Forecasting Dynamic Term Structure Models with Auto Encoders

Carlos Castro and Julian Ramirez

Universidad del Rosario

July 5, 2021

Highlights

Autoencoders

- A type of neural network that by design provides dimension reduction.
- PCA is a particular case of a linear autoencoder, Baldi and Hornik (1989).
- Recently applied for financial factor models Gu et al. (2020).

Forecasting dynamic term structure models

- State space representation: Nelson-Siegel, PCA (three factor models)
- Data: US Historical monthly synthetic yields, structural change.

Contribution

- Propose a new Dynamic term structure model.
- Benchmark: Recurrent Neural Network (Hewamalage et al. 2021).

Preliminary results

- The simple two step auto-encoder model with VAR on neurons has a state space representation.
  - Measurement equation is the decoder.
  - State equation is a VAR(1) on neurons.
  - Encoder is an auxiliary equation of the neurons as a function of the variables.
- Out-of-sample forecasting Performance (rolling window 8 year exercise).
  - PCA and simple linear autoencoder outperform Nelson and Siegel and more complex models: non-linear or deep autoencoder. 1-month 6-month 12-month
  - Recurrent neural network models: Elman recurrent unit, Gated recurrent unit, LSTM, undeperform. 1-month
  - Too few data (right trade off to the number of parameters) and structural changes.

breakout room session

Dynamic term structure models

Term Structure of Interest rates

- Central banks.
  - Expectations of market participants on changes in interest rates.
  - General shape of the curve.
- Academic finance, practitioners.
  - Term and risk premiums (non-observable).
  - Reference curves for pricing (tracking and curve fitting), forecasting and risk management.
- Types of models
  - Structural: Affine term structure models.
  - Reduced Form: Diebold and Li (2006),
  - Hybrid: Ang and Piazzesi (2006), Adrian et al. (2013)

Autoencoders.

A type of neural network model where the inputs and the outputs are the same $Y := X$. Consider them in the following context.

- Multivariate: τ outputs and inputs; $i = 1, \ldots, \tau$.
- Objective is Forecasting hence a regression type problem (vs classification). Linear function when decoding.
- One or more layers but not deep.
- Linear and non linear activation functions, $\tanh(x) \in [-1, 1]$.

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

- Linear autoencoders and dimension reduction.

AutoEncoder with 1 layer.

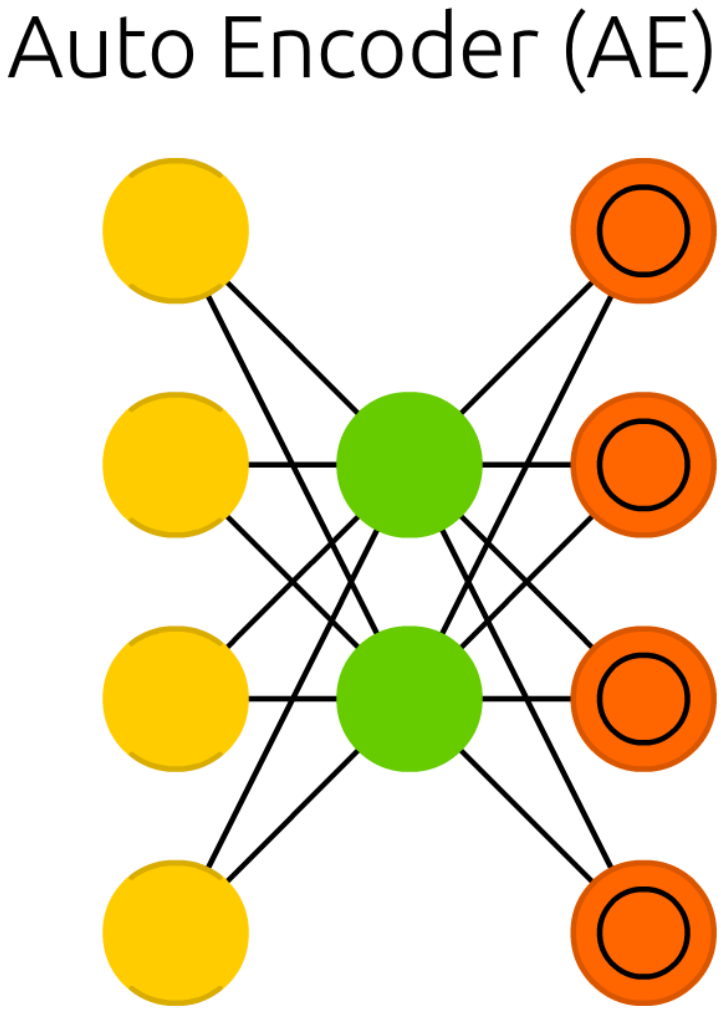


Figure: Simple autoencoder (Van Veen and Leijnen, 2019)

Difference between linear autoencoders and principal components.

- The lower dimensional components in PCA Z are by construction orthogonal where as the neurons are not.
- There is a natural ordering in PCA: the first principal component is the most important.
- The lower dimensional components using Autoencoders contain all of the information. In theory reconstruction error should be smaller because in PCA you only keep the K first principal components.

Empirically, solving the reconstruction problem is equivalent to testing the fit of the model over all of the sample and gives us an initial comparison based on some data.

Neural Networks and Autoencoders in Finance.

- Dixon, Halperin and Bilokon (2020), Machine learning in Finance: Form Theory to Practice, Springer.
- Heaton, Polson and Witte (2017), Deep learning for finance: deep portfolios, Applied Stochastic Models in Business and Industry.
- Gu, Kelly and Xiu (2021), Autoencoder asset pricing models, Journal of Econometrics.
- Suimon, Sakaji, Izumi and Matsushima (2020), Autoencoder-Based Three-Factor Model for the Yield Curve of Japanese Government Bonds and a Trading Strategy, Journal of Risk and Financial Management.

