Dropout: A regularization technique

Prof. Dr.-Ing. Katharina Breininger

katharina.breininger@fau.de

„AI and Data Science“

Lecture slides & materials

Course Context

- Lecture “Introduction to AI” or “Introduction to Machine Learning”
- Semester 4+
- Prerequisites & prior knowledge:
  - Basic principles of symbolic/subsymbolic AI
  - Multilayer perceptrons
  - Loss functions, optimization and training of NNs
  - Overfitting & underfitting

Foto von Oleksandr Pidvalnyi von Pexels
https://www.pexels.com/de-de/foto/grune-und-graue-schere-2831794/

Embedding

Last week: Optimization, overfitting & underfitting

This week: Regularization techniques

- **Recap**: Motivation & general concept of regularization
- L1-, L2-, Lp (norm constraint) regularization
- Normalization & Data augmentation
- Initialization
- Early stopping
- Dropout
- Advanced techniques
- Outlook and open research questions

Goals for Today

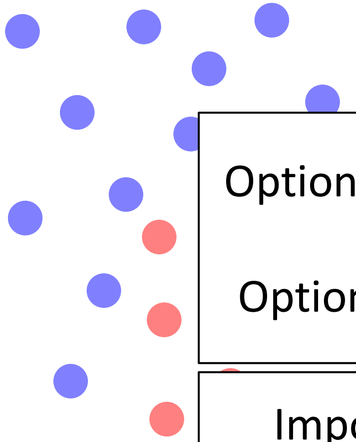
You will be able to …

- summarize the general concept of over- and underfitting
- explain the general working principle of dropout
- identify effects of dropout in neural network optimization
- discuss the connection to ensembling and identify further usecases

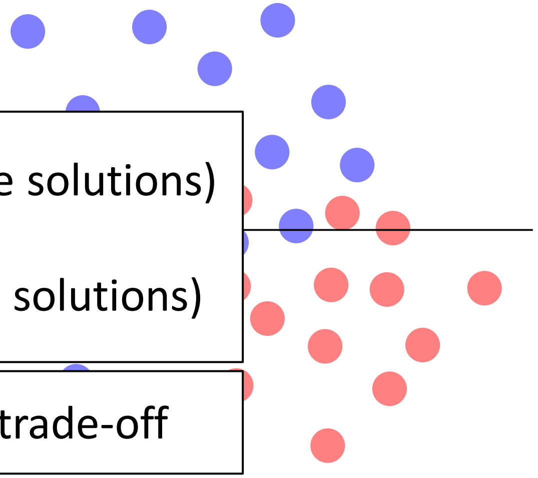
… and you will know where to find more information

Motivation & Recap: Fitting appropriately

Data



Underfitting

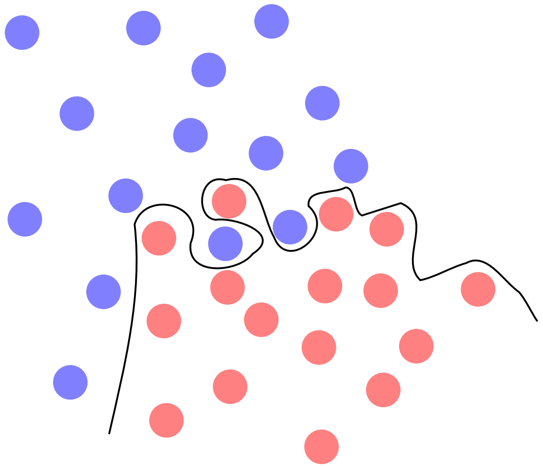


Option 1: **Model selection** (prohibit undesirable solutions)

