Risk versus Uncertainty in Deep Learning: Bayes, Bootstrap and the Dangers of Dropout

Ian Osband
Google Deepmind
iosband@google.com

1 Introduction

The "Big Data" revolution is spawning systems designed to make decisions from data. In particular, deep learning methods have emerged as the state of the art method in many important breakthroughs [18, 20, 28]. This is due to the statistical flexibility and computational scalability of large and deep neural networks which allows them to harness the information of large and rich datasets. At the same time, elementary decision theory shows that the only admissible decision rules are Bayesian [5, 30]. Colloquially, this means that any decision rule which is not Bayesian can be strictly improved (or even exploited) by some Bayesian alternative [6]. The implication of these results is clear: combine deep learning with Bayesian inference for the best decisions from data.

There is a persistent history of research in Bayesian neural nets which never quite gained mainstream traction [19, 21]. The majority of deep learning research has evolved outside of Bayesian (or for that matter statistical) analysis [26, 18]. Recently, Bayesian deep learning has experienced a resurgence of interest [16, 1, 12]; one explanation for this revival is the rise of automated deep learning decision systems for which effective uncertainty estimates are essential [30]. In this paper we investigate several popular approaches for uncertainty estimation in neural networks. We find that several popular approximations to the *uncertainty* of a unknown neural net model are in fact approximations to the *risk* given a fixed model [17]. We review that conflating risk with uncertainty can lead to arbitrarily poor performance in a sequential decision problem [9]. We present a simple and practical solution to this problem based upon smoothed bootstrap sampling [7, 22].

2 Risk versus uncertainty

In sequential decision problems there is an important distinction between *risk* and *uncertainty* [17]. In this document we identify risk as inherent stochasticity in a model and uncertainty as the confusion over which model parameters apply. For example, a coin may have a fixed $p = 0.5$ of heads and so the outcome of any single flip holds some risk; a learning agent may also be uncertain of p. The demarcation between risk and uncertainty is tied to the specific model class, in this case a Bernoulli random variable; with a more detailed model of flip dynamics even the outcome of a coin may not be risky at all. Our distinction is that unlike risk, uncertainty captures the variability of an agent's posterior belief which *can* be resolved through statistical analysis of the appropriate data. For a learning agent looking to maximize cumulative utility through time, this distinction represents a crucial dichotomy [2].

Consider the reinforcement learning problem of an agent interacting with its environment while trying to maximize cumulative utility through time. At each timestep, the agent faces a fundamental tradeoff: by exploring uncertain states and actions the agent can learn to improve its future performance, but it may attain better short-run performance by exploiting its existing knowledge. At a high level this effect means uncertain states are more attractive since they can provide important information to the agent going forward. On the other hand, states and action with high risk are actually less attractive for an agent in both exploration and exploitation. For exploitation, any concave utility will naturally penalize risk [4]. For exploration, risk also makes any single observation less informative [27]. Although colloquially similar, risk and uncertainty can require radically different treatment.

Workshop on Bayesian Deep Learning, NIPS 2016, Barcelona, Spain.

3 A didactic example

To highlight the practical distinction between risk and uncertainty we begin with a maximally simplistic example. Fix $n \in \mathbb{N}$ and let $y_1, .., y_n \in \mathbb{R}$ be i.i.d. samples from an unknown distribution F. Given this data, we might be interested in estimating the mean $\mathbb{E}[Y]$, the variance $\mathrm{Var}(Y)$ and also to give some measure of our uncertainty over these quantities. Of course, this problem is horribly ill-defined: imagine if I gave you a dataset $\{y_1{=}7\}$ and asked you estimate the mean $\mathbb{E}[Y]$... we have no idea if the expectation is well-defined! To even attempt to answer these sorts of questions requires some form of prior information.

The subjective Bayesian framework provides a coherent solution to this problem. First the agent specifies a prior distribution for the parameters of interest, then it should update its beliefs according to Bayes rule based upon the data it observes [14, 6]. For clarity, in the example above we might model that the observations $y_i \sim N(\mu^*, \sigma^2)$ where $\mu^* \sim N(\mu_0, \sigma_0^2)$ is unknown implies

$$\mu^* | y_1, .., y_n \sim N\left(\mu_n = \frac{\mu_0/\sigma_0^2 + \sum_{i=1}^n y_i/\sigma^2}{1/\sigma_0^2 + n/\sigma^2}, \sigma_n^2 = \frac{1}{1/\sigma_0^2 + n/\sigma^2}\right). \tag{1}$$

The precise form of this posterior is relatively unimportant compared its qualitative characteristics. First, this distribution models the uncertainty in the unknown parameter μ^* by σ_n^2 as distinct from the risk σ^2. Second, this uncertainty is concentrates with observed data so that in the limit of infinite data it is resolved $\sigma_n^2 \to 0$. These high level properties are not specific to this example but are general characteristics of coherent Bayesian inference [6].

