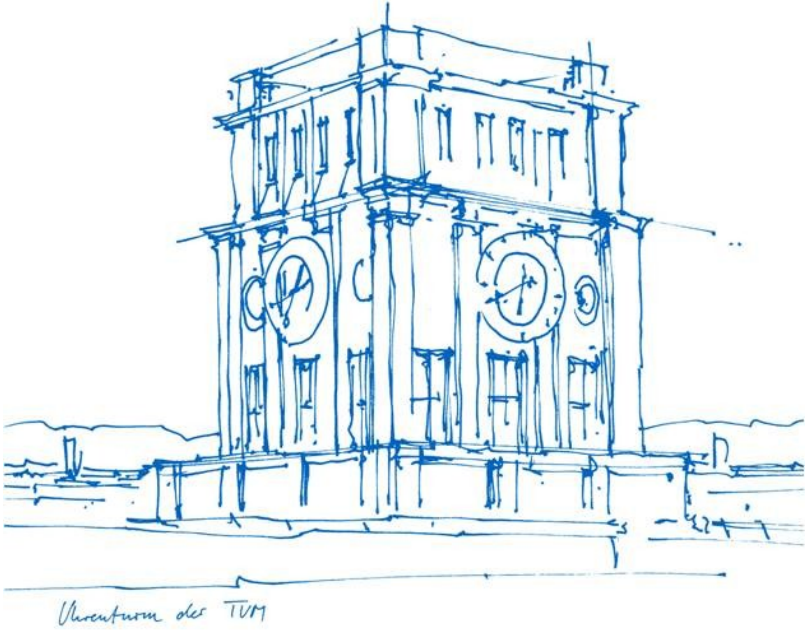
Beyond Deep Learning: Uncertainty Aware Models

Christian Tomani, Yuesong Shen

Technical University of Munich

Chair of Computer Vision and Artificial Intelligence

Garching, Feb. 4th, 2020



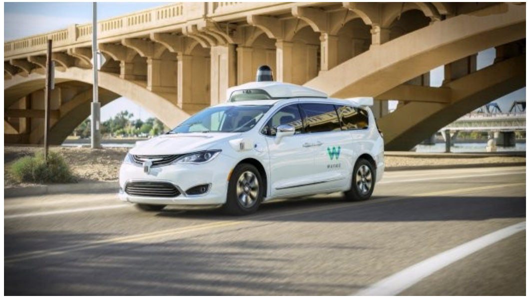
Chair of Computer Vision & Artificial Intelligence
Department of Informatics
Technical University of Munich

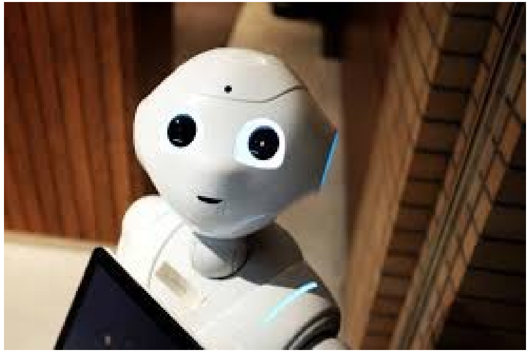
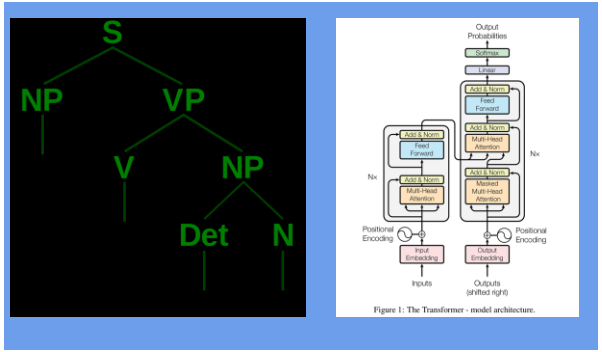
Agenda

- Why do we need uncertainty aware models?
- What is our focus for this course?
- What are the topics we will cover?
- How is the course organized?
- How to apply?