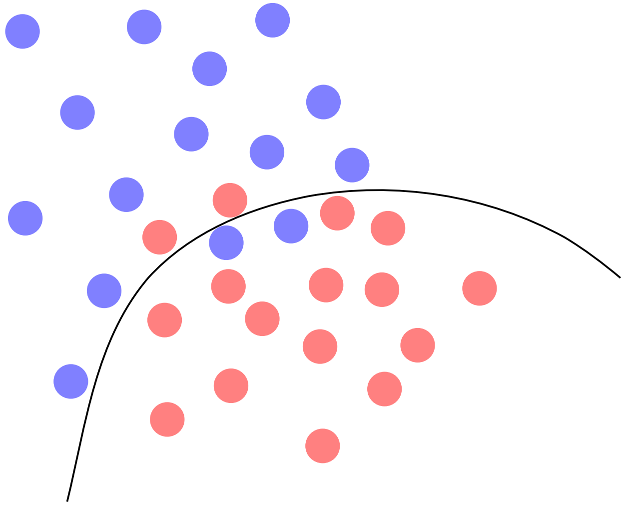
Option 2: **Regularization** (penalize undesirable solutions)

Important additional concept: bias-variance trade-off

Overfitting



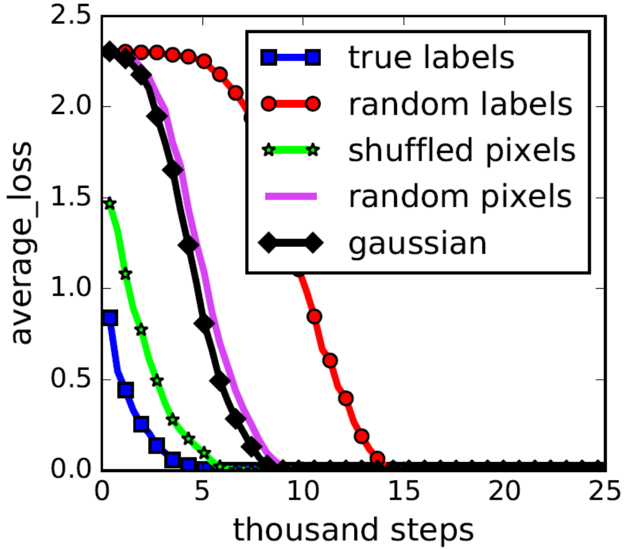
Sensible boundary



Motivation: Neural network capacity

- NNs contain millions of parameters (or more)
- Often trained on datasets << # parameters
- Networks can fit random data & random labels with 100% train accuracy [1]

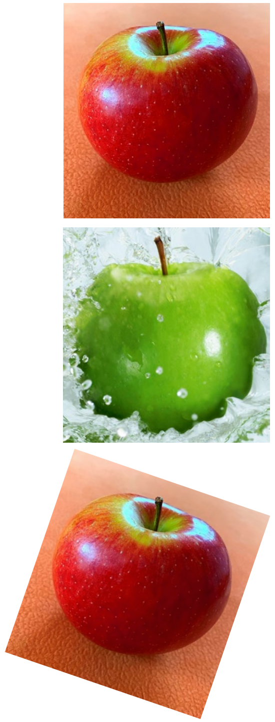
Adapted from: http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/

