Correlated Parameters to Accurately Measure Uncertainty in Deep Neural Networks

Konstantin Posch, Juergen Pilz

Alpen-Adria-University Klagenfurt

27.11.2020

Motivation and Research Objectives

- Deep learning has to struggle with two problems:
  - Overfitting (often modern deep nets include millions of learnable parameters)
  - Lack of model uncertainty information (only point estimates of the network parameters are computed)
- Model uncertainty directly translates to prediction uncertainty
- Missing model uncertainty information is critical in the medical field or for self-driving vehicles
- Both problems are well addressed by using Bayesian statistics
- A new approach for training deep nets in a Bayesian way will be presented

Content

Classification Networks

- A classification network is a mapping $\mathbf{f} : D \subseteq \mathbb{R}^{\alpha} \to [0,1]^{c}$
- $\mathbf{f}(\mathbf{x};\mathbf{w})_k$ is used to model the probability that the input $\mathbf{x} \in D$ belongs to class $k \in \{1,...,c\}$, i.e. $P(\mathbf{x} \text{ belongs to class } k) = \mathbf{f}(\mathbf{x};\mathbf{w})_k$
- Artificial neurons are the units neural networks consist of
- A neuron η is a real valued mapping:

$$\eta : \mathbb{R}^{n} \to \mathbb{R} \qquad \tilde{\mathbf{x}} \mapsto g\left(\mathbf{w}_{\eta}^{T}\tilde{\mathbf{x}} + b_{\eta}\right)$$

  - $\tilde{\mathbf{x}}$ denotes outputs of other neurons
  - g denotes an activation function, such as the rectified linear unit:

$$g(x) = \max\{0, x\}$$

Deep Networks

- Deep networks consist of *(many)* disjoint sets $S_1, ..., S_d$ of neurons; connections are allowed between two successive sets

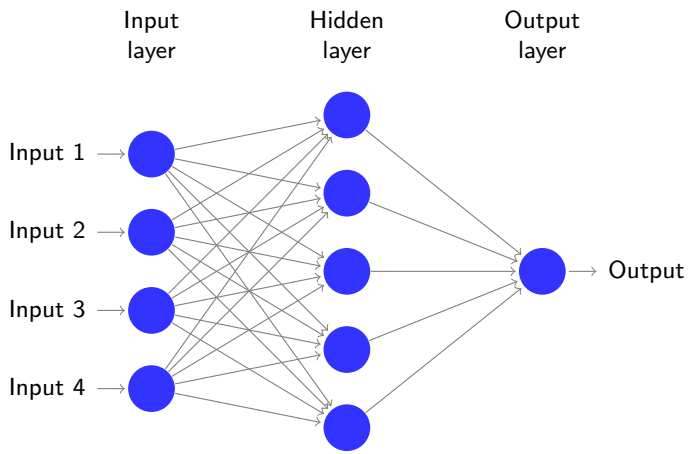

Figure: Neural network

- Often deep networks require exponential less parameters than shallow ones to approximate a given mapping well

Convolutional Neural Networks (CNNs)
State of the art image classifiers consist of three layer-types:

1. Convolutional layers
   - Extract features *(2d arrays)* of an image
   - Feature extraction: a kernel *(small 3d array)* is slidden over the previous layer
2. Pooling layers: compress the extracted features
3. Fully connected layers: use the compressed features for classification

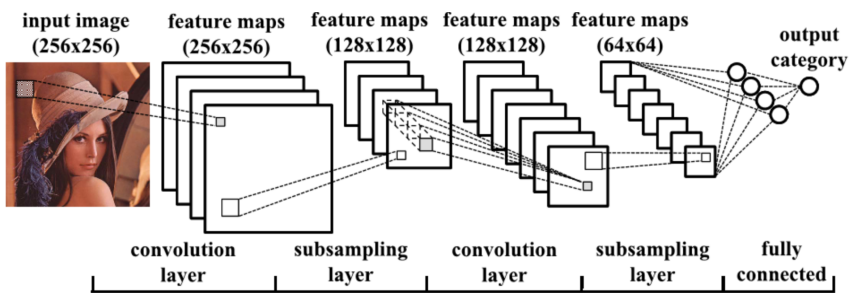

Figure: CNN example

Variational Inference

- Let $\mathbf{W}$ denote the random network parameters and $D = \{\mathbf{X}, \mathbf{y}\}$ the observed data *(instances & corresponding class labels)*
- The posterior $p(\mathbf{w}|\mathbf{y}, \mathbf{X})$ is intractable, because of the high dimensional integral in the denominator:

