Nonlinear Forecasting in a Big Data Environment

Deep Learning Approach

Ali Habibnia

DepartmentofEconomics,VirginiaTech

alihabibnia@gmail.com

2nd Big Data Economics Summer School (Aug 8th , 2018)

Reference

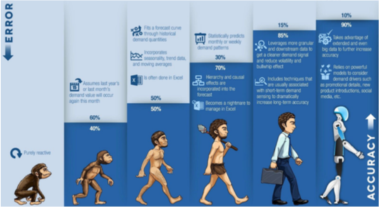
Habibnia, Ali (2016) Essays in high-dimensional nonlinear time series analysis. PhD thesis, London School of Economics and Political Science (LSE).

- *Past, Present and Future of Testing for Nonlinearity in and between Time Series*
- *Nonlinear Forecasting Using a Large Number of Predictors: a Nonlinear Generalization of the Statistical Factor Models*
- *Forecasting in Big Data Environments: a Shrinkage Estimation of Skip-layer Neural Networks*, with E. Maasoumi

Motivation and Inspirations

Design novel statistical learning techniques to model the complexity of large datasets.

- Curse of dimensionality → Blessing of dimensionality
- Relaxing unrealistic assumptions of the classical models
- A resurgence in the field of machine learning & neural networks
- Real world series are rarely purely linear or nonlinear


ERROR
Purely reactive
Assumes last year's or last month's demand value will occur again this month
60%
40%
Fits a forecast curve through historical demand quantities
Incorporates seasonality, trend data, and moving averages
Is often done in Excel
50%
50%
Statistically predicts monthly or weekly demand patterns
30%
70%
Hierarchy and causal effects are incorporated into the forecast
Becomes a nightmare to manage in Excel
15%
85%
Leverages more granular and downstream data to get a cleaner demand signal and reduce volatility and bullwhip effect
Includes techniques that are usually associated with short-term demand sensing to dramatically increase long-term accuracy
10%
90%
Takes advantage of extended and even big data to further increase accuracy
Relies on powerful models to consider demand drivers such as promotional details, new product introductions, social media, etc.
ACCURACY

Question

Is it possible to forecast with a high-dimensional panel of predictors while considering nonlinear dynamic among variables?

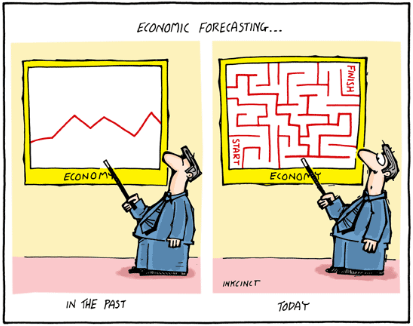
- **Curse of dimensionality**
  - Feature extraction (i.e, Factor Models - Stock and Watson (2002,2006); Bai and Ng (2002); Deistler and Hamman (2005); Forni, Hallin, Lippi, and Reichlin (2005); Lam and Yao (2012), and several others.)
  - Feature selection (i.e, Ridge - Hoerl and Kennard (1970); LASSO -Tibshirani (1996); Elastic Net - Zou and Hastie (2005), Bayesian regression - Mol, Giannone, and Reichlin(2008); Selecting variables -Bai and Ng(2008a))
- **To model complex and nonlinear data**
  - Parametric, semiparametric and nonparametric nonlinear regression models (i.e, TAR & STAR - Teräsvirta, Tjøstheim, and Granger (2010); Neural nets - Kuan and White (1994), Teräsvirta, van Dijk, and Medeiros (2005), Mederios, Träsvirta and Rech (2005) and Varian (2014))

ML for complex and nonlinear phenomena

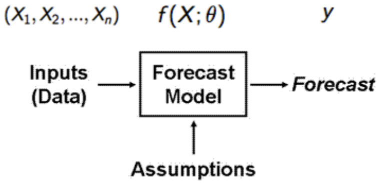
- However linear regression models are adequate to explain many phenomena in the world, most important economic and financial phenomena are complex and nonlinear in nature.

