Reliable Uncertainty Estimates in Deep Neural Networks using Noise Contrastive Priors

Danijar Hafner [1] **Dustin Tran** [1] **Alex Irpan** [1] **Timothy Lillicrap** [2] **James Davidson** [1]

Abstract

Obtaining reliable uncertainty estimates of neural network predictions is a long standing challenge. Bayesian neural networks have been proposed as a solution, but it remains open how to specify the prior. In particular, the common practice of a standard normal prior in weight space imposes only weak regularities, causing the function posterior to possibly generalize in unforeseen ways on out-of-distribution inputs. We propose noise contrastive priors (NCPs). The key idea is to train the model to output high uncertainty for data points outside of the training distribution. NCPs do so using an input prior, which adds noise to the inputs of the current mini batch, and an output prior, which is a wide distribution given these inputs. NCPs are compatible with any model that represents predictive uncertainty, are easy to scale, and yield reliable uncertainty estimates throughout training. Empirically, we show that NCPs offer clear improvements as an addition to existing baselines. We demonstrate the scalability on the flight delays data set, where we significantly improve upon previously published results.

1. Introduction

Many successful applications of neural networks (Krizhevsky et al., 2012; Sutskever et al., 2014; van den Oord et al., 2016) are in restricted settings where predictions are only made for inputs similar to the training distribution. In real-world scenarios, neural networks can face truly novel data points during inference, and in these settings it can be valuable to have good estimates of the model's uncertainty. For example, in healthcare, reliable uncertainty estimates can prevent overconfident decisions for rare or novel patient conditions (Schulam and Saria, 2015). Similarly, autonomous agents that actively explore their environment can use uncertainty estimates to decide what data points will be most informative (MacKay, 1992a).

Uncertainty describes the degree of missing knowledge about the data generating function. This uncertainty can in principle be completely reduced by observing more data points at the right locations and training on them. In contrast, the data generating function may also have inherent randomness, which we call data noise. This noise can be captured by models outputting a distribution rather than a point prediction. Obtaining more data points will move the noise estimate closer to its true value, which is usually different from zero.[1] For active learning, it is crucial to separate the two types of randomness: we want to acquire labels in regions of high uncertainty but low noise.

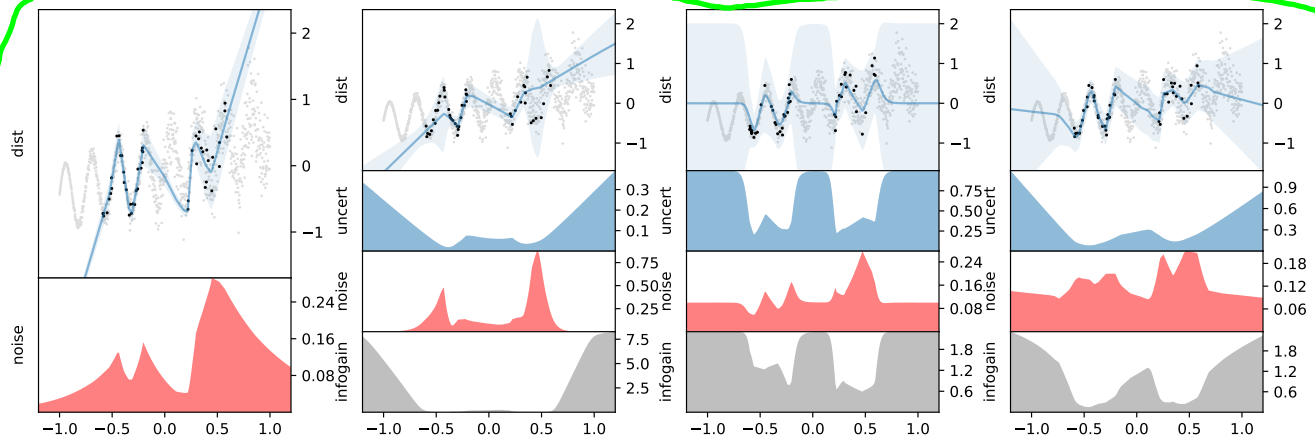
Bayesian analysis provides a principled approach to modeling uncertainty in neural networks (Denker et al., 1987; MacKay, 1992b). Namely, one places a prior over the network's weights and biases. This effectively places a distribution over the functions that the network can represent, capturing uncertainty in which function best fits the data. Unfortunately, specifying the prior remains an open challenge. Common practice is to use a standard normal prior in weight space, which imposes weak shrinkage regularities analogous to weight decay. It is neither informative about the induced function class nor the data (e.g., it is sensitive to parameterization). This can cause the induced function posterior to generalize in unforeseen ways on out-of-distribution (OOD) inputs.

The most well-studied function priors, Gaussian processes (GPs), may offer a path toward improving neural network priors. Namely, GPs can be understood as the limit of infinitely wide networks (Neal, 1994; Matthews et al., 2018; Lee et al., 2018). This may inspire finite weight priors that mimic GP properties. However, GPs are not necessarily a gold standard, as the infinite network analogy has limitations: it only holds for feedforward networks, correlations between output units vanish, and all hidden units contribute infinitesimally to all inputs (i.e., there are no salient features)

[1]Google Brain [2]DeepMind. Correspondence to: Danijar Hafner <mail@danijar.com>.

