COMPOUND VARIATIONAL AUTO-ENCODER

Shang-Yu Su *Shan-Wei Lin* *Yun-Nung Chen*

National Taiwan University, Taipei, Taiwan

f05921117@ntu.edu.tw r03922067@csie.ntu.edu.tw y.v.chen@ieee.org

ABSTRACT

Amortized variational inference (AVI) enables efficient training of deep generative models to scale to large datasets. The quality of the approximate inference is determined by various reasons, such as the ability of producing proper variational parameters for each datapoint in the recognition network and whether the variational distribution matches the true posterior, etc. This paper focuses on the inference sub-optimality of variational auto-encoders (VAE), where the goal is to reduce the difference caused by amortizing the variational distribution parameters over the entire training set instead of optimizing for each training example individually, which is also known as the amortization gap. This paper extends Bayesian inference in VAE from the latent level to both latent and weight levels by adopting Bayesian neural networks (BNN) in the encoder, so that each datapoint obtains its own distribution for better modeling. The hybrid design in the proposed compound VAE is empirically demonstrated to be capable of mitigating the amortization gap.

Index Terms— VAE, Variational Inference, Bayesian

1. INTRODUCTION

Recently, generative models have drawn huge attention from the artificial intelligence community by great success in various tasks. Variational auto-encoders (VAE) [1], generative adversarial networks (GAN) [2] and their variants have been proposed and widely studied. These generative models have achieved huge success in many tasks, such as unconditional or conditional image synthesis [3, 4], image-to-image translation [5], speech synthesis [6], and etc. Among them, the family of VAE models [7, 8] has the elegant design combining a directed graphical model and a deep learning framework for modeling the joint distribution between the data and a set of hidden variables that capture latent factors of variation.

Direct optimization and inference are intractable, so variational inference (VI) provides a framework to optimize a surrogate and tractable distribution to approximate the intractable true posterior [9]. The amortized variant of VI (AVI) for VAE is popular for its feasibility and training efficiency. Previous work [10] investigated inference suboptimality in VAE: the mismatch between the true and approximate posterior. One of the suboptimality is the difference resulting from amortizing variational parameters over the whole training dataset instead of optimizing an individual parameter for each datapoint, called *Amortization Gap* illustrated in Figure 1.

Traditional deep learning techniques used for supervised learning lack the ability of measuring uncertainty in the training and inference processes. It becomes an issue when the trained model encounters a scenario where the real data distribution differs a lot from the distribution of training data; when exposed to data outside the distribution it was trained on, the networks may be forced to extrapolate, leading to an unpredictable behavior. Bayesian methods provide a principled way to model uncertainty by learning the posterior distribution over model parameters [11, 12], which have attempted to extend neural networks into a Bayesian setting. While Bayesian neural networks (BNN) have been studied for decades, unfortunately, BNNs could not scale well and struggle to adapt into modern deep learning architectures. Incorporating the Bayesian framework into deep learning has become a popular topic in machine learning.

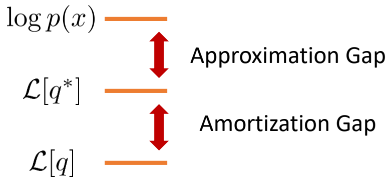

Fig. 1. Gaps in inference.

Prior work [13, 14] developed a practical solution to obtain uncertainty estimates by incorporating dropout techniques into training in typical deep neural networks as a Bayesian approximation of a Gaussian process; it provided theoretical and empirical evidence showing that any network trained with dropout is an approximate Bayesian model, and uncertainty estimates can be obtained by computing the variance on multiple predictions with different dropout masks. While the prior work attempted at incorporating the Bayesian framework in deep learning, none of them introduced BNN into VAE to extend the Bayesian inference to weight distribution modeling (weight level). In addition, considering that the prior work [15] showed that training a deep network using batch normalization is equivalent to approximate inference in Bayesian models, this paper focuses on estimating the model uncertainty without modifying network architectures or training procedure.

