Variational autoencoders in time series analysis

Péter Nemesi
László Varga

AI Research Group
Artificial Intelligence & Data Science
Eötvös Loránd University

ELTE AI Research Seminar

June 17, 2025

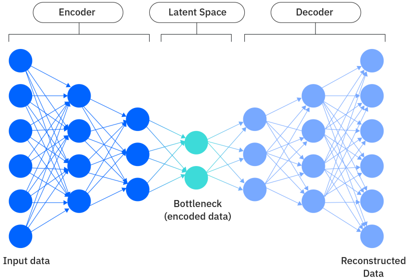
Outline

Goals of the research

- Strategic goal: mimic an **arbitrary** empirical time series using variational autoencoders
- Tactical goals: mimic samples from important, well-known time series models (SARIMA, GARCH, fractional Brownian motion) with variational autoencoders
- Possible applications: forecasting, classification/clustering, scenario analysis, volatility estimation

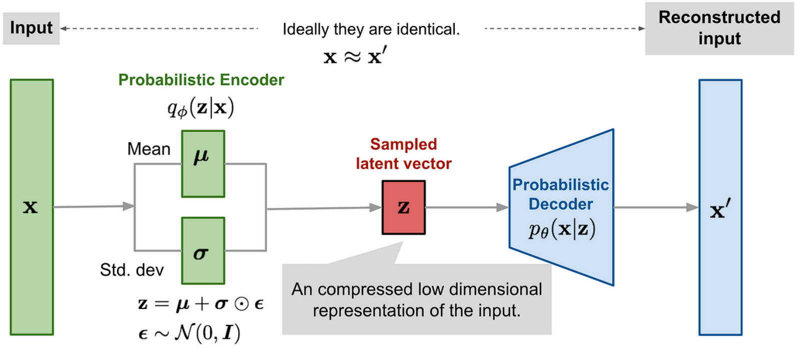
Autoencoder

- Encoder: input vector $\mathbf{x} \in \mathbb{R}^d$ is mapped to latent vector $\mathbf{h} \in \mathbb{R}^k$, where usually $k < d$:
$$\mathbf{h} = f_{\text{enc}}(\mathbf{x}),$$
where f_{enc} is a multi-layer neural network.
- Decoder: from the latent vector, we reconstruct vector $\hat{\mathbf{x}} \in \mathbb{R}^d$ adatot:
$$\hat{\mathbf{x}} = f_{\text{dec}}(\mathbf{h}).$$
- Goal: $\hat{\mathbf{x}} \approx \mathbf{x}$, as well as we can



Variational Autoencoder

- We teach a distribution in the latent space
  - $p_\theta(\mathbf{z} \mid \mathbf{x})$ is not tractable
  - the encoder (inference network) does not return a single vector, but a distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$, where ϕ is a vector of parameters and is usually Gaussian.
- The decoder (generative network) tries to reconstruct $\mathbf{x}$ using the conditional distribution $p_\theta(\mathbf{x} \mid \mathbf{z})$



- Score function
  - $D_{\mathrm{KL}}\big(q_\phi(z \mid x) \,\|\, p_\theta(z \mid x)\big) \longrightarrow \min_\phi$
  - Equivalent formulation:

$$\underbrace{\mathbb{E}_{z\sim q_\phi(z|x)}\big[\log p_\theta(x \mid z)\big]}_{\text{reconstruction element}} - \underbrace{D_{\mathrm{KL}}\big(q_\phi(z \mid x) \,\|\, p(z)\big)}_{\text{KL element}}$$

- β-VAE (Miroslav et al., 2021): the KL element is weighted by an additional parameter:

$$\mathbb{E}_{q_\phi}\big[\log p_\theta(x \mid z)\big] - \beta\, D_{\mathrm{KL}}\big(q_\phi(z \mid x) \,\|\, p(z)\big).$$

  - If $\beta > 1$, then the KL element is punished stronger, so the quality of the reconstruction may deteriorate
  - If $\beta < 1$, then the KL element is less punished, the latent space will be of less importance

VAE – score function and β-VAE

- Score function
  - $D_{\mathrm{KL}}\left(q_\phi(z \mid x) \parallel p_\theta(z \mid x)\right) \longrightarrow \min_\phi$
  - Equivalent formulation:

$$\underbrace{\mathbb{E}_{z \sim q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right]}_{\text{reconstruction element}} - \underbrace{D_{\mathrm{KL}}\left(q_\phi(z \mid x) \parallel p(z)\right)}_{\text{KL element}}$$

- β-VAE (Miroslav et al., 2021): the KL element is weighted by an additional parameter:

$$\mathbb{E}_{q_\phi}\left[\log p_\theta(x \mid z)\right] - \beta\, D_{\mathrm{KL}}\left(q_\phi(z \mid x) \parallel p(z)\right).$$

  - If $\beta > 1$, then the KL element is punished stronger, so the quality of the reconstruction may deteriorate
  - If $\beta < 1$, then the KL element is less punished, the latent space will be of less importance

