Adaptive Neural Connections for Sparsity Learning

Alex Gain*
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21210
again1@jhu.edu

Prakhar Kaushik*
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21210
pkaushi1@jhu.edu

Hava Siegelmann
School of Computer Science
University of Massachussetts Amherst
Amherst, MA, 01003
hava@cs.umass.edu

Abstract

Sparsity learning aims to decrease the computational and memory costs of large deep neural networks (DNNs) via pruning neural connections while simultaneously retaining high accuracy. A large body of work has developed sparsity learning approaches, with recent large-scale experiments showing that two main methods, magnitude pruning and Variational Dropout (VD), achieve similar state-of-the-art results for classification tasks. We propose Adaptive Neural Connections (ANC), a method for explicitly parameterizing fine-grained neuron-to-neuron connections via adjacency matrices at each layer that are learned through backpropagation. Explicitly parameterizing neuron-to-neuron connections confers two primary advantages: 1. Sparsity can be explicitly optimized for via norm-based regularization on the adjacency matrices; and 2. When combined with VD (which we term, ANC-VD), the adjacencies can be interpreted as learned weight importance parameters, which we hypothesize leads to improved convergence for VD. Experiments with ResNet18 show that architectures augmented with ANC outperform their vanilla counterparts.

1. Introduction

Deep neural networks (DNNs) have achieved great success in recent years, continuously breaking benchmarks in a breadth of applications and research areas. With the aim of improving state-of-the-art (SOTA) results, architectures have become increasingly large and complex, intensive with respect to both memory and computation. Thus, it is of great interest to decrease the number of parameters a neural network uses while retaining similar quality of inference. There are a number of categories of approaches for decreasing the parameters of a neural network, such as the well-known student teacher methods . In this paper, we address the subset of methods where connections between neurons are set to zero or pruned in some way, termed sparsity learning . By sparsifying models, memory and computational intensiveness of neural networks can be drastically reduced.

A large number of methods in recent years have been proposed to sparsify models. Recently, a large scale study with tight methodology (Gale et al. (2019)) has shown that most of these methods fail to outperform simple magnitude pruning, a sparsity method where all weights below a certain magnitude threshold are set to 0, with retraining occurring afterwards. In the case of classification tasks, Variational Dropout (VD), a method in which parameterizations of Gaussian priors on all individual weights are learned via training allowing for pruning of weights based on the learned distributional parameters, outperforms magnitude pruning and all other tested methods in terms of test accuracy at nearly all sparsity levels and especially at the highest sparsity levels. A slightly modified version of magnitude pruning taking into the account the depth of the layers was shown to marginally outperform or approximately equal the test accuracy of VD at most sparsity levels, with VD achieving higher test accuracy at the sparsest levels (greater than 95% parameters pruned).

In this paper, we improve upon VD by explicitly parameterizing the adjacencies of neuron-to-neuron connections. At each layer, the weight matrix, or weight kernel, is multiplied by a binary mask of the same shape. A soft mask, with values between 0 and 1, is passed through a differentiable

*Equal contribution.

rounding function that approximates a hard round function. In this sense, neuron to neuron connections can be learned adaptively through backpropagation via this method. We term this layer-wise method, Adaptive Neural Connections (ANC). Through explicit representation of neuron-to-neuron connections, ANC allows us to explicitly induce sparsity through norm based regulation on the adjacency parameters. Additionally, we give a Bayesian interpretation for why this is relevant and beneficial to VD, when combined together, termed Adaptive Neural Connection Augmented Variational Dropout (ANC-VD).

Experimentally, ADC-VD outperforms VD in terms of test accuracy at nearly all sparsity levels. ADC-VD is able to achieve higher sparsity with less test accuracy degradation, decreased computational wall time, decreased memory cost, and decreased number computational operations at inference, compared to all other methods. These outcomes are achieved in significantly less training iterations than VD. Our experiments and comparisons make use of a ResNet18 architecture, as well as a VGG18 architecture, with training and testing done primarily on CIFAR-10. We leave preliminary experiments showing similar results on CINIC-10 and ImageNet-32 as well.

Contributions: We introduce an easily implementable, general method for sparsity learning that can be applied to virtually any sparsity learning method or network architecture. We give an interpretation as to why this method is particularly suited to Variational Dropout (VD), which is state-of-the-art or nearly state-of-the-art for sparse image classification. For computational expediency and to ease the complexity of comparisons, we take the top two methods from (Gale et al., 2019) (VD and magnitude pruning) and focus on improving upon them. Experimental results show our method moderately improves upon VD and marginally improves upon magnitude pruning. Overall, ANC shows promise as a practical method that can be broadly applied in future sparsification research.

