

# Variational Bayesian Dropout with a Hierarchical Prior

Yuhang Liu<sup>1,2</sup>, Wenyong Dong<sup>1</sup>, Lei Zhang<sup>3,2</sup>, Dong Gong<sup>2</sup>, and Qinfeng Shi<sup>2</sup>

<sup>1</sup>School of Computer Science, Wuhan University, Hubei, China

<sup>2</sup>The University of Adelaide, Australia <sup>3</sup>Inception Institute of Artificial Intelligence, UAE

<https://sites.google.com/view/yuhangliu/pro>

## Abstract

*Variational dropout (VD) is a generalization of Gaussian dropout, which aims at inferring the posterior of network weights based on a log-uniform prior on them to learn these weights as well as dropout rate simultaneously. The log-uniform prior not only interprets the regularization capacity of Gaussian dropout in network training, but also underpins the inference of such posterior. However, the log-uniform prior is an improper prior (i.e., its integral is infinite), which causes the inference of posterior to be ill-posed, thus restricting the regularization performance of VD. To address this problem, we present a new generalization of Gaussian dropout, termed variational Bayesian dropout (VBD), which turns to exploit a hierarchical prior on the network weights and infer a new joint posterior. Specifically, we implement the hierarchical prior as a zero-mean Gaussian distribution with variance sampled from a uniform hyper-prior. Then, we incorporate such a prior into inferring the joint posterior over network weights and the variance in the hierarchical prior, with which both the network training and dropout rate estimation can be cast into a joint optimization problem. More importantly, the hierarchical prior is a proper prior which enables the inference of posterior to be well-posed. In addition, we further show that the proposed VBD can be seamlessly applied to network compression. Experiments on classification and network compression demonstrate the superior performance of the proposed VBD in regularizing network training.*

## 1. Introduction

Deep neural networks have gained great success in various artificial intelligence research areas, e.g., computer vision [12, 8], natural language processing [4], etc. Nevertheless, due to the limited samples with annotation in practice, training deep neural networks with extensive parameters often suffers from over-fitting problem [35]. Dropout proves to be a practical technique to alleviate this problem, which stochastically regularizes network weights by

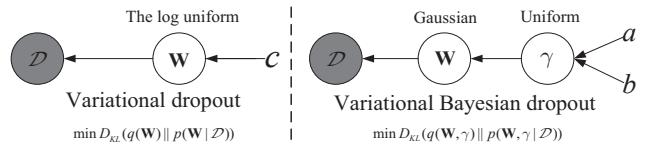


Figure 1: Variational dropout v.s. the proposed variational Bayesian dropout.

randomly enforcing multiplicative noise on input features during training [13]. Over the past several years, various dropout methods have been proposed [13, 29, 33]. Among them, *Gaussian dropout* [33] provides a general framework, which introduces the distribution of network weights into model training and thus can well approximate the conventional dropout with different types of noise, such as binary noise [13] or Gaussian noise [29]. While these methods have shown promising regularization performance in various deep network architectures [20, 28, 16, 18, 9], the reason behind such success is not clear, and their performance heavily depends on a predefined dropout rate, for which traditional grid-search based methods is a prohibitive operation for large network models.

*Variational dropout* (VD) [19] is a generalization of Gaussian dropout, which focuses on inferring the posterior of network weights based on a log-uniform prior. By doing this, VD can address these two aspects of the problems mentioned above. 1) *Bayesian Interpretation*. It has been proved [19] that VD can be consistent with Gaussian dropout for a fixed dropout rate by enforcing the log-uniform prior on network weights. This implies that incorporating Gaussian dropout into network training amounts to variational inference on the network weights, where these weights are regularized by the Kullback-Leibler (KL) divergence between the variational posterior, i.e., the distribution of network weights introduced by Gaussian dropout, and the log-uniform prior. In other words, the log-uniform prior endows Gaussian dropout with the regularization capacity. 2) *Adaptive dropout rate*. Based on the log-uniform prior, VD [19] can simultaneously learn network weights as well as dropout rate via inferring the posterior on these

weights. To sum up, the log-uniform prior as the footstone of VD underpins these two advantages above. However, recent theoretical progress in [14, 15, 26] demonstrates that the log-uniform prior is an improper prior (*i.e.*, its integral is infinite), which causes inferring the posterior of network weights to be ill-posed. Such ill-posed inference can degenerate variational inference on these weights into penalized maximum likelihood estimation [14]. Thus, the interpretation of the regularization capacity of Gaussian dropout is not in a full Bayesian way. And more importantly, the regularization capacity of VD is still limited.

To address this problem, we propose a variational Bayesian dropout (VBD) framework, which is a new generalization of Gaussian dropout. A visual comparison between the proposed VBD and VD can be seen in Figure 1. In VBD, we assume network weights to come from a two-level hierarchical prior. Instead of only inferring the posterior over network weights, we propose to infer the joint posterior over both network weights and their hyper-parameters defined in their first-level prior. Through implementing the hierarchical prior as a zero-mean Gaussian distribution with variance sampled from a uniform distribution, we can theoretically prove that the proposed VBD can be consistent with Gaussian dropout [33] for a fixed dropout rate as VD. Thus, VBD also can interpret the regularization capacity of Gaussian dropout. In contrast to the improper log-uniform prior, the proposed hierarchical prior is a proper prior, which enables the inference of posterior in VBD well-posed. This not only leads to a full Bayesian justification for Gaussian dropout, but also improves the regularization capacity obviously. In addition, we further find that the proposed VBD can be seamlessly applied to neural network compression as [25, 26]. Experimental results on classification as well as network compression tasks show the effectiveness of VBD in handling over-fitting.

