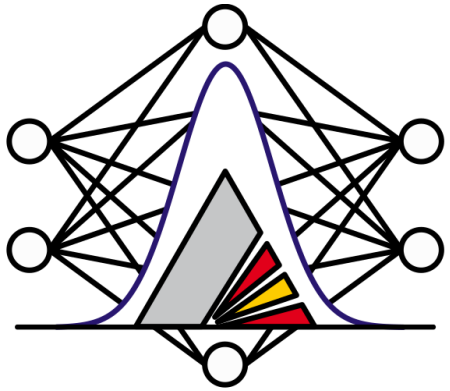
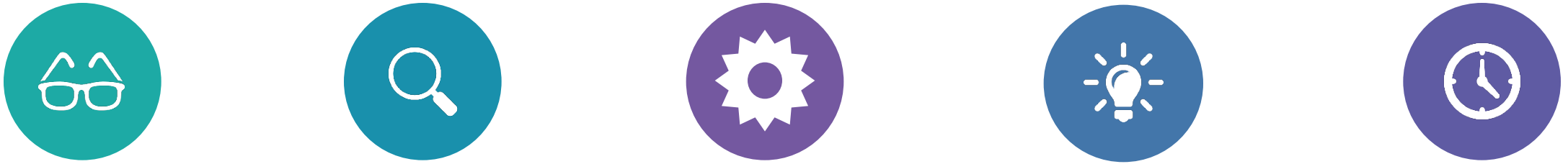
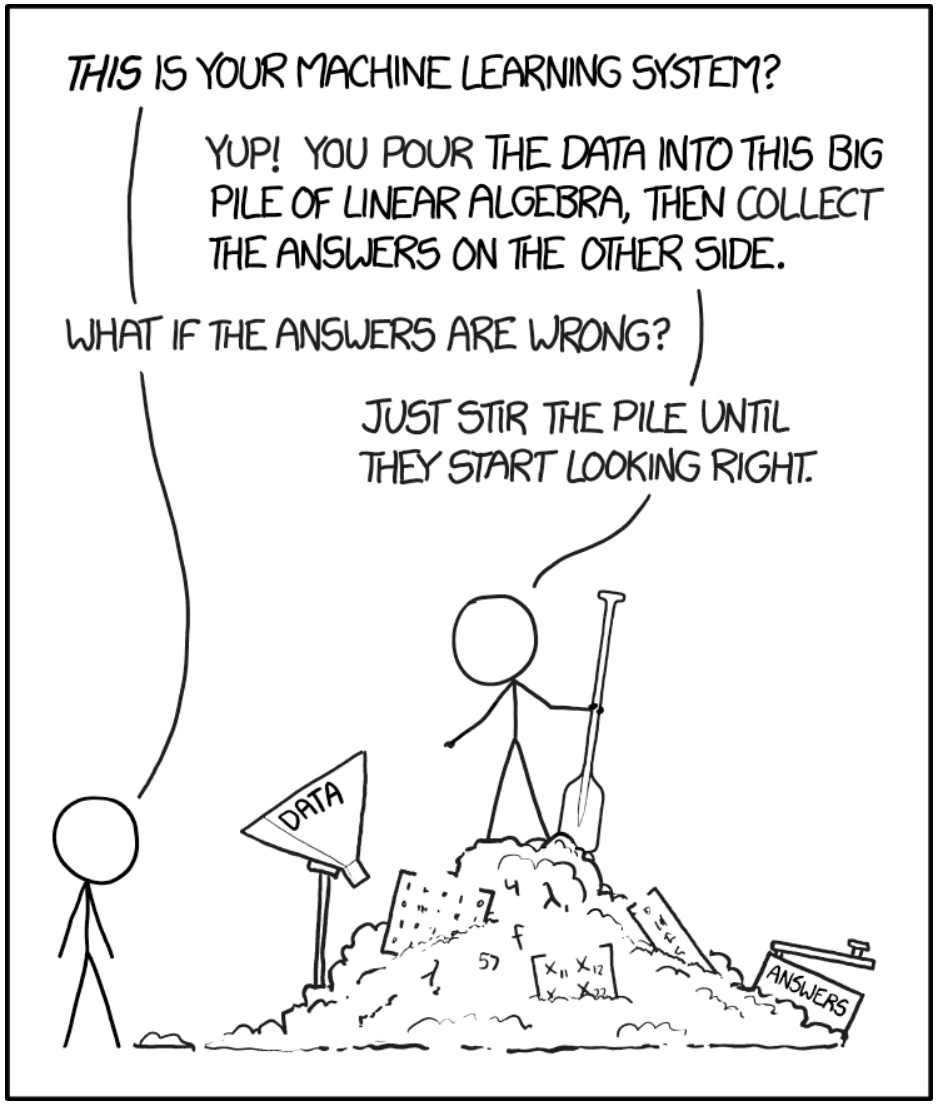


Introduction to TensorFlow with Application to Autonomous Driving



Christian Hubschneider
FZI Forschungszentrum Informatik

24.05.2017


THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Topics

Overview over Frameworks

TensorFlow Basics

Coding Examples

TensorFlow for Autonomous Driving

How to get involved

Deep Learning Frameworks

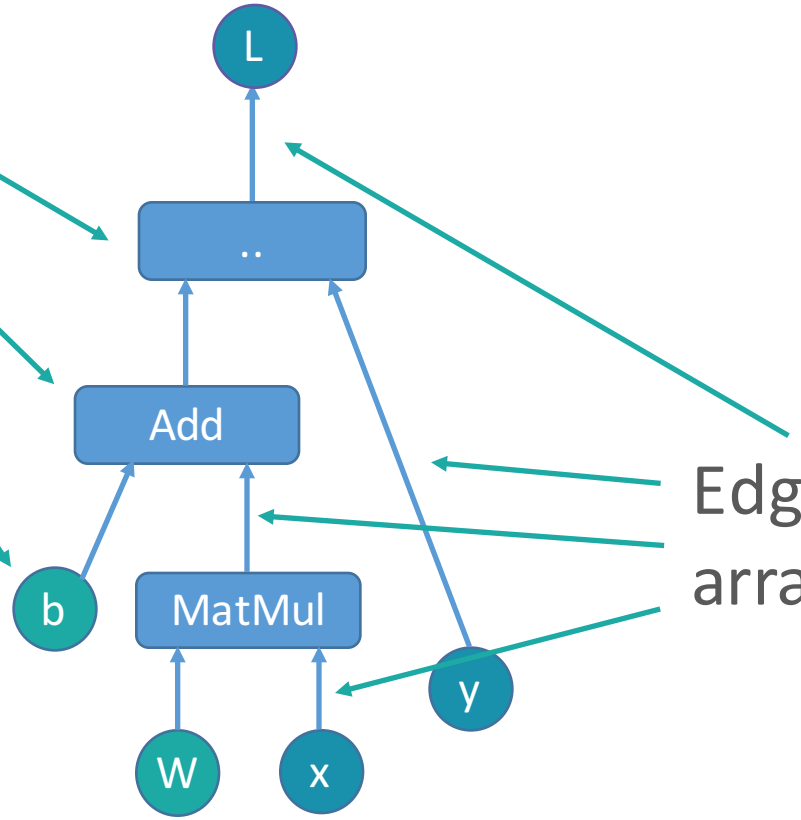
OVERVIEW OVER CROSS-PLATTFORM MACHINE LEARNING TOOLS

- Caffe: caffe.berkeleyvision.org/ - *U.C. Berkeley*
  - **C++**, Python, MATLAB
- Caffe2: caffe2.ai/ - *Facebook*
  - **C++, Python**
  - Released: April 18th 2017: https://caffe2.ai/blog/2017/04/18/caffe2-open-source-announcement.html
- MXNet: mxnet.io/ - *Amazon / AWS*
  - **C++**, Python, Julia, Scala, R, MATLAB, ..
- TensorFlow: https://www.tensorflow.org/ - *Google*
  - C/**C++, Python**, Java, Go
- Theano: http://www.deeplearning.net/software/theano/
  - **Python**
- Torch: http://torch.ch/ - *New York University*
  - **C**/C++, **Lua**, LuaJIT
- Keras, Lasagne, ...

A Simple Graph

TENSORFLOW REPRESENTS THE LEARNING PROBLEM AS A GRAPH

Nodes are Operations



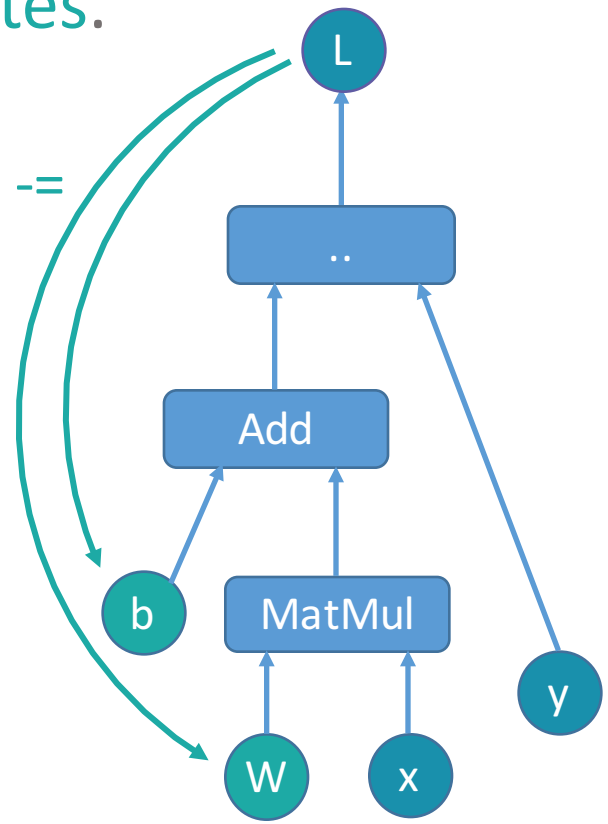
Edges are n-dimensional arrays: Tensors

Each node has zero or more inputs and zero or more outputs.

