Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

Yeseul Jeon

January 10, 2021

Contents

About paper:

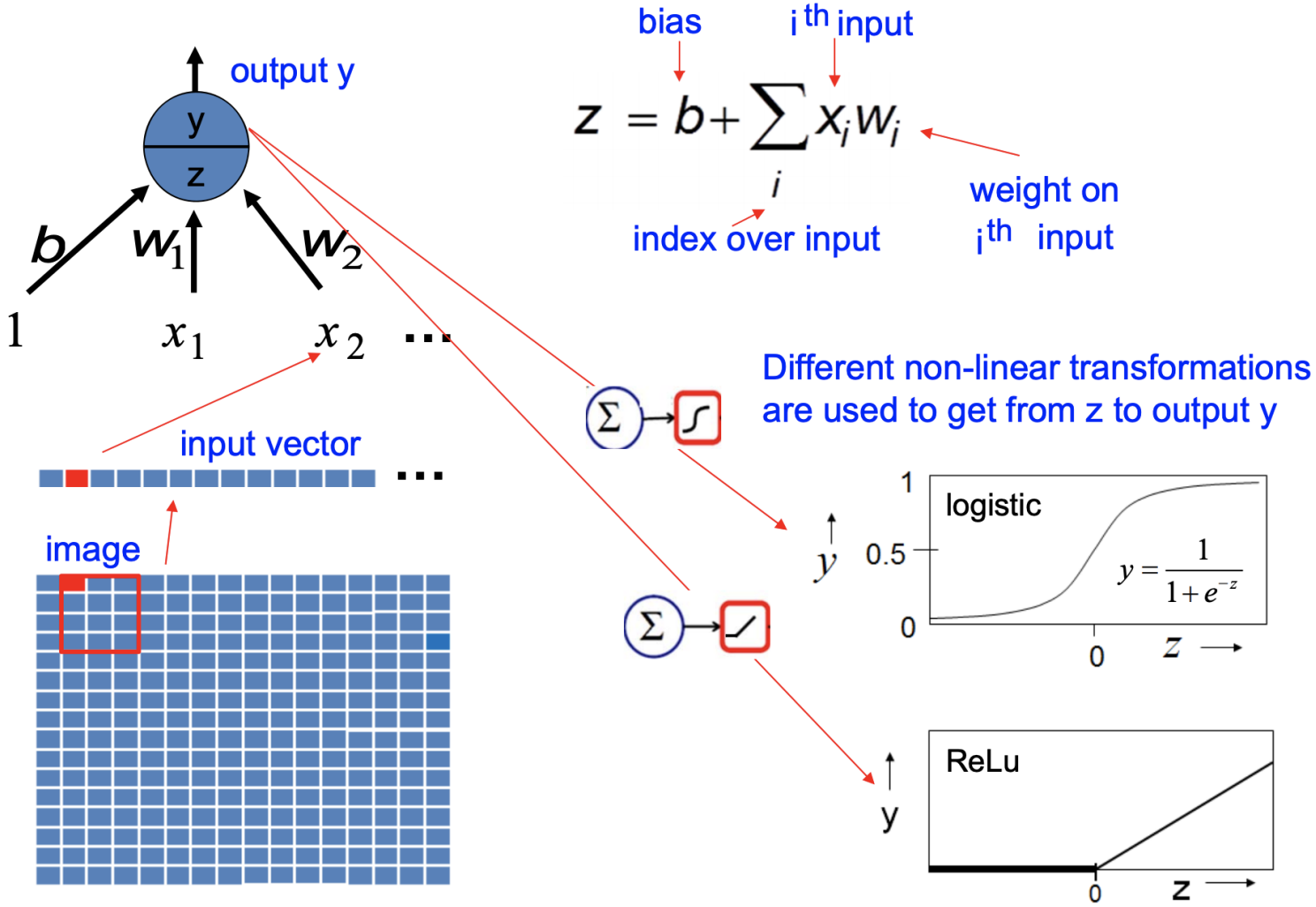
Dropout as a Bayesian Approximation:
Representing Model Uncertainty in Deep Learning

Yarin Gal YG279@CAM.AC.UK
Zoubin Ghahramani ZG201@CAM.AC.UK
University of Cambridge

