AI1103 Presentation

Uncertainity Estimations by Softplus normalization in Bayesian Convolutional Neural Networks with Variational Inference

Kumar Shridhar, Felix Laumann

Hritik Sarkar

May 1, 2022

Introduction

- Convolutional neurl networks (CNNs) with frequintist inference (i.e. weights are single point estimates) require substantial amount of data examples to train on and are prone to overfitting on datasets with few examples per class.
- Introducing Bayesian methods to CNNs allows to express the uncertainty via its parameters in the form of probability distribution.
- At the same time, using a prior distribution to integrate out the parameters, lets us compute an average across many models during training. It has a regularization effect to the network, which prevents overfitting.

Continued

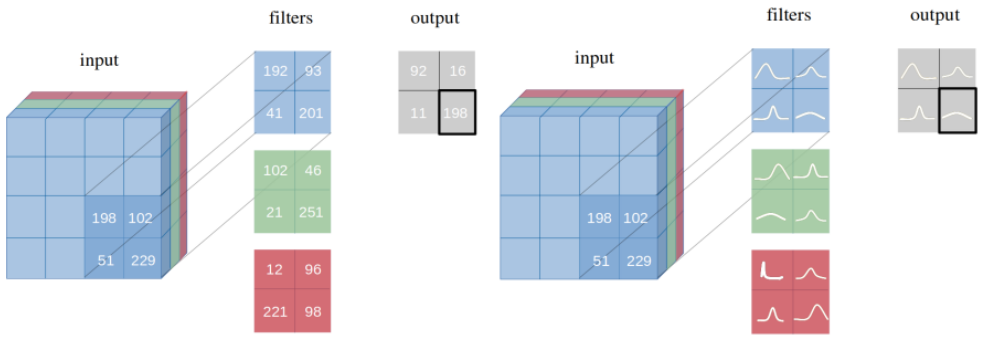

Figure: Input image with exemplary filters and outputs for point-estimates (left) and probability distributions (right) over weights

- In Bayesian approach we try to approximate the intractable true probability distribution $p(w|D)$ with the variational probability distributions $q_\theta(w|D)$

Bayes by backprop - Introduction

- Bayes by backprop is a variational inference method to learn the posterior distribution on the weights $w \sim q_\theta(w|D)$ of a NN from which w can be sampled during backpropagation.
- We try to minimize the KL divergence between the intractable true posterior $p(w|D)$ and the approximate distribution $q_\theta(w|D)$.
- The optimal parameters θ^{opt} can be defined as

$$\begin{aligned}\theta^{opt} &= \arg\min_\theta KL[q_\theta(w|D)||p(w|D)] \\ &= \arg\min_\theta KL[q_\theta(w|D)||p(w)] - \mathbb{E}_{q_\theta(w|D)}[logp(w|D)] + log\ p(D)\end{aligned} \tag{1}$$

- the KL divergence is defined as

$$KL[q_\theta(w|D)||p(w)] = \int q_\theta(w|D) log \frac{q_\theta(w|D)}{p(w)} d\theta \tag{2}$$

Variational free energy

- The cost function $KL[q_\theta(w|D)||p(w)] - \mathbb{E}_{q_\theta(w|D)}[logp(w|D)] + logp(D)$ is widely known as *variational free energy*.
- The first term $KL[q_\theta(w|D)||p(w)]$ is dependent on the definition of prior $p(w)$ and thus called complexity cost and the next term $\mathbb{E}_{q_\theta(w|D)}[logp(w|D)]$ is dependent on the data $p(D|w)$, thus called the likelihood cost. The term $logp(D)$ is a constant and can be omitted in optimization.
- The KL divergence also is intractable to compute exactly. We follow a stochastic variational method and consequently reach at a tractable cost that can be optimized with respect to θ during training.

$$F(D, \theta) \approx \Sigma_{i=1}^{n} logq_\theta(w^{(i)}|D) - logp(w^{(i)}) - logp(D|w^{(i)}) \qquad (3)$$

where n is the no. of draws. We sample $w(i)$ from $q_\theta(w|D)$.

