

Presentation Overview


- ▶ Review of equity fundamental factor models
- ▶ Deep fundamental factor models
 - ▶ Problem formulation
 - ▶ Interpretability of factors with Russell 1000 portfolio example
- ▶ Interaction effects with same example

Linear Cross-sectional Factor Models

- ▶ Consider the fundamental equity factor model (Barra style²):

$$\mathbf{r}_t = B_t \mathbf{f}_t + \mathbf{e}_t, \quad t = 1, \dots, T$$

- ▶ $B_t = [\mathbf{1} \mid b_{1,t} \mid \dots \mid b_{K,t}]$ is the $N \times K + 1$ matrix of known fundamental factor exposures
- ▶ $(b_{i,t})_k$ is the exposure (a.k.a. loading) of asset i to factor k at time t
- ▶ $\mathbf{f}_t = [\alpha, f_{1,t}, \dots, f_{K,t}]$ is the $K + 1$ vector of unknown factor realizations
- ▶ \mathbf{r}_t is the N vector of asset returns
- ▶ $\rho(\mathbf{f}_t, \mathbf{e}_t) = 0$ and cross-sectional homoskedasticity $\mathbb{E}[\mathbf{e}_{t,i}^2] = \sigma_i^2$.

²See Grinold and Kahn (2000), Conner et al. (2010), and Cariño et al. (2010) 

Non-linear/Non-parametric Factor Models

- ▶ Consider the non-linear fundamental factor model:

$$\mathbf{r}_t = F(B_t) + \mathbf{e}_t$$

- ▶ $F : \mathbb{R}^K \rightarrow \mathbb{R}$ is a (differentiable) non-linear function
- ▶ Do not assume that e_t is Gaussian or require any other restrictions on \mathbf{e}_t , i.e. error distribution is non-parametric
- ▶ The model shall just be used to predict the next period asset returns only and stationarity of the factor returns is not required
- ▶ This model maps on to popular deep learning based predictive models.

Taxonomy of Most Popular Deep Learning Architectures

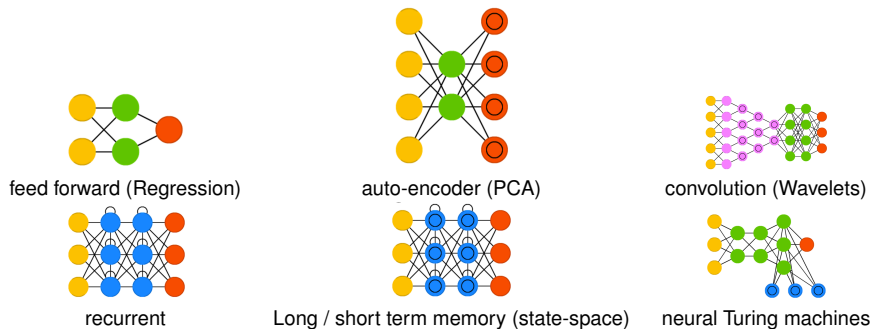


Figure: Most commonly used deep learning architectures for modeling. Source: <http://www.asimovinstitute.org/neural-network-zoo>

Quick Quiz

Which of the following statements are true?

- A Machine learning is different from statistics: Machine learning methods assume the data generation process is unknown
- B Machine learning uses the bias-variance tradeoff to avoid over-fitting
- C We need multiple hidden layers in the neural network to capture non-linearity
- D Neural networks provide no statistical interpretability, they are 'black-boxes' and the importance of the features is unknown

Quick Quiz

Which of the following statements are true?

- A Machine learning is different from statistics: Machine learning methods assume the data generation process is unknown [True]
- B Machine learning uses regularization and other techniques to tradeoff bias and variance, with the emphasis on out-of-sample performance [True]
- C We need multiple hidden layers in the neural network to capture non-linearity [False³]
- D Neural networks provide no statistical interpretability, they are 'black-boxes' and the importance of the features is unknown [False]

³The Universal representation theorem suggests that we only need one hidden layer: Andrei Nikolaevich Kolmogorov, On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. AMS Translation, 28(2):55-59, 1963.

Quick Overview of Supervised Machine Learning

Property	Statistical Inference	Supervised Machine Learning
Goal	Causal models with explanatory power	Prediction performance, often with limited explanatory power
Data	The data is generated by a model	The data generation process is unknown
Framework	Probabilistic	Algorithmic & Probabilistic
Expressability	Typically linear	Non-linear
Model selection	Based on information criteria	Numerical optimization
Scalability	Limited to lower dimensional data	Scales to higher dimensional input data
Robustness	Prone to over-fitting	Designed for out-of-sample performance

