Communications for Statistical Applications and Methods
2019, Vol. 26, No. 5, 497–506

https://doi.org/10.29220/CSAM.2019.26.5.497
Print ISSN 2287-7843 / Online ISSN 2383-4757

The roles of differencing and dimension reduction in machine learning forecasting of employment level using the FRED big data

Ji-Eun Choi[a], Dong Wan Shin[1,a]

[a]Department of Statistics, Ewha Womans University, Korea

Abstract

Forecasting the U.S. employment level is made using machine learning methods of the artificial neural network: deep neural network, long short term memory (LSTM), gated recurrent unit (GRU). We consider the big data of the federal reserve economic data among which 105 important macroeconomic variables chosen by McCracken and Ng (*Journal of Business and Economic Statistics*, **34**, 574–589, 2016) are considered as predictors. We investigate the influence of the two statistical issues of the dimension reduction and time series differencing on the machine learning forecast. An out-of-sample forecast comparison shows that (LSTM, GRU) with differencing performs better than the autoregressive model and the dimension reduction improves long-term forecasts and some short-term forecasts.

Keywords: employment forecast, deep neural network, differencing, dimension reduction, long short term memory, gated recurrent unit

1. Introduction

Employment level plays an important role in making government labor policy, understanding the overall economic conditions and planning business investment. Therefore, forecasting employment level is crucial for government policy makers, investors, and many others. Accordingly, many studies for it have been conducted, see Rapach and Strauss (2010, 2012), Siliverstovs (2013), Lehmann and Weyh (2016) and many others for recent studies. The forecasting methods of these authors are based on statistical or economic models, such as autoregressive integrated moving average (ARIMA) model, vector autoregression and factor analysis. The recent hot applications of machine learning methods to diverse statistical problems of classification and forecasting render us to consider the artificial neural network (ANN), one of the machine learning methods, for employment level forecast.

ANN is an interesting forecasting method in that the method is capable of addressing a nonlinear structure between the employment level and predictor variables without econometric intuition. Substantial forecast efficiency gain will be demonstrated for the machine learning forecast of employment level using ANN methods of deep neural network (DNN), long short term memory (LSTM), and gated recurrent unit (GRU) over the standard AR forecast if a big data approach is used with a careful statistical consideration of dimension reduction and time series differencing. In ANN, in order to handle more complex nonlinear relationships, two or more hidden layers are added and this model is called DNN. However, DNN has limitations in that it does not address the serial dependence

[1] Corresponding author: Department of Statistics, Ewha Womans University, 52, Ewhayeodae-gil, Seodaemun-gu, Seoul 03760, Korea. E-mail: shindw@ewha.ac.kr

Published 30 September 2019 / journal homepage: http://csam.or.kr

of most of economic time series. A recurrent neural network (RNN) is then proposed to address serial dependence. Improved modifications of the RNN have appeared: the LSTM of Hochreiter and Schmidhuber (1997) and GRU of Cho *et al.* (2014) are proposed. DNN and RNN have been used for stock market forecasting in many recent studies: Arevalo *et al.* (2016) for US Apple stock price, Chong (2017) for KOSPI returns, Chiang *et al.* (2016) for trading signal of the world 22 stock market indices, and Qju *et al.* (2016) for Japan Nikkei 225 index return.

For the machine learning forecast methods of ANN for employment level, we consider the big data of the federal reserve economic data (FRED) as predictors. The FRED is a huge big database managed by the Federal Reserve Bank of St. Louis and is composed of more than 500,000 economic time series related to banking, employment, population, and consumer price indexes.

The FRED is huge and contains unit root series. Therefore, we need to consider two statistical issues of dimension reduction and time series differencing. McCracken and Ng (2016) choose 105 important macroeconomic variables among the over-500,000 variables in the FRED. They also provided background information for these important variables and the transformation method of each series, such as degree of differencing and log transformation. The recommendation by McCracken and Ng (2016) will be applied to machine learning forecasting of employment level.

We identify that consideration of the two statistical issues improves the forecast performance of the ANN methods. Out-of-sample forecast comparison with a model confidence set (MCS) analysis of Hansen *et al.* (2011) is conducted to compare the methods of (AR, DNN, LSTM, GRU) combined with (differencing, non-differencing) and (dimension reduction, non-dimension reduction). The comparison reveals that (LSTM, GRU) forecasts with differencing are substantially better than benchmarking AR forecast and that dimension reduction improves long-term forecast and some short-term forecast.

