2019 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, OCT. 13–16, 2019, PITTSBURGH, PA, USA

EXTENDED VARIATIONAL INFERENCE FOR PROPAGATING UNCERTAINTY IN CONVOLUTIONAL NEURAL NETWORKS

Dimah Dera, Ghulam Rasool and Nidhal Bouaynaya

Rowan University, Electrical and Computer Engineering, Glassboro, NJ

ABSTRACT

Model confidence or uncertainty is critical in autonomous systems as they directly tie to the safety and trustworthiness of the system. The quantification of uncertainty in the output decisions of deep neural networks (DNNs) is a challenging problem. The Bayesian framework enables the estimation of the predictive uncertainty by introducing probability distributions over the (unknown) network weights; however, the propagation of these high-dimensional distributions through multiple layers and non-linear transformations is mathematically intractable. In this work, we propose an extended variational inference (eVI) framework for convolutional neural network (CNN) based on tensor Normal distributions (TNDs) defined over convolutional kernels. Our proposed eVI framework propagates the first two moments (mean and covariance) of these TNDs through all layers of the CNN. We employ first-order Taylor series linearization to approximate the mean and covariances passing through the non-linear activations. The uncertainty in the output decision is given by the propagated covariance of the predictive distribution. Furthermore, we show, through extensive simulations on the MNIST and CIFAR-10 datasets, that the CNN becomes more robust to Gaussian noise and adversarial attacks.

1. INTRODUCTION

The estimation of uncertainty or confidence in the output decisions of deep neural networks (DNNs) is pivotal for their deployment in real-world scenarios [1]. In modern applications, including autonomous driving [2] and medical diagnosis [3], the reliability of the predicted decision and the robustness of the model to input noise are crucial. One possible way of estimating uncertainty/confidence in the output of DNNs includes Bayesian treatment of unknown parameters, i.e., introducing prior distributions and later estimating the posterior distributions over the network weights. However, posterior inference in DNNs is analytically intractable and approximations such as variational inference (VI) are often used [4]. Recent work has shown that VI approximation can be scaled to large and modern DNN architectures [5]. However, the challenge remains, i.e., the propagation of distributions introduced over the weights through multiple layers (consisting of linear and nonlinear transformations) of DNNs.

Recently, Roth and Pernkopf proposed a VI framework for the propagation of moments of the approximate distribution through layers of fully-connected neural networks [6]. They presented a closed-form solution for propagating moments through rectified linear unit (ReLU) activation functions with Gaussian distributions. Later, Wu *et al* extended [6] framework for the Heaviside activation function and developed an empirical Bayes method for tuning prior distributions of the weights during training [7]. Hernandez-Lobato and Adams proposed probabilistic back-propagation (PBP) [8] to propagate probabilities through a fully-connected network using assumed density filtering (ADF) [9]. ADF is known to be sensitive to observation ordering, an undesirable property in the batch context [10]. PBP is restricted to ReLU activation and continuous regression problems. Ghosh *et al* extended PBP to multi-class classification [11]. Gast and Roth propagated observations uncertainty (non-Bayesian framework) through deterministic CNN by using ADF framework and ReLU and Leaky ReLU activations [12].

Blundell *et al* introduced Bayes-by Backprop (BBB) and defined a fully-factorized Gaussian distribution over the weights of the neural network [13]. In a followup work, Shridhar *et al* extended BBB to Bayes-CNN by introducing a fully-factorized Gaussian distribution over the convolutional kernels [14]. On the other hand, Gal and Ghahramani extended Bayesian dropout approximation for CNN (termed Dropout CNN) by formulating the VI problem with Bernoulli distributions over the convolutional kernels [15]. In all these approaches [13–15], the second moment (covariance matrix) of the weights is *not propagated* from one layer of the neural network to the next layer. The uncertainty of the network output is estimated at the test time using Monte Carlo runs by sampling from the estimated distribution of weights.

Recent approaches proposed for propagating model uncertainty considered only the fully-connected network and limited choice of activation function such as (ReLU, leaky ReLU and/or Heaviside functions). To the best of our knowledge, none of the recent methods for propagating model uncertainty considered a CNN with a general choice of activation function which enables flexibility in extending the framework for

978-1-7281-0824-7/19/$31.00

various network architecture and different datasets.