In this paper, we propose a new variant of Bayesian neural networks, *dynamic Bayesian neural networks* (DBNN), which models the distribution over the distribution of variational parameters based on input signals; considering the characteristics of Bayesian neural networks–modeling distribution over model parameters with a sampling procedure during inference, we further propose a brand new variant of VAE by incorporating DBNN, batch normalization, and various techniques, called *compound variational auto-encoders*. The contributions are summarized below:

- This paper proposes a new variant of Bayesian neural networks, *dynamic Bayesian neural networks* (DBNN), which models the distribution over the distribution of variational parameters based on input signals.
- A brand new variant of VAE by incorporating DBNN, batch normalization, and various techniques, called *compound variational auto-encoders* is also proposed.
- The proposed framework is empirically validated to have a narrower amortization gap than the vanilla VAE.

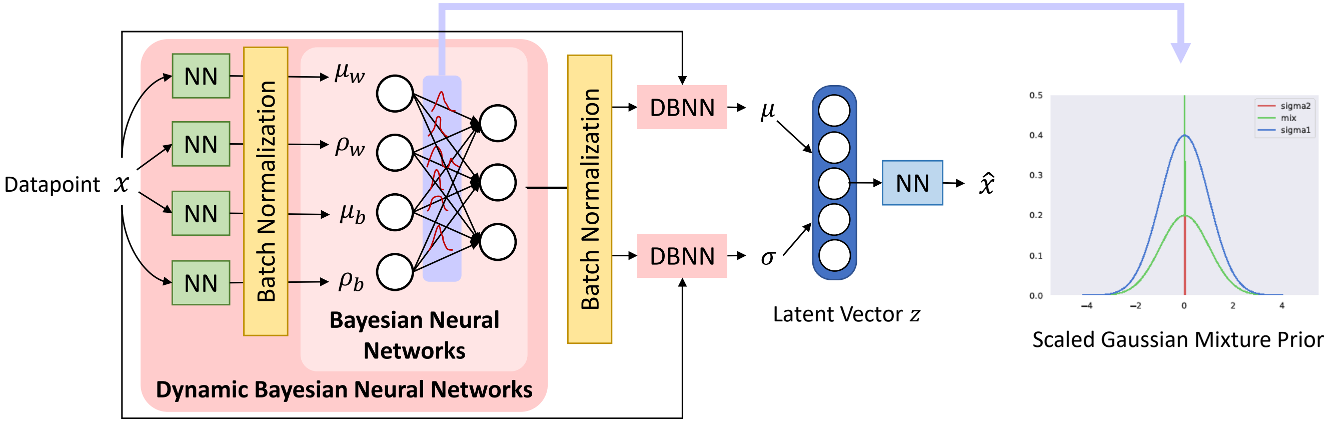

Fig. 2. The model illustration of the proposed compound variational auto-encoder.

2. COMPOUND VARIATIONAL AUTO-ENCODER

One of the inference sub-optimality in VAE, the amortization gap, comes from optimizing model parameters ϕ over the whole dataset instead of assigning and optimizing individual parameters for each individual datapoint. An intuitive approach is to extend static deep neural networks to networks with dynamic parameters. Following the idea, we incorporate Bayesian neural networks into the encoder of VAE, named *compound variational auto-encoder*. The proposed compound VAE is illustrated in Figure 2.

2.1. Improved Variational Learning Objective

To derive the learning objective of the proposed model, first, we define the joint generative distribution $p_\theta(x, z, w)$ and the inference distribution $q_\phi(z, w \mid x)$:

$$p_\theta(x, z, w) = p_\theta(z \mid w)p_\theta(w)p_\theta(x \mid z, w), \tag{1}$$

$$q_\phi(z, w \mid x) = q_\phi(z \mid w, x)q_\phi(w \mid x), \tag{2}$$

where z is the latent representation of VAE, w is the weights of BNN, and x is the input datapoint. In the variational inference approach, the goal is to find the parameters ϕ and θ that minimize KL divergence between the approximate distribution and the true posterior:

$$\begin{aligned} & D_{KL}(q_\phi(z, w \mid x) \parallel p_\theta(z, w \mid x)) \qquad (3)\\ &= \mathbb{E}_{q_\phi(z,w|x)}[\log q_\phi(z, w \mid x) - \log p_\theta(z, w \mid x)] \\ &= \mathbb{E}_{q_\phi(z,w|x)}[\log q_\phi(z, w \mid x) - \log p_\theta(z, w, x)] + \log p_\theta(x). \end{aligned}$$