2. Background and Related Work

Many techniques have been developed for sparsity learning. (Wen et al., 2016) makes use of specific norm-based regularization expressions to adapt filter-wise and depth-wise capacity for sparsity, and (Liu et al., 2017) use a similar category of methods to induce channel-wise sparsity. (Zhu and Gupta, 2017) is where magnitude pruning is introduced via simple norm-based thresholds and scheduling of the pruning. (Kingma et al., 2015) introduced variational dropout, a Gaussian-prior formulation of dropout, and (Zhu and Gupta, 2017) applied variational dropout to the sparsity learning setting. (Liu et al., 2018) argue that sparsification is a form of neural architecture search, though (Gale et al., 2019) conducted experiments that contradict that claim, to some extent. Other related sparsity approaches include Louizos et al. (2017) which uses an L0-norm based regularization for sparsity on the weights and Liu et al. (2018), which uses a variational technique to compress weights.

As for similarities to ANC, any method that performs some sort of masking will fall into the same operational category, since ANC can be formulated as equivalent to a specific soft mask formulation. For example, since it does channel-wise masking, (Liu et al., 2017) is a specific case of ANC. (Bejnordi et al., 2019) is also a special case of ANC with different non-linearities and priors on gating functions. However, current methods do not attempt to explicitly represent *all* neuron-to-neuron connections adaptively, nor are there masking techniques that are depth-wise modulated (shown in section 3).

3. Adaptive Neural Connections

As quick illustration, we give ANC as defined for a simple multi-layer perceptron, denoted as f. Let mapping f from $\mathbb{R}^{d_1} \longrightarrow \mathbb{R}^{d_3}$ be defined as

$$f(x) \triangleq \sigma(W_2\sigma(W_1 x)) \tag{1}$$

where $x \in \mathbb{R}^{d_1}$, $W_1 \in \mathbb{R}^{d_2 \times d_1}$, and $W_2 \in \mathbb{R}^{d_3 \times d_2}$.

Denote the set of weights as $\mathbf{W} \triangleq \{W_1, W_2\}$. Define a corresponding set of masks as $\mathbf{A} = \{A_1, A_2\}$, where $A_1 \in \{0,1\}^{d_2 \times d_1}$ and $A_2 \in \{0,1\}^{d_3 \times d_2}$. In other words, A_1 and A_2 are the same shape as their corresponding weights, and explicitly represent the neuron-to-neuron connections of the network. Applying $\mathbf{A}$ to f, we define as

$$f(x; \mathbf{W}, \mathbf{A}) \triangleq \sigma((W_2 \odot A_2)\sigma((W_1 \odot A_2)x)) \tag{2}$$

where $\odot$ denotes element-wise multiplication. The definition for MLPs can be generalized to *any* architecture with weight tensors.

However, $\mathbf{A}$ cannot learned through backpropagation due to its non-differentiability (you would need to incorporate a hard rounding function, which is not differentiable). One way around this is to approximate the hard rounding function through a sigmoidal function:

$$soft\_round(x; \beta) = \frac{1}{1 + \exp(-(\beta(x - 0.5)))} \tag{3}$$

This function is element-wise, and β is a "tightening" parameter that controls how closely $soft\_round$ approximates the hard rounding function. Here, we assume $x \in [0, 1]$, and in fact we initialize our pre-soft-rounded $\mathbf{A}$ parameters in experiments with skewed, clipped Gaussians with support $[0, 1]$. In experiments, β can be restricted to be non-decreasing so as to coerce the network towards $\mathbf{A}$ parameters that can be safely converted into binary values and combined with their corresponding W parameters at inference time so as to not superfluously introduce a higher parameter count.

3.1. Bayesian Interpretation

We now discuss ANC from a Bayesian point of view, and discuss how this relates to VD.

3.1.1 Automatic Relevance Determination and ANC

Automatic relevance determination (ARD) and the closely-related sparse Bayesian learning (SBL) framework are effective tools for pruning large numbers of irrelevant features leading to a sparse explanatory subset (Wipf and Nagarajan, 2008).

