Laboratory Work 1 RN&DL Spring 2023

General Instructions:

The labs must be uploaded to ELSE as a PDF file before the deadline. Labs submitted after the deadline will be penalized by 1 point.

To answer each question, first copy- paste the question itself, then provide the answer bellow. The answer must contain the code that answers each question, the output of the code and most importantly the explanations when these are required.

The PDF files should be saved as follows: **firstname_lastname_LAB1_RNDL**

Assignment:

Use the data from the **dataset.zip** file to train a convolutional neural network that will distinguish between images of cats and dogs. You will find this file in Teams, Files from our RNDL class, CNN folder, Lab 1 Besliu CNN folder (here is the link dataset.zip).

The dataset has already been split into a training set and a test set, so work with this split. There is also a file called "single prediction" which contains just one image of a dog and one image of a cat, you will it when making a single prediction in steps 11 & 12.

For the first 13 steps you can use the code provided to you on ELSE and provide explanations where required. Running the code without explanations can add only up to 1/3 of the total points.

Note: *The dataset contains thousands of images, thus you will not be able to implement this lab in a Google Colab notebook. If you did work in Google Colab for your previous labs, you may want to download Anconda and work in Jupyter Notebook for this lab.*

Part 1: Building and Training the Model

1. Import the necessary libraries and the ImageDataGenerator class to perform image augmentation; Import the training set and perform the required steps for image augmentation (rescale, zoom, flip, etc.). You can use the same steps as in the code provided to you.

It is advisable to rescale the images to make training faster (64x64 is a good choice as the training will be faster and the performance will not suffer much).

Perform the same steps for the training set. Be careful though when it comes to the image augmentation steps.

Explain what each of the image transformations in the image augmentation part does and why image augmentation is necessary when working with CNNs**.** Explain why these steps should not be performed on the test set – remember the idea of information leakage. **(10 points)**

2. Initialize the CNN as an object of the Sequential class in Keras **(1 point)**
3. Add the first convolutional layer. Specify the batch size = 32, choose the filter size 3x3, the rectifier function as the activation function, and don't forget to specify the input size as it is your first layer. **(1 point)**
4. Add a pooling layer with the pool size = 2x2 and a stride = 2 pixels. **(1 point)**
5. Add a second convolutional layer (don't forget to remove the shape of the input, you need it only in the first layer). Add a second Max Pool layer. **(1 point)**
6. Flatten the final output in order to pass it through a fully connected layer. **(1 point)**
7. Add the Dense layer and choose 128 neurons in this layer. Choose ReLu as the activation function. **(1 point)**
8. Add the output layer and specify the sigmoid as the activation function. Explain why we can use a sigmoid in the output layer, but not in the previous layers - remember the vanishing gradient problem? - explain it. **(5 points)**
9. Compile the CNN with the Adam optimizer. Choose the appropriate loss function for your use case (remember you have a classification problem with 2 classes) and choose the appropriate accuracy metric for evaluation. **(2 points)**
10. Train the CNN for 25 epochs and evaluate it using the training set. **(2 points)**

Total max points for Part 1: 25 points

Part 2: Making Predictions & Evaluating the Model

11. Create a single prediction for each of the two pictures in the "single prediction" folder. Print the predictions. Did the model predict correctly? **(5 points)**
12. From the test set, the folder with dog images, choose the image #4689, copy paste it into your single prediction folder. Open this image first and comment if you expect the model to make a good prediction. Make the prediction. Did the model predict correctly? **(5 points)**
13. Do the same for image #4538 in the test set with cat images. Open this image and look at it first. Given your knowledge of CNNs would you expect that the model may have difficulty making the right prediction for this image? Explain why. **(5 points)**
14. Compute the confusion matrix and comment on the result (i.e. false positives, false negatives, etc.). **(5 points)**
15. Notice in the confusion matrix that the model seems to be doing a better job at classifying dog images rather than cat images. Do you have an intuition about why this could be the case? Tip: browse among the dog images versus cat images, consider the positions in which they were photographed, how often they wear something a-typical for the animal, etc. Consider which class seems to have a greater number of outliers. **(5 points)**

Total max points for Part 2: 25 points

Part 3: Fine Tuning & Regularization

16. Plot the accuracy scores for the training set and validation set on the same graph. Do you see signs of your model overfitting? Explain based on the graph. **(5 points)**
17. To address overfitting add a drop-out layer where 50% of neurons in the Dense layer are dropped at random. Plot again the accuracy scores of the training and validation data and comment. Did this improve the situation? Do you still see signs of overfitting? **(5 points)**
18. To further reduce overfitting add more operations to the image augmentation part. Specifically:

Change *rotation_range = 40;*

Add *width_shift_rage = 0.2*

Add *height_shift_rage = 0.2*

Add *fill_mode = 'nearest'*

Explain for each of these steps why this should help with overfitting. With the newly augmented images retrain and refit the model, plot the accuracy scores like in point (18) and discuss the effect this had on your model and its propensity to overfit.

Notice the training accuracy – has it decreased? **(10 points)**

19. Since performance has declined try the following steps to bring it back up again:

a. Reduce the range for the data augmentation transformations. Specifically change the following:

rotation_range = 20

width_shift_range = 0.1

height_shift_range = 0.1

shear_range = 0.1

b. Increase the target image size to 128x128 so we don't lose as much data from downscaling
c. Change the architecture of the CNN by adding two convolutional layers and changing the number of filters so that it doubles for each consecutive set of convolutional layers.

Explain why you expect the doubling of filters for each consecutive convolution layer to have a positive impact on model's accuracy (one easy place to find the answer is in Keras documentation).

Once again plot the accuracies for train and validation set, and comment on the effect of making these changes. **(15 points)**

20. Notice on your graphs that the validation accuracy starts stagnating at some point during learning. To address that reduce the learning rate when the validation accuracy stops improving.
    Quote from the Keras documentation: "Models often benefit from reducing the learning rate by a factor of 2-10 once learning stagnates. This callback monitors a quantity and if no improvement is seen for a 'patience' number of epochs, the learning rate is reduced."

To reduce the learning rate appropriately use the ReduceLROnPlateau class with the following arguments:

monitor = 'val_accuracy',

patience = 3,

verbose = 1,

factor = 0.5,

min_lr = 0.00001

Explain what each of these preset argument values will do to the model.

<u>If you decide not to answer the extra credit question below, finish the lab by doing the following:</u>

Plot the training and validation accuracy scores, compute the confusion matrix and discuss the improvements in the final accuracy scores. **(15 points)**

If you answer the Extra Credit question below perform these steps after training the model longer.

<u>Total max points for Part 3: 50 points</u>

<u>Extra Credit Question:</u>

Below is an Extra credit Question. Providing a solution to this question will add an extra **0.2 points to your final grade!** (not your lab grade, but your final grade for this class!).

<u>Question:</u> Plot the train and validation accuracies again. Notice that the validation accuracy continues improving until the end of training. Discuss what should be done in such case? (Tip: this a sign that we stopped training the model too early and that we could boost model's performance by training it longer).

Make the necessary adjustment, plot the training and validation accuracy scores, compute the confusion matrix and discuss the improvements in the final accuracy scores.