Dropout as a Bayesian approximation: Representing model uncertainty in deep learning

Authors	Yarin Gal, Zoubin Ghahramani
Publication date	2015/6/6
Conference	Proceedings of the 33rd International Conference on Machine Learning (ICML-16)
Total citations	Cited by 11812

The authors introduce the limitations of traditional deep learning models in capturing model uncertainty and highlight that Bayesian models, while capable of handling uncertainty through probabilistic reasoning, often come with substantial computational costs, making them less practical for deep learning applications. This presentation provides a detailed yet concise understanding of its contributions to the field of deep learning and Bayesian methods.

Bayesian Interpretation of Dropout

Dropout can be viewed as a variational approximation to a deep Gaussian process (GP). The covariance function of a GP with an element-wise non-linearity $\sigma(\cdot)$ is given by:

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

In this framework, instead of considering a fixed weight W, they model the weights as random variables, thereby capturing the uncertainty in the model predictions.

1. **Define Model**
   Express neural network with weights θ under dropout.
2. **Stochastic Passes**
   Run T stochastic forward passes through the network.
3. **Weight Sampling**
   Sample weights after applying dropout.
4. **Bayesian Approximation**
   View dropout as approximation to Bayesian inference.

Dropout Training Objective & Variational Formulation

The standard dropout training objective with L2 regularization is expressed as:

$$L_{\text{dropout}} = \frac{1}{N} \sum_{i=1}^{N} E(y_i, \hat{y}_i) + \lambda \sum_{i=1}^{L} \left(\|W_i\|_2^2 + \|b_i\|_2^2 \right),$$

where $E(\cdot, \cdot)$ is the loss function (e.g., softmax loss or Euclidean loss) and λ is the weight decay parameter.

For each layer i, dropout is implemented via:

$$W_i = M_i \cdot \text{diag}([z_{i,j}]_{j=1}^{K_i}), \quad z_{i,j} \sim \text{Bernoulli}(p_i),$$

which introduces binary random variables

that drop out neurons with probability 1−p_i

Deep Gaussian Process & Predictive Distribution

The predictive probability for a deep Gaussian process model is defined as:

$$p(y|x, X, Y) = \int p(y|x, \omega)\, p(\omega|X, Y)\, d\omega,$$

where ω={W_1, …, W_L} represents the set of all weight matrices.

Since the posterior p(ω | X,Y) is intractable, they introduce a variational distribution q(ω) that approximates it. This variational form effectively turns dropout into a Bayesian treatment where the model averages over many possible weight configurations.

KL Divergence Minimization & Monte Carlo Approximation

The variational objective minimizes the Kullback–Leibler (KL) divergence between the approximate posterior q(ω) and the true posterior p(ω∣X,Y):

$$\mathcal{L} = -\int q(\omega)\ \log p(Y|X,\omega)\, d\omega + \mathrm{KL}(q(\omega)\ \|\ p(\omega)).$$

This integral is approximated using Monte Carlo sampling:

$$-\frac{1}{T}\sum_{t=1}^{T}\log p(y_n|x_n,\omega_t),$$

where each ω_t is a sample drawn from q(ω).

Monte Carlo Dropout for Uncertainty Estimation

The predictive distribution under the variational approximation is:

$$q(y^*|x^*) = \int p(y^*|x^*, \omega)\, q(\omega)\, d\omega.$$

The predictive mean is approximated by averaging over T stochastic forward passes:

$$\mathbb{E}[y^*|x^*] \approx \frac{1}{T} \sum_{t=1}^{T} \hat{y}^*(x^*, \omega_t),$$

and the predictive variance is estimated as:

$$\mathbf{Var}[y^*|x^*] \approx \tau^{-1} I_D + \frac{1}{T} \sum_{t=1}^{T} \hat{y}^*(x^*, \omega_t) \hat{y}^*(x^*, \omega_t)^T - \mathbb{E}[y^*|x^*] \mathbb{E}[y^*|x^*]^T.$$

Model Precision & Predictive Log-Likelihood

The model precision τ is related to the dropout probability p, the prior length-scale l, the number of data points N, and the weight decay λ via:

$$\tau = \frac{p\, l^2}{2N\lambda}.$$

Furthermore, the predictive log-likelihood for regression is approximated by:

$$\log p(y^*|x^*, X, Y) \approx \log\left(\frac{1}{T}\sum_{t=1}^{T}\exp\left(-\frac{\tau}{2}\|y - \hat{y}_t\|^2\right)\right) - \log T - \frac{1}{2}\log(2\pi) - \frac{1}{2}\log\left(\tau^{-1}\right).$$

Empirical Evaluation & Performance Metrics

The approach is validated on both regression and classification tasks. For regression, key performance metrics include the Root Mean Square Error (RMSE) and predictive log-likelihood:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}.$$

In classification, uncertainty is reflected in the distribution of softmax outputs. Multiple stochastic forward passes reveal the variability in predictions, ensuring that high softmax scores do not necessarily imply high model confidence.

