Appendix: Bayesian Uncertainty Estimation for Batch Normalized Deep Networks

1. Appendix

1.1. Variational Approximation

Assume we were to come up with a family of distributions parameterized by $\boldsymbol{\theta}$ in order to approximate the posterior, $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$. Our goal is to set $\boldsymbol{\theta}$ such that $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ is as similar to the true posterior $p(\boldsymbol{\omega}|\mathbf{D})$ as possible.

For clarity, $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ is a distribution over stochastic parameters $\boldsymbol{\omega}$ that is determined by a set of learnable parameters $\boldsymbol{\theta}$ and some source of randomness. The approximation is therefore limited by our choice of parametric function $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ as well as the randomness.[1] Given $\boldsymbol{\omega}$ and an input $\mathbf{x}$, an output distribution $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) = p(\mathbf{y}|f_{\boldsymbol{\omega}}(\mathbf{x})) = f_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{y})$ is induced by observation noise (the conditionality of which is omitted for brevity).

One strategy for optimizing $\boldsymbol{\theta}$ is to minimize $\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\omega})||p(\boldsymbol{\omega}|\mathbf{D}))$, the KL divergence of $p(\boldsymbol{\omega}|\mathbf{D})$ w.r.t. $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$. Minimizing $\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\omega})||p(\boldsymbol{\omega}|\mathbf{D}))$ is equivalent to maximizing the ELBO:

$$\int_{\boldsymbol{\omega}} q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) \ln p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} - \mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\omega})||p(\boldsymbol{\omega}))$$

Assuming i.i.d. observation noise, this is equivalent to minimizing:

$$\mathcal{L}_{\mathrm{VA}}(\boldsymbol{\theta}) := -\sum_{n=1}^{N} \int q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) \ln p(\mathbf{y}_i|f_{\boldsymbol{\omega}}(\mathbf{x}_i)) \mathrm{d}\boldsymbol{\omega} + \mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\omega})||p(\boldsymbol{\omega}))$$

Instead of making the optimization on the full training set, we can use a subsampling (yielding an unbiased estimate of $\mathcal{L}_{\mathrm{VA}}(\boldsymbol{\theta})$) for iterative optimization (as in mini-batch optimization):

$$\hat{\mathcal{L}}_{\mathrm{VA}}(\boldsymbol{\theta}) := -\frac{N}{M} \sum_{i \in B} \int_{\boldsymbol{\omega}} q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) \ln p(\mathbf{y}_i|f_{\boldsymbol{\omega}}(\mathbf{x}_i)) \mathrm{d}\boldsymbol{\omega} + \mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\omega})||p(\boldsymbol{\omega}))$$

During optimization, we want to take the derivative of the expected log likelihood w.r.t. the learnable parameters $\boldsymbol{\theta}$. (Gal, 2016) provides an intuitive interpretation of a MC estimate for NNs trained with a SRT (equivalent to the reparametrisation trick in (Kingma & Welling, 2014)), and this interpretation is followed here. We let an auxillary variable $\boldsymbol{\epsilon}$ represent the stochasticity during training such that $\boldsymbol{\omega} = g(\boldsymbol{\theta}, \boldsymbol{\epsilon})$. The function g and the distribution of $\boldsymbol{\epsilon}$ are such that $p(g(\boldsymbol{\theta}, \boldsymbol{\epsilon})) = q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$.[2] Assuming $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ can be written $\int_{\boldsymbol{\epsilon}} q_{\boldsymbol{\theta}}(\boldsymbol{\omega}|\boldsymbol{\epsilon}) p(\boldsymbol{\epsilon}) \mathrm{d}\boldsymbol{\epsilon}$ where $q_{\boldsymbol{\theta}}(\boldsymbol{\omega}|\boldsymbol{\epsilon}) = \delta(\boldsymbol{\omega} - g(\boldsymbol{\theta}, \boldsymbol{\epsilon}))$, this auxiliary variable yields the estimate (full proof in (*Gal*, 2016)):

$$\hat{\mathcal{L}}_{\mathrm{VA}}(\boldsymbol{\theta}) = -\frac{N}{M} \sum_{i \in B} \int_{\epsilon} p(\epsilon) \ln p(\mathbf{y}_i|f_{g(\boldsymbol{\theta}, \epsilon)}(\mathbf{x}_i)) \mathrm{d}\epsilon + \mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\omega})||p(\boldsymbol{\omega}))$$

where taking a single sample MC estimate of the integral yields the optimization objective in Eq. 1.

[1] In an approx. Bayesian view of a NN, $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ would correspond to the distribution of weights in the network given by some SRT.

[2] In a NN trained with BN, it is easy to see that g exists if we let $\boldsymbol{\epsilon}$ represent a mini-batch selection from the training data, since all BN units' means and variances are completely determined by $\boldsymbol{\epsilon}$ and θ.

1.2. KL Divergence of factorized Gaussians

If $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ and $p(\boldsymbol{\omega})$ factorize over all stochastic parameters:

$$\begin{aligned}
\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\omega})||p(\boldsymbol{\omega})) &= -\int_{\boldsymbol{\omega}} \prod_i \left[q_{\boldsymbol{\theta}}(\omega_i)\right] \ln \frac{\prod_i p(\omega_i)}{\prod_i q_{\boldsymbol{\theta}}(\omega_i)} \mathrm{d}\boldsymbol{\omega} \\
&= -\int_{\boldsymbol{\omega}} \prod_i \left[q_{\boldsymbol{\theta}}(\omega_i)\right] \sum_i \left[\ln \frac{p(\omega_i)}{q_{\boldsymbol{\theta}}(\omega_i)}\right] \prod_i \mathrm{d}\omega_i \\
&= \sum_j \Big[-\int_{\boldsymbol{\omega}} \prod_i \left[q_{\boldsymbol{\theta}}(\omega_i)\right] \ln \frac{p(\omega_j)}{q_{\boldsymbol{\theta}}(\omega_j)} \prod_i \mathrm{d}\omega_i \Big] \\
&= \sum_j \Big[-\int_{\omega_j} q_{\boldsymbol{\theta}}(\omega_j) \ln \frac{p(\omega_j)}{q_{\boldsymbol{\theta}}(\omega_j)} \mathrm{d}\omega_j \int_{\omega_{i\neq j}} \prod_{i\neq j} q_{\boldsymbol{\theta}}(\omega_i) \mathrm{d}\omega_i \Big] \\
&= \sum_i -\int_{\omega_i} q_{\boldsymbol{\theta}}(\omega_i) \ln \frac{p(\omega_i)}{q_{\boldsymbol{\theta}}(\omega_i)} \mathrm{d}\omega_i \\
&= \sum_i \mathrm{KL}(q_{\boldsymbol{\theta}}(\omega_i)||p(\omega_i))
\end{aligned} \tag{3}$$

such that $\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\omega})||p(\boldsymbol{\omega}))$ is the sum of the KL divergence terms for the individual stochastic parameters ω_i. If the factorized distributions are Gaussians, where $q_{\boldsymbol{\theta}}(\omega_i) = \mathcal{N}(\mu_q, \sigma_q^2)$ and $p(\omega_i) = \mathcal{N}(\mu_p, \sigma_p^2)$ we get:

$$\begin{aligned}
\mathrm{KL}(q_{\boldsymbol{\theta}}(\omega_i)||p(\omega_i)) &= \int_{\omega_i} q_{\boldsymbol{\theta}}(\omega_i) \ln \frac{q_{\boldsymbol{\theta}}(\omega_i)}{p(\omega_i)} \mathrm{d}\omega_i \\
&= -H(q_{\boldsymbol{\theta}}(\omega_i)) - \int_{\omega_i} q_{\boldsymbol{\theta}}(\omega_i) \ln p(\omega_i) \mathrm{d}\omega_i \\
&= -\frac{1}{2}(1 + \ln(2\pi\sigma_q^2)) \\
&\quad - \int_{\omega_i} q_{\boldsymbol{\theta}}(\omega_i) \ln \frac{1}{(2\pi\sigma_p^2)^{1/2}} \exp\Big\{ -\frac{(\omega_i - \mu_p)^2}{2\sigma_p^2} \Big\} \mathrm{d}\omega_i \\
&= -\frac{1}{2}(1 + \ln(2\pi\sigma_q^2)) \\
&\quad + \frac{1}{2}\ln(2\pi\sigma_p^2) + \frac{\mathbb{E}_q[\omega_i^2] - 2\mu_p\mathbb{E}_q[\omega_i] + \mu_p^2}{2\sigma_p^2} \\
&= \ln\frac{\sigma_p}{\sigma_q} + \frac{\sigma_q^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} - \frac{1}{2}
\end{aligned} \tag{4}$$

for each KL divergence term. Here $H(q_{\boldsymbol{\theta}}(\omega_i)) = \frac{1}{2}(1 + \ln(2\pi\sigma_q^2))$ is the differential entropy of $q_{\boldsymbol{\theta}}(\omega_i)$.

1.3. Distribution of $\mu_{\mathbf{B}}^u, \sigma_{\mathbf{B}}^u$

Here we approximate the distribution of mean and standard deviation of a mini-batch, separately to two Gaussians – This has also been empirically verified, see Figure 1 for 2 sample plots and the appendix section 1.9 for more. For the mean we get:

$$\mu_{\mathrm{B}} = \frac{\Sigma_{\mathrm{m}=1}^{\mathrm{M}} \mathbf{W}^{(j)} \mathbf{x}_{\mathrm{m}}}{\mathrm{M}}$$

where $\mathbf{x}_{\mathrm{m}}$ are the examples in the sampled batch. We will assume these are sampled i.i.d.[3]. Samples of the random variable $\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}}$ are then i.i.d.. Then by central limit theorem (CLT) the following holds for sufficiently large M (often ≥ 30):

$$\mu_{\mathrm{B}} \sim \mathcal{N}(\mu, \frac{\sigma^2}{\mathrm{M}})$$

[3] Although in practice with deep learning, mini-batches are sampled without replacement, stochastic gradient descent samples with replacement in its standard form.