Presented at the ICML 2018 workshop on *Theoretical Foundations and Applications of Deep Generative Models*, Stockholm, Sweden.

[1]What we describe as uncertainty is sometimes referred to as epistemic uncertainty. The data noise is then referred to as aleatoric uncertainty.



(a) Det (b) BBB (c) Det + NCP (d) BBB + NCP

Figure 1: Comparison of predictive distributions on a low-dimensional active learning task. The predictive distributions are visualized as mean and two standard deviations shaded. They decompose into predictive uncertainty (blue) and predicted data noise (red). Data points (black) are only available within two bands, and are selected using the expected information gain (grey). **(a)** A deterministic network conflates model uncertainty as part of data noise and is overconfident outside of the data distribution. **(b)** A variational Bayesian neural network separates model uncertainty but remains overconfident. **(c)** On the OOD classifier model, NCP prevents overconfidence. **(d)** On the Bayesian neural network, NCP produces smooth uncertainty estimates that generalize well to unseen data points.

(Neal, 1994; MacKay, 1998; Der and Lee, 2006).

A different path to selecting network priors is in existing regularization techniques. However, these typically act on the hidden units or weights—such as weight decay, dropout (Srivastava et al., 2014), and batch normalization (Ioffe and Szegedy, 2015)—and they focus on generalizing predictions from finite data rather than uncertainty on OOD inputs.

Promising recent work applies a classifier to detect OOD samples; unlike the above, OOD detection acts on the function and lets the network avoid overconfident predictions (Hendrycks and Gimpel, 2016; Lee et al., 2017; Liang et al., 2018). The main challenge with this approach is to efficiently generate OOD training examples for the classifier: it must generate examples from the complement of the training distribution, which is an ill-defined problem.

Motivated by these lines of work and their challenges, this paper makes several contributions:

- We develop *noise contrastive priors (NCPs)*, a prior for neural networks in data space that encourages network weights to both explain the training data and capture high uncertainty on OOD inputs. Unlike priors in weight space, data priors let one easily express informative assumptions about how the input and output of the model should be related.

- We develop a simple strategy for training neural networks with noise contrastive priors: add noise to a mini-batch's inputs and train the network to output high uncertainty by minimizing the KL-divergence to a wide prior distribution. NCP is compatible with any model that represents predictive uncertainty, is easy to scale, and yields reliable uncertainty estimates throughout active data acquisition.

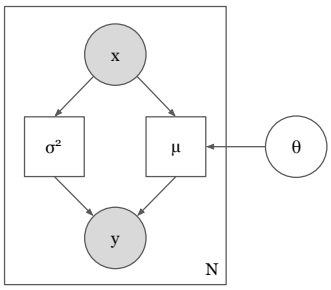
- We apply NCPs to active learning, and find that NCPs provide clear improvements when added to existing baselines. On a large flight delays data set, we significantly improve upon state of the art by 0.5 nats, showing the scalability of NCPs.

2. Noise Contrastive Priors

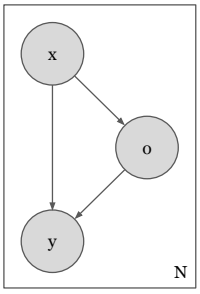
Specifying priors is intuitive for small probabilistic models, since each variable typically has a clear interpretation (Blei, 2014). It is less intuitive for neural networks, where the parameters serve more as adaptive basis coefficients in a nonparametric function. For example, neural network models are non-identifiable due to weight symmetries that can produce the same function output (Müller and Insua, 1998). This complicates placing priors on specific weights, making it difficult to express priors such as high uncertainty on unfamiliar examples. We investigate an alternative prior based in data space.

2.1. Data priors

Unlike a prior in weight space, a *data prior* lets one easily express informative assumptions about input-output relationships. Here, we use the example of a prior over a labeled



(a) Bayesian neural network



(b) Out-of-distribution classifier

Figure 2: Graphical representations of the two uncertainty-aware models that we study. Circles denote random variables, squares denote deterministic variables, shading denotes observations during training. **(a)** The Bayesian neural network captures a belief over parameters for the predictive mean, while the predictive variance is a deterministic function of the input. In practice, we only use weight uncertainty for the mean's output layer and share earlier layers between the mean and variance. **(b)** The out-of-distribution classifier model uses a binary auxiliary variable o to determine if a given input is out-of-distribution; given its value, the output is drawn from either a neural network prediction or a wide output prior.

data set (x, y), although the prior can also be on x and another variable in the model that represents model uncertainty and has a clear interpretation. The prior takes the form,

$$p_{\text{prior}}(x, y) = p_{\text{prior}}(x)\, p_{\text{prior}}(y \mid x), \tag{1}$$

where $p_{\text{prior}}(x)$ represents the *input prior* and $p_{\text{prior}}(y \mid x)$ denotes the *output prior*.

To achieve reliable uncertainty estimates, a good input prior $p_{\text{prior}}(x)$ should include OOD examples so that we can learn from them, and a good output prior $p_{\text{prior}}(y \mid x)$ should be a wide distribution, representing high uncertainty about the model output.

2.2. Generating OOD inputs

Exactly generating OOD data is difficult. A priori, we must uniformly represent the input domain. A posteriori, we must represent the complement of the training distribution. Both distributions are uniform over infinite support, making them ill-defined. To estimate the OOD inputs, we develop an algorithm that uses the idea of noise contrastive estimation (Gutmann and Hyvärinen, 2010a; Mnih and Kavukcuoglu, 2013), where a complement distribution is approximated using random noise.