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y}|\mathbf{w}, \mathbf{X})p(\mathbf{w})}{\int p(\mathbf{y}|\mathbf{w}, \mathbf{X})p(\mathbf{w})\ d\mathbf{w}}$$

- Variational inference aims at approximating the posterior $p(\mathbf{w}|\mathbf{y}, \mathbf{X})$ with another parametric distribution $q_\phi(\mathbf{w})$
- The Kullback-Leibler divergence

$$D_{KL}(q_\phi(\mathbf{w})||p(\mathbf{w}|\mathbf{y}, \mathbf{X})) := \mathbb{E}_{q_\phi(\mathbf{w})}\left(\ln \frac{q_\phi(\mathbf{w})}{p(\mathbf{w}|\mathbf{y}, \mathbf{X})}\right)$$

  is minimized

- Since the posterior is unknown $D_{KL}(q_\phi(\mathbf{w})||p(\mathbf{w}|\mathbf{y},\mathbf{X}))$ cannot directly be minimized
- Therefore, the negative log evidence lower bound L_{VI} is minimized

$$L_{VI} = -\mathbb{E}_{q_{\phi(\mathbf{w})}} \ln p(\mathbf{y}|\mathbf{X},\mathbf{w}) + D_{KL}(q_\phi(\mathbf{w})||p(\mathbf{w}))$$

- Minimization of L_{VI} is equivalent to minimization of $D_{KL}(q_\phi(\mathbf{w})||p(\mathbf{w}|\mathbf{y},\mathbf{X}))$
- Comparison to the classical approach:
  - Since $\mathbf{W}$ is random the expected likelihood is maximized
  - Instead of the L2-norm the KL-divergence is penalized

Prediction and Prediction Uncertainty

- The posterior predictive distribution $p(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X})$ reflects the belief in a class label y^* for a given example $\mathbf{x}^*$ after observing data $\mathbf{y}, \mathbf{X}$:

$$p(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X}) = \int p(y^*|\mathbf{w}, \mathbf{x}^*) p(\mathbf{w}|\mathbf{y}, \mathbf{X}) \, d\mathbf{w}$$

- $p(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X})$ can be approximated by computing the mean of multiple network outputs with parameters sampled from $q_\phi(\mathbf{w})$
- One can estimate credible intervals for the probability that $\mathbf{x}^*$ belongs to class y^* by computing empirical quantiles of the random network outputs

Overview of Selected Variational Distributions from Literature

- Gaussian distribution with a diagonal covariance matrix

  Graves (2011), Blundell et al. (2015)

- Dropout as Bernoulli variational distribution (randomly dropping a neuron in layer $i-1$ is equivalent setting all weights in layer i to zero, which represent connections to this one neuron)

  Gal and Ghahramani (2015)

Proposed Variational Distribution

The variational distribution presented aims at satisfying the following requirements:

- Network parameters can be correlated
  - Enables an exchange of information between different parts of the network
  - Leads to more accurate uncertainty estimates than methods which assume independence
- The number of parameters to be optimized does not differ significantly from the non-Bayesian case
  - Bernoulli dropout (Gal and Ghahramani, 2015) works well and does not introduce additional parameters (except of the dropping rate)
  - Allows for an easy interpretation of the additional parameters
  - Should guarantee that the difficulty of the network optimization does not increase significantly

