Predicting Inflation with Neural Networks

Livia Paranhos

Department of Economics, University of Warwick

Highlights

- This paper applies neural networks to predict US CPI inflation, and in particular a **recurrent neural network**
- Neural nets present better performance than usual benchmarks, especially at the **one and two-year forecast**
- Recurrent neural nets are **at least as good as** the traditional feed forward neural net at medium-long horizons
- **Macroeconomic information** is important during periods of high uncertainty
- The paper also addresses the impact of the **stochastic initialization** of parameters on forecasting performance

Econometric framework

Consider two sets of predictive variables:
$\mathbf{x_t} = (x_{1t}, ..., x_{Nt})'$: pool of economic predictors
$\mathbf{y_t} = (y_{1t}, ..., y_{Mt})'$: CPI and its components

Let $\mathbf{z_t^L}$ be the set collecting the current and lagged values of $\mathbf{z_t} = \mathbf{x_t}, \mathbf{y_t}$ or $(\mathbf{x_t}, \mathbf{y_t})'$

I suppose that inflation, $y_t \in \mathbb{R}$, evolves nonlinearly wrt $\mathbf{z_t^L}$ through a function G

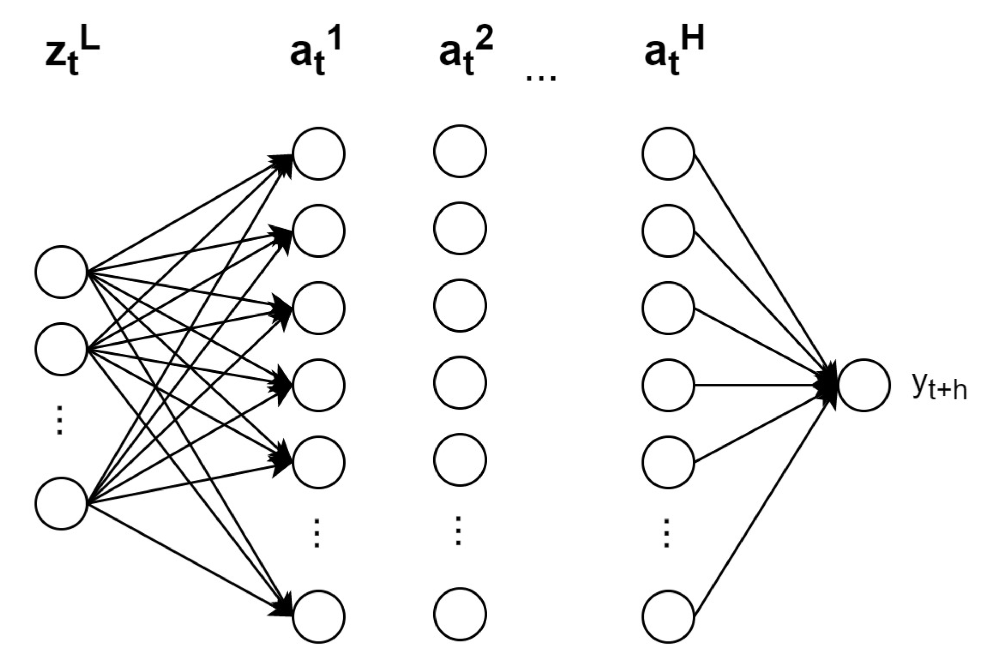
$$y_{t+h} = G(\mathbf{z_t^L}; \Theta_h) + \varepsilon_{t+h}$$

Fitting the unknown function $G : \mathbf{z_t^L} \to y_{t+h}$ to the data corresponds to estimating Θ_h given a network architecture, $\mathcal{A}_G$, by minimizing

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\left(y_{t+h} - G(\mathbf{z_t^L}; \Theta_h)\right)^2$$

- $\mathcal{A}_G$: neural net model & tunning parameters
- Universal approximation theorem (Cybenko, 1989): simple neural net model can approximate any continuous function up to an arbitrary degree of accuracy

