Where is the uncertainty in Neural Networks?

Moritz Kniebel

Universität Tübingen

June 26, 2020

Why being Bayesian?

Why being Bayesian?/Problem

- Neural Networks output point estimates.
- Uncertainties of the outputs are unknown.
- Bad for safety-critical applications (e.g. self-driving cars).

$\Rightarrow$ Need for posterior distribution over weights.

Solution

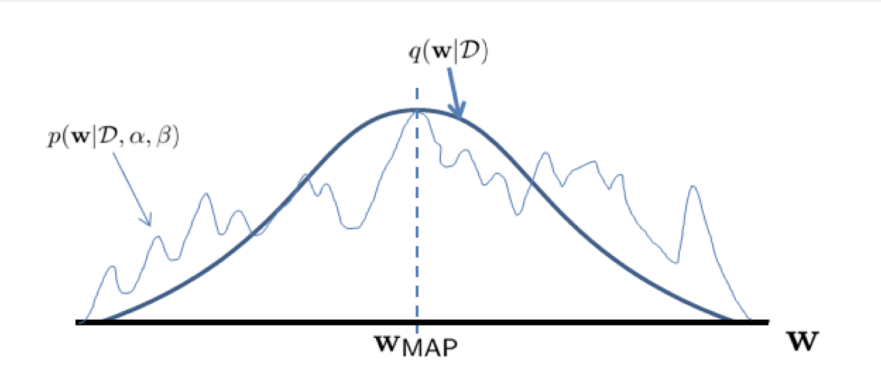
- Bayesian Neural Networks apply uncertainty to NNs.
- Methods to get a posterior over the weights.
- Yields a bound of confidence for weights and outputs.
- benefits in decision-making (e.g. to brake, or not to brake).

In many cases the posterior over the weights is intractable
$\Rightarrow$ Need for approximating methods.

Approximating methods: LAPLACE

Laplace approximation for NNs

- approx. true posterior by Gaussian centered at the maximum-a-posteriori/mode of the weights W_{MAP}
- curvature is given by Hessian H w.r.t to the Loss L evaluated at W_{MAP}
- KFAC:
  - Hessian of every layer gets approximated by a Kronecker-product of two smaller matrices.

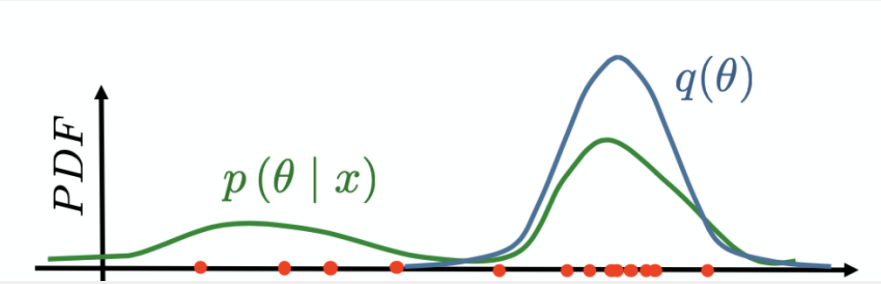


Placeholder, will create similar one

Approximating methods: Variational Inference

Variational Inference

- approx. true posterior $p(W|\mathcal{D})$ with parameterized variational distribution $q(W|\theta)$.
- objective: minimize the Kullback-Leibler divergence $\mathrm{KL}(q(W|\theta)||p(W|\mathcal{D}))$.
- tractable objective: maximize ELBO instead.



Placeholder, will create similar one

Research Goals

Research Goals

What I want to achieve/know:

- main objectives:
  - are there differences in the weight distributions (i.e. the uncertainties) of the layers in a trained network?
  - are there differences in the weight distribution of a single layer?
- extension: are there differences in the weight distributions of the layers during training?

Research Goals

Possible implications:

- Uneven distributions in uncertainty (e.g. much in the last layer) could implicate to not even consider the uncertainties in the other layers.
- If a weight or layer stays unidentified during training, the network could be pruned at this location.
  - yields possible improvements in model's training procedure and architecture.
- Tracking the uncertainty over the weights during training might give insights into convergence criteria (extension).

PROCEDURE

First goal:

- Use a network with simple architecture (starting with MNIST).
- Apply Laplace approximation and Variational Inference to get uncertainty estimates.
- Find the location of the uncertainty.
- create visualisation tools to make findings more comprehensible.

Second goal:

- Use more complex network, such as VGG
- Transfer previous methods
- visualization

PROCEDURE

possible extensions:

- observe, how the uncertainty behaves during training
- measure the influence of the size of the weights to the resulting uncertainty.
- add a third method (e.g. KFAC) two get uncertainty estimates from used NNs.

Results so far

TDB

- tbd
- tbd
- ...