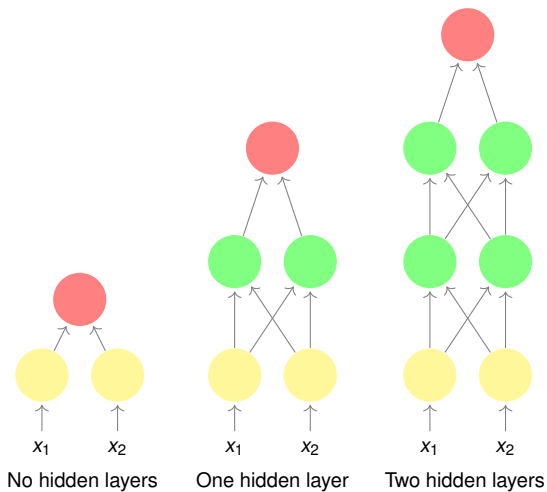
Supervised Machine Learning

- ▶ Machine learning addresses a fundamental prediction problem: Construct a nonlinear predictor, $\hat{Y}(X)$, of an output, Y , given a high dimensional input matrix $X = (X_1, \dots, X_P)$ of P variables.
- ▶ Machine learning can be simply viewed as the study and construction of an input-output map of the form

$$Y = F(X) \quad \text{where} \quad X = (X_1, \dots, X_P).$$

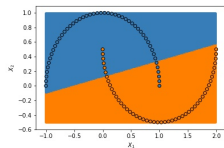
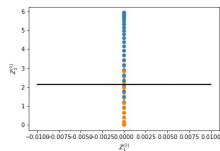
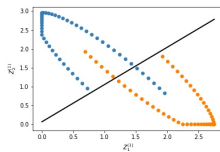
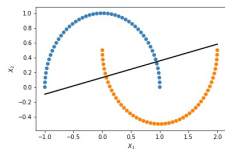
- ▶ The output variable, Y , can be continuous, discrete or mixed.

Geometric Interpretation of Neural Networks

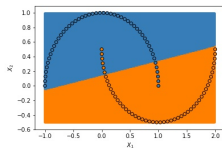


Geometric Interpretation of Neural Networks

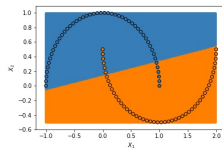
Half-Moon Dataset



No hidden layers

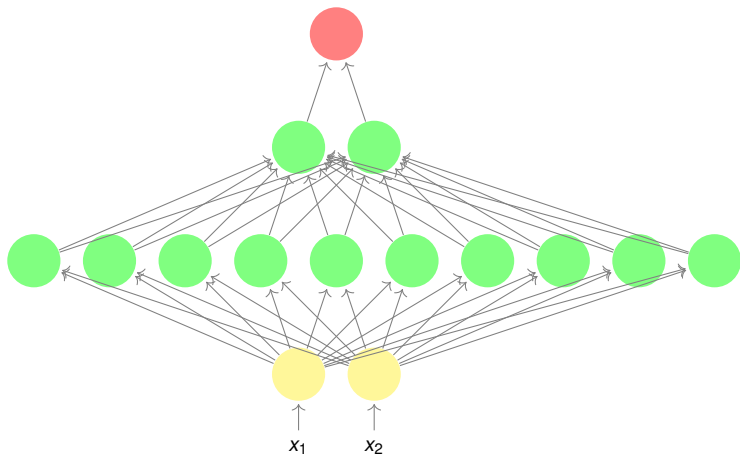


One hidden layer



Two hidden layers

More Neurons?



Geometric Interpretation of Neural Networks

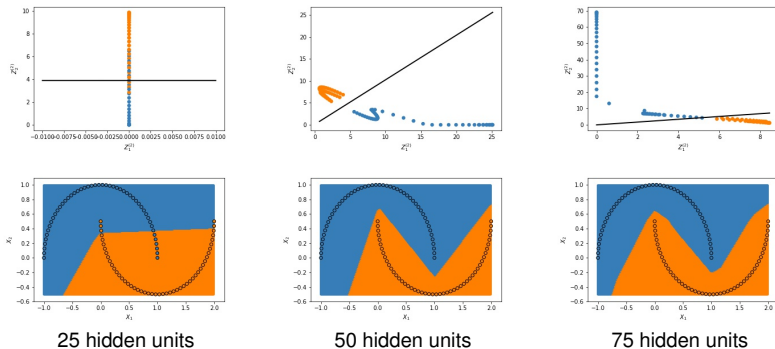


Figure: The number of hidden units is adjusted according to the requirements of the classification problem and can be very high for data sets which are difficult to separate.

Feedforward Networks

- ▶ The activation functions are essential for the network to approximate non-linear functions. For example if there is one hidden layer and $f^{(1)}$ is the identity function then

$$\hat{Y}(X) = W^{(2)}(W^{(1)}X + b^{(1)}) + b^{(2)} = W^{(2)}W^{(1)}X + W^{(2)}b^{(1)} + b^{(2)} = W'X + b'$$


is just linear regression, i.e. an affine transformation⁴

- ▶ Clearly, if there are no hidden layers, the architecture recovers standard linear regression

$$Y = WX + b.$$

(or logistic regression $\sigma(WX + b)$)

- ▶ => **Key insight for developing interpretability**

⁴While the functional form of the map is the same as linear regression, neural networks do not assume a data generation process and hence inference is not identical to ordinary least squares. 

