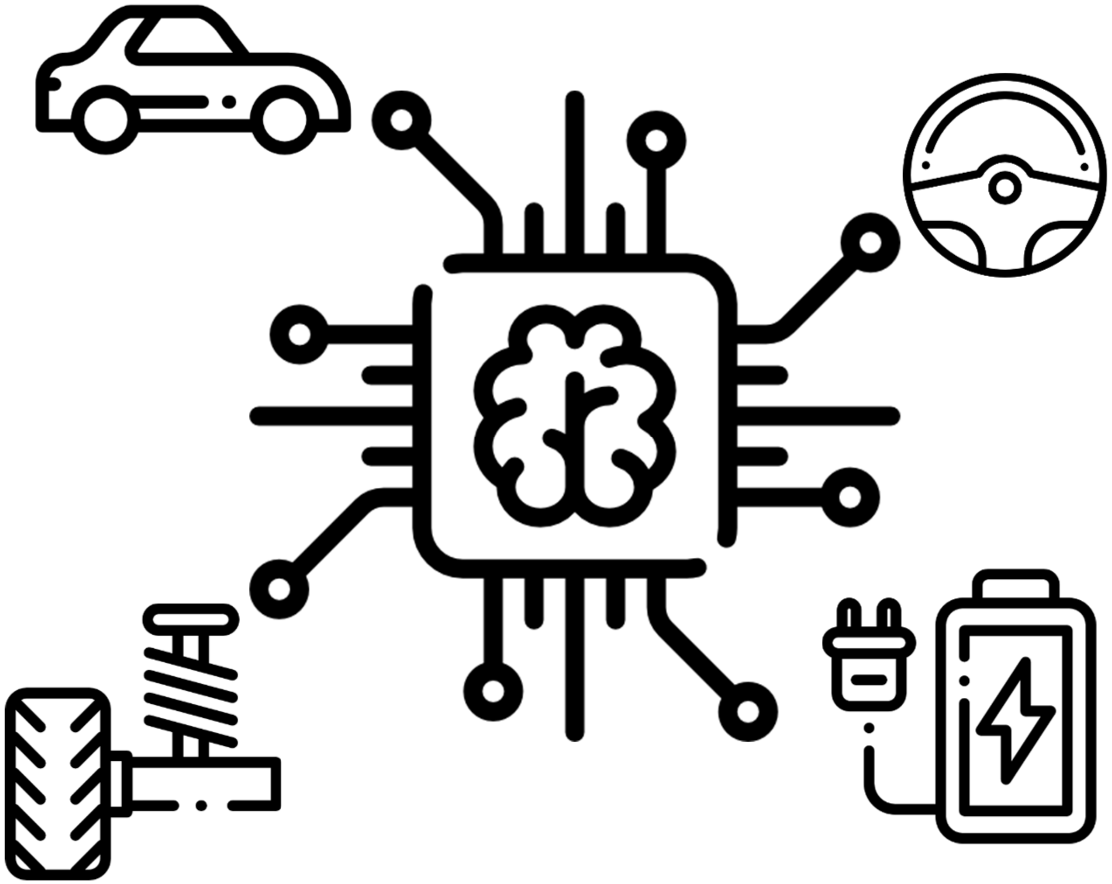
Lehrstuhl für Fahrzeugtechnik
Fakultät für Maschinenwesen
Technische Universität München

Artificial Intelligence in Automotive Technology

Johannes Betz / Prof. Dr.-Ing. Markus Lienkamp/ Prof. Dr.-Ing. Boris Lohmann

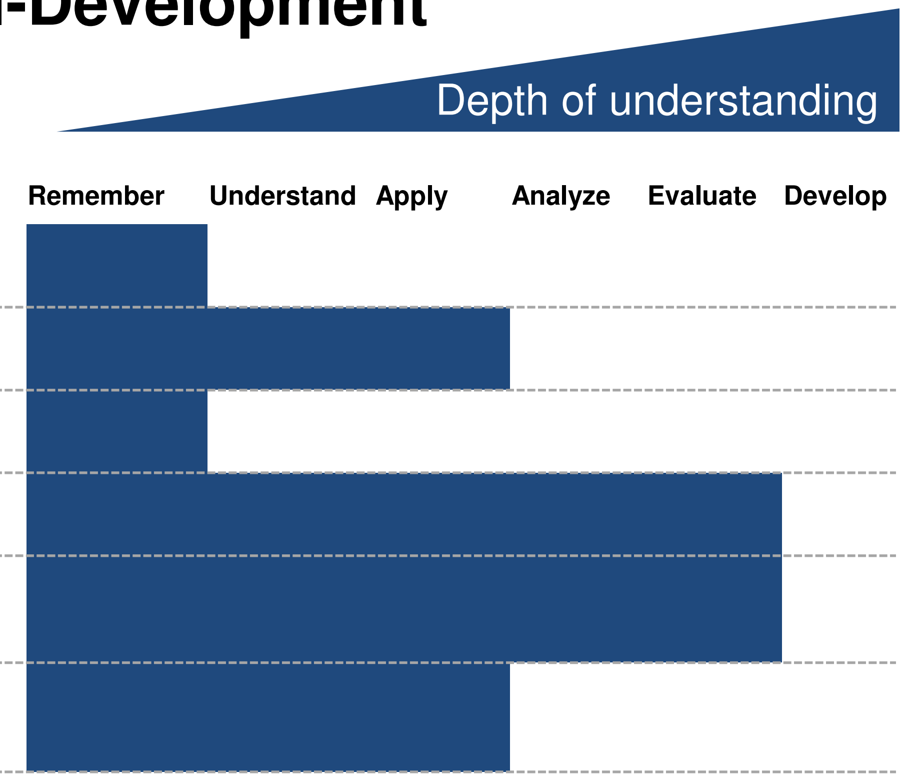


Lecture Overview

1 Introduction: Artificial Intelligence 18.10.2018 – Betz Johannes	**6 Pathfinding: From British Museum to A*** 29.11.2018 – Lennart Adenaw	**11 Reinforcement Learning** 17.01.2019 – Christian Dengler
Practice 1 18.10.2018 – Betz Johannes	**Practice 6** 29.11.2018 – Lennart Adenaw	**Practice 11** 17.01.2019 – Christian Dengler
2 Perception 25.10.2018 – Betz Johannes	**7 Introduction: Artificial Neural Networks** 06.12.2018 – Lennart Adenaw	**12 AI-Development** 24.01.2019 – Johannes Betz
Practice 2 25.10.2018 – Betz Johannes	**Practice 7** 06.12.2018 – Lennart Adenaw	**Practice 12** 24.01.2019 – Johannes Betz
3 Supervised Learning: Regression 08.11.2018 – Alexander Wischnewski	**8 Deep Neural Networks** 13.12.2018 – Jean-Michael Georg	**13 Guest Lecturer** 07.02.2019 – Rasmus Rothe
Practice 3 08.11.2018 – Alexander Wischnewski	**Practice 8** 13.12.2018 – Jean-Michael Georg	
4 Supervised Learning: Classification 15.11.2018 – Jan Cedric Mertens	**9 Convolutional Neural Networks** 20.12.2018 – Jean-Michael Georg	
Practice 4 15.11.2018 – Jan Cedric Mertens	**Practice 9** 20.12.2018 – Jean-Michael Georg	
5 Unsupervised Learning: Clustering 22.11.2018 – Jan Cedric Mertens	**10 Recurrent Neural Networks** 10.01.2019 – Christian Dengler	
Practice 5 22.11.2018 – Jan Cedric Mertens	**Practice 10** 10.01.2019 – Christian Dengler	

Objectives for Lecture 12: AI-Development

After the lecture you are able to…



… remember the pipeline for developing DL algorithms

… apply transfer learning regarding a given problem

… apply data augmentation to a given set of data

… remember the important facts of the single and multiple GPU usage in the field of DL

… analyze results from the training of DL algorithm and evaluate the hyperparameter regarding the performance of the algorithm

… analzye results from the inference of a DL algorithm and evaluate the algorithm regarding his performance

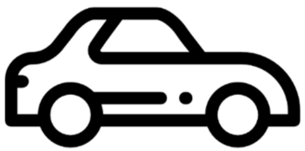
AI-Development

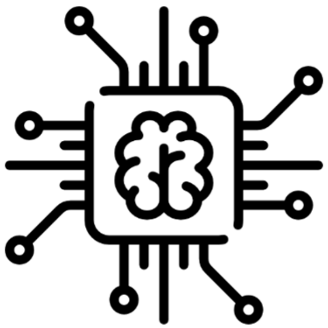
Johannes Betz / Prof. Dr. Markus Lienkamp /
Prof. Dr. Boris Lohmann

(Johannes Betz, M. Sc.)

Agenda

1. **Chapter: AI-Development Pipeline**
2. Chapter: Transfer Learning
3. Chapter: AI-Frameworks
4. Chapter: Data and Labeling
5. Chapter: GPU Computing
6. Chapter: Hyperparamter Tuning
7. Chapter: AI-Inference
8. Chapter: Summary





AI-Development

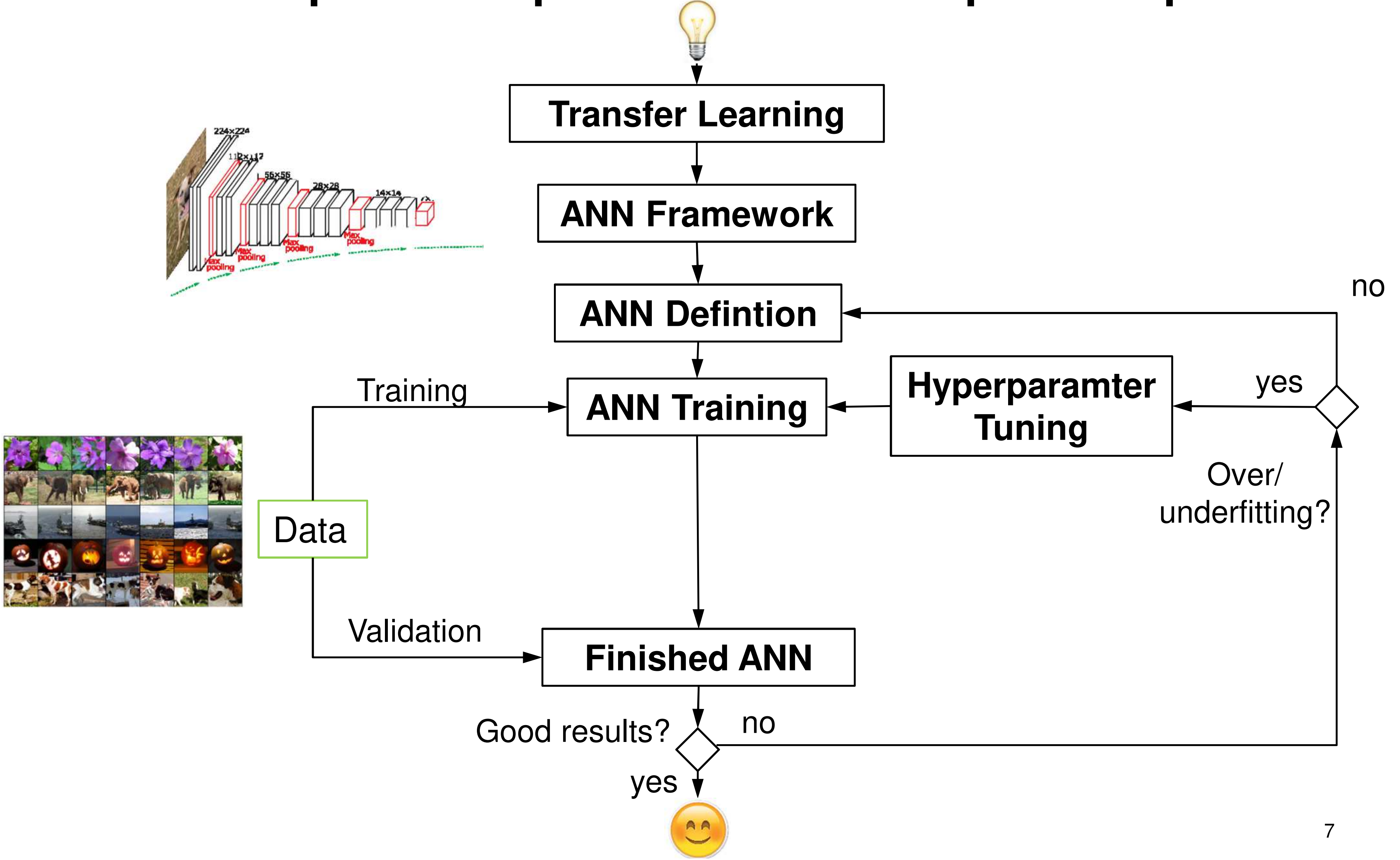
Deep Learning

Source: http://p.migdal.pl/2017/04/30/teaching-deep-learning.html

AI-Development – General AI-Development Pipeline

1. What kind of **problem** do I have?
2. What kind of **machine learning method** can I use: Regression, Classification, Clustering?
3. If i need **deep learning**, what kind of neuronal networks are useful regarding the problem I have?
4. Which **programming language** do I have to use?
5. Do i have **enough data** for the problem?
6. Can i **vary the data** I have?
7. Do i have **small scale GPU Power** for first shots?
8. Do i know how to **vary my hyperparamters**?
9. Is my problem too big but scalable so I need **multiple GPUs**?
10. What hardware do I need for the **inference**?
11. Is my hardware **connectable** with other hardware?

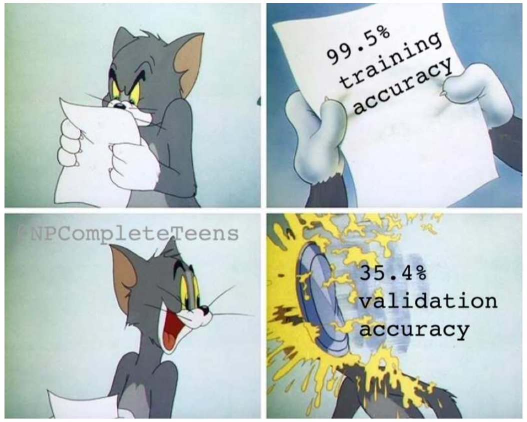
AI-Development – Specific ANN-Development Pipeline



AI-Development – Be aware!

Before you start to develop:

1. Think through the whole **general pipeline** first!
2. DL Development ≠ DL Training ≠ DL Inference



General Scaling up:

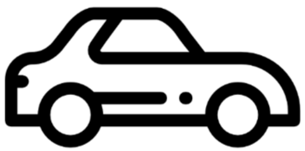
1. Make bigger models: More Layers, bigger Layers,..
2. Tackle more data: More data, variation of data, data augmentation,…
3. Reduce research cycle time with fast computing: Parallel computing, GPU usage…

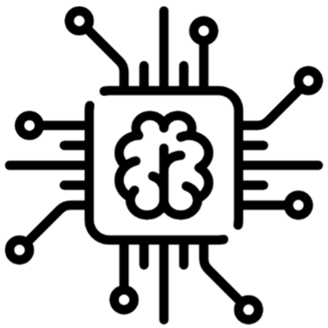
AI-Development
Johannes Betz / Prof. Dr. Markus Lienkamp / Prof. Dr. Boris Lohmann

(Johannes Betz, M. Sc.)

Agenda

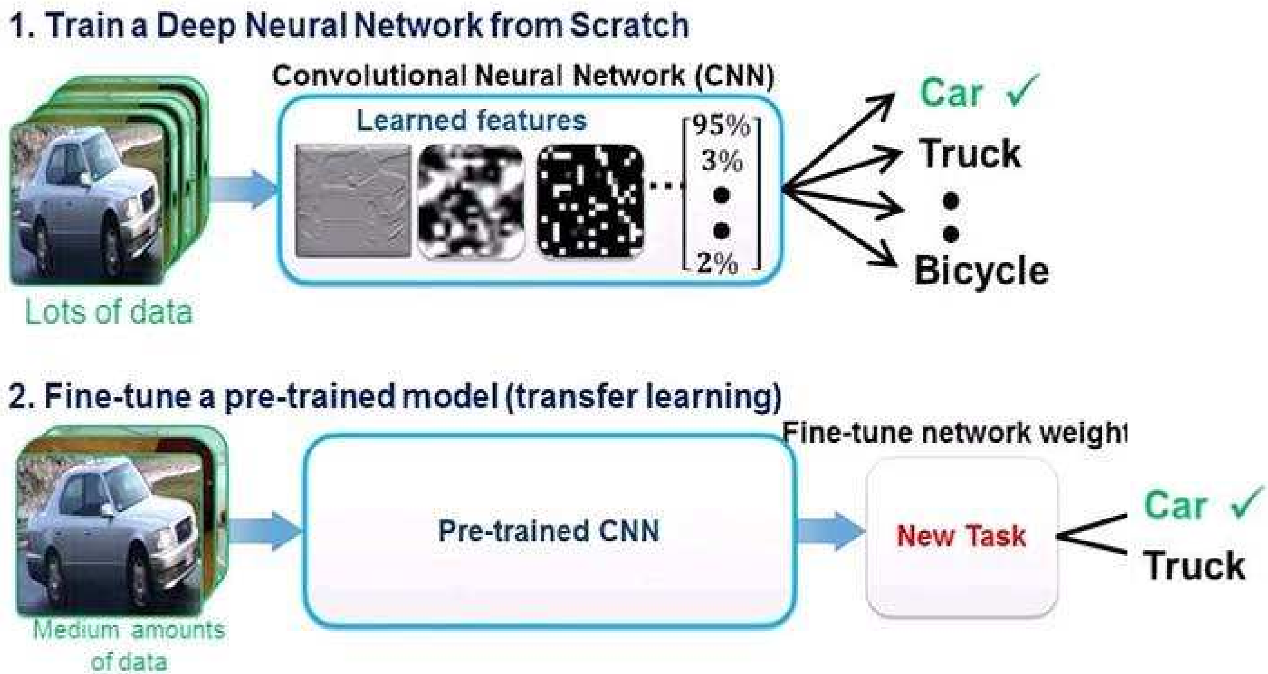
1. Chapter: AI-Development Pipeline
2. **Chapter: Transfer Learning**
3. Chapter: AI-Frameworks
4. Chapter: Data and Labeling
5. Chapter: GPU Computing
6. Chapter: Hyperparamter Tuning
7. Chapter: AI-Inference
8. Chapter: Summary





Transfer Learning

- When you tackeling a new problem with an ANN, it might help to look at **existing ANNs** that were built for a similar task
- **Accelerate your work:** People but a lot of effort in developing the architecture and training the ANN
- Take this ANN and adjust it for your problem

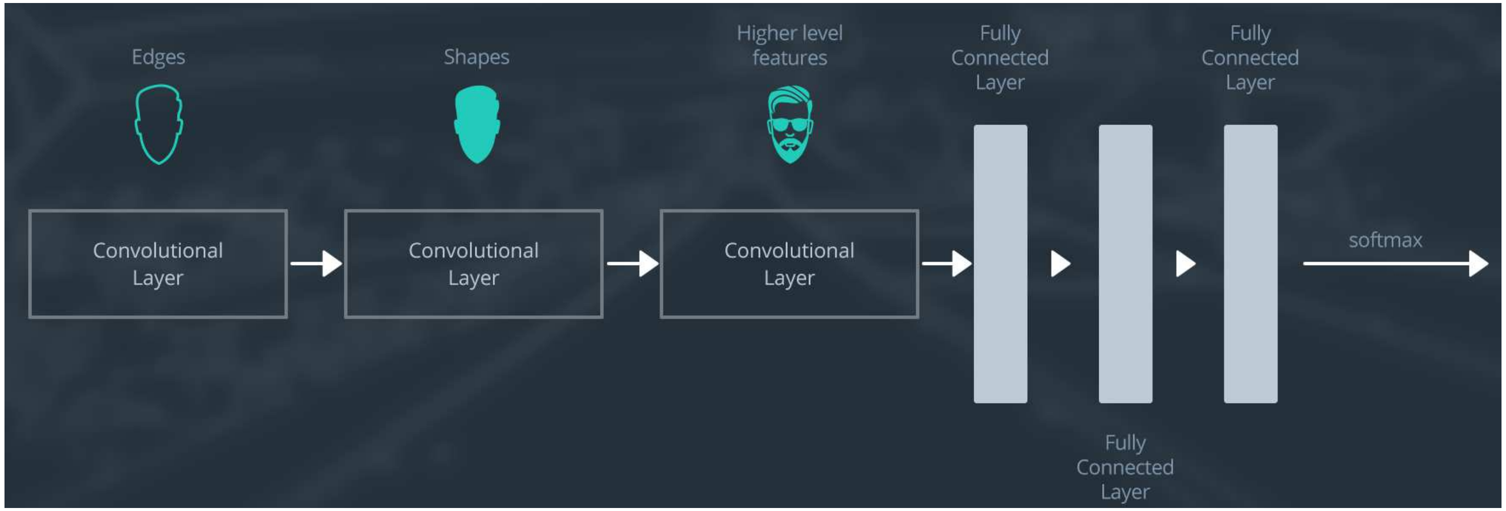


Source: https://de.mathworks.com/videos/training-deep-learning-models-with-transfer-learning-1486670648501.html

Transfer Learning

Examples:

- **AlexNet:** CNN for classification
- **VGG:** CNN for classification
- **GoogLeNet:** CNN for classification and detection
- **MobileNet:** ObjectDetection, Object Classification, Landmark Recognition



Example Pretrained CNN

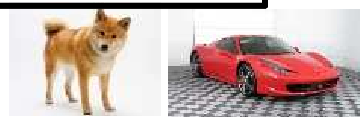
Source: Udacity Self-Driving Car Nanodegree – Part 1, Lesson 15

Transfer Learning – What can we do?

