Convolutional Neural Networks: Applications and a short timeline

7th Deep Learning Meetup

Kornel Kis

Vienna, 1.12.2016.

Introduction

- Currently a master student
- Master thesis at BME SmartLab
- Started deep learning 2.5 years ago

Deep MultiLayer-Perceptrons (feedforward)

New idea: Backprop. through time

LSTM, GRU etc. (recurrent)

New idea: Input is highly redundant

ConvNETs (feedforward)

The „naive approach”

- Let’s apply MLP networks to real-world images…
- CIFAR-10 database: 32*32*3 pixels
- 32*32*3 = 3072 inputs !
  - Does not scale with image size
  - Really inefficient knowledge representation
  - MLPs work well with de-correlated input

Key Idea

Information on real-world images are really redundant

Learn patterns, not data

Feature engineering -> Feature learning

ConvNets

- Feedforward networks
- Use different types of layers
  - Convolutional layer
  - Pooling layer
  - ReLU layer
  - (Sometimes FC layer)
  - Etc.

Convolutional Layer

- Each unit is only connected to a small part of the image
  - „receptive field” of a neuron
- Weights are shared inside one „depth slice”
  - Each „depth slice” is using the same weights
  - Drastically reduces parameter count
- A 3D volume of neurons
  - Width, Height, „Depth”

Convolutional Layer

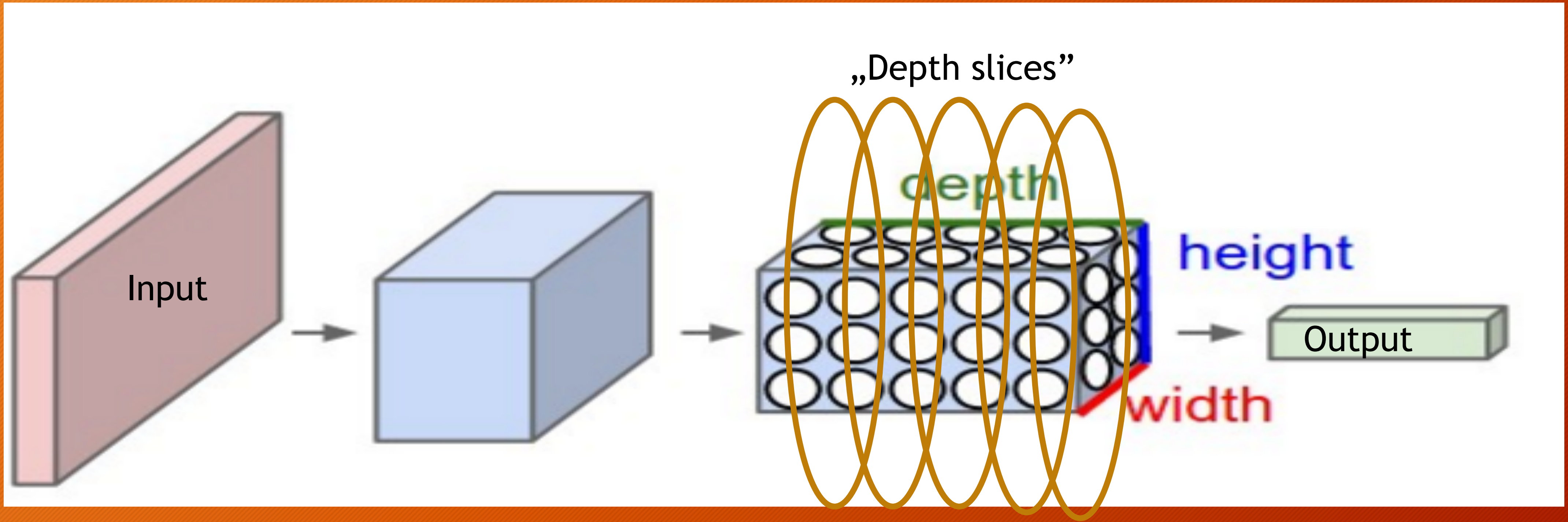


Image source: http://cs231n.github.io/convolutional-networks/

Pooling layer

- Downsampling -> further control of parameters
- Types: Max-pooling, Avg-pooling Etc.
- Changes spatial structure, but not parameter count

ReLU layer

- Applies a non-linear function
- Does change neither parameter count, nor structure

Important ConvNet architectures

- AlexNet (2012) – 8 layers
- VGG (2014) – 16/19 layers
- GoogLeNet (2014-) – 22 layers (Inception v1)
- ResNet (2015-) – 152 layers !!!
- Inception-ResNet (2016) - (Inception v4)

What about vanishing gradients?

