Machine Intelligence:: Deep Learning
Week 7

Beate Sick, Pascal Bühler, Oliver Dürr

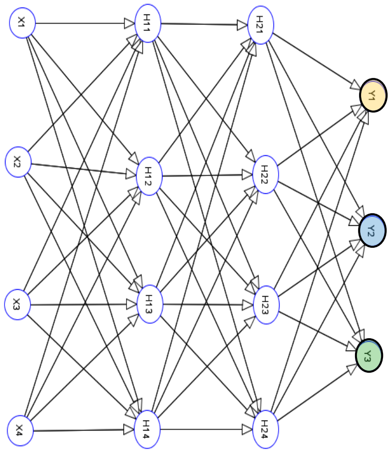
Ensembling approaches for improving the performance and uncertainty estimates of NN models by taking into account the algorithmic epistemic uncertainty.

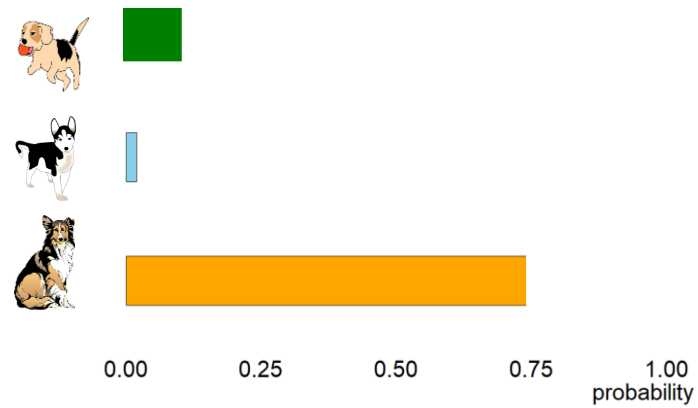
Outline:

- Uncertainty in DL models
  - Epistemic uncertainty
  - Algorithmic uncertainty
  - Aleatoric uncertainty

- Approaches to take algorithmic and/or algorithmic & epistemic uncertainty into account:
  - Deep Ensembling
  - MC Dropout
  - Bayes
    - Theoretical Background for all
    - Variational Inference (possible next week)

Probabilistic CNNs as we know them so far

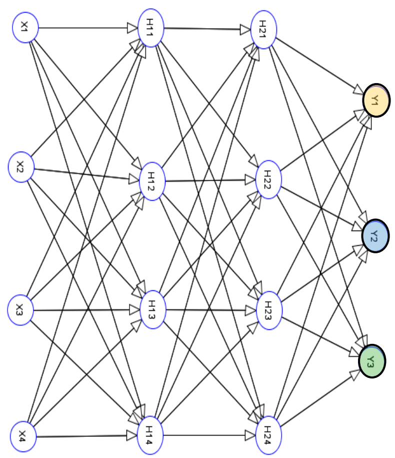




CNNs yield high accuracy and calibrated (=unbiased) probabilities, but…

How good do we know probabilistic CNNs?

What happens if we present a novel class to the CNN?

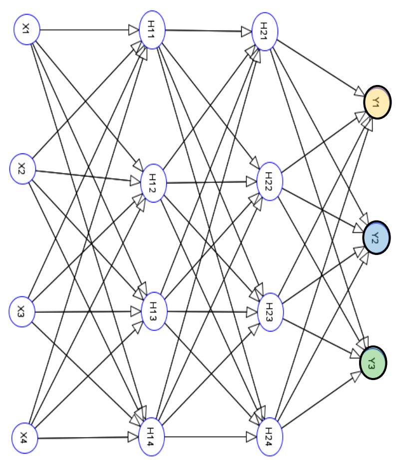


You might expect:

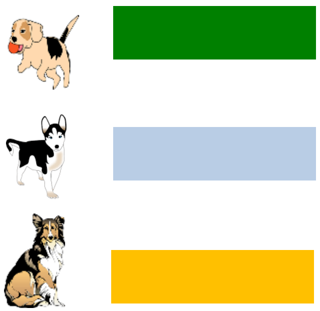
What do you expect?

How good do we know probabilistic CNNs?

What happens if we present a novel class to the CNN?

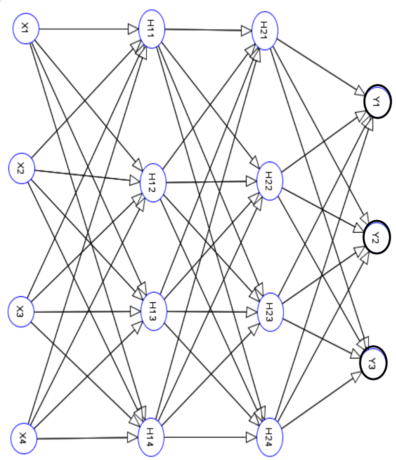


You might expect:

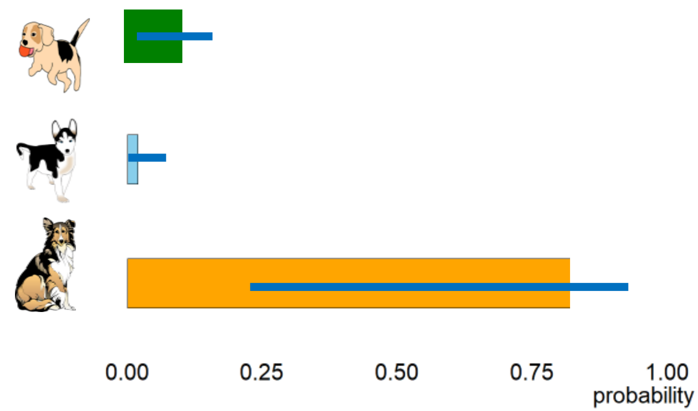


A non-Bayesian NN cannot ring the alarm

What happens if we present a novel class to the CNN?