Like mentioned in Wipf and Nagarajan (2008), if we present a model,

$$y = \gamma x + \eta \tag{4}$$

where $\gamma \in \mathbb{R}^{nXm}$ is a dictionary of features, $x \in R^m$ is a vector of unknown weights, y is an observation vector, and η is uncorrelated noise distributed as $N(\eta; 0, \lambda I)$ and if the number of features is too large, the problem becomes ill-poised, complex and may overfit.

ARD addresses this problem by regularizing the solution space using a parameterized, data-dependent prior distribution that effectively prunes away redundant or superfluous features (Neal, 2012). As from Wipf and Nagarajan (2008), the basic (ARD) prior that SBL incorporates can be defined as $p(x; y) = \mathcal{N}(x; 0, diag[\lambda])$ where $\lambda \in \mathbb{R}_+^m$ is a vector of m non-negative hyperparameters governing the prior variance of each unknown coefficient.

These hyperparameters are estimated from the data by minimizing

$$\begin{aligned}\mathbb{L}(\lambda) \triangleq &- \log \int p(y|x)p(x;\lambda)dx \\ &= -\log(p(y;\lambda)) \equiv \log|\sum_y| + y^T \sum_y^{-1} y \end{aligned} \tag{5}$$

where, a flat hyperprior on λ is assumed. This minimization is also referred to as *evidence maximization* (MacKay, 1992). If any $\lambda_{*,i} = 0$, as often happens while the model is learning, then $x_{ARD,i} = 0$ (the posterior mean), effectively pruning the respective feature from the model, resulting in a relatively sparse weight vector.

With respect to the above-mentioned Automatic Relevance Determination (ARD) work, which we can interpret as a Bayesian framework of placing Gaussian priors on the Neural Networks' weights and then structured hyper-priors on the Gaussian prior, we can draw a parallel to our experiment of combining the neuron connection matrix A with Variational Dropout (Kingma et al., 2015). Instead of tying the weights together in the layer as a form of combined regularization, the neuron connection matrix A computes the *importance* of the connected layers' weights/parameters.

Kharitonov et al. (2018) actually shows ARD applied to Bayesian DNNs with Gaussian approximate posterior distributions leads to a variational bound which is similar to that of variational dropout, and in the case of a *fixed dropout rate*, objectives are exactly the same. In our case, we believe the adjacency matrix models the hyperprior defined above.

3.2. DropConnect and ANC

Dropconnect (Wipf and Nagarajan, 2008) was introduced in 2013 as a generalization of the previously available Dropout. Dropconnect sets a randomly selected subset of weights within the network to zero instead of Dropout's method of doing the same with a random sunset of units in the previous layer.

For a DropConnect layer, the output is given as:

$$r = a((M * W)v) \tag{6}$$

where $*$ denotes element wise product, W represents the weights, v is the input vector, a is the non-linear activation function, M is a binary matrix encoding the connection information and $M_{ij} \sim Bernoulli(p)$. Each element of the mask M is drawn independently for each example during training, essentially instantiating a different connectivity for each example seen. In our case, we can see the adjacency layer as a soft Dropconnect Mask assignment. Instead of hard random assignment $\{0, 1\}$, we assign softer values (which are randomly initialised) to individual weights.

4. Experiments

Here, we compare the sparsity learning methods Variational Dropout and magnitude pruning to their ANC-augmented counter-parts. Experiments are primarily conducted on ResNet18 and VGG18-like architectures, and the CIFAR-10 dataset, though less extensive results are shown for MNIST, CINIC-10, and ImageNet32 as well. Generally, we compare test accuracy at each sparsity level for each method.

Applied to MNIST, sparsity learning methods can easily achieve greater than 99% sparsity. Coming from the ANC perspective, we can visualize the exact connections learned simply through observing the neuron connection representation values $\mathbf{A}$. As a simple illustration of this sparsity learned over training time, Figure 1 shows the learned adjacency matrices of a VGG network. Here, we can see large sparsity in the convolutional layer shown, and grid-like patterns suggesting local feature-dependencies were learned.