GoogLeNet

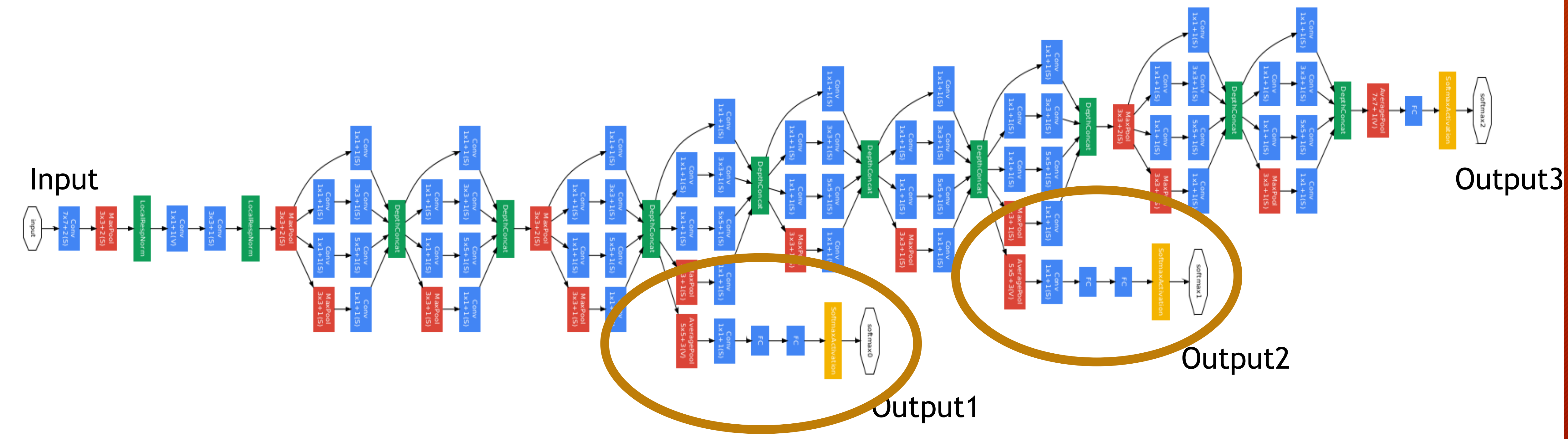
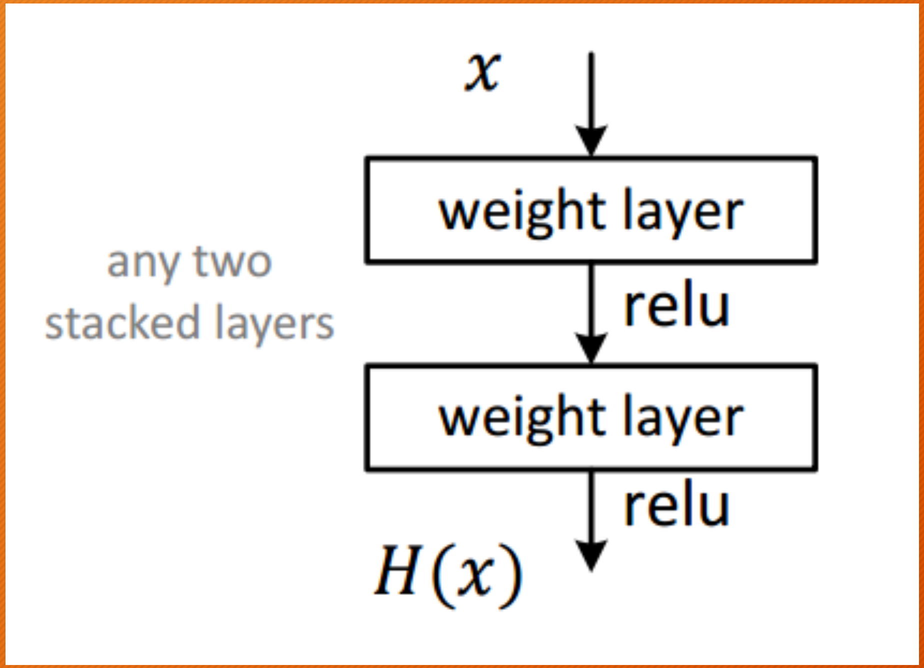


