Using LSTMs with Variational Autoencoders to Model Time Series Data

T. Ansah-Narh

February 19, 2025

AutoEncoder (AE): A type of unsupervised neural networks

- The **encoder** network transforms the input data from a high-dimensional space into codes in a low-dimensional space.
- The **decoder** network reconstructs the inputs from the corresponding codes.

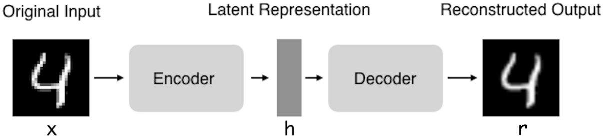


Figure 1: Representation of a simple autoencoder. *Credit*: http://i-systems.github.io

AutoEncoder (AE):Loss function

- Reconstruction error: $||x - \tilde{x}||$

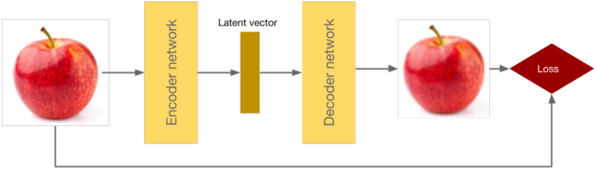


Figure 2: Basic principle of an AutoEncoder. *Credit*: Dehaene's blog, 2018.

AutoEncoder (AE): Applications

- Image compression
- Image denoising

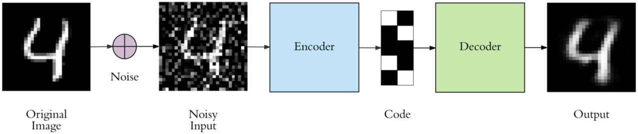


Figure 3: Basic principle of an AutoEncoder. *Credit*: Dehaene's blog, 2018.

Variational AutoEncoder (VAE):

Consider

- Observable Data: $\mathbf{X} = \{x_1, x_2, x_3, \ldots, x_m\}$
- Hidden Variable: $\mathbf{z} = \{z_1, z_2, z_3, \ldots, z_n\}$

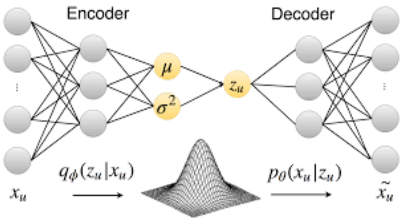


Figure 4: Architecture graph of a variational autoencoder. *Credit*: Karamanolakis et al., 2018.

VAE:

Training a VAE, the loss function becomes:

$$\mathcal{L}(\theta, \phi; \boldsymbol{x}, \boldsymbol{z}) = \mathbb{E}_z[log\ p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \frac{\mathbb{E}_z[log\ q_\phi(\boldsymbol{z}|\boldsymbol{x})]}{p_\theta(z)} \tag{1a}$$

$$= \mathbb{E}_z[log\ p_\theta(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z})) \tag{1b}$$

RNN

1. Periodic formula: $S_t = R_w(S_{t-1}, X_t)$
   - X_t – Input at time t
   - S_t – Current state at time t
   - S_{t-1} – Initial state at time $t-1$
   - R_w – Periodic function
2. Simple case: $S_t = \tanh(W_s S_{t-1} + W_x X_t)$

How RNNs works

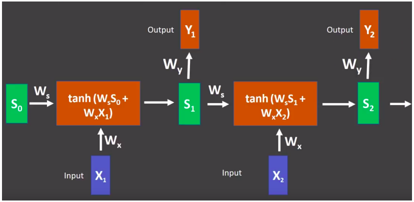


Figure 5: Basic RNN (unfolded). *Credit*: The Semicolon RNN lecture, 2018.

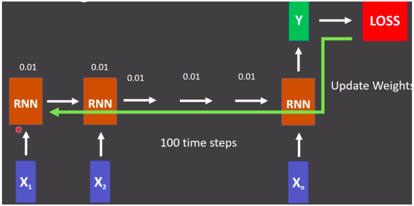
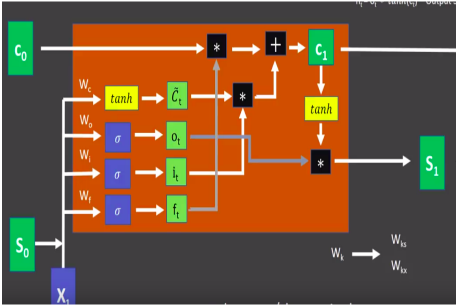


Figure 6: Problem of vanishing gradient. *Credit*: The Semicolon RNN lecture, 2018.

LSTM

- Input Gate:
  $i_t = \sigma(W_i S_{t-1} + W_i X_i)$
- Forget Gate:
  $f_t = \sigma(W_f S_{t-1} + W_f X_f)$
- Output Gate:
  $o_t = \sigma(W_o S_{t-1} + W_o X_o)$
- Cell State:
  $c_t = (i_t * \tilde{C}_t) + (f_t * c_{t-1})$
  - Intermediate cell state:
    $\tilde{C}_t = \tanh(W_c S_{t-1} + W_c X_c)$



Architecture graph of LSTM. *Credit*: The Semicolon RNN lecture, 2018.

LSTM-VAE

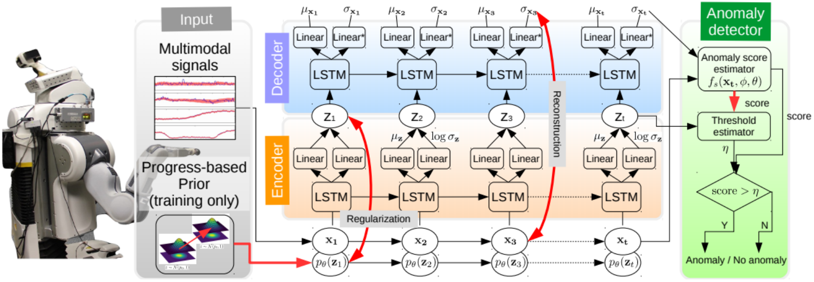
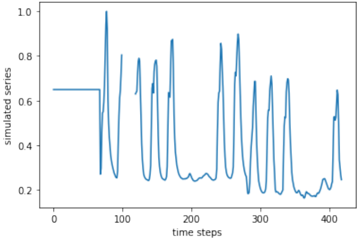
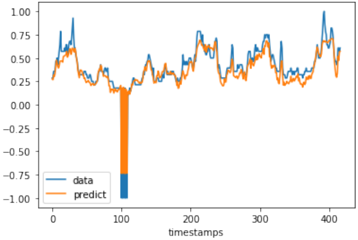
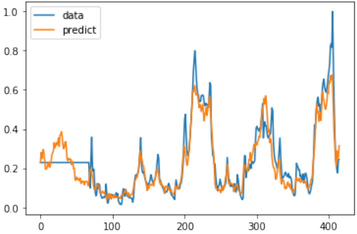


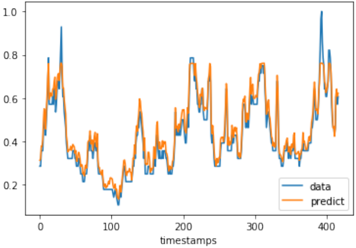
Figure 7: Architecture graph of an LSTM-VAE. Credit: Park et al., 2017.

Simulation Test






1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
data
predict


1.0
0.8
0.6
0.4
0.2
0
100
200
300
400
timestamps
data
predict

Simulation Test

- Run an RNN on the imputed inputs (i.e. mean, median, fill nans)
- Use weighted average approach to interpolate missing values
- Gaussian processes to interpolate missing values


That's all Folks!