Appendix to Uncertainty in Neural Networks: Approximately Bayesian Ensembling

A Proofs

Definition 1. Data likelihood and parameter likelihood

We take care to define two versions of the likelihood, one in output space, $P_{\mathcal{D}}(\mathcal{D}|\boldsymbol{\theta})$ (data likelihood), and one in parameter space, $P_{\boldsymbol{\theta}}(\mathcal{D}|\boldsymbol{\theta})$ (parameter likelihood). Both return the same values given some data set $\mathcal{D}$ and parameter values $\boldsymbol{\theta}$, and hence are exchangeable, but their forms are subtly different.

The data likelihood, $P_{\mathcal{D}}(\mathcal{D}|\boldsymbol{\theta})$, is defined on the output domain. Typically the log of this, $\log(P_{\mathcal{D}}(\mathcal{D}|\boldsymbol{\theta}))$, might be optimised as the cross entropy loss or (negative) mean squared error.

In contrast, $P_{\boldsymbol{\theta}}(\mathcal{D}|\boldsymbol{\theta})$ defines a likelihood function in the parameter domain.

Illustrative Example

Consider a linear regression model with dataset, $\mathcal{D}$, consisting of tuples, $\{\mathbf{x}, y\}$; a vector of predictor variables $\mathbf{x} \in \mathbb{R}^p$, predicting a single scalar $y \in \mathbb{R}$. If the model is of the form, $\boldsymbol{\theta}^T\mathbf{x}$, a Gaussian data likelihood with variance σ_ϵ^2 on the output might be assumed.

This leads to a data likelihood for the target, y,

$$P_{\mathcal{D}}(\mathcal{D}|\boldsymbol{\theta}) = \mathcal{N}(y|\boldsymbol{\theta}^T\mathbf{x}, \sigma_\epsilon^2). \tag{10}$$

For this linear model, the corresponding parameter likelihood is a multivariate normal distribution,

$$P_{\boldsymbol{\theta}}(\mathcal{D}|\boldsymbol{\theta}) \propto \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{like}, \boldsymbol{\Sigma}_{like}). \tag{11}$$

where, $\boldsymbol{\mu}_{like} \in \mathbb{R}^p$ & $\boldsymbol{\Sigma}_{like} \in \mathbb{R}^{p\times p}$, can be found analytically. They are implicitly functions of the dataset, $\mathcal{D}$, although to lighten notation we do not write this. Subsequently we also drop the explicit referral to $\boldsymbol{\theta}$.

Note that whilst both the data and parameter likelihood follow a normal distribution, they are defined in different domains.

The correspondence between a Gaussian data likelihood and multivariate normal parameter likelihood is only exact for a linear regression model. For non-linear models with Gaussian data likelihoods, and other data likelihoods, the parameter likelihood is not in general multivariate normal. Nevertheless it can be convenient to model it as such.

Standard Result 1. Product of two multivariate Gaussians (§8.1.8, The Matrix Cookbook, 2008)

$$\mathcal{N}(\boldsymbol{\mu}_{like}, \boldsymbol{\Sigma}_{like})\mathcal{N}(\boldsymbol{\mu}_{prior}, \boldsymbol{\Sigma}_{prior}) \propto \mathcal{N}(\boldsymbol{\mu}_{post}, \boldsymbol{\Sigma}_{post}) \tag{12}$$

$$\boldsymbol{\Sigma}_{post} = (\boldsymbol{\Sigma}_{prior}^{-1} + \boldsymbol{\Sigma}_{like}^{-1})^{-1}, \tag{13}$$

$$\boldsymbol{\mu}_{post} = \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\mu}_{prior} + \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{like}^{-1}\boldsymbol{\mu}_{like}. \tag{14}$$

Standard Result 2. Affine transform of a normal random variable (§8.1.4, The Matrix Cookbook, 2008)

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{15}$$

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}, \tag{16}$$

$$\mathbf{y} \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T). \tag{17}$$

Theorem 1. *Assume that a model's parameter likelihood follows a multivariate normal distribution,* $P_{\boldsymbol{\theta}}(\mathcal{D}|\boldsymbol{\theta}) \propto \mathcal{N}(\boldsymbol{\mu}_{like}, \boldsymbol{\Sigma}_{like})$*, and the prior also,* $P(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{prior}, \boldsymbol{\Sigma}_{prior})$*. The posterior is then also multivariate normal,* $P(\boldsymbol{\theta}|\mathcal{D}) = \mathcal{N}(\boldsymbol{\mu}_{post}, \boldsymbol{\Sigma}_{post})$*.*

Further assume availability of some function which returns MAP parameter estimates taking as input the location of the prior centre, $\boldsymbol{f}_{MAP}(\boldsymbol{\theta}_{anc})$*. In order that,* $P(\boldsymbol{f}_{MAP}(\boldsymbol{\theta}_{anc})) = P(\boldsymbol{\theta}|\mathcal{D})$*, then the required distribution of* $\boldsymbol{\theta}_{anc}$ *is also multivariate normal,* $P(\boldsymbol{\theta}_{anc}) = \mathcal{N}(\boldsymbol{\mu}_{anc}, \boldsymbol{\Sigma}_{anc})$*, where,* $\boldsymbol{\mu}_{anc} = \boldsymbol{\mu}_{prior}$*, and,* $\boldsymbol{\Sigma}_{anc} = \boldsymbol{\Sigma}_{prior} + \boldsymbol{\Sigma}_{prior}\boldsymbol{\Sigma}_{like}^{-1}\boldsymbol{\Sigma}_{prior}$*.*

Proof. Consider a model's parameters having a multivariate normal prior,

$$P(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{prior}, \boldsymbol{\Sigma}_{prior}), \tag{18}$$

where, $\boldsymbol{\theta} \in \mathbb{R}^p$, $\boldsymbol{\mu}_{prior} \in \mathbb{R}^p$, $\boldsymbol{\Sigma}_{prior} \in \mathbb{R}^{p \times p}$.

This theorem makes the assumption that the form of the parameter likelihood (def. 1) is multivariate normal,

$$P_{\boldsymbol{\theta}}(\mathcal{D}|\boldsymbol{\theta}) \propto \mathcal{N}(\boldsymbol{\mu}_{like}, \boldsymbol{\Sigma}_{like}) \tag{19}$$

where, $\boldsymbol{\mu}_{like} \in \mathbb{R}^p$, $\boldsymbol{\Sigma}_{like} \in \mathbb{R}^{p \times p}$. Here $\propto$ is used since it is not a true probability distribution in $\boldsymbol{\theta}$ so need not sum to 1.

The posterior is calculated by Bayes rule. Recalling that data likelihood and parameter likelihood are exchangeable (def. 1), and using Standard Result 1,

$$P(\boldsymbol{\theta}|\mathcal{D}) = \frac{P_{\mathcal{D}}(\mathcal{D}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\mathcal{D})} = \frac{P_{\boldsymbol{\theta}}(\mathcal{D}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\mathcal{D})} \propto \mathcal{N}(\boldsymbol{\mu}_{like}, \boldsymbol{\Sigma}_{like})\mathcal{N}(\boldsymbol{\mu}_{prior}, \boldsymbol{\Sigma}_{prior}) \propto \mathcal{N}(\boldsymbol{\mu}_{post}, \boldsymbol{\Sigma}_{post}), \tag{20}$$

where, $\boldsymbol{\mu}_{post}$ & $\boldsymbol{\Sigma}_{post}$ are given by eq. 14 & 13.

We introduce a further distribution, termed 'anchor distribution', which we enforce as multivariate normal,

$$P(\boldsymbol{\theta}_{anc}) = \mathcal{N}(\boldsymbol{\mu}_{anc}, \boldsymbol{\Sigma}_{anc}). \tag{21}$$

It will be used as described in the main text (see figure 2 and algorithm 1) so that samples are drawn from the anchor distribution, $\boldsymbol{\theta}_{anc} \sim P(\boldsymbol{\theta}_{anc})$, with a prior then recentred at each sample, denoted $P_{anc}(\boldsymbol{\theta})$,

$$P_{anc}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}_{anc}, \boldsymbol{\Sigma}_{prior}). \tag{22}$$

Note that this anchor distribution is in the same position as a hyperprior on $\boldsymbol{\mu}_{prior}$, but will have a subtly different role. $\boldsymbol{\Sigma}_{prior}$ is unchanged from eq. 18.

Denote $\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})$ as the MAP estimates given this recentred prior and the original likelihood from eq. 19.

$$\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc}) \coloneqq \text{argmax}_{\boldsymbol{\theta}} P_{anc}(\boldsymbol{\theta}) P_{\boldsymbol{\theta}}(\mathcal{D}|\boldsymbol{\theta}) \tag{23}$$

In order to prove the theorem, three things regarding $\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})$ must be shown:

1. Its distribution is multivariate normal - denote mean and covariance $\boldsymbol{\mu}_{post}^{\text{RMS}}, \boldsymbol{\Sigma}_{post}^{\text{RMS}}$,

$$P(\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})) = \mathcal{N}(\boldsymbol{\mu}_{post}^{\text{RMS}}, \boldsymbol{\Sigma}_{post}^{\text{RMS}}), \tag{24}$$

2. That $\boldsymbol{\mu}_{anc}$ & $\boldsymbol{\Sigma}_{anc}$ can be selected in such a way that the mean of the distribution is equal to that of the original posterior

$$\boldsymbol{\mu}_{post}^{\text{RMS}} = \boldsymbol{\mu}_{post}, \tag{25}$$

3. And also so that the covariance of the distribution is equal to that of the original posterior

$$\boldsymbol{\Sigma}_{post}^{\text{RMS}} = \boldsymbol{\Sigma}_{post}. \tag{26}$$

For a multivariate normal distribution, the MAP solution is simply equal to the mean of the posterior, $\boldsymbol{\mu}_{post}$. For the typical case this is given by eq. 14. In our procedure, the location of the prior mean has been replaced by $\boldsymbol{\theta}_{anc}$, so the MAP solution is given by,

$$\begin{aligned}
\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc}) = &\ \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\theta}_{anc} + \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{like}^{-1}\boldsymbol{\mu}_{like} && (27)\\
= &\ \mathbf{A}_1\boldsymbol{\theta}_{anc} + \mathbf{b}_1 && (28)
\end{aligned}$$

where two constants have been defined for convenience,

$$\begin{aligned}
\mathbf{A}_1 &= \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1} && (29)\\
\mathbf{b}_1 &= \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{like}^{-1}\boldsymbol{\mu}_{like}, && (30)
\end{aligned}$$

which is the same form as eq. 16. Hence, from Standard Result 2, if $\boldsymbol{\theta}_{anc}$ is normally distributed, $\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})$ **will also be normally distributed**.

