

# **Bachelor's Thesis**

## **Literature search**

March 24, 2023

### **The main four papers I will use for my thesis:**

#### **Error perception classification in Brain-Computer Interfaces using CNN [2]**

The main goal of this paper was to implement a CNN to predict error potentials which could be used for error correction in Brain-Computer Interfaces (BCIs). Error-related Potentials (ErrP) have been used to include the perception of subjects on errors caused by these BCIs. ErrPs originate from the anterior cingulate cortex and are elicited as a response to error perception after a feedback presentation. It used the publicly available dataset from the BNCI Horizon 2020 project website under the name Monitoring error-related potentials. The data was generated by subjects who monitored a moving cursor without controls. There was a 20% chance for the cursor to move the wrong way. The only preprocessing that was done was a 1 to 10hz Butterworth bandpass filter. All 64 channels were used in windows of 600ms after the stimuli. This gave better results than the two selected channels (FCZ and FZ). They achieved an accuracy of 80%, 76% sensitivity, and 85% specificity.

I will use the same dataset this author used for my thesis since this dataset is the best for the ErrP classification and only uses 'mental' tasks without physical movements, so there will be fewer physiological artifacts.

#### **Estimating the Mean and Variance of the Target Probability Distribution [10]**

This paper aims to develop a feed-forward network that can also predict its uncertainty in a regression task. To achieve this, they created a network with two heads, each head having its own hidden layers. They only share the input layer without any shared hidden layers. For this particular task, it showed better performance. First, only the weights of the y-headed side (the actual regression task) were trained, and the  $\sigma^2$  (Variance) head was ignored. The reasoning is to reduce the computing load since training the variance predictor does not make sense if the regression task is nowhere near accurate.

This paper aims for regression tasks with a clear difference between the predicted and actual continuous values. This strategy becomes more challenging in a classification task, but the following paper has some solutions to this problem.

#### **What uncertainties do we need in bayesian deep learning for computer vision [5]**

Deep learning models have two major types of uncertainty. The first one is epistemic uncertainty. This is the uncertainty of the model parameters. This can often be explained away, given enough data. The other type is aleatoric uncertainty. This is noise inherent to the observations, such as faulty sensors. The latter can be further categorized into homoscedastic uncertainty, constant regardless of the input,

and heteroscedastic uncertainty, which depends on the inputs. The latter sparks the most interest in my research topic since I want to track down artifacts in the signals. These uncertainties can be modeled using Bayesian deep learning approaches. To capture epistemic uncertainty in NNs, a prior distribution is put over all weights, such as a Gaussian distribution. These are called Bayesian Networks. They try to capture how much these weights vary on the data to model this uncertainty. For the aleatoric uncertainty, a distribution is placed over the model's output, and then try to learn the noise's variance based on the inputs.

The paper suggests using a single network with its head split to predict  $\hat{y}$  and  $\hat{\sigma}^2$ . The model does need 'uncertainty labels' to learn the uncertainty. Instead, it infers it from the loss function during the regression task. To make the model work for classification tasks, some adjustments must be made. In their example, the model predicts a vector of unaries, which forms a probability vector when passed through a softmax function. They place a Gaussian distribution over this vector. Then they approximate through Monte Carlo integration and sample unaries through the softmax function.

All formulas and a better description can be found in the paper in section 3.

As far as I can understand from this paper, the implementation should look like the following:

First, the model's body will be a 'normal' (not bayesian) convolutional network, such as the EEGNet model, which will be explained in the following paper. This model then has two output heads. One which predicts the mean ( $\mu$ ) and one to predict the variance ( $\sigma^2$ ). Each produces a tensor of length 2 (one for each class: ErrP, not Errp). Next,  $T$  samples are taken from these distributions and passed to the softmax function. Finally, the loss can be calculated using something like  $\beta$ -NLL.

## **EEGNet: a compact convolutional neural network for EEG-based brain-computer interfaces [7]**

This paper proposes a neural network architecture for machine learning models that classify EEG signals, mainly for BCIs. They present an architecture called 'EEGNet.' This architecture is a compact convolutional network (CNN) specifically designed and tailored for EEG signals. These signals generally have a low signal-to-noise ratio and high inter-subject variability. The architecture of EEGNet consists of multiple temporal convolutional layers and a max pooling layer, followed by a spatial convolutional layer and a global averaging layer.

The authors demonstrated the capabilities of this architecture by training and testing it on various tasks and datasets. A few examples are motor imagery tasks and auditory attention tasks. They compared their architecture to various pre-existing architectures and showed that it received higher accuracies than the pre-existing architectures while using fewer parameters. The model was also quite robust to variations; even training runs on small datasets showed promising results.

I will likely use this architecture and convert it to a multi-headed network for my project. Or at least take inspiration from the internal construction of this architecture.

## **Papers that are useful for the introduction and comparison with different methods:**

### **On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks [12]**

Typically, aleatoric uncertainty in regression tasks can be predicted using a neural network to predict the parameters of the distribution, when assuming a heteroscedastic Gaussian distribution. Usually, using stochastic descent, MLE (Maximum Log Likelihood) is used to learn by minimizing the NLL (Negative Log Likelihood). This method is pretty much de-facto and was introduced by Nix[10] However, this

method highly depends on the gradients of the predictive variance, creating an unstable model. This paper suggests an alternative loss function  $\beta - NLL$ , which counteracts by weighting each data point's contributions to the overall loss by its  $\beta$  estimate, which controls the dependency of gradients on the predictive variance. Refer back to the paper for the exact theory on how this works.

