

Deep Probabilistic Programming for Financial Modeling

Matthew F. Dixon

Abstract

Deep probabilistic programming (DPP) combines three fields: Bayesian statistics and machine learning, deep learning, and probabilistic programming. Many applications in trading and risk management involve uncertainty quantification such as portfolio optimization and risk estimation. DPP provides a general purpose Bayesian framework for fitting high dimensional models to large datasets. Wielded correctly, it provides a richer set of statistical properties and more explanatory power over deterministic models such as deep learning. Furthermore, increasing low-cost compute power made available through many-core processors, can be readily exploited by open-source DPP packages such as **Edward**. This short positional article provides a brief and informal background on deep probabilistic programming and discusses its implications for trading, risk management and financial stability modeling.

1 Introduction

Deep probabilistic programming (DPP) combines three fields: Bayesian statistics and machine learning, deep learning (DL), and probabilistic programming. Although in its infancy, DPP is a powerful combination of a number of different probabilist modeling approaches and inference techniques that have historically been treated as separate mathematical ideas in the financial modeling literature. Modeling approaches includes Bayesian neural networks (BNNs, Bayesian NNs), directed graphical models and the broad availability of a wide range of techniques for efficient and robust model training. DPP can be regarded as a compositional modeling framework, composing previously disparate modeling concepts and algorithms over a scalable computational graph implementation. Just as **TensorFlow**'s (Abadi et al., 2016) computational graphs have made DL synonymous with modeling high dimensional input spaces with very large data sets, so to is **Edward** (Tran et al., 2017) poised to make DPP a tool of choice along side STAN (Carpenter et al., 2017) and **PyMC3** (Salvatier, Wiecki, & Fonnesbeck, 2016b) for Bayesian modeling and inference of computationally intractable problems.

1.1 Gaussian Processes

In order to explore some of the key capabilities of DPP for financial modeling and uncertainty quantification, we shall first revisit more familiar territory - Gaussian processes (GPs). The idea of GPs is to, without parameterizing $F(X)$, place a prior $\mathbb{P}[F(x)]$ directly on the space of functions (MacKay, 1998). Gaussian processes can hence be viewed as a generalization of Gaussian distributions from finite dimensional vector spaces to infinite dimensional function spaces. The main advantage of using GPs is modeling co-movements of time series. GPs learn the relationship between the time series directly building up an empirical model of the co-movements rather than assuming linear correlations between stochastic processes.

1.2 Bayesian Neural Networks

Gaussian processes have been shown to be a limiting case of Bayesian Neural networks (BNNs). These offer a probabilistic interpretation of neural network models by inferring distributions over the models' weights. To model uncertainty, BNNs assume that model parameters (weights and biases) are random variables. A prior distribution is placed over the weights and biases, which induces a distribution over a parametric set of functions. As the number of weights goes to infinity (Neal, 2012; Williams, 1997) show that the Bayesian

neural networks converge to GPs when a standard matrix of Gaussian prior distributions is placed over the weights and point estimates used for the priors. Of course, there is no reason to restrict modeling assumptions to these limiting conditions and, in fact, one of the main benefits of either approach is the flexibility to combine different distributions and include interaction effects.

2 Deep Learning

DL has shown remarkable success in a wide field of applications, including Artificial Intelligence (AI) (DeepMind, 2016; Kubota, 2017; Esteva et al., 2017), image processing (Simonyan & Zisserman, 2014), learning in games (DeepMind, 2017), neuroscience (Poggio, 2016), energy conservation (DeepMind, 2016), skin cancer diagnostics (Kubota, 2017; Esteva et al., 2017). Current advances in hardware (GPU/TPU) and software (Theano/TensorFlow) allow for developing scalable deep learners to find a **deterministic** approximation to an unknown nonlinear function. However, in many trading and risk management applications, it is necessary to model uncertainty of a predictive model. Deep learners are typically used to build deterministic approximations to an unknown nonlinear function and are largely unusable when uncertainty needs to be explicitly represented. And so deep probabilistic programming builds on deep learners by providing inferential techniques to estimate the posterior distribution and hence characterize the uncertainty in a model estimate.

2.1 Bayesian Computations

As a Bayesian approach, the choice of architecture of a deep network, combined with regularization, can be viewed as defining a prior probability distribution over non-linear functions and deep learning can be viewed as finding the posterior probability distribution of the unknown function over the network weights (Gal, 2016). In general, the posterior distribution is analytically intractable and Bayesian computational methods are required to approximate it. One common class of solution approaches, referred to as *variational inference*, solves an optimization problem by minimizing the Kullback-Leiber (KL) distance between an approximating variational distribution and the posterior obtained from the original model. Bayes’ filtering techniques, such as Kalman filters, HMMs and particle filters, are examples of variational inference techniques common place in trading and risk management, albeit for low dimensional input spaces. The other well known class of posterior approximation techniques use sampling techniques, rather than variational techniques - a prime example being the use of MCMC simulation. Neither of these inference engines, however, scale well with respect to model and data size.