The remaining of the paper is organized as follows. Section 2 describes the FRED. Section 3 explains forecast methods. Section 4 makes an out-of-sample forecast comparison. Section 5 gives the conclusion.

2. Federal reserve economic data

We consider the FRED in forecasting the U.S. monthly civilian employment level for the period of 01/01/1985–12/01/2018 of $T = 408$ months. The employment data set and the FRED data set can be downloaded from the FRED website (https://fred.stlouisfed.org). In the website, the employment level is defined to be "the number of persons of 16 years of age and older residing in the 50 U.S. states and the District of Columbia, who are not inmates of institutions and who are not on active duty in the Armed Forces", see the FRED website for more details.

The FRED is a huge big database maintained by the Federal Reserve Bank of St. Louis. The FRED are collected from global financial institutions and U.S. government agencies such as the U.S. Census and Bureau of Labor Statistics. The database contains various categories of economic and financial data: banking, employment and population, gross domestic product, interest rates, and consumer price indexes. McCracken and Ng (2016) reduced the over-500,000-dimensional FRED, say X^F, to smaller dimensional data, say X^{MN}, of important macroeconomic variables and discussed background information of X^{MN}. They also discussed transformation method of each series in X^{MN} such as log transformation and the degree of differencing by checking whether the series is $I(0)$, $I(1)$, or $I(2)$. The reduced data set X^{MN} is also a big one containing $K = 105$ variables. As discussed by McCracken and Ng (2016), the 105 variables X^{MN} is chosen to satisfy the four criteria established by Stock and Watson (1996) to include important macroeconomic categories of leading economic indicators and to

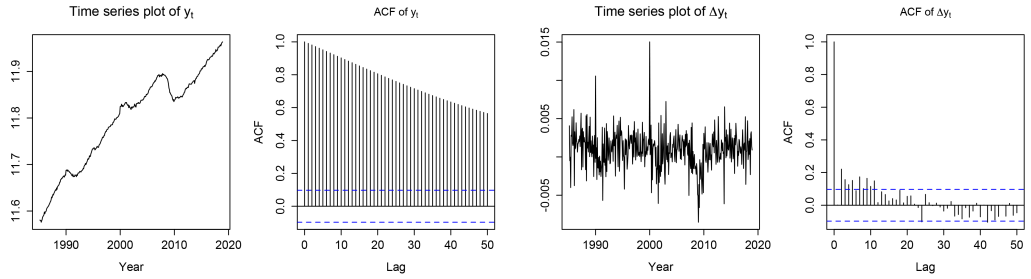

Figure 1: *Time series plots and ACF plots of the log employment level Y_t and its difference ΔY_t. ACF = autocorrelation function.*

represent different broad categories of macroeconomic variables not selected. Therefore, the set of reduced variables X^{MN} is a good summary of the original huge big data X^F including majority of the important macroeconomic categories. We consider this reduced big data X^{MN} to improve forecast of log employment level, say Y, via machine learning.

The summary analysis by McCracken and Ng (2016) are mainly differencing and dimension reduction. When we use the reduced FRED big data X^{MN} in forecasting the employment level, Y say, the issue of differencing and dimension reduction is non-trivial because the big data set X^{MN} contains a large number of variables having different dynamics. For example, many variables in X^{MN} have one unit root, other variables have zero or two unit roots. We demonstrate that machine learning forecast fails unless proper differencing is considered for X^{MN}. Further reduction of X^{MN} to X^R will be considered to show that dimension reduction matters in terms of forecast horizons. Let T be the time series dimension and $K = 105$ be the number of variables in X^{MN}. Then, $Y = (Y_1, \ldots, Y_T)'$ and $X^{\mathrm{MN}} = (X^{\mathrm{MN}}_{it},\ t = 1, \ldots, T,\ i = 1, \ldots, K)_{T\times K}$.

We will briefly discuss what difference order will be considered for each element in Y and X^{MN}. The log employment level Y_t is differenced by order one according to the following analysis. Figure 1 is the time series plots and the autocorrelation function (ACF) plots of original and differenced log employment level. From the increasing trend in the figure and very slowly decreasing ACF, we can identify a need for first-order differencing for the log employment level. The need is also confirmed by the p-value = 0.26 of ADF test statistic with AIC order = 11 and time trend. We will use those recommended by McCracken and Ng (2016) for the order $d_i \in \{0, 1, 2\}$ of differencing of each series X^{MN}_{it} in X^{MN}.