Local reparameterization trick

- During implementation, we utilize the local reparameterization trick. We do not sample w, but we sample layer activations b instead due to its consequent computational accleration.
- The variational posterior probability distribution $q_\theta(w_{ijhw}|D) = \mathcal{N}(\mu_{ijhw}, \alpha_{ijhw}\mu_{ijhw}^2)$ (where i is the input layer, j is the output layer, h and w are the height and width of the filter respectively) allows to implement the local reparametrization trick in convolutional layers.
- The above results leads to the following equation

$$b_j = A_i \circledast \mu_i + \epsilon \odot \sqrt{A_i^2 \circledast (\alpha_i \odot \mu_i^2)} \tag{4}$$

where $\circledast$ is convolution and $\odot$ is component-wise multiplication.

Uncertainty estimation in Bayesian CNN's

In classification tasks we are interested in the predictive distribution $p_D(y^*|x^*)$, where x^* is an unseen data example and y^* is its predicted class. For a Bayesian neural network, this quantity is given by -

$$p_D(y^*|x^*) = \int p_w(y^*|x^*)p_D(w)dw \tag{5}$$

In Bayes by Backprop, Gaussian distribution $q_\theta(w|D) \sim \mathcal{N}(w|\mu, \sigma^2)$, where $\theta = \mu, \sigma$ are learned with some dataset $D = \{x_i, y_i\}_{i=1}^n$. Due to discrete finite nature of most classification tasks, the predictive distribution is commonly assumed to be categorical. Incorporating this we get

$$\begin{aligned} p_D(y^*|x^*) &= \int Cat(y^*|f_w(x^*))\mathcal{N}(w|\mu, \sigma^2)dw \\ &= \int \prod_{c=1}^{C} f(x_c^*|w)^{y_c^*} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(w-\mu)^2}{2\sigma^2}} \end{aligned} \tag{6}$$

Continued

where C is the total no. of classes and $\Sigma_c f(x_c^*|w) = 1$. There is no closed form solution due to lack of conjugacy between Categorical and Gaussian distribution. However we can construct an unbiased estimator of the expectation by sampling from $q_\theta(w|D)$.

$$\mathbb{E}_q[p_D(y^*|x^*)] = \int q_\theta(w|D)p_w(y|x)dw \approx \frac{1}{T}\Sigma_{t=1}^{T} p_{w_t}(y^*|x^*) \quad (7)$$

where T is the predefined no. of samples.

Continued

The variance of the estimator allows us to calculate the uncertainty of our predictions, hence called *predictive variance* and is denoted as -

$$\begin{aligned} Var_q(p(y^*|x^*)) &= \mathbb{E}_q[y^* {y^*}^T] - \mathbb{E}_q[y^*]\mathbb{E}_q[y^*]^T \\ &= \int [diag(\mathbb{E}_p[y^*]) - \mathbb{E}_p[y^*]\mathbb{E}_p[y^*]^T] q_\theta(w|D) dw \\ &+ \int (\mathbb{E}_p[y^*] - \mathbb{E}_q[y^*])(\mathbb{E}_p[y^*] - \mathbb{E}_q[y^*])^T q_\theta(w|D) dw \quad (8) \end{aligned}$$

where $\mathbb{E}_p[y*] = \mathbb{E}_{p(y^*|x^*)}[y^*]$ and $\mathbb{E}_q[y^*] = \mathbb{E}_{q_\theta(y^*|x^*)}[y^*]$

Continued

- The first term on the right hand side is aleatoric uncertainty (a measure for the variation of "noisy" data) and epistemic uncertainty (caused by the model). By comparing these two values one can see whether the quality of the data is low (high aleatoric uncertainity) or the model itself is the cause for poor performances (high epistemic uncertainty).
- The former can be improved by gathering more data whereas the latter requires that the model is refined.

