Contextual Dropout for Bayesian Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Dropout has been demonstrated as a simple and effective tool to not only regularize the training process of deep neural networks, but also estimate the prediction uncertainty. It is common to assume that the dropout distribution is independent of the input covariates and set the same across all data samples. In this paper, we propose contextual dropout as a scalable sample-dependent dropout method, which makes the dropout probabilities in the variational posterior depend on the input covariates of each data sample, in a particular way that only slightly increases the model size and computational complexity. We learn the covariate-dependent dropout probabilities with a variational objective, which we show is compatible with both Bernoulli dropout and Gaussian dropout. We conduct experiments on both image classification and visual question answering where dropout is applied to three representative types of neural network layers. Our experimental results show that contextual dropout outperforms baseline methods in terms of both accuracy and quality of uncertainty estimation.

1 Introduction

Deep neural networks (NNs) have become ubiquitous in engineering and scientific studies, achieving state-of-the-art results in a wide variety of research problems [1]. With ever increasing computational power, we are able to train large NNs on an unprecedented scale. To prevent over-parameterized NNs from overfitting, we often need to appropriately regularize their training. One way to do so is to use Bayesian NNs that treat the NN weights as random variables and regularize them with appropriate prior distributions [2, 3]. More importantly, the posteriors of the NN weights can naturally be used to evaluate the uncertainty for out-of-sample predictions. For example, by evaluating the consistency between the predictions that are conditioned on different posterior samples of NN weights, we can obtain the model's confidence on its predictions, which is critical for real-life deployment of artificial intelligence (AI) systems. However, despite significant recent efforts in developing various types of approximate inference for Bayesian NNs [3–9], the large number of NN weights makes it difficult to well model their distributions, and the requirement of drawing multiple posterior samples of NN weights for uncertainty estimation makes it difficult to scale to real-world applications.

Another regularization strategy that has been demonstrated to be simple and effective is dropout, which randomly shuts down neurons during training [10–12]. Relating dropout to Bayesian inference helps provide a much simpler and more efficient way than using Bayesian NNs to provide uncertainty estimation [13], as there is no more need to explicitly instantiate multiple sets of NN weights. However, whether the uncertainty estimation is well-calibrated depends heavily on the dropout probabilities [14]. To allow different NN layers to have different dropout probabilities but avoid computationally prohibitive grid-search, Gal et al. [15] develops a concrete relaxation of binary dropout. Gaussian dropout is another type of dropout. It multiplies the neurons with independent, and identically distributed (iid) Gaussian random variables drawn from $\mathcal{N}(1, \alpha)$, where the variance

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

α is a tuning parameter [11]. Variational dropout generalizes Gaussian dropout by reformulating it under a Bayesian setting and allowing α to be learned under a variational objective [16, 17].

Consider an observed data with covariates $\boldsymbol{x}$ and label y. In existing methods, the dropout probabilities are treated as global parameters and hence are independent of $\boldsymbol{x}$. Instead we propose parameterizing them as a function of $\boldsymbol{x}$, making them become data-specific local parameters. From another perspective, applying conventional dropouts can be viewed as imposing a single distribution over the NN weights [13, 16], while applying covariate-dependent dropouts makes different data to have different distributions over the NN weights. Instead of treating the weights as global variables, we now treat them as data-specific local variables. This generalization, which makes the distribution of the NN weights become covariate-dependent, has the potential to greatly enhance the expressiveness of a Bayesian NN. However, learning covariate-dependent dropout rates is challenging. Ba and Frey [18] propose *standout* where a binary belief network is laid over the original network and develop a heuristic approximation to optimize free energy. But, as pointed out by Gal et al. [15], it is difficult to scale this method to a large model due to its need to significantly increase the model size.

In this paper, we propose contextual dropout, whose dropout rates depend on the covariates $\boldsymbol{x}$, as a new approximate Bayesian inference method for NNs, providing well calibrated uncertainty estimation. With a novel design that reuses the decoder network to define how the covariate-dependent dropout rates are produced in its variational encoder, it boosts the performance while only slightly increases the memory and computational cost. In contrast to conventional Bayesian NNs, contextual dropout maintains the inherent advantage of dropout in not requiring drawing multiple random samples of the NN weights for uncertainty estimation. Another benefit of this design is allowing contextual dropout to be directly plugged into various types of NN layers, including fully connected, convolutional, and attention layers. On a variety of supervised learning tasks, contextual dropout achieves good performance in terms of accuracy, and quality of uncertainty estimation.

2 Contextual dropout

Consider a supervised learning problem with training data $\mathcal{D} := \{\boldsymbol{x}_i, y_i\}_{i=1}^N$, where we model the conditional probability $p_{\boldsymbol{\theta}}(y_i \,|\, \boldsymbol{x}_i)$ using a NN parameterized by $\boldsymbol{\theta}$. Applying dropout to a NN often means element-wisely reweighing each layer with a data-specific Bernoulli/Gaussian distributed random mask $\boldsymbol{z}_i$, which are iid drawn from prior $p_{\boldsymbol{\eta}}(\boldsymbol{z})$ parameterized by $\boldsymbol{\eta}$ [10, 11]. This implies dropout training can be viewed as approximate Bayesian inference [13]. More specifically, one may view the learning objective of a supervised learning model with dropout as a log-marginal-likelihood: $\log \int \prod_{i=1}^N p(y_i \,|\, \boldsymbol{x}_i, \boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}$. To maximize this often intractable log-marginal, it is common to resort to variational inference [19, 20] that introduces a variational distribution $q(\boldsymbol{z})$ on the random mask $\boldsymbol{z}$ and optimizes an evidence lower bound (ELBO) as

$$\mathcal{L}(\mathcal{D}) = \mathbb{E}_{q(\boldsymbol{z})}\left[\log \frac{\prod_{i=1}^N p_{\boldsymbol{\theta}}(y_i \,|\, \boldsymbol{x}_i, \boldsymbol{z}) p_{\boldsymbol{\eta}}(\boldsymbol{z})}{q(\boldsymbol{z})}\right] = \left(\sum_{i=1}^N \mathbb{E}_{\boldsymbol{z}_i \sim q(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(y_i \,|\, \boldsymbol{x}_i, \boldsymbol{z}_i)\right]\right) - \mathrm{KL}(q(\boldsymbol{z})||p_{\eta}(\boldsymbol{z})), \quad (1)$$

