Representing Model Uncertainty in Deep Learning Using Dropout

Bayesian methods in Statistics and Machine Learning

Pouya Moeini

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

Proceedings of The 33rd International Conference on Machine Learning, PMLR 48:1050-1059, 2016

What are we seeking in this paper ?

Outline

- Introduction
- Mathematical Background
- Dropout as a Bayesian Approximation
- Obtaining Model Uncertainty
- Conclusions and Experiments

Neural Network

- What is Neural Networks and How do they work?
- Pros and Cons of Deep Learning
- Model Uncertainty in Deep Learning

Model Uncertainty

- What is Model uncertainty?
- Why is it useful?
- How is Model uncertainty in Deep Learning?
- How to measure it?

Bayesian inference as Model uncertainty

- Bayesian Inference?
- How to measure model uncertainty using Bayesian inference?
- Challenges within?

Outline

- Introduction
- Mathematical Background
- Dropout as a Bayesian Approximation
- Obtaining Model Uncertainty
- Conclusions and Experiments

Dropout

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Nitish Srivastava NITISH@CS.TORONTO.EDU
Geoffrey Hinton HINTON@CS.TORONTO.EDU
Alex Krizhevsky KRIZ@CS.TORONTO.EDU
Ilya Sutskever ILYA@CS.TORONTO.EDU
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
10 Kings College Road, Rm 3302
Toronto, Ontario, M5S 3G4, Canada.

Editor: Yoshua Bengio

Dropout

- What is dropout and how does it work?
- How is it useful?
- How to formulate it?

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

Dropout

- Consider the case of a single hidden layer
- Denote by W_1, W_2 the weight matrices connecting the first layer to the hidden layer and connecting the hidden layer to the output layer respectively
- Denote by b the biases by which we shift the input of the non-linearity
- A standard NN model with input X would output:

$$\hat{Y} = \sigma(xW_1 + b)W_2$$

- Applying dropout method: $z_{1,q} \sim Bernouli(p_1)$, $z_{2,k} \sim Bernouli(p_2)$
- The results after applying dropout would be:

$$\hat{Y} = \sigma(x(z_1 W_1) + b)z_2 W_2$$

- Minimization objective (with l2-reg):
  - $\mathcal{L}_{dropout} = E + \lambda_1 \parallel W_1 \parallel_2^2 + \lambda_2 \parallel W_2 \parallel_2^2 + \lambda_3 \parallel b \parallel_2^2$

Gaussian Process

Gaussian Process

- The Gaussian process (GP) is a powerful tool in statistics that allows us to <u>model distributions over functions.</u>
- Given a training dataset consisting of N inputs $\{x_1, \dots, x_N\}$ and their corresponding outputs $\{y_1, \dots, y_N\}$
- we would like to estimate a function $y = f(x)$ that is likely to have generated our observations
- What is a function that is likely to have generated our data?

$$p(f|X, Y) \propto p(f)p(Y|f, X)$$

- $p(f)$: prior distribution over the space of functions, represents our prior belief as to which functions are more likely and which are less likely to have generated our data

Gaussian Process

- A Gaussian process is a stochastic process such that any finite sub collection of random variables has a multivariate Gaussian distribution
- A collection of random variables $\{f(x): x \in \mathcal{X}\}$ is said to be drawn from a Gaussian process with mean function $m(.)$ and covariance function $k(.,.)$ f for any finite set of elements $x_1, \dots, x_n \in \mathcal{X}$, the associated finite set of random variables $f(x_1), f(x_2), \dots f(x_n)$ have distribution:

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ \dots \\ f(x_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ m(x_2) \\ \dots \\ m(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{bmatrix}\right)$$

$$f(.) \sim \mathcal{GP}(m(.), k(.,.))$$

Gaussian Process

Variational Inference

Variational Inference: A Review for Statisticians

David M. Blei[a], Alp Kucukelbir[b], and Jon D. McAuliffe[c]

[a]Department of Computer Science and Statistics, Columbia University, New York, NY; [b]Department of Computer Science, Columbia University, New York, NY; [c]Department of Statistics, University of California, Berkeley, CA

ABSTRACT

One of the core problems of modern statistics is to approximate difficult-to-compute probability densities. This problem is especially important in Bayesian statistics, which frames all inference about unknown quantities as a calculation involving the posterior density. In this article, we review variational inference (VI), a method from machine learning that approximates probability densities through optimization. VI has been used in many applications and tends to be faster than classical methods, such as Markov chain Monte Carlo sampling. The idea behind VI is to first posit a family of densities and then to find a member of that family which is close to the target density. Closeness is measured by Kullback–Leibler divergence. We review the ideas behind mean-field variational inference, discuss the special case of VI applied to exponential family models, present a full example with a Bayesian mixture of Gaussians, and derive a variant that uses stochastic optimization to scale up to massive data. We discuss modern research in VI and highlight important open problems. VI is powerful, but it is not yet well understood. Our hope in writing this article is to catalyze statistical research on this class of algorithms. Supplementary materials for this article are available online.