- Parametric nonlinear regression models:
  - The shape of the functional relationships between the response and the predictors are predetermined
  - Can take the form of a polynomial, exponential, trigonometric, power, or any other nonlinear function

- Nonparametric and semiparametric models :
  - In many situations, the relationship is unknown
  - The shape of the functional relationships between variables can be adjusted to capture unusual or unexpected features of the data
  - Artificial Neural Networks, Kernel-based methods & Tree-based regression models



2011-503 © INKCINCT Cartoons www.inkcinct.com.au



$(X_1, X_2, ..., X_n)$ $f(X;\theta)$ y

Inputs (Data) → Forecast Model → *Forecast*

Assumptions

$$y_{t+h|t} = f(X_{t-i|t};\theta)$$

- Building accurate forecast models in economics and finance is a complex and challenging task.

- **In this talk:** we will see how to apply appropriate and novel techniques to design data driven forecast models in few steps from data mining and model selection to forecasts evaluation and comparison. Each step has its own tricks!

Big Data

- An important step in designing modern predictive models is to cope with high-dimensional data, which contain large numbers of correlated variables and present complex properties.

- “Big data” is both an increase in the number of samples collected over time, and an increase in the number of potential explanatory variables and predictors that are simultaneously measured.

- When using nonlinear tools such as artificial neural networks. Most nonlinear models involve more parameters than the dimension of the data space which may result in a lack of model identifiability, instability, and overfitting.

Nonlinearity:

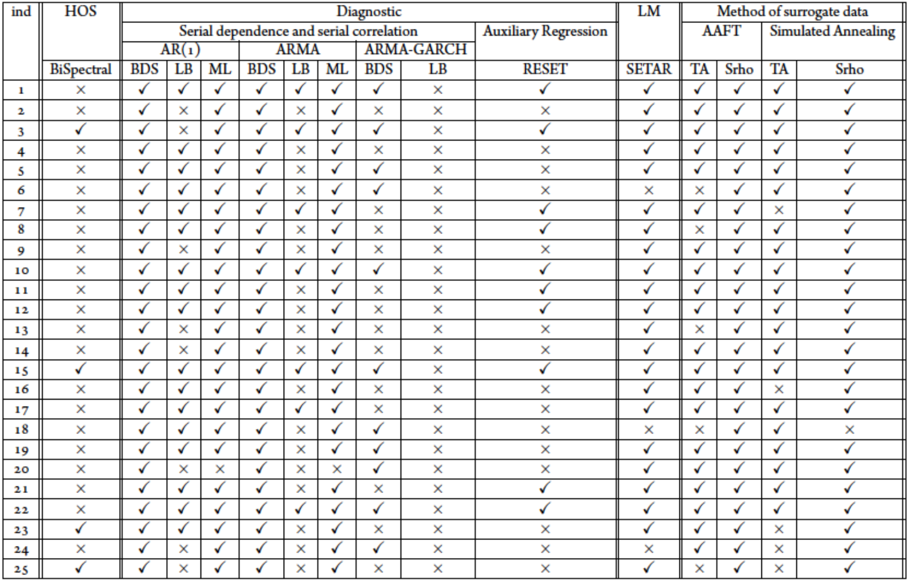
- A linear stochastic process can be represented in terms of an arithmetic sequence of independent and identically distributed random variables in time domain or the power spectrum in the frequency domain. Any stochastic process that does not satisfy the condition of the those representations is said to be nonlinear.

- Nonlinearity may arise in different ways. The characteristic of nonlinear time series such as higher-moment structures, time-varying variance, asymmetric fluctuations, thresholds and breaks can be only modelled by an appropriate nonlinear function like $f(.)$ and a linear process is not adequate to model these features.

- Before we apply nonlinear techniques, such as those inspired by machine learning theories, to real-world financial data, it is logical to first ask if the use of such techniques is justified by the data.

