DEEP WEIGHT PRIOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Bayesian inference is known to provide a general framework for incorporating prior knowledge or specific properties into machine learning models via carefully choosing a prior distribution. In this work, we propose a new type of prior distributions for convolutional neural networks, deep weight prior, that in contrast to previously published techniques, favors empirically estimated structure of convolutional filters e.g., spatial correlations of weights. We define deep weight prior as an implicit distribution and propose a method for variational inference with such type of implicit priors. In experiments, we show that deep weight priors can improve the performance of Bayesian neural networks on several problems when training data is limited. Also, we found that initialization of weights of conventional convolutional networks with samples from deep weight prior leads to faster training.

1 INTRODUCTION

Bayesian inference is a tool that, after observing training data, transforms a *prior distribution* over parameters of machine learning models to posterior distribution. Recently, stochastic variational inference (Hoffman et al., 2013) – a method for approximate Bayesian inference – has been successfully adopted to obtain *variational approximation* of posterior distribution over weights of a deep neural network (Kingma et al., 2015). Currently, there are two major directions for the development of Bayesian deep learning. The first direction can be summarized as the improvement of approximate inference with richer variational approximations and tighter variational bounds (Louizos & Welling, 2017; Dikmen et al., 2015). The second direction is the design of probabilistic models, in particular weight prior distributions, that widen the scope of applicability of the Bayesian approach.

It has been shown that prior distribution plays an important role for model sparsification (Molchanov et al., 2017; Neklyudov et al., 2017), quantization (Ullrich et al., 2017) and compression (Louizos et al., 2017). Although these prior distributions proved to be helpful, they are limited to fully-factorized distributions. Thus, the often observed spatial structure of convolutional filters cannot be enforced with such priors. Convolutional neural networks are an example of the model family, where a correlation of the weights plays an important role, thus it may benefit from a more flexible prior distributions.

In this work, based on the fact that convolutional neural networks tend to learn the similar convolutional kernels on different datasets from similar domains (Sharif Razavian et al., 2014; Yosinski et al., 2014), we propose a method for building prior distributions for convolutional neural networks. For a specific data domain, we consider the distribution of kernels of a trained convolutional network. In the rest of the paper, we refer to this distribution as the *source kernel distribution*. Our main assumption is that within a specific domain the source kernel distribution can be efficiently approximated with kernel distributions of models trained on a small subset of tasks from the specific domain. For example, we expect that a good approximation for the kernel distribution models trained on NotMNIST – a dataset of grayscale images, will be a good approximation for kernel distribution of MNIST trained models. However, some difficulties appear due to an inability to access the density function of the source kernel distribution, in Section 3 we propose a technique to overcome this problem. Our contributions can be summarized as follows:

1. We propose *deep weight prior*, a framework that approximates the *source kernel distribution* and incorporates prior knowledge about the structure of convolutional filters into the prior distribution. We also propose to use an implicit form of this prior (Section 3.1).

2. We develop a method for variational inference with the specific type of implicit priors (Section 3.2).
3. In experiments (Section 4), we show that variational inference with *deep weight prior* significantly improves classification performance upon a number of popular prior distributions in the case of limited training data. We also find that just initialization of the conventional convolution networks with samples from deep weight prior leads to faster convergence and better random feature extraction.

2 DEEP BAYES

In a Bayesian setting, after observing a dataset $\mathcal{D} = \{x_1, \dots, x_N\}$ of N points, the goal is to transform our prior knowledge $p(\omega)$ of the unobserved distribution parameters ω to the posterior distribution $p(\omega \mid \mathcal{D})$. However, computing a posterior distribution through the Bayes rule $p(\omega \mid \mathcal{D}) = p(\mathcal{D} \mid \omega)p(\omega)/p(\mathcal{D})$ may involves computationally intractable integrals. This problem, nonetheless, can be solved approximately.

Variational Inference (Jordan et al., 1999) is one of such approximation methods. It reduces the inference to an optimization problem, where we optimize parameters θ of a variational approximation $q_\theta(\omega)$, so that KL-divergence between $q_\theta(\omega)$ and $p(\omega \mid \mathcal{D})$ is minimized. This divergence in practice is minimized by maximizing so-called *variational lower bound* $\mathcal{L}(\theta)$ of the marginal log-likelihood of the data w.r.t parameters θ of the variational approximation $q_\theta(W)$.

$$\mathcal{L}(\theta) = L_{\mathcal{D}} - D_{\mathrm{KL}}(q_\theta(\omega) \| p(\omega)) \to \max_\theta \tag{1}$$

$$\text{where} \quad L_{\mathcal{D}} = \mathbb{E}_{q_\theta(\omega)} \log p(D \mid \omega) \tag{2}$$

The variational lower bound $\mathcal{L}(\theta)$ consists of two terms (conditional) *expected log likelihood* $L_{\mathcal{D}}$ and regularizer $D_{\mathrm{KL}}(q_\theta(\omega) \| p(\omega))$. Since $\log p(\mathcal{D}) = \mathcal{L}(\theta) + D_{\mathrm{KL}}(q_\theta(\omega) \| p(\omega \mid \mathcal{D}))$ and $p(\mathcal{D})$ does not depend on $q_\theta(w)$ maximizing of $\mathcal{L}(\theta)$ minimizes $D_{\mathrm{KL}}(q_\theta(\omega) \| p(\omega \mid \mathcal{D}))$. However, in case of intractable expectations in equation 1 neither the variational lower bound $\mathcal{L}(\theta)$ nor its gradients can be computed in close form.

