KI Labor - Sommersemester 22

Termin 1 - Organisation & Computer Vision

Robin Baumann, Sven Müller, Maximilian Blanck,
Matthias Richter, Tim Bossenmaier, Pascal Fecht

Karlsruhe, 18. Mär. 2022

Agenda für Heute

1. Organisation
    a. Teamvorstellung
    b. Modus
    c. Bewertung
2. Computer Vision

Vorstellung

Modus

Computer Vision

2 Sprints

→

Natural Language Processing

2 Sprints

→

Reinforcement Learning

2 Sprints

- Angelehnt an Scrum-Prozess
  - Sprint dauert 2 Wochen
  - **Sprint 1:** Einarbeitung (Übungsaufgaben)
  - **Sprint 2:** Assignment
- Teilweise in Präsenz und Remote

Sprint-Modus

- **Sprintwechsel** (i.d.R) alle 2 Wochen in Präsenz
  - **Review**: Lösungen präsentieren
  - **Retro**: Was lief gut? Was lief schlecht?
  - **Planning**: Neue Aufgaben / Assignment

- Im Sprint: Remote-Betreuung
  - Freitags: **Q&A Timeslots** (per Zoom / Meet)

    30 Minuten gemeinsam,

    15 Minuten individuell für jede Gruppe
  - Dazwischen über Slack-Workspace:
    https://join.slack.com/t/kilaborss22/shared_invite/zt-151obzfgg-eNeQXE2Iseb3vrLvt8zmIQ

Semesterplan

Datum	Thema	Inhalt	Präsenz
18.03.22	**Allg.**	**Organisation, Teamfindung, Vorstellung CV**	Ja
25.03.22	CV	Q&A Sessions	Nein
01.04.22	CV	Sprintwechsel, Vorstellung Assignment	Ja
08.04.22	CV	Q&A Sessions	Nein
05.04.22	Ostern		
22.04.22	**CV / NLP**	**Abgabe CV, Vorstellung NLP**	**Ja**
29.04.22	NLP	Q&A Sessions	Nein
06.05.22	NLP	Sprintwechsel, Vorstellung Assignment	Ja
13.05.22	NLP	Q&A Sessions	Nein
20.05.22	**NLP / RL**	**Abgabe NLP, Vorstellung RL**	**Ja**
27.05.22	RL	Sprintwechsel, Vorstellung Assignment	Ja
03.06.22	Sommerplenum		
10.06.22	Pfingsten (H-KA zu)		
17.06.22	RL	Q&A Sessions (Brückentag)	Nein
24.06.22	**RL**	**Abgabe RL, Abschluss KI Labor**	**Ja**
01.07.22		Puffer	

Hardware

- Bearbeitung der Aufgaben auf
  - Pool-Rechnern
  - Eigener Hardware
  - Cloud

- Cloud: Google Colab
  - Kostenlose GPUs und TPUs!
  - Benötigt Google-Account (und Internet)

- Cloud 2: Kaggle
  - Training auch mit geschlossenem Browser-Fenster möglich

Gruppenfindung

1. Zu 3er/4er Teams zusammenfinden
   - 16 Teilnehmer:innen → 4-5 Teams

2. Team-Name überlegen

3. Mitglieder und Team-Name mitteilen (slack)

Bewertung

- Labor wird **benotet**
- Übungsaufgaben werden nicht benotet (aber müssen bestanden werden)
- Je Themenblock ein bewertetes Assignment und Präsentation
- Assignment
  - Jupyter Notebook
  - Abgabe vor der Präsentation
  - Jedes Team gibt eigene Lösung ab
- Präsentation
  - Jede/r in den Gruppen sollte Redeanteil haben
  - Müssen keine Slides sein (Notebook zeigen)
  - 15 - 20 Minuten je Gruppe

Folien und Aufgaben

siehe Github Repository

⇒ https://github.com/inovex/ai-lab

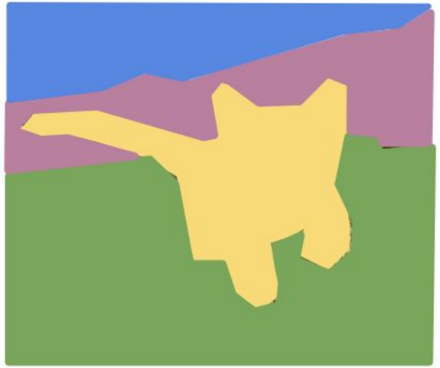
Computer Vision

Agenda for today

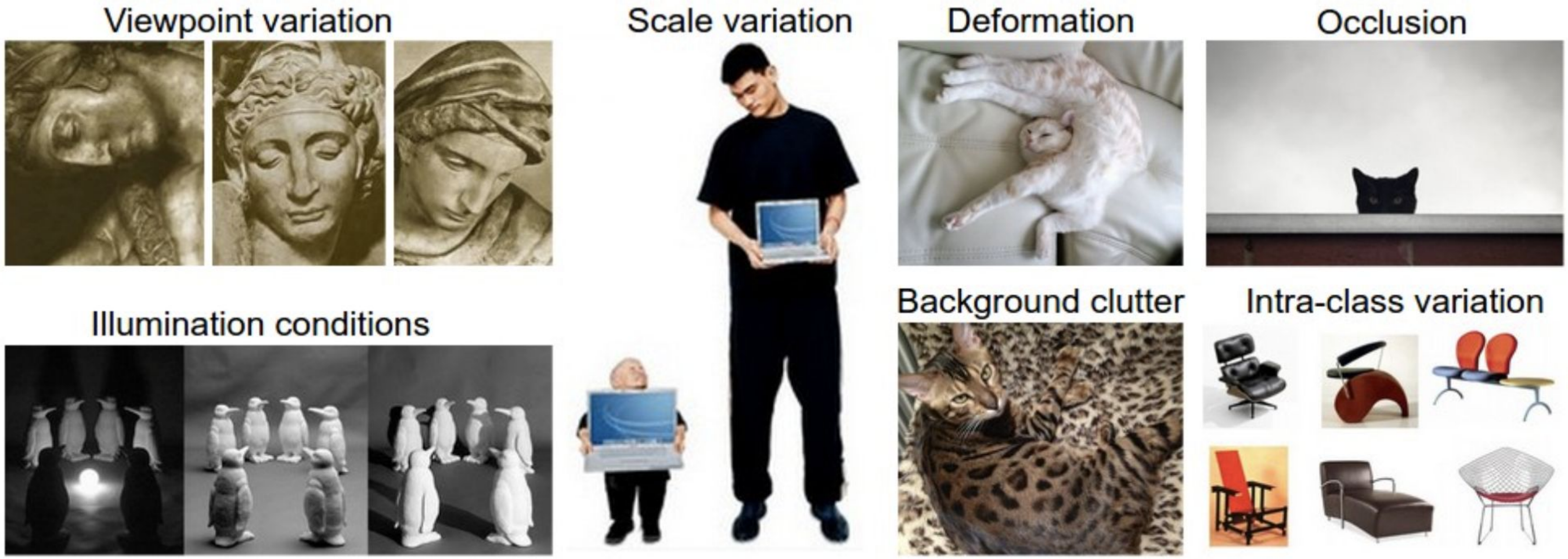
1. Introduction
2. Deep Learning
3. Analysing Model/Training performance
4. Exercise Notebooks

Introduction

What is Computer Vision (CV)?