In this paper, we propose an extended VI (eVI) approach for propagating model uncertainty in CNN. The convolutional kernels are considered as random tensors and their first and second moments are propagated through all layers (convolution, max-pooling and fully-connected). The covariance of the predictive distribution, which represents the uncertainty associated with the prediction, is the covariance of the distribution of the weights propagated through layers of the CNN. Our contributions include:

1. Introducing tensor Normal distributions (TNDs) over convolutional kernels. TND captures the correlation and variance heterogeneity, both within and among dimensions [16].
2. Approximating the means and covariances of the TNDs after propagating them through nonlinear activation functions using Taylor series. Propagation of moments through layers of CNN make it robust to noise (additive, inherent or adversarial) in the data as well as variations in the model parameters (kernels).
3. Experimental results showing superior robustness of eVI-CNN against Gaussian noise and adversarial attacks on MNIST and CIFAR-10 datasets.

2. PROBABILITY DISTRIBUTIONS OVER THE WEIGHTS IN DEEP NEURAL NETWORKS

We view a neural network as a probabilistic model $p(\mathbf{y}|\mathcal{X}, \mathbf{\Omega})$: given an input $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times K}$, the neural network assigns a probability distribution to each possible output $\mathbf{y}$, using the set of weights $\mathbf{\Omega}$. The weight parameters define all network layers, $\mathbf{\Omega} = \{\{\{\mathcal{W}^{(k_c)}\}_{k_c=1}^{K_c}\}_{c=1}^{C}, \{\mathbf{W}^{(l)}\}_{l=1}^{L}\}$, where $\{\{\mathcal{W}^{(k_c)}\}_{k_c=1}^{K_c}\}_{c=1}^{C}$ is the set of C convolutional layers with K_c kernels in the c^{th} convolutional layer, and $\{\mathbf{W}^{(l)}\}_{l=1}^{L}$ is the set of L fully-connected layers.

In deterministic setting, the optimal weights are obtained by maximizing the likelihood $p(\mathcal{D}|\mathbf{\Omega})$ given the training data $\mathcal{D} = \{\mathcal{X}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$ or by maximizing the posterior $p(\mathbf{\Omega}|\mathcal{D})$, where the prior distribution is considered as a regularization term. The likelihood distribution $p(\mathbf{y}|\mathcal{X}, \mathbf{\Omega})$, in deterministic models, is generally, the cross-entropy loss for classification problems or squared loss for regression problems, and network parameters are updated through back-propagation.

In our setting, we introduce a prior distribution $\mathbf{\Omega} \sim p(\mathbf{\Omega})$ over network parameters. By estimating the posterior distribution of the weights given the data $p(\mathbf{\Omega}|\mathcal{D})$, we can find the predictive distribution of any new unseen data point $\tilde{\mathcal{X}}$,

$$p(\tilde{\mathbf{y}}|\tilde{\mathcal{X}}, \mathcal{D}) = \int p(\tilde{\mathbf{y}}|\tilde{\mathcal{X}}, \mathbf{\Omega})\, p(\mathbf{\Omega}|\mathcal{D})\, d\mathbf{\Omega}. \tag{1}$$

An illustration of a probabilistic convolutional neural network with one convolutional layer, one max-pooling and one fully-connected layer is shown in Fig. 1.

3. EXTENDED VARIATIONAL INFERENCE (EVI)

3.1. Tensor Normal Distribution (TND)

A fully factorized Gaussian distribution defined over the kernel tensor imposes a restrictive independence assumption between the kernel elements. Instead, we use TNDs, which are defined over n-dimensional arrays [16]. Specifically, a TND of order 3 is defined as: $\mathcal{W} \sim \mathcal{TN}_{n_1,n_2,n_3}(\mathcal{M}, \mathcal{T})$, where $\mathcal{M} = E[\mathcal{W}]$, and $\mathcal{T}$ is the covariance tensor of order six. It can be shown that this covariance tensor is positive semi-definite. In a separable or Kronecker structured model [16], the covariance matrix of the vectorized multi-dimensional array is the Kronecker product of covariance matrices equal to the number of dimensions, i.e., $\mathcal{T} = \bigotimes_{j=3}^{1} \mathbf{U}^{(j)}$, where $\{\mathbf{U}^{(j)}\}_{j=1}^{3} \in \mathbb{R}^{n_j \times n_j}$ are positive semi-definite matrices. We note that this factorization reduces the number of parameters to be estimated. In a separable model, an equivalent formulation of the TND is essentially a multivariate Gaussian distribution, i.e.,