Explanatory Power of Deep Networks

- ▶ In a linear regression model

$$\hat{Y} = F_{\beta}(X) := \beta_0 + \beta_1 X_1 + \dots + \beta_K X_K$$

the sensitivities are

$$\partial_{X_i} \hat{Y} = \beta_i$$

- ▶ In a FFWD neural network, we can use the chain rule to obtain the sensitivities

$$\partial_{X_i} \hat{Y} = \partial_{X_i} F_{W,b}(X) = \partial_{X_i} f_{W^{(L)},b^{(L)}}^{(L)} \circ \dots \circ f_{W^{(1)},b^{(1)}}^{(1)}(X)$$

- ▶ => **Key Idea: Resolve the interpretability at the sub-graph level rather than down to individual edges**

Example: Step test

- ▶ The model is trained to the following data generation process where the coefficients of the features are stepped:

$$\hat{Y} = \sum_{i=1}^{10} iX_i, \quad X_i \sim \mathcal{U}(0, 1).$$

Example: Step test

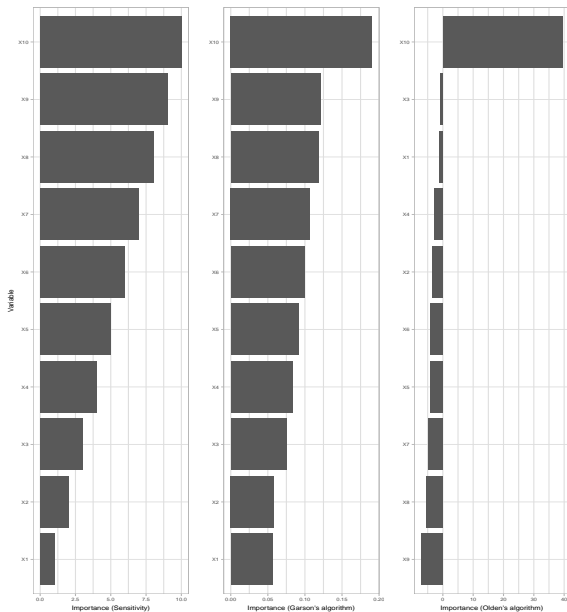


Figure: Step test: This figure shows the ranked importance of the input variables in a fitted neural network with one hidden layer.

Robustness of Interpretability

- ▶ If the data is generated from a linear model, can the neural network recover the correct weights?

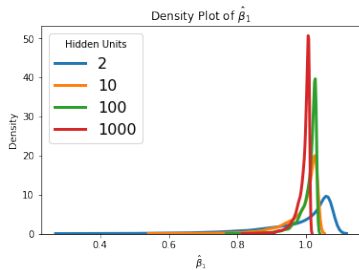
$$Y = \beta_1 X_1 + \beta_2 X_2 + e, \quad X_1, X_2, e \sim N(0, 1), \quad \beta_1 = 1, \beta_2 = 1$$

- ▶ Compare OLS estimators with zero hidden layer NNs and single hidden layers NNs.
- ▶ Use *tanh* activation functions for smoothness of the Jacobian because $\max(x, 0)$ gives a piecewise constant Jacobian.
- ▶ Increase the number of hidden units and show that the variance of the sensitivities converges.

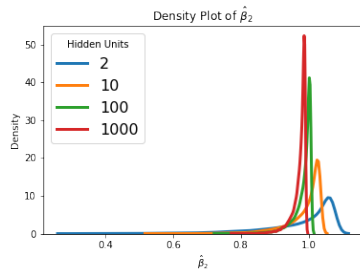
Robustness of Interpretability

Method	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$
OLS	0	1.0154	1.018
NN (zero hidden layers)	0.03	1.0184141	1.02141815
NN (1 hidden layers)	0.02	1.013887	1.02224

Robustness of Interpretability



(a) density of $\hat{\beta}_1$



(b) density of $\hat{\beta}_2$

Robustness of Interpretability ($\hat{\beta}_1$)

Hidden Units	Mean	Median	Std.dev	1% C.I.	99% C.I.
2	0.980875	1.0232913	0.10898393	0.58121675	1.0729908
10	0.9866159	1.0083131	0.056483902	0.76814914	1.0322522
50	0.99183553	1.0029879	0.03123002	0.8698967	1.0182846
100	1.0071343	1.0175397	0.028034585	0.89689034	1.0296803
200	1.0152218	1.0249312	0.026156902	0.9119074	1.0363332

The confidence interval narrows with increasing number of hidden units.

Robustness of Interpretability ($\hat{\beta}_2$)

Hidden Units	Mean	Median	Std.dev	1% C.I.	99% C.I.
2	0.98129386	1.0233982	0.10931312	0.5787732	1.073728
10	0.9876832	1.0091512	0.057096474	0.76264584	1.0339714
50	0.9903236	1.0020974	0.031827927	0.86471796	1.0152498
100	0.9842479	0.9946766	0.028286876	0.87199813	1.0065105
200	0.9976638	1.0074166	0.026751818	0.8920307	1.0189484

The confidence interval narrows with increasing number of hidden units.

Deep Factor Models

- ▶ We arrive at our deep factor model:

$$\mathbf{r}_t = F_{W,b}(B_t) + \mathbf{e}_t$$

where $F_{W,b}(X)$ is a deep network with L layers

$$\hat{\mathbf{r}}(X) := F_{W,b}(X) = f_{W^{(L)},b^{(L)}}^{(L)} \circ \cdots \circ f_{W^{(1)},b^{(1)}}^{(1)}(X)$$

Experimental Setup

- ▶ We define the estimation universe as the Russell 1000 index
- ▶ The model has 18 fundamental factors and 31 GICS sector dummy variables
- ▶ Factor exposures are given by Bloomberg and reported monthly
- ▶ Remove symbols with missing factor exposures and any symbols dropped from the index are carried for the next 12 months to avoid excessive turnover
- ▶ Use a 30 year period (with 3290 stocks in total), fit a Deep factor model or an OLS based factor model at each period and use the model to forecast the next period monthly returns.

