Bayesian Networks: Weight Uncertainty in Neural Networks

Simon Aertssen[1], Peyman Kor[1], Didrik Nielsen[1] and Ole Winther[1]

1 DTU Compute, Technical University of Denmark

Introduction

As is noted in the field of machine learning, a machine can make the wrong decisions with almost a 100% confidence. We investigated how uncertainty can be built into Neural Networks, to improve generalization when learning.
This project builds on the publication "Weight Uncertainty in Neural Networks" [1], which introduced a new backpropogation algorithm that could be used in Bayesian neural networks: "Bayes by Backprop". The method infers the posterior distribution over the weights in the network, using simple prior assumptions. This construction has a regularizing effect as something like an ensemble of networks is obtained and introduces uncertainty on the prediction.

Bayesian Neural Network Model

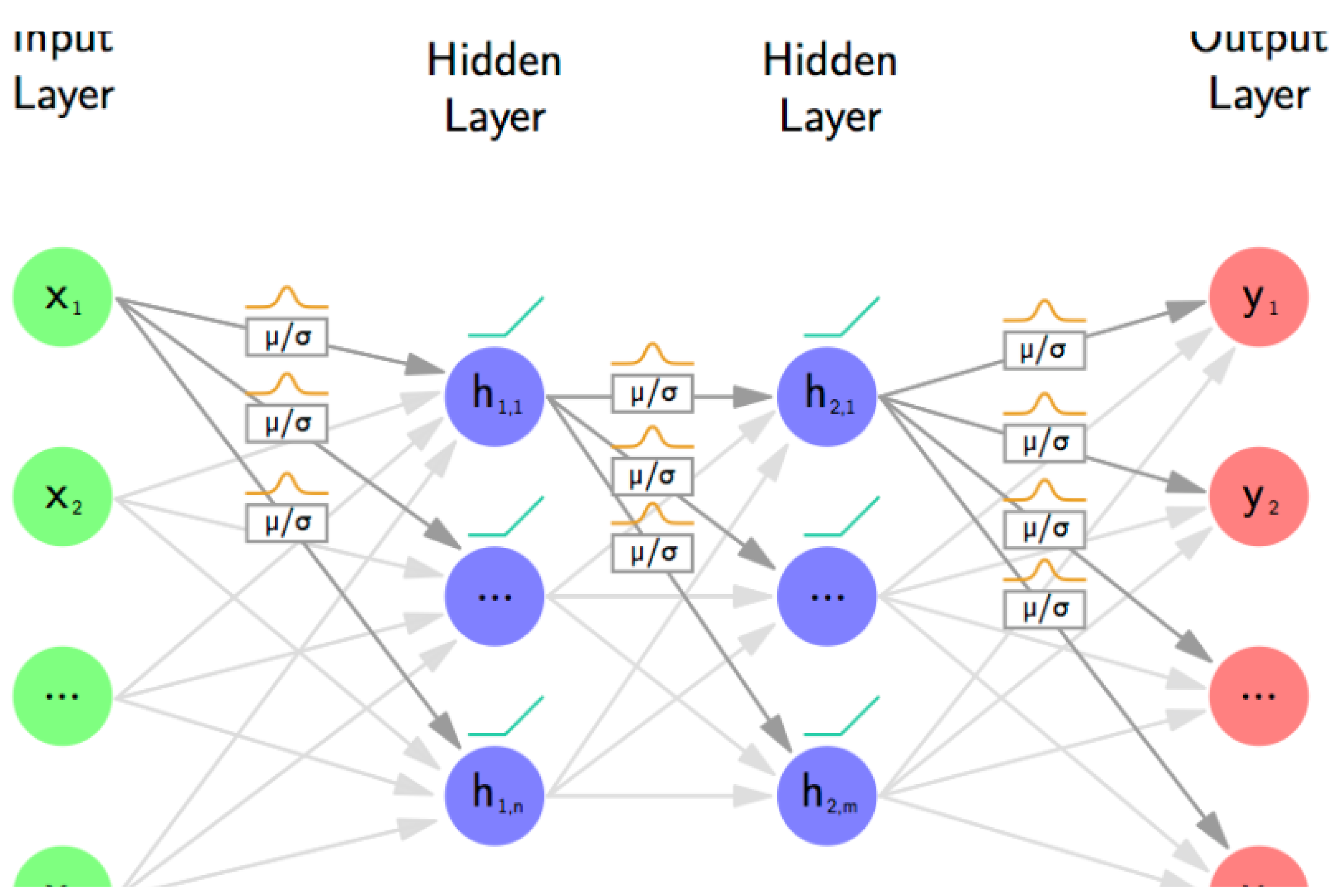

Figure 1: Model architecture representation.

Datasets : Classification and Regression Cases

Table 1 shows that models trained on the HÃ¶glund dataset have poor generalization performance on our new dataset, which reflects the high level of homology and possibly erroneous annotations in this dataset. This is further corroborated on Figure 2, for the HÃ¶glund trained model, all locations are almost perfectly separated.

Table 1: Comparison of generalization performances between the DeepLoc dataset and the Höglund dataset.

Training set	Test set	Accuracy	Gorodkin
DeepLoc	DeepLoc	0.7511	0.6988
Höglund	DeepLoc	0.6426	0.5756
DeepLoc	Höglund	0.8301	0.8010
Höglund	Höglund	0.9138	0.8979