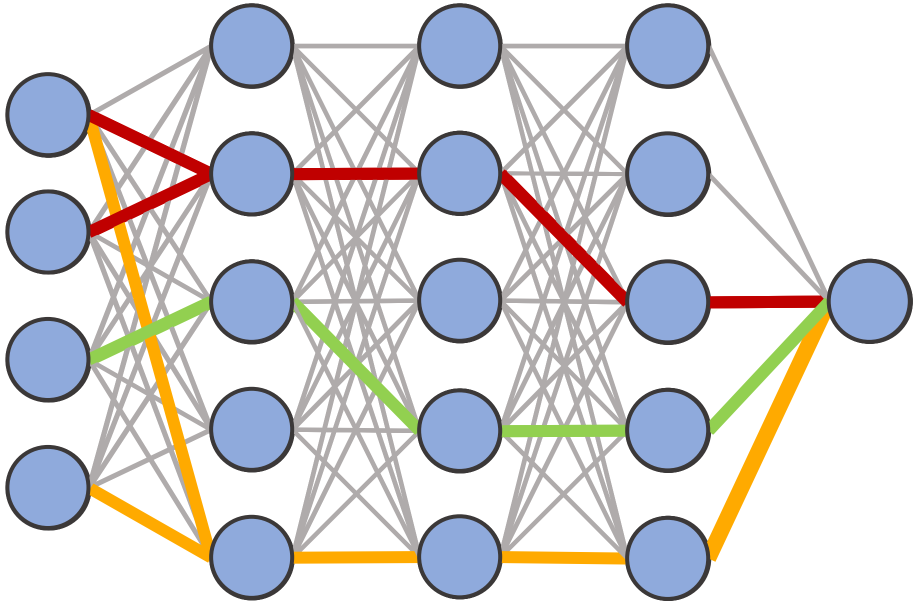


[1] Zhang et al. "Understanding deep learning requires rethinking generalization", ICLR 2017

Motivation:
Co-dependence in neural networks

Co-dependence:
Deeper neurons strongly depend on the activation of (few & specific) earlier neurons