1. **Vary Layers:** Vary layer in the end or the beginngy of the ANN, the rest of the network remains fixed
2. **Finetuning**: Train the entire network end-to-end, start with **pre-trained** weights
3. **Training from scratch:** Train the entire network end-to-end, start from **random** weights

Size of Dataset

	Similar	Different
Large	**Fine-tune the ANN**	**Fine-tune or Retrain the ANN**
Small	**Vary the end of the ANN**	**Vary the beginning levels of the ANN**

Similar  Different 

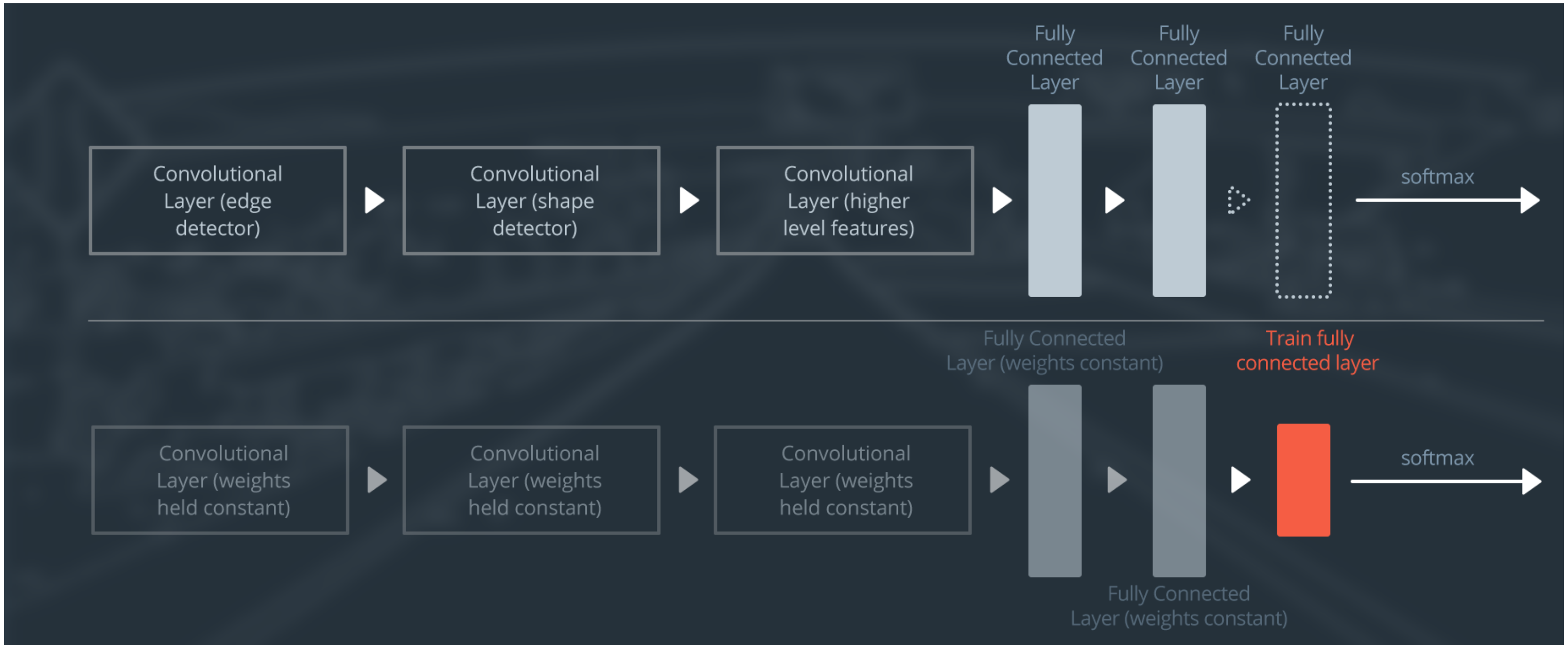
Similarity to Training Data

Transfer Learning – Small Data Set & Similar Data

Consider vary the end of the ANN when ...

... the new dataset is small and similar to the original dataset. The higher-level features learned from the original dataset should transfer well to the new dataset.

Old ANN | New ANN

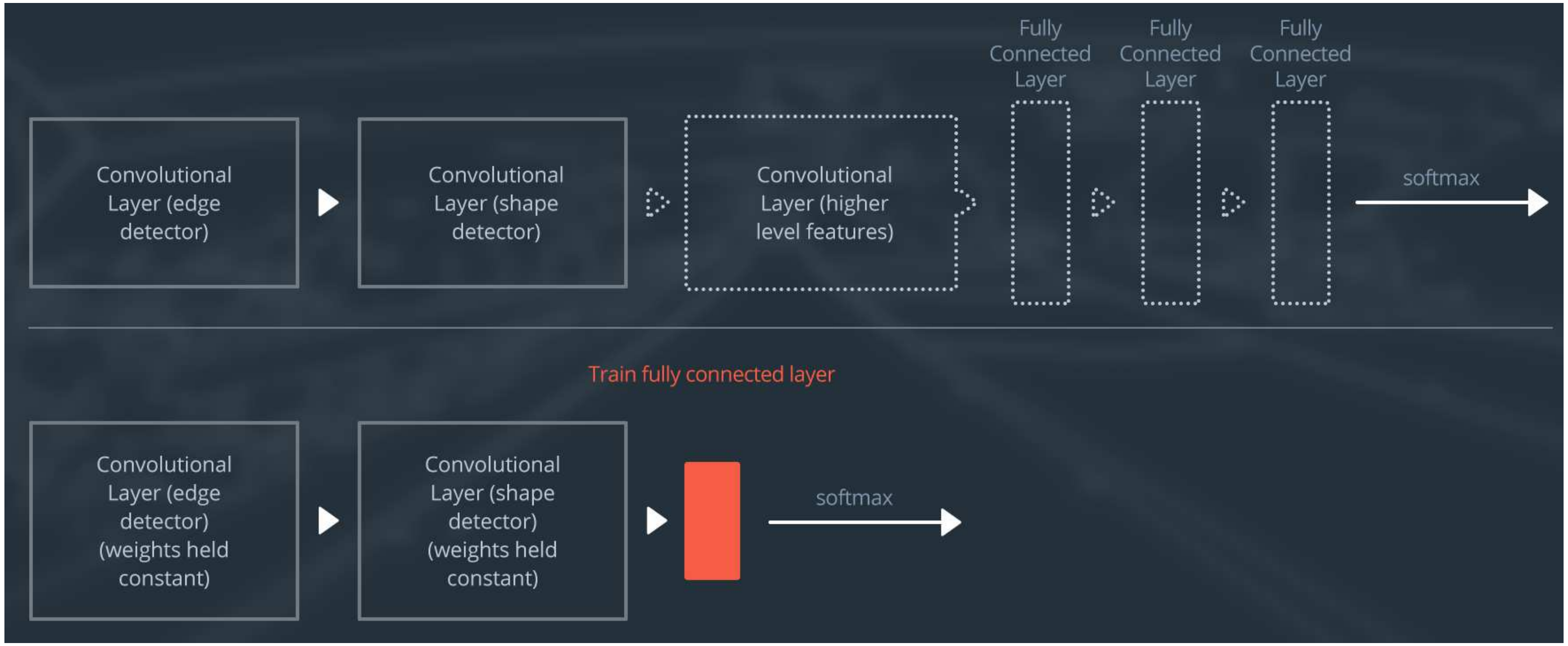


Source: Udacity Self-Driving Car Nanodegree – Part 1, Lesson 15

Transfer Learning - Small Data Set & Different Data

Consider vary the beginning of the ANN when ...
…the new dataset is small and very different from the original dataset. You could also make the case for training from scratch. In this case we will only use features from the first few layers of the pre-trained network → features from the final layers of the pre-trained network might be too specific to the original dataset.

Old ANN | New ANN

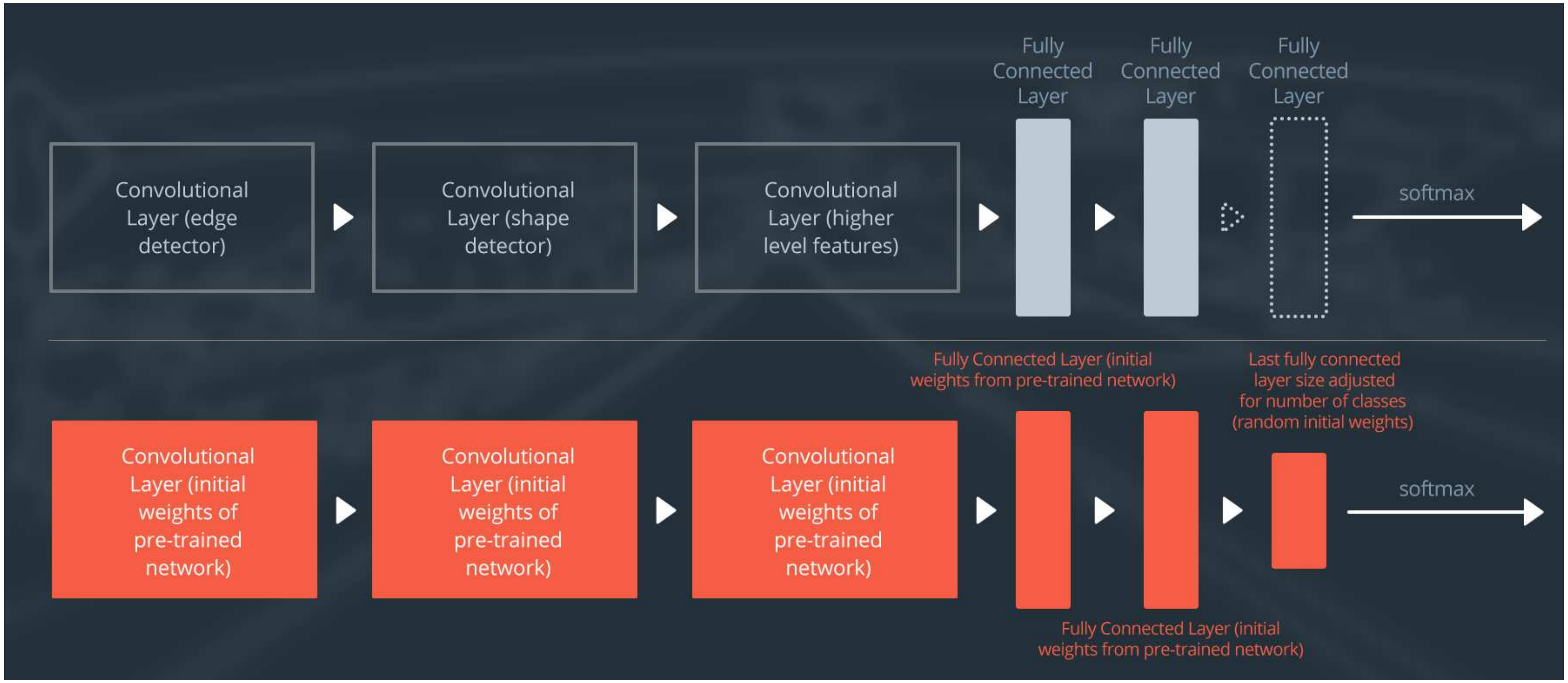


Source: Udacity Self-Driving Car Nanodegree – Part 1, Lesson 15

Transfer Learning - Large Data Set & Similar Data

Consider finetuning when ...
... the new dataset is large and similar to the original dataset. Altering the original weights should be safe because the network is unlikely to overfit the new, large dataset.

New ANN | Old ANN



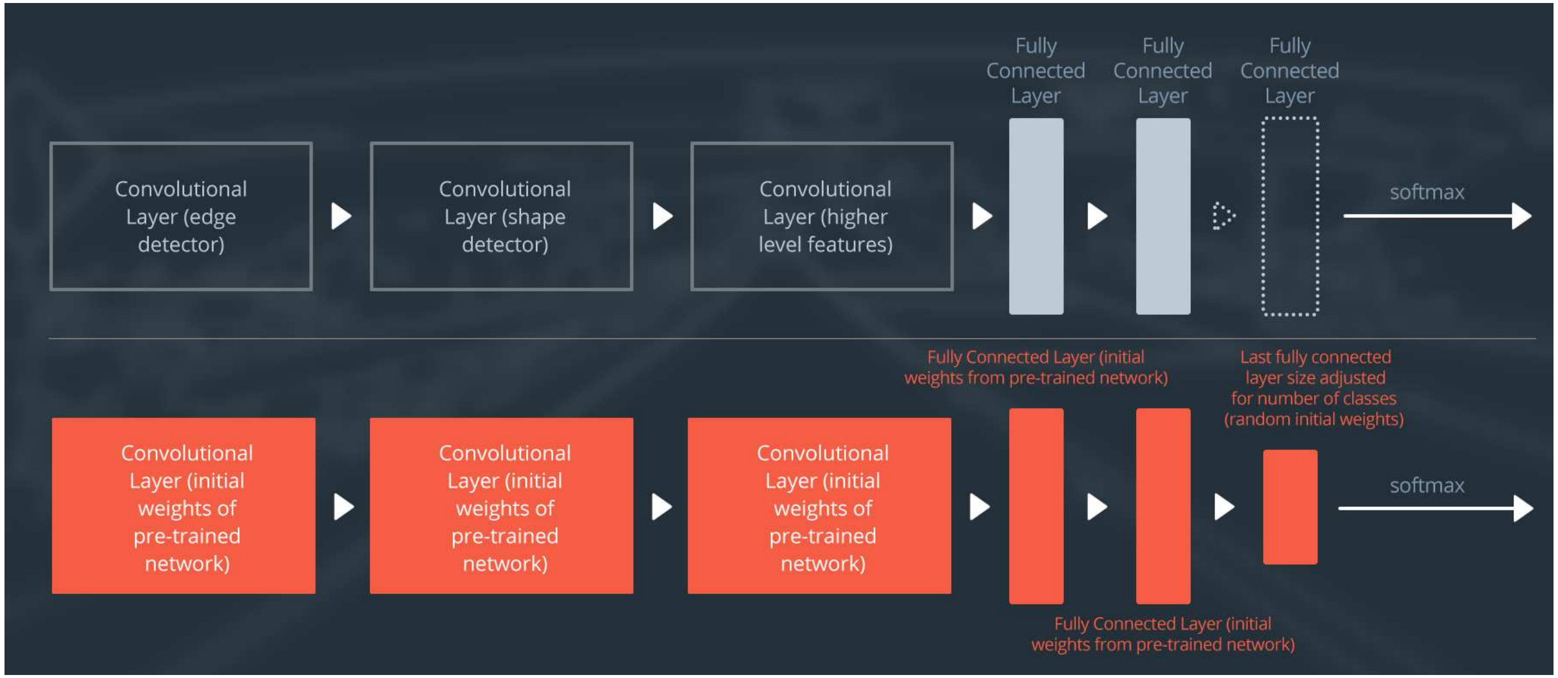
Source: Udacity Self-Driving Car Nanodegree – Part 1, Lesson 15

Transfer Learning – Large Data Set & Different Data

Consider training from scratch when ...

... the dataset is large and very different from the original dataset. In this case we have enough data to confidently train from scratch. However, even in this case it might be beneficial to initialize the entire network with pretrained weights and finetune it on the new dataset.

New ANN | Old ANN



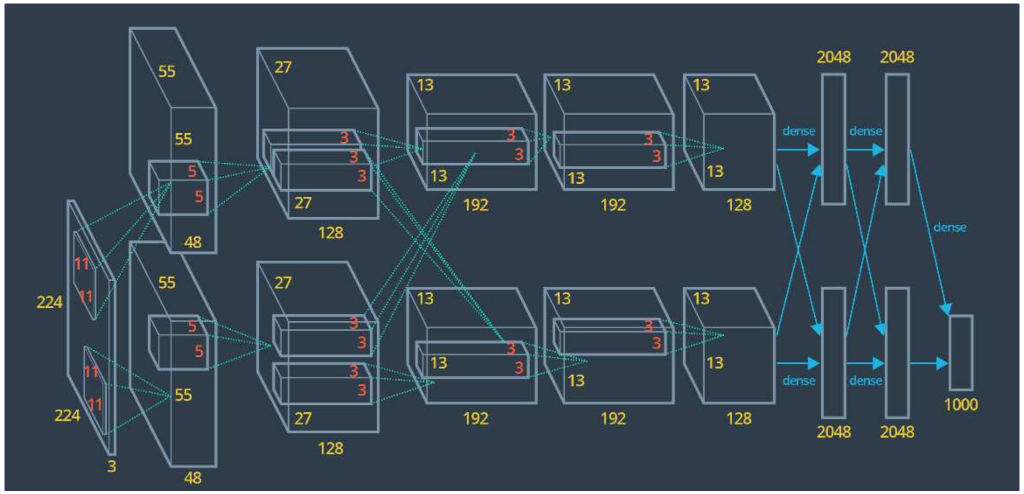
Source: Udacity Self-Driving Car Nanodegree – Part 1, Lesson 15

Additional Slide

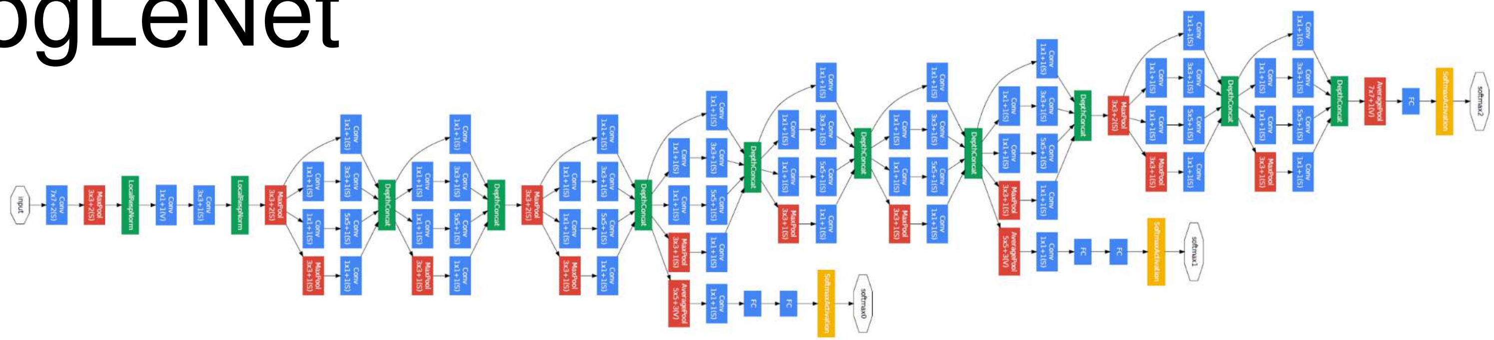
Keep in mind that for a lot of problems you won't need an architecture as complicated and powerful as VGG, Inception, or ResNet.

These architectures were made for the task of classifying thousands of complex classes. A smaller network might be a better fit for a smaller problem, especially if you can comfortably train it on moderate hardware.

AlexNet



GoogLeNet



Source: Udacity Self-Driving Car Nanodegree – Part 1, Lesson 15 / https://medium.com/coinmonks/paper-review-of-googlenet-inception-v1-winner-of-ilsvlc-2014-image-classification-c2b3565a64e7

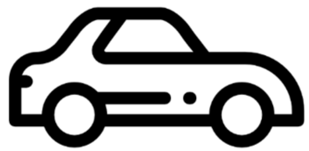
AI-Development

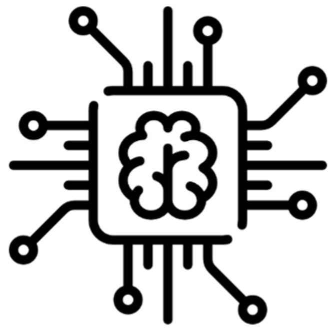
Johannes Betz / Prof. Dr. Markus Lienkamp /
Prof. Dr. Boris Lohmann

(Johannes Betz, M. Sc.)

Agenda

1. Chapter: AI-Development Pipeline
2. Chapter: Transfer Learning
3. **Chapter: AI-Frameworks**
4. Chapter: Data and Labeling
5. Chapter: GPU Computing
6. Chapter: Hyperparamter Tuning
7. Chapter: AI-Inference
8. Chapter: Summary





AI-Frameworks – Whats that ?