Regarding the mean of $\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})$, we have,

$$\begin{aligned}
\mathbb{E}[\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})] &= \mathbb{E}[\mathbf{A}_1\boldsymbol{\theta}_{anc} + \mathbf{b}_1] && (31)\\
&= \mathbf{A}_1\mathbb{E}[\boldsymbol{\theta}_{anc}] + \mathbf{b}_1. && (32)
\end{aligned}$$

By choosing the anchor distribution to be centred about the original prior, $\mathbb{E}[\boldsymbol{\theta}_{anc}] = \boldsymbol{\mu}_{prior}$, we have,

$$\begin{aligned}
&= \mathbf{A}_1\boldsymbol{\mu}_{prior} + \mathbf{b}_1 && (33)\\
&= \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\mu}_{prior} + \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{like}^{-1}\boldsymbol{\mu}_{like}, && (34)
\end{aligned}$$

This is consistent with eq. 14 and proves that **the means of the distributions are aligned when $\boldsymbol{\mu}_{anc} = \boldsymbol{\mu}_{prior}$.**

Finally we consider the variance of $\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})$, which we wish to equal $\boldsymbol{\Sigma}_{post}$ by choosing $\boldsymbol{\Sigma}_{anc}$. Using the form from eq. 28 and applying Standard Result 2,

$$\begin{aligned}
\mathbb{V}ar[\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})] &= \mathbb{V}ar[\mathbf{A}_1\boldsymbol{\theta}_{anc} + \mathbf{b}_1] && (35)\\
&= \mathbf{A}_1\mathbb{V}ar[\boldsymbol{\theta}_{anc}]\mathbf{A}_1^T && (36)\\
&= \mathbf{A}_1\boldsymbol{\Sigma}_{anc}\mathbf{A}_1^T && (37)
\end{aligned}$$

We require $\mathbb{V}ar[\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})] = \boldsymbol{\Sigma}_{post}$.

$$\boldsymbol{\Sigma}_{post} = \mathbf{A}_1\boldsymbol{\Sigma}_{anc}\mathbf{A}_1^T. \tag{38}$$

Note that transposes of covariance matrices may be ignored since they are symmetric.

$$\begin{aligned}
\boldsymbol{\Sigma}_{anc} &= \mathbf{A_1}^{-1}\boldsymbol{\Sigma}_{post}\mathbf{A_1}^{-1T} && (39)\\
&= (\boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1})^{-1}\boldsymbol{\Sigma}_{post}(\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{post})^{-1} && (40)\\
&= \boldsymbol{\Sigma}_{prior}\boldsymbol{\Sigma}_{post}^{-1}\boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{post}^{-1}\boldsymbol{\Sigma}_{prior} && (41)\\
&= \boldsymbol{\Sigma}_{prior}\boldsymbol{\Sigma}_{post}^{-1}\boldsymbol{\Sigma}_{prior} && (42)\\
&= \boldsymbol{\Sigma}_{prior}(\boldsymbol{\Sigma}_{prior}^{-1} + \boldsymbol{\Sigma}_{like}^{-1})\boldsymbol{\Sigma}_{prior} && (43)\\
&= \boldsymbol{\Sigma}_{prior} + \boldsymbol{\Sigma}_{prior}\boldsymbol{\Sigma}_{like}^{-1}\boldsymbol{\Sigma}_{prior}. && (44)
\end{aligned}$$

This proves that **the covariances of the two distributions are aligned when $\boldsymbol{\Sigma}_{anc} = \boldsymbol{\Sigma}_{prior} + \boldsymbol{\Sigma}_{prior}\boldsymbol{\Sigma}_{like}^{-1}\boldsymbol{\Sigma}_{prior}$.**

□

Corollary 1.1. *Following from theorem 1 (and under the same assumptions), set* $\boldsymbol{\mu}_{anc} \coloneqq \boldsymbol{\mu}_{prior}$ *and* $\boldsymbol{\Sigma}_{anc} \coloneqq \boldsymbol{\Sigma}_{prior}$*. The RMS approximate posterior is* $P(\boldsymbol{f}_{MAP}(\boldsymbol{\theta}_{anc})) = \mathcal{N}(\boldsymbol{\mu}_{post}, \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{post})$.

Proof. Independent of the choice of anchor distribution covariance $\boldsymbol{\Sigma}_{anc}$, theorem 1 demonstrated that the resulting posterior, $P(\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc}))$ is normally distributed, with mean equal to that of the true posterior $\boldsymbol{\mu}_{post}$.

To discover the covariance of the resulting distribution, $\mathbb{V}ar[\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})]$, we take eq. 37 and simply set, $\boldsymbol{\Sigma}_{anc} \coloneqq \boldsymbol{\Sigma}_{prior}$.

$$\begin{aligned}
\mathbb{V}ar[\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})] &= \mathbf{A}_1\boldsymbol{\Sigma}_{anc}\mathbf{A}_1^T && (45)\\
&= \mathbf{A}_1\boldsymbol{\Sigma}_{prior}\mathbf{A}_1^T && (46)\\
&= \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{prior}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{post} && (47)\\
&= \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{post} && (48)
\end{aligned}$$

□

Lemma 1.1. *Following from corollary 1.1 (and under the same assumptions), when* $\boldsymbol{\mu}_{anc} \coloneqq \boldsymbol{\mu}_{prior}$, $\boldsymbol{\Sigma}_{anc} \coloneqq \boldsymbol{\Sigma}_{prior}$, *the RMS approximate posterior will in general underestimate the marginal variance compared to the true posterior,* $\mathbb{V}ar[\boldsymbol{f}_{MAP}(\theta_{anc})] < \mathbb{V}ar[\theta|\mathcal{D}]$.

Proof. We consider the marginal posterior of a single parameter, $\theta \coloneqq \boldsymbol{\theta}_i$, again assuming multivariate normal prior and parameter likelihood. First consider the following rearrangement of eq. 48, beginning by noting, $\boldsymbol{\Sigma}_{prior}^{-1} = \boldsymbol{\Sigma}_{post}^{-1} - \boldsymbol{\Sigma}_{like}^{-1}$.

$$\begin{aligned}
\boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{post} &= \boldsymbol{\Sigma}_{post}(\boldsymbol{\Sigma}_{post}^{-1} - \boldsymbol{\Sigma}_{like}^{-1})\boldsymbol{\Sigma}_{post} && (49)\\
&= (\mathbb{I} - \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{like}^{-1})\boldsymbol{\Sigma}_{post} && (50)\\
&= \boldsymbol{\Sigma}_{post} - \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{like}^{-1}\boldsymbol{\Sigma}_{post} && (51)
\end{aligned}$$

To show that RMS generally underestimates the marginal variance, it must hold that diagonal elements of the true posterior covariance matrix are greater than or equal to the same diagonal element of the RMS posterior.

$$\begin{aligned}
\mathbb{V}ar[\boldsymbol{f}_{\text{MAP}}(\theta_{anc})] &< \mathbb{V}ar[\theta|\mathcal{D}] && (52)\\
\text{diag}(\boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{post})_i &< \text{diag}(\boldsymbol{\Sigma}_{post})_i && (53)
\end{aligned}$$

substituting in the diagonal of the rearrangement in eq. 51,

$$\text{diag}(\boldsymbol{\Sigma}_{post})_i - \text{diag}(\boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{like}^{-1}\boldsymbol{\Sigma}_{post})_i < \text{diag}(\boldsymbol{\Sigma}_{post})_i \quad (54)$$

We know that ABA^T is positive definite if A, B are positive definite, and also that the inverse of a positive definite matrix is positive definite (§9.6.4, §9.6.10, The Matrix Cookbook, 2008). The diagonal of a positive definite matrix is positive. Hence, $\text{diag}(\boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{like}^{-1}\boldsymbol{\Sigma}_{post})_i > 0$, and we have shown that $\mathbb{V}ar[\boldsymbol{f}_{\text{MAP}}(\theta_{anc})] < \mathbb{V}ar[\theta|\mathcal{D}]$.

□

Lemma 1.2. *This lemma follows from corollary 1.1. Again parameter likelihood and prior are assumed normally distributed. The prior is additionally assumed isotropic. When* $\boldsymbol{\mu}_{anc} \coloneqq \boldsymbol{\mu}_{prior}$, $\boldsymbol{\Sigma}_{anc} \coloneqq \boldsymbol{\Sigma}_{prior}$ *the eigenvectors (or 'orientation') of the RMS approximate posterior equal those of the true posterior.*

Proof. From eq. 48, $\mathbb{V}ar[\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})] = \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{post}$. If the prior is isotropic, $\boldsymbol{\Sigma}_{prior} = \sigma_{prior}^2\mathbb{I}$, then, $\mathbb{V}ar[\boldsymbol{f}_{\text{MAP}}(\theta_{anc})] = 1/\sigma_{prior}^2\boldsymbol{\Sigma}_{post}^2$. Hence the prior only scales the eigenvalues, and doesn't affect the eigenvectors. (Note that this won't be the case for non-isotropic $\boldsymbol{\Sigma}_{prior}$.)

Consider some matrix A and a specific eigenvalue λ_i and eigenvector $\mathbf{v}_i$ so that, $A\mathbf{v}_i = \lambda_i\mathbf{v}_i$. It then follows that if A is squared, $A^2\mathbf{v}_i = A(A\mathbf{v}_i) = \lambda_i A\mathbf{v}_i = \lambda_i^2\mathbf{v}_i$. Hence eigenvalues are squared but eigenvectors are unaffected. This applies to the transformation $\boldsymbol{\Sigma}_{post}^2$.

Hence both the square and the multiplication of prior covariance, $1/\sigma_{prior}^2\boldsymbol{\Sigma}_{post}^2$, do not modify the original eigenvectors of $\boldsymbol{\Sigma}_{post}$ and its orientation is unaffected.

□

Theorem 2. *For a two parameter model with normally distributed parameter likelihood and isotropic prior, the RMS approximate posterior will in general overestimate the magnitude of the true posterior parameter correlation coefficient, $|\rho|$. However, if $|\rho| = 1$, then it will recover it precisely. We set $\boldsymbol{\mu}_{anc} \coloneqq \boldsymbol{\mu}_{prior}, \boldsymbol{\Sigma}_{anc} \coloneqq \boldsymbol{\Sigma}_{prior}$.*

Proof. From corollary 1.1, we have that the RMS approximate posterior is given by $\boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{post}$. Let $\boldsymbol{\Sigma}_{prior} \coloneqq \sigma_{prior}^2\mathbb{I}$, the RMS approximate posterior is then given by $1/\sigma_{prior}^2\boldsymbol{\Sigma}_{post}^2$. Denote the true posterior covariance as the following 2×2 matrix.

$$\boldsymbol{\Sigma}_{post} = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \tag{55}$$

For general covariance matrices, the correlation coefficient, ρ, can be found by solving, $b = \rho\sqrt{ac}$.

Our RMS approximate posterior is given as follows.

$$1/\sigma_{prior}^2\boldsymbol{\Sigma}_{post}^2 = 1/\sigma_{prior}^2 \begin{bmatrix} a & b \\ b & c \end{bmatrix}\begin{bmatrix} a & b \\ b & c \end{bmatrix} \tag{56}$$

$$= 1/\sigma_{prior}^2 \begin{bmatrix} a^2+b^2 & ab+bc \\ ab+bc & b^2+c^2 \end{bmatrix} \tag{57}$$

The correlation coefficient here, denoted ρ_{RMS}, is found by solving, $ab + bc = \rho_{\text{RMS}}\sqrt{(a^2+b^2)(b^2+c^2)}$.