The Bayesian solution to this toy problem is disarmingly simple because the specific prior model admits conjugate updates [5]. With more complex models, such as deep neural networks, this will not be the case. Here, computational approximations must take the place of some analytical manipulations [3, 31, 1]. One of the most popular recent suggestions has been to use dropout sampling [29, 12] (where individual neurons are independently set to zero with probability p) to "get uncertainty information from these deep learning models for free – without changing a thing" [10]. Unfortunately, as we now show, dropout sampling can be better thought of as an approximation the risk in y, rather than the uncertainty of the learned model. Further, using a fixed dropout rate p, rather than optimizing this variational parameter can lead an arbitrarily bad approximation to the risk [15].

Consider a linear network $f = \sum_{k=1}^K d_k w_k$ with dropout $d_k \sim \mathrm{Ber}(p)$ and weights $w_k \in \mathbb{R}$. Minimizing the mean squared error on $\{y_1, .., y_n\}$ for $\overline{y}{:=}\sum_{i=1}^n y_i/n$ the resulting optimum $w_k = \overline{y}/K(1-p)$ produces a binomial distribution with mean $\overline{y}$ and variance $p\overline{y}/(1-p)K$. The resulting "dropout posterior" can have arbitrarily small or large variance depending on the interaction between the dropout rate p and the model size K. This distribution does not concentrate as more data is gathered and has no dependence on the amount of data n, nor the observed variance in the data. Optimizing the dropout rate [15] or switching to a heteroskedastic loss [11] can lead to a more accurate approximation of the *risk*, but does not address the fundamental flaws in this application for model *uncertainty* [11].

An alternative pragmatic approach to non-parametric computational uncertainty estimation is given by the bootstrap, a method for data-based simulation [7]. At a high level, the bootstrap samples multiple realizations of a given dataset perturbed by some noise, fits an estimator on each sampled dataset and then uses the resulting distribution to approximate uncertainty [8]. For certain choices of sampling/perturbations this process is precisely equivalent to Bayesian inference [25, 23]. Note that simply adding i.i.d. $N(0, \sigma^2)$ noise to the labels y_i is equivalent to limit of prior variance $\sigma_0^2 \to \infty$ in (1) [13].

4 Paper outline

We extend the analysis from Section 3 to linear functions and argue that this behavior also carries over to deep learning; extensive computational results support this claim. We investigate the importance of risk and uncertainty in sequential decision problems and why this setting is crucially distinct from standard supervised learning tasks. We highlight the dangers of a naive applications dropout (or any other approximate risk measure) as a proxy for uncertainty. We present analytical regret bounds for algorithms based upon smoothed bootstrapped uncertainty estimates that complement their strong performance in complex nonlinear domains [24, 22].

References

[1] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *ICML*, 2015.

[2] Apostolos N Burnetas and Michael N Katehakis. Optimal adaptive policies for Markov decision processes. *Mathematics of Operations Research*, 22(1):222–255, 1997.

[3] Siddhartha Chib and Edward Greenberg. Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335, 1995.

[4] Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-sensitive and robust decision-making: a cvar optimization approach. In *Advances in Neural Information Processing Systems*, pages 1522–1530, 2015.

[5] David Roxbee Cox and David Victor Hinkley. *Theoretical statistics*. CRC Press, 1979.

[6] Bruno De Finetti. La prévision: ses lois logiques, ses sources subjectives. In *Annales de l'institut Henri Poincaré*, volume 7, pages 1–68, 1937.

[7] Bradley Efron. *The jackknife, the bootstrap and other resampling plans*, volume 38. SIAM, 1982.

[8] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[9] Valerii Vadimovich Fedorov. *Theory of optimal experiments*. Elsevier, 1972.

[10] Yarin Gal. What my deep model doesn't know... *Personal blog post*, 2015.

[11] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[12] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

[13] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[14] John Maynard Keynes. *A treatise on probability*. Courier Corporation, 2013.

[15] Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *NIPS*, 2015.

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014.