Experimental Design

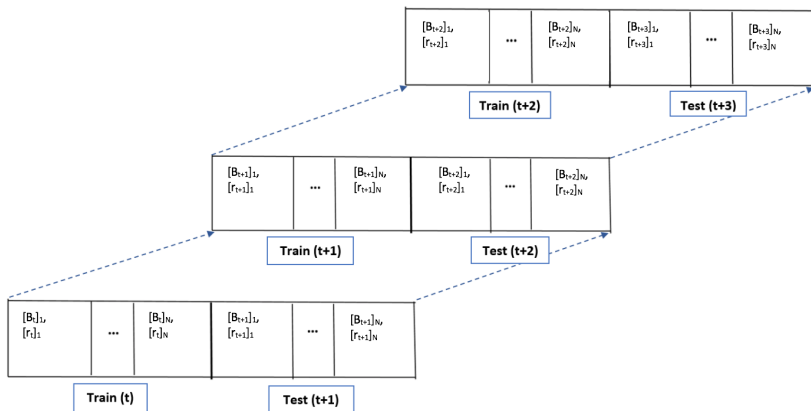
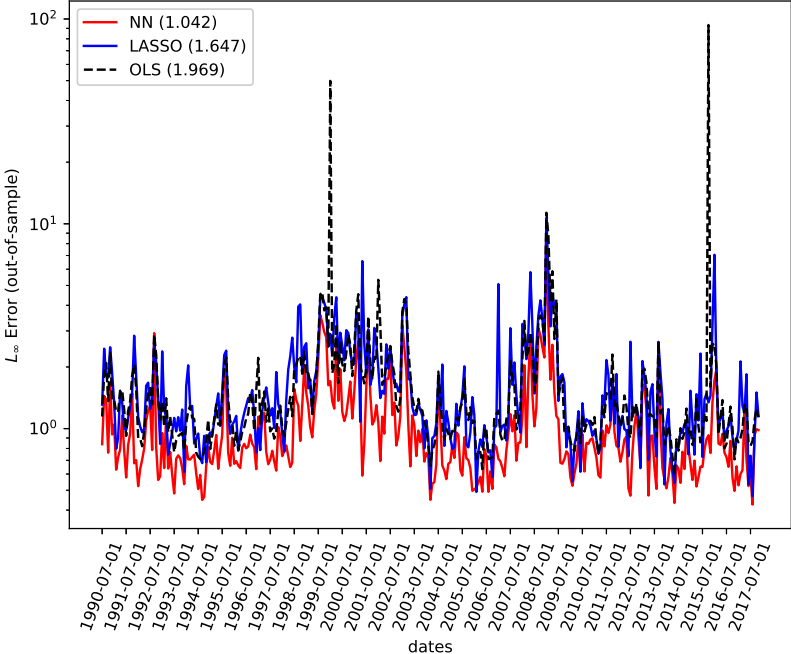


Figure: The experiment is designed so that the factor model is fitted over period t and then tested over period $t + 1$. Each labeled training set is a factor loading matrix B_t and return vector r_t over N assets. Note that the assets need not be fixed over the periods (i.e. permit turnover).

Estimation Error



Estimation Error

- ▶ We observe the ability of the neural network to capture outliers, with the L_∞ norm of the error in the NN being an order of magnitude smaller than in the OLS model at two dates, 2000-01-01 and 2015-10-01
- ▶ The average L_∞ norms over all periods is shown in parenthesis and is a factor of 2x smaller for NNs
- ▶ The L_∞ norm of the OLS error falls to 1.5483 if these two dates are excluded
- ▶ The out-of-sample MSEs for NNs and OLS are 0.026 and 0.254 - the latter decreases to 0.028 with these dates removed.

Portfolio Performance

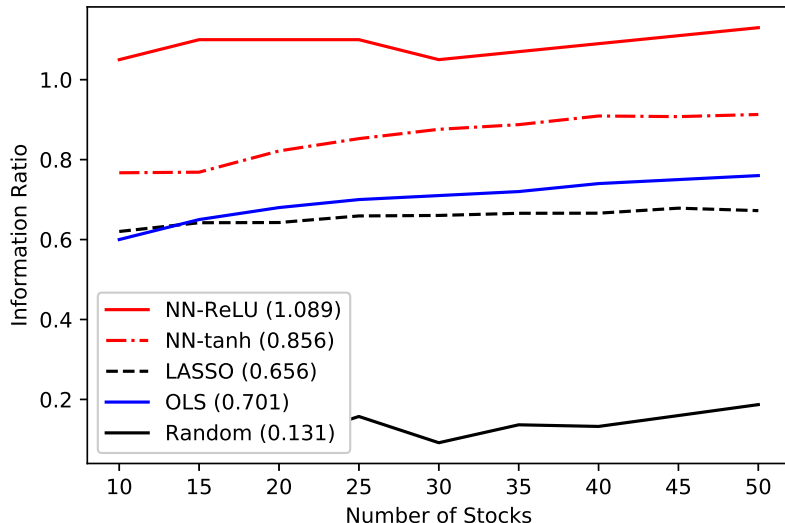


Figure: The information ratios of a portfolio selection strategy which selects the n stocks from the universe with the highest predicted monthly returns. The information ratios are evaluated for various equally weighted portfolios whose number of stocks are shown by the x-axis.