To prove the correlation coefficient is generally overestimated, we must show that $\rho_{\text{RMS}}^2 > \rho^2$ when $\rho^2 < 1$.

$$\rho_{\text{RMS}}^2 > \rho^2 \tag{58}$$

$$\frac{(ab+bc)^2}{(a^2+b^2)(b^2+c^2)} > \frac{b^2}{ac} \tag{59}$$

$$(ab+bc)^2ac > b^2(a^2+b^2)(b^2+c^2) \tag{60}$$

$$(a+c)^2ac > (a^2+b^2)(b^2+c^2) \tag{61}$$

$$a^3c + ac^3 + 2a^2c^2 > a^2b^2 + a^2c^2 + b^4 + b^2c^2 \tag{62}$$

$$a^3c + ac^3 + a^2c^2 > a^2b^2 + b^4 + b^2c^2 \tag{63}$$

We now note that from the set up of the proof, $b^2 = \rho^2 ac$. Since $\rho^2 < 1$, we have, $b^2/ac < 1 \implies b^2 < ac$. We can use this to provide an upper bound on the right hand side of eq. 63.

$$a^2b^2 + b^4 + b^2c^2 < a^2(ac) + (ac)^2 + (ac)c^2 \tag{64}$$

$$< a^3c + a^2c^2 + ac^3 \tag{65}$$

Coincidentally, this is precisely the inequality in eq. 63, that we were proving.

Alternatively, if $|\rho| = 1 \implies b^2 = ac$, and $\rho_{\text{RMS}}^2 = \rho^2$. □

Definition 2. Extrapolation Parameters

We define extrapolation parameters as model parameters which have no effect on the data likelihood of a training dataset, but which nevertheless could influence model predictions made on a new data point.

Illustrative Example

Consider a fully-connected NN trained on the MNIST digit dataset. Further consider a preprocessing such that the pixel values by default are set to 0, and where they contain part of the digit take values $(0, 1]$.

Certain pixels may be zero across the entire training dataset, such as those in the corners of the image. First layer weights connected to such pixels will never receive input across the whole training dataset. Hence, the values of these parameter weights have no effect on the data likelihood. However, the weights would still influence predictions for some test image containing values for these pixels. Hence, these are named extrapolation parameters, since they influence extrapolation properties of the model.

The top row of figure 5 empirically shows examples of flat likelihoods for precisely these types of weights on MNIST.

Theorem 3. *For extrapolation parameters (definition 2) of a model, setting $\boldsymbol{\mu}_{anc} \coloneqq \boldsymbol{\mu}_{prior}$, $\boldsymbol{\Sigma}_{anc} \coloneqq \boldsymbol{\Sigma}_{prior}$, means the marginal RMS approximate posterior equals that of the marginal true posterior. This holds for any distributional form of parameter likelihood.*

Proof. Extrapolation parameters (definition 2) do not have any effect on the data likelihood, therefore their parameter likelihoods are flat. This means that their marginal posterior equals their marginal prior.

This results in a posterior covariance matrix structure as follows, where parameter i is an extrapolation parameter (here shown in the first row for convenience), so has marginal variance equal to the prior variance and is uncorrelated with all other parameters.

$$\boldsymbol{\Sigma}_{post} = \begin{bmatrix} \sigma^2_{prior,i} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & a_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{D2} & \dots & a_{DD} \end{bmatrix}$$

From corollary 1.1, $\mathbb{V}ar[\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})] = \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}^{-1}_{prior}\boldsymbol{\Sigma}_{post}$.

$$\begin{aligned} \mathbb{V}ar[\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})] &= \begin{bmatrix} \sigma^2_{prior,i} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & a_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{D2} & \dots & a_{DD} \end{bmatrix} \begin{bmatrix} 1/\sigma^2_{prior,i} & 0 & \dots & 0 \\ 0 & b_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_{DD} \end{bmatrix} \begin{bmatrix} \sigma^2_{prior,i} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & a_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{D2} & \dots & a_{DD} \end{bmatrix} \\ &= \begin{bmatrix} \sigma^2_{prior,i} & 0 & \dots & 0 \\ 0 & c_{22} & \dots & c_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & c_{D2} & \dots & c_{DD} \end{bmatrix} \end{aligned}$$

This shows that the marginal variance of the RMS approximate posterior equals that of the true posterior, $\mathbb{V}ar[\boldsymbol{f}_{\text{MAP}}(\boldsymbol{\theta}_{anc})]_{i,i} = [\boldsymbol{\Sigma}_{post}]_{i,i}$, for extrapolation parameters. Note that the values of the rest of the covariance matrices (a's, b's, c's) are irrelevant since these have no effect on the marginals of interest.

This proof did not assume any specific distributional form of parameter likelihood, only that it is flat for these extrapolation parameters.

□

Theorem 4. *Set $\boldsymbol{\mu}_{anc} \coloneqq \boldsymbol{\mu}_{prior}, \boldsymbol{\Sigma}_{anc} \coloneqq \boldsymbol{\Sigma}_{prior}$. The RMS approximate posterior will exactly equal the true posterior, $\boldsymbol{\Sigma}_{post}$, when all eigenvalues of a scaled version of $\boldsymbol{\Sigma}_{post}$ (scaled such that the prior equals the identity matrix) are equal to either 0 or 1. This corresponds to posteriors that are a mixture of perfectly correlated and perfectly uncorrelated parameters.*

Proof. We will initially consider a scaled version of the parameter space. This conveniently allows standard results for idempotent matrices to apply to the posterior covariance. A reverse scaling is subsequently applied to show that results hold for the original unscaled version. Finally, we articulate arguments allowing relaxation of the distributional assumptions.

Corollary 1.1 showed that the RMS approximate posterior is normally distributed and centered at the true posterior mean, but with modified variance, $\mathcal{N}(\boldsymbol{\mu}_{post}, \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{post})$. This proof requires specifying conditions that allow $\boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{post} = \boldsymbol{\Sigma}_{post}$ to hold.

One solution is given by $\boldsymbol{\Sigma}_{post} = \boldsymbol{\Sigma}_{prior}$, and is a trivial extension of theorem 3. Here we consider alternative solutions.

The two inputs into the inference process are the likelihood and prior covariances. Consider a scaling $\boldsymbol{\Sigma}'_{like} \coloneqq \boldsymbol{\Sigma}_{prior}^{-1/2}\boldsymbol{\Sigma}_{like}\boldsymbol{\Sigma}_{prior}^{-1/2}$ and $\boldsymbol{\Sigma}'_{prior} \coloneqq \boldsymbol{\Sigma}_{prior}^{-1/2}\boldsymbol{\Sigma}_{prior}\boldsymbol{\Sigma}_{prior}^{-1/2} = \mathbb{I}$. The posterior for this scaled version will be denoted by $\boldsymbol{\Sigma}'_{post}$, and is given as follows.

$$\begin{aligned}
\boldsymbol{\Sigma}'_{post} &= (\boldsymbol{\Sigma}'^{-1}_{like} + \boldsymbol{\Sigma}'^{-1}_{prior})^{-1} && (66)\\
&= (\boldsymbol{\Sigma}_{prior}^{1/2}\boldsymbol{\Sigma}_{like}^{-1}\boldsymbol{\Sigma}_{prior}^{1/2} + \boldsymbol{\Sigma}_{prior}^{1/2}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{prior}^{1/2})^{-1} && (67)\\
&= \boldsymbol{\Sigma}_{prior}^{-1/2}(\boldsymbol{\Sigma}_{like}^{-1} + \boldsymbol{\Sigma}_{prior}^{-1})^{-1}\boldsymbol{\Sigma}_{prior}^{-1/2} && (68)\\
&= \boldsymbol{\Sigma}_{prior}^{-1/2}\boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1/2} && (69)
\end{aligned}$$

Hence, unsurprisingly the same scaling applies to the posterior covariance, $\boldsymbol{\Sigma}'_{post} = \boldsymbol{\Sigma}_{prior}^{-1/2}\boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1/2}$.

We now consider conditions under which $\boldsymbol{\Sigma}'_{post}\boldsymbol{\Sigma}'^{-1}_{prior}\boldsymbol{\Sigma}'_{post} = \boldsymbol{\Sigma}'_{post}$ holds. From our choice of rescaling, we have that $\boldsymbol{\Sigma}'^{-1}_{prior} = \mathbb{I}$. So we require that $\boldsymbol{\Sigma}'^{2}_{post} = \boldsymbol{\Sigma}'_{post}$.

This conveniently allows use of results for idempotent matrices - defined as a square matrix, A, for which $A^2 = A$. Aside from the case when $A = \mathbb{I}$ (which corresponds to $\boldsymbol{\Sigma}_{post} = \boldsymbol{\Sigma}_{prior}$), a matrix is idempotent if and only if it is singular and all eigenvalues are 0 or 1.

In order that, $\boldsymbol{\Sigma}'_{post}\boldsymbol{\Sigma}'^{-1}_{prior}\boldsymbol{\Sigma}'_{post} = \boldsymbol{\Sigma}'_{post}$, it is therefore sufficient that our scaled posterior, $\boldsymbol{\Sigma}'_{post}$, is singular with all eigenvalues 0 or 1. Any possible permutation is allowed.

Naturally, applying a reverse scaling recovers the original parameter space, $\boldsymbol{\Sigma}_{post} = \boldsymbol{\Sigma}_{prior}^{1/2}\boldsymbol{\Sigma}'_{post}\boldsymbol{\Sigma}_{prior}^{1/2}$.

Remark. *To summarise, we have shown that provided the RMS approximate posterior equals the true posterior in the scaled space, it will also be equal in the original unscaled space. In order for this equality to hold, eigenvalues must be 0 or 1 in the scaled space.*

See section B.1.2 for numerical examples in a three parameter model when this condition holds.

□

A.1 MAP solution and regularisation interpretation

For completeness, we write out the MAP solution for the case of normally distributed prior, and data likelihoods often used in regression and classification. From this derives our interpretation of the regularisation matrix, $\mathbf{\Gamma}$.

$$\begin{aligned}\boldsymbol{\theta}_{MAP} &= \text{argmax}_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathcal{D}) \\ &= \text{argmax}_{\boldsymbol{\theta}} P_{\mathcal{D}}(\mathcal{D}|\boldsymbol{\theta}) P(\boldsymbol{\theta}) \\ &= \text{argmax}_{\boldsymbol{\theta}} \log(P_{\mathcal{D}}(\mathcal{D}|\boldsymbol{\theta})) + \log(P(\boldsymbol{\theta}))\end{aligned}$$

If prior is normally distributed, $P(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$,

$$\begin{aligned}&= \text{argmax}_{\boldsymbol{\theta}} \log(P_{\mathcal{D}}(\mathcal{D}|\boldsymbol{\theta})) - \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}) + \text{const.} \\ &= \text{argmax}_{\boldsymbol{\theta}} \log(P_{\mathcal{D}}(\mathcal{D}|\boldsymbol{\theta})) - \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}).\end{aligned}$$

Typically in BNNs the prior covariance is chosen as diagonal. This is sometimes set as isotropic, $\mathbf{\Sigma} = \lambda \mathbb{I}$, but here we will keep it in matrix form (but assuming it is diagonal) so that different prior variances can be assigned to different layer weights.

$$= \text{argmax}_{\boldsymbol{\theta}} \log(P_{\mathcal{D}}(\mathcal{D}|\boldsymbol{\theta})) - \frac{1}{2}\|\mathbf{\Sigma}^{-1/2} \cdot (\boldsymbol{\theta} - \boldsymbol{\mu})\|_2^2$$

In the case that the prior mean is zero $\boldsymbol{\mu} = \mathbf{0}$,

$$= \text{argmax}_{\boldsymbol{\theta}} \log(P_{\mathcal{D}}(\mathcal{D}|\boldsymbol{\theta})) - \frac{1}{2}\|\mathbf{\Sigma}^{-1/2} \cdot \boldsymbol{\theta}\|_2^2.$$

One is free to choose any suitable expression for $\log(P_{\mathcal{D}}(\mathcal{D}|\boldsymbol{\theta}))$. Next we describe the resulting forms for common choices of log likelihood in regression and classification tasks.