CV has a number of challenges to overcome

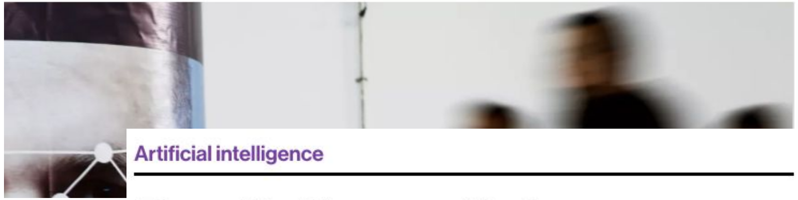


Can we trust machines to make fair decisions?

Research shows AI is often biased. Here's how to make algorithms work for all of us

RETAIL OCTOBER 11, 2018 / 1:04 AM / UPDATED 3 YEARS AGO

Amazon scraps secret AI recruiting tool that showed bias against women



Artificial intelligence

Predictive policing algorithms are racist. They need to be dismantled.

Lack of transparency and biased training data mean these tools are not fit for purpose. If we can't fix them, we should ditch them.

by **Will Douglas Heaven** July 17, 2020

Battling bias and other toxicities in natural language generation

Despite numerous and concerted efforts to train NLG systems to generate content without offensive elements, success is still elusive.

Your favorite A.I. language tool is toxic

BY **JONATHAN VANIAN**
September 29, 2020 5:25 PM GMT+2

inovex

The SOTA algorithms for solving CV problems are based on deep learning

Deep Learning

What is deep learning?

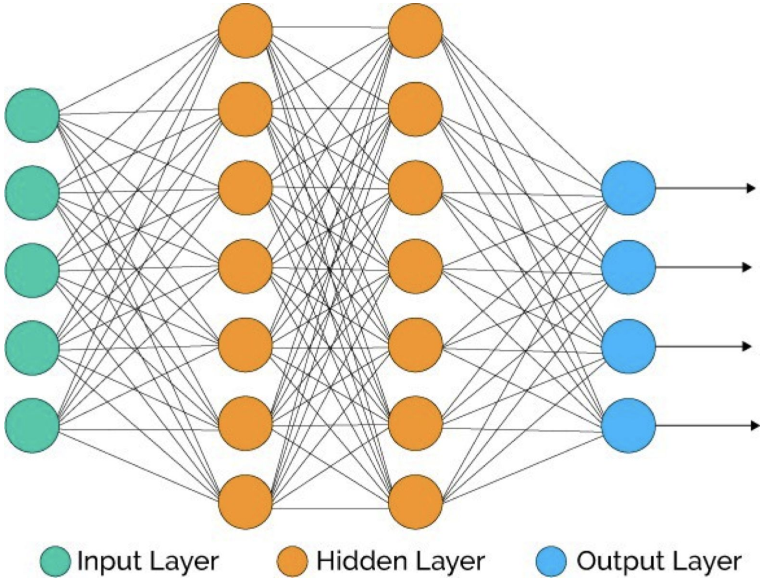
*“[...] very large neural networks we can now have and …
huge amounts of data that we have access to [...]” - Andrew Ng (2015)*

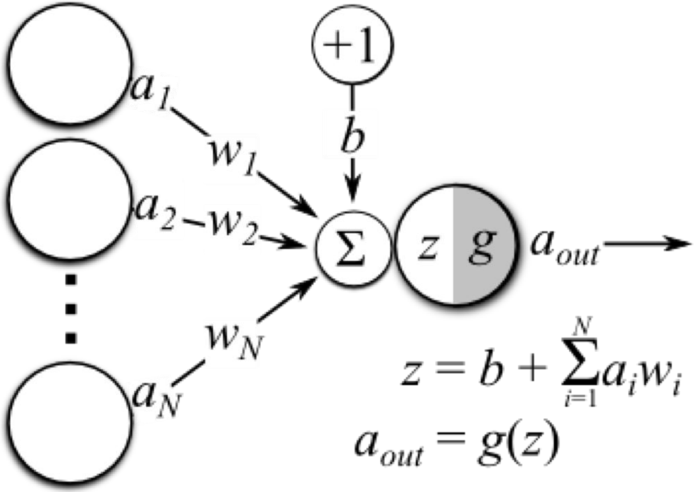
“Deep learning methods aim at learning feature hierarchies [...] at multiple levels of abstraction allow[ing] a system to learn complex functions mapping the input to the output directly from data, without depending completely on human-crafted features.” - Yoshua Bengio (2009)

“It has been obvious since the 1980s that backpropagation through deep autoencoders would be very effective for nonlinear dimensionality reduction, provided that computers were fast enough, data sets were big enough, and the initial weights were close enough to a good solution. All three conditions are now satisfied.” - Geoffrey Hinton (2006)

Let’s start with building a simple Neural Network

Multi-Layer Perceptron

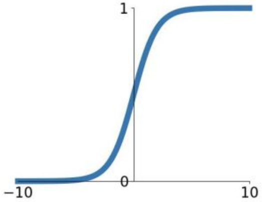




We can learn complex functions by applying non-linear activation functions

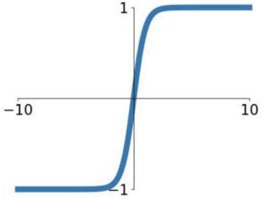
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



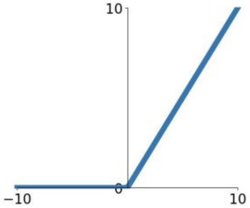
tanh

$\tanh(x)$



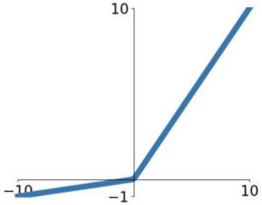
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

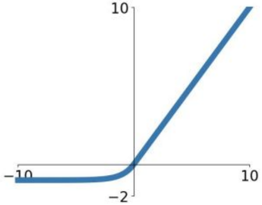


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

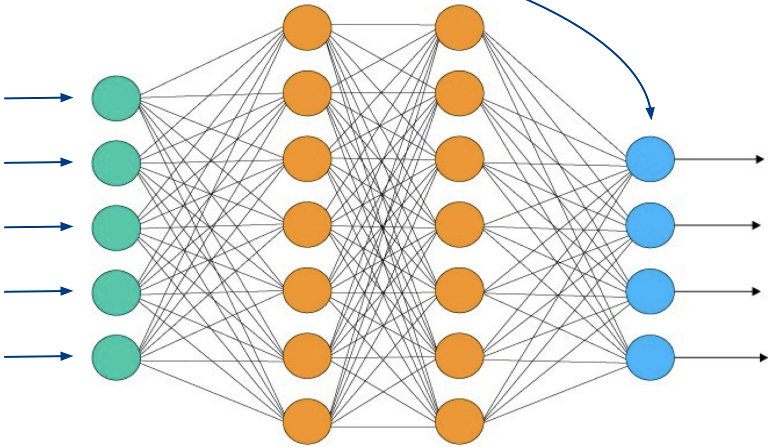
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



But how do we actually learn?