A Simple Graph

The graph has a state that can be used to perform gradient descent (or any other) weight updates.

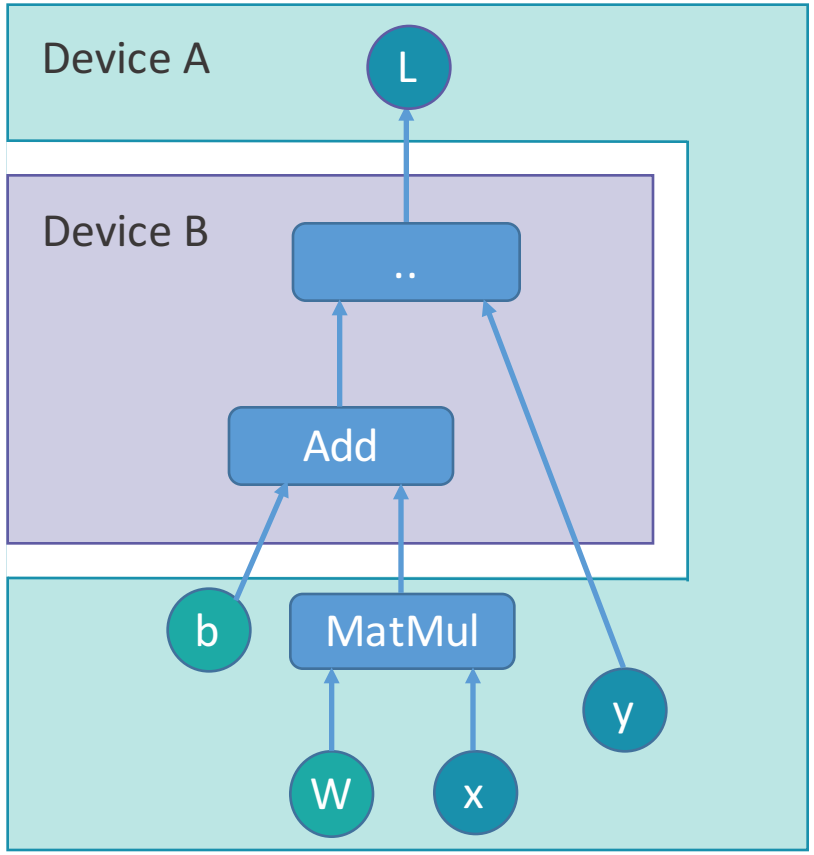
Update bias and weights.



A Simple Graph

SCALABILITY THROUGH THE GRAPH STRUCTURE

In TensorFlow the graph makes distributed processing on multiple devices easy.



Devices: Processes, Machines, CPUs, GPUs, ..

Overview over Operators

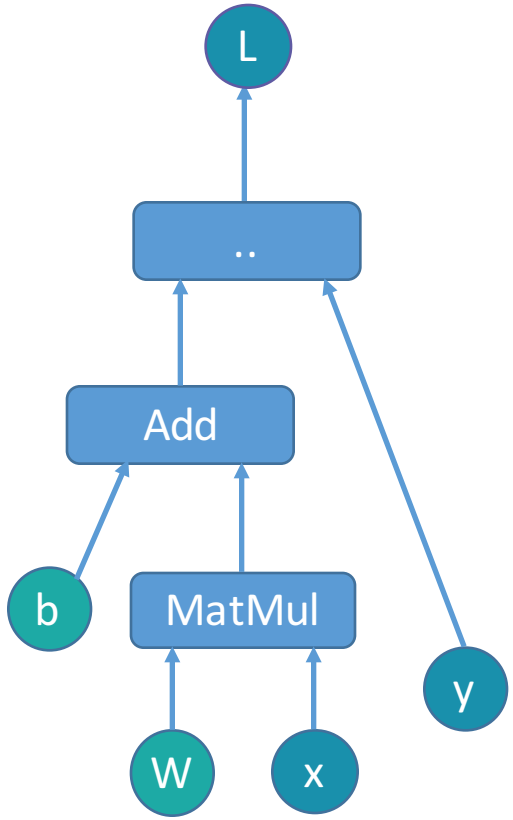
OPERATIONS REPRESENT ABSTRACT COMPUTATIONS ON A NUMBER OF NODES

Category	Example
Element-wise mathematical operations	Add, Sub, Mul, Div, Exp, Log, Compare, ..
Array operations	Concat, Slice, Split, Shuffle, ..
Matrix operations	MatMul, MatrixInverse, MatrixDeterminant, ..
Stateful operations	Variable, Assign, AssignAdd, ..
Neural-net building blocks (layers, non-linearities)	SoftMax, Sigmoid, ReLU, Conv2D, MaxPool, ..
Checkpoint operations	Save, Restore
Queue and sync operations	Enqueue, Dequeue, ..
Control flow operations	Merge, Switch, Enter, Leave, NextIteration

A Simple Graph

NUMERICAL COMPUTATION USING DATA FLOW GRAPHS

A graph is defined in Python by attaching nodes to the global graph.




```
import tensorflow as tf

# Placeholders for input
x = tf.placeholder(tf.float32)
y = tf.placeholder(tf.float32)

# Variables to be learned
W = tf.get_variable("W", [1, 1])
b = tf.get_variable("b", [1])

y_pred = tf.matmul(x, W) + b
loss = tf.reduce_mean(tf.square(y - y_pred))
```

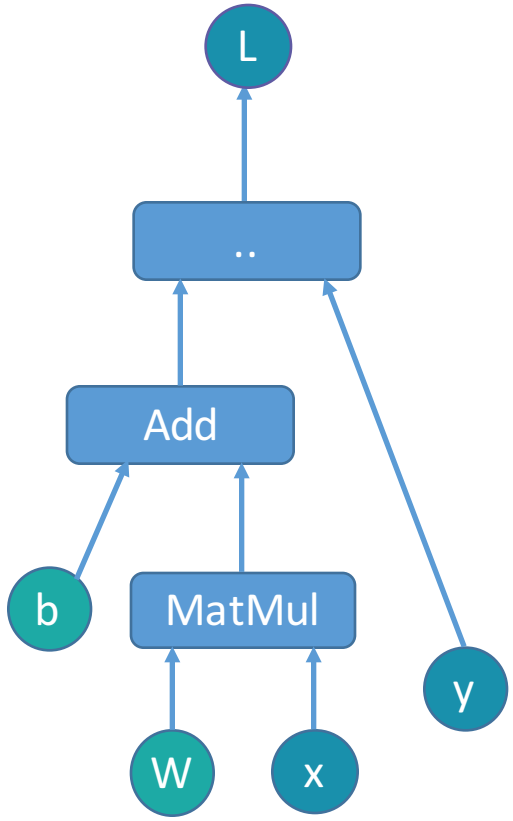
Nothing is executed yet!

$$L(W,b) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (Wx_i + b))^2$$

Tensorflow Sessions

NUMERICAL COMPUTATION USING DATA FLOW GRAPHS

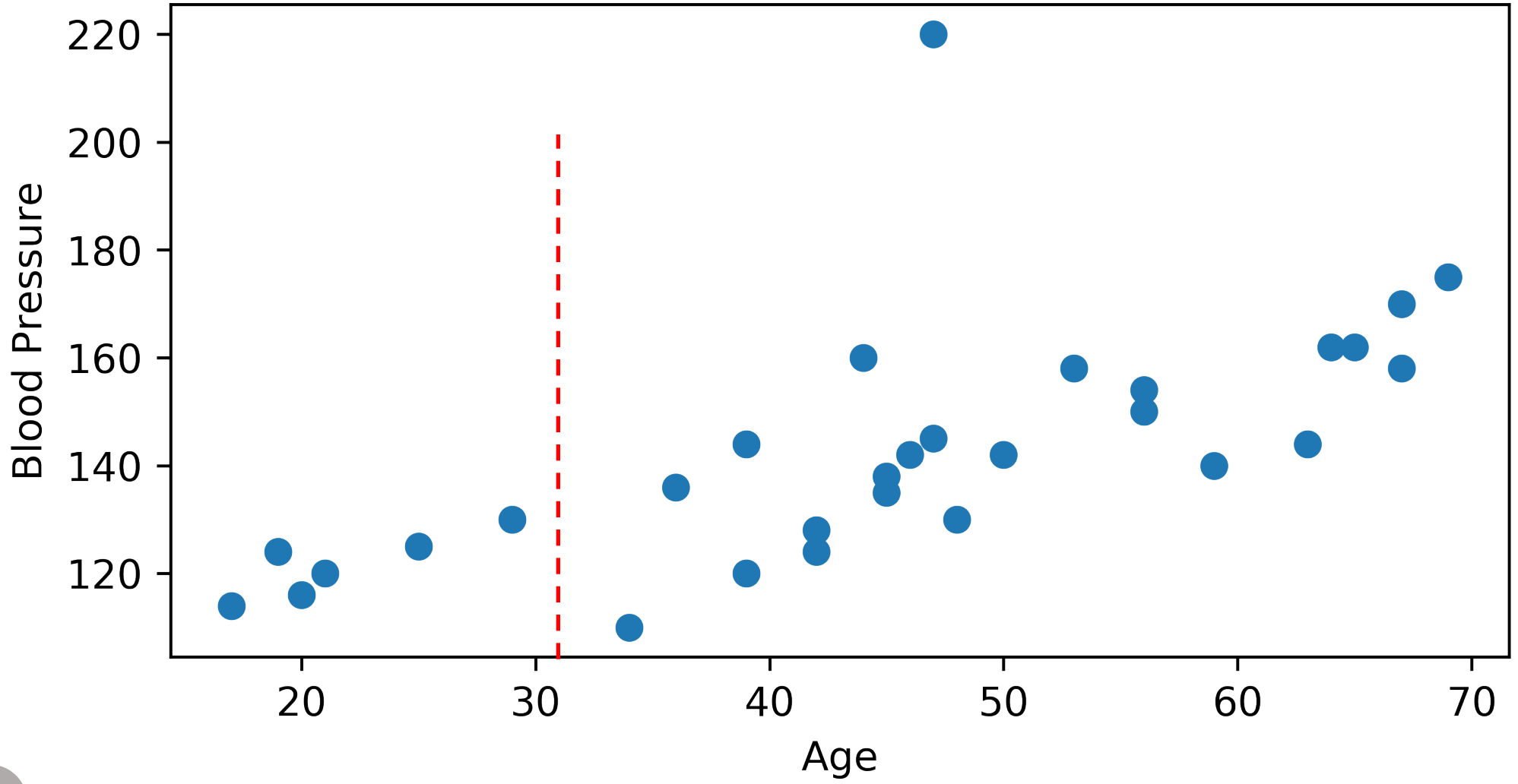
Clients interact with and run the TensorFlow engine using Sessions.




```
training_step =
tf.train.GradientDescentOptimizer(0.001).minimize(loss)


with tf.Session() as sess:
   # Initialize all variables in graph
   sess.run(tf.global_variable_initializer())

   xs, ys = data[0], data[1]  # choose data for step

   for _ in range(10000):
       # Run gradient descent step
       sess.run(training_step, feed_dict={x: xs, y: ys})
```

Linear Regression

WHAT SHOULD MY BLOOD PRESSURE BE WHEN I AM 31?



