Automatic Depth Determination for Bayesian ResNets

Eric Nalisnick
Department of Engineering
University of Cambridge
e.nalisnick@eng.cam.ac.uk

José Miguel Hernández-Lobato
Department of Engineering
University of Cambridge,
Microsoft Research Cambridge,
Alan Turing Institute
jmh233@cam.ac.uk

1 Introduction

The size of neural networks (NNs) is increasing at a steady pace, and as these models gain ever more capacity, proper regularization and model selection become increasingly important. Currently, a deep learning practitioner often needs a GPU cluster to thoroughly search over architectures, regularization strength, and other hyper-parameters. Automatic methods for hyper-parameter tuning based on genetic algorithms [19], reinforcement learning [32], and Bayesian optimization [25] have achieved some success but still can be slow simply due to the challenges of combinatorial optimization. Bayesian inference presents an alternative approach that integrates over parameter uncertainty to find the most suitable model. This is known as the *evidence framework* [16] and has been called the *occam's razor* effect of Bayesian inference [17, 20].

Automatic relevance determination (ARD) [18, 22, 28] is the best known Bayesian method that can automatically tune the size of a neural architecture. Specifically, ARD selects a NN's number of hidden units by placing structured priors on the NN's weights. Denoting a weight in row i and column j as $w_{i,j}$, the ARD prior is defined as

$$w_{i,j} \sim \mathrm{N}(w_{i,j}; 0, \sigma_0^2 \lambda_i), \quad \lambda_i \sim p(\lambda_i), \tag{1}$$

where σ_0^2 is a constant, λ_i is a random scalar, and $p(\lambda_i)$ is a hyper-prior acting on the first-level Gaussian prior's variance. The crucial detail is that each λ_i has a *row index* i, meaning that all of the weights occupying the same row of the weight matrix share the same scale. If $p(\lambda_i)$ places sufficient density near zero, then Bayesian inference can shrink entire rows of weights, effectively pruning them from the NN. Figure 1 shows how the ARD prior partitions a weight matrix $\boldsymbol{W}$: the red lines denote the groups that share the same scale (i.e. the rows).

In this paper, we propose a novel ARD-inspired framework that can automatically select the number of *layers* in a residual network [9]. We term the framework *automatic depth determination* (ADD) as it naturally extends ARD to layers. Moreover, we derive a light-weight EM algorithm to perform approximate inference under ADD priors. The algorithm can be implemented in only a few additional lines of code to most Bayes-by-backprop-style [2] implementations.

2 Automatic Depth Determination

Residual networks (resnets) [9] are NNs with *residual connections* (a.k.a. skip connections) [14, 9, 27] between their hidden layers. Residual connections simply add the previous hidden state to the usual non-linear transformation: $\boldsymbol{h}_l = f_l(\boldsymbol{h}_{l-1}\boldsymbol{W}_l + \boldsymbol{b}_l) + \boldsymbol{h}_{l-1}$ where $\boldsymbol{h}_l$ denotes a vector of hidden units at layer l, $\boldsymbol{W}_l$ the weights, $\boldsymbol{b}_l$ the bias parameters, and f_l the activation function. Resnets have achieved a notable jump in performance on object recognition benchmarks and enabled the training of NNs with $1000+$ layers [9].

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

Since a residual connection allows information to bypass the non-linear transformation, entire weight matrices can be shrunk to zero without obstructing the NN's forward propagation. Thus we can create a prior that selects for *layers* by tying the variance of all weights in the same matrix. By collectively shrinking all the weights in coordination, we can reduce the layer's influence, effectively pruning it in the case of absolute shrinkage to zero. We term this prior *automatic depth determination* (ADD) as it is the natural analog of ARD for network depth. ADD is specified as

$$w_{l,i,j} \sim \mathrm{N}(w_{l,i,j}; 0, \sigma_0^2 \tau_{l,\cdot,\cdot}), \quad \tau_{l,\cdot,\cdot} \sim p(\tau_l), \tag{2}$$

where we have introduced the variable $\tau_{l,\cdot,\cdot}$ that acts as a *per-layer* group variance. We denote this structure by giving τ a layer index l but not a row or column index. Figure 1 shows the ADD prior's structure in comparison to ARD: the blue box encloses all weights with the same scale. As $p(\tau_l)$ places more density near zero, Bayesian inference will increasingly prefer to prune whole weight matrices, i.e. $\mathbf{h}_l \approx f_l(\boldsymbol{h}_{l-1}\mathbf{0}) + \boldsymbol{h}_{l-1} = \boldsymbol{h}_{l-1}$ (assuming f_l is a ReLU and ignoring the bias term), making the network effectively more shallow.

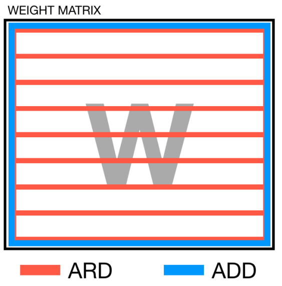


Figure 1: Structure of ARD and ADD Priors

Combining ARD and ADD Yet if Bayesian inference does not decide to prune all of the weights of a layer, regularization still may be necessary. The natural progression is to then select the layer's number of hidden units—as ARD does. Therefore it makes sense to combine ARD and ADD so that the former takes effect when the latter imposes little to no regularization. The joint ARD-ADD prior is specified as

$$w_{l,i,j} \sim \mathrm{N}(w_{l,i,j}; 0, \sigma_0^2 \lambda_{l,i,\cdot}\tau_{l,\cdot,\cdot}), \quad \lambda_{l,i,\cdot} \sim p(\lambda_{l,i}), \quad \tau_{l,\cdot,\cdot} \sim p(\tau_l). \tag{3}$$

