Applied Economics Letters

ISSN: 1350-4851 (Print) 1466-4291 (Online) Journal homepage: https://www.tandfonline.com/loi/rael20

Can machine learning on economic data better forecast the unemployment rate?

Aaron Kreiner & John V. Duca

To cite this article: Aaron Kreiner & John V. Duca (2019): Can machine learning on economic data better forecast the unemployment rate?, Applied Economics Letters, DOI: 10.1080/13504851.2019.1688237

To link to this article: https://doi.org/10.1080/13504851.2019.1688237

Published online: 11 Nov 2019.

APPLIED ECONOMICS LETTERS
https://doi.org/10.1080/13504851.2019.1688237

Routledge
Taylor & Francis Group

ARTICLE

Can machine learning on economic data better forecast the unemployment rate?

Aaron Kreiner[a] and John V. Duca[b,c]

[a]Global Markets Division, Nomura Securities, New York, NY, USA; [b]Economics Department, Oberlin College, Oberlin, OH, USA; [c]Research Department, Federal Reserve Bank of Dallas, Dallas, TX, USA

ABSTRACT
Using FRED data, a machine-learning model outperforms the Survey of Professional Forecasters and other models since 2001 in forecasting the unemployment rate.

KEYWORDS
Machine learning; forecasting; artificial intelligence; unemployment

JEL CLASSIFICATION
C10; C45; E24

I. Introduction

While some conventional models have outperformed the mean response from the Survey of Professional Forecasters (SPF) in forecasting the unemployment rate since Montgomery et al. (1998) (e.g. Barnichon and Nekarda 2012; Meyer and Tasci 2015), few have used machine learning. Exceptions are Two Sigma (2016), who use taxi data to forecast the New York City unemployment rate, and Xu, Li, and Chen (2013), who use Google searches and slightly outperform the mean SPF forecast. Cook and Hall (2017) use neural networks on one indicator, unemployment lags, and out-forecast the SPF over short horizons. We outperform the SPF over one-, two- and four-quarter horizons, following Cook and Hall (2017) and Xu, Li, and Chen (2013) in using neural networks, but use machine learning to select data from over 600,000 variables in the FRED database, structured as a recursive tree with data binned into eight categories (Table 2).

II. Constructing a neural network model

Our algorithm recursively calls this tree to acquire and clean data. Variables must be non-forecast data that are monthly or quarterly (higher frequency is converted into monthly), not reported as lags or leads, and continuously available over 1970:01–-2018:12. Values are from a variable's first revision and include the current and four lags of unemployment. Under 2% of FRED variables are culled, with 'international' variables the most numerous and none from the 'academic' category. To these, we apply a principal components analysis (PCA) algorithm with a variance threshold of 0.99 yielding a dataset with fewer variables (185 linearly independent data columns) to reduce over-fitting.

The principal components are inputted into an artificial neuron network (ANN) in which the neurons perform three functions: propagation, activation, and output (Kriesel 2009). Our PCA is a pre-processing phase reducing the dimensionality of initial inputs. These principal components enter our ANN as inputs (t-1 to t-4) received by neurons in the first propagation phase, which neurons transform into intermediate signals in an activation phase using computational methods to convert the inputs into more usable intermediate signals. In the output phase, neurons produce four-quarter ahead forecasts of unemployment. Since the latter are continuous, positive real numbers, we can use a multilayer perceptron (MLP) regression in the activation function.

Networks are comprised of neurons, which contain propagation, activation, and output functions. Neurons are grouped into 'hidden layers' transform inputs into outputs. Each layer produces net output (net_j) from a member neuron j that receives outputs $o_{i_1}, o_{i_2} \ldots, o_{i_n}$ from neurons $i_1, i_2, \ldots, i_n$. Neuron j's propagation function transforms the o_i into a weighted sum of the outputs by calculating

CONTACT John V. Duca jduca@oberlin.edu Economics Department, Oberlin College, Oberlin, OH, USA

weights (w_{ij}) to determine the slope of the parameters of the activation function. It also estimates an associated bias ($bias_j$) or intercept of the activation function. These yield the optimal output (net_j, a number with singular dimension) at each network layer:

$$net_j = bias_j + \sum_{i \in I} o_i * w_{i,j} \tag{1}$$

The propagation function thus reduces the dimensionality of the inputs of neuron j from n to 1.