Abstract

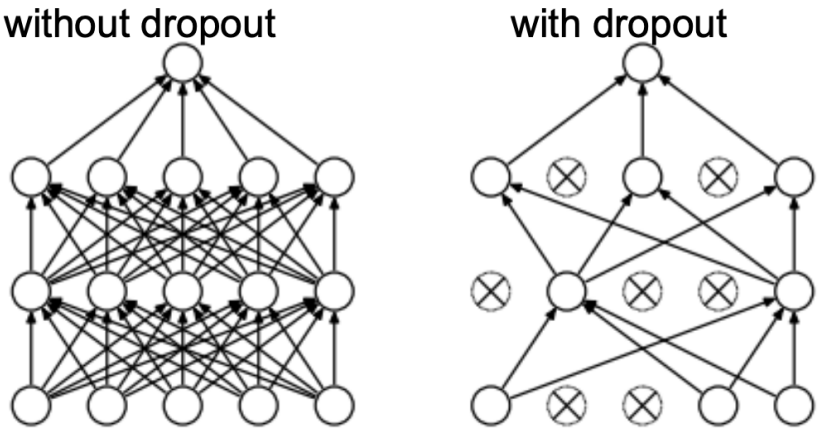
- **Limitation:**
  - Deep NN gives us only point estimates with no uncertainty information.
  - Bayesian modeling we can get a measure of uncertainty by evaluating the posterior distribution of the NN weights.
  - Bayesian model usually come with a prohibitive computational cost.
- **Idea:**
  - Develop a new theoretical framework casting dropout training in deep neural networks as approximate Bayesian inference in deep Gaussian process
  - This theory gives us tools to model uncertainty with dropout NNs - extracting information from existing models that has been thrown away so far.

- **Logistic Regression or NN with 1 neuron:**



Introduction

- **Dropout:**
  - At each training step we remove random nodes with a probability of p resulting in a sparse version of the full net and we use back-propagation to update the weights
  - By averaging over theses models we should be able to "reduce noise", "over-fitting".



(a) Standard Neural Net (b) After applying dropout.

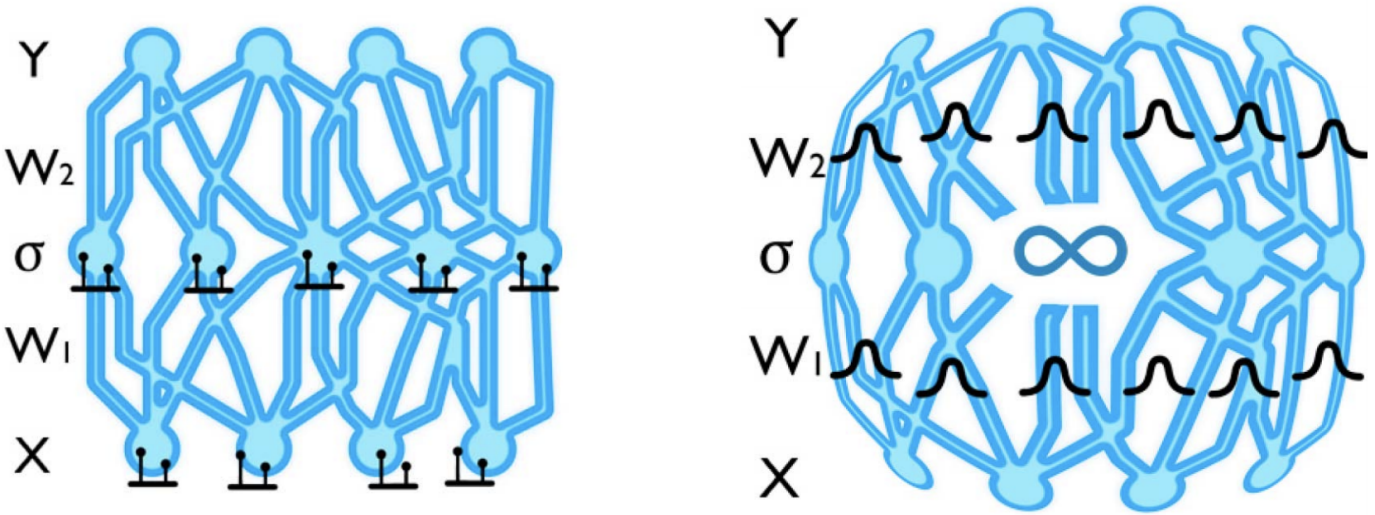
Dropout

$$\mathcal{L}_{\text{dropout}} := \frac{1}{N}\sum_{i=1}^{N} E(\mathbf{y}_i, \widehat{\mathbf{y}}_i) + \lambda \sum_{i=1}^{L} \left(||\mathbf{W}_i||_2^2 + ||\mathbf{b}_i||_2^2 \right).$$

- NN Optimisation
  - A regularisation term is added.
  - E($y_i, \hat{y}_i$): Error function
  - Sample binary variables for every input point and for every network unit in each layer.