Image source: https://indico.io/blog/wp-content/uploads/2015/08/googlenet.png

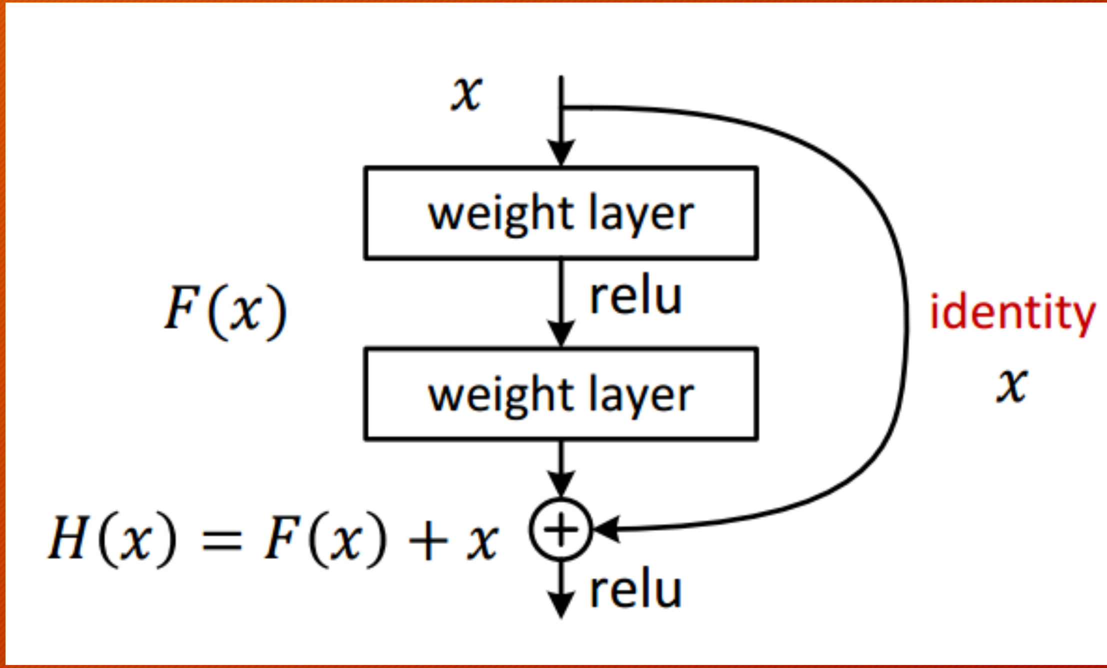
ResNet

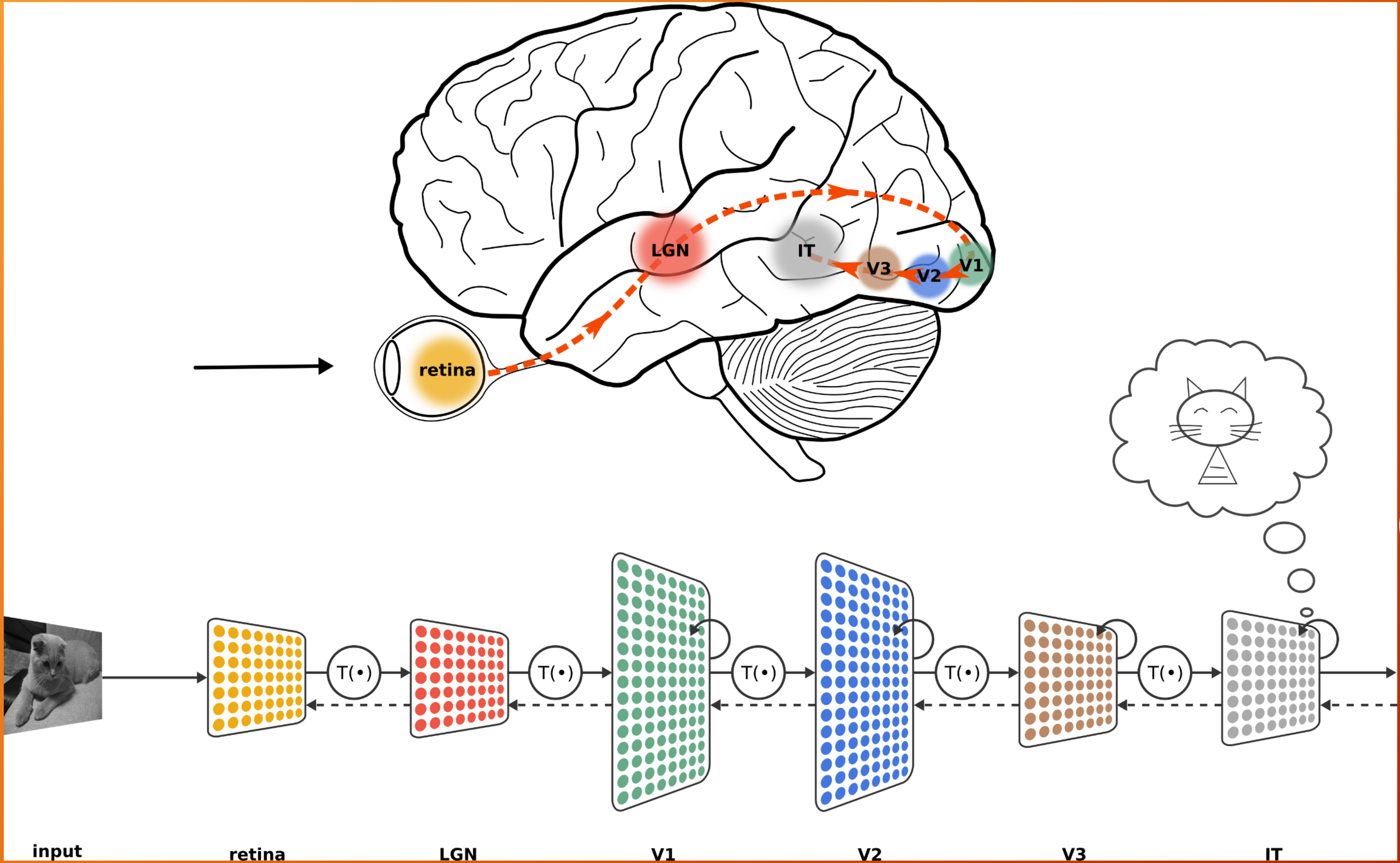
- Not just stacking layers on top of each other…

„Normal”



„Residual”





A brain analogy (hypothesis)

Image source: https://figshare.com/articles/Ventral_visual_stream/106794 (Jonas Kubilius)

Spiking neural networks

- More similar to real neurons
- Neurons do not fire at each cycle, only after reaching some threshold
- Output information is carried by frequency or timing between spikes

- Possible applications (?)
  - Direct modeling of the nervous system
  - Not yet widespread due to hardware constraints and stability problems

Applications

- „Features Matter". ([Girshick et al. 2014])
  - Meaning: better feature learning directly yield better overall performance
- Frameworks are being developed in many areas which can use any deep ConvNet as feature learner
  - Wide range of applications (images, videos, speech technology, etc.)