where $\mathrm{KL}(q(\boldsymbol{z})||p_{\boldsymbol{\eta}}(\boldsymbol{z})) = \mathbb{E}_{q(\boldsymbol{z})}[\log q(\boldsymbol{z}) - \log p(\boldsymbol{z})]$ is a Kullback-Leibler (KL) divergence based regularization term. Whether the KL term is explicitly imposed is a key distinction between regular dropout [10, 11] and their Bayesian generalizations [13–17]. Note while $\boldsymbol{z}$ in regular dropout, as shown in the first expression of $\mathcal{L}(\mathcal{D})$ in (1), is treated as a global variable independent of $\boldsymbol{x}_i$, it is a common practice to follow the second expression of $\mathcal{L}(\mathcal{D})$ in (1) to draw iid data-specific random dropout masks $\boldsymbol{z}_i \sim q(\boldsymbol{z})$ when estimating $\mathcal{L}(\mathcal{D})$, which often leads to lower gradient variance [16].

2.1 Covariate-dependent weight uncertainty

In regular dropout, as shown in (1), while we make the dropout masks data specific during optimization, we keep their distributions the same. This implies that while the NN weights can vary from data to data, their distribution is kept data invariant. In this paper, we propose *contextual dropout*, in which the distributions of dropout masks $\boldsymbol{z}_i$ depend on covariates $\boldsymbol{x}_i$ for each data $(\boldsymbol{x}_i, y_i)$. Specifically, we define the variational distribution as $q_{\boldsymbol{\phi}}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i)$, where $\boldsymbol{\phi}$ denotes its NN parameters. In the framework of amortized variational Bayes [21, 22], we can view $q_{\boldsymbol{\phi}}$ as an inference network (encoder) trying to approximate the posterior $p(\boldsymbol{z}_i \,|\, y_i, \boldsymbol{x}_i) \propto p(y_i \,|\, \boldsymbol{x}_i, \boldsymbol{z}_i) p(\boldsymbol{z}_i)$. Note as we have no access to y_i during testing, we parameterize our encoder in a way that it depends on $\boldsymbol{x}_i$ but not y_i. From the optimization point of view, what we propose corresponds to the ELBO of

$\log \prod_{i=1}^{N} \int p(y_i \mid \boldsymbol{x}_i, \boldsymbol{z}_i) p(\boldsymbol{z}_i) d\boldsymbol{z}_i$ given $q_{\boldsymbol{\phi}}(\boldsymbol{z}_i \mid \boldsymbol{x}_i)$ as the encoder, which can be expressed as

$$\mathcal{L}(\mathcal{D}) = \sum_{i=1}^{N} \mathcal{L}(\boldsymbol{x}_i, y_i),\ \mathcal{L}(\boldsymbol{x}_i, y_i) = \mathbb{E}_{\boldsymbol{z}_i \sim q_{\boldsymbol{\phi}}(\cdot \mid \boldsymbol{x}_i)}[\log p_{\boldsymbol{\theta}}(y_i \mid \boldsymbol{x}_i, \boldsymbol{z}_i)] - \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}_i \mid \boldsymbol{x}_i) || p_{\boldsymbol{\eta}}(\boldsymbol{z}_i)). \quad (2)$$

This ELBO differs from that of regular dropout in (1) in that the dropout distributions for $\boldsymbol{z}_i$ are now parameterized by $\boldsymbol{x}_i$ and a single KL regularization term is replaced with the aggregation of N data-dependent KL terms. Unlike conventional Bayesian NNs, as $\boldsymbol{z}_i$ is now a local random variable, the impact of the KL terms will not diminish as N increases, and from the viewpoint of uncertainty quantification, contextual dropout relies only on aleatoric uncertainty to model its uncertainty on y_i given $\boldsymbol{x}_i$. Like conventional BNNs, we may add epistemic uncertainty by imposing a prior distribution on $\boldsymbol{\theta}$ and/or $\boldsymbol{\phi}$, and infer their posterior given $\mathcal{D}$. As contextual dropout with a point estimate on both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ is already achieving state-of-the-art performance, we leave that extension for future research. In what follows, we omit the data index i to simplify the notation.

Note imposing random dropout masks on the neurons at each layer can be equivalently expressed as drawing random NN weights from some specific distribution [16]. Thus contextual dropout enables us to provide covariate-dependent weight uncertainty, achieving data-dependent distributions for NN weights, which helps improve training and calibrate uncertainty. Moreover, this enhanced Bayesian modeling ability is realized in the dropout framework that does not require instantiating multiple samples of NN weights for uncertainty estimation. Below we formally define its model structure.

Cross-layer dependence: For a NN with L layers, we denote $\boldsymbol{z} = \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^L\}$, with $\boldsymbol{z}^l$ representing the dropout masks at layer l. As we expect $\boldsymbol{z}^l$ to be dependent on the dropout masks in previous layers $\{\boldsymbol{z}^j\}_{j<l}$, we introduce an autoregressive distribution as $q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) = \prod_{l=1}^{L} q_{\boldsymbol{\phi}}(\boldsymbol{z}^l \mid \boldsymbol{x}^{l-1})$, where $\boldsymbol{x}^{l-1}$, the output of layer $l-1$, is a function of $\{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^{l-1}, \boldsymbol{x}\}$.