## 2. Related work

**Dropout.** Dropout plays an important role in improving the generalization capacity of deep neural networks. At first, dropout is employed to randomly drop input features with Bernoulli distribution during training to prevent feature co-adaptation [13]. This amounts to training an exponential number of different networks with shared parameters. In the test phase, the prediction is determined by averaging the outputs of all these different networks. The idea of dropout is then generalized by multiplying the input features with random noise drawn from other distributions, *e.g.*, Gaussian [29]. While these early methods have shown effectiveness in some cases, repeatedly dropping a random subset of input features makes training a network much slower. To address this problem, Gaussian dropout [33] proposes to sample the output features from a Gaussian distribution instead of input features in dropout training and shows vir-

tually identical regularization performance but faster convergence. This is inspired by the observation that enforcing multiplicative noise on input features, whatever the noise is generated from a Bernoulli distribution or a Gaussian, making use of the central limit theorem, makes the corresponding outputs to be approximately Gaussian [33]. However, these conventional dropout methods fail to clarify the intrinsic principle for their regularization capacity. In addition, their performance depends a lot on the pre-defined dropout rate. In contrast, the proposed VBD can provide a Bayesian interpretation for dropout as well as automatically estimating the dropout rate.

**Variational Dropout.** VD is a generalization of Gaussian dropout, which can interpret the regularization capacity of dropout as well as automatically estimating the dropout rate via inferring the posterior of network weights. For example, literature in [19] proves that training network with variational dropout framework implicitly imposes the log-uniform prior on weights for preventing over-fitting. Since the dropout rate can be automatically determined, some works of literature [25, 26] further apply VD to compress neural networks. However, the log-uniform prior is an improper prior which causes the inference of posterior over network weights in VD to be ill-posed [15, 26], thus limiting its performance in preventing over-fitting. In this study, the proposed VBD imposes a proper hierarchical prior on network weights, which induces a well-posed Bayesian inference over network weights and thus improves the regularization capacity.

**Concrete Dropout and Adversarial Dropout.** In addition, recent works have proposed another two dropout methods, *e.g.*, concrete dropout [6] and adversarial dropout [27]. They are different from the proposed VBD. Specifically, concrete dropout provides Bayesian generalization for dropout with Bernoulli distribution [13], while the proposed VBD provides Bayesian generalization for Gaussian dropout [33]. Besides, adversarial dropout [27] proposes to handle over-fitting by training the network in an adversarial way. In contrast, the proposed method focuses on introducing hierarchical prior on network weights to regularize the network training.

## 3. Preliminaries

Consider a supervised learning problem on a dataset  $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$  of observation-label pairs. We train a fully connected neural network with  $L$  hidden layers. For each layer, we have:

$$\mathbf{B} = \mathbf{A}\theta, \quad (1)$$

where  $\mathbf{A}$  denotes the  $M \times K$  matrix of input features for current minibatch,  $\theta$  is the  $K \times D$  weight matrix,  $\mathbf{B}$  is the  $M \times D$  output matrix before activation function.

### 3.1. Gaussian Dropout

To prevent over-fitting, dropout applies multiplicative noise on the input of each layer of neural networks during training as follows:

$$\mathbf{B} = (\mathbf{A} \circ \xi) \theta, \quad (2)$$

where  $\xi$  is the  $M \times K$  noisy matrix, and  $\circ$  denotes the element-wise (Hadamard) product. In conventional dropout methods, the elements of the noise  $\xi$  are either sampled from a Bernoulli distribution with probability  $1 - p$  of being 1, with the *dropout rate*  $p$  [13], or sampled from a Gaussian distribution with mean 1 and variance  $\alpha = p/(1 - p)$  [29]. Regardless which strategy of the above two is used, according to the central limit theorem and equation Eq. (2), one can directly produce  $B_{m,d}$  by sampling from the following Gaussian distribution:

$$q(B_{m,d}|\mathbf{A}) = \mathcal{N}(B_{m,d}|\mu_{m,d}, \delta_{m,d}^2), \quad (3)$$

where mean  $\mu_{m,d} = \sum_{k=1}^K A_{m,k} \theta_{k,d}$  and variance  $\delta_{m,d}^2 = \alpha \sum_{k=1}^K A_{m,k}^2 \theta_{k,d}^2$ . Here  $A_{m,k}$  denotes an element in  $\mathbf{A}$ . This means  $q(\mathbf{W})$  can be factorized as follows:

$$q(\mathbf{W}) = \prod_{k=1}^K \prod_{d=1}^D q(W_{k,d}) = \prod_{k=1}^K \prod_{d=1}^D \mathcal{N}(W_{k,d}|\theta_{k,d}, \alpha \theta_{k,d}^2), \quad (4)$$

where each element  $W_{k,d}$  in  $\mathbf{W}$  can be sampled from  $q(\mathbf{W}_{k,d})$  in Eq. (4). Finally, the objective function for network training with Gaussian dropout becomes:

$$\max_{\theta} L_{\mathcal{D}}(\theta) \simeq \max_{\theta} \sum_{i=1}^N \mathbb{E}_{q(\mathbf{W})} \log p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{W}). \quad (5)$$

The viewpoint in Eq. (5) provides a opportunity to bridge the gap between Bayesian inference and dropout, if we use Eq. (4) as an approximate posterior distribution for a network model with a special prior on the weights. The challenge in this gap is what is the special prior.

### 3.2. Variational Dropout

VD uses  $q(\mathbf{W})$  in Eq. (4) as a variational posterior to approximate the true posterior  $p(\mathbf{W}|\mathcal{D})$  in terms of minimal KL divergence,

$$\min_{\theta, \alpha} D_{KL}(q(\mathbf{W})||p(\mathbf{W}|\mathcal{D})). \quad (6)$$

Here  $D_{KL}(\cdot)$  denotes the KL divergence. Given  $\mathbf{W}$  from a prior  $p(\mathbf{W})$ , according to the Bayesian rule, i.e.,  $p(\mathbf{W}|\mathcal{D}) \propto p(\mathcal{D}|\mathbf{W})p(\mathbf{W})$ , minimizing the KL divergence in Eq. (6) is equivalent to maximizing the variational lower bound of the marginal likelihood of data as:

$$\max_{\theta, \alpha} L_{\mathcal{D}}(\theta, \alpha) - D_{KL}(q(\mathbf{W})||p(\mathbf{W})), \quad (7)$$

where  $L_{\mathcal{D}}(\theta, \alpha)$  with fixed parameter  $\alpha$  is the same as one in Eq. (5) and known as the expected log-likelihood term.