The Feed Forward Network



Optimal tunning parameters

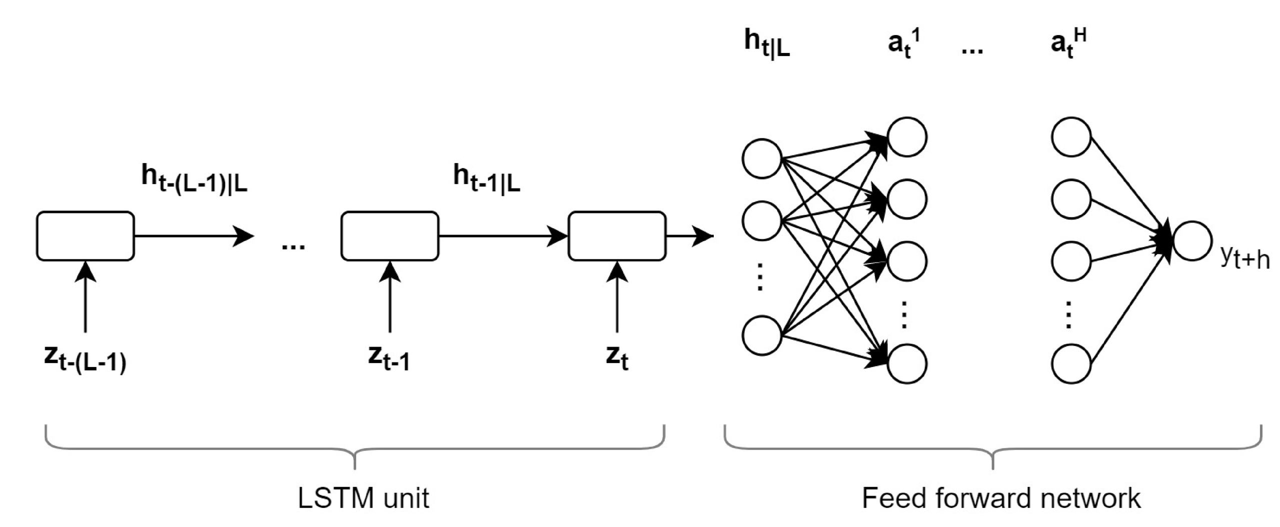
	lags	nodes	layers	factors	# par.
FF-cpi	24	128	4	n/a	80K
FF-pool	48	128	3	n/a	760K
LSTM-pool	48	128	4	2	50K
LSTM-all	48	128	4	2	50K
FF-LSTM	24,48	128	4	2	80K

Out-of-sample performance

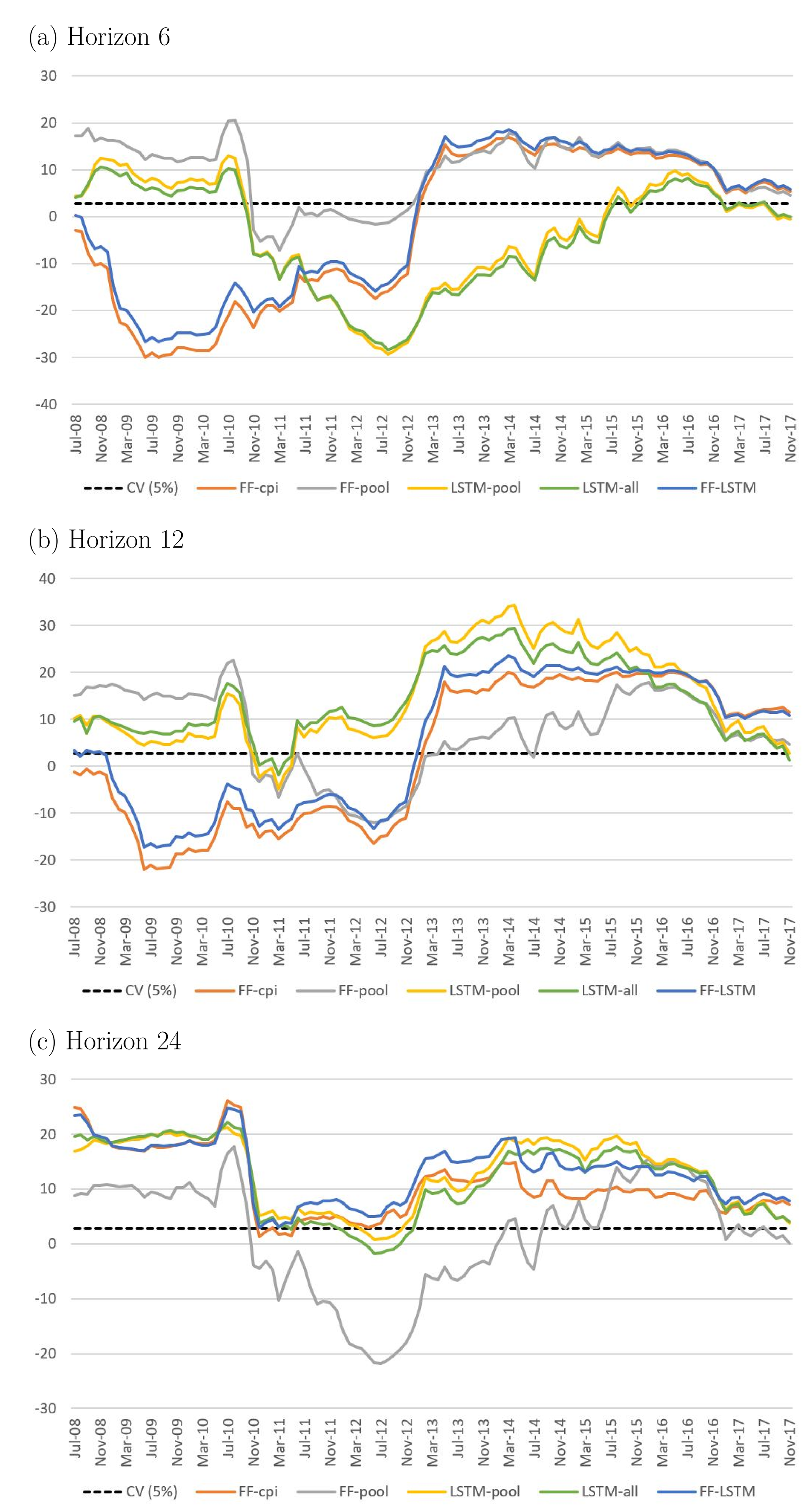
		Horizon (months)				
Model	Data	1	3	6	12	24
UCSV	CPI	1.13	1.05	1.03	1.02	1.00
FADL	Pool	1.05	1.09	1.08	1.01	1.00
FF-cpi	CPI	1.07	1.01	1.01	0.98	0.91
FF-pool	Pool	1.09	**0.92**	**0.90**	0.94	0.99
LSTM-pool	Pool	1.00	0.93	1.03	0.93	0.92
LSTM-all	All	**0.98**	0.94	1.04	**0.92**	0.91
FF-LSTM	All	1.06	0.99	0.99	0.97	**0.89**