Parameter sharing between encoder and decoder: We aim to build an encoder by modeling $q_{\boldsymbol{\phi}}(\boldsymbol{z}^l \mid \boldsymbol{x}^{l-1})$, where $\boldsymbol{x}$ may come from complex and highly structured data such as images and natural languages. Thus, extracting useful features from $\boldsymbol{x}$ to learn the encoder distribution $q_{\boldsymbol{\phi}}$ itself becomes a problem as challenging as the original one, *i.e.*, extracting discriminative features from $\boldsymbol{x}$ to predict y. As intermediate layers in the decoder network $p_{\boldsymbol{\theta}}$ are already learning useful features from the input, we choose to reuse them in the encoder, instead of extracting the features from scratch. If we denote layer l of the decoder network by $g_{\boldsymbol{\theta}}^l$, then the output of layer l, given its input $\boldsymbol{x}^{l-1}$, would be $\mathbf{U}^l = g_{\boldsymbol{\theta}}^l(\boldsymbol{x}^{l-1})$. Considering this as a learned feature for $\boldsymbol{x}$, as illustrated in Figure 1, we build the encoder on this output as $\boldsymbol{\alpha}^l = h_{\boldsymbol{\varphi}}^l(\mathbf{U}^l)$, draw $\boldsymbol{z}^l$ conditioning on $\boldsymbol{\alpha}^l$, and element-wisely multiply $\boldsymbol{z}^l$ with $\mathbf{U}^l$ (with broadcast if needed) to produce the output of layer l as $\boldsymbol{x}^l$. In this way, we use $\{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ to parameterize the encoder, which reuses parameter $\boldsymbol{\theta}$ of the decoder. To produce the dropout rates of the encoder, we only need extra parameter $\boldsymbol{\varphi}$, the added memory and computational cost of which are often insignificant in comparison to these of the decoder (see Table 4 in Appendix for the model sizes of different dropout methods).

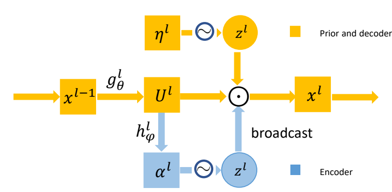

Figure 1: A contextual dropout block.

2.2 Efficient parameterization of contextual dropout

Denote the output of layer l by a multidimensional array (tensor) $\mathbf{U}^l = g_{\boldsymbol{\theta}}^l(\boldsymbol{x}^{l-1}) \in \mathbb{R}^{C_1^l \times \ldots \times C_{D^l}^l}$, where D^l denotes the total number of dimensions of $\mathbf{U}^l$ and C_d^l denotes the number of elements along dimension $d \in \{1, \ldots, D^l\}$. For efficiency, the output shape of $h_{\boldsymbol{\varphi}}^l$ is not matched to the shape of $\mathbf{U}^l$. Instead, we make it smaller and broadcast the contextual dropout masks $\boldsymbol{z}^l$ across the dimensions of $\mathbf{U}^l$ [23]. Specifically, we parameterize dropout logits $\boldsymbol{\alpha}^l$ of the variational distribution to have C_d^l elements, where $d \in \{1, ..., D^l\}$ is a specified dimension of $\mathbf{U}^l$. We sample $\boldsymbol{z}^l$ from the encoder and broadcast them across all but dimension d of $\mathbf{U}^l$. We sample $\boldsymbol{z}^l \sim \mathrm{Ber}(\sigma(\boldsymbol{\alpha}^l))$ under contextual Bernoulli dropout, and follow Srivastava et al. [11] to use $\boldsymbol{z}^l \sim N(1, \sigma(\boldsymbol{\alpha}^l)/(1-\sigma(\boldsymbol{\alpha}^l)))$ for contextual Gaussian dropout. To obtain $\boldsymbol{\alpha}^l \in \mathbb{R}^{C_d^l}$, we first take the average pooling of $\mathbf{U}^l$ across all but dimension d, with the output denoted as $F_{\text{avepool},d}(\mathbf{U}^l)$, and then apply two fully-connected layers Φ_1^l and Φ_2^l connected by F_{NL}, a (Leaky) ReLU based nonlinear activation function, as

$$\boldsymbol{\alpha}^l = h_{\boldsymbol{\varphi}}^l(\mathbf{U}^l) = \Phi_2^l(F_{\text{NL}}(\Phi_1^l(F_{\text{avepool},d}(\mathbf{U}^l)))), \quad (3)$$

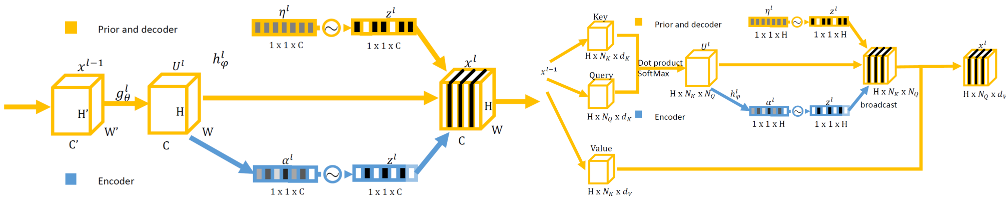

Figure 2: Left: Contextual dropout in convolution layers. Right: Contextual dropout in attention layers.

where Φ^l_1 is a linear transformation mapping from $\mathbb{R}^{C^l_d}$ to $\mathbb{R}^{C^l_d/\gamma}$, while Φ^l_2 is from $\mathbb{R}^{C^l_d/\gamma}$ back to $\mathbb{R}^{C^l_d}$, with γ being a reduction ratio controlling the complexity of $h^l_{\boldsymbol{\varphi}}$. Below we describe how to apply contextual dropout to three representative types of NN layers.

Contextual dropout for fully-connected layers: If layer l is a fully-connected layer and $\mathbf{U}^l \in \mathbb{R}^{C^l_1 \times \cdots \times C^l_{D^l}}$, we set $\boldsymbol{\alpha}^l \in \mathbb{R}^{C^l_{D^l}}$, where D^l is the dimension that the linear transformation is applied to. Note, if $\mathbf{U}^l \in \mathbb{R}^{C^l_1}$, then $\boldsymbol{\alpha}^l \in \mathbb{R}^{C^l_1}$, and $F_{\text{avepool},1}$ is an identity map, so $\boldsymbol{\alpha}^l = \Phi^l_2 F_{\text{NL}}(\Phi^l_1(\mathbf{U}^l))$.