$$\text{vec}(\mathcal{W}) \sim \mathcal{N}_{\prod_{j=1}^{3} n_j}(\text{vec}(\mathcal{M}), \bigotimes_{j=3}^{1} \mathbf{U}^{(j)}), \tag{2}$$

where $\text{vec}(.)$ denotes the vectorization operation. We assume that convolutional kernels are independent of each other within as well as across layers. The independence assumption allows convolutional layers to extract independent features within and across layers in a CNN.

3.2. VI with Tensor Normal Distributions (TNDs)

We use the variational learning approach for estimating the posterior distribution of the weights given data by minimizing the Kullback-Leibler (KL) divergence between a proposed approximate distribution $q_\phi(\mathbf{\Omega})$ (i.e., TNDs over convolutional kernels) and the true posterior distribution of the weights.

$$\begin{aligned} \phi^* &= \text{argmin}\, \text{KL}\left[q_\phi(\mathbf{\Omega}) \| p(\mathbf{\Omega}|\mathcal{D})\right], \\ &= \text{argmin}\, \text{KL}\left[q_\phi(\mathbf{\Omega}) \| p(\mathbf{\Omega})\right] - E\left\{\log p(\mathcal{D}|\mathbf{\Omega})\right\}, \end{aligned} \tag{3}$$

where $E = E_{q_\phi(\mathbf{\Omega})}$. Let us denote by $\mathfrak{L}(\phi; \mathbf{y}|\mathcal{X})$ the (variational) or evidence lower bound (ELBO) as:

$$\mathfrak{L}(\phi; \mathbf{y}|\mathcal{X}) = E(\log p(\mathbf{y}|\mathcal{X}, \mathbf{\Omega})) - \text{KL}(q_\phi(\mathbf{\Omega}) \| p(\mathbf{\Omega})). \tag{4}$$

An optimal approximation to posterior distribution is obtained by maximizing the ELBO objective function, which consists of two parts: the expected log-likelihood of the training data given the weights, and a regularization term. The expected log-likelihood is defined as a multivariate Gaussian with the mean and covariance estimated by propagating the mean and covariances of the approximate distribution $q_\phi(\mathbf{\Omega})$ through the network.

3.3. Propagation of the First two Moments

Without loss of generality, we demonstrate propagation of means and covariances of the approximate distribution $q_\phi(\mathbf{\Omega})$

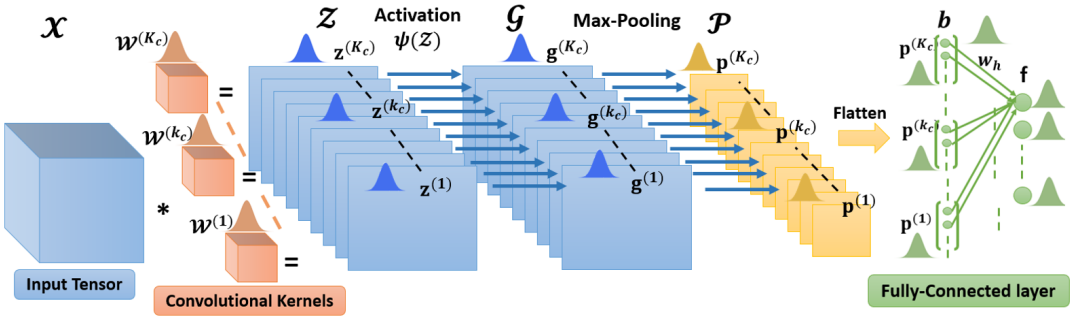

Fig. 1. Propagation of the mean and covariance of the approximate distribution $q_\phi(\mathbf{\Omega})$ through multiple layers of a CNN. The network consists of a single convolutional layer followed by a non-linear activation function, a single max-pooling layer and one fully-connected layer.

through a CNN with one convolutional layer ($C = 1$) followed by the activation function, one max-pooling and one fully-connected layer ($L = 1$) in Fig. 1. Our goal is to obtain the mean and covariance of the likelihood distribution, $p(\mathbf{y}|\mathcal{X}, \mathbf{\Omega})$, which represent the network's prediction (mean) and the uncertainty associated with it (variances in the covariance matrix).