Nonlinearity in & Between (Financial) Series

Table 2.4.1: The results of the application of linearity tests to equity returns. A rejection of the null hypothesis of linearity is shown by ✓.



ind	HOS	Diagnostic									LM	Method of surrogate data			
		Serial dependence and serial correlation								Auxiliary Regression		AAFT		Simulated Annealing	
		AR(1)			ARMA			ARMA-GARCH							
	BiSpectral	BDS	LB	ML	BDS	LB	ML	BDS	LB	RESET	SETAR	TA	Srho	TA	Srho
1	×	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓
2	×	✓	×	✓	✓	×	✓	×	×	×	✓	✓	✓	✓	✓
3	✓	✓	×	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓
4	×	✓	✓	✓	✓	×	✓	×	×	×	✓	✓	✓	✓	✓
5	×	✓	✓	✓	✓	×	✓	✓	×	×	✓	✓	✓	✓	✓
6	×	✓	✓	✓	✓	×	✓	✓	×	×	×	×	✓	✓	✓
7	×	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓	✓	×	✓
8	×	✓	✓	✓	✓	×	✓	×	×	✓	✓	×	✓	✓	✓
9	×	✓	×	✓	✓	×	✓	×	×	×	✓	✓	✓	✓	✓
10	×	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓
11	×	✓	✓	✓	✓	×	✓	×	×	✓	✓	✓	✓	✓	✓
12	×	✓	✓	✓	✓	×	✓	×	×	✓	✓	✓	✓	✓	✓
13	×	✓	×	✓	✓	×	✓	×	×	×	✓	×	✓	✓	✓
14	×	✓	×	✓	✓	×	✓	×	×	×	✓	✓	✓	✓	✓
15	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓
16	×	✓	✓	✓	✓	×	✓	×	×	×	✓	✓	✓	×	✓
17	×	✓	✓	✓	✓	✓	✓	×	×	×	✓	✓	✓	✓	✓
18	×	✓	✓	✓	✓	×	✓	✓	×	×	×	×	✓	✓	×
19	×	✓	✓	✓	✓	×	✓	✓	×	×	✓	✓	✓	✓	✓
20	×	✓	×	×	✓	×	×	✓	×	×	✓	✓	✓	✓	✓
21	×	✓	✓	✓	✓	×	✓	×	×	✓	✓	✓	✓	✓	✓
22	×	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓
23	✓	✓	✓	✓	✓	×	✓	×	×	×	✓	✓	✓	×	✓
24	×	✓	×	✓	✓	×	✓	✓	×	×	×	✓	✓	×	✓
25	✓	✓	×	✓	✓	×	✓	×	×	×	✓	×	✓	×	✓

Forecasting with Factor Models

Linear (statistical) factor models:

Given a high-dimensional matrix of stationary time series (i.e. financial returns), denoted by x_{it} $(i = 1, ..., m, t = 1, ..., T)$

- **Factor estimation step**

  (PCA, MLE, Kalman-Filter,...) PCA finds the projection such that the best linear reconstruction of the data is as close as possible to the original data.

$$x_{it} = \lambda_i' u_t + \xi_{it}$$

- **Forecasting step**

$$(\hat{y}_{T+1|T}) = \hat{x}_{iT+1|T} = \hat{\lambda}_i' u_{T+1|T}$$

Neural Networks: one of the oldest and one of the newest areas

- Formulation of a feedforward neural network model with one hidden layer can be generalized to

$$y_t = \Phi(x; w) = \phi_k \left[\sum_{j \to k} \phi_j \left(\sum_{i \to j} x_{it} w_{ij} \right) w_{jk} \right] + \varepsilon_t$$

  where Φ describes network by a vector function. We associate subscript i with the input layer, subscript j with the hidden layer, and subscript k with the output layer.

- To show that the neural network models can be seen as a generalization of linear models, we assumed that the output transfer function $\{\phi_k(.)\}$ is linear, then the model becomes

$$y_t = \sum_{j \to k} \phi_j \left(\sum_{i \to j} x_{it} w_{ij} \right) w_{jk} + \varepsilon_t$$

Model 1 - Nonlinear Factor Models

- **Factor estimation using neural network PCA (for details see; Oja (1982); Kramer (1991); Hsieh (2004); Hinton and Salakhutdinov (2006))**

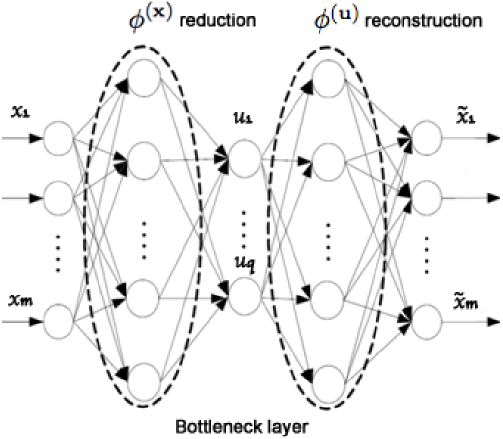


Figure: Schematic diagram of the standard autoassociative neural network architecture for calculating the nonlinear principal component analysis (NLPCA).