Linear Regression

WHAT SHOULD MY BLOOD PRESSURE BE WHEN I AM 31?

Assume a model

$$y = aX + b + \epsilon,$$

where *a* and *b* are unknown constants that represent **slope** and **intercept** and error term ϵ.

Given some estimates $\hat{a}$ and $\hat{b}$ for the model coefficients, we can predict the blood pressure for arbitrary ages X.

How can we find those estimates? (e.g. using Gradient Descent*)

*Be aware that for this simple case way better solutions than Gradient Descent do exist! ;-)

Simple Linear Regression using TensorFlow

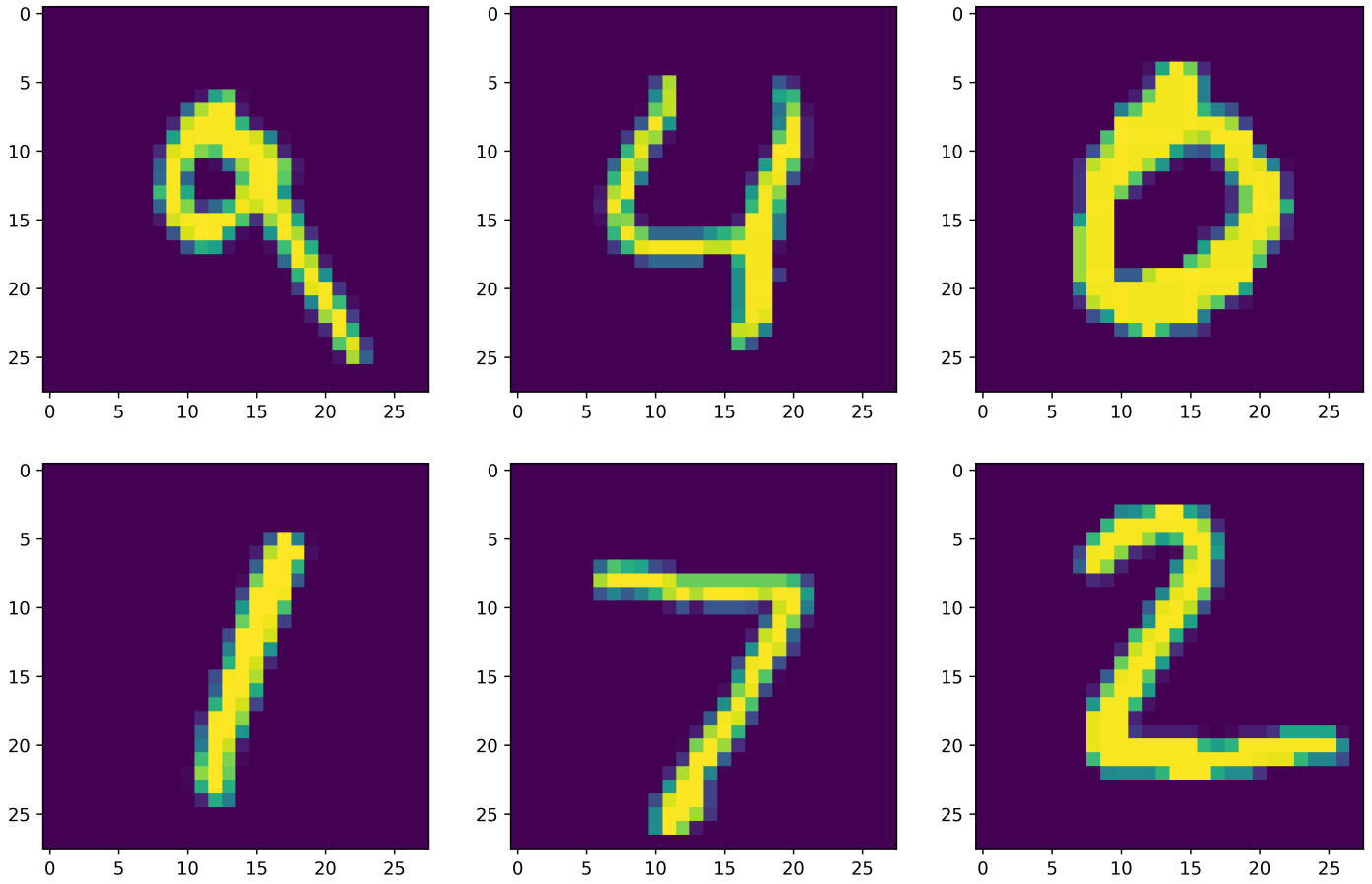
WHAT SHOULD MY BLOOD PRESSURE BE WHEN I AM 31?

Live demo

linear_regression.ipynb

The MNIST Dataset

THE MNIST DATABASE OF HANDWRITTEN DIGITS (by Yann LeCun et al.)

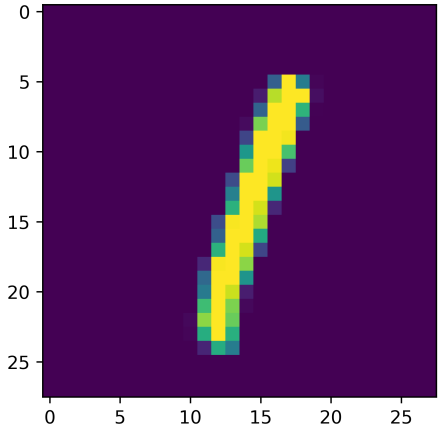


Dataset containing handwritten digits (0-9)

- 60’000 training samples
- 10’000 test samples
- Resolution: 28x28
- State of the art: **0.21% classification error rate**

Logistic Regression, Softmax and Classification

CLASSIFICATION INSTEAD OF REGRESSION



$$L_i = -\log \frac{e^{s_{y_i}}}{\sum_j e^{s_j}}$$

unnormalized probabilities

0	2.8	exp	16.4	normalize	0.74
1	**1.7**	→	**5.5**	→	**0.25**
2	-1.2		0.3		0.01

unnormalized log probabilities

probabilities

Softmax Loss:

L_i = -log(0.25) = 0.61

„Multinomial Logistic Regression“ on MNIST

USING LOGISTIC REGRESSION TO SOLVE A CLASSIFICATION TASK ON MNIST

Live demo

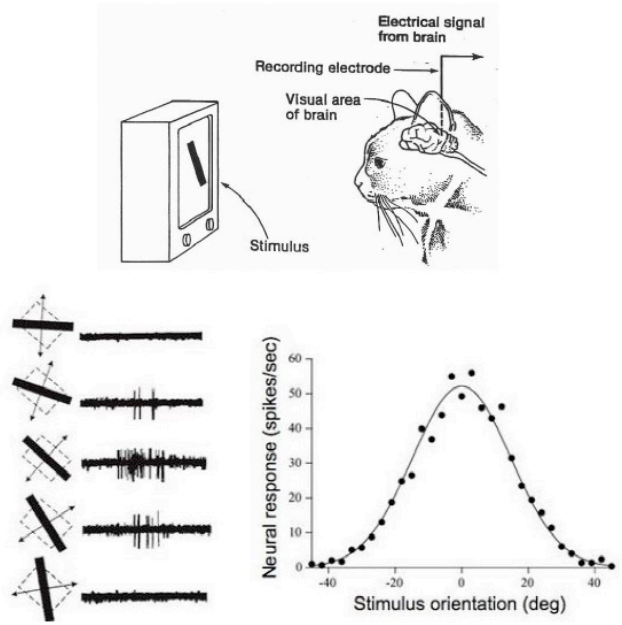
softmax_classification_mnist.ipynb

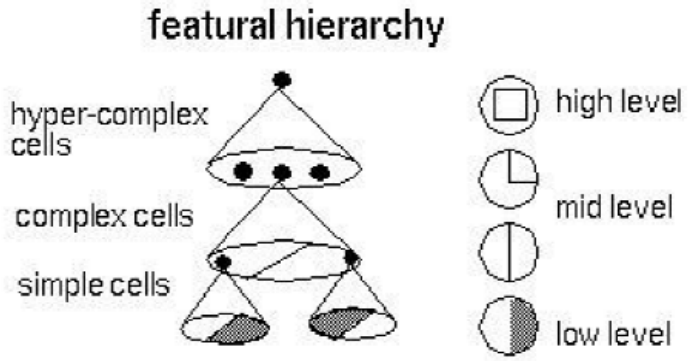
Convolutional Neural Networks

OR HOW TO GET EVEN BETTER CLASSIFICATION RESULTS

How to get even better classification results?

Use spatial relationships between pixels in the image (inspired by Biology)!