Estimator for variance

The following estimator is used for the variance -

$$Var_q(p(y^*|x^*)) = \frac{1}{T}\Sigma_{t=1}^{T} diag(\hat{\sigma}_t^2) + \frac{1}{T}\Sigma_{t=1}^{T}(\hat{\mu}_t - \bar{\mu})(\hat{\mu}_t - \bar{\mu})^T \quad (9)$$

where $\hat{\mu}_t$ and $\hat{\sigma}_t^2$ are the mean and variance of the logits and $\bar{\mu} = \hat{\mu}_t/T$. A better estimator is the following -

$$Var_q(p(y^*|x^*)) = \frac{1}{T}\Sigma_{t=1}^{T} diag(\hat{p}_t) - \hat{p}_t\hat{p}_t^T + \frac{1}{T}\Sigma_{t=1}^{T}(\hat{p}_t - \bar{p})(\hat{p}_t - \bar{p})^T \quad (10)$$

where $\bar{p} = \hat{p}_t/T$ and $\hat{p} = Softmax(f_{w_t}(x^*))$.

Softplus

- In this paper the authors defined the doftplus function as below -

$$Softplus(x) = \frac{1}{\beta}.log(1 + exp(\beta.x))$$

  default value of β is 1.
- Deploying the traditional Softmax function at the output layer assumes that the classification is done according to logistic sigmoid function. Applying Softplus removes this additional non-linear transformation. We change $\hat{p} = Softmax(f_{w_t}(x^*))$ to $\hat{p} = Softplus(f_{w_t}(x^*))$ in (10)
- The implementation is done in two steps. Softplus function is applied to output layer just as any other layer. Then, the outputs are normalized as below

$$y^*_{norm} = \frac{y^*_c}{\Sigma_C y^*_c}$$

Intuition behind Softplus

- We classify according to a categorical distribution, this can be seen as an approximation of the outputs with one-hot vectors. If those vectors already contains a lot of zeros in it, the approximation is going to be more accurate. For a Softmax, predicting zeros requires a logit of $-\infty$, which is hard to achieve in practice. For the Softplus normalization we need a negative value around -4 to get a nearly 0 output. Consequenty, Softplus normalization can easily produce vectors that are in prectice zero, whereas Softmax cannot.