Recent advances in variational inference techniques and software can now represent probabilistic models as a computational graph (Blundell, Cornebise, Kavukcuoglu, & Wierstra, 2015; Tran et al., 2017; Salvatier, Wiecki, & Fonnesbeck, 2016a). The key benefit here is the ability to build scalable probabilistic deep learners can, without having to perform testing (forward propagation) or inference (gradient-based optimization, with back propagation and automatic differentiation) from scratch. Alternatives to variational and MCMC algorithms were recently proposed by (Gal, 2016) and build on efficient Dropout regularization techniques, a variable selection approach which has fueled the popularity of DL.

The combination of **Edward** and **TensorFlow** is the first time that we have seen large-scale Bayesian modeling fused with AI and HPC and that’s likely to spawn critical developments in applications where uncertainty and model risk play a central role, such as finance. The remainder of this article briefly discusses some areas which are likely suitable financial modeling applications for DPP.

3 Financial Applications

Predictive Modeling Oftentimes, an automated decision to trade or a price forecast may be shrouded in uncertainty arising from noisy data or model risk, either through incorrect model assumptions or parameter error. This uncertainty should be accounted for in predictions so that the model output can be more clearly interpreted. DPP thrives in high dimensional input spaces. In trading and risk management applications, high dimensionality may arise naturally where there is spatial structure in the data. Examples include deep portfolios (Heaton, Polson, & Witte, 2017), large scale loan modeling (Sirignano, Tsoukalas, & Giesecke,

2016), predictors for prices using the many depth levels in the limit order book (Dixon, Polson, & Sokolov, 2017; Dixon, 2017; Sirignano, 2016), predictors arising from combining several data sets as in mortgage risk (Sirignano, Sadhwani, & Giesecke, 2016) or consumer lending (Dixon, Giesecke, Sheshardi, & Troha, 2017), or from text mining and natural language processing of news and other documents (Glasserman & Mamaysky, 2015).

Portfolio Optimization and Market Risk For some of these applications, we need look no further than existing applications of Gaussian processes to finance. (da Barrosa, Salles, & de Oliveira Ribeiro, 2016) present a spatio-temporal GP method for optimizing financial asset portfolios which allows for approximating the risk surface. (Cousin, Maatouk, & Rulliere, 2016) propose a new term-structure interpolation method that extends classical spline techniques by quantifying uncertainty. Other examples of GPs include spatial meta-modeling for Expected Shortfall through nested simulation (Liu & Staum, 2012). Spatio-temporal GPs are used to infer portfolio values in a scenario based on inner-level simulation of nearby scenarios. This significantly reduces the required computational effort by avoiding inner-level simulation in every scenario and takes account of the variance that arises from inner-level simulation. DPP could improve these existing applications through Bayesian neural networks and robust regularization techniques for generalizing modeling and improving robustness and scalability.

Contagion and Credit Risk Capturing dependencies between financial institutions is a significant aspect of contagion and counterparty credit risk modeling. Examples in contagion modeling include the inter-dependency between banks participating in Federal Reserve System’s emergency programs (Battiston, Puliga, Kaushik, Tasca, & Caldarelli, 2012), joint dependencies of banks and other institutional investors through transactions in the repurchase agreement market or through various other channels of systemic risk (Bisias, Flood, Lo, & Valavanis, 2012). Other examples in counterparty credit risk modeling include wrong way risk assessment of OTC portfolios. It is well known that the dependence structure between variables can be explicitly model with a directed graph. Given this capability, a promising research direction is how to use probabilistic graphical programming to build Bayesian network copulas (Elidan, 2010) for contagion and credit risk?

3.1 Summary

Deep probabilistic programming (DPP) combines three fields: Bayesian statistics and machine learning, deep learning, and probabilistic programming. Many applications in trading and risk management involve uncertainty quantification such as portfolio optimization and risk estimation. DPP provides a general purpose Bayesian framework for fitting high dimensional models to large datasets to provide a richer set of statistical properties and more explanatory power.

This article provides some background on deep probabilistic programming and discusses its implications for trading and risk management. In particular, we see merit in exploring high dimensional modeling problems arising in large-scale portfolios, systemic risk and contagion in financial networks and where spatial structure is present in the application, such as limit order book modeling. Tools such as **Edward** lower the barrier to entry and add salt to the fire ignited by **TensorFlow**. The barrier to entry is deceptively simple however and some discipline is needed. The programming convenience of **Keras** for **TensorFlow** is alluring, but the ability to build effective and interpretable financial models is still in its infancy and should be investigated more fully by cross-disciplinary researchers in computational science, statistics and finance, before any definitive and practical conclusions can be drawn.