An optimization procedure minimizes the sum of in-sample squared errors. Assume there are p pairings of inputs and outputs. Let x(i) be a n-dimensional vector for the i-th input into a neuron x, d(i) be a singleton for the output of neuron x(i), w be the unknown weight matrix of the ANN and b be the unknown bias of the ANN. If $y(x:\ w;\ b)$ denotes the predicted output of x conditional on w and b, minimizing the squared error of the predicted output implies the objective function:

$$\min \sum_{i=1}^{p} \|y(x(i) : w; b) - d(i)\|^2 \tag{2}$$

The y function depends on the activation of the neurons and the numbers of hidden layers and inputs. This process generates weights and biases from the training data.

Several choices were made in constructing an optimal ANN that minimize the root mean squared error for predicting the unemployment rate four-quarters ahead. The size and number of hidden layers are chosen from tests that jointly evaluated four computational methods used by the neurons from activation functions in Python's ANN library: *Identity, Logistic, Tanh*, and *Relu*. One, two, and three hidden layers were tested. More were not tested due to computational complexity and runtime. The size (number of neurons) of each hidden layer was varied between 1 and 150. More layers were not tested because the RMSE was at a minimum for activation at about 120 neurons for each number of hidden layer sizes. For a four-quarter ahead forecast, the best configuration minimized the RMSE at 0.20 over the longest training sample using three hidden layers and 97 neurons per layer, with a size of 120 neutrons that used Python's logistic activation function. These hyper-parameters are fixed as different rolling forecasts are made.

The first ANN forecast uses 1970:q1-2000:q3 PCA data to train the model – estimate the weights, biases, and other metrics – to forecast the unemployment rate in 2001:q3. Each subsequent forecast repeats this process (including the PCA algorithm).[1] We tested a nonrolling window variant that fixes the beginning of training at 1970:q1 and sequentially extends the end-of-training period by one quarter against a variant, which roles the start and end training points forward by one quarter. We select the non-rolling window whose RMSE was lower, implying that old business cycles are informative, consistent with Montgomery et. al.'s (1998) finding that including more data lags increased forecast accuracy.[2]

III. Lasso regression

As an alternative, we construct a Lasso model, which uses variable selection and regularization to maximize prediction accuracy and coefficient interpretation. While this model uses the same overall structure as our ANN and benefits from reducing raw data with PCA (Fonti 2017), the only hyper-parameter tested is $0 \le \lambda \le 1$, which shrinks the number of beta coefficients, reducing over-fitting. For four-quarter ahead forecasts from 2001–2018, $\lambda = 0.15$ minimizes the RMSE at 0.31.

IV. Results

We compare four-quarter ahead unemployment rate forecasts from the ANN and Lasso models with the mean SPF forecast and a naïve forecast equal to the sum of the current unemployment rate plus its change over the prior four quarters. The Lasso and ANN models use 185 principal components derived from FRED and SPF data through 2018:12 and retrieved on 7 January 2019.

The naive forecast severely underperforms, having five and eight times the RMSE of the Lasso and

[1]For each data row, we include four observations of each variable from the current date to a year before.
[2]For one- and two-quarter ahead forecasts, the RMSE was also minimized using non-rolling forecasts, and a logistic activation function with three hidden layers.