Forward Pass



Common loss function

$$-\frac{1}{N}\sum_{i=0}^{N}\sum_{c=1}^{M} y_c log(\hat{y}_c)$$

OR

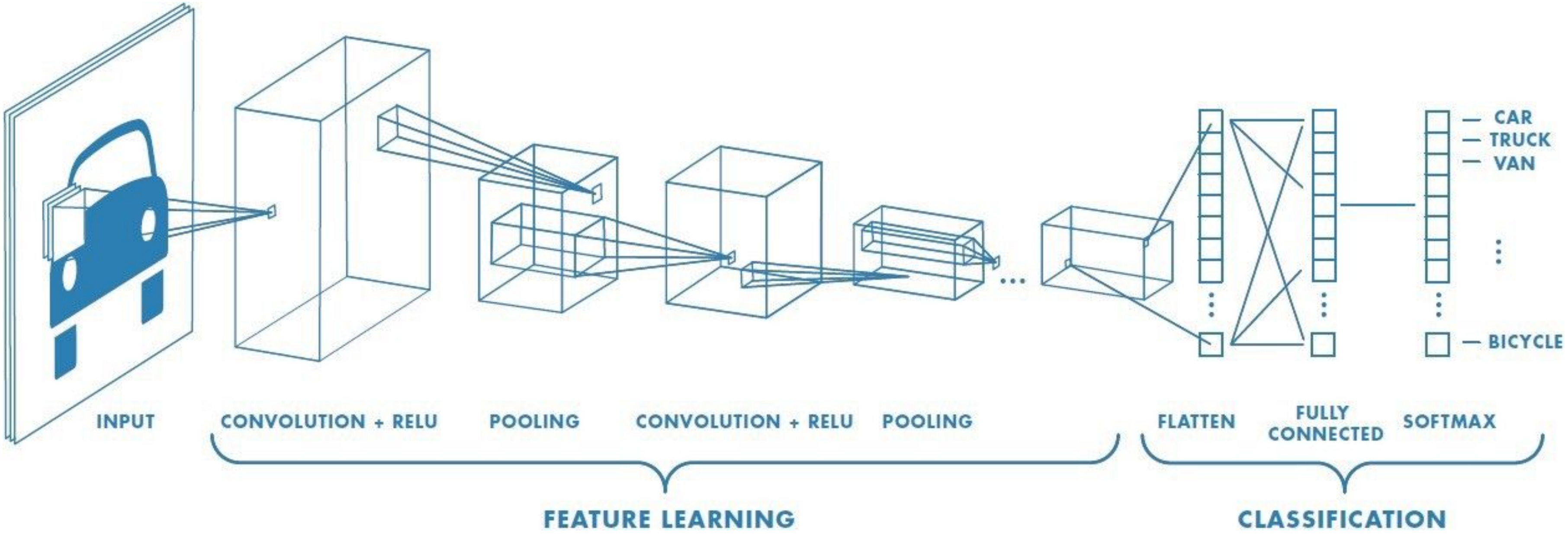
$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$w_i \leftarrow w_i - \alpha \frac{\partial L(w)}{\partial w_i}$$

Backward Pass

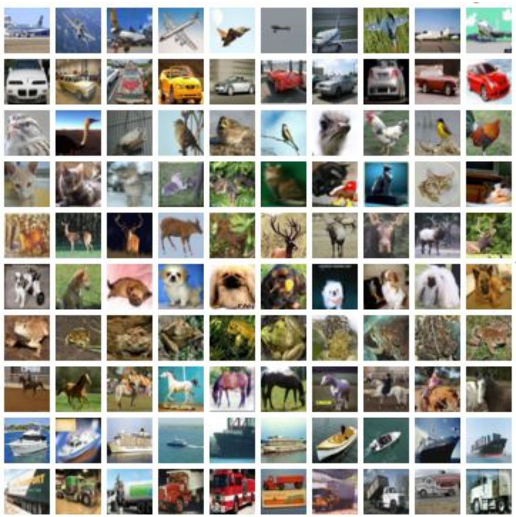
Let’s advance to more complex Neural Networks

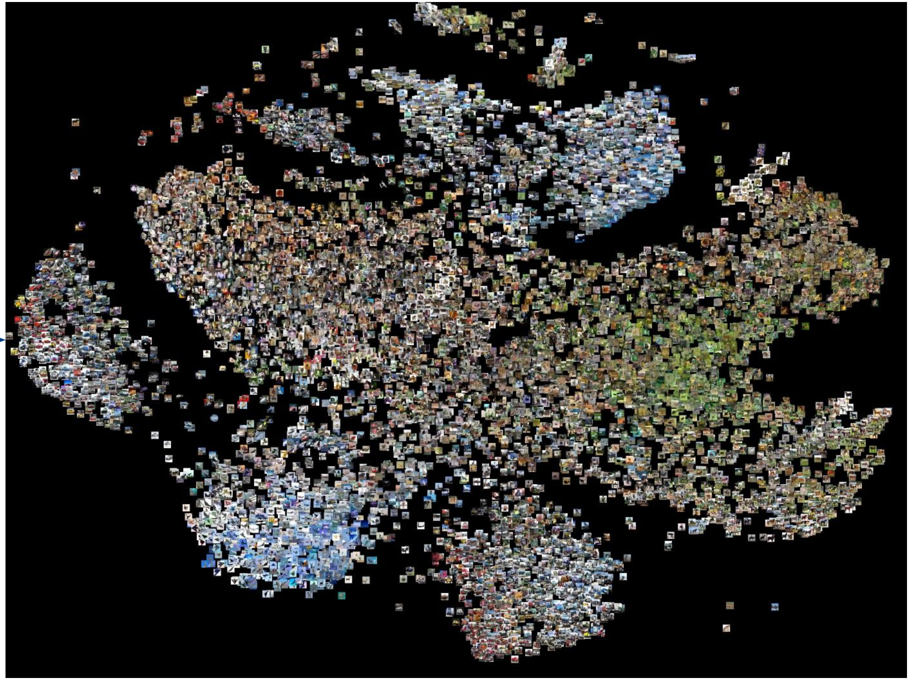
Convolutional Neural Network



Can we show the discriminative power of NNs?

t-Distributed Stochastic Neighbor Embedding (t-SNE)