Contextual dropout for convolutional layers: Assume layer l is a convolutional layer with C^l_3 as convolutional channels and $\mathbf{U}^l \in \mathbb{R}^{C^l_1 \times C^l_2 \times C^l_3}$. Similar to Spatial Dropout [23], we set $\boldsymbol{\alpha}^l \in \mathbb{R}^{C^l_3}$ and broadcast its corresponding $\boldsymbol{z}^l$ spatially as illustrated in Figure 2. Such parameterization is similar to the squeeze-and-excitation unit for convolutional layers, which has been shown to effective in image classification tasks [24]. However, in squeeze-and-excitation, $\sigma(\boldsymbol{\alpha}^l)$ is used as channel-wise soft attention weights instead of dropout probabilities, therefore it serves as a deterministic mapping in the model instead of a stochastic unit used in variational inference.

Contextual dropout for attention layers: Dropout has been widely used in attention layers [25–27]. For example, it can be applied to multi-head attention weights after the softmax operation (see illustrations in Figure 2). The weights are of dim $[H, N_K, N_Q]$, where H is the number of heads, N_K the number of keys, and N_Q the number of queries. In this case, we find that setting $\boldsymbol{\alpha}^l \in \mathbb{R}^H$ gives good performance. Intuitively, this coincides with the choice of channel dimension for convolutional layers, as heads in attention could be analogized as channels in convolution.

2.3 Variational inference for contextual dropout

In contextual dropout, we choose $\mathcal{L}(\mathcal{D}) = \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \mathcal{L}(\boldsymbol{x}, y)$ shown in (2) as the optimization objective. Note in our design, the encoder $q_{\boldsymbol{\phi}}$ reuses the decoder parameter $\boldsymbol{\theta}$ to define its own parameter. Therefore, we copy the values of $\boldsymbol{\theta}$ into $\boldsymbol{\phi}$ and stop the gradient of $\boldsymbol{\theta}$ when optimizing $q_{\boldsymbol{\phi}}$. This is theoretically sound and for verification, we find that the performance often clearly drops without stop gradient. We use a simple prior $p_{\boldsymbol{\eta}}$, making the prior distributions for all dropout masks the same within each layer. The gradients with respect to $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$ can be expressed as

$$\nabla_{\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{x}, y) = \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\cdot \,|\, \boldsymbol{x})}[\nabla_{\boldsymbol{\eta}} \log p_{\boldsymbol{\eta}}(\boldsymbol{z})], \quad \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, y) = \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\cdot \,|\, \boldsymbol{x})}[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(y \,|\, \boldsymbol{x}, \boldsymbol{z})], \tag{4}$$

which are both estimated via Monte Carlo integration, using a single $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z} \,|\, \boldsymbol{x})$ for each $\boldsymbol{x}$.

Now, we consider the gradient of $\mathcal{L}$ with respect to $\boldsymbol{\varphi}$, the components of $\boldsymbol{\phi} = \{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ not copied from the decoder. For Gaussian contextual dropout, we estimate the gradients via the reparameterization trick [21]. For $\boldsymbol{z}^l \sim N(\mathbf{1}, \sigma(\boldsymbol{\alpha}^l)/(1-\sigma(\boldsymbol{\alpha}^l)))$, we rewrite it as $\boldsymbol{z}^l = 1 + \sqrt{\sigma(\boldsymbol{\alpha}^l)/(1-\sigma(\boldsymbol{\alpha}^l))}\boldsymbol{\epsilon}^l$, where $\boldsymbol{\epsilon}^l \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Similarly, sampling a sequence of $\boldsymbol{z} = \{\boldsymbol{z}^l\}_{l=1}^L$ from $q_{\boldsymbol{\phi}}(\boldsymbol{z} \,|\, \boldsymbol{x})$ can be rewritten as $f_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x})$, where $f_{\boldsymbol{\phi}}$ is a deterministic differentiable mapping and $\boldsymbol{\epsilon}$ are iid standard Gaussian. The gradient $\nabla_{\boldsymbol{\varphi}} \mathcal{L}(\boldsymbol{x}, y)$ can now be expressed as (see pseudo code in Appendix Algorithm 3)

$$\nabla_{\boldsymbol{\varphi}} \mathcal{L}(\boldsymbol{x}, y) = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{1})}[\nabla_{\boldsymbol{\varphi}}(\log p_{\boldsymbol{\theta}}(y \,|\, \boldsymbol{x}, f_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x})) - \tfrac{\log q_{\boldsymbol{\phi}}(f_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x})|\boldsymbol{x})}{\log p_{\boldsymbol{\eta}}(f_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{x}))})]. \tag{5}$$

For Bernoulli contextual dropout, backpropagating the gradient efficiently is not straightforward, as the Bernoulli distribution is not reparameterizable, restricting the use of the reparameterization trick. In this case, a commonly used gradient estimator is the REINFORCE estimator [28] (see details in Appendix A). This estimator, however, is known to have high Monte Carlo estimation variance. To this end, we estimate $\nabla_{\boldsymbol{\varphi}} \mathcal{L}$ with the augment-REINFORCE-merge (ARM) estimator [29], which provides unbiased and low-variance gradients for the parameters of Bernoulli distributions. We defer the details of this estimator to Appendix A.

At the testing stage, to obtain a point estimate, we follow the common practice in dropout [11] to multiply the neurons by the expected values of random dropout masks, which means we predict y with $p_{\boldsymbol{\theta}}(y \mid \boldsymbol{x}, \bar{\boldsymbol{z}})$, where $\bar{\boldsymbol{z}} = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})}[\boldsymbol{z}]$ under the proposed contextual dropout. When uncertainty estimation is needed, we draw K random dropout masks to approximate the posterior predictive distribution of y given $\boldsymbol{x}$ using $\hat{p}(y \mid \boldsymbol{x}) = \frac{1}{K} \sum_{k=1}^{K} p_{\boldsymbol{\theta}}(y \mid \boldsymbol{x}, \boldsymbol{z}^{(k)})$, where $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(K)} \overset{iid}{\sim} q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$. Note for uncertainty estimation on y, unlike conventional Bayesian NNs requiring instantiate multiple sets of NN weights, here one only needs multiple sets of random dropout masks, with the NN weights $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ being kept the same, leading to much lower memory and computational costs.