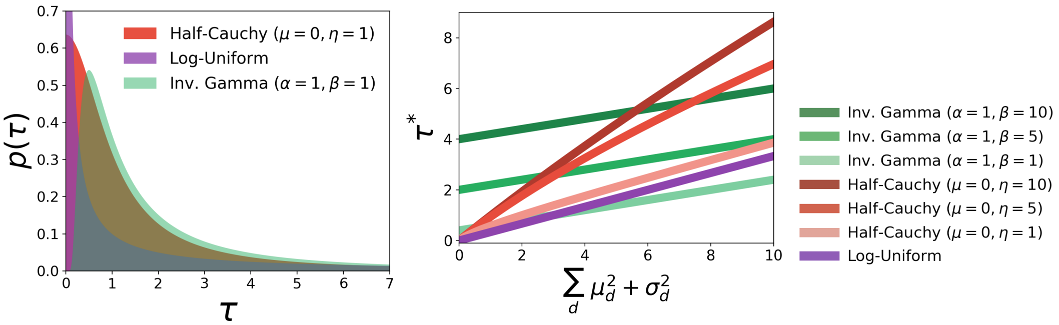
The priors remained essentially unchanged from their original definitions in Equations 1 and 2. The two multiplicatively interact in the first-level prior's variance, and therefore when $\tau_l \to 0$, the product $\lambda_{l,i}\tau_l \approx 0$, effectively turning off the influence of ARD. Conversely, when $\tau_l > 0$, then ARD will act as usual but have its effect modified by a factor of τ_l.

Choosing the Hyper-Priors Having defined the structure of the priors, the next point of consideration is which distributions to choose for $p(\lambda)$ and $p(\tau)$. As pointed out by Neal [22], NNs with Gaussian priors converge to Gaussian processes (GPs) as the network width increases. This can be undesirable because the influence of each hidden unit is diminished, which then stifles the NN's ability to represent latent features. Neal [22] proposes using the inverse gamma as the ARD hyper-prior, in turn making the marginal distribution on the weights a student's-t. The student's-t's heavy tails slow or entirely halt (depending on the setting of the degrees of freedom) the aforementioned convergence to a GP. For even stronger shrinkage, Carvalho et al. [4] and Gelman [7] recommend the half-Cauchy prior on the Gaussian's scale. The resulting marginal distribution is known as the *horseshoe prior*, and it has several beneficial properties such as bounded influence [3]. A stronger prior still is the log-uniform distribution of the form $p(\tau) \propto 1/\tau$. Williams [30] and Toussaint et al. [29] recommend this prior for Bayesian NNs due to its invariance properties, and Kingma et al. [13] derived the prior from a Bayesian interpretation of *dropout* [26]. Figure 2 (a) shows the densities of the three hyper-priors discussed.

3 Inference via Generalized Variational EM

Unfortunately, even approximate inference for scale mixture priors can be challenging—especially when the hyper-priors are half-Cauchy or log-uniform. Previous work performing variational inference for these and similar priors has had to incorporate truncated approximations [24], auxiliary variables [8, 15], non-centered parametrizations [12, 8, 15], and quasi-divergences [11] for the sake of tractability. Instead, we opt for a light-weight inference procedure derived through variational expectation-maximization (VEM) [1]. For most variational Bayesian NN implementations, using VEM with ADD or ARD priors can be implemented in a few lines of code—possibly in as little as one.

Below we detail the inference procedure for ADD, leaving the description of inference for the ARD-ADD prior to Appendix B. Following Wu et al. [31], we assume the posterior approximation



(a) Hyper-Priors

(b) EM Updates for Variance

Figure 2: *Hyper-Priors and EM Updates.* Subfigure (a) shows the density functions of the three hyper-priors considered: half-Cauchy, log-uniform, and inverse gamma. Subfigure (b) shows the EM updates for the posterior variance as a function of $q(\boldsymbol{W})$'s parameters.

	Test Set RMSE				
	Dropout [6]	Prob. Backprop [10]	ARD	ADD	ARD-ADD
Boston Housing	2.80 ± 0.13	2.795 ± 0.16	2.223 ± 0.28	2.221 ± 0.27	$\mathbf{2.208 \pm 0.30}$
Energy Efficiency	$\mathbf{0.47 \pm 0.01}$	0.903 ± 0.05	0.852 ± 0.12	0.855 ± 0.08	0.796 ± 0.15
Yacht	$\mathbf{0.66 \pm 0.06}$	0.848 ± 0.05	0.938 ± 0.08	0.801 ± 0.10	0.793 ± 0.07

Table 1: *Test Set RMSE.* We compare test set RMSE for three UCI regression data sets. As baselines, we use previously reported results for dropout [6] and probabilistic backpropagation [10] applied to two-hidden-layer networks. Our results use the log-uniform prior in all cases.

$p(\boldsymbol{W}, \tau|\boldsymbol{y}, \boldsymbol{X}) \approx q(\boldsymbol{W};\boldsymbol{\phi})q(\tau) = \mathrm{N}(\boldsymbol{W};\boldsymbol{\mu}_{\boldsymbol{\phi}}, \mathrm{daig}\{\boldsymbol{\Sigma}_{\boldsymbol{\phi}}\})\delta[\bar{\tau}_l]$[1] where $\boldsymbol{\phi} = \{\boldsymbol{\mu}_{\boldsymbol{\phi}}, \mathrm{diag}\{\boldsymbol{\Sigma}_{\boldsymbol{\phi}}\}\}$ and $\bar{\tau}_l$ are the variational parameters. The evidence lower bound (ELBO) for this approximation is