- AI software can be developed from scratch, but it is tediously and complex

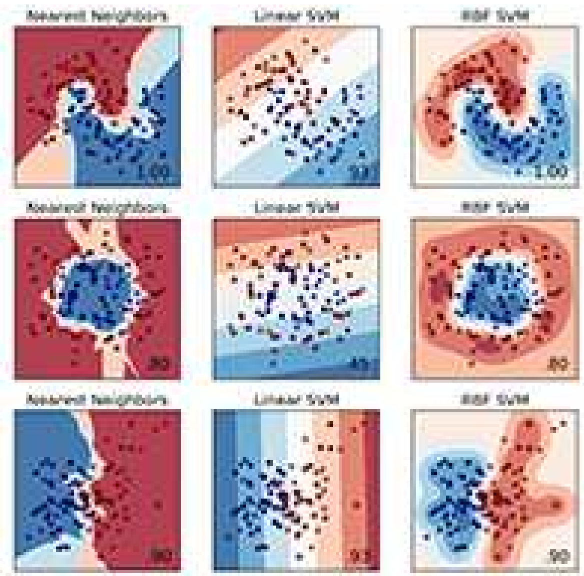
- A lot of people put time and effort in the development computer programs, application programming interface (APIs), libraries and software frameworks, which make the development much easier

  → **USE IT !!!!!**

- A **software framework** is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code
  - The software framework provides a standard way to build and deploy applications
  - Better then normal library: Inversion of control, extensibility, non modifiable framework code

AI-Frameworks – Scikit-learn Library

- Free software framework
- Machine Learning Library
- **Language:** Python, C, C++
- **Operating System:** Linux, macOS, Windows
- **Includes:** Clustering, Regression, Clustering with algorithms like SVM, Nearest Neighbors, Gaussian Process, Decision Trees
- **Pros:** Everything you need, Good Documentation, powerful, GPU boost
- **Cons:** Not for hardcore statistics, limited in parameters



Source: https://scikit-learn.org/stable/

AI-Frameworks – Matlab

- Commercial Software (Free for **students)**
- Machine and Deep Learning Toolbox
- **Language:** Matlab, Simulink
- **Operating System:** Linux, macOS, Windows
- **Includes:**
  - Machine Learning
  - Deep Learing
- **Pros:** Easy to use, good documentation, GPU boost
- **Cons:** Closed Environment, Performance

Source: https://www.cbcity.de/portfolio/matlab-simulink

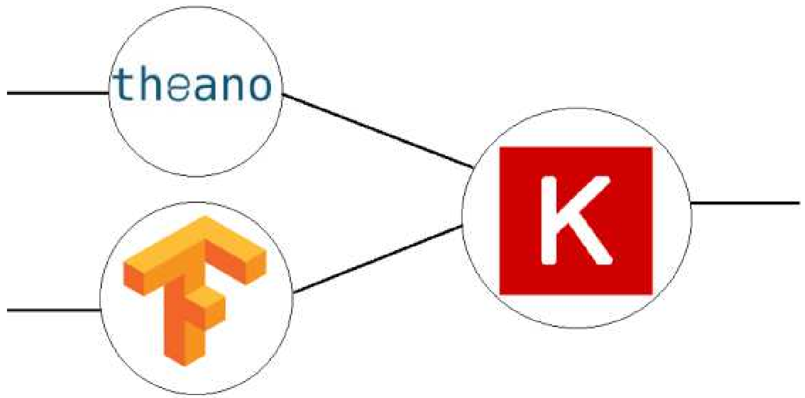
AI-Frameworks – Tensorflow

- Free software framework
- Deep Learning software framework
- **Language:** Python, C++
- **Operating System:** Linux, macOS, Windows
- **Includes:** CNN, RNN, → Voice and image recognition
- **Pros:** High performance, multiple GPU, connects research and Production, true portability, tensorboard for visualization, good documentation
- **Cons:** Hard to learn in comparison to other frameworks

Source: https://www.tensorflow.org/

AI-Frameworks – Keras

- Free software framework
- Neural network library
- **Language:** Python
- **Operating System:** Linux, macOS, Windows
- **Includes:** Neural Network interface for CNN and RNN – Speach recognition, image classification
- **Pros:** Makes Tensorflow and Theano easier to use, easy prototyping, fully configurable modles, GPU support
- **Cons:** It might be too high-level and not always easy to customize



Source: https://keras.io/

AI-Frameworks – Caffe and Caffe 2

- Free software framework
- Deep Learning software framework
- **Language:** C, C++, Python, MATLAB
- **Operating System:** Linux, macOS, Windows
- **Includes: CNN**
- **Pros:** Pre-trained networks available, fast and scalable, GPU usage
- **Cons:** a few input formats and only one output format, no exact layer definition like Tensorflow

Source: https://caffe2.ai/

AI-Frameworks – mxnet

- Free software framework
- Deep Learning software framework
- **Language:** Python, C++, Javascript, R,….
- **Operating System:** Linux, macOS, Windows
- **Includes:** CNN, RNN, LSTM
- **Pros:** Fast and flexible, GPU support, can run on any device (productivity), highly scalable, different languages
- **Cons:** Small community

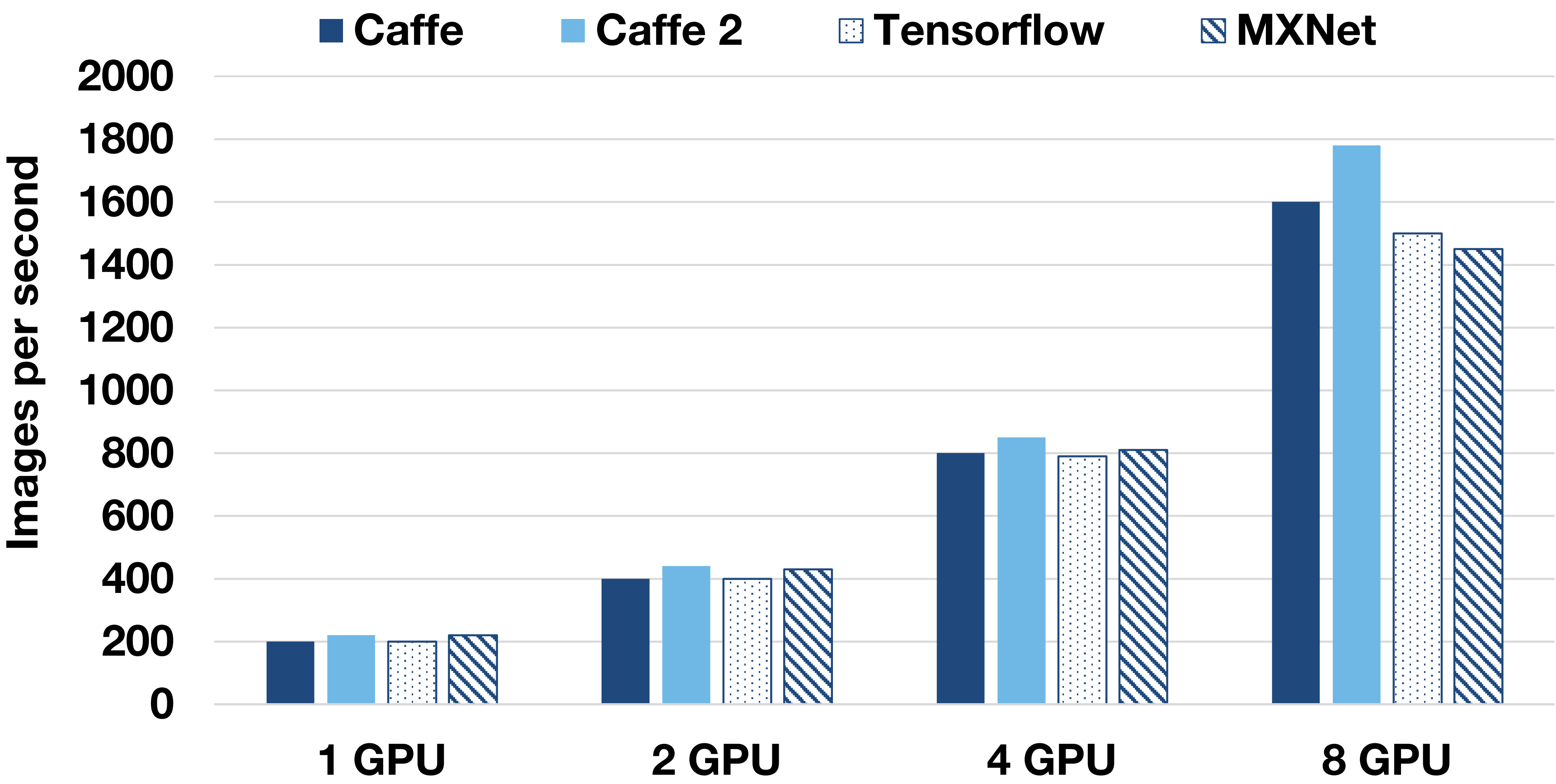
Source: https://mxnet.apache.org/

Additional Slide

	Languages	Tutorials and training materials	CNN modeling capability	RNN modeling capability	Architecture: easy-to-use and modular front end	Speed	Multiple GPU support	Keras compatible
Theano	Python, C++	++	++	++	+	++	+	+
Tensor-Flow	Python	+++	+++	++	+++	++	++	+
Torch	Lua, Python (new)	+	+++	++	++	+++	++	
Caffe	C++	+	++		+	+	+	
MXNet	R, Python, Julia, Scala	++	++	+	++	++	+++	
Neon	Python	+	++	+	+	++	+	
CNTK	C++	+	+	+++	+	++	+	

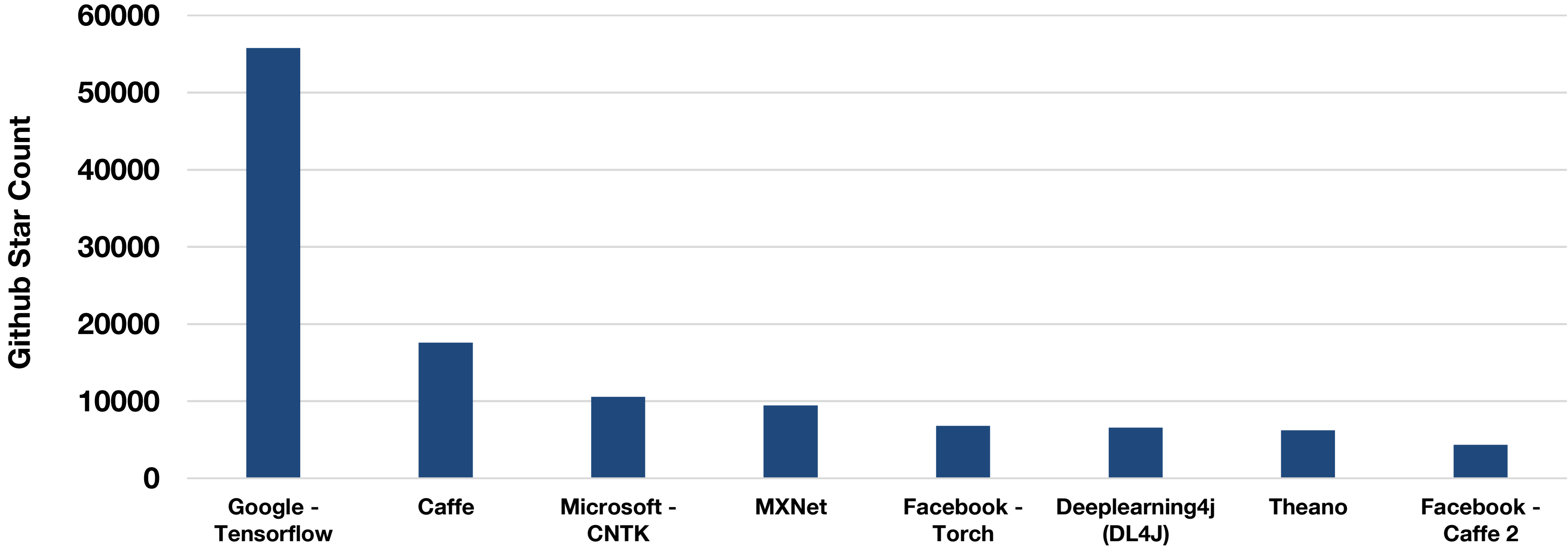
Quelle: https://www.datasciencecentral.com/profiles/blogs/open-source-deep-learning-frameworks-and-visual-analytics

AI Frameworks - Comparison



RESNET-50 FP32 Performance

AI Frameworks - Comparison



new contributors from 2017-02-11 to 2017-04-12		
#1:	131	tensorflow/tensorflow
#2:	63	fchollet/keras
#3:	51	pytorch/pytorch
#4:	49	dmlc/mxnet
#5:	18	Theano/Theano
#6:	11	BVLC/caffe
#7:	11	Microsoft/CNTK
#8:	9	tflearn/tflearn
#9:	9	pfnet/chainer
#10:	8	torch/torch7
#11:	5	deeplearning4j/deeplearning4j
#12:	4	NVIDIA/DIGITS
#13:	3	baidu/paddle

new forks from 2017-02-11 to 2017-04-12		
#1:	4192	tensorflow/tensorflow
#2:	991	fchollet/keras
#3:	810	BVLC/caffe
#4:	517	deeplearning4j/deeplearning4j
#5:	414	dmlc/mxnet
#6:	307	pytorch/pytorch
#7:	244	Microsoft/CNTK
#8:	211	tflearn/tflearn
#9:	134	torch/torch7
#10:	131	Theano/Theano
#11:	116	baidu/paddle
#12:	88	NVIDIA/DIGITS
#13:	55	pfnet/chainer

Source: https://www.cio.com/article/3193689/artificial-intelligence/which-deep-learning-network-is-best-for-you.html / http://p.migdal.pl/2017/04/30/teaching-deep-learning.html

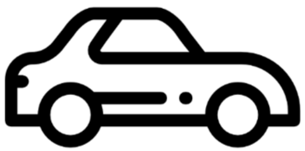
AI-Development

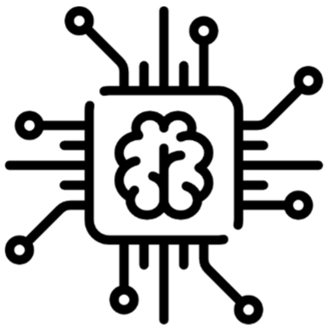
Johannes Betz / Prof. Dr. Markus Lienkamp /
Prof. Dr. Boris Lohmann

(Johannes Betz, M. Sc.)

Agenda

1. Chapter: AI-Development Pipeline
2. Chapter: Transfer Learning
3. Chapter: AI-Frameworks
4. **Chapter: Data and Labeling**
5. Chapter: GPU Computing
6. Chapter: Hyperparamter Tuning
7. Chapter: AI-Inference
8. Chapter: Summary

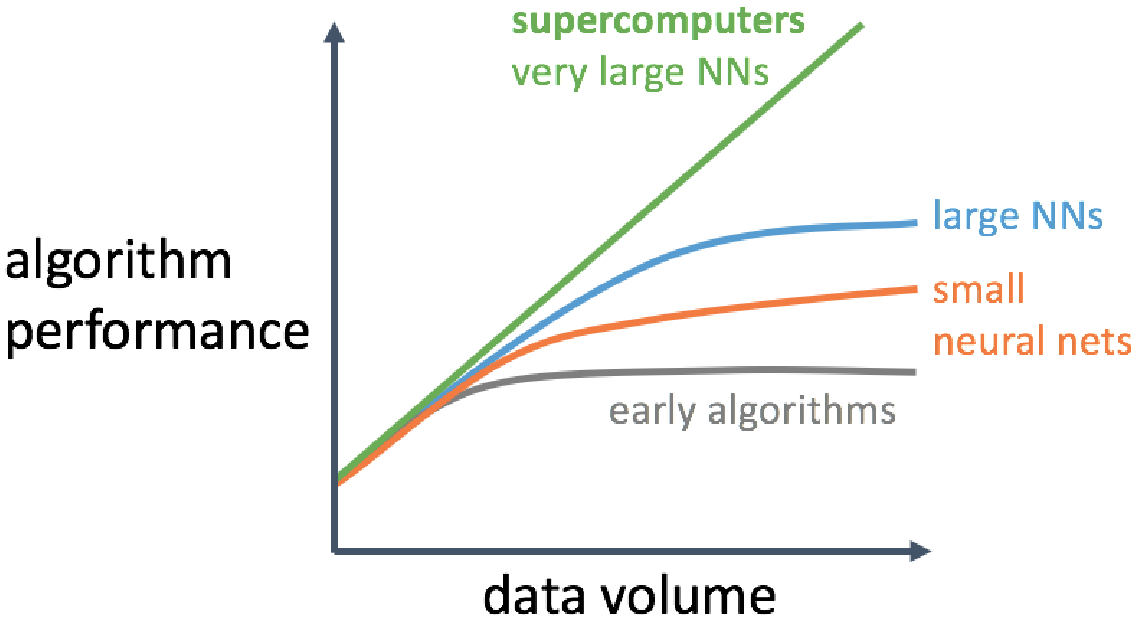




Data

Data is the crucial part of Machine Learning:

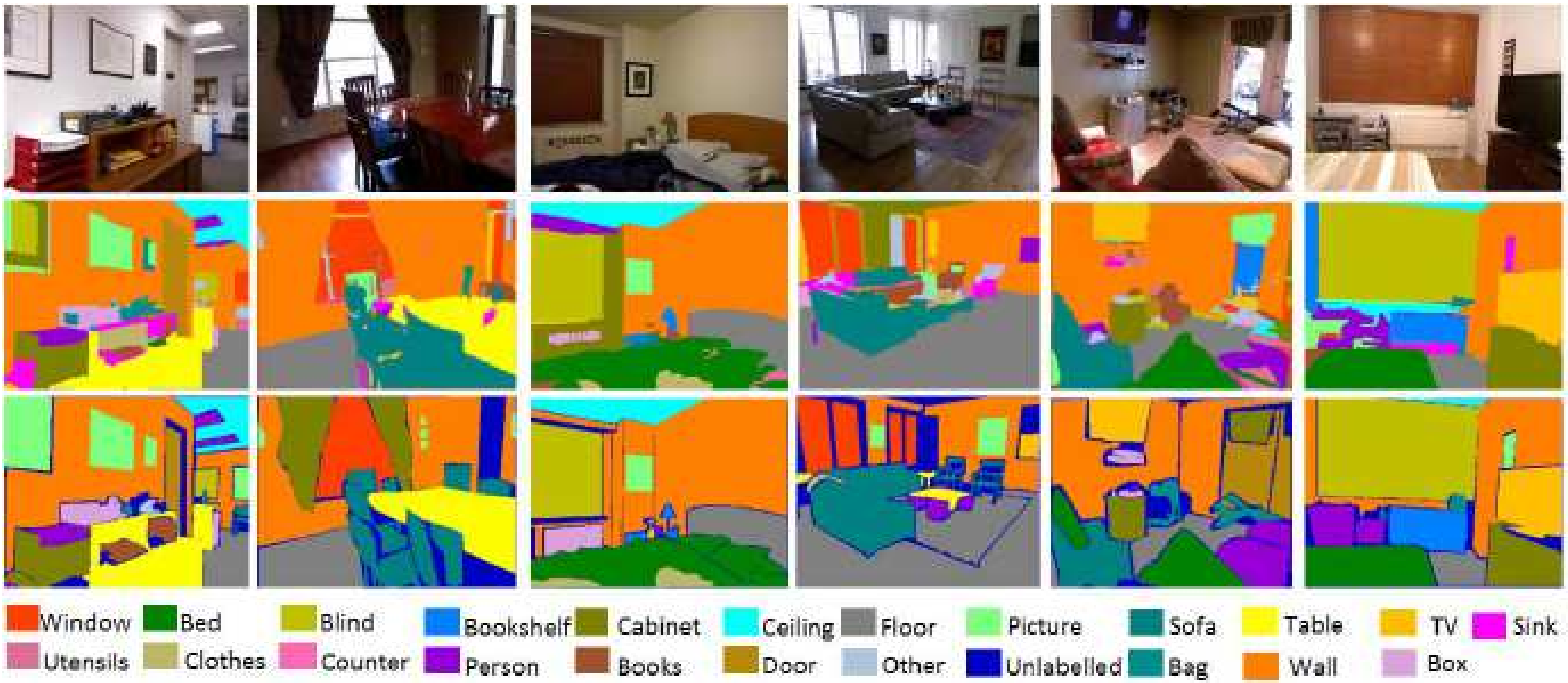
- We need **data** for training our algorithms
- We need **labeled data** for training our algorithms
- We need **more and different data** for not overfitting our training
- We need **even more data** for good regression, clustering, classification



Source: http://visualdl.paddlepaddle.org/docs/develop/documentation/fluid/en/design/memory/memory_optimization.html

Data – Labeled Datesets

- We need data that is labeled
- Label = What does the data include?
- The more specific, the better the regression, classification, clustering



Source: https://www.researchgate.net/publication/274780609_Geometry_Driven_Semantic_Labeling_of_Indoor_Scenes/figures?lo=1

Data – Labeled Datesets

1. Search for datasets online, other people have done a lot in this area
   - **Cityscapes:** Pixel based label of streets
   - **Kitti Dataset:** Pixeld based label of over 5000 pictures
   - **Berkeley Deep Drive:** Labeld pictures of streets, GPS locations, IMU data…

2. Create your own dataset
   - Aggregate your data
   - Label the data with the information you want to be later detected
   - Takes a lot of time

Source: https://www.cityscapes-dataset.com/examples/

Data – Labeled Datesets


Forbes

167,739 views | Jan 17, 2019, 12:35 pm

Was The Facebook '10 Year Challenge' A Way To Mine Data For Facial Recognition AI?

