You’re your own best teacher: A Self-Supervised Learning Approach For Expressive Representations

Johan Vik Mathisen

May 19, 2024

Our work in this thesis can be seen as a tangent of the paper "Vector Quantized Time Series Generation with a Bidirectional Prior Model" [**TimeVQVAE**]. We simplify the model architecture by omitting the high-low frequency split, which reduces the model to what they refer to as "naive TimeVQVAE" in their paper. We expand on naive TimeVQVAE with a self-supervised extension.

A schematic figure of our proposed tokenization model is given in **??**. To improve on the reconstruction we add a regularizing term by reconstructing augmented views. We hypothesize that the model generalizes better to unseen data by letting the decoder "see" the augmented views.
To separate classes better we introduce a non contrastive self supervised loss. The intuition being that the representation of original and augmented views are pushed closer together by the SSL loss. We further enforce this hypothesis by using augmentations that preserve the overall semantics of the class conditional distributions.

0.1 Proposed model: NC-VQVAE

Our model, termed NC-VQVAE, is a generative time series model which learns expressive discrete latent representations by combining VQVAE [**VQVAE**] with non contrastive self supervised learning algorithms. NC-VQVAE uses the two staged modelling approach presented in [**TimeVQVAE**], and can be considered an extension of their "naive TimeVQVAE". Our model mainly extends the tokenization stage, where we incorporate Barlow Twins [**zbontar2021barlow**] and VIbCReg [**lee2024computer**] as our non contrastive SSL, but the framework is flexible. For stage two we model the prior using a bidirectional transformer as MaskGIT [**chang2022maskgit**].

0.1.1 Stage 1: NC-VQVAE

The architecture of the tokenization model **??** consists of two branches. The top and bottom branch is referred to as the original and augmented branch respectively. The model takes a time series x as input and creates an augmented view x'. The original branch is simply the naive TimeVQVAE from [**TimeVQVAE**], while the augmented branch is an autoencoder, constructed by omitting the quantization layer. The views are passed through their respective branches, where we compute the SSL loss derived from the original discrete latent representation z_q and the augmented continuous latent z', before the decoder reconstructs each latent representation.
The SSL loss is calculated by concatenate the average and max pool of both representations individually and pass the resulting vectors to the projector.

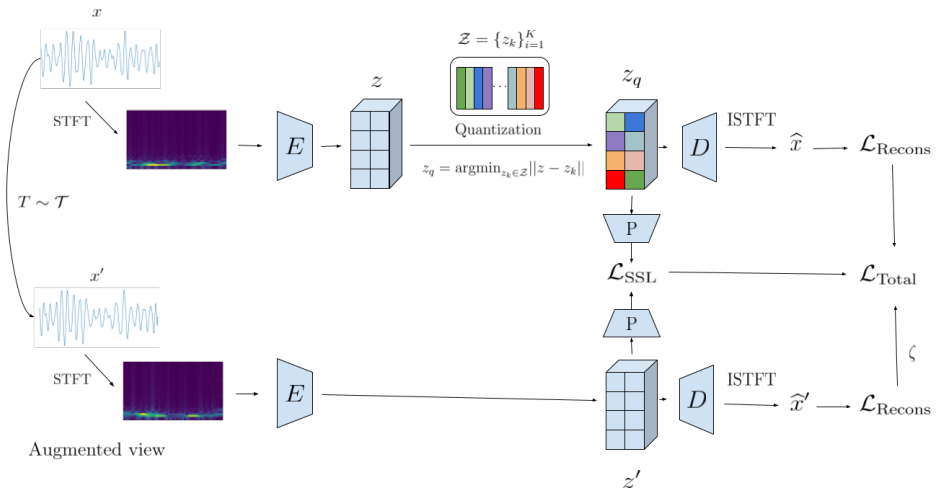

Figure 1: Overview of proposed model. NC-VQVAE

Implementation details

We follow [**TimeVQVAE**] closely in the encoder/decoder/codebook implementation, and

Time Frequency Modelling

$n_fft = 8$, other params are default parameters of pytorch implementation of STFT.

Encoder

Encoder block: Conv2d + Batchnorm + LeakyReLU
Convlayer has kernel size $(3, 4)$, stride $(1, 2)$ and padding $(1, 1)$ (padding mode = "replicate").

Residual block:LeakyReLU + Conv2d + LeakyReLU + Conv2d
First conv: kernel size=3, stride=1, padding=1
Second conv: kernel size=1, stride=1, padding=0

Decoder

Codebook

codebook: size = 32, dim = 64

Projector

0.1.2 Stage 2: Prior Learning

For prior learning we follow MaskGIT and refer to related works for a description of the learning and generation procedure. Token context MaskGIT, learnable codebook MaskGIT