Overview of New Approach

- Let $\mathbf{W}_i$ denote the random weights of layer i and $\mathbf{B}_i$ denote the random bias terms of layer i $(i = 1, ..., d)$
- Variational distribution: $q_{\phi}(\mathbf{w}) = \prod_{i=1}^{d} q_{\phi_i}(\mathbf{w}_i) q_{\phi_{bi}}(\mathbf{b}_i)$
- $q_{\phi_i}(\mathbf{w}_i)$, $q_{\phi_{bi}}(\mathbf{b}_i)$ denote the densities of normal distributions with expectation vectors $\boldsymbol{m}_i, \boldsymbol{m}_{bi}$ and tridiagonal covariance matrices
- The variances are given by $\tau_i^2 \mathbf{m}_i^2$ and $\tau_{bi}^2 \mathbf{m}_{bi}^2$, respectively
- The correlations are assumed to be identical and given by ρ_i and ρ_{bi}, respectively
- Only 2 parameters are used *(per layer)* to regulate the variance in proportion to the expectation

Prior

- $p(\boldsymbol{w}) = \prod_{j=1}^{d} p(\boldsymbol{w}_j) p(\boldsymbol{b}_j)$,
  where $p(\boldsymbol{w}_j)$ denotes the density of $N(\boldsymbol{\mu}_j, \zeta_j^2 \mathbf{I}_{K_j})$ and $p(\boldsymbol{b}_j)$ denotes the density of $N(\boldsymbol{\mu}_{bj}, \zeta_{bj}^2 \mathbf{I}_{k_j})$
- For $\boldsymbol{\mu}_j = \boldsymbol{\mu}_{bj} = \mathbf{0}$ $(j = 1, ..., d)$ the network parameters are shrunken towards zero (compare to Ridge regularization)
- For $\boldsymbol{\mu}_j, \boldsymbol{\mu}_{bj} \neq \mathbf{0}$ $(j = 1, ..., d)$ a priori knowledge regarding the network parameters can be modeled

Preliminary Remarks

- The presented approach was implemented by extending the fully-connected layer and the convolutional layer of the popular deep learning framework Caffe (developed by Berkeley AI Research)
- In the experimental studies the following holds:
  - The prior is used with the specification $\boldsymbol{\mu}_j = \boldsymbol{\mu}_{bj} = \mathbf{0}$ $(j = 1, ..., d)$
  - The Model is trained according to the following approaches:
    - Proposed approach (Gauss cor.)
    - Proposed approach without correlations (Gauss ind.)
    - Bernoulli Variational distribution (Gal and Ghahramani, 2015)

MNIST & LeNet

- The proposed Bayesian approach was evaluated on the MNIST dataset with the architecture LeNet
- MNIST dataset consists of $70,000$ images of handwritten digits *(*$60,000$ *for training,* $10,000$ *for testing)*
- LeNet:
  1. Convolutional layer with 20 kernels of size $5 \times 5 \times 3$
  2. Convolutional layer with 50 kernels of size $5 \times 5 \times 20$
  3. Fully connected layer with 250 neurons
  4. Fully connected layer with 10 neurons

Figure: Sample images from the MNIST dataset.

Prediction Accuracy Comparison

Table

Relative frequency of misclassified images. The predictions are based on 200 samples from the corresponding variational distribution per test image, respectively.

Model	Test error
Gauss cor.	0.61%
Gauss ind.	1.00%
Bernoulli	0.78%

Overview Variational Parameters

Table

Variational Parameters $\rho_j, \rho_{bj}, \tau_j$ and τ_{bj} $(j = 1, ..., 4)$.

Layer	τ_j	τ_{bj}	ρ_j	ρ_{bj}
Convolutional 1	0.03	0.05	-0.44	0.03
Convolutional 2	0.35	0.05	-0.21	-0.01
Fully connected 1	2.02	0.06	-0.15	-0.01
Fully connected 2	0.06	0.05	-0.18	0.01