The GitHub repository contains an implementation of a neural net for a regression task which could be helpful.

### **Uncertainty quantification for multilabel text classification [1]**

They used Bayesian Networks to quantify aleatoric uncertainties in a multilabel text classification environment. They mainly aimed at capturing the homoscedastic aleatoric uncertainty, which is constant across all inputs. In our case, we are primarily interested in capturing the heteroscedastic aleatoric uncertainty.

### **Uncertainty estimation in medical image classification: systematic review [6]**

Medical image classification is an essential task in medical diagnosis and treatment planning. However, obtaining large and well-labeled datasets for training deep neural networks in medical image analysis is often challenging, which can lead to uncertainty in the model's predictions. The authors propose a method to estimate the uncertainty in the model's predictions using a Bayesian convolutional neural network. This method resulted in better results than previous state-of-the-art methods.

### **Inhibited softmax for uncertainty estimation in neural networks [8]**

The author suggests a modified version of the softmax function in this paper, commonly used in classification tasks. This modified version is called the Inhibited softmax function. It takes in another parameter called the inhibition parameter. This parameter controls the degree to which confidence in its prediction is inhibited. The authors show that this modified softmax can predict both aleatoric and epistemic uncertainty.

### **A deeper look into aleatoric and epistemic uncertainty disentanglement [14]**

For many applications where you wish to predict uncertainty, attempting to disentangle this 'global' uncertainty into its aleatoric and epistemic counterpart can be beneficial. The author reviews existing methods for disentangling this uncertainty, such as maximum likelihood estimation and bayesian deep learning and ensembles. They also propose a technique called Monte Carlo Dropout with Gaussian Smoothing (MCD-GS). This method uses dropout regularization with Gaussian smoothing to estimate both types of uncertainty.

### **Optimal Training of Mean Variance Estimation Neural Networks [13]**

The authors propose a novel approach to MVE (Mean Variance Estimation) tasks in this paper. This paper focuses on regression tasks, such as financial and economic predictions, where a mean and variance are desired. They propose two output layers to predict the mean and variance of the distribution separately. The novel contribution of this paper is a new training algorithm that improves the accuracy of the mean and variance estimations. They combine a regularised maximum likelihood objective with a variance clipping method, which stops the network from predicting excessively high or low variances.

They demonstrate the capabilities of their approach on various MVE regression tasks, such as portfolio optimization and option pricing. They compare them with existing state-of-the-art methods and show that their approach outperforms the existing methods.

## Further possible readings

- BayesNetCNN: incorporating uncertainty in neural networks for image-based classification tasks [3]
- Confidence estimation methods for neural networks: A practical comparison [11]
- Practical uncertainty quantification for brain tumor segmentation [4]
- Reliable multi-class classification based on pairwise epistemic and aleatoric uncertainty[9]

## References

- [1] Wenshi Chen, Bowen Zhang, and Mingyu Lu. Uncertainty quantification for multilabel text classification. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(6):e1384, 2020.
- [2] J Rafael Correia, J Miguel Sanches, and Luca Mainardi. Error perception classification in brain-computer interfaces using cnn. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 204–207. IEEE, 2021.
- [3] Matteo Ferrante, Tommaso Boccato, and Nicola Toschi. Bayesnetcnn: incorporating uncertainty in neural networks for image-based classification tasks. *arXiv preprint arXiv:2209.13096*, 2022.
- [4] Moritz Fuchs, Camila Gonzalez, and Anirban Mukhopadhyay. Practical uncertainty quantification for brain tumor segmentation. In *Medical Imaging with Deep Learning*, 2021.
- [5] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.
- [6] Alexander Kurz, Katja Hauser, Hendrik Alexander Mehrtens, Eva Krieghoff-Henning, Achim Hekler, Jakob Nikolas Kather, Stefan Fröhling, Christof von Kalle, and Titus Josef Brinker. Uncertainty estimation in medical image classification: systematic review. *JMIR Medical Informatics*, 10(8):e36427, 2022.
- [7] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain-computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.
- [8] Marcin Możejko, Mateusz Susik, and Rafał Karczewski. Inhibited softmax for uncertainty estimation in neural networks. *arXiv preprint arXiv:1810.01861*, 2018.
- [9] Vu-Linh Nguyen, Sébastien Destercke, Marie-Hélène Masson, and Eyke Hüllermeier. Reliable multi-class classification based on pairwise epistemic and aleatoric uncertainty. In *27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*, pages 5089–5095, 2018.
- [10] David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, volume 1, pages 55–60. IEEE, 1994.
- [11] Georgios Papadopoulos, Peter J Edwards, and Alan F Murray. Confidence estimation methods for neural networks: A practical comparison. *IEEE transactions on neural networks*, 12(6):1278–1287, 2001.

- [12] Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. *arXiv preprint arXiv:2203.09168*, 2022.
- [13] Laurens Sluijterman, Eric Cator, and Tom Heskes. Optimal training of mean variance estimation neural networks. *arXiv preprint arXiv:2302.08875*, 2023.
- [14] Matias Valdenegro-Toro and Daniel Saromo Mori. A deeper look into aleatoric and epistemic uncertainty disentanglement. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1508–1516. IEEE, 2022.