$$\begin{aligned} \log p(\boldsymbol{y}|\boldsymbol{X}) &\geq \mathbb{E}_{q(\boldsymbol{W})}\left[\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{W})\right] - \mathbb{E}_{q(\tau)}\mathrm{KL}\left[q(\boldsymbol{W};\boldsymbol{\phi})||p(\boldsymbol{W}|\tau)\right] - \mathrm{KL}\left[q(\tau)||p(\tau)\right] \\ &= \mathbb{E}_{\mathrm{N}(\boldsymbol{W})}\left[\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{W})\right] - \mathrm{KL}\left[\mathrm{N}(\boldsymbol{W};\boldsymbol{\phi})||\mathrm{N}(\boldsymbol{W}|\bar{\tau}_l)\right] - \mathrm{KL}\left[\delta[\bar{\tau}_l]||p(\tau)\right] \\ &= \mathbb{E}_{\mathrm{N}(\boldsymbol{W})}\left[\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{W})\right] - \mathrm{KL}\left[\mathrm{N}(\boldsymbol{W};\boldsymbol{\phi})||\mathrm{N}(\boldsymbol{W}|\bar{\tau}_l)\right] + \log p(\bar{\tau}_l) + \mathcal{C}, \end{aligned} \tag{4}$$

where $\mathcal{C} = \mathbb{H}[\delta[\bar{\tau}_l]]$ is a constant. For the three hyper-priors we consider, $\bar{\tau}_l$ has a closed-form solution that can be found by differentiating the ELBO and setting to zero:

$$\frac{\partial}{\partial \bar{\tau}_l}\mathcal{J}_{\mathrm{ELBO}}(\boldsymbol{\phi}, \bar{\tau}_l) = -\frac{\partial}{\partial \bar{\tau}_l}\mathrm{KL}\left[\mathrm{N}(\boldsymbol{W};\boldsymbol{\phi})||\mathrm{N}(\boldsymbol{W}|\bar{\tau}_l)\right] + \frac{\partial}{\partial \bar{\tau}_l}\log p(\bar{\tau}_l) = 0. \tag{5}$$

We denote the solution to Equation 5 as $\bar{\tau}_l^*$ and give its formula for each hyper-prior in Appendix A. No closed-form exists for updating $q(\boldsymbol{W};\boldsymbol{\phi})$, and hence we perform gradient ascent updates using

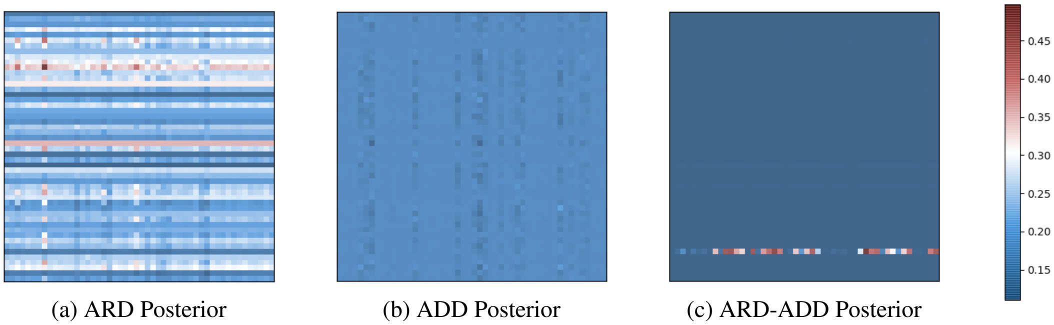
$$\frac{\partial}{\partial \boldsymbol{\phi}}\mathcal{J}_{\mathrm{ELBO}}(\boldsymbol{\phi}, \bar{\tau}_l^*) = \frac{\partial}{\partial \boldsymbol{\phi}}\mathbb{E}_{\mathrm{N}(\boldsymbol{W};\boldsymbol{\phi})}\left[\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{W})\right] - \frac{\partial}{\partial \boldsymbol{\phi}}\mathrm{KL}\left[\mathrm{N}(\boldsymbol{W};\boldsymbol{\phi})||\mathrm{N}(\boldsymbol{W}|\bar{\tau}_l^*)\right]. \tag{6}$$

Since the M-step for $\boldsymbol{\phi}$ in our VEM framework is incremental, we are technically performing *generalized* VEM [23]. Figure 2 (b) shows the value for $\bar{\tau}_l^*$ as a function of the variational parameters $\boldsymbol{\mu}_{\boldsymbol{\phi}}$ and $\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}$. The slope and intercept of each line convey the prior's shrinkage properties. Only the log-uniform and half-Cauchy provide true sparsity, allowing for $\bar{\tau}_l^* = 0$ when $\boldsymbol{\mu}^2_{\boldsymbol{\phi}} + \boldsymbol{\sigma}^2_{\boldsymbol{\phi}} = 0$, no matter the setting of the prior's scale. The inverse Gamma, on the other hand, can set $\bar{\tau}_l^* = 0$ only in the limit as $\alpha \to 0$ and $\beta \to 0$.

4 Experiments

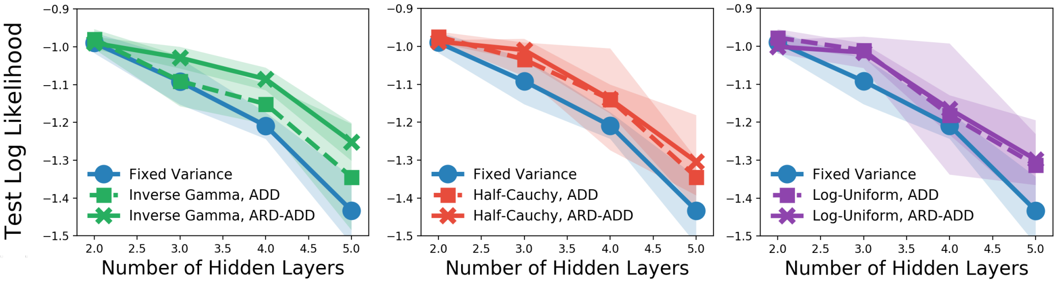
We first test our proposed priors and their EM algorithm on benchmark regression tasks from the UCI repository [5]. We report test set RMSE in Table 1 for the Boston housing, energy efficiency, and

[1] We assume $\delta[\bar{\tau}_l]$ is a *pseudo-Dirac delta* [21] so that the distribution has finite entropy.