4 Acknowledgements

The author would like to thank Keiran Thompson, Stanford University, and Saeed Amen, Cuemacro, for their useful feedback.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). TensorFlow: A System for Large-scale Machine Learning. In *Proceedings of the 12th usenix conference on operating systems design and implementation* (pp. 265–283).
- Battiston, S., Puliga, M., Kaushik, R., Tasca, P., & Caldarelli, G. (2012, August 2). DebtRank: Too central to fail? financial networks, the FED and systemic risk. *Scientific Reports*, 2. Retrieved from <http://dx.doi.org/10.1038/srep00541> doi: 10.1038/srep00541
- Bisias, D., Flood, M., Lo, A., & Valavanis, S. (2012). A survey of systemic risk analytics. *Annual Review of Financial Economics*, 4(1), 255–296. Retrieved from <http://EconPapers.repec.org/RePEc:anr:refeco:v:4:y:2012:p:255-296>
- Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015, May). Weight Uncertainty in Neural Networks. *arXiv:1505.05424 [cs, stat]*.
- Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., ... Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1).
- Cousin, A., Maatouk, H., & Rulliere, D. (2016). Kriging of financial term-structures. *European Journal of Operational Research*, 255(2), 631–648. Retrieved from <http://EconPapers.repec.org/RePEc:eee:ejores:v:255:y:2016:i:2:p:631-648>
- da Barrosa, M. R., Salles, A. V., & de Oliveira Ribeiro, C. (2016). Portfolio optimization through kriging methods. *Applied Economics*, 48(50), 4894–4905. doi: 10.1080/00036846.2016.1167827
- DeepMind. (2016). *DeepMind AI Reduces Google Data Centre Cooling Bill by 40%*. <https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/>.
- DeepMind. (2017). *The story of AlphaGo so far*. <https://deepmind.com/research/alphago/>.
- Dixon, M. (2017). Sequence classification of the limit order book using recurrent neural networks. *arXiv preprint arXiv:1707.05642*.
- Dixon, M., Giesecke, K., Sheshardi, A., & Troha, N. (2017). Optimal investment in online loans. *working paper*.
- Dixon, M., Polson, N., & Sokolov, V. (2017). Deep learning for spatio-temporal modeling: Dynamic traffic flows and high frequency trading. *arXiv preprint arXiv:1705.09851*.
- Elidan, G. (2010). Copula bayesian networks. In *Advances in neural information processing systems* (pp. 559–567).
- Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115–118.
- Gal, Y. (2016). *Uncertainty in deep learning* (Unpublished doctoral dissertation). PhD thesis, University of Cambridge.
- Glasserman, P., & Mamaysky, H. (2015). Does unusual news forecast market stress? *Columbia Business School Research Paper*(15-70).
- Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1), 3–12.
- Kubota, T. (2017, January). *Artificial intelligence used to identify skin cancer*. <http://news.stanford.edu/2017/01/25/artificial-intelligence-used-identify-skin-cancer/>.
- Liu, M., & Staum, J. (2012). Stochastic kriging for efficient nested simulation of expected shortfall. *Journal of Risk*.
- MacKay, D. J. (1998). Introduction to gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, 168, 133–166.
- Neal, R. M. (2012). *Bayesian learning for neural networks* (Vol. 118). Springer Science & Business Media. (bibtex: neal2012bayesian)
- Poggio, T. (2016). Deep Learning: Mathematics and Neuroscience. *A Sponsored Supplement to Science, Brain-Inspired intelligent robotics: The intersection of robotics and neuroscience*, 9–12.
- Salvatier, J., Wiecki, T. V., & Fonnesbeck, C. (2016a). Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2, e55.
- Salvatier, J., Wiecki, T. V., & Fonnesbeck, C. (2016b, April). Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2, e55. Retrieved from <https://doi.org/10.7717/peerj-cs.55> doi: 10.7717/peerj-cs.55

- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.
- Sirignano, J., Sadhwani, A., & Giesecke, K. (2016, July). Deep Learning for Mortgage Risk. *ArXiv e-prints*.
- Sirignano, J. A. (2016). Deep learning for limit order books. *arXiv preprint arXiv:1601.01987*.
- Sirignano, J. A., Tsoukalas, G., & Giesecke, K. (2016). Large-scale loan portfolio selection. *Operations Research*, 64(6), 1239-1255. doi: 10.1287/opre.2016.1537
- Tran, D., Hoffman, M. D., Sauros, R. A., Brevdo, E., Murphy, K., & Blei, D. M. (2017, January). Deep Probabilistic Programming. *arXiv:1701.03757 [cs, stat]*.
- Williams, C. K. (1997). Computing with infinite networks. In *Advances in neural information processing systems* (pp. 295–301).