Impact of Dropout Rate

The authors find a notable relationship between the dropout probability p and the model's predictive log-likelihood and RMSE. Their findings show that:

$$\frac{\partial \text{Log-Likelihood}}{\partial p} > 0$$

indicating that as the dropout rate increases, the uncertainty estimation improves significantly, supporting the use of dropout in practical applications.

1	Increase Dropout Rate
2	Improve Uncertainty Estimation
3	Enhance Predictive Performance

Proposed idea: Adaptive Dropout Rates

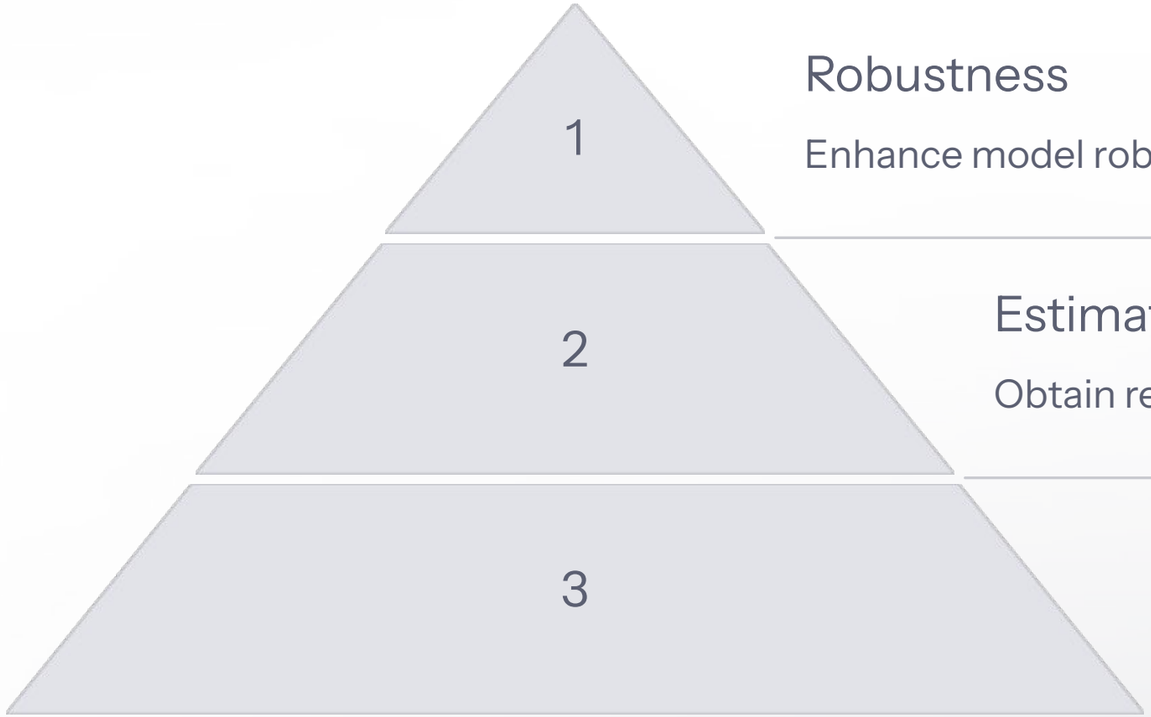
A promising extension is to develop an **adaptive dropout mechanism** where the dropout probability p is adjusted dynamically based on the model's uncertainty. One can propose an adaptive dropout rate as a function of the predictive variance U(x):

$$p(x) = \sigma(\alpha U(x) + \beta),$$

where $\sigma(\cdot)$ is the sigmoid function, and α and β are parameters that could be learned during training. This adaptive mechanism would allow the network to modulate its regularization strength in regions with high uncertainty, potentially improving both prediction accuracy and uncertainty calibration.

Conclusion

The authors conclude that interpreting dropout as a Bayesian approximation opens new avenues for uncertainty representation in deep learning. By utilizing dropout effectively, practitioners can enhance model robustness and obtain reliable uncertainty estimates without incurring significant computational costs. This approach enhances model reliability.



Robustness

Enhance model robustness

Estimates

Obtain reliable uncertainty estimates

Efficiency

Avoid significant computational costs