Prediction Uncertainty - Correct Classification

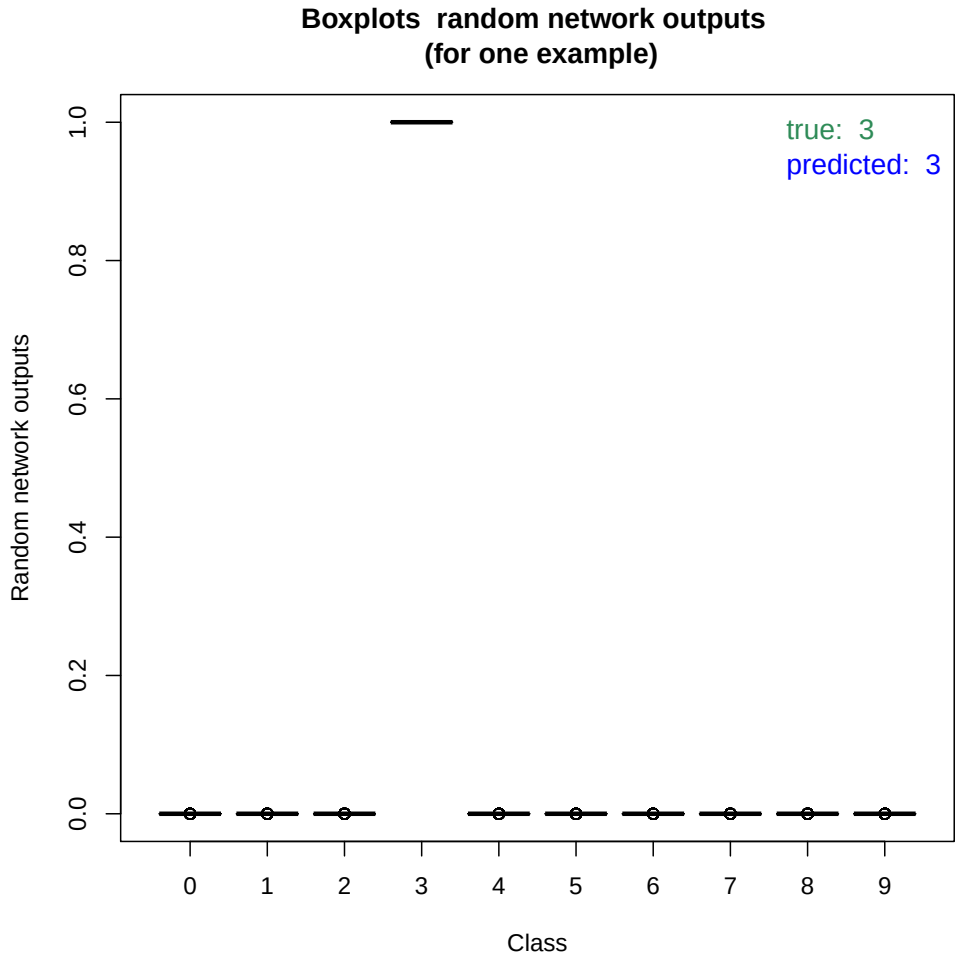

Figure: Boxplots of 200 random network outputs for a representative correct classification result.