(a) ARD Posterior (b) ADD Posterior (c) ARD-ADD Posterior

Figure 3: *Posterior Structure.* The heatmaps above show the posterior structure inferred by the three structured priors considered. We plot the sum of moments $\mu^2 + \sigma^2$ in order to visualize each weight's ability to deviate from zero.



(a) Inverse Gamma $(\alpha = 1, \beta = 1)$ (b) Half-Cauchy $(\mu = 0, \eta = 2)$ (c) Log-Uniform

Figure 4: *Regression on Yacht Data Set.* Results for Bayesian resnets with ADD and ARD-ADD priors are shown for the yacht data set from the UCI repository [5]. The number of hidden layers (x-axis) is varied from two to five to test if the network can remain robust despite changes in depth.

yacht hydrodynamics data sets. A NN with only ARD priors and previously reported dropout [6] and probabilistic backpropagation [10] results serve as baselines. All NNs have two hidden layers, and our ARD, ADD, and ARD-ADD implementations use only the log-uniform prior, as it has no hyperparameters and we wanted to test its utility as a default prior. $q(\boldsymbol{W}; \boldsymbol{\phi})$ was set as a factorized Gaussian. We used the same optimization settings that Gal & Ghahramani [6] used for their two-layer results (batch size of 32, 4000 epochs, default Adam settings, averaged over twenty $90\% - 10\%$ train-test splits). From the table, we see that dropout is a strong baseline and performs best in two out of the three data sets. However, both ADD and ARD-ADD outperform probabilistic backpropagation in all cases and have better RMSE than dropout on Boston housing (2.208 vs 2.80). Furthermore, we see that using the joint ARD-ADD prior is an improvement over ADD for all three data sets, which is expected since it combines the benefits of ARD and ADD.

Next we examine the posterior structure of the Bayesian NNs from Table 1. Figure 3 shows heat maps for the hidden-to-hidden weight matrices when given each prior. The colors are determined by the summed posterior moments $\mu_{\phi}^2 + \sigma_{\phi}^2$ as this quantifies the ability of the parameter to deviate from zero. From subfigure (a), we see that the ARD prior works as depicted in Figure 1: each row learns an independent scale and some rows become almost entirely shrunk (dark blue). Subfigure (b) shows the ADD posterior. ADD shares one scale across the entire matrix, and we see evidence of this in matrix's uniform blue coloring. Lastly, the ARD-ADD posterior is shown in Subfigure (c). Interestingly, the outgoing weights of only one hidden unit remained active, as shown by the line containing a mixture of reds, whites, and blues. The rest of the weights shrunk to near zero (dark blue). This demonstrates the power of the joint prior since a few rows can have large scales while the rest of the matrix remains shrunk. ADD cannot do this since the one row with large weights increases the scale of all the other rows. Conversely, ARD shares no information across rows, but this makes it unlikely to shrink and prune aggressively.

Lastly, we test how well ADD and ARD-ADD can make a NN robust to depth. Regression experiments on the yacht data set from the UCI repository [5] are reported in Figure 3. The x-axis shows the number of hidden layers $(2-5)$, and the y-axis represents test set log likelihood as computed from the posterior predictive distribution (1000 Monte Carlo samples). $q(\boldsymbol{W};\boldsymbol{\phi})$ was again set as a factorized Gaussian. Results are shown for ADD and ARD-ADD priors for the three hyper-priors discussed. The blue lines denote the performance of a Bayesian resnet with a fixed variance that we chose *based on test set performance* over the set $\sigma_0^2 = \{.1, .5, 1, 2, 5, 10\}$. We see that ADD and ARD-ADD were able to best the test-set-chosen variance in each case. The ARD-ADD prior performed better than the ADD prior in each case as well, but the differences are within statistical error.

Conclusions We have proposed two structured priors—*automatic depth determination* (ADD) and joint ARD-ADD—to enable Bayesian reasoning about a neural network's depth. Moreover, their implementation incurs little additional memory or runtime costs to Bayes-by-backprop. Future work includes experiments on larger data sets, comparison against other variational inference strategies, and use of structured variational approximations.

References

[1] Matthew J. Beal and Zoubin Ghahramani. The Variational Bayesian EM Algorithm for Incomplete Data: with application to scoring graphical model structures. *Bayesian Statistics*, 7:453–464, 2003.

[2] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Networks. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1613–1622, 2015.

[3] Carlos M Carvalho, Nicholas G Polson, and James G Scott. Handling Sparsity via the Horseshoe. In *International Conference on Artificial Intelligence and Statistics*, pages 73–80, 2009.

[4] Carlos M Carvalho, Nicholas G Polson, and James G Scott. The Horseshoe Estimator for Sparse Signals. *Biometrika*, page asq017, 2010.

[5] Dua Dheeru and Efi Karra Taniskidou. UCI Machine Learning Repository, 2017.

[6] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1050–1059, 2016.

[7] Andrew Gelman. Prior Distributions for Variance Parameters in Hierarchical Models (Comment on Article by Browne and Draper). *Bayesian Analysis*, 1(3):515–534, 2006.

[8] Soumya Ghosh, Jiayu Yao, and Finale Doshi-Velez. Structured Variational Learning of Bayesian Neural Networks with Horseshoe Priors. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[10] José Miguel Hernández-Lobato and Ryan Adams. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. In *International Conference on Machine Learning*, pages 1861–1869, 2015.