Model 1 - Nonlinear Factor Models

- **Factor estimation using neural network PCA**
  PCA can be nonlinearly generalized NLPCA/Autoencoders (Bottleneck, Autoassociative)
  An autoencoder is a neural network which is trained to replicate its input at its output. Autoencoders can be used as tools to learn deep neural networks. Training an autoencoder is unsupervised in the sense that no labeled data is needed. The training process is still based on the optimization of a cost function. The cost function measures the error between the input and its reconstruction at the output. An autoencoder is composed of an encoder (mapping) and a decoder (demapping). The encoder and decoder can have multiple layers.
  Here, both mapping $\mathbf{u}_t = \phi^{(x)}(\mathbf{x}_t)$ and demaping $\tilde{\mathbf{x}}_t = \phi^{(u)}(\mathbf{u}_t)$ functions are approximated by neural nets.

  The loss of information is again measured by $\xi_t = x_t - \tilde{x}_t$, and analogous to PCA, the functions $\phi^{(x)}$ and $\phi^{(u)}$ are selected to minimise $||\xi||$.

$$
\begin{aligned}
y_t^{(x)} &= \phi_j^{(x)}(z^{(x)}) \\
u &= \phi_k^{(x)}(y_t^{(x)}) \\
y_t^{(u)} &= \phi_j^{(u)}(z^{(u)}) \\
\tilde{x}_t &= \phi_k^{(u)}(y_t^{(u)})
\end{aligned}
$$

Model 1 - Nonlinear Factor Models

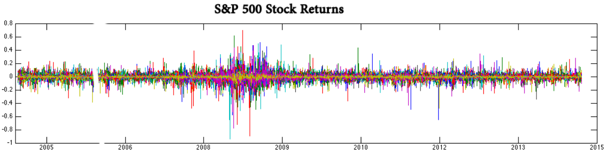
- **Nonlinear forecasting step**

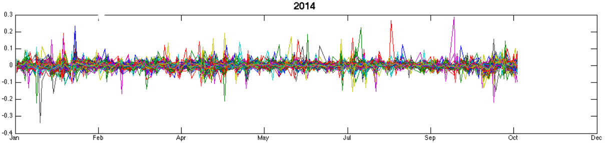
$$\text{Linear Factor Model} \begin{cases} \hat{x}_{iT+1|T} = \hat{\beta}_i' \hat{u}_T \\ \hat{x}_{iT+1|T} = \hat{\lambda}_i' u_{T+1|T} \\ \hat{x}_{iT+1|T} = \hat{\lambda}_i' u_{T+1|T} + \hat{\xi}_{iT+1|T} \end{cases}$$

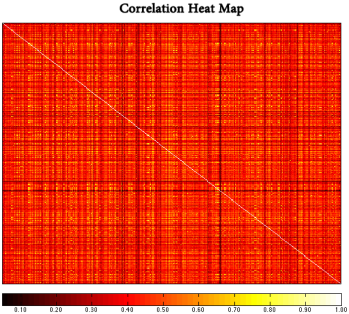
$$\text{Nonlinear Factor Model} \begin{cases} \hat{x}_{iT+1|T} = \Phi(\hat{u}_T^{(NL)}) \\ \hat{x}_{iT+1|T} = \phi_k^{(u^{NL})}(\phi_j^{(u^{NL})}(\hat{u}_{T+1|T}^{(NL)})) \\ \hat{x}_{iT+1|T} = \phi_k^{(u^{NL})}(\phi_j^{(u^{NL})}(\hat{u}_{T+1|T}^{(NL)})) + \hat{\xi}_{iT+1|T}^{(NL)} \end{cases}$$

