Inhibited Softmax for Uncertainty Estimation in Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

We present a new method for uncertainty estimation and out-of-distribution detection in neural networks with softmax output. We extend softmax layer with an additional constant input. The corresponding additional output is able to represent the uncertainty of the network. The proposed method requires neither additional parameters nor multiple forward passes nor input preprocessing nor out-of-distribution datasets. We show that our method performs comparably to more computationally expensive methods and outperforms baselines on our experiments from image recognition and sentiment analysis domains.

1 Introduction

The applications of computational learning systems might cause intrusive effects if we assume that predictions are always as accurate as during the experimental phase. Examples include misclassified traffic signs (Evtimov et al., 2018) and an image tagger that classified two African Americans as gorillas (Curtis, 2015). This is often caused by overconfidence of models that has been observed in the case of deep neural networks (Guo et al., 2017). Such malfunctions can be prevented if we estimate correctly the uncertainty of the machine learning system. Beside AI safety, uncertainty is useful in the active learning setting in which data collection process is expensive or time consuming (Houlsby et al., 2011; Rottmann et al., 2018).

While uncertainty estimation in neural networks is an active field of research, the current methods are rarely adopted. It is desirable to develop a method that does not create an additional computational overhead. Such a method could be used in environments that focus on quick training and/or inference. If such a method is simple, the ease of implementation should encourage practitioners to develop danger-aware systems in their work.

We suggest a method that measures uncertainty of the neural networks with a softmax output layer. We replace this layer with Inhibited Softmax layer (Saito et al., 2016), and we show that it can be used to express the uncertainty of the model. In our experiments the method outperforms baselines and performs comparably with more computationally expensive methods on the out-of-distribution detection task.

We contribute with:

- The mathematical explanation why the additional Inhibited Softmax output can be interpreted as an uncertainty measure.
- The additions to the Inhibited Softmax that improve its uncertainty approximation properties.
- The benchmarks comparing Inhibited Softmax, baseline and contemporary methods for measuring uncertainty in neural networks.

2 Related Work

Certainty of classification models can be represented by maximum of probabilities (Hendrycks & Gimpel, 2017). It has been shown, however, that deep neural networks are prone to the overconfidence problem (Guo et al., 2017), and thus so simple a method might not measure certainty well.

The modern Bayesian Neural Networks (Blundell et al., 2015; Hernández-Lobato & Adams, 2015; Louizos & Welling, 2017; Malinin & Gales, 2018; Wang et al., 2016; Hafner et al., 2018; Zhang et al., 2018; Khan et al., 2018) aim to confront this issue by inferring distribution over the models' weights. This approach has been inspired by Bayesian approaches suggested as early as the nineties (Buntine & Weigend, 1991; Neal, 1993). A very popular regularisation mean - dropout - also can be a source of approximate Bayesian inference (Gal & Ghahramani, 2016). Such technique, called Monte Carlo dropout (Gal & Ghahramani, 2015), belongs to the Bayesian Neural Networks class and has been since used in the real-life scenarios (e.g. Leibig et al., 2017). In the Bayesian Neural Networks the uncertainty is modelled by computing the predictive entropy or mutual information over the probabilities coming from stochastic predictions (Smith & Gal, 2018).

Other methods to measure uncertainty of neural networks include a non-Bayesian ensemble (Lakshminarayanan et al., 2017), a student network that approximates the Monte Carlo posterior predictive distribution (Korattikara Balan et al., 2015), modelling Markov chain Monte Carlo samples with a GAN (Wang et al., 2018), Monte Carlo Batch Normalization (Teye et al., 2018) and the nearest neighbour analysis of penultimate layer embedding (Mandelbaum & Weinshall, 2017).

The concept of uncertainty is not always considered as a homogeneous whole. Some of the authors distinguish two types of uncertainties that influence predictions of machine learning models (Kendall & Gal, 2017): epistemic uncertainty and aleatoric uncertainty. Epistemic uncertainty represents the lack of knowledge about the source probability distribution of the data. This uncertainty can be reduced by increasing the size of the training data. Aleatoric uncertainty arises from homoscedastic, heteroscedastic and label noises and cannot be reduced by the model. We will follow another source (Malinin & Gales, 2018) that defines the third type: distributional uncertainty. It appears when the test distribution differs from the training distribution, i.e. when new observations have different nature then the ones the model was trained on.

A popular benchmark for assessing the ability of the models to capture the distributional uncertainty is distinguishing the original test set from out-of-distribution dataset (Hendrycks & Gimpel, 2017). There are works that focus only on this type of uncertainty (Lee et al., 2018). ODIN (Liang et al., 2018) does not require changing already existing network and relies on gradient-based input preprocessing. Another work (DeVries & Taylor, 2018) is close to the functionality of our method, as it only adds a single densely connected layer and uses a single forward pass for a sample.

