Forecasting Dynamic Term Structure Models with Autoencoders

Carlos Castro-Iragorri [1] Julian Ramirez [1]

[1]Universidad del Rosario

Autoencoders

Autoencoders are a particular type of neural network with the same input and output variables. If there are fewer neurons $\rightarrow$ dimension reduction.

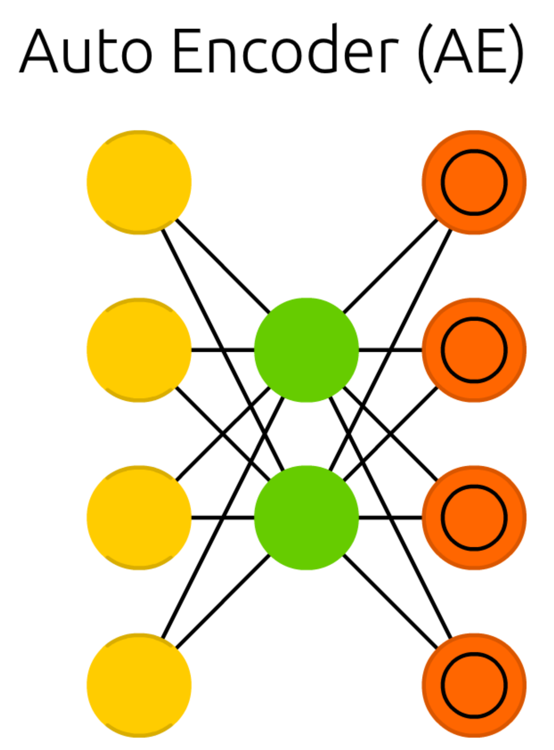
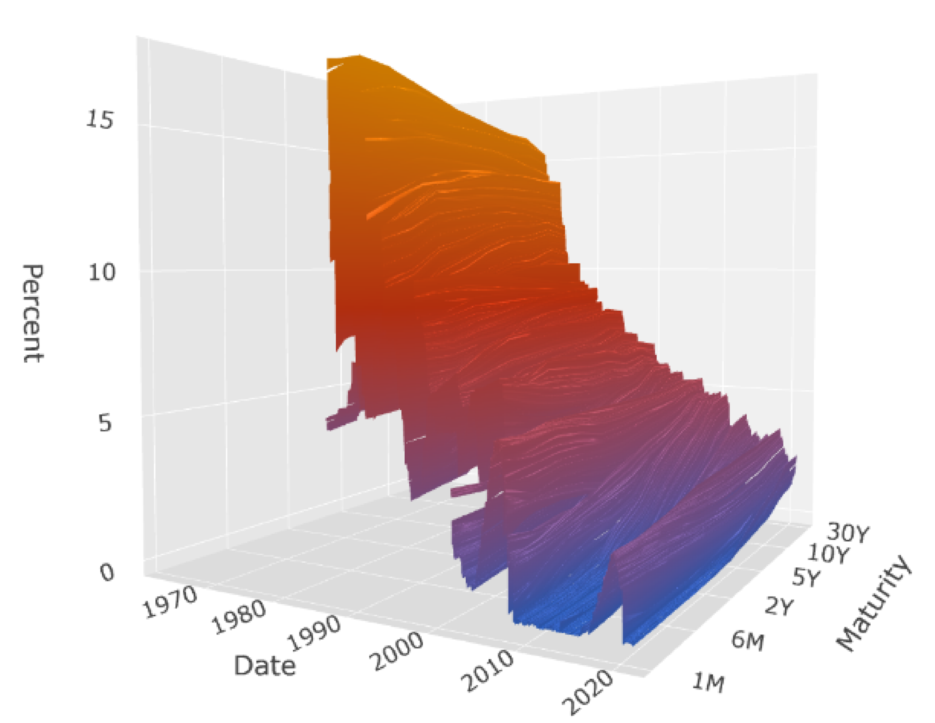


Figure 1. Simple autoencoder (Van Veen and Leijnen, 2019)

$$Y_t(\tau) = W^{(2)} Z_t + b^{(2)}$$
$$Z_t = \sigma(W^{(1)} Y_t(\tau) + b^{(1)})$$

Linear and non linear activation functions, σ. Principal component analysis (PCA) is a particular case of a linear autoencoder. The neurons have no natural ordering and are not ortogonal. However, they contain all of the information vs the selected PCAs.

US historical synthetic yields



- 16 maturities
- monthly data November 1985 to December 2020 (400 observations)
- source: CRSP, US Treasury and Fed data.

Dynamic term structure models

State-space representation (measurement and state transition equation),

$$y_{t+1}(\tau) = F_t(\tau) B_t + \varepsilon_{t+1}(\tau)$$
$$B_t = \Phi B_{t-1} + v_t$$

where $F_t(\tau)$ predetermined or time invariant ($F_t(\tau) = F(\tau) \forall t$) factor loadings and B_t are lower dimensional $K << \tau$ time varying factors.
Measurement equations

- NS3: Nelson-Siegel 3 factor model

$$y_{t+1}(\tau) = \beta_{1,t} 1 + \beta_{2,t}\left(\frac{1 - e^{-\lambda_t \tau}}{\lambda_t \tau}\right) + \beta_{3,t}\left(\frac{1 - e^{-\lambda_t \tau}}{\lambda_t \tau} - e^{-\lambda_t \tau}\right) + \varepsilon_{t+1}(\tau)$$

  where $B_t = (\beta_{1,t}, \beta_{2,t}, \beta_{3,t})$.
- PCA 3 factor model

$$y_{t+1}(\tau) \approx Z_{1,t} V_1(\tau) + Z_{2,t} V_2(\tau) + Z_{3,t} V_3(\tau) + \varepsilon_{t+1}(\tau)$$

  where $B_t = (Z_{1,t}, Z_{2,t}, Z_{3,t})$.