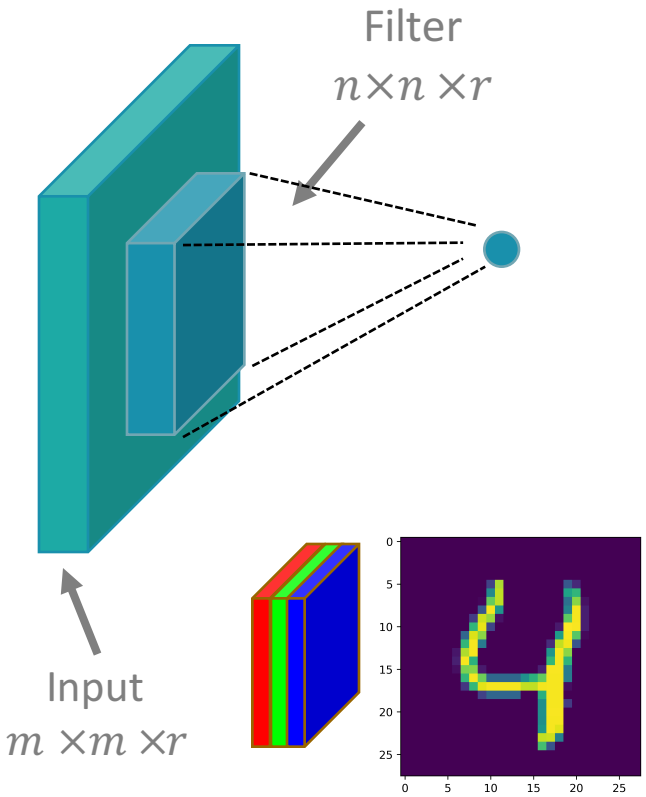


[Hubel & Wiesel 1959, 1968]

Building Blocks of a CNN: Convolutional Layer

OR HOW TO GET EVEN BETTER CLASSIFICATION RESULTS

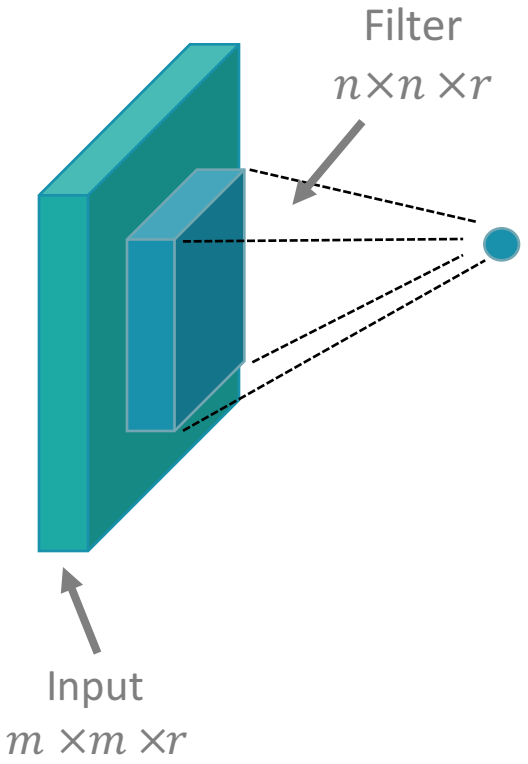
Each neuron only sees part of the input (“receptive field”).

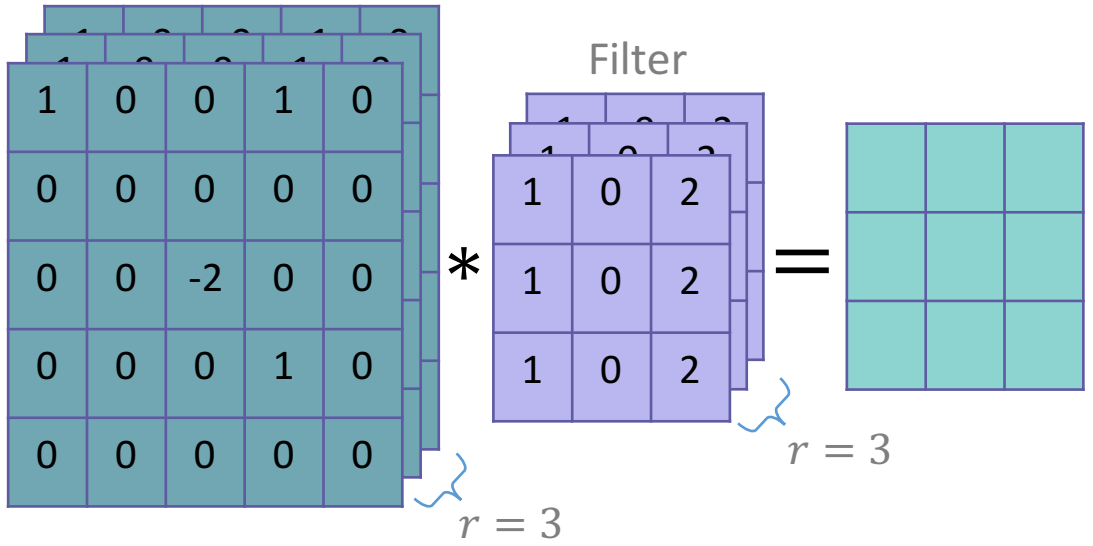


Building Blocks of a CNN: Convolutional Layer

OR HOW TO GET EVEN BETTER CLASSIFICATION RESULTS

Output of the neuron: discrete convolution of all input-features for one filter.



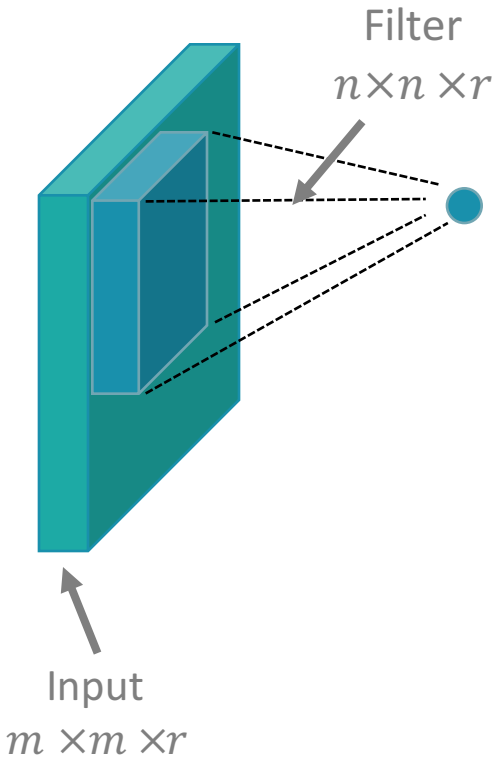


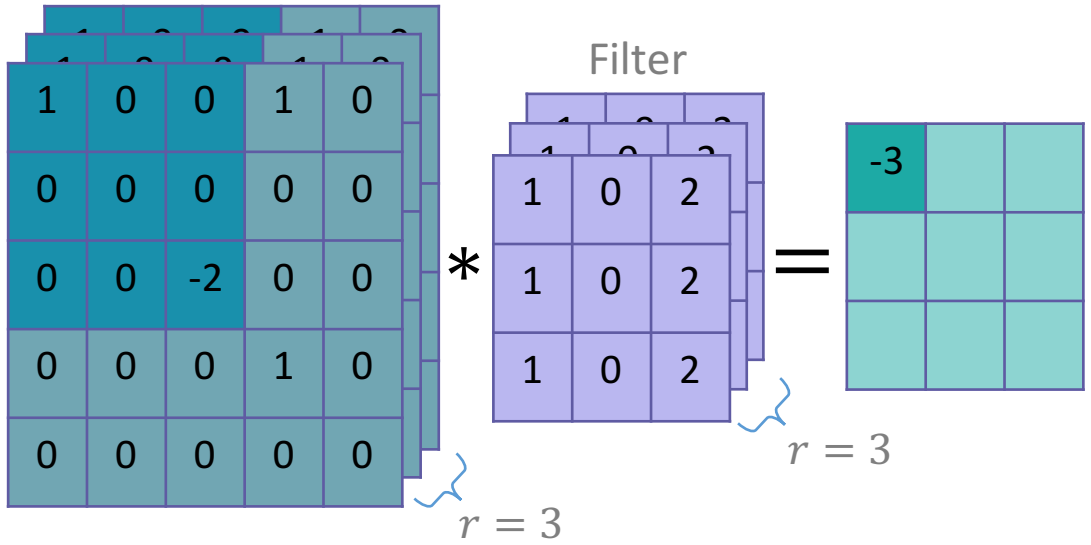
$$(I * F)_{r,s} := \sum_{u=0}^{r-1} \sum_{v=0}^{r-1} I_{r+u,s+v} \cdot F_{u,v}$$

Building Blocks of a CNN: Convolutional Layer

OR HOW TO GET EVEN BETTER CLASSIFICATION RESULTS

Output of the neuron: discrete convolution of all input-features for one filter.