Dropout as a Bayesian Approximation

- **Dropout vs Bayesian NN**



- Dropout: Remove random nodes with a probability p
- Bayesian NN: Update the posterior distribution of the weights

Approximation of full Bayesian learning

- **Posterior Distribution**

likelihood prior

$$p(\omega \mid \mathrm{X}, \mathrm{Y}) = \frac{p(\mathrm{Y} \mid \omega, \mathrm{X}) \cdot p(\omega)}{p(\mathrm{Y} \mid \mathrm{X})}$$

normalizer=marginal likelihood

- We can approximate the posterior distribution for the model parameters via Variation Inference
- replacing the posterior distribution at the observed data $p(w|X, Y)$ with a member $q(w)$ of a simpler distribution family Q that minimizes the Kullback–Leibler divergence to the posterior

Variational Inference Approximated full Bayesian Learning

- **Variational Inference**

$$\mathrm{KL}\left(q_\theta(\omega) \,\|\, \mathrm{p}(\omega \,|\, \mathbf{X}, \mathbf{Y})\right) = \int q_\theta(\omega) \log \frac{q_\theta(\omega)}{\mathrm{p}(\omega \,|\, \mathbf{X}, \mathbf{Y})} d\omega = E_q\left[\log\left(q_\theta(\omega)\right) - \log\left(\mathrm{p}(\omega \,|\, \mathbf{X}, \mathbf{Y})\right)\right]$$

- Approximated $p(w|X, Y)$ with simple distribution $q_\theta(w)$
- Minimize Kullback Leibler divergence of q from the posterior w.r.t to the variational parameters θ:

- **Variational Inference**

$$\mathrm{L} = \log\left(p(\mathrm{Y}\mid\mathrm{X})\right) = \log\int p\left(\mathrm{Y}|\mathrm{X},\omega\right)\cdot p\left(\omega\right)d\omega = \log\int p\left(\mathrm{Y}|\mathrm{X},\omega\right)\cdot p\left(\omega\right)\frac{q_\theta(\omega)}{q_\theta(\omega)}d\omega = \log\left(E_{q_\theta}\left[\frac{p\left(\mathrm{Y}|\mathrm{X},\omega\right)\cdot p\left(\omega\right)}{q_\theta(\omega)}\right]\right)$$

$$\geq E_{q_\theta}\left[\log\left(\frac{p\left(\mathrm{Y}|\mathrm{X},\omega\right)\cdot p\left(\omega\right)}{q_\theta(\omega)}\right)\right] = E_{q_\theta}\left[\log\left(p\left(\mathrm{Y}|\mathrm{X},\omega\right)\right)+\log\left(\frac{p\left(\omega\right)}{q_\theta(\omega)}\right)\right] = E_{q_\theta}\left[\log\left(p\left(\mathrm{Y}|\mathrm{X},\omega\right)\right)\right]-E_{q_\theta}\left[\log\left(\frac{q_\theta(\omega)}{p\left(\omega\right)}\right)\right]$$

$$= \int q_\theta(\omega)\cdot\log\left(p\left(\mathrm{Y}|\mathrm{X},\omega\right)\right)d\omega - KL\left(q_\theta(\omega)\,\|\,\mathrm{p}(\omega)\right)$$

- Minimizing the KL divergence of q from the posterior distribution w.r.t θ is
- equivalent to maximizing a lower bound of the log marginal likelihood w.r.t θ

Variational Inference Approximated full Bayesian Learning

- **MC integration to approximate L**

$$\mathrm{L}_{VI}(\theta) := \int q_\theta(\omega) \cdot \log\Big(p\big(\mathrm{Y}|\mathrm{X},\omega\big)\Big) d\omega - KL\big(q_\theta(\omega) \,\|\, \mathrm{p}(\omega)\big)$$

- Since this integral is not tractable for almost all q therefore we will MC integration to approximate this quantity.
- sample $\hat{w}$ from q and each sampling step the integral is replaced by $log(p(Y|X, \hat{w}))$.