Recently, Kingma & Welling (2013) and Rezende et al. (2014) proposed an efficient mini-batch based approach to stochastic variational inference, so-called *stochastic gradient variational Bayes* or *doubly stochastic variational inference*. The idea behind this framework is reparamtetrization, that represents samples from parametric distribution $q_\theta(\omega)$ as a deterministic differentiable function $\omega = f(\theta, \epsilon)$ of parameters θ and an (auxiliary) noise variable $\epsilon \sim p(\epsilon)$. Using this trick we can efficiently compute an unbiased stochastic gradient $\nabla_\theta \mathcal{L}$ of variational lower bound w.r.t parameters of variational approximation (Kingma & Welling, 2013).

2.1 BAYESIAN NEURAL NETWORKS

Stochastic gradient variational Bayes framework has been applied to approximate posterior distributions over parameters of deep neural networks (Kingma et al., 2015). We consider a discriminative problem, where dataset $\mathcal{D}$ consists of N object-label pairs $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$. For this problem we maximize the variational lower bound $\mathcal{L}(\theta)$ with respect to parameters θ of variational approximation:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \mathbb{E}_{q_\theta(W)} \log p(y_i \mid x_i, W) - D_{\mathrm{KL}}(q_\theta(W) \| p(W)) \to \max_\theta \tag{3}$$

where, W denotes weights of neural network, $q_\theta(W)$ is a variational distribution, that allows reparametrization (Kingma & Welling, 2013; Figurnov et al., 2018), $p(W)$ is a prior distribution. In a simple case $q_\theta(W)$ can be fully-factorized normal distribution, however, more expressive variational approximations may lead to better quality of variational inference (Louizos & Welling, 2017; Yin & Zhou, 2018). Typically, Bayesian neural networks use fully-factorized normal or log-uniform prior distributions (Kingma et al., 2015; Molchanov et al., 2017; Louizos & Welling, 2017). The *local reparametrization trick* proposed by Kingma et al. (2015) replaces sampling of weights with sampling of activations before non-linearity, that allows to reduce the variance of the gradient for a number of variational approximations.

2.2 VARIATIONAL AUTO-ENCODER

Stochastic gradient variational Bayes has also been applied for building generative models. Variational auto-encoder proposed by Kingma & Welling (2013) maximizes a variational lower bound $\mathcal{L}(\theta, \phi)$ on the marginal log-likelihood by amortized variational inference:

$$\mathcal{L}(\theta, \phi) = \sum_{i=1}^{N} \mathbb{E}_{q_\theta(z_i \,|\, x_i)} \log p_\phi(x_i \,|\, z_i) - D_{\mathrm{KL}}(q_\theta(z_i \,|\, x_i) \| p(z_i)) \to \max_{\theta,\phi}, \tag{4}$$

where, *inference model* $q_\theta(z_i \,|\, x_i)$ approximates posterior distribution over local latent variables z_i, *reconstruction model* $p_\phi(x_i \,|\, z_i)$ transforms distribution over latent variables to conditional distribution in object space and a prior distribution over latent variables $p(z_i)$. The vanilla VAE defines $q_\theta(z \,|\, x), p_\phi(x \,|\, z), p(z)$ as fully-factorized distributions, however, a number of richer variational approximations and prior distributions have been proposed (Kingma et al., 2016; Tomczak & Welling, 2017). The approximation of the data distribution then can be defined as an intractable integral $p(x) \approx \int p_\phi(x \,|\, z) p(z)\, dz$ which we will refer to as an implicit distribution.

3 DEEP WEIGHT PRIOR

We propose *deep weight prior* – an adaptive prior distribution which allows us to encode the information about the interdependence of learned convolutional filters. In this section, we consider a neural network with L convolutional layers. We denote parameters of l-th convolutional layer as $w^l \in \mathbb{R}^{I_l \times O_l \times H_l \times W_l}$, where I_l is the number of input channels, O_l is the number of output channels, H_l and W_l are spatial dimensions of kernels. Parameters of the neural network are denoted as $W = (w^1, \dots w^L)$. We consider a variational approximation $q_\theta(W)$ and a prior distribution $p(W)$ with the following factorization over layers, filters and channels:

$$q_\theta(W) = \prod_{l=1}^{L} \prod_{i=1}^{I_l} \prod_{j=1}^{O_l} q(w_{ij}^l \,|\, \theta_{ij}^l) \qquad p(W) = \prod_{l=1}^{L} \prod_{i=1}^{I_l} \prod_{j=1}^{O_l} p_l(w_{ij}^l), \tag{5}$$

where $w_{ij}^l \in \mathbb{R}^{H_l \times W_l}$ is a kernel of j-th channel in i-th filter of l-th convolutional layer. We also assume that $q_\theta(W)$ allows reparametrization.

For a specific domain, we consider *source kernel distribution* – a distribution of trained convolutional kernels of l-th convolutional layer. The source kernel distribution is a very natural candidate to be a prior distribution $p_l(w_{ij}^l)$ for convolutional kernels of l-th layer. Unfortunately, we do not have an access to its probability density function. As a result, we cannot use this distribution for the Bayesian inference on a new problem from the same domain. We assume that within a specific data domain the source kernel distribution can be efficiently approximated with the distribution of kernels of a single trained model. In the next subsection, we propose to approximate this intractable probability density function of the source kernel distribution using a framework of generative models.

3.1 MODEL OF PRIOR DISTRIBUTION

In this section, we discuss explicit and implicit approximations $\hat{p}_l(w)$ of the probability density function $p_l(w)$ of the source kernel distribution of l-th layer. We assume to have a trained convolutional neural network, and treat kernels from l-th layer of this network $w_{ij}^l \in R^{H_l \times W_l}$ as samples from the source kernel distribution of l-th layer $p_l(w)$.