The core hypothesis of our work is that it is enough to encourage high uncertainty output near the *boundary* of the training distribution, and that this effect will propagate to the entire OOD space. This hypothesis is backed up by previous work (Lee et al., 2017) as well as our experiments (see Figure 1). This is important because we no longer need to sample arbitrary OOD inputs. It is enough to sample OOD points that lie close to the boundary of the training distribution, and to apply our desired prior at those points.

2.3. Noise contrastive priors

Noise contrastive priors (NCPs) are data-dependent priors where we approximate OOD inputs near the boundary by perturbing training inputs x with noise, $\tilde{x} = x + \epsilon$. By definition, some training inputs lie near the boundary of the training distribution. Adding noise to those inputs gives inputs that tend to be more OOD than the training data:

$$p_{\text{prior}}(x) = p_{\text{train}}(x) + \mathcal{N}(0, \sigma_x^2), \tag{2}$$

$$p_{\text{prior}}(y \mid x) = \mathcal{N}(0, \sigma_y^2). \tag{3}$$

The variances σ_x^2, σ_y^2 are hyperparameters that tune how far from the boundary we sample, and how large we want the output uncertainty to be. In binary and categorical input domains, we approximate OOD inputs by randomly flipping each feature to a different class with a certain probability.

Technically, NCPs add noise to all data inputs rather than manually selecting the subset on the boundary. In our experiments, we found that this does not affect performance: noised-up inputs that remain in the training distribution's support can be seen as a form of label smoothing, and it avoids a potentially sensitive preprocessing step.

For training, we minimize the loss function

$$\begin{aligned}\mathcal{L}(\theta) = \mathbb{E}_{p_{\text{train}}(x)}[\text{KL}(p_{\text{train}}(y \mid x) \,\|\, p(y \mid x, \theta))] \\ + \mathbb{E}_{p_{\text{prior}}(x)}[\text{KL}(p_{\text{prior}}(y \mid x) \,\|\, p(y \mid x, \theta))].\end{aligned} \tag{4}$$

The first term represents typical maximum likelihood, in which one minimizes the KL divergence to the training distribution $p_{\text{train}}(y \mid x)$ over training inputs. The second term represents the analogous term on a data prior.

The noise contrastive prior can be interpreted as inducing a function prior. This is formalized through the predictive

distribution $p(y \mid x)$, which takes the form

$$p(y \mid x) = \int p(y \mid x, \theta)\, p(\theta \mid \tilde{x}, \tilde{y})\, p_{\text{prior}}(\tilde{x}, \tilde{y})\, \mathrm{d}\theta\, \mathrm{d}\tilde{x}\, \mathrm{d}\tilde{y}.$$

The distribution marginalizes over network parameters θ as well as data fantasized from the data prior. The distribution $p(\theta \mid \tilde{x}, \tilde{y})$ represents the distribution of model parameters after fitting the prior data. That is, the belief over weights is shaped to make $p(y \mid x)$ highly variable.

Because network weights are constrained to fit the data prior, the prior acts as "pseudo-data." This is similar to classical work on conjugate priors: a $\text{Beta}(\alpha, \beta)$ prior on the probability of a Bernoulli likelihood implies a Beta posterior, and if the posterior mode is chosen as an optimal parameter setting, then the prior translates to $\alpha - 1$ successes and $\beta - 1$ failures. It is also similar to pseudo-data in sparse Gaussian processes (Quiñonero-Candela and Rasmussen, 2005).

Data priors encourage learning parameters that not only capture the training data well but also the prior data. In practice, we only apply data priors to capture OOD uncertainty, so our network priors are noninformative about in-distribution predictions. We can combine NCP with other priors, for example the typical standard normal prior in weight space for Bayesian neural networks.

Next, we show how to apply NCP to two models, a Bayesian neural network and an OOD classifier model.

2.4. Bayesian Neural Networks with NCP

We model continuous data in which $p(y \mid x, \theta)$ is a Gaussian likelihood whose mean μ and variance σ^2 are predicted by a neural network from the inputs. We use a weight belief for the parameters of only the output layer that predicts the mean,

$$\theta \sim p(\theta) \qquad y \sim \mathcal{N}(\mu(x, \theta), \sigma^2(x)). \tag{5}$$

We place an NCP on the distribution of the mean, $p(\mu(\tilde{x}, \theta \sim q))$, which is induced by the weight belief, giving the loss function

$$\begin{aligned}\mathcal{L} = &-\mathbb{E}_{q_\phi(\theta)}[\log p(y \mid x, \theta)] + \text{KL}(q_\phi(\theta) \,\|\, p(\theta)) \\ &+ \text{KL}\left(\mathcal{N}(0, \sigma_\mu^2) \,\|\, p(\mu(\tilde{x}, \theta \sim q))\right),\end{aligned} \tag{6}$$

where $q_\phi(\theta)$ forms an approximate posterior over weights.[2] We set $\sigma_\mu^2 = 1$ for all regression experiments, which can be seen as an empirical prior for a normalized data set with mean 0 and variance 1, which is standard for many machine learning algorithms.

[2] To derive the loss, set $p(y \mid x, \theta) = \mathbb{E}_{q_\phi(\theta)}[p(y \mid x, \theta)]$ in Equation 4 and apply Jensen's inequality. Note because the likelihood is Gaussian, the KL divergence can be computed analytically.