Firstly, in VD, the dropout rate  $\alpha$  can be automatically determined by data characteristics. Secondly, VD can provide a Bayesian interpretation for the success of dropout in preventing over-fitting. To clarify this, VD requires that the optimization of  $\theta$  in Eq. (7) is consistent with that in Gaussian dropout in Eq. (5), i.e., maximizing the expected log-likelihood. To this end, the prior  $p(\mathbf{W})$  has to be such that  $D_{KL}(q(\mathbf{W})||p(\mathbf{W}))$  in Eq. (7) does not depend on weight parameters  $\theta$ . With such a requirement, we have the following proposition.

**Proposition 1** ([19]). *The only prior in VD, which enables  $D_{KL}(q(\mathbf{W})||p(\mathbf{W}))$  not depending on weight parameters  $\theta$ , is the log-uniform prior:*

$$p(|W_{k,d}|) = \frac{c}{|W_{k,d}|} \Leftrightarrow p(\log(|W_{k,d}|)) = c. \quad (8)$$

The above discussion demonstrates that training network with VD implicitly imposes the log-uniform prior on weights. With such a prior, the KL term in Eq. (7) is able to regularize the number of significant digits stored for the weights  $\mathbf{W}$  in the floating-point format, thus being able to mitigate over-fitting at some extent. However, the log-uniform prior is an improper prior, which reaches ill-posed variational inference, e.g., the KL divergence between the variational posterior in Eq. (4) and the log uniform prior in Eq. (8) is infinite. Although leveraging truncated techniques relieves this problem at some extent [19, 26], pathological behaviour still remains [14], e.g., the resultant infinite KL divergence theoretically degenerate VD into a maximum likelihood estimation that fails to avoid over-fitting. More details for the theoretical justification can be found in [15, 26, 14]. Therefore, the performance of VD in preventing over-fitting need to be further improved.

In this section, we will introduce the details of the proposed VBD. In the following, we first introduce the proposed VBD framework. Then, we show a specifically designed hierarchical prior in VBD framework and prove that VBD with this prior can be consistent with Gaussian dropout for a fixed dropout rate.

### 3.3. Variational Bayesian Dropout

In contrast to a one-level prior in VD, we turn to propose a two-level hierarchical prior  $p(\mathbf{W}, \gamma) = p(\mathbf{W}|\gamma)p(\gamma)$ . This brings two aspects of advantages. Firstly, two kinds of very simple distributions in hierarchical structure can produce a much more complicated distribution, e.g., a hierarchical sparse prior [36], a zero-mean Gaussian distribution with variance depicted by Gamma distribution [37], and the super-Gaussian scale mixture model [22]. Thus the two-level structure increase the possible solution spaces for the proper and feasible prior to interpret Gaussian dropout.

Secondly, the hierarchical structure enables the two-level prior separable in the involved Bayesian inference and thus is possible to simplify the Bayesian inference or makes the intractable inference tractable, which will be further clarified in the following subsections.

Similar to VD, the proposed VBD aims at optimizing a variational posterior to approximate the true posterior [17, 23]. Unlike that VD only considers the posterior of network weights, we propose to model the joint posterior of both the network weights (e.g.,  $\mathbf{W}$ ) and the hyper-parameters (e.g.,  $\gamma$ ) in their prior as illustrated in Figure 1. We thus arrive the objective of Bayesian inference in the proposed VBD:

$$\min_{\theta, \alpha, \gamma} D_{KL}(q(\mathbf{W}, \gamma) \parallel p(\mathbf{W}, \gamma | \mathcal{D})), \quad (9)$$

where  $q(\mathbf{W}, \gamma)$  denotes a corresponding variational joint posterior for  $p(\mathbf{W}, \gamma | \mathcal{D})$ . Note that when the hyper-parameter  $\gamma$  is fixed, the proposed VBD will reduce to VD. Thus, the proposed VBD is a more general version of VD. According to variational Bayesian inference technique [17], we use the variational posterior  $q(\mathbf{W}, \gamma) = q(\mathbf{W})q(\gamma)$  to approximate the true posterior  $p(\mathbf{W}, \gamma | \mathcal{D})$ , and then the objective in Eq. (9) can be reformulated as the variational lower bound of the marginal likelihood of data as:

$$\max_{\alpha, \theta, \gamma} L_{\mathcal{D}}(\alpha, \theta) - D_{KL}(q(\mathbf{W}) \parallel p(\mathbf{W} | \gamma)) - D_{KL}(q(\gamma) \parallel p(\gamma)), \quad (10)$$

where  $L_{\mathcal{D}}$  is the expected log-likelihood term in Eq. (7). Derivations can be found in supplementary material. For the proposed VBD, the key is to exploit a proper hierarchical prior  $p(\mathbf{W}, \gamma)$  for supporting Gaussian dropout. In the following, we will provide such prior and discuss its advantages.

### 3.4. The Proposed Hierarchical Prior

Inspired by the hierarchical prior in sparse Bayesian learning [30], we assume the network weights  $\mathbf{W}$  come from a zero-mean Gaussian distribution. Then, a uniform hyper-prior is imposed on the variance of the Gaussian distribution to adjust the shape of the ultimate prior. When each element  $W_{k,d}$  in  $\mathbf{W}$  is independent identically distributed, the proposed two-level hierarchical prior  $p(\mathbf{W}, \gamma)$  is formulated as:

$$\begin{aligned} p(\mathbf{W} | \gamma) &= \prod_{k=1}^K \prod_{d=1}^D p(W_{k,d} | \gamma_{k,d}) = \prod_{k=1}^K \prod_{d=1}^D \mathcal{N}(W_{k,d} | 0, \gamma_{k,d}), \\ p(\gamma) &= \prod_{k=1}^K \prod_{d=1}^D p(\gamma_{k,d}) = \prod_{k=1}^K \prod_{d=1}^D \mathcal{U}(\gamma_{k,d} | a, b), \end{aligned} \quad (11)$$

where  $\mathcal{U}(\gamma_{k,d} | a, b)$  denotes an uniform distribution with range  $[a, b]$ . By imbedding this prior into the proposed VBD framework in Eq. (10), we give the following theoretical result.