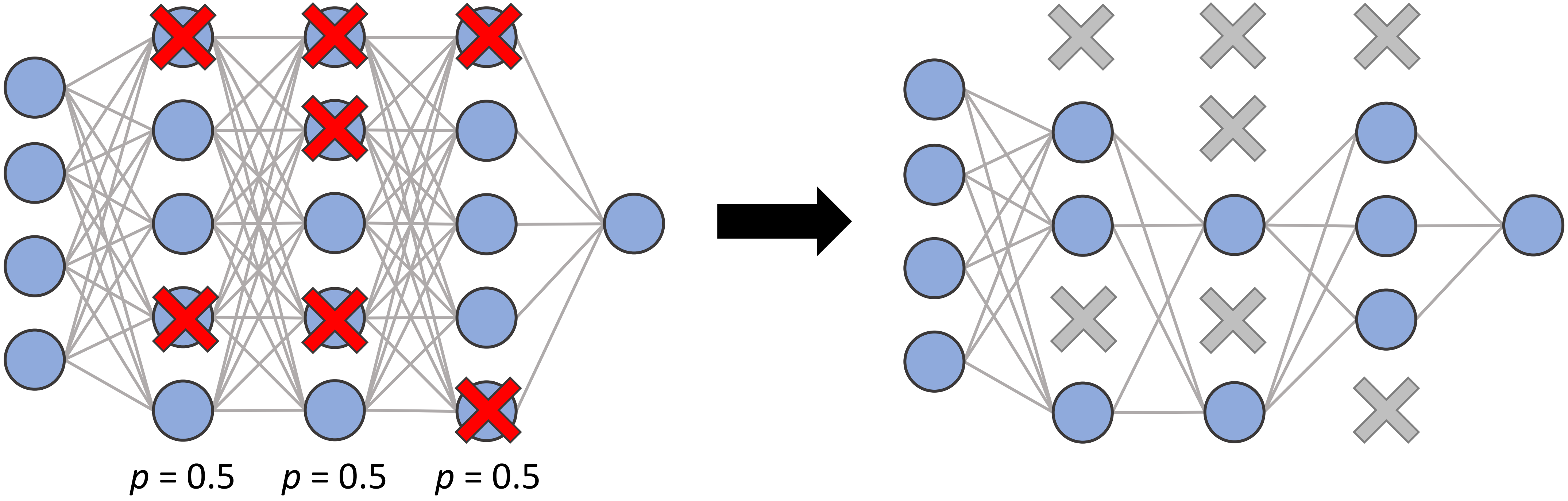




→Highly specific features
→Little feature reuse
→Low generalization

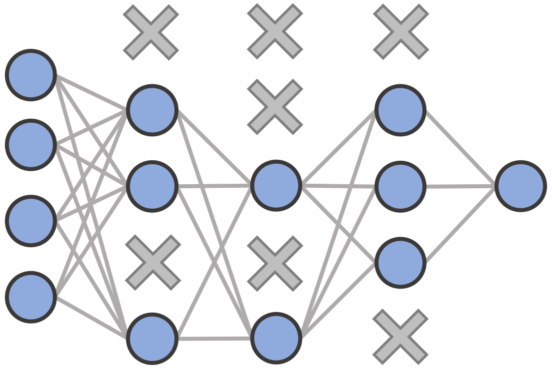
Breaking co-dependence: Dropout

Keep neurons in (hidden) layers with probability *p (per iteration)*

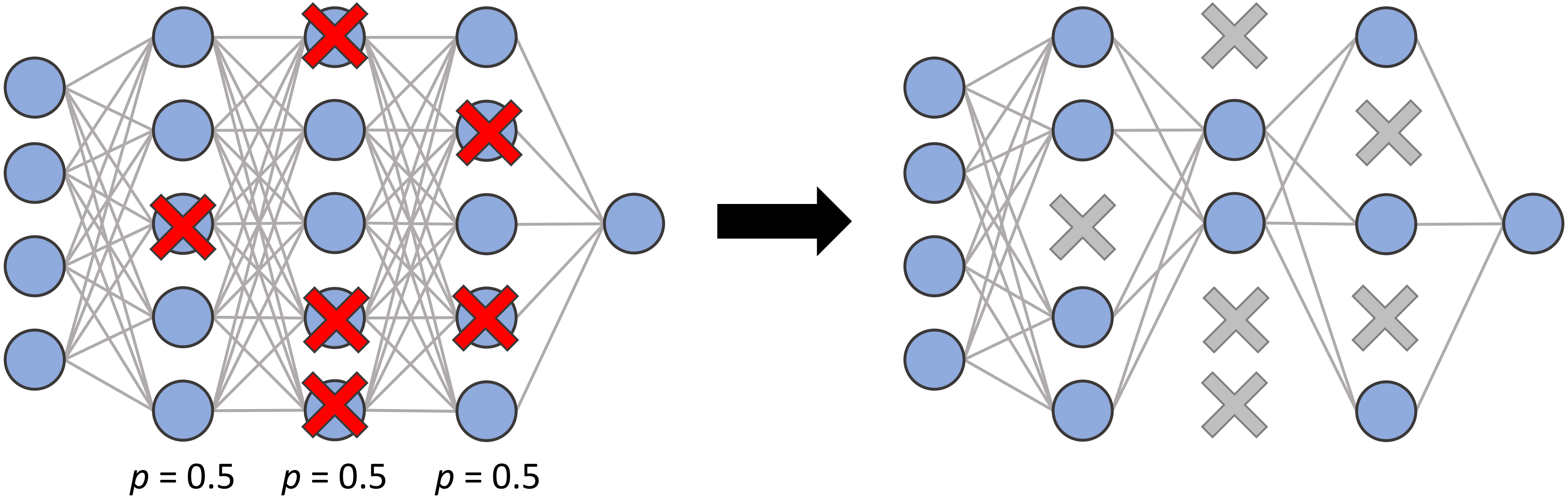


[2] Hinton et al. "Improving neural networks by preventing co-adaptation of feature detectors", 2012.