Plain wrong !



We need some error bars!

Importance to detect OOD

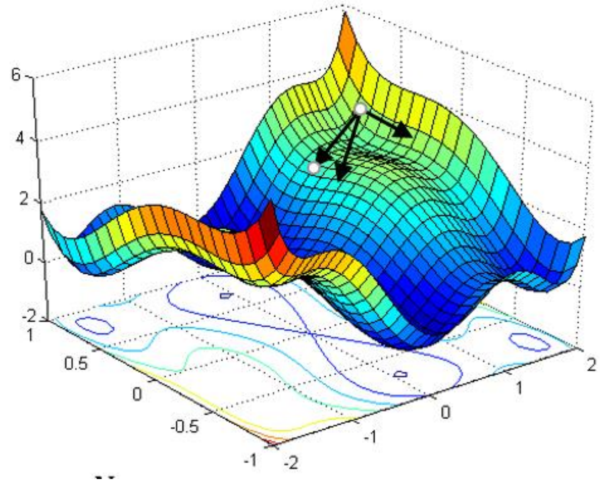
- Current DL Systems bad in out of distribution OOD situations
- Application need at least to detect OOD situations

Foto: https://www.crn.de/software-services/halloween-training-fuer-google-autos.108409.html

Algorithmic uncertainty in Deep Learning

If we train the same NN model twice on the same data, we get two (slightly) different trained models due to algorithmic uncertainty, i.e.

- Because we initialize the weights randomly before starting the training
- Becuase we split the train data randomly in mini-batches and the determined gradients $\partial L/\partial w_i$ dependi n the mini-batch on which they are determined. After each epoch a new split in mini-batches is done.
- Because we often work with data augmentation methods, where the augmented samples differe randomly from the a sample in the mini-batch



Extrapolation: Causes Problems

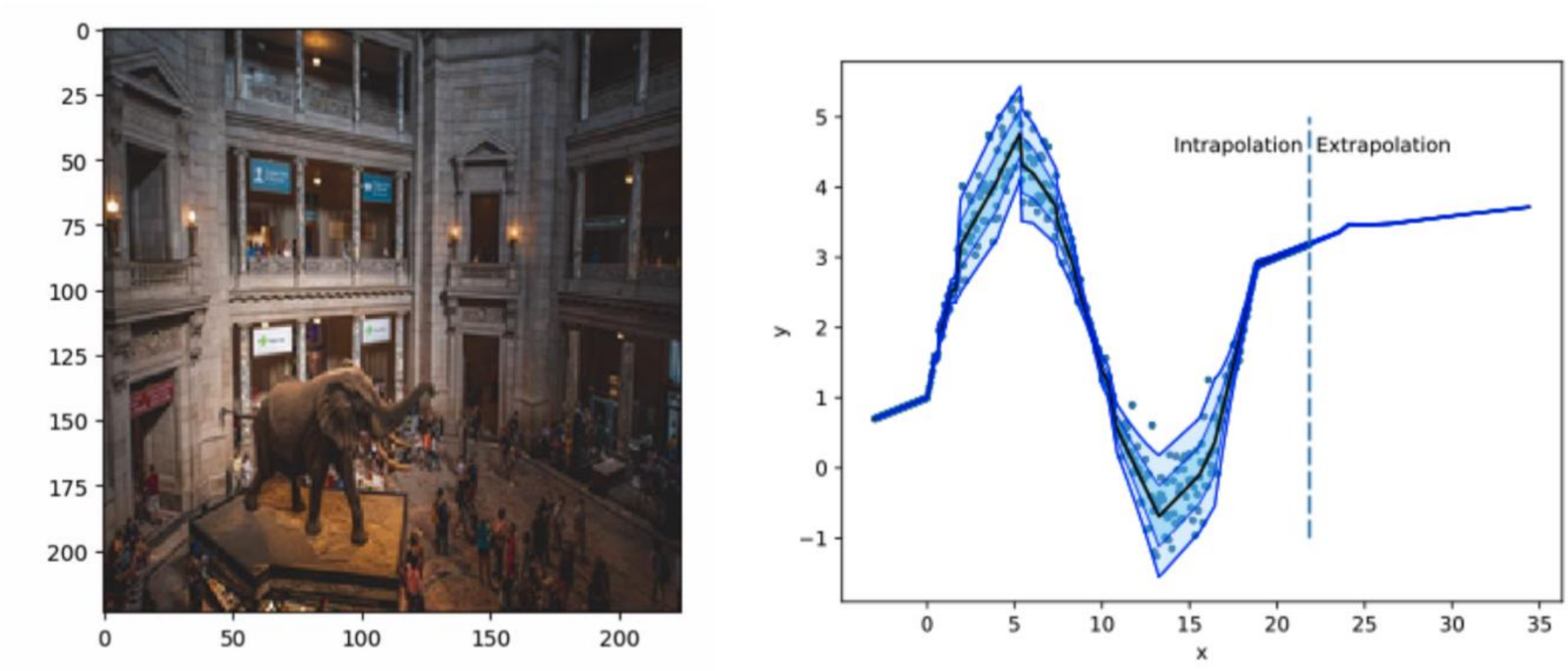
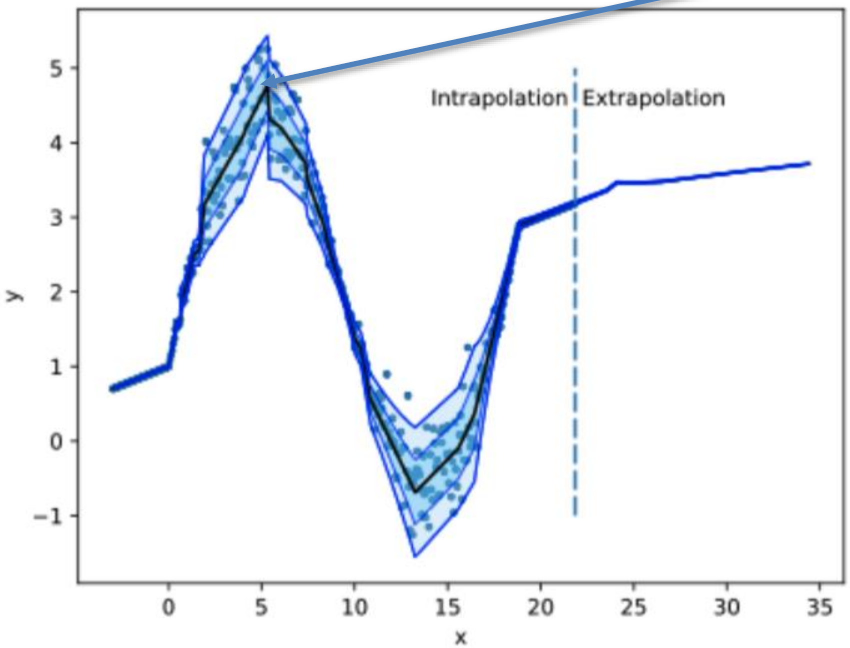
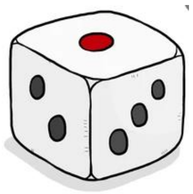