The loss function applies weight regularization in order for network weights to regress to a standard normal prior; like other regularization techniques, this assists in improving the network's generalization in-distribution. The data regularization encourages the network's generalization OOD by matching the mean distribution to the output prior. Minimizing the KL divergence to a wide output prior results in high uncertainty on OOD inputs, so the model will explore these data points during active learning.

2.5. OOD Classifier Model with NCP

An alternative approach to capture model uncertainty is to use explicit predictions about whether an input is OOD. Figure 2b shows such a mixture model via a binary variable o,

$$\begin{aligned}o &\sim \text{Bernoulli}(\pi(x)) \\ y &\sim \begin{cases}\mathcal{N}(\mu(x), \sigma^2(x)) & \text{if } o = 0 \\ \mathcal{N}(0, \sigma_y^2) & \text{if } o = 1,\end{cases}\end{aligned} \tag{7}$$

where $p(o = 1 \mid x)$ is the OOD probability of x. If $o = 0$ ("in distribution"), the model outputs the neural network prediction. Otherwise, if $o = 1$ ("out of distribution"), the model uses a fixed wide output prior.

During training, x, y, o are all observed as by definition, training data are in-distribution ($o = 0$) and noise-up inputs are assumed to be OOD ($o = 1$). Following Equation 4, the loss is

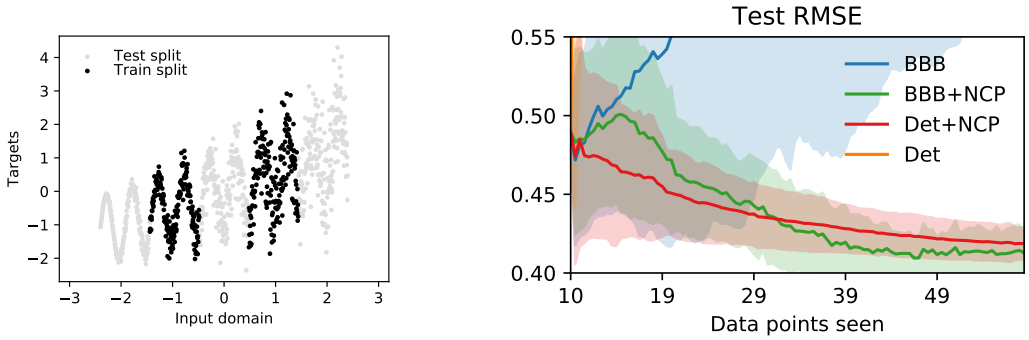
$$\begin{aligned}\mathcal{L} &= -\log p(y, o = 0 \mid x) - \log p(y, o = 1 \mid \tilde{x}) \\ &\overset{+}{=} -\log \mathcal{N}(y \mid \mu(x), \sigma^2(x)) - \log \text{Bernoulli}(0 \mid \pi(x)) \\ &\quad - \log \text{Bernoulli}(1 \mid \pi(\tilde{x})),\end{aligned} \tag{8}$$

where we drop $p(y \mid o = 1, \tilde{x})$ as it is constant with respect to model parameters.

In our experiments, we implement the OOD classifier model using a single neural network with two output layers that parameterize the Gaussian distribution $p(y|o = 0, x)$ and the binary distribution $p(o|x)$, respectively.

3. Related Work

Priors for neural networks. Most recent work in Bayesian neural networks has been in variational inference, either speeding up its training (Graves, 2011; Blundell et al., 2015; Hernández-Lobato and Adams, 2015) or improving its posterior approximation (Louizos and Welling, 2016; Zhang et al., 2017; Krueger et al., 2017). Recent works analyzing priors have been specific to compression and model selection (Ghosh and Doshi-Velez, 2017; Louizos et al., 2017); this is with the exception of Flam-Shepherd et al. (2017), who propose general-purpose weight priors based on approximating Gaussian processes. Classic work has investigated entropic priors (Buntine and Weigend, 1991) and



(a) Data set with structured split

(b) Model performance

Figure 3: Active learning on a 1-dimensional regression problem. Starting from 10 seed labels, the models select one additional label every 1000 epochs. These epochs are small because only few data points are visible to the model. The root mean squared error (RMSE) of the models trained with NCP decreases during the active learning run, while the deterministic and Bayes by Backprop models select less informative data and overfit. The deterministic network is barely visible in the plot as it overfits quickly. Figure 1 shows the predictive distributions of the four models.

hierarchical priors (MacKay, 1992b; Neal, 2012; Lampinen and Vehtari, 2001). Instead of a prior in weight space, NCPs take the functional view by imposing explicit regularities in terms of the network's inputs and outputs.

Input and output regularization. There is classic work on adding noise to inputs for improved generalization (Matsuoka, 1992; An, 1996). For example, Bishop (1995) connects input noise to training with a penalty, and denoising autoencoders (Vincent et al., 2008) encourage reconstructions given noisy encodings. All these methods train the model to make correct predictions in the presence of noise, which improves the model's local interpolation behavior. NCP's intention is the opposite: train the model to predict high uncertainty in the presence of noise, which improves the model's extrapolation behavior.