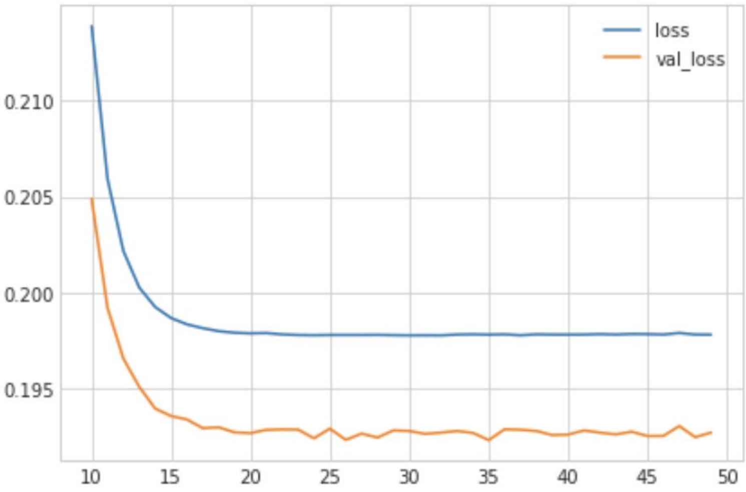




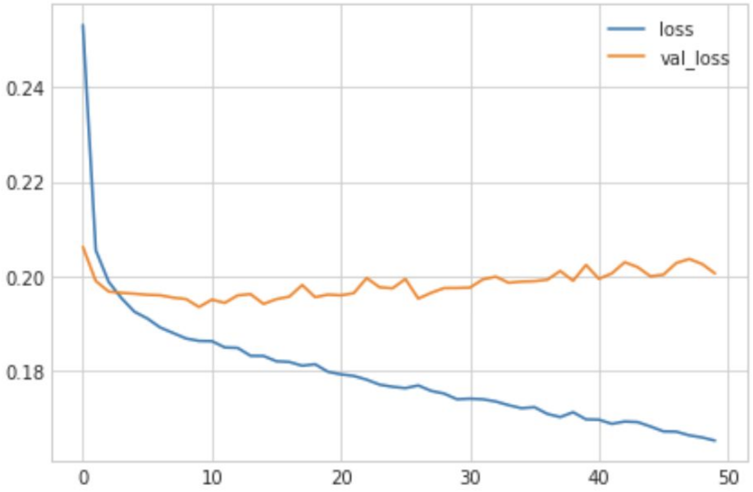
inovex

Analysing Model/Training performance

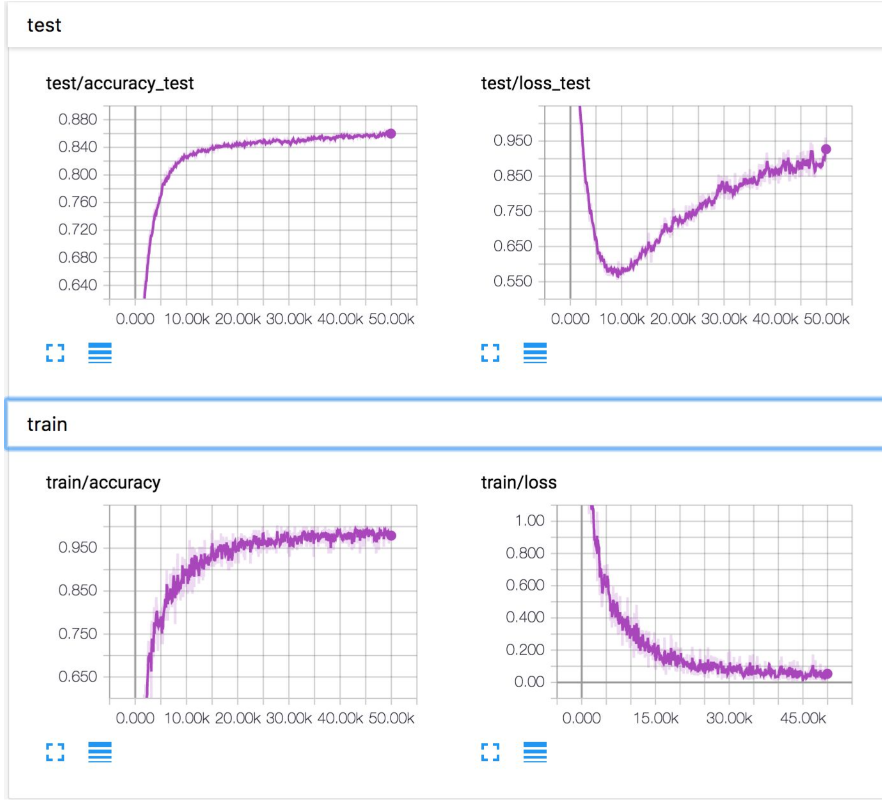
What happened?



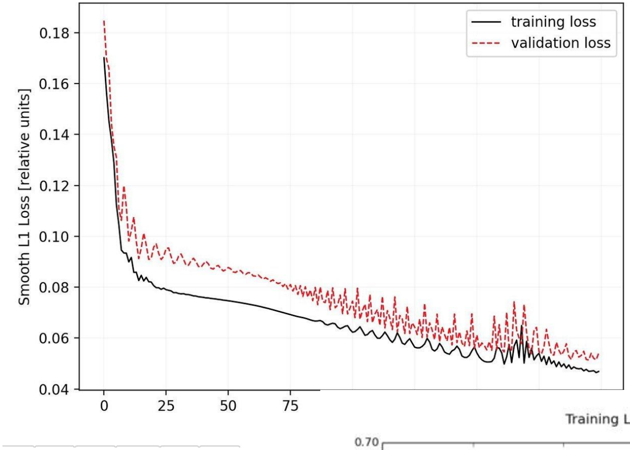
What happened?

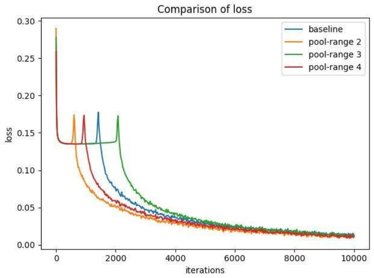


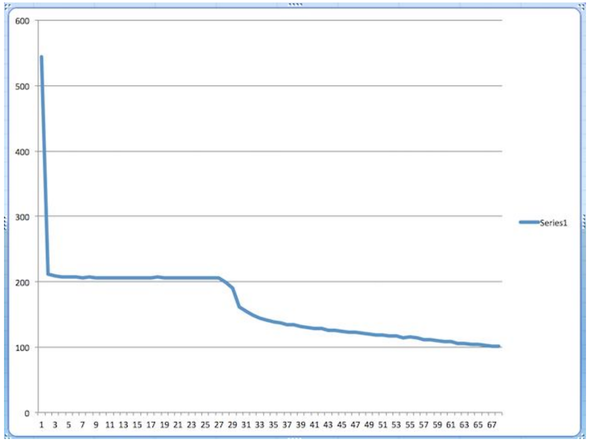
What happened?