Nicole Martin Contributor

AI & Big Data

I write about technology, data and privacy.

Source: https://www.sadanduseless.com/ten-year-challenge/
https://www.forbes.com/sites/nicolemartin1/2019/01/17/was-the-facebook-10-year-challenge-a-way-to-mine-data-for-facial-recognition-ai/#27e43bcf5859

Additional Slide

Berkeley DeepDrive - Explore 100,000 HD video sequences of over 1,100-hour driving experience across many different times in the day, weather conditions, and driving scenarios. Our video sequences also include GPS locations, IMU data, and timestamps.

Udacity - Udacity driving datasets released for Udacity Challenges. Contains ROSBAG training data. (~80 GB).

Comma.ai - 7 and a quarter hours of largely highway driving. Consists of 10 videos clips of variable size recorded at 20 Hz with a camera mounted on the windshield of an Acura ILX 2016. In parallel to the videos, also recorded some measurements such as car's speed, acceleration, steering angle, GPS coordinates, gyroscope angles. These measurements are transformed into a uniform 100 Hz time base.

KITTI Vision Benchmark Suite - 6 hours of traffic scenarios at 10-100 Hz using a variety of sensor modalities such as highresolution color and grayscale stereo cameras, a Velodyne 3D laser scanner and a high-precision GPS/IMU inertial navigation system.

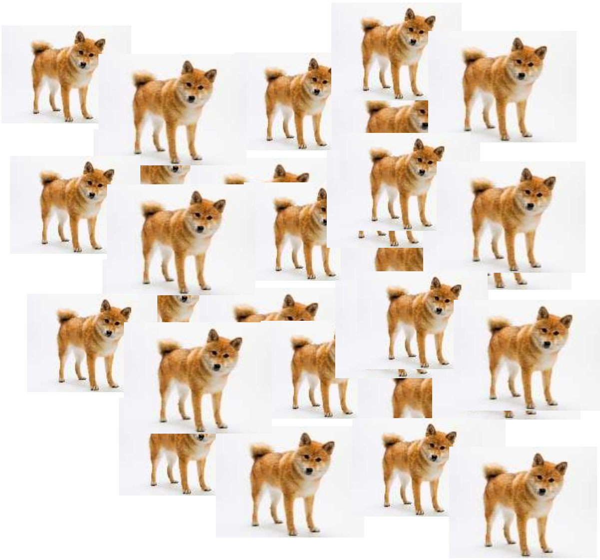
University of Michigan North Campus Long-Term Vision and LIDAR Dataset - consists of omnidirectional imagery, 3D lidar, planar lidar, GPS, and proprioceptive sensors for odometry collected using a Segway robot. pedestrian, cyclist, lanemarking.

Cityscape Dataset - focuses on semantic understanding of urban street scenes. large-scale dataset that contains a diverse set of stereo video sequences recorded in street scenes from 50 different cities, with high quality pixel-level annotations of 5 000 frames in addition to a larger set of 20 000 weakly annotated frames. The dataset is thus an order of magnitude larger than similar previous attempts. Details on annotated classes and examples of our annotations are available.

MIT AGE Lab - a small sample of the 1,000+ hours of multi-sensor driving datasets collected at AgeLab.

Data – Check your dataset

Is there a bias in your dataset? If yes, it will impact the training



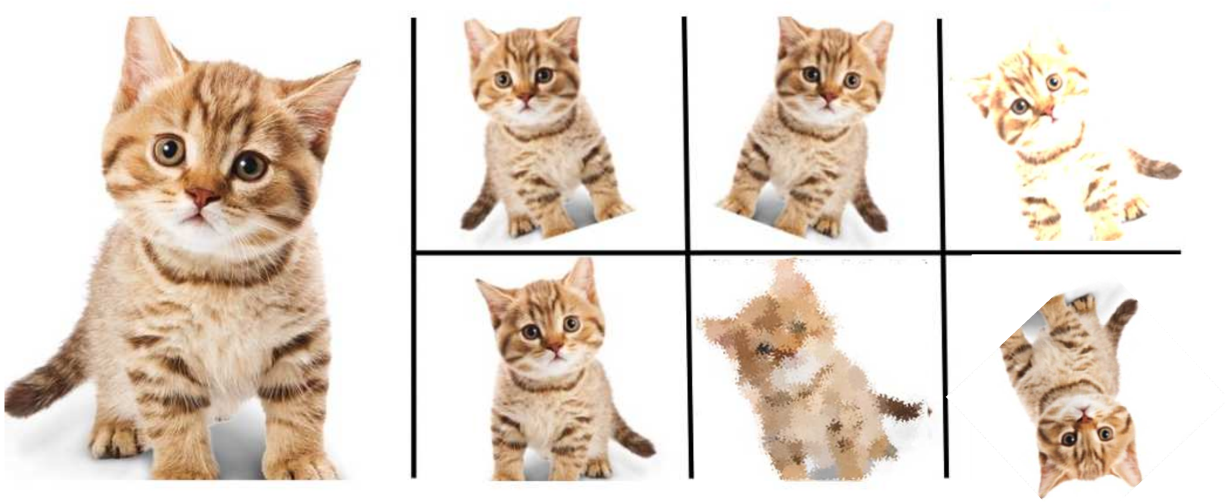
1000 dogs

1 cat

Source: https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced

Data – More data with data augmentation

How to use Deep Learning when you have limited data?



Enlarge your available Dataset with

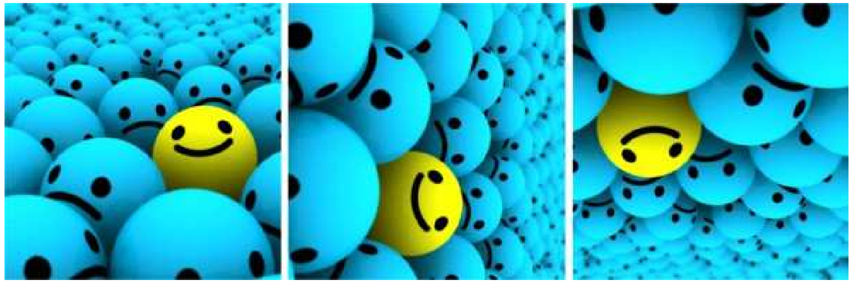
Data Augmentation

Source: https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced

Data – More data with data augmentation

1. Flip the images

2. **Rotate** the images



3. **Scale** the images outward or inward

Source: https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced

Data – More data with data augmentation

4. **Crop**

5. **Translation** of objects in x,y-position

6. **Gaussian Noise**

Source: https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced

Data – More data with data augmentation

Original photo   Reference photo   Result

7. **Deep Photo Style Transfer:** Transform one image from one domain to an image to another domain using Deep Learing

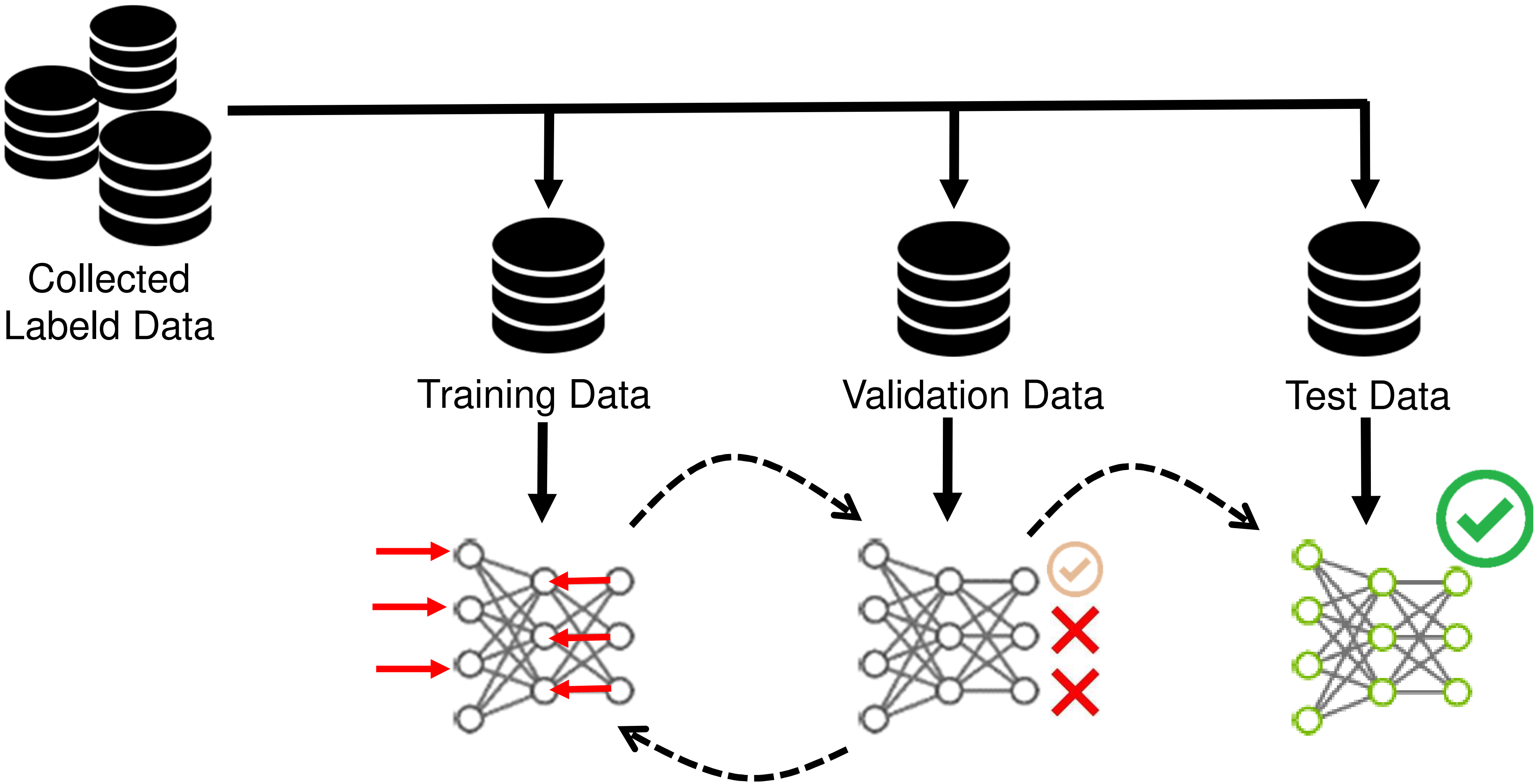
Source: https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced

Data – Improvement with data augmentation

	Top-1 Accuracy	Top-5 Accuracy
Baseline	$48.13 \pm 0.42\%$	$64.50 \pm 0.65\%$
Flipping	$49.73 \pm 1.13\%$	$67.36 \pm 1.38\%$
Rotating	$50.80 \pm 0.63\%$	$69.41 \pm 0.48\%$
Cropping	$\mathbf{61.95 \pm 1.01\%}$	$\mathbf{79.10 \pm 0.80\%}$
Color Jittering	$\mathbf{49.57 \pm 0.53\%}$	$67.18 \pm 0.42\%$
Edge Enhancement	$49.29 \pm 1.16\%$	$66.49 \pm 0.84\%$
Fancy PCA	$49.41 \pm 0.84\%$	$\mathbf{67.54 \pm 1.01\%}$

Source: https://arxiv.org/pdf/1708.06020.pdf

Data – Provide the data in your dataset



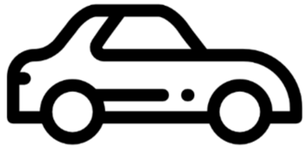
Split your data into Training (60%), Validation (20%) and test (20%) dataset

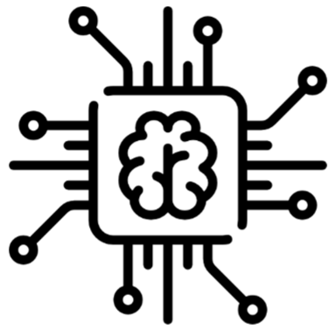
AI-Development
Johannes Betz / Prof. Dr. Markus Lienkamp / Prof. Dr. Boris Lohmann

(Johannes Betz, M. Sc.)

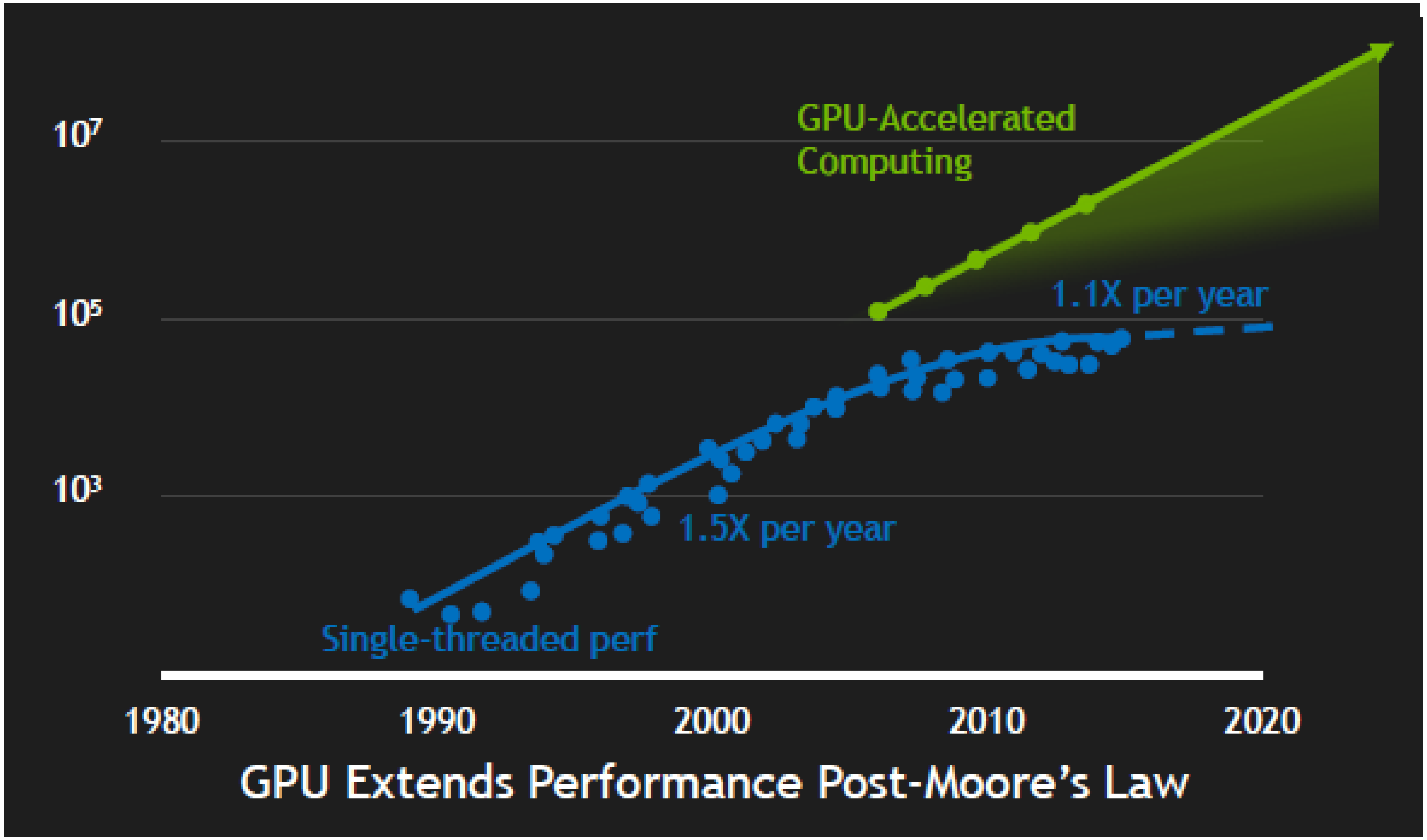
Agenda

1. Chapter: AI-Development Pipeline
2. Chapter: Transfer Learning
3. Chapter: AI-Frameworks
4. Chapter: Data and Labeling
5. **Chapter: GPU Computing**
6. Chapter: Hyperparamter Tuning
7. Chapter: AI-Inference
8. Chapter: Summary





GPU Computing – What is a GPU?



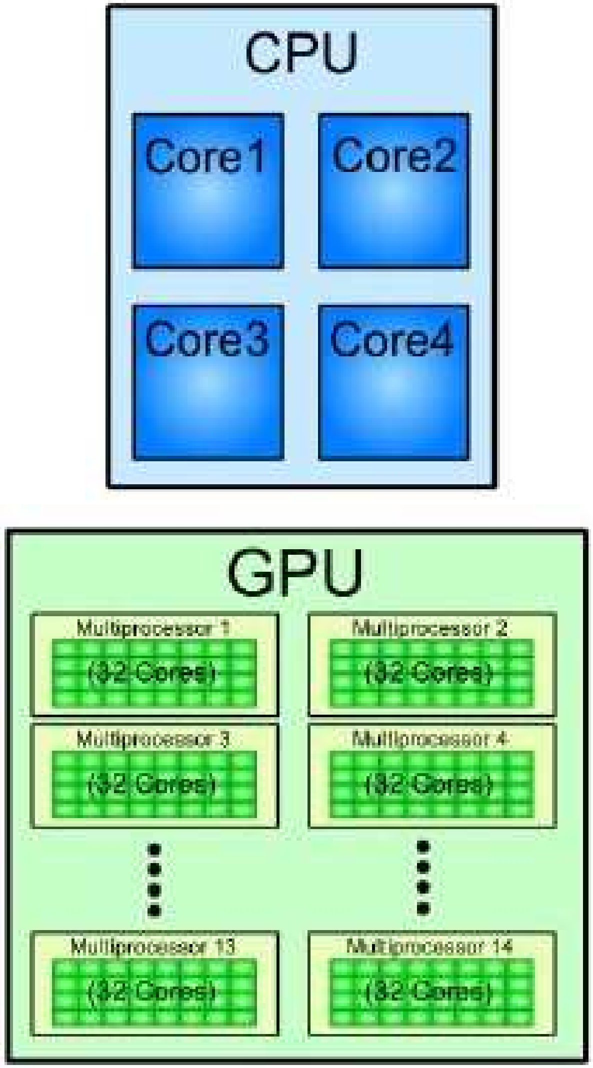
https://www.fedscoop.com/era-ai-computing/

GPU Computing – What is a GPU?

- A GPU is an **Graphical Processing Unit**
- Specialized electronic circuit design
- Rapidly manipulate and alter memory to **accelerate the creation of images in a frame buffer**
- By 2012, GPUs had evolved into highly **parallel multi-core systems** allowing very efficient manipulation of large blocks of data.
- This design is **more effective** than general-purpose central processing unit (CPUs) for algorithms in situations where processing large blocks of data is done in parallel, such as:
  - Push-relabel maximum flow algorithm
  - Fast sort algorithms of large lists
  - Two-dimensional fast wavelet transform
  - Molecular dynamics simulations
  - Artificial Neural Networks

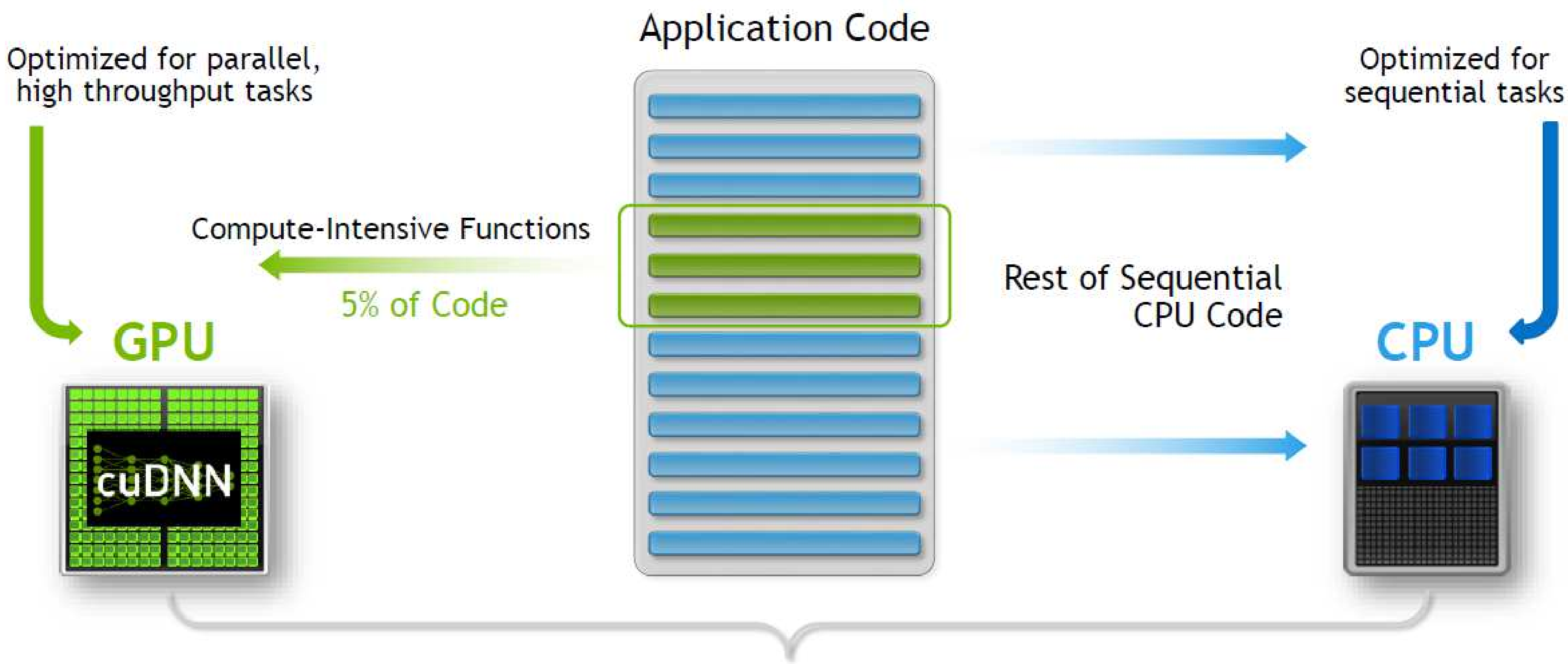
Source: https://www.vrnerds.de/nvidia-pascal-ab-27-mai/
https://www.notebookcheck.net/Like-AMD-Nvidia-promises-to-ramp-up-GPU-production.282024.0.html

GPU Computing – What is a GPU?