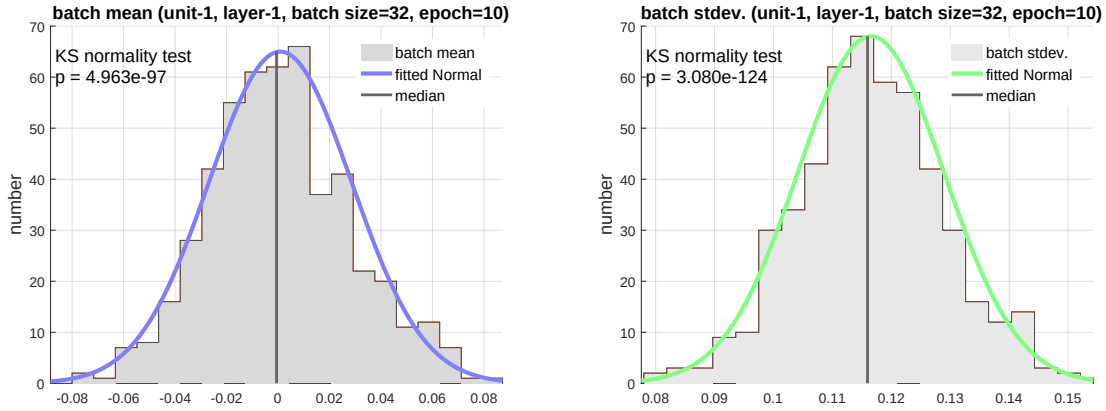

Figure 1. Batch statistics used to train the network are normal. A one-sample Kolmogorov-Smirnov test checks that $\mu_{\mathbf{B}}$ and $\sigma_{\mathbf{B}}$ come from a standard normal distribution. More examples are available in Appendix 1.9.

For standard deviation:

$$\sigma_{\mathrm{B}} = \sqrt{\frac{\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2}{M}}$$

Then

$$\sqrt{M}(\sigma_{\mathrm{B}} - \sigma) = \sqrt{M}\Big(\sqrt{\frac{\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2}{M}} - \sqrt{\sigma^2}\Big)$$

We want to rewrite $\sqrt{\frac{\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2}{M}}$. We take a Taylor expansion of $f(x) = \sqrt{x}$ around $a = \sigma^2$. With $x = \frac{\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2}{M}$:

$$\sqrt{x} = \sqrt{\sigma^2} + \frac{1}{2\sqrt{\sigma^2}}(x - \sigma^2) + \mathcal{O}[(x - \sigma^2)^2]$$

so

$$\begin{aligned}
\sqrt{M}(\sigma_{\mathrm{B}} - \sigma) &= \sqrt{M}\Bigg(\frac{1}{2\sqrt{\sigma^2}}\Big(\frac{\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2}{M} - \sigma^2\Big) + \\
&\qquad \mathcal{O}\left[\Big(\frac{\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2}{M} - \sigma^2\Big)^2\right]\Bigg) \\
&= \frac{\sqrt{M}}{2\sigma}\Big(\frac{1}{M}\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2 - \sigma^2\Big) + \\
&\qquad \mathcal{O}\left[\sqrt{M}\Big(\frac{\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2}{M} - \sigma^2\Big)^2\right] \\
&= \frac{1}{2\sigma\sqrt{M}}\Big(\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2 - M\sigma^2\Big) + \\
&\qquad \mathcal{O}\left[\sqrt{M}\Big(\frac{\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2}{M} - \sigma^2\Big)^2\right]
\end{aligned}$$

consider $\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2$. We know that $E[\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}}] = \mu$ and write

$$
\begin{aligned}
&\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2 \\
=&\Sigma_{\mathrm{m}=1}^{\mathrm{M}}((\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu) - (\mu_{\mathrm{B}} - \mu))^2 \\
=&\Sigma_{\mathrm{m}=1}^{\mathrm{M}}((\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2 + (\mu_{\mathrm{B}} - \mu)^2 - 2(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)(\mu_{\mathrm{B}} - \mu)) \\
=&\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2 + M(\mu_{\mathrm{B}} - \mu)^2 - 2(\mu_{\mathrm{B}} - \mu)\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu) \\
=&\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2 - M(\mu_{\mathrm{B}} - \mu)^2 \\
=&\Sigma_{\mathrm{m}=1}^{\mathrm{M}}((\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2 - (\mu_{\mathrm{B}} - \mu)^2)
\end{aligned}
$$

then

$$
\begin{aligned}
\sqrt{M}(\sigma_{\mathrm{B}} - \sigma) &= \frac{1}{2\sigma\sqrt{M}}\Big(\Sigma_{\mathrm{m}=1}^{\mathrm{M}}((\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2 - (\mu_{\mathrm{B}} - \mu)^2) - M\sigma^2\Big) + \\
&\qquad \mathcal{O}\left[\sqrt{M}\Big(\frac{\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2}{M} - \sigma^2\Big)^2\right] \\
&= \frac{1}{2\sigma\sqrt{M}}\Big(\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2 - \Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mu_{\mathrm{B}} - \mu)^2 - M\sigma^2\Big) + \\
&\qquad \mathcal{O}\left[\sqrt{M}\Big(\frac{\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2}{M} - \sigma^2\Big)^2\right] \\
&= \frac{1}{2\sigma\sqrt{M}}\Big(\Sigma_{\mathrm{m}=1}^{\mathrm{M}}((\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2 - \sigma^2) - \Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mu_{\mathrm{B}} - \mu)^2\Big) + \\
&\qquad \mathcal{O}\left[\sqrt{M}\Big(\frac{\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2}{M} - \sigma^2\Big)^2\right] \\
&= \frac{1}{2\sigma\sqrt{M}}\Sigma_{\mathrm{m}=1}^{\mathrm{M}}((\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2 - \sigma^2) \\
&\quad - \frac{1}{2\sigma\sqrt{M}}\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mu_{\mathrm{B}} - \mu)^2 \\
&\quad + \mathcal{O}\left[\sqrt{M}\Big(\frac{\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2}{M} - \sigma^2\Big)^2\right] \\
&= \underbrace{\frac{1}{2\sigma\sqrt{M}}\Sigma_{\mathrm{m}=1}^{\mathrm{M}}((\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2 - \sigma^2)}_{\text{term A}} \\
&\quad \underbrace{- \frac{\sqrt{M}}{2\sigma}(\mu_{\mathrm{B}} - \mu)^2}_{\text{term B}} \\
&\quad \underbrace{+ \mathcal{O}\left[\sqrt{M}\Big(\frac{\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2}{M} - \sigma^2\Big)^2\right]}_{\text{term C}}
\end{aligned}
$$

We go through each term in turn

Term A
We have

$$\text{Term A} = \frac{1}{2\sigma\sqrt{M}}\Sigma_{\mathrm{m}=1}^{\mathrm{M}}((\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2 - \sigma^2)$$

where $\Sigma_{\mathrm{m}=1}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2$ is the sum of M RVs $(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2$. Note that since $E[\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}}] = \mu$ it holds that $E[(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2] = \sigma^2$. Since $(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2$ is sampled approximately iid (by assumptions above), for large enough M

by CLT it holds approximately that

$$\Sigma_{\mathrm{m=1}}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2 \sim \mathcal{N}(M\sigma^2, M\mathrm{Var}((\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2))$$

where

$$\begin{aligned}\mathrm{Var}((\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2) &= E[(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^{2*2}] - E[(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2]^2 \\ &= E[(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^4] - \sigma^4\end{aligned}$$

Then

$$\Sigma_{\mathrm{m=1}}^{\mathrm{M}}((\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2 - \sigma^2) \sim \mathcal{N}(0, M * E[(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^4] - M\sigma^4)$$

so

$$\text{Term A} \sim \mathcal{N}(0, \frac{E[(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^4] - \sigma^4}{4\sigma^2})$$

Term B
We have

$$\text{Term B} = \frac{\sqrt{M}}{2\sigma}(\mu_{\mathrm{B}} - \mu)^2 = \frac{1}{2\sigma}\sqrt{M}(\mu_{\mathrm{B}} - \mu)(\mu_{\mathrm{B}} - \mu)$$

Consider $(\mu_{\mathrm{B}} - \mu)$. As $\mu_{\mathrm{B}} \xrightarrow{p} \mu$ when $M \to \infty$ we have $\mu_{\mathrm{B}} - \mu \xrightarrow{p} 0$. We also have

$$\sqrt{M}(\mu_{\mathrm{B}} - \mu) = \frac{\Sigma_{\mathrm{m=1}}^{\mathrm{M}}\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}}}{\sqrt{M}} - \sqrt{M}\mu$$

which by CLT is approximately Gaussian for large M. We can then make use of the Cramer-Slutzky Theorem, which states that if $(X_n)_{n\geq 1}$ and $(Y_n)_{n\geq 1}$ are two sequences such that $X_n \xrightarrow{d} X$ and $Y_n \xrightarrow{p} a$ as $n \to \infty$ where a is a constant, then as $n \to \infty$, it holds that $X_n * Y_n \xrightarrow{d} X * a$. Thus, Term B is approximately 0 for large M.

Term C
We have

$$\text{Term C} = \mathcal{O}\left[\sqrt{M}\Big(\frac{\Sigma_{\mathrm{m=1}}^{\mathrm{M}}(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu_{\mathrm{B}})^2}{M} - \sigma^2\Big)^2\right]$$

Since $E[(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^2] = \sigma^2$ we can make the same use of Cramer-Slutzky as for *Term B*, such that Term C is approximately 0 for large M.