Regression

For regression, a common choice is that the NN predicts the mean of the function, $\hat{\mathbf{y}}$, and there is additive noise on the true targets $\mathbf{y}$, $P_{\mathcal{D}}(\mathcal{D}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}|\hat{\mathbf{y}}, \sigma_\epsilon^2)$

$$\begin{aligned}\boldsymbol{\theta}_{MAP} &= \text{argmax}_{\boldsymbol{\theta}} - \frac{1}{2\sigma_\epsilon^2}\|\hat{\mathbf{y}} - \mathbf{y}\|_2^2 + \text{const.} - \frac{1}{2}\|\mathbf{\Sigma}^{-1/2} \cdot \boldsymbol{\theta}\|_2^2 \\ &= \text{argmax}_{\boldsymbol{\theta}} - \frac{1}{2\sigma_\epsilon^2}\|\hat{\mathbf{y}} - \mathbf{y}\|_2^2 - \frac{1}{2}\|\mathbf{\Sigma}^{-1/2} \cdot \boldsymbol{\theta}\|_2^2\end{aligned}$$

Generally the mean squared error is minimised,

$$= \text{argmin}_{\boldsymbol{\theta}} \frac{1}{N}\|\hat{\mathbf{y}} - \mathbf{y}\|_2^2 + \frac{1}{N}\|\sigma_\epsilon \mathbf{\Sigma}^{-1/2} \cdot \boldsymbol{\theta}\|_2^2$$

More compactly, we can define $\mathbf{\Gamma} \coloneqq \sigma_\epsilon^2 \mathbf{\Sigma}^{-1}$, as a diagonal matrix with, $\text{diag}(\mathbf{\Gamma})_i = \sigma_\epsilon^2 / \sigma_{prior,i}^2$,

$$= \text{argmin}_{\boldsymbol{\theta}} \frac{1}{N}\|\hat{\mathbf{y}} - \mathbf{y}\|_2^2 + \frac{1}{N}\|\mathbf{\Gamma}^{1/2} \cdot \boldsymbol{\theta}\|_2^2$$

Classification

The data likelihood is commonly chosen as a multinomial distribution, $P_{\mathcal{D}}(\mathcal{D}|\boldsymbol{\theta}) \propto \prod_{n=1}^{N} \prod_{c=1}^{C} \hat{y}_{n,c}^{y_{n,c}}$, for C classes, and N data points, where $\hat{y} \in [0, 1]$ denotes predicted probability, and $y_{n,c} \in \{0, 1\}$ the true targets.

$$\boldsymbol{\theta}_{MAP} = \text{argmax}_{\boldsymbol{\theta}} \sum_{n=1}^{N} \sum_{c=1}^{C} y_{n,c} \log(\hat{y}_{n,c}) - \frac{1}{2}\|\mathbf{\Sigma}^{-1/2} \cdot \boldsymbol{\theta}\|_2^2$$

Cross entropy is typically minimised,

$$= \text{argmin}_{\boldsymbol{\theta}} - \sum_{n=1}^{N} \sum_{c=1}^{C} y_{n,c} \log(\hat{y}_{n,c}) + \frac{1}{2}\|\mathbf{\Sigma}^{-1/2} \cdot \boldsymbol{\theta}\|_2^2$$

Here we can simply define $\mathbf{\Gamma} \coloneqq \frac{1}{2}\mathbf{\Sigma}^{-1}$, with $\text{diag}(\mathbf{\Gamma})_i = 1/2\sigma_{prior,i}^2$.

B Numerical Examples

B.1 Numerical Examples of Proofs

In this section we print covariance matrices that illustrate theoretical results numerically.

B.1.1 General case: Example of lemma 1.1, 1.2, theorem 2

For general $\mathbf{\Sigma}_{post}$, RMS will return a posterior with underestimated marginal variancesa and overestimated correlations. Since the prior is isotropic, the orientation (eigenvectors) of the RMS approximate posterior will be unchanged. Here a three parameter is shown. Note $\mathbf{\Sigma}_{post}\mathbf{\Sigma}_{prior}^{-1}\mathbf{\Sigma}_{post}$ represents the RMS approximate posterior.

$$\mathbf{\Sigma}_{prior} = \begin{bmatrix} 2.0 & 0.0 & 0.0 \\ 0.0 & 2.0 & 0.0 \\ 0.0 & 0.0 & 2.0 \end{bmatrix} \qquad \mathbf{\Sigma}_{like} = \begin{bmatrix} 2.0 & 0.707 & 0.283 \\ 0.707 & 1.0 & 0.4 \\ 0.283 & 0.4 & 1.0 \end{bmatrix}$$

$$\mathbf{\Sigma}_{post} = \begin{bmatrix} 0.953 & 0.238 & 0.067 \\ 0.238 & 0.589 & 0.166 \\ 0.067 & 0.166 & 0.638 \end{bmatrix} \qquad \mathbf{\Sigma}_{post}\mathbf{\Sigma}_{prior}^{-1}\mathbf{\Sigma}_{post} = \begin{bmatrix} 0.485 & 0.189 & 0.073 \\ 0.189 & 0.215 & 0.11 \\ 0.073 & 0.11 & 0.22 \end{bmatrix}$$

Note, $\mathbf{\Sigma}_{post_{i,i}} > \mathbf{\Sigma}_{post}\mathbf{\Sigma}_{prior}^{-1}\mathbf{\Sigma}_{post_{i,i}}, \forall i.$

$$\text{Correlation}(\mathbf{\Sigma}_{post}) = \begin{bmatrix} 1.0 & 0.317 & 0.086 \\ 0.317 & 1.0 & 0.27 \\ 0.086 & 0.27 & 1.0 \end{bmatrix} \qquad \text{Correlation}(\mathbf{\Sigma}_{post}\mathbf{\Sigma}_{prior}^{-1}\mathbf{\Sigma}_{post}) = \begin{bmatrix} 1.0 & 0.585 & 0.224 \\ 0.585 & 1.0 & 0.504 \\ 0.224 & 0.504 & 1.0 \end{bmatrix}$$

Note, $\text{Correlation}(\mathbf{\Sigma}_{post})_{i,j} < \text{Correlation}(\mathbf{\Sigma}_{post}\mathbf{\Sigma}_{prior}^{-1}\mathbf{\Sigma}_{post})_{i,j}, \forall i \neq j.$

Eigenvalues and eigenvectors of $\mathbf{\Sigma}_{post}$,

$$\lambda = 1.1101, \mathbf{v} = \begin{bmatrix} -0.8352 & -0.471 & -0.284 \end{bmatrix}$$
$$\lambda = 0.6667, \mathbf{v} = \begin{bmatrix} -0.465 & 0.3288 & 0.822 \end{bmatrix}$$
$$\lambda = 0.4032, \mathbf{v} = \begin{bmatrix} 0.2938 & -0.8186 & 0.4936 \end{bmatrix}$$

Eigenvalues and eigenvectors of $\mathbf{\Sigma}_{post}\mathbf{\Sigma}_{prior}^{-1}\mathbf{\Sigma}_{post}$,

$$\lambda = 0.6162, \mathbf{v} = \begin{bmatrix} -0.8352 & -0.471 & -0.284 \end{bmatrix}$$
$$\lambda = 0.2222, \mathbf{v} = \begin{bmatrix} -0.465 & 0.3288 & 0.822 \end{bmatrix}$$
$$\lambda = 0.0813, \mathbf{v} = \begin{bmatrix} 0.2938 & -0.8186 & 0.4936 \end{bmatrix}$$

B.1.2 Special case: Examples of theorem 3, 4

We again print out covariance matrices for a three parameter model. Firstly we provide an example where two parameters are perfectly correlated, and one has no effect on the likelihood. We print unscaled and scaled versions. Note that all eigenvalues of the scaled posterior, $\mathbf{\Sigma}'_{post}$, are either 0 or 1.

$$\mathbf{\Sigma}_{post} = \begin{bmatrix} 1.0 & 1.0 & 0.0 \\ 1.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 2.0 \end{bmatrix} \quad \mathbf{\Sigma}_{prior} = \begin{bmatrix} 2.0 & 0.0 & 0.0 \\ 0.0 & 2.0 & 0.0 \\ 0.0 & 0.0 & 2.0 \end{bmatrix} \quad \mathbf{\Sigma}_{post}\mathbf{\Sigma}_{prior}^{-1}\mathbf{\Sigma}_{post} = \begin{bmatrix} 1.0 & 1.0 & 0.0 \\ 1.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 2.0 \end{bmatrix}$$

$$\mathbf{\Sigma}'_{post} = \begin{bmatrix} 0.5 & 0.5 & 0.0 \\ 0.5 & 0.5 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad \mathbf{\Sigma}'_{prior} = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad \mathbf{\Sigma}'_{post}\mathbf{\Sigma}'^{-1}_{prior}\mathbf{\Sigma}'_{post} = \begin{bmatrix} 0.5 & 0.5 & 0.0 \\ 0.5 & 0.5 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

Eigenvalues and eigenvectors of $\mathbf{\Sigma}_{post} = \mathbf{\Sigma}_{post}\mathbf{\Sigma}_{prior}^{-1}\mathbf{\Sigma}_{post}$,

$$\lambda = 2.0, \mathbf{v} = \begin{bmatrix}0.7071 & 0.7071 & 0.\end{bmatrix}$$
$$\lambda = 0.0, \mathbf{v} = \begin{bmatrix}-0.7071 & 0.7071 & 0.\end{bmatrix}$$
$$\lambda = 2.0, \mathbf{v} = \begin{bmatrix}0. & 0. & 1.\end{bmatrix}$$

Eigenvalues and eigenvectors of $\mathbf{\Sigma}'_{post} = \mathbf{\Sigma}'_{post}\mathbf{\Sigma}'^{-1}_{prior}\mathbf{\Sigma}'_{post}$,

$$\lambda = 1.0, \mathbf{v} = \begin{bmatrix}0.707 & 0.707 & 0.\end{bmatrix}$$
$$\lambda = 0.0, \mathbf{v} = \begin{bmatrix}-0.707 & 0.707 & 0.\end{bmatrix}$$
$$\lambda = 1.0, \mathbf{v} = \begin{bmatrix}0. & 0. & 1.\end{bmatrix}$$

Following is the same set up as the previous case, but now all parameters are perfectly correlated. Eigenvalues of the scaled posterior, $\mathbf{\Sigma}'_{post}$, are again either 0 or 1.