Figure 7.2 Bad case of DL. The high performant VGG16-CNN trained on ImageNet data fails to see the elephant in the room. The five highest-ranked class predictions of the objects in the image are horse_cart, shopping_cart, palace, streetcar, gondola; the elephant is not found! This image is an extrapolation of the training set. In the regression problem on the right side of the dashed vertical line, there's zero uncertainty in the regions where there's no data (extrapolation).

Definition Aleatroic vs. Epistemic Uncertainty

Much spread in data, aleatroic (from “Alea Acta est”))



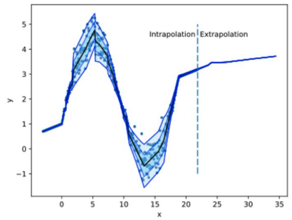


- *Aleatoric* uncertainty is due to the uncertainty in the data.
- The uncertainty when leaving the ‘known ground’ is called *epistemic* uncertainty.

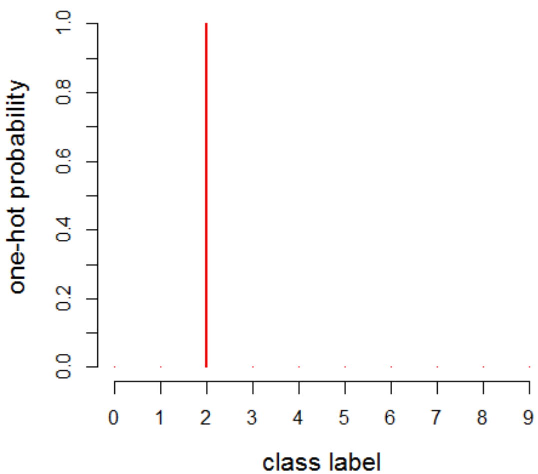
We can model this uncertainty when we take the uncertainty with which we know the weights (called parameter uncertainty) into account. This can be done with Bayesian reasoning or phenomenological.

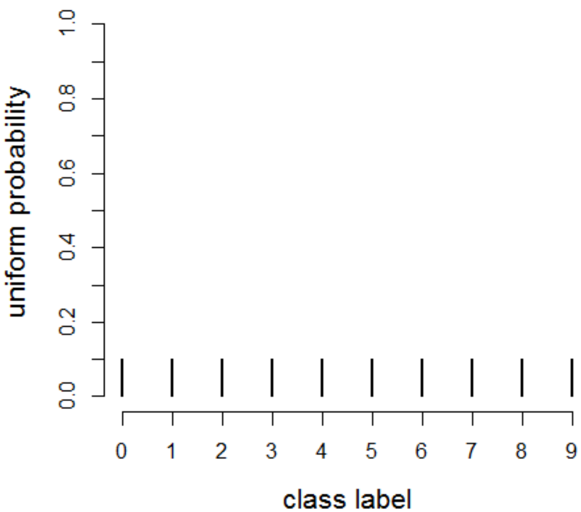
Aleatoric Unctertainty

- Regression
  - The spread of the Data.



- Classification
  - Spread?





$$H(P) = -\sum_i p_i \cdot \log(p_i)$$

One has H=~2.3 and on H=0. Which one

How to model the epistemic and/or algorithmic uncertainty?

Deep Ensembling

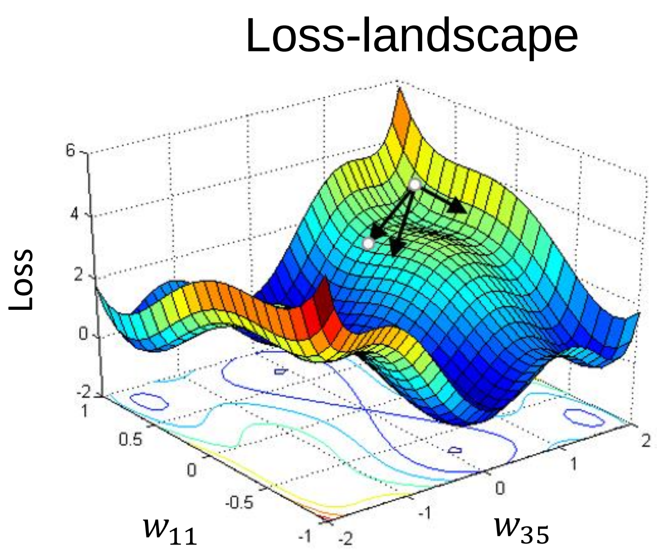
Original paper by Lakshminarayanan et al.: https://arxiv.org/pdf/1612.01474.pdf