**Proposition 2.** *With the prior  $p(\gamma)$  in Eq. (11), we employ mean-field variational approximation, viz.,  $q(\gamma) = \prod_{k=1}^K \prod_{d=1}^D q(\gamma_{k,d})$ , and assume that  $q(\gamma_{k,d})$  comes from a delta distribution. Then the proposed VBD framework in Eq. (10) reduces to*

$$\max_{\alpha, \theta, \gamma} L_{\mathcal{D}}(\alpha, \theta) - D_{KL}(q(\mathbf{W}) \parallel p(\mathbf{W} | \gamma)). \quad (12)$$

Given the prior in Eq. (11) and the variational posterior  $q(\gamma_{k,d})$ ,  $D_{KL}(q(\gamma) \parallel p(\gamma))$  in Eq. (10) will collapse to a constant and thus can be neglected in optimization. Similar trick can be found in [2, 1]. Specifically, to simplify the representation, we assume  $\gamma_{k,d}$  as a one-dimensional scalar. As defined in Eq. (11), we have  $p(\gamma_{k,d}) = 1/(b - a)$ . Note that the delta distribution  $q(\gamma_{k,d})$  either lies in or out of  $[a, b]$ , e.g.,  $\delta(\gamma_{k,d} - \gamma'_{k,d})$  and  $\gamma'_{k,d}$  lies in or out of  $[a, b]$ . If  $q(\gamma_{k,d})$  is out of  $[a, b]$ , there is  $D_{KL}(q(\gamma_{k,d}) \parallel p(\gamma_{k,d})) = +\infty$ . To avoid this case,  $[a, b]$  is generally regarded as a large enough interval [2, 1]. As a result, we arrive  $D_{KL}(q(\gamma_{k,d}) \parallel p(\gamma_{k,d})) = \log(b - a)$ , which is independent to the unknown variables  $\alpha, \theta$  and  $\gamma$  and thus can be neglected. Therefore, we do not need to set specific values for the hyper-parameters  $a$  and  $b$  in practice. The detailed proof can be found in supplementary material.

According to [19], the key property of the log-uniform prior is to enable the KL divergence  $D_{KL}(q(\mathbf{W}) \parallel p(\mathbf{W}))$  in Eq. (7) not depending on weight parameters  $\theta$  as mentioned in **Proposition 1**. With this property, learning  $\theta$  in VD will be consistent with that in conventional Gaussian dropout for a fixed dropout rate  $\alpha$ . In the following, we will demonstrate that the proposed hierarchical prior Eq. (11) also shows a similar property in VBD framework Eq. (12). To this end, we give the following theoretical result.

**Proposition 3.** *Given the objective Eq. (12), together with the prior Eq. (11) and the variational posterior Eq. (4),  $D_{KL}(q(\mathbf{W}) \parallel p(\mathbf{W} | \gamma)) = \sum_{k=1}^K \sum_{d=1}^D 0.5 \log(1 + \alpha^{-1})$ , which does not depend on weight parameters  $\theta$ .*

*Proof.* Since the variational posterior in Eq. (4) and the prior  $p(\mathbf{W} | \gamma)$  in Eq. (11) are fully factorized, the KL-divergence  $D_{KL}(q(\mathbf{W}) \parallel p(\mathbf{W} | \gamma))$  in (12) can be decomposed into a sum as:

$$\sum_{k=1}^K \sum_{d=1}^D D_{KL}(q(W_{k,d}) \parallel p(W_{k,d} | \gamma_{k,d})). \quad (13)$$

Since both the prior  $q(W_{k,d})$  and the posterior  $p(W_{k,d} | \gamma_{k,d})$  follow Gaussian distributions, the KL divergence  $D_{KL}(q(W_{k,d}) \parallel p(W_{k,d} | \gamma))$  in Eq. (13) can be calculated as:

$$\begin{aligned} D_{KL}(q(W_{k,d}) \parallel p(W_{k,d} | \gamma)) &= \\ 0.5 \log\left(\frac{\gamma_{k,d}}{\alpha \theta_{k,d}^2}\right) + \frac{\alpha \theta_{k,d}^2 + \theta_{k,d}^2}{2\gamma_{k,d}} - 0.5. \end{aligned} \quad (14)$$

By introducing Eq. (14) into Eq. (12), we arrive:

$$\max_{\alpha, \theta, \gamma} L_D(\alpha, \theta) - \sum_{k=1}^K \sum_{d=1}^D 0.5 \log\left(\frac{\gamma_{k,d}}{\alpha \theta_{k,d}^2}\right) + \frac{\alpha \theta_{k,d}^2 + \theta_{k,d}^2}{2\gamma_{k,d}}. \quad (15)$$

To find the optimal  $\gamma_{k,d}$ , referred to as  $\gamma_{k,d}^*$ , by setting the partial differential of Eq. (15) with respect to  $\gamma_{k,d}$  to zero, we have:

$$\gamma_{k,d}^* = \alpha \theta_{k,d}^2 + \theta_{k,d}^2. \quad (16)$$

Replacing  $\gamma_{k,d}$  in Eq. (14) by Eq. (16) completes the proof.  $\square$

In summary, with **Propositions 2** and **3**, the final objective for the proposed VBD with the hierarchical prior can be given as:

$$\max_{\alpha, \theta} L_D(\alpha, \theta) - \sum_{k=1}^K \sum_{d=1}^D 0.5 \log(1 + \alpha^{-1}). \quad (17)$$

Built on this loss function, we can see that: 1) The proposed hierarchical prior also meets the requirement such that  $D_{KL}(q(\mathbf{W})||p(\mathbf{W}|\gamma))$  does not depend on weight parameters  $\theta$ . In other words, the proposed VBD with the hierarchical prior is consistent with Gaussian dropout when  $\alpha$  is fixed. Hence, the proposed VBD can give Bayesian interpretation for Gaussian dropout. 2) The dropout rate parameter  $\alpha$  also can be automatically learned as VD.