[11] Jiri Hron, Alexander Matthews, and Zoubin Ghahramani. Variational Bayesian Dropout: Pitfalls and Fixes. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1–8, 2018.

[12] John Ingraham and Debora Marks. Variational Inference for Sparse and Undirected Models. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1607–1616, 2017.

[13] Diederik P Kingma, Tim Salimans, and Max Welling. Variational Dropout and the Local Reparameterization Trick. In *Advances in Neural Information Processing Systems*, 2015.

[14] K. J. Lang and M. Witbrock. Learning to Tell Two Spirals Apart. In *1988 Connectionist Models Summer School*, 1988.

[15] Christos Louizos, Karen Ullrich, and Max Welling. Bayesian Compression for Deep Learning. In *Advances in Neural Information Processing Systems*, pages 3288–3298, 2017.

[16] David Mackay. The Evidence Framework Applied to Classification Networks. *Neural Computation*, 4(5):720–736, 1992.

[17] David JC MacKay. Bayesian Interpolation. *Neural Computation*, 4(3):415–447, 1992.

[18] David JC MacKay. Bayesian Non-Linear Modeling for the Prediction Competition. In *Maximum Entropy and Bayesian Methods*, pages 221–234. Springer, 1994.

[19] Geoffrey F Miller, Peter M Todd, and Shailesh U Hegde. Designing Neural Networks using Genetic Algorithms. In *ICGA*, volume 89, pages 379–384, 1989.

[20] Iain Murray and Zoubin Ghahramani. A Note on the Evidence and Bayesian Occam's Razor. *Gatsby Unit Technical Report*, 2005.

[21] Shinichi Nakajima and Masashi Sugiyama. Analysis of Empirical MAP and Empirical Partially Bayes: Can They be Alternatives to Variational Bayes? In *Artificial Intelligence and Statistics*, pages 20–28, 2014.

[22] Radford M Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1994.

[23] Radford M Neal and Geoffrey E Hinton. A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants. In *Learning in Graphical Models*, pages 355–368. Springer, 1998.

[24] Kirill Neklyudov, Dmitry Molchanov, Arsenii Ashukha, and Dmitry P Vetrov. Structured Bayesian Pruning via Log-Normal Multiplicative Noise. In *Advances in Neural Information Processing Systems*, pages 6778–6787, 2017.

[25] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.

[26] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[27] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training Very Deep Networks. In *Advances in Neural Information Processing Systems*, pages 2377–2385, 2015.

[28] Michael E Tipping. Sparse Bayesian Learning and the Relevance Vector Machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

[29] Udo v Toussaint, Silvio Gori, and Volker Dose. Invariance Priors for Bayesian Feed-Forward Neural Networks. *Neural Networks*, 19(10):1550–1557, 2006.

[30] Peter M Williams. Matrix Logarithm Parametrizations for Neural Network Covariance Models. *Neural Networks*, 12(2):299–308, 1999.

[31] A. Wu, S. Nowozin, E. Meeds, R. E. Turner, J. M. Hernández-Lobato, and A. L. Gaunt. Fixing Variational Bayes: Deterministic Variational Inference for Bayesian Neural Networks. *ArXiv e-prints*, 2018.

[32] Barret Zoph and Quoc V Le. Neural Architecture Search with Reinforcement Learning. *International Conference on Learning Representations (ICLR)*, 2017.

A ADD EM Updates

Below we derive the EM updates for the various hyper-priors considered. For all expressions, we assume the variational distribution on the weights factorizes, i.e. $\text{N}(\boldsymbol{W};\boldsymbol{\phi}) = \prod_i \prod_j \text{N}(w_{i,j};\boldsymbol{\phi}_{i,j})$ where i denotes row and j column indices.

A.1 Inverse Gamma

We begin with the ELBO terms that depend on the variational variance τ, Equation 4:

$$\begin{aligned}
\mathcal{J}_{\text{ELBO}}(\boldsymbol{\phi}, \bar{\tau}_l) &= -\text{KL}\left[\text{N}(\boldsymbol{W};\boldsymbol{\phi})||\text{N}(\boldsymbol{W}|\bar{\tau}_l)\right] + \log p(\bar{\tau}_l) \\
&= \frac{-1}{2}\sum_i\sum_j\left[\log\frac{\bar{\tau}_l}{\sigma_{i,j}^2} + \frac{\sigma_{i,j}^2}{\bar{\tau}_l} + \frac{\mu_{i,j}^2}{\bar{\tau}_l} - 1\right] + \log\Gamma^{-1}(\bar{\tau}_l;\alpha,\beta) \\
&= \frac{-1}{2}\left[\sum_i\sum_j\log\frac{\bar{\tau}_l}{\sigma_{i,j}^2} + \sum_i\sum_j\frac{\sigma_{i,j}^2+\mu_{i,j}^2}{\bar{\tau}_l} - D\right] - \frac{\beta}{\bar{\tau}_l} - (\alpha+1)\log\bar{\tau}_l + \log\frac{\beta^\alpha}{\Gamma(\alpha)}.
\end{aligned}$$