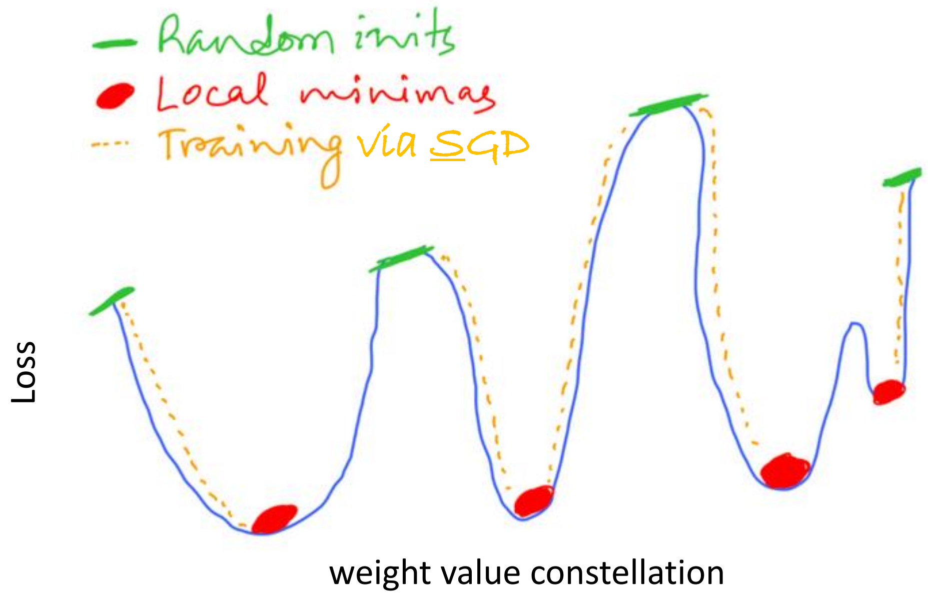
Basic Idea of deep ensembling

- Use an ensemble of n-different models and see if they agree
- How to get different ensembles?
- Traditional Statics?
  - Bagging Boot Strapping and Aggregating
- Deep Learning
  - Deep Learning just average over different solutions of gradient descent

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, *30*.

The loss-landscape in DL is usually not convex



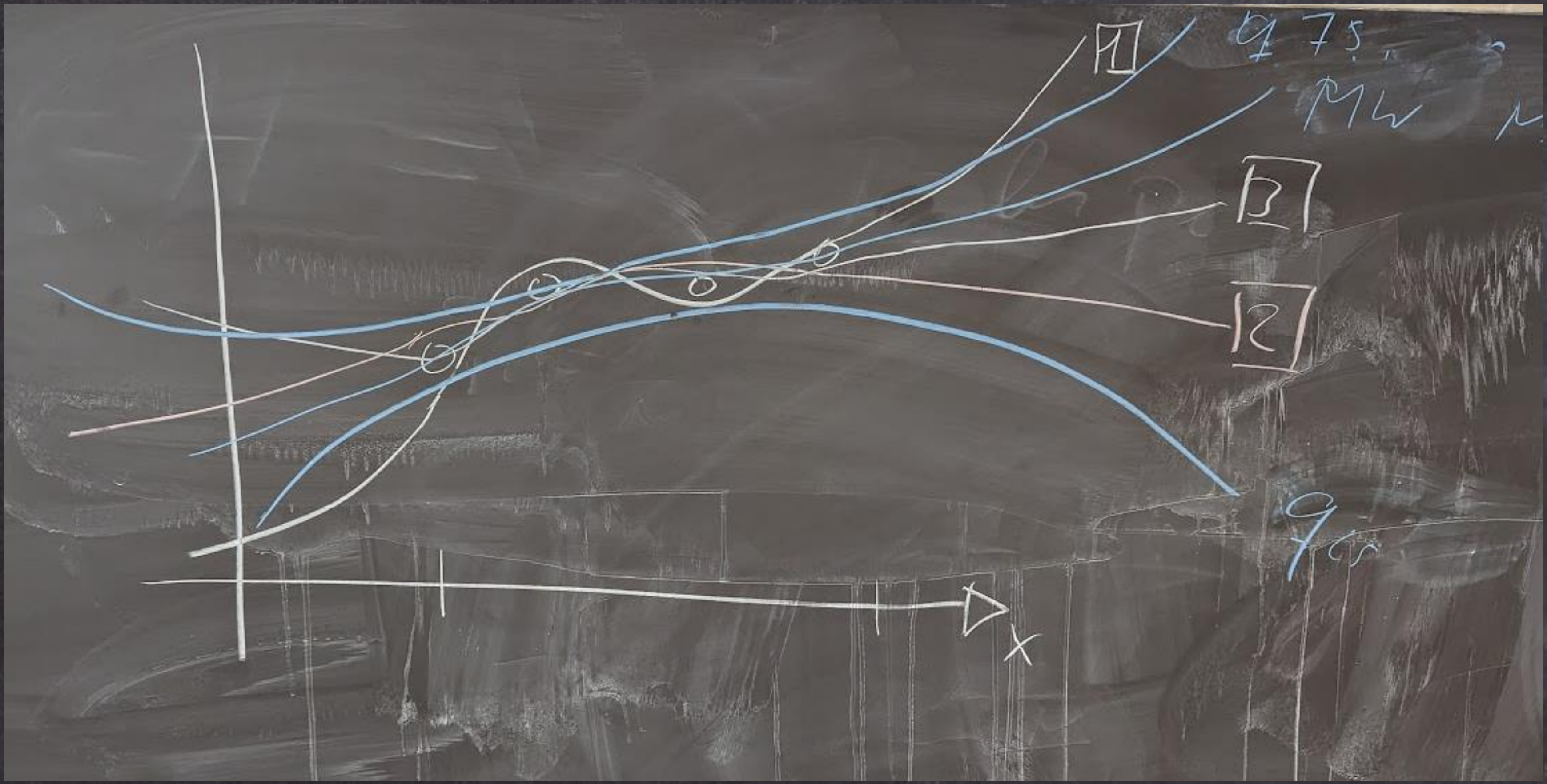


The loss-landscape of DL models has many local minima with similar depth.

