School of Electrical and Computer Engineering- University of Tehran
Artificial Intelligence Course- 2024
Project 1-Due Date: May 2, 2024, before 4 PM

Project 1

Part I

A. **Use VGG16 and ResNet50 and classify three different skin diseases (Nevus, Melanoma, and Carcinoma) with and without data augmentation. Compare the classification report for all four models. You should use TensorFlow. The dataset is provided in the following link.**
https://drive.google.com/drive/u/0/folders/1XC3H25tMjoNllXqU0UGY9jc2cUiaXzuF
B. Explain the architecture of VGG16 and Resnet50. [Length of your recording $\leq$ 5 minutes].
C. Explain the codes for VGG16 and ResNet50 with data augmentation. [Length of your recording $\leq$ 10 minutes].
D. What are the advantages and disadvantages of data augmentation in Convolutional Neural Networks? [Length of your recording $\leq$ 5 minutes].

Note: Data augmentation is one of the most common tricks for improving the recognition performance is to augment the training data in an intelligent way. There are multiple strategies to achieve this effect:

Translation and rotation invariance: For the network to learn translation as well as rotation invariances, it is often suggested to augment a training dataset of images with the different perspective transformation of images. For instance, you can take an input image and flip it horizontally and add it to the training dataset. Along with horizontal flips, you can translate them by a few pixels among other possible transformations.

Scale invariance: One of the limitations of a CNN is its ineffectiveness to recognize objects at different scales. To address this shortcoming, it is often a good idea to augment the training set with random crops of the input images. These random crops act as a sub-sampled version of training images. You can also take these random crops and up-sample them to the original height and width of the input image.

Color perturbation: One of the more interesting data transformations is perturbing the color values of the input image directly.

[Source: Deep Learning Essentials, Wei Di, Anurag Bhardwaj, Jianing Wei, 2018]

You can use the following codes to implement image augmentation.

```
from tensorflow.keras.preprocessing.image import
ImageDataGenerator

data_generator = ImageDataGenerator(
  width_shift_range=0.1,
  height_shift_range=0.1,
  horizontal_flip=True)
```

Other args: `rotation_range, width_shift_range, height_shift_range, brightness_range, shear_range, zoom_range, horizontal_flip, and vertical_flip`

```
data_generator = ImageDataGenerator(...)
train_generator = data_generator.flow(
  x_train, y_train, batch_size)

steps_per_epoch = x_train.shape[0] // batch_size
r = model.fit_generator(
  train_generator,
  steps_per_epoch=steps_per_epoch,
  epochs=50)
```

Part II

A. Create a Deep Convolutional Generative Adversarial Network for the CIFAR-10 dataset and explain your codes. You should use TensorFlow. The dataset can be imported using [from keras.datasets import cifar10]. [Length of your recording $\leq$ 10 minutes].

B. I stated in the lecture that when training GANs, optimizing the generator's objective function, which is given below, does not often produce good results.

$$\min_{\theta_g} \mathrm{E}_{z\sim p(z)} log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$ **Gradient Descent**

As a result, practitioners optimize the following objective function for the generator.

$$\max_{\theta_g} \mathrm{E}_{z\sim p(z)} log(D_{\theta_d}(G_{\theta_g}(z)))$$ **Gradient Ascent**

Why is this the case? Provide an example and demonstrate the superiority of Gradient Ascent over Gradient Descent for optimizing the generator's objective function. [Length of your recording $\leq$ 5 minutes].

Part III

I stated in the lecture that Transfer Learning is used to train a model on large amounts of data for a long time for task A and then to use that pre-trained model for task B. For instance, if you know how to play tennis, you can learn how to play ping pong very quickly because your knowledge in one domain is transferred to the new skills you're learning in another.

A. How should you employ Transfer Learning If your dataset is small and very different from the original dataset used to train the pre-trained model? Provide an example using Python and TensorFlow. [Length of your recording $\leq$ 5 minutes].
B. How should you employ Transfer Learning If your dataset is small and similar to the original dataset used to train the pre-trained model? Provide an example using Python and TensorFlow. [Length of your recording $\leq$ 5 minutes].

Important Note

1. For this project, you should record a video. You should show your face. Your voice must be clear. It is your responsibility to make sure your video is working. You should put your video and notebook in Google Drive. Your folder should have your [student ID number, first name, and last name]. Only one video should be submitted for all parts of this project.
2. Your notebook should be well structured, and before each code block, you should explain the codes of the next block. You should follow the format of the notebooks I presented in the lectures.
3. You should make your Google link open access and submit your video to the auxiliary TA of our course, Mr Behzad Mohasel Afshari, via e-mail on May 2, 2024, before 4 PM. His e-mail for this course is ai.2024.cs@gmail.com.
4. You should use Google Colab. Insert all the slides into your Google Colab environment, then record all parts of this project.
5. You should submit the following two items
A. Your Google Colab notebook.
B. Your video file. The extension of your video file should be .mp4. Other formats will not be accepted.
6. Your video file and notebook should be saved as [student ID number, first name, and last name]. Your total video file should not be more than 45 minutes. Your recording must be in English.
7. You should ensure that your codes and mathematical algorithms are flawless.

If further elucidation is warranted, please don't hesitate to contact me.