$$\mathbf{\Sigma}_{post} = \begin{bmatrix}0.667 & 0.667 & 0.667\\ 0.667 & 0.667 & 0.667\\ 0.667 & 0.667 & 0.667\end{bmatrix} \quad \mathbf{\Sigma}_{prior} = \begin{bmatrix}2.0 & 0.0 & 0.0\\ 0.0 & 2.0 & 0.0\\ 0.0 & 0.0 & 2.0\end{bmatrix} \quad \mathbf{\Sigma}_{post}\mathbf{\Sigma}_{prior}^{-1}\mathbf{\Sigma}_{post} = \begin{bmatrix}0.667 & 0.667 & 0.667\\ 0.667 & 0.667 & 0.667\\ 0.667 & 0.667 & 0.667\end{bmatrix}$$

$$\mathbf{\Sigma}'_{post} = \begin{bmatrix}0.333 & 0.333 & 0.333\\ 0.333 & 0.333 & 0.333\\ 0.333 & 0.333 & 0.333\end{bmatrix} \quad \mathbf{\Sigma}'_{prior} = \begin{bmatrix}1.0 & 0.0 & 0.0\\ 0.0 & 1.0 & 0.0\\ 0.0 & 0.0 & 1.0\end{bmatrix} \quad \mathbf{\Sigma}'_{post}\mathbf{\Sigma}'^{-1}_{prior}\mathbf{\Sigma}'_{post} = \begin{bmatrix}0.333 & 0.333 & 0.333\\ 0.333 & 0.333 & 0.333\\ 0.333 & 0.333 & 0.333\end{bmatrix}$$

Eigenvalues and eigenvectors of $\mathbf{\Sigma}_{post} = \mathbf{\Sigma}_{post}\mathbf{\Sigma}_{prior}^{-1}\mathbf{\Sigma}_{post}$,

$$\lambda = 2.0, \mathbf{v} = \begin{bmatrix}-0.5774 & -0.5774 & -0.5774\end{bmatrix}$$
$$\lambda = 0.0, \mathbf{v} = \begin{bmatrix}-0. & -0.7071 & 0.7071\end{bmatrix}$$
$$\lambda = 0.0, \mathbf{v} = \begin{bmatrix}-0.6667 & -0.0749 & 0.7416\end{bmatrix}$$

Eigenvalues and eigenvectors of $\mathbf{\Sigma}'_{post} = \mathbf{\Sigma}'_{post}\mathbf{\Sigma}'^{-1}_{prior}\mathbf{\Sigma}'_{post}$,

$$\lambda = 1.0, \mathbf{v} = \begin{bmatrix}0.577 & 0.577 & 0.577\end{bmatrix}$$
$$\lambda = 0.0, \mathbf{v} = \begin{bmatrix}0. & -0.707 & 0.707\end{bmatrix}$$
$$\lambda = 0.0, \mathbf{v} = \begin{bmatrix}-0.521 & -0.284 & 0.805\end{bmatrix}$$

Now we detail an example of a non-isometric prior.

$$\mathbf{\Sigma}_{post} = \begin{bmatrix}1.818 & 0.0 & 1.818\\ 0.0 & 2.0 & 0.0\\ 1.818 & 0.0 & 1.818\end{bmatrix} \quad \mathbf{\Sigma}_{prior} = \begin{bmatrix}20.0 & 0.0 & 0.0\\ 0.0 & 2.0 & 0.0\\ 0.0 & 0.0 & 2.0\end{bmatrix} \quad \mathbf{\Sigma}_{post}\mathbf{\Sigma}_{prior}^{-1}\mathbf{\Sigma}_{post} = \begin{bmatrix}1.818 & 0.0 & 1.818\\ 0.0 & 2.0 & 0.0\\ 1.818 & 0.0 & 1.818\end{bmatrix}$$

$$\mathbf{\Sigma}'_{post} = \begin{bmatrix}0.091 & 0.0 & 0.287\\ 0.0 & 1.0 & 0.0\\ 0.287 & 0.0 & 0.909\end{bmatrix} \quad \mathbf{\Sigma}'_{prior} = \begin{bmatrix}1.0 & 0.0 & 0.0\\ 0.0 & 1.0 & 0.0\\ 0.0 & 0.0 & 1.0\end{bmatrix} \quad \mathbf{\Sigma}'_{post}\mathbf{\Sigma}'^{-1}_{prior}\mathbf{\Sigma}'_{post} = \begin{bmatrix}0.091 & 0.0 & 0.287\\ 0.0 & 1.0 & 0.0\\ 0.287 & 0.0 & 0.909\end{bmatrix}$$

Eigenvalues and eigenvectors of $\boldsymbol{\Sigma}_{post} = \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{post}$,

$$\lambda = 3.636, \mathbf{v} = \begin{bmatrix}0.707 & 0. & 0.707\end{bmatrix}$$
$$\lambda = 0.0, \mathbf{v} = \begin{bmatrix}-0.707 & 0. & 0.707\end{bmatrix}$$
$$\lambda = 2.0, \mathbf{v} = \begin{bmatrix}0. & 1. & 0.\end{bmatrix}$$

Eigenvalues and eigenvectors of $\boldsymbol{\Sigma}'_{post} = \boldsymbol{\Sigma}'_{post}\boldsymbol{\Sigma}'^{-1}_{prior}\boldsymbol{\Sigma}'_{post}$,

$$\lambda = 0.0, \mathbf{v} = \begin{bmatrix}-0.953 & 0. & 0.302\end{bmatrix}$$
$$\lambda = 1.0, \mathbf{v} = \begin{bmatrix}-0.302 & 0. & -0.953\end{bmatrix}$$
$$\lambda = 1.0, \mathbf{v} = \begin{bmatrix}0. & 1. & 0.\end{bmatrix}$$

B.2 Mixtures of Parameter Types

Here, we provide examples of a five parameter model containing a mixture of perfectly correlated, partially correlated, and extrapolation parameters.

First we consider distinct blocks of perfectly and partially correlated parameters, as well as one extrapolation parameter. In this situation both the perfectly correlated block and the extrapolation parameter posterior are recovered exactly. The RMS approximate posterior of the partially correlated block is biased as per the general case.

$$\boldsymbol{\Sigma}_{post} = \begin{bmatrix}1.0 & 1.0 & 0.0 & 0.0 & 0.0\\ 1.0 & 1.0 & 0.0 & 0.0 & 0.0\\ 0.0 & 0.0 & 0.5 & 0.2 & 0.0\\ 0.0 & 0.0 & 0.2 & 0.8 & 0.0\\ 0.0 & 0.0 & 0.0 & 0.0 & 2.0\end{bmatrix} \quad \boldsymbol{\Sigma}_{prior} = \begin{bmatrix}2 & 0 & 0 & 0 & 0\\ 0 & 2 & 0 & 0 & 0\\ 0 & 0 & 2 & 0 & 0\\ 0 & 0 & 0 & 2 & 0\\ 0 & 0 & 0 & 0 & 2\end{bmatrix} \quad \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{post} = \begin{bmatrix}1.0 & 1.0 & 0.0 & 0.0 & 0.0\\ 1.0 & 1.0 & 0.0 & 0.0 & 0.0\\ 0.0 & 0.0 & 0.145 & 0.13 & 0.0\\ 0.0 & 0.0 & 0.13 & 0.34 & 0.0\\ 0.0 & 0.0 & 0.0 & 0.0 & 2.0\end{bmatrix}$$

Secondly the perfectly correlated block overlaps with the partially correlated block. In this scenario, a small amount of bias is introduced on the perfectly correlated block, but not in terms of the correlation. The extrapolation parameter is unaffected.

$$\boldsymbol{\Sigma}_{post} = \begin{bmatrix}1.0 & 1.0 & 0.1 & 0.2 & 0.0\\ 1.0 & 1.0 & 0.1 & 0.2 & 0.0\\ 0.1 & 0.1 & 0.5 & 0.2 & 0.0\\ 0.2 & 0.2 & 0.2 & 0.8 & 0.0\\ 0.0 & 0.0 & 0.0 & 0.0 & 2.0\end{bmatrix} \quad \boldsymbol{\Sigma}_{prior} = \begin{bmatrix}2 & 0 & 0 & 0 & 0\\ 0 & 2 & 0 & 0 & 0\\ 0 & 0 & 2 & 0 & 0\\ 0 & 0 & 0 & 2 & 0\\ 0 & 0 & 0 & 0 & 2\end{bmatrix} \quad \boldsymbol{\Sigma}_{post}\boldsymbol{\Sigma}_{prior}^{-1}\boldsymbol{\Sigma}_{post} = \begin{bmatrix}1.03 & 1.03 & 0.15 & 0.29 & 0.0\\ 1.03 & 1.03 & 0.15 & 0.29 & 0.0\\ 0.15 & 0.15 & 0.16 & 0.15 & 0.0\\ 0.29 & 0.29 & 0.15 & 0.38 & 0.0\\ 0.0 & 0.0 & 0.0 & 0.0 & 2.0\end{bmatrix}$$

B.3 Example of Perfectly Correlated Parameters in a Neural Network

Section 4.2.2 stated that perfectly correlated parameters can exist in a NN. Here, we provide a concrete example of such a case, and show that anchored ensembling *does* recover the true posterior for these parameters when $\mathbf{\Sigma}_{anc} = \mathbf{\Sigma}_{prior}$.

We consider finding the posterior of final layer weights in a small single layer ReLU NN of two hidden nodes for a regression problem with two data points. (Choosing the final layer weights for our analysis allows analytical equations associated with linear regression to be used, simplifying our analysis, though these results would also apply to the first layer weights and biases.)

We design the problem such that the point where both hidden nodes becomes greater than zero (the elbow points of ReLU units) falls in between the two data points, and the active half of the output is also shared, so that the final layer weights are perfectly correlated.

Figure 9 illustrates our set up. Data points and NN parameters are as follows,

$\mathcal{D} = \{x_1 = -5, y_1 = 0; x_2 = 5, y_2 = 0\}$, $\sigma_\epsilon^2 = 0.01$,

$\mathbf{W}_1 = [-0.8, -0.4], \mathbf{b}_1 = [-1, 0.1], \hat{y} = \mathbf{W}_2^T \max(x\mathbf{W}_1 + \mathbf{b}_1, 0)$

We set prior means to zero, with isotropic covariance according to $1/H$,

$$\mathbf{\Sigma}_{prior} = \begin{bmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{bmatrix}$$

We print out matrices of interest,

$$\mathbf{\Sigma}_{like} = \begin{bmatrix} -6.89e+13 & 9.85e+13 \\ 9.85e+13 & -1.40e+14 \end{bmatrix}$$

$$\mathbf{\Sigma}_{like}^{-1} = \begin{bmatrix} 90.0 & 63.0 \\ 63.0 & 44.1 \end{bmatrix}$$

$$\mathbf{\Sigma}_{post} = \begin{bmatrix} 0.169 & -0.231 \\ -0.231 & 0.338 \end{bmatrix}$$

$$\mathbf{\Sigma}_{post}\mathbf{\Sigma}_{prior}^{-1}\mathbf{\Sigma}_{post} = \begin{bmatrix} 0.166 & -0.235 \\ -0.235 & 0.338 \end{bmatrix}$$

The anchored ensembling posterior, $\mathbf{\Sigma}_{post}\mathbf{\Sigma}_{prior}^{-1}\mathbf{\Sigma}_{post}$, provides a close approximation of the true posterior covariance (and would be exact discounting numerical rounding issues). Note the similarity of the posterior in figure 9 (middle) with the perfect correlations example shown in figure 3 (B).