General approach

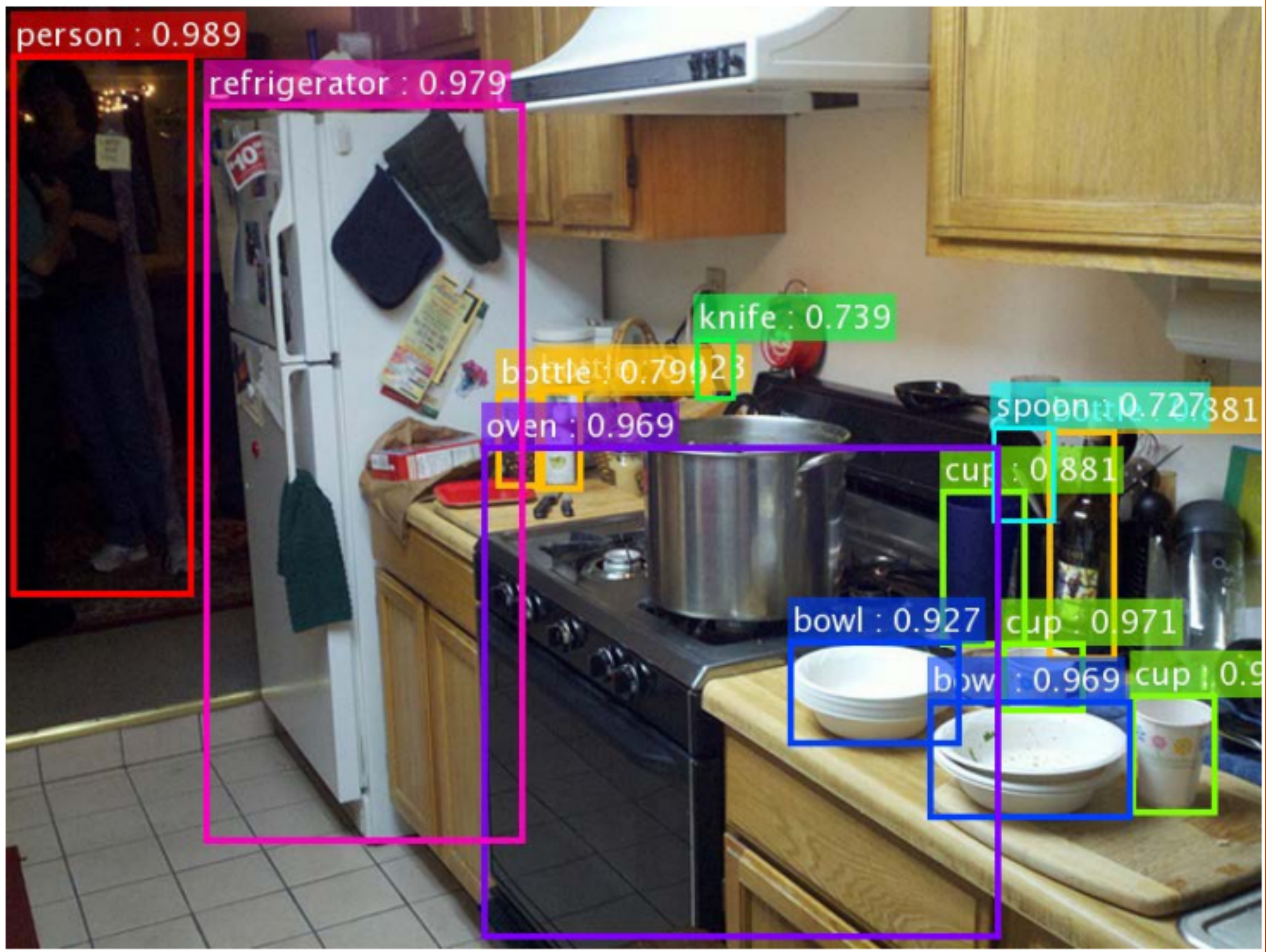
- Networks are rarely trained from „scratch”
  - Pre-trained weights are often available for each network
- Fine-tuning (transfer learning) for specific application
  - Only FC layers (the classifier) are re-trained from random initials

A brief example: object detection

- Problem: Finding multiple objects on an image
  - Output: bounding-boxes given with 4 coordinates and detection probability

- Massive improvements in performance in the recent years, thanks to ConvNets

- Frameworks include R-CNN, Fast R-CNN, Faster R-CNN, R-FCN



R-FCN/ Faster R-CNN example

Image source: Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015. Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.

R-FCN

- Latest generation of object detection frameworks
  - Based on Faster R-CNN
  - Regression to boundary box coordinates + object classification inside boundary box
  - Uses an arbitrary ConvNet (most often ResNet) as backend
  - Surpasses human-level performance
  - Detection works very close to real-time (!)

Our current project

- Skin cancer image detection/classification
  - Huge database of medical images
  - Using Faster R-CNN (currently)
  - Image labeling/preprocessing is neccessary
  - Related work:
    - [Liao, Haofu. "A Deep Learning Approach to Universal Skin Disease Classification."]
    - Already surpassing human expert performance on a much smaller dataset

Challenges

- Huge unlabeled dataset
- Many preprocessing steps
- Pre-trained models may not be applicable

Thank you for your attention!

Image source: https://www.carmudi.ae/journal/wp-content/uploads/2015/06/Google-Self-Driving-Car.jpg