ID1: 208560086
ID2: 301613501

TEL AVIV UNIVERSITY

EX1 - DL Basics

1. **General**
   This readme file describes how to run the code in the notebook of exercise 1.
   The code is written such that we train all the cases and show the required graphs on a *Tensorboard*.

2. **Run the code**
   Prior to the training or loading weight step, please run the following steps \ functions \ cells:
   - Cell 1: Mount to google drive.
   - Cell 2: Import libraries.
   - Cell 3: Set the device that PyTorch will use for computation based on the available hardware resources.
   - Cell 4: Load Tensorboard extension.
   - Cell 5: Transform the data into a PyTorch tensor.
   - Cell 6: Initializes the train_dataset and test_dataset, using the Fashion-MNIST dataset.
   - Cell 7: Implementation of LeNet-5 and get_optimizer method
   - Cell 8: Run the train_one_ephoc function
   - Cell 9: Run the evaluate the function
   - Cell10: Run the get_model function
   - Cell 11: Run the train_lenet5 function
   - Cell 12: Run the test_lenet5 function

3. **How to train**
   Run the *train_lenet5* function as shown in the following table.

Regularization	Command
Without Regularization	train_lenet5(batch_size=64, initial_lr=0.01, num_epochs=30, batch_norm_enabled=False, dropout_enabled=False, weight_decay_enabled=False, weight_decay=0.0001)
Batch Normalization	train_lenet5(batch_size=64, initial_lr=0.01, num_epochs=30, batch_norm_enabled=True, dropout_enabled=False, weight_decay_enabled=False,weight_decay=0.0001)
Dropout	train_lenet5(batch_size=64, initial_lr=0.01, num_epochs=30, batch_norm_enabled=False, dropout_enabled=True, weight_decay_enabled=False,weight_decay=0.0001)
Weight Decay	train_lenet5(batch_size=64, initial_lr=0.01, num_epochs=30, batch_norm_enabled=False, dropout_enabled=False, weight_decay_enabled=True,weight_decay=0.0001)

Function Description

Trains a LeNet-5 model on a given dataset and returns the final training and testing accuracies.
Args:
- batch_size (int): the number of samples per batch to load in the data loaders.
- initial_lr (float): the initial learning rate to use for the optimizer.
- num_epochs (int): the number of epochs to train the model for.
- batch_norm_enabled (bool): whether to enable batch normalization in the model.
- dropout_enabled (bool): whether to enable dropout in the model.
- weight_decay_enabled (bool): whether to enable weight decay in the model.
- weight_decay (float): the weight decay value to use for the optimizer.
- weights_path (str): (optional) the file path to save the trained model weights to. If no value is provided then wights_path will be assigned with 'lenet_5_bn{}_dp{}_wd{}_weights.pth' when {} will be true or false according to the function values.

Returns:
- train_acc (float): the final training accuracy of the model on the train set.
- test_acc (float): the final testing accuracy of the model on the test set.

4. **How to test with saved weights**

Run the *test_lenet5* function as shown in the following table.

Regularization	Command
Without Regularization	`test_lenet5(batch_size=64, batch_norm_enabled=False, dropout_enabled=False, weight_decay_enabled=False,weight_decay=0.0001,weights_path ='lenet5_bnFalse_dpFalse_wdFalse_weights.pth')`
Batch Normalization	`test_lenet5(batch_size=64, batch_norm_enabled=False, dropout_enabled=False, weight_decay_enabled=False,weight_decay=0.0001,weights_path ='lenet5_bnTrue_dpFalse_wdFalse_weights.pth')`
Dropout	`test_lenet5(batch_size=64, batch_norm_enabled=False, dropout_enabled=False, weight_decay_enabled=False,weight_decay=0.0001,weights_path ='lenet5_bnFalse_dpFalse_wdTrue_weights.pth')`
Weight Decay	`test_lenet5(batch_size=64, batch_norm_enabled=False, dropout_enabled=False, weight_decay_enabled=False,weight_decay=0.0001,weights_path ='lenet5_bnFalse_dpFalse_wdTrue_weights.pth')`

Function Description

Test a LeNet-5 model on a given dataset and return the final training and testing accuracies on a pre-trained model.
Args:
- batch_size (int): the number of samples per batch to load in the data loaders.
- batch_norm_enabled (bool): whether to enable batch normalization in the model.
- dropout_enabled (bool): whether to enable dropout in the model.
- weight_decay_enabled (bool): whether to enable weight decay in the model.
- weight_decay (float): the weight decay value to use for the optimizer.
- weights_path (str): the file path to load the trained model weights to.

Returns:
- train_acc (float): the final testing accuracy of the model on the train set.
- test_acc (float): the final testing accuracy of the model on the test set.

5. By running Cell 20 you can see a **table of the summary**

	Train Accuracy [%]	Test Accuracy [%]
base	93.53	90.32
batch normalization	97.39	91.93
dropout	92.11	89.67
weight decay	94.30	91.07

6. By running Cell 21 (`%tensorboard --logdir logs`) you can see the Tensorboard visualization of all the graphs. Then follow the illustration below in order to choose the relevant graph\s you want to see. It is possible to compare all the train-test combinations.

7. For example - if we want to compare the test and the train of batch normalization we will choose test/lenet_5_**bnTrue**… and train/lenet5_**bnTrue**.