Source: Udacity Self-Driving Car Nanodegree – Part 1, Lesson 15

Additional Slide



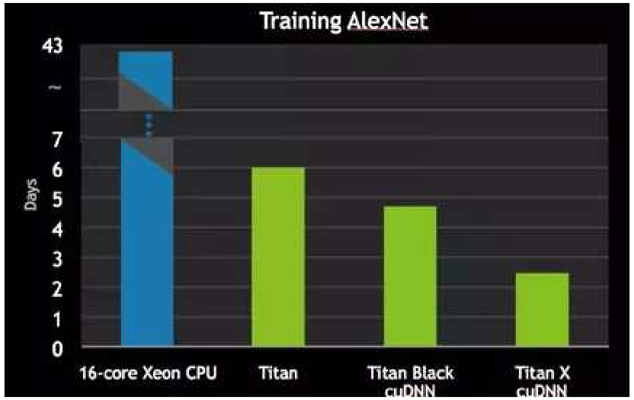
GTX 1080 Ti

GPU Engine Specs
CUDA Cores
3584
Graphics Clock (MHz)
1480
Processor Clock (MHz)
1582

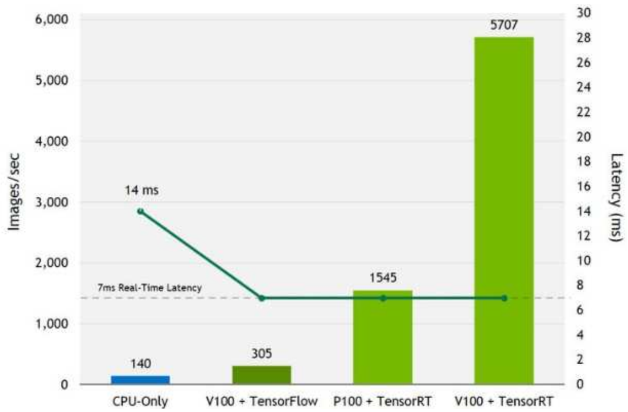
Memory Specs
Standard Memory Config
11 GB GDDR5X
Memory Interface Width
352-bit
Memory Bandwidth (GB/sec)
11 Gbps

Source: Nvidia Deep Learning Workshop @LRZ

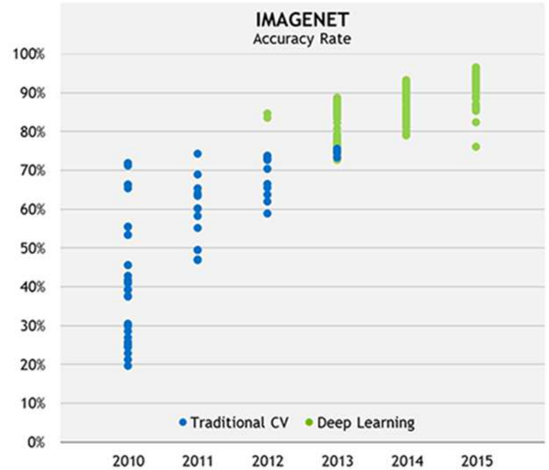
GPU Computing – Why a GPU for DL?



1. Faster Training



2. Faster Inference



3. Better Results

Source: https://www.quora.com/Would-the-success-of-deep-learning-have-been-possible-without-GPUs-and-CUDA/ https://www.tomshardware.co.uk/nvidia-ai-technology-improves-efficiency,news-56853.html

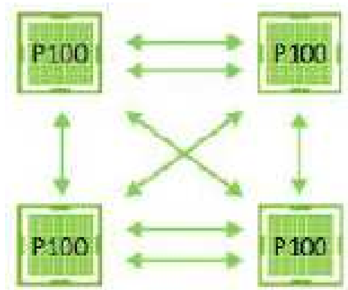
GPU Computing – Nvidia GPU for DL

- Using the NVIDIA hard- and software for your DL development
- NVIDIA is one of the leading GPU manufacteres
- NVIDIA is one of the leading Deep Learning developer and is building a complete Environment

Consumer GPUs
~800 €

HPC GPUs
~16.000 €

GPU Cluster (8x)
~120.000 €



NVLINK
>5x-12x PCI

CUDA API
Parallel Computing

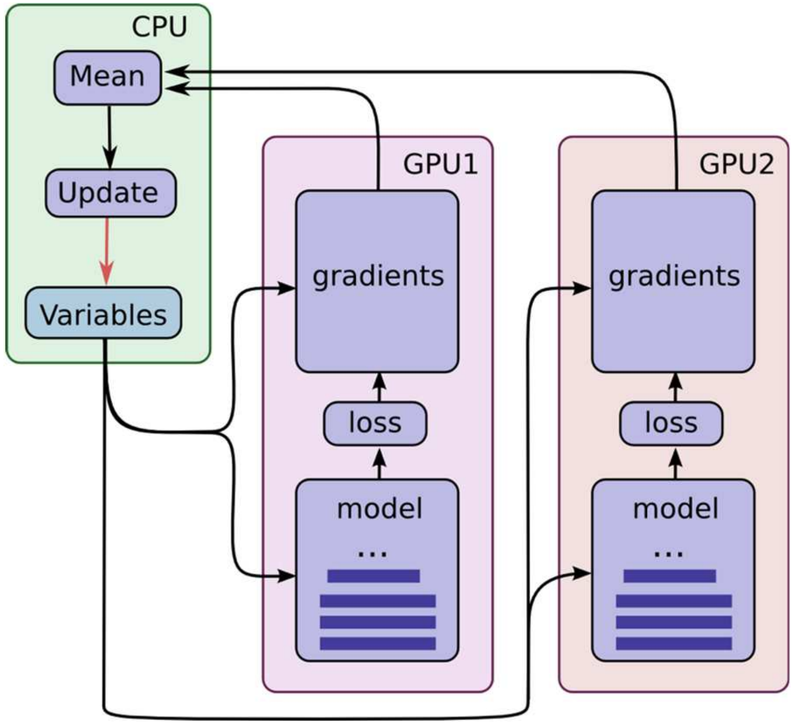
cuDNN
DL Library

Digits
DL Training

TensorRT
Faster Inference

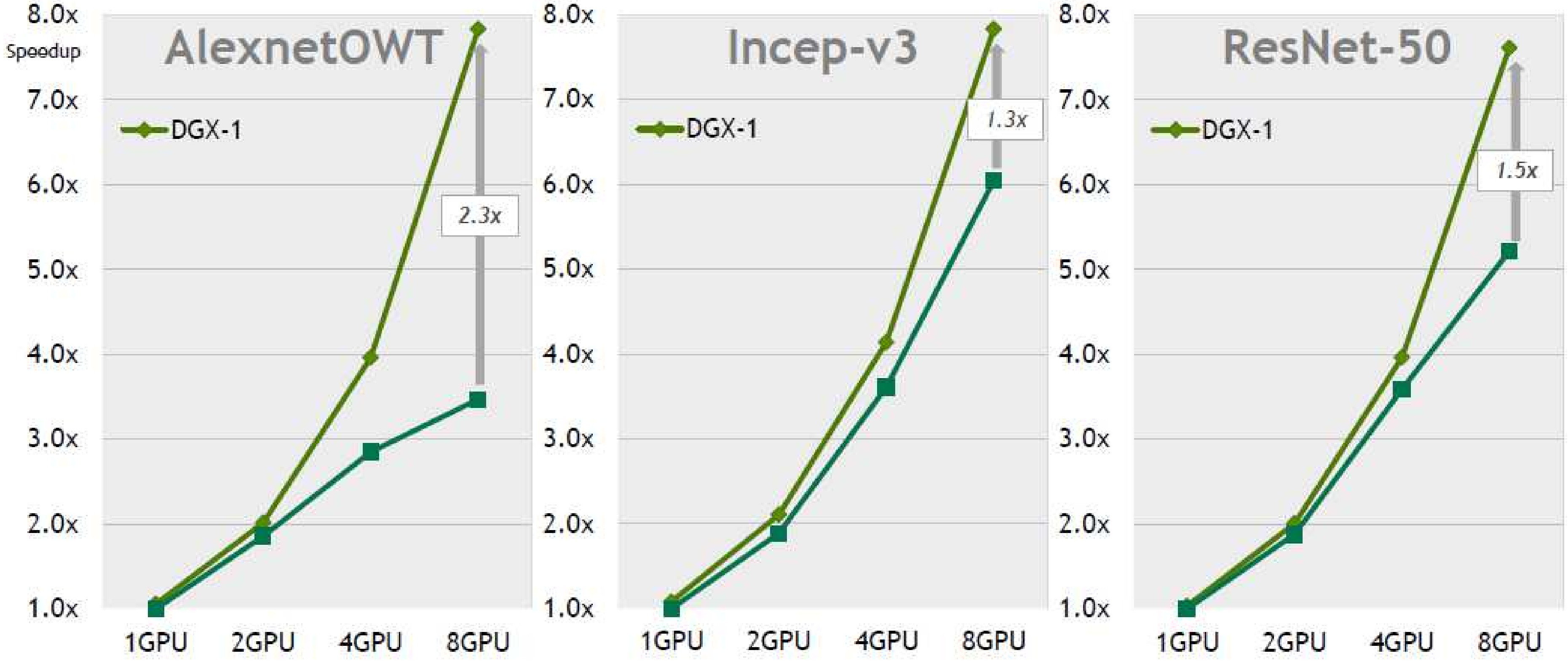
GPU Computing – Mulitple GPU

- What is better than training faster on One GPU? → Training on **Multiple GPUs!**
- You can either use a normal PC with a motherboard that fits multiple GPUs (2x, 3x, 4x) or use a special GPU Cluster (NVIDIA DGX-1)
- Multi-GPU can be used in different ways:
  - Use a single GPU for the training of specific Hyperparamter set and do the same on the others
  - Use multiple GPUs for simultanous calculation → Takes time to adjust the model



Source: https://blog.rescale.com/deep-learning-with-multiple-gpus-on-rescale-tensorflow/

GPU Computing – Mulitple GPU Training

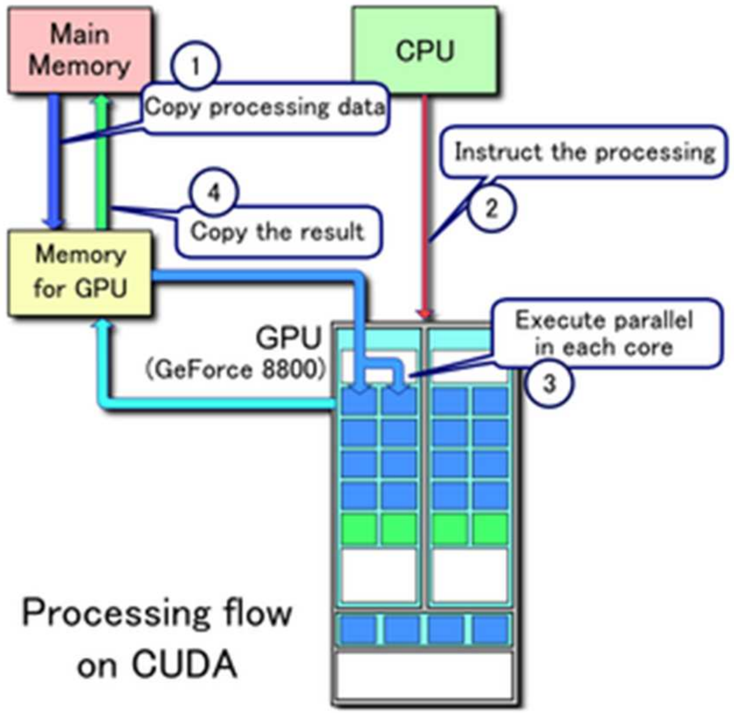


Multiple GPU: DGX vs. Normal Motherboard

Source: https://blog.slavv.com/picking-a-gpu-for-deep-learning-3d4795c273b9

GPU Computing – Cuda

- **CUDA** is a parallel computing platform and application programming interface (API) model created by Nvidia.
- It allows to use a CUDA-enabled GPU for general purpose processing
- The CUDA platform is a software layer that gives direct access to the GPU's virtual instruction set and parallel computational elements, for the execution of compute kernels
- Programming languages in C, C++
- Full support for integer and bitwise operations, including integer texture lookups
- Unified Memory



GPU Computing – Cuda

Standard C Code

```c
void saxpy_serial(int n,
                  float a,
                  float *x,
                  float *y)
{
  for (int i = 0; i < n; ++i)
    y[i] = a*x[i] + y[i];
}

// Perform SAXPY on 1M elements
saxpy_serial(4096*256, 2.0, x, y);
```

Parallel C Code

```c
__global__
void saxpy_parallel(int n,
                    float a,
                    float *x,
                    float *y)
{
  int i = blockIdx.x*blockDim.x +
          threadIdx.x;
  if (i < n) y[i] = a*x[i] + y[i];
}

// Perform SAXPY on 1M elements
saxpy_parallel<<<4096,256>>>(n,2.0,x,y);
```

Source: https://www.youtube.com/watch?v=nRSxp5ZKwhQ/
https://intranet.birmingham.ac.uk/it/teams/infrastructure/research/bear/documents/public/CUDA-2013-07-31/GPU-overview.pdf

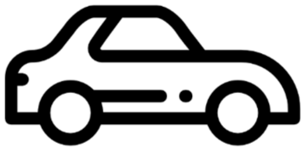
AI-Development

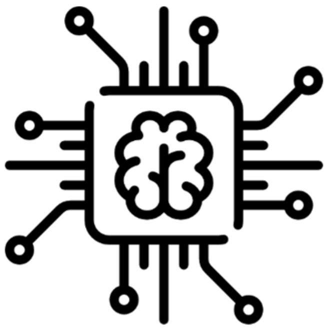
Johannes Betz / Prof. Dr. Markus Lienkamp /
Prof. Dr. Boris Lohmann

(Johannes Betz, M. Sc.)

Agenda

1. Chapter: AI-Development Pipeline
2. Chapter: Transfer Learning
3. Chapter: AI-Frameworks
4. Chapter: Data and Labeling
5. Chapter: GPU Computing
6. **Chapter: Hyperparamter Tuning**
7. Chapter: AI-Inference
8. Chapter: Summary





Hyperparamter Tuning – Whats that?

- When we train the ANN, we focus on **not overfitting** in the training loss and getting a **good evaluation** at the end
- To achieve that, we can vary different paramters for the training of the ANN → These are the **Hyperparameters**
- The tuning of the Hyperparameters is **the „magic“** behind a good ANN and changes the ANN from a Black Box to a system we can understand
- Hyperparamter Tuning is nothing official, more like a **„best practice“** or a **„tipps and tricks“** collection

Important: Hyperparameter tuning differs from problem to problem!!!

Hyperparamter Tuning – Whats that?

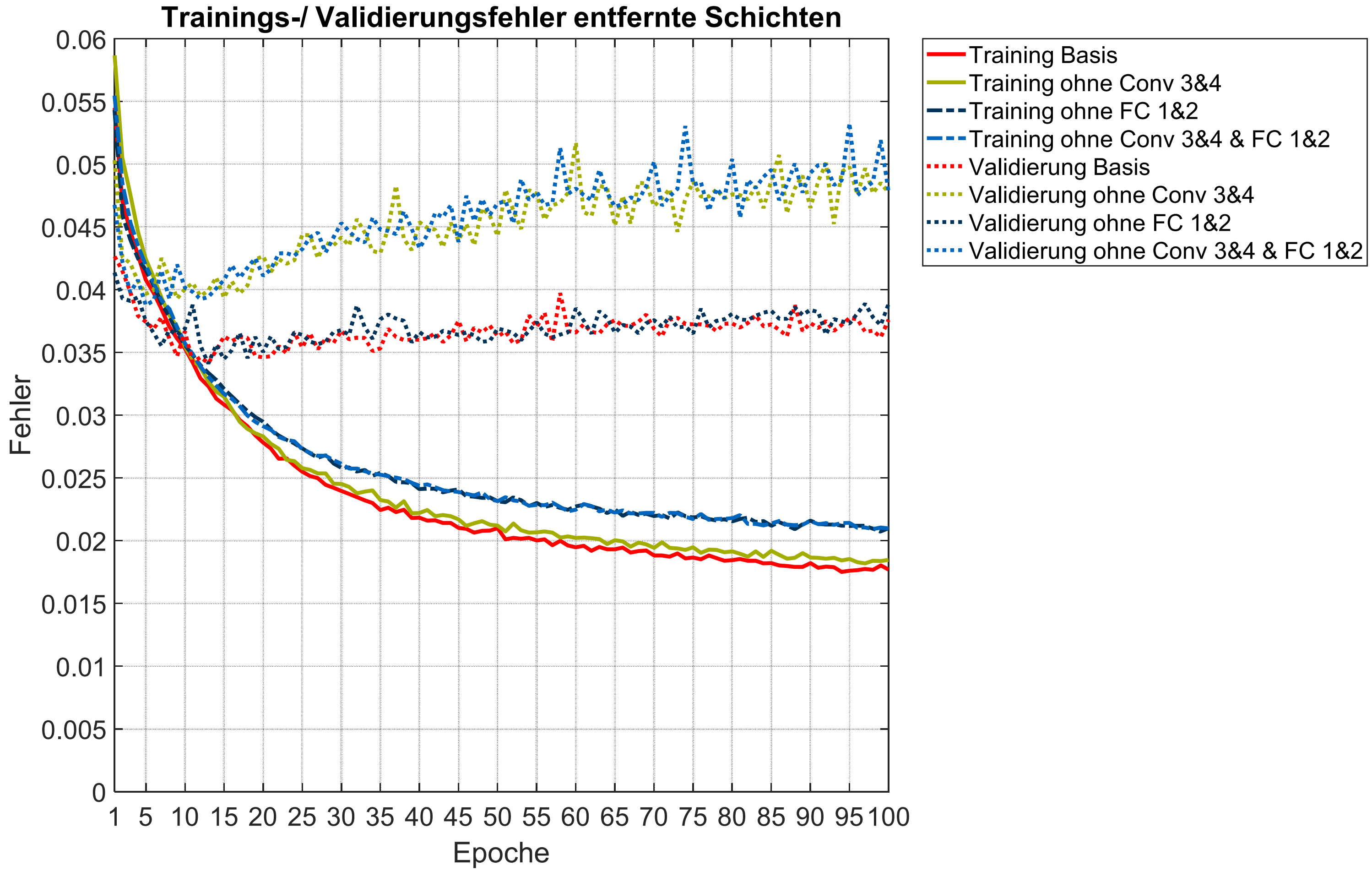
Things you can vary in your ANN:

- Number of hidden layers
- Number of fully connected layers
- Number of neurons in one layer
- Number of training epochs
- Weight Initialization
- Learning rate
- Batch size
- Activation function
- Dropout Rate
- ……