Output regularization is a classic idea from the maximum entropy principle (Jaynes, 1957), where it has motivated label smoothing (Szegedy et al., 2016) and entropy penalties (Pereyra et al., 2017). Also related is virtual adversarial training (Miyato et al., 2015), which includes examples that are close to the current input but cause a maximal change in the model output, and mixup (Zhang et al., 2018), which minimizes the risk under the viscinity of training data. Again, these methods are orthogonal to NCPs: they aim to improve generalization from finite data (but under the same distribution); our approach aims to improve uncertainty estimates outside of the training distribution.

Classifying in vs out-of-distribution data. A simple approach for neural network uncertainty is to classify whether data points belong to the data distribution, or are OOD (Hendrycks and Gimpel, 2016). This is core to noise contrastive estimation (Gutmann and Hyvärinen, 2010b), where similar ideas are used in natural language processing to approximate softmax losses for large vocabularies (Mikolov et al., 2013). More recently, Lee et al. (2017) introduce a GAN to generate OOD samples. A classifier is then trained to predict whether a data point is inside or outside of the data distribution. Liang et al. (2018) add perturbations to the input, applying an "OOD detector" to improve softmax scores on OOD samples according to a scaled temperature. Extending these directions of research, we connect to Bayesian principles and focus on uncertainty estimates that are useful for active data acquisition.

Active learning. Active learning is often employed in domains where data is cheap but labeling is expensive, and is motivated by the idea that not all data points are equally valuable when it comes to learning (Settles, 2009; Dasgupta, 2004). Active learning techniques can be coarsely grouped into three categories. Ensemble methods (Seung et al., 1992; McCallumzy and Nigamy, 1998; Freund et al., 1997) generate queries that have the greatest disagreement between a set of classifiers. Error reduction approaches incorporate the select data based on the predicted reduction in classifier error based on information (MacKay, 1992a), Monte Carlo estimation (Roy and McCallum, 2001), or hard-negative example mining (Sung, 1994; Rowley et al., 1998).

Uncertainty-based techniques select samples for which the classifier is most uncertain. Approaches include maximum entropy (Joshi et al., 2009), distance from the decision boundary (Tong and Koller, 2001), pseudo labelling high confidence examples (Wang et al., 2017), and mixtures of information density and uncertainty measures (Li and Guo, 2013). Within this category, the area most related to our work are Bayesian methods. Kapoor et al. (2007) estimate expected improvement using a Gaussian process. Other ap-

proaches use classifier confidence (Lewis and Gale, 1994), predicted expected error (Roy and McCallum, 2001), or model disagreement (Houlsby et al., 2011). Recently, Gal et al. (2017) applied a convolutional neural network with dropout uncertainty to images.

4. Experiments

To demonstrate their usefulness, we evaluate NCPs on various tasks where uncertainty estimates are desired. Our focus is on active learning, where only few labels are visible in the beginning, and additional labels are selected regularly based on an acquisition function. We perform active learning experiments on three data sets: a toy example, a large flights data set, and a hyper parameter optimization data set with binary features. Moreover, we show that NCP scales to large data sets by training on the full flights data set in a passive setting. Our implementation uses TensorFlow Probability (Dillon et al., 2017; Tran et al., 2016) and is available at `https://github.com/brain-research/ncp`.

We compare Bayes by Backprop (BBB) (Blundell et al., 2015; Kucukelbir et al., 2017), which is a Bayesian neural network trained via gradient-based variational inference, Bayes by Backprop with NCP (BBB+NCP), and the OOD classifier model with NCP (Det+NCP). All models use leaky ReLU as activation functions (Maas et al., 2013), and are trained using Adam (Kingma and Ba, 2014). We scale down gradients if their norm exceeds 100.

In our active learning experiments, we select new data points by maximizing the expected information gain. We use the approximation for Gaussian posterior predictive distributions described in MacKay (1992a),

$$\mathbb{E}_{p(y|x)}\left[\mathrm{KL}(q(\theta \mid \{x,y\}) \,\|\, q(\theta))\right] \approx \frac{1}{2}\ln\left(1+\frac{\alpha}{\sigma^2}\right), \tag{9}$$

where σ^2 is the expected data noise for the output and α is the variance of the mean, integrating out the parameter belief. We place a softmax distribution on the information gain for all available data points and acquire labels by sampling with a temperature of $\tau = 0.5$ to get diversity when selecting batches of labels at once.

We only model parameter beliefs θ for the output layer that predicts the mean in our Bayesian neural networks (Lázaro-Gredilla and Figueiras-Vidal, 2010; Calandra et al., 2014) as described in Section 2.4. Therefore, $\alpha = \mathrm{Var}(\mu(x,\theta))$ which is Gaussian and can be computed in closed form. Modeling uncertainty only in the mean fits well with the information gain approximation that also only considers uncertainty around the mean. In the classifier model, we use the OOD probability as proxy for $\alpha = p(o=1|x)$.

Table 1: Performance on all 700k data points of the flights data set. While uncertainty estimates are not necessary when a large data set that is similar to the test data set is available, it shows that our method scales easily to large data sets.