Convolutional Layer: The convolution operation between a set of kernels and the input tensor is formulated as a matrix-vector multiplication. We first form sub-tensors $\mathcal{X}_{i:i+r_1-1,j:j+r_2-1}$ from the input tensor $\mathcal{X}$, having the same size as the kernels $\mathcal{W}^{(k_c)} \in \mathbb{R}^{r_1 \times r_2 \times K}$. These sub-tensors are subsequently vectorized and arranged as the rows of a matrix $\tilde{\mathbf{X}}$. Thus, we have $\mathcal{X} * \mathcal{W}^{(k_c)} \Longleftrightarrow \tilde{\mathbf{X}}\ \text{vec}(\mathcal{W}^{(k_c)})$, where $*$ denotes the convolution operation.

We denote the output of the convolution of the k_c^{th} kernel with the input by $\mathbf{z}^{(k_c)} = \tilde{\mathbf{X}}\ \text{vec}(\mathcal{W}^{(k_c)})$. We endow the kernels with TNDs, which are equivalent to multivariate Gaussian distribution over the vectorized kernels, i.e. $\text{vec}(\mathcal{W}^{(k_c)}) \sim \mathcal{N}\left(\mathbf{m}^{(k_c)}, \mathbf{\Sigma}^{(k_c)}\right)$, where $\mathbf{m}^{(k_c)} = \text{vec}(\mathcal{M}^{(k_c)})$ and $\mathbf{\Sigma}^{(k_c)} = \mathbf{U}^{(1,k_c)} \otimes \mathbf{U}^{(2,k_c)} \otimes \mathbf{U}^{(3,k_c)}$. It follows that, $\mathbf{z}^{(k_c)} \sim \mathcal{N}\left(\tilde{\mathbf{X}}\mathbf{m}^{(k_c)},\ \tilde{\mathbf{X}}\mathbf{\Sigma}^{(k_c)}\tilde{\mathbf{X}}^T\right)$.

Non-linear Activation Function: We approximate mean and covariance passing through the non-linear activation function ψ using Taylor series (first-order approximation [17]). Let $\mathbf{g}_i^{(k_c)} = \psi[\mathbf{z}_i^{(k_c)}]$ be the element-wise i^{th} output of ψ. Thus, elements of $\boldsymbol{\mu}_{\mathbf{g}^{(k_c)}}$ and $\mathbf{\Sigma}_{\mathbf{g}^{(k_c)}}$ are derived as:

$$
\begin{aligned}
\text{E}[\mathbf{g}_i^{(k_c)}] &\approx \psi(\text{E}[\mathbf{z}_i^{(k_c)}]),\\
\text{Var}[\mathbf{g}_i^{(k_c)}] &\approx \sigma^2_{\mathbf{z}_i^{(k_c)}} \left(\frac{d\psi(\mu_{\mathbf{z}_i^{(k_c)}})}{d\mathbf{z}_i^{(k_c)}}\right)^2,\\
\text{Cov}[\mathbf{g}_i^{(k_c)}, \mathbf{g}_j^{(k_c)}] &\approx \sigma_{\mathbf{z}_i^{(k_c)}\mathbf{z}_j^{(k_c)}} \left(\frac{d\psi(\mu_{\mathbf{z}_i^{(k_c)}})}{d\mathbf{z}_i^{(k_c)}}\right)\left(\frac{d\psi(\mu_{\mathbf{z}_j^{(k_c)}})}{d\mathbf{z}_j^{(k_c)}}\right),
\end{aligned}
\tag{5}
$$

where $i \neq j$.

Max-Pooling Layer: For the max-pooling, $\boldsymbol{\mu}_{\mathbf{p}^{(k_c)}} = \text{pool}(\boldsymbol{\mu}_{\mathbf{g}^{(k_c)}})$ and $\mathbf{\Sigma}_{\mathbf{p}^{(k_c)}} = \text{co-pool}(\mathbf{\Sigma}_{\mathbf{g}^{(k_c)}})$, where pool represents the max-pooling operation on the mean and co-pool represents down-sampling the covariance, i.e., we keep only the rows and columns of $\mathbf{\Sigma}_{\mathbf{g}^{(k_c)}}$ corresponding to the pooled means.