2.4 Related work

There are related works that also use data-dependent variational posteriors [30, 31]. Deng et al. [30] model attentions as latent-alignment variables and optimize a tighter lower bound (compared to hard attention) using a learned inference network. To balance exploration and exploitation for contexual bandits problems, Wang and Zhou [31] introduce local variable uncertainty under the Thompson sampling framework. However, their inference networks of are both independent of the decoder, which may considerably increase memory and computational cost for the considered applications. In addition, while the scope of Deng et al. [30] is limited to attention units and that of Wang and Zhou [31] limited to contextual bandits, we demonstrate the general applicability of contextual dropout to fully connected, convolutional, and attention layers in supervised learning models.

When applied to convolutional layers, contextual dropout can be viewed as a variational binary version of squeeze-and-excitation block [24], which has proven to be an effective unit to boost accuracy at minimal additional computational cost in computer vision tasks, especially in image classification [32, 33]. However, in contrast to the deterministic characteristic of squeeze-and-excitation, contextual dropout is a probabilistic inference unit that is able to produce uncertainty.

Conditional computation [34–37] is an area trying to increase model capacity without a proportional increase in computation, where an independent gating network decides turning which part of a network active and which inactive for each example. In contextual dropout, the encoder works much like a gating network choosing the distribution of sub-networks for each sample. But the potential gain in model capacity is even larger, $e.g.$, there are potentially $\sim O((2^d)^L)$ combinations of nodes for L fully-connected layers, where d is the scale of the number of nodes for one layer.

3 Experiments

We evaluate contextual dropout on three representative types of NN layers: fully connected, convolutional, and attention layers. We apply contextual dropout to the fully connected layers in multi-layer perceptrons (MLPs) [38], and to the convolutional layers in wide residual networks (WRNs) [39]. Both networks are evaluated on image classification tasks. Further, we apply contextual dropout to the attention layers of modular co-attention network (MCAN) [27], an attention-based state-of-the-art model for the task of visual question answering (VQA). The added computation and memory of contextual dropout is insignificant (see model size comparison for MLP, WRN, and MCAN in Table 4 in Appendix). We conduct experiments on MNIST [40] with MLP, CIFAR 10 and 100 [41] with WRN, and VQA-v2 [42] with MCAN. The training data size ranges from 60k to 444k. To investigate the model's robustness to noise, we also construct a noisy version for each dataset by adding Gaussian noises to image inputs [43]. All experiments are conducted using a single Nvidia Tesla V100 GPU.

For evaluation, we consider both the accuracy and uncertainty on predicting y given $\boldsymbol{x}$. Many metrics have been proposed to evaluate the quality of uncertainty estimation. On one hand, researchers are generating calibrated probability estimates to measure model confidence [44–46]. While expected calibration error and maximum calibration error have been proposed to quantitatively measure calibration, such metrics do not reflect how robust the probabilities are with noise injected into the network input, and cannot capture epistemic or model uncertainty [13]. On the other hand, the entropy of the predictive distribution as well as the mutual information, between the predictive distribution and posterior over network weights, are used as metrics to capture both epistemic and aleatoric uncertainty [47]. However, it is often unclear how large the entropy or mutual information is large enough to be classified as uncertain, so such metric only provides a relative uncertainty measure.

Hypothesis testing based uncertainty estimation: Unlike previous information theoretic metrics, we use a statistical test based method to estimate uncertainty, which works for both single-label and multi-label classification models. One advantage of using hypothesis testing over information theoretic metrics is that the p-value of the test can be more interpretable, making it easier to be deployed in practice to obtain a binary uncertainty decision. To quantify how confident our model is about this prediction, we evaluate whether the difference between the empirical distributions of the two most possible classes from multiple posterior samples is statistically significant. Please see Appendix D for a detailed explanation of the test procedure.

Uncertainty evaluation via PAvPU: With the p-value of the testing result and a given p-value threshold, we can determine whether the model is certain or uncertain about one prediction. To evaluate the uncertainty estimates, we uses Patch Accuracy vs Patch Uncertainty (PAvPU) [47], which is defined as $\text{PAvPU} = (n_{ac} + n_{iu})/(n_{ac} + n_{au} + n_{ic} + n_{iu})$, where $n_{ac}, n_{au}, n_{ic}, n_{iu}$ are the numbers of accurate and certain, accurate and uncertain, inaccurate and certain, inaccurate and uncertain samples, respectively. This PAvPU evaluation metric would be higher if the model tends to generate the accurate prediction with high certainty and inaccurate prediction with high uncertainty.

3.1 Contextual dropout on fully connected layers

We consider an MLP with two hidden layers of size 300 and 100, respectively, with ReLU activations. Dropout is applied to the input layer and the outputs of first two full-connected layers. We use MNIST as the benchmark. We compare contextual dropout with MC dropout [13], concrete dropout [15], Gaussian dropout [11], and Bayes by Backprop [7]. For hyperparameter tuning, we hold out $10,000$ samples randomly selected from the training set for validation. We use the chosen hyperparameters to train on the full training set ($60,000$ samples) and evaluate on the testing set ($10,000$ samples). Please see the complete hyperparameter setting in Appendix C.1.

Table 1: Results on MNIST with MLP.