Differentiating w.r.t. $\bar{\tau}_l$ and setting to zero, we have

$$\begin{aligned}
0 &= \frac{\partial}{\partial\bar{\tau}_l}\mathcal{J}_{\text{ELBO}}(\boldsymbol{\phi}, \bar{\tau}_l) \\
&= \frac{-1}{2}\left[\sum_i\sum_j\frac{1}{\bar{\tau}_l} - \frac{\sigma_{i,j}^2+\mu_{i,j}^2}{\bar{\tau}_l^2}\right] + \frac{\beta}{\bar{\tau}_l^2} - \frac{\alpha+1}{\bar{\tau}_l} \\
&= \frac{D}{\bar{\tau}_l} - \frac{\sum_i\sum_j\sigma_{i,j}^2+\mu_{i,j}^2}{\bar{\tau}_l^2} - \frac{2\beta}{\bar{\tau}_l^2} + \frac{2\alpha+2}{\bar{\tau}_l} \\
\frac{2\beta + \sum_i\sum_j\sigma_{i,j}^2+\mu_{i,j}^2}{\bar{\tau}_l^2} &= \frac{D+2\alpha+2}{\bar{\tau}_l} \\
\bar{\tau}_l^* &= \frac{2\beta + \sum_i\sum_j\sigma_{i,j}^2+\mu_{i,j}^2}{D+2\alpha+2}
\end{aligned}$$

where D are the number of dimensions (i.e. parameters) in $\boldsymbol{W}$.

A.2 Half-Cauchy

Starting with Equation 5, we have

$$\begin{aligned}
0 &= \frac{\partial}{\partial\bar{\tau}_l}\mathcal{J}_{\text{ELBO}}(\boldsymbol{\phi}, \bar{\tau}_l) \\
&= \frac{-1}{2}\frac{\partial}{\partial\bar{\tau}_l}\sum_i\sum_j\left[\log\frac{\bar{\tau}_l}{\sigma_{i,j}^2} + \frac{\sigma_{i,j}^2}{\bar{\tau}_l} + \frac{\mu_{i,j}^2}{\bar{\tau}_l} - 1\right] + \frac{\partial}{\partial\bar{\tau}_l}\log C^+(\sqrt{\bar{\tau}_l};0,\eta) \\
&= \frac{-1}{2}\left[\sum_i\sum_j\frac{1}{\bar{\tau}_l} - \frac{\sigma_{i,j}^2+\mu_{i,j}^2}{\bar{\tau}_l^2}\right] + \frac{\partial}{\partial\bar{\tau}_l}\log\frac{2}{\pi\eta(1+\bar{\tau}_l/\eta^2)} \\
&= \frac{D}{\bar{\tau}_l} - \frac{\sum_i\sum_j\sigma_{i,j}^2+\mu_{i,j}^2}{\bar{\tau}_l^2} + \frac{2}{\eta^2+\bar{\tau}_l} \\
&= \frac{2\bar{\tau}_l}{\eta^2+\bar{\tau}_l} - \frac{\sum_i\sum_j\sigma_{i,j}^2+\mu_{i,j}^2}{\bar{\tau}_l} + D.
\end{aligned}$$

Solving[2] the above equation for $\bar{\tau}_l$ gives for a positive solution:

$$\bar{\tau}_l^* = \frac{M - \eta^2 D + \sqrt{M^2 + (2D+8)\eta^2 M + \eta^4 D^2}}{2D+4} \quad \text{where} \quad M = \sum_i\sum_j\sigma_{i,j}^2 + \mu_{i,j}^2,$$

D is again the dimensionality, and η is the half-Cauchy's scale.

[2] We plugged the equation into Wolfram Alpha.

A.3 Log-Uniform

Again beginning with Equation 5, we have

$$\begin{aligned}
0 &= \frac{\partial}{\partial \bar{\tau}_l} \mathcal{J}_{\text{ELBO}}(\boldsymbol{\phi}, \bar{\tau}_l) \\
&= \frac{-1}{2} \left[\sum_i \sum_j \frac{1}{\bar{\tau}_l} - \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l^2} \right] + \frac{\partial}{\partial \bar{\tau}_l} \log \frac{c}{\bar{\tau}_l} \\
&= \frac{D}{\bar{\tau}_l} - \frac{\sum_i \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l^2} + \frac{2}{\bar{\tau}_l} \\
\frac{\sum_i \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l^2} &= \frac{D+2}{\bar{\tau}_l} \\
\bar{\tau}_l^* &= \frac{\sum_i \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{D+2}.
\end{aligned}$$

Notice that this update is the same as the inverse Gamma's as $\alpha \to 0$ and $\beta \to 0$.

B ARD-ADD EM Updates

For approximate posterior inference for the joint ARD-ADD prior, we assume the posterior approximation

$$p(\boldsymbol{W}, \lambda, \tau | \boldsymbol{y}, \boldsymbol{X}) \approx q(\boldsymbol{W}; \boldsymbol{\phi})\, q(\lambda)\, q(\tau) = \text{N}(\boldsymbol{W}; \boldsymbol{\mu}_{\boldsymbol{\phi}}, \text{daig}\{\boldsymbol{\Sigma}_{\boldsymbol{\phi}}\})\, \delta[\bar{\lambda}_i]\, \delta[\bar{\tau}_l]$$

where $\boldsymbol{\phi} = \{\boldsymbol{\mu}_{\boldsymbol{\phi}}, \text{diag}\{\boldsymbol{\Sigma}_{\boldsymbol{\phi}}\}\}$, $\bar{\lambda}_i$, and $\bar{\tau}_l$ are the variational parameters. The ELBO for this approximation is

