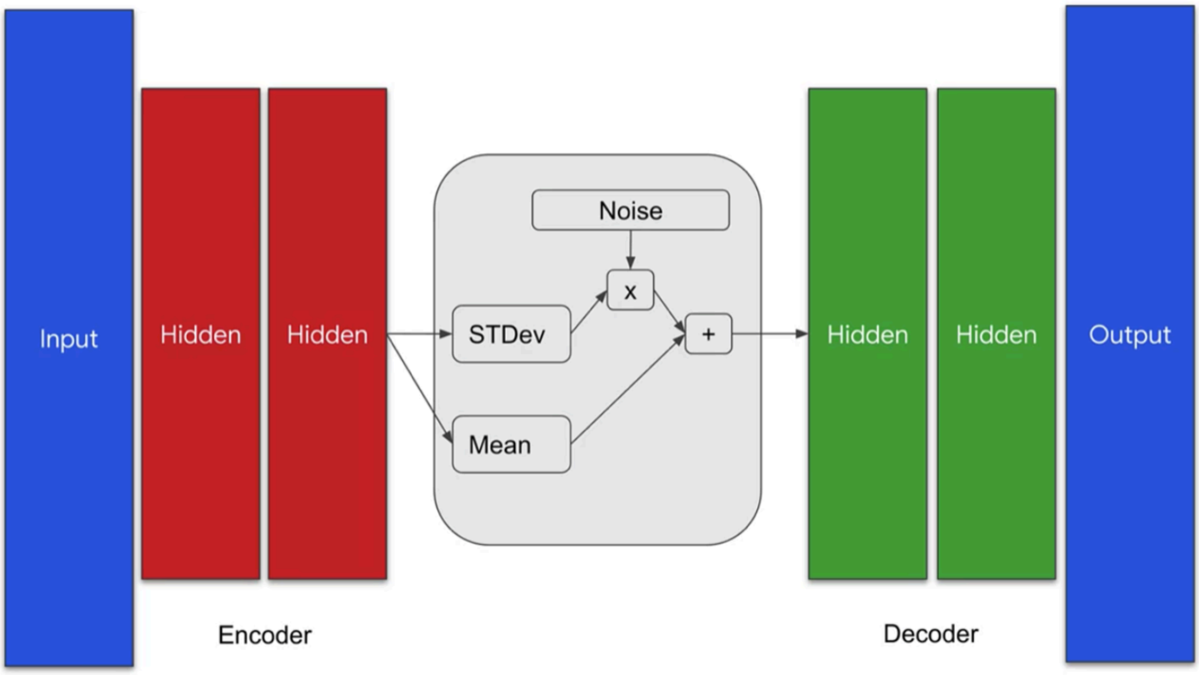
Explain the hyperparameters that you eventually used, and narrate the training experience pointwise and the analysis of the final results. Suggest in what ways the error in the model be improved.

Strategy in nutshell



GIT LINK:- https://github.com/Aarav1666/dl

Hyper Parameters used : Learning Rate : 0.001

The learning rate defines how quickly a network updates its parameters.

Observation during training
Low learning rate slows down the learning process but converges smoothly. Larger learning rate speeds up the learning but may not converge.
Usually a decaying Learning rate is preferred.

Number of Epochs : 100

Number of epochs is the number of times the whole training data is shown to the network while training.

Observation during training

Increase the number of epochs until the validation accuracy starts decreasing even when training accuracy is increasing(overfitting).

Batch Size : 200

Mini batch size is the number of sub samples given to the network after which parameter update happens.
A good default for batch size might be 32. Also try 32, 64, 128, 256, and so on.

For my dataset it was 200

During the learning process at first I started by taking 800 images each category 80 and epoch 50, batch size 200 and learning rate 0.07

I observed that the regeneration of images was not that good

After that I increased the datasets to 1200 images each category 120 and epoch 100, batch size 200 and learning rate 0.003

And didn’t get much difference

Then I took with Larger datasets like 5000, 8000 and so on because I noticed although the learning rate was decreased due to less supply of dataset in each category Newly generated images were not at all visible

Finally at the end when I set the Learning rate 0.001 Epoch 100 and batch size 100
Almost 90 percent of regenerating images were similar to real ones

After this I started generating new Image using the variational AutoEncoder and the learning model
And got few Good Sample Images out of 750 images

Generated Images
8x8 grid of 12 sets . Total 768 images .

the custom layer to provide the Gaussian noise input along with the mean (mu) and standard deviation (sigma) of the encoder's output. Recall the equation to combine these:

$z = \mu + e^{0.5\sigma} * \epsilon$

where μ = mean, σ = standard deviation, and ϵ = random sample

The general steps are:

- feed a training batch to the VAE model
- compute the reconstruction loss (hint: use the mse_loss defined above instead of bce_loss in the ungraded lab, then multiply by the flattened dimensions of the image
- add the KLD regularization loss to the total loss (you can access the losses property of the vae model)
- get the gradients
- use the optimizer to update the weights

How to improve the model ?

Learning rate (LR):

- Perform a learning rate range test to identify a “large” learning rate.
- Using the 1-cycle LR policy with a maximum learning rate determined from an LR range test, set a minimum learning rate as a tenth of the maximum.

Weight decay:

- A grid search to determine the proper magnitude but usually does not require more than one significant figure accuracy.

Batch Size:

- Use as large batch size as possible to fit your memory then you compare performance of different batch sizes.
- Small batch sizes add regularization while large batch sizes add less, so utilize this while balancing the proper amount of regularization.
- It is often better to use a larger batch size so a larger learning rate can be used.