Statistical Perspectives in Teaching Deep Learning from Fundamentals to Applications

David Han, Ph.D., Nathan Kim
The University of Texas at San Antonio, 1 UTSA Circle, San Antonio, TX 78249

Abstract
The use of Artificial Intelligence (A.I.), Machine Learning (ML) and Deep Learning (DL) have gained a lot of attention and become increasingly popular in many areas of application. Historically machine learning and theory had strong connections to statistics; however, the current deep learning context is mostly in computer science perspectives and lacks statistical perspectives. In this work, we address this research gap and discuss how to teach deep learning to the next generation of statisticians. We first describe some backgrounds and how to get motivated. We discuss different terminologies in computer science and statistics, and how deep learning procedures work without getting into mathematics. In response to a question regarding what to teach, we address organizing deep learning contents and focus on the statistician's view; form basic statistical understandings of the neural networks to the latest hot topics on uncertainty quantifications for prediction of deep learning, which has been studied in the Bayesian frameworks. Furthermore, we discuss how to choose computational environments and help develop programming skills for the students. We also discuss how to develop homework incorporating the idea of experimental design. Finally, we discuss how to expose students to the domain knowledge and help to build multi-discipline collaborations.

Key Words: artificial intelligence, curriculum development, deep learning, machine learning, artificial neural network, statistical education

1. Practical Challenges

The use of Artificial Intelligence (A.I.), Machine Learning (ML) and Deep Learning (DL) have gained a lot of attention and become increasingly popular in many areas of application. Historically machine learning and theory had strong connections to statistics; however, the current deep learning context is mostly in computer science perspectives and lacks statistical perspectives. To address this gap, academic statisticians must be prepared to develop practice-oriented curricula in teaching deep learning to the next generation of statisticians, rather than traditional proof-based curricula. This is because the students' appetite and drive are to do DL, rather than to prove theorems about DL. Also, the textbooks should be selected with a variety of choices and the focus on statistical learning, other than the popular *Deep Learning* by Goodfellow et al. (2016). It is also paramount for the instructors to have good software management due to the existence of diverse programming environments and abundant open-source tools available in the market, including Google TensorFlow, Keras, PyTorch, Theano, Caffe, Apache MxNet, MS CNTK, DL4J, Chainer, etc. Adopting different software tools, the instructors must consider the learning curves in mastering various tools, software stability

and developer/community support, tool ecosystem already established in public. Another important aspect to consider is the potential cognitive burden to statistics faculty and students all together due to the inconsistent terminology and jargon used between statistics and computer science; for example, the definition of bias term, noise robustness, normalization, hyperparameter, etc.

2. Course Contents

The course contents should cover from the fundamentals to good practices. These include

- the basic concept of perceptron, (deep) feed-forward neural network, back-propagation, non-convex optimization, empirical risk minimization;
- regularization and dropout, feature selection and dimension reduction, interpretable deep networks;
- convolutional neural network (CNN), recurrent neural network (RNN) and long short-term memory (LSTM);
- variational auto-encoders (VAE), generative adversarial network (GAN);
- representation learning, reinforcement learning;
- uncertainty quantification (UQ) via deep Gaussian process (DGP) and Bayesian variational inference, etc.

Please see Figure 1 below for the architectural references. The course should also cover other ML techniques such as support vector machines (SVM), random forests, boosting and bagging, natural language processing (NLP), etc. The instructors must discuss how all these techniques can be viewed as statistical techniques by connecting them to the fundamental statistical theory and/or methods. Just like any good statistical practices, the instructors also need to address the respective model assumptions and potential drawbacks so that the students do not misuse or overuse the learned techniques. The common pitfalls of DL include the tendency of overfitting and poorly approximated UQ, not to mention the sub-optimal network architectures due to the trial and error approach or construction based on abstract.

3. Pedagogical Approach

The DL curriculum for statisticians should be dynamical, adjustable, and adaptable. The instructors should prepare each lecture in an interactive and dynamic platform (*e.g.*, Jupyter Notebooks). Any popular MOOC can be also utilized in a complementary manner to support the students' learning (*i.e.*, bootcamp, preparatory and supplementary works). The pedagogical development must also consider the intended audience level of the course: for beginners or for advanced? and for which disciplines? For instance, even with the same course contents, the courses targeting (senior) undergraduates should be delivered in a different manner to the courses serving the graduate (M.S./Ph.D.) students.

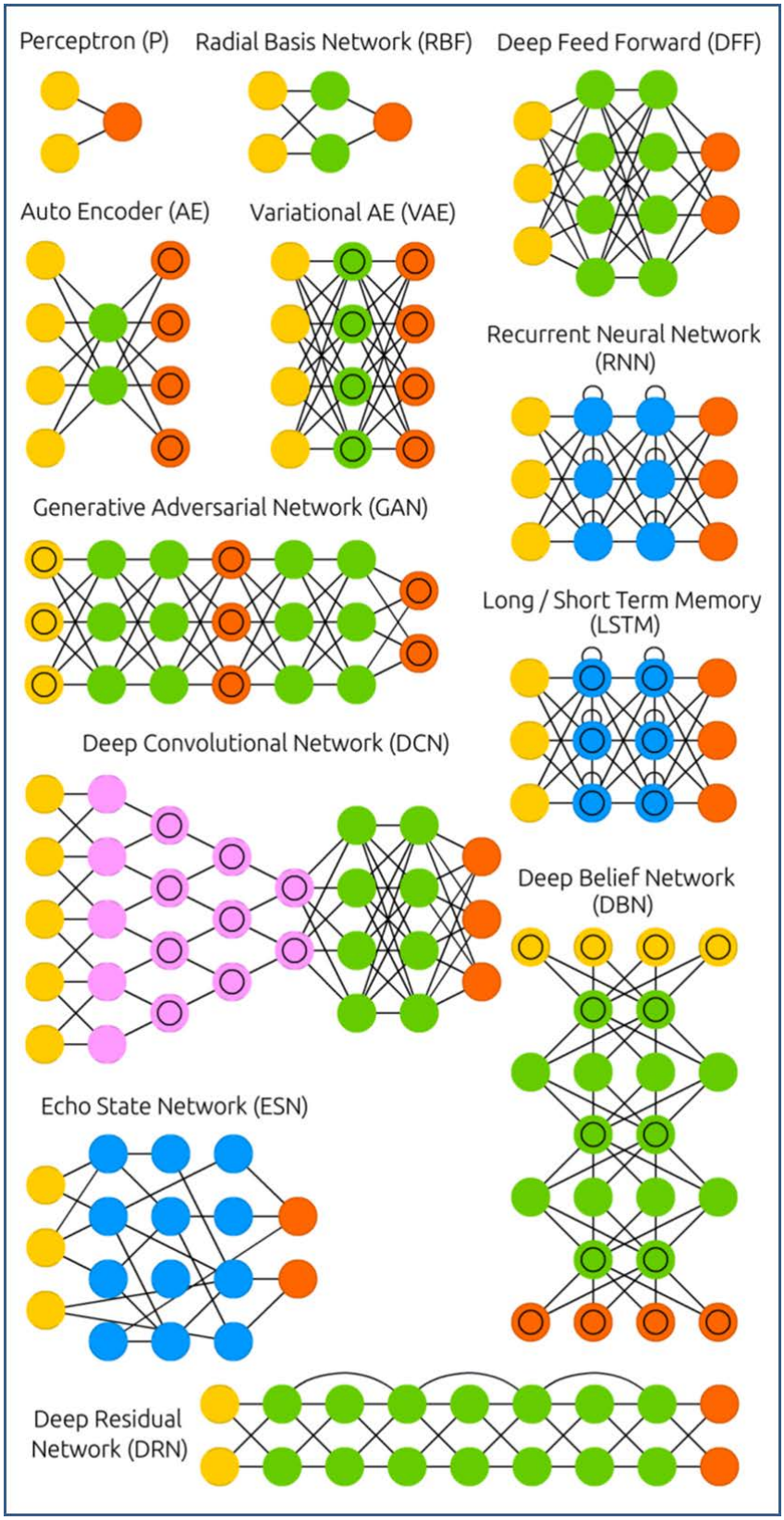

Figure 1: Various neural network designs a good DL course should discuss about (*source*: The Asimov Institute)

4. Programming Skill

Before enrolling to the course, it should be made very clear to the students that they are expected to have good programming skills to practice DL. This is indeed a significant prerequisite for the students, and such skills should be enhanced over the course. Currently, Python is the main programming language to practice DL, not R. However, both the instructors and students must be able to adapt as software tools can/will change constantly. To practice DL in a classroom environment, the course instructors should secure adequate computational resources and infra (*e.g.*, HPC servers, cloud computing) as most algorithms are highly computation-intensive.

5. Homework & Projects

When developing the homework and assignments, it needs to balance between theory and practice. Any exercise must be prescriptive and application-oriented if possible. One could start with some classical datasets to help the students gain familiarity, such as MNIST, Cifar-10, SVHN, etc. Large open datasets such as from Kaggle are also recommended for implementation and practices with Big Data. Another important pedagogical aspect to consider is to expose the students to the domain knowledge specific to each given problem from various fields. Introducing an adequate level of the domain knowledge will help building cross-/inter-/multi-disciplinary collaborations with the domain experts. This will also encourage and help incorporating the team-based learning, which is a common practice in industry.

References

Geron, A. (2017). *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. Sebastopol, CA: O'Reilly Media, Inc.

Goodfellow, I., Bengio, Y., and Courville, Y. (2016). *Deep Learning*. Cambridge, UK: MIT Press.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement Learning*. Cambridge, UK: MIT Press.

van Veen, F. and Leijnen, S. (2019). A mostly complete chart of neural networks (asimovinstitute.org)