Chair of Computer Vision & Artificial Intelligence
Department of Informatics
Technical University of Munich

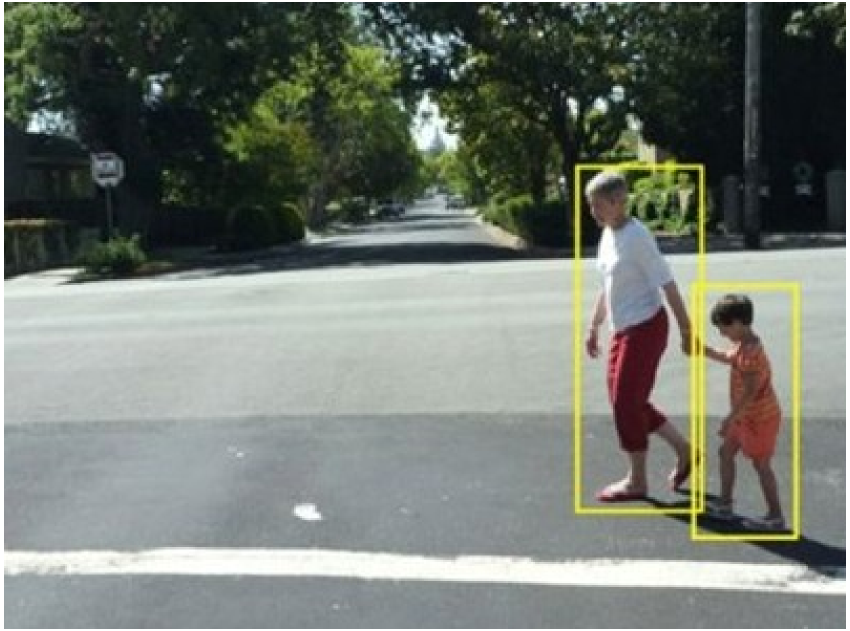
Current applications for Deep Learning



Vaswani et al. - Attention Is All You Need, 2017 https://www.heise.de/newsticker/meldung/Autonomes-Fahren-Waymo-Roboterautos-bald-bei-Lyft-Milliarde-fuer-GM-44170html91., 03/01/2020
https://www.nytimes.com/interactive/2018/08/17/technology/alexa-siri-conversation.html, 03/01/2020

Safety critical applications

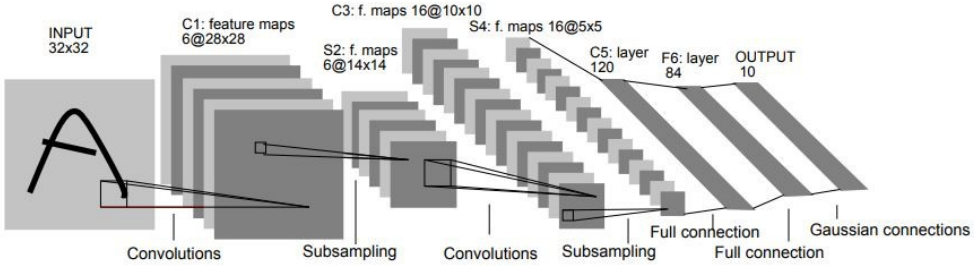


Chair of Computer Vision & Artificial Intelligence
Department of Informatics
Technical University of Munich

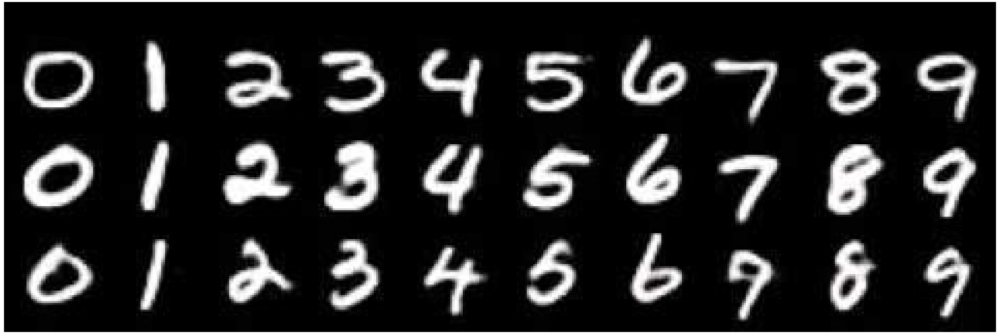
The issue with Deep Learning - Can we trust the model?