3. Forecast methods

We forecast the log employment level Y using machine learning methods of ANN for which we need to consider two statistical issues of differencing and dimension reduction. We describe ANN forecast methods in Section 3.1 and discuss the two issues in Sections 3.2 and 3.3.

3.1. Artificial neural network

The forecast methods based on ANN have received significant attention. ANN is one of the machine learning methods inspired by biological neural networks. Keeping in mind of the implementation of the ANN methods in Section 4 for forecasting Y from predictor X, we describe the key concept of ANN learning. Let a data set $\{X_t = (X_{1t}, \ldots, X_{Kt})', Y_t, t = 1, \ldots, T\}$ be given for forecasting Y with predictor X. ANN consists of an input layer receiving predictors $X_t = (X_{1t}, \ldots, X_{Kt})'$, a hidden layer

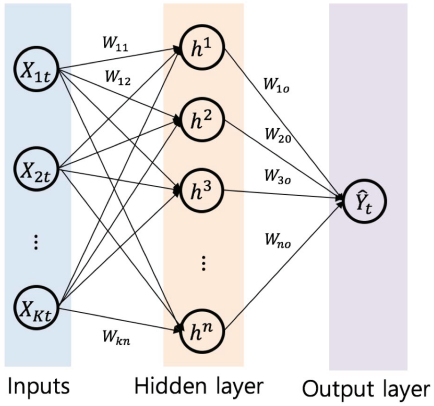

Figure 2: *Artificial neural network model with K predictors, n hidden nodes and one output.*

composed of hidden nodes, and an output layer giving forecast value $\hat{Y}_t$ (Figure 2).

In ANN, forecasting is made through two step processes: first, the ANN transmits the nonlinear function value of the linear combination of predictors received in the input nodes to each hidden node, for example $h^1 = \phi(\sum_{i=1}^{K} W_{i1}X_{it} + b_1) = \phi(W_1'X_t + b_1)$; second, the ANN transmits the nonlinear function value of the linear combination $\hat{Y}_t = g(\sum_{j=1}^{n} W_{jo}h^j + b_o)$ of hidden nodes to the output node, which becomes a forecast value, where $\{W_{ij}, W_{jo}\}$, $\{b_1, b_o\}$ are real numbers called weights and biases. The nonlinear functions ϕ, g are called activation functions such as tanh. Let $\{B_1, \ldots, B_m\}$ be a random partition of $\{1, \ldots, T\}$ which correspond to sets of time points of a partition of the data having m batches $\{(X_t, Y_t), t \in B_j\}$, $j = 1, \ldots, m$. For example, if $T = 380$ and we choose batch size 20, we have $m = 19$ batches. The ANN gets learning from a sequence of batches $\{(X_t, Y_t), t \in B_j\}$, $j = 1, \ldots, m$ by repeatedly updating the weights W and biases b to minimize a measure, loss function such as mean squared error, of forecast error $Y_t - \hat{Y}_t$, $t \in B_j$, $j = 1, \ldots, m$. This learning procedure from a given partition $B_1, \ldots, B_m$ of $\{1, \ldots, T\}$ makes an epoch. The learning procedure is repeated with other random partitions of $\{1, \ldots, T\}$ many times. The number of repetition is called the number of epochs. In ANN, two or more hidden layers between input and output layers are added in order to address more complex non-linear relationship between predictors and forecast, which is called a DNN. In many cases, deeper NN shows better forecast performance, but not always.

However, DNN does not perform well for temporal data sets whose elements are serially correlated as are the most economic time series data. The RNN is proposed for temporally-structured data to address the serial dependence. In RNN, hidden node values have time-dependent nonlinear AR(1) structure. When we use X_t as a predictor at time t, the value h_t^1 of a hidden node, node 1 say, is given by

$$h_t^1 = \phi\left(W_1'X_t + U_1'H_{t-1} + b_1\right),$$

where ϕ is an activation function such as tanh, $H_{t-1} = (h_{t-1}^1, \ldots, h_{t-1}^n)'$ is the vector of hidden state values at time $t-1$. U_1 and W_1 are also vectors of weights and b_1 is a bias that will be updated by machine learning. For X_t, we will consider X_t^{MN}, ΔX_t^{MN}, ΔX_t^R, where X_t^{MN}, X_t^R are subvectors of X^{MN}, X^R, respectively, corresponding to time t and ΔX_t^{MN} and ΔX_t^R are differences of X_t^{MN} and X_t^R, respectively. The hidden layer in the RNN has the role of remembering previous information, but it cannot selectively remember the previous information. It makes all inputs at all times be remembered

with the same weight and makes the effect of remote inputs disappearing rapidly (gradient vanishing) or (exploding) like in stationary or explosive AR(1) models. LSTM of Hochreiter and Schmidhuber (1997) and GRU of Cho *et al.* (2014) resolves the gradient vanishing and exploding problems of RNN by adding the memory cell with (forget, input, and output) gates and (reset and update) gates, respectively. We consider DNN, LSTM, and GRU as the forecasting method of the U.S. employment level. See Section 4 for the implementations of the ANN methods.