Bayesian neural networks are more computationally demanding as they usually require multiple stochastic passes and/or additional parameters to capture the priors specification.

To the best of our knowledge, our method is the first that improves upon the baseline, and meets all the following criteria:

- No additional learnable parameters required.
- Only single forward pass needed.
- No additional out-of-distribution or adversarial observations required.
- No input preprocessing.

The technique we use, Inhibited Softmax, has been successfully used for the prediction of background class in the task of extraction the objects out of aerial imagery (Saito et al., 2016). The original work does not mention other possible applications of this softmax modification.

3 Inhibited Softmax

In this section we will define the Inhibited Softmax function. We will provide mathematical rationale on why it can provide uncertainty estimation when used as the output function of a machine learning model. Later we will present several adjustments which we have made to the model architecture when applying Inhibited Softmax to a multilayer neural network.

Inhibited Softmax function IS_c is given by:

Let $x \in \mathbb{R}^n$ and $c \in \mathbb{R}$, then IS_c is a function which maps $\mathbb{R}^n$ to $\mathbb{R}^n$. The i-th output is equal to:

$$IS_c(x)_i = \frac{\exp x_i}{\sum_{i=1}^n \exp x_i + \exp c} \in (0, 1). \tag{1}$$

Following equation holds:

$$IS_c(x)_i = S(x)_i P_c(x), \tag{2}$$

where:

$$P_c(x) = \frac{\sum_{i=1}^n \exp x_i}{\sum_{i=1}^n \exp x_i + \exp c} \in (0, 1). \tag{3}$$

and $S(x)$ is the standard softmax function applied to vector x. We will later refer to $P_c(x)$ as ”certainty factor”.

Now let’s assume that IS_c is the output of a multiclass classification model trained with the *cross-entropy* loss function l_{IS}. Assuming that the true class of a given example is equal to t the loss is equal to:

$$l_{IS}(x,t) = -\log IS_c(x)_t = -\log S(x)_t - \log P_c(x) = l_S(x,t) - \log P_c(x), \tag{4}$$

where l_S is the *cross-entropy* loss function for a model with a standard softmax output. As one may see - the optimisation process both minimises classification error (given by l_S) and maximises the certainty factor $P_c(x)$ for all training examples. This is the intuition that explains why Inhibited Softmax serves as an uncertainty estimator - as $P_c(x)$ is maximised only for cases from training distribution.

If P_c estimates the certainty of the model, in order to provide a valid uncertainty score we will introduce the following function:

$$P_u(x) = 1 - P_c(x) = \frac{\exp c}{\sum_{i=1}^n \exp x_i + \exp c}, \tag{5}$$

which is minimised during the optimisation process. It is worth to mention that it might be interpreted as an artificial softmax output from the additional channel.

3.1 Adjustments and Regularisation

Although P_u is minimised during the optimisation process we would like to ensure that its low values are obtained solely because of the training process and neither because of the trivial solutions nor accidental network structure. Because of that we applied the following network adjustments:

- removing bias terms from the inhibited softmax layer. This was done in order to prevent the network from minimising $P_u(x)$ that can be achieved by increasing the values of bias terms which are independent of data.
- changing the activation function to a kernel function in the penultimate layer of the network. The main aim of this adjustment was to make activations of the layer noticeably greater from 0 only for a narrow, learnable region in the input space of the penultimate network layer. Therefore, the activations of that layer corresponding to out-of-domain examples are likely to be close to 0, which, combined with the lack of bias term, results in vanishing input to IS.

In order to combat the overconfidence of the network we:

- add the activity regularisation. Standard softmax classification is invariant to translation along the all-ones vector. On the other hand, $-\log P_c$ is a decreasing function of x_i. As l_{is}

is a sum of standard classification error and $-\log P_c$ increasing all values of x_i by a constant decreases the loss, which causes gradient optimisation methods to increase x_i boundlessly. In order to address this issue we introduced the following regularisation method:

$$l'_{IS}(x,t) = l_{IS}(x,t) + \lambda \sum_{i=1}^{n} x_i, \tag{6}$$

where the gradient of the additional term is parallel to the all-ones vector and thus does not affect the standard softmax classification loss.

- apply l_2 regularisation to the weights of the output layer. It indirectly limits the values of x_i, as we use a bounded activation function.

These adjustments significantly increased the certainty estimation properties of Inhibited Softmax. The dependency between performance and applying these changes to model architecture is presented in Appendix 1.