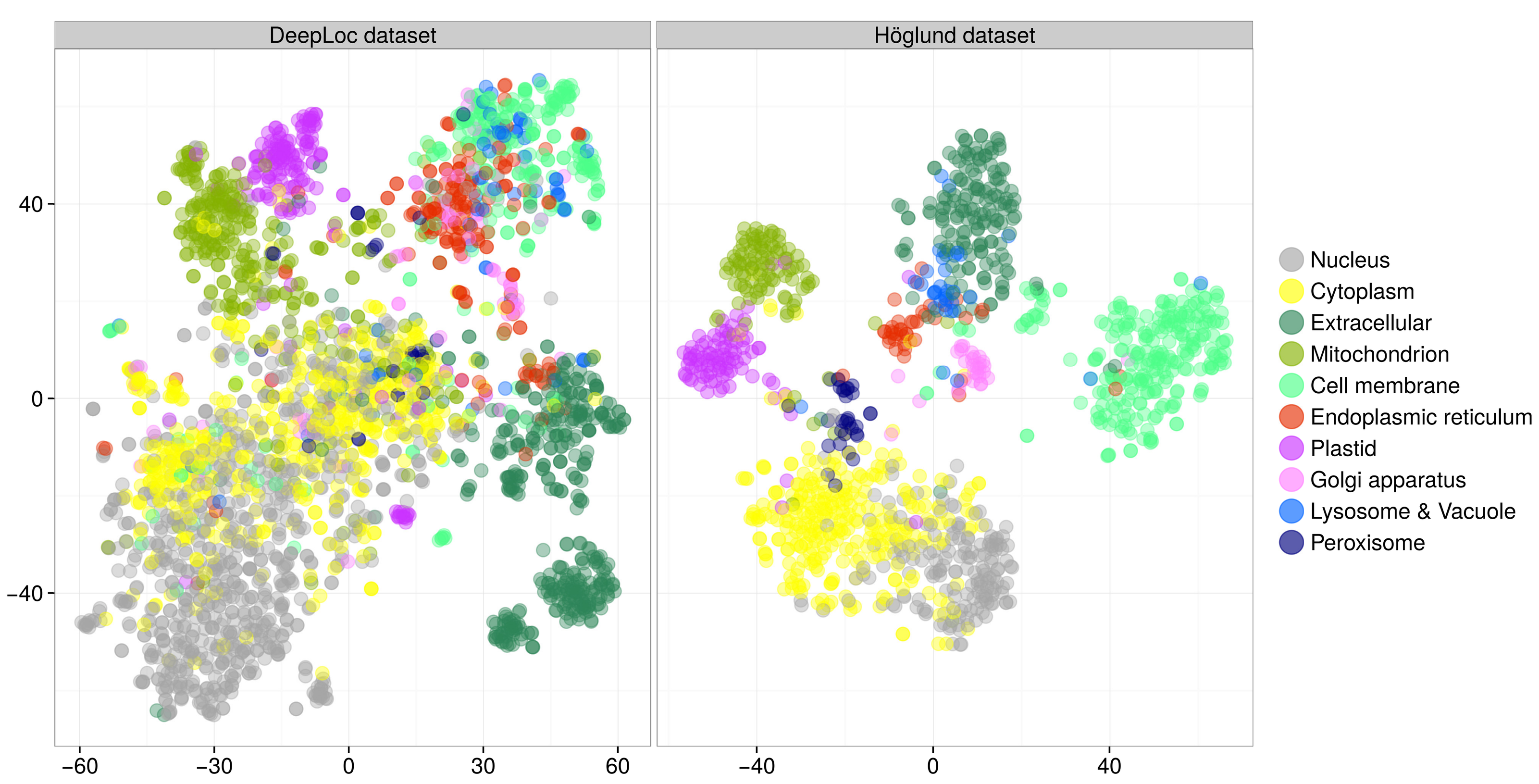

Figure 2: t-SNE representation of the context vector for the model trained on the DeepLoc and HÃ¶glund dataset and visualized for the respective test sets.

Key points

- **Setup:** the neural network is a probabilistic model $P(\mathbf{y}|\mathbf{x},\mathbf{w})$.
- **Objective:** predict unknown $\mathbf{y}^*$ from test input $\mathbf{x}^*$ using train data $\mathcal{D} = (\mathbf{x},\mathbf{y})$.
- Use the predictive distribution:

$$P(\mathbf{y}^*|\mathbf{x}^*) = \mathbb{E}_{P(\mathbf{w}|\mathcal{D})}[P(\mathbf{y}^*|\mathbf{x}^*,\mathbf{w})] = \int P(\mathbf{y}^*|\mathbf{x}^*,\mathbf{w})P(\mathbf{w}|\mathcal{D})\mathrm{d}\mathbf{w}$$

- **Challenge:** an analytical solution for the posterior $P(\mathbf{w}|\mathcal{D})$ is **intractable**.