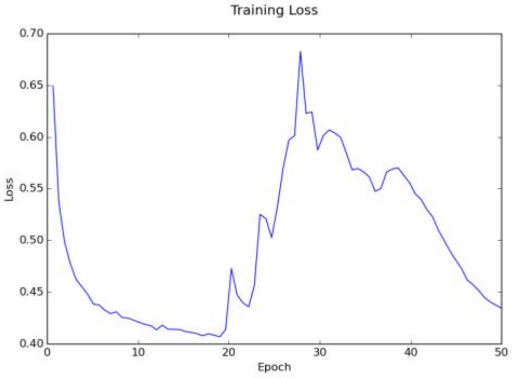


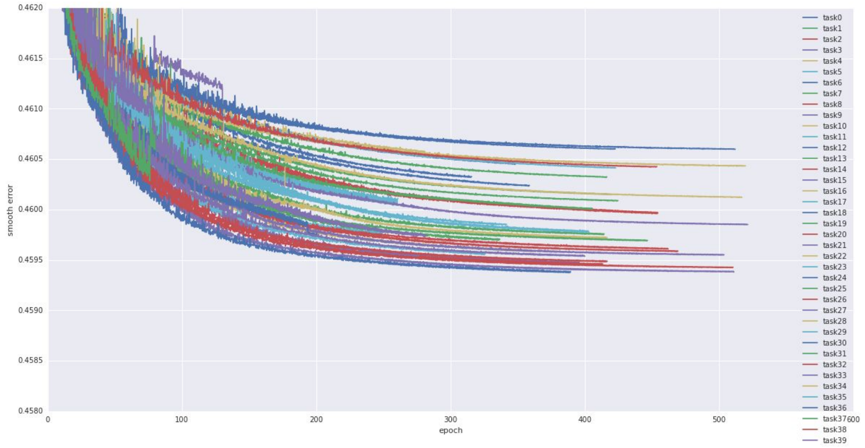
What happened?







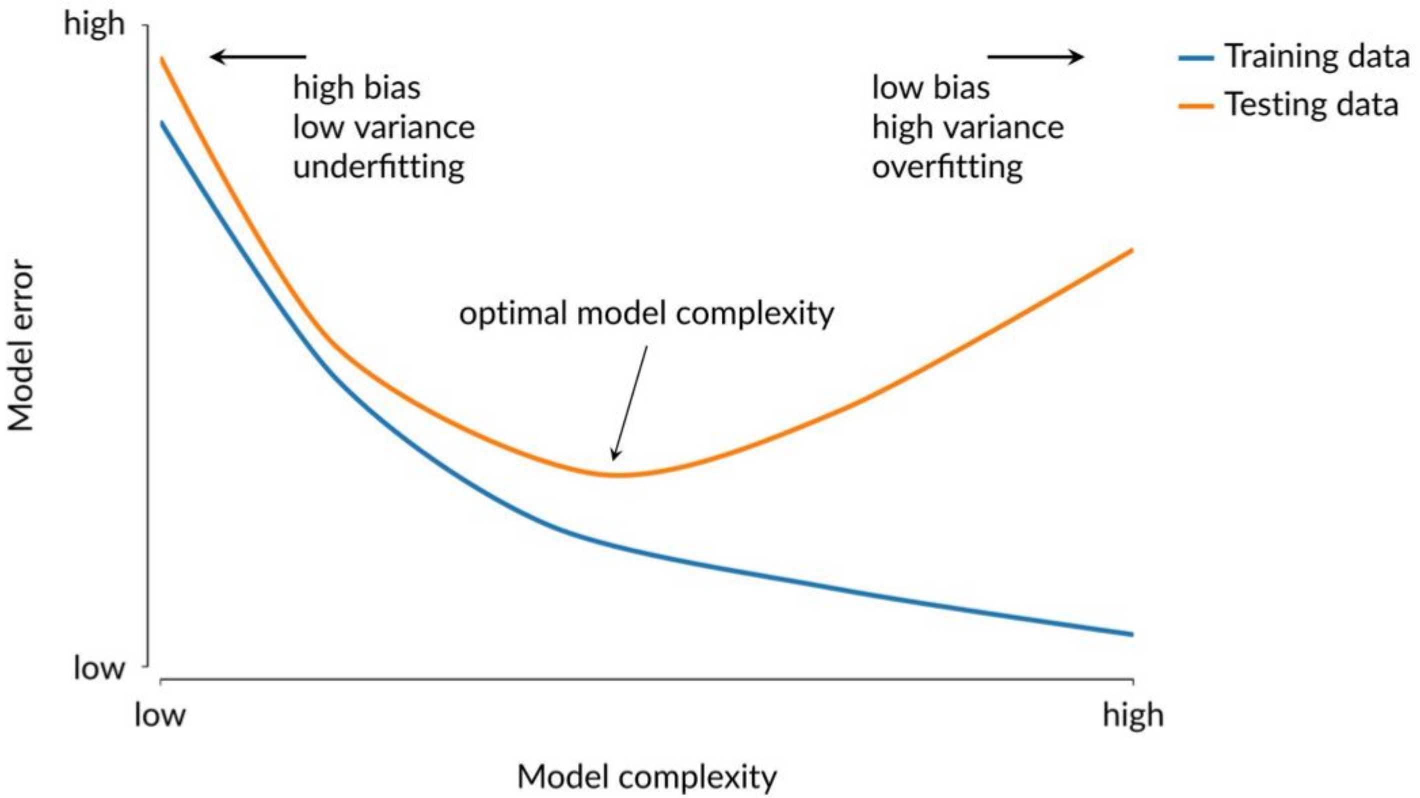




Quelle: https://lossfunctions.tumblr.com/ (11.03.2022)