Results

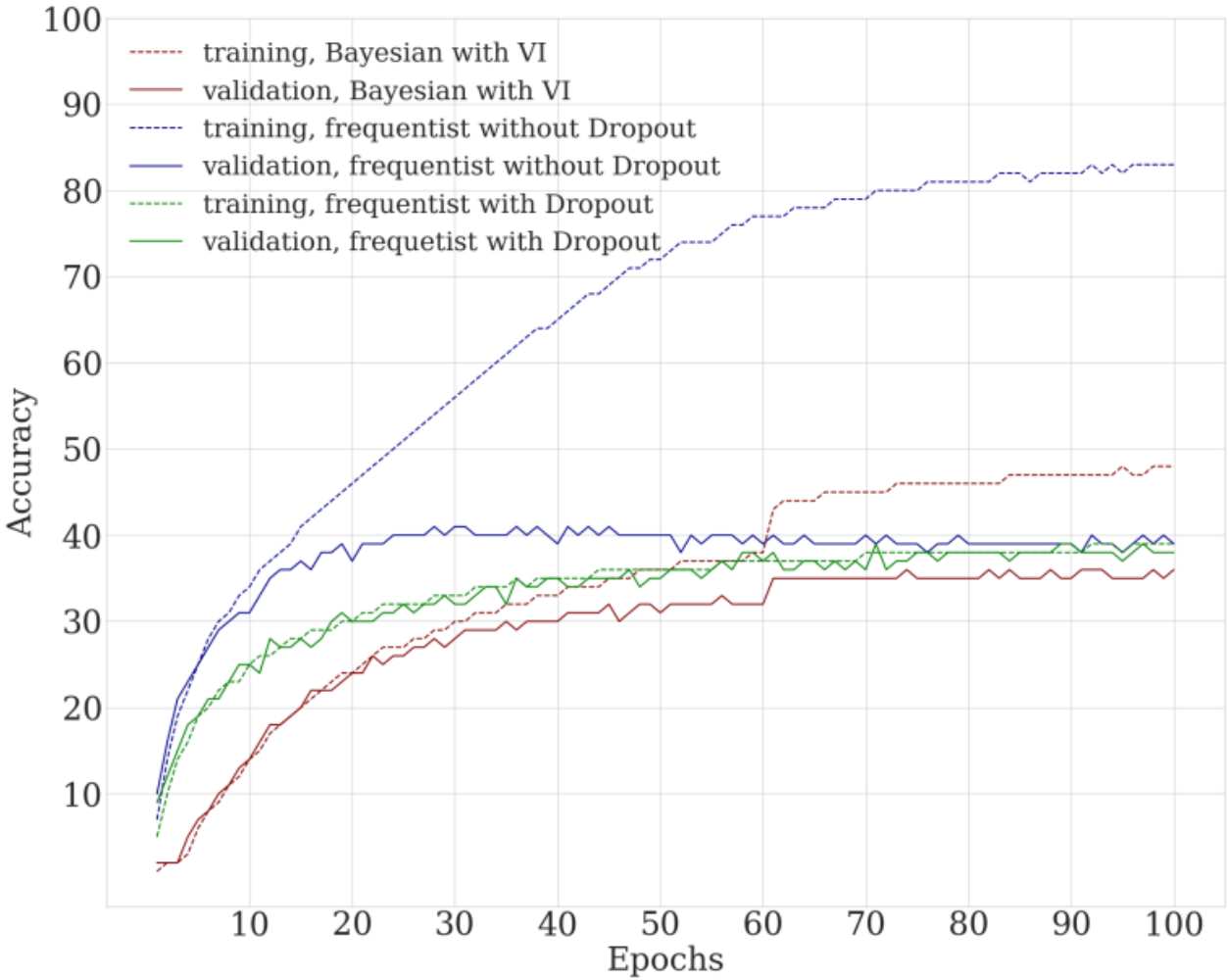

Figure: AlexNet trained on CIFAR-100 by Bayesian and frequintis inference

Results

	MNIST	CIFAR-10	CIFAR-100
Bayesian VGG (with VI)	99	86	45
Frequentist VGG	99	85	48
Bayesian AlexNet (with VI)	99	73	36
Frequentist AlexNet	99	73	38
Bayesian LeNet-5 (with VI)	98	69	31
Frequentist LeNet-5	98	68	33
Bayesian LeNet-5 (with Dropout)	99	83	

Figure: Comparison of validation accuracies for different architectures with variational inference and frequentist inference

Results

	Aleatoric uncertainty	Epistemic uncertainty	Validation accuracy
Bayesian VGG (MNIST)	0.00110	0.0004	99
Bayesian VGG (CIFAR-10)	0.00099	0.0013	85
Bayesian AlexNet (MNIST)	0.00110	0.0019	99
Bayesian AlexNet (CIFAR-10)	0.00099	0.0002	73
Bayesian LeNet-5 (MNIST)	0.00110	0.0026	98
Bayesian LeNet-5 (CIFAR-10)	0.00099	0.0404	69

Figure: Aleatoric and epistemic uncertainity for Bayesian VGG, ALexNet, LeNet-5 calculated for MNIST and CIFAR-10, computed using Softplus normalization

Concluding remarks

- Bayesian CNNs trained by variational inference achieve valiadation accuracies comparable to the ones trained by frequentist inference.
- Bayesian networks incorporate naturally effects of regularization and are less prone to overfitting compared to frequentists networks.
- We see correlating pattern between validation accuracies and epistemic uncertainties, with increasing validation accuracy , epistemic uncertainity decreases.

Appendix

Proof of (1)

$$
\begin{aligned}
KL[q_\theta(w|D)||p(w|D)] &= \int q_\theta(w|D) log \frac{q_\theta(w|D)}{p(w|D)} d\theta \\
&= \int q_\theta(w|D) log \frac{q_\theta(w|D)p(D)}{p(D|w)p(w)} d\theta \\
&= \int q_\theta(w|D) log \frac{q_\theta(w|D)}{p(w)} d\theta + \mathbb{E}_q[log\ p(D)] \\
&- \mathbb{E}_q[logp(D|w)] \\
&= KL[q_\theta(w|D)||p(w)] - \mathbb{E}_{q_\theta(w|D)}[logp(w|D)] \\
&+ log\ p(D)
\end{aligned}
$$

References

- Uncertainity Estimations by Softplus normalization in Bayesian Convolutional Neural Networks with Variational Inference Kumar Shridhar, Felix Laumann