3.2. Differencing

As discussed in Section 2, the FRED data sets include $I(1)$ or $I(2)$ series, which need to be differenced. In the forecast based on statistical methods, the degree of differencing is an important issue. In ARIMA forecasting, forecast (especially long-term) performance depends on the degree of differencing: long-term forecasts tend to be seriously biased toward the overall sample mean if a unit root of a nonstationary data is estimated rather than specified. We will demonstrate that proper differencing is more important in machine learning forecasting than in ARIMA forecasting: we have very bad machine learning forecast if nonstationary data are not differenced. In the machine learning methods, data-normalization is a crucial factor for good weight estimation in network learning, see Sola and Sevilla (1997). The normalization improves network convergence speed of neural network algorithm and avoids falling into local optimum by changing the range of values of all predictors to have a common scale without a large difference. However, since mean and variance are not defined for nonstationary series, normalization by sample mean and sample standard deviation would be unstable for nonstationary series, hence not working good in machine learning forecasting as will be demonstrated in Section 4.

3.3. Dimension reduction

We are interested in whether dimension reduction improves the machine learning forecast performance. Accordingly, we consider a type of linear regression, the least absolute shrinkage and selection operator (LASSO) regression which is widely used as a dimension reduction method, see for example Tarassow (2019), Uniejewski *et al.* (2019), Cepni and Swanson (2019) and many others for application in time series forecasting. Let T be the time series dimension of the data set. Dimension reduction is made for X^{MN} to X^R, say, in view of forecasting $Y_{T+h} - Y_T$ using ΔX_T^{MN}. Let Δ_h be the operator such that $\Delta_h Y_t = Y_t - Y_{t-h}$. Let $\Delta = \Delta_1$ be the difference operator. In LASSO regression, the LASSO coefficients for h-step ahead forecast $\Delta_h Y_{t+h} = Y_{t+h} - Y_t$ are obtained by minimizing the sum of squares of h-step forecast errors,

$$\sum_{t=1}^{T}\left(\Delta_h Y_{t+h} - \beta_0 - \sum_{i=1}^{K}\Delta^{d_i} X_{it}^{\mathrm{MN}}\beta_i\right)^2 \quad \text{subject to } \sum_{i=1}^{K}|\beta_i| \le \lambda,$$

where X_{it}^{MN} is the $(i,t)^{th}$ element of X^{MN} and $d_i \in \{0,1,2\}$ is the order of X_{it}^{MN} identified by McCracken and Ng (2016). The constraint $\sum_{i=1}^{K}|\beta_i| \le \lambda$ makes many coefficients of $\Delta X_{it}^{\mathrm{MN}}$ be zero reducing the dimension K of X^{MN} to a substantially smaller one (Table 1) below and preventing over-fitting of the regression. We will demonstrate that dimension reduction improves long-term forecast and some short-term forecast.

Table 1 shows predictors selected by LASSO regression for $h = 1, 3, 6, 12$ step forecasts. The predictors selected three or more times in each step are related to the number of employees in specific industries (PAYEMS, MANEMP, and USFIRE) and to the number of housing units authorized

Table 1: Predictors selected by LASSO

h-step	Variables						
1	PAYEMS	SRVPRD	CLF16OV	USWTRADE	DPCERA3M086SBEA		
3	PAYEMS	MANEMP	USFIRE	PERMITNE	HOUST	HOUSTMW	NDMANEMP
6	PAYEMS	MANEMP	USFIRE	PERMITNE	USGOOD	HOUSTMW	HOUSTNE
	TB3MS	T5YFFM	USWTRADE				
12	PAYEMS	MANEMP	USFIRE	PERMITNE	HOUST	HOUSTMW	CES1021000001
	EXSZUS	T10YFFM	USWTRADE	USTPU	USGOOD	UMCSENT	