Explicit models. A number of approximations allow us to evaluate probability density functions explicitly. Such families include but are not limited to Kernel Density Estimation (Silverman, 1986), Normalizing Flows (Rezende & Mohamed, 2015; Dinh et al., 2017) and PixelCNN (van den Oord et al., 2016). For these families, we can estimate KL-divergence $D_{KL}(q(w \,|\, \theta_{ij}^l) \| \hat{p}_l(w))$ and its gradients without a systematic bias, and then use them for variational inference. Despite the fact that these methods provide flexible approximations, they usually demand high memory or computational cost (Louizos & Welling, 2017).

Implicit models. Implicit models, in contrast, can be more computationally efficient, however, they don't provide an access to an explicit form of probability density function $\hat{p}_l(w)$. We consider an

Algorithm 1 Stochastic Variational Inference With Implicit Prior Distribution

Require: dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$
Require: variational approximations $q(w \mid \theta_{ij}^l)$ and reverse models $r(z \mid w; \psi_l)$
Require: reconstruction models $p(w \mid z; \phi_l)$, priors for auxiliary variables $p_l(z)$
while not converged **do**
  $\hat{M} \leftarrow$ mini-batch of objects form dataset $\mathcal{D}$
  $\hat{w}_{ij}^l \leftarrow$ sample weights from $q(w|\theta_{ij}^l)$ with reparametrization
  $\hat{z}_{ij}^l \leftarrow$ sample auxiliary variables from $r(z \mid \hat{w}_{ij}^l; \psi_l)$ with reparametrization
  $\hat{\mathcal{L}}^{aux} \leftarrow L_{\hat{M}} + \sum_{l,i,j} -\log q(\hat{w}_{ij}^l \mid \theta_{ij}^l) - \log r(\hat{z}_{ij}^l \mid \hat{w}_{ij}^l; \psi_l) + \log p_l(\hat{z}_{ij}^l) + \log p(\hat{w}_{ij}^l \mid \hat{z}_{ij}^l; \phi_l)$
  Obtain unbiased estimate $\hat{g}$ with $\mathbb{E}[\hat{g}] = \nabla \mathcal{L}^{aux}$ by differentiating $\hat{\mathcal{L}}^{aux}$
  Update parameters θ and ψ using gradient $\hat{g}$ and a stochastic optimization algorithm
end while
return Parameters θ, ψ

approximation of the prior distribution $p_l(w)$ in the following implicit form:

$$\hat{p}_l(w) = \int p(w \mid z; \phi_l) p_l(z)\, dz, \tag{6}$$

where the conditional distribution $p(w \mid z; \phi_l)$ is an explicit parametric distribution and $p_l(z)$ is an explicit prior distribution that does not depend on trainable parameters. Parameters ϕ_l of the conditional distribution $p(w \mid z; \phi_l)$ can be modeled by a differentiable function $g(z; \phi_l)$ e.g. neural network. Note, that while the conditional distribution $p(w \mid z; \phi_l)$ usually is a simple explicit distribution, e.g. fully-factorized Gaussian, the marginal distribution $\hat{p}_l(w)$ is generally a more complex intractable distribution.

Parameters ϕ_l of the conditional distribution $p(w \mid z; \phi_l)$ can be fitted using the framework of variational auto-encoder. In contrast to the methods with explicit access to the probability density, variational auto-encoders combine low memory cost and fast sampling. However, we cannot compute the probability density function $\hat{p}_l(w)$ and therefore cannot build an unbiased estimation of the variational lower bound (equation 3). In order to overcome this limitation we propose a modification of variational inference for implicit prior distributions.

3.2 Variational Inference With Implicit Prior Distribution

Stochastic variational inference approximates true posterior distribution by maximizing the variational lower bound $\mathcal{L}(\theta)$ (equation 1), which includes KL-divergence $D_{\mathrm{KL}}(q(W) \| p(W))$ between a variational approximation $q_\theta(W)$ and a prior distribution $p(W)$. In case of simple prior and variational distributions (e.g. Gaussians), this KL-divergence can be computed in a closed form or estimated without a systematic bias. Unfortunately, it does not hold anymore in case of implicit prior distribution $\hat{p}(W) = \prod_{l,i,j} \hat{p}_l(w_{ij}^l)$. In this case, the KL-divergence becomes an intractable and cannot be unbiasedly estimated since the prior $\hat{p}(W)$ is intractable. However, we can make the variational lower bound tractable again by upper bounding the KL-divergence:

$$D_{\mathrm{KL}}(q(W) \| \hat{p}(W)) = \sum_{l,i,j} D_{\mathrm{KL}}(q(w_{ij}^l | \theta_{ij}^l) \| \hat{p}_l(w_{ij}^l)) \leq \sum_{l,i,j} \big(-H(q(w_{ij}^l \mid \theta_{ij}^l)) +$$
$$+ \mathbb{E}_{q(w_{ij}^l \mid \theta_{ij}^l)} \big[D_{\mathrm{KL}}(r(z \mid w_{ij}^l; \psi_l) \| p_l(z)) - \mathbb{E}_{r(z \mid w_{ij}^l; \psi_l)} \log p(w_{ij}^l \mid z; \phi_l) \big] \big) = D_{\mathrm{KL}}^{bound}, \tag{7}$$

where $r(z \mid w; \psi_l)$ is an auxiliary inference model for the prior of l-th layer $\hat{p}_l(w)$. The final auxiliary variational lower bound has a folowing form:

$$\mathcal{L}^{aux}(\theta, \psi) = L_D - D_{\mathrm{KL}}^{bound} \leq L_D - D_{\mathrm{KL}}(q_\theta(W) \| \hat{p}(W)) = \mathcal{L}(\theta) \tag{8}$$

In case when $q_\theta(w)$, $p(w \mid z; \phi_l)$ and $r(z \mid w; \psi_l)$ are explicit parametric distributions which can be reparametrized, we can perform an unbiased estimation of a gradient of the auxiliary variational lower bound $\mathcal{L}^{aux}(\theta, \psi)$ (equation 8) w.r.t. parameters θ of the variational approximation $q_\theta(W)$ and parameters ψ of reverse models $r(z \mid w; \psi_l)$. Then we can maximize the auxiliary lower bound

w.r.t. parameters of the variational approximation and reversed models $\mathcal{L}^{aux}(\theta, \psi) \to \max_{\theta,\psi}$. For more details see Appendix A. Note, that parameters ϕ of the prior distribution $\hat{p}(W)$ are fixed during variational inference, in contrast to Empirical Bayes framework (MacKay, 1992).