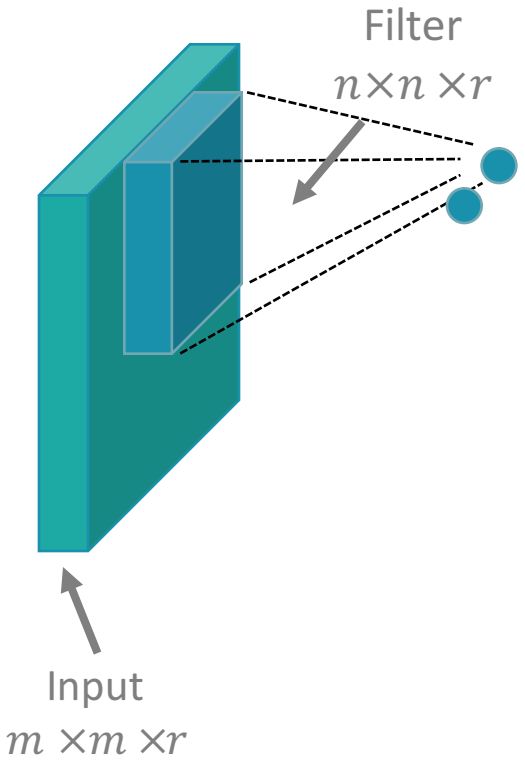


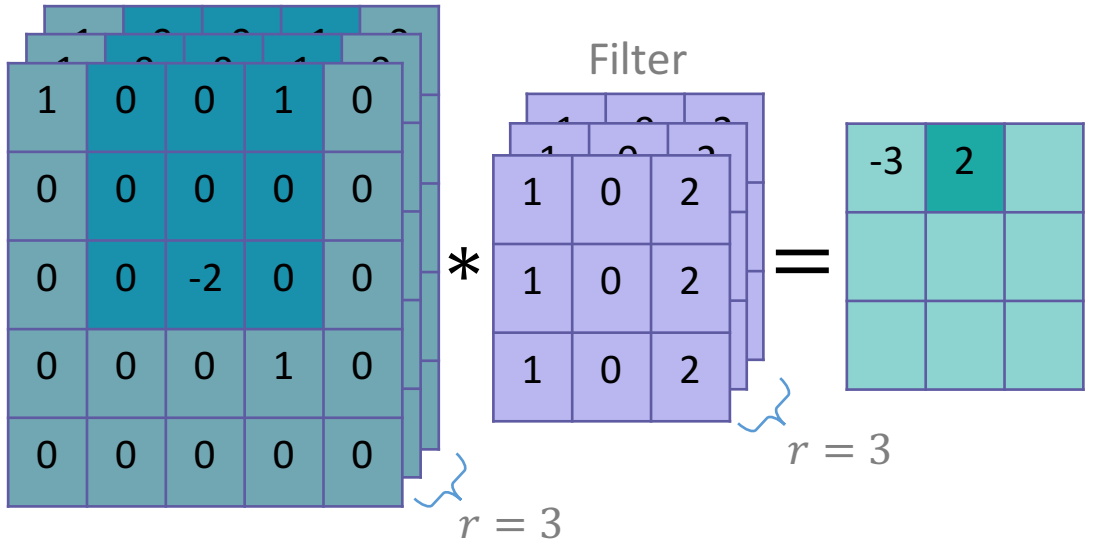
$$(I * F)_{r,s} := \sum_{u=0}^{r-1} \sum_{v=0}^{r-1} I_{r+u,s+v} \cdot F_{u,v}$$

Building Blocks of a CNN: Convolutional Layer

OR HOW TO GET EVEN BETTER CLASSIFICATION RESULTS

Output of the neuron: discrete convolution of all input-features for one filter.



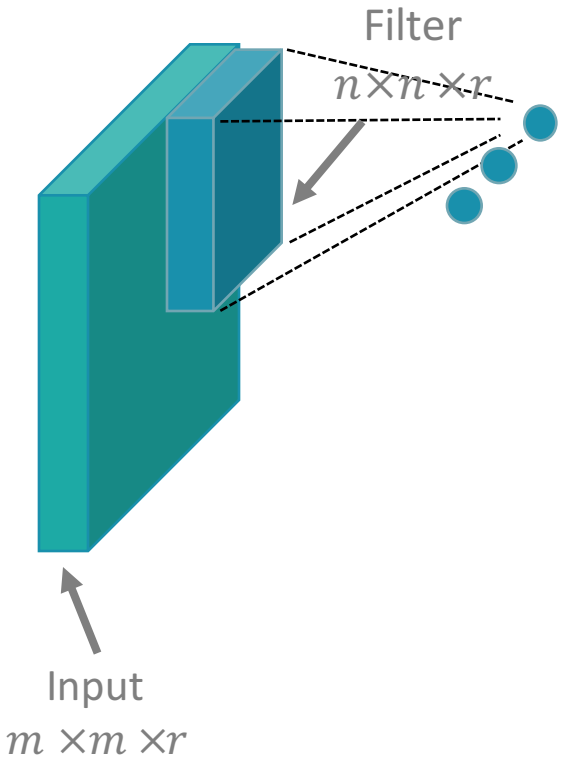


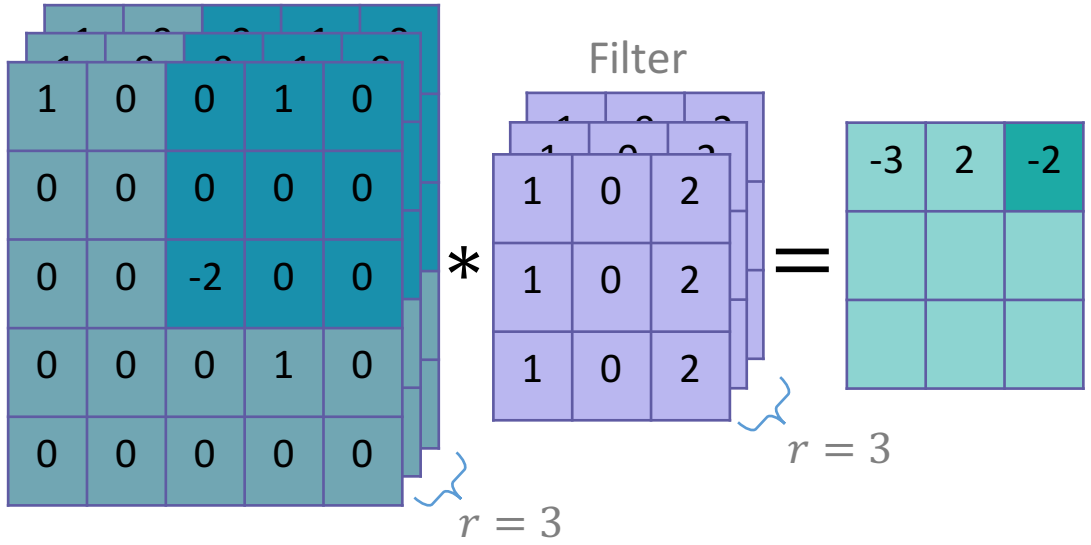
$$(I * F)_{r,s} := \sum_{u=0}^{r-1} \sum_{v=0}^{r-1} I_{r+u,s+v} \cdot F_{u,v}$$

Building Blocks of a CNN: Convolutional Layer

OR HOW TO GET EVEN BETTER CLASSIFICATION RESULTS

Output of the neuron: discrete convolution of all input-features for one filter.



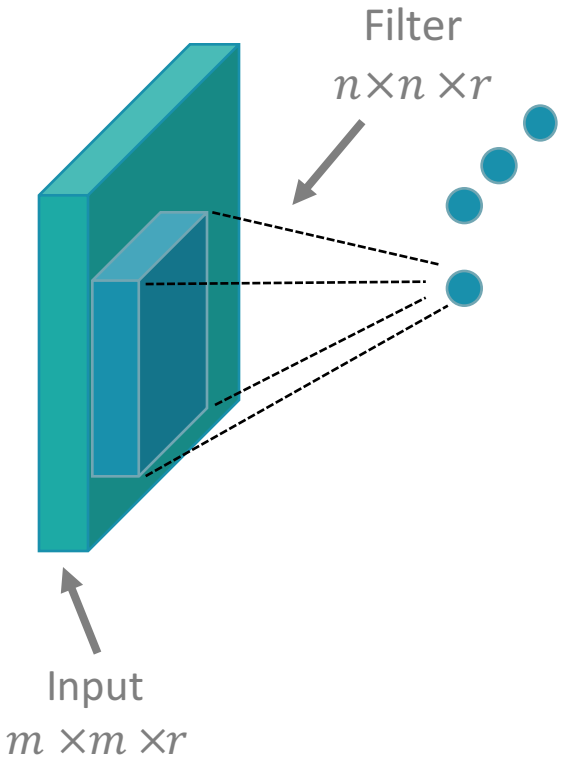


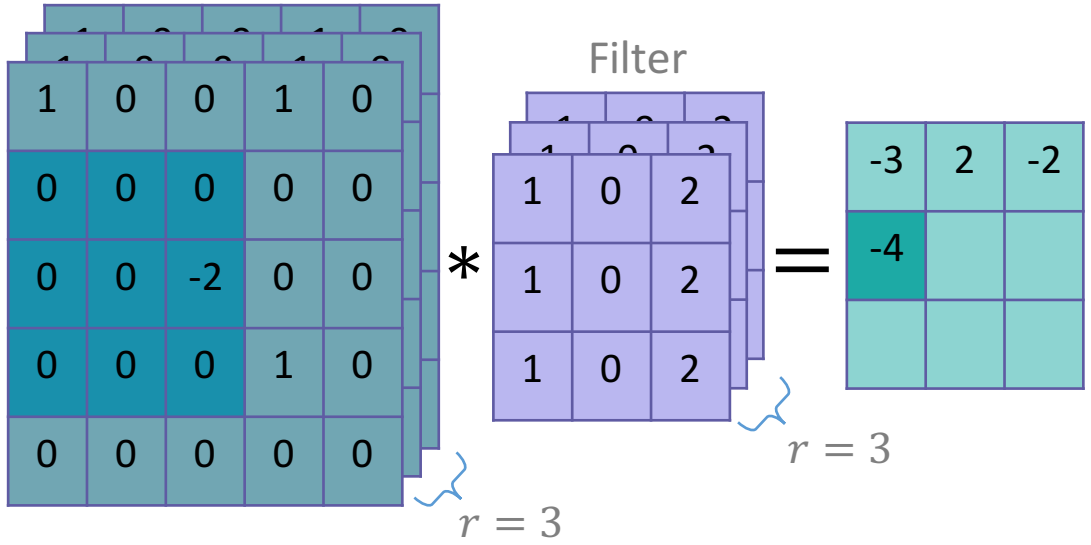
$$(I * F)_{r,s} := \sum_{u=0}^{r-1} \sum_{v=0}^{r-1} I_{r+u,s+v} \cdot F_{u,v}$$

Building Blocks of a CNN: Convolutional Layer

OR HOW TO GET EVEN BETTER CLASSIFICATION RESULTS

Output of the neuron: discrete convolution of all input-features for one filter.