Note that we apply a uniform prior on  $\gamma$  to be able to update  $\gamma$ . If  $\gamma$  was treated as the variance hyper-parameter of the Gaussian prior, we could not update it in the proposed framework. This is because the prior could not see any data. Then, if we cannot update  $\gamma$ , we cannot produce Eq. (16), which means that we could not produce **Propositions 3**. To allow  $\gamma$  to be updated, we impose a uniform prior on it and utilize delta variational posterior leading to **Propositions 2**. This is also why many variational approximation methods (e.g., [2, 1]) employ priors on hyper-parameters.

More importantly, the improper log-uniform prior in traditional VD induces ill-posed Bayesian inference, since it leads to infinite  $D_{KL}(q(\mathbf{W})||p(\mathbf{W}))$  as mentioned in [14, 15, 26]. By contrast, the proposed hierarchical prior in VBD gives well-posed Bayesian inference, since it produces reasonable and tractable  $D_{KL}(q(\mathbf{W})||p(\mathbf{W}|\gamma))$  as shown in **Proposition 3**. Hence, the proposed hierarchical prior in VBD shows obvious superiority over the log-uniform prior in VD, and we argue that the proposed VBD framework with the hierarchical prior provides a full Bayesian interpretation for the success of Gaussian dropout in preventing over-fitting.

According to the discussion above, we can conclude as follows. 1) The proposed VBD is a more general VBD. VD focuses on incorporating one-level priors, while VBD can contain two-level priors as shown in Figure 1. 2) VD

claims that the improper log-uniform prior interprets Gaussian dropout, which fails to give full Bayesian interpretation for Gaussian dropout and suffers limited regularization capacity, while VBD proposes a proper prior to handle those drawbacks. In the following, we will show that these two main differences further lead to the apparent advantage of VBD in network compression compared with VD.

## 4. Extension to Neural Network Compression

Since VD can adaptively learn the dropout rate  $p$  (or  $\alpha$ ) from training dataset, it can be utilized for neural networks compression [25, 26]. Inspired by this, in this section we turn to exploit the ability to apply the proposed VBD to neural networks compression under the frameworks proposed in [25, 26]. In addition, we will also discuss the advantage of the proposed framework on networks compression.

### 4.1. Compressing Weights

We first extend the proposed VBD to compressing weights under the framework in [25]. Further details about the framework can be found in that paper. To this end, the proposed hierarchical prior Eq. (11) is used to model weights in a neural network. For convenience, we replace the original  $\alpha$  in Eq. (4) to learn specific  $\alpha$  for each weight, in which **Proposition 3** holds. The distribution Eq. (4) as a variational posterior,  $\mathcal{N}(\theta_{k,d}, \alpha_{k,d}\theta_{k,d}^2)$ , is used to approximate the true posterior. In this way, to learn  $\theta_{k,d}$  and  $\alpha_{k,d}$ , the objective function in Eq. (17) is rewritten as:

$$\max_{\alpha, \theta} L_D(\alpha, \theta) - \sum_{k=1}^K \sum_{d=1}^D 0.5 \log(1 + \alpha_{k,d}^{-1}). \quad (18)$$

Furthermore, since the natural gradient of  $\theta_{k,d}$  faces with high variance, we follow the re-parameterization trick in [25], viz,  $\sigma_{k,d}^2 = \alpha_{k,d}\theta_{k,d}^2$ .

The difference between the proposed method and the method in [25] is on the regularization term  $-D_{KL}(q(\mathbf{W})||p(\mathbf{W}))$ . It is equivalent to  $\sum_{k=1}^K \sum_{d=1}^D -0.5 \log(1 + \alpha_{k,d}^{-1})$  in the proposed VBD, while it is  $\sum_{k=1}^K \sum_{d=1}^D k_1 S(k_2 + k_3 \log(\alpha_{k,d})) - 0.5 \log(1 + \alpha_{k,d}^{-1})$  in the method proposed in [25], where  $k_1$ ,  $k_2$  and  $k_3$  are constant, and  $S(\cdot)$  denotes the sigmoid function. The term  $-0.5 \log(1 + \alpha_{k,d}^{-1})$  in the latter is heuristically designed in [25] to model the behaviour that the negative KL-divergence goes to a constant as  $\log \alpha_{k,d}$  goes to minus infinity. In contrast, the term  $-0.5 \log(1 + \alpha_{k,d}^{-1})$  in the proposed VBD is naturally derived from Bayesian inference as mentioned above. Further discussion on this term will be given in section 4.3.

## 4.2. Structured Compressing

Although the method in [25] can be employed to compress weights in a neural network, it fails to accelerate neural networks in the testing phase, since resultant compression is unstructured. Recently, structured Bayesian pruning in [26] employs VD to remove neurons and/or convolutional channels in convolutional neural networks for structured compression, resulting in satisfactory performance. Similar to VD, the proposed VBD also can be employed for structured pruning by constructing a dropout-like layer under the framework in [26].

Specifically, we construct a single dropout-like layer with an input matrix  $f(\mathbf{B})$  as follows:

$$\mathbf{B}' = f(\mathbf{B}) \circ \mathbf{W}', \quad (19)$$

where  $\mathbf{W}'$  denotes the dropout noise and  $f(\cdot)$  denotes the activation function. The output of this layer  $\mathbf{B}'$  is of the same size as the input  $\mathbf{B}$ , and would serve as an input matrix for the following layer. Similar to that in the previous section, we enforce the proposed hierarchical prior Eq. (11) on  $\mathbf{W}'$ , and imposes the variational posterior,  $\mathcal{N}(\theta_{m,d}, \alpha_{m,d}\theta_{m,d}^2)$  (For convenience, we re-utilize the same symbol  $\alpha_{m,d}$  and  $\theta_{m,d}$  that is originally used to model weights in the previous section, however, they are served to  $\mathbf{W}'$  in this section). Again, the objective function Eq. (18) is used for learning  $\theta_{m,d}$  and  $\alpha_{m,d}$ , and the re-parameterization [25] is adopted.