4 EXPERIMENTS

We have compared various ways of estimating uncertainty in neural networks (hereinafter referred to as ”methods”). For the benchmarks we implement these methods on top of the same *base* neural network. We use following experiments to check their quality:

- Out-of-distribution (OOD) examples detection - following (Hendrycks & Gimpel, 2017) we use ROC AUC and average precision (AP) metrics to check the classifier’s ability to distinguish between the original test set and a dataset coming from another probability distribution. This experiments show whether the method measures well the distributional uncertainty on a small sample of out-of-distribution datasets.
- Predictive performance experiment - given a dataset, we split it into train, test and validation sets. We report accuracy and negative log loss on the test set. Any method should not deteriorate predictive performance of the network.
- Wrong prediction detection - we expect that the more confident the model is, the more accurate its predictions on in-distribution dataset should be. In this experiment the ground truth labels are used to construct two classes after the prediction on the test dataset is performed. The classes represent the correctness of the classifier prediction. Then, the uncertainty measure is used to compute TPRs and FPRs. We report ROC AUC scores on this setting. This experiment shows whether the method measures well the combination of epistemic and aleatoric uncertainty on a small sample of datasets. In this experiment we do not report average precision score, as it would be distorted by different levels of misclassification in the predictions.

Method	**Uncertainty measure**	**Abbreviation**
Inhibited Softmax	probability of the artificial softmax output	IS
Base network	$1 - max(p_i)$	BASE
Base network	entropy of the probabilities	BASEE
Monte Carlo Dropout (Gal & Ghahramani, 2016)	predictive entropy of the probabilities from 50 stochastic forward passes	MCD
Bayes By Backprop with a Gaussian prior (Blundell et al., 2015)	predictive entropy of the probabilities from 10 stochastic forward passes	BBP
Deep Ensembles without adversarial training (Lakshminarayanan et al., 2017)	predictive entropy of the probabilities from 5 base neural networks	DE

Table 1: methods used for benchmarks. Both the base network methods will serve as baselines.

In-distribution dataset	**Out-of-distribution datasets**	**Base network**
CIFAR-10 (Krizhevsky, 2009)	SVHN (Netzer et al., 2011) LFW-A (Learned-Miller et al., 2015)	Custom small network trained with Adadelta (Zeiler, 2012)
MNIST	NOTMNIST (Bulatov, 2011) black and white CIFAR-10 Omniglot (Lake et al., 2015)	Lenet-5 (Lecun et al., 1998) with an average pooling instead of a subsampling and a softmax layer instead of a gaussian connection trained with Adadelta (Zeiler, 2012)
IMDB (Maas et al., 2011)	Customer Reviews (Hu & Liu, 2004) Movie Reviews (Pang & Lee, 2004) Reuters-21578	Linear classifier on top of an embedding (as in Hendrycks & Gimpel, 2017) trained with RMSProp (Tieleman & Hinton, 2012)

Table 2: Datasets and neural architectures used.

Table 1 shows the methods and respective uncertainty measures that will be benchmarked[1]. We establish two baselines. Both of them work on the unmodified base neural network, but uncertainty is measured in different ways, using either the maximum of probabilities over classes or entropy of probabilities. The method we suggest to use is referred as *IS*.

We have chosen these methods as they have been already used for benchmarking (e.g. Louizos & Welling, 2017), and they are well-known in the Bayesian Neural Network community. In the case of Inhibited Softmax we set *l2* penalty to 0.01, activity regularisation to 10^{-6}, c to 1 and we use rescaled Cauchy distribution's PDF ($f(x) = \frac{1}{1+x^2}$). The datasets[2] and the respective base neural networks we have chosen for the experiments are reported in Table 2.

The base network for CIFAR-10 consists of 3 2D convolutional layers with 2D batch norm and 0.25 dropout. The convolving filter size was 3. Each convolutional layer was followed by 2D maximum pooling over 3x3 neurons with stride 2. The number of filters in the consecutive layers are 80, 160 and 240. Then there are 3 fully-connected layers. After the first fully-connected layer we apply 0.25 dropout. The number of neurons in the consecutive dense layers are 200, 100, 10.

In the experiments we report averages over three training and prediction procedures on the same training-test splits.

In computer vision OOD tasks, Inhibited Softmax improves upon baselines with an exception of the task of discriminating *MNIST* from black and white *CIFAR-10* (Table 3). Our method still achieves very high detection performance (0.996 ROC AUC and 0.999 AP). This dataset is the least similar to *MNIST*. In contrast to other tested datasets against the digit recognition networks, various shades of gray dominate the images. IS is better than BASE on *NOTMNIST* (0.977 ROC AUC vs 0.958) and Omniglot (0.97 ROC AUC vs 0.956). IS' ROC AUC performance on *MNIST/NOTMNIST* and *CIFAR-10/SVHN* is similar to MCD (resp. 0.977 vs 0.974 and 0.923 vs 0.927). IS achieves the best result on *CIFAR-10/LFW-A* task and all the methods vastly outperform the baselines.