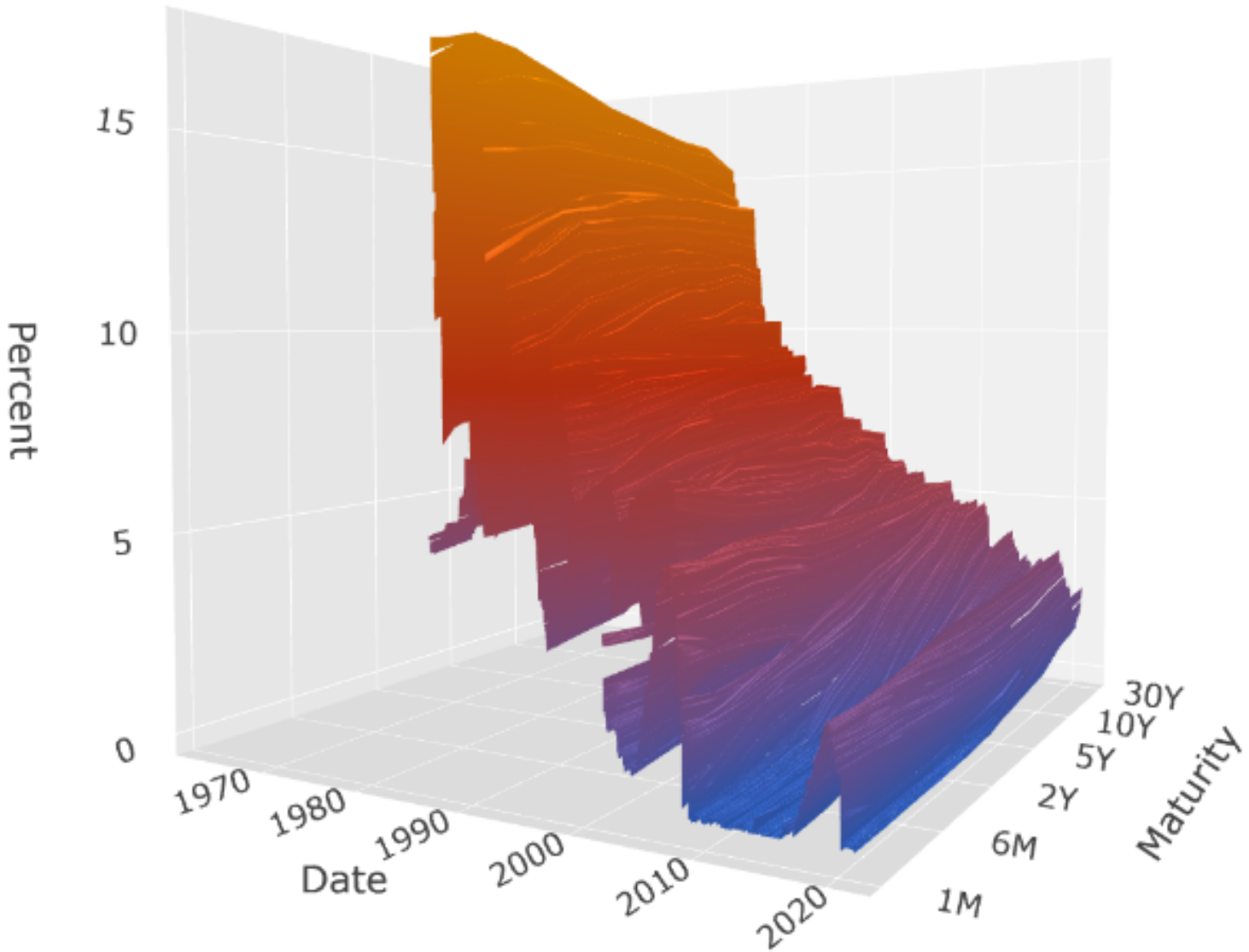
US Yield Data

Monthly synthetic yields from November 1985 to December 2020 (400 observations).

- CRSP T-bill, 1,2,3,4,6 months.
- CRSP T-bond, 1,2,3,4,5 years.
- U.S. Treasury constant maturity yields, 7, 10, 20, 30 years.
- Federal reserve nominal yield curve, 15 and 25 years.

$\tau = 16$ maturities (observed nodes).

US Historical Synthetic Yields



In-sample performance, RMSE in bp

Maturity	NS3	PCA	LA3	NA3	DA3
0.1	17.78	12.50	12.62	19.97	17.25
0.2	6.78	4.72	4.73	11.344	9.98
0.3	5.92	4.05	4.07	9.76	8.14
0.4	7.53	4.45	4.48	8.68	7.55
0.5	10.39	5.55	5.55	8.70	7.43
1.0	38.93	8.59	8.64	10.99	9.70
2.0	20.03	5.56	5.54	13.18	10.72
3.0	7.15	3.98	4.02	13.08	11.11
4.0	5.99	5.56	5.57	13.06	11.24
5.0	11.86	5.86	5.91	12.24	10.72
7.0	11.76	5.99	5.98	10.09	8.22
10.0	16.63	5.75	5.75	7.46	5.87
15.0	9.98	5.29	5.52	9.82	7.97
20.0	8.37	8.13	8.17	11.45	8.58
25.0	10.96	5.70	5.71	11.43	10.36
30.0	9.05	7.37	7.48	11.44	10.92

10 basis points is 0.10%

Dynamic Term Structure Models

Dynamic models are required for forecasting because they provide a representation of the conditional mean.
Proposition: Every dynamic term structure model has a 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. These time varying factor will provide a mechanism to forecast the variables of interest $y_{t+1}(\tau)$ given a set of loading factors and $E_{t-1}(B_t) = \hat{\Phi}B_{t-1}$

NS: Nelson-Siegel(1987) three factor model

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

where $B_t = (\beta_{1,t}, \beta_{2,t}, \beta_{3,t})$. The Nelson-Siegel is a polynomial approximation using exponential functions for the factor loadings, where λ_t is an exponential decay parameter.

- Factor loading represent: level, slope and curvature.
- $\lambda_t = \lambda \forall t$ (Diebold and Li, 2006), the you can use OLS to measure B_t and forecast using a Vector autoregreessive model.
- λ_t Trade-off: low (high) λ better fit for longer (shorter) part of the curve.
- Empirically fix λ at maturity that maximizes curvature or estimate using Kalman-Filter.
- Four factor model $\lambda_{1,t}, \lambda_{2,t}$, Svensson(1994), Bjork and Christensen(1999)

PCA: Three first principal components

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

where $B_t = (Z_{1,t}, Z_{2,t}, Z_{3,t})$ and the factor loadings are given by the matrix of the first three eigenvectors $V(\tau) = (V_1(\tau), V_2(\tau), V_3(\tau))$.

- Factor loading represent: level, slope and curvature.
- Use the variance-covariance matrix of the data to obtain the eigenvalue decomposition and obtain the principal components to measure B_t and forecast using a Vector autoregreessive model.
- Extend the model to a four or a k-factor model.