Empirical Analysis

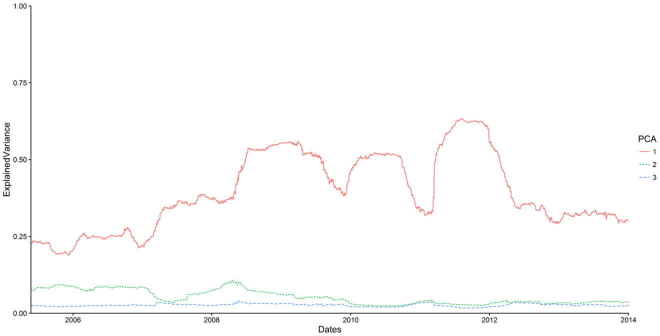
- The data are daily returns of $m = 418$ equities on the S&P 500 index from 04.01.2005 through 31.12.2014.
- We calculate 1-step (here one day) ahead forecasts of targets ($\hat{x}_{it+1|t}$ return series to be forecast) based on a rolling (moving) estimation window.

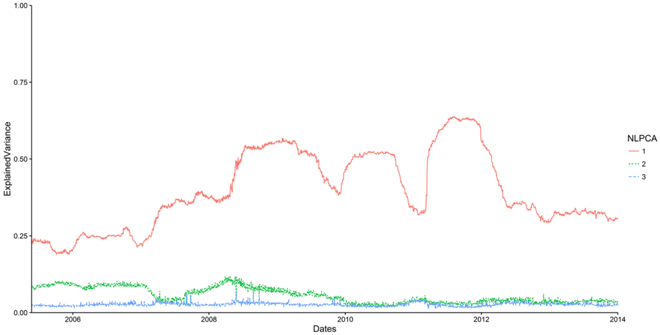

S&P 500 Stock Returns


2014


Correlation Heat Map

Fraction of the variance explained by the first three PCs





Nonlinearity Between (Financial) Series

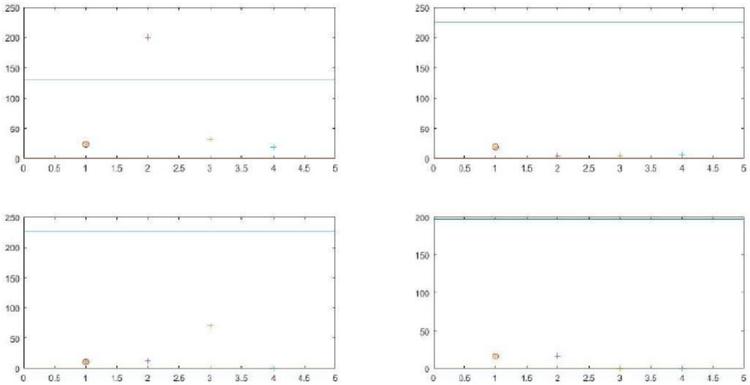


Figure 3.4.8: Accuracy bounds and residual variances. Sample is divided into smaller disjunct regions; and accuracy bounds are determined for the sum of the discarded eigenvalues of each region. If this sum is within the accuracy bounds for each region, the process is assumed to be linear. Conversely, if at least one of these sums is outside, the process is assumed to be nonlinear. As the figure illustrates, the recorded financial data is nonlinear.

Comparing the forecastability of alternative quantitative models

- Time series approach (R2, RMSE, MAE, Hit Rate, Mean Profit per Day, DM-test ...)

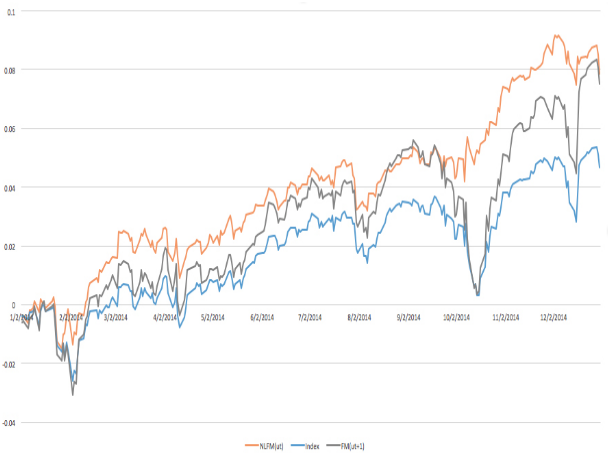
  $$RMSE = \sqrt{\frac{1}{N}\sum_{t=h+1}^{N}(y(t)-\hat{y}(t))^2}$$