Algorithm 1 describes the stochastic variational inference with an implicit prior distribution. In case when we can calculate an entropy $H(q)$ or the divergence $D_{\mathrm{KL}}(r(z \,|\, w; \psi_l) \| p_l(z))$ explicitly, the variance of the estimation of the gradient $\nabla \hat{\mathcal{L}}^{aux}(\theta, \psi)$ can be reduced. This algorithm can also be applied to an implicit prior that is defined in the form of Markov chain:

$$\hat{p}(w) = \int dz_0 \dots dz_T p(w \,|\, z_T) p(z_0) \prod_{t=0}^{T-1} p(z_{t+1} \,|\, z_t), \tag{9}$$

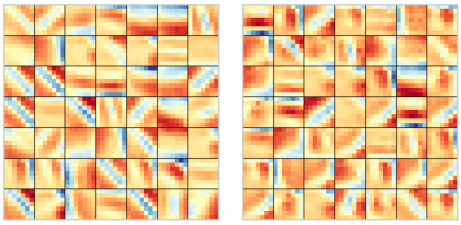
where $p(z_{t+1} \,|\, z_t)$ is transition operator (Salimans et al., 2015), see Appendix A. We provide more details related to the form of $p(w \,|\, z; \phi_l)$, $r(z \,|\, w; \psi_l)$ and $p_l(z)$ distributions in Section 4.

4 Experiments

We start the section with a detailed description of learning deep weight prior. Then we apply this prior to variational inference, random feature extraction and initialization for convolutional neural networks. In our experiments we used MNIST (LeCun et al., 1998), NotMNIST (Bulatov, 2011), CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009) datastets. Experiments were implemented using PyTorch (Paszke et al., 2017). For optimization we used Adam (Kingma & Ba, 2014) with default hyperparameters.

4.1 Learning Deep Weight Prior

In this subsection we explain how to train deep weight prior models for a particular problem. First, we collect source datasets of kernels from convolutional networks (source networks), that were trained on data from a similar domain. Then we train prior models (equation 6) on these collected datasets by using the framework of variational auto-encoder (Section 2.2). Examples of samples from learned prior distribution are presented at figure 1.

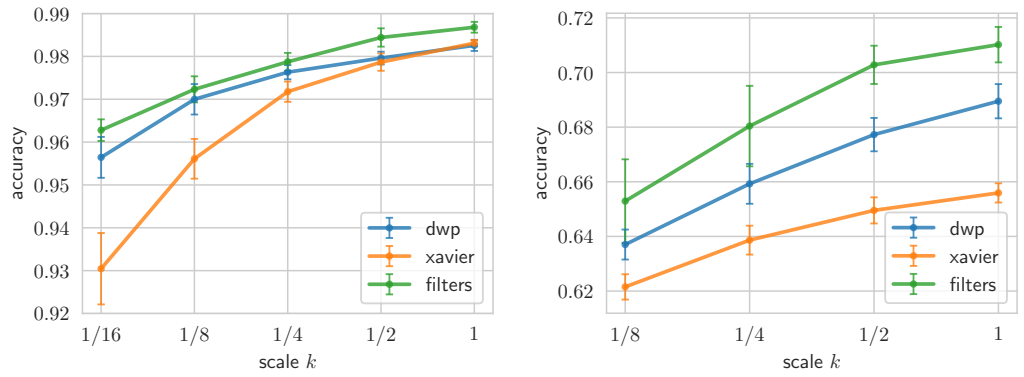


(a) Learned filters (b) Samples from DWP

Figure 1: At subfig. 1(a) we show a batch of learned kernels of shape 7×7 form the first convolutional layer of CNN trained on NotMNIST dataset, at subfig. 1(b) we show samples form deep weight prior that was learned on these kernels.

Source datasets of kernels. For kernels of particular convolutional layer l, we use an individual prior distribution $\hat{p}_l(w) = \int p(w \,|\, z; \phi_l) p_l(z)\, dz$, where reconstruction model $p(w \,|\, z; \phi_l)$ was learned on a separate dataset of learned kernels. In our experiments, we found that regularization is crucial for a learning of source kernels, it helps to learn more structured and less noisy kernels. Thus, source models were learned with L_2 regularization. We removed kernels of small norm as they make no influence to predictions Molchanov et al. (2017), but they make learning of a generative model more challenging. We trained prior distributions on 10 source networks witch were learned from different initial points on NotMNIST and CIFAR-100 datasets.

