EEE7331-01

SPECIAL TOPICS IN DEEP LEARNIING

2022 FALL

Assignment 1

Due Date: 30th of October before midnight to LearnUs.

INSTRUCTIONS:

1. This question paper consists of 6 pages.
2. Attempt ALL questions.
3. This is an individual-based assignment. Discussions among students are welcome. However, any cheating attempt will be penalized.
4. Attach your code as an appendix. You may show code snippets to aid your explanation.
5. Upload the solution in a single pdf file.
6. Use English in the report.

Question 1 (50%):

Part A: CIFAR 10 Classification with a customized CNN

We will use the CIFAR-10 Dataset (https://www.cs.toronto.edu/~kriz/cifar.html). You can find the tutorial according to the major deep learning library. Most major deep learning libraries come with tutorials (TensorFlow, PyTorch, MxNet, MATLAB etc) with CIFAR10 examples.

Create a network with the following specifications and train it with CIFAR-10:

	Stage 1		**Stage 2**		**Stage 3**		**Stage 4**	**Stage 5**
Layer	**Conv**	**Pool**	**Conv**	**Pool**	**Conv**	**Pool**	**FC**	**Output**
	$(5,32)_{/1,1}$ RELU	$2_{/2,0}$	$(3,64)_{/1,1}$ RELU	$2_{/2,0}$	$(3,128)_{/1,1}$ RELU	$2_{/2,0}$	500	10, softmax

where $(m, c)_{/a,b}$ with c filters of size $m \times m$, where the stride and padding are a and b, respectively. $p_{/r,s}$ denotes the max-pooling layers with a window of $p \times p$, where the stride and padding are r and s, respectively.

Fill in the table below based on the above information:

Layer	**#of filters/ filtersize(Window) /Stride/ZeroPadding**	**Feature Map size**	**#of weights (including bias)**	**# of Neurons**
Input	-	32x32x3	-	-
Conv1-1	32/5x5/1/1	30x30x32		
Pool-1				
Conv2-1				
Pool-2				
…	…	…	…	…
Output				
Total				

0. Describe the deep learning library you use in this assignment, e.g., name, version number, etc.

1. Describe your training setup, e.g., data augmentation, parameters such as learning rate schedule, momentum, dropout rate, batch number, batch normalization etc. Present them systematically in *table form*.

2. Plot the following:

(a) Training and test loss vs. epoch.

(b) Classification accuracy on the training and test set vs. epoch.

3. Your testing accuracy should be more than 70%. Show the final training and testing accuracy you obtain from the experiments.

Part B: CNN with Depth-wise Separable Convolution

Based on the CNN specified in part A, replace the convolution layer at stages 2 and 3 with *depth-wise separable convolution*. Note depth-wise separable convolution composed of **depth-wise convolution** and **point-wise convolution**.

Fill in the table below:

Layer	**#of filters/ filtersize(Window) /Stride/ZeroPadding**	**Feature Map size**	**#of weights (including bias)**	**# of Neurons (Memory)**
Input	-	32x32x3	-	-
Conv1-1	32/5x5/1/1	30x30x32		
…	…	…	…	…
Output				
Total				

1. Train the modified network with a similar configuration (data augmentation, momentum, dropout rate, batch number, batch normalization) as in the original network. However, the learning rate schedule can be changed according to the network convergence circumstance.

2. Plot the following:

(a) Training and test loss vs. epoch.

(b) Classification accuracy on the training and test set vs. epoch.

3. Fill in the table:

	Total Weight Numbers (including bias)	**Training Accuracy (%)**	**Testing Accuracy (%)**
CNN at Part A			
CNN with depth-wise separable convolution			

4. Discuss your results, and compare the original CNN (Part A) with Part B.

Question 2: StyleGAN Inversion (50%):

In this question, you will follow the provided tutorial (EEE7331_GAN_Inversion.ipynb) to invert face images into StyleGAN latent space. The tutorial uses **Pytorch**, and works with **Google Colaboratory** by default.

Since there are many version issues regarding StyleGAN, running the ipynb notebook in Google Colaboratory is **highly** recommended. However, you can also run this notebook in your local environment if you meet the requirements for the stylegan2-pytorch repository.

To open the .ipynb file on Google Colaboratory, upload the provided folder in your Google Drive, then download Google Colaboratory from Google Workspace Marketplace. With this, you will be able to open the .ipynb file.

1.(a) In the tutorial, we invert the **Obama.jpg** using a simple MSE loss. Perform the same experiment for all 20 target images and fill in the table below. You must report the average and the standard deviation for the three metrics. Additionally, report all the hyperparameters that you used. This may differ from the tutorial, but you must use identical **hyperparameters** across different latent spaces.

Latent Space	**Loss Function**	**L2 Distance (↓)***	**LPIPS (↓)**	**Cosine Distance (↓)**
Z	MSE	mean (±std)		
W	MSE			
W+	MSE			

*(↓): Lower value means better

Hyperparameters:

- iteration: 1,000
- optimizer (lr): Adam (0.01)
- lr_scheduler: NaN
- etc.

1.(b). Quantitatively analyze the achieved results. Which latent space "seems" better overall? Do the metrics **agree** with your perception? Which seems to be the best metric?

2. Instead of using only MSE loss, devise your own loss function for optimization. You can use the LPIPS and Cosine Distance that is already given in the tutorial as below:

$$\mathcal{L} = \mathcal{L}_{MSE} + w_1\mathcal{L}_{LPIPS} + w_2\mathcal{L}_{cos}$$

If you do so, report the weighting factor for each loss function and give a brief explanation. If you devise your own function, also give a brief explanation of that function. Compare the result against that of MSE. Note that for a fair comparison, the **hyperparameters** should be kept identical to the ones used for the MSE experiment.

Latent Space	**Loss Function**	**L2 Distance (↓)***	**LPIPS (↓)**	**Cosine Distance (↓)**
Z	MSE	mean (±std)		
Z	Your Function			
W	MSE			
W	Your Function			
W+	MSE			
W+	Your Function			

*(↓): Lower value means better

Optional for bonus points:

Propose **any method(s)** to improve the final performance. It can be **anything** as long as you do not significantly change the hyperparameters such that the comparison is not fair anymore. Explain your method and how it improves upon other baselines from 1 and 2.

Latent Space	**Loss Function**	**L2 Distance (↓)***	**LPIPS (↓)**	**Cosine Distance (↓)**
Z	MSE	mean (±std)		

Z	Your Function			
W	MSE			
W	Your Function			
W+	MSE			
W+	Your Function			
Proposed	**Proposed**			

*(↓): Lower value means better

References:

[1] R. Abdal, Y. Qin, and P. Wonka, "Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?," in 2019 CVPR.

[2] R. Abdal, Y. Qin, and P. Wonka, "Image2StyleGAN++: How to Edit the Embedded Images?," in 2020 CVPR.