- **Solution:** use variational approximation $q(\mathbf{w}|\theta) \approx P(\mathbf{w}|\mathcal{D})$ by minimizing the difference between the distributions. Bayes gives an expression for the posterior:

$$P(\mathbf{w}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{P(\mathcal{D})}$$

- Use the KL-divergence

$$\begin{aligned}\theta_{opt} &= \arg\min_\theta \mathbf{KL}\,[\,q(\mathbf{w}|\theta)\,\|\,P(\mathbf{w}|\mathcal{D})\,] \\ &= \arg\min_\theta \mathbf{KL}\,[\,q(\mathbf{w}|\theta)\,\|\,P(\mathbf{w})\,] - \mathbb{E}_{q(\mathbf{w}|\theta)}[\log P(\mathcal{D}|\mathbf{w})\,] \\ &= \arg\min_\theta \mathcal{F}(\mathcal{D},\theta)\end{aligned}$$

- Compute $\mathcal{F}$ by MC sampling n times from the learned distribution. The authors propose the following approximated cost function:

$$\mathcal{F}(\mathcal{D},\theta) \approx \sum_{i=1}^{n} \log q\left(\mathbf{w}^{(i)}|\theta\right) - \log P\left(\mathbf{w}^{(i)}\right) - \log P\left(\mathcal{D}|\mathbf{w}^{(i)}\right)$$