Training is started with a **random** weight value initialization → training the NN with **S**GD and the same data several times is usually ending in different local minima.

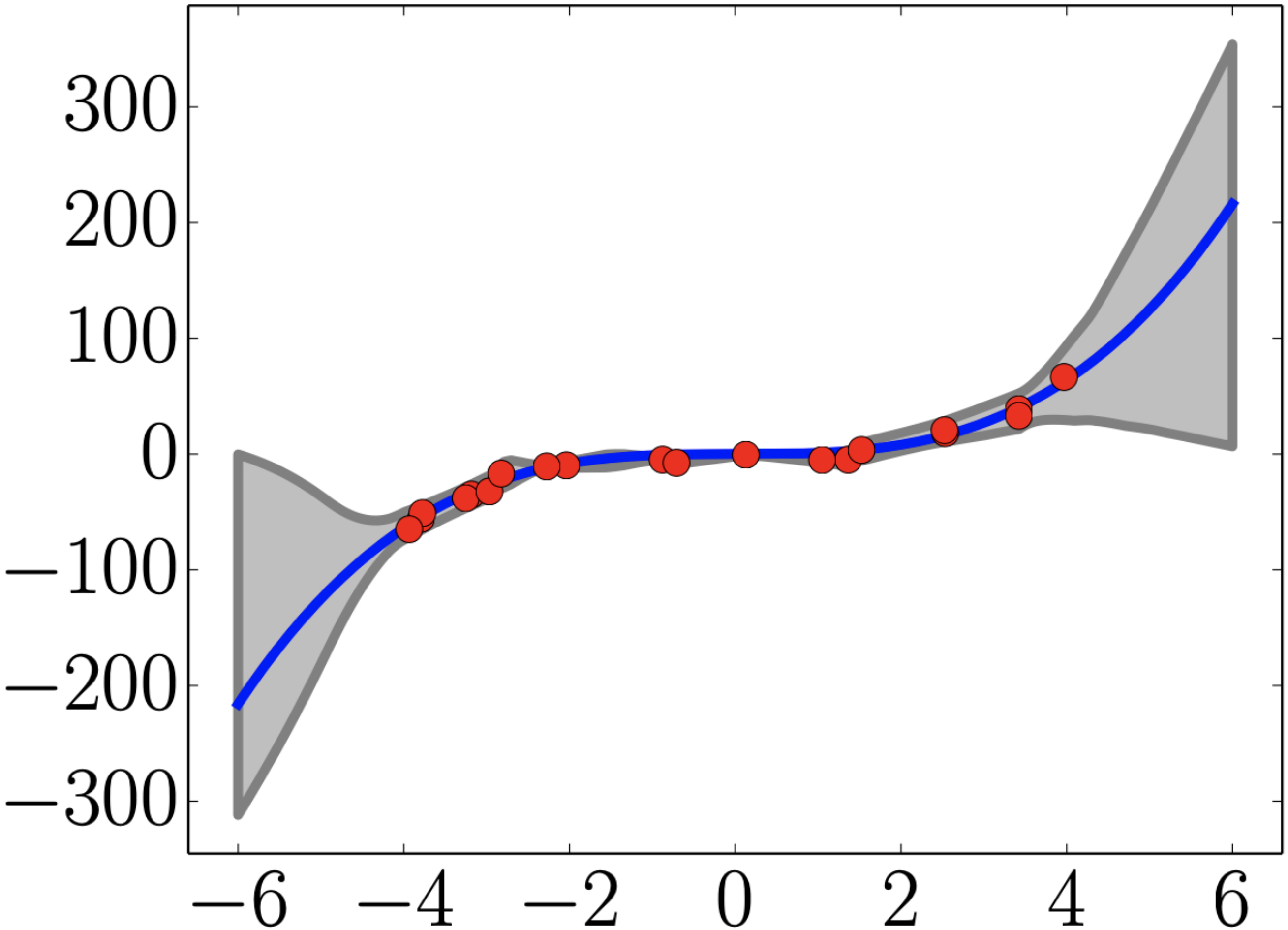
Image credit: https://wandb.ai/authors/loss-landscape/reports/Understanding-the-Effectivity-of-Ensembles-in-Deep-Learning--VmlldzoxODAxNjA