[17] Frank H Knight. *Risk, uncertainty and profit*. Courier Corporation, 2012.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[19] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

[20] Volodymyr et al. Mnih. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[21] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[22] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *arXiv preprint arXiv:1602.04621*, 2016.

[23] Ian Osband and Benjamin Van Roy. Bootstrapped Thompson sampling and deep exploration. *NIPS*, 2016.

[24] Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. *ICML*, 2015.

[25] Donald B Rubin et al. The Bayesian bootstrap. *The annals of statistics*, 9(1):130–134, 1981.

[26] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

[27] Dan Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. In *Advances in Neural Information Processing Systems*, pages 1583–1591, 2014.

[28] David Silver et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[29] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[30] Abraham Wald. Statistical decision functions. In *Breakthroughs in Statistics*, pages 342–357. Springer, 1992.

[31] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

APPENDIX

In this appendix we present some early computational results that serve to highlight the practical implications of using dropout for uncertainty estimates. We compare six distinct methods for uncertainty estimation in a sequence of one-dimensional regression tasks.

1. **Naive ensemble** - 10 models trained from different random initialization.
2. **Smoothed ensemble** - 10 models each trained with target smoothing $\tilde{y}_i^k = y_i + w_k \sim N(0, 1/4)$.
3. **Bootstrap ensemble** - 10 models each trained on bootstrapped subsets of the data.
4. **SmoothedBoot ensemble** - 10 models each trained on bootstrapped subsets of the data with additional target smoothing $\tilde{y}_i^k = y_i + w_k \sim N(0, 1/4)$.
5. **Dropout $p = 0.5$** - a single model evaluated for 1000 monte-carlo dropout samples with $p = 0.5$.
6. **Dropout $p = 0.9$** - a single model evaluated for 1000 monte-carlo dropout samples with $p = 0.9$.

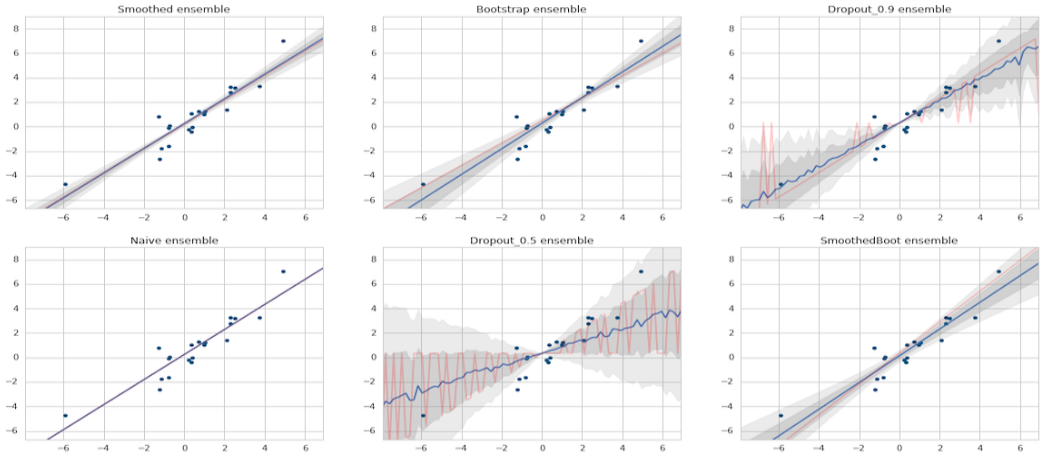

Figure 1: Linear regression with little data.