Prediction Uncertainty - Incorrect Classification

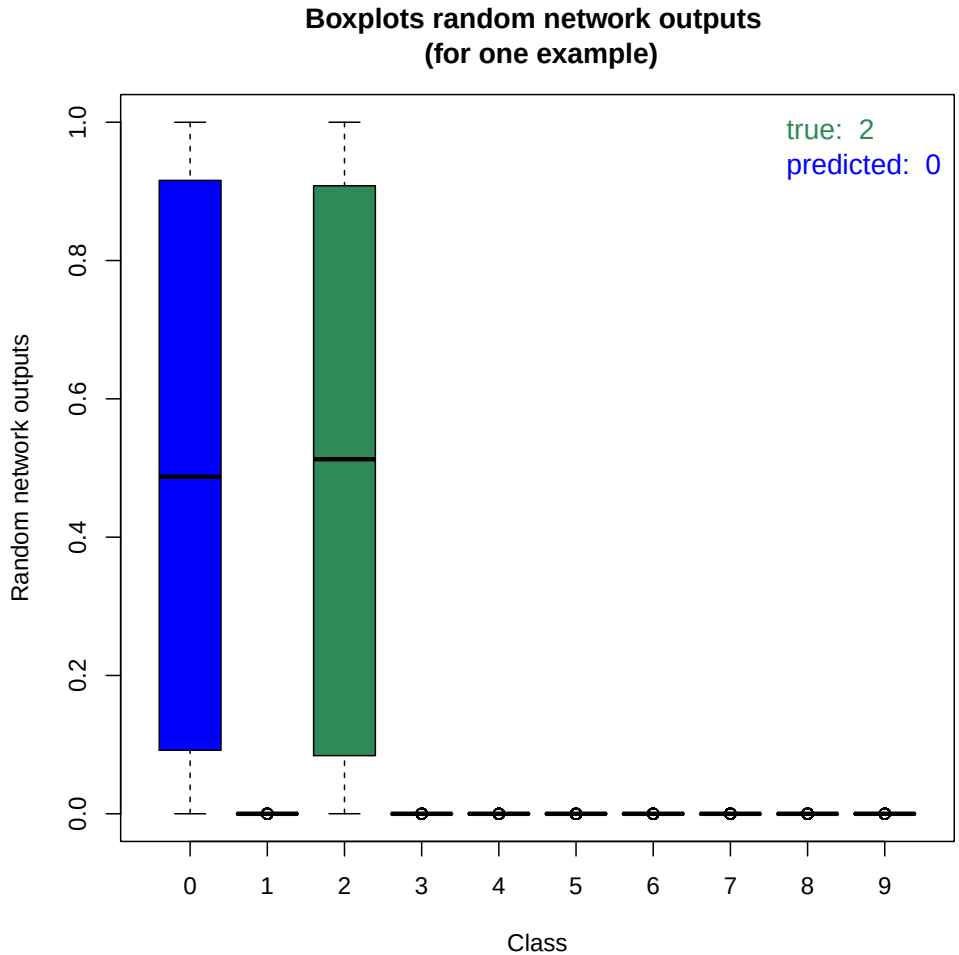

Figure: Boxplots of 200 random network outputs for a representative incorrect classification result.

Usefulness of Prediction Uncertainty

Consider a predicted class *(class with highest a posteriori probability)* as quite certain according to

- criterion (i) if the α credible interval of this class does not overlap with the intervals of the other classes
- criterion (ii) if its posterior probability is greater than or equal to α.

Table

Overview of certain and uncertain prediction results (proposed model)

α	Prediction	Certain (i)	Uncertain (i)	Certain (ii)	Uncertain (ii)
95%	correct	9587	352	9563	376
95%	wrong	8	53	8	53
99%	correct	9371	568	9133	806
99%	wrong	6	55	4	57
99.999%	correct	9112	827	4665	5274
99.999%	wrong	5	56	0	61

Experimental Results

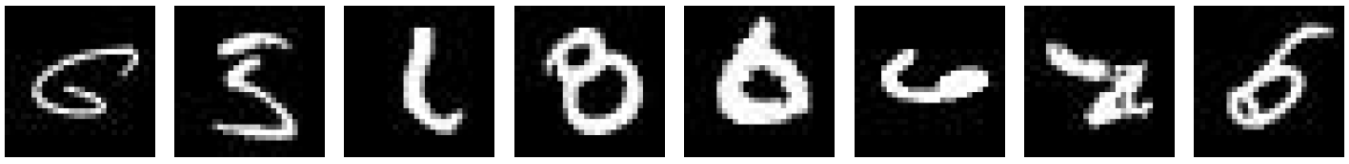

Figure: Images for which the proposed model is certain about the wrong predictions ($\alpha = 95\%$, criterion (i) and (ii) lead to the same images).

Quality of Prediction Uncertainty

Consider two approaches to measure the overall quality of the uncertainty information:

- Log-likelihood of test data $\log[p(\mathbf{y}_{test}|\mathbf{X}_{test}, \mathbf{y}_{train}, \mathbf{X}_{train})]$
- Brier score of test data: Mean of squared Euclidean distance between true class probabilities and estimated class probabilities

Table

Log-likelihood and Brier score of the test dataset

Model	Log-likelihood	Brier score
Gauss cor.	-220.9156	0.01041743
Gauss ind.	-336.4502	0.01522011
Bernoulli	-270.3255	0.01253694

References

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622.

Gal, Y. and Ghahramani, Z. (2015). Bayesian convolutional neural networks with bernoulli approximate variational inference. *ArXiv*, abs/1506.02158.

Graves, A. (2011). Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems 24*, pages 2348–2356. Curran Associates, Inc.

Thank you for your attention!
Any questions?