Model	NLPD	RMSE
gPoE (Deisenroth & Ng 2015)	8.1	—
SAVIGP (Bonilla et al. 2016)	5.02	—
SVI GP (Hensman et al. 2013)	—	32.60
HGP (Ng & Deisenroth 2014)	—	27.45
MF (Lakshminarayanan et al. 2016)	4.89	26.57
BBB	**4.38**	**24.59**
BBB+NCP	**4.38**	**24.71**
ODC+NCP	**4.38**	**24.68**

4.1. Low-dimensional active learning

For visualization purposes, we start with experiments on a low-dimensional regression data set shown in Figure 3a. Training data can be acquired within two bands, and the model is evaluated on all data points that are not visible to the model. This structured split between training and testing data causes a strong distributional shift at inference time, forcing successful models to have reliable uncertainty estimates to avoid mispredictions for OOD inputs.

We use two layers of 50 hidden units and a learning rate of 3×10^{-4} for all models. NCP models use $\epsilon \sim \mathcal{N}(0, 0.5)$. In addition to the models described above, we compare to a deterministic neural network (Det) that fits a Gaussian to the data and uses the proxy $\alpha = \mathrm{Var}(x)$, since it has no uncertainty estimate. We start with 10 randomly selected initial labels, and select 1 additional label every 1000 epochs. We train on full batches for this task, since the number of data points is small enough.

Figure 3b shows the root mean squared error throughout learning. The two models trained with NCP show improved performance compared to the two baseline models which severely overfit to the training distribution on this small task. Models with NCP outperform BBB, which in turn outperforms the deterministic baseline. Figure 1 visualizes the predictive distributions of the 4 models at the end of learning, and shows that NCP prevents overconfident generalization.

4.2. Active learning on flight delays

We consider the flight delay data set (Hensman et al., 2013; Deisenroth and Ng, 2015; Lakshminarayanan et al., 2016), a large scale regression benchmark with several published results. The data set has 8 input variables describing a flight, and the target is the delay of the flight in minutes. There are 700k training examples and 100k test examples. The test set has a subtle distributional shift, since the 100k data points temporally follow after the training data.

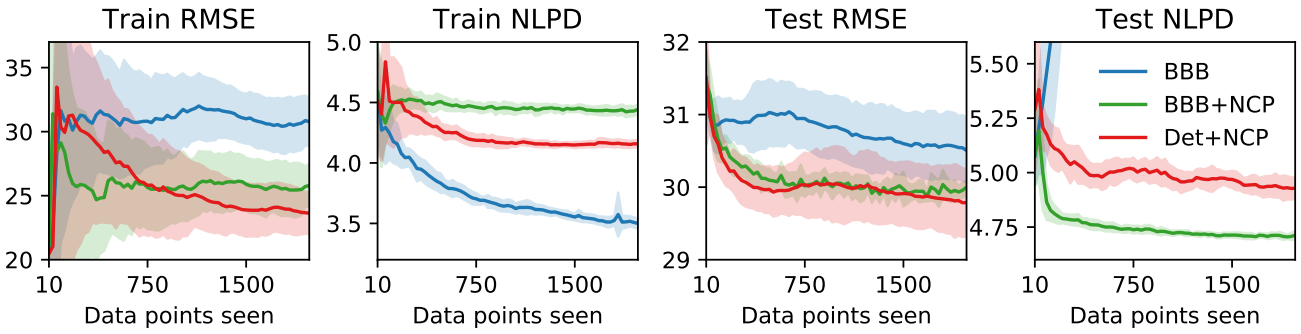


Figure 4: Active learning run on the flights data set. Starting from 10 data points, the models select 10 additional labels every 50 epochs. The models trained with NCP achieve significantly lower root mean squared error (RMSE) and negative log predictive density (NLPD) on the test set. The test NLPD for the baseline model diverges as it overfits to the visible data points. The plots show mean and standard deviation over 10 runs.

We use models of two layers with 50 units each, a batch size of 10 and a learning rate of 10^{-4}. For NCP models, $\epsilon \sim \mathcal{N}(0, 0.1)$. Since BBB+NCP is already regularized by NCP, we scale down the divergence loss for the weight prior by 10^{-3}. Starting from 10 labels, the models select a batch of 10 additional labels every 50 epochs. The 700k data points of the training data set are available for acquisition, and we evaluate performance on the normal test split.

Figure 4 shows the performance for the visible data points and the test set respectively. Bayes by Backprop overfits on this task, while the error continues decreasing for the two models with NCP. We note that BBB and BBB+NCP show similar error on the visible data points, but the NCP models generalize better to unseen data.

4.3. Predicting Hyper Parameter Performance

To evaluate NCP on a different input domain, we apply active learning to the results of a grid search over 1024 different CIFAR-10 models from a previous project. The input is 10 binary ± 1 features, and the output is the final accuracy achieved by the hyper parameter configuration. Our model uses three layers of 29 units each. To generate a test set with distributional shift, we randomly pick a set of 5 features, then order the data points by the number of $+1$ features in that set, breaking ties randomly. The test set is the last 124 points in this ordering. This encourages the picked feature set to be biased towards $+1$ in the test set and -1 in the train set.

To approximate OOD samples for NCP on binary features, we randomly flip each feature with probability 0.25. We start with 20 initial labels and select 5 new labels every 5 epochs. As baseline, we compare to Flipout (Wen et al., 2018), an improved version of BBB that decorrelates mini-batch gradients. Figure 5 shows BBB+NCP is competitive with Flipout, performing better given enough data points.

4.4. Large scale regression of flight delays

In addition to the active learning experiments, we perform a passive learning run on all 700k data points of the flights data set to explore the scalability of NCP. For this, we use networks of 3 layers and 1000 units each. The root mean squared error (RMSE) and negative log predictive density (NLPD) are displayed in Figure 6. Table 1 compares the performance of our models to previously published results. We significantly improve state of the art performance on this data set.