Reconstruction and inference models for prior distribution. In our experiments, inference models $r(z \,|\, w; \psi_l)$ are fully-factorized normal-distributions $\mathcal{N}(z \,|\, \mu_{\psi_l}(w), \mathrm{diag}(\sigma^2_{\psi_l}(w)))$, where parameters $\mu_{\psi_l}(w)$ and $\sigma_{\psi_l}(w)$ are modeled by convolutional neural network. Convolutional part of the network is constructed from several convolutional layers that are divided with ELU (Clevert et al., 2015) and max-pooling layers. Convolution layers are followed by a fully-connected layer with $2 \cdot z^l_{dim}$ output neurons, where z^l_{dim} is a dimension of hidden representation. Dimension z^l_{dim} of hidden space is specific for particular layer. Reconstruction models $p(w \,|\, z; \phi_l)$ are also modeled by a fully-factorized normal-distribution $\mathcal{N}(w \,|\, \mu_{\phi_l}(z), \mathrm{diag}(\sigma^2_{\phi_l}(z)))$ and networks μ_{ϕ_l} and $\sigma^2_{\phi_l}$ have the similar architectures, but use transposed convolutions. We use the same architectures for all prior models, but with slightly different hyperparameters, due to different sizes of kernels. We provide a more detailed description at Appendix D.



(a) Results for MNIST

(b) Results for CIFAR-10

Figure 2: We study an influence of initialization of convolutional filter on the performance of random feature extraction. In this experiment, weights of convolutions layers were initialized, with: samples from deep weight prior (dwp), learned filters (filters) and samples from Xavier distibution (xavier). A number of filters in all convolutional layers were scaled linearly by k. For every size of model results were averaged by 10 runs. We found that initialization with samples from deep weight prior and learned filters significantly outperform Xavier initialization. Although, initialization with filters performs marginally better, deep weight prior does not require to store potentially big dataset of all learned filters. We present result for MNIST dataset at subfig. 2(a) and on CIFAR-10 dataset at subfig. 2(b).

4.2 Random Feature Extraction

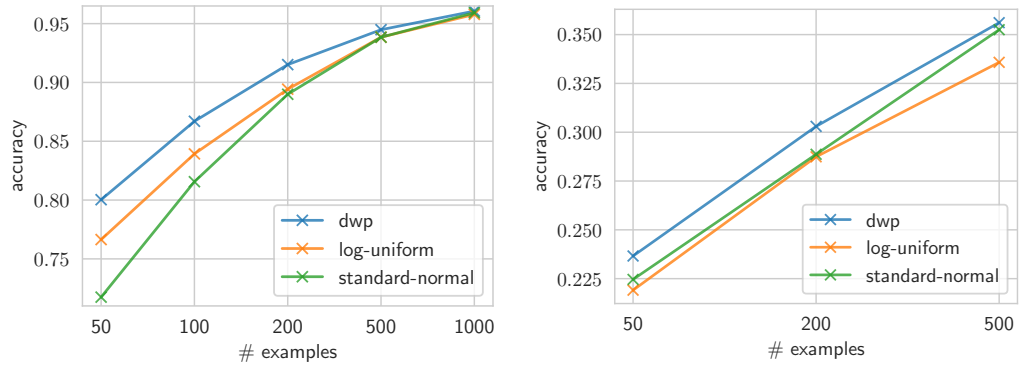
Convolutional neural networks produce useful features even if they are initialized randomly (Saxe et al., 2011; He et al., 2016; Ulyanov et al., 2017). In this experiment, we study an influence of different initializations on convolutional networks of different size. We use three initializations for weights of convolutional layers: learned kernels, samples from deep weight prior, samples from Xavier distribution (Glorot & Bengio, 2010).

On MNIST dataset we used a neural network with two convolutional layers with $32, 128$ filters of shape $7 \times 7,\ 5 \times 5$ respectively, followed by one linear layer with 10 neurons. On CIFAR dataset we used neural network with four convolutional layers with $128,\ 256,\ 256$ filters of shape $7 \times 7,\ 5 \times 5,\ 5 \times 5$ respectively, followed by two fully connected layers with 512 and 10 neurons. After the first convolutional layer max-pooling layer (Nagi et al., 2011) was applied. All layers were divided with leaky ReLU nonlinearities (Nair & Hinton, 2010).

We found that initializations with samples from deep weight prior and learned kernels significantly outperform the standard Xavier initialization when the size of the network is small. Initializations with samples form deep weight prior and learned filters perform similarly, but with deep weight prior we can aviod storing all learned kernels. At Figure 2, we show results on MNIST and CIFAR-10 for different networks sizes. Specifically, we set the number of filters to be proportional to the scale parameter k and varied k.

4.3 Few-short Classification

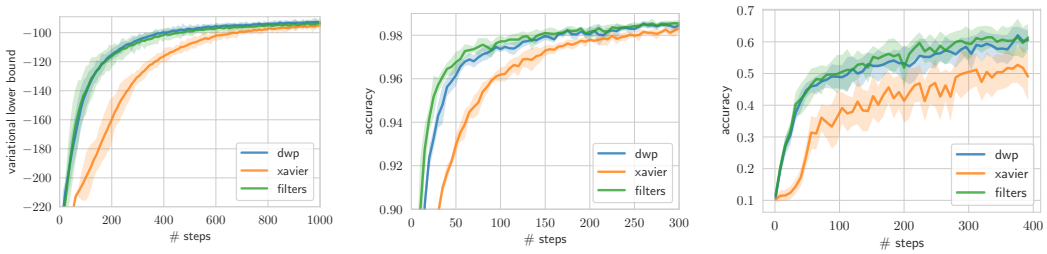
In this experiment, we perform variational inference over weights of discriminative convolutional neural network (Section 3) with three different prior distributions for the weights of convolutional layers: deep weight prior (dwp), standard normal and log-uniform (Kingma et al., 2015). We do not perform variational inference over the parameters of fully connected layers. We used a fully-factorized variational approximation with additive parameterization proposed by Molchanov et al. (2017) and local reparametrization trick proposed by Kingma et al. (2015). Note, that our method can be combined with more complex variational approximations, in order to improve quality of variational inference. We use the same architectures as in Section 4.2.