LA3, NA3, DA3: Autoencoders with 3 neurons and one or more layers

$$y_{t+1}(\tau) = Z_{1,t} W_1^{(2)}(\tau) + Z_{2,t} W_2^{(2)}(\tau) + Z_{3,t} W_3^{(2)}(\tau) + b^{(2)} 1(\tau) + \varepsilon_{t+1}(\tau)$$

$$Z_{i,t} = \tanh(W_i^{(1)} y_t(\tau) + b_i^{(1)})$$

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).

- Factor loading represent: level, slope and curvature.
- Fit the neural network to measure B_t and forecast using a Vector autoregreessive model. This is a naive forecasting approach because we are not optimally adjusting the weights.
- Extend the model:
  - 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 forecasting exercise

- Estimation/Training window 8 years of data.
- Evaluation/Testing window is a rolling window starting $1-1994$ and ending $12-2020$.
- Loss function is the Root Mean Square Error (RMSE) in basis points (bp).
- Forecast horizon: 1, 6 and 12 months.
- Benchmark is a random walk (RW), $E_t(y_{t+1}(\tau)) = y_t(\tau)$.
- Diebold-Mariano test for forecast accuracy,
  $H_0 : y_{t+1}(\tau) - \hat{y}_{t+1}^{model1}(\tau) = y_{t+1}(\tau) - \hat{y}_{t+1}^{model2}(\tau)$
- Cumulative square prediction error (CSPE) with tespect to the random walk.

$$CSPE_t = \sum_{i=1}^{t}((\hat{y}_i^{RW}(\tau) - y_i(\tau))^2 - (\hat{y}_i^{model}(\tau) - y_i(\tau))^2)$$

Out-of-sample performance, Horizon 1 month

Maturity	RW	NS3	PCA	LA3	NA3	DA3
0.1	25.63	28.28	22.25	22.65	36.44	24.25
0.2	17.28	19.25	17.33	17.71	40.66	19.66
0.3	16.69	18.80	17.02	17.44	48.78	19.12
0.4	16.29	19.17	16.89	17.09	55.09	18.75
0.5	17.79	20.47	17.56	18.14	59.96	19.61
1.0	21.57	42.23	24.42	25.30	97.89	26.10
2.0	24.12	31.61	27.95	28.85	126.31	29.23
3.0	25.63	28.34	27.69	28.19	131.61	29.17
4.0	26.84	29.96	28.41	28.83	136.09	29.92
5.0	26.52	31.74	27.66	27.95	139.56	29.53
7.0	26.28	30.49	27.23	27.17	140.50	29.95
10.0	25.55	33.35	26.04	26.23	146.72	29.14
15.0	24.10	26.19	26.89	27.31	131.67	31.39
20.0	23.54	25.74	26.46	27.16	131.03	29.92
25.0	22.27	24.94	25.37	26.18	128.60	31.32
30.0	22.47	27.58	24.83	25.76	133.91	31.15

10 basis points is 0.10%. Diebold-Mariano CSPE

Out-of-sample performance, Horizon 6 months

Maturity	RW	NS3	PCA	LA3	DA3
0.1	63.52	69.06	58.66	59.57	60.34
0.2	61.61	68.10	62.13	62.72	62.66
0.3	61.52	70.30	65.32	65.70	65.38
0.4	61.67	71.69	67.01	67.40	66.92
0.5	62.32	72.96	68.55	69.05	67.97
1.0	74.81	82.18	87.88	88.53	86.97
2.0	74.51	84.00	88.58	89.00	87.90
3.0	73.01	84.73	84.46	85.08	84.74
4.0	72.11	86.15	82.22	82.53	82.73
5.0	70.62	86.87	80.39	80.68	81.23
7.0	66.70	80.50	76.10	76.43	77.71
10.0	64.15	79.95	72.77	73.03	74.86
15.0	57.50	63.68	69.31	69.67	71.59
20.0	56.59	64.02	67.82	68.32	69.78
25.0	53.95	60.65	65.11	65.81	68.32
30.0	55.00	64.68	63.25	63.88	67.52

10 basis points is 0.10% Diebold-Mariano CSPE

Out-of-sample performance, Horizon 12 months

Maturity	RW	NS3	PCA	LA3	DA3
0.1	104.54	124.07	108.36	109.29	104.82
0.2	105.63	125.54	115.59	116.29	109.81
0.3	105.55	128.01	119.61	120.31	112.75
0.4	105.25	129.26	121.68	122.41	114.40
0.5	105.60	130.50	123.59	124.45	115.64
1.0	121.92	136.52	153.04	153.64	141.31
2.0	112.28	137.22	146.94	147.39	135.32
3.0	104.17	135.27	136.33	136.77	127.64
4.0	98.44	132.60	128.05	128.25	121.50
5.0	94.46	129.92	122.07	122.36	116.97
7.0	88.33	118.76	113.38	113.66	111.58
10.0	83.35	113.12	106.01	106.60	107.47
15.0	74.42	89.98	98.29	98.85	103.73
20.0	72.85	88.58	94.81	95.56	100.56
25.0	68.90	83.52	91.36	92.19	99.63
30.0	69.63	87.85	88.41	89.45	97.16

10 basis points is 0.10% Diebold-Mariano CSPE

Recurent Neural Networks (RNN) and Time Series Forecasting, Hewamalage et al. (2021)

- Elman recurrent unit, ERNN(1).

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

- Gated recurrent unit, GRNN(1).

