Where is the uncertainty in Neural Networks?

Moritz Kniebel

Universität Tübingen

July 15, 2020

Why do we need uncertainty in Neural Networks?

Problems

- NNs output point estimates
- Unknown uncertainties and overconfidence
- Especially problematic in safety-critical applications (e.g. self-driving cars)

Problems

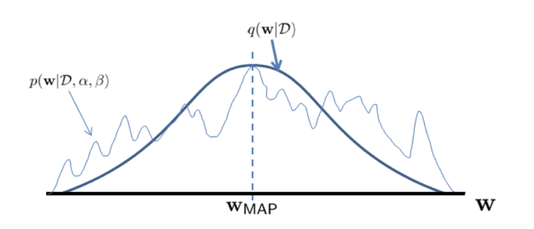
- NNs output point estimates
- Unknown uncertainties and overconfidence
- Especially problematic in safety-critical applications (e.g. self-driving cars)

Solutions

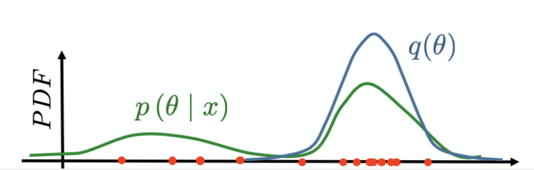
- Bayesian Neural Networks add a prior to the weights
- Posterior over weights can be formulated using Bayes theorem
- Posterior lets us make predictions about new data with a bound of confidence

But...

- Posterior over the weights becomes intractable
- Needs to be approximated



- approx. true posterior by Gaussian centered at the 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.



- 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.

Research Goals

Research Goals

main objectives

- observe the differences in weight distributions to locate uncertainty
- locate the uncertainty in a single layer
- create visualizations of the uncertainty

extensions

- observe uncertainty during training

Possible implications

- training methods can focus on certain parts first
- unidentified parts could be pruned from a network
- Tracking uncertainty during training might give insights into convergence criteria (extension).

First goal

- Use network with simple architecture
- Apply Laplace approximation and Variational Inference to get uncertainty estimates.
- Find the location of the uncertainty
- create visualization tools to make findings more comprehensible

First goal

- Use network with simple architecture
- Apply Laplace approximation and Variational Inference to get uncertainty estimates.
- Find the location of the uncertainty
- create visualization tools to make findings more comprehensible

Second goal

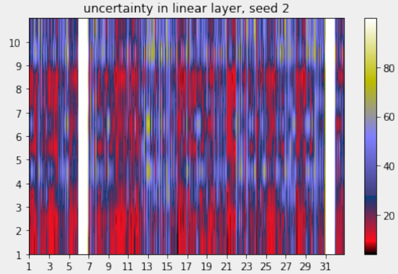
- Use more complex network, such as VGG
- Transfer previous methods
- create visualization

possible extensions

- observe uncertainty during training
- measure the influence of the size of the weights to the resulting uncertainty
- add a third method (e.g. KFAC) to get uncertainty estimates

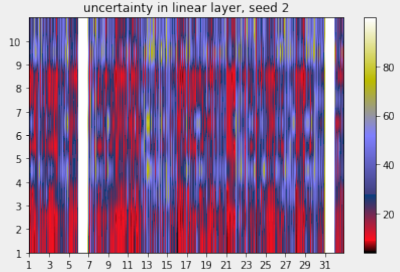
Results so far

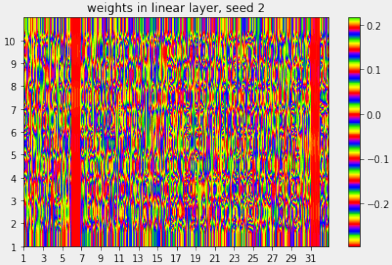
Uncertainty in linear layer



- occurrence of features with high uncertainty

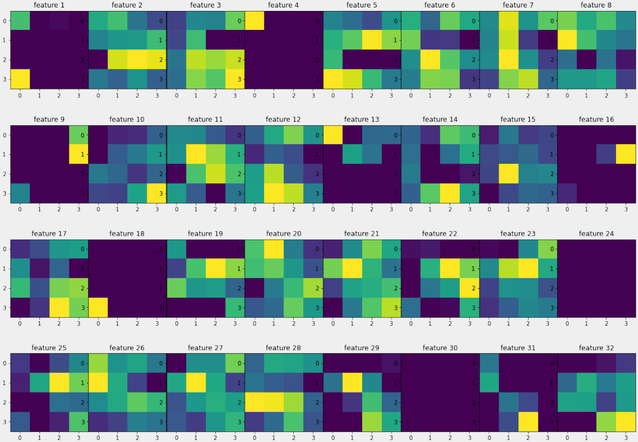
Uncertainty in linear layer





- occurrence of features with high uncertainty
- correlation with size of weights

Uncertainty in linear layer



- no noticeable correlation with inputs of linear layer

Network's Uncertainty distribution

prior precision	0.weight	0.bias	3.weight	3.bias	7.weight	7.bias
1	0.9988	0.9945	0.9998	0.9995	0.9953	0.9988
0.1	3.1283	3.0128	3.1587	3.1483	3.0441	3.1264
0.01	9.1787	7.5770	9.8925	9.5996	8.2552	9.0650
0.001	20.8763	13.0098	29.0170	24.2664	18.1852	19.0137
0.0001	31.0822	15.7158	66.6581	42.2745	34.2089	22.4717
0.00001	34.9811	16.8371	112.8993	67.5562	64.5365	27.2227

Table: mean standard deviations of each layer given a prior precision

- uncertainty always maximal in third layer

Feedback and Questions?