Inhibited Softmax improves upon other methods on the sentiment analysis task. Especially large improvement can be observed on the test against the *Movie Reviews* dataset. For example, the ROC AUC of IS (0.875) is much greater than ROC AUC of MCD (0.836). Methods other than IS are not much better than the baseline (BBP's 0.845 ROC AUC), sometimes being worse (DE's 0.835 ROC AUC). IS is also the best on the test against *Reuters-21578* and *Customer reviews* (resp. 0.822 and 0.731). Two baselines achieve the same results on sentiment analysis experiment as there is no difference in ranking of the examples between the chosen uncertainty measures. We do not corroborate the results from the baseline publication (Hendrycks & Gimpel, 2017). We discovered that in that paper the out-of-distribution samples for *Movie Reviews* were constructed by taking single lines from the dataset file, while the reviews span over few lines. Our results show that the

[1] The choice of hyperparameters and training details for methods other than Inhibited Softmax is further discussed in the appendix

[2] Preprocessing is discussed in the appendix

Datasets (In/Out)	**Score**	**MCD**	**IS**	**BASE**	**BASEE**	**BBP**	**DE**
MNIST/ NOTMNIST	ROC	0.974	0.977	0.958	0.956	0.982	0.979
	AP	0.984	0.982	0.938	0.955	0.989	0.988
MNIST/ CIFAR-10 B&W	ROC	0.999	0.996	0.997	0.996	0.999	0.999
	AP	0.9997	0.9992	0.9994	0.999	0.9997	0.9997
MNIST/ Omniglot	ROC	0.977	0.97	0.956	0.953	0.975	0.977
	AP	0.992	0.99	0.983	0.981	0.991	0.99
CIFAR-10/ SVHN	ROC	0.927	0.923	0.866	0.865	0.913	0.946
	AP	0.987	0.984	0.961	0.958	0.981	0.99
CIFAR-10/ LFW-A	ROC	0.693	0.775	0.593	0.594	0.723	0.755
	AP	0.142	0.194	0.127	0.126	0.169	0.181
IMDB/ Customer Reviews	ROC	0.723	0.731	0.717	0.717	0.729	0.718
	AP	0.027	0.088	0.027	0.027	0.028	0.026
IMDB/ Movie Reviews	ROC	0.836	0.875	0.837	0.837	0.845	0.835
	AP	0.755	0.875	0.756	0.756	0.769	0.753
IMDB/ Reuters-21578	ROC	0.817	0.822	0.816	0.816	0.805	0.815
	AP	0.735	0.818	0.727	0.727	0.715	0.724

Table 3: Out of distribution detection results. The green colour shows the best results, the red - results worse than any of the baselines.

		MCD	**IS**	**BASE(E)**	**BBP**	**DE**
MNIST	Accuracy	0.992	0.992	0.992	0.991	0.994
	NLL	0.034	0.028	0.035	0.031	0.019
CIFAR10	Accuracy	0.854	0.853	0.851	0.841	0.88
	NLL	0.527	0.668	1.661	0.514	0.385
IMDB	Accuracy	0.883	0.882	0.883	0.882	0.885
	NLL	0.291	0.305	0.295	0.302	0.289
IMDB model on Movie Reviews	Accuracy	0.848	0.851	0.857	0.849	0.851
	NLL	0.378	0.345	0.362	0.365	0.586
Number of forward passes		50	1	1	10	1
Params (vs BASE)		x	~x (no bias in last layer)	x	~2x	5x

Table 4: Predictive performance experiment results and computational overhead. Only the baseline and Inhibited Softmax have neither additional parameters nor require multiple forward passes. The green color shows the best results, the red - results worse than the baseline. We compare Inhibited Softmax and baselines with methods that require more forward passes and/or more params.

detection is a tougher task when full reviews are used (BASE achieves 0.837 ROC AUC vs 0.94 ROC AUC (Hendrycks & Gimpel, 2017)).

To understand where the improvements of IS in the sentiment OOD tasks come from, we trained the base network with the same regularisation consisting of *l2* penalty on weights and the activity regularizer. Such an improved baseline achieved 0.853 on *IMDB/Movie Reviews* and 0.838 on *IMDB/Reuters-21578*. Both results are better than all the methods but IS, with the latter improving also upon IS. On the other hand, this enhanced baseline did not improve on *IMDB/Customer Reviews* achieving 0.712 ROC AUC.

In our experiments Inhibited Softmax does not deteriorate the predictive performance of the neural network (Table 4). Its accuracy was similar to the baselines on every task, for example on *IMDB* dataset the accuracy is 0.04% lower and on *CIFAR-10* 0.19% higher. Ensembling the networks gives the best predictive performance. We observed that text models perform very well on *Movie Reviews* dataset. Despite coming from a different probability distribution the latter dataset contains strong sentiment retrieved by the networks for the prediction of the correct label.