(a) Results for MNIST

(b) Results for CIFAR-10

Figure 3: For different number of training examples from MNIST and CIFAR-10 datasets, we demonstrate the performance of variational inference with a fully-factorized variational approximation with three different prior distributions: deep weight prior (dwp), log-uniform, and standard normal. We found that variational inference with deep weight prior distribution achieves better mean test accuracy comparing to learning with standard normal and log-uniform prior distributions.



(a) VAE on MNIST

(b) ConvNet on MNIST

(c) ConvNet on CIFAR-10

Figure 4: We found that initialization of weights of model with deep weight prior or learned filters significantly increases the training speed, comparing to Xavier initialization. At subplot 4(a) we report a variational lower bound for variational auto-encoder, at subplots 4(b) and 4(c) accuracy for convolution networks on MINTS and CIFAR-10.

At figure 3 we report accuracy for variational inference with different sizes of training datasets and prior distributions. Variational inference with deep weight prior leads to better mean test accuracy, in comparison to log-uniform and standard normal prior distributions. Note that the difference gets more significant as the training set gets smaller.

4.4 Convergence

Deep learning models are sensitive to initialization of model weights. In particular, it may influence the speed of the convergence or even on a convergence point. In this experiment, we study the influence of initialization on the convergence speed of two problems: variational auto-encoder on MNIST, and convolutional network on MNIST and CIFAR-10. We compare three different initializations of weights of conventional convolutional layers: learned filters, samples from deep weight prior and samples form Xavier distribution.

Figure 4 provides results for convolutional variational auto-encoder trained on MNIST and for convolutional classification network trained on CIFAR-10. We found that deep weight prior and filters initializations perform similarly and lead to significantly faster convergence comparing to standard Xavier initialization. Deep weight prior initialization however does not require us to store possibly large set of filters. Also, we plot samples form variational auto-encoders at different training steps Appendix C.

5 Related Work

The recent success of transfer learning (Yosinski et al., 2014) shows that convolutional networks produce similar weights of convolutional filters while being trained on different datasets from the same domain e.g. photo-realistic images. In contrast to Bayesian techniques (Kingma et al., 2015; Kochurov et al., 2018), these methods do not allow to obtain a posterior distribution over parameters of the model, and in most cases, they require to store convolutional weights of pre-trained models and careful tuning of hyperparameters.

Bayesian approach provides a framework that incorporates prior knowlege about the weights of the machine learning model by choosing or leaning prior distribution $p(w)$. There is a long line of research on prior distributions for Bayesian inference (MacKay, 1992; Williams, 1995), where empirical Bayes – approach that tunes parameters of the prior distribution by training data – plays an important role (MacKay, 1992). These methods are widely used for regularization and sparsification of linear models (Bishop & Tipping, 2003), however, applied to deep neural networks (Kingma et al., 2015; Ullrich et al., 2017), they do not take into account the structure of the model weights, e.g. spatial correlations, which does matter in case of convolutional networks. Our approach, allows to perform variational inference with implicit prior distribution, that is based on previously observed convolutional kernels. In contrast to empirical Bayes approach, parameters ϕ of deep weight prior (equation 6) are adjusted before variational inference and then remain fixed.

Implicit models have been applied to variational inference that result in a number of flexible variational approximations. Yin & Zhou (2018) proposed to use semi-implicit variational approximation. Salimans et al. (2015) proposed to use an implicit variational approximation in the form of Markov chain.

In this work, we propose to use an implicit prior distribution and develop a method for variational inference with the specific type of implicit priors. We show how to use this framework to learn a flexible prior distribution over kernels of Bayesian convolutional neural networks. Our approach can be generalized to prior distributions in the form of a Markov chain.

6 Discussion & Conclusion

In this work we propose *deep weight prior* – a framework for building prior distribution for convolutional neural networks, that exploits prior knowledge about the structure of learned convolutional filters. This framework opens a new direction for applications of Bayesian deep learning.

Factorization. The *lottery ticket hypothesis*, proposed by Frankle & Carbin (2018), argues that to converge to good solution we need just a few filters or neurons in the network with luckily good initialization. The factorization of deep weight prior, however, does not take into account inter-layer dependencies of the weights, although this dependencies might be useful. Accounting inter-layer dependencies may give us an opportunity to recover a distribution in space of trained networks rather then in space of trained kernels.

Inference. An alternative to variational inference with auxiliary variables is semi-implicit variational inference (Yin & Zhou, 2018). This algorithm might provide a better way for variational inference with deep weight prior, however it was developed only for semi-implicit variational approximations, so it might be interesting to adapt it for implicit prior distributions.

Parameterization. Generative models for weights of neural network, also can be applied to reduce memory usage via representing kernels of conditional filters w_{ij} by parameters of its hidden representations $\mu(w_{ij}), \sigma^2(w_{ij})$. Moreover optimization by $\mu(w_{ij}), \sigma^2(w_{ij})$ directly, may poses a better optimization problem.

References

Christopher M Bishop and Michael E Tipping. Bayesian regression and classification. *Nato Science Series sub Series III Computer And Systems Sciences*, 190:267–288, 2003.

Yaroslav Bulatov. Notmnist dataset. technical report. 2011. URL `http://yaroslavvb.blogspot.it/2011/09/notmnist-dataset.html`.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Onur Dikmen, Zhirong Yang, and Erkki Oja. Learning the information divergence. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1442–1454, 2015.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *5th International Conference on Learning Representations*, 2017.

Michael Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. *arXiv preprint arXiv:1805.08498*, 2018.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Training pruned neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Kun He, Yan Wang, and John Hopcroft. A powerful generative model using random weights for the deep image representation. In *Advances in Neural Information Processing Systems*, pp. 631–639, 2016.

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.

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.

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

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

Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pp. 2575–2583, 2015.