Setup

LeNet-5 Model with weight decay

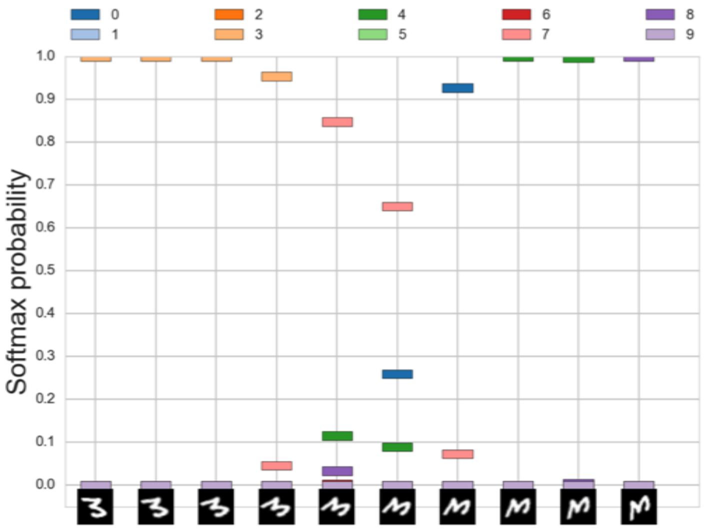


MNIST Dataset



LeCun et al. - Gradient Based Learning Applied to Document Recognition, 1998
https://github.com/cazala/mnist

The issue with Deep Learning - Can we trust the model?



Vanilla LeNet-5 Model on MNIST

- Model is unreliable and not calibrated
- Gives totally wrong but highly confident predictions if data is perturbed
- wrong predictions cannot be distinguished from correct ones

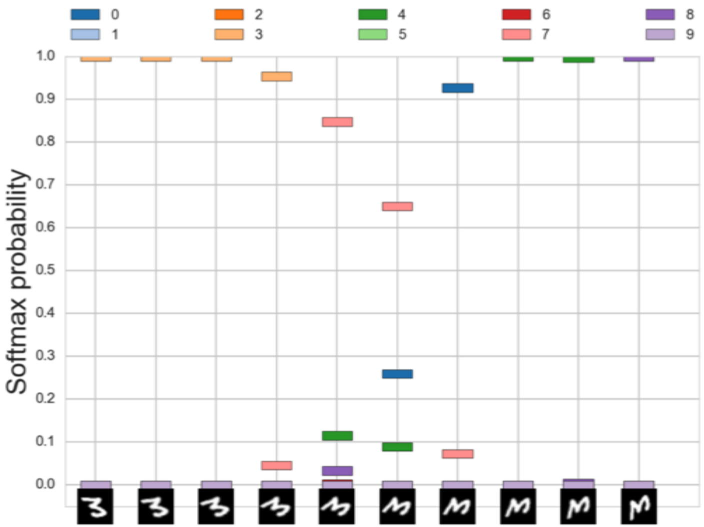
Softmax Probability

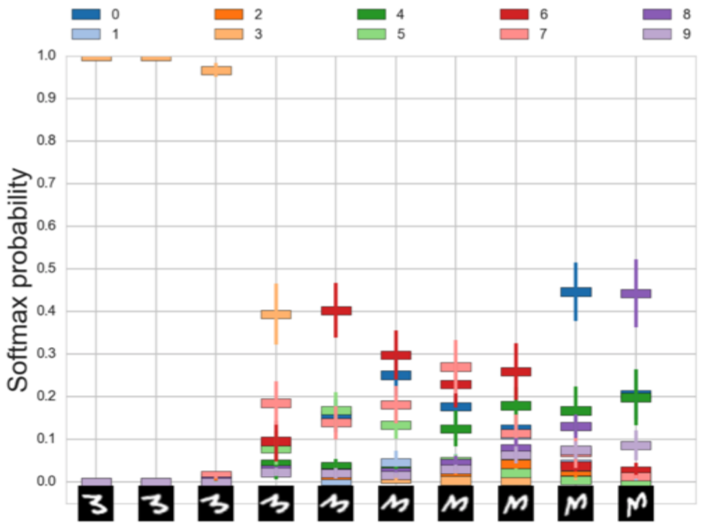
≠

Confidence

Chair of Computer Vision & Artificial Intelligence
Department of Informatics
Technical University of Munich

The issue with Deep Learning - Can we trust the model?