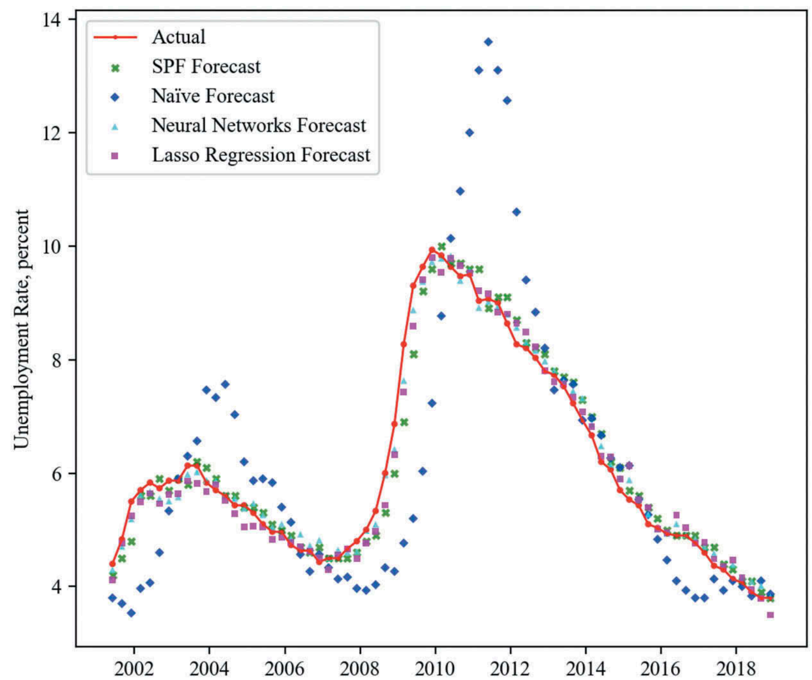

Figure 1. Four-quarter ahead forecasts of the unemployment rate 2001:q3-2018:q4.
Sources: FRED and authors' calculations.

ANN models over 2001–18, respectively, with similar results for 2001–2007 and 2007–2012 (Figure 1 and Table 1). This reflects that the SPF, Lasso, and ANN forecasts capture information from other variables. Importantly, the ANN model performs best in each period, particularly 2007–2012, which spans the Great Recession. Similar qualitative rankings obtain for one and two-quarter ahead forecasts (Table 1 reports 2001–18 to preserve space), with neural networks outperforming more at the four-quarter horizon. The relative performance of the Lasso model is unchanged if continuously available FRED variables replace principal components, except that the latter version of Lasso outperforms one- and two-quarters ahead.

Table 2 reports which types of variables are most informative at a four-quarter horizon, showing how much the RMSE rose above that of the full baseline model when each category is dropped. The biggest loss over the full and 2001–06 samples occurs if international data are excluded, consistent with globalization affecting US labour markets, especially after China entered the WTO in the early 2000s (see Autor, Dorn, and Hanson 2016). Nevertheless, the marginal information of international data has fallen since the late-2000s, consistent with slower pace of globalization. The second largest loss for the full and 2001–06 periods and the highest over 2007–2012 comes from omitting labour-demographic variables. The third largest loss in the 2001–18, 2001–2006 and 2007–2012 samples arises from dropping financial data, likely reflecting the role of asset prices in the recessions of 2001 and 2007–09.[3]

Table 1. Forecast errors.

s	4Q Ahead				1Q ahead	2Q ahead
Source	RMSE 2001–18	RMSE 2001–06	RMSE 2007–12	RMSE 2013–18	RMSE 2001–18	RMSE 2001–18
SPF	0.35	0.23	0.51	0.23	0.15	0.25
Naive	1.60	1.10	2.50	0.45	0.23	0.74
Neural Networks	0.20	0.19	0.23	0.18	0.12	0.16
Lasso, PCA	0.31	0.20	0.34	0.37	0.13	0.21
Lasso all variables	0.31	0.20	0.34	0.37	0.17	0.23

Sources: FRED and authors' calculations.

[3]The three most informative categories for the full sample were international, production and prices for two-quarter ahead forecasts, and international, prices, and labour for one-quarter ahead forecasts.