Wrong prediction detection results (Table 5) show that IS is the only method that is able to detect misclassified observations better than a random classifier (0.687 ROC AUC) on the sentiment task.

Dataset	**MCD**	**IS**	**BASE**	**BASEE**	**BBP**	**DE**
MNIST	0.982	0.982	0.979	0.979	0.979	0.987
CIFAR-10	0.869	0.855	0.875	0.877	0.878	0.886
IMDB	0.418	0.687	0.501	0.501	0.398	0.391

Table 5: Wrong prediction detection results (ROC AUC). The green color shows the best results, the red - results worse than any of the baselines.

The baseline trained with an activity regularization and l2 penalty does not improve much achieving 0.516 ROC AUC. All the methods improve slightly over the baselines on the *MNIST* dataset with DE improving the most (0.987) and *BBP* improving the least (0.979). The Inhibited Softmax and Monte Carlo Dropout are worse than the baseline on *CIFAR-10* (resp. 0.869 and 0.855 vs 0.875).

5 VISUALISATION

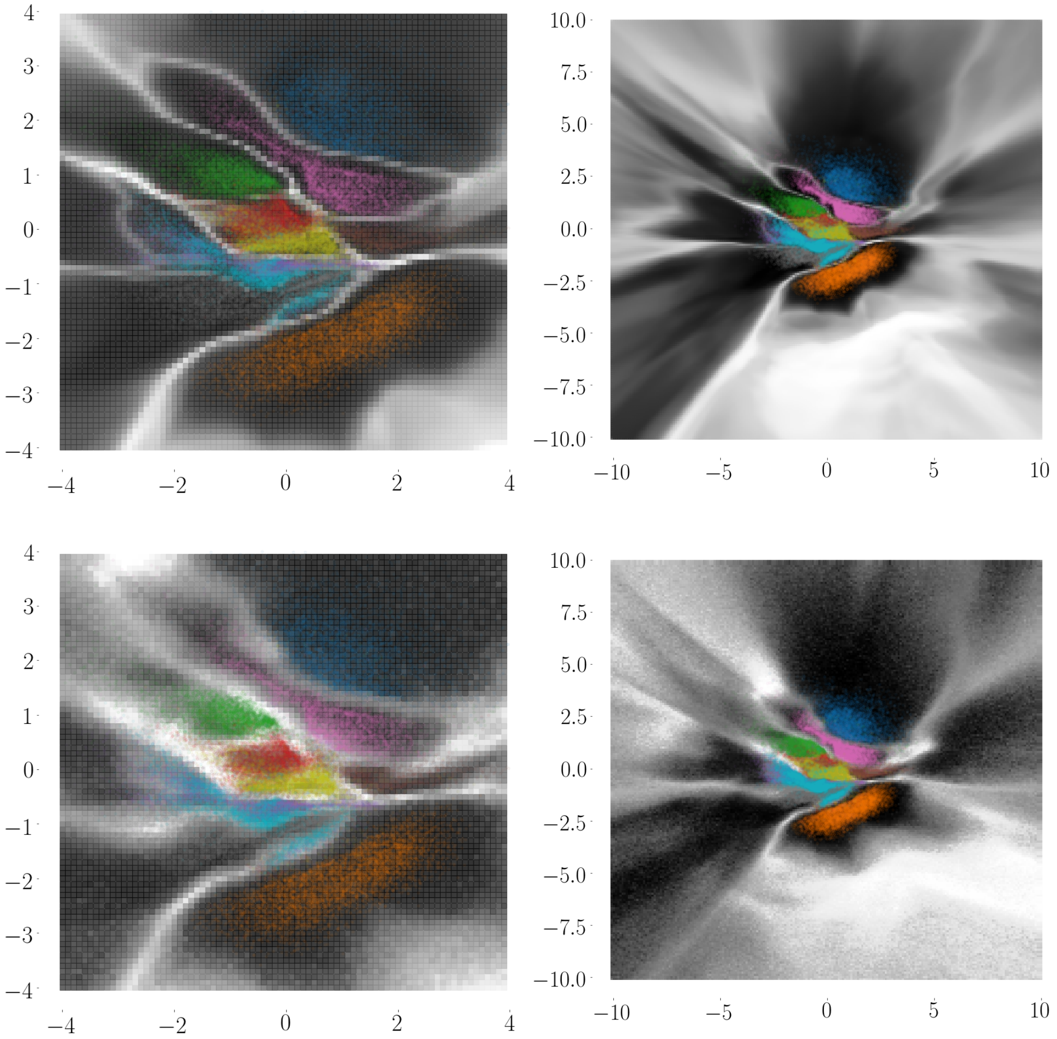

Figure 1: Visualisation of uncertainty measures: Inhibited Softmax (top) and Monte Carlo Dropout (bottom) on the VAE's latent space. The shade of grey represent the normalised uncertainty on the samples generated from the latent space. The lighter the more uncertainty. The points represent the encoded test set and the colours are the classes. The axes show coordinates in the latent space. Note the similarity in the regions between the methods.