Fully-Connected Layer: The output tensor of the max-pooling, i.e., $\mathcal{P}$ (as shown in Fig. 1) is vectorized to form an input vector $\mathbf{b}$ to the fully-connected layer such that, $\mathbf{b} = \left[\mathbf{p}^{(1)T}, \cdots, \mathbf{p}^{(K_c)T}\right]^T$. The mean and covariance matrix of $\mathbf{b}$ are given by:

$$
\boldsymbol{\mu}_{\mathbf{b}} = \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{p}^{(1)}} \\ \vdots \\ \boldsymbol{\mu}_{\mathbf{p}^{(K_c)}} \end{bmatrix}, \mathbf{\Sigma}_{\mathbf{b}} = \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{p}^{(1)}} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{\Sigma}_{\mathbf{p}^{(K_c)}} \end{bmatrix} \tag{6}
$$

Let $\mathbf{w}_h \sim \mathcal{N}(\mathbf{m}_h, \mathbf{\Sigma}_h)$ be the weight vectors of the fully-connected layer, where $h = 1, \cdots, H$, and H is the number of output neurons. We note that f_h is the product of two independent random vectors $\mathbf{b}$ and $\mathbf{w}_h$. Let $\mathbf{f}$ be the output vector of the fully-connected layer, then we can prove that the elements of $\boldsymbol{\mu}_{\mathbf{f}}$ and $\mathbf{\Sigma}_{\mathbf{f}}$ are derived as:

$$
\begin{aligned}
\text{E}[f_h] &= \mathbf{m}_h^T \boldsymbol{\mu}_{\mathbf{b}},\\
\text{Var}[f_h] &= \text{tr}\left(\mathbf{\Sigma}_h \mathbf{\Sigma}_{\mathbf{b}}\right) + \mathbf{m}_h^T \mathbf{\Sigma}_{\mathbf{b}} \mathbf{m}_h + \boldsymbol{\mu}_{\mathbf{b}}^T \mathbf{\Sigma}_h \boldsymbol{\mu}_{\mathbf{b}},\\
\text{Cov}[f_{h_i}, f_{h_j}] &= \mathbf{m}_{h_i}^T \mathbf{\Sigma}_{\mathbf{b}} \mathbf{m}_{h_j}, i \neq j.
\end{aligned}
\tag{7}
$$

Assuming diagonal covariance matrices for the distributions defined over network weights, i.e., $\textbf{vec}\left(\mathcal{W}^{(k_c)}\right) \sim \mathcal{N}(\text{vec}(\mathcal{M}^{(k_c)}); \sigma^2_{r_1,k_c}\mathbf{I}, \sigma^2_{r_2,k_c}\mathbf{I}, \sigma^2_{K,k_c}\mathbf{I})$, and $\mathbf{w}_h \sim \mathcal{N}(\mathbf{m}_h; \sigma_h^2\mathbf{I})$, N independently and identically distributed (iid) data points and using M Monte Carlo samples to approximate the expectation by a summation, the ELBO objective function is re-formulated as:

$$\mathfrak{L}(\phi;\mathcal{D}) \approx -\frac{NH}{2}\log(2\pi) - \frac{1}{M}\sum_{m=1}^{M}\Big[\frac{N}{2}\log(|\mathbf{\Sigma_f}|) + \frac{1}{2}\sum_{i=1}^{N}(\mathbf{y}^{(i)} - \boldsymbol{\mu}_\mathbf{f})^T(\mathbf{\Sigma_f})^{-1}(\mathbf{y}^{(i)} - \boldsymbol{\mu}_\mathbf{f})\Big]$$
$$- \frac{1}{2}\sum_{k_c=1}^{K_c}\left(r_1 r_2 K\,\sigma^2_{r_1,k_c}\,\sigma^2_{r_2,k_c}\,\sigma^2_{K,k_c} + \|\mathfrak{M}^{(k_c)}\|_F^2 - r_1 r_2 K - r_1 r_2 K\left(\log\sigma^2_{r_1,k_c} + \log\sigma^2_{r_2,k_c} + \log\sigma^2_{K,k_c}\right)\right) \quad (8)$$
$$- \frac{1}{2}\sum_{h=1}^{H}\Big(n_f\,\sigma_h^2 + \|\mathbf{m}_h\|_F^2 - n_f - n_f\,\log\sigma_h^2\Big),$$