Labor Market: PAYEMS = all employees for total nonfarm, SRVPRD = all employees for service-providing industries, CLF16OV = civilian labor force, NDMANEMP = all employees for nondurable goods, USWTRADE = all employees for wholesale trade, MANEMP = all employees for manufacturing, USFIRE = all employees for financial activities, USGOOD = all employees for goods-producing industries, CES1021000001 = all employees for mining, USTPU = all employees for trade, transportation and utilities; **Orders and inventories**: DPCERA3M086SBEA = real personal consumption expenditures, UMCSENT = consumer sentiment index; **Consumption and orders**: HOUST = housing starts for total new privately owned, PERMITNE = new private housing permits, northeast (SAAR), HOUSTMW = housing starts, midwest, HOUSTNE = housing starts, northeast; **Interest rate and exchange rates**: TB3MS = 3-month treasury bill, T5YFFM = 5-year treasury C minus fedfunds, EXSZUS = Switzerland/U.S. foreign exchange rate.

by building permits (PERMITNE and HOUSTMW). The table also shows that more predictors are selected for long-term forecast than for short-term forecast.

4. Out-of-sample forecast

Focusing on the role of differencing and dimension reduction, we make on out-of-sample forecast comparison of log employment level Y for some ANN methods and a benchmarking statistical method: DNN, LSTM, GRU discussed in Section 2 and AR model. We demonstrate that the ANN methods of LSTM and GRU have a better forecast performance than the benchmarking method of AR forecasting if statistical issues of differencing and dimension reduction are properly addressed. The benchmarking AR forecast is based on an AR(p) fitting to the differenced series ΔY_t

$$(1 - \phi_1 B - \cdots - \phi_p B^p)\Delta Y_t = a_t, \quad a_t \sim \left(0, \sigma^2\right), \tag{4.1}$$

where B is backshift operator and p is selected by the Bayesian information criterion (BIC).

Let T be the data length. Out-of-sample forecasts are computed from expanding window samples starting from $t_0 = 0.7T$. The h-step ahead forecasts $\hat{Y}_{t+h|t}$, $h = 1, 3, 6, 12$ are computed from the expanding window sample $\{(X_s, Y_s), s = 1, \ldots, t\}$ for $t = t_0, \ldots, T - h$, where X_t is the vector of predictors at time t. In AR(p) model, h-step out-of-sample forecast is recursively computed from (4.1). Machine learning forecasts are computed by the ANN methods of DNN, LSTM, and GRU using all non-differenced predictors $X_t = X_t^{\text{MN}}$, all differenced predictors $X_t = \Delta X_t^{\text{MN}}$ and dimension-reduced differenced predictors $X_t = \Delta X_t^R$. Dimension reduction is made for each time $t \in \{t_0, \ldots, T - h\}$ and for each $h = 1, 3, 6, 12$ by fitting LASSO regression of $\Delta_h Y_{t+h}$ on $\{\Delta X_s^{\text{MN}}, s = 1, \ldots, t\}$. The reduced predictors in Table 1 are for $t = T - h$. As measures of the forecast performance, we consider the root mean square error (RMSE) and the mean absolute error (MAE),

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{t=t_0}^{T-h}\left(Y_{t+h} - \hat{Y}_{t+h|t}\right)^2}, \qquad \text{MAE} = \frac{1}{M}\sum_{t=t_0}^{T-h}\left|Y_{t+h} - \hat{Y}_{t+h|t}\right|, \quad M = T - h - t_0 + 1.$$

For each forecast method, efficiency gain relative to the benchmarking AR forecast is compared. For

Table 2: The RMSE and the MAE efficiency gain (%) of the ANN forecasts relative to the AR forecasts and the Diebold-Mariano test results

h-step		AR(p)	ANN methods with 105 FRED predictors, X^{MN}						ANN methods with LASSO predictors		
			Non-differenced data, X^{MN}			Differenced data, ΔX^{MN}			Differenced data, ΔX^{R}		
			DNN	LSTM	GRU	DNN	LSTM	GRU	DNN	LSTM	GRU
1	RMSE	0.0023	**−97.3**	**−96.5**	**−96.3**	1.9	−2.4	4.0	5.3	0.0	2.1
	MAE	0.0018	**−96.5**	**−97.0**	**−97.0**	3.7	2.6	7.6	2.5	3.4	**6.0**
3	RMSE	0.0046	**−92.2**	**−92.9**	**−93.3**	−4.6	20.1	21.7	17.1	**16.9**	10.3
	MAE	0.0031	**−92.3**	**−94.8**	**−95.1**	−4.8	14.8	**17.1**	3.3	**13.9**	3.0
6	RMSE	0.0071	**−86.5**	**−89.3**	**−89.9**	25.5	25.9	58.8	−2.6	50.7	35.0
	MAE	0.0047	**−88.6**	**−92.7**	**−92.8**	16.3	**24.5**	**48.4**	−8.0	**41.8**	18.4
12	RMSE	0.0110	**−85.2**	**−84.6**	**−85.8**	36.7	47.2	38.7	40.3	60.0	50.5
	MAE	0.0071	**−88.0**	**−89.9**	**−90.5**	16.4	**36.4**	29.2	21.3	**45.6**	31.7