## 4.3. Analysis

In these two kinds of compression schemes above, effective compression depends on high dropout rate, e.g.,  $\alpha_{k,d} \rightarrow \infty$  or  $\alpha_{k,d}^{-1} \rightarrow 0$ , which corresponds to a binary dropout rate that approaches  $p = 1$ . This effectively means that the corresponding weight or neuron is always ignored and can be removed [25]. In this subsection, we will show that the proposed variational Bayesian dropout explicitly imposes a sparse regularization for optimizing  $\alpha_{k,d}^{-1}$ , and thus is able to effectively compress the deep neural networks. To this end, we firstly rewrite the objective Eq. (18) as:

$$\min_{\alpha, \theta} \sum_{k=1}^K \sum_{d=1}^D 0.5 \log(1 + \alpha_{k,d}^{-1}) - L_{\mathcal{D}}(\alpha, \theta). \quad (20)$$

The expected log-likelihood term  $L_{\mathcal{D}}$  can be viewed as the data fit-term for  $\alpha_{k,d}^{-1}$  and the remainder derived from KL divergence works as the regularization term. For such a regularization term, we have the following theoretical results.

**Proposition 4.** *The regularization term  $0.5 \log(1 + \alpha_{k,d}^{-1})$  in Eq. (20) is a concave, non-decreasing function on the domain  $[0, +\infty)$ , with respect to  $\alpha_{k,d}^{-1}$ .*

According to [3], introducing such a regularization term  $0.5 \log(1 + \alpha_{k,d}^{-1})$  into the objective is beneficial to promote

the sparsity of the solution. Therefore, with optimizing  $\alpha_{k,d}^{-1}$ , we can obtain sparse  $\alpha_{k,d}^{-1}$ . Note that this regularization term coincides with that in [5] which is motivated by information bottleneck principle. In contrast, the regularization term in this study stems from variational Bayesian inference. Besides, this sparsity-promoting regularization comes from a result of the particular variational approximation, which is different from previous methods with sparse priors to compress network, e.g., [24, 7].

## 5. Experiments

In this section, we conduct experiments on classification task to demonstrate the effectiveness of the proposed variational Bayesian dropout in preventing over-fitting. Then, we further evaluate its performance in neural network compression including weight compression and structured compression. Note that neural network compression can also imply the ability of preventing over-fitting in term of the final test error.

### 5.1. Classification

**MNIST Dataset** Following the settings in [19], we first take the hand-written digit classification task on MNIST dataset as a standard benchmark to evaluate the performance of dropout methods in preventing over-fitting. On this task, we compare the proposed variational Bayesian dropout with other five existing dropout methods, namely no dropout, standard dropout with Bernoulli noise [13], dropout with Gaussian noise [29], Gaussian dropout [33] and VD [19], and concrete dropout [6] which is able to learn adaptive dropout rate. We follow the network architecture in [29], which adopt a fully connected neural network consisting of 3 hidden layers with different number of units and rectified linear units. For experimental setting, all networks are trained for 50 epochs. More details about the network architecture can be found in supplementary material.

The number of units	100	340	580	820	1060
No Dropout	1.80	1.77	1.78	1.78	1.71
Dropout, Bernoulli	1.69	1.68	1.67	1.67	1.63
Dropout, Gaussian	1.70	1.66	1.68	1.69	1.62
Gaussian Dropout	1.66	1.64	1.71	1.65	1.64
Concrete Dropout	2.39	1.61	1.55	1.54	1.51
VD	1.69	1.62	1.67	1.62	1.63
Ours	<b>1.56</b>	<b>1.53</b>	<b>1.53</b>	<b>1.52</b>	<b>1.45</b>

Table 1: Test error (%) on the MNIST dataset.

Table 1 shows the test error for all methods with various choices of the number of units per layer. We observe that although VD can adaptively learn dropout rate during training, it only obtains slightly better performance compared with conventional dropout methods with fixed dropout rate,

including dropout with Bernoulli noise and dropout with Gaussian noise. This is because the Bayesian inference in VD with the improper log-uniform prior is ill-posed, which hence ultimately restricts the capacity in preventing overfitting as discussed before. Conversely, the proposed VBD with adaptive dropout rate gains impressive performance, which is better than that of the standard dropout as well as VD. This profits from that the proposed hierarchical prior is a proper prior and thus it can appropriately regularize the network weights in the Bayesian inference. Besides, the proposed VBD is superior to concrete dropout.

**CIFAR-10 Dataset** We further compare the proposed method with dropout with Bernoulli noise [13], dropout with Gaussian noise [29], VD [19], concrete dropout [6] on CIFAR-10 dataset. We also compare with adversarial dropout recently proposed in [27]. In this case, we follow the network architecture with different *scale* in [19] for experimental setting, all networks are trained for 100 epochs. More details for the network architecture can be found in supplementary material.

<i>scale</i>	1	1.5	2
No Dropout	48.34	48.13	47.56
Dropout, Bernoulli	45.55	43.38	42.69
Dropout, Gaussian	46.52	43.50	42.79
Adversarial Dropout	45.50	42.35	42.50
Concrete Dropout	43.47	42.68	42.19
VD	44.12	42.99	42.60
Ours	<b>39.50</b>	<b>38.88</b>	<b>38.52</b>

Table 2: Test error (%) on the CIFAR-10 dataset.

Figure 2 shows the test error for the all methods with different *scale*. We can see that concrete dropout only performs on par with traditional dropout methods and VD. In addition, we found that due to its negative effect of the improper log-uniform prior, VD only provides comparable results to those methods with fixed dropout rate. In contrast, profiting from the proper hierarchical prior, the proposed variational Bayesian dropout performs impressively well in preventing over-fitting. For example, when *scale* = 2, compared with no dropout method, the proposed method reduces the test error by 9.04%. The improvement is even up to 4.08% when compared with VD.

