AER 1515 - Perception for Robotics

Assignment 1

Submission guidelines: This assignment is due on Friday, October 14, 2022 at 11:59 pm. Please submit only a single *typed* solutions as a PDF file.

Classification Using Convolutional Neural Networks

The convolutional neural network (CNN) represents a huge breakthrough in image recognition and is frequently used in image classification tasks. In this assignment, you will build a CNN for an animal classification task. The images of animal faces are from LHI-Animal-Faces dataset. It consists of 20 classes, 19 classes of different animals, and one human faces class. You are provided with 2313 training images and 100 test images.

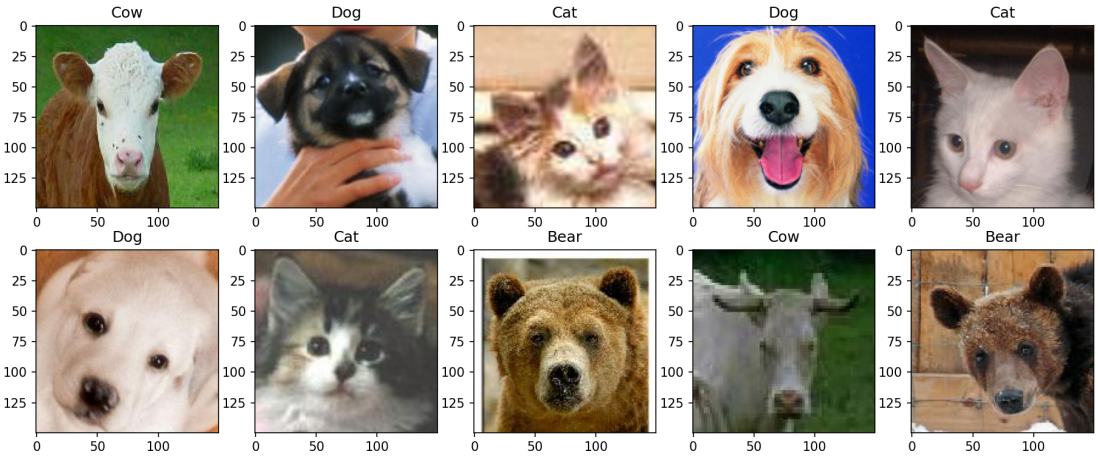

Figure 1: Sample images from the LHI-Animal-Faces dataset.

To Start With

We used python 3.8.13 and PyTorch 1.12.1 when developing the start code, and students are highly recommended to use the same version. For more details, please go through the "readme.txt" in the download package.

1 Run and Extend to Multi-Class Classification (15 pts)

We provide students the start code to train a binary classifier on "Dog" and "Cat" using the CNN model. The evaluation code is also provided for testing the trained model. After setting up the python environment, you should be able to run file "**train.py**" and train your model on the training set (2313 images). When training is done, it will generate a file called "**model.pt**" and a plot for the training loss (shown in figure 2). To test your trained model, you can run the file "**test.py**" for evaluation purpose. This will report the overall accuracy result (%) on the test set (100 images).

Questions

1. (*5 pts*) Run the code "**train.py**" and train a CNN model with at least 30 epochs. Save your trained model ("**model.pt**") on the disk and show a plot of the **training loss**. Here we show an example of the plot with only ten epochs.

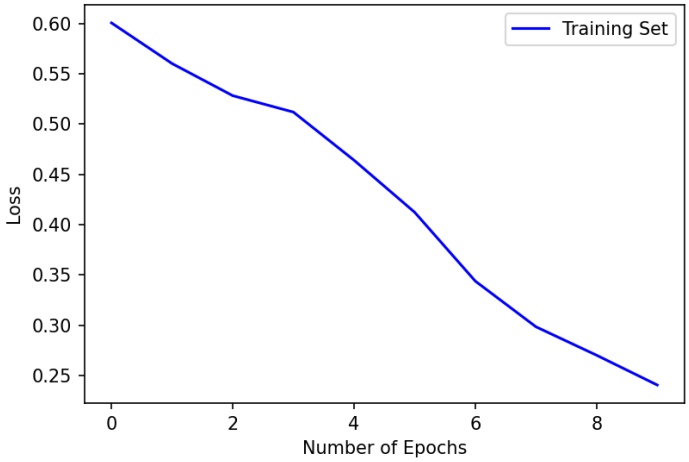

Figure 2: An example of the training loss for the binary classification task.

2. (*5 pts*) Run the code "**test.py**" with your saved CNN model ("**model.pt**"), and report the overall classification accuracy. 80%

3. (*5 pts*) Modify the code "**train.py**" and "**test.py**" to extend the binary classification to a multi-class classification (20 classes) task. Show your training loss plot and report the overall classification accuracy here. 67%

2 Change CNN Architecture (25 pts)

Our default CNN architecture is defined in the file "**Animal_Classification_Network.py**". It consists of 4 convolutional layers, 3 max-pooling layers, and 2 fully connected layers. We also use Rectified Linear Unit (ReLU) as our activation function. To improve the performance, students are asked to add additional layers to the current CNN architecture.