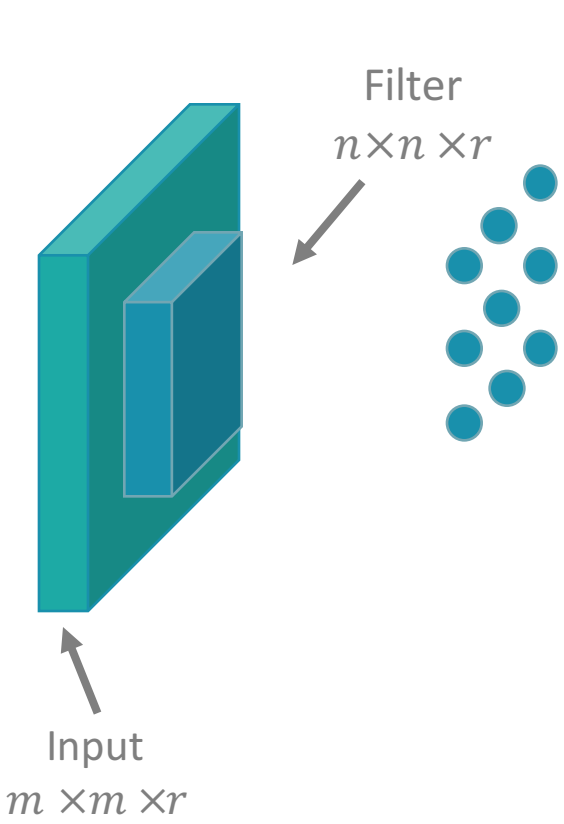


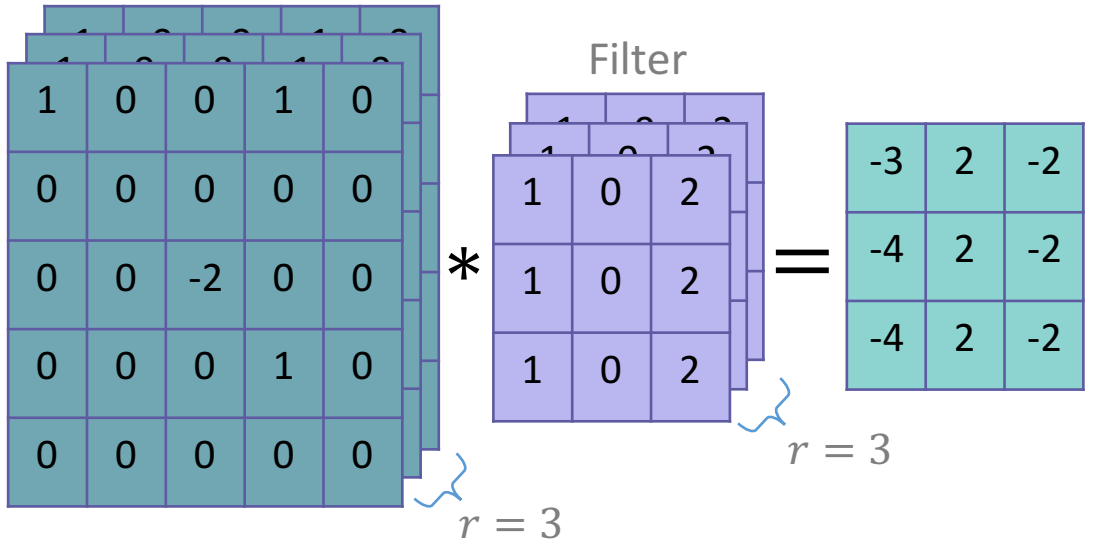
$$(I * F)_{r,s} := \sum_{u=0}^{r-1} \sum_{v=0}^{r-1} I_{r+u,s+v} \cdot F_{u,v}$$

Building Blocks of a CNN: Convolutional Layer

OR HOW TO GET EVEN BETTER CLASSIFICATION RESULTS

Output of the neuron: discrete convolution of all input-features for one filter.



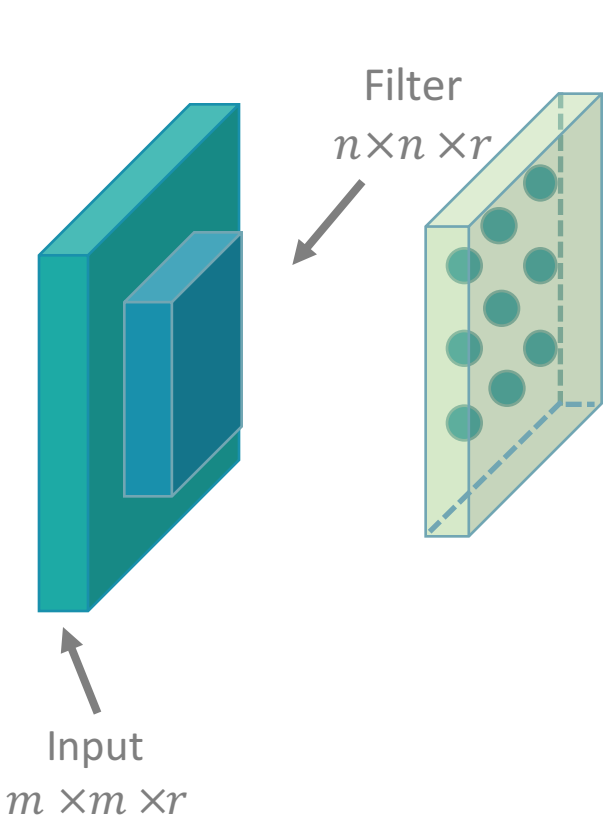


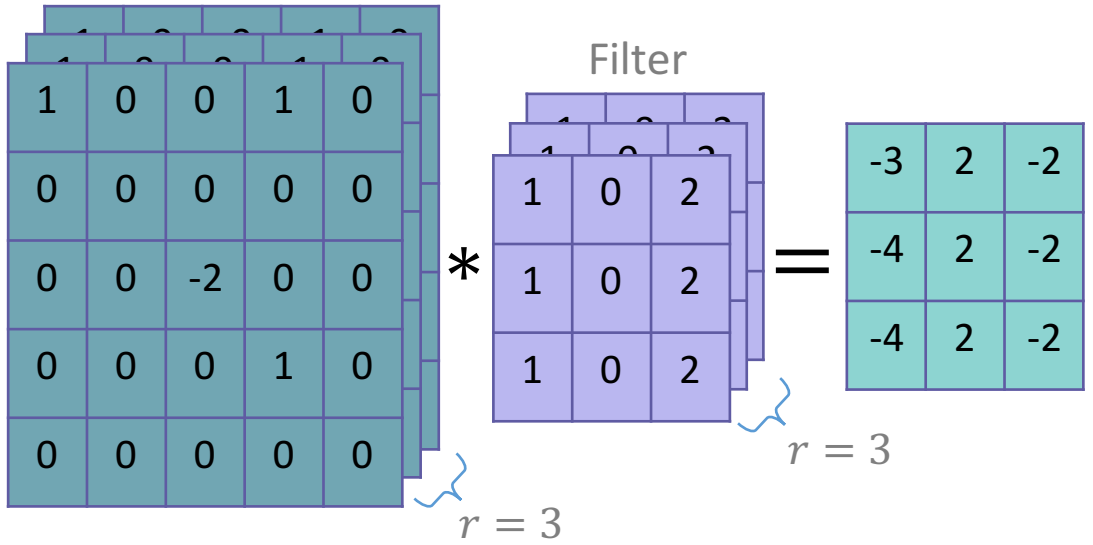
$$(I * F)_{r,s} := \sum_{u=0}^{r-1} \sum_{v=0}^{r-1} I_{r+u,s+v} \cdot F_{u,v}$$

Building Blocks of a CNN: Convolutional Layer

OR HOW TO GET EVEN BETTER CLASSIFICATION RESULTS

Output of the neuron: discrete convolution of all input-features for one filter.