In practice, the overlap of the correctly detected out-of-distribution observations between Inhibited Softmax and Bayesian methods is surprisingly large. To demonstrate it, we compare Monte Carlo dropout and our method on an experiment from (Smith & Gal, 2018). We train a fully connected variational autoencoder (VAE) on the *MNIST* dataset. Then, we create a grid in the latent space and for each point we generate a sample. We plot the uncertainty estimation of the methods on generated samples from these points together with labelled latent encoding of the test samples (Figure 1). Both methods are unable to detect out of distribution samples generated from the bottom left corner of the 2D latent space. Another example for similarity is that both of the methods do not estimate high uncertainty in area where blue and purple classes intersect in the latent space. This leads to a hypothesis that there exist samples that are tougher to detect by uncertainty measures for any recently proposed method. Similarly to the ideas from adversarial attacks field, it might be worth to investigate how to construct such samples. We believe it might be a way to improve uncertainty sampling performance.

6 Further Work & Limitations

We notice that working on following aspects can enhance the uncertainty estimation:

- Developing an analogous to IS method for regression.
- Limiting the number of required hyperparameters for Inhibited Softmax.
- Expanding the method to hidden layers. This is especially promising as the Inhibited Softmax performs better than other methods on a shallow network in our sentiment analysis experiment. On deeper networks IS has not have yet such advantage and it might be possible to outperform other methods.

Although we showed by experiments that the architecture adjustments applied to the network architecture are beneficial, we are still lacking the full and sound mathematical explanation of their influence on model behaviour. Such analysis could result in both better procedure for setting Inhibited Softmax hyperparameters as well as new adjustments to the network structure.

7 Conclusion & Discussion

We presented a new method for uncertainty estimation - Inhibited Softmax. The method can be easily applied to various multilayer neural network architectures and does not require additional parameters, multiple stochastic forward passes or OOD examples.

The results show that the method outperforms baseline and performs comparably to the other methods. The method does not deteriorate predictive performance of the classifier.

The predictive performance from *IMDB/Movie Reviews* experiment suggests that even if the observation comes from another probability distribution and the uncertainty measure is able to detect it, the network can still serve as a useful classifier.

The improvement of the baseline on the sentiment task after adding suggested regularisation indicates it might be worth to apply such measures to other uncertainty estimation methods.

References

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 1613–1622, 2015.

Yaroslav Bulatov. notMNIST dataset. `http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html`, Aug 2011.

Wray Buntine and Andreas Weigend. Bayesian back-propagation. *Complex Systems*, 5:603–643, 1991.

Sophie Curtis. Google photos labels black people as 'gorillas'. https://www.telegraph.co.uk/technology/google/11710136/Google-Photos-assigns-gorilla-tag-to-photos-of-black-people.html, Jul 2015.

Terrance DeVries and Graham W. Taylor. Learning Confidence for Out-of-Distribution Detection in Neural Networks. *ArXiv e-prints*, 2018.

Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on machine learning models. *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, abs/1707.08945, 2018.

Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *ArXiv e-prints*, 2015.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, 2016.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330, 2017.

Danijar Hafner, Dustin Tran, Alex Irpan, Timothy P. Lillicrap, and James Davidson. Reliable uncertainty estimates in deep neural networks using noise contrastive priors. *CoRR*, abs/1807.09289, 2018.

Dan Hendrycks and Kevin Gimpel. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *International Conference on Learning Representations*, 2017.

José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 1861–1869, 2015.

N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel. Bayesian Active Learning for Classification and Preference Learning. *ArXiv e-prints*, December 2011.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *KDD*, 2004.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems 30*, pp. 5574–5584. Curran Associates, Inc., 2017.

Mohammad Emtiyaz Khan, Didrik Nielsen, Voot Tangkaratt, Wu Lin, Yarin Gal, and Akash Srivastava. Fast and scalable bayesian deep learning by weight-perturbation in adam. In *ICML*, 2018.

Anoop Korattikara Balan, Vivek Rathod, Kevin P Murphy, and Max Welling. Bayesian dark knowledge. In *Advances in Neural Information Processing Systems 28*, pp. 3438–3446. Curran Associates, Inc., 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. doi: 10.1126/science.aab3050.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30*, pp. 6402–6413. Curran Associates, Inc., 2017.

Erik G. Learned-Miller, Gary Huang, Aruni RoyChowdhury, Haoxiang Li, and Gang Hua. Labeled faces in the wild : A survey. 2015.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. ISSN 0018-9219. doi: 10.1109/5.726791.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018.