5. Discussion

We develop *noise contrastive priors (NCPs)*, a prior for neural networks in data space. Unlike priors in weight space, data priors let one easily express informative assumptions about input-output relationships. NCPs encourage network weights that not only explain the training data but also capture high uncertainty on OOD inputs. Empirically, we show that NCPs offer clear improvements as an addition to existing baselines, and scale to large regression tasks.

In this work, we focused on active learning, where uncertainty is crucial for determining which data points to select next. In future work it would be interesting to apply NCPs to alternative settings where uncertainty is important, such as supervised learning with sparse or missing data, and longitudinal analyses. In addition, NCPs are only one form of a data prior, designed to encourage uncertainty on OOD inputs. Priors in data space can easily capture other properties such as periodicity or spatial invariance, and they may provide a scalable alternative to Gaussian process priors.

Acknowledgements. We thank Rif Saurous, Balaji Lakshminarayanan, Jascha Sohl-Dickstein, and Matthew D. Hoffman for their comments.

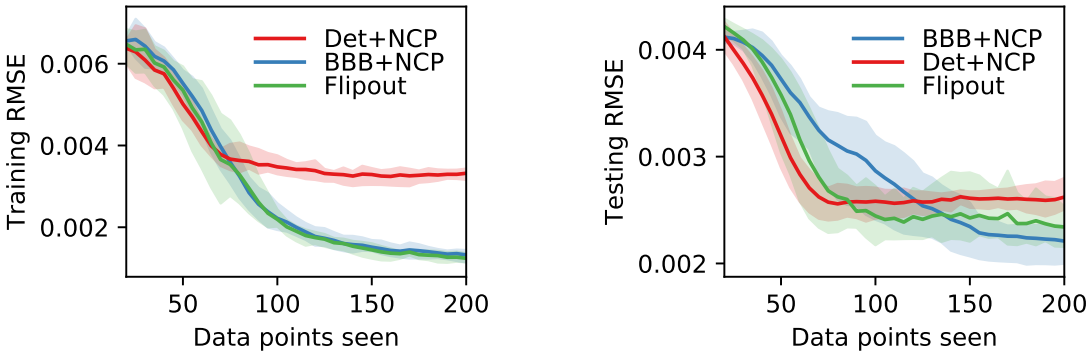


Figure 5: Active learning on our hyper parameter data set. Solid lines show mean and shaded areas percentiles 5 to 95 over multiple scores and 5 seeds. Please see Figure 1 for visualizations of the predictive distributions on this task.

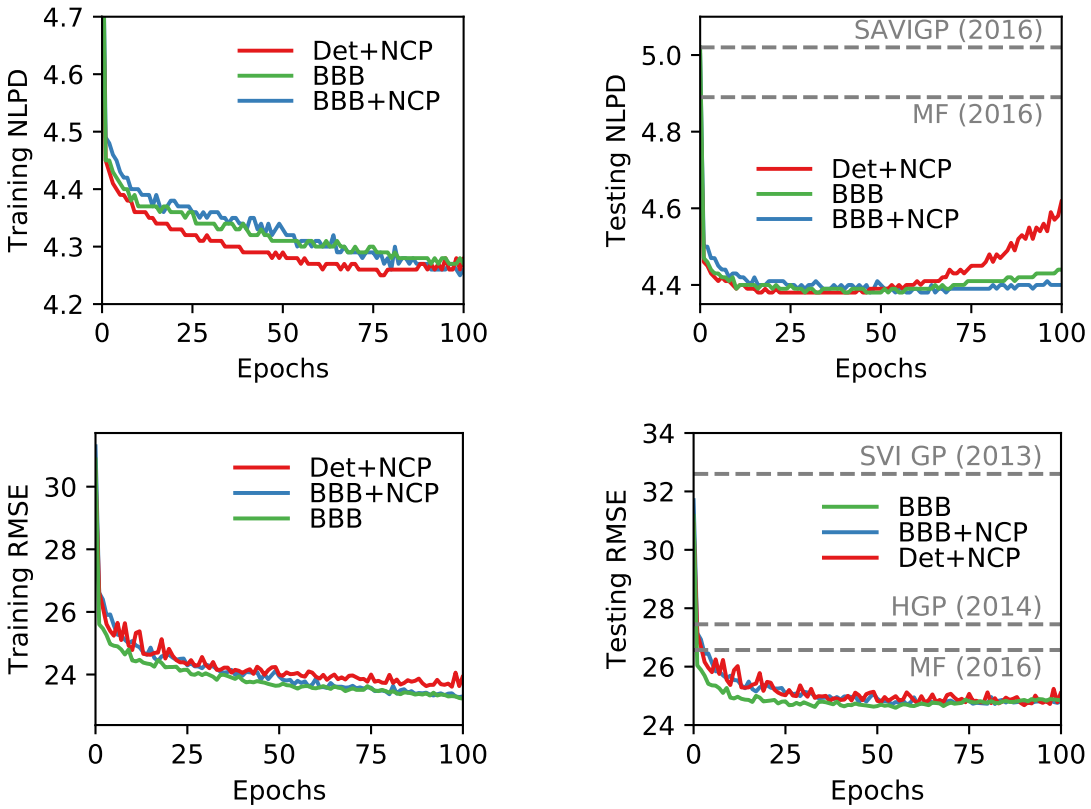


Figure 6: Large-scale regression on the 700k data points of the flights data set. Models trained with NCP achieve the same performance as Bayes by Backprop on this task, displaying the scalability of the approach. All three models significantly improve upon previously published results on this data set.