**SVHN Dataset** In this case, we follow the network architecture with *scale* = 1 used in experiments on **CIFAR-10 Dataset**, and all networks are trained for 100 epochs. Table 3 shows the results of test error (%) in SVHN dataset. Since traditional methods for setting dropout rate, such as Grid-search based methods, are computationally expensive. We set the dropout rate to be 0.5 for all layers of the network in our experiments for simplicity. Under this simple setting, traditional dropouts, e.g., Dropout with Bernoulli

Methods	Error (%)
No dropout	22.01
Dropout, Bernoulli	20.31
Dropout, Gaussian	20.22
Concrete Dropout	18.95
Adversarial Dropout	18.66
VD	19.74
Ours	<b>17.46</b>

Table 3: Test error on the SVHN dataset.

noise and Dropout with Gaussian noise, are slightly superior to no dropout. Again, due to the improper log-uniform prior, VD only provides comparable results to traditional dropout with fixed dropout rate. Conversely, compared with these fixed dropout rate based methods, the proposed method with adaptive dropout rate gets the best test error, 17.46%.

## 5.2. Network compression

**Compressing Weights** In this part, we turn to evaluate the effectiveness of the proposed method in weight compression in neural networks for classification on the MNIST dataset. Here we adopt two kinds of basic neural networks, the fully-connected LeNet-300-100 [21] and a convolutional LeNet-5-Caffe<sup>1</sup>. We compare the proposed method with four network compression methods, including Pruning [11], Dynamic Network Surgery (DNS) [10], Soft Weight Sharing (SWS) [32] and VD [25]. In the experiments, we strictly follow the settings in [25].

Table 4 shows the results of compressing weights in LeNet-300-100. Compared with traditional VD, with the similar sparsity  $\frac{|W|}{|W_{\neq 0}|} = 66$ , the proposed method gets 1.67% test error that is better than the result of traditional VD 1.94%. Further, although traditional VD has better results in compression ratio  $\frac{|W|}{|W_{\neq 0}|} = 68$ , compared with Pruning, DNS and SWS, it also reports higher test error 1.94% due to its limited ability to avoid over-fitting. On the contrary, since the proposed VBD provides a real Bayesian interpretation for dropout, it can effectively prevent overfitting and gains the better test error 1.76% as well as higher compression ratio  $\frac{|W|}{|W_{\neq 0}|} = 81$ .

Table 4 also shows the results of compressing weights in LeNet-5-Caffe. We can see that compared with Pruning, DNS and SWS, traditional VD reports better compression ratio  $\frac{|W|}{|W_{\neq 0}|} = 280$  and test error 0.75%. Further, compared with the all methods, the proposed method obtains the best compression ratio  $\frac{|W|}{|W_{\neq 0}|}$ , without the loss of test error.

**Structured Compressing** We here test the performances of the proposed method on structured compressing for neu-

<sup>1</sup><https://github.com/BVLC/caffe/tree/master/examples/mnist>

		LeNet-300-100			LeNet-5-Caffe		
Methods	Error %	Sparsity per Layer %	$\frac{ \mathbf{W} }{ \mathbf{W}_{\neq 0} }$	Error %	Sparsity per Layer %	$\frac{ \mathbf{W} }{ \mathbf{W}_{\neq 0} }$	
Original	1.64		1	0.8		1	
Pruning	<b>1.59</b>	92.0-91.0-74.0	12	0.77	34-88-92.0-81	12	
DNS	1.99	98.2-98.2- <b>94.5</b>	56	0.91	<b>86</b> -97-99.3-96	111	
SWS	1.94		23	0.97		200	
VD	1.94	<b>98.9</b> -97.2-62.0	68	0.75	<b>67</b> - <b>98</b> - <b>99.8</b> -95	280	
Ours (low test error)	1.67	98.7-97.4-87.4	66	<b>0.74</b>	69- <b>98</b> -99.7-95	198	
Ours (high sparsity)	1.76	<b>98.9</b> - <b>98.1</b> -90.9	<b>81</b>	0.81	<b>65</b> - <b>98</b> - <b>99.8</b> - <b>97</b>	<b>290</b>	

Table 4: Compressing weights in LeNet-300-100 and LeNet-5-Caffe.  $|\mathbf{W}|$  and  $|\mathbf{W}_{\neq 0}|$  denote the number of weights, the number of weights with nonzero value, respectively. The sparsity per layer is computed by the rate between the number of weights with zero value and the number of weights in each layer.

ral networks. The used architecture of neural networks is a fully-connected LeNet-500-300 and a convolutional LeNet-5-Caffe. We compare VD [25], SSL [34] and SBP [26].

Table 5: Compressing neurons.

Models	Methods	Error %	Neurons per Layer
LeNet-500-300	Original	1.54	784 - 500 - 300 - 10
	VD	1.57	537 - 217 - 130 - 10
	SSL	1.49	434 - 174 - 78 - 10
	SBP	1.55	245 - <b>160</b> - <b>55</b> - 10
	Ours	<b>1.35</b>	<b>179</b> - <b>160</b> - 60 - 10
LeNet-5-Caffe	Original	0.80	20 - 50 - 800 - 500
	VD	0.75	17 - 32 - 329 - 75
	SSL	1.00	<b>3</b> - <b>12</b> - 800 - 500
	SBP	0.86	<b>3</b> - 18 - 284 - 283
	Ours	<b>0.66</b>	16 - 34 - <b>123</b> - <b>62</b>

Table 5 shows the results of compressing neurons in LeNet-500-300. As discussed in [31, 14], VD based SBP corresponds to maximum likelihood estimation, which leads to overly pruning neurons, and hence SBP gets the higher test error 1.55%. The proposed method not only prunes the most neurons but also gains the lowest test error. By contrast, due to the improper log-uniform prior, VD produces the highest test error and only slightly compresses neurons.

Table 5 also shows the results of neurons compression in LeNet-5-Caffe for all methods. We find that the proposed method gains the lowest test error as well as the least neurons, e.g., the test error of the proposed method is even up to 0.66%. In addition, in LeNet-5-Caffe, the first two layers are convolutional layers, and the following two layers are fully-connected layers. Differing from SSL and SBP that mainly focus on pruning neurons in convolutional layers, the result shows that the proposed method prefers to pruning neurons in the fully-connected layers. This means that the proposed method emphasizes feature extraction.