	Original Data			Noisy Data		
	Accuracy	PAvPU(0.05)	log likelihood	Accuracy	PAvPU(0.05)	log likelihood
MC - Bernoulli	98.62±0.05	98.39±0.09	-1.4840 ±0.0004	86.36±0.19	85.63±0.31	-1.72 ±0.01
MC - Gaussian	98.67±0.04	98.41 ±0.04	-1.4820±0.0003	86.31±0.36	85.64±0.49	-1.72±0.01
Concrete	98.61±0.06	98.50±0.16	-1.4822 ±0.0012	86.52±0.35	86.77±0.23	-1.68 ±0.01
Bayes By Backprop	98.44±0.04	98.42±0.07	-1.4806 ±0.0007	86.55±0.37	87.13±0.31	-2.30 ±0.01
Bernoulli Contextual	**98.69**±0.04	98.50±0.08	-1.4816 ±0.0005	**87.06**±0.39	87.25±0.23	**-1.65** ±0.01
Gaussian Contextual	98.68±0.09	**98.57** ±0.08	**-1.4786**±0.0005	87.05±0.33	**87.61**±0.29	-1.66±0.01

Results and analysis: In Table 1, we show accuracy, PAvPU (p-value threshold equal to 0.05 and test predictive loglikelihood with error bars (5 random runs) for models with different dropouts under the original data and noisy data (added Gaussian noise with mean 0, variance 1). Note that we find the uncertainty results for p-value threshold 0.05 is in general consistent with the results for other p-value thresholds (see more in Table 5 in Appendix). We observe that contextual dropout outperforms other methods in all metrics, especially on the more challenging noisy data. Moreover, compared to Bayes by Backprop, contextual dropout is more memory and computationally efficient. As shown in Table 4 in Appendix, contextual dropout only introduces 16% additional parameters in this case. However, Bayes by Backprop doubles the memory and increases the computations significantly as we need multiple draws of NN weights for uncertainty estimation. Due to this reason, we do not include it for comparison for the following large model evaluations.

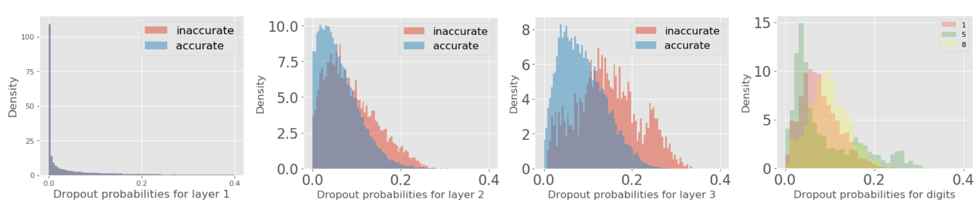

Figure 3: Visualization of dropout probabilities of Bernoulli contextual on the MNIST dataset.

Bernoulli contextual dropout probabilities visualization: In Figure 3, we visualize the probability distributions of Bernoulli contextual dropout. We observe that the learned dropout probabilities seem to increase as we go to higher-level layers, as also observed in Gal et al. [15]. Also, with contextual dropout, different samples own different dropout probabilities. Inaccurate ones often have higher dropout probabilities corresponding to higher uncertainties, which confirms our intuition. Further, we compare the dropout distributions across 3 representative digits. The dropout probabilities are overall higher for digit 8 compared to digit 1, meaning 1 is easier to classify. The distribution for 5 has longer tails than others showing there are more variations in the uncertainty for digit 5.

Combine contextual dropout with Deep Ensemble: Deep ensemble proposed by Lakshminarayanan et al. [48] is a simple way to obtain uncertainty by ensembling models trained independently from different random initializations. In Figure 4, we show the performance of combining different dropouts with deep ensemble on noisy MNIST data. We observe that as the number of NNs increases, both accuracy and PAvPU increase for all dropouts. However, Bernoulli contextual dropout outperforms both MC-bernoulli and concrete dropouts by a large margin in both metrics, showing contextual dropout is compatible with deep ensemble and their combination can lead to significant improvement.

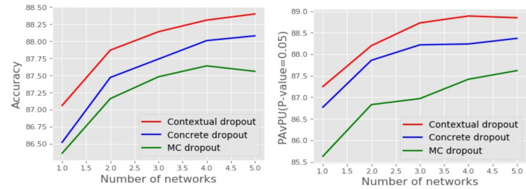

Figure 4: Combine dropouts with deep ensemble.

3.2 Contextual dropout on convolutional layers

We apply dropout to the convolutional layers in WRN [39]. In Figure 6 in Appendix, we show the architecture of WRN, where dropout is applied to the first convolutional layer in each network block; in total, dropout is applied to 12 convolutional layers. We use CIFAR-10 and CIFAR-100 [41] as benchmarks. The setting of hyperparameters is provided in Appendix C.1.

Table 2: Results on CIFAR-10 and CIFAR-100 with WRN.

Dropout	CIFAR-10 Original Data		CIFAR-10 Noisy Data		CIFAR-100 Original Data		CIFAR-100 Noisy Data	
	Accuracy	PAvPU (0.05)	Accuracy	PAvPU (0.05)	Accuracy	PAvPU (0.05)	Accuracy	PAvPU (0.05)
Bernoulli	94.58±0.19	82.34±2.33	79.51±0.18	74.43±0.77	79.03±0.17	61.54±0.25	52.01±0.45	54.25±0.80
Gaussian	93.81±0.06	93.84±0.21	79.33±0.37	80.15±0.36	76.63±0.11	78.05±0.10	51.38±0.24	57.02±0.23
Concrete	94.60±0.23	78.41±1.52	79.34±0.38	73.89±0.84	79.19±0.29	64.14±0.37	51.58±0.20	56.61±0.50
Bernoulli Contextual	94.94±0.10	94.96±0.12	79.40±0.19	80.53±0.31	79.08±0.05	80.59±0.31	51.42±0.45	**58.45**±0.50
Gaussian Contextual	**95.02**±0.10	**95.05**±0.07	**79.64**±0.31	**81.04**±0.36	**79.43**±0.18	**80.90**±0.16	**52.36**±0.34	57.72±0.43

Results and analysis: We show the results for CIFAR-10 and CIFAR-100 in Table 2 (see complete results in Tables 6 and 7 in Appendix). Accuracies and PAvPUs are incorporated for both the original and noisy data (see test predictive loglikelihood in Appendix Table 8). We consistently observe contextual dropout outperforms other models in accuracy, uncertainty estimation, and loglikelihood.

