

ASSIGNMENT: 4

Assignment Document

edureka!

edureka!

© Brain4ce Education Solutions Pvt. Ltd.

Scenario – 1

Problem Statement:

The dataset is like MNIST but includes images of certain clothing and accessory. The objective is to classify images into specific classes.

Total Images: 70,000

Train Images: 60,000

Test Images: 10,000

Image size: 28x28

Classes (In order): ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle Boot']

Homepage: <https://github.com/zalandoresearch/fashion-mnist>

Create Generator and Discriminator function for Fashion MNIST Data

Question 1:

Define Generator model with following architecture:

HINT:

```
nch = 200
g_input = Input(shape=[100])
H = Dense(nch*14*14, init='glorot_normal')(g_input)
H = BatchNormalization()(H)
H = Activation('relu')(H)
H = Reshape( [nch, 14, 14] )(H)
H = UpSampling2D(size=(2, 2))(H)
```

Layer1:

- Conv2d with 100 filter, 3*3
- Batch Normalization

- Activation Relu

Layer 2:

- Conv2d with nch/4 filter, 3*3
- Batch Normalization
- Activation Relu

Layer 3:

- Conv2d with 1 filter, 1*1
- Activation Sigmoid

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 100)	0
dense_1 (Dense)	(None, 39200)	3959200
batch_normalization_1 (Batch)	(None, 39200)	156800
activation_1 (Activation)	(None, 39200)	0
reshape_1 (Reshape)	(None, 200, 14, 14)	0
up_sampling2d_1 (UpSampling2)	(None, 200, 28, 28)	0
conv2d_1 (Conv2D)	(None, 100, 28, 28)	180100
batch_normalization_2 (Batch)	(None, 100, 28, 28)	112
activation_2 (Activation)	(None, 100, 28, 28)	0
conv2d_2 (Conv2D)	(None, 50, 28, 28)	45050
batch_normalization_3 (Batch)	(None, 50, 28, 28)	112
activation_3 (Activation)	(None, 50, 28, 28)	0
conv2d_3 (Conv2D)	(None, 1, 28, 28)	51
activation_4 (Activation)	(None, 1, 28, 28)	0
<hr/>		
Total params:	4,341,425	
Trainable params:	4,262,913	
Non-trainable params:	78,512	

Question 2:

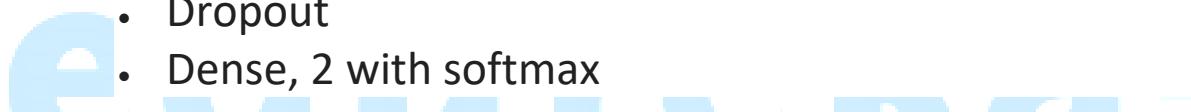
Define Discriminator model with following architecture:

HINT:

```
shp = X_train.shape[1:]
d_input = Input(shape=shp)
H = Convolution2D(256, 5, 5, subsample=(2, 2), border_mode = 'same',
activation='relu')(d_input)
H = LeakyReLU(0.2)(H)
H = Dropout(dropout_rate)(H)
```

Layer2:

- Conv2d with 512 filters, 5*5 and subsample (2,2)
- LeakyRelu, 0.2
- Dropout
- Flatten
- Dense 256
- LeakyRelu, 0.2
- Dropout
- Dense, 2 with softmax



Layer (type)	Output Shape	Param #
input_2 (InputLayer)	(None, 1, 28, 28)	0
conv2d_4 (Conv2D)	(None, 256, 14, 14)	6656
leaky_re_lu_1 (LeakyReLU)	(None, 256, 14, 14)	0
dropout_1 (Dropout)	(None, 256, 14, 14)	0
conv2d_5 (Conv2D)	(None, 512, 7, 7)	3277312
leaky_re_lu_2 (LeakyReLU)	(None, 512, 7, 7)	0
dropout_2 (Dropout)	(None, 512, 7, 7)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_2 (Dense)	(None, 256)	6422784
leaky_re_lu_3 (LeakyReLU)	(None, 256)	0
dropout_3 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 2)	514

```
=====
Total params: 9,707,266
Trainable params: 9,707,266
Non-trainable params: 0
```

edureka!