Breaking co-dependence: Dropout

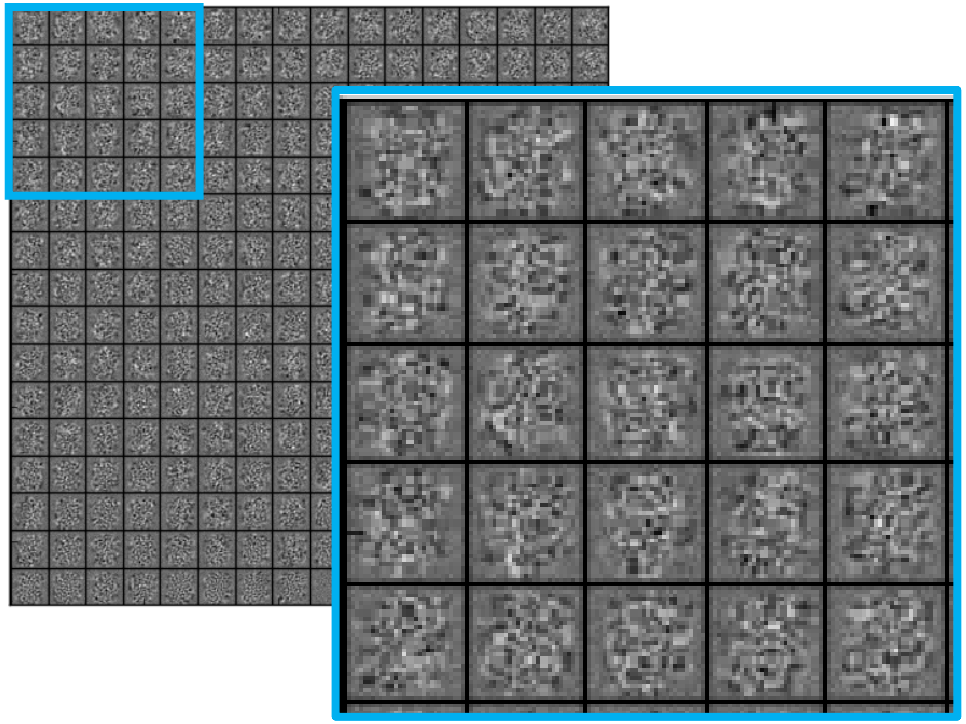


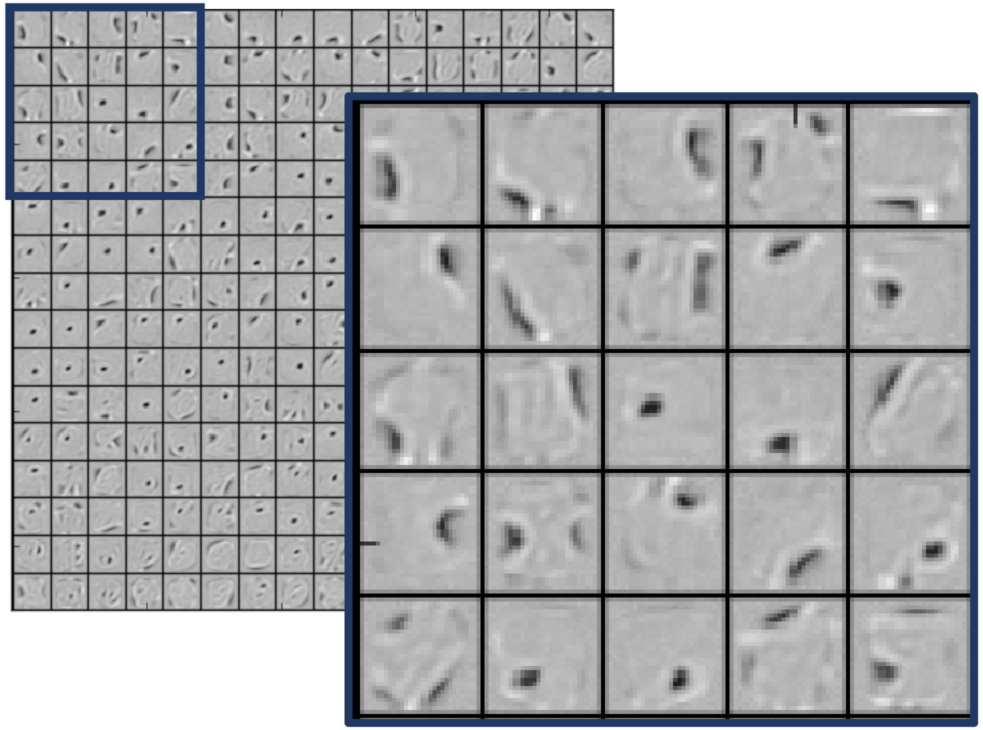
Keep neurons in (hidden) layers with probability *p (per iteration)*



[2] Hinton et al. “Improving neural networks by preventing co-adaptation of feature detectors”, 2012.