$$\begin{aligned}
&\log p(\boldsymbol{y}|\boldsymbol{X}) \geq \\
&\mathbb{E}_{q(\boldsymbol{W})} \left[\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{W})\right] - \mathbb{E}_{q(\lambda)} \mathbb{E}_{q(\tau)} \text{KL} \left[q(\boldsymbol{W}; \boldsymbol{\phi}) || p(\boldsymbol{W}|\lambda, \tau)\right] - \text{KL} \left[q(\lambda) || p(\lambda)\right] - \text{KL} \left[q(\tau) || p(\tau)\right] \\
&= \mathbb{E}_{\text{N}(\boldsymbol{W})} \left[\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{W})\right] - \text{KL} \left[\text{N}(\boldsymbol{W}; \boldsymbol{\phi}) || \text{N}(\boldsymbol{W}|\bar{\lambda}_i, \bar{\tau}_l)\right] - \text{KL} \left[\delta[\bar{\lambda}_i] || p(\lambda)\right] - \text{KL} \left[\delta[\bar{\tau}_l] || p(\tau)\right] \\
&= \mathbb{E}_{\text{N}(\boldsymbol{W})} \left[\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{W})\right] - \text{KL} \left[\text{N}(\boldsymbol{W}; \boldsymbol{\phi}) || \text{N}(\boldsymbol{W}|\bar{\lambda}_i, \bar{\tau}_l)\right] + \log p(\bar{\lambda}_i) + \log p(\bar{\tau}_l) + \mathcal{C}
\end{aligned} \tag{7}$$

where $\mathcal{C} = \mathbb{H}[\delta[\bar{\lambda}_i]] + \mathbb{H}[\delta[\bar{\tau}_l]]$ is again a constant. Next we must find solutions for both $\bar{\lambda}_i$ and $\bar{\tau}_l$. Again we can differentiate the ELBO and set to zero:

$$\begin{aligned}
\frac{\partial}{\partial \bar{\lambda}_i} \mathcal{J}_{\text{ELBO}}(\boldsymbol{\phi}, \bar{\lambda}_i, \bar{\tau}_l) &= -\frac{\partial}{\partial \bar{\lambda}_i} \text{KL} \left[\text{N}(\boldsymbol{W}; \boldsymbol{\phi}) || \text{N}(\boldsymbol{W}|\bar{\lambda}_i, \bar{\tau}_l)\right] + \frac{\partial}{\partial \bar{\lambda}_i} \log p(\bar{\lambda}_i) = 0, \\
\frac{\partial}{\partial \bar{\tau}_l} \mathcal{J}_{\text{ELBO}}(\boldsymbol{\phi}, \bar{\lambda}_i, \bar{\tau}_l) &= -\frac{\partial}{\partial \bar{\tau}_l} \text{KL} \left[\text{N}(\boldsymbol{W}; \boldsymbol{\phi}) || \text{N}(\boldsymbol{W}|\bar{\lambda}_i, \bar{\tau}_l)\right] + \frac{\partial}{\partial \bar{\tau}_l} \log p(\bar{\tau}_l) = 0.
\end{aligned} \tag{8}$$

We denote the solutions as $\bar{\lambda}_i^*$ and $\bar{\tau}_l^*$, deriving them for the three hyper-priors below. Again, we update $q(\boldsymbol{W}; \boldsymbol{\phi})$ via gradient ascent:

$$\frac{\partial}{\partial \boldsymbol{\phi}} \mathcal{J}_{\text{ELBO}}(\boldsymbol{\phi}, \bar{\lambda}_i^*, \bar{\tau}_l^*) = \frac{\partial}{\partial \boldsymbol{\phi}} \mathbb{E}_{\text{N}(\boldsymbol{W}; \boldsymbol{\phi})} \left[\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{W})\right] - \frac{\partial}{\partial \boldsymbol{\phi}} \text{KL} \left[\text{N}(\boldsymbol{W}; \boldsymbol{\phi}) || \text{N}(\boldsymbol{W}|\bar{\lambda}_i^*, \bar{\tau}_l^*)\right]. \tag{9}$$

B.1 Inverse Gamma

Starting with the first line of Equation 8, we have

$$
\begin{aligned}
0 &= \frac{\partial}{\partial \bar{\lambda}_i} \mathcal{J}_{\text{ELBO}}(\boldsymbol{\phi}, \bar{\lambda}_i, \bar{\tau}_l) \\
&= \frac{-1}{2} \frac{\partial}{\partial \bar{\lambda}_i} \sum_i \sum_j \left[\log \frac{\bar{\lambda}_i \bar{\tau}_l}{\sigma_{i,j}^2} + \frac{\sigma_{i,j}^2}{\bar{\lambda}_i \bar{\tau}_l} + \frac{\mu_{i,j}^2}{\bar{\lambda}_i \bar{\tau}_l} - 1 \right] + \frac{\partial}{\partial \bar{\lambda}_i} \log \Gamma^{-1}(\bar{\lambda}_i; \alpha, \beta) \\
&= \frac{-1}{2} \left[\sum_j \frac{1}{\bar{\lambda}_i} - \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\lambda_i^2 \bar{\tau}_l} \right] + \frac{\beta}{\bar{\lambda}_i^2} - \frac{\alpha + 1}{\bar{\lambda}_i} \\
&= \frac{D_i}{\bar{\lambda}_i} - \frac{\sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\lambda}_i^2 \bar{\tau}_l} - \frac{2\beta}{\bar{\lambda}_i^2} + \frac{2\alpha + 2}{\bar{\lambda}_i} \\
\frac{2\beta \bar{\tau}_l + \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\lambda}_i^2 \bar{\tau}_l} &= \frac{D_i + 2\alpha + 2}{\bar{\lambda}_i} \\
\bar{\lambda}_i^* &= \frac{2\beta \bar{\tau}_l + \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l (D_i + 2\alpha + 2)}
\end{aligned}
$$

where D_i is the number of parameters in the ith row (i.e. λ's corresponding row) of the weight matrix $\boldsymbol{W}$. Furthermore, notice that the sum over posterior parameters is across only columns (j index).