Bold type is significant at 10% level by the Diebold Mariano test. RMSE = root mean square error; MAE = mean absolute error; ANN = artificial neural network; AR = autoregressive; FRED = federal reserve economic data; LASSO = least absolute shrinkage and selection operator; DNN = deep neural network; LSTM = long short term memory; GRU = gated recurrent unit.

example, MAE efficiency gain of the DNN method relative to the AR method is

$$\text{DNN MAE efficiency gain (\%)} = \left(\frac{\text{MAE of the AR(p) forecast}}{\text{MAE of the DNN forecast}} - 1\right) \times 100.$$

It means better forecast performance of one forecast model than AR(p) model if its efficiency gain is greater than 0.

For the ANN forecasts described in Section 3.1, we need to specify the hyperparameters: learning rate, optimization method, loss function, the number of epochs (e), batch size (b), the number of hidden nodes (n), the number of hidden layers (l), dropout rate, weight regularization. The optimization method is chosen to be 'Adam' proposed by Kingma and Ba (2014), respectively, which are known to give a good weight estimate in learning network among several recent methods, see Ruder (2016). The loss function is set to mean squared error. The number of epochs, the number of repetitions for network learning, is set to 500. The learning rate, the fraction of the weights being updated during network learning, is set to 0.001 from the widely considered range $(0.0001, 0.1)$. In order to prevent overfitting in the ANN methods, we consider the most widely used 0.5 dropout rate and (L_1, L_2) weight regularization. The dropout rate is the fraction of nodes whose inbound and outbound weights are all randomly set to 0. The (L_1, L_2) weight regularization restricts the sums of $|W_{ij}|$, W_{ij}^2 to below given numbers. We also consider batch normalization proposed by Ioffe and Szegedy (2015), which has become an essential consideration for training acceleration and stable behavior of the gradients in ANN methods, see Cooijmans *et al.* (2017), Laurent *et al.* (2016), and Santurkar *et al.* (2018).

For each $h = 1, 3, 6, 12$, the batch size (b), the numbers of nodes (n), and layers (l) are selected by minimizing average of h-step out-of-sample forecast RMSE and MAE over the grid of $\{16, 32, 64\} \times \{8, 16, 32, 64\} \times \{1, 2, 3\}$ for LSTM, GRU and the grid of $\{16, 32, 64\} \times \{8, 16, 32, 64\} \times \{1, 2, 3, 6, 9\}$ for DNN. Determination of hyperparameters by grid search is commonly considered in the literature, see for example Kim and Baek (2019). Since Li *et al.* (2018) shows that the recurrent methods of LSTM and GRU with four or more layers gives usually bad performance, we consider the number of layers among $\{1, 2, 3\}$. For DNN, larger number of layers $\{1, 2, 3, 6, 9\}$ are considered because DNN is sometimes implemented with large l.

Table 2 shows the RMSE and the MAE efficiency gains of the ANN forecasts relative to the AR forecast. ANN forecasts based on differenced predictors gain efficiency over the AR forecast,