Bayes backprop and Vanilla SGD

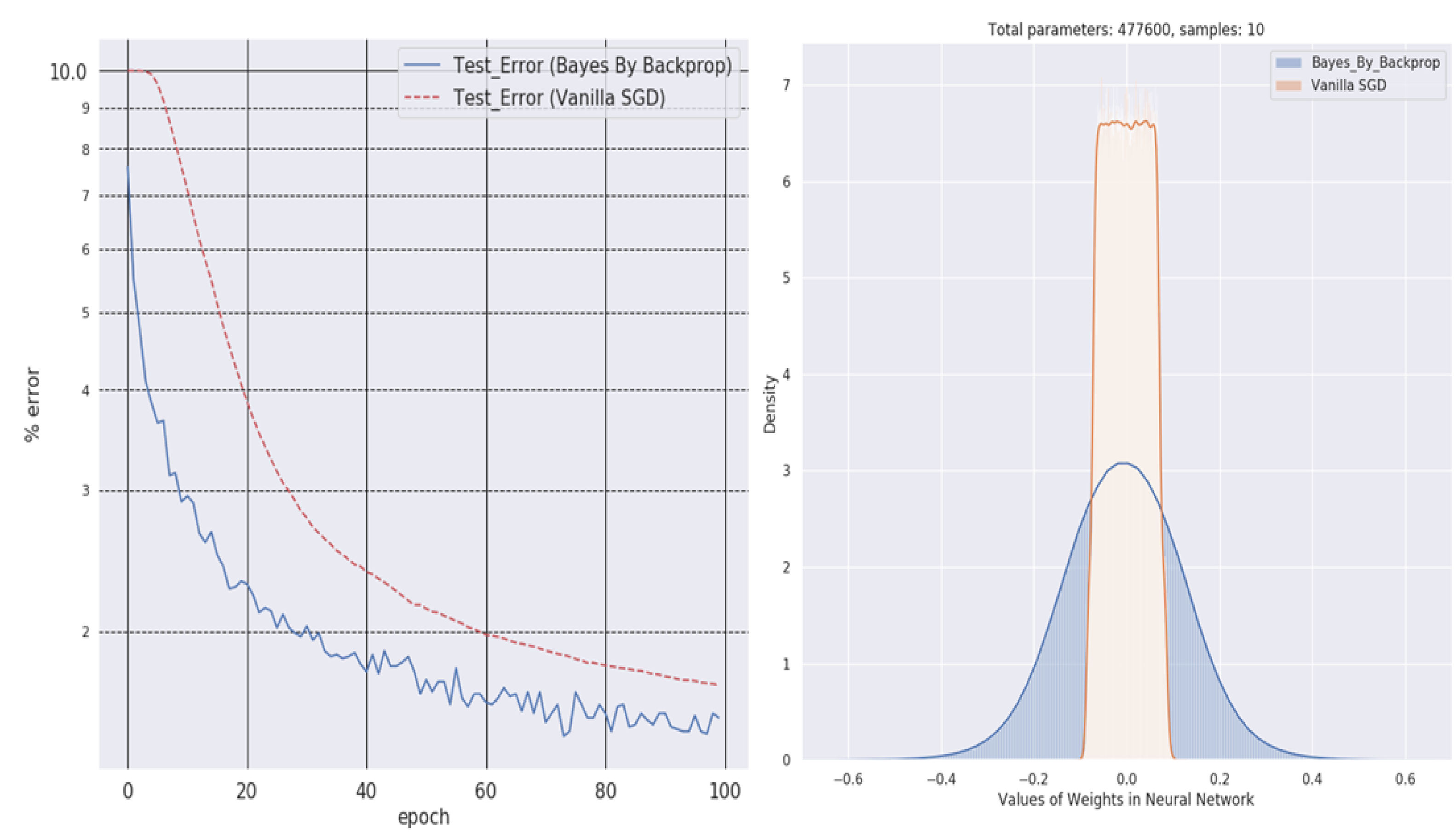

Figure 3: Sequence importance across the protein sequence of DeepLoc test set when making the prediction.