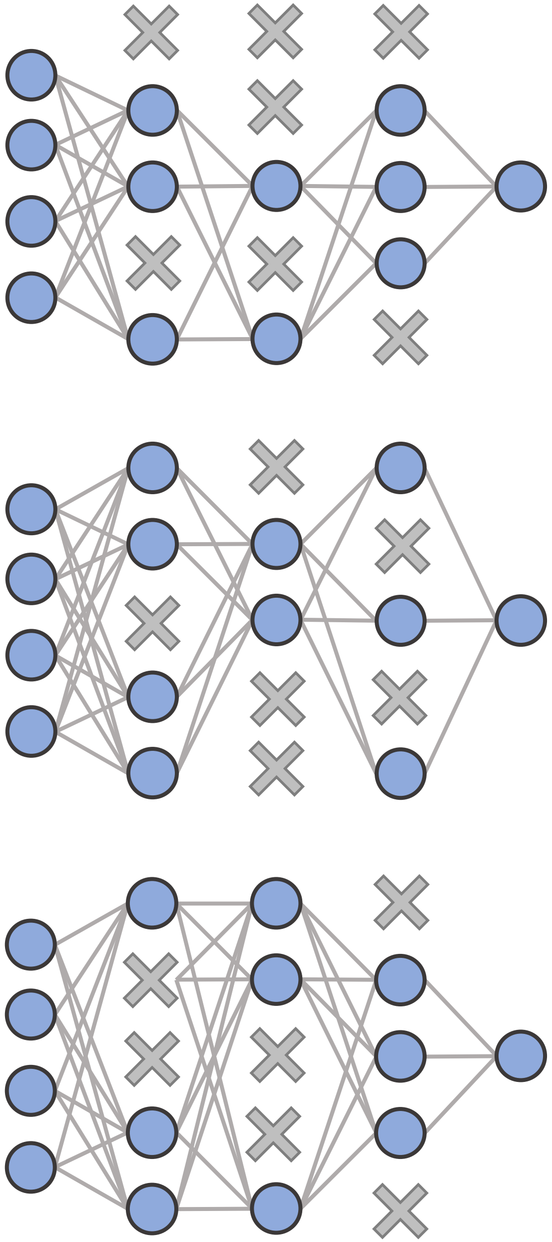
Effect of dropout for FCNs on MNIST





[3] Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, PMLR 2014.

