Bayesian Deep Learning via Subnetwork Inference

Nikitina Maria

MIPT

October 10, 2024

Problem

A critical shortcoming of deep neural networks (NNs) is that they tend to be poorly calibrated and overconfident in their predictions, especially when there is a shift between the train and test data distributions.

Bayesian deep learning

Exact posterior inference is intractable in NNs. Existing methods invoke unrealistic assumptions to scale to NNs with large numbers of weights.

The proposal of the paper

Posterior predictive distribution of a full network can be well represented by that of a subnetwork.

Let $\mathbf{w} \in \mathbb{R}^D$ be the D-dimensional vector of all neural network weights. $\mathcal{D} = \{\mathbf{y}, \mathbf{X}\}$ – the training data. We then wish to infer their full *posterior distribution*:

$$p(\mathbf{w}|\mathcal{D}) = p(\mathbf{w}|\mathbf{y}, \mathbf{X}) \propto p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})$$

Finally, predictions for new data points $\mathbf{X}^*$ are made through marginalisation of the posterior:

$$p(\mathbf{y}^*|\mathbf{X}^*, \mathcal{D}) = \int_{\mathbf{w}} p(\mathbf{y}^*|\mathbf{X}^*, \mathbf{w})p(\mathbf{w}|\mathcal{D})d\mathbf{w}$$

$p(\mathbf{w}|\mathcal{D}) \approx \prod_{d=1}^{D} q(w_d)$ – crude posterior approximation.

Overparameterization

[Maddox et al.] have shown that, in the neighborhood of local optima, there are many directions that leave the NN's predictions unchanged. NNs can be heavily pruned without sacrificing test-set accuracy.

Inference over submodels

Inference can be effective even when not performed on the full parameter space.

Let's combine these two ideas and make the following two-step approximation of the posterior. Let S be small subset of weights:

$$p(\mathbf{w}|\mathcal{D}) \approx p(\mathbf{w}_S|\mathcal{D}) \prod_r \delta(w_r - \hat{w}_r) \approx q(\mathbf{w}_S) \prod_r \delta(w_r - \hat{w}_r) = q_S(\mathbf{w})$$

Maddox, W. J. et.al. Rethinking parameter counting in deep models: Effective dimensionality revisited, 2020

Let's denote NN function as $\mathbf{f} : \mathbb{R}^I \to \mathbf{R}^O$. A prior over NN's weights $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}, \mathbf{0}, \lambda \mathbf{I})$. MAP setting of the weights:

$$\hat{\mathbf{w}} = \text{argmax}_{\mathbf{w}}[\log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) + \log p(\mathbf{w})]$$

The posterior is then approximated with a second order Taylor expansion around the MAP estimate with the Hessian of the negative logposterior density. Thus, the approximate posterior takes the form of a full covariance Gaussian with Hessian as covariance matrix:

$$p(\mathbf{w}|\mathcal{D}) \approx q(\mathbf{w}) = \mathcal{N}(\mathbf{w}, \hat{\mathbf{w}}, \mathbf{H}^{-1})$$

Posterior

In practise, the Hessian $\mathbf{H}$ is commonly replaced with the generalized Gauss-Newton matrix (GGN) $\tilde{\mathbf{H}} \in \mathbf{R}^{D \times D}$. The resulting approximate posterior:

$$q_S(\mathbf{w}) = \mathcal{N}(\mathbf{w}, \hat{\mathbf{w}}, \tilde{\mathbf{H}}_S^{-1}) \prod_r \delta(w_r - \hat{w}_r)$$

Prediction

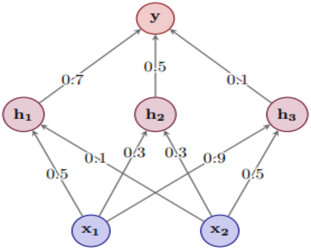
Perform a local linearization of the NN while fixing w_r to $\hat{w}_r$:

$$\mathbf{f}_{lin}(\mathbf{x}, \mathbf{w}_S) = \mathbf{f}(\mathbf{x}, \hat{\mathbf{w}}_S) + \hat{\mathbf{J}}_S(\mathbf{x})(\mathbf{w}_S - \hat{\mathbf{w}}_S)$$

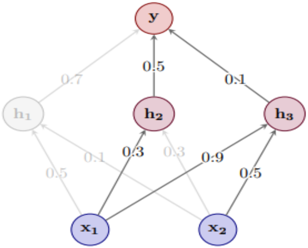
The corresponding predictive distributions are:

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \mathcal{N}(\mathbf{y}^*; \mathbf{f}(\mathbf{x}^*, \hat{\mathbf{w}}), \Sigma_S(\mathbf{x}^*) + \sigma^2\mathbf{I}),$$

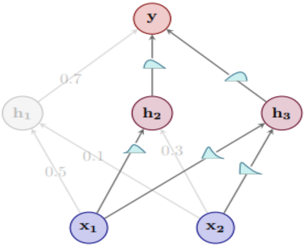
$\Sigma_S(\mathbf{x}^*) = \hat{\mathbf{J}}_S(\mathbf{x}^*)^\top \tilde{\mathbf{H}}_S^{-1} \hat{\mathbf{J}}_S(\mathbf{x}^*)$



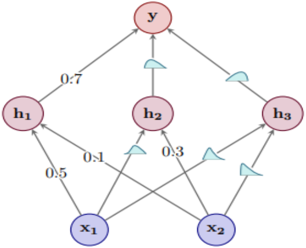
(a) Point Estimation



(b) Subnetwork Selection



(c) Bayesian Inference



(d) Prediction

Computing the exact posterior remains intractable

The true posterior for the linearized model is Gaussian or approximately Gaussian:

$$p(\mathbf{w}|\mathcal{D}) \approx \mathcal{N}(\mathbf{w}, \hat{\mathbf{w}}, \tilde{\mathbf{H}^{-1}})$$

Dirac delta distributions

For the case of a product of a full covariance Gaussian with Dirac deltas, the squared 2-Wasserstein distance takes the following form:

$$W_2(p(\mathbf{w}|\mathcal{D}), q_S(\mathbf{w}))^2 = \mathrm{Tr}\left(\tilde{\mathbf{H}}^{-1} + \tilde{\mathbf{H}}_{S+}^{-1} - 2(\tilde{\mathbf{H}}_{S+}^{-1/2}\tilde{\mathbf{H}}^{-1}\tilde{\mathbf{H}}_{S+}^{-1/2})^{1/2}\right)$$

$$W_2(p(\mathbf{w}|\mathcal{D}), q_S(\mathbf{w}))^2 \approx \sum_{d=1}^{D} \sigma_d^2(1 - m_d)$$

Erik Daxberger, Eric Nalisnick, James Urquhart Allingham, Javier Antoran, Jose Miguel Hernandez-Lobato (2021)
Bayesian Deep Learning via Subnetwork Inference