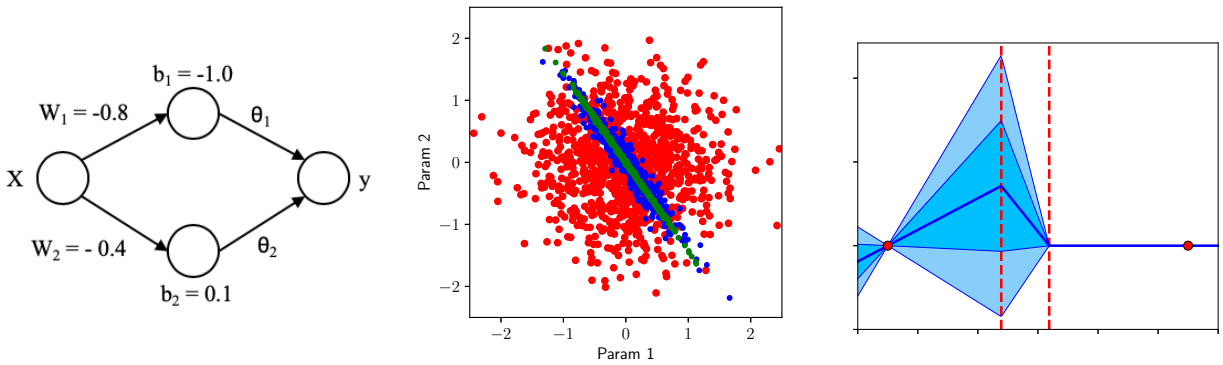

Figure 9: Left: Single layer NN of two hidden nodes. Middle: Draws in parameter space for prior (red), analytical posterior (blue) and anchored posterior (green). Right: Posterior predictive distribution - dashed red lines are elbows of ReLU units.

C Further Results

C.1 Uncertainty-Aware Model-Free Reinforcement Learning

An anchored ensemble of 5xNNs, each with two hidden layers, was trained to complete a discretised version of FetchPush - an agent controls a robotic arm, with rewards received when a randomly placed cube is pushed to a goal.

We used Bayesian Q-learning (Dearden et al., 1998), similar to regular Q-learning, but with Q-values modelled as *distributions* rather than point estimates - the wider the distribution, the less certain the agent. This is beneficial both to drive the exploration/exploitation process via Thompson sampling, and for identifying OOD examples.

Figure 10 shows the agent's awareness of its uncertainty. After training for 40,000 episodes, its confidence over actions was plotted for three scenarios: A) Cube and goal are in positions often encountered during training, the agent has learnt that it must move the arm left - the narrow distributions with significantly different means reflect its confidence in this. B) The goal has already been achieved - narrow overlapping distributions with higher means. C) A peculiar goal position that has never been encountered - the broad distributions over all actions reflect its high uncertainty.

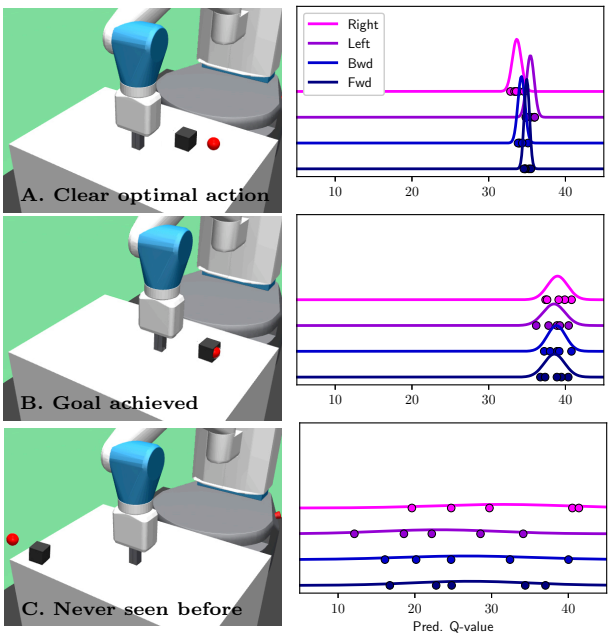

Figure 10: Anchored ensembling creates uncertainty-aware agents.

C.2 Model-Based Reinforcement: Learning in Noisy Environments

We tested the benefit of using an anchored ensemble in noisy RL environments. We modified the classic cartpole swingup environment such that different levels of stochastic noise could be added to the future state; consider a state action pair given by, s, a, and a noisy state, $\bar{s}$ such that $P(\bar{s}_{t+1}|a_t, s_t) = \mathcal{N}(s_{t+1}, \sigma_\epsilon^2)$.

The value of σ_ϵ^2 was given three settings: low, medium and high, $\sigma_\epsilon^2 \in \{0.001, 0.002, 0.005\}$. The task was learnt using a model-based RL approach similar to the heteroskedastic ensembles used by Chua et al. (2018). Fully-connected three-layer NNs learnt to predict the dynamics of the environment given some state and action. Planning was performed using the cross-entropy method, rolling out for a horizon of 25 steps, with 10 particles.

Figure 11 (A, B, C) shows learning curves for the three noise levels. All ensemble techniques perform similarly in the low noise setting. As noise is increased the overall performance of all methods drops. Anchored ensembles are most resiliant, followed by the unconstrained ensemble. In panel D, we compared against an unconstrained ensemble employing early stopping - this corresponds to the proposal that applying early stopping to an ensemble can produce approximate inference (Duvenaud et al., 2016). Whilst careful tuning did offer some improvement over the default setting, a performance gap to anchored ensembling remained.

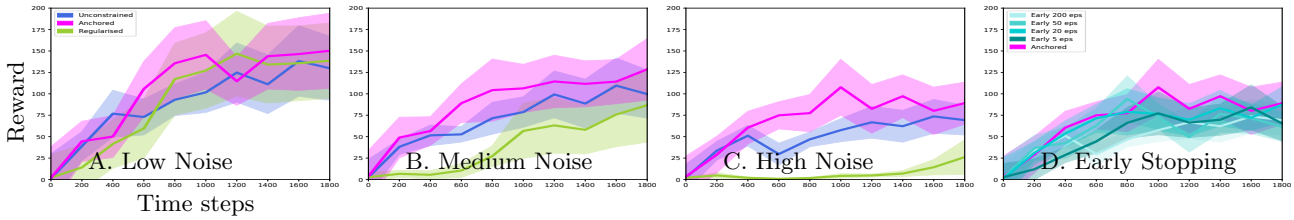

Figure 11: Anchored ensembling creates robust MBRL agents in noisy environments. Mean and standard error over five runs.

C.3 Regression Benchmarking

Tables 3 & 4 show all experiments run on the regression benchmarking datasets. The below discussion focuses on NLL results in table 4.

ERF GP refers to the equivalent GP for an infinite width, single-layer BNN with ERF activations. It was tuned and implemented as for the ReLU GP. We were interested to discover how different activation functions would affect uncertainty estimates. In general the ReLU GP performed better than the ERF GP, with some exceptions, such as for Wine. The target variable for Wine is ordinal, containing five factors, it is therefore understandable that the ReLU GP, which extrapolates linearly, is at a slight disadvantage.

10x 50 NNs refers to an anchored ensemble of ten NNs with 50 hidden nodes. We find that these results fall in between the 5x 50 NNs and the ReLU GP. This agrees with the convergence analysis done in section 5.2.

We also implemented an anchored ensemble of five two-layer NNs, 5x 50-50 NNs. Even with minimal hyperparameter tuning (section E) we found an extra layer gave a performance boost over the 5x 50 NNs. We expect with more careful tuning this margin would increase.

Single 50 NN refers to a single regularised NN, of one hidden layer with 50 hidden nodes, for which we used a constant value of predictive variance. Although this performs poorly in several cases, e.g. Boston and Yacht, the results are surprisingly close to those achieved by both our method and Deep Ensembles, even surpassing them on the Energy dataset. A method outputting constant predictive variance should not perform well in experiments designed to test uncertainty quantification, and this raises questions over the validity of the benchmarks.

Table 5 compares anchored ensembles against results reported for other methods.

Table 3: Variants of our method on benchmark regression datasets, RMSE.

			RMSE					
	N	D	ReLU GP	ERF GP	5x 50 NNs	10x 50 NNs	5x 50-50 NNs	Single 50 NN
Boston	506	13	2.86 ± 0.16	2.94 ± 0.18	3.09 ± 0.17	3.09 ± 0.17	3.00 ± 0.18	3.40 ± 0.20
Concrete	1,030	8	4.88 ± 0.13	5.21 ± 0.12	4.87 ± 0.11	4.73 ± 0.11	4.75 ± 0.12	5.17 ± 0.13
Energy	768	8	0.60 ± 0.02	0.78 ± 0.03	0.35 ± 0.01	0.34 ± 0.01	0.40 ± 0.01	0.40 ± 0.01
Kin8nm	8,192	8	0.07 ± 0.00	0.08 ± 0.00	0.07 ± 0.00	0.07 ± 0.00	0.06 ± 0.00	0.07 ± 0.00
Naval	11,934	16	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Power	9,568	4	3.97 ± 0.04	3.94 ± 0.04	4.07 ± 0.04	4.07 ± 0.04	4.03 ± 0.04	4.23 ± 0.04
Protein	45,730	9	4.34 ± 0.02	4.23 ± 0.02	4.36 ± 0.02	4.34 ± 0.02	4.23 ± 0.02	4.56 ± 0.02
Wine	1,599	11	0.61 ± 0.01	0.60 ± 0.01	0.63 ± 0.01	0.62 ± 0.01	0.62 ± 0.01	0.64 ± 0.01
Yacht	308	6	0.60 ± 0.08	1.48 ± 0.15	0.57 ± 0.05	0.54 ± 0.05	0.85 ± 0.08	0.81 ± 0.07
Song Year	515,345	90	9.01 ± NA	8.90 ± NA	8.82 ± NA	8.82 ± NA	8.66 ± NA	8.77 ± NA

Table 4: Variants of our method on benchmark regression datasets, NLL.