Blackboard Regression Ensembling with 3 solutions



In the data points, solutions are similar.
Outside the data there are difference

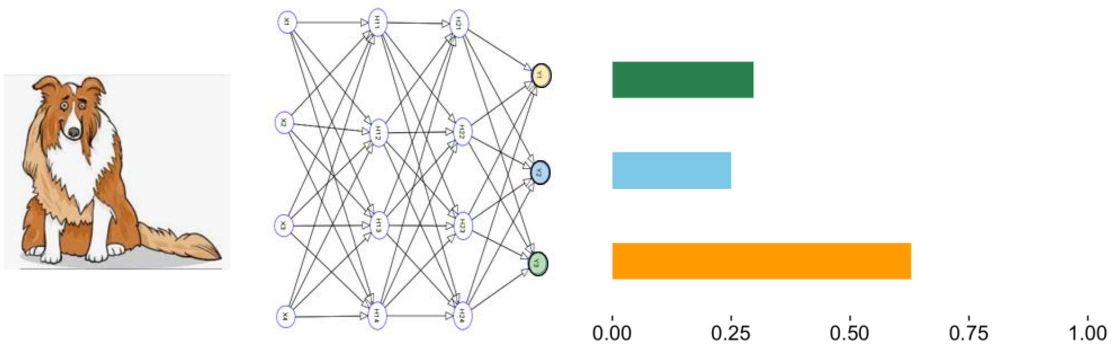
Ensemble of 5 Networks (Regression)

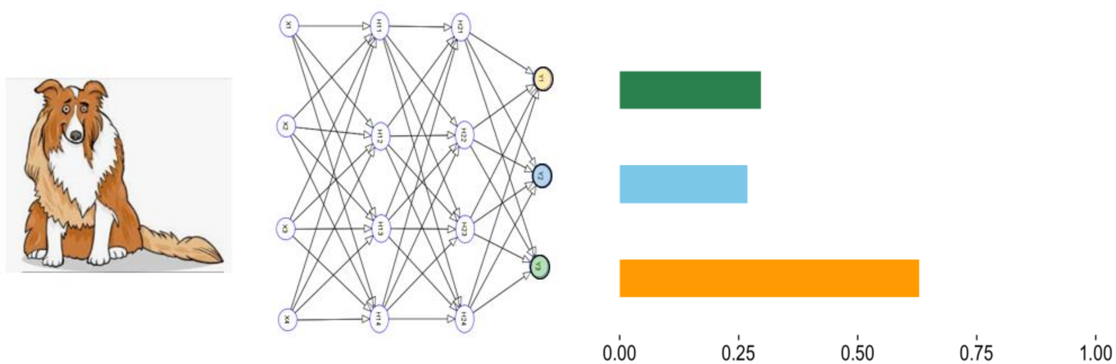


Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, *30*.

Classification

What happens if we haved trained the same CNN twice with the same data? Present example in training set.

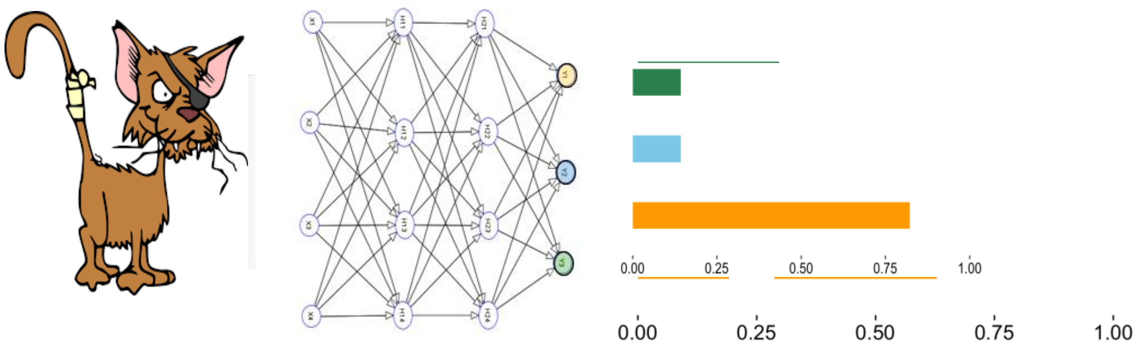


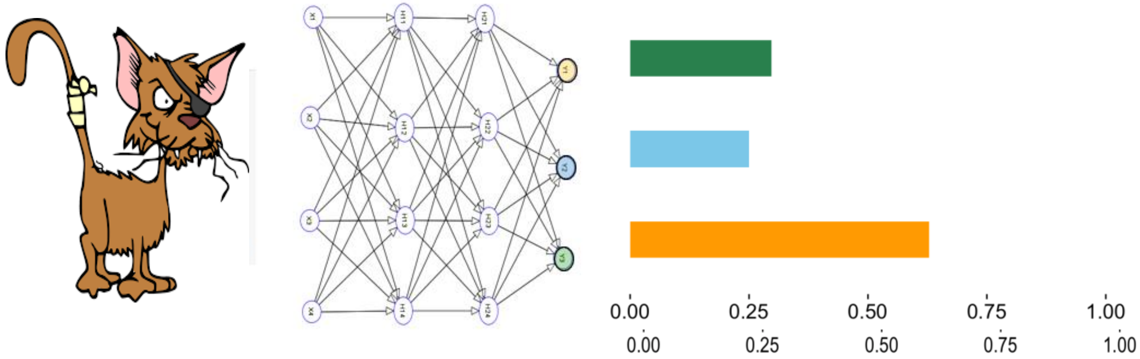


Small difference if example is know

Classification

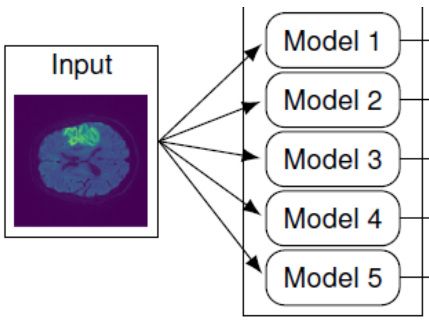
What happens if we haved trained the same CNN twice with the same data? But present OOD example.





Larger difference if not (cat).