Moving on the ADD parameter, we begin with the second line of Equation 8:

$$
\begin{aligned}
0 &= \frac{\partial}{\partial \bar{\tau}_l} \mathcal{J}_{\text{ELBO}}(\boldsymbol{\phi}, \bar{\lambda}_i, \bar{\tau}_l) \\
&= \frac{-1}{2} \frac{\partial}{\partial \bar{\tau}_l} \sum_i \sum_j \left[\log \frac{\bar{\lambda}_i \bar{\tau}_l}{\sigma_{i,j}^2} + \frac{\sigma_{i,j}^2}{\bar{\lambda}_i \bar{\tau}_l} + \frac{\mu_{i,j}^2}{\bar{\lambda}_i \bar{\tau}_l} - 1 \right] + \frac{\partial}{\partial \bar{\tau}_l} \log \Gamma^{-1}(\bar{\tau}_l; \alpha, \beta) \\
&= \frac{-1}{2} \left[\sum_i \sum_j \frac{1}{\bar{\tau}_l} - \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\lambda}_i \bar{\tau}_l^2} \right] + \frac{\beta}{\bar{\tau}_l^2} - \frac{\alpha + 1}{\bar{\tau}_l} \\
&= \frac{D}{\bar{\tau}_l} - \frac{\sum_i \bar{\lambda}_i^{-1} \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l^2} - \frac{2\beta}{\bar{\tau}_l^2} + \frac{2\alpha + 2}{\bar{\tau}_l} \\
\frac{2\beta + \sum_i \bar{\lambda}_i^{-1} \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l^2} &= \frac{D + 2\alpha + 2}{\bar{\tau}_l} \\
\bar{\tau}_l^* &= \frac{2\beta + \sum_i \bar{\lambda}_i^{-1} \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{D + 2\alpha + 2}.
\end{aligned}
$$

B.2 Half-Cauchy

Starting with the first line of Equation 8, we have

$$
\begin{aligned}
0 &= \frac{\partial}{\partial \bar{\lambda}_i} \mathcal{J}_{\text{ELBO}}(\boldsymbol{\phi}, \bar{\lambda}_i, \bar{\tau}_l) \\
&= \frac{-1}{2} \frac{\partial}{\partial \bar{\lambda}_i} \sum_i \sum_j \left[\log \frac{\bar{\lambda}_i \bar{\tau}_l}{\sigma_{i,j}^2} + \frac{\sigma_{i,j}^2}{\bar{\lambda}_i \bar{\tau}_l} + \frac{\mu_{i,j}^2}{\bar{\lambda}_i \bar{\tau}_l} - 1 \right] + \frac{\partial}{\partial \bar{\lambda}_i} \log C^{+}(\sqrt{\bar{\lambda}_i}; 0, \eta) \\
&= \sum_j \frac{1}{\bar{\lambda}_i} - \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\lambda_i^2 \bar{\tau}_l} + \frac{2}{\eta^2 + \bar{\lambda}_i} \\
&= \frac{2\bar{\lambda}_i}{\eta^2 + \bar{\lambda}_i} - \frac{\bar{\tau}_l^{-1} \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\lambda}_i} + D_i
\end{aligned}
$$

where D_i is the number of parameters in the ith row. This final expression is of the same form we solved for ADD and therefore has the same solution with appropriately adjusted constants:

$$\bar{\lambda}_i^* = \frac{M_i - \eta^2 D_i + \sqrt{M_i^2 + (2D_i + 8)\eta^2 M_i + \eta^4 D_i^2}}{2D_i + 4} \quad \text{where} \;\; M_i = \bar{\tau}_l^{-1} \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2.$$

Moving on the ADD parameter, we consider the second line of Equation 8:

$$\begin{aligned}
0 &= \frac{\partial}{\partial \bar{\tau}_l} \mathcal{J}_{\text{ELBO}}(\boldsymbol{\phi}, \bar{\lambda}_i, \bar{\tau}_l) \\
&= \frac{-1}{2} \frac{\partial}{\partial \bar{\tau}_l} \sum_i \sum_j \left[\log \frac{\bar{\lambda}_i \bar{\tau}_l}{\sigma_{i,j}^2} + \frac{\sigma_{i,j}^2}{\bar{\lambda}_i \bar{\tau}_l} + \frac{\mu_{i,j}^2}{\bar{\lambda}_i \bar{\tau}_l} - 1 \right] + \frac{\partial}{\partial \bar{\tau}_l} \log C^+(\sqrt{\bar{\tau}_l}; 0, \eta) \\
&= \sum_i \sum_j \frac{1}{\bar{\tau}_l} - \frac{\sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\lambda}_i \bar{\tau}_l^2} + \frac{2}{\eta^2 + \bar{\tau}_l} \\
&= \frac{D}{\bar{\tau}_l} - \frac{\sum_i \bar{\lambda}_i^{-1} \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l^2} + \frac{2}{\eta^2 + \bar{\tau}_l} \\
&= \frac{2\bar{\tau}_l}{\eta^2 + \bar{\tau}_l} - \frac{\sum_i \bar{\lambda}_i^{-1} \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l} + D.
\end{aligned}$$

Again solving the same equation yields the solution:

$$\bar{\tau}_l^* = \frac{M_\lambda - \eta^2 D + \sqrt{M_\lambda^2 + (2D + 8)\eta^2 M_\lambda + \eta^4 D^2}}{2D + 4} \quad \text{where} \;\; M_\lambda = \sum_i \bar{\lambda}_i^{-1} \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2.$$

B.3 Log-Uniform

As mentioned earlier, the solution for the log-uniform distribution can be attained in the limit of the inverse Gamma. Plugging $\alpha = 0$ and $\beta = 0$ into the inverse gamma's solutions we obtain:

$$\bar{\lambda}_i^* = \frac{\sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{\bar{\tau}_l (D_i + 2)} \quad \text{and} \quad \bar{\tau}_l^* = \frac{\sum_i \bar{\lambda}_i^{-1} \sum_j \sigma_{i,j}^2 + \mu_{i,j}^2}{D + 2}.$$