ARTICLE HISTORY

Received January 2016
Revised December 2016

KEYWORDS

Algorithms; Computationally intensive methods; Statistical computing

Variational Inference

- Try to compute $p(w|X,Y)$ and challenges within

- Try to approximate it with $q(w)$

$$q^* = \min_q \mathcal{D}_{KL}(q(w) \parallel p(w|X,Y))$$

- Maximize log evidence lower bound as follows

$$\mathcal{L} = \int q(w)p(Y|w,X)dw - \mathcal{D}_{KL}(p(w) \parallel q(w))$$

Outline

- Introduction
- Mathematical Background
- Dropout as a Bayesian Approximation
- Obtaining Model Uncertainty
- Conclusions and Experiments

Dropout as a Bayesian Approximation

Goal: Showing that **a neural network** with **arbitrary depth** and non-linearities, **with dropout applied before every weight layer**, is mathematically <u>equivalent</u> to an approximation to the probabilistic **deep Gaussian process**

A net with nonparametric activation functions corresponding to a 3-layer deep GP

A Gaussian Process Approximation

- Covariance Function in Gaussian process
- Definition of predictive function:

$$p(y^*|x^*, X, Y) = \int p(y^*|x^*, w)p(w|X, Y)\, dw$$

- For a deep gaussian Process:

$$p(y^*|x^*, w) = \mathcal{N}(\hat{y}(X, w), \tau^{-1}I)$$

- What about $p(w|X, Y)$?

Variational Inference

$\longrightarrow$

Variational methods to Approx. posterior

- The goal was try to obtain posterior distribution
- Use variational methods to approximate wanted distribution!

$$q(w) \approx p(w|X,Y)$$

- Suppose we define q as follows:

$$q(w): W_i \sim M_i.diag([z_{i,j}]^{k_i}{}_{j=1})$$

$$z_{i,j} \sim Bernouli(p_i)$$

$$M_i, p_i \leftarrow variational\ hyperparameters$$

Evaluation and Optimization ELBO

- Variational inference → Maximize ELBO

- After applying some approximation and Monte-Carlo methods, the optimization problem would be as follows:

$$\mathcal{L}_{GP} \propto \frac{1}{N}\sum_{n} -\frac{1}{\tau}\log p(y_n|x_n,\widehat{w}) + \sum_{i}\left(\frac{p_i l^2}{2\tau N} \parallel M_i \parallel_2^2 + \frac{l^2}{2\tau N} \parallel m_i \parallel_2^2\right)$$

$\tau, l \rightarrow$ variational hyper parameters

$\widehat{w} \rightarrow$ obtained from Mont-Carlo methods

Loss function in Deep Gaussian Process

Loss Function with dropout usage

- Let $\hat{y}$ be be the output of a NN model with L layers and a loss function $E(.,.)$ (such as the softmax loss or the Euclidean loss (square loss))
- We denote by W_i the NN's weight matrices of dimensions $K_{i-1} \times K_i$ and by b_i the bias vectors of dimensions K_i for each layer i = 1, ..., L
- We often use L2-regularisation weighted by some weight decay λ
- Resulting in a minimization objective as:

$$\mathcal{L}_{dropout} = \frac{1}{N} \sum_i E(y_i, \hat{y}_i) + \lambda \sum_j (\| w_j \|_2^2 + \| b_j \|_2^2)$$

Comparison of two losses

We can show that the optimization solution of both losses, results in the same answer!

$$\text{argmax}\ \mathcal{L}_{Gp} \equiv \text{argmin}\ \mathcal{L}_{dropout}$$

Outline

- Introduction
- Mathematical Background
- Dropout as a Bayesian Approximation
- Obtaining Model Uncertainty
- Conclusions and Experiments

Obtaining Model Uncertainty

- Predictive function:

$$p(y|x, D) = \int p(y|x, w)p(w|D)dw$$

- Variational inference approximation: $q(w) \equiv p(w|D)$

$$q(y|x) = \int p(y|x, w)q(w)dw$$

Obtaining Model Uncertainty

- Use moment matching to approximate the mentioned equation

$$\mathbb{E}_{q(y|x)}[y] \approx \frac{1}{T}\sum_{t}\hat{y}(x,W) \quad , \qquad \mathbb{E}_{q(y|x)}[y^T y] \approx \tau^{-1} + \frac{1}{T}(\hat{y})^T\hat{y}$$

- Results in:

$$\log p(y|x,D) \approx logsumexp\left(-\frac{1}{2}\tau \parallel y-\hat{y} \parallel_2^2\right) - \log(T) - \frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\tau^{-1})$$

Outline

- Introduction
- Mathematical Background
- Dropout as a Bayesian Approximation
- Obtaining Model Uncertainty
- Conclusions and Experiments

Conclusions

We have shown that 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

We have built a probabilistic interpretation of dropout which allowed us to obtain model uncertainty out of existing deep learning models

Experiments

(a) Standard dropout with weight averaging

(b) Gaussian process with SE covariance function

(c) MC dropout with ReLU non-linearities

(d) MC dropout with TanH non-linearities

Outline

- Introduction
- Mathematical Background
- Dropout as a Bayesian Approximation
- Obtaining Model Uncertainty
- Conclusions and Experiments

References

- Z. G. Yarin Gal, “Dropout as a bayesian approximation: Representing model uncertaintyin deep learning,”Proceedings of The 33rd International Conference on Machine Learning,2016
- N. D. L. Andreas C. Damianou, “Deep gaussian processes,”Proceedings of the SixteenthInternational Conference on Artificial Intelligence and Statistics, 2013.
- A. K. Nitish Srivastava, Geoffrey Hinton, “Dropout: A simple way to prevent neuralnetworks from overfitting,”Journal of Machine Learning Research, 2014.

THANK YOU

For any question, please contact me: pooyamoini2000@gmail.com