Uncertainty Visualization on CIFAR-10: In Figures 13-15 in Appendix F.2, we visualize 15 CIFAR-10 images (with true label) and compare the corresponding probability outputs of different dropouts in boxplots. As most samples in CIFAR-10 are not difficult to classify, we only visually inspect challenging ones. We observe that contextual dropout predicts the correct answer if it is certain, and it is certain and predicts the correct answers on many images for which MC or concrete dropout is uncertain. In addition, MC or concrete dropout is uncertain about some easy examples or certain on some wrong predictions (see details in Appendix F.2). Moreover, on an image that all three methods have high uncertainty, concrete dropout places a higher probability on the correct answer than the other two. These observations verify that contextual dropout provides better calibrated uncertainty.

3.3 Contextual dropout on attention layers

We further apply contextual dropout to the attention layers of VQA models, whose goal is to provide an answer to a question relevant to the content of a given image.

Dataset and evaluation: We conduct experiments on the commonly used VQA benchmark, VQA-v2 [42], which contains human-annotated question-answer (QA) pairs for images from MS-COCO dataset [49]. There are three types of questions: Yes/No, Number, and Other. In Figure 5, we show one example for each question type. There are 10 answers provided by 10 different human annotators for each question. As shown in the examples, VQA is generally so challenging that there are often several different human annotations for a given image. Therefore, good uncertainty estimation becomes even more necessary. Moreover, the evaluation for VQA is different from image classification. The accuracy for a single answer could be a number between 0 and 1 [42]: $\mathrm{Acc}(ans) = \min\{(\#\text{human that said } ans)/3, 1\}$. We generalize the uncertainty evaluation accordingly:

$$n_{ac} = \sum_i \mathrm{Acc}_i \mathrm{Cer}_i,\ n_{iu} = \sum_i (1 - \mathrm{Acc}_i)(1 - \mathrm{Cer}_i)\,, n_{au} = \sum_i \mathrm{Acc}_i (1 - \mathrm{Cer}_i),\ n_{ic} = \sum_i (1 - \mathrm{Acc}_i)(\mathrm{Cer}_i)$$

where for the ith prediction Acc_i is the accuracy and $\mathrm{Cer}_i \in \{0, 1\}$ is the certainty indicator.

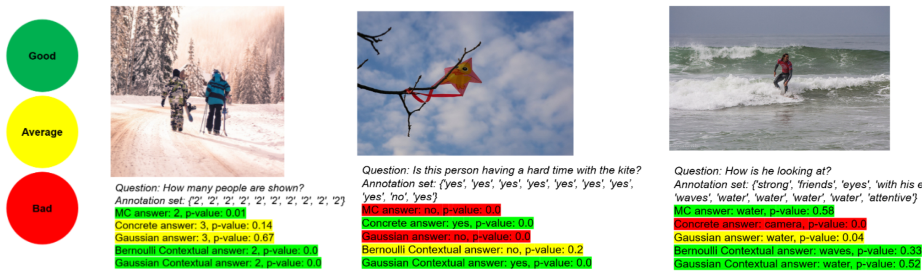

Figure 5: VQA visualization: for each question type (Num, Yes/No, Others), we present an image-question pair along with human annotations. We show the predictions and uncertainty estimates of different dropouts and highlight the good, average, and bad answers with green, yellow, and red, respectively.

Model and training specifications: We use MCAN [27], an attention-based state-of-the-art model for VQA. Bottom-up features extracted from images by Faster R-CNN [50] are used as visual features. Pretrained word-embeddings [51] and LSTM [52] are used to extract question features. Then, self-attention layers for question features and visual features, as well as the question-guided attention layers of visual features, are stacked one over another to build a deep MCAN. We adopt the encoder-decoder structure in MCAN with six co-attention layers. Dropout is applied in every attention layer (after the softmax and before residual layer [26]) and fully-connected layer to prevent overfitting [27], resulting in 62 layers in total with dropout. We follow the same model hyperparameters and training settings in Yu et al. [27] (also see details in Appendix C.2).

Table 3: Accuracy and PAvPU on visual question answering

Dropout	Accuracy		PAvPU	
	Original Data	Noisy Data	Original Data	Noisy Data
MC - Bernoulli	66.95	61.45	70.04	66.11
MC - Gaussian	66.96	62.75	70.77	67.42
Concrete	66.82	61.47	71.02	65.94
Bernoulli Contextual	**67.19**$\pm$0.06	63.06$\pm$0.08	**71.26**$\pm$0.06	67.41$\pm$0.11
Gaussian Contextual	66.97	**63.54**	71.15	**67.49**

Results and analysis: As in image classification, we compare different dropouts on both the original VQA dataset and a noisy version, where we add Gaussian noise with standard deviation 5 to the visual features. In Tables 3, we show the overall accuracy and uncertainty estimation of each method on both the original and noisy data (see complete results of per-type accuracy and uncertainty for each of three question types of in Tables 9-11 in Appendix E). The results show that on the original data, contextual dropout achieves better accuracy and uncertainty estimation than the other two. Moreover, on noisy data, where the prediction becomes more challenging and requires more model flexibility and robustness, contextual dropouts outperform their regular dropout counterparts by a large margin in terms of accuracy, and the improvement is consistent across all three question types.

Visualization: In Figures 16-19 in Appendix F.3, we visualize some image-question pairs, along with the human annotations and compare the predictions and uncertainty estimations of different dropouts. We show three of them in Figure 5. We manually classify each prediction by different methods based on their answers and p-values. For questions that have a clear answer, we define the good as certain & accurate, the average as uncertain & accurate or uncertain & inaccurate, and the bad as certain & inaccurate. Otherwise, we define the good as uncertain & accurate, the average as certain & accurate or uncertain & inaccurate, and the bad as certain & inaccurate. As shown in the plots, overall contextual dropout is more conservative on its wrong predictions and more certain on its correct predictions than other methods(see more detailed explanations in Appendix F.3).