Variational Inference Approximated full Bayesian Learning

- **Stochastic Inference**

$$\mathrm{L}_{VI}(\theta) := \int q_\theta(\omega) \cdot \log\Big(p\big(\mathrm{Y}|\mathrm{X},\omega\big)\Big) d\omega - KL\big(q_\theta(\omega) \,\|\, \mathrm{p}(\omega)\big)$$

$$\hat{\mathrm{L}}(\theta) := \log\Big(p\big(\mathrm{Y}|\mathrm{X},\hat{\omega}\big)\Big) - KL\big(q_\theta(\omega) \,\|\, \mathrm{p}(\omega)\big)$$

  - For inference repeatedly do:
- Sample $\hat{w} \sim q_\theta(w)$
- Do one step of minimization w.r.t θ: $\hat{L}(\theta)$

What kind of q-distribution should we use?

- **The deep Gaussian process**

$$\mathbf{K}(\mathbf{x},\mathbf{y}) = \int p(\mathbf{w})p(b)\sigma(\mathbf{w}^T\mathbf{x}+b)\sigma(\mathbf{w}^T\mathbf{y}+b)\mathbf{dw}db$$

$$\mathbf{w}_k \sim p(\mathbf{w}),\ b_k \sim p(b),$$
$$\mathbf{W}_1 = [\mathbf{w}_k]_{k=1}^K, \mathbf{b} = [b_k]_{k=1}^K$$
$$\widehat{\mathbf{K}}(\mathbf{x},\mathbf{y}) = \frac{1}{K}\sum_{k=1}^{K}\sigma(\mathbf{w}_k^T\mathbf{x}+b_k)\sigma(\mathbf{w}_k^T\mathbf{y}+b_k)$$
$$\mathbf{F} \mid \mathbf{X}, \mathbf{W}_1, \mathbf{b} \sim \mathcal{N}(\mathbf{0}, \widehat{\mathbf{K}}(\mathbf{X},\mathbf{X}))$$
$$\mathbf{Y} \mid \mathbf{F} \sim \mathcal{N}(\mathbf{F}, \tau^{-1}\mathbf{I}_N),$$

- W_i be a random matrix of dimensions $K_i \times K_{i-1}$ for each layer i.
- A prior let each row of W_i distribute according to the p(w) above.
- Assume vectors m_i of dimensions K_i for each GP layer.

What kind of q-distribution should we use?

- **Prediction Probability of deep GP model**

$$p(\mathbf{y}|\mathbf{x},\mathbf{X},\mathbf{Y}) = \int p(\mathbf{y}|\mathbf{x},\boldsymbol{\omega})p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})\mathrm{d}\boldsymbol{\omega}$$

$$p(\mathbf{y}|\mathbf{x},\boldsymbol{\omega}) = \mathcal{N}\left(\mathbf{y};\widehat{\mathbf{y}}(\mathbf{x},\boldsymbol{\omega}),\tau^{-1}\mathbf{I}_D\right)$$

$$\widehat{\mathbf{y}}\left(\mathbf{x},\boldsymbol{\omega}=\{\mathbf{W}_1,...,\mathbf{W}_L\}\right)$$

- The posterior distribution $p(w|X,Y)$ is intractable.
- Use $q(w)$, a distribution over matrices whose columns are randomly set to zero
- To approximated the intractable posterior

Define the structure of the approximate distribution q

- **q(w) distribution**

$$\mathbf{W}_i = \mathbf{M}_i \cdot \text{diag}([\mathbf{z}_{i,j}]_{j=1}^{K_i})$$

$$\mathbf{z}_{i,j} \sim \text{Bernoulli}(p_i) \text{ for } i = 1, ..., L,\ j = 1, ..., K_{i-1}$$

$$-\int q(\boldsymbol{\omega}) \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} + \text{KL}(q(\boldsymbol{\omega})||p(\boldsymbol{\omega})).$$

We rewrite the first term as a sum

$$-\sum_{n=1}^{N} \int q(\boldsymbol{\omega}) \log p(\mathbf{y}_n|\mathbf{x}_n, \boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega}$$

- Some probabilities p
- Matrices M_i as variational parameters.
- The variational dsitribuiton q(w) is highly mulitmodal
- This corresponds to the frequencies in the sparse spectrum GP approximation

Define the structure of the approximate distribution q

- **KL with GP and MC**

$$\mathcal{L}_{\text{GP-MC}} \propto \frac{1}{N}\sum_{n=1}^{N}\frac{-\log p(\mathbf{y}_n|\mathbf{x}_n,\widehat{\boldsymbol{\omega}}_n)}{\tau} + \sum_{i=1}^{L}\left(\frac{p_i l^2}{2\tau N}||\mathbf{M}_i||_2^2 + \frac{l^2}{2\tau N}||\mathbf{m}_i||_2^2\right).$$

$$\mathcal{L}_{\text{GP-MC}} \propto -\frac{1}{2N}\sum_{n=1}^{N}||\mathbf{y}_n - \widehat{\mathbf{y}}_n||_2^2 - \frac{l^2 p_1}{2\tau N}||\mathbf{M}_1||_2^2 - \frac{K p_2}{2\tau N}||\mathbf{M}_2||_2^2 - \frac{l'^2}{2\tau N}||\mathbf{m}||_2^2.$$

- The sampled $\hat{w}$ result in realization from the Bernoulli distribution z_i^n equivalent to the binary variables in the dropout case.