References

G. An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3):643–674, 1996.

C. M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

D. M. Blei. Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, 1:203–232, 2014.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

W. L. Buntine and A. S. Weigend. Bayesian backpropagation. *Complex Systems*, 5(6):603–643, 1991.

R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth. Manifold gaussian processes for regression. *arXiv preprint arXiv:1402.5876*, 2014.

S. Dasgupta. Analysis of a greedy active learning strategy. In *Neural Information Processing Systems (NIPS)*, 2004.

M. P. Deisenroth and J. W. Ng. Distributed gaussian processes. *arXiv preprint arXiv:1502.02843*, 2015.

J. Denker, D. Schwartz, B. Wittner, S. Solla, R. Howard, L. Jackel, and J. Hopfield. Large automatic learning, rule extraction, and generalization. *Complex Systems*, 1(5): 877–922, 1987.

R. Der and D. D. Lee. Beyond gaussian processes: On the distributions of infinite networks. In *Neural Information Processing Systems*, pages 275–282, 2006.

J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

D. Flam-Shepherd, J. Requeima, and D. Duvenaud. Mapping gaussian process priors to bayesian neural networks. In *NIPS Workshop*, 2017.

Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2-3):133–168, 1997.

Y. Gal, R. Islam, and Z. Ghahramani. Deep bayesian active learning with image data. *arXiv preprint arXiv:1703.02910*, 2017.

S. Ghosh and F. Doshi-Velez. Model selection in bayesian neural networks via horseshoe priors. *arXiv preprint arXiv:1705.10388*, 2017.

A. Graves. Practical variational inference for neural networks. In *Neural Information Processing Systems*, 2011.

M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010a.

M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010b.

D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

J. M. Hernández-Lobato and R. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869, 2015.

N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

E. T. Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2372–2379. IEEE, 2009.

A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems*, pages 1097–1105, 2012.

D. Krueger, C.-W. Huang, R. Islam, R. Turner, A. Lacoste, and A. Courville. Bayesian hypernetworks. *arXiv preprint arXiv:1710.04759*, 2017.

A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei. Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18(1):430–474, 2017.

B. Lakshminarayanan, D. M. Roy, and Y. W. Teh. Mondrian forests for large-scale regression when uncertainty matters. In *Artificial Intelligence and Statistics*, pages 1478–1487, 2016.

J. Lampinen and A. Vehtari. Bayesian approach for neural networks—review and case studies. *Neural networks*, 14 (3):257–274, 2001.

M. Lázaro-Gredilla and A. R. Figueiras-Vidal. Marginalized neural network mixtures for large-scale regression. *IEEE transactions on neural networks*, 21(8):1345–1351, 2010.

J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

K. Lee, H. Lee, K. Lee, and J. Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017.

D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

X. Li and Y. Guo. Adaptive active learning for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 859–866. IEEE, 2013.

S. Liang, Y. Li, and R. Srikant. Principled detection of out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2018.

C. Louizos and M. Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *International Conference on Machine Learning*, pages 1708–1716, 2016.

C. Louizos, K. Ullrich, and M. Welling. Bayesian compression for deep learning. In *Neural Information Processing Systems*, pages 3290–3300, 2017.

A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning (ICML)*, 2013.

D. J. MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992a.

D. J. MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992b.

D. J. MacKay. Introduction to Gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, 168:133–166, 1998.

K. Matsuoka. Noise injection into inputs in back-propagation learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):436–440, 1992.

A. G. d. G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations*, 2018.

A. K. McCallumzy and K. Nigamy. Employing em and pool-based active learning for text classification. In *Proc. International Conference on Machine Learning (ICML)*, pages 359–367. Citeseer, 1998.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

T. Miyato, S. Maeda, M. Koyama, K. Nakae, and S. Ishii. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*, 2015.

A. Mnih and K. Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in neural information processing systems*, pages 2265–2273, 2013.

P. Müller and D. R. Insua. Issues in bayesian analysis of neural network models. *Neural Computation*, 10(3):749–770, 1998.

R. M. Neal. Priors for infinite networks. *Technical Report CRG-TR-94-1*, pages 29–53, 1994.

R. M. Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IIEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1998.

N. Roy and A. McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, pages 441–448, 2001.

P. Schulam and S. Saria. A framework for individualizing predictions of disease trajectories by exploiting multi-resolution structure. In *Advances in Neural Information Processing Systems*, pages 748–756, 2015.

B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 1992.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

K.-K. Sung. Learning and example selection for object and pattern detection. *MIT A.I. Memo No. 1521*, 1994.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Neural Information Processing Systems*, 2014.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

D. Tran, A. Kucukelbir, A. B. Dieng, M. Rudolph, D. Liang, and D. M. Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016.

A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2017.

Y. Wen, P. Vicol, J. Ba, D. Tran, and R. Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*, 2018.

G. Zhang, S. Sun, D. Duvenaud, and R. Grosse. Noisy natural gradient as variational inference. *arXiv preprint arXiv:1712.02390*, 2017.

H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.