Portfolio Performance

- ▶ The information ratios are evaluated for equally weighted portfolios with varying numbers of stocks.
- ▶ Also shown, for control, are randomly selected portfolios, without the use of a predictive signal.
- ▶ The mean information ratio for each model, across all portfolios, is shown in parentheses.
- ▶ We observe that the information ratio of the portfolio returns, using the deep learning model (with ReLU), is approximately 1.5x greater than the OLS model.
- ▶ We also observe that the information ratio of the baseline random portfolio is small, but not negligible, suggesting sampling bias and estimation universe modification have a small effect.

Sector Tilts

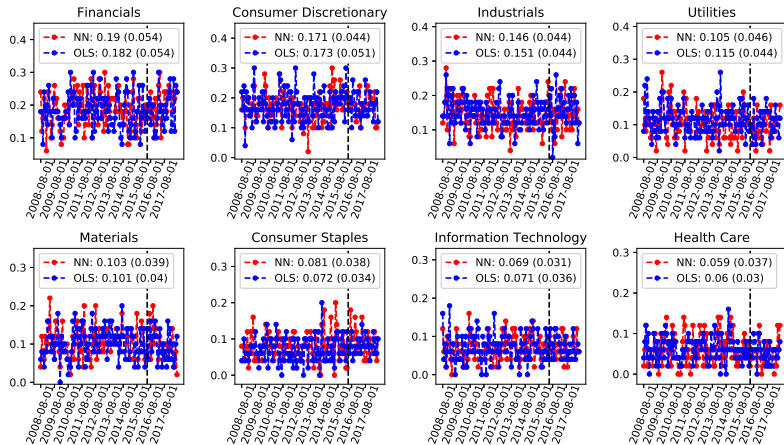


Figure: The sector tilts are shown over time for each sector, in descending order. The mean (and std. dev.) of the sector ratios, over the ten year period, are shown in the legends.

Sector Tilts

- ▶ The figure shows the sector tilts of equally weighted portfolios constructed from the predicted top performing 50 stocks, in each monthly period, over the most recent ten year period in the data.
- ▶ The sectors are ranked by their time averaged ratios, but their tilts vary each month as the portfolios turn-over.
- ▶ Financials is the most dominant sector, with almost 20% time averaged representation. This is followed by Consumer Discretionary. Note that three of the least representative sectors are excluded: Energy, Communication Services, and Real Estate. The sector tilts across the NN and the OLS are found to be comparable on average.
- ▶ The outlier date, 2015-10-01, is marked with a vertical dashed line.

Sector Tilts on Outlier Date

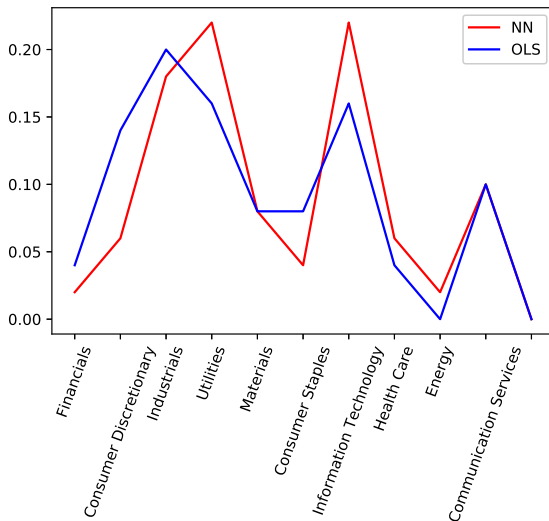


Figure: The sector tilts are compared on the outlier date, 2015-10-1, between OLS and Deep Factor model driven portfolios.

Factor Tilts

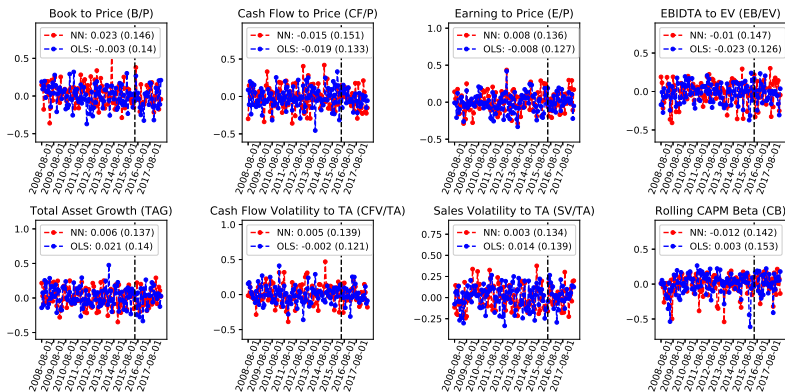
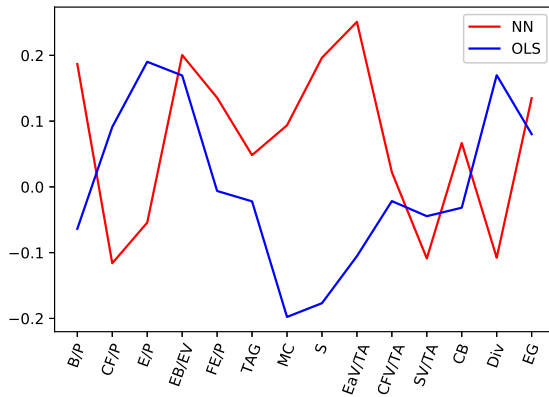