Table 3: The forecasting MCS performance: p-value (rank) of the MCS test

h-step		AR(p)	ANN methods with 105 FRED predictors, X^{MN}						ANN methods with LASSO predictors		
			Non-differenced data, X^{MN}			Differenced data, ΔX^{MN}			Differenced data, ΔX^{R}		
			DNN	LSTM	GRU	DNN	LSTM	GRU	DNN	LSTM	GRU
1	RMSE	0.98(5)	0.00	0.00	0.00	1.00(4)	0.94(7)	1.00(2)	1.00(1)	0.96(6)	1.00(3)
	MAE	0.28(7)	0.00	0.00	0.00	0.99(3)	0.91(5)	1.00(1)	0.96(4)	0.90(6)	1.00(2)
3	RMSE	0.34(6)	0.00	0.00	0.00	0.20(7)	1.00(2)	1.00(1)	1.00(3)	0.99(4)	0.49(5)
	MAE	0.12(5)	0.00	0.00	0.00	0.03(7)	1.00(2)	1.00(1)	0.05(6)	1.00(3)	0.18(4)
6	RMSE	0.40(4)	0.00	0.00	0.00	0.29(5)	0.84(3)	1.00(1)	0.00	0.97(2)	0.06(6)
	MAE	0.07(5)	0.00	0.00	0.00	0.14(4)	0.70(3)	1.00(1)	0.00	0.98(2)	0.00
12	RMSE	0.70(5)	0.00	0.00	0.00	0.12(7)	0.98(2)	0.71(4)	0.12(6)	1.00(1)	0.71(3)
	MAE	0.13(5)	0.00	0.00	0.00	0.11(6)	0.99(2)	0.70(3)	0.10(7)	1.00(1)	0.57(4)

MCS = model confidence set; ANN = artificial neural network; FRED = federal reserve economic data; LASSO = least absolute shrinkage and selection operator; AR = autoregressive; DNN = deep neural network; LSTM = long short term memory; GRU = gated recurrent unit; RMSE = root mean square error; MAE = mean absolute error.

while those based on non-differenced predictors lose substantial gains. We find that, for 1, 3, 6 step forecasts, the GRU method with all the 105 FRED differenced predictors ΔX^{MN} is the best; for 12 step forecast, LSTM with differenced LASSO predictors ΔX^{R} is the best. For all the three ANN methods, the dimension reduction improves forecast performances of the longer-term forecast, 12-step forecast, and of some shorter-term forecasts of $h = 1, 3, 6$. We also identify that the GRU and LSTM methods tend to perform substantially better than the DNN method. The reason is that the significant autocorrelation of ΔY_t depicted in Figure 1 is properly addressed by the recurrent structure of GRU and LSTM, but is neglected by DNN.

From the efficiency gain of the ANN forecasts based on the non-differenced data X^{MN}, we see the ANN forecasts based on X^{MN} are very worse than the AR forecasts as well as the ANN forecasts based on the differenced data ΔX^{MN}.

In the table, we check statistical significances of the efficiency gain by the test of Diebold and Mariano (1995). The DM test, for example for RMSE efficiency gain, is the t-test for the equality of the mean of the squared forecast error of an ANN method and that of AR(p) method in which serial correlations of forecast errors are addressed by the heteroscedasticity and autocorrelation consistent (HAC) standard error of the sample mean difference. The DM test shows that the LSTM and GRU methods have significantly better forecast performance than the AR method for some h. We also identify that the ANN methods based on non-differenced predictor X^{MN} has a significantly worse forecast performance than the AR forecast.

For more formal comparison, we make a MCS analysis of Hansen *et al.* (2011) at a given level of confidence $\alpha = 0.05$. The MCS is a set of one or more models with a good forecast performance and is constructed by multiple comparison under the assumption that there is no true model. The MCS provides forecasting performance rank and p-value, the latter of which is the probability of the model being contained in the MCS. Table 3 reports the result of MCS analysis. The table shows a results similar to that of Table 2. The GRU method with all 105 differenced predictors ΔX^{MN} has p-value of 1.00 and is mostly ranked 1 for all short-term forecast, 1, 3, 6 steps. The LSTM model with dimension reduced differenced predictors ΔX^{R} has p-value 1.00 and is ranked 1 for long-term forecast, 12 steps. ANN methods based on non-differenced data X^{MN} has p-values close to zero, indicating poor forecast performance. Therefore, the MCS analysis shows that, for $h = 1, 3, 6$ forecasts, GRU with ΔX^{MN} is the best and, for $h = 12$ forecast, LSTM with ΔX^{R} is best.

5. Conclusion

In forecasting the U.S. employment level, machine learning methods of DNN, LSTM, and GRU are considered. The predictors are chosen to be the 105 important macroeconomic variables selected by McCracken and Ng (2016) among the big data of the FRED. We consider the two statistical issues of dimension reduction and time series differencing in the machine learning forecast. An out-of-sample comparison shows substantial efficiency gain for the machine learning forecasts over the AR forecast if proper differencing is considered. The comparison reveals that, for $h = 1, 3, 6$ step forecasts, the GRU method with all the 105 FRED differenced predictors is the best and, for 12 step forecast, LSTM with differenced and dimension reduced predictors is the best. We also find that the dimension reduction improves long-term forecast of $h = 12$ and some short-term forecast of $h = 1, 3, 6$.