where n_f is length of $\mathbf{w}_h$. Last two terms in Eq. (8) are result of KL-divergence between prior and approximate distributions [18] and act as regularizations. Equation (8) can be extended to multiple layers as well as to different network types, i.e., recurrent neural networks.

4. EXPERIMENTS AND DISCUSSION

The uncertainty propagation in DNNs results in an increased robustness against noise and adversarial attacks. We assess the performance of eVI-CNN by adding various levels of synthetic Gaussian and adversarial noise to the test sets of MNIST [19] and CIFAR-10 [20] datasets. We compare with Bayes-by-backprop (BBB) [13], Bayes-CNN [14], Dropout CNN [15] and with a deterministic CNN (no uncertainty propagation).

For MNIST handwritten digits dataset, we use a small size network architecture with one convolutional layer followed by the ReLU activation [21], one max-pooling, and one fully-connected layer. There are 32 kernels in the convolutional layer, and they are of size (5×5). While, for the CIFAR-10 dataset, we extend the network architecture to six convolutional layers followed by exponential linear unit (ELU) activations [22], three max-pooling, and one fully-connected layer. The convolutional kernels are of size (3×3), and they are ordered, from the first to the last convolutional layer, as $(32, 32, 64, 64, 128$ and $128)$ kernels. Figure 2 shows the architecture of the proposed eVI-CNN network for MNIST and CIFAR-10 datasets.

In Table 1, we present test accuracy of eVI-CNN, Bayes-CNN and Dropout CNN on CIFAR-10 dataset before and after adding of 0.05 (5%) level of adversarial noise. The adversarial examples were generated using fast gradient sign method (FGSM) to fool each network into predicting the class label as a "cat" [23]. We note that all three networks perform well on noise-free test data; however, eVI-CNN maintains its performance under adversarial attacks while other methods start failing (i.e., 17% decrease in Bayes-CNN test accuracy and 34% decrease in Dropout CNN). Figure 3 shows three randomly chosen images from the CIFAR-10 dataset corrupted by FGSM adversarial noise generated to fool each network (Dropout CNN, Bayes-CNN and proposed eVI-CNN) into predicting the class label as a cat [23]. Under each image, we present the true label and the predicted one.

In Table 2, we present test accuracy of eVI-CNN, BBB and a deterministic CNN on MNIST dataset before and after inserting various levels of adversarial and additive Gaussian noise to the test set. The adversarial examples were generated using FGSM to fool each network into predicting digit "3" [23]. We note that the performance of all networks decreases when Gaussian and especially adversarial noise is added; however, our proposed eVI-CNN is significantly more robust than other state-of-the-art CNNs.

In Figure 4, we present the variance of the output predictions (diagonal element corresponding to predicted class label from the output covariance matrix) of eVI-CNN plotted against the signal to noise ratio (SNR in dB) of different test images. The SNR was decreased by adding synthetic Gaussian noise. We note that the output variance increases with the decreasing SNR; thus, provide a metric for the confidence (or uncertainty) in the network's output. , i.e., the higher the noise in the input data, the lower the confidence in the prediction made by eVI-CNN.

We hypothesize that the robustness of the proposed eVI-CNN against both Gaussian noise and adversarial attacks is linked with their ability to propagate uncertainty across layers. In particular, eVI-CNN associates variance with every convolutional kernel which quantifies confidence in the features learned by the kernel. Since these confidence values are propagated across layers, the eVI-CNN weighs the features according to their confidence and is able to better resist noise and adversarial attacks.

Adversarial Noise	eVI	Bayes-CNN	Dropout CNN
5%	80%	68%	52%
Zero (No noise)	86%	85%	86%

Table 1. Test accuracy of eVI-CNN, Bayes-CNN and Dropout CNN on the CIFAR-10 dataset before and after adding 5% level of the FGSM adversarial noise to the test set.