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 29*, pp. 4743–4751. 2016.

Max Kochurov, Timur Garipov, Dmitry Podoprikhin, Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Bayesian incremental learning for deep neural networks. *arXiv preprint arXiv:1802.07329*, 2018.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

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.

Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. *arXiv preprint arXiv:1703.01961*, 2017.

Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. In *Advances in Neural Information Processing Systems*, pp. 3288–3298, 2017.

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

Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 2498–2507, 2017.

Jawad Nagi, Frederick Ducatelle, Gianni A Di Caro, Dan Cireşan, Ueli Meier, Alessandro Giusti, Farrukh Nagi, Jürgen Schmidhuber, and Luca Maria Gambardella. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*, pp. 342–347. IEEE, 2011.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

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*, pp. 6775–6784, 2017.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, pp. 1530–1538, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1278–1286, 2014.

Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pp. 1218–1226, 2015.

Andrew M Saxe, Pang Wei Koh, Zhenghao Chen, Maneesh Bhand, Bipin Suresh, and Andrew Y Ng. On random weights and unsupervised feature learning. In *ICML*, pp. 1089–1096, 2011.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813, 2014.

B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1986. ISBN 9780412246203. URL `https://books.google.ru/books?id=e-xsrjsL7WkC`.

Jakub M Tomczak and Max Welling. Vae with a vampprior. *arXiv preprint arXiv:1705.07120*, 2017.

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

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *arXiv:1711.10925*, 2017.

Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pp. 4790–4798, 2016.

Peter M Williams. Bayesian regularization and pruning using a laplace prior. *Neural computation*, 7(1):117–143, 1995.

Mingzhang Yin and Mingyuan Zhou. Semi-implicit variational inference. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 5660–5669, 2018.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pp. 3320–3328, 2014.

A VARIATIONAL INFERENCE WITH IMPLICIT PRIOR DISTRIBUTION

We consider a variational lower bound $\mathcal{L}$ with variational approximation $q(w)$ and prior distribution defined in a form of Markov chain $p(w) = \int dz_0 \dots dz_T p(w \mid z_T) \prod_{t=0}^{T} p(z_{t+1} \mid z_t) p(z_0)$ and joint distribution $p(w, \boldsymbol{z}) = p(w \mid z_T) \prod_{t=0}^{T} p(z_{t+1} \mid z_t) p(z_0)$. Where $p(z_{t+1} \mid z_t)$ is a transition operator, and $\boldsymbol{z} = (z_0, \dots, z_T)$ (Salimans et al., 2015). Unfortunately, gradients of $\mathcal{L}$ cannot be efficiently estimated, but we construct a tractable lower bound $\mathcal{L}^{aux}$ for $\mathcal{L}$:

$$\mathcal{L} = \mathbb{E}_{q(w)} \left[\log p(x \mid w) p(w) - \log q(w) \right] = \mathbb{E}_{q(w)} \mathbb{E}_{r(\boldsymbol{z} \mid w)} \left[\log p(x \mid w) p(w) - \log q(w) \right] = \tag{10}$$

$$= \mathbb{E}_{q(w)} \mathbb{E}_{r(\boldsymbol{z} \mid w)} \left[\log p(x \mid w) \frac{p(w, \boldsymbol{z})}{p(\boldsymbol{z} \mid w)} \frac{r(\boldsymbol{z} \mid w)}{r(\boldsymbol{z} \mid w)} - \log q(w) \right] = \tag{11}$$

$$= \mathbb{E}_{q(w)} \mathbb{E}_{r(\boldsymbol{z} \mid w)} \left[\log p(x \mid w) \frac{p(w, \boldsymbol{z})}{r(\boldsymbol{z} \mid w)} - \log q(w) \right] + \mathbb{E}_{q(w)} D_{\mathrm{KL}}(r(\boldsymbol{z} \mid w) \| p(\boldsymbol{z} \mid w)) = \tag{12}$$

$$= \mathcal{L}^{aux} + \mathbb{E}_{q(w)} D_{\mathrm{KL}}(r(\boldsymbol{z} \mid w) \| p(\boldsymbol{z} \mid w)) \geq \mathcal{L}^{aux}. \tag{13}$$

Inequality laux has a very natural interpretation. The lower bound $\mathcal{L}^{aux}$ is tight if and only if the KL-divergence between the auxillary reverse model and the posterior intractable distribution $p(z \mid w)$ is zero.

The deep weight prior (Section 3) is a special of Markov chain prior for $T = 0$ and $p(w) = \int p(w \mid z) p(z) dz$. The auxiliary variational bound has the following form:

$$\mathcal{L}^{aux} = \mathbb{E}_{q(w)} \mathbb{E}_{r(z \mid w)} \left[\log p(x \mid w) \frac{p(w \mid z) p(z)}{r(z \mid w)} - \log q(w) \right] = \tag{14}$$

$$= \mathbb{E}_{q(w)} \left[\log p(x \mid w) \right] + H(q) - \mathbb{E}_{q(w)} \left[D_{\mathrm{KL}}(r(z \mid w) \| p(z) - \mathbb{E}_{r(z \mid w)} \log p(w \mid z)) \right]. \tag{15}$$

where the gradients in equation 14 can be efficiently estimated in case $q(w)$, for explicit distributions $q(w), p_\phi(w \mid z), r(z \mid w)$ that can be reparametrized.