→ **Usage of optimization algorithms for hyperparamter tuning possible**

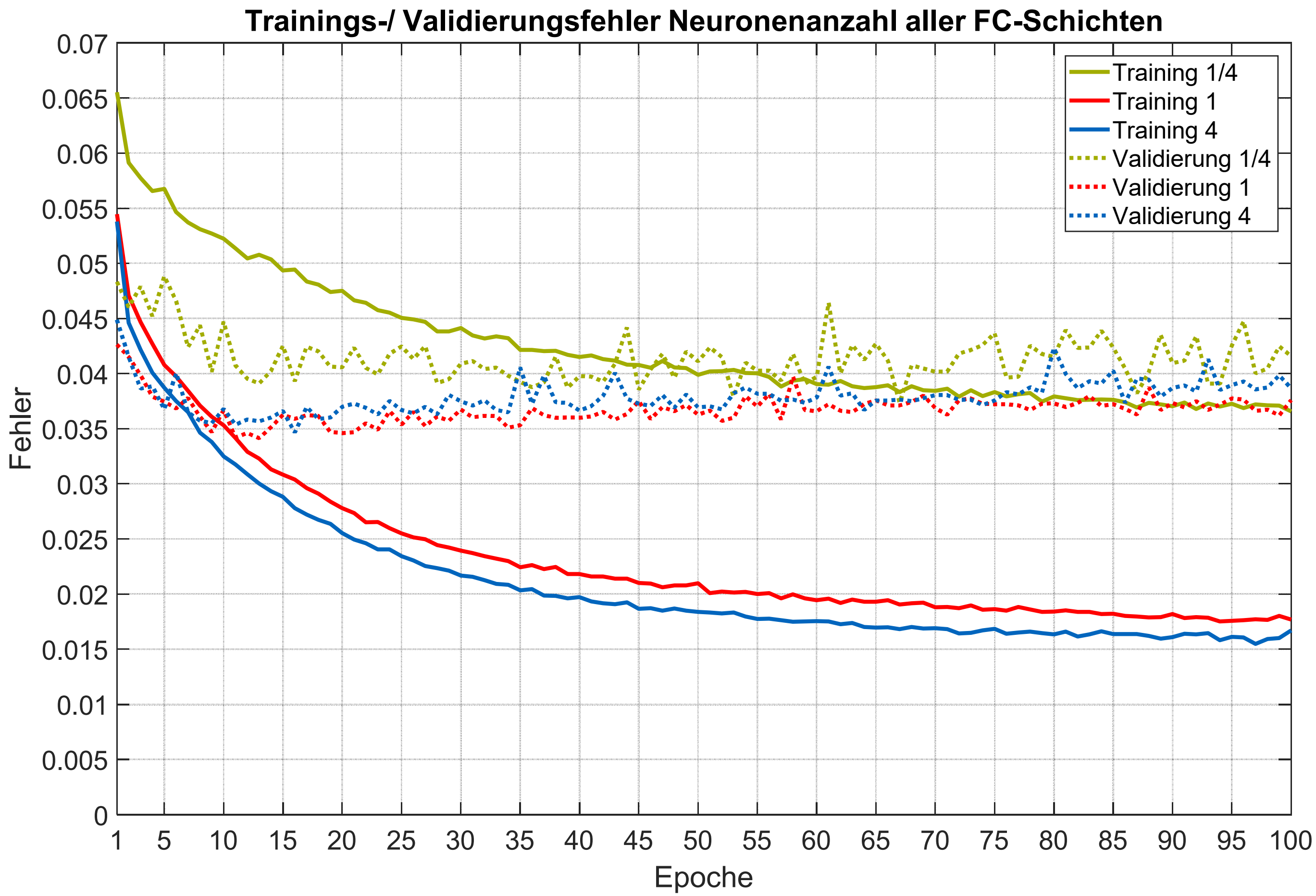
Source: http://colinraffel.com/wiki/neural_network_hyperparameters

Hyperparamter Tuning – Size of the net



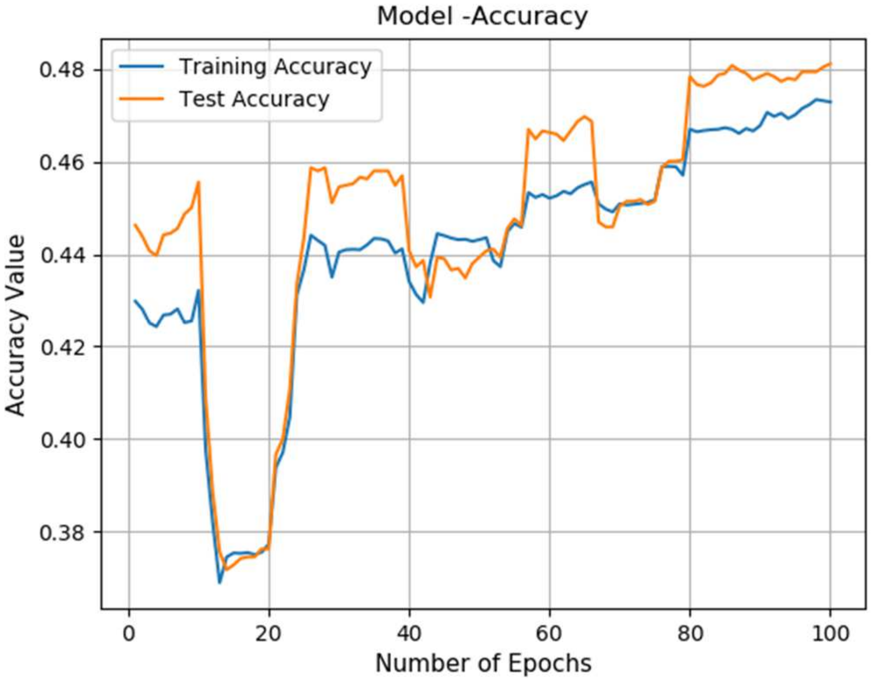
Variable Net-Size

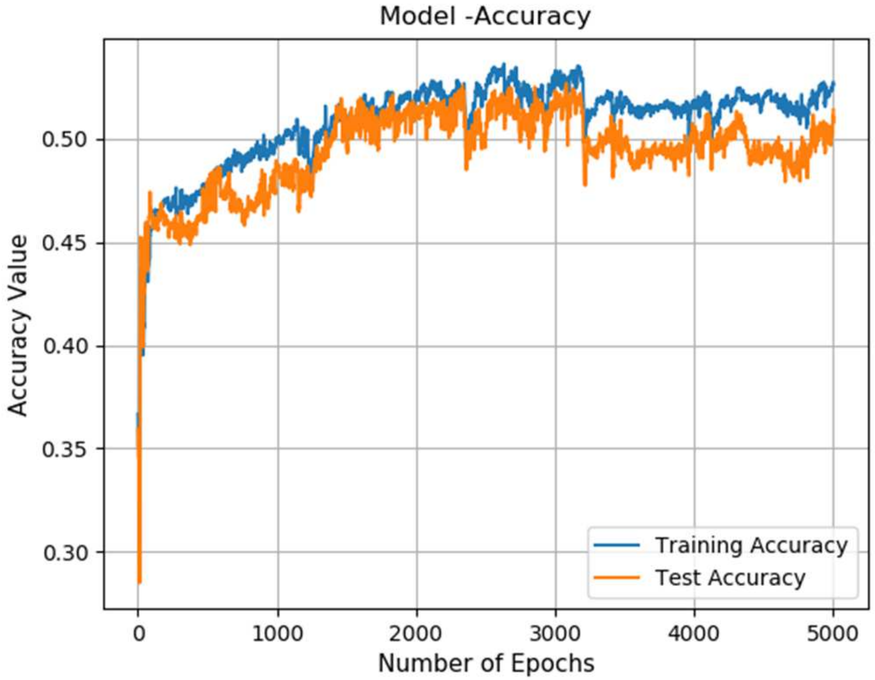
Hyperparamter Tuning – Size of the net



Variable Number of Neurons

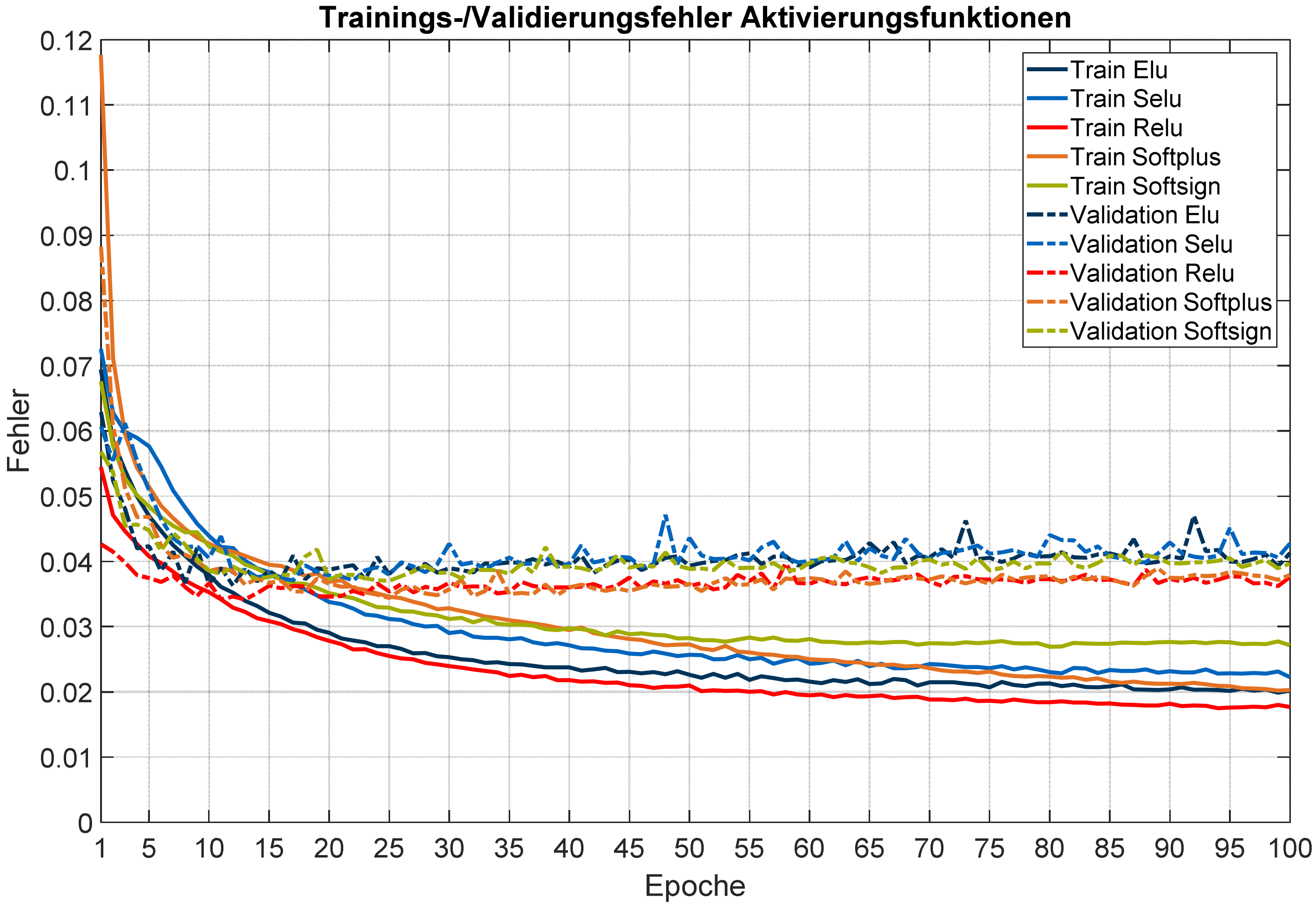
Hyperparamter Tuning – Number of Epochs



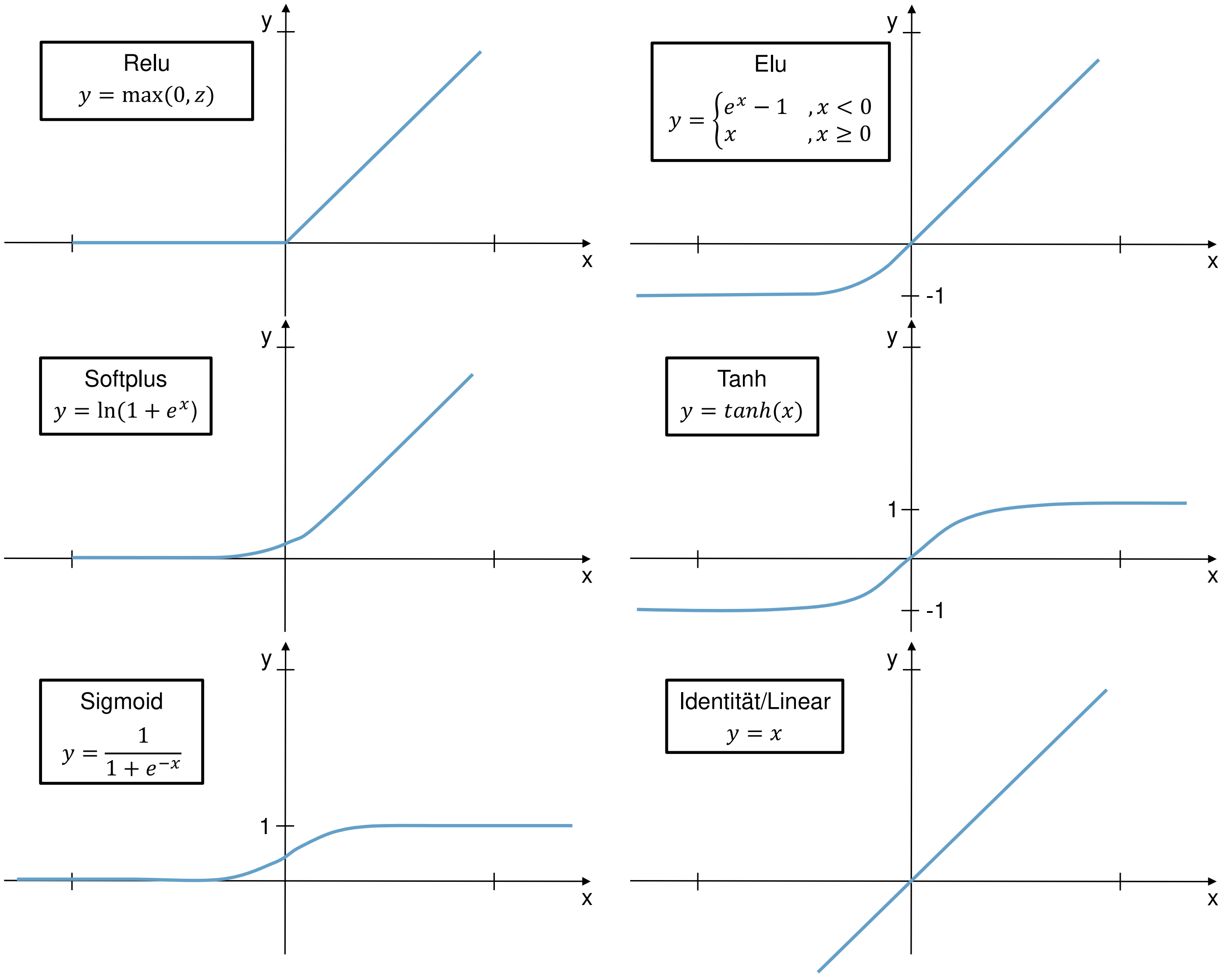


Same ANN structure, Same Input Data, Different Epochs

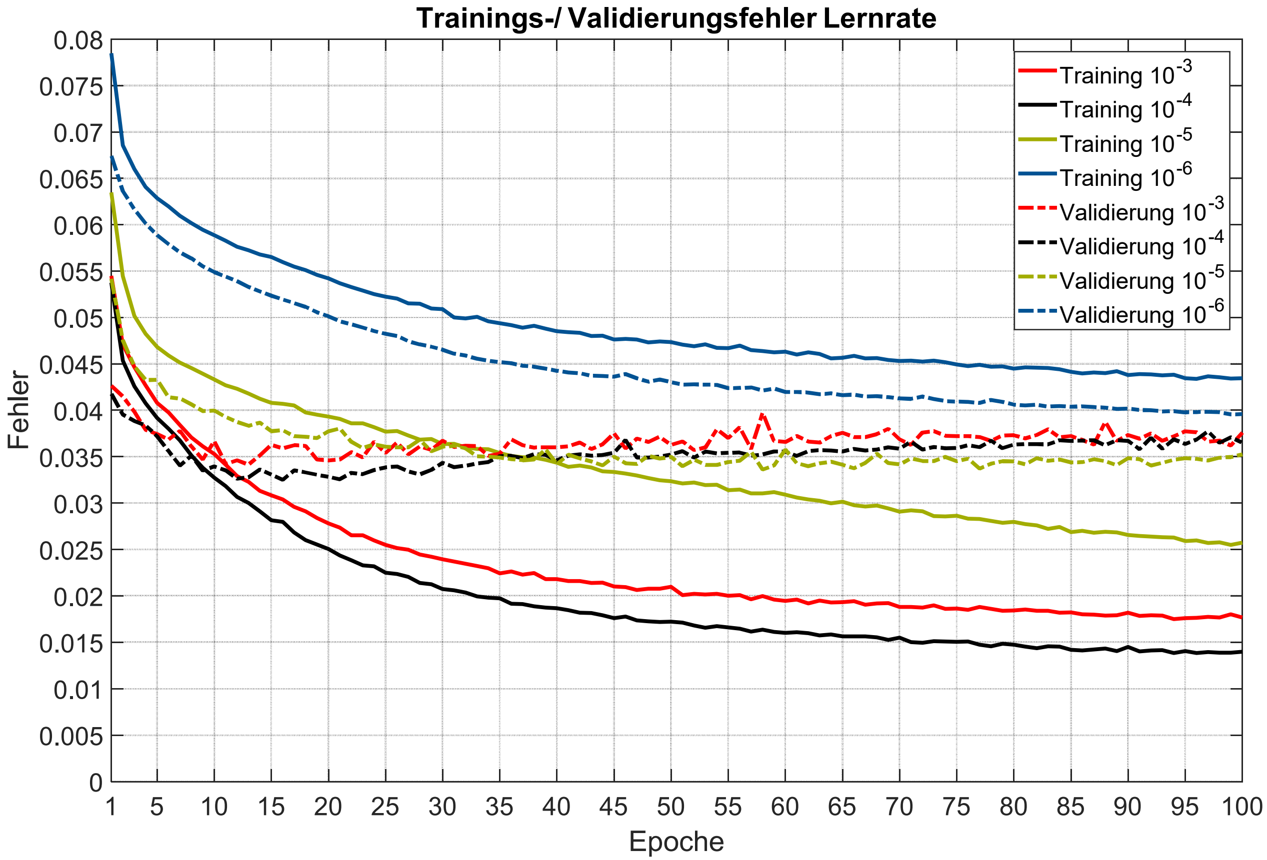
Hyperparamter Tuning – Activation Function