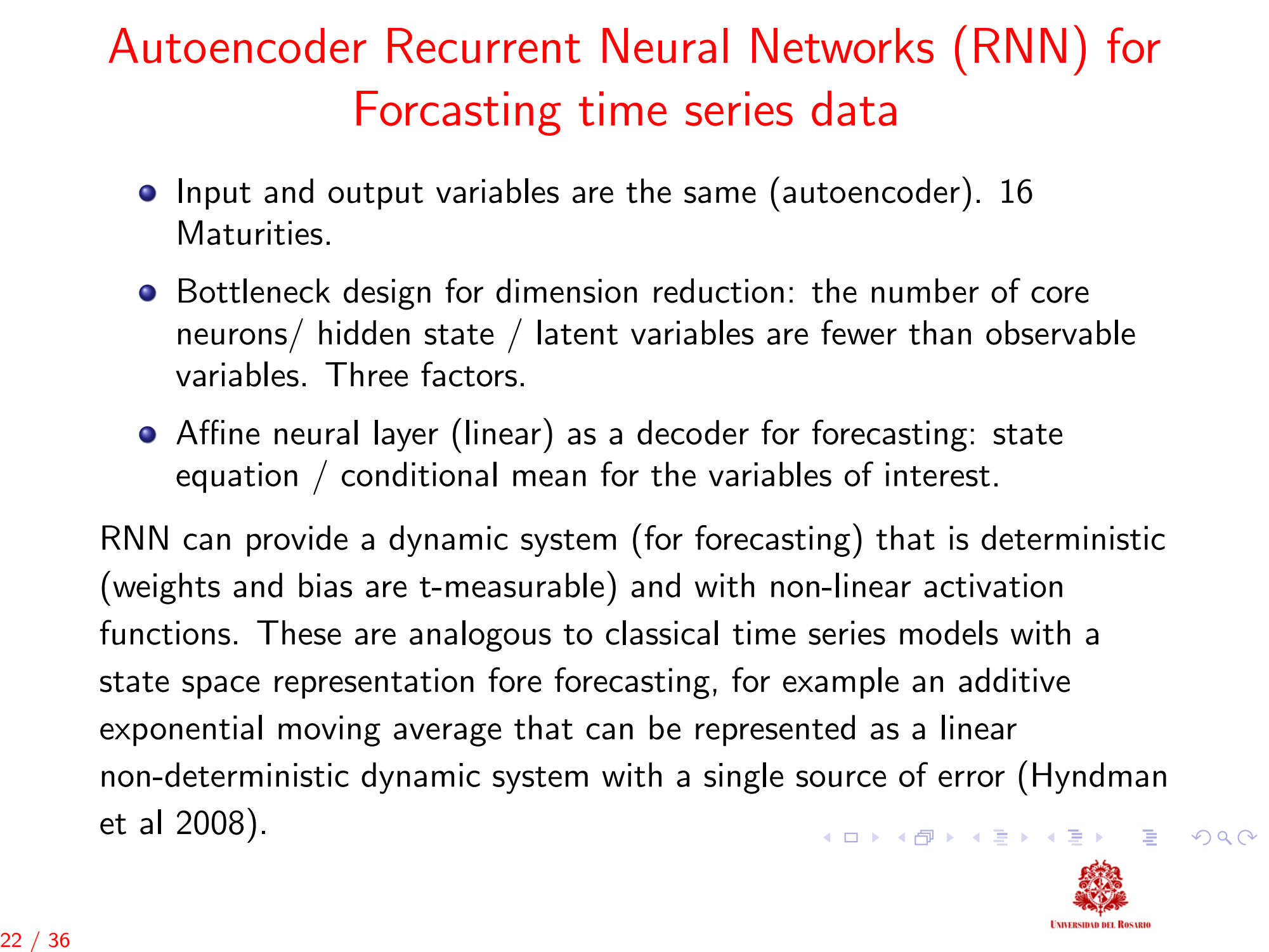
$$y_{t+1}(\tau) = Z_t W^{(2)}(\tau) + b^{(2)} 1(\tau) + \varepsilon_{t+1}(\tau)$$
$$Z_t = u_t \circ \tilde{Z}_t + (1 - u_t) \circ Z_{t-1}$$
$$\tilde{Z}_t = \tanh(y_t(\tau) W_{z,y}^{(1)} + Z_{t-1} W_{z,z}^{(1)} r_t + 1 b_z^{(1)})$$
$$r_t = \sigma(W_{r,y}^{(1)} y_t(\tau) + W_{r,z}^{(1)} Z_{t-1} + 1 b_r^{(1)})$$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

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

Autoencoder Recurrent Neural Networks (RNN) for Forcasting time series data

- Input and output variables are the same (autoencoder). 16 Maturities.
- Bottleneck design for dimension reduction: the number of core neurons/ hidden state / latent variables are fewer than observable variables. Three factors.
- Affine neural layer (linear) as a decoder for forecasting: state equation / conditional mean for the variables of interest.

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).



In-sample performance, RMSE in bp

τ	ERNN_{20}	GRNN_{20}	LSTM_{20}	ERNN_{200}	GRNN_{200}	LSTM_{200}
0.1	25.59	24.93	23.38	8.37	12.54	3.76
0.2	15.77	12.91	11.86	5.46	6.25	3.38
0.3	15.87	11.40	10.99	5.17	5.88	3.14
0.4	16.07	10.54	10.71	4.64	6.36	3.51
0.5	17.31	11.18	11.57	5.67	6.84	3.03
1.0	21.24	12.81	13.77	6.20	7.60	3.80
2.0	21.30	13.62	14.03	5.83	6.82	3.65
3.0	21.52	13.66	13.65	5.13	5.94	3.53
4.0	21.94	14.90	13.40	5.03	6.18	3.14
5.0	21.62	14.47	13.15	5.08	6.23	3.09
7.0	21.44	14.83	12.47	4.16	5.62	2.96
10.0	21.33	17.28	12.37	4.95	5.98	2.95
15.0	20.85	13.83	12.52	4.95	5.89	2.97
20.0	21.22	13.24	12.12	5.15	6.04	3.01
25.0	19.80	14.80	12.25	5.04	6.51	3.11
30.0	19.78	19.01	11.95	5.10	6.56	2.82

10 basis points is 0.10%

Out-of-sample performance, Horizon 1 month

τ	ERNN_{20}	GRNN_{20}	LSTM_{20}	ERNN_{200}	GRNN_{200}	LSTM_{200}
0.1	173.52	171.96	171.96	185.83	176.92	177.19
0.2	185.93	194.31	182.31	191.18	185.40	186.52
0.3	191.14	200.68	185.58	191.67	188.61	188.67
0.4	193.32	208.67	192.57	192.33	190.77	191.06
0.5	191.83	208.40	192.61	192.74	191.67	192.01
1.0	244.05	273.30	246.16	236.09	238.86	237.81
2.0	257.33	281.97	256.88	246.32	253.73	251.10
3.0	253.63	266.66	253.84	250.62	251.66	252.00
4.0	261.19	275.55	264.36	253.40	260.20	259.35
5.0	260.69	268.47	257.13	254.38	260.13	259.75
7.0	260.56	272.74	257.86	254.20	258.63	259.32
10.0	255.46	255.79	247.35	245.18	250.81	252.00
15.0	255.57	291.27	253.35	252.04	254.43	256.53
20.0	257.90	242.50	254.61	239.78	254.95	254.52
25.0	256.35	275.83	256.27	254.92	255.84	257.45
30.0	236.37	232.55	235.30	236.65	236.36	236.97