Christian Leibig, Vaneeda Allken, Murat Seckin Ayhan, Philipp Berens, and Siegfried Wahl. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific Reports*, 7, 12 2017.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

C. Louizos and M. Welling. Multiplicative Normalizing Flows for Variational Bayesian Neural Networks. *ArXiv e-prints*, 2017.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, June 2011.

A. Malinin and M. Gales. Predictive Uncertainty Estimation via Prior Networks. *ArXiv e-prints*, 2018.

Amit Mandelbaum and Daphna Weinshall. Distance-based confidence score for neural network classifiers. *CoRR*, abs/1709.09844, 2017.

Radford M. Neal. Bayesian learning via stochastic dynamics. In S. J. Hanson, J. D. Cowan, and C. L. Giles (eds.), *Advances in Neural Information Processing Systems 5*, pp. 475–482. Morgan-Kaufmann, 1993.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity. In *Proceedings of ACL*, pp. 271–278, 2004.

Matthias Rottmann, Karsten Kahl, and Hanno Gottschalk. Deep bayesian active semi-supervised learning. *CoRR*, abs/1803.01216, 2018.

Shunta Saito, Takayoshi Yamashita, and Yoshimitsu Aoki. Multiple object extraction from aerial imagery with convolutional neural networks. *Electronic Imaging*, 2016(10):1–9, feb 2016. doi: 10.2352/issn.2470-1173.2016.10.robvis-392. URL `https://doi.org/10.2352/issn.2470-1173.2016.10.robvis-392`.

Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. In *34th Conference on Uncertainty in Artificial Intelligence (UAI)*, 03 2018.

Mattias Teye, Hossein Azizpour, and Kevin Smith. Bayesian uncertainty estimation for batch normalized deep networks. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 4907–4916, 2018.

T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

Hao Wang, Xingjian SHI, and Dit-Yan Yeung. Natural-parameter networks: A class of probabilistic neural networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 118–126. Curran Associates, Inc., 2016.

Kuan-Chieh Wang, Paul Vicol, James Lucas, Li Gu, Roger Grosse, and Richard S. Zemel. Adversarial distillation of bayesian neural network posteriors. In *ICML*, 2018.

Matthew D. Zeiler. Adadelta: An adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

Guodong Zhang, Shengyang Sun, David K. Duvenaud, and Roger B. Grosse. Noisy natural gradient as variational inference. In *ICML*, 2018.

Appendix 1 - Ablation Study

We show the performance of our methods in the experiments on *CIFAR-10* and *MNIST* datasets if the hyperparameters are changed (Figure 2). The results are averages over three runs of experiments. Without *l2* penalty or with too strong a penalty (e.g. 0.1) the performance in terms of accuracy on *CIFAR-10*, wrong prediction detection on *CIFAR-10* and out-of-distribution detection on *CIFAR-10/SVHN* deteriorates. Moreover, without *l2* penalty, the network performs worse on OOD detection on *MNIST/NOTMNIST* and *MNIST/CIFAR-10* tasks. Similarly, the activity regularizer penalty is important. The networks without it performed worse on all the checked tasks with an exception of OOD detection on *CIFAR-10/SVHN*. With too much of the regularization, the networks are unable to fit the data well. It results in a sharp drop in results of all experiments on *CIFAR-10*. We show also that it is possible to replace the rescaled Cauchy PDF function with another kernel function. Here, we show a comparison with rescaled Gaussian PDF ($\exp\frac{-x^2}{2}$) and a custom nonlinear function:

$$f(x) = \begin{cases} min(x+1, -x+1), & \text{if } |x| < 1 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Still, non-kernel activation functions like ReLU do not perform well.

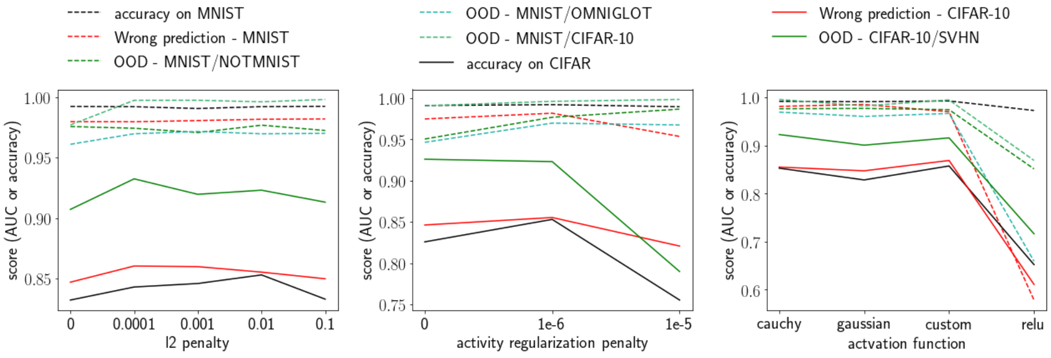

Figure 2: Results of ablation experiments. The plots show the wrong prediction experiment's ROC AUC, out of-distribution experiment's ROC AUC and accuracies on *MNIST* and *CIFAR* datasets. We check the performance with changed *l2* penalty (left), changed activation function (middle) and changed activity regularization penalty (right).