Next, we compare the performance of ANC-VD and VD via a VGG variant on CIFAR-10. The sparsity learned for an α parameter[1] threshold value of 3.0 is shown over training

[1] α corresponds to how stringently connections are pruned, though its exact value does not seem to have much effect in classification settings, as found in Gale et al. (2019)

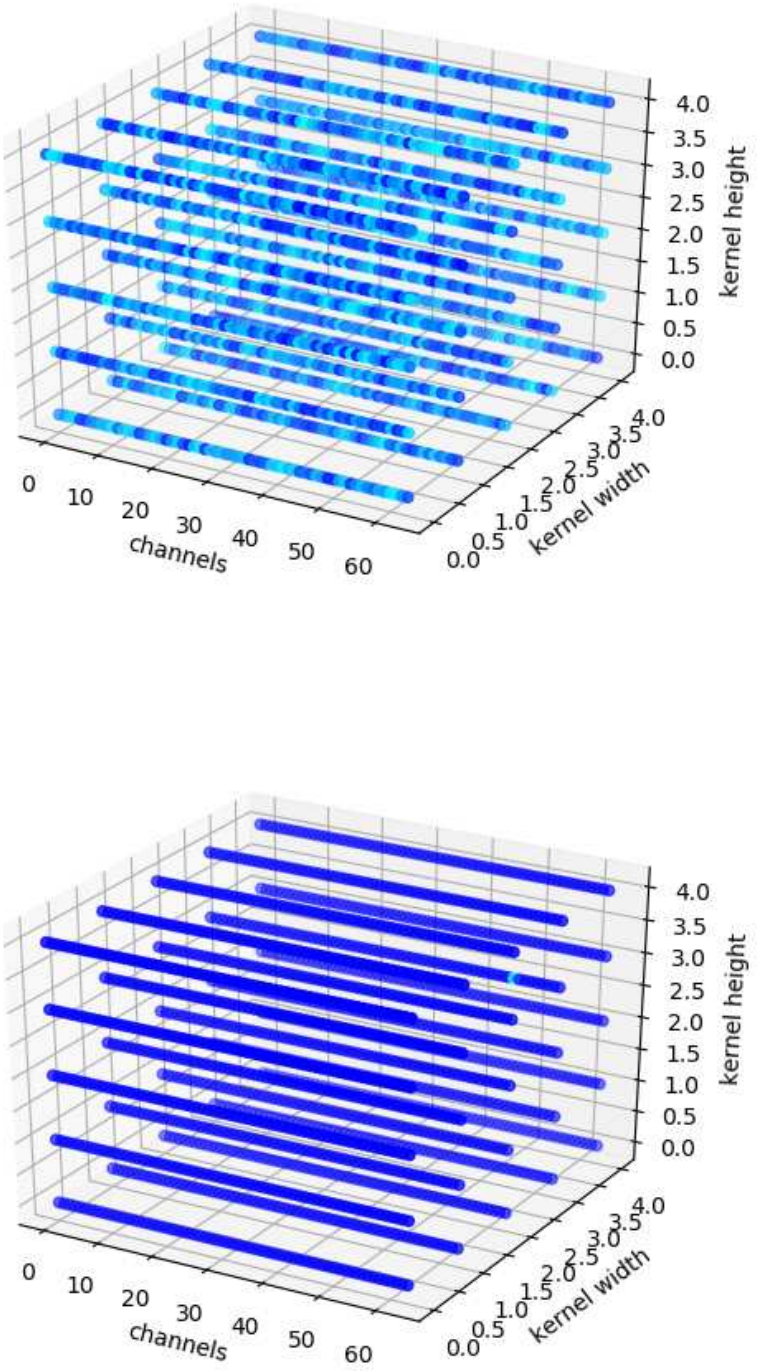

Figure 1. 3D visualization of the learned adjacency kernel for a VGG architecture. The **upper figure** shows the initialized adjacency kernel and the **lower figure** shows a learned kernel with grid-like patterns suggesting local feature-dependencies were learned (most of the kernel is inactive with few active (*light blue*) neurons).

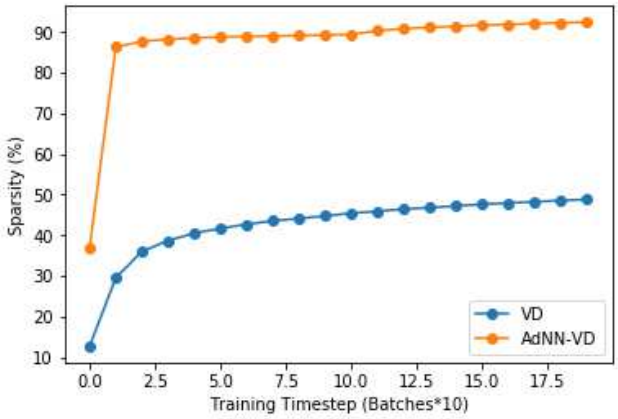

Figure 2. Total sparsity for each method over training time for CIFAR-10 for α value of 0.3