Deep ensembling:
Train several NN models and average their predicitions



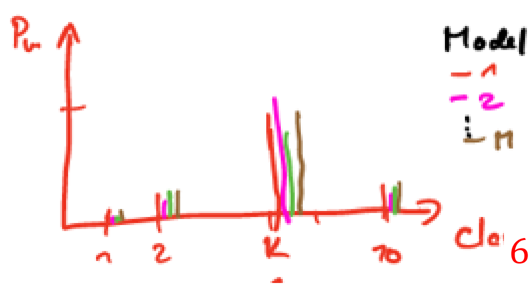
Each model predictis probabilities for the 6 classes

p_1	p_2	p_3	p_4	p_5	p_6
0.1	0.1	0.2	0.4	0.1	0.1
0.05	0.05	0.2	0.3	0.2	0.2
0.2	0.2	0.1	0.05	0.05	0.4
0.05	0.05	0.05	0.05	0.2	0.6
0.3	0.25	0.25	0.1	0.1	0

We get the ensemble predictions by averaging the probabilities that were predicted by the different models for each class

p_1^E	p_2^E	p_3^E	p_4^E	p_5^E	p_6^E
0.14	0.13	0.16	0.18	0.13	0.26

We can get the aleatoric ensemble uncertainties by computing the entropy of the ensemble prediction $H(P^E) = -\sum_i p_i^E \cdot \log(p_i^E)$.



We can get the epistemic ensemble uncertainties by e.g. computing the standard deviation of the probabilities that were predicted by the different models for each class

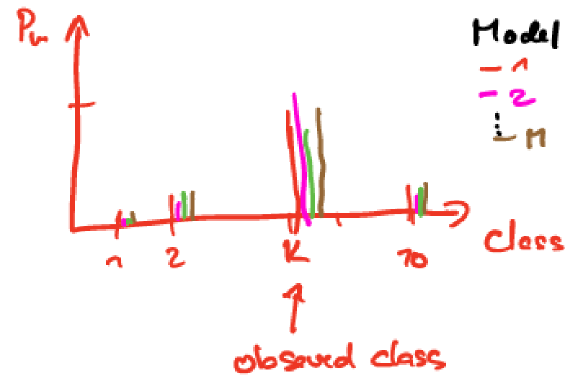
$sd_1^E \quad sd_2^E \quad sd_3^E \quad sd_4^E \quad sd_5^E \quad sd_6^E$

Nice:
For the convex NLL loss, it is guaranteed, that the NLL of the ensemble prediction is better (smaller or equal) than the average NLL of the individual models.

Original paper by Lakshminarayanan et al.: https://arxiv.org/pdf/1612.01474.pdf

Ensembling improves the NLL performance

Ensemble prediction for an observation with observed class = k, based on M models:



$$p_k^E = \frac{1}{M}\sum_{m=1}^{M} p_k \qquad k = 1\ldots10 \text{ for 10 classes}$$

associated NLL contribution ℓ:

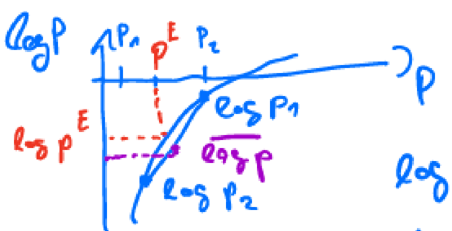
Model m: $\ell_m = -\log p_{km}$; Ensemble: $\ell^E = -\log p_k^E$

to show $\bar{\ell} \geq \ell^E$ with $\bar{\ell} = \frac{1}{M}\sum_{m=1}^{M} \ell_m = \frac{1}{M}\sum_m -\log p_{km}$

$$= -\underbrace{\frac{1}{M}\sum_m \log p_{km}}_{\leq \log \frac{1}{M}\sum_m p_{km}}$$

$$\geq -\log p_k^E = \ell^E \quad \square$$

use Jensen inequality



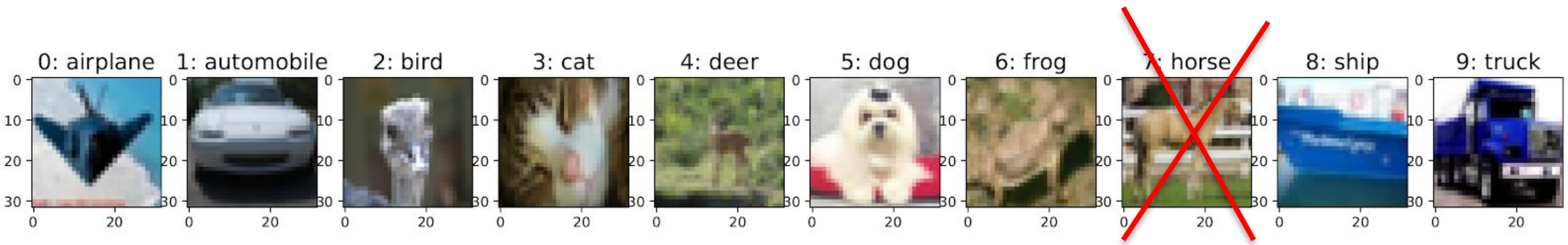
$$\log \bar{p} \geq \overline{\log p}$$

$$\log \frac{1}{M}\sum_m p_m \geq \frac{1}{M}\sum_m \log p_m$$

Hands-on Time

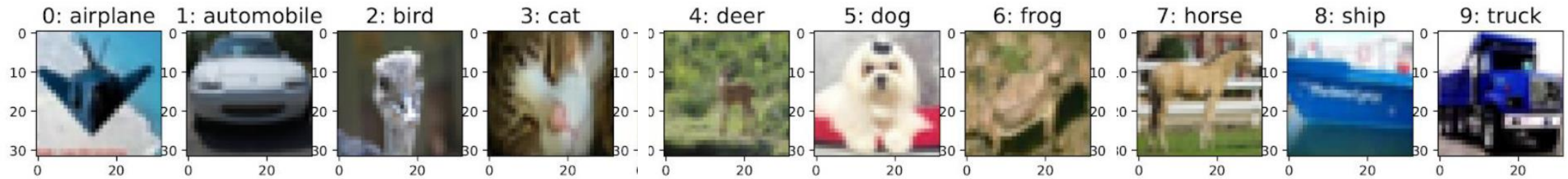
[Notebook 17](#)