## 6. Conclusion

In this study, we propose a new generalization (*i.e.*, VBD) for Gaussian dropout to address the drawback of VD brought by the improper log-uniform prior, e.g., the ill-posed inference of posterior over network weights. Towards this goal, we exploit a hierarchical prior to the network weights and propose to infer the joint posterior over both these weights and the hyper-parameters defined in their first-level prior. Through implementing the hierarchical prior as a zero-mean Gaussian distribution with variance sampled from a uniform hyper-prior, the proposed VDB can cast the network training and the dropout rate estimation into a joint optimization problem. In VBD, the hierarchical prior is a proper prior which enables the inference of posterior to be well-posed, thus not only leading to a full Bayesian justification for Gaussian dropout but also improving regularization capacity. In addition, we also show that the proposed VBD can be seamlessly applied to network compression. In the experiments on both classification and network compression tasks, the proposed VBD shows superior performance in terms of regularizing network training.

VBD is a general dropout framework, which exploits a promising direction for dropout, *i.e.*, investigating hierarchical priors of network weights. The Gaussian-uniform prior in this study is one feasible choice, but no means the only option. In the future, more effort will be made to exploit other possible choices to regularize the training process of deep neural networks better.

**Acknowledgements:** This work was partially supported by National Key R&D Program of China (2018YFB0904200), National Natural Science Foundation of China (No. 61672024, 61170305 and 60873114) and Australian Research Council (DP140102270 and DP160100703). Yuhang was supported by a scholarship from the China Scholarship Council.

## References

[1] S Derin Babacan, Rafael Molina, Minh N Do, and Aggelos K Katsaggelos. Bayesian blind deconvolution with general sparse image priors. In *European Conference on Computer Vision (ECCV)*, pages 341–355. Springer, 2012. [4](#), [5](#)

[2] Kwokleung Chan, Te-Won Lee, and Terrence J Sejnowski. Variational learning of clusters of undercomplete nonsymmetric independent components. *Journal of Machine Learning Research (JMLR)*, 3(Aug):99–114, 2002. [4](#), [5](#)

[3] Yichen Chen, Dongdong Ge, Mengdi Wang, Zizhuo Wang, Yinyu Ye, and Hao Yin. Strong NP-hardness for sparse optimization with concave penalty functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 740–747, 2017. [6](#)

[4] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning (ICML)*, pages 160–167, 2008. [1](#)

[5] Bin Dai, Chen Zhu, Baining Guo, and David Wipf. Compressing neural networks using the variational information bottleneck. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 1135–1144, 2018. [6](#)

[6] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3581–3590, 2017. [2](#), [6](#), [7](#)

[7] Soumya Ghosh, Jiayu Yao, and Finale Doshi-Velez. Structured variational learning of bayesian neural networks with horseshoe priors. *arXiv preprint arXiv:1806.05975*, 2018. [6](#)

[8] Dong Gong, Jie Yang, Lingqiao Liu, Yanning Zhang, Ian Reid, Chunhua Shen, Anton Van Den Hengel, and Qinfeng Shi. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2319–2328, 2017. [1](#)

[9] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016. [1](#)

[10] Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic network surgery for efficient dnns. In *Advances In Neural Information Processing Systems (NIPS)*, pages 1379–1387, 2016. [7](#)

[11] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1135–1143, 2015. [7](#)

[12] 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 (CVPR)*, pages 770–778, 2016. [1](#)

[13] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. [1](#), [2](#), [3](#), [6](#), [7](#)

[14] Jiri Hron, Alexander GG Matthews, and Zoubin Ghahramani. Variational bayesian dropout: pitfalls and fixes. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 2024–2033, 2018. [2](#), [3](#), [5](#), [8](#)

[15] Jiri Hron, Alexander G de G Matthews, and Zoubin Ghahramani. Variational gaussian dropout is not bayesian. *arXiv preprint arXiv:1711.02989*, 2017. [2](#), [3](#), [5](#)

[16] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. [1](#)

[17] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999. [4](#)

[18] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015. [1](#)

[19] Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2575–2583, 2015. [1](#), [2](#), [3](#), [4](#), [6](#), [7](#)

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012. [1](#)

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. [7](#)

[22] Yuhang Liu, Wenyong Dong, Dong Gong, Lei Zhang, and Qinfeng Shi. Deblurring natural image using

super-gaussian fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–468, 2018. 3

[23] Yuhang Liu, Wenyong Dong, and Mengchu Zhou. Frame-based variational bayesian learning for independent or dependent source separation. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 29(10):4983–4996, 2018. 4

[24] Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3288–3298, 2017. 6

[25] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 2498–2507, 2017. 2, 5, 6, 7, 8

[26] 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 (NIPS)*, pages 6775–6784, 2017. 2, 3, 5, 6, 8

[27] Sungrae Park, JunKeon Park, Su-Jin Shin, and Il-Chul Moon. Adversarial dropout for supervised and semi-supervised learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 2, 7

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[29] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958, 2014. 1, 2, 3, 6, 7

[30] Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research (JMLR)*, 1(Jun):211–244, 2001. 4

[31] Brian Trippe and Richard Turner. Overpruning in variational bayesian neural networks. *arXiv preprint arXiv:1801.06230*, 2018. 8

[32] Karen Ullrich, Edward Meeds, and Max Welling. Soft weight-sharing for neural network compression. *arXiv preprint arXiv:1702.04008*, 2017. 7

[33] Sida Wang and Christopher Manning. Fast dropout training. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 118–126, 2013. 1, 2, 6

[34] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2074–2082, 2016. 8

[35] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016. 1

[36] Lei Zhang, Wei Wei, Yanning Zhang, Chunhua Shen, Anton van den Hengel, and Qinfeng Shi. Cluster sparsity field: An internal hyperspectral imagery prior for reconstruction. *International Journal of Computer Vision (IJCV)*, 126(8):797–821, 2018. 3

[37] Lei Zhang, Wei Wei, Yanning Zhang, Chunna Tian, and Fei Li. Reweighted laplace prior based hyperspectral compressive sensing for unknown sparsity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2274–2281, 2015. 3