Finalizing the distribution
We have approximately

$$\sqrt{M}(\sigma_{\mathrm{B}} - \sigma) \sim \mathcal{N}(0, \frac{E[(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^4] - \sigma^4}{4\sigma^2})$$

so

$$\sigma_{\mathrm{B}} \sim \mathcal{N}(\sigma, \frac{E[(\mathbf{W}^{(j)}\mathbf{x}_{\mathrm{m}} - \mu)^4] - \sigma^4}{4\sigma^2 M})$$

1.4. Prior

Here we make use of the stochasticity from BN modeled in the Appendix section 1.3, to evaluate the implied prior on the stochastic variables for a BN network. Specifically, we consider a BN network with fully connected layers and BN applied to each layer, trained with L2-regularization (weight decay). In the following, we make use of the simplifying assumptions of no scale and shift tranformations, BN applied to each layer, and independent input units to each layer.

Equivalence between the objectives of Eq. (1) and (2) requires:

$$\begin{aligned}\frac{\partial}{\partial\theta_k}\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\omega})||p(\boldsymbol{\omega})) &= N\tau\frac{\partial}{\partial\theta_k}\Omega(\boldsymbol{\theta}) \\ &= N\tau\frac{\partial}{\partial\theta_k}\sum_{l=1}^{L}\lambda_l||\mathbf{W}^l||^2\end{aligned} \tag{5}$$

where $\theta_k \in \boldsymbol{\theta}$, and $\boldsymbol{\theta}$ is the set of weights in the network. To proceed with the LHS of Eq. (5) we first need to find the approximate posterior $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ that batch normalization induces. As shown in Appendix 1.3, with some weak assumptions and approximations the Central Limit Theorem (CLT) yields Gaussian distributions of the stochastic variables $\mu^u_{\mathbf{B}}, \sigma^u_{\mathbf{B}}$, for large enough M. For any BN unit u:

$$\begin{aligned}&\mu^u_{\mathbf{B}} \overset{\propto}{\sim} \mathcal{N}(\mu^u, \frac{(\sigma^u)^2}{M}), \\ &\sigma^u_{\mathbf{B}} \overset{\propto}{\sim} \mathcal{N}(\sigma^u, \frac{\mathbb{E}[(\mathbf{W}^{(u)}\boldsymbol{x} - \mu^u)^4] - (\sigma^u)^4}{4(\sigma^u)^2 M})\end{aligned} \tag{6}$$

where μ^u and σ^u are *population-level* moments (i.e. moments over $\mathbf{D}$).

We assume that $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ and $p(\boldsymbol{\omega})$ factorize over all stochastic variables.[4] We use i as an index of the set of stochastic variables. As shown in Eq. (3) in Appendix 1.2, the factorized distributions yield:

$$\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\omega})||p(\boldsymbol{\omega})) = \sum_i \mathrm{KL}(q_{\boldsymbol{\theta}}(\omega_i)||p(\omega_i))$$

Note that each BN unit produces two $\mathrm{KL}(q_{\boldsymbol{\theta}}(\omega_i)||p(\omega_i))$ terms: one for $\omega_i = \mu^u_{\mathbf{B}}$ and one for $\omega_i = \sigma^u_{\mathbf{B}}$. We consider these terms for one particular BN unit u, and drop the index i for brevity. We use a Gaussian prior $p(\omega_i) = \mathcal{N}(\mu_p, \sigma_p^2)$ and, for consistency, use the notation $q_{\boldsymbol{\theta}}(\omega_i) = \mathcal{N}(\mu_q, \sigma_q^2)$. As shown in Eq. (4) in Appendix 1.2:

$$\mathrm{KL}(q_{\boldsymbol{\theta}}(\omega_i)||p(\omega_i)) = \ln\frac{\sigma_p}{\sigma_q} + \frac{\sigma_q^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} - \frac{1}{2}$$

Since θ_k changes during training, a prior cannot depend on θ_k so $\frac{\partial}{\partial\theta_k}(\mu_p) = \frac{\partial}{\partial\theta_k}(\sigma_p) = 0$. Letting $(\cdot)'$ denote $\frac{\partial}{\partial\theta_k}(\cdot)$:

$$\frac{\partial}{\partial\theta_k}\mathrm{KL}(q_{\boldsymbol{\theta}}(\omega_i)||p(\omega_i)) = \frac{\sigma_q\sigma_q' + \mu_q\mu_q' - \mu_p\mu_q'}{\sigma_p^2} - \frac{\sigma_q'}{\sigma_q} \tag{7}$$

We need not consider θ_k past a previous layer's BN, since a normalization step is performed before scale and shift. In the general case with a given Gaussian $p(\boldsymbol{\omega})$, Eq. 7 evaluated on all BN units' means and standard deviations w.r.t. all θ_k up to a previous layer's BN, would yield an expression for a custom $N\tau\frac{\partial}{\partial\theta_k}\Omega(\boldsymbol{\theta})$ that could be used for an exact VI treatment of BN.

In our reconciliation of weight decay however, given our assumptions of no scale and shift and BN applied to each layer, we need only consider the *weights* in the same layer as the BN unit. This means that the stochastic variables in layer l are only affected by weights in $\theta_k \in \boldsymbol{W}^l$ (i.e. not the scale and shift variables operating on the input to the layer). We denote a weight connecting the k:th input unit to the u:th BN unit by $\mathbf{W}^{(u,k)}$. For such weights, we need to derive μ_q' and σ_q', for two cases: $\omega_i = \mu^u_{\mathbf{B}}$ and $\omega_i = \sigma^u_{\mathbf{B}}$. We denote the priors of the mean and std. dev for $\mu^u_{\mathbf{B}}$ by $\mu_{\mu,q}$ and $\sigma_{\mu,q}$, and for $\sigma^u_{\mathbf{B}}$ by $\mu_{\sigma,q}$ and $\sigma_{\sigma,q}$. Using the distributions modeled in Eq. 6:

[4] The empirical distributions have been numerically checked to be linearly independent and the joint distribution is close to a bi-variate Gaussian.

Case 1: $\omega_i = \mu^u_{\mathbf{B}}$

$$\mu_{\mu,q} = \sum_{\boldsymbol{x}\in\mathbf{D}} \frac{\mathbf{W}^{(u)}\boldsymbol{x}}{N} = \mathbf{W}^{(u)}\bar{\boldsymbol{x}}$$
$$\mu'_{\mu,q} = \sum_{\boldsymbol{x}\in\mathbf{D}} \frac{\boldsymbol{x}_k}{N} = \bar{\boldsymbol{x}}_k$$
$$\sigma_{\mu,q} = \sqrt{\frac{(\sigma^u)^2}{M}} = \sqrt{\frac{\sum_{\boldsymbol{x}\in\mathbf{D}}(\mathbf{W}^{(u)}\boldsymbol{x} - \mu_q)^2}{NM}}$$
$$\sigma'_{\mu,q} = \frac{1}{2}\sigma_q^{-1}\sum_{\boldsymbol{x}\in\mathbf{D}} \frac{2(\mathbf{W}^{(u)}\boldsymbol{x} - \mu_q)(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k)}{NM} = \sigma_q^{-1}\bigg(\sum_{i=1}^{K}\mathbf{W}^{(u,i)}\mathrm{Cov}(x_i, x_k)\bigg)M^{-1}$$

where there are K input units to the layer.

Case 2: $\omega_i = \sigma^u_{\mathbf{B}}$

$$\mu_{\sigma,q} = \sqrt{\frac{\sum_{\boldsymbol{x}\in\mathbf{D}}(\mathbf{W}^{(u)}\boldsymbol{x} - \mu_q)^2}{N}} = \sigma_{\mu,q}M^{\frac{1}{2}}$$
$$\mu'_{\sigma,q} = \sigma_{\mu,q}^{-1}M^{-\frac{1}{2}}\bigg(\sum_{i=1}^{K}\mathbf{W}^{(u,i)}\mathrm{Cov}(x_i, x_k))\bigg)$$
$$\sigma_{\sigma,q} = \frac{\mathbb{E}[(\mathbf{W}^{(u)}\boldsymbol{x} - \mu^u)^4] - (\sigma^u)^4}{4(\sigma^u)^2M}$$
$$\sigma'_{\sigma,q} = \frac{\mathbb{E}[(\mathbf{W}^{(u)}\boldsymbol{x} - \mu^u)^4]'\sigma^u - 2(\sigma^u)^4(\sigma^u)' - 2(\sigma^u)'\mathbb{E}[(\mathbf{W}^{(u)}\boldsymbol{x} - \mu^u)^4]}{4(\sigma^u)^3M}$$

