CMPE 260 – Reinforcement Learning (Spring 2023)

Homework1

Submitted by: Kavan Vipinbhai Soni (016085966)

---------------------------------------------------------------------------------------------------------------------

Updated code at : <https://github.com/kavan-soni/252_260/tree/main/260_HW1>

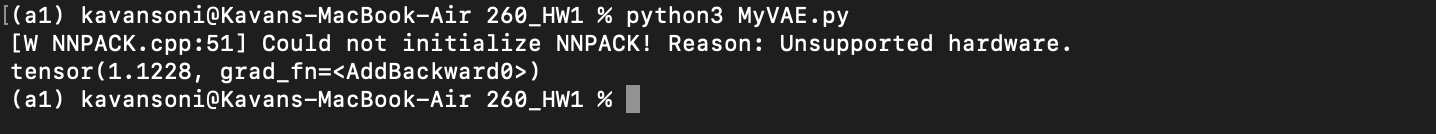
Activity 1 :

Finish the `\_\_init\_\_()` in `MyVAE.py` model.

A picture containing graphical user interface

Description automatically generated

Run `MyVAE.py` to quickly test if your model is working



Run `train\_vae.py` to train.

Text

Description automatically generated

Table

Description automatically generated with low confidence

Run `sample\_vae.py` to generate a few images with your model. Save 2 generated images.

Chart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generated

What model components are used in the forward pass and in sampling?

* Following components are used in forward pass and sampling:
* Encoder
* Latent space
* Decoder
* Image sampling

Activity 2:

Upgrade to VAE by modifying `forward()`, `encode()`, and `reparameterize()`.

Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Train and save two generated images.

Table

Description automatically generated with medium confidenceChart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generated

Describe the difference between the AE and VAE models.

* AE and VAE are both neural networks used for unsupervised learning and dimensionality reduction. They differ in following ways:
* VAE is generative model while AE is not.
* Latent space of VAE is continuous and structured, while it is unstructured for AE.
* AE do not use regularization, while VAE uses Kullback-Leibler divergence for regularization of latent space.
* AE only use reconstruction loss, while VAE uses reconstruction loss and KL divergence to assess how well the model can reconstruct the input data.

What is the reparameterization trick?

* The reparameterization trick is a technique used in Variational Autoencoder (VAE) models to sample from a probability distribution in a differentiable way. The reparameterization trick enables the gradient-based optimization of VAE models, which makes it possible to train these models using backpropagation and stochastic gradient descent.
* In a VAE, the goal is to learn a latent space that follows a specific probability distribution, typically a normal distribution. During training, the VAE learns the parameters of this distribution, namely the mean and variance, from the input data. To generate new samples from this distribution, the VAE must sample from it.
* However, sampling from a probability distribution is a non-differentiable operation, which makes it difficult to use gradient-based optimization techniques for training the VAE. The reparameterization trick solves this problem by separating the sampling operation from the parameters of the distribution.
* Specifically, the trick involves introducing a noise variable epsilon that is drawn from a fixed standard normal distribution. The mean and variance parameters learned by the VAE are then used to transform this noise variable into a sample from the desired distribution, as follows:

z = mu + sigma \* epsilon

* This transformation is differentiable with respect to the mean and standard deviation, which allows for the gradient-based optimization of the VAE during training.
* In summary, the reparameterization trick is a technique used in VAE models to enable the differentiable sampling from a probability distribution, which is required for gradient-based optimization during training.

Activity 3:

Update the `train\_vae.py` to reset the environment after the first 20 observations from each episode.

Graphical user interface, text, application

Description automatically generated

Train and save two generated images.

Table

Description automatically generated with medium confidence

Chart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generated

Activity 4:

update the `train\_vae.py` train vae on observations with a custom angle range.

Graphical user interface, text

Description automatically generated

Train and save two generated images.

Table

Description automatically generated

Chart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generated

Activity 5:

pick [some other gym environment] and train vae on it.

Graphical user interface, text, application

Description automatically generated

Train and save two generated images.

Background pattern

Description automatically generated

Chart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generated