

# Assignment 4\_proj

March 2, 2021

## 1 Deep Learning Course

### 1.1 Assignment Four

#### 1.1.1 Assignment Goals:

- Implementing Fully Connected AutoEncoders
- Implement naive generative model
- Understand VAE and GAN, then implement a classical generative model: VAE-GAN.

#### 1.1.2 DataSet

In this Assignment, you will use the Fashion-MNIST dataset. The dataset is not given in the assignment package, please download/load by yourself. *Hint:* You can use

```
(x_train, _), (x_test, _) = keras.datasets.fashion_mnist.load_data()
```

to load the dataset.

#### 1.1.3 Requirements

##### 1. (20 points) Implement a Fully Connected AutoEncoder

- Your AutoEncoder should have a bottleneck with two neurons and use Mean Squared Error (MSE) as the objective function. Design the model structure by yourself. Notice that in an AutoEncoder, the layer with the least number of neurons is referred to as a bottleneck.
- Train your model on MNIST. Plot the train and test loss.
- Randomly select 10 images from the test set, encode the selected 10 images, visualize the original images and the decoded images.

##### 2. (30 points) Naive generative model

This question is about using an AutoEncoder to generate similar but not identical hand digits. We use a naive approach: Try to see if a trained decoder can map randomly generated inputs (random numbers) to a recognizable hand-written digit.

1. Start with your Fully Connected AutoEncoder from part 1. Try to generate new images by inputting some random numbers to the decoder (i.e. the bottleneck layer). Visualize 10 generated images. (10 points)

2. Now restrict each neuron of the bottleneck layer to have a distribution with mean zeroes and variance one (i.e.  $N(0,1)$ ). Retrain the Fully Connected AutoEncoder with the normalized bottleneck. Now randomly generate inputs to the bottleneck layer that are drawn from the multi-variate standard normal distribution, and use the randomly generated inputs to generate new images. Visualize 10 generated images. (15 points)
3. Are the output images different between A) and B)? If so, why do you think this difference occurs? (5 points)

### 3. (50 points) Advanced generative model

In this part, you are asking to implement a VAE-GAN model. A VAE-GAN is a Generative Adversarial Network whose generator is an Variational Autoencoder. Here is the paper which proposed the VAE-GAN: [\[PAPER\]](#). You may need to read this paper before implementing this model.

1. Implement a Variational Autoencoder based on your Fully Connected AutoEncoder from part 1. Use your VAE to randomly generate 10 images. Does the VAE produce a different quality of output image? (30 points)
2. Implement a VAE-GAN based on your implemented VAE. Train the VAE-GAN. (20 points)
  - Then use your VAE-GAN to randomly generate 10 images from  $p(z)$ .
  - Randomly select 10 images from the test dataset and reconstruct them using your model, then visualize the original and reconstructed images.

*Hint:* For the generation and reconstruction tasks, refer to Section 4.1 in the paper.

#### 1.1.4 Submission Notes:

Please use Jupyter Notebook. The notebook should include the final code, results, and answers. You should submit your Notebook in .pdf and .ipynb format. (penalty 10 points).

Notice that your AutoEncoders should have only one bottleneck.

#### Instructions:

The university policy on academic dishonesty and plagiarism (cheating) will be taken very seriously in this course. Everything submitted should be your writing or coding. You must not let other students copy your work. Spelling and grammar count.

## 1.2 Your implementation

[ ]:

[ ]:

[ ]: