Applied Stochastic Models in Business and Industry

Discussion

(wileyonlinelibrary.com) DOI: 10.1002/asmb.2228 Published online in Wiley Online Library

Discussion of 'Deep learning for finance: deep portfolios'

The goal of this discussion is to present the connection between deep learning models and more familiar nonparametric statistical models. We also demonstrate connection between deep learning auto-encoders and a more classical way to find low-dimensional structures in the data, namely, singular value decomposition.

1. Nonparametric model perspective

Authors provided a rigorous and a clear introduction into the deep learning models. They described the problem of model estimation (back propagation), model quality assessment, and the issue of trade-off between bias and variance of the model that can be addressed via regularization or drop-out techniques. Because deep learning models are currently not widely used in econometrics or engineering applications, we present yet another point of view on those models that hopefully will make the content of the paper more accessible and less mysterious for some of the readers. The neural networks and deep learning models (networks with large number of links in the path from inputs to outputs), in particular have been widely used in the last three decades for the model-free regression and classification problems. Deep learning models are also called black-box models [1], in the sense that no domain insights (in our case financial principals) are used to build a model. A very flexible model structure is used to compensate for the lack of any assumption. It was shown in other applications that flexible black-box models can outperform typically 'nice' approximations that dominated the application before. For example, deep learning methods outperform other parametric and nonparametric statistical methods on such tasks as speech recognition [2] and image processing [3,4]. The main difficulty with the flexible of nonlinear approximations is that a very large set of possible models needs to be handled, in the context of deep learning that manifested in the network architecture. Although, for some applications, there are efficient methods for finding an optimal network architecture [5].

Thus, the main advantage of deep learning models is that they do not make any assumption on a functional form of the map $F : R^p \to R^r$ to be learned. This trait of deep learners echoes a well-studied in statistical literature class of nonparametric models [6]. This is not a coincidence. We will show that in fact those approaches are very similar. Understanding this similarity helps us gaining some insights into deep learning methodology. This, hopefully, will make understanding of deep learning models easier for the readers unfamiliar with the technique.

A statistical learning problem considers recovering the true regression function $F(X)$ given N observations

$$Y_i = F(X_i) + e_i,\ i = 1, \dots, N.$$

The nonparametric approach seeks to approximate the unknown map F using a family of functions defined by the following expression

$$F(X) = \sum_{k=0}^{\infty} \alpha_k f_k(X). \tag{1}$$

Functions f_k are called basis functions and play similar role of a functional space basis, that is, they are chosen to give a good approximation to the unknown map F. In some cases, $\{f_k\}_{k=1}^{\infty}$ actually do form a basis of a space, for example, Fourier ($f_k(x) = \cos(kx)$) and wavelet bases.

In a multivariate case ($p > 1$), the basis functions are usually constructed using functions of a single variable. Two examples are radial functions and ridge functions. The radial function has the following form

$$f_k(X) = \kappa\left(||X - \gamma_k||_2\right), \tag{2}$$

Applied Stochastic Models in Business and Industry

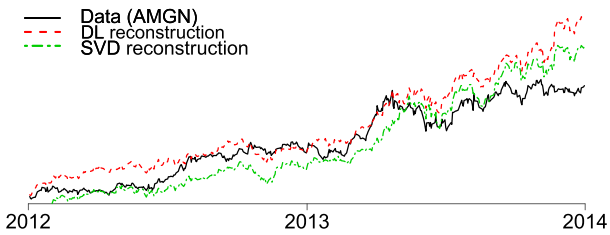


Figure 1. AMGN opening price data reconstructed from deep learning and SVD projections. [Colour figure can be viewed at wileyonlinelibrary.com]

where κ typically chosen to be a bell-shaped function, such as $1/e^{x^2}$ or $1/\cosh(x)$. A ridge function relies on inner product and is given by

$$f_k(X) = \kappa(w^T X + w_0). \tag{3}$$

The ridge function, which is a composition of inner-product and nonlinear univariate function, is arguably one of the simplest nonlinear multivariate function. Two of the most popular types of neural networks are constructed as a composition of radial or ridge functions. The famous radial basis neural networks [7] are nothing but an approximation using radial function (2). Each layer of a feed-forward neural network used in the paper (for auto-encoding, calibration, and validation steps) can be seen as the representation (1) with ridge basis function (3) and κ being sigmoidal (e.g., $1/(1+e(-x))$, $\cosh(x)$, or $\tanh(x)$), heaviside gate functions (e.g., $I(x > 0)$), or rectified linear units $\max\{x, 0\}$. The function κ is called an activation function in the context of deep learning; thus, a deep learning network can be seen as a superposition of approximations (1).