Figure: The (scaled) factors, averaged over the portfolio, are shown over time for a subset of the factors.

Factor Tilts

- ▶ The figure shows the corresponding (scaled) factors, averaged over the portfolio, for a subset of the factors with non-trivial differences in tilts between OLS and NNs.
- ▶ In comparison with OLS, we observe that NNs favor assets with higher Book to Price, Earning to Price, and Cash Flow Volatility to Total Assets.
- ▶ OLS favors stocks with higher Total Asset Growth, Sales Volatility to Total Assets, and Rolling CAPM Beta.
- ▶ Note on the outlier date, the NN portfolio overweights Market Cap, Earnings Volatility to Total Assets, and Sales.

Factor Tilts on Outlier Date



Factor Tilts on Outlier Date

Factor	Description
B/P	Book to Price
CF/P	Cash Flow to Price
E/P	Earning to Price
EB/EV	EBIDTA to EV
FE/P	Forecasted E/P
TAG	Total Asset Growth
MC	Log (Market Capitalization)
S	Log (Sales)
EaV/TA	Earnings Volatility to Total Assets
CFV/TA	Cash Flow Volatility to Total Assets
SV/TA	Sales Volatility to Total Assets
CB	Rolling CAPM Beta
DIV	Dividend yield
EG	Earnings Growth

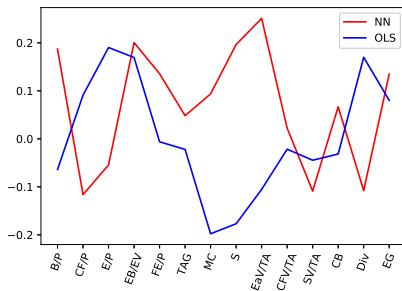
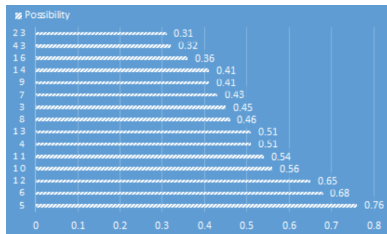


Table: A brief description of the factor abbreviations.⁵

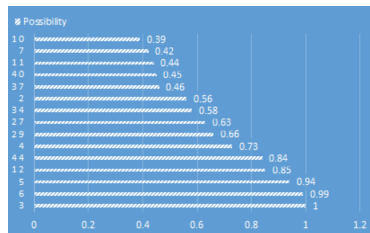
Figure: The factor tilts are compared on the outlier date, 2015-10-1, between OLS and Deep Factor model driven portfolios.

⁵A more detailed description of these factors is provided on Slide 38.

Factor Sensitivities (against OLS)



(a) Deep Neural Network



(b) OLS

Figure: Both the OLS and DNN are fitted to cross-sectional data for every time period. The overall probability that the following factors are in the **top fifteen**, as ranked by sensitivities, is compared between the DNN (left) and OLS (right). For the DNN, EBIDTA to EV (5), Forecasted Earning/Price (6) and Earnings Volatility to Total Assets (12) are the three most likely factors. For the OLS model, Earnings/Price (3), Forecasted Earning/Price (6) and EBIDTA to EV (5) are the top three most likely factors. Hence both models agree on many of the factor importances.