B VISUALIZATION OF DEEP WEIGHT PRIOR LATENT SPACE

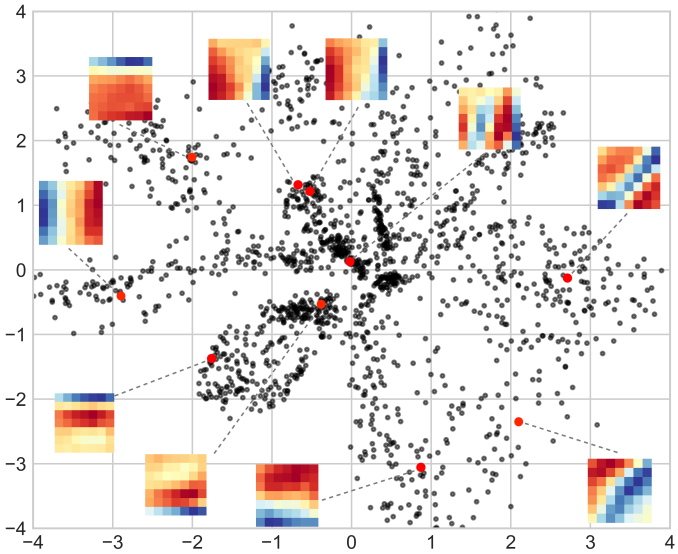
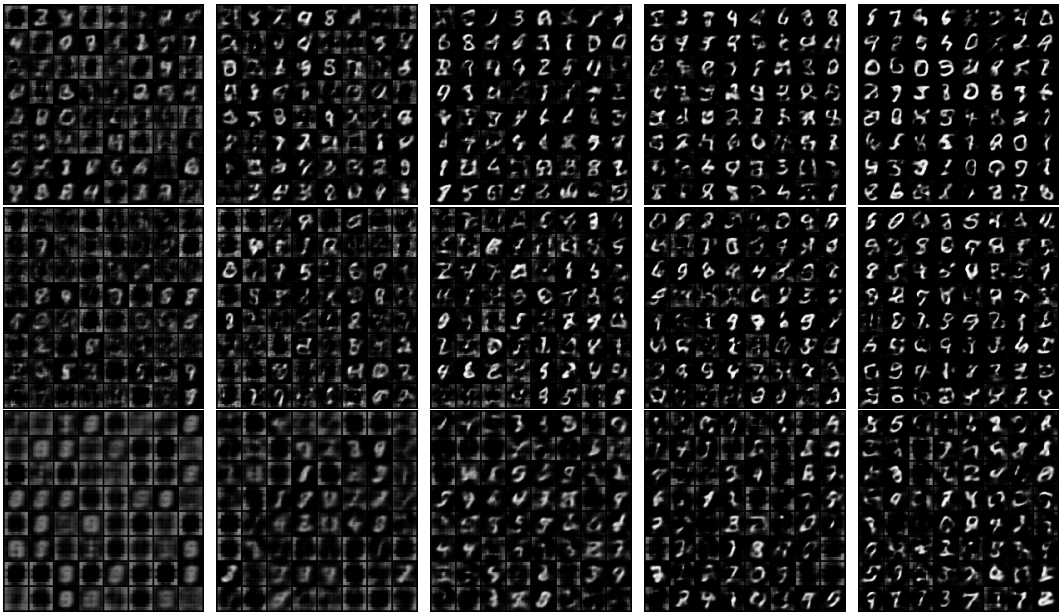


Figure 5: An illustration for Section 4.3. We visualize latent representations of convolutional filters for ConvNet on NotMNIST. Every point corresponds to mean of latent representation $q(z \mid w_i)$, where w_i is a kernel of shape 7×7 from the first convolutional layer, and $q(z \mid w_i)$ is an inference network with a two-dimensional latent represenation.

C SAMPLES FORM VARIATIONAL AUTO-ENCODERS



(a) 100 steps (b) 200 steps (c) 300 steps (d) 400 steps (e) 500 steps

Figure 6: An illustration for the Section 4.4 of samples from variational auto-encoder for three different types of initialization of convolutional layers after 100, 200, 300, 400 and 500 steps of optimization. The first row corresponds to deep weight prior initialization, the second to initialization with learned kernels, and the third to xavier initialization.

D PRIOR ARCHITECTURES

Encoder5x5	**Decoder5x5**	**Encoder7x7**	**Decoder7x7**
Conv, 64, 3×3	Conv, 128, 1×1	Conv, 32, 3×3	ConvT, 64, 3×3
Conv, 64, 3×3	ConvT, 128, 3×3	Conv, 64, 3×3	ConvT, 64, 3×3
Conv, 128, 3×3	ConvT, 128, 3×3	Conv, 64, 3×3	ConvT, 32, 3×3
Conv, 128, 3×3	ConvT, 64, 1×1	$2 \times$ Linear, z_{hid}	$2 \times$ ConvT, 1, 1×1
$2 \times$ Linear, z_{hid}	$2 \times$ Conv, 1, 1×1		
= 260040 params	= 304194 params	= 56004 params	= 56674 params

Table 1: Architectures of variational auto-encoders for prior distributions. On the left for filters of shapes 5×5 and for filters of shape 7×7 on the right. See more details at Section 4. All layers were divided with ELU non-literary, layers of Encoders were also divided with max-polling2×2 layer.

E NETWORK ARCHITECTURES

Classification MNIST	**Classification CIFAR**	**Variational Auto-encoder MNIST**
Conv, 32, 7×7	Conv2d, 128, 7×7	Conv2d, 64, stride 2, 7×7
Conv, 128, 5×5	Conv2d, 256, 5×5	Conv2d, 128, 5×5
Linear, 10	Conv2d, 256, 5×5	$2\times$ Linear, z_{hid}
	Linear, 512	ConvT, 128, 5×5
	Linear, 10	ConvT, 64, stride 2, 5×5
		ConvT, 1, stride 2, 5×5
= 115658 params	= 5759498 params	= 1641665 params

Table 2: Network Architectures for MNIST and CIFAR-10/CIFAR-100 datasets (Section 4).