Christos Louizos - Multiplicative Normalizing Flows for Variational Bayesian Neural Networks, 2017

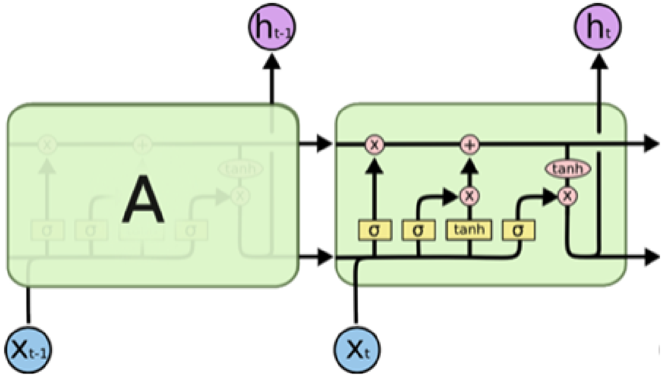
What is our focus for this course?

- **Work with different uncertainty aware methods**
- **Develop models that are uncertainty aware**
- **Apply uncertainty aware models to real world data**

Trustworthy and uncertainty aware methods as a solution to the problem of unreliable models

- Temperature-Scaling
- Deep Ensemble Models
- Monte-Carlo Dropout
- Stochastic Variational Inference Networks
- Multiplicative Normalizing Flow Networks
- Evidential Deep Learning Models - Dirichlet Distribution

Long-Short-Term-Memory N



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Beyond NN: scaling up probabilistic models

Holy grail: scalable and uncertainty-aware model

- NN (scalable) + uncertainty-awareness: Bayesian DL, deep ensemble …

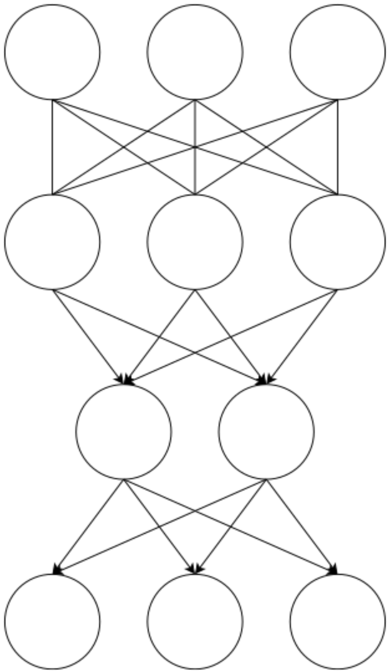
  → c.f. previous slides

- probabilistic models (uncertainty-aware) + scalability ?

  Alternative approach to our goal

  →examples: next slides

Hierarchical graphical models



- Graphical model is a well studied family of ML models
  - + Flexible probabilistic representation
  - + Unsupervised learning, generative modeling
  - - Hard inference and learning (approximative methods available)
  - → source of inspiration for many BDL methods

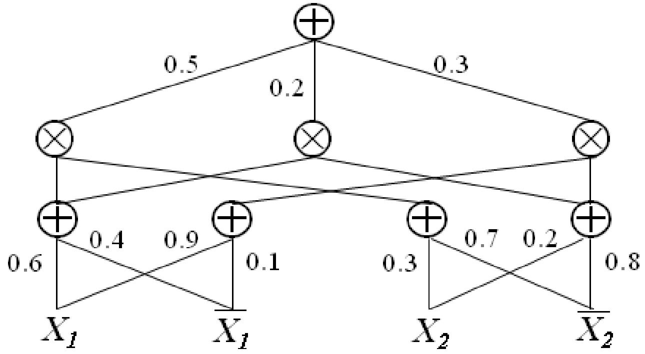
- Hierarchical graphical models

  Deep belief network, deep Boltzmann machine, sigmoid belief network ...

- Special training techniques:

  contrastive divergence, annealed importance sampling, wake-sleep algorithm ...

Sum-product network



Hierarchical mixture (sum) layers and product layers.