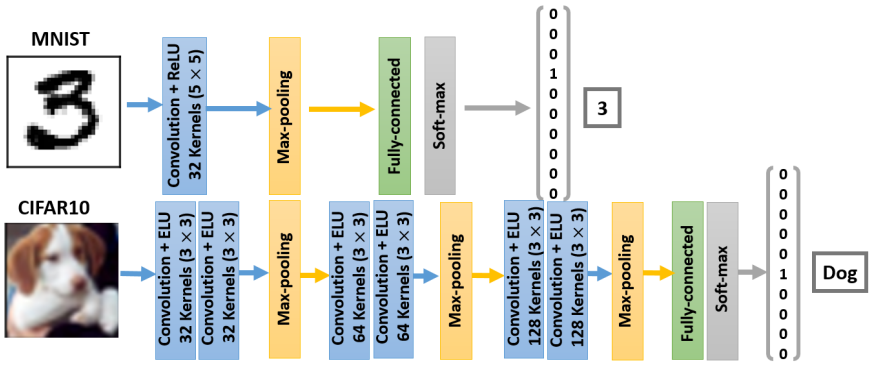

Fig. 2. Architecture of the proposed eVI-CNN network for the MNIST and CIFAR-10 datasets.

Adversarial Noise	eVI	BBB	CNN
0.1	95%	91%	58%
0.2	81%	45%	14%
0.3	**50%**	**16%**	**14%**
Gaussian Noise			
0.1	94%	86%	79%
0.2	85%	76%	70%
0.3	**74%**	**63%**	**55%**
Zero (No noise)	96%	96%	96%

Table 2. Test accuracy of eVI-CNN, BBB and Deterministic CNN on the MNIST dataset before and after adding different levels of FGSM adversarial noise and Gaussian noise.

5. CONCLUSION

We introduced an extended variational inference (eVI) approach for estimating model uncertainty by propagating the first two moments of the approximate posterior distributions defined over the weights through the layers of a CNN. We defined tensor Normal distributions (TNDs) over the convolutional kernels and propagated the means and covariances of the TNDs through the network layers and non-linearities using Taylor series. In particular, the variance of the predictive output was computed as the result of propagating input and weight (model) covariances across the network layers. The proposed eVI-CNN showed superior robustness to added Gaussian noise as well as FGSM adversarial attacks as compared to its deterministic counterpart and other state-of-the-art Bayesian networks: BBB, Bayes-CNN and Dropout CNN, on the MNIST and CIFAR-10 datasets.

6. ACKNOWLEDGMENT

This work was supported by the National Science Foundation Awards NSF ECCS-1903466 and NSF CCF-1527822. Dimah Dera was also supported by an ACM SIGHPC/Intel Computational and Data Science Fellowship Award. The authors would like to acknowledge the insights and collaboration of Dr. Hassan M. Fathallah-Shaykh, Professor of Neurology at the University of Alabama at Birmingham School of Medicine.

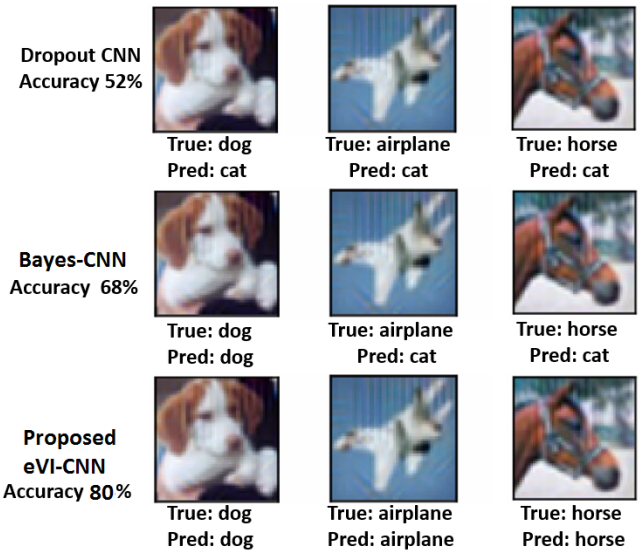

Fig. 3. Predictions of the Dropout CNN [15], Bayes-CNN [14] and the proposed eVI-CNN for three randomly chosen images from CIFAR-10 test dataset corrupted by FGSM adversarial noise built at the same level, i.e., 5% for all networks.

