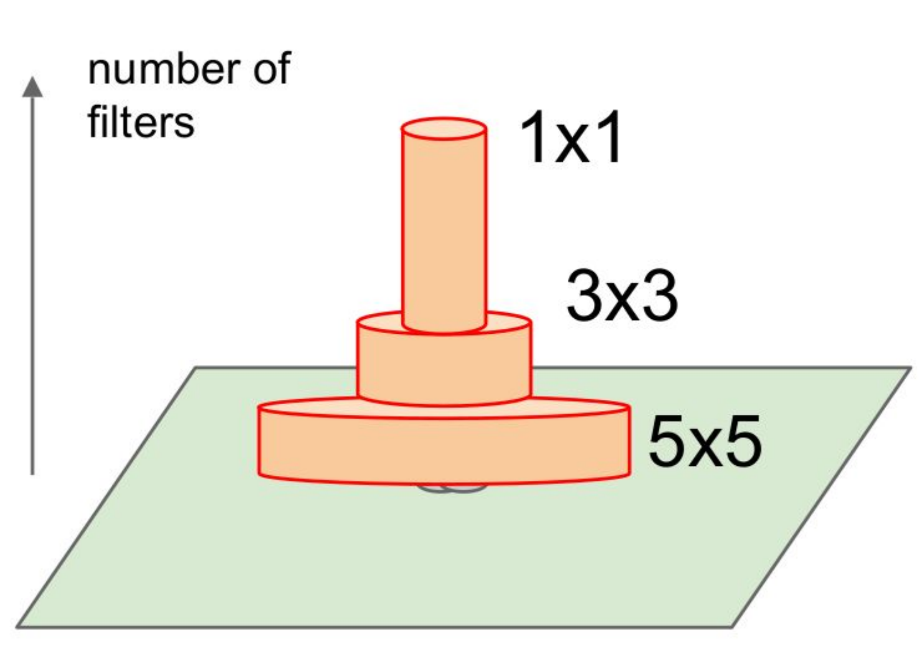
Idea of Local Neighborhood



Idea of Local Neighborhood

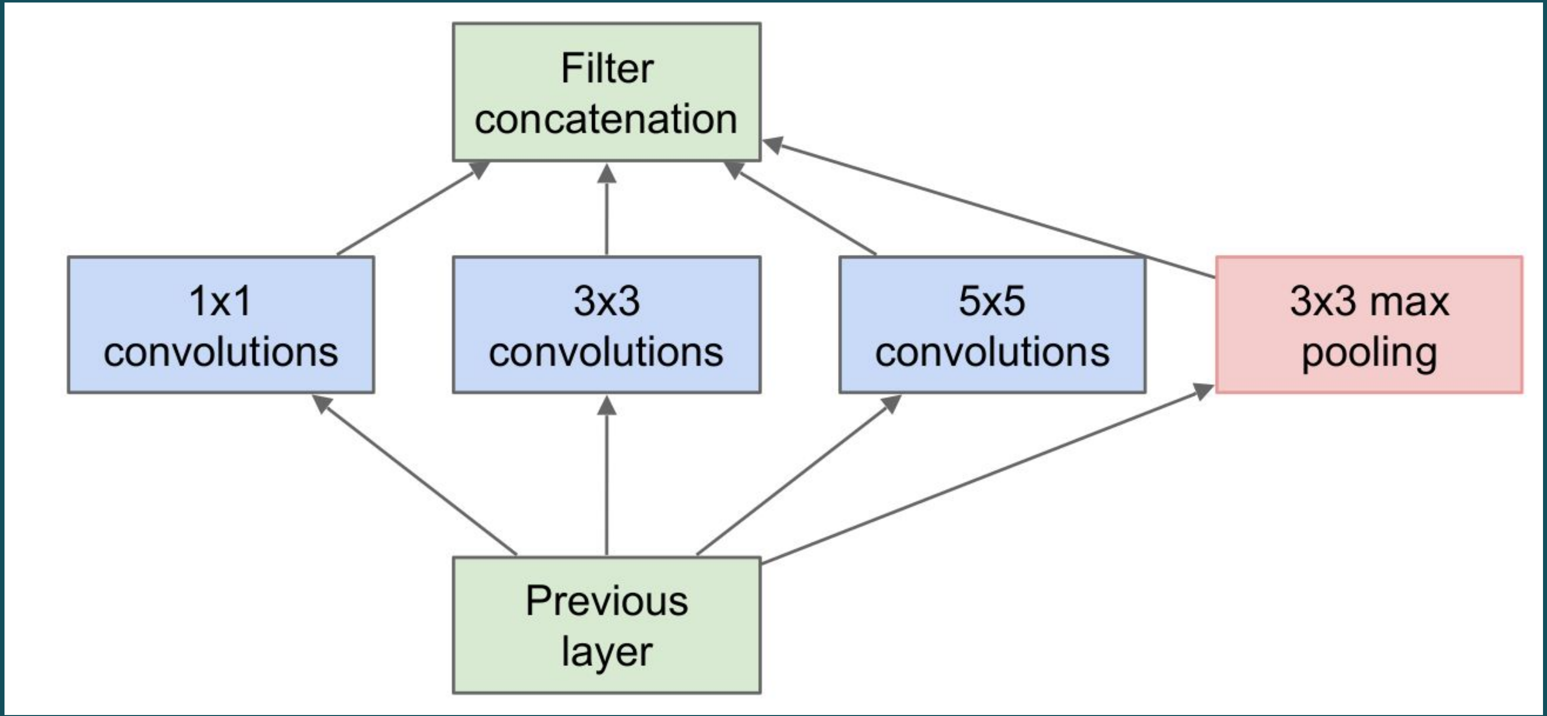


Idea of Local Neighborhood



Source: Princeton 2015

A Naive Approach (does not scale)



