

Functional Autoencoder with Time Series Forecasting

1 Functional Autoencoder with Latent Time Series

Let $\{X^{(i)}(\cdot)\}_{i=1}^N$ denote functional observations, each observed on a common grid $\mathcal{T} = \{t_1, \dots, t_J\}$. We denote the discretized i th curve by

$$\mathbf{x}^{(i)} = (X^{(i)}(t_1), \dots, X^{(i)}(t_J))^\top \in \mathbb{R}^J.$$

The encoder maps each curve to a K -dimensional latent representation

$$\mathbf{h}^{(i)} = E_\theta(\mathbf{x}^{(i)}) \in \mathbb{R}^K,$$

and the decoder reconstructs it in a basis representation,

$$\mathbf{b}^{(i)} = D_\theta(\mathbf{h}^{(i)}) \in \mathbb{R}^{M^{(O)}}, \quad \hat{\mathbf{x}}^{(i)} = \sum_{m=1}^{M^{(O)}} \mathbf{b}_m^{(i)} \phi_m^{(O)}(t),$$

For each functional observation, we assume that $J + L$ time points are available. The first J points are used to train the autoencoder, and the remaining L points are used for forecasting. After encoding the first J points, the latent dynamics are iteratively applied to generate future latent states for $t = J, \dots, J + L - 1$, which are then decoded and compared with the corresponding observed values. To impose this temporal structure on the latent space, we introduce a latent dynamical model acting on the sequence of latent representations for each sample.

$$\tilde{\mathbf{h}}_{t+1}^{(i)} = g_\psi(h_1^{(i)}, \dots, h_t^{(i)}); \quad t = K, \dots, K + L - 1$$

where g_ψ is a parametric time-series model (VAR, GRU, LSTM, etc.) and $\tilde{\mathbf{h}}^{(i)}$ denotes the predicted latent state. The corresponding functional prediction is obtained by decoding,

$$\tilde{\mathbf{b}}_{t+1}^{(i)} = D_\theta(\tilde{\mathbf{h}}_{t+1}^{(i)}), \quad \tilde{\mathbf{x}}_{t+1}^{(i)} = \sum_{m=1}^{M^{(O)}} \tilde{\mathbf{b}}_{m,t+1}^{(i)} \phi_m^{(O)}(t).$$

The training objective combines reconstruction and forecasting:

$$\begin{aligned} \mathcal{L}_{\text{rec}} &= \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)}\|_2^2, \\ \mathcal{L}_{\text{for}} &= \frac{1}{NL} \sum_{i=1}^N \sum_{t=J+1}^{J+L} \|\mathbf{x}_t^{(i)} - \tilde{\mathbf{x}}_t^{(i)}\|_2^2, \end{aligned}$$

and the overall loss is

$$\mathcal{L}(\theta, \psi) = \mathcal{L}_{\text{rec}} + \alpha \mathcal{L}_{\text{for}}$$

One can add a smoothness, sparsity, or other types of penalty term to the objective function.

Joint Training Algorithm

Algorithm 1 Training FAE with Latent Time Series

```

1: Initialize  $\theta^{(0)}, \psi^{(0)}$ , set  $r=0$ .
2: repeat
3:   for  $i = 1, \dots, N$  do
4:     Encode and reconstruct first  $J$  points
5:     for  $t = 1, \dots, J$  do
6:        $\mathbf{h}_t^{(i,r)} = E_{\theta^{(r)}}(\mathbf{x}_t^{(i)})$ 
7:        $\hat{\mathbf{x}}_t^{(i,r)} = D_{\theta^{(r)}}(\mathbf{h}_t^{(i,r)})$ 
8:     end for
9:     Latent forecasting for  $t = J, \dots, J + L - 1$ 
10:    for  $t = J, \dots, J + L - 1$  do
11:       $\tilde{\mathbf{h}}_{t+1}^{(i,r)} = g_{\psi^{(r)}}(\mathbf{h}_1^{(i,r)}, \dots, \mathbf{h}_t^{(i,r)})$ 
12:       $\tilde{\mathbf{x}}_{t+1}^{(i,r)} = D_{\theta^{(r)}}(\tilde{\mathbf{h}}_{t+1}^{(i,r)})$ 
13:    end for
14:  end for
15:  Compute  $\mathcal{L}_{\text{rec}}^{(r)}$  and  $\mathcal{L}_{\text{for}}^{(r)}$  using the current  $\{\mathbf{x}_t^{(i)}, \hat{\mathbf{x}}_t^{(i,r)}, \tilde{\mathbf{x}}_t^{(i,r)}\}$ 
16:  Compute additional penalty terms, e.g. smoothness or sparsity
17:  Form total loss

$$\mathcal{L}^{(r)} = \mathcal{L}_{\text{rec}}^{(r)} + \alpha \mathcal{L}_{\text{for}}^{(r)} + (\text{penalty terms}).$$

18:  Update parameters

$$\theta^{(r+1)} = \theta^{(r)} - \eta_\theta \nabla_\theta \mathcal{L}^{(r)}, \quad \psi^{(r+1)} = \psi^{(r)} - \eta_\psi \nabla_\psi \mathcal{L}^{(r)}.$$

19:   $r \leftarrow r + 1$ 
20: until convergence

```

Forecasting

Given an observed sequence $\{\mathbf{x}_t^{(i)}\}_{t=1}^J$ for sample i :

Algorithm 2 Forecasting L Steps Ahead

1: Encode the first J observed curves:

$$\mathbf{h}_t^{(i)} = E_{\hat{\theta}}(\mathbf{x}_t^{(i)}), \quad t = 1, \dots, J.$$

2: **for** $s = 1, \dots, L$ **do**
3: $\tilde{\mathbf{h}}_{J+s}^{(i)} = g_{\hat{\psi}}(h_1^{(i)}, \dots, h_{J+s-1}^{(i)})$
4: $\tilde{\mathbf{x}}_{J+s}^{(i)} = D_{\hat{\theta}}(\tilde{\mathbf{h}}_{J+s}^{(i)})$
5: **end for**
6: Return $\{\tilde{\mathbf{x}}_{J+s}^{(i)}\}_{s=1}^L$.