Autoencoders with 3 neurons and one or more layers

$$y_{t+1}(\tau) = Z_{1,t} W_1^{(2)}(\tau) + Z_{2,t} W_2^{(2)}(\tau) + Z_{3,t} W_3^{(2)}(\tau) + b^{(2)} \mathbf{1}(\tau) + \varepsilon_{t+1}(\tau)$$
$$Z_{i,t} = \tanh(W_i^{(1)} y_t(\tau) + b_i^{(1)})$$

where $B_t = (Z_{1,t}, Z_{2,t}, Z_{3,t})$ and the factor loadings are given by the weight matrices $W_i^{(2)}$ of of each of the neurons at the dencoder level (2).

- LA3: three neurons, one layer, linear activation.
- NA3: three neurons, one layer, non-linear activation.
- DA3: three core-neurons, three layers (6 and 8 non-core neurons), non-linear activation.

Out-of-sample performance, Horizon 1 month

τ	RW	NS3	PCA	LA3	NA3	DA3	ERNN_{20}	GRNN_{20}	LSTM_{20}
0.1	25.63	28.28	22.25	22.65	36.44	24.25	173.52	171.96	171.96
0.2	17.28	19.25	17.33	17.71	40.66	19.66	185.93	194.31	182.31
0.3	16.69	18.80	17.02	17.44	48.78	19.12	191.14	200.68	185.58
0.4	16.29	19.17	16.89	17.09	55.09	18.75	193.32	208.67	192.57
0.5	17.79	20.47	17.56	18.14	59.96	19.61	191.83	208.40	192.61
1.0	21.57	42.23	24.42	25.30	97.89	26.10	244.05	273.30	246.16
2.0	24.12	31.61	27.95	28.85	126.31	29.23	257.33	281.97	256.88
3.0	25.63	28.34	27.69	28.19	131.61	29.17	253.63	266.66	253.84
4.0	26.84	29.96	28.41	28.83	136.09	29.92	261.19	275.55	264.36
5.0	26.52	31.74	27.66	27.95	139.56	29.53	260.69	268.47	257.13
7.0	26.28	30.49	27.23	27.17	140.50	29.95	260.56	272.74	257.86
10.0	25.55	33.35	26.04	26.23	146.72	29.14	255.46	255.79	247.35
15.0	24.10	26.19	26.89	27.31	131.67	31.39	255.57	291.27	253.35
20.0	23.54	25.74	26.46	27.16	131.03	29.92	257.90	242.50	254.61
25.0	22.27	24.94	25.37	26.18	128.60	31.32	256.35	275.83	256.27
30.0	22.47	27.58	24.83	25.76	133.91	31.15	236.37	232.55	235.30

10 basis points is 0.10%. RW random walk benchmark

Benchmarks: Recurent Neural Networks (RNN) and Time Series Forecasting

- Elman recurrent unit, ERNN(1).

$$y_{t+1}(\tau) = Z_t W^{(2)}(\tau) + b^{(2)} \mathbf{1}(\tau) + \varepsilon_{t+1}(\tau)$$
$$Z_t = \tanh(y_t(\tau) W_y^{(1)} + Z_{t-1} W_z^{(1)} + b^{(1)})$$

- Gated recurrent unit, GRNN(1).

$$y_{t+1}(\tau) = Z_t W^{(2)}(\tau) + b^{(2)} \mathbf{1}(\tau) + \varepsilon_{t+1}(\tau)$$
$$Z_t = u_t \circ \tilde{Z}_t + (1 - u_t) \circ Z_{t-1}$$
$$\tilde{Z}_t = \tanh(y_t(\tau) W_{z,y}^{(1)} + Z_{t-1} W_{z,z}^{(1)} r_t + \mathbf{1} b_z^{(1)})$$
$$r_t = \sigma(W_{r,y}^{(1)} y_t(\tau) + W_{r,z}^{(1)} Z_{t-1} + \mathbf{1} b_r^{(1)})$$
$$u_t = \sigma(W_{u,y}^{(1)} y_t(\tau) + W_{u,z}^{(1)} Z_{t-1} + \mathbf{1} b_u^{(1)})$$

- Long Short Term Memory (LSTM) with or whithout peephole.

RNN can provide a dynamic system (for forecasting) that is deterministic (weights and bias are t-measurable) and with non-linear activation functions. These are analogous to classical time series models with a state space representation fore forecasting, for example an additive exponential moving average that can be represented as a linear non-deterministic dynamic system with a single source of error (Hyndman et al 2008).

Conclusions and work ahead

- PCA and simple linear autoencoder outperform Nelson and Siegel and more complex models: non-linear or deep autoencoder.
- Recurrent neural network models: Elman recurrent unit, Gated recurrent unit, LSTM, undeperform.
- Too few data (right trade off to the number of parameters) and structural changes.

References

- Shihao Gu, Bryan T. Kelly, and Dacheng Xiu. Autoencoder asset pricing models. Journal of Econometrics, 2020.
- Francis Diebold and Canlin Li. Forecasting the term structure of government bond yields. J. Econometrics, 2006.
- Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. International Journal of Forecasting, 37:388-427, 2021.
- Yoshiyuki Suimon, Hiroki Sakaji, Kiyoshi Izumi, and Hiroyasu Matsushima. Autoencoder-based three-factor model for the yield curve of japanese government bonds and a trading strategy. 13:82, 2020.
- P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. Neural Networks, 2:53-58, 1989.
- R.J. Hyndman, Koehler A.B., J.K. Ord, and R.D. Snyder. Forecasting with exponential smoothing: The state space approach. Springer, 2008.

carlos.castro@urosario.edu.co