Combining these results with Eq. 7 we find that taking $\mathrm{KL}(q_{\boldsymbol{\theta}}(\omega_i)||p(\omega_i))$ for the mean and variance of a single BN unit u wrt the weight from input unit k:

$$\begin{aligned}
&\frac{\partial}{\partial\mathbf{W}^{(u,k)}}\mathrm{KL}(q_{\boldsymbol{\theta}}(\mu^u_{\mathbf{B}})||p(\mu^u_{\mathbf{B}})) + \frac{\partial}{\partial\mathbf{W}^{(u,k)}}\mathrm{KL}(q_{\boldsymbol{\theta}}(\sigma^u_{\mathbf{B}})||p(\sigma^u_{\mathbf{B}})) \\
=&\frac{\sigma_{\mu,q}\sigma'_{\mu,q} + \mu_{\mu,q}\mu'_{\mu,q} - \mu_{\mu,p}\mu'_{\mu,q}}{\sigma^2_{\mu,p}} - \frac{\sigma'_{\mu,q}}{\sigma_{\mu,q}} \\
+&\frac{\sigma_{\sigma,q}\sigma'_{\sigma,q} + \mu_{\sigma,q}\mu'_{\sigma,q} - \mu_{\sigma,p}\mu'_{\sigma,q}}{\sigma^2_{\sigma,p}} - \frac{\sigma'_{\sigma,q}}{\sigma_{\sigma,q}} \\
=&\frac{\mathcal{O}(M^{-1}) + \bar{\boldsymbol{x}}_k\mathbf{W}^{(u)}\bar{\boldsymbol{x}} - \mu_{\mu,p}\bar{\boldsymbol{x}}_k}{\sigma^2_{\mu,p}} - \mathcal{O}(M^{-1}) \\
+&\frac{\mathcal{O}(M^{-2}) + \sum_{i=1}^{K}\mathbf{W}^{(u,i)}\mathrm{Cov}(x_i, x_k) - \mu_{\sigma,p}\mathcal{O}(M^{-\frac{1}{2}})}{\sigma^2_{\sigma,p}} \\
-&\frac{\mathbb{E}[(\mathbf{W}^{(u)}\boldsymbol{x} - \mu^u)^4]'\sigma^u - 2(\sigma^u)^4(\sigma^u)' - 2(\sigma^u)'\mathbb{E}[(\mathbf{W}^{(u)}\boldsymbol{x} - \mu^u)^4]}{\mathbb{E}[(\mathbf{W}^{(u)}\boldsymbol{x} - \mu^u)^4]\sigma^u - (\sigma^u)^5}
\end{aligned}$$

where we summarize the terms scaled by M with $\mathcal{O}$-notation. We see that if we let $M \to \infty$, $\mu_{\mu,p} = 0$, $\sigma_{\mu,p} \to \infty$, $\mu_{\sigma,p} = 0$ and $\sigma_{\sigma,p}$ is small enough, then:

$$\frac{\partial}{\partial\mathbf{W}^{(u,k)}}\bigg(\mathrm{KL}(q_{\boldsymbol{\theta}}(\mu^u_{\mathbf{B}})||p(\mu^u_{\mathbf{B}})) + \mathrm{KL}(q_{\boldsymbol{\theta}}(\sigma^u_{\mathbf{B}})||p(\sigma^u_{\mathbf{B}}))\bigg) \approx \frac{\sum_{i=1}^{K}\mathbf{W}^{(u,i)}\mathrm{Cov}(x_i, x_k)}{\sigma^2_{\sigma,p}}$$

such that each BN layer yields the following:

$$\sum_u\sum_{i=1}^{K}\frac{\partial}{\partial\mathbf{W}^{(u,i)}}\bigg(\mathrm{KL}(q_{\boldsymbol{\theta}}(\mu^u_{\mathbf{B}})||p(\mu^u_{\mathbf{B}})) + \mathrm{KL}(q_{\boldsymbol{\theta}}(\sigma^u_{\mathbf{B}})||p(\sigma^u_{\mathbf{B}}))\bigg) \approx \sum_u\frac{\sum_{i=1}^{K}\mathbf{W}^{u,i}\sum_{i2=1}^{K}\mathrm{Cov}(x_i, x_{i2})}{\sigma^2_{\sigma,p,u}} \tag{8}$$

where we denote the prior for the std. dev. of the std. dev. of BN unit u by $\sigma_{\sigma,p,u}$. Given our assumptions of no scale and shift from the previous layer, and independent input features in every layer, Eq. 8 reduces to:

$$\sum_u \sum_{i=1}^{K} \frac{\mathbf{W}^{u,i}}{\sigma_{\sigma,p}^2}$$

if the same prior is chosen for each BN unit in the layer. We therefore find that Eq. 5 is reconciled by $p(\mu_{\mathbf{B}}^u) \to \mathcal{N}(0, \infty)$ and $p(\sigma_{\mathbf{B}}^u) \to \mathcal{N}(0, \frac{1}{2N\tau\lambda_l})$, if $\frac{1}{2N\tau\lambda_l}$ is small enough, which is the case if N is large.

1.5. predictive distribution properties

This section provides derivations of properties of the predictive distribution $p^*(\mathbf{y}|\mathbf{x}, \mathbf{D})$ in section 3.4, following (Gal, 2016). We first find the first two modes of the approximate predictive distribution (with the second mode applicable to regression), then show how to estimate the predictive log likelihood, a measure of uncertainty quality used in the evaluation.

Predictive mean Assuming Gaussian iid noise defined by model precision τ, i.e. $f_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|f_{\boldsymbol{\omega}}(\mathbf{x})) = \mathcal{N}(\mathbf{y}; f_{\boldsymbol{\omega}}(\mathbf{x}), \tau^{-1}\mathbf{I})$:

$$\begin{aligned}
\mathbb{E}_{p^*}[\mathbf{y}] &= \int \mathbf{y} p^*(\mathbf{y}|\mathbf{x}, \mathbf{D}) d\mathbf{y} \\
&= \int_{\mathbf{y}} \mathbf{y} \Big(\int_{\boldsymbol{\omega}} f_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{y}) q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) d\boldsymbol{\omega} \Big) \mathrm{d}\mathbf{y} \\
&= \int_{\mathbf{y}} \mathbf{y} \Big(\int_{\boldsymbol{\omega}} \mathcal{N}(\mathbf{y}; f_{\boldsymbol{\omega}}(\mathbf{x}), \tau^{-1}\mathbf{I}) q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \Big) \mathrm{d}\mathbf{y} \\
&= \int_{\boldsymbol{\omega}} \Big(\int_{\mathbf{y}} \mathbf{y} \mathcal{N}(\mathbf{y}; f_{\boldsymbol{\omega}}(\mathbf{x}), \tau^{-1}\mathbf{I}) \mathrm{d}\mathbf{y} \Big) q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \\
&= \int_{\boldsymbol{\omega}} f_{\boldsymbol{\omega}}(\mathbf{x}) q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \\
&\approx \frac{1}{T} \sum_{i=1}^{T} f_{\hat{\boldsymbol{\omega}}_i}(\mathbf{x})
\end{aligned}$$

where we take the MC Integral with T samples of $\boldsymbol{\omega}$ for the approximation in the final step.

Predictive variance For regression, our goal is to estimate:

$$\mathrm{Cov}_{p^*}[\mathbf{y}] = \mathbb{E}_{p^*}[\mathbf{y}^\intercal \mathbf{y}] - \mathbb{E}_{p^*}[\mathbf{y}]^\intercal \mathbb{E}_{p^*}[\mathbf{y}]$$

We find that:

$$\begin{aligned}
\mathbb{E}_{p^*}[\mathbf{y}^\intercal \mathbf{y}] &= \int_{\mathbf{y}} \mathbf{y}^\intercal \mathbf{y} p^*(\mathbf{y}|\mathbf{x}, \mathbf{D}) \mathrm{d}\mathbf{y} \\
&= \int_{\mathbf{y}} \mathbf{y}^\intercal \mathbf{y} \Big(\int_{\boldsymbol{\omega}} f_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{y}) q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) d\boldsymbol{\omega} \Big) \mathrm{d}\mathbf{y} \\
&= \int_{\boldsymbol{\omega}} \Big(\int_{\mathbf{y}} \mathbf{y}^\intercal \mathbf{y} f_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{y}) \mathrm{d}\mathbf{y} \Big) q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \\
&= \int_{\boldsymbol{\omega}} \Big(\mathrm{Cov}_{f_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{y})}(\mathbf{y}) + \mathbb{E}_{f_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{y})}[\mathbf{y}]^\intercal \mathbb{E}_{f_{\boldsymbol{\omega}}(\mathbf{x}, \mathbf{y})}[\mathbf{y}] \Big) q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \\
&= \int_{\boldsymbol{\omega}} \Big(\tau^{-1}\mathbf{I} + f_{\boldsymbol{\omega}}(\mathbf{x})^\intercal f_{\boldsymbol{\omega}}(\mathbf{x}) \Big) q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \\
&= \tau^{-1}\mathbf{I} + E_{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})}[f_{\boldsymbol{\omega}}(\mathbf{x})^\intercal f_{\boldsymbol{\omega}}(\mathbf{x})] \\
&\approx \tau^{-1}\mathbf{I} + \frac{1}{T} \sum_{i=1}^{T} f_{\hat{\boldsymbol{\omega}}_i}(\mathbf{x})^\intercal f_{\hat{\boldsymbol{\omega}}_i}(\mathbf{x})
\end{aligned}$$

where we use MC integration with T samples for the final step. The predictive covariance matrix is given by:

$$\text{Cov}_{p^*}[\mathbf{y}] \approx \tau^{-1}\mathbf{I} + \frac{1}{T}\sum_{i=1}^{T} f_{\hat{\boldsymbol{\omega}}_i}(\mathbf{x})^\intercal f_{\hat{\boldsymbol{\omega}}_i}(\mathbf{x}) - \mathbb{E}_{p^*}[\mathbf{y}]^\intercal \mathbb{E}_{p^*}[\mathbf{y}]$$

which is the sum of the variance from observation noise and the sample covariance from T stochastic forward passes though the network.

The form of p^* can be approximated by a Gaussian for each output dimension (for regression). We assume bounded domains for each input dimension, wide layers throughout the network, and a uni-modal distribution of weights centered at 0. By the Liapounov CLT condition, the first layer then receives approximately Gaussian inputs (a proof can be found in (Lehmann, 1999)). Having sampled $\mu_{\mathbf{B}}^u$ and $\sigma_{\mathbf{B}}^u$ from a mini-batch, each BN unit's output is bounded. CLT thereby continues to hold for deeper layers, including $f_{\boldsymbol{\omega}}(\mathbf{x}) = \mathbf{W}^L\mathbf{x}^L$. A similar motivation for a Gaussian approximation of Dropout has been presented by (Wang & Manning, 2013).