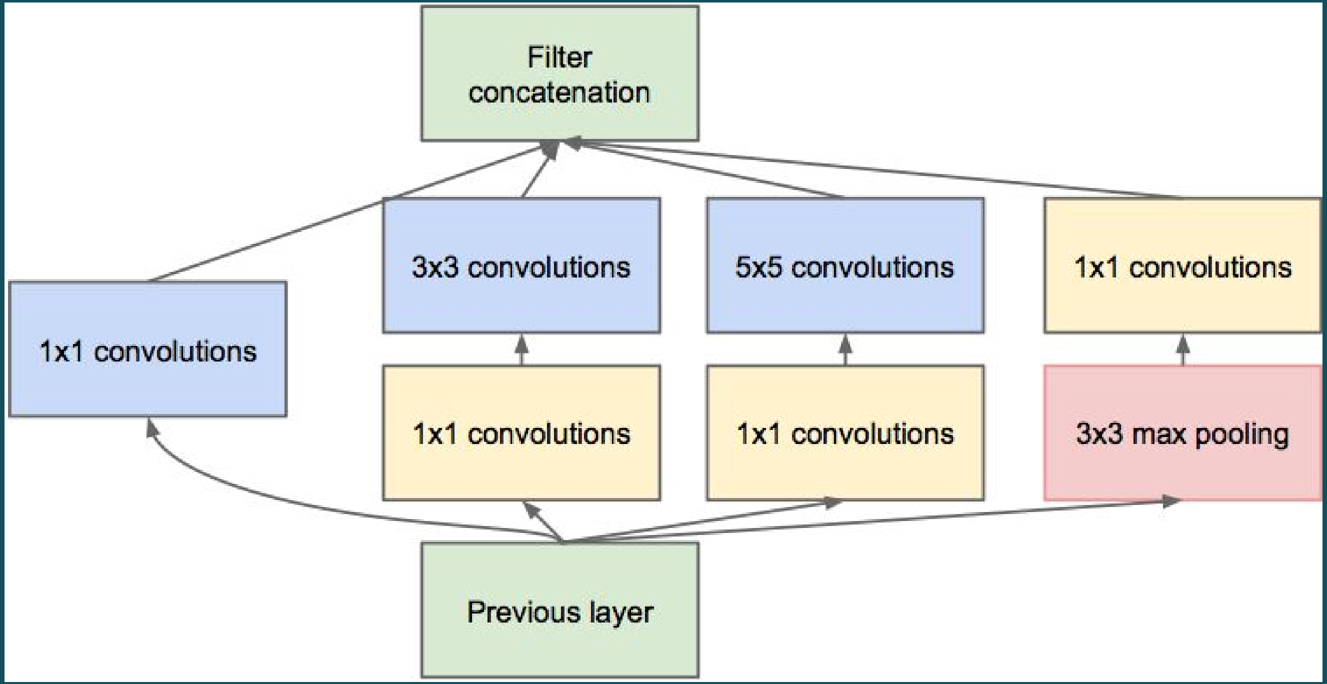
Source: [Princeton 2015]

Ingredient 2: Bottleneck Convolutions

- Input features are correlated
- Remove redundancy by Aggregation
- Compute Convolutions
- Expand features afterwards (if needed)