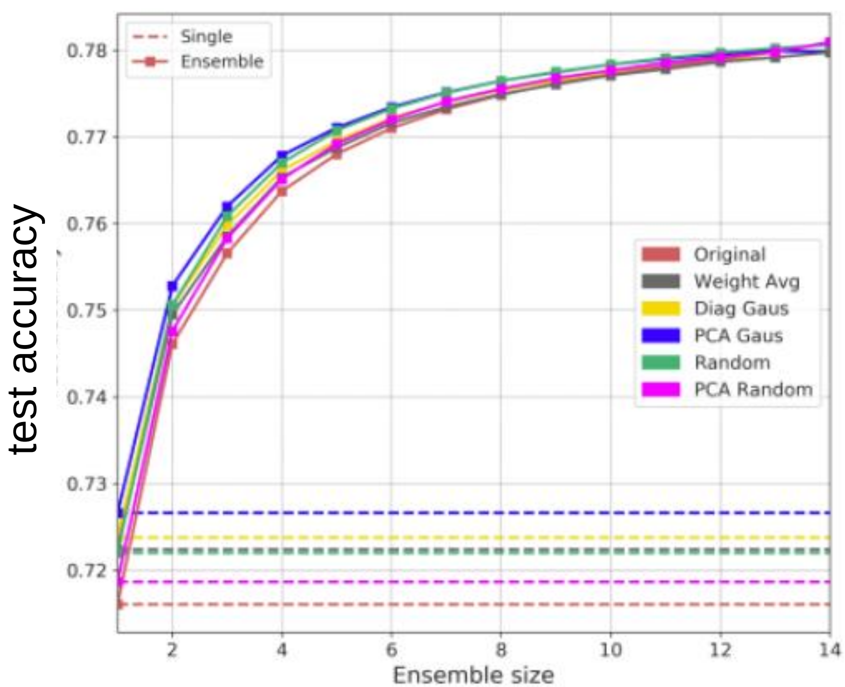
Train a CNN with only 9 of the 10 classes and investigate if the uncertainties are different when predicting images from known or unknown classes.

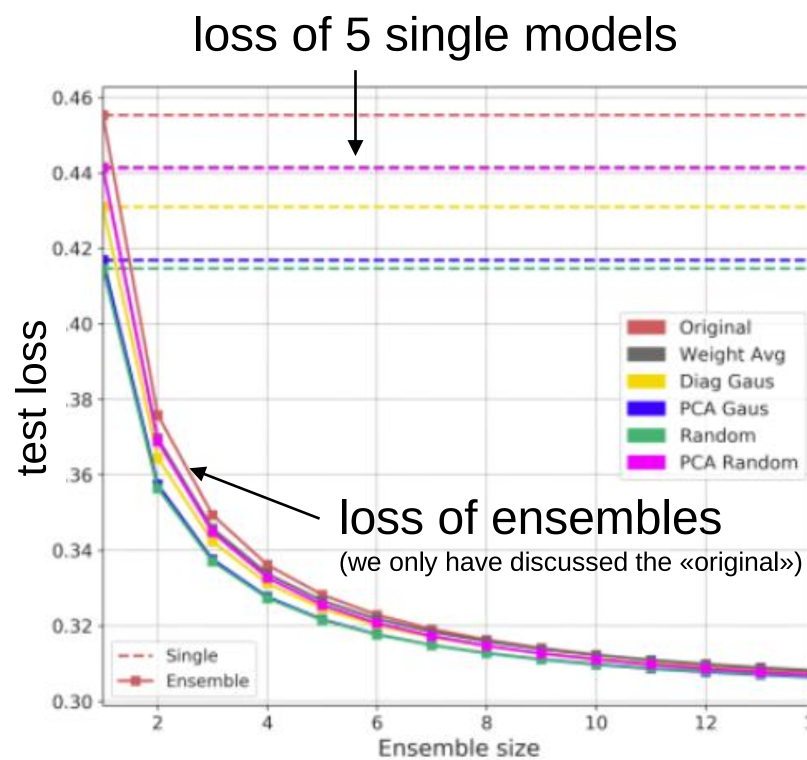


Work through [NB17](#) until and including Deep Ensembling.

Deep ensembling improves prediction power







Ensembes with as few as 3 or 5 members are typically enough to achieve a perfromance gain.

Image credit: https://wandb.ai/authors/loss-landscape/reports/Understanding-the-Effectivity-of-Ensembles-in-Deep-Learning--VmlldzoxODAxNjA

Take home messagte

- We have different uncertainty components when working with NN
  - Model choice uncertainty
    - which model/architecture should we use?
  - Algorithmic uncertainty
    - Training twice the same NN-architecture with the same data does not yield the same trained model
    - Random intialization, random mini-batch splits, random augmentation
  - Aleatoric uncertainty = data inherent variablilty
    - We capture aleatoric uncertainty by the spread of the predicted conditional probability distribution, e.g. Variance/Entropy for numeric/categorial data
  - Epistemic uncertainty
    - The lack of knowledge due to a lack of information, such as too few data or a lack of understanding leading to the wrong model choice

- Deep Ensembling is always good to get a better model
    - Better prediction performance: The NLL of an ensemble is better or equal than the mean NLL of the members
    - We can quantify the algorithmic uncertainty of the model