Generative ML for time series

Johan Vik Mathisen

February 22, 2024

0.1   Stage 1

We compare the latent representations learned from basic TimeVQVAE with a Barlow Twins extended version on downstream tasks as classification and reconstruction.

0.1.1   VQ-VAE

The VQ-VAE model is baseline for our experiments.

An encoder, decoder, and codebook are to be optimized by compressing the input into discrete latent space, minimizing information loss by comparing input to the output, which ideally are equal. We follow [**TimeVQVAE**] and augment time-series into time-frquency domain, but leave the high-low frequency split for future work.

A schematic overwiew of the VQ-VAE model is presented in "Figure here"

A time series is first augmented into time-frequency domain using the Short-time Fourier Transform (cite pytorch stft). Then it is encoded into the continious latent space, and is discretized by the codebook via the argmin process. In the argmin process the continious token is compared to every discrete token in the codebook, and replaced by the closes discrete token in terms of euclidean distance. Then, the decoder maps the discrete token back to time-frequency domain, before finally being mapped back to time domain using the ISTFT.

0.1.2   Barlow Twins VQ-VAE

An encoder, decoder, codebook, and projector are to be optimized.

Produce two augmented views of the time-series, augment views into time-frequency domain and encode into latent space. Choose one view for quantization, decoding and comparrison to original time series (VQVAE loss). Project both latent embeddings and calculate Barlow loss. Update using both VQVAE and Barlow loss.

A schematic overwiew of the BT-VQ-VAE model is presented in "Figure here"

Augmentations

We used the following collection of augmentation techniques throughout. Never dataset specific.

- Flip
- Jitter
- Amplitude Resizing
- Adding slope
- STFT Augmentation

Training

0.2 Stage 2