Minimizing the first expectation term is equal to maximizing the likelihood, so it can be expanded by (1) and (2):

$$\begin{aligned} & \mathbb{E}_{q_\phi(z,w|x)}[\log q_\phi(z, w \mid x) - \log p_\theta(z, w, x)] \qquad (4)\\ =& \mathbb{E}_{q_\phi(z,w|x)}[\log q_\phi(z|w, x) + \log q_\phi(w \mid x) \\ & - \log p_\theta(z \mid w) - \log p_\theta(w) - \log p_\theta(x \mid z, w)] \\ =& \mathbb{E}_{q_\phi(w|x)q_\phi(z|w,x)}[\log q_\phi(z \mid w, x) - \log p_\theta(z \mid w)] \\ & + \mathbb{E}_{q_\phi(z,w|x)}[\log q_\phi(w \mid x) - \log p_\theta(w)] \\ & - \mathbb{E}_{q_\phi(z,w|x)}[\log p_\theta(x \mid z, w)] \\ =& \mathbb{E}_{q_\phi(w|x)}[D_{KL}(\log q_\phi(z \mid w, x) \parallel \log p_\theta(z|w))] \\ & + \mathbb{E}_{q_\phi(z,w|x)}[\log q_\phi(w \mid x) - \log p_\theta(w)] \\ & - \mathbb{E}_{q_\phi(z,w|x)}[\log p_\theta(x \mid z, w)]. \end{aligned}$$

The derived evidence lower bound objective (ELBO) now has three terms: the first term is the Kullback-Leibler (KL) divergence between the latents and their prior, the second term is the divergence between the weights and their priors, and the last term can be viewed as the reconstruction likelihood. The whole objective embodies the capability of modeling data distribution (the last term) and soft constraints regularizing both latent representation and model parameters to approximate specified priors.

2.2. Dynamic Bayesian Recognition Network

After deriving the objective, we endow every tractable term a physical meaning. Among them, the second term of the derived objective is the KL divergence between the approximate distribution from the true posterior of weights in the BNN-based encoder $D_{KL}(\log q_\phi(w \mid x) \parallel \log p_\theta(w))$. However, the approximate distribution term is $\log q_\phi(w \mid x)$, indicating the distribution over weights w given the input x. In other words, it is not a conventional formulation of Bayesian neural networks where the distribution over weights is independent of the input $\log q_\phi(w)$.

Therefore we propose a new variant of Bayesian neural networks, called *dynamic Bayesian neural network* (DBNN) to model weight distribution based on the input ($\log q_\phi(w \mid x)$); specifically, we apply linear projection on the input x, for each DBNN module:

$$\mu_w = W_{w,\mu} \cdot x \quad \rho_w = W_{w,\rho} \cdot x,$$
$$\mu_b = W_{b,\mu} \cdot x \quad \rho_b = W_{b,\rho} \cdot x,$$

then obtain the weights w and the bias b via reparameterization, and further compute the hidden vector h:

$$\begin{aligned} & \mathit{sample}\ \epsilon \sim \mathcal{N}(0, I), \\ & w = \mu_w + \log(1 + \exp(\rho_w)) \odot \epsilon, \\ & b = \mu_b + \log(1 + \exp(\rho_b)) \odot \epsilon, \\ & h = w \cdot x + b. \end{aligned}$$

By this design, the proposed model is capable of generating different weight distributions based different input signals, where the extension allows the model to have more capacity and flexibility of learning the distribution over the variational parameters. Furthermore, the proposed concept about DBNN is general and can be utilized by various deep learning architectures.

2.2.1. *Scaled Mixture Gaussian Prior*

In this paper, we conduct a scaled mixture of two Gaussians as the prior distribution on weights; both Gaussians come with zero mean, but with different variances.

$$P(w) = \prod_j \pi N(w_j|0, \sigma_1^2) + (1-\pi)N(w_j|0, \sigma_2^2)$$

where w_j is the jth weight of the network and π is the ratio of the composition of two Gaussian density function. By setting $\sigma_1 > \sigma_2$ and $1 \gg \sigma_2$, we obtain a prior composed of a heavy tail (σ_1) with a concentration around the 0 mean (σ_2), which is similar to the spike-and-slab prior [16, 17, 18], as illustrated in the right part of Figure 2. The spike-and-slab prior has various advantages over other common prior distribution in Bayesian approaches. In this work, π is set to 0.5, σ_1 is 1, and σ_2 is $\exp(-6)$, all the weights have the same prior parameters, which makes it amenable and feasible to optimize.