4 Conclusion

We propose contextual dropout with dropout probabilities dependent on the covariates in its variational inference network. We show contextual dropout masks can be defined using either the Bernoulli or Gaussian distribution. With an efficient parameterization of the coviariate-dependent variational distribution, contextual dropout boosts the flexibility of Bayesian neural networks and enables the model to better estimate uncertainty, at the expense of only slightly increased memory and computational cost. We demonstrate the general applicability of contextual dropout on fully connected, convolutional, and attention layers. On both image classification and visual question answering tasks, we verify that contextual dropout improves both accuracy and quality of uncertainty estimation.

Broader Impact

Deep learning systems have been or have the potential to be adopted in a wide range of domains to help activities of our daily living, such as self-driving [53], healthcare [54], and robotics [55]. Such systems could greatly benefit our daily life and liberate us from repeating labors. However, deploying these systems in real life is challenging. One of the main challenges is that deep learning systems can be over-confident on its predictions, and unaware of its mistakes, which could significantly restrict their usage as making mistakes in real life could lead to catastrophic events. Our work could greatly enhance models' capacity and capability to better estimate uncertainty. Unlike previous work, such as deep ensemble or Bayes by Backprop, our work introduces little extra computational or memory cost. While we show improvements by our work on image classification and visual question answering, our framework is general enough that it could be used to improve potentially any supervised learning models. In this regard, our work could mitigate the general over-confidence issue of any deep learning systems, and help to build mistake-aware and efficient deep learning systems, knowing when to ask for human-aid if needed.

We, as human beings, make mistakes, and machines also do (even though we are constantly improving them). We have built a legal systems for accountability when human makes mistakes. But what if machines make mistakes? Closely related to the visual question answering (VQA) task considered in this paper, we can ask even more specific questions: 1) What if a machine makes a wrong decision with grave consequences, such as significant property damages or injuries, when its algorithm is CERTAIN it is making a right decision? 2) What if a machine makes a wrong decision with grave consequences when its algorithm is very UNCERTAIN it is making a right decision? In these two different unfortunate scenarios, what would be the differences of the implied legal liabilities for the company that designs the machine, and the individual that is operating the machine? These are important questions to ask, but an important step before these questions can be meaningful addressed is probably to have a Bayesian deep learning model that provides high quality uncertainty estimation. We hope the proposed contextual dropout can help enhance the quality of uncertainty estimation in a wide variety of domains, especially avoiding the possibility of making a decision with high certainty that endangers workers and jeopardizes public safety.

References

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[2] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4 (3):448–472, 1992.

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

[4] Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.

[5] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

[6] Chunyuan Li, Changyou Chen, David Carlson, and Lawrence Carin. Preconditioned stochastic gradient Langevin dynamics for deep neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[7] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

[8] Christos Louizos and Max Welling. Multiplicative normalizing flows for variational Bayesian neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2218–2227. JMLR. org, 2017.

[9] Jiaxin Shi, Shengyang Sun, and Jun Zhu. Kernel implicit variational inference. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1l4eQW0Z.

[10] 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.

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

[12] Sida Wang and Christopher Manning. Fast dropout training. In *international conference on machine learning*, pages 118–126, 2013.

[13] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[14] Shahin Boluki, Randy Ardywibowo, Siamak Zamani Dadaneh, Mingyuan Zhou, and Xiaoning Qian. Learnable Bernoulli dropout for Bayesian deep learning. In *Artificial Intelligence and Statistics*, 2020.

[15] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *Advances in Neural Information Processing Systems*, pages 3581–3590, 2017.

[16] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 2015.

[17] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2498–2507. JMLR. org, 2017.

[18] Jimmy Ba and Brendan Frey. Adaptive dropout for training deep neural networks. In *Advances in Neural Information Processing Systems*, pages 3084–3092, 2013.

[19] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[20] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

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

[22] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pages 1278–1286, 2014.

[23] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656, 2015.

[24] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[25] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[27] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6281–6290, 2019.

[28] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer, 1992.

[29] Mingzhang Yin and Mingyuan Zhou. ARM: Augment-REINFORCE-merge gradient for discrete latent variable models. *Preprint*, May 2018.

[30] Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander Rush. Latent alignment and variational attention. In *Advances in Neural Information Processing Systems*, pages 9712–9724, 2018.

[31] Zhendong Wang and Mingyuan Zhou. Thompson sampling via local uncertainty. *arXiv preprint arXiv:1910.13673*, 2019.

[32] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[33] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. Mnasnet: Platform-aware neural architecture search for mobile. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[34] Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*, 2015.

[35] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[36] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[37] Ravi Teja Mullapudi, William R Mark, Noam Shazeer, and Kayvon Fatahalian. Hydranets: Specialized dynamic architectures for efficient inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8080–8089, 2018.

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

[39] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[40] Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2:18, 2010.

[41] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[42] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017.

[43] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pages 473–480, 2007.

[44] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

[45] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[46] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. *arXiv preprint arXiv:1807.00263*, 2018.

[47] Jishnu Mukhoti and Yarin Gal. Evaluating bayesian deep learning methods for semantic segmentation. *arXiv preprint arXiv:1811.12709*, 2018.

[48] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[49] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[50] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[51] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[52] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[53] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[54] Igbe Tobore, Jingzhen Li, Liu Yuhang, Yousef Al-Handarish, Abhishek Kandwal, Zedong Nie, and Lei Wang. Deep learning intervention for health care challenges: Some biomedical domain considerations. *JMIR mHealth and uHealth*, 7(8):e11966, 2019.

[55] Harry A Pierson and Michael S Gashler. Deep learning in robotics: a review of recent research. *Advanced Robotics*, 31(16):821–835, 2017.

[56] Yang Li and Shihao Ji. L0-ARM: Network sparsification via stochastic binary optimization. In *The European Conference on Machine Learning (ECML)*, 2019.

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

[58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[59] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[60] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[61] Damien Teney, Peter Anderson, Xiaodong He, and Anton Van Den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4223–4232, 2018.

[62] Asghar Ghasemi and Saleh Zahediasl. Normality tests for statistical analysis: a guide for non-statisticians. *International journal of endocrinology and metabolism*, 10(2):486, 2012.