  $$Hit - Rate = \frac{|\{t|y(t)\hat{y}(t)>0,t=1,..,N\}|}{|\{t|y(t)\hat{y}(t)\neq0,t=1,..,N\}|}$$

  how often the sign of the return is correctly predicted. HR>0.5 better than RW

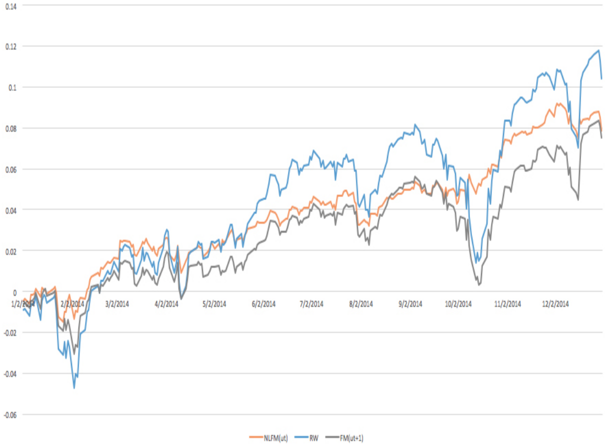
- Trading simulation approach (Profit of a portfolio with one asset or more than one assets)
  - In time series approach, models aim to minimise Out-of-sample forecasting errors, however, the model with minimum statistical errors does not necessarily guarantee maximised trading profits, which is often deemed as the ultimate objective of financial application.
  - Since the ultimate goal of investment is to make profit, the best way to evaluate alternative financial forecast model is therefore to evaluate their trading performance.
- Benchmark for trading simulation:
  - The performance of AR(1), RW or the stock market index during the same out-of-sample period.

Comparison of linear and nonlinear factor models and the benchmark models based on the performance of the portfolio simulation



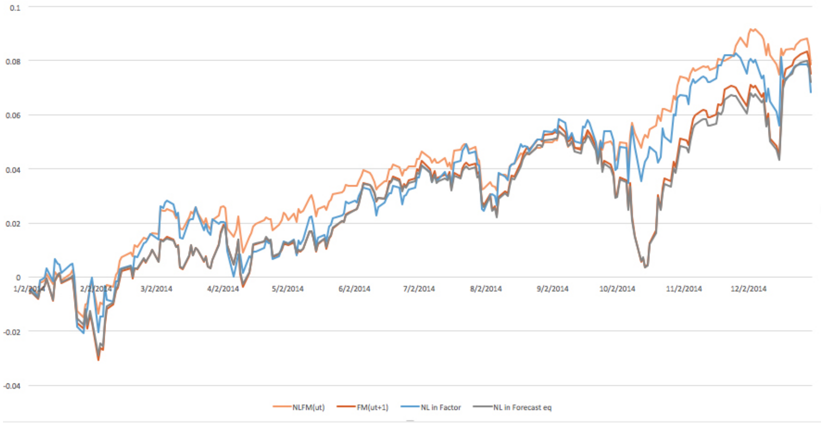
(a) Linear and nonlinear factor models against an investment on S&P 500 index

Comparison of linear and nonlinear factor models and the benchmark models based on the performance of the portfolio simulation



(b) Linear and nonlinear factor model against Random walk

Comparison of linear and nonlinear factor models, and the models with only one nonlinear step based on the performance of the portfolio simulation



Portfolio	Return	Sharp Ratio
Linear FM	7.51%	17.4927
Nonlinear FM	7.87%	25.0019
Nonlinear in factor estimation step	7.61%	18.6259
Nonlinear in forecast equation step	6.83%	17.8577

Comparison of linear and nonlinear factor models based on the performance of the portfolio simulation during an out-of-sample period