2.2.2. *Complexity Loss Reweighting*

When training VAE, there are several ways to improve training efficiency and stability, one of the notable method is KL annealing [19]: the idea is to anneal the KL divergence term in the traditional ELBO objective; the approach gives the reconstruction loss and the KL divergence regularization term different weight coefficients and makes the weight of KL term gradually increase, from 0 to 1.

When training BNN, we formulate a similar learning objective by variational inference; therefore we rewrite the loss term:

$$\mathcal{L}_{B,i} = \pi_i D_{KL}(q_\phi(w) \parallel (w)) - \mathbb{E}_{q_\phi(w)}[\log P(D_i \mid w)], \quad (5)$$

where $\pi_i \in [0,1]$ and $\sum_i^n \pi_i = 1$. Then $\mathbb{E}_n[\sum_i^n \mathcal{L}_{B,i}] = \mathcal{L}_B$, where $\mathbb{E}_n$ denotes an expectation over the random partitioning of mini-batches. We apply the scheme $\pi_i = \frac{2^{n-1}}{2^n-1}$, where the first few mini-batches are heavily influenced by the complexity cost, and the later mini-batches are largely influenced by the data. At the beginning of learning, the scheme is particularly useful, because for the first few mini-batches, changes of weights due to the data are subtle, and data become more influential and the prior less influential when seeing more data.

2.2.3. *Batch Normalization*

When training neural networks, the change in the distributions of layers inputs is a notable problem in training efficiency, because the layers need to continuously adapt to the new distribution. Prior work [20] defined *internal covariate shift* as the change in the distribution of network activations due to the change in network parameters during training, and further proposed the *batch normalization* technique to address the issue.

Due to mini-batch training by gradient descent methods exploited here, we estimate the mean and variance of each mini-batch and normalize the hidden vector by each dimension, making the normalized representations to have zero mean and the variance equal to one:

$$\hat{h} = \frac{h - \mathbb{E}[h]}{\sqrt{\mathrm{Var}[h] + \delta}}, \quad (6)$$

where δ is a small scalar constant added for numerical stability. Simply normalizing the distribution sometimes severely restricts the expressiveness of the neural networks due to the nature of some commonly-used activation functions, such as the sigmoid function. Therefore we instead shift and scale the features as below:

$$\hat{h} = \gamma \odot \hat{h} + \beta. \quad (7)$$

Even conventional deep neural networks would have the issue about covariate shift, we argue that Bayesian neural networks would suffer from internal covariate shift much more due to the sampling characteristics. Therefore we propose several mechanisms to enable model training on the parameters of distribution over variational parameters (μ_w, μ_b, ρ_w, ρ_b) and the hidden vector used to predict the parameters of distribution over the latent representation (μ, σ). The model framework with the proposed mechanisms is illustrated in Figure 2.

3. EXPERIMENT

3.1. Experiment Settings

The experiments are conducted on MNIST[21] and EMNIST [22] datasets, both of which consist of a training and test set with 60000 and 10000 datapoints respectively. Each example is a 28x28 grey-scale image and pixel values are scaled to be within the range $[0, 1]$.

The size of all the hidden layer is 100, Rectified Linear Unit (RELU) is chosen to be the activation function between hidden layers, and the dimension of the latent representation is set to 5. All models are optimized via Adam [23] with initial learning rate 0.001 for 50 epochs of batch size 32 without early-stop methods.

The performed models in the experiments include:

- **Vanilla VAE** composes of symmetric encoder and decoder module as a baseline for comparison [1].
- **Simplified Compound VAE** is the simplified version of the proposed model that incorporates BNN for modeling the weight distribution. This model replaces feed-forward neural networks in the encoder with typical BNNs to approximate the distribution of the variational parameters over the training set, but the distribution is fixed after training process.
- **Compound VAE** is the complete version of the proposed model that incorporates DBNN for modeling the distribution over the weight distribution.