RMSE Loss ratios wrt AR(1)

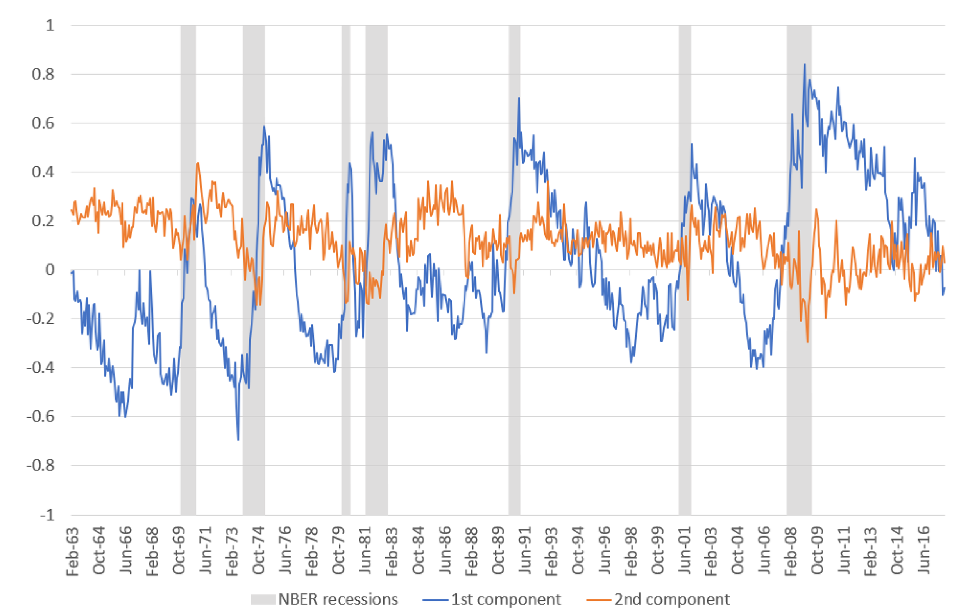
The Long-short Term Memory (LSTM) model



RMSE over time



LSTM hidden states



Initialization of parameters

- Random initialization (Glorot and Bengio, 2010, He et al., 2015)
- Non-convexity increases the sensitivity of the algorithm to initial values
- Empirical solution. Repeat estimation large number of times (here 1400) and average the predictions
- Counterfactual analysis. Compare errors of initial value with better performance and empirical solution adopted

	Horizons (months)				
Model	1	3	6	12	24
FF-cpi	0.96	0.97	0.95	0.98	0.97
FF-pool	0.96	0.96	0.95	0.93	0.89
LSTM-pool	0.97	0.93	0.89	0.96	0.93
LSTM-all	0.91	0.99	0.86	0.99	0.96
FF-LSTM	0.96	0.97	0.97	0.96	0.94

References

Cybenko, G. (1989). "Approximation by superposition of a sigmoidal function." Mathematics of Control, Signals and Systems 2, 303–314.

Glorot, X., and Y. Bengio. (2010). "Understanding the difficulty of training deep feedforward neural networks." Journal of Machine Learning Research - Proceedings Track 9, 249–256.

He, K., X. Zhang, Ren S., and J. Sun. (2015). "Delving deep into rectifiers: surpassing human-level performance on imagenet classification." In "Proceedings of the IEEE international conference on computer vision.", 1026–1034.