Table 2. Impact on forecast RMSE of omitting a variable category versus baseline.

Category Omitted	RMSE 2001–18 vs. Baseline	RMSE 2001–06 vs. Baseline	RMSE 2007–12 vs. Baseline	RMSE 2013–18 vs. Baseline
Academic	N/A	N/A	N/A	N/A
International	0.09	0.12	0.10	0.04
Money, Banking, Finance	0.05	0.05	0.08	0.01
National Accounts	0.04	0.04	0.04	0.05
Population, Employment, Labour	0.06	0.07	0.08	0.02
Prices	0.03	0.00	0.06	0.03
Production and Business Activity	0.03	0.00	0.08	−0.03
Regional	0.03	0.03	0.04	0.01

Sources: FRED, authors' calculations. 'vs. baseline': RMSE of model omitting listed category minus RMSE from baseline.

The non-PCA Lasso can assess which individual variables are most important. For four-quarter ahead forecasts, the 10 most informative, in order, were: housing starts, the change in nonfinancial corporate commercial mortgages, unfilled German job vacancies, retail vehicle registrations, weekly aggregate payrolls, household net lending, 3-month Australian Treasury rates, and producer prices for pharmaceuticals and tractors.[4] The top two are real estate variables, but are classified into different FRED categories, which reflects a limitation of assessing FRED categories. On the other hand, the marginal impact of a variable in an ANN versus a Lasso model can differ reflecting the nonlinear nature of the ANN and the degree of multicollinearity across variables.

V. Conclusion

For forecasting the unemployment rate, our neural network/machine learning model outperforms the SPF, as does a Lasso approach to a lesser extent. Our method can be efficiently implemented using an SQL database to update data and produce forecasts in minutes.

Acknowledgments

We thank an anonymous referee for suggestions. Views and errors are those of the authors and are not necessarily those of the Federal Reserve Bank of Dallas or the Federal Reserve System.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Autor, D. H., D. Dorn, and G. H. Hanson. 2016. "The China Shock: Learning from Labor Market Adjustment to Large Changes in Trade." *Annual Review of Economics* 8 (1): 205–240. doi:10.1146/annurev-economics-080315-015041.

Barnichon, R., and C. Nekarda. 2012. "The Ins and Outs of Forecasting Unemployment: Using Labor Force Flows to Forecast the Labor Market." *Brookings Papers on Economic Activity* 2: 83–117. doi:10.1353/eca.2012.0018.

Cook, T., and A. Hall. 2017. "Macroeconomic Indicator Forecasting with Deep Neural Networks." Research Working Paper RWP 17–11, Federal Reserve Bank of Kansas City.

Fonti, V. 2017. "Feature Selection Using LASSO - VU." Manuscript, University of Amsterdam. beta.vu.nl/nl/Images/werkstuk-fonti_tcm235-836234.pdf

Kriesel, D. 2009. *A Brief Introduction to Neural Networks*. Self-Published.

Meyer, B., and M. Tasci. 2015. "Lessons for Forecasting Unemployment in the U.S.: Use Flow Rates, Mind the Trend." Working Paper 1502, Federal Reserve Bank of Cleveland.

Montgomery, A. L., V. Zarnowitz, R. S. Tsay, and G. C. Tiao. 1998. "Forecasting the U.S. Unemployment Rate." *Journal of the American Statistical Association* 93: 478–493. doi:10.1080/01621459.1998.10473696.

Two Sigma. 2016. *Predicting New York City Unemployment Rate Using Taxi Cab Data*.

Xu, W., Z. Li, and Q. Chen. 2013. "Forecasting the Unemployment Rate by Neural Networks Using Search Engine Query Data." *Service Oriented Computing Applications* 7 (1): 33–42. doi:10.1007/s11761-012-0122-2.

[4]These were in the top-ten for one- and two-quarter-ahead forecasts, with slightly different rankings from four to ten.