

Pattern Analysis

Pattern Analysis of various datasets by COMP3710 students in 2024 at the University of Queensland.

We create pattern recognition and image processing library for Tensorflow (TF), PyTorch or JAX.

This library is created and maintained by The University of Queensland [COMP3710 \(https://my.uq.edu.au/programs-courses/course.html?course_code=comp3710\)](https://my.uq.edu.au/programs-courses/course.html?course_code=comp3710) students.

The library includes the following implemented in Tensorflow:

- fractals
- recognition problems

In the recognition folder, you will find many recognition problems solved including:

- segmentation
- classification
- graph neural networks
- StyleGAN
- Stable diffusion
- transformers etc.

VQ-VAE for Medical Image Generation

Description

This repository implements a Vector Quantized Variational Autoencoder (VQ-VAE) model to generate medical images from 2D slices of prostate MRI scans. The VQ-VAE is a generative model that combines the principles of variational inference with vector quantization, allowing for a discrete representation that improves the quality of image reconstructions. The goal of this model is to map MRI scans of Prostate Cancer into a lower-dimensional latent space and decode the quantized representation back to the original image, providing a method for generating clear MRI images with improved similarity to real medical data.

Problem

The challenge this VQ-VAE addresses is generating clear medical images from noisy data while maintaining high similarity to the original MRI scans. Specifically, the model is trained on the HipMRI Study dataset containing 2D prostate MRI slices and attempts to reconstruct these images with a focus on structural similarity.

How It Works

The VQ-VAE model works by encoding input images, which is then quantized to discrete vectors. This quantized space is then decoded to reconstruct the original image. The key components include:

1. Encoder: Converts input images into a lower-dimensional latent representation using convolutional layers.
2. Vector Quantizer: Discretizes the latent space by mapping the continuous latent vectors to the nearest embedding in the codebook.
3. Decoder: Reconstructs the input image from the quantized latent vectors using transposed convolution layers.
4. Loss Function: The model optimizes a combination of reconstruction loss (MSE) and a commitment loss to ensure the latent vectors stay close to the embeddings.

A high-level flow:

- Input MRI image -> Encoder -> Latent Space
- Latent Space -> Vector Quantizer -> Quantized Latent Space
- Quantized Latent Space -> Decoder -> Reconstructed MRI image

Dependencies

To install all dependencies, run: `pip install torch torchvision nibabel matplotlib tqdm pathlib`

Usage

1. Preprocessing: The .nii MRI image files are sliced into 2D images and resized. Normalization to the range [0, 1] is applied
2. Training: The dataset is split into 70% for training, 15% for validation, and 15% for testing. This split is chosen to ensure a sufficiently large training set while keeping a balanced validation and test set for hyperparameter tuning and performance evaluation.
3. To Run the training on current variables and epochs, while saving the model to reconstruct the images later

- `python train.py -save`

4. To reconstruct the images after training and save the reconstructed images (top 5)

- `python predict.py -save`

Results

Validation SSIM

The model achieves a structural similarity index (SSIM) of 0.75 averaged over 5 images, indicating good image quality and resemblance of the input MRI Scans

Working Principles

Algorithm:

- The VQ-VAE model solves the problem of medical image generation by learning discrete latent representations and creates high-quality MRI reconstructions

Comments:

- All key components and methods in the modules.py file and all other files are well-commented and structures to explain the core logic behind the model

Formatting:

- The README is properly formatted using GitHub markdown

Examples:

- Examples of the model's reconstructed images can be found at ./results

Author

Harrison Cleland, 2024 47433386