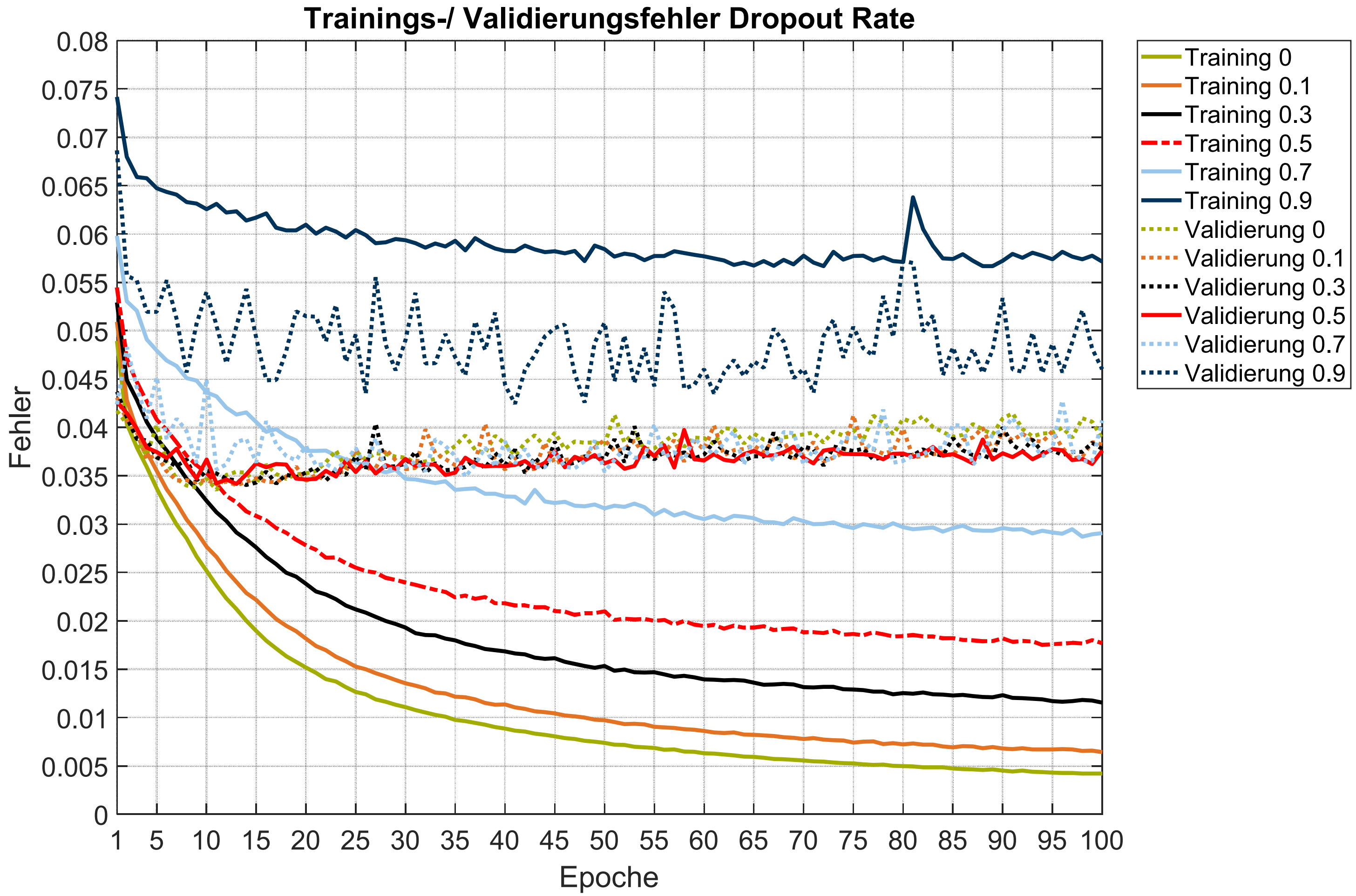
Additional Slide



Hyperparamter Tuning – Learningrate

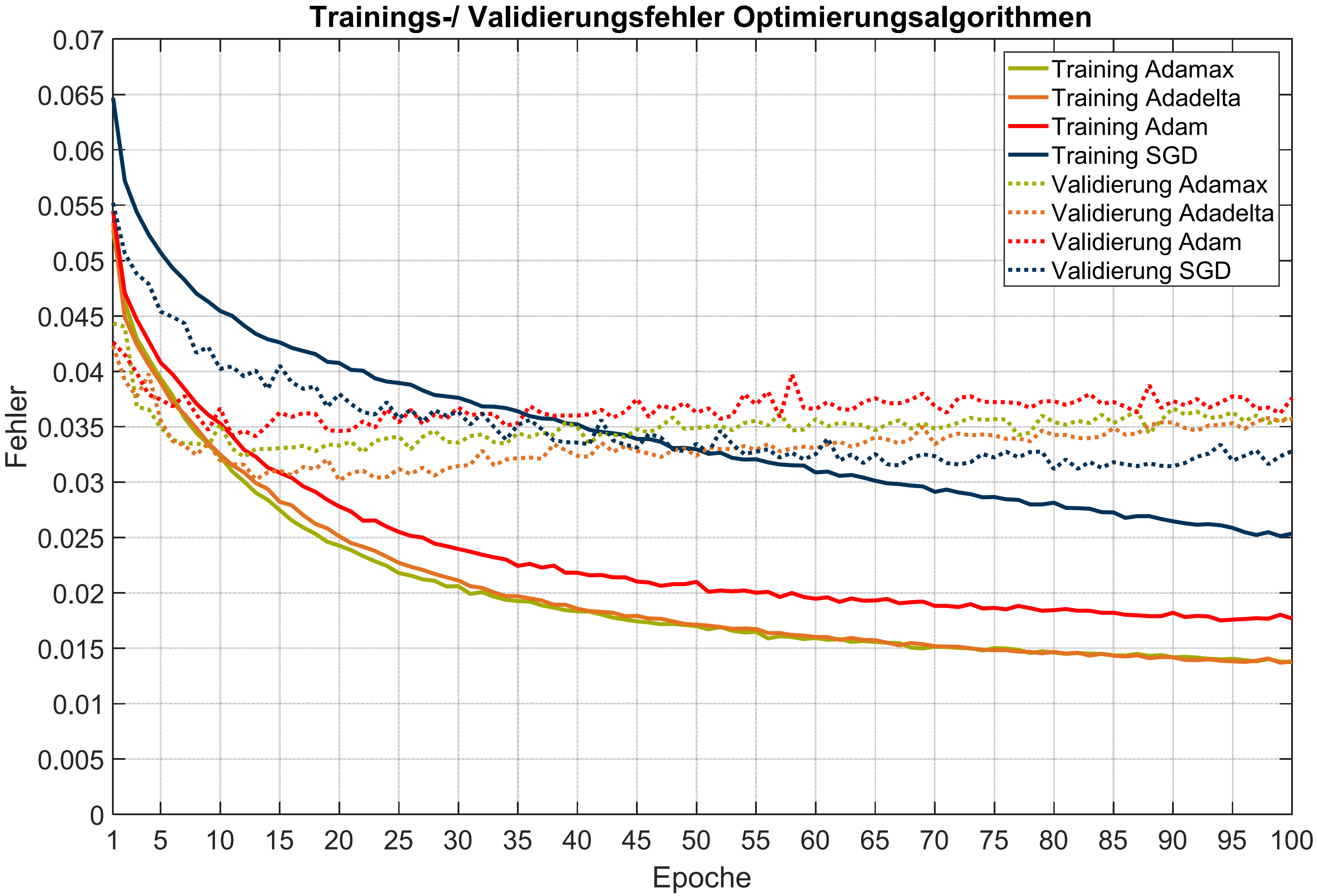


Hyperparamter Tuning – Dropout



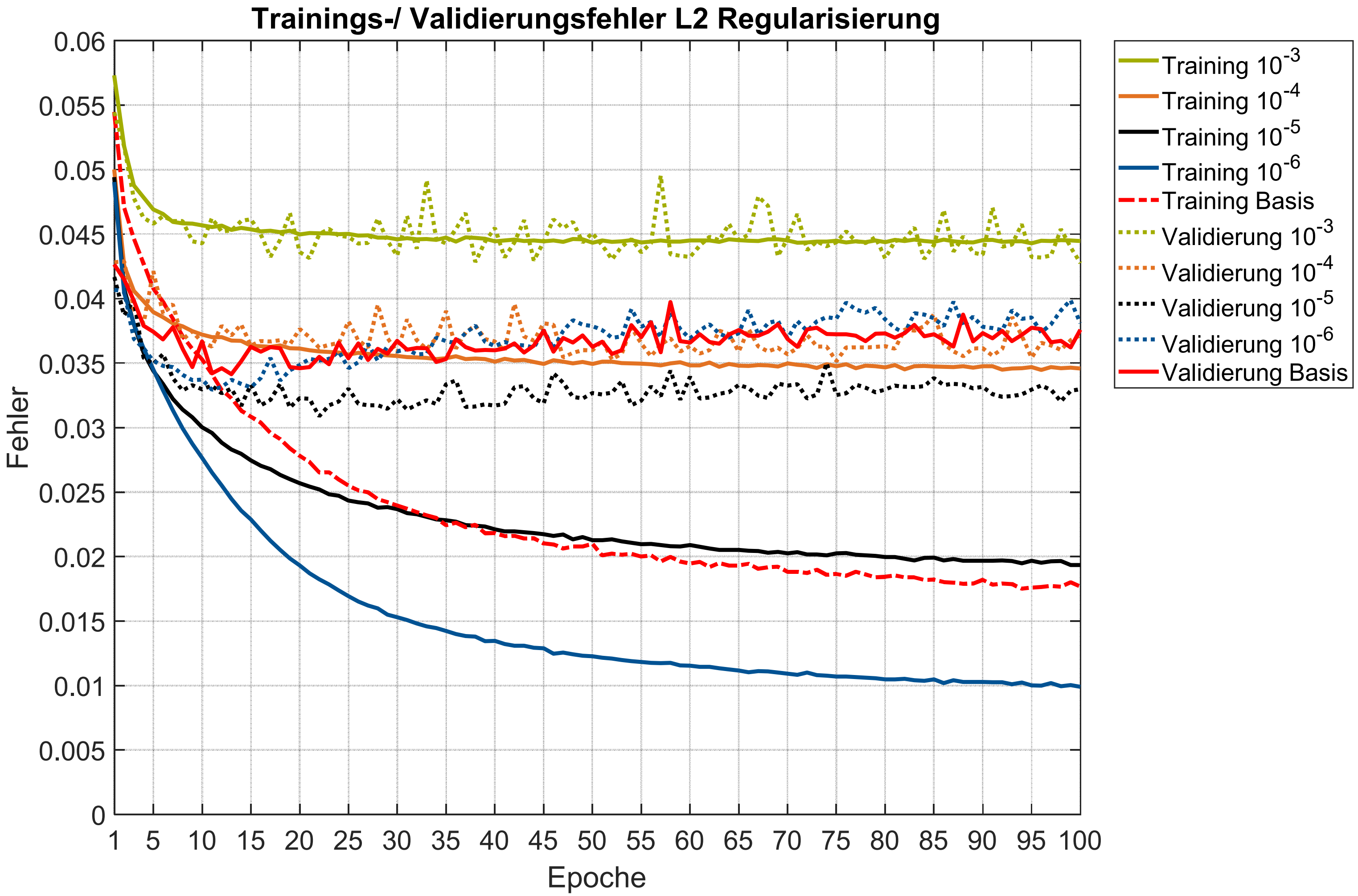
Additional Slide

Hyperparamter Tuning – Optimization Function



Additional Slide

Hyperparamter Tuning – Regularization

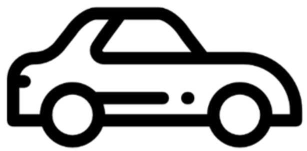


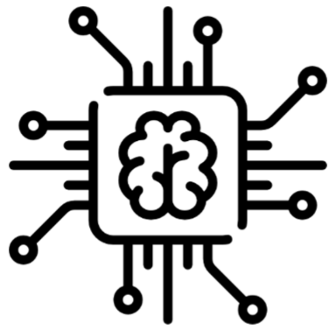
AI-Development
Johannes Betz / Prof. Dr. Markus Lienkamp / Prof. Dr. Boris Lohmann

(Johannes Betz, M. Sc.)

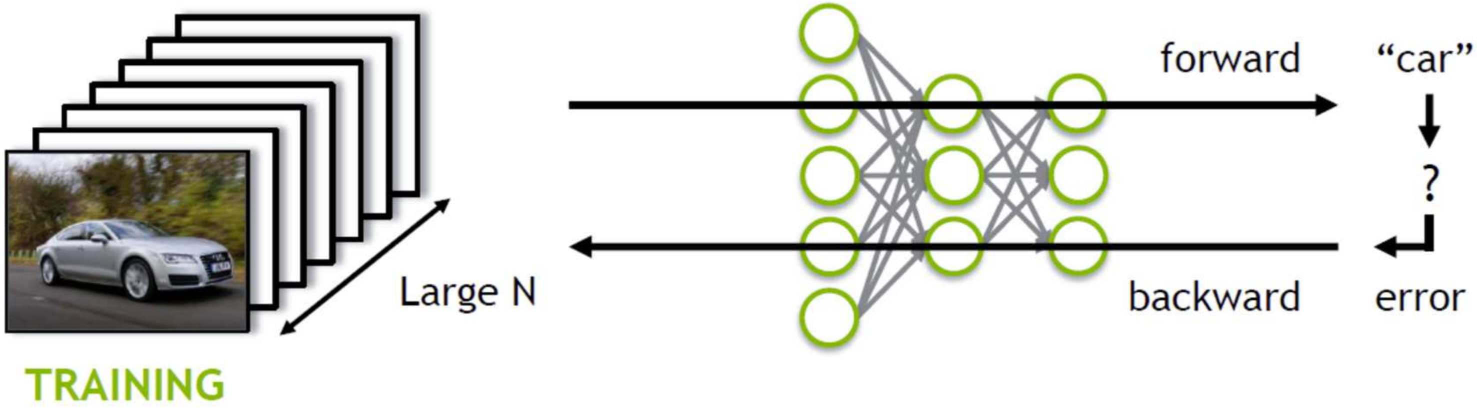
Agenda

1. Chapter: AI-Development Pipeline
2. Chapter: Transfer Learning
3. Chapter: AI-Frameworks
4. Chapter: Data and Labeling
5. Chapter: GPU Computing
6. Chapter: Hyperparamter Tuning
7. **Chapter: AI-Inference**
8. Chapter: Summary





AI-Inference – What is Inference?



Inference: The actual application and usage of your ANN

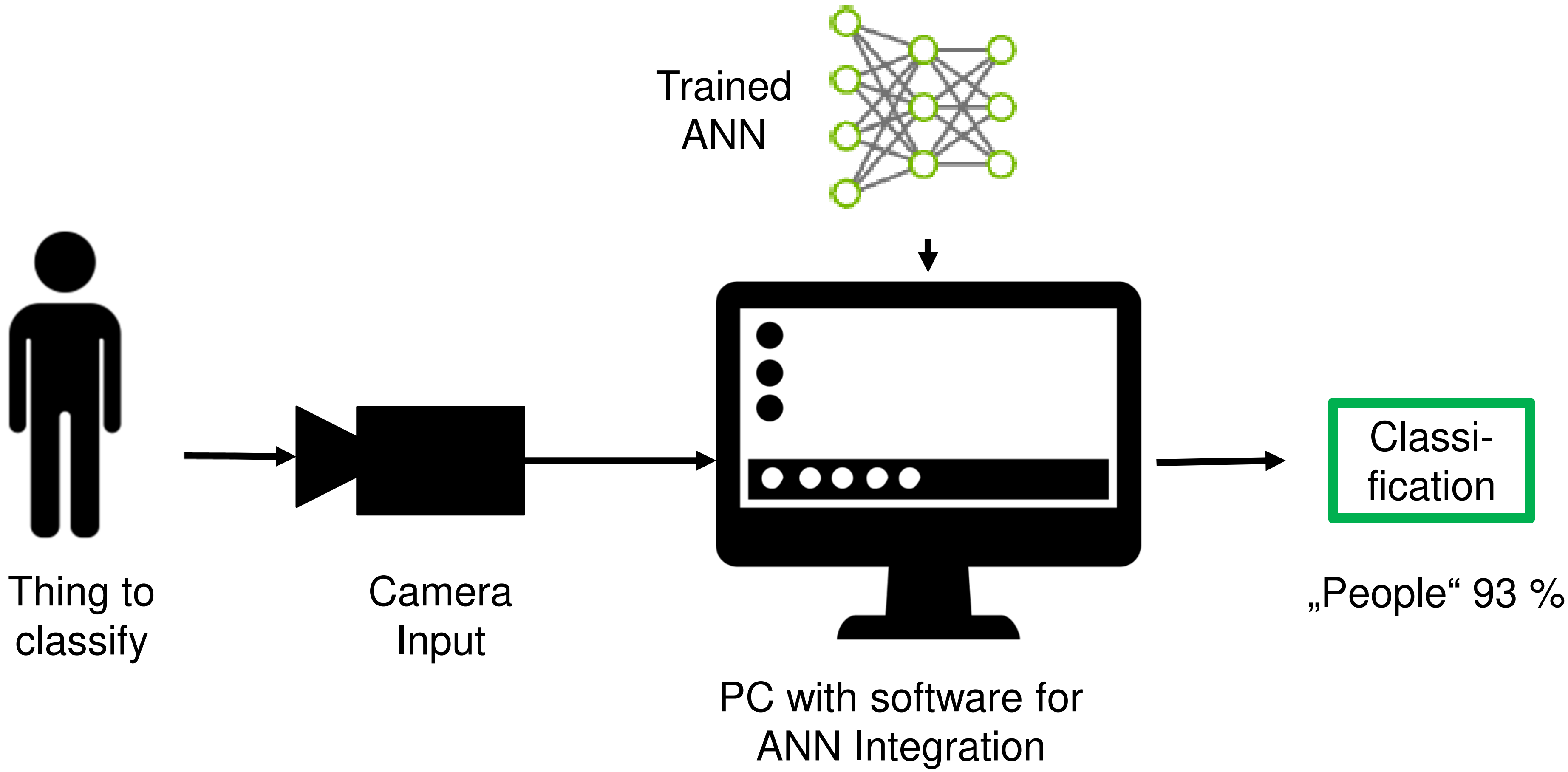
Source: Nvidia Deep Learning Workshop @LRZ

Additional Slide

Both DNN training and Inference start out with the same *forward propagation* calculation, but training goes further. As Figure 1 illustrates, after forward propagation, the results from the forward propagation are compared against the (known) correct answer to compute an error value.
A *backward propagation* phase propagates the error back through the network's layers and updates their weights using gradient descent in order to improve the network's performance at the task it is trying to learn. It is common to batch hundreds of training inputs (for example, images in an image classification network or spectrograms for speech recognition) and operate on them simultaneously during DNN training in order to prevent overfitting and, more importantly, amortize loading weights from GPU memory across many inputs, increasing computational efficiency.

For inference, the performance goals are different. To minimize the network's end-to-end response time, inference typically batches a smaller number of inputs than training, as services relying on inference to work (for example, a cloud-based image-processing pipeline) are required to be as responsive as possible so users do not have to wait several seconds while the system is accumulating images for a large batch. In general, we might say that the per-image workload for training is higher than for inference, and while high *throughput* is the only thing that counts during training, *latency* becomes important for inference as well.

AI-Inference – What is Inference?



Example: Live Image Classification

Source: https://github.com/dusty-nv/jetson-inference
https://www.google.de/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwj7oZOz7LncAhUGmrQKHThBDIMQjRx6BAgBEAU&url=https%3A%2F%2Fwww.onlinewebfonts.com%2Ficon%2F395988&psig=AOvVaw35HUnkc45PR1hxkOdoZQ42&ust=1532594180959688

AI-Inference – Inference Hardware: Nvidia Drive PX2

- Linux Ubuntu based „Mini-Computer“ (SoC)
- Automotive Grade!

Tegra X2 Architecture („Parker“)

- 6x ARM Cortex-A57
- 1x 256-core Pascal GPU
- 16GB LPDDR4, 128-bit interface
- Peripherie: USB, HDMI, SATA, UART, SPI, I2C, GPIO,CAN, LIN,…

+Software Package „Driveworks SDK“: Cuda, CudNN, TensorRT + Autonomous Driving Functions API

Additional Slide

Software SDK for inference of AI software for autonomous driving

	DETECTION	LOCALIZATION	PLANNING	VISUALIZATION
DRIVEWORKS SDK	Detection/Classification	Map Localization	Vehicle Control	Streaming to cluster
	Sensor Fusion	HD-Map Interfacing	Scene understanding	ADAS rendering
	Segmentation	Egomotion (SFM, Visual Odometry)	Path Planning solvers	Debug Rendering
System SW	V4L/V4Q, CUDA , cuDNN, NPP, OpenGL, ...			
Hardware	Tegra , dGPU			
Sensors	Camera, LIDAR, Radar, GPS, Ultrasound, Odometry, Maps			