Visualization of BNN Concept

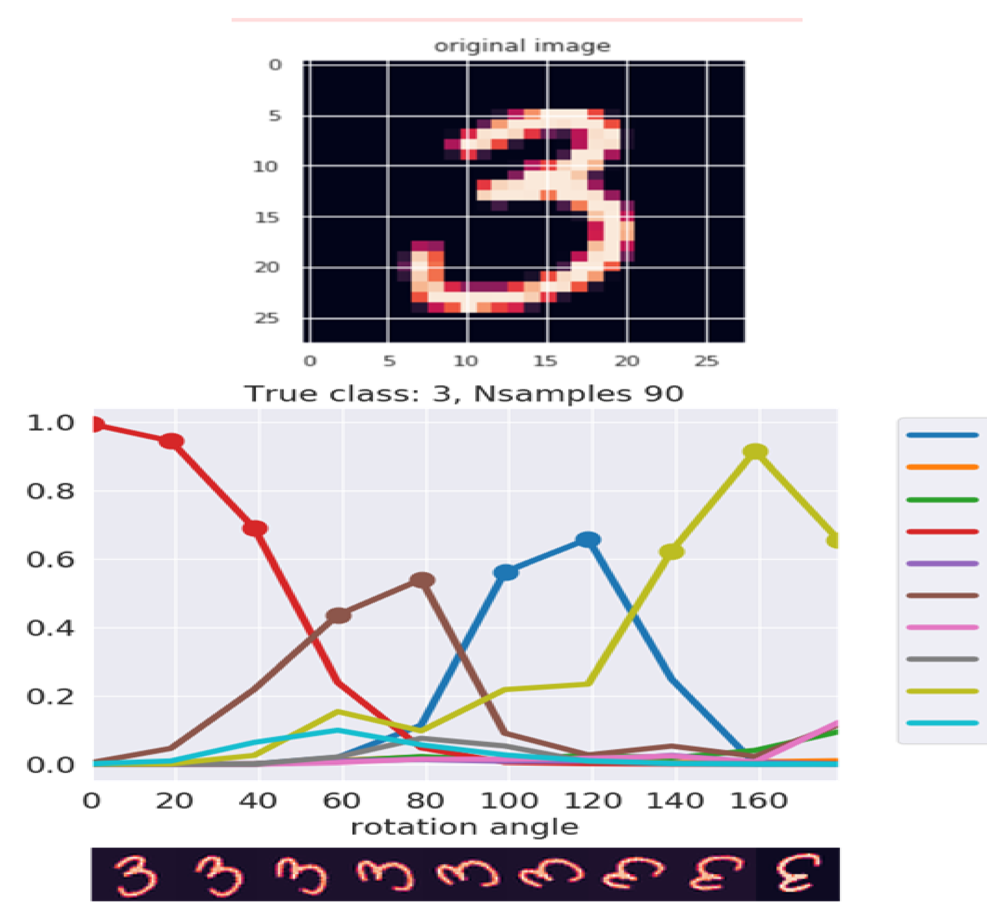

Figure 4: Sequence importance across the protein sequence of DeepLoc test set when making the prediction.

Acknowledgements

The authors wish to thank Konstantinos Tsirigos and Arne Elofsson of Stockholm University for permission to use their fast profile construction method in DeepLoc, even though it has not been published yet. In addition, we want to thank Fabian Aicheler of University of TÃ¼bingen for kindly running the DeepLoc test set on YLoc.

References

[1] K. K. . D. W. C. Blundell, J. Cornebise. Weight uncertainty in neural networks. *Google Deepmind*, May 2015.

jjaa@bioinformatics.dtu.dk