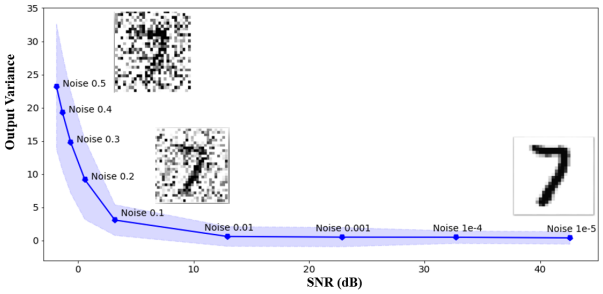

Fig. 4. The variance of the output prediction of eVI-CNN vs. signal-to-noise ratio (SNR). The eVI-CNN was tested on the MNIST test dataset corrupted with different levels of Gaussian noise.

7. REFERENCES

[1] Alex Kendall and Yarin Gal, "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2017, pp. 5574–5584.

[2] Evan Ackerman, "How drive.ai is mastering autonomous driving with deep learning," *IEEE Spectrum Magazine*, Mar. 2017.

[3] Dejun Shi, Yaling Pan, Chunlei Liu, Yao Wang, Deqi Cui, and Yong Lu, "Automatic localization and segmentation of vertebral bodies in 3d CT volumes with deep learning," in *Proceedings of the International Symposium on Image Computing and Digital Medicine*, 2018, pp. 42–46.

[4] Geoffrey E. Hinton and Drew van Camp, "Keeping the neural networks simple by minimizing the description length of the weights," in *Proceedings of the Sixth Annual Conference on Computational Learning Theory.*, 1993, pp. 5–13.

[5] Alex Graves, "Practical variational inference for neural networks," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, 2011, pp. 2348–2356.

[6] Wolfgang Roth and Franz Pernkopf, "Variational inference in neural networks using an approximate closed-form objective," in *Neural Information Processing Systems workshop*, 2016.

[7] Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E. Turner, Jose Miguel Hernandez-Lobato, and Alexander L. Gaunt, "Deterministic variational inference for robust Bayesian neural networks," in *Proceedings of 7th International Conference on Learning Representations*, 2019.

[8] Jose Miguel Hernandez-Lobato and Ryan Adams, "Probabilistic backpropagation for scalable learning of Bayesian neural networks," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, vol. 37, pp. 1861–1869.

[9] Manfred Opper, *On-line Learning in Neural Networks*, Cambridge University Press, NY, USA, 1998.

[10] Thomas P. Minka, "Expectation propagation for approximate Bayesian inference," in *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, 2001, pp. 362–369.

[11] Soumya Ghosh, Francesco Maria Delle Fave, and Jonathan S. Yedidia, "Assumed density filtering methods for learning Bayesian neural networks.," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 1589–1595.

[12] Jochen Gast and Stefan Roth, "Lightweight probabilistic deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3369–3378.

[13] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra, "Weight uncertainty in neural networks," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, 2015, vol. 37, pp. 1613–1622.

[14] Kumar Shridhar, Felix Laumann, Adrian Llopart Maurin, and Marcus Liwicki, "Bayesian convolutional neural networks," *arXiv preprint arXiv:1806.05978*, 2018.

[15] Yarin Gal and Zoubin Ghahramani, "Bayesian convolutional neural networks with Bernoulli approximate variational inference," in *Proceedings of 4th International Conference on Learning Representations, workshop track*, 2016.

[16] A. M. Manceur and P. Dutilleul, "Maximum likelihood estimation for the tensor Normal distribution: Algorithm, minimum sample size, and empirical bias and dispersion," *Journal of Computational and Applied Mathematics*, vol. 239, pp. 37 49, 2013.

[17] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill Higher Education, 4 edition, 2002.

[18] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press, 2005.

[19] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, pp. 141–142, 2012.

[20] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[21] Vinod Nair and Geoffrey E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010, pp. 807–814.

[22] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *Proceedings of 4th International Conference on Learning Representations*, 2016.

[23] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *Proceedings of 5th International Conference on Learning Representations*, 2017.