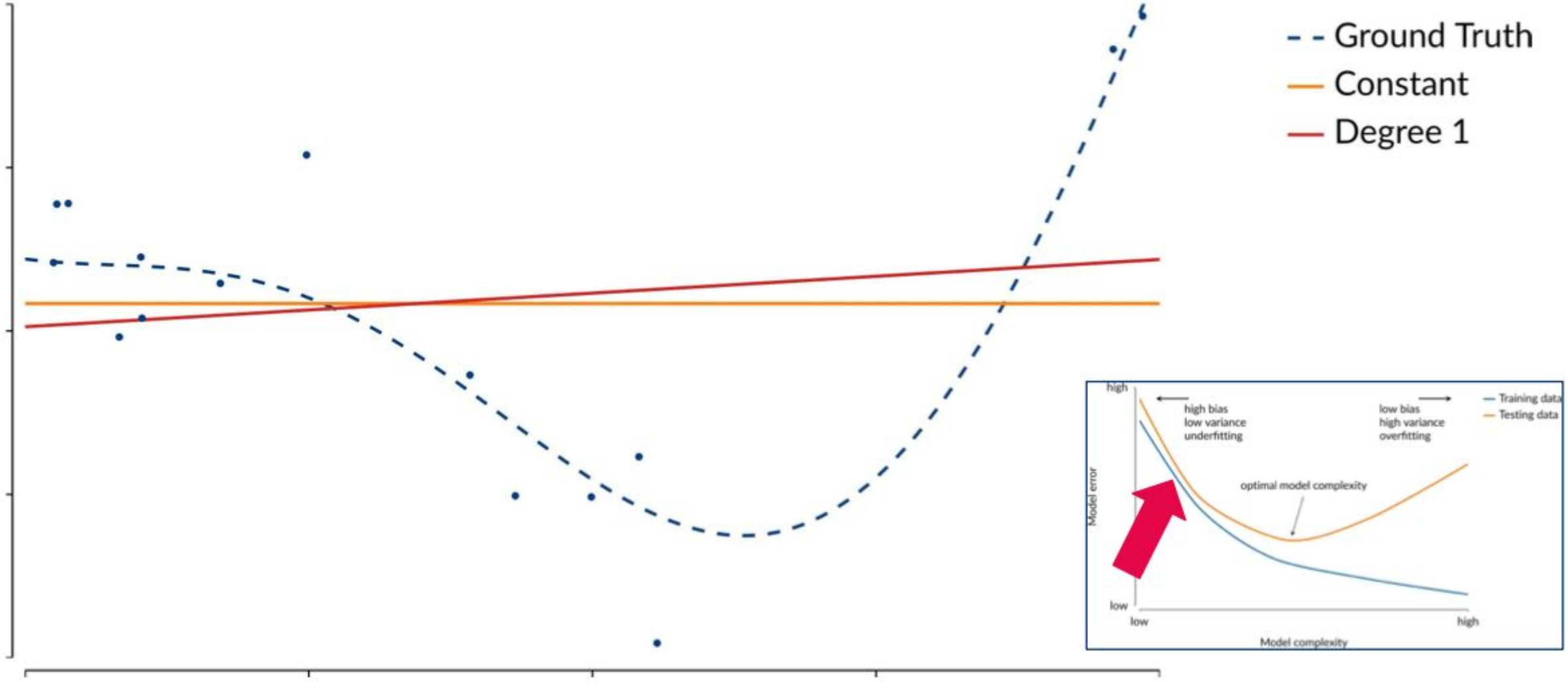
inovex

Bias-Variance-Tradeoff



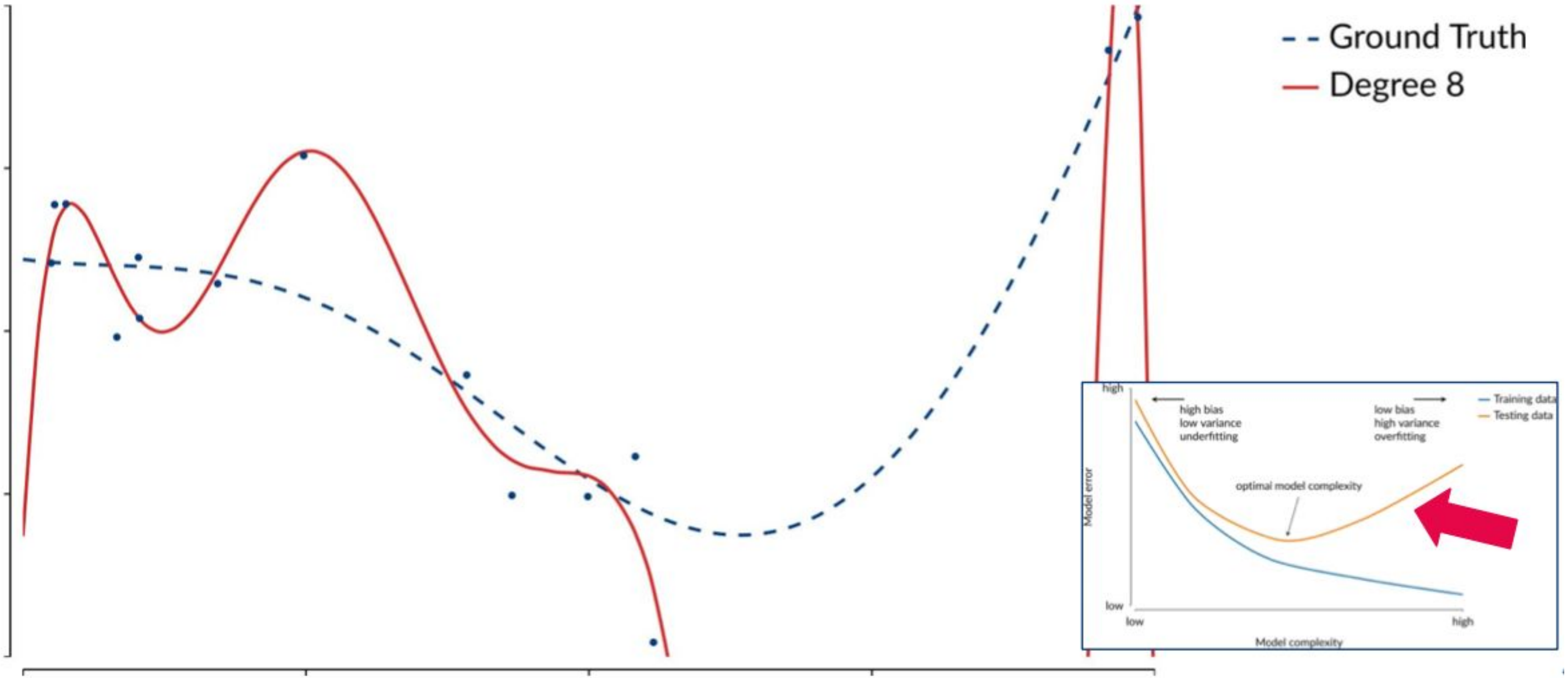
Bias-Variance-Tradeoff

Underfitting: Model is too simple



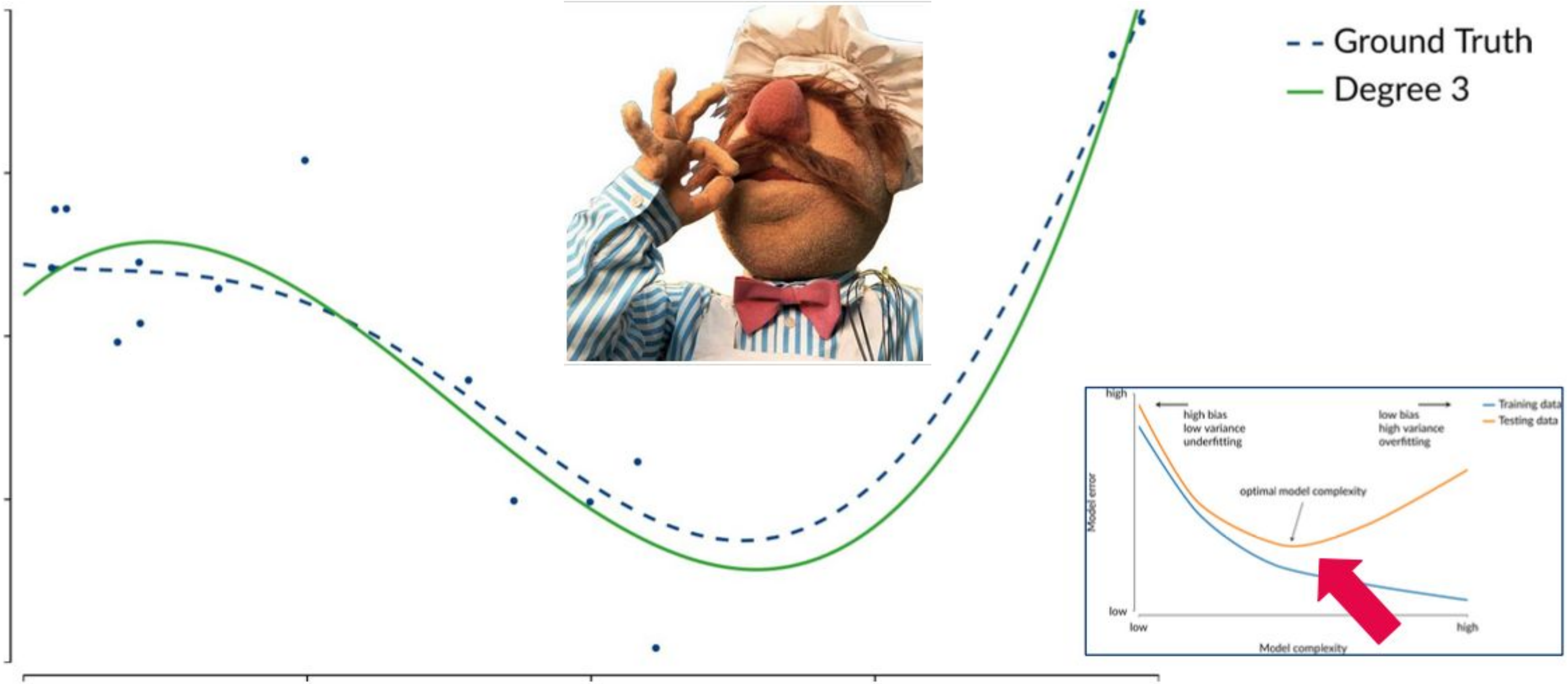
Bias-Variance-Tradeoff

Overfitting: Model is too complex



Bias-Variance-Tradeoff

Optimal model complexity



What about neural networks?

- Compared to polynomials, the complexity / variance of neural networks is extremely high