Predictive Log Likelihood We use the Predictive Log Likelihood (PLL) as a measure to estimate the model's uncertainty quality. For a certain test point $(\mathbf{y}_i, \mathbf{x}_i)$, the PLL definition and approximation can be expressed as:

$$\begin{aligned}
\text{PLL}(f_{\boldsymbol{\omega}}(\mathbf{x}), (\mathbf{y}_i, \mathbf{x}_i)) &= \log p(\mathbf{y}_i | f_{\boldsymbol{\omega}}(\mathbf{x}_i)) \\
&= \log \int f_{\boldsymbol{\omega}}(\mathbf{x}_i, \mathbf{y}_i) p(\boldsymbol{\omega}|\mathbf{D}) \text{d}\boldsymbol{\omega} \\
&\approx \log \int f_{\boldsymbol{\omega}}(\mathbf{x}_i, \mathbf{y}_i) q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) \text{d}\boldsymbol{\omega} \\
&\approx \log \frac{1}{T}\sum_{j=1}^{T} p(\mathbf{y}_i | f_{\hat{\boldsymbol{\omega}}_j}(\mathbf{x}_i))
\end{aligned}$$

where $\hat{\boldsymbol{\omega}}_j$ represents a sampled set of stochastic parameters from the approximate posterior distrubtion $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ and we take a MC integration with T samples. For regression, due to the iid Gaussian noise, we can further develop the derivation into the form we use when sampling:

$$\begin{aligned}
\text{PLL}(f_{\boldsymbol{\omega}}(\mathbf{x}), (\mathbf{y}_i, \mathbf{x}_i)) &= \log \frac{1}{T}\sum_{j=1}^{T} \mathcal{N}(\mathbf{y}_i | f_{\hat{\boldsymbol{\omega}}_j}(\mathbf{x}_i), \tau^{-1}\mathbf{I}) \\
&= \text{logsumexp}_{j=1,\dots,T}\big(-\frac{1}{2}\tau||\mathbf{y}_i - f_{\hat{\boldsymbol{\omega}}_j}(\mathbf{x}_i)||^2\big) \\
&\quad - \log T - \frac{1}{2}\log 2\pi + \frac{1}{2}\log \tau
\end{aligned}$$

Note that PLL makes no assumption on the form of the approximate predictive distribution.

1.6. Data

To assess the uncertainty quality of the various methods studied we rely on eight standard regression datasets, listed in Table 1. Publicly available from the UCI Machine Learning Repository (University of California, 2017) and Delve (Ghahramani, 1996), these datasets have been used to benchmark comparative models in recent related literature (see (Hernández-Lobato & Adams, 2015), (Gal & Ghahramani, 2015), (Bui et al., 2016) and (Li & Gal, 2017)).

For image classification, we applied MCBN using ResNet32 to CIFAR10.

For the image segmentation task, we applied MCBN using Bayesian SegNet on data from CamVid and PASCAL-VOC using models published in (Kendall et al., 2015).

Table 1. **Regression dataset summary.** Properties of the eight regression datasets used to evaluate MCBN. N is the dataset size and Q is the n.o. input features. Only one target feature was used – we used heating load for the Energy Efficiency dataset, which contains multiple target features.

Dataset name	N	Q
Boston Housing	506	13
Concrete Compressive Strength	1,030	8
Energy Efficiency	768	8
Kinematics 8nm	8,192	8
Power Plant	9,568	4
Protein Tertiary Structure	45,730	9
Wine Quality (Red)	1,599	11
Yacht Hydrodynamics	308	6

1.7. Extended experimental results

Below, we provide extended results measuring uncertainty quality. In Tables 2 and 3, we provide tables showing the mean $\overline{\text{CRPS}}$ and $\overline{\text{PLL}}$ values for MCBN and MCDO. These results indicate that MCBN performs on par or better than MCDO across several datasets. In Table 4 we provide the raw PLL and CRPS results for MCBN and MCDO. In Table 5 we provide RMSE results of the MCBN and MCDO networks in comparison with non-stochastic BN and DO networks. These results indicate that the procedure of multiple forward passes in MCBN and MCDO show slight improvements in the accuracy of the network.

In Figure 2 and Figure 3, we provide a full set of our uncertainty quality visualization plots, where errors in predictions are sorted by estimated uncertainty. The shaded areas show the model uncertainty and gray dots show absolute prediction errors on the test set. A gray line depicts a running mean of the errors. The dashed line indicates the optimized constant uncertainty. In these plots, we can see a correlation between estimated uncertainty (shaded area) and mean error (gray). This trend indicates that the model uncertainty estimates can recognize samples with larger (or smaller) potential for predictive errors.

We also conduct a sensitivity analysis to estimate how the uncertainty quality varies with batch size M and the number of stochastic forward passes T. In tables 6 and 7 we evaluate $\overline{\text{CRPS}}$ and $\overline{\text{PLL}}$ respectively for the regression datasets when trained and evaluated with varying batch sizes, but other hyperparameters fixed (T was fixed at 100). The results show that results deteriorate when batch sizes are too small, likely stemming from the large variance of the approximate posterior. In tables 8 and 9 we evaluate $\overline{\text{CRPS}}$ and $\overline{\text{PLL}}$ respectively for the regression datasets when trained and evaluated with varying n.o. stochastic forward samples, but other hyperparameters fixed (M was fixed at 128). The results are indicative of performance improvements with larger T, although we see improvements over baseline for some datasets already with $T = 50$ ($1/10$:th of the T used in our main experiments).

Table 2. **Uncertainty quality measured by $\overline{\text{CRPS}}$ on regression dasets.** $\overline{\text{CRPS}}$ measured on eight datasets over 5 random 80-20 splits of the data with 5 different random seeds each split. Mean values for MCBN, MCDO and MNF are reported along with standard error. A significance test was performed to check if $\overline{\text{CRPS}}$ significantly exceeds the baseline. The p-value from a one sample t-test is reported.

	$\overline{\text{CRPS}}$					
Dataset	MCBN	p-value	MCDO	p-value	MNF	p-value
Boston Housing	8.50±0.86	6.39E-10	3.06±0.33	1.64E-09	5.88±1.09	2.01E-05
Concrete	3.91±0.25	4.53E-14	0.93±0.41	3.13E-02	3.13±0.81	6.43E-04
Energy Efficiency	5.75±0.52	6.71E-11	1.37±0.89	1.38E-01	1.10±2.63	6.45E-01
Kinematics 8nm	2.85±0.18	2.33E-14	1.82±0.14	1.64E-12	0.52±0.26	7.15E-02
Power Plant	0.24±0.05	2.32E-04	-0.44±0.05	2.17E-08	-0.89±0.15	3.36E-06
Protein	2.66±0.10	2.77-12	0.99±0.08	2.34E-12	0.57±0.03	8.56E-16
Wine Quality (Red)	0.26±0.07	1.26E-03	2.00±0.21	1.83E-09	0.93±0.12	6.19E-08
Yacht Hydrodynamics	-56.39±14.27	5.94E-04	21.42±2.99	2.16E-07	24.92±3.77	9.62E-06

Table 3. **Uncertainty quality measured by $\overline{\text{PLL}}$ on regression dasets.** $\overline{\text{PLL}}$ measured on eight datasets over 5 random 80-20 splits of the data with 5 different random seeds each split. Mean values for MCBN, MCDO and MNF are reported along with standard error. A significance test was performed to check if $\overline{\text{PLL}}$ significantly exceeds the baseline. The p-value from a one sample t-test is reported.

			$\overline{\text{PLL}}$			
Dataset	MCBN	p-value	MCDO	p-value	MNF	p-value
Boston Housing	10.49±1.35	5.41E-08	5.51±1.05	2.20E-05	1.76±1.12	1.70E-01
Concrete	-36.36±12.12	6.19E-03	10.92±1.78	2.34E-06	-2.16±4.19	6.79E-01
Energy Efficiency	10.89±1.16	1.79E-09	-14.28±5.15	1.06E-02	-33.88±29.57	2.70E-01
Kinematics 8nm	1.68±0.37	1.29E-04	-0.26±0.18	1.53E-01	0.42±0.43	2.70E-01
Power Plant	0.33±0.14	2.72E-02	3.52±0.23	1.12E-13	-0.86±0.15	7.33E-06
Protein	2.56±0.23	4.28E-11	6.23±0.19	2.57E-21	0.52±0.07	1.81E-07
Wine Quality (Red)	0.19±0.09	3.72E-02	2.91±0.35	1.84E-08	0.83±0.16	2.27E-05
Yacht Hydrodynamics	45.58±5.18	5.67E-09	-41.54±31.37	1.97E-01	46.19±4.45	2.47E-07

Table 4. **Raw (unnormalized) CRPS and PLL scores on regression datasets.** CRPS and PLL measured on eight datasets over 5 random 80-20 splits of the data with 5 different random seeds each split. Mean values and standard errors are reported for MCBN, MCDO and MNF.