			NLL				
	$\hat{\sigma}_\epsilon^2$	ReLU GP	ERF GP	5x 50 NNs	10x 50 NNs	5x 50-50 NNs	Single 50 NN
Boston	0.08	2.45 ± 0.05	2.46 ± 0.05	2.52 ± 0.05	2.50 ± 0.05	2.50 ± 0.07	2.70 ± 0.05
Concrete	0.05	2.96 ± 0.02	3.06 ± 0.02	2.97 ± 0.02	2.94 ± 0.02	2.94 ± 0.02	3.08 ± 0.03
Energy	1e-7	0.86 ± 0.02	1.06 ± 0.03	0.96 ± 0.13	0.52 ± 0.06	0.61 ± 0.07	0.57 ± 0.03
Kin8nm	0.02	-1.22 ± 0.01	-1.17 ± 0.00	-1.09 ± 0.01	-1.16 ± 0.01	-1.25 ± 0.01	-1.17 ± 0.01
Naval	1e-7	-10.05 ± 0.02	-9.66 ± 0.04	-7.17 ± 0.03	-7.29 ± 0.02	-7.08 ± 0.13	-6.58 ± 0.04
Power	0.05	2.80 ± 0.01	2.79 ± 0.01	2.83 ± 0.01	2.83 ± 0.01	2.82 ± 0.01	2.86 ± 0.01
Protein	0.5	2.88 ± 0.00	2.86 ± 0.00	2.89 ± 0.01	2.88 ± 0.01	2.86 ± 0.01	2.94 ± 0.00
Wine	0.5	0.92 ± 0.01	0.91 ± 0.01	0.95 ± 0.01	0.94 ± 0.01	0.94 ± 0.01	0.97 ± 0.01
Yacht	1e-7	0.49 ± 0.07	1.50 ± 0.13	0.37 ± 0.08	0.18 ± 0.03	0.04 ± 0.08	1.50 ± 0.02
Song Year	0.7	3.62 ± NA	3.61 ± NA	3.60 ± NA	3.60 ± NA	3.57 ± NA	3.59 ± NA

Table 5: Comparison against inference methods on UCI benchmark regression datasets, log likelihood. Adapted from Mukhoti et al. (2018).

	Log Likelihood (*not* negative)							
	Anch. Ens.	Drop conv.	Drop tune	VMG	HS-BNN	PBP-MV	SGHMC tune	SGHMC adap.
Boston	-2.52 ± 0.05	$\mathbf{-2.40 \pm 0.04}$	$\mathbf{-2.40 \pm 0.04}$	-2.46 ± 0.09	-2.54 ± 0.15	-2.54 ± 0.08	-2.49 ± 0.15	-2.54 ± 0.04
Concrete	-2.97 ± 0.02	-2.97 ± 0.02	$\mathbf{-2.93 \pm 0.02}$	-3.01 ± 0.03	-3.09 ± 0.06	-3.04 ± 0.03	-4.17 ± 0.72	-3.38 ± 0.24
Energy	$\mathbf{-0.96 \pm 0.13}$	-1.72 ± 0.01	-1.21 ± 0.01	-1.06 ± 0.03	-2.66 ± 0.13	-1.01 ± 0.01	$--$	$--$
Kin8nm	1.09 ± 0.01	0.97 ± 0.00	1.14 ± 0.01	1.10 ± 0.01	1.12 ± 0.03	$\mathbf{1.28 \pm 0.01}$	$--$	$--$
Naval	$\mathbf{7.17 \pm 0.03}$	3.91 ± 0.01	4.45 ± 0.00	2.46 ± 0.00	5.52 ± 0.10	4.85 ± 0.06	$--$	$--$
Power	-2.83 ± 0.01	-2.79 ± 0.01	-2.80 ± 0.01	-2.82 ± 0.01	-2.81 ± 0.03	$\mathbf{-2.78 \pm 0.01}$	$--$	$--$
Protein	-2.89 ± 0.01	-2.87 ± 0.00	-2.87 ± 0.00	-2.84 ± 0.00	-2.89 ± 0.00	$\mathbf{-2.77 \pm 0.01}$	$--$	$--$
Wine	-0.95 ± 0.01	$\mathbf{-0.92 \pm 0.01}$	-0.93 ± 0.01	-0.95 ± 0.01	-0.95 ± 0.05	-0.97 ± 0.01	-1.29 ± 0.28	-1.04 ± 0.17
Yacht	$\mathbf{-0.37 \pm 0.08}$	-1.38 ± 0.01	-1.25 ± 0.01	-1.30 ± 0.02	-2.33 ± 0.01	-1.64 ± 0.02	-1.75 ± 0.19	$\mathbf{-1.10 \pm 0.08}$

C.4 Fashion MNIST

Table 6 provides a breakdown of results from the OOD classification test in section 5.4 for fashion MNIST. Also included are results for entropy, where high entropy represents high uncertainty. These correlated strongly with the proportion metrics, which was true across all three OOD experiments.

Table 6: Fashion MNIST results: proportion of predictions made with $\geq 90\%$ probability, and entropy of predicted categorical distribution. Also shown is relative advantage (percentage change) for each method compared to anchored ensembles. Averaged over five runs/random seeds, mean $\pm$ 1 standard error. Best result in blue.

		—Edge Cases—		—Out-of-distribution—		—Natural Adversarial—			—Pure Adversarial—	
	Train	Sneaker	Trouser	CIFAR	MNIST	Rotate	Flip	Invert	Noise	Sparse
					Proportion $\geq 90\%$ (smaller better)					
reg 1xNN	0.660 ± 0.006	0.739 ± 0.056	0.429 ± 0.047	0.143 ± 0.008	0.160 ± 0.007	0.609 ± 0.007	0.330 ± 0.009	0.349 ± 0.015	0.271 ± 0.007	0.456 ± 0.006
free 5xNN	0.733 ± 0.001	0.781 ± 0.015	0.380 ± 0.030	0.301 ± 0.013	0.104 ± 0.010	0.571 ± 0.011	0.300 ± 0.011	0.222 ± 0.052	0.042 ± 0.005	0.048 ± 0.003
reg 5xNN	0.634 ± 0.002	0.589 ± 0.054	0.269 ± 0.020	0.115 ± 0.004	0.072 ± 0.007	0.556 ± 0.007	0.256 ± 0.012	0.213 ± 0.002	0.112 ± 0.005	0.174 ± 0.005
anc 5xNN	0.631 ± 0.002	0.578 ± 0.049	0.325 ± 0.037	0.065 ± 0.002	0.041 ± 0.002	0.497 ± 0.003	0.215 ± 0.005	0.025 ± 0.010	0.006 ± 0.001	0.006 ± 0.001
					Proportion Relative Advantage					
1xNN Reg. to 5xNN Anch.	-4.4%	-21.8%	-24.2%	-54.5%	-74.4%	-18.4%	-34.8%	-92.8%	-97.8%	-98.7%
5xNN Uncons. to 5xNN Anch.	-13.9%	-26.0%	-14.5%	-78.4%	-60.6%	-13.0%	-28.3%	-88.7%	-85.7%	-87.5%
5xNN Reg. to 5xNN Anch.	-0.5%	-1.9%	20.8%	-43.5%	-43.1%	-10.6%	-16.0%	-88.3%	-94.6%	-96.6%
					Entropy (larger better)					
1xNN Reg.	0.328 ± 0.005	0.253 ± 0.043	0.575 ± 0.050	1.176 ± 0.010	0.984 ± 0.015	0.484 ± 0.008	0.713 ± 0.009	0.836 ± 0.035	0.808 ± 0.010	0.580 ± 0.008
5xNN Uncons.	0.230 ± 0.001	0.161 ± 0.010	0.535 ± 0.021	0.688 ± 0.009	1.016 ± 0.021	0.453 ± 0.011	0.685 ± 0.011	0.573 ± 0.037	1.036 ± 0.014	0.992 ± 0.012
5xNN Reg.	0.352 ± 0.001	0.365 ± 0.039	0.707 ± 0.019	1.239 ± 0.009	1.161 ± 0.012	0.564 ± 0.008	0.807 ± 0.017	1.014 ± 0.014	1.048 ± 0.008	0.919 ± 0.009
5xNN Anch.	0.349 ± 0.001	0.327 ± 0.034	0.623 ± 0.042	1.251 ± 0.011	1.295 ± 0.013	0.624 ± 0.006	0.868 ± 0.002	1.098 ± 0.035	1.238 ± 0.013	1.191 ± 0.014
					Entropy Relative Advantage					
1xNN Reg. to 5xNN Anch.	6.4%	29.2%	8.3%	6.4%	31.6%	28.9%	21.7%	31.3%	53.2%	105.3%
5xNN Uncons. to 5xNN Anch.	51.7%	103.1%	16.4%	81.8%	27.5%	37.7%	26.7%	91.6%	19.5%	20.1%
5xNN Reg. to 5xNN Anch.	-0.9%	-10.4%	-11.9%	1.0%	11.5%	10.6%	7.6%	8.3%	18.1%	29.6%

D Additional Material

D.1 Algorithms

Algorithm 1 Implementing anchored ensembles of NNs

Input: Training data, $\mathbf{X}$ & $\mathbf{Y}$, test data point, $\mathbf{x}^*$, prior mean and covariance, $\boldsymbol{\mu}_{prior}$, $\boldsymbol{\Sigma}_{prior}$, ensemble size, M, data noise variance estimate, $\hat{\sigma}^2_\epsilon$ (regression only).
Output: Estimate of mean and variance, $\hat{\mathbf{y}}$, $\hat{\sigma}^2_y$ for regression, or class probabilities, $\hat{\mathbf{y}}$ for classification.

 # Set regularisation matrix
$\boldsymbol{\Gamma} \Leftarrow \hat{\sigma}^2_\epsilon \boldsymbol{\Sigma}^{-1}_{prior}$ (regression) OR $\boldsymbol{\Gamma} \Leftarrow \frac{1}{2}\boldsymbol{\Sigma}^{-1}_{prior}$ (classification)

 # Create ensemble
$\boldsymbol{\mu}_{anc} \Leftarrow \boldsymbol{\mu}_{prior}, \boldsymbol{\Sigma}_{anc} \Leftarrow \boldsymbol{\Sigma}_{prior}$
for $j = 1$ **to** M
  $\boldsymbol{\theta}_{anc,j} \sim \mathcal{N}(\boldsymbol{\mu}_{anc}, \boldsymbol{\Sigma}_{anc})$ *# Sample anchor points*
  NN_j.create($\boldsymbol{\Gamma}, \boldsymbol{\theta}_{anc,j}$) *# Create custom regulariser*
  NN_j.initialise() *# Initialisations independent of* $\boldsymbol{\theta}_{anc,j}$

 # Train ensemble
for $j = 1$ **to** M
  NN_j.train($\mathbf{X}, \mathbf{Y}$), loss in eq. 8 (regression) or eq. 9 (classification) or eq. 7 (custom)

 # Predict with ensemble
for $j = 1$ **to** M
  $\hat{\mathbf{y}}_j \Leftarrow NN_j$.predict($\mathbf{x}^*$)

 # Regression - combine ensemble estimates
$\hat{\mathbf{y}} = \frac{1}{M}\sum_{j=1}^{M} \hat{\mathbf{y}}_j$, *# Mean prediction*
$\hat{\sigma}^2_{model} = \frac{1}{M-1}\sum_{j=1}^{M}(\hat{y}_j - \hat{y})^2$ *# Epistemic var.*
$\hat{\sigma}^2_y = \hat{\sigma}^2_{model} + \hat{\sigma}^2_\epsilon$ *# Total var. = epistemic + data noise*

 # Classification - combine ensemble estimates
$\hat{\mathbf{y}} = \frac{1}{M}\sum_{j=1}^{M} \hat{\mathbf{y}}_j$, *# Average softmax output*
$\hat{\sigma}^2_y =$ None *# N/A for classification*

return $\hat{\mathbf{y}}$, $\hat{\sigma}^2_y$

E Experimental Details

E.1 Introduction to Anchored Ensembles

Experimental details for figure 1 are as follows.

Six randomly generated data points were used.

Hyperparameters: activation = ERF, $\sigma_\epsilon^2 = 0.003$, b_1 variance = 1, W_1 variance = 1, $H = 100$, $M = 3$ (number of ensembles), optimiser = adam, epochs = 400, learning rate = 0.005.