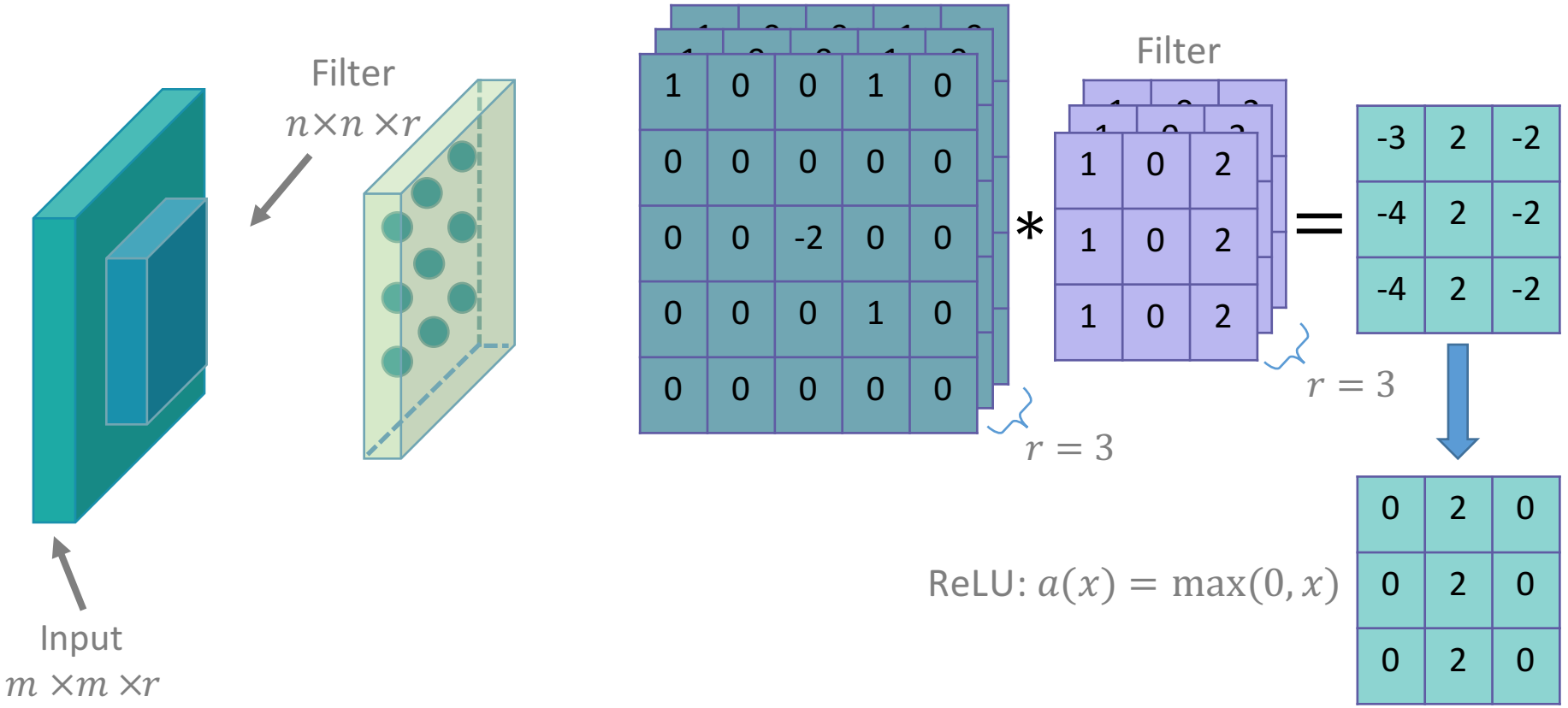


$$(I * F)_{r,s} := \sum_{u=0}^{r-1} \sum_{v=0}^{r-1} I_{r+u,s+v} \cdot F_{u,v}$$

Building Blocks of a CNN: Convolutional Layer

OR HOW TO GET EVEN BETTER CLASSIFICATION RESULTS

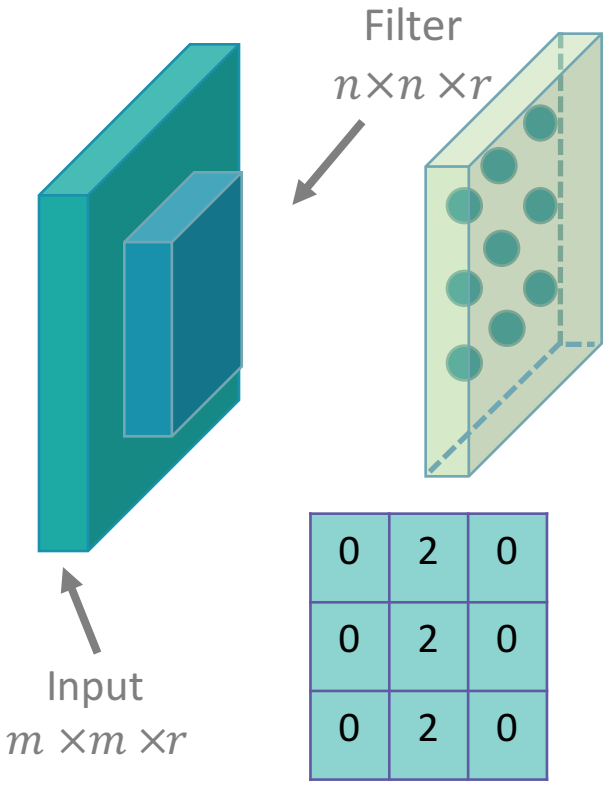
Output of the neuron: discrete convolution of all input-features for one filter.



Building Blocks of a CNN: Convolutional Layer

OR HOW TO GET EVEN BETTER CLASSIFICATION RESULTS

Output of the neuron: discrete convolution of all input-features for one filter.



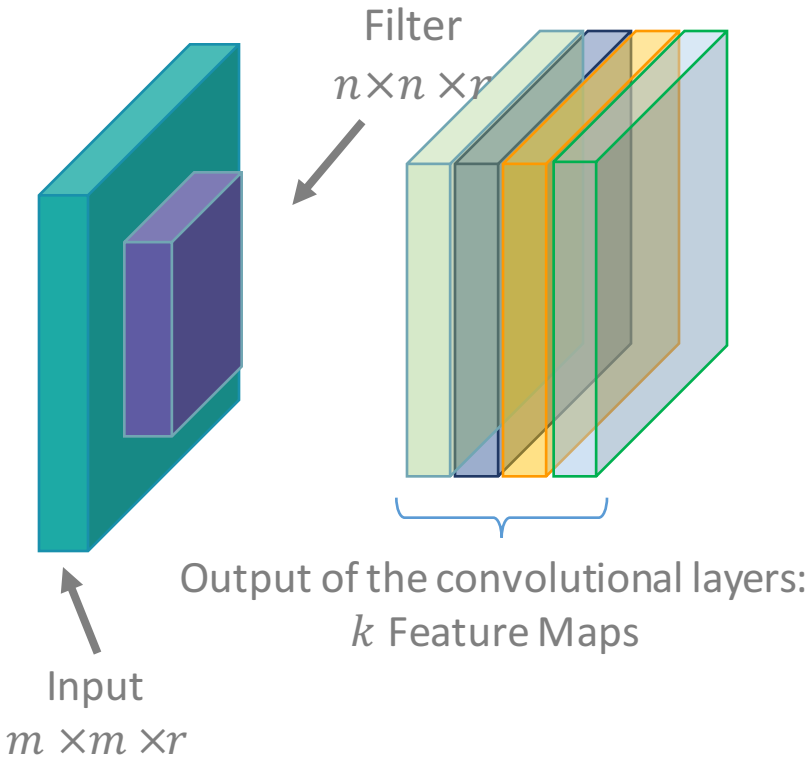
0	2	0
0	2	0
0	2	0

All neurons of a filter share the same weights ("weight sharing")!

Output: Feature Map of a filter

Building Blocks of a CNN: Convolutional Layer

OR HOW TO GET EVEN BETTER CLASSIFICATION RESULTS



Input of a convolutional layer: *r* feature maps

Output: *k* filters create *k* feature maps

Simple CNN using TensorFlow on MNIST

OR HOW TO GET EVEN BETTER CLASSIFICATION RESULTS

Live demo

cnn_mnist.ipynb

Integrated Perception

SEGMENTATION, DETECTION AND CLASSIFICATION – ALL AT ONCE

Application

- Vehicle Detection
- Road Segmentation
- Road Type Classification

Technique

- Fully Convolutional Network
- Supervised Learning

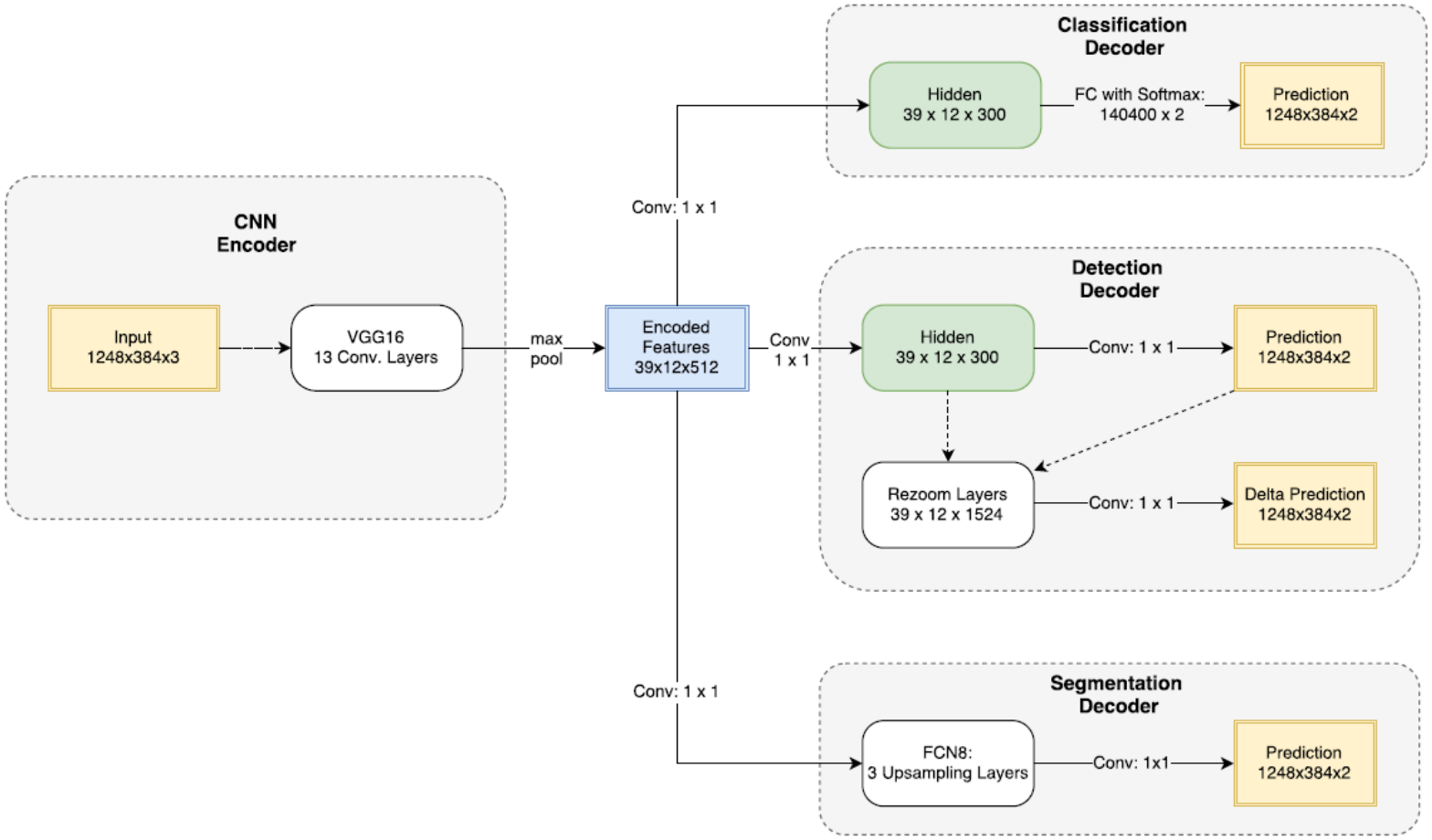
Data

- KITTI Dataset

Teichmann, Weber, Zöllner, Cipolla, Urtasun: “MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving“ arXiv preprint