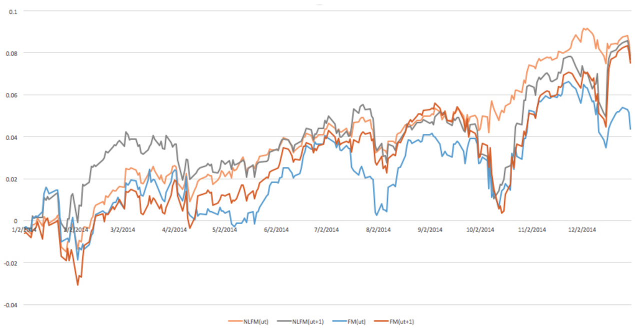


Table:

Portfolio	Return	Sharp Ratio
FM(u_t)	4.35%	9.1770
FM(u_{t+1})	7.51%	17.4927
NLFM(u_t)	7.87%	25.0019
NLFM(u_{t+1})	7.41%	18.8963

Comparison of linear and nonlinear factor models and the Hybrid model based on the performance of the portfolio simulation during out-of-sample period

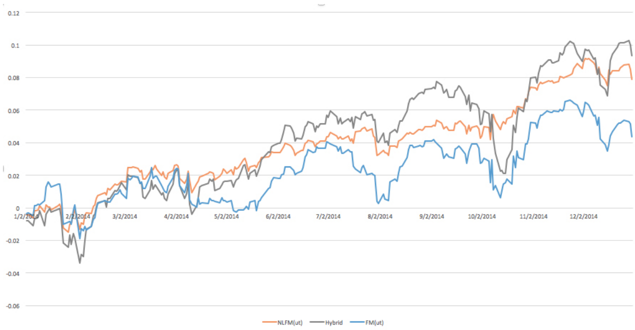


Table:

Portfolio	Return	Sharp Ratio
Linear FM	7.51%	17.4927
Nonlinear FM	7.87%	25.0019
Hybrid model	9.32%	19.2152

Model 2 - Shrinkage Estimation of Skip-layer Neural Networks

It is challenging to determine if complex real world time series behave in a linear or nonlinear fashion. The experimental results from different

linearity tests suggest that the real world series are rarely purely linear or nonlinear. They consists of both linear and nonlinear patterns. We allow that series are composed of a linear structure ($\mathcal{L}_t$) plus a nonlinear component ($\mathcal{N}_t$).

$$y_t = \mathcal{L}_t + \mathcal{N}_t$$

Two different approaches to model and forecast time series with both linear and nonlinear patterns are imaginable. Hybrid methodology which, first we estimate the linear component using a linear model and then we collect the residuals obtained from the fitted model $\hat{e}_t = y_t - \hat{\mathcal{L}}_t$. Finally we let a nonlinear approach (i.e, GARCH family models, neural nets) to model the residuals which may contain information about nonlinearity.

Model 2 - Shrinkage Estimation of Skip-layer Neural Networks

Model includes both linear and nonlinear structures. This is a high-dimensional learning approach including both sparsity L_1 and smoothness L_2 penalties, allowing high-dimensionality and nonlinearity to be accommodated in one step.

$$y_t = \Phi(x; w) = \sum_{i \to k} x_{it} w_{ik} + \sum_{j \to k} \phi_j \left(\sum_{i \to j} x_{it} w_{ij} \right) w_{jk} + \varepsilon_t,$$

where Φ describe network by a vector function. We associate subscript i with the input layer, subscript j with the hidden layer, and subscript k with the output layer. $x_{it} = (x_{1t}, x_{2t}, ..., x_{mn})$ is the value of the ith input node, which can be a constant input representing biases, a matrix of lagged values of y_t and some exogenous variables. $\phi_j(.)$ and J are activation functions and number of neurons used at the hidden layer.

