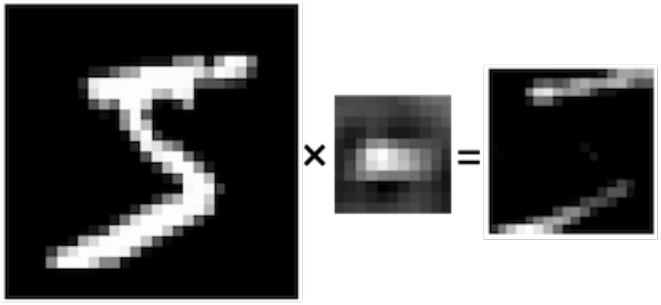
Universität Potsdam

Introduction to Deep Learning

Optimization



Andreas Krug, M.Sc.

ankrug@uni-potsdam.de

07. January 2019

Feedback evaluation

Aspects you liked:

- Reading material (with papers, blogs)
- In-class activities
- Inverted classroom
- Programming exercises
- Good choice and structure of the topics

Feedback evaluation

Aspects you disliked:

- Too much workload
- Too little/no discussion of programming in class
- Programming assignments too complex and/or vague
- the „right answer“ is not always given
  especially, if no answers pop up during group work

Feedback evaluation

Suggestions:

- stretch contents to IDL1 and IDL2
- Emphasize/point out the „correct answer“ after discussions
- Provide solutions for programming tasks
- More non-DL-book material
- More guidance in the beginning
  (with theory, in-class exercises, programming tasks)
- PyTorch (more intuitive than TensorFlow)

Feedback evaluation

Topics you like to revisit:

- RNNs, LSTMs
- Attention + Transformer
- Backpropagation
- Complex architectures

Feedback evaluation

Topic you like to be covered:

- Debugging code
- Hyperparameter tuning
- in-class project presentations for feedback
- Reinforcement Learning
- Autoencoder

Universität Potsdam

remaining semester schedule

date	topic	project
Jan 14	Auto-Encoder	
Jan 21	Introspection (brief overview by Andreas)	Short presentation for feedback
Jan 28	Revisiting topics	
Feb 4	Final course evaluation	final presentation
Feb 11 – Mar 31	oral exam (vote via doodle, enroll via moodle)	

Group work

Expert groups

Group work

- 4 sets of topics → 4 groups of experts
- Within each expert group (20'):
  - Prepare to explain concept behind each topic by extracting the key points together (feel free to use the book/internet)
- in mixed-expert groups (20' = 5' per expert):
  - Each in turn explains his/her topics briefly (3-4')
  - Others may ask questions and give feedback (concise information? Missing points?) (1-2')

Group work

- Example question:
  Is learning the same as optimization?
  - Not the same
  - Often the cost/loss function is optimized,
    this in turn maximizes a performance measure
    e.g. use negative log likelihood to optimize 0-1-loss
  - As we want a model to generalize well, we don't even optimize the cost function completely, e.g. in applying early stopping
  - Moreover, ML algorithms often don't optimize the complete cost function, as they use batches for gradient computation instead of the complete data set

Group work

question set 1 • Why is ill-conditioning challenging for optimization/learning? • What is RMSProp? • How does batch normalization work?	**question set 2** • Why are flat regions (saddle points, plateaus) challenging for optimization/learning? • What is AdaGrad? • How does (greedy) supervised pre-training work?
question set 3 • Why are local minima challenging for optimization/learning? • What is Nesterov Momentum and how is it different to standard Momentum? • How does (block) coordinate descent work?	**question set 4** • Why are cliffs and long-term dependencies challenging for optimization/learning? • What is Stochastic Gradient Descent with Momentum? • How does curriculum learning work?

Assignments

- Responsible for recap: Sonu & Reza
- Reading on Auto-Encoders
- (no programming task)
- Time for your project

Slides & assignments on: https://mlcogup.github.io/idl_ws18/schedule