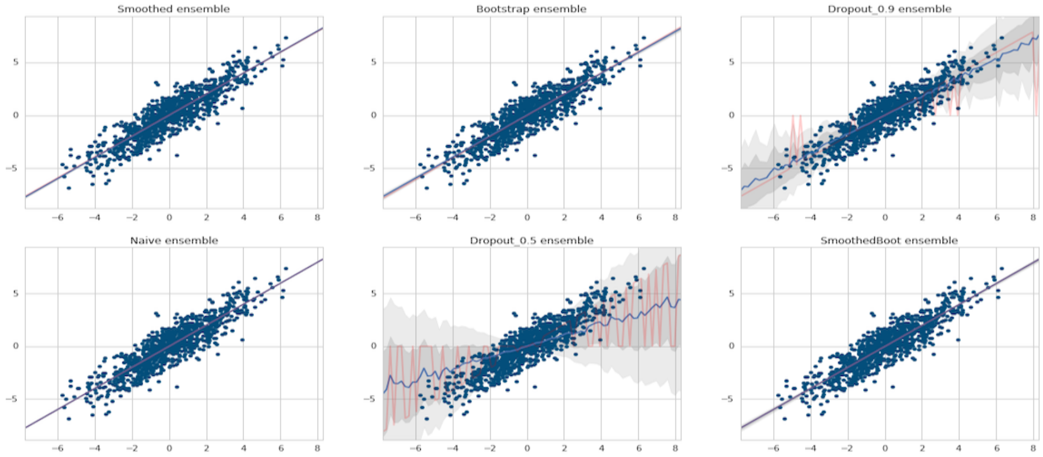

Figure 2: Linear regression with lots of data.

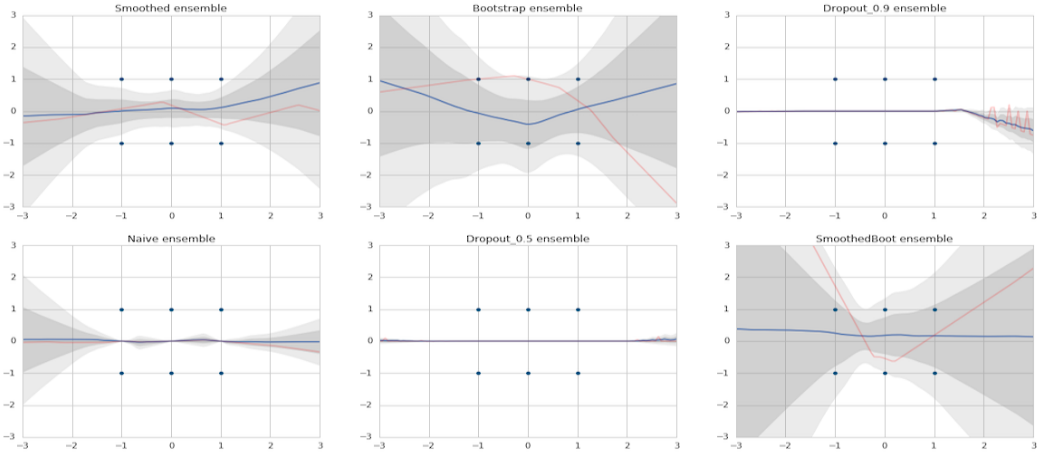

Figure 3: Nonlinear regression with (50, 50) MLP.

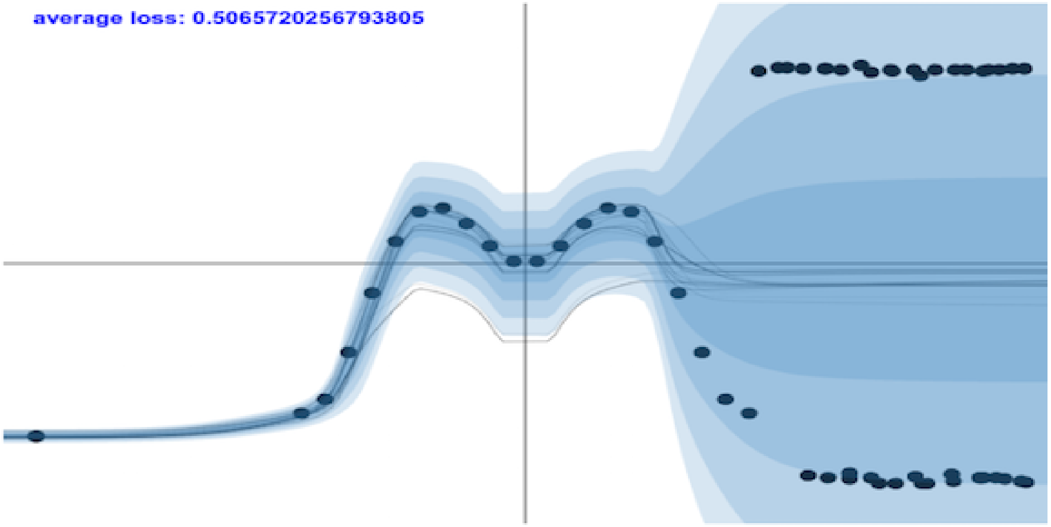
average loss: 0.5065720256793805

Figure 4: Dropout with heteroskedastic noise approximates the **risk**, but is a precisely the opposite of the **uncertainty** of the regression mean. Screenshot taken from `http://mlg.eng.cam.ac.uk/yarin/`