	CRPS			PLL		
Dataset	MCBN	MCDO	MNF	MCBN	MCDO	MNF
Boston Housing	1.45±0.02	1.41±0.02	1.57±0.02	-2.38±0.02	-2.35±0.02	-2.51±0.06
Concrete	2.40±0.04	2.42±0.04	3.61±0.02	-3.45±0.11	-2.94±0.02	-3.35±0.04
Energy Efficiency	0.33±0.01	0.26±0.00	1.33±0.04	-0.94±0.04	-0.80±0.04	-3.18±0.07
Kinematics 8nm	0.04±0.00	0.04±0.00	0.05±0.00	1.21±0.01	1.24±0.00	1.04±0.00
Power Plant	2.00±0.01	2.00±0.01	2.31±0.01	-2.75±0.00	-2.72±0.01	-2.86±0.01
Protein	1.95±0.01	1.95±0.00	2.25±0.01	-2.73±0.00	-2.70±0.00	-2.83±0.01
Wine Quality (Red)	0.34±0.00	0.33±0.00	0.34±0.00	-0.95±0.01	-0.89±0.01	-0.93±0.00
Yacht Hydrodynamics	0.68±0.02	0.32±0.01	0.94±0.01	-1.39±0.03	-2.57±0.69	-1.96±0.05

Table 5. **Prediction accuracy measured by RMSE on regression datasets.** RMSE measured on eight datasets over 5 random 80-20 splits of the data with 5 different random seeds each split. Mean values and standard errors are reported for for MCBN, MCDO and MNF as well as conventional non-Bayesian models BN and DO.

			RMSE		
Dataset	MCBN	BN	MCDO	DO	MNF
Boston Housing	2.75±0.05	2.77±0.05	2.65±0.05	2.69±0.05	2.98±0.06
Concrete	4.78±0.09	4.89±0.08	4.80±0.10	4.99±0.10	6.57±0.04
Energy Efficiency	0.59±0.02	0.57±0.01	0.47±0.01	0.49±0.01	2.38±0.07
Kinematics 8nm	0.07±0.00	0.07±0.00	0.07±0.00	0.07±0.00	0.09±0.00
Power Plant	3.74±0.01	3.74±0.01	3.74±0.02	3.72±0.02	4.19±0.01
Protein	3.66±0.01	3.69±0.01	3.66±0.01	3.68±0.01	4.10±0.01
Wine Quality (Red)	0.62±0.00	0.62±0.00	0.60±0.00	0.61±0.00	0.61±0.00
Yacht Hydrodynamics	1.23±0.05	1.28±0.06	0.75±0.03	0.72±0.04	2.13±0.05

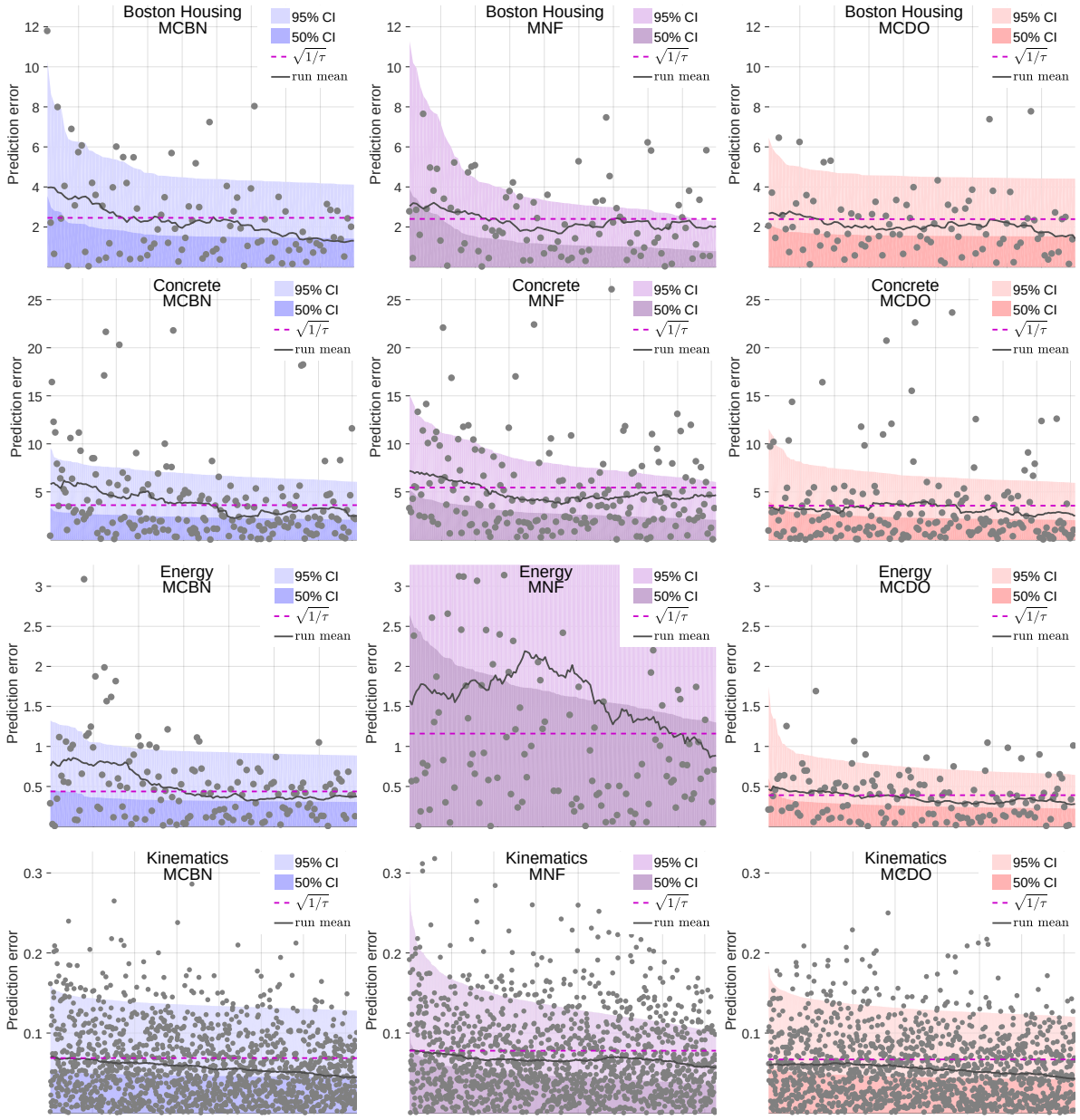

Figure 2. Errors in predictions (gray dots) sorted by estimated uncertainty on select datasets. The shaded areas show model uncertainty (light area 95% CI, dark area 50% CI). Gray dots show absolute prediction errors on the test set, and the gray line depicts a running mean of the errors. The dashed line indicates the optimized constant uncertainty. A correlation between estimated uncertainty (shaded area) and mean error (gray) indicates the uncertainty estimates are meaningful for estimating errors.

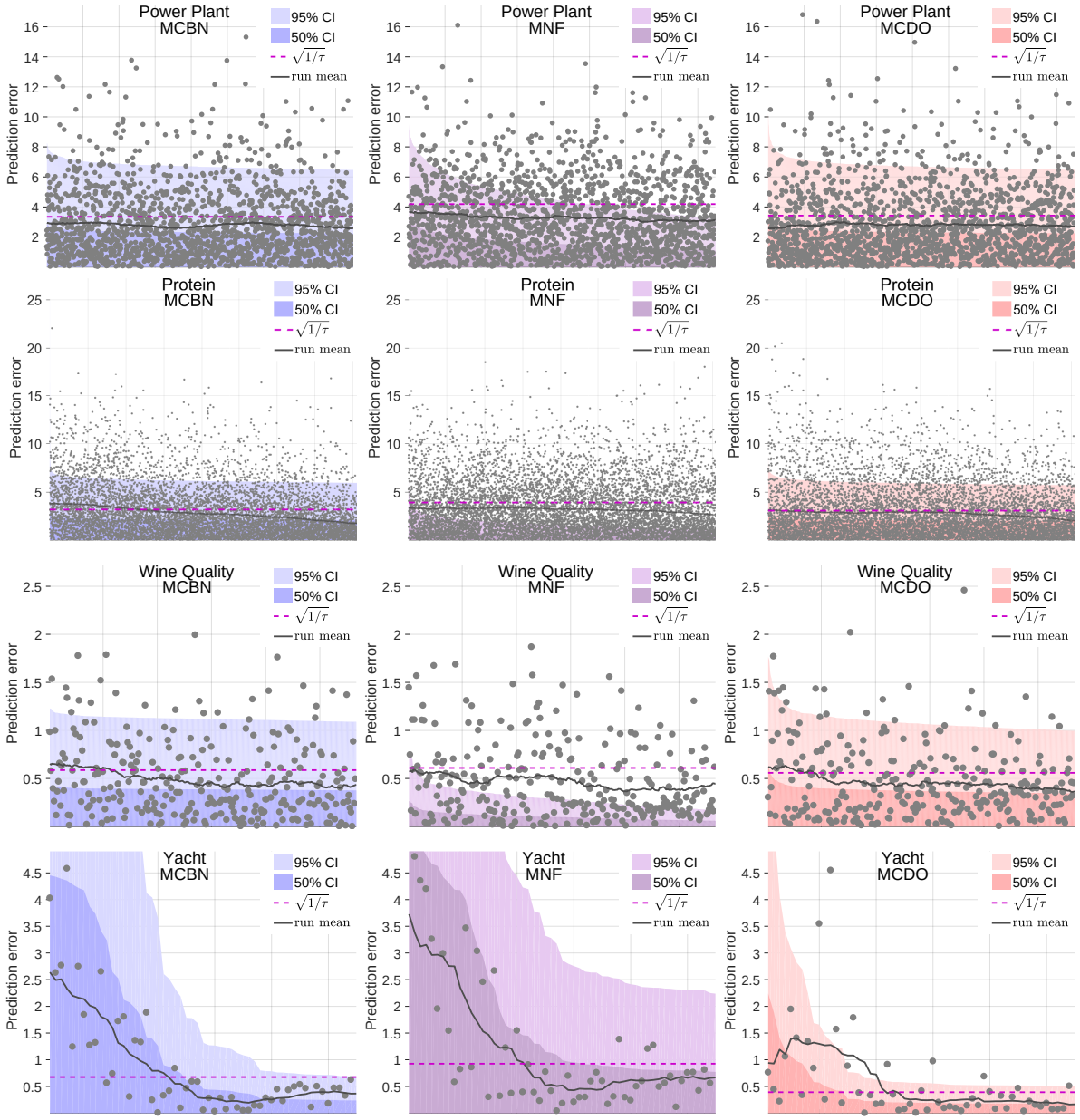

Figure 3. Errors in predictions (gray dots) sorted by estimated uncertainty on select datasets. The shaded areas show model uncertainty (light area 95% CI, dark area 50% CI). Gray dots show absolute prediction errors on the test set, and the gray line depicts a running mean of the errors. The dashed line indicates the optimized constant uncertainty. A correlation between estimated uncertainty (shaded area) and mean error (gray) indicates the uncertainty estimates are meaningful for estimating errors.