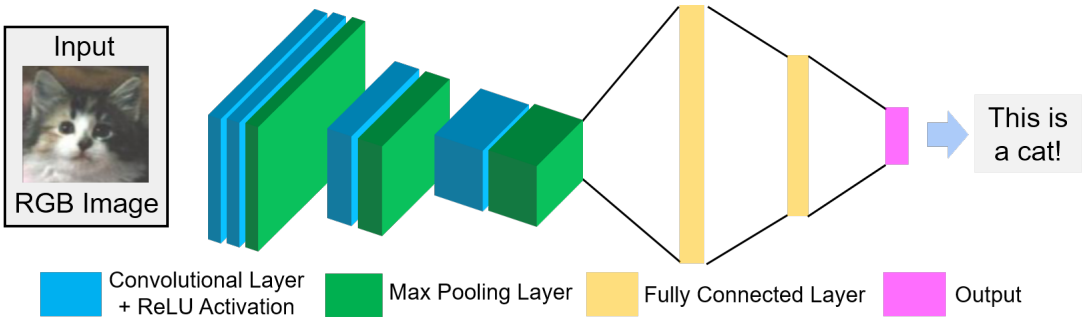

Figure 3: Default Network Structure for the animal classification task.

Questions

1. (*15 pts*) Based on current CNN architecture, add batch normalization layers and show your training loss plot and report the overall classification accuracy with the modified CNN architecture. For students who are not familiar with batch normalization, please refer to here. (*Hint: you can add one batch normalization layer before each max pooling layer.*)

2. (*10 pts*) Based on the modified CNN architecture from the previous Question, add dropout layers and show your training loss plot and report the overall classification accuracy. For students who are not familiar with dropout, please refer to here.

3 Training Neural Network with Validation (35 pts)

In the start code, we split the full training data into the training set (80%) and validation set (20%). However, we only use the training set and have not leveraged the validation set yet. In general, we are unable to achieve optimal performance using only the training set. Moreover, we save the trained CNN model at the end of the training with the last epoch. It may happen that your last epoch is NOT the one that gave you the least loss.

In this section, students will need to keep track of the loss on the validation set at each training epoch and also save the CNN model with the minimal validation loss.

Questions

1. (*25 pts*) Based on your modified CNN architecture, add the validation loop within the training epochs and save the CNN model that corresponds to the least validation loss. Please also show a plot that includes both **training loss** and **validation loss** with at least 30 epochs. Here we show an example of the plot with only ten epochs.

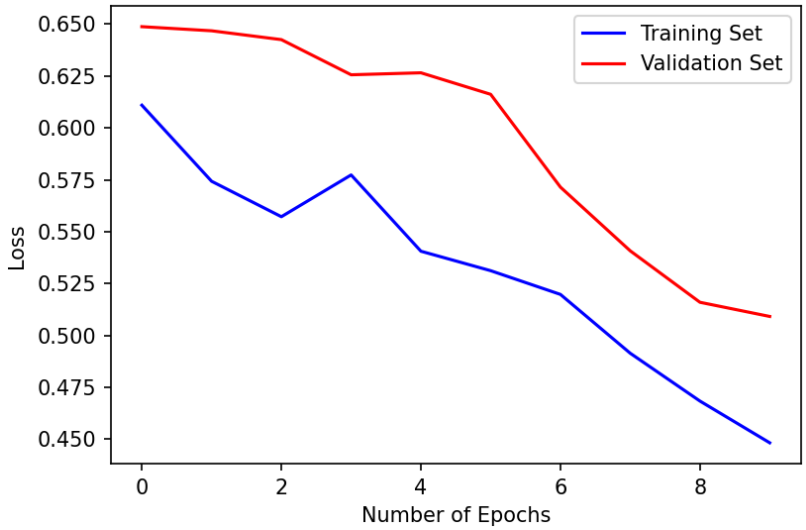

Figure 4: The example plot of training loss and validation loss.

2. (*10 pts*) Run the evaluation on the test dataset with your saved CNN model. Please compare the classification accuracy with the one that does not use the validation set.

4 Hyperparameter Tuning (25 pts)

In the start code, we use the RMSProp as our optimizer for the network training. The batch size is 32, and the learning rate is set to $1 \times e^{-4}$.

In this section, students will need to tune these hyperparameters and discover new combinations of hyperparameters, resulting in a better classification result on the test set.

Questions

1. (*7 pts*) Change the optimizer to another one and report the classification results on the test set. A brief introduction of different optimizers can be found here.

2. (*9 pts*) Select an optimizer and tune the learning rate. Please show the training and validation loss plots with different learning rates and report the classification result on the test set. Briefly describe how the training rate affects the training progress (training and validation loss) and the final result on the test set.

3. (*9 pts*) Select an optimizer and tune the batch size. Please show the training and validation plots with different batch sizes and report the classification result on the test set. Briefly describe how the batch size affects the training progress and the final result on the test set.

4. (*Extra Credit, 10 pts*) Try your best to discover different combinations of hyperparameters (different optimizer, learning rate and batch size), and report the best classification result you have achieved on the test set. Please also report the corresponding hyperparameters for your best result.