For both versions of compound VAE, different Gaussian weight priors are adopted and the results are reported.

Table 1 reports the performance ($\mathcal{L}_{rec}$, D_{KL}, $\mathcal{L}_{rec} + \mathcal{L}_{KL}$) of the baseline models and the proposed compound VAE models with different settings.

3.2. Results

Table 1 shows that the simplified version of the proposed compound VAE slightly outperform the vanilla VAE. Intuitively, extending the original static feed-forward neural networks to dynamic ones could improve the model capacity, and the experimental results show that it also indirectly reduces the amortization gap. In other words, instead of approximating a set of static variational parameters, sampling parameters from a learned posterior during inference is empirically demonstrated to be a better mechanism in amortized variational inference.

Because the KL divergence (D_{KL}) is the regularization term and highly depends on the size of latent representation, the reconstruction loss becomes an important performance indicator. Note that all models in Table 1 have the same size of latent representations, indicating that the reconstruction loss reflects the richness of relevant

Model		MNIST			EMNIST		
		$\mathcal{L}_{rec}$	D_{KL}	$\mathcal{L}_{rec} + D_{KL}$	$\mathcal{L}_{rec}$	D_{KL}	$\mathcal{L}_{rec} + D_{KL}$
Vanilla VAE		114.02	11.39	125.41	177.71	11.99	189.70
		112.33	11.09	123.42	176.07	11.84	187.91
Simplified Compound VAE	single Gaussian	113.24	11.52	124.76†	177.67	12.16	189.84
		111.02	11.45	122.47†	175.94	12.10	188.05
	Gaussian mixture	113.21	11.49	124.70†	176.23	12.25	188.48†
		110.90	11.37	122.28†	174.60	12.24	186.84†
Compound VAE	single Gaussian	109.53	11.95	121.48†	168.22	12.85	181.08†
		160.37	7.19	167.56	244.32	3.54	247.87
	Gaussian mixture	110.96	11.81	122.77†	172.18	12.46	184.65†
		107.77	11.80	**119.56**†	168.13	12.45	**180.57**†
Compound VAE (Gaussian mixture)		118.01	11.06	129.07	179.19	11.92	191.11
without DBNN modules for μ and σ		110.52	11.20	121.72	171.70	11.74	183.43
Compound VAE (Gaussian mixture)		154.39	7.87	162.26	228.31	7.50	235.80
without KL reweighting		141.22	7.40	148.62	216.12	7.34	223.46
Compound VAE (Gaussian mixture)		124.08	12.59	136.67	189.92	13.78	203.71
without batch normalization		119.87	12.29	132.16	185.00	13.64	198.64

Table 1. The performance of the baselines and the proposed compound VAE on training/testing set (upper row: training, lower row: testing).

information in the latent representation. Experimental results show that almost all models have nearly the same performance in KL divergence, and the proposed compound VAEs significantly outperform the baseline vanilla VAE in reconstruction loss and the learning target ELBO ($\mathcal{L}_{\text{ELBO}}$). The exception is the proposed model with the standard Gaussian weight prior ($\mathcal{N}(0, I)$), which achieves the best performance in training set; however, the model performs worst among all results (shown as the underlined numbers). Its reconstruction loss is much higher and the KL divergence is much lower than others, meaning that the model suffers from the well known issue in VAE called *posterior collapse*. The issue refers to the phenomenon that VAEs tend to fit the latent distribution to the standard Gaussian prior and fail to extract relevant information into latents.

However, the simplified compound VAE does not suffer from the posterior collapse issue, and the reason may be that the complete version has remarkable flexibility to model the distribution over the distribution of variational parameters by predicting the distribution parameters for the distribution of variational parameters based on the input signal. On the other hand, the simplified version only replaces feed-forward networks with BNNs, where all parameters are randomly initialized and adjusted by gradient-based method. Hence, because the complete version has more capability to fit the specified prior, the output of the DBNN probably contains more characteristics of the prior, which is the standard Gaussian distribution. However, the prior of the latent representation is also the standard Gaussian distribution, so we argue that the model is easier to fit the prior of latents and then results in the posterior collapse issue.