Nvidia Driveworks SDK

AI-Inference – Inference Hardware: Nvidia Drive AGX

Nvidia Drive AGX Hardware + Driveworks Software SDK

Source: https://www.youtube.com/watch?v=KS_4xjXNTxg&t=20s

AI-Inference – Project Roborace

Hardware

Nvidia Drive PX2

Speedgoat

Software

Perception → Strategy → Control

Software Language

C++
ROS

python

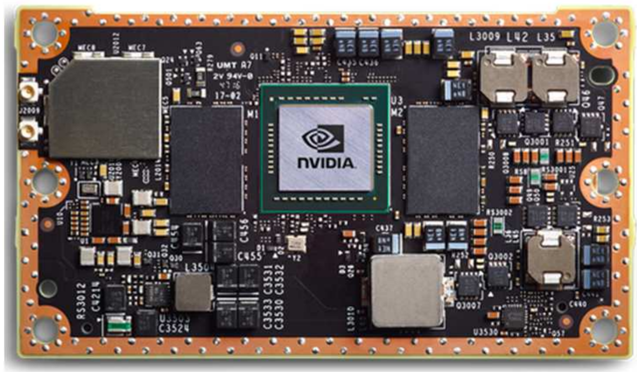
MATLAB SIMULINK

Interface

ØMQ

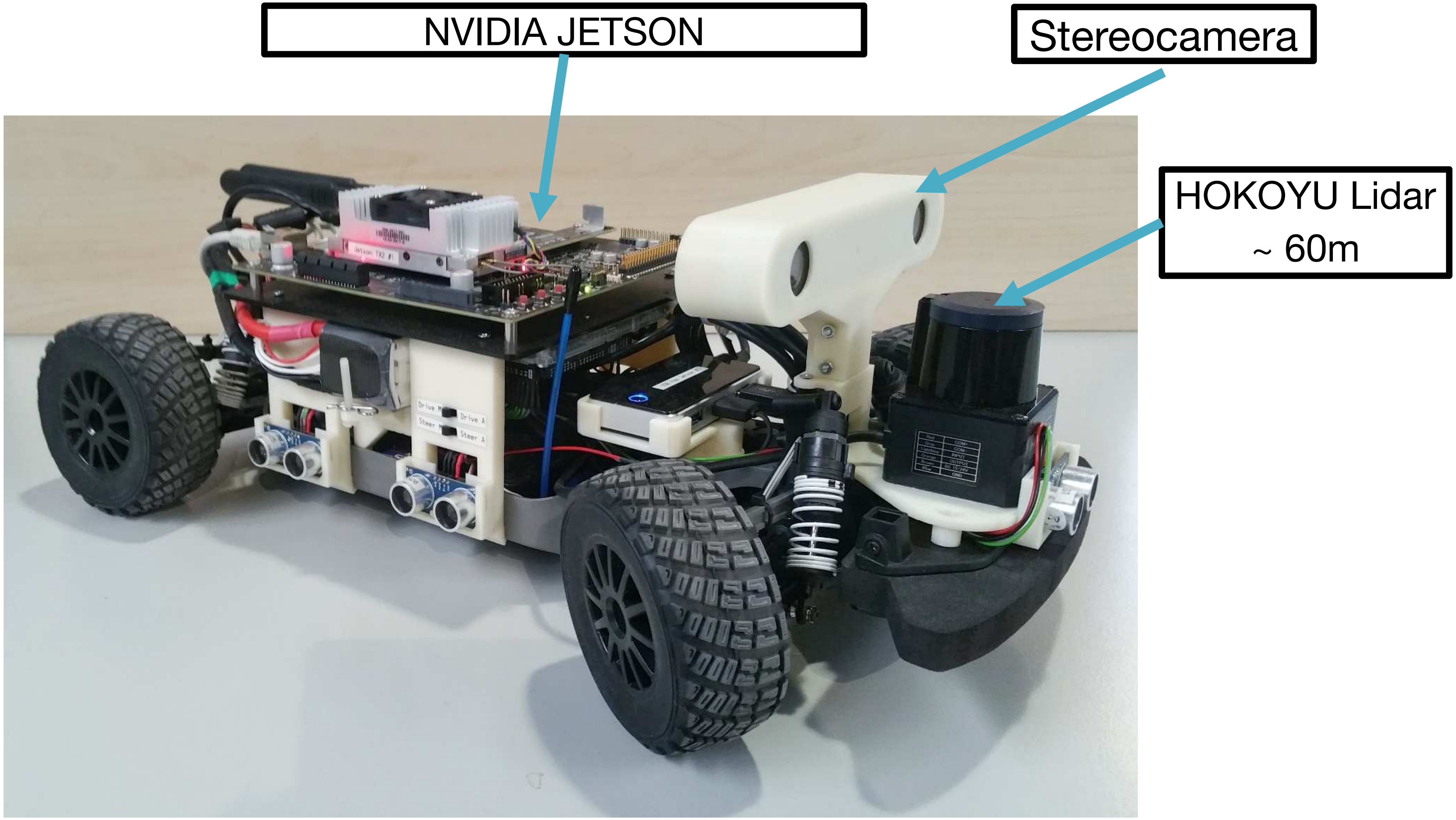
UDP
Ethernet

AI-Inference – Inference Hardware: Nvidia Jetson



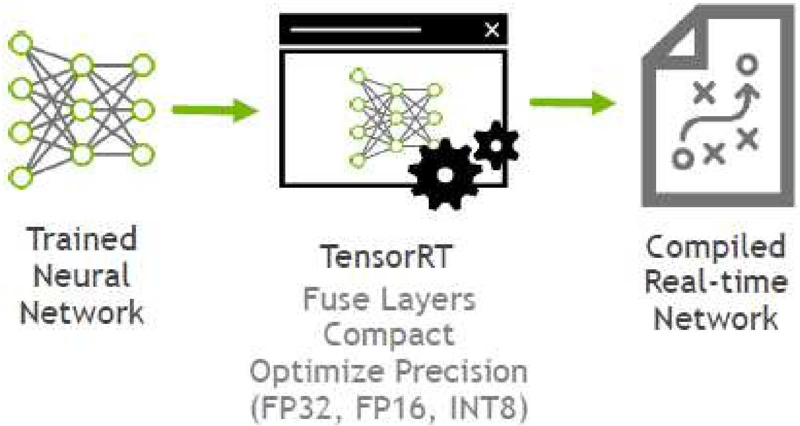
- Linux Ubuntu based „Mini-Computer“
- Size 50x87mm
- Quad-core ARM Cortex-A57
- 256-core Pascal GPU
- 8GB LPDDR4, 128-bit interface
- Peripherie: USB, HDMI, SATA, UART, SPI, I2C, GPIO,….
- Software Package „Jetpack“; Cuda, CudNN, TensorRT,…

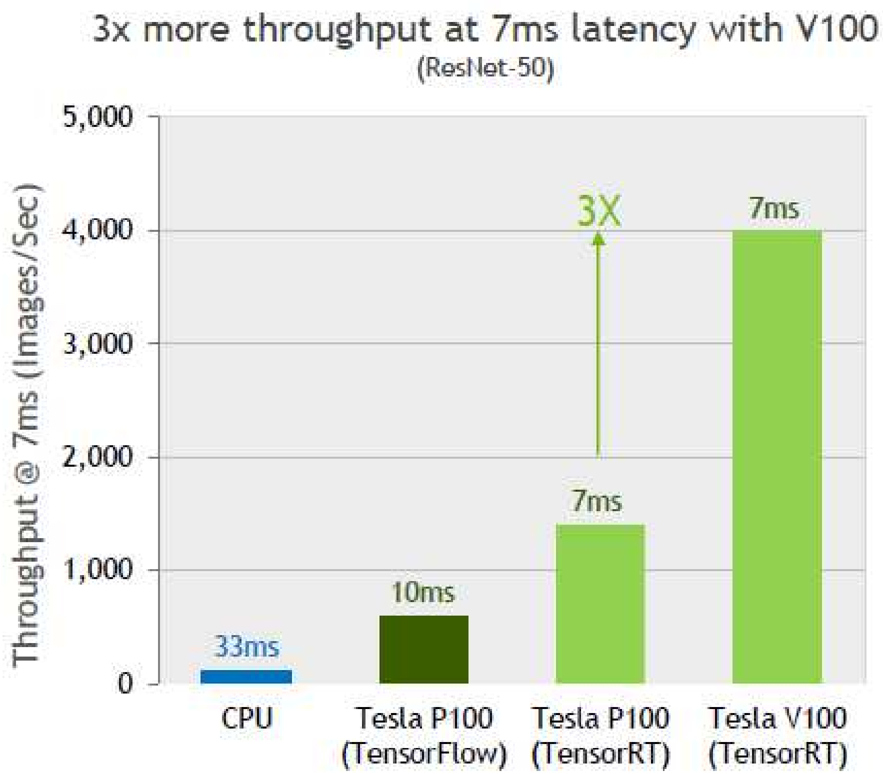
AI-Inference – Inference Hardware: Nvidia Jetson



FTM Autonomous RC-Car

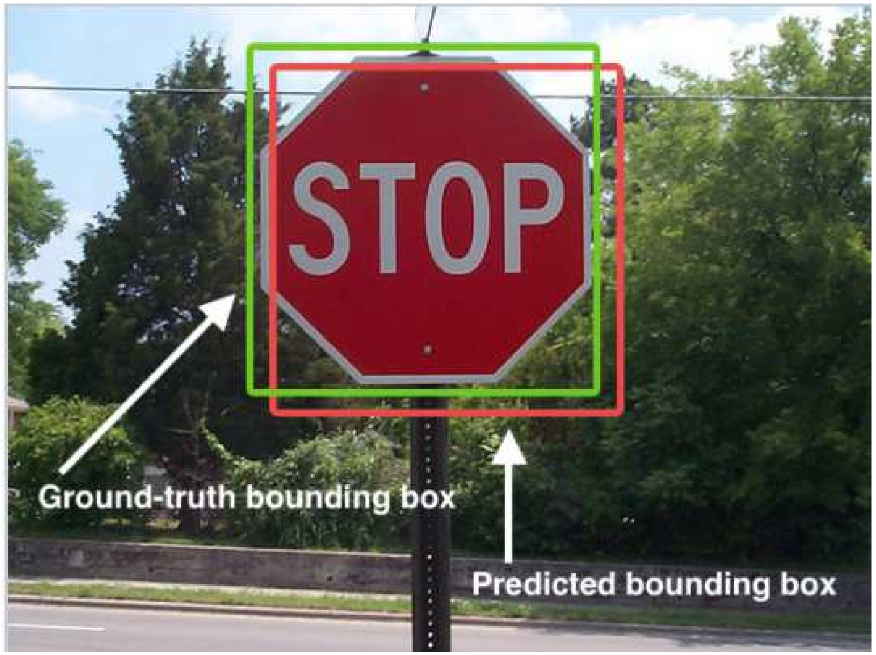
AI-Inference – Fasten the Inference

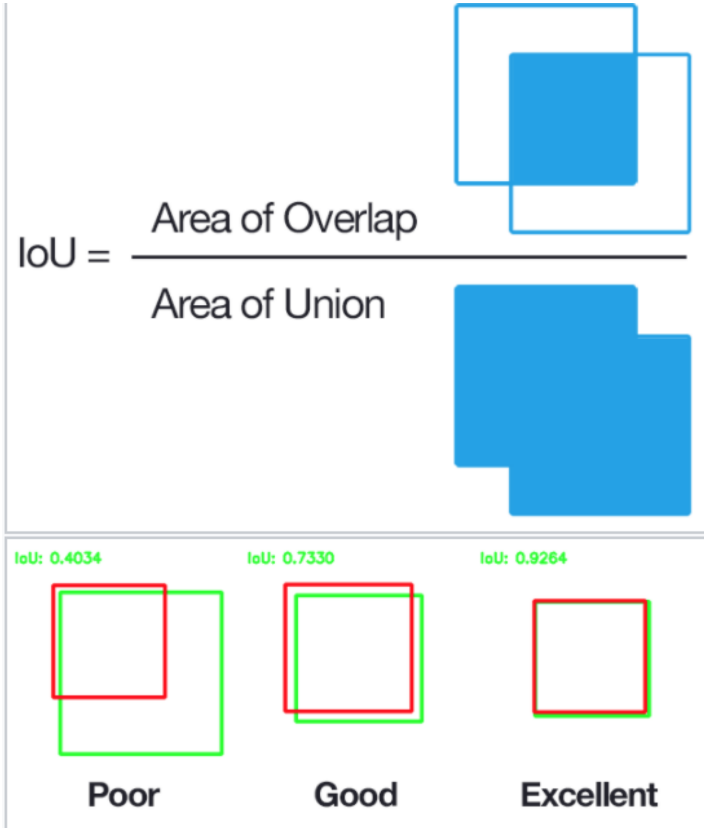




TensorRT can help to otimize the trained ANN

AI-Inference – Rate your Inference

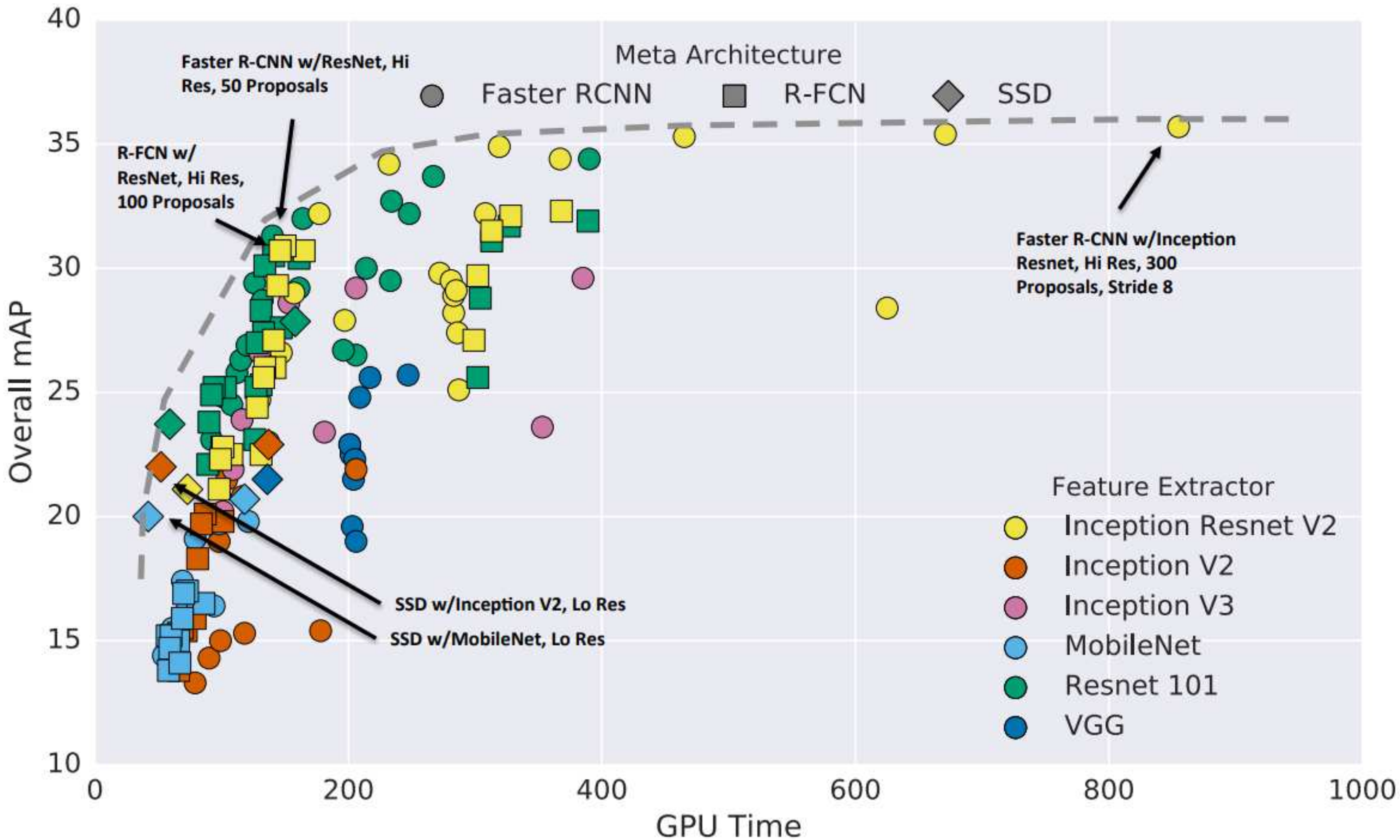




Intersection over Union (IoU)

Source: https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab

AI-Inference – Rate your Inference



GPU load / Images per second / GPU calculation time / mean average precision (mAP)

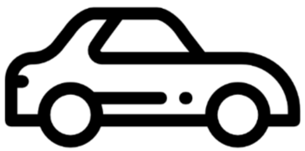
Source: https://arxiv.org/pdf/1611.10012.pdf

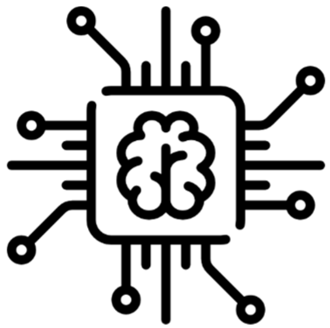
AI-Development
Johannes Betz / Prof. Dr. Markus Lienkamp / Prof. Dr. Boris Lohmann

(Johannes Betz, M. Sc.)

Agenda

1. Chapter: AI-Development Pipeline
2. Chapter: Transfer Learning
3. Chapter: AI-Frameworks
4. Chapter: Data and Labeling
5. Chapter: GPU Computing
6. Chapter: Hyperparamter Tuning
7. Chapter: AI-Inference
8. **Chapter: Summary**





Summary

What did we learn today:

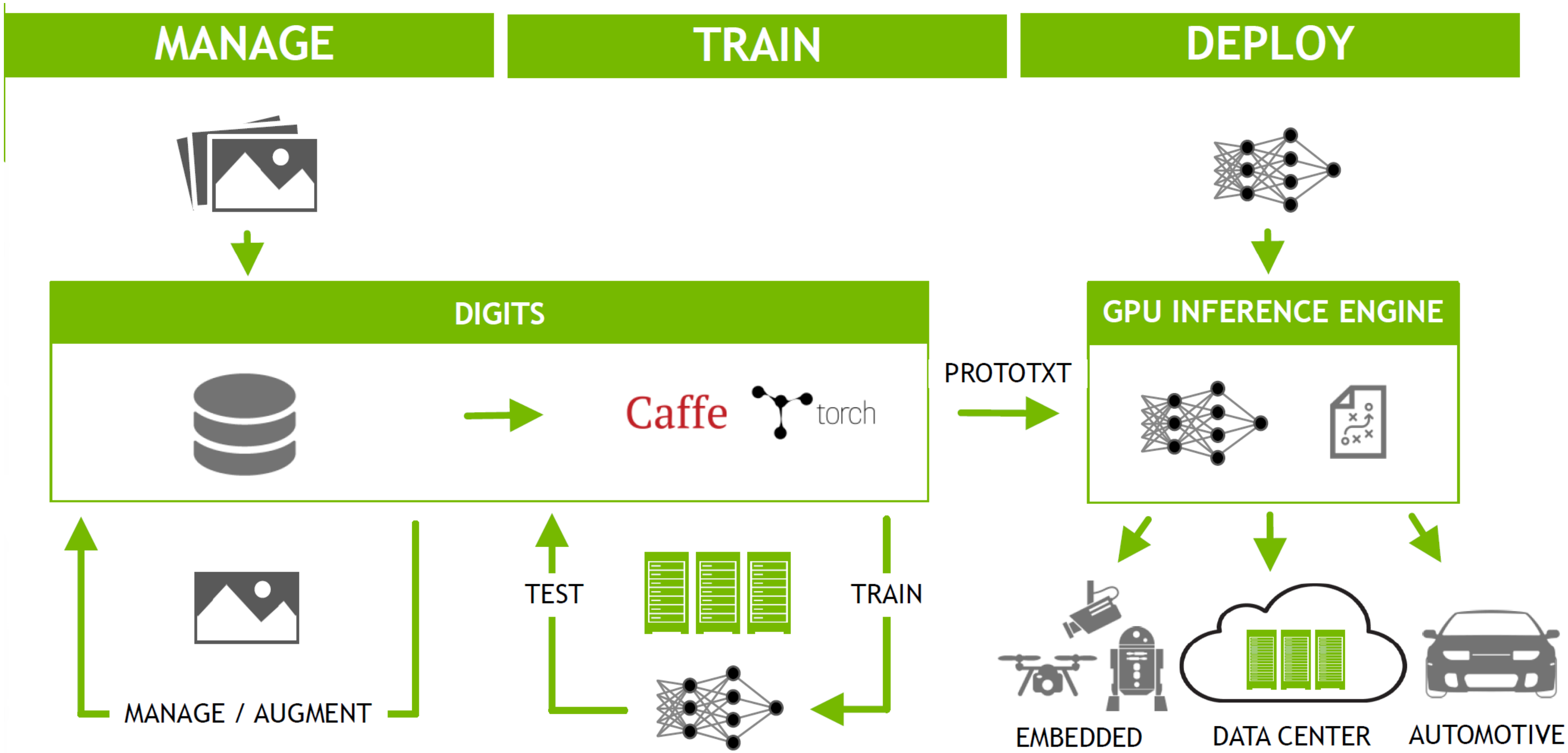
- Deep Learning algorithm development is all about the problem: **Same same but different**
- Take advantage of **pre-trained networks** and use them for your problem. In addition you can used published ANN architectures for your problem by using **transfer learning.** You will save a lot time
- There are a lot of **frameworks** for setting up your ANN in code. Check first what do you need (e. g. multiple GPUs, inference hardware) and decide what you want to use
- Be aware of the data you need. Search for **labeled datasets** first before you acquire your own data
- **Data is the crucial part** in deep learning:
  - The better and specific your data, the better your results
  - The more data you have, the better your results

Summary

What did we learn today:

- **GPUs** are your number one tool for fast training and fast inference
- Scale your model for training on multiple GPUs
- ANN development is all about **hyperparamter tuning.** Check the crucial parameters first and try to understand your ANN.
- When your ANN is ready, it is time for the real live experience that we call **Inference**. You can test your ANN model in your application and evaluate it afterwards
- Hardware like the **Jetson (embedded)** and **DrivePX2 (automotive)** enable software SDK and GPU accelerated Inference for your development

Summary



MANAGE
TRAIN
DEPLOY
DIGITS
Caffe
torch
PROTOTXT
GPU INFERENCE ENGINE
TEST
TRAIN
MANAGE / AUGMENT
EMBEDDED
DATA CENTER
AUTOMOTIVE

Guest Lecturer Rasmus Rothe – 07.02.2018

What is the talk about?

- Insights on the differences between academia and industry when applying deep learning
- Technical tricks for building robust real-world deep learning applications
- Advice on how to start an AI company

Rasmus Rothe

- Born in Bremen, 29 years
- PhD in Deep Learning from ETH Zurich, Master from Oxford / Princeton
- Co-Founder and CTO of Merantix, a Berlin-based AI company
- Founded and runs KI Bundesverband
- Founded HackZurich
- World champion in Robocup Junior

Evaluation

Evaluation

- In this lecture we are doing in regularly evaluation
- We want **your** feedback for every **individual** session
- We evaluate the session each week
- We give feedback based on the evaluation the week after

Evaluation – Step by Step

1. Get out your smartphones
2. Open an app for QR-code reading
3. Read the following QR-code on the right side →
4. Open the website
5. Answer the questions
6. Send the evaluation

OR

1.Open the following website in your browser: https://evasys.zv.tum.de/evasys/online.php?p=AIAT-12

2. Answer the questions

3.Send the evaluation