10 basis points is 0.10%

Appendix

Linear Autoencoders and dimension reduction

There is dimension reduction if the number of neurons Z are smaller than the number of variables, $M << \tau$.

$$Y_i = w^{(2)} z_i + b^{(2)}$$

$$z_i = W^{(1)} Y_i + b^{(1)}$$

Training the model requires solving the following reconstruction (optimization) problem,

$$min_{W^{(1)},\mathrm{b}^{(1)},W^{(2)},\mathrm{b}^{(2)}} \; || \; \mathrm{Y} - W^{(2)}(W^{(1)}\mathrm{Y} + \mathrm{b}^{(1)}1'_N) + \mathrm{b}^{(2)}1'_N \; ||^2_F$$

- $Y_{n\mathrm{x}\tau}, Z_{M\mathrm{x}N}$.
- Decoder $W^{(2)}_{n\mathrm{x}M}$.
- Encoder $W^{(1)}_{M\mathrm{x}n}$.
- Bias or intercept: $\mathrm{b}^{(1)}_n, \mathrm{b}^{(2)}_n$

The problem can be simplified by transforming Y as deviation from columns means Y_0 eliminating bias terms.

Linear Autoencoders and Principal Components.

Simplified reconstruction (optimization) problem

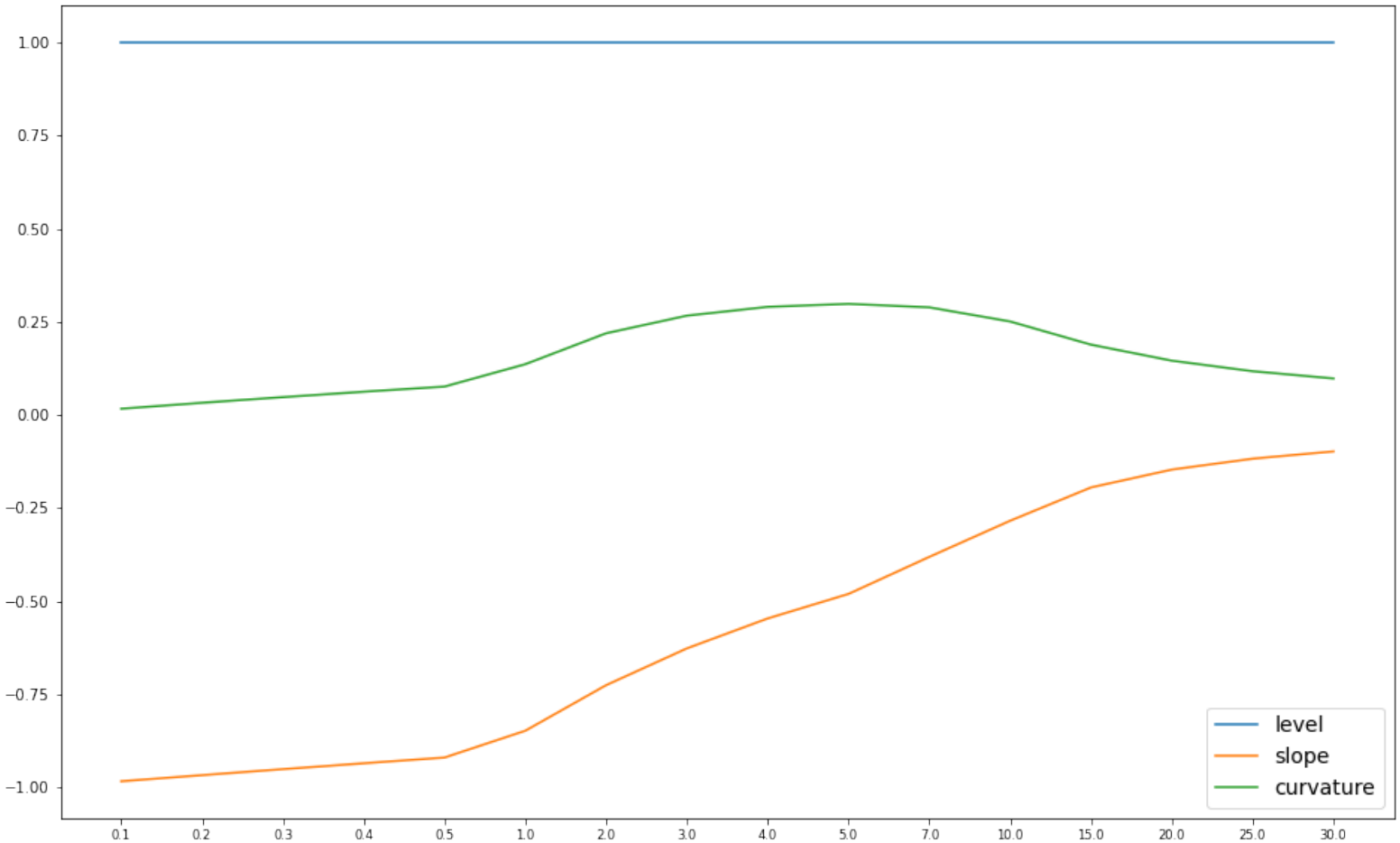
$$min_{W^{(2)}} \mid\mid \mathrm{Y}_0 - W^{(2)} W^{(2)+} \mathrm{Y}_0 \mid\mid_F^2$$

where $P_{W^{(2)}} := W^{(2)} W^{(2)+} = W^{(2)} (W^{(2)\prime} W^{(2)})^{-1} W^{(2)\prime}$ is an ortogonal projection matriz over the space generated by $W^{(2)}$, (Baldi y Hornik, 1989). Note that the columns of $W^{(2)}$ are not necessarily ortonormal. Principal components is a special case of a linear autoencoder because in the reconstruction problem there is also a projection onto a lower dimensional subspace,