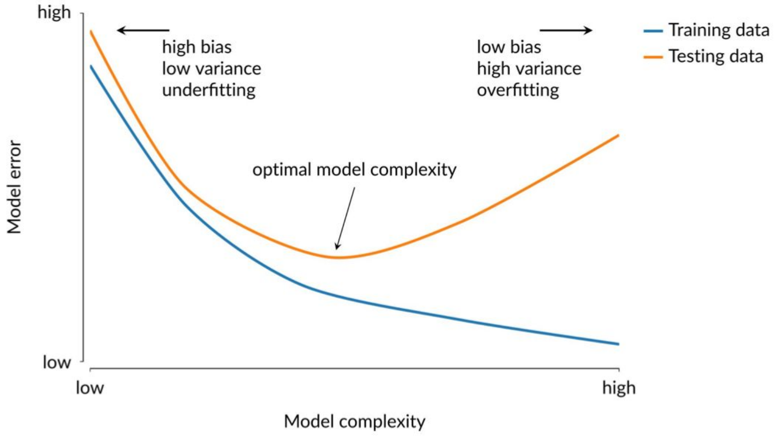
- Universal Approximation Theorem

- The tasks / targets are also very complex

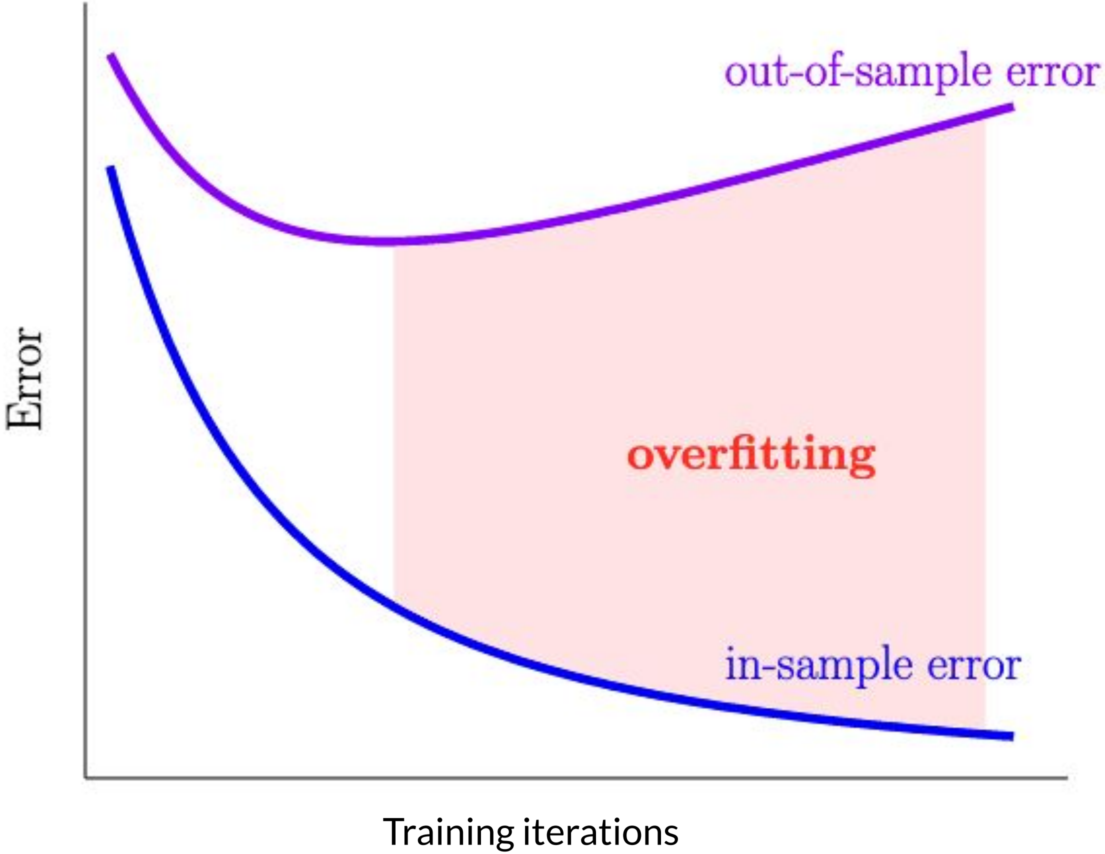
inovex

Preventing overfitting in neural networks

- Using Validation set
  - Network size fine-tuning
  - Early stopping
- Regularization methods
  - Weight regularization
  - Dropout
- Data based methods
  - Data Augmentation
  - Noise
  - Extending the dataset