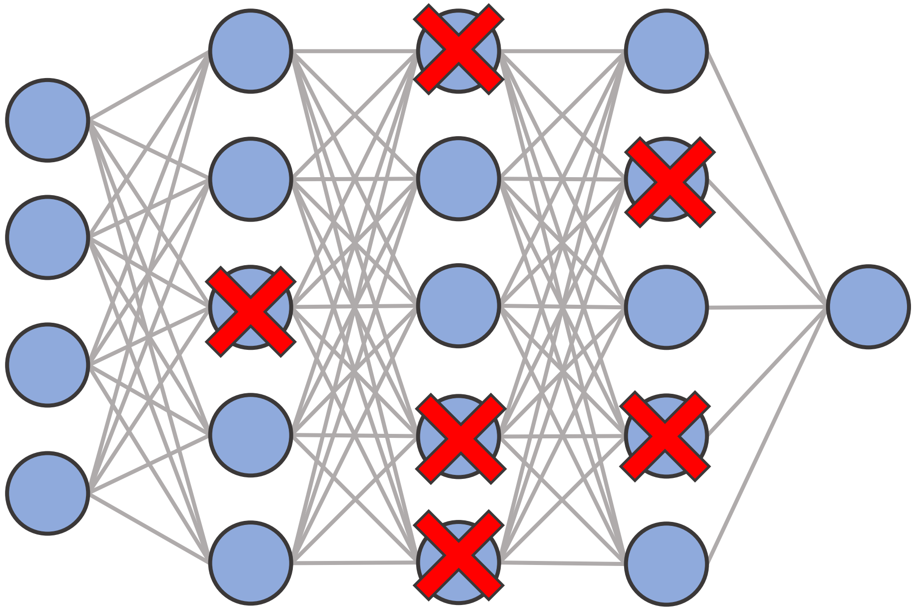
Dropout: Observations



- Size of sub-networks is controlled by keep probability *p*
- Penalizes co-dependence,
  promotes redundance and independence
- Different training iterations use different “sub-networks“
  → “Ensemble”-like behavior
- Biological correlation: redundant structures/synapses in the brain

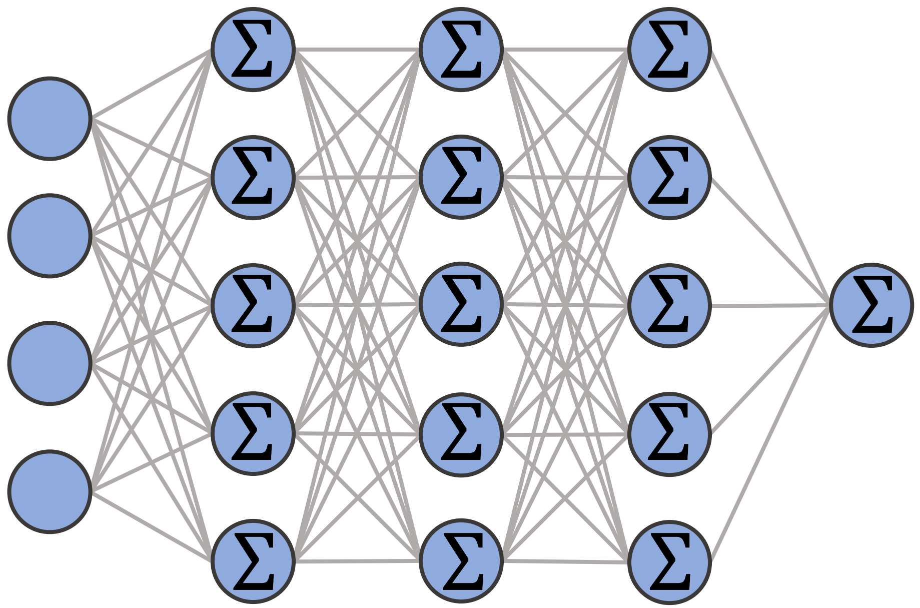
Dropout during training vs. during testing

During training



$p = 0.5$ $p = 0.5$ $p = 0.5$

During training

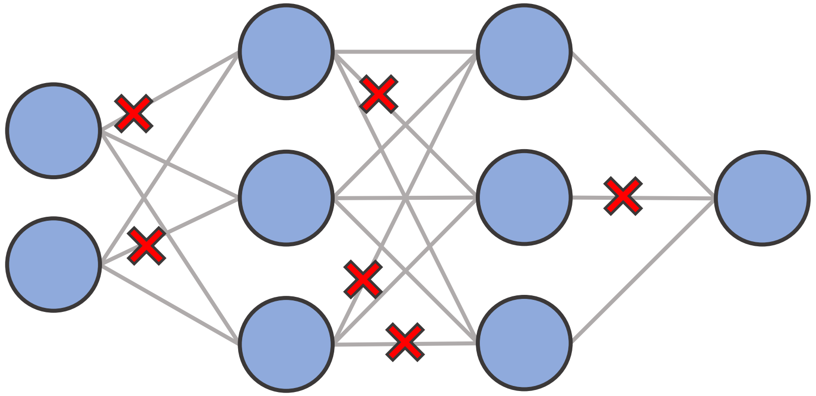


$p^* = 1.0$ $p^* = 1.0$ $p^* = 1.0$

→ Most frequent approach: **inverted dropout**

→ Multiply all “active” units with $1/p$

Dropout variants (examples)