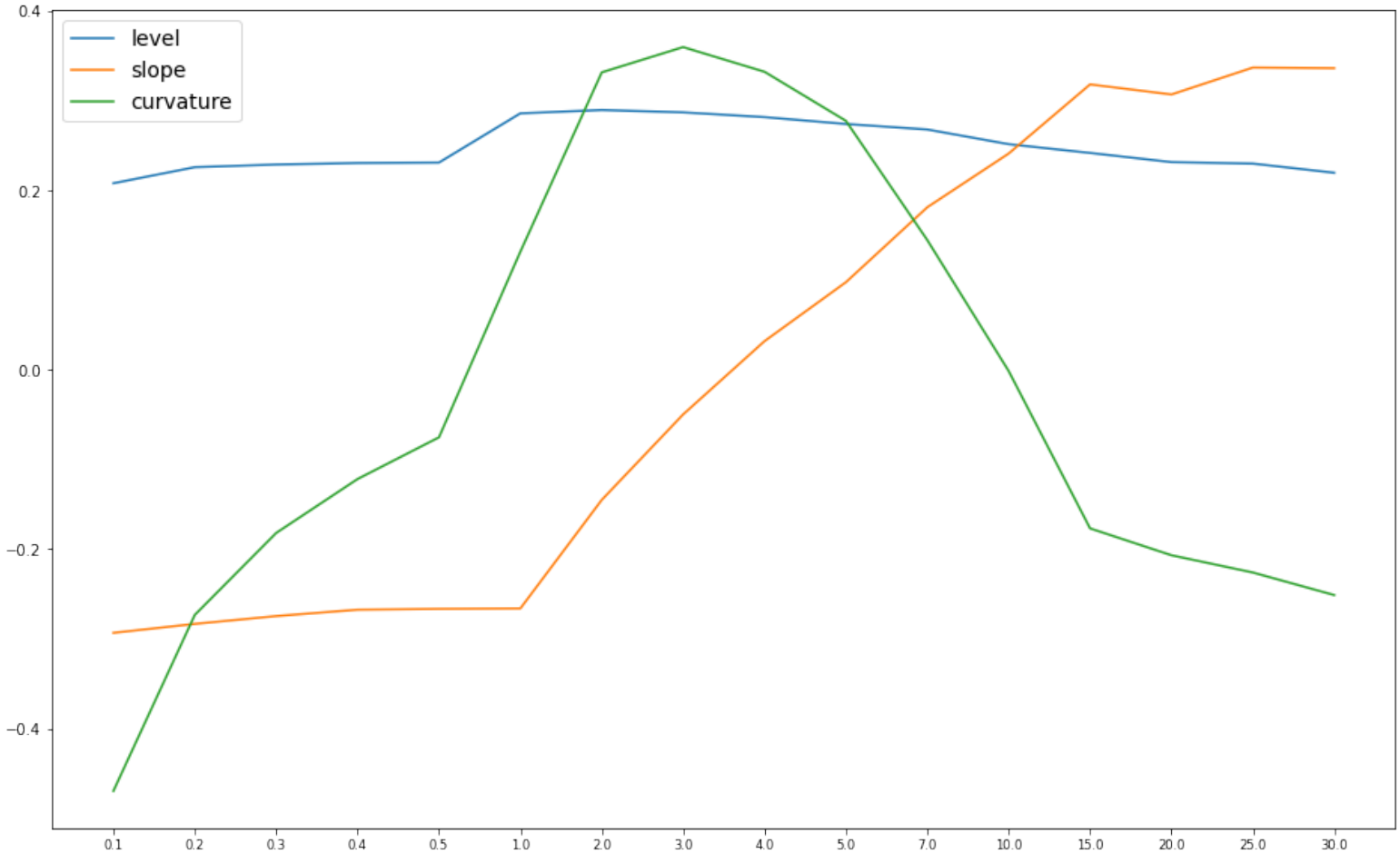
$$min_V \mid\mid \mathrm{Y}_0 - VZ' \mid\mid_F^2 = \mid\mid \mathrm{Y}_0 - VV'\mathrm{Y}_0 \mid\mid_F^2, \text{s.a.} V'V = I_M$$

where Z is the matrix of M first principal components and V is the matrix of the corresponding eigen vectors.

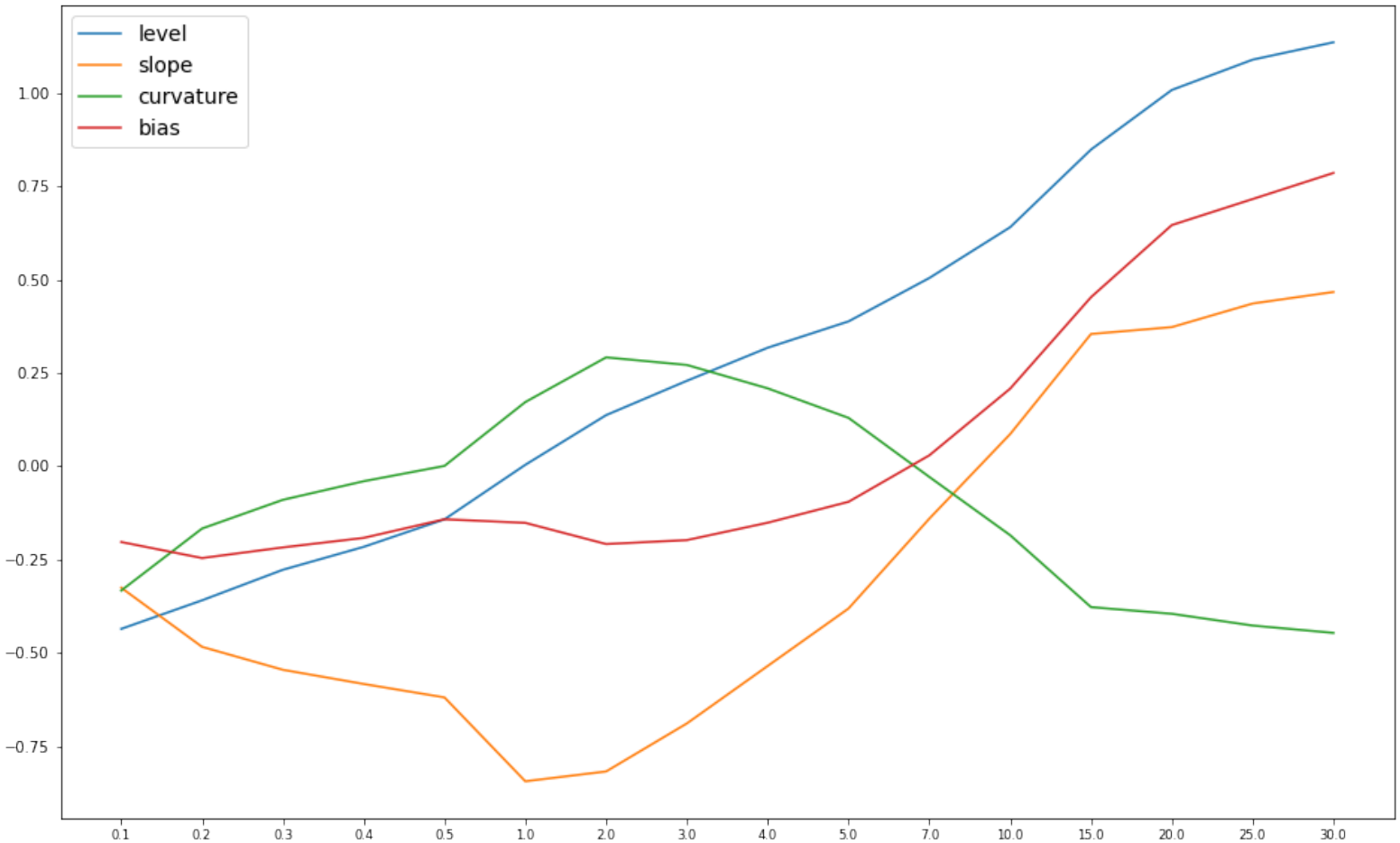
Factor Loadings Nelson-Siegel: level, slope and curvature