Table 6. **Uncertainty quality sensitivity to batch size.** A sensitivity analysis to determine how MCBN uncertainty quality varies with batch size is measured on eight regression datasets using $\overline{\text{CRPS}}$ as the quality measure. Results are measured over 3 random 80-20 splits of the data with 5 different random seeds each split.

	$\overline{\text{CRPS}}$							
Batch size	8	16	32	64	128	256	512	1024
Boston Housing	-7.1	**16.6**	11.8	7.2	2.5	0.9	-	-
Concrete	-34.5	**6.0**	5.0	5.1	2.9	1.4	0.6	0.0
Energy Efficiency	-61.6	-3.0	2.7	9.8	**11.1**	0.8	4.9	-
Kinematics 8nm	-1.4	-4.3	0.2	2.8	**2.7**	1.7	0.9	0.5
Power Plant	-10.5	0.8	0.0	-0.1	0.0	0.0	**0.2**	0.0
Protein	**14.5**	4.8	3.6	2.8	2.5	1.6	1.0	0.5
Wine Quality (Red)	**2.2**	1.6	0.6	0.6	0.3	0.0	0.2	0.0
Yacht Hydrodynamics	15.1	-23.0	-30.4	21.0	**34.4**	-	-	-

Table 7. **Uncertainty quality sensitivity to batch size.** A sensitivity analysis to determine how MCBN uncertainty quality varies with batch size is measured on eight regression datasets using $\overline{\text{PLL}}$ as the quality measure. Results are measured over 3 random 80-20 splits of the data with 5 different random seeds each split.

	$\overline{\text{PLL}}$							
Batch size	8	16	32	64	128	256	512	1024
Boston Housing	**13.9**	-36.7	10.0	7.9	3.7	1.5	-	-
Concrete	-113.3	-528.4	-10.0	**2.9**	0.0	1.4	0.2	0.0
Energy Efficiency	-64.4	5.2	-0.2	-9.6	-14.5	1.4	**10.4**	-
Kinematics 8nm	-4.9	-5.4	-3.1	1.6	**2.3**	1.5	0.7	0.4
Power Plant	-135.0	-1.4	-1.0	-1.1	-0.4	0.1	-0.1	**0.4**
Protein	**44.9**	15.7	4.6	2.9	2.8	2.2	1.2	0.6
Wine Quality (Red)	**2.2**	2.0	0.0	0.5	0.6	0.4	0.0	0.0
Yacht Hydrodynamics	**99.6**	74.9	76.8	48.5	44.9	-	-	-

Table 8. **Uncertainty quality sensitivity to n.o. stochastic forward passes.** A sensitivity analysis to determine how MCBN uncertainty quality varies with the n.o. stochastic forward passes measured on eight regression datasets using $\overline{\text{CRPS}}$ as the quality measure. Results are measured over 3 random 80-20 splits of the data with 5 different random seeds each split.

	$\overline{\text{CRPS}}$		
Forward passes	250	100	50
Boston Housing	**6.1**	2.7	3.2
Concrete	3.3	2.3	**3.3**
Energy Efficiency	**13.2**	4.2	7.9
Kinematics 8nm	3.2	2.7	**4.2**
Power Plant	0.2	**0.5**	0.1
Protein	2.3	**2.7**	2.4
Wine Quality (Red)	**0.9**	-0.4	0.6
Yacht Hydrodynamics	**32.9**	32.2	32.1

Table 9. **Uncertainty quality sensitivity to n.o. stochastic forward passes.** A sensitivity analysis to determine how MCBN uncertainty quality varies with the n.o. stochastic forward passes measured on eight regression datasets using $\overline{\text{PLL}}$ as the quality measure. Results are measured over 3 random 80-20 splits of the data with 5 different random seeds each split.

	$\overline{\text{PLL}}$		
Forward passes	250	100	50
Boston Housing	**7.8**	1.9	2.6
Concrete	3.8	**7.1**	0.1
Energy Efficiency	**15.7**	-30.5	-47.3
Kinematics 8nm	2.5	2.2	**3.4**
Power Plant	-0.9	**0.7**	-0.9
Protein	1.8	2.0	**2.4**
Wine Quality (Red)	**1.7**	-0.9	1.1
Yacht Hydrodynamics	**38.0**	35.9	35.5

1.8. Uncertainty in image segmentation

We applied MCBN to an image segmentation task using Bayesian SegNet with the main CamVid and PASCAL-VOC models in (Kendall et al., 2015). Here, we provide more image from Pascal VOC dataset in Figure 4.

1.9. Batch normalization statistics

In Figure 5 and Figure 6, we provide statistics on the batch normalization parameters used for training. The plots show the distribution of BN mean and BN variance over different mini-batches of an actual training of Yacht dataset for one unit in the first hidden layer and the second hidden layer. Data is provided for different epochs and for different batch sizes.

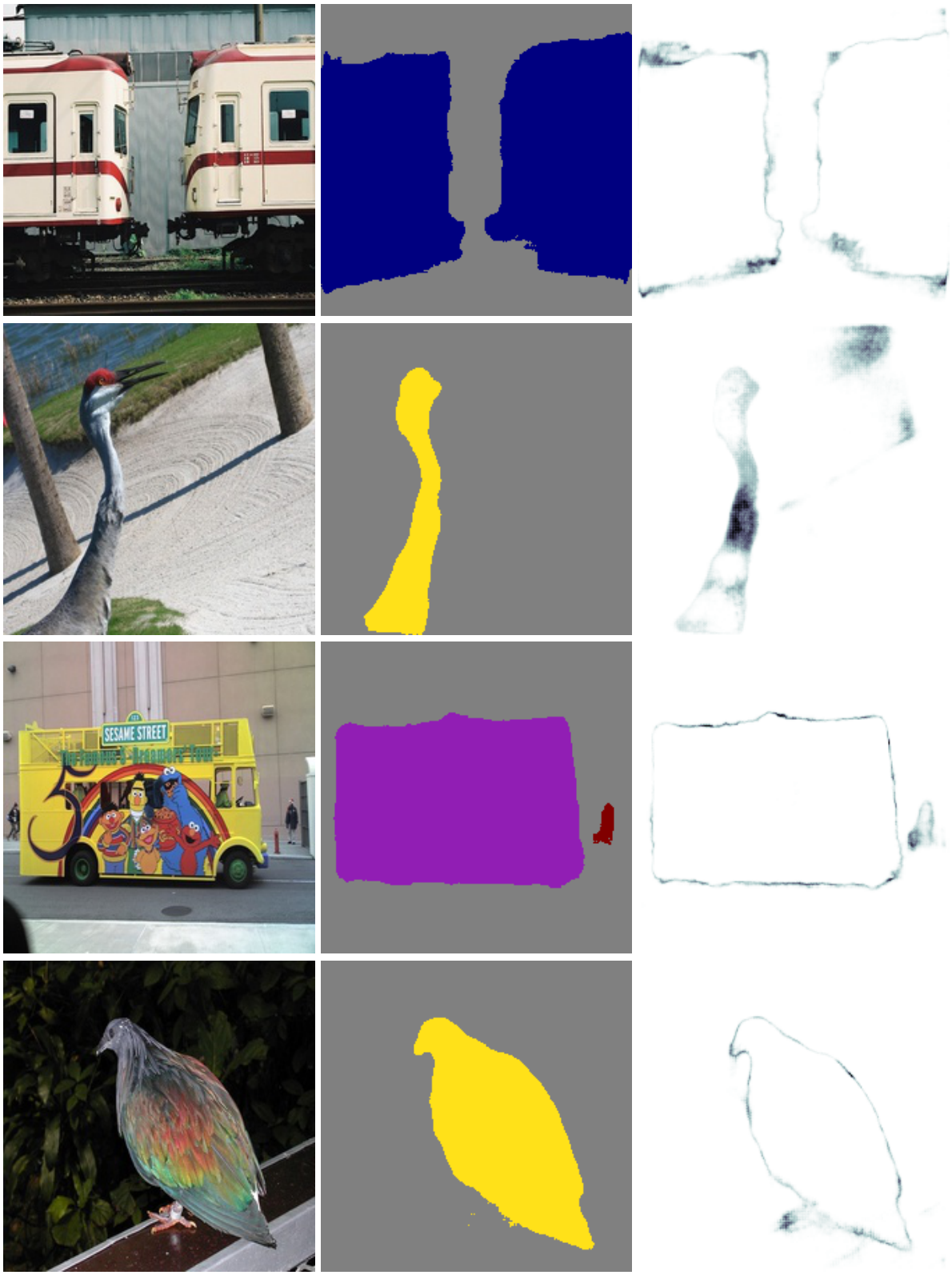

Figure 4. **Uncertainty in image segmentation**. Results applying MCBN to Bayesian SegNet (Kendall et al., 2015) on images from PASCAL-VOC (right). Left: original. Middle: the Bayesian estimated segmentation. Right: estimated uncertainty using MCBN for all classes. Mini-batches of size 36 were used for PASCAL-VOC on images of size 224x224. 20 inferences were conducted to estimate the mean and variance of MCBN.