Description of Fundamental Factors

ID	Symbol	Value Factors
1	B/P	Book to Price
2	CF/P	Cash Flow to Price
3	E/P	Earning to Price
4	S/EV	Sales to Enterprise Value (EV). EV is given by $EV = \text{Market Cap} + \text{LT Debt} + \max(\text{ST Debt} - \text{Cash}, 0)$, where LT (ST) stands for long (short) term
5	EB/EV	EBIDTA to EV
6	FE/P	Forecasted E/P. Forecast Earnings are calculated from Bloomberg earnings consensus estimates data. For coverage reasons, Bloomberg uses the 1-year and 2-year forward earnings.
17	DIV	Dividend yield. The exposure to this factor is just the most recently announced annual net dividends divided by the market price. Stocks with high dividend yields have high exposures to this factor.
Size Factors		
8	MC	Log (Market Capitalization)
9	S	Log (Sales)
10	TA	Log (Total Assets)
Trading Activity Factors		
11	TrA	Trading Activity is a turnover based measure. Bloomberg focuses on turnover which is trading volume normalized by shares outstanding. This indirectly controls for the Size effect. The exponential weighted average (EWMA) of the ratio of shares traded to shares outstanding: In addition, to mitigate the impacts of those sharp short-lived spikes in trading volume, Bloomberg winsorizes the data: first daily trading volume data is compared to the long-term EWMA volume (180 day half-life), then the data is capped at 3 standard deviations away from the EWMA average.
Earnings Variability Factors		
12	EaV/TA	Earnings Volatility to Total Assets. Earnings Volatility is measured over the last 5 years/Median Total Assets over the last 5 years
13	CFV/TA	Cash Flow Volatility to Total Assets. Cash Flow Volatility is measured over the last 5 years/Median Total Assets over the last 5 years
14	SV/TA	Sales Volatility to Total Assets. Sales Volatility over the last 5 years/Median Total Assets over the last 5 year
Volatility Factors		
15	RV	Rolling Volatility which is the return volatility over the latest 252 trading days
16	CB	Rolling CAPM Beta which is the regression coefficient from the rolling window regression of stock returns on local index returns
Growth Factors		
7	TAG	Total Asset Growth is the 5-year average growth in Total Assets divided by the Average Total Assets over the last 5 years
18	EG	Earnings Growth is the 5-year average growth in Earnings divided by the Average Total Assets over the last 5 years

Interaction terms

- ▶ Even with one hidden layer, the activation function introduces interaction terms $X_i X_j$.
- ▶ Hence the off-diagonals of the Hessian give the interaction terms.
- ▶ Let's consider a famous toy example, the Friedman test, to illustrate how deep learning captures interaction terms.

$$Y = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + e,$$

Example: Friedman test for interaction terms

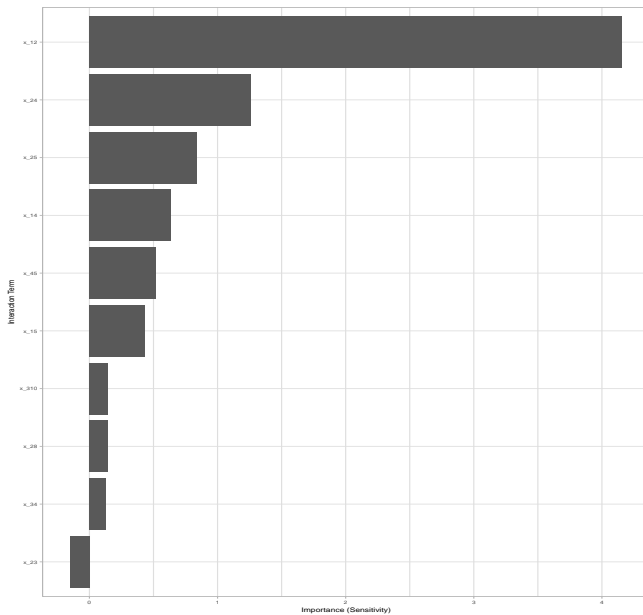


Figure: Friedman test: Estimated interaction terms in the fitted neural network to the input.

Factor Sensitivities (against LASSO)

Factor	Description
E/P	Earning to Price
FE/P	Forecasted E/P
S/EV	Sales to Enterprise Value (EV)
MC	Log (Market Cap.)
TrA	Trading Activity
TAG	Total Asset Growth
EB/EV	EBIDTA to EV
TA	Total Assets
EaV/TA	Earnings Volatility to Total Assets
SV/TA	Sales Volatility to Total Assets
CF/P	Cash Flow to Price
S	Log (Sales)
CFV/TA	Cash Flow Volatility to Total Assets
B/P	Book to Price

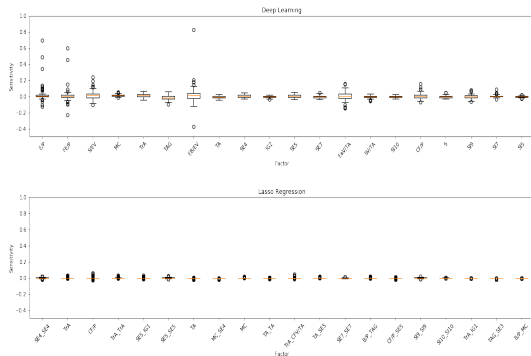


Figure: The distribution of factor model sensitivities and interaction terms over the entire ten year period using the deep neural network applied to the Russell 3000 asset factor loadings (top). The sensitivities are sorted in descending order from left to right by their absolute median values and the top 20 terms have been shown. The same sensitivities using LASSO regression are shown (bottom). Terms with XX.YY denote pairwise interaction effects between factors XX and factors YY. We note that interaction terms feature prominently in the LASSO model but not in the DNN. Trading Activity (TrA), Log Market Cap (MC) and Log (Total Assets) (TA) are observed to be the most important factors which are common between the models.