Acknowledgements

This study was supported by a grant from the National Research Foundation of Korea (2019R1A2C10 04679), by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2019R1A6A1A11051177) and by the Ewha Womans University Scholarship of 2017.

References

Arevalo A, Nino J, Hernandez G, and Sandoval J (2016). High-frequency trading strategy based on deep neural networks, Intelligent Computing Methodologies. ICIC 2016, 9773, Springer, Cham.

Cepni O and Swanson NR (2019). Nowcasting and forecasting GDP in emerging markets using global financial and macroeconomic diffusion indexes, *International Journal of Forecasting*, **35**, 555–572.

Chiang WC, Enke D, Wu T, and Wang R (2016). An adaptive stock index trading decision support system, *Expert Systems with Applications*, **59**, 195–207.

Cho K, Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, and Bengio Y (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, 1724–1734.

Chong E, Han C, and Park FC (2017). Deep learning networks for stock market analysis and prediction: methodology, data representations, and case studies, *Expert Systems With Applications*, **83**, 187–205.

Cooijmans T, Ballas N, Laurent C, Gulcehre C, and Courville A (2017). Recurrent batch normalization, arXiv preprint, arXiv: 1603.09025.

Diebold FX and Mariano RS (1995). Comparing predictive accuracy, *Journal of Business and Economic Statistics*, **13**, 253–263.

Hansen PR, Lunde A, and Nason JM (2011). The model confidence set, *Econometrica*, **79**, 453–497.

Hochreiter S and Schmidhuber J (1997). Long short-term memory, *Neural Computation*, **9**, 1735–1780.

Ioffe S and Szegedy C (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift, arXiv preprint, arXiv: 1502.03167.

Kim J and Baek C (2019). Bivariate long range dependent time series forecasting using deep learning, *The Korean Journal of Applied Statistics*, **32**, 69–81.

Kingma DP and Ba J (2014). Adam: a method for stochastic optimization, arXiv preprint, arXiv: 1412.6980.

Laurent C, Pereyra G, Brakel P, Zhang Y, and Bengio Y (2016). Batch normalized recurrent neural networks. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, 2657–2661.

Lehmann R and Weyh A (2016). Forecasting employment in Europe: are survey results helpful?, *Journal of Business Cycle Research*, **12**, 81–117.

Li S, Li W, Cook C, Zhu C, and Gao Y (2018). Independently recurrent neural network (IndRNN): building a longer and deeper RNN, arXiv preprint, arXiv: 1803.04831.

McCracken MW and Ng S (2016). FRED-MD: a monthly database for macroeconomic research, *Journal of Business and Economic Statistics*, **34**, 574–589.

Qiu M, Song Y, and Akagi F (2016). Application of artificial neural network for the prediction of stock market returns: the case of the Japanese stock market, *Chaos, Solitons and Fractals*, **85**, 1–7.

Rapach DE and Strauss JK (2010). Bagging or combining (or both)? an analysis based on forecasting U.S. employment growth, *Econometric Reviews*, **29**, 511–533.

Rapach DE and Strauss JK (2012). Forecasting US state-level employment growth: an amalgamation approach, *International Journal of Forecasting*, **28**, 315–327.

Ruder S (2016). An overview of gradient descent optimization algorithms, arXiv preprint, arXiv:1600.04747.

Santurkar S, Tsipras D, Ilyas A, and Madry A (2018). How does batch normalization help optimization (no, it is not about internal covariate shift), arXiv preprint, arXiv: 1805.11604.

Siliverstovs B (2013). Do business tendency surveys help in forecasting employment? A real-time evidence for Switzerland, *OECD Journal: Journal of Business Cycle Measurement and Analysis*, 2013/2.

Sola J and Sevilla J (1997). Importance of input data normalization for the application of neural networks to complex industrial problems, *IEEE Transactions on Nuclear Science*, **44**, 1464–1468.

Stock JH and Watson MW (1996). Evidence on structural instability in macroeconomic time series relations, *Journal of Business and Economic Statistics*, **14**, 11–30.

Tarassow A (2019). Forecasting U.S. money growth using economic uncertainty measures and regularisation techniques, *International Journal of Forecasting*, **35**, 443–457.

Uniejewski B, Marcjasz G, and Weron R (2019). Understanding intraday electricity markets: Variable selection and very short-term price forecasting using LASSO, published online, doi.org/10.1016/j.ijforecast.2019.02.001

Received June 27, 2019; Revised August 23, 2019; Accepted August 26, 2019