Define the structure of the approximate distribution q

- $q_\theta(w)$

$$q_{\mathbf{M}_i}(\mathbf{W}_i) = \mathbf{M}_i \cdot \mathrm{diag}\left(\lfloor \mathbf{z}_{i,j} \rfloor\right)$$

$$\mathbf{z}_{i,j} \sim \mathrm{Bernoulli}(\mathrm{p}_i)$$

$$\mathbf{W}_i \sim q_{\mathbf{M}_i}(\mathbf{W}_i)$$

Approximate posterior of model parameters

$$\mathbf{M}_i = \mathrm{mean}(\mathbf{W}_i)$$

$\mathbf{M}_i$ is as variational parameter of q

Sampling the diagonal elements **z** from a Bernoulli is identical to randomly setting columns of M to zero which is identical to randomly setting units of the network to zero -> dropout!

- Bernoullis are computationally cheap to get multi-modality

Obtaining Model Uncertainty

- **Prediction distribution for uncertainty estimation**
  - To get an approximation of the posterior via training
  1) Randomly set columns of M_i to zero (do dropout)
  2) Update the weights by doing one step
  - To sample from the learned approximate posterior we just can do dropout during the test time when using the trained NN for prediction.
  - From the received predictions we can estimate the predictive distribution and from this different uncertainty measures such as the variance.

Obtaining Model Uncertainty

- **Prediction distribution for uncertainty estimation**
  - To sample from the learned approximate posterior do dropout during the test time when using the trained NN for prediction.
  - From the received predictions we can estimate the predictive distribution and from this different uncertainty measures such as the variance.

Obtaining Model Uncertainty

- **Prediction distribution for uncertainty estimation**

$$\begin{aligned}\log p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) &= \log \int p(\mathbf{y}^*|\mathbf{x}^*, \boldsymbol{\omega}) p(\boldsymbol{\omega}|\mathbf{X}, \mathbf{Y}) \mathrm{d}\boldsymbol{\omega} \\ &\approx \log \int p(\mathbf{y}^*|\mathbf{x}^*, \boldsymbol{\omega}) q(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \\ &\approx \log \left(\frac{1}{T} \sum_{t=1}^{T} p(y^*|\mathbf{x}^*, \boldsymbol{\omega}_t) \right)\end{aligned}$$

- Given a dataset X, Y and a new data point x^* we can calculate the probability of possible output values y^* using the predictive probability $p(y^*|x^*, X, Y)$.

Obtaining Model Uncertainty

- **Mean and Variance using MC-dropout**

$$\mathbb{E}_{q(\mathbf{y}^*|\mathbf{x}^*)}(\mathbf{y}^*) \approx \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathbf{y}}^*(\mathbf{x}^*, \mathbf{W}_1^t, ..., \mathbf{W}_L^t)$$

$$\begin{aligned}\mathrm{Var}_{q(\mathbf{y}^*|\mathbf{x}^*)}(\mathbf{y}^*) &\approx \tau^{-1}\mathbf{I}_D \\ &+ \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathbf{y}}^*(\mathbf{x}^*, \mathbf{W}_1^t, ..., \mathbf{W}_L^t)^T\widehat{\mathbf{y}}^*(\mathbf{x}^*, \mathbf{W}_1^t, ..., \mathbf{W}_L^t) \\ &- \mathbb{E}_{q(\mathbf{y}^*|\mathbf{x}^*)}(\mathbf{y}^*)^T\mathbb{E}_{q(\mathbf{y}^*|\mathbf{x}^*)}(\mathbf{y}^*)\end{aligned}$$

Conclusion

- **Estimate Model uncertainty with MC-dropout**
  - Can represents model uncertainty in deep learning, better model regularisation, computationally efficient Bayesian convolutional neural networks.
  - A neural network with arbitrary depth and non-linearities and with dropout applied before every weight layer is mathematically equivalent to an approximation to the deep Gaussian process (marginalised over its covariance function parameters).