```
import
import
import          as
import                        as
from                         import

import
from       import
print "berkay"
           50
             15
           10
                      'cuda:0' if                            else 'cpu'
# customize Transform variable is to take input and return a tensor object
# Also by setting normalizer, I scaled pixel values between -1 and 1
#directly taken from https://medium.com/@aaysbt/fashion-mnist-data-training-using-pytorch-7f6ad
                                                                               0 5    0 5

# training set
                                               './data'          True             False

#Splitting data
#https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
                                                            0 1               42

# test set
                                              './data'          False


                                                                                        False
print "ipek"

# example mlp classifier
class mlp1
    def __init__ self
        super       self
        self
        self
        self
        self
    def forward self
                    1  self
                 self
               self
                 self
        return
# example mlp2 classifier
class mlp2
    def __init__ self
        super       self
        self
        self
        self                                                      False
        self
        self
    def forward self
                    1  self
                 self
               self
                  self
                 self
        return
```

```
class cnn_3
    #Layer Definition
    #https://pyimagesearch.com/2021/07/19/pytorch-training-your-first-convolutional-neural-netwo
    def __init__ self
        super        self
        self
        #in_channel = input_size m1 0 m1 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
        self                                   1               16              3         1
        self
        self                                   16               8              7         1
        self
        self                                            2  2          2         0
        self                                   8               16              5         1
        self                                            2 2          2         0
        #https://stackoverflow.com/questions/53580088/calculate-the-output-size-in-convolution-
        self                                   144
    def forward self
        #It didin't work ??????
        #x = x.view(-1, self.input_size)
                  self
                self
                  self
                self
                self
                  self
                self
        #Reshaping linear input
                                    144
                 self
        return


class cnn_4
    #Layer Definition
    #https://pyimagesearch.com/2021/07/19/pytorch-training-your-first-convolutional-neural-netwo
    def __init__ self
        super        self
        self
        self                                   1               16              3         1
        self
        self                                   16               8              5         1
        self
        self                                   8               8              3         1
        self
        self                                            2  2           2
        self                                   8               16              5         1
        self
        self                                            2  2           2
        self                                   144
    def forward self
        #x = x.view(-1, self.input_size)
        #It didin't work ??????
                  self
                self
                  self
                self
                  self
                self
                self
                  self
                self
                self
                         50 144
        #Reshaping linear input
                 self
        return
class cnn_5
    #Layer Definition
    #https://pyimagesearch.com/2021/07/19/pytorch-training-your-first-convolutional-neural-netwo
    def __init__ self
        super        self
        self
        self                                       1              16              3         1
        self
        self                                       16              8              3         1
        self
        self                                       8              8              3         1
        self
        self                                       8              8              3         1
        self
        self                                                2  2           2
        self                                       8              16              3         1
        self
        self                                       16              16              3         1
        self
        self                                                2  2           2
        self                                       144
    def forward self
        #x = x.view(-1, self.input_size)
        #It didin't work ??????
                  self
                self
                  self
                self
                  self
                self
                  self
                self
                self
        #pool4=pool4.view()
                  self
                self
                  self
                self
                self
        #Reshaping linear input
                         50 144
                 self
        return
#Dictionary for json

    'name'  None
    'loss_curve' None
    'train_acc_curve' None
    'val_acc_curve' None
    'test_acc' None
    'weights' None

#Model Types
        'cnn_4'
# models=['mlp_1','mlp_2','cnn_3','cnn_4','cnn_5']
for                in
```

```
print f"Training is started for model {modelselected}"



            None
                   0
for       in range
    print f"Step {stepX+1} is started"
    if                  'mlp_1'
                         784 64 10
    elif                  'mlp_2'
                        784 16 64 10
    elif                  'cnn_3'
                          784 10
    elif                  'cnn_4'
                          784 10
    elif                  'cnn_5'
                          784 10

                                                              0 01

    #Recorded values for each try




    for       in range
        print f"Epoch is {epoch+1}/{epoch_size}"
              0
                0

                     len
        #https://stackoverflow.com/questions/62833157/cnn-model-using-pytorch
        #Train DATA
        #https://androidkt.com/calculate-total-loss-and-accuracy-at-every-epoch-and-plot-usi
        for                     in enumerate

            # Move tensors to the configured device


            # Forward pass



            # Backward and optimize





            if   +1    10    0

                #Train calculation
                #Directly taken from https://discuss.pytorch.org/t/how-does-one-get-the-pred
                                        1
                                       0


                                  100

                          0
```

```
                            0
                #Valid
                for                             in enumerate
                    # Accuracy Calculation
                    # https://androidkt.com/calculate-total-loss-and-accuracy-at-every-epoch


                    # Forward pass

                                                       1
                                                 0

                                                  100
                #print ('Step {} Epoch [{}/{}], Step [{}/{}], Loss: {:.4f}, Train Accuracy
                #       .format(stepX,epoch+1, epoch_size, i+1, total_step, running_loss,tra






    #print(train_losses_total)

    #Test
    with


                  0
                0
        for                in

 

                                       1
                                 0

                                   100

        if

                         'cpu'



    print 'For Step {} It is finished'            +1
#https://www.geeksforgeeks.org/python-column-wise-sum-of-nested-list/
                        sum             len           for          in zip
                      sum             len           for          in zip
                      sum             len           for          in zip


#Dictionary for json

     'name'
     'loss_curve'
     'train_acc_curve'
    'val_acc_curve'
    'test_acc'
    'weights'                      #https://stackoverflow.com/questions/26646362/numpy-array

#JSON Writing a file (geeksforgeeks.com)
```

```
with open "DUMMY_q2_"               ".json" "w"  as


     'name'
     'train_losses_total'
     'train_accus_total'
     'valid_accus_total'
     'test_acc'
     'weights'                      #https://stackoverflow.com/questions/26646362/numpy-arra

#JSON Writing a file (geeksforgeeks.com)
with open "am_"               ".json" "w"  as

                                       'resultQ2'           'input_weights_'
```