Factor Loadings PCA3: level, slope and curvature.



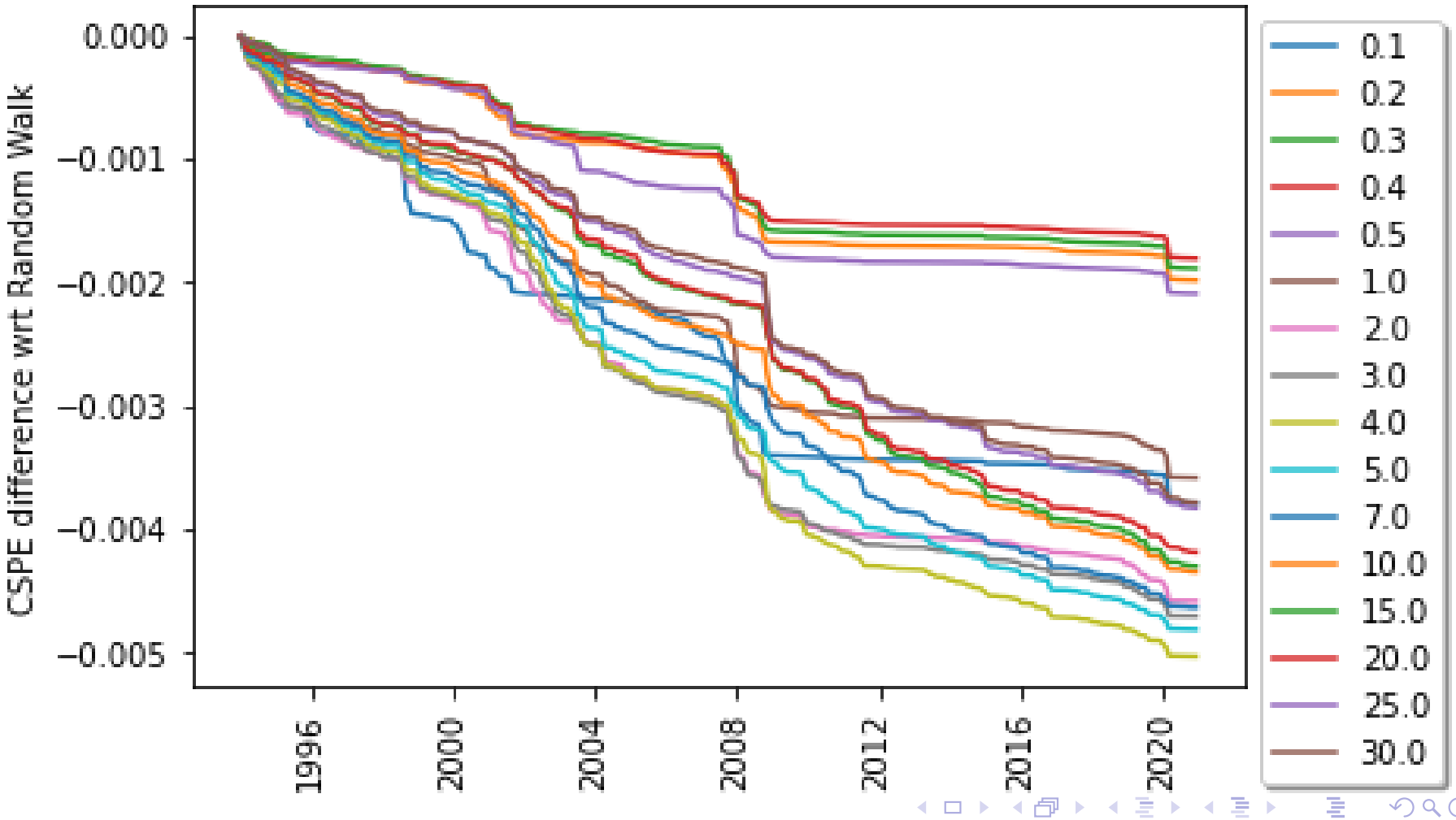
Factor Loadings Linear Autoencoders 3 neurons: level, slope and curvature.



Out-of-sample performance, Horizon 1 month, Diebold-Mariano p-values

Maturity	PCA/LA3	RW/LA3	NS/LA3	DA3/LA3	RW/DA3
0.1	0.454	0.071	0.000	0.054	0.413
0.2	0.091	0.588	0.000	0.007	0.010
0.3	0.045	0.292	0.000	0.032	0.008
0.4	0.367	0.213	0.000	0.035	0.003
0.5	0.023	0.686	0.000	0.062	0.072
1.0	0.008	0.000	0.000	0.382	0.000
2.0	0.007	0.000	0.006	0.675	0.000
3.0	0.101	0.000	0.768	0.165	0.000
4.0	0.058	0.010	0.031	0.187	0.000
5.0	0.100	0.003	0.000	0.056	0.001
7.0	0.639	0.173	0.000	0.006	0.003
10.0	0.152	0.222	0.000	0.023	0.012
15.0	0.083	0.000	0.255	0.006	0.000
20.0	0.038	0.000	0.027	0.026	0.000
25.0	0.091	0.000	0.152	0.001	0.000
30.0	0.030	0.001	0.050	0.001	0.000