Implementation

- The codebase is available on Github:
  `https://github.com/kornyik/ML-StochTS`,
  on branch *Playground*
- Runs are governed by a yaml config file
- Possible layers in the VAE network
  - Linear
  - GRU
  - LSTM
  - Conv1d, ConvTranspose1d
- Activation functions: ReLU, LeakyReLU, Tanh, Sigmoid, ELU, GELU

Results – ARMA models

- First attempts – no success
- Breakthrough:
  - following the ides of Acciaio et al. [2024] and Bühler et al. [2020]: the shape of the VAE should be a 'barrel', instead of the typical 'hourglass'
  - include LSTM layers
- AR(1) sample with AR parameter $\alpha \in [0.6, 0.99]$ could be reconstructed
- ARMA(1) sample with large AR parameter and small MA parameter could be reconstructed

Layer structure to reconstruct ARMA processes

Following Bühler et al (2020), we used the following network structure:

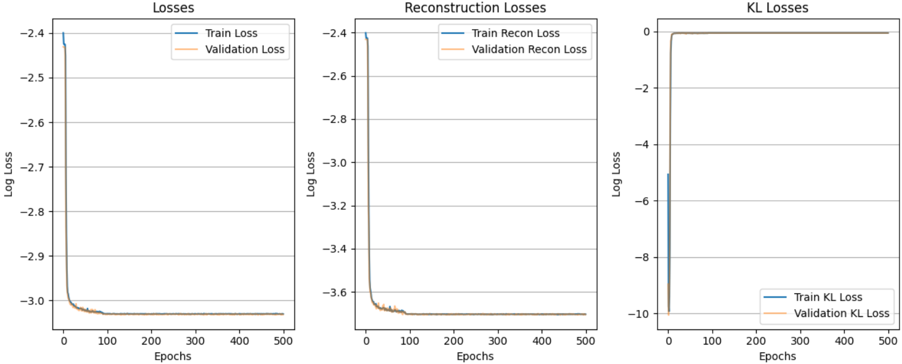
Layer	Output Shape	Nr of parameters
Input	[32, 128, 1]	
Encoder		
Linear	[32, 128, 16]	32
LeakyReLU		–
LSTM	[32, 128, 16]	4 352
Tanh		–
Linear	[32, 128, 16]	272
Tanh		–
Reparametrization		–
Linear (μ)	[32, 128, 2]	34
Linear (σ)	[32, 128, 2]	34
Decoder		
Linear	[32, 128, 16]	48
Tanh		–
LSTM	[32, 128, 16]	4 352
Tanh		–
Linear	[32, 128, 16]	272
Tanh		–
Linear	[32, 128, 1]	17
Total parameters		9 413

Results

Posterior collapse: the approximate posterior becomes almost equal to the prior: $q_\phi(z \mid x) \approx p(z)$. This means that the

- latent space is underutilized
- model behaves like a regular autoencoder

To avoid posterior collapse, both parts of the score function have to be assessed.



Conclusions, further work

- Achievements:
  - We can mimic samples from stationary ARMA processes with small positive AR polynomial roots
  - We have developed a codebase useful for wider testing purpose
- Challenge: runtime (the bottleneck is the time needed for ARMA model fitting)
- Further plans:
  - Calibrate the network so that it is able to recostruct stationary ARMA processes with small negative AR polynomial roots
  - Enhance the deep network so that it is able to reconstruct samples from SARIMA and GARCH processes
  - Forecasting on real life data (US GNP, Johnson&Johnson EPS, M5 competition: Walmart sales)
  - Reconstructing of stochastic volatility (GARCH)

Conclusions, further work

- Achievements:
  - We can mimic samples from stationary ARMA processes with small positive AR polynomial roots
  - We have developed a codebase useful for wider testing purpose
- Challenge: runtime (the bottleneck is the time needed for ARMA model fitting)
- Further plans:
  - Calibrate the network so that it is able to recostruct stationary ARMA processes with small negative AR polynomial roots
  - Enhance the deep network so that it is able to reconstruct samples from SARIMA and GARCH processes
  - Forecasting on real life data (US GNP, Johnson&Johnson EPS, M5 competition: Walmart sales)
  - Reconstructing of stochastic volatility (GARCH)

Thank you for the attention!

Questions / comments?

Acknowledgement: András Zempléni and Gábor Fáth

References

- B. Acciaio, S. Eckstein, S. Hou: Timecausal VAE: Robust Financial time series generator, *arXiv preprint arXiv:2411.02947*, 2024, https://arxiv.org/abs/2411.02947.
- H. Bühler, B. Horvath, T. Lyons, I.P. Arribas, B. Wood: A data-driven market simulator for small data environments, *arXiv preprint arXiv:2006* 2020, https://arxiv.org/abs/2006.14498.
- A. Desai, C. Freeman, Z. Wang, I. Beaver: Timevae: A variational auto-encoder for multivariate time series generation, *arXiv preprint arXiv:2111.08095*, 2021, https://arxiv.org/abs/2111.08095.
- F. Miroslav, M. Munib, M. Matthew, W. Jonas: Beta-VAE reproducibility: Challenges and extensions. *ArXiv preprint arXiv:2112.14278*, 2021, https://arxiv.org/abs/2112.14278.