Appendix 2 - Experiments details & Preprocessing

Omniglot consists of black letters on white background. We negated the images so that they resemble more the images from *MNIST*. Without the negation, all the methods performed very well (between 0.999 and 1 in ROC AUC) on the out-of-distribution detection task.

In the sentiment analysis task, before feeding the data to the networks we preprocessed it by removing stopwords and words that did not occur in the pretrained embedding. We use a pretrained embedding in order to model vocabulary that exists in the OOD sets and was not present in the in-distribution dataset.

Regarding the baseline publication (Hendrycks & Gimpel, 2017): we were able to corroborate the results on *IMDB/Movie Reviews* experiment when we split the observations from *Movie Reviews* into single lines and use the same randomly initialized embeddings. The model was trained on full reviews from *IMDB*. We argue that in such setting the use of average pooling after the embedding invalidates the experiment. The input is padded with zeros to 400 words. Now, if the sentence is very short, say 10 words, the true average of the embed words will be diminished by all the zeros after the sentence. Thus, the uncertainty estimation method needs only to correctly work in a very narrow region centred at zero in order to achieve high scores in the experiment.

For the state-of-the art methods we compared with we made following choices:

- Deep Ensembles - we skipped adversarial training, as adversarial training is a way to improve performance of any of the methods used in the paper. We use an ensemble of 5 base networks.
- Monte Carlo Dropout - for *MNIST* we use dropout probability 0.25 on all but last layers, 0.5 on the only trainable layer in the sentiment experiment, and on *CIFAR-10* network 0.25 only on the last but one layer. In larger networks setting dropout on many layers required greater number of epochs to achieve top performance. We run 50 forward passes for variational prediction.
- Bayes By Backprop - we observed that there is a trade-off between accuracy and OOD detection performance that depends on the initialisation of the variance. We chose initialisation that led to the best combination of accuracy and OOD detection performance in our view. We run 10 forward passes for variational prediction.

We followed the original publications when possible. For example, the number of networks in DE and number of inferences in BBP and MCD are taken from the original descriptions of the algorithms.

Appendix 3 - Visualization

In the visualisation section of the paper the uncertainties were normalised so that the predictive entropy and IS' probabilities could be visually compared. The normalisation for a method was performed by ranking the uncertainties and splitting them into 400 equal bins. Then, the bins are plotted. White colour represents the bin with the most uncertainty, the black - with the least.

For better understanding of the latent space we visualise the images decoded from the grid from the latent space (Figure 3).

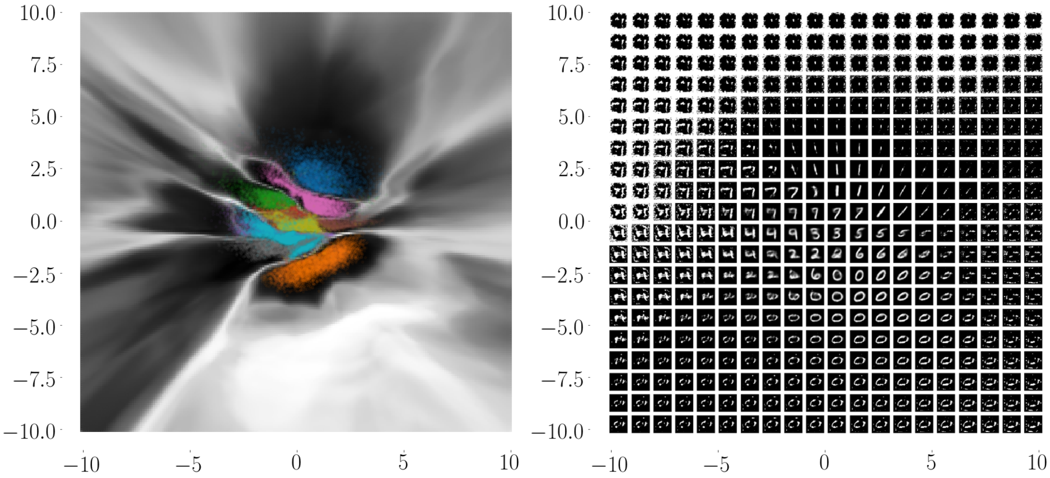

Figure 3: Visualisation of the images generated from the grid in the latent space (right) next to the uncertainty measure visualisation (left).