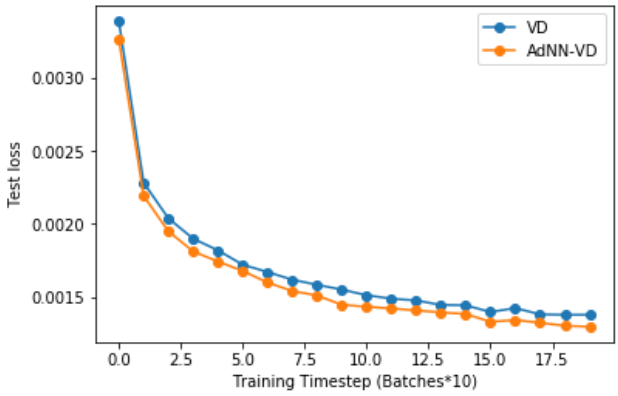

Figure 3. Test loss for each method over training time for CIFAR-10.

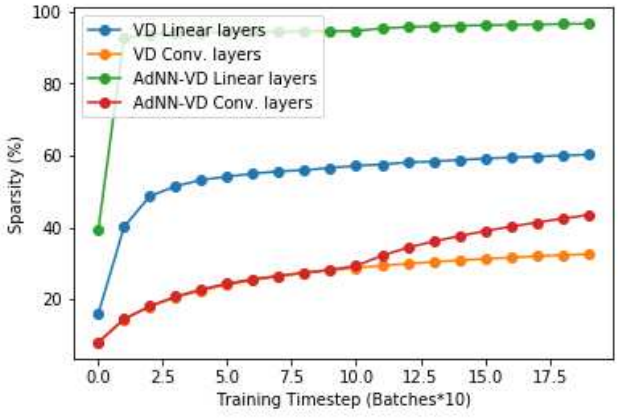

Figure 4. Sparsity for each layer type for each method over training time for CIFAR-10 for α value of 0.3

time for each method, shown in figure 2. Each network was trained until training loss converged. ANC-VD learns higher sparsity for equivalent hyperparameter settings. Yet, as shown in figure 3, its test loss does not suffer.

Table 1. CINIC-10 and ImageNet Results

Method	Dataset	Test Loss	Sparsity ($\alpha = 3.0$)
VD	CINIC-10	0.9159	49.08%
ANC-VD	CINIC-10	0.9000	92.43%
VD	ImageNet-32	6.2818	89.43%
ANC-VD	ImageNet-32	4.7892	93.63%

Additionally, we observe that the linear layers are pruned faster for ANC-VD, while keeping convolutional sparsity consistent with VD levels, as shown in figure 4. This suggests the importance of convolutional parameters is learned end-to-end by ANC more robustly than what occurs for vanilla VD.

Next, we test ResNet variants on complex classification datasets, ImageNet-32[2] and CINIC-10[3] These results are shown in Table 1. The architectures were of low capacity for the difficulty of the task and did not converge to great test accuracy, thus we show test loss only below. Given the models had difficulty converging and fitting to the training set, we present these CINIC-10 and ImageNet-32 as preliminary results only, yet show positive empirical support for the ANC method nonetheless.

Lastly, we present extensive experiments on ResNet-18 pre-trained on CIFAR-10 with follow-on magnitude pruning or VD. We compare the vanilla methods to the ANC-augmented methods for sparsity levels of 80%, 85%, 90%, 95%, and 97.5%. For VD and ANC-VD, we do fine-tuning of 100 epochs of post-training, with a sigma initialization of -30. For magnitude pruning, we do fine-tuning of 60 epochs of post-training, with uniform pruning distribution across layers. When comparing between ANC and vanilla methods, architecture capacity and optimization procedures are kept identical for as close a comparison as possible. Across all experiments, Adam optimizer was used with very low learning rates, since fine-tuning is being done.

Figure 5 shows a comparison of the ANC-VD and VD methods. The difference in test accuracy across all sparsity levels is both significant and consistent, showcasing the fact that ANC is well suited as a method applied to VD in the pretrained-fine-tuning setting as well.

Similarly, we show the same figure set-up for magnitude pruning and ANC-augmented magnitude pruning shown, in figure 6. In the case for magnitude pruning, there appears to be minor but consistent increase in its performance when augmented with ANC.