3.3. Analysis

In order to validate each component in the proposed compound VAE, a set of ablation tests is conducted based on the complete version of the proposed model shown in the lower part of Table 1.

3.3.1. Expressiveness of DBNN

To validate the expressiveness of the design in the proposed dynamic Bayesian neural network, we replace the DBNN modules for predicting parameters of latent distribution (μ and σ) with typical feed-forward neural networks and keep other setting unchanged. Experimental results demonstrate the better model capacity for the proposed DBNN with exact the same amount of weights used to transform the input signal.

3.3.2. Effectiveness of KL Reweighting

We further examine the effectiveness of the KL reweighting mechanism in the framework; Table 1 shows that, without KL reweighting, the models would have lower KL divergence but higher reconstruction loss, which is also a sign of tendency toward posterior collapse. Hence, KL reweighting is shown to be a critical mechanism in the training procedure.

3.3.3. Effectiveness of Batch Normalization

The proposed compound VAE approximates the distribution over the distribution of variational parameters, while the vanilla VAE only models a set of variational parameters. To validate if the flexibility and sampling property would result in a severer internal covariate shift issue, we test the models trained without batch normalization. Experimental results show that both KL divergence and reconstruction loss would raise without batch normalization, showing the importance of the technique.

In sum, three mechanisms are critical to enable better and efficient training procedure, and our proposed compound VAE is empirically demonstrated effectiveness in two benchmark experiments.

4. CONCLUSION

This paper proposes a new variant of Bayesian neural networks, named *dynamic Bayesian neural networks* (DBNN), which models the distribution over the distribution of variational parameters based on input signals; such model design is able to generalize to other deep learning architectures. Also, a brand new variant of VAE that incorporates DBNN, batch normalization, and various techniques, called *compound variational auto-encoders* is further proposed. The proposed framework is empirically proven to have a narrower amortization gap than the vanilla VAE, and experimental results on two datasets also derive the conclusions.

5. REFERENCES

[1] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[3] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, et al., "Fader networks: Manipulating images by sliding attributes," in *Advances in Neural Information Processing Systems*, 2017, pp. 5967–5976.

[4] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool, "Pose guided person image generation," in *Advances in Neural Information Processing Systems*, 2017, pp. 406–416.

[5] Ming-Yu Liu, Thomas Breuel, and Jan Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 700–708.

[6] Sercan Arik, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," *arXiv preprint arXiv:1705.08947*, 2017.

[7] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov, "Importance weighted autoencoders," *arXiv preprint arXiv:1509.00519*, 2015.

[8] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling, "Improved variational inference with inverse autoregressive flow," in *Advances in Neural Information Processing Systems*, 2016, pp. 4743–4751.

[9] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

[10] Chris Cremer, Xuechen Li, and David Duvenaud, "Inference suboptimality in variational autoencoders," *arXiv preprint arXiv:1801.03558*, 2018.

[11] Radford M Neal, *Bayesian learning for neural networks*, vol. 118, Springer Science & Business Media, 2012.

[12] Zoubin Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452, 2015.

[13] Yarin Gal, "Uncertainty in deep learning," *University of Cambridge*, 2016.

[14] Yarin Gal and Zoubin Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[15] Mattias Teye, Hossein Azizpour, and Kevin Smith, "Bayesian uncertainty estimation for batch normalized deep networks," in *Proceedings of the 35th International Conference on Machine Learning*, Jennifer Dy and Andreas Krause, Eds., Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 4907–4916, PMLR.

[16] Hugh Chipman, "Bayesian variable selection with related predictors," *Canadian Journal of Statistics*, vol. 24, no. 1, pp. 17–36, 1996.

[17] Edward I George and Robert E McCulloch, "Variable selection via gibbs sampling," *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 881–889, 1993.

[18] Toby J Mitchell and John J Beauchamp, "Bayesian variable selection in linear regression," *Journal of the American Statistical Association*, vol. 83, no. 404, pp. 1023–1032, 1988.

[19] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio, "Generating sentences from a continuous space," *arXiv preprint arXiv:1511.06349*, 2015.

[20] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik, "Emnist: an extension of mnist to handwritten letters," *arXiv preprint arXiv:1702.05373*, 2017.

[23] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.