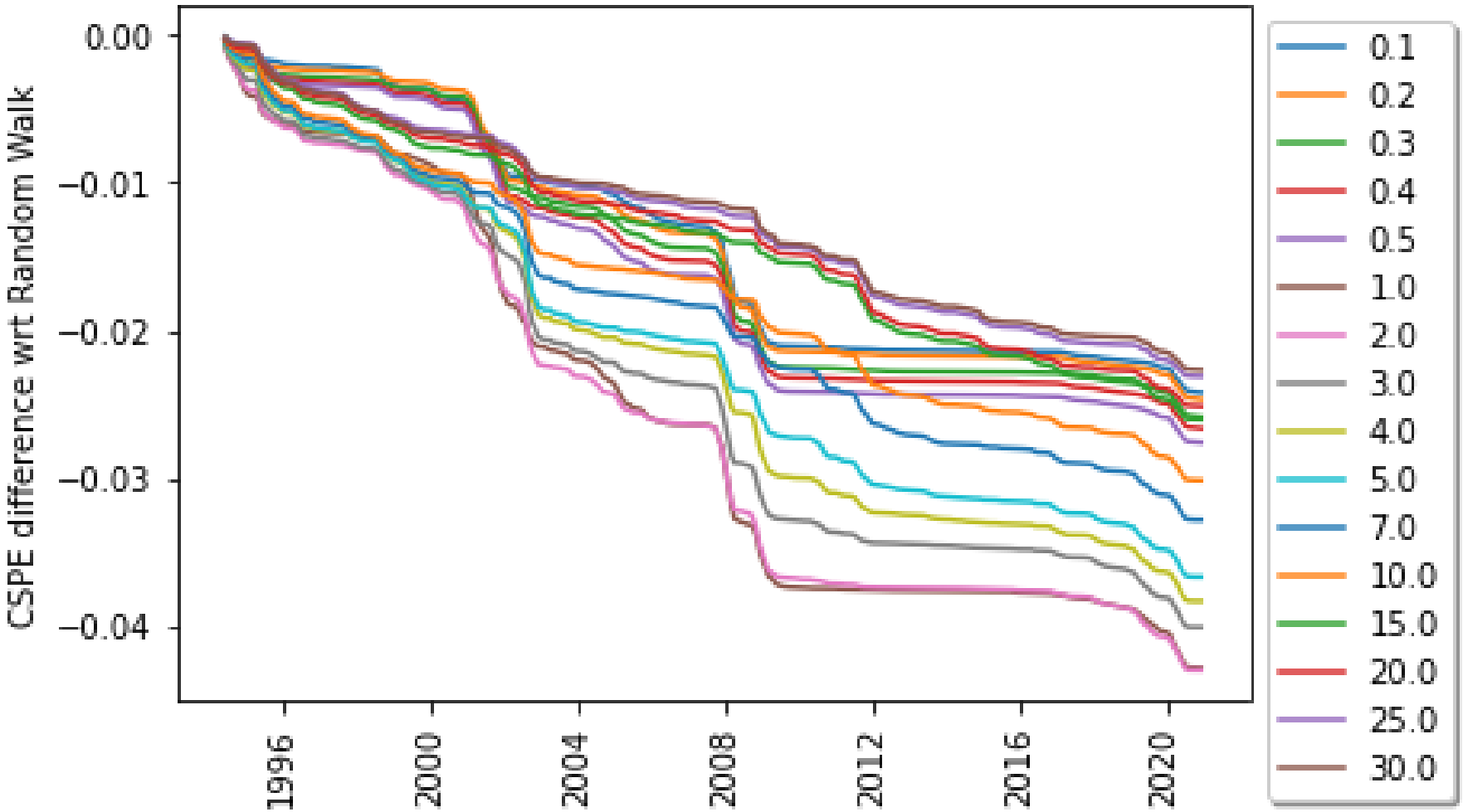
CSPE-Linear Autoencoder 3 neurons, Horizon 1 Month.



Out-of-sample performance, Horizon 6 months, Diebold-Mariano p-values

Maturity	PCA/LA3	RW/LA3	NS/LA3	DA3/LA3	RW/DA3
0.1	0.035	0.413	0.000	0.697	0.473
0.2	0.242	0.820	0.000	0.977	0.815
0.3	0.464	0.387	0.001	0.892	0.378
0.4	0.484	0.232	0.013	0.841	0.229
0.5	0.422	0.160	0.083	0.674	0.197
1.0	0.363	0.023	0.281	0.615	0.028
2.0	0.496	0.012	0.062	0.701	0.008
3.0	0.079	0.035	0.692	0.895	0.011
4.0	0.266	0.066	0.001	0.939	0.016
5.0	0.213	0.077	0.005	0.832	0.020
7.0	0.161	0.088	0.054	0.593	0.020
10.0	0.099	0.116	0.006	0.486	0.031
15.0	0.030	0.018	0.011	0.483	0.003
20.0	0.013	0.013	0.047	0.594	0.001
25.0	0.026	0.019	0.091	0.406	0.003
30.0	0.007	0.038	0.735	0.200	0.008

CSPE-Linear Autoencoder 3 neurons, Horizon 6 Months.



Out-of-sample performance, Horizon 12 months, Diebold-Mariano p-values

Maturity	PCA/LA3	RW/LA3	NS/LA3	DA3/LA3	RW/DA3
0.1	0.083	0.545	0.002	0.438	0.294
0.2	0.327	0.226	0.001	0.478	0.192
0.3	0.340	0.131	0.001	0.558	0.171
0.4	0.329	0.106	0.009	0.590	0.168
0.5	0.266	0.093	0.056	0.722	0.161
1.0	0.508	0.060	0.056	0.720	0.140
2.0	0.509	0.069	0.051	0.638	0.147
3.0	0.453	0.085	0.537	0.471	0.154
4.0	0.660	0.103	0.020	0.372	0.175
5.0	0.585	0.120	0.007	0.331	0.183
7.0	0.630	0.129	0.113	0.290	0.193
10.0	0.424	0.147	0.028	0.275	0.210
15.0	0.318	0.054	0.046	0.277	0.174
20.0	0.215	0.062	0.160	0.281	0.190
25.0	0.169	0.049	0.142	0.262	0.173
30.0	0.125	0.061	0.742	0.268	0.194

CSPE, Horizon 12 Months.