Early Stopping



inovex

Weight regularization

- Penalizing a NN based on the size of the weights

$$weight\ decay\ L1 = \sum_i |\theta_i|$$

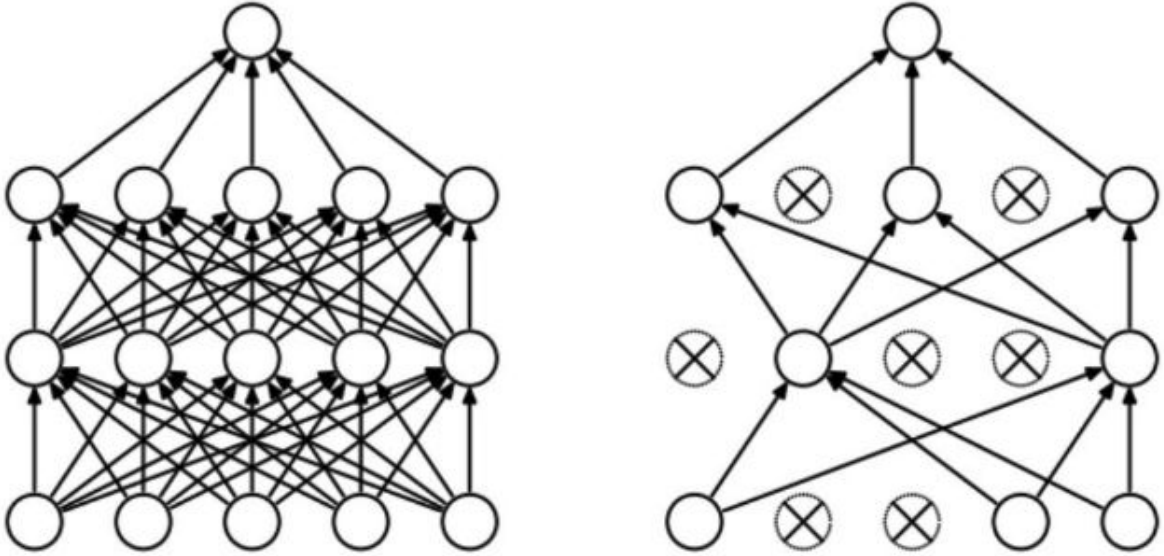
$$weight\ decay\ L2 = \sum_i \theta_i^2$$

- Variation of scales of input variables causes the scale of the weights of the network to vary accordingly
  - Problematic for weight regularization
  - Solution: normalization, standardization

inovex

Dropout

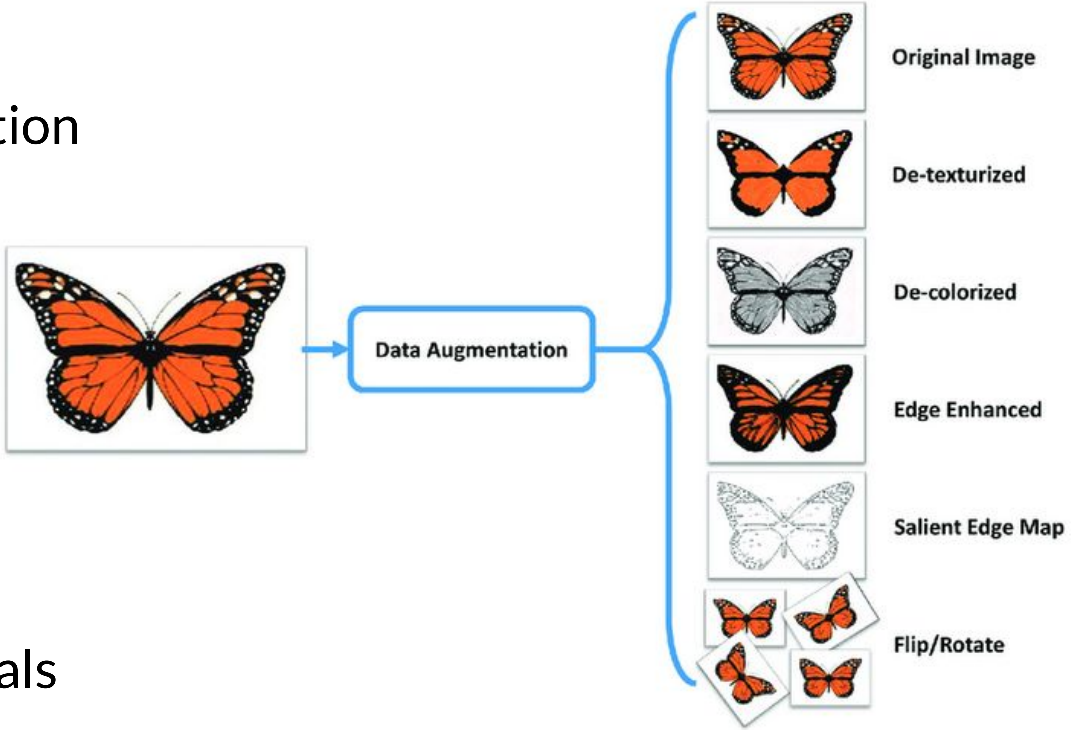
- “Ephemeral sparsity”
- applied only during the training phase



(a) Standard Neural Net (b) After applying dropout.

Data based methods

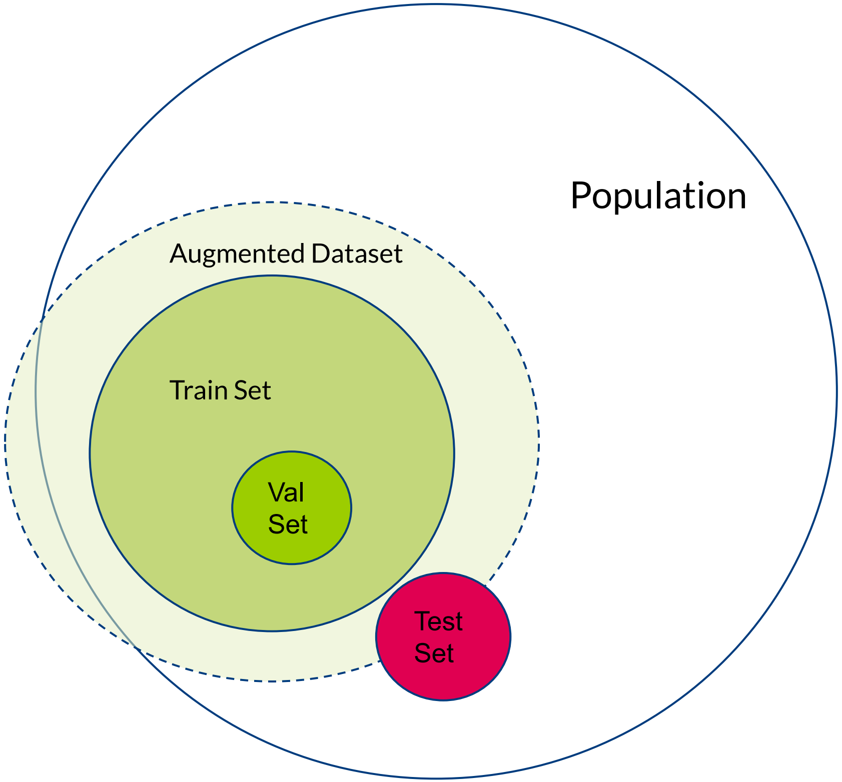
- Data Augmentation
  - Synthetic data generation
  - Data modification



- Noise
  - Useful for natural signals
- Extending the dataset
  - Used in real world & kaggle competitions

Figure from "[Data augmentation-assisted deep learning of hand-drawn partially colored sketches for visual search](https://doi.org/10.1371/journal.pone.0183838)" DOI:10.1371/journal.pone.0183838

Augmentation



Vielen Dank

inovex GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe

inovex