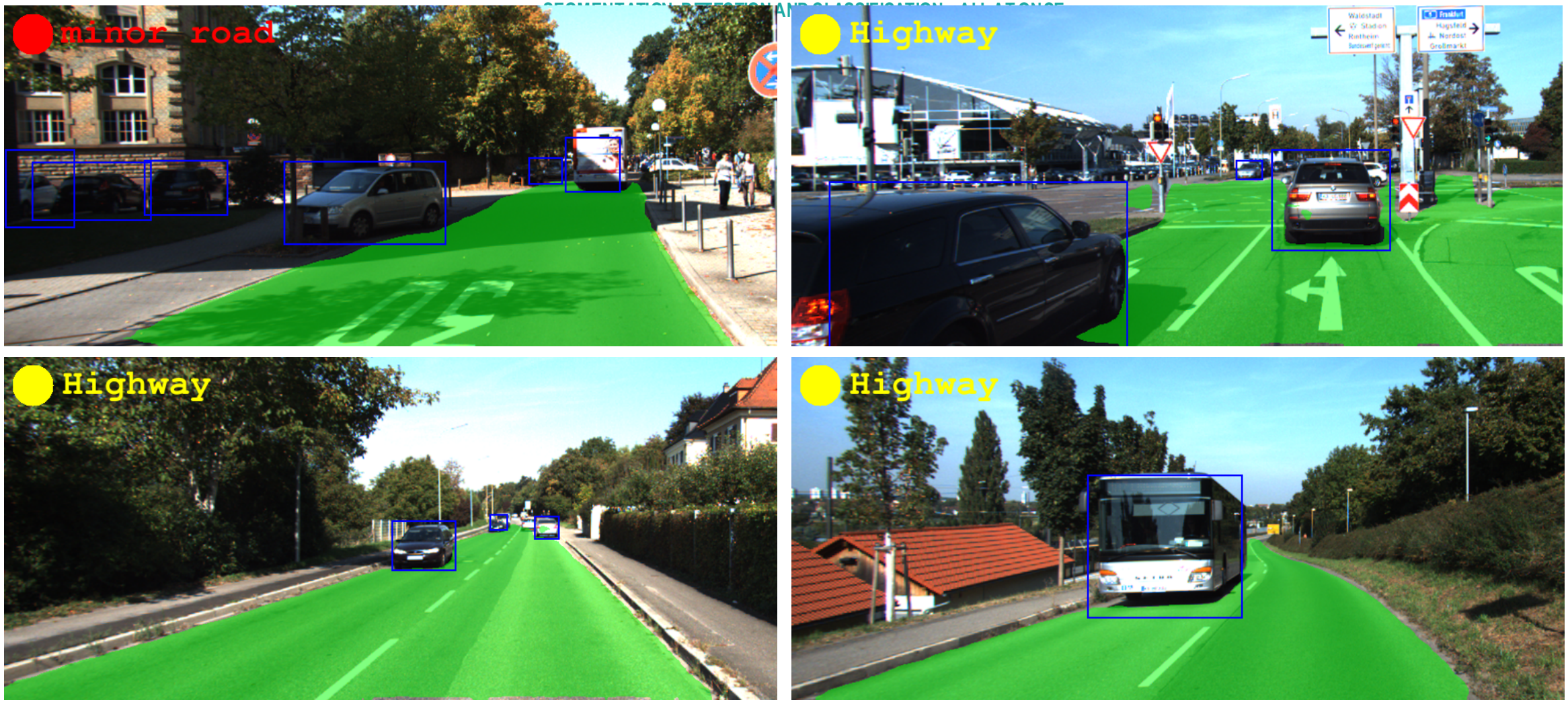
MultiNet Architecture

SEGMENTATION, DETECTION AND CLASSIFICATION – ALL AT ONCE



Teichmann, Weber, Zöllner, Cipolla, Urtasun: "MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving" arXiv preprint

MultiNet Results



Teichmann, Weber, Zöllner, Cipolla, Urtasun: “MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving“ arXiv preprint

DeepTLR

DETECTION AND CLASSIFICATION OF TRAFFIC LIGHTS

CNN to detect traffic lights

- Implicit sliding window approach
- Result contains probability maps and bound box proposals

Post-Processing

- Clustering of bounding boxes to achieve detection

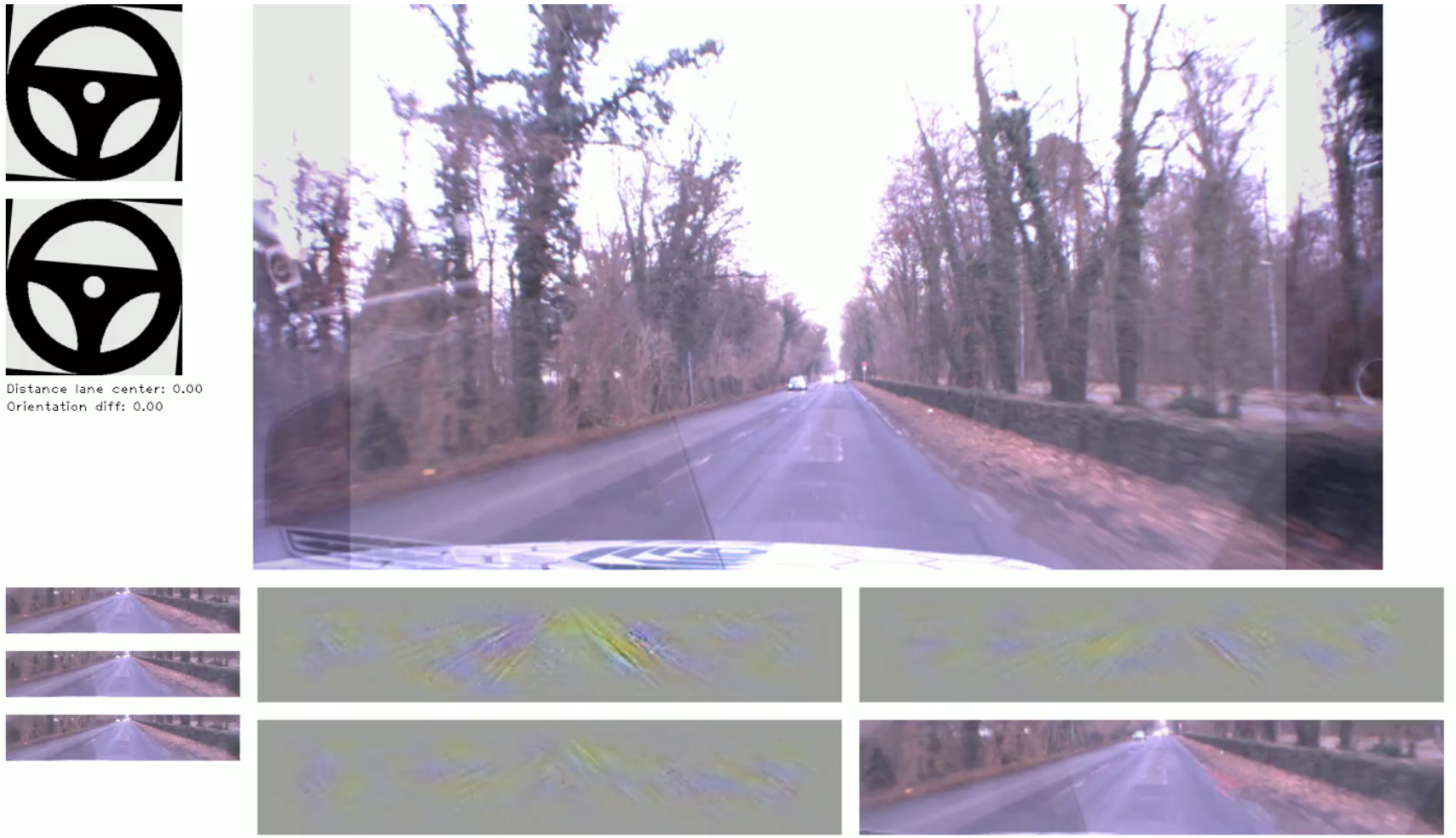
Performance

- Runnable with 33 Hz (640x480 px)
- Precision: 95.6%
- Recall: 91.4%

Weber, Wolf, Zöllner: "DeepTLR: A single deep convolutional network for detection and classification of traffic lights"" Intelligent Vehicles Symposium (IV), 2016 IEEE

End-to-End Driving

END-TO-END LEARNING FOR AUTONOMOUS DRIVING



Hubschneider, Bauer, Weber et al., 2017

TensorFlow Yourself

HOW TO GET INVOLVED?

Online Learning Resources

- TensorFlow Online Tutorial: https://www.tensorflow.org/get_started/
- Stanford CS231n. Full course available on YouTube: https://www.youtube.com/playlist?list=PLkt2uSq6rBVctENoVBg1TpCC7OQi31AlC
- .. many many more! TensorFlow has a large community and is pretty well documented.

ML-KA Team Tensorflow

- Tuesdays, 5:30 pm
- Dates will be added to Meetup!

ML-KA Einsteiger Track

- Dates will also be added to Meetup!

.. or: Come talk to me!

Conclusion

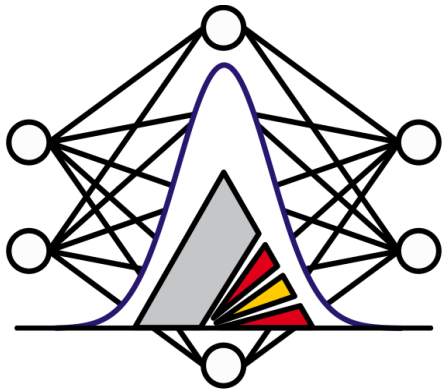
Overview over Frameworks

TensorFlow Basics

Coding Examples

TensorFlow for Autonomous Driving

How to get involved



Karlsruhe Machine Learning,
Statistics and AI Meetup

Any questions?