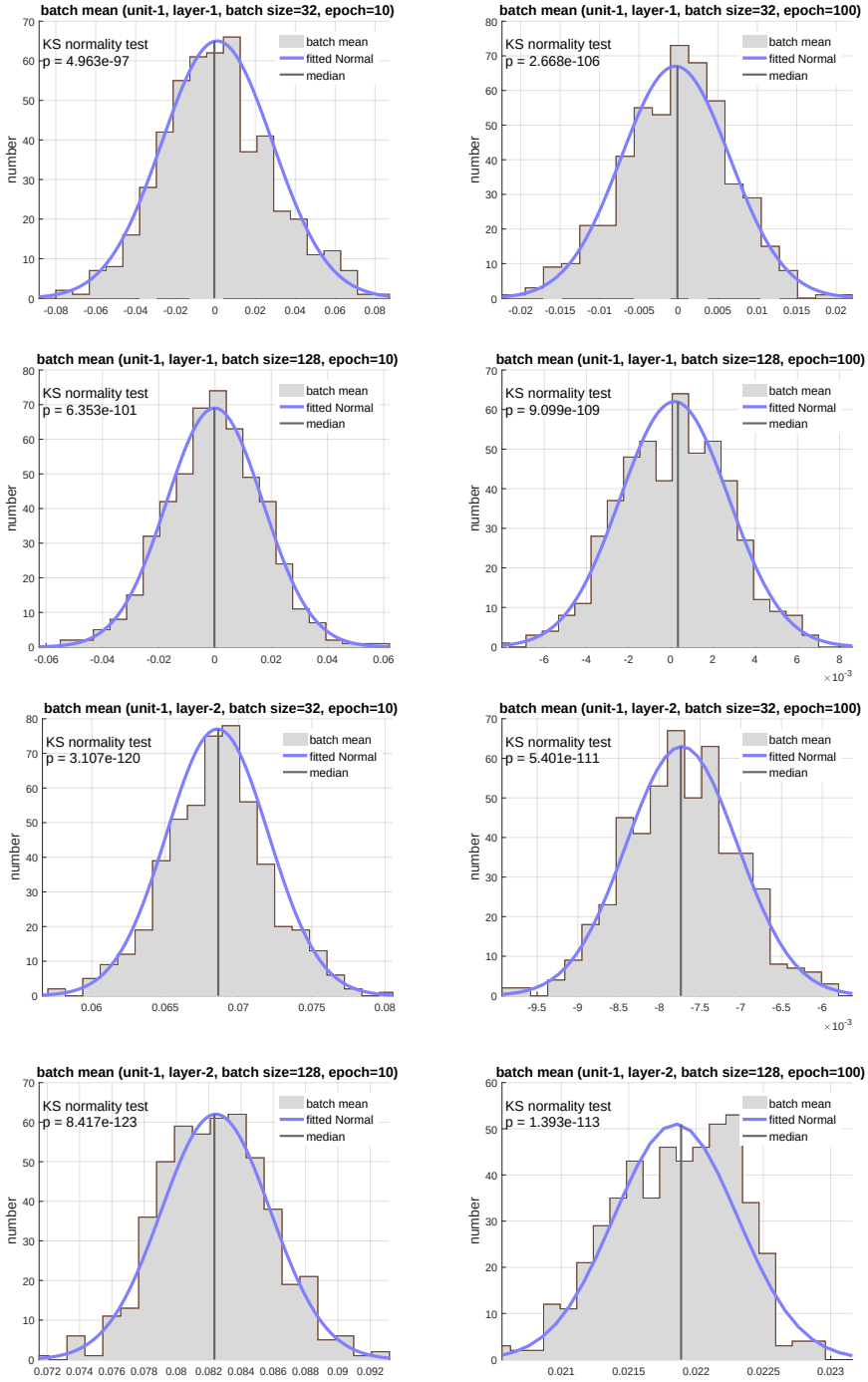

Figure 5. **The distribution of means** of mini-batches during training of one of our datasets. The distribution closely follows our analytically approximated Gaussian distribution. The data is collected for one unit of each layer and is provided for different epochs and for different batch sizes.

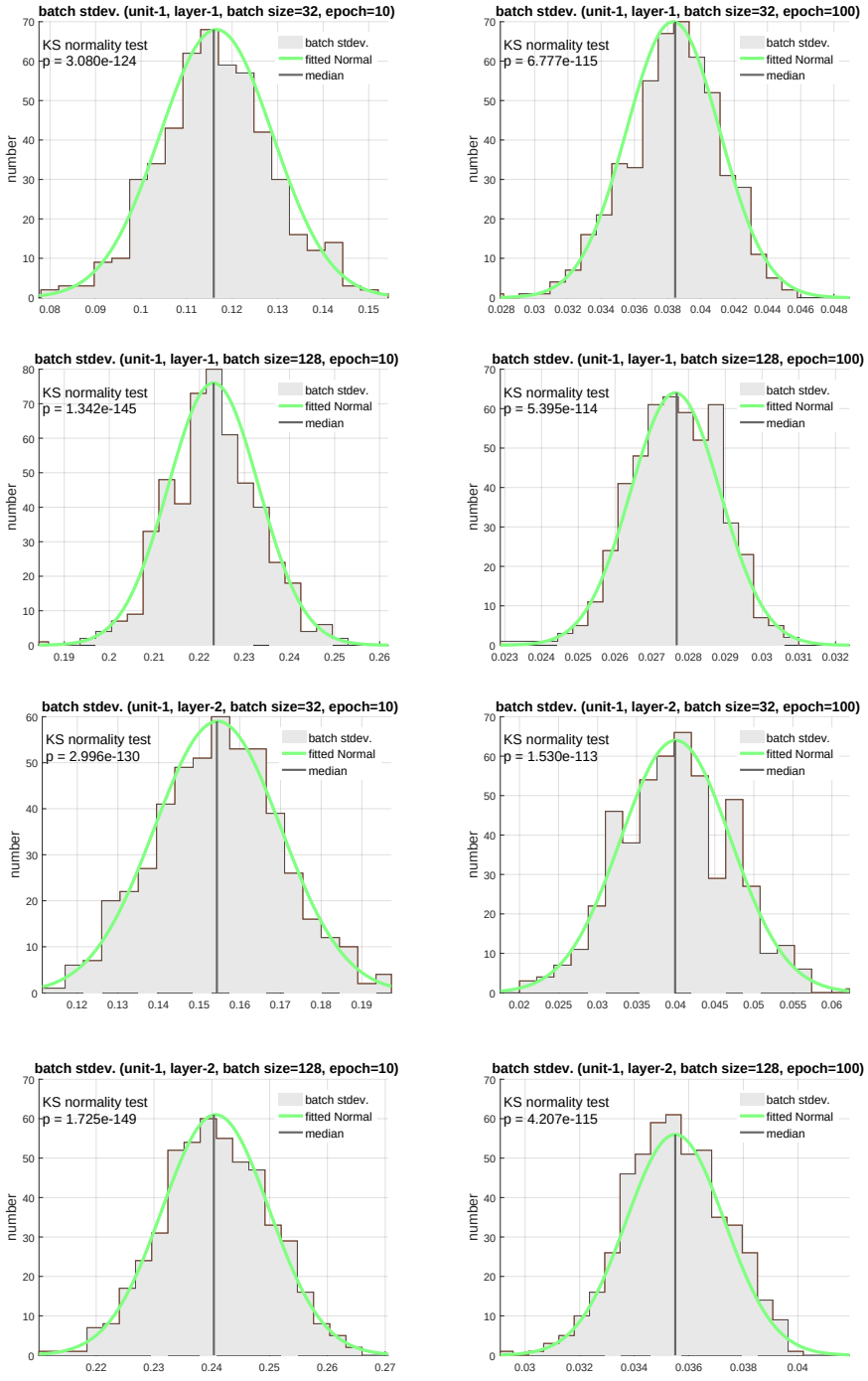

Figure 6. **The distribution of standard deviation** of mini-batches during training of one of our datasets. The distribution closely follows our analytically approximated Gaussian distribution. The data is collected for one unit of each layer and is provided for different epochs and for different batch sizes.

References

Bui, T. D., Hernández-Lobato, D., Li, Y., Hernández-Lobato, J. M., and Turner, R. E. Deep Gaussian Processes for Regression using Approximate Expectation Propagation. In *ICML*, 2016.

Gal, Y. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian Approximation : Representing Model Uncertainty in Deep Learning. *ICML*, 48:1–10, 2015.

Ghahramani, Z. *Delve Datasets*. University of Toronto, 1996. URL http://www.cs.toronto.edu/{~}delve/data/kin/desc.html.

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

Kendall, A., Badrinarayanan, V., and Cipolla, R. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. *CoRR*, abs/1511.0, 2015. URL http://arxiv.org/abs/1511.02680.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *ICLR*, 2014.

Lehmann, E. L. *Elements of Large-Sample Theory*. Springer Verlag, New York, 1999. ISBN 0387985956.

Li, Y. and Gal, Y. Dropout Inference in Bayesian Neural Networks with Alpha-divergences. *arXiv*, 2017.

University of California, I. UC Irvine Machine Learning Repository, 2017. URL https://archive.ics.uci.edu/ml/index.html.

Wang, S. I. and Manning, C. D. Fast dropout training. *Proceedings of the 30th International Conference on Machine Learning*, 28:118–126, 2013. URL http://machinelearning.wustl.edu/mlpapers/papers/wang13a.