- Fully probabilistic
- Efficient inference (feed-forward)
- Great for generative modeling

Curated list of SPN related materials:

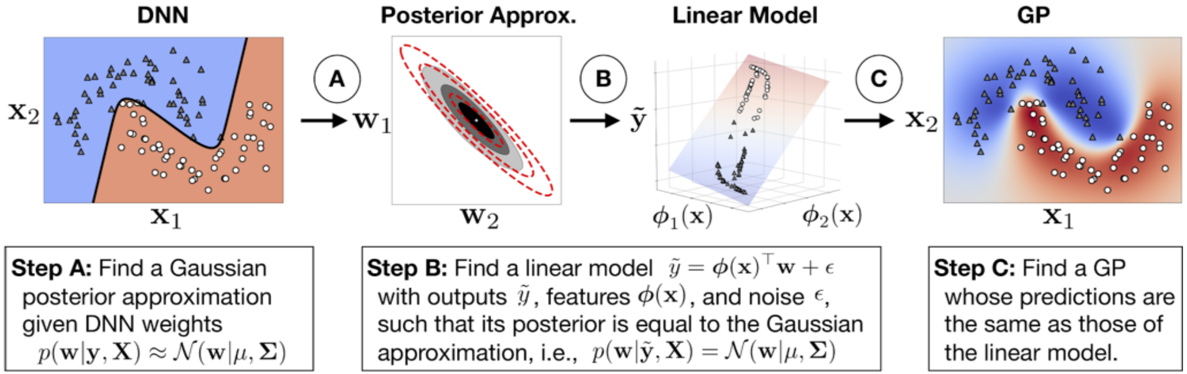
https://github.com/arranger1044/awesome-spn

Original Image published in:
"Sum-Product Networks: A New Deep Architecture, Poon and Domingos, 2011"

Gaussian process view of NN

NN with infinite width can be interpreted as a gaussian process.

- Non-linearity ⇒ Kernel function
- Some related works based on this interpretation: Neural tangent kernel, MC-dropout, ...



Original Image from https://github.com/team-approx-bayes/dnn2gp
“Approximate Inference Turns Deep Networks into Gaussian Processes, Khan et al., 2019”

Topics that we will cover in this course

Models:

- Recurrent neural networks for time series prediction
- Models with differentiable memory
- Convolutional neural networks

Datasets:

- Medical Datasets
- Bioinformatics
- Natural Language Processing
- Computer Vision datasets

Methods for inducing uncertainty awareness:

- MC-Dropout
- Temperature Scaling
- Deep Ensemble Models
- Evidential Deep Learning Models - Dirichlet Distribution

Non-NN uncertainty-aware models:

- deep GP, NTK
- Hierarchical GM (SBN, DBN, DBM)
- SPN

Chair of Computer Vision & Artificial Intelligence
Department of Informatics
Technical University of Munich

Course Organization

Course website: https://vision.in.tum.de/teaching/ss2020/bdluam_ss2020

Course email: bdluam-ss20@vision.in.tum.de

- Course structure:
  - Approx. 2 student teams
  - Weekly progress report & (if needed) weekly team meeting

- 4 joint meetings along the semester:
  - Kickoff meeting: project intro & matching (April 22nd)
  - 2 progress meetings: project progress check points & exchange ideas (May 27th & June 24th)
  - Final presentations: final project delivery and presentation (July 22nd)

Prerequisites

- Programming skills:

  Python, tensor ops. (numpy), DL framework (PyTorch / TensorFlow / ...)

- Machine learning & deep learning knowledge:

  Basic ML concepts and ML/DL models

  Min. Requirement: passed one ML/DL related course (I2ML, I2DL, ADL4CV, PGM ...)

- Soft skills:

  Manage regular workflow, collaborate with teammates and communicate with tutors efficiently

- We also value:
  - prior experience with ML/DL projects
  - solid basis & interest for maths

How to apply

1. Apply via the **TUM Matching system** (Feb. 7th - 12th, 2020)
    - If you like our course, make sure to give it a high priority :)
2. **Send us an email** to show your interest and fulfillment of prerequisites
    - Crucial for us to give you a priority

- The email should be sent to bdluam-ss20@vision.in.tum.de **latest Feb 11th** with the title "[Application] <Firstname> <Lastname>" and contain
    - Filled information form (template on course website, rename to "firstname_lastname.xlsx")
    - Transcript
    - CV

Thank you! Questions?