Source: Eugenio Culurciello's blog

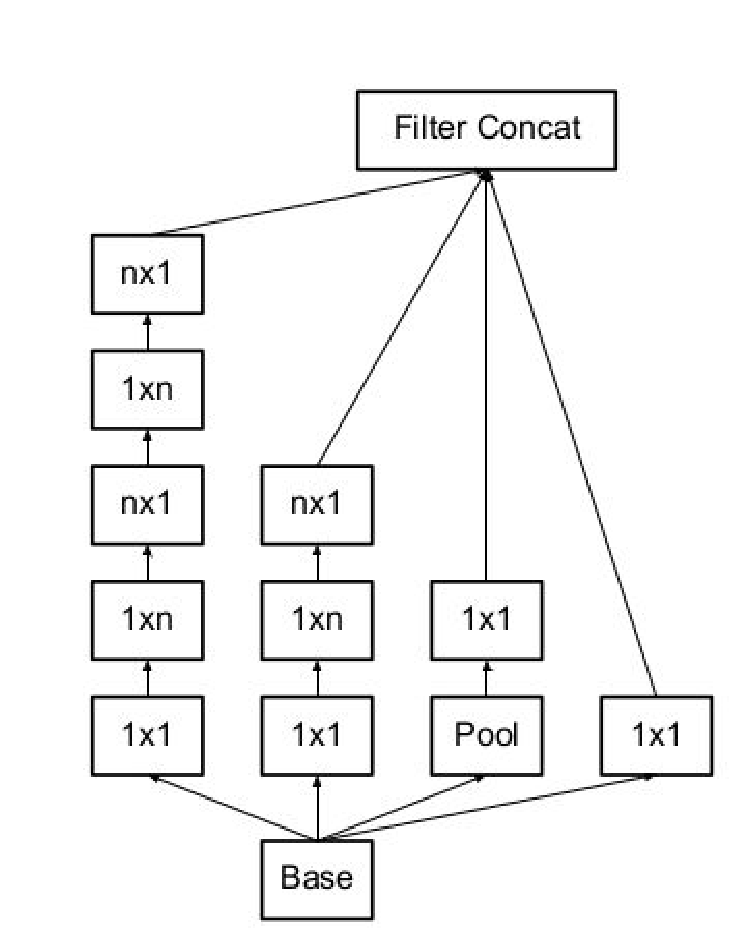
Inception Module



Source: Princeton 2015

Extensions to v1

- Batch Norm 2015
- Inception v3 2105



Source: Eugenio Culurciello's blog

CaffeJS - DL in the Browser

Source: CaffeJS

CaffeJS

- Started as Proof of Concept
- Run trained DL models in the browser
  → especially GoogLeNet

- Based on ConvNetJS by Andrej Karpathy

Why CaffeJS

- Client-side model evaluation
  - Fraud detection
  - Image upscaling
  - Sentiment detection
  - Gender + age detection (webcam)
  - Image filters

Why CaffeJS

- Teaching & Learning
    - No requirements (but a browser)
    - Understand & analyze Deep Nets
    - Debugging of Deep Nets (FF and BP)
    - Visualize the filters, layers, activations, etc.
    - Feed webcam stream into Deep Nets

Using pre-trained Models

- Offline training (many GPUs, long time)
- Export weights
- Perform only forward pass (no BP needed)

CaffeJS - Caffe Models in the Browser

- Work in progress...
- Prases *.prototxt models
- Transforms *.caffemodel weights into bin files (one file per layer) - Thanks to #1669 in Keras
- Updates weights in ConvNetJS models
- Forward and backward pass

CaffeJS vs. ConvNetJS

- Graph structure for layers + layer iterator
- Gradient correction for wide models
- Fixed Local Response Normalization
- Abstractions for visualizations
- New Layers (Concat, AVG Pool, etc.)

Demos with CaffeJS

- Visualize models and structure
- Demo using Webcam and DNN
- DeepDream ported to JS

Problems of CaffeJS

- Forward pass still slow
  - 6s for GoogLeNet
  - Most time spend in early convolutions
- ~~Too much overhead for weights (fc6 and fc7)~~
- ~~Memory issues (above 200MB)~~

Whats next

- Fully Convolutional Nets (FCN)
- WebCL: Heterogeneous parallel computing
- WebAssembly: Compilation to the web
- Deep Compression: AlexNet on 7MB
- More Layers!

Some more useful resources

- Tensorflow Playground: Neural Nets
- CS231n: Lecture, Github, and videos
- CS231n: Caffe Tutorial
- Udacity Tensorflow
- DeepDream

Thank you.