Model 2 - Shrinkage Estimation of Skip-layer Neural Networks

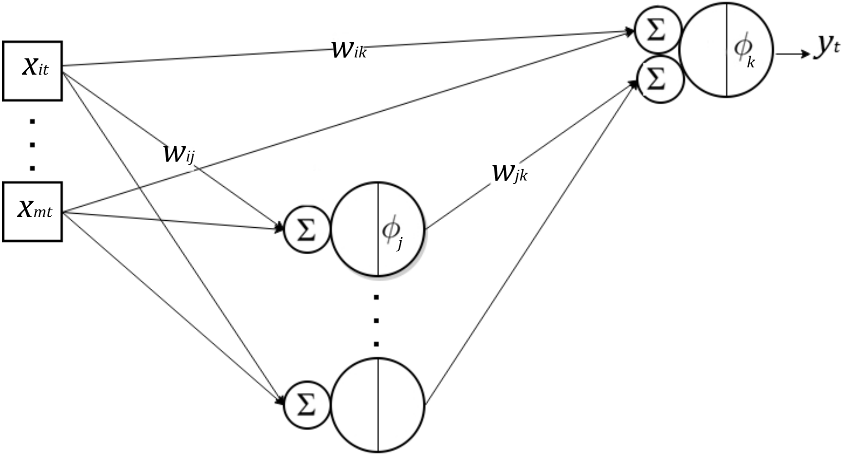


Figure: A single-hidden-layer neural network with skip-layer connections

Model 2 - Shrinkage Estimation of Skip-layer Neural Networks

Network parameters are the solutions to the following optimization problem:

$$w^* = \underset{w}{\operatorname{argmin}}\ E(w) \ + \ \frac{\lambda_2}{2} \sum_{i \to k} w_{ik}^2 \ + \ \lambda_1 (\sum_{i \to j} |w_{ij}| + \sum_{j \to k} |w_{jk}|)$$

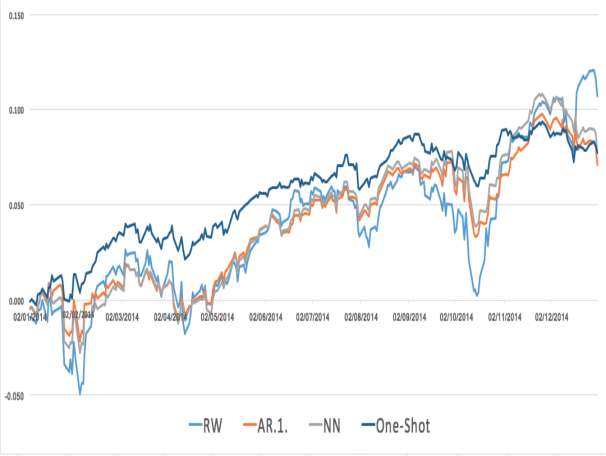
Then the learning rule for the weights becomes:

$$\begin{cases} w_{ik}^{new} = w_{ik}^{old} \ - \ \eta(\frac{\partial E(w)}{\partial w_{ik}} + \lambda_2 \ w_{ik}^{old}) \\ w_{ij}^{new} = w_{ij}^{old} \ - \ \eta(\frac{\partial E(w)}{\partial w_{ij}} + \lambda_1 \ sgn(w_{ij}^{old})) \\ w_{jk}^{new} = w_{jk}^{old} \ - \ \eta(\frac{\partial E(w)}{\partial w_{jk}} + \lambda_1 \ sgn(w_{jk}^{old})) \end{cases}$$

The optimal λ can be found by a gradient descent scheme instead of setting that manually using grid search or cross-validation (Larsen et al.(2012) and Maclaurin et al.(2015)).

$$\lambda^{new} = \lambda^{old} - \gamma \ \frac{\partial E_V}{\partial \lambda}(\hat{w}(\lambda^{old}))$$

Comparison of one-shot model, and the competing models based on the performance of the portfolio simulation



(a) One-shot model against competing models

Nonlinearity: we let the data speak for themselves as much as possible

- Classification of different statistical approaches which are testing nonlinearity in time series is a challenging task as they entail consideration of various types of nonlinear dynamics and are coming from different disciplines. Granger and Tersvirta (1993), Tersvirta, Tjstheim and Granger (1994) and recently Giannerini (2012)

- The main idea behind various linearity tests is a hypothesis testing procedure. Every hypothesis test starts with a null hypothesis (H_0) and an alternative (H_1). In general, the null hypothesis of linearity tests states that observed series are generated by Gaussian linear stochastic processes against an alternative hypothesis that states observed series are rooted in nonlinear dynamics.

- To be more precise, H_0 tests the hypothesis that the time series is completely specified by its first and second order statistics (i.e. mean, variance, and autocorrelation or its frequency domain counterpart, power spectrum) .