E.2 Panel of Inference Methods

Experimental details for figure 4 are as follows.

Same six data points were used for all methods and activation functions, generated by $y = x\sin(5x)$, evaluated at, [-0.8, -0.1, 0.02, 0.2, 0.6, 0.8].

Hyperparameters: b_1 variance = 10, W_1 variance = 10, $H = 100$, $M = 10$, epochs= 4,000, $\sigma_\epsilon^2 = 0.001$, leaky ReLU $\alpha = 0.2$, optimiser = adam, MC Dropout probability = 0.4, MC Dropout samples = 200, HMC step size = 0.001, HMC no. steps = 150, HMC burn in = 500, HMC total samples = 1000, HMC predict samples = 50, VI predict samples = 50, VI iterations = 2000, VI gradient samples = 200.

E.3 Ensembling Loss Functions

Experimental details for figure 6 are as follows.

E.3.1 Regression

Generated $\mathbf{X}$ by sampling 20 points linearly spaced from the interval [-1.5, 1.5], $y = sin(2x)+\epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.2^2)$. The y value corresponding to the largest x value was shifted -0.4 to produce a slight outlier.

Sub-plot A was trained via mean square error, B was regularised, C was anchored. D shows a ReLU GP.

Hyperparameters: activation = ReLU, $\sigma_\epsilon^2 = 0.08$, b_1 variance = 10, W_1 variance = 10, $H = 1000$, optimiser = adam, epochs = 2,000, learning rate = 0.003, $M = 10$, hidden layers = 1.

E.3.2 Classification

Generated $\mathbf{X}$ using sklearn's 'make blobs' function, n samples = 30.

Sub-plot A was trained via cross entropy, B was regularised, C was anchored. D shows inference with HMC.

Hyperparameters: activation = ReLU, b_1 variance = 15/2, W_1 variance = 15/2, b_2 variance = 1/50, W_2 variance = 1/50, W_3 variance = 10/50, $H = 50$, optimiser = adam, epochs = 100, learning rate = 0.001, $M = 10$, hidden layers = 2.

E.4 1-D Convergence Plots

Experimental details for figure 8 are as follows.

Data as in section E.2 was used, with M = [3,5,10,20].

Hyperparameters: activation = ReLU, $\sigma_\epsilon^2 = 0.001$, b_1 variance = 20, W_1 variance = 20, $H = 100$, optimiser = adam, epochs = 4,000, learning rate = 0.005.

E.5 KL Convergence Results

Experimental details for figure 7 are as follows.

Training was done on 50% of the data, with KL computed over the other 50%. Results were averaged over ten runs. The 'ideal' line shows the metric when posterior samples from the GP itself, rather than anchored NNs,

were used.

The Boston Housing dataset was used, with 50% of data used for training, and testing on the other 50%.

Hyperparameters: activation = ReLU, $\sigma_\epsilon^2 = 0.1$, b_1 variance = 2, W_1 variance = 2, H = [4, 16, 64, 256, 1024], M = [3,5,10,20,40], optimiser = adam, no. runs = 10, epochs = 1,000, learning rate = 0.001 when $H < 20$ else learning rate = 0.0002.

E.6 Regression Benchmarking Experiments

We complied with the established protocol (Hernández-Lobato and Adams, 2015). Single-layer NNs of 50 nodes were used, experiments repeated 20 times with random train/test splits of 90%/10%. The larger Protein and Song datasets allow 100 node NNs, and were repeated five and one time respectively.

The hyperparameter tuning process and final settings for experiments in table 1, 3 & 4 are as follows.

E.6.1 Hyperparameter Tuning

Hyperparameter tuning was done on a single train/validation split of 80%/20%. We found it convenient to begin by tuning data noise variance and prior variances. We restricted the prior variance search space by enforcing, $\sigma_{W_1}^2 = \sigma_{b_1}^2/D$, and $\sigma_{W_2}^2 = 1/H$. We therefore had only two hyperparameters to optimise initially: $\sigma_{b_1}^2$ and σ_ϵ^2. We did this with the GP model, using grid search, maximising marginal log likelihood over the training portion, and minimising NLL of the validation portion. For the larger datasets, when inference over the 80% training portion was too slow, we reduced the training split to 2,000 data points.

Hyperparameters for priors and data noise estimates were shared between the GP and anchored ensembles. Hyperparameters requiring tuning specifically for anchored ensembles were batch size, learning rate, number of epochs and decay rate. This was done on the same 80%/20% split used to select data noise and prior variance. We used random search, directed by our knowledge of the optimisation process (e.g. a lower learning rate requires more epochs to converge), minimising NLL on the validation portion.

We did not retune hyperparameters from scratch for the double layer NN (5x 50-50 NNs). We used settings as for the single-layer NNs (5x 50 NNs), but divided learning rate by 4, and multiplied epochs by 1.5.

For the single regularised NN with constant noise, we again used hyperparameters as for the single-layer ensemble (5x 50 NNs), tuning only the constant amount of variance to be added on the same 80%/20% split.

E.6.2 Hyperparameter Settings

Table 7 provides the key hyperparameters used. The adam optimiser was used for all experiments. ReLU activations were used for all except the ERF GP (prior variance was separately tuned for this, values aren't given in the table).

Table 7: Hyperparameters used for regression benchmark results.

	N	Batch Size	Learn Rate	$\hat{\sigma}_\epsilon^2$	b_1 variance	W_1 variance	No. Epochs	Decay Rate	Single NN var.
Boston	506	64	0.05	0.06	10	0.77	3000	0.995	0.45
Concrete	1,030	64	0.05	0.05	40	5.00	2000	0.997	0.28
Energy	768	64	0.05	1e-7	12	1.50	2000	0.997	0.03
Kin8nm	8,192	256	0.10	0.02	40	5.00	2000	0.998	0.32
Naval	11,934	256	0.10	1e-7	200	12.50	1000	0.997	0.03
Power	9,568	256	0.20	0.05	4	1.00	1000	0.995	0.24
Protein	45,730	8192	0.10	0.5	50	5.56	3000	0.995	0.71
Wine	1,599	64	0.05	0.5	20	1.82	500	0.997	0.77
Yacht	308	64	0.05	1e-7	15	2.50	3000	0.997	0.10
Song Year	515,345	32768	0.01	0.7	2	0.02	500	0.996	0.84

E.7 Out-of-Distribution Classification

E.7.1 Fashion MNIST

We trained a three-layer NN on eight of ten classes of Fashion MNIST. We trained on 48,000 examples, tested on 8,000.

Experiments were repeated 5 times with a different random seed for each run.

Data categories were created as suggested by their name in table 6. Examples are shown in figure 12.

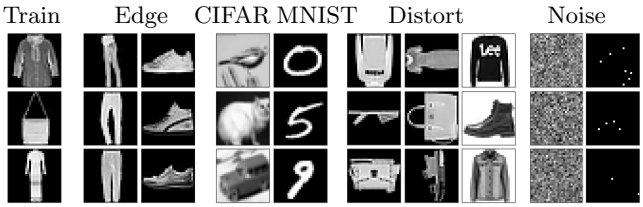

Figure 12: Fashion MNIST OOD data examples.

- **Distort** comprised of rotations, vertical flips, and pixel value inversions.
- **Noise** comprised of iid Gaussian noise, mean = 0.0, standard deviation = 2.0.
- **Sparse** comprised of iid Bernoulli noise, pixles were given a value of 50.0 with p = 0.005, else 0.0.

Hyperparameters: activation = ReLU, optimiser = adam, epochs = 30, learning rate = 0.005, batch size = 256, hidden layers = 3, hidden units = 100

E.7.2 CIFAR-10

CIFAR-10 contains 50,000 32x32 color training images, labelled over 10 categories, and 10,000 test images.

We removed 2 categories during training (ships, dogs) so trained over 40,000 examples.

OOD data classes are as show in the images in table 2.

- **Scramble** permuted each row of pixels in a given image.
- **Invert** took the negative of the pixel values.
- **Noise** sampled pixels from bernoulli distribution (p=0.005) of large magnitude (pixel value=50).

NN architecture: A convolutional NN was used, with the following structure, 64-64-maxpool-128-128-maxpool-256-256-256-maxpool-512-512-512-maxpool-flatten-2048fc-softmax.

All convolutional kernels were [3 x 3 x number of channels in previous layer]. All maxpooling kernels were [2 x 2]. The total number of parameters was 8,689,472.

Hyperparameters: activation = ReLU, optimiser = adam, learning rate = 0.001 decreasing to 0.0005 after 10 epochs and to 0.0001 after 20 epochs, batch size = 300.

In order to bring test accuracies and confidence on the training dataset roughly in line, it was necessary to train for a different number of training epochs for each method (this effectively applies early stopping to the unconstrained case). Anchored eps = 25, Regularise eps = 30, Unconstrained eps = 15.

Experiments were repeated 3 times with a different random seed for each run.

E.7.3 IMDb

Dataset of 25,000 movie reviews, labelled as positive or negative.

Example: *"this movie is the best horror movie bar none i love how stanley just dumps the women into the lake i have been a fan of judd nelson's work for many years and he blew me away its a blend of horror and ... "*

OOD data classes were generated as follows.

- **Reuters** - taken from the Reuters news dataset.

  Example: *"said it has started talks on the possible ... of the company with various parties that it did not identify the company said the talks began after it ... "*

- **Random 1** - A single integers sampled uniformly at random from {1...vocabulary size} and converted to a repeated sequence of words.

  Example: *"member member member member member member member member member member member member member ... "*

- **Random 2** - One integer per word sampled uniformly at random from {1...vocabulary size} and converted to words.

  Example: *"twists mentally superb finest will dinosaur variety models stands knew refreshing member spock might mode lose leonard resemble began happily names... "*

- **Random 3** - As for Random 2, but now only sample from least commonly used 100 words.

  Example: *"computers towers bondage braveheart threatened rear triangle refuse detectives hangs bondage firmly btw token 1990s mermaid reeves landed dylan remove hum natives insightful demonic... "*

NN architecture: used an embedding layer (outputting 20 dimensions), followed by 1D convolutional layer using 50 filters with kernel size of 3 words. Finally a hidden layer with 200 hidden nodes.

Hyperparameters: activation = ReLU, optimiser = adam, learning rate = 0.001, batch size = 64, max sentence length = 200, vocabulary size = 6000

Experiments were repeated 5 times with a different random seed for each run.

E.8 Reinforcement Learning

E.8.1 Uncertainty-Aware Reinforcement Learning

The FetchPush environment from OpenAI Gym was used with the sparse rewards setting. We modified the environment slightly. The goal was positioned at a fixed radius from the block (but at varying angle). Actions were discretised and vertical movements removed so the agent had a choice of moving 0.4 units forward/backwards/left/right. Gaussian noise was added to the actions to make the problem stochastic. Inputs were preprocessed so that relative coordinates of gripper to cube and cube to goal were provided directly to the NNs.

We used fixed target NNs which were updated every 500 episodes.

The simulation was run for 40,000 episodes, with final average rewards around -0.4. Two-layer NNs of 50 nodes were used. Learning rate = 0.001, batch size = 100, episodes in between training = 100, $\gamma = 0.98$, buffer size = 100,000.