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

Johan Vik Mathisen

May 9, 2024

TODO: relate our work to TimeVQVAE, Neural Representation, Barlow, MaskGIT,

TODO: Include something on time series generation / representation learning

Our work in this thesis can be seen as a tangent of the paper "Vector Quantized Time Series Generation with a Bidirectional Prior Model" [1]. The TimeVQVAE model is a two staged process. For this part of the thesis we focused on the first stage (tokenization) of the model. Meaning that we did not fit a prior on the latent space, keeping it uniform. Further we simplified the model by not separating the high and low frequency components of the data.

To our knowledge the joint embedding VQ-VAE models presented in this thesis are new.

0.1 TimeVQVAE

0.2 MaskGIT

The Masked Generative Image Transformer is a generative transformer model for image synthesis developed by Google Research. The novelty of the model lies in the token generation. Unlike popular autoregressive generative transformers, who treat images as a sequence of tokens, MaskGIT introduces an image synthesis paradigm using a bi-directional transformer decoder. This means that during training MaskGIT learns to predict tokens in all directions, an intuitively more natural way to consider images. At inference time MaskGIT starts out with a blank canvas and predicts the entire image, and iteratively keeps and conditions on the most confident pixels.

TODO: Intuitive introduction of masked modelling. Figures and such.

The model assumes a tokenization procedure for stage 1, and in the original paper they used VQGAN [2]. As MaskGIT only focuses on improving stage 2, present only that part.

0.2.1 Prior learning

Start out with a sequence s (b,n) of codebook indeces corresponding to a discrete latent representation z_q. Determine the proportion of tokens to mask according to the mask scheduling function $\gamma(t) \in (0, 1]$. Sample a random subset of s and replace values by [MASK] token in order to create the masked sequence s_M (b,n).

By a forward pass of the bi-directional transformer with s_M as input obtain unnormalized logits (b,n,K), defining a distribution over the codebook indeces at each element. Calculate the loss as the binary cross-entropy of the logits and s.

0.2.2 Iterative decoding

The bi-directional transformer could in principle predict all [MASK] tokens and generate a sample in a single pass by simply sampling from the logits obtained from a forward pass of an all masked sequence. However, there are challenges with this approach. In their original article [3] proposes a novel non-autoregressive decoding method to sythesize samples in a constant number of steps.

spør om siteringsstil, og fiks denne setningen

The decoding process goes from $t = 0$ to T. To genereate a sample at inference time one starts out with a all masked sequence which we denote by $s_M^{(0)}$. At iteration t the model predicts the probabilities for all the [MASK] tokens, $p(\hat{s}_{ij}^{(t)}|s_M^{(t)})$, in parallell. Then at each masked entry ij we sample a token index based on its predicted distribution.

0.2.3 Masking design

For image generation, cosine schedulig function proved best across all experiments in the original paper. Start out by selecing just a few

0.3 SSL

Our model leverages SSL algorithms in order to learn more expressive latent representations. Here we present the relevant algorithms for our work.

0.3.1 Barlow Twins

What is it?

Barlow Twins is a non-constrastive SSL method based on applying the *redundancy-reduction principle* (or efficient coding hypothesis) [5] from the neroscientist H. Barlow to a pair of identical networks.

In essence the models encurage representations of similar samples to be similar, while simultaneously reducing the amount of redundancy between the components of the vectors. This is done by producing two distorted views of each sample and embedding these in a vast feature space, in such a way that their cross-correalion is close to the identity.

How does it work?

Start out with a sample X and creates two augmented (distorted) views X_1 and X_2. The views are then mapped to a latent space by two identical encoders, giving Y_1 and Y_2. Then the projector embeds the latent representations in a vast space,

giving Z_1 and Z_2. Finally the similarity of the two embeddings are measured by the empirical cross-corelation.

TODO: Ask for premission?? to use this or make own

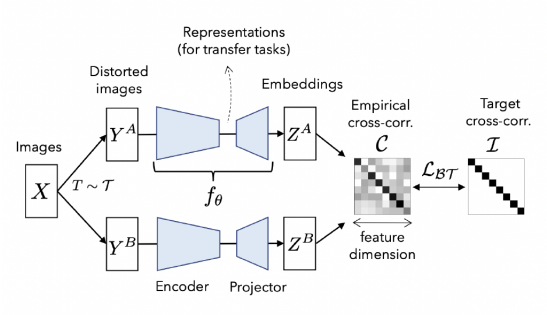

Figure 1: [6]

The loss function is calculated as the difference of the empirical cross-correlations of Z_1 and Z_2 is then calculated and the identiry matrix.

0.3.2 VIbCReg

TODO: What is it?

VIbCReg [7] is a non-contrastive SSL model based on VICReg [8], but with better covariance regularization. It has a joint embedding architecture.

TODO: How it works

Two different views of the input data is encoded into representations Y Y'. The representations are further mapped to a larger space by a *projector* with an IterNorm [9] layer. The loss is computed using the projected values Z and Z'.
The loss consists of a similarity loss between the branches, and feature decoration (FD) loss together with a feature component expressiveness (FcE) term at each branch.

TODO: Loss

Input data is processed in batches. Denote $Z = [z_1, ..., z_B]^T \in \mathbb{R}^{B \times F}$, and similarly for Z', where B and F denotes the batch and feature sizes respectively. Var() is a variance estimator, γ is a target value for the standard deviation, which both in VIbCReg and VICReg is set to 1. ϵ is a small scalar preventing numerical instabilities.
Similarity loss

$$s(Z, Z') = \frac{1}{B} \sum_{b=1}^{B} ||Z_b - Z'_b||_2^2 \tag{1}$$

FcE/Variance term
FD/covariance term

$$C(Z) = \frac{1}{B-1}\left(\frac{Z-\bar{Z}}{||Z-\bar{Z}||_2}\right)^T\left(\frac{Z-\bar{Z}}{||Z-\bar{Z}||_2}\right) \text{ where } \bar{Z} = \sum_{b=1}^{B} Z_b \qquad (2)$$