

Machine Learning and Computational Physics

Fall 2021

Assignment 4

Due: Oct. 22nd 2021, 11:59:59 PM PDT

Convolutional Neural Network (CNN)

In this assignment, you will train a deep convolutional network to classify images from the CIFAR-10 dataset. The primary objective of this assignment is to learn how to build and train a CNN for image processing tasks and visualize the learned filters.

1. For this assignment we will be using CIFAR-10 dataset, which is a dataset of color images of objects belonging to 10 different classes.
2. Download these images and re-scale them to have values between 0 and 1. For downloading and splitting these into a training and testing set use the command: `keras.datasets.cifar10.load_data()`
3. Plot 25 training images in a 5×5 grid. Include the class name for each image. To do this (a) create a dictionary whose keys are the labels 0-9 and the values are the corresponding class names, and (b) use the command `plt.xlabel` to print the label for each image.
4. Then construct a CNN based classifier with the following architecture :
 - Convolution layer with 32 (3×3) filters, zero padding and ReLU activation.
 - A 2×2 Max. pooling layer.
 - Convolution layer with 128 (3×3) filters, zero padding and ReLU activation.
 - A 2×2 Max. pooling layer.
 - Convolution layer with 256 (3×3) filters, no padding and ReLU activation.
 - Flatten the output of the previous layer.
 - A fully connected layer with 256 neurons and ReLU activation.
 - A final fully connected layer with width = number of classes = 10 and no activation.
5. Train this network with the following parameters:
 - l_2 regularization of 1.0e-5.
 - optimizer = Adam.
 - learning rate set to the default value.
 - number of epochs = 20.
 - loss function = sparse categorical cross-entropy.
 - batch size = 64.
6. Create a plot of accuracy as a function of number of epoch for the training and test set.

7. Plot 25 test images in a 5×5 grid. Include the predicted class name for each image. Report how many of these labels are correct.
8. For the trained network extract the learned filters from the first convolutional layer. (Hint: you might want to use combination of `model.layers` and `Layer.get_weights()`). For each of the 32 filters of the first layer, subtract their respective mean and then plot all 32 filters of size (3×3) . For each subplot show the colorbar, and choose its limits symmetrically about zero. Note that the point of creating these plots is to see if you can identify these filters (see the next task).
9. Identify at least one filter that approximate a first derivative and one filter that approximates a second derivative. Justify your assertion. Also say along which direction these derivatives are computed.

Instructions:

- At the very beginning of your notebook insert a text cell and write your name and **USC email address**.
- You need to submit your work as a single notebook saved as `A1_FirstName.LastName.ipnyb` (for example `A1_Tommy_Trojan.ipnyb`). You can create this notebook locally (on your computer using Jupyter notebook) or on cloud using Google Colab (which we recommend). If you are using Google Colab, then please make sure that you are signed in to your USC Google account before starting. This will make sharing your saved work little easier.
- For questions requiring descriptive output (such as question 5 in this assignment) use individual text cell.
- Make sure that your entire notebook runs successfully on Google Colab before submitting it. It is your responsibility to ensure this.
- Once you finish the assignment save it and share it with `hramaswa@usc.edu`. (If you are using Google Colab, then the notebook will automatically be saved to your Google Drive. Once you locate it in your Google Drive, right click on it and share it with `hramaswa@usc.edu`). While sharing make sure that you enable “editor” option, so that we can run your notebook on our end while grading it.