Some of the well-known nonparametric techniques can be represented using (1) as well. The *kernel estimator* can be obtained by choosing

$$f_k(X) = \kappa\left(\frac{X - \gamma_k}{h}\right).$$

Another example is *Fourier series* that is used for time series analysis. It can obtained by choosing $f_k(x) = \cos(x)$. A *spline* approximation can be derived by using polinomial functions with finite support as a basis. Popular nonparametric *tree-based models* [8], can be represented as (1), by choosing

$$f_k(X) = \bar{Y}_k I(X \in C_k).$$

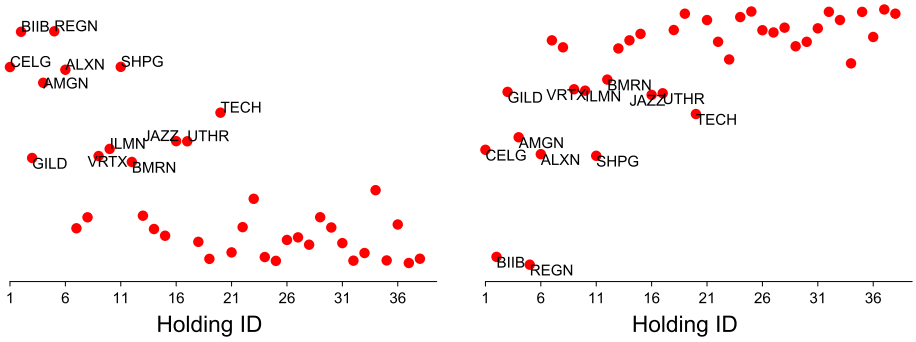
In regression, $\bar{Y}_k$ is the average of $(Y_i \mid X_i \in C_k)$ and C_k is a box set in R^p with zero or more extreme directions (open sides).

Ridge-based approximations, which are used in the paper, can efficiently represent high-dimensional data sets with a small number of parameters. We can think of deep features (outputs of hidden layers) as projections of the input data into a lower-dimensional space. Deep learners can deal with the curse of dimensionality because ridge functions determine directions in (ϕ^{k-1}, ϕ^k) space, where the variance is very high. Here, ϕ^k is the input to the kth layer of a deep learning network. Those directions are chosen as global ones and represent the most significant patterns in the data. This approach has a clear connection to the other well-studied techniques such as projection pursuit [9] and principal component analysis (PCA). In the next section, we show, in fact, that in the case of IBB index (iShares Nasdaq Biotechnology ETF (Exchange-Traded Fund)) analysis, both deep learning and PCA identified the same set of holdings that deviate from communal information.

2. Connection with principal component analysis

The goal of both PCA and deep learning is to find projections of the data onto lower-dimensional spaces. In the PCA, the projection space is linear (vector space), and a set of orthogonal vectors is used as a basis. The orthogonality of basis is equivalent to the assumption that projected features are not correlated. A deep learner projects data several times (at each layer of the network) and does not assume that projected features are uncorrelated.

We compare the data reconstructed from the PCA projections, calculated using singular value decomposition (SVD) [10] of the data matrix and the projection generated by stacked auto-encoder deep learning model. Figure 1 compares the opening price of AMGN holding (Amgen, Inc. stock) data with the reconstructions obtained from SVD and stacked auto-encoder deep learning network. In both cases, we used projection into a one-dimensional space. For the auto-encoding network, we used a bottleneck structure with five hidden layers, each having 100, 50, 1, 50, and 100 neurons correspondingly.



(a) Deep learning auto-encoder (b) SVD

Figure 2. One-dimensional projection performed via deep learning auto-encoder network and SVD. [Colour figure can be viewed at wileyonlinelibrary.com]

We can see that reconstructed data is very similar to the original data. Further, Figure 2 compares the projections obtained via the deep-learning auto-encoder network and SVD. The deep learning projection is the output of the 'bottleneck' layer that has one neuron. The SVD projection is the projection onto the first principal component. To make presentation clear, we plotted only a selected set of holdings from the IBB index.

Both methods identified the same communal and non-communal sets of holdings. The separation in the SVD case is not as strong as in the deep learning case.

3. Conclusion

We demonstrated a connection of the deep learning models with the nonparametric statistical methods and deep learning auto-encoders with more familiar PCA. One of the major criticism of the deep learning models is the lack of interoperability, which is especially important in econometric analysis. However, deep learning models are no more and no less interpretable than many nonparametric statistical models. Thus, there is no difference in interoperability between those two classes of models. The computational cost associated with finding an optimal network structure might prevent using the deep learners in certain applications. On another hand, deep learning provides much higher level of flexibility in terms of which patterns can be modeled in the data.

VADIM SOKOLOV
Department of Systems Engineering and Operations Research
George Mason University, USA
E-mail: *vsokolov@gmu.edu*

References

1. Sjöberg J, Zhang Q, Ljung L, Benveniste A, Delyon B, Glorennec P-Y, Hjalmarsson H, Juditsky A. Nonlinear black-box modeling in system identification: a unified overview. *Automatica* 1995; **31**(12):1691–1724.
2. Deng L, Li J, Huang J-T, Yao K, Yu D, Seide F, Seltzer M, Zweig G, He X, Williams J, Gong Y, Acero A, Seltzer M. Recent advances in deep learning for speech research at Microsoft. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE: Vancouver, Canada, 2013,8604–8608.
3. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015; **521**(7553):436–444.
4. Bengio Y. Learning deep architectures for AI. *Foundations and trends in Machine Learning* 2009; **2**(1):1–127.
5. Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. *Neural Computation* 2006; **18**(7):1527–1554.
6. Gibbons JD, Chakraborti S. *Nonparametric Statistical Inference*. Chapman and Hall/CRC, 2011.
7. Chen S, Cowan CFN, Grant PM. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks* 1991; **2**(2):302–309.
8. Breiman L, Friedman J, Stone CJ, Olshen RA. *Classification and Regression Trees*. CRC Press, 1984.
9. Friedman JH, Tukey JW. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers* 1974; **C-23**(9):881–890.
10. Golub GH, Van Loan CF. *Matrix Computations*, vol. 3. JHU Press, 2012.