Overall, the experiments in this section showcase the applicability of ANC across a broad range of scenarios, and

[2]A harder variant of ImageNet in which the images are downsampled Chrabaszcz et al. (2017).

[3]A dataset that combines aspects of ImageNet and CIFAR-10 Darlow et al. (2018). It is shown empirically to be of similar difficulty to ImageNet.

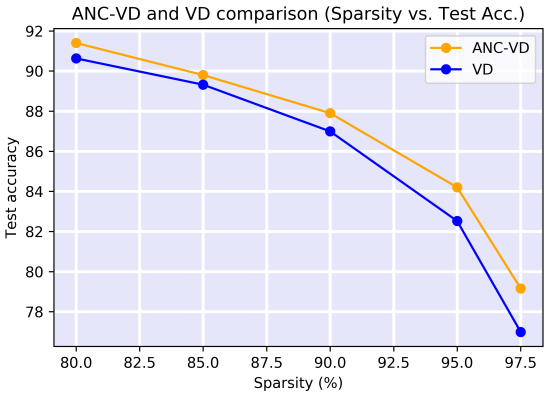

Figure 5. Sparsity versus test accuracy of ResNet18 trained on CIFAR-10. ANC-VD (orange) shows consistent, moderate gains compared to VD (blue) across all sparsity levels.

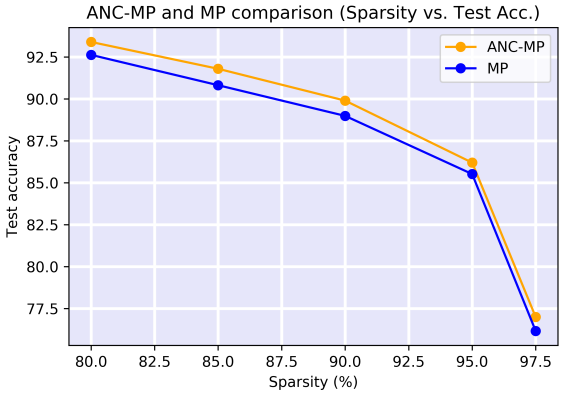

Figure 6. Sparsity versus test accuracy of ResNet18 trained on CIFAR-10. ANC-magnitude-pruning (orange) shows consistent, minor gains compared to vanilla magnitude pruning (blue).

show that ANC is helpful both when training from scratch for sparsity and when starting with a pre-trained model. ANC appears most helpful when applied to variational dropout.

5. Discussion

In conclusion, we introduce ANC, a method that is broadly applicable in the sparsity learning setting. ANC learns neuron-to-neuron connections adaptively which can confer positive benefits, which we primarily showcase with Variational Dropout, providing modest improvement gains. ANC is a general, easy to implement method that can be applied to virtually any architecture or sparsity method, showing potential as being widely used in the sparsity learning sub-field, both as a stand-alone method, and in conjunction with other methods.

References

Bejnordi, B. E., Blankevoort, T., and Welling, M. (2019). Batch-shaped channel gated networks. *arXiv preprint arXiv:1907.06627*.

Chrabaszcz, P., Loshchilov, I., and Hutter, F. (2017). A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*.

Darlow, L. N., Crowley, E. J., Antoniou, A., and Storkey, A. J. (2018). Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*.

Gale, T., Elsen, E., and Hooker, S. (2019). The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*.

Kharitonov, V., Molchanov, D., and Vetrov, D. (2018). Variational dropout via empirical bayes. *arXiv preprint arXiv:1811.00596*.

Kingma, D. P., Salimans, T., and Welling, M. (2015). Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583.

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. (2017). Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2736–2744.

Liu, Z., Sun, M., Zhou, T., Huang, G., and Darrell, T. (2018). Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*.

Louizos, C., Welling, M., and Kingma, D. P. (2017). Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*.

MacKay, D. J. (1992). Bayesian interpolation. *Neural computation*, 4(3):415–447.

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

Wen, W., Wu, C., Wang, Y., Chen, Y., and Li, H. (2016). Learning structured sparsity in deep neural networks. In *Advances in neural information processing systems*, pages 2074–2082.

Wipf, D. P. and Nagarajan, S. S. (2008). A new view of automatic relevance determination. In *Advances in neural information processing systems*, pages 1625–1632.

Zhu, M. and Gupta, S. (2017). To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*.