Deep Fundamental Factor Sensitivities

Factor	Description
E/P	Earning to Price
FE/P	Forecasted E/P
S/EV	Sales to Enterprise Value (EV)
MC	Log (Market Cap.)
TrA	Trading Activity
TAG	Total Asset Growth
EB/EV	EBIDTA to EV
TA	Total Assets
EaV/TA	Earnings Volatility to Total Assets
SV/TA	Sales Volatility to Total Assets
CF/P	Cash Flow to Price
S	Log (Sales)
CFV/TA	Cash Flow Volatility to Total Assets
B/P	Book to Price

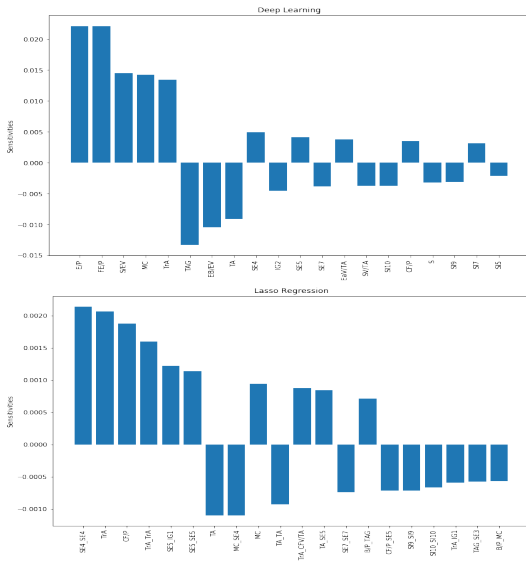


Figure: The medians of factor model sensitivities and interaction terms over the entire ten year period using the deep neural network applied to the Russell 1000 asset factor loadings (top). The same sensitivities using LASSO regression are shown (bottom). The sensitivities to Trading Activity (TrA) and Log Market Cap (MC) are both positive and Log Total Assets (TA) are negative in LASSO and the DNN.

Summary

- ▶ Deep fundamental factor models are developed to automatically capture non-linearity and interaction effects in factor modeling
- ▶ Uncertainty quantification provides interpretability with interval estimation, ranking of factor importances and estimation of interaction effects
- ▶ With no hidden layers we recover a linear factor model and for one or more hidden layers, uncertainty bands for the sensitivity to each input naturally arise from the network weights
- ▶ Using 3290 assets in the Russell 1000 index over a period of December 1989 to January 2018, we assess a 50 factor model and generate information ratios that are approximately 1.5x greater than the OLS factor model. Furthermore, we compare our deep fundamental factor model with a quadratic LASSO model and find less sensitivity to interaction effects (in addition to superior performance)

References

- ▶ Dixon, M.F., I. Halperin, and P. Bilokon (2020). Machine Learning in Finance: From Theory to Practice, Springer
- ▶ Dixon, M.F. and N. Polson (2020). Deep Fundamental Factor Modeling, arXiv:1903.07677
- ▶ Chen, L., M. Pelger, and J. Zhuz (2019, March). Deep learning in asset pricing, arXiv:1904.00745
- ▶ Gu, S., B. T. Kelly, and D. Xiu (2018). Empirical asset pricing via machine learning, Chicago Booth Research Paper 18-04
- ▶ Feng, G., J. He, and N. G. Polson (2018, Apr). Deep Learning for Predicting Asset Returns, arXiv:1804.09314

Definition

A feedforward network is a particular class of multivariate function $F(X)$ constructed using a sequence of L layers via a composite map

$$\hat{Y}(X) := F_{W,b}(X) = \left(f_{W^{(L)},b^{(L)}}^{(L)} \cdots \circ f_{W^{(1)},b^{(1)}}^{(1)} \right) (X).$$

- ▶ $f_{W^{(\ell)},b^{(\ell)}}^{(\ell)}(X) := f^{(\ell)}(W^{(\ell)}X + b^{(\ell)})$ where $f^{(\ell)}$ is a univariate and continuous semi-affine function.
- ▶ $W = (W^{(1)}, \dots, W^{(L)})$ and $b = (b^{(1)}, \dots, b^{(L)})$ are weight matrices and offsets respectively.

Feedforward Networks

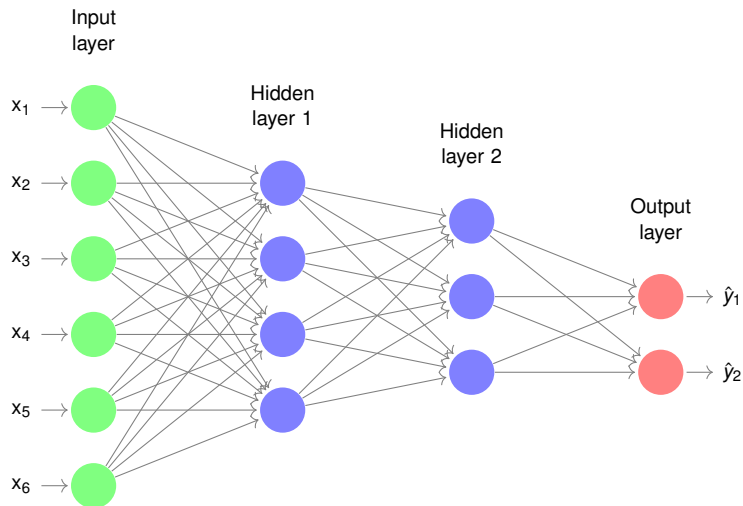


Figure: An illustrative example of a feed-forward neural network with two hidden layers, six features and two output states. Deep learning network classifiers typically have many more layers, use a large number of features and several output states or classes. The goal of learning is to find the weight on every edge that minimizes the out-of-sample error measure.