- DropConnect [4]: Drop connections instead of nodes
- SpatialDropout [5] / DropBlock [6]:
  Variants for CNNs – drop feature maps or regions
- Gaussian Dropout[7]: Gaussian approximation
- …
- Monte Carlo Dropout [8]: Exploit ensemble-like behavior during test-time
  → Uncertainty / Bayesian deep learning

[4] Srivastava et al.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting, PMLR 2014.
[5] Tompson et al.: Efficient Object Localization Using Convolutional Networks, Proc. CVPR 2015
[6] Ghiasi et al.: DropBlock: A regularization method for convolutional networks, Proc. NeurIPS 2018
[7] Wang et al.: Fast dropout training, Proc. NeurIPS 2013
[8] Gal and Ghahramani: Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning, PMLR 2016

Summary and Outlook

- Regularization limits solution/model space by penalizing undesirable solutions
- Main goal: preventing overfitting & improving generalization
- Dropout drops nodes during training to prevent co-dependence
- “Inverted” dropout to achieve distribution match between train & test

- Dropout relates to & connects with
  - Model pruning
  - Early stopping
  - Norm constraints
  - Bayesian Deep Learning

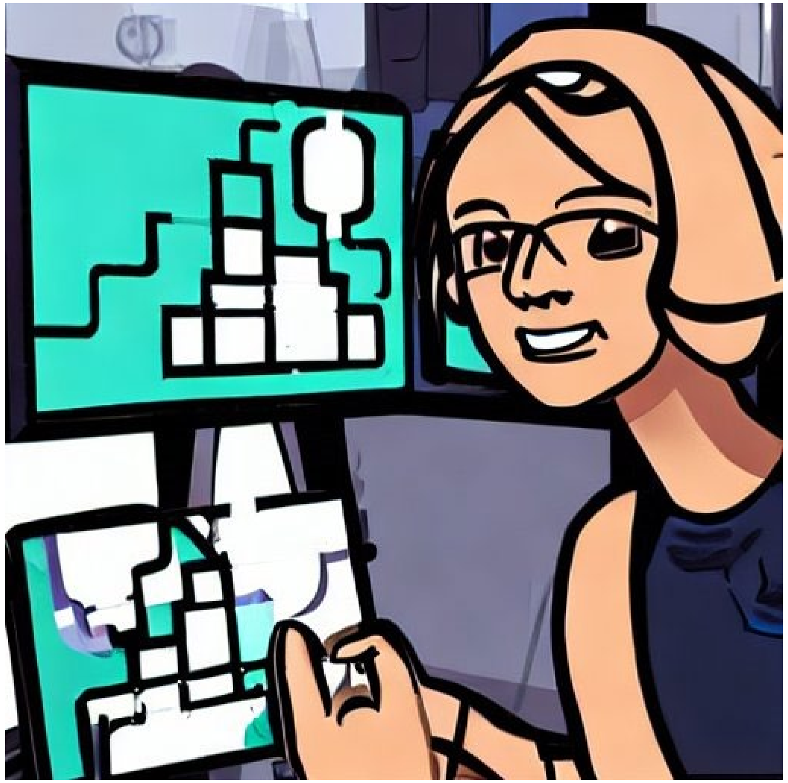
→Ongoing research in optimization

→Highly relevant also in current architectures (e.g., Transformers)

NEXT TIME on “Introduction to AI”: “Inductive bias and the rise of CNNs”

Exercise/Homework (2 weeks)

- Investigate **norm constraints** for different ML approaches
- Extend neural network framework with **dropout** for FCN and CNN layers
- Investigate learning behavior with and without dropout
- Visualize learned weights under different settings
- Investigate convergence for downstream tasks
    - Binary classification
    - Multi-class classification
    - With transfer learning



< a female AI researcher playing with an AI, in comic style >

Thank you!

Prof. Dr.-Ing. Katharina Breininger

katharina.breininger@fau.de

Lecture slides & materials