

CPSC 4430/6430 Project 4: Linear Classifier and Neural Network

Learning Objectives

- Implement and apply a Multiclass Support Vector Machine (SVM) classifier
- Implement and apply a Two-layer Neural Network classifier
- Understand the differences and tradeoffs between these classifiers
- Understand how a Two-layer Neural Network can approximate an arbitrary function
- Practice implementing vectorized gradient code by checking against naive implementations, and using numeric gradient checking

Download Assignment Files

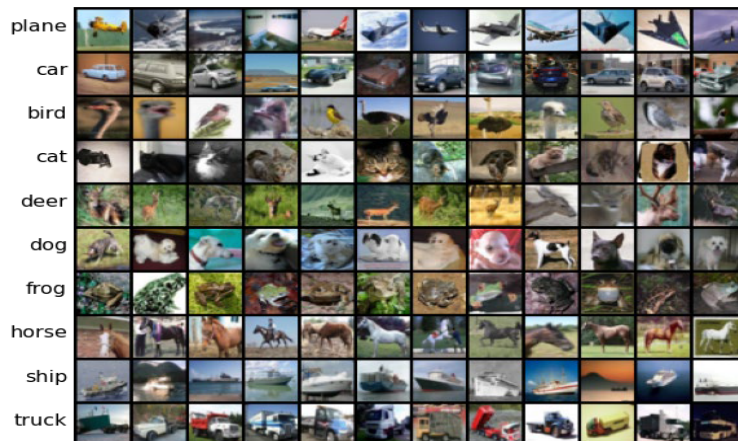
- Starter code for this project can be downloaded on the project page on Canvas

Setup the Google Colab

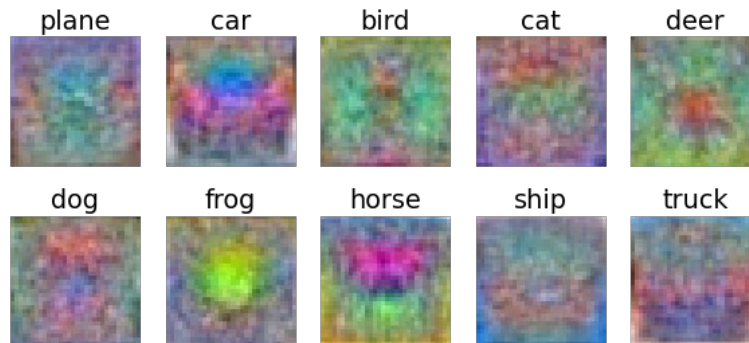
- Install Colab by going to your Google Drive (drive.google.com) and selecting My Drive > More > + Connect more apps. Search the marketplace for Colaboratory, select it, and click Install.
- Extract the starter code from the .zip file and upload them to a folder in your Google Drive.
- Double-click on ipynb file to open the file in Colab. If this doesn't happen automatically with a double-click, right-click and select Open with > Google Colaboratory.
- For more detailed setup instruction using Google Colab, please watch the video in the following website: <https://towardsdatascience.com/getting-started-with-google-colab-f2fff97f594c>

Part 1: Linear Classifiers

- Open **linear_classifier.ipynb** and follow the instructions in the notebook. If you are using the Palmetto Cluster, remember to skip the cell that mounts the Google Drive, or you will get an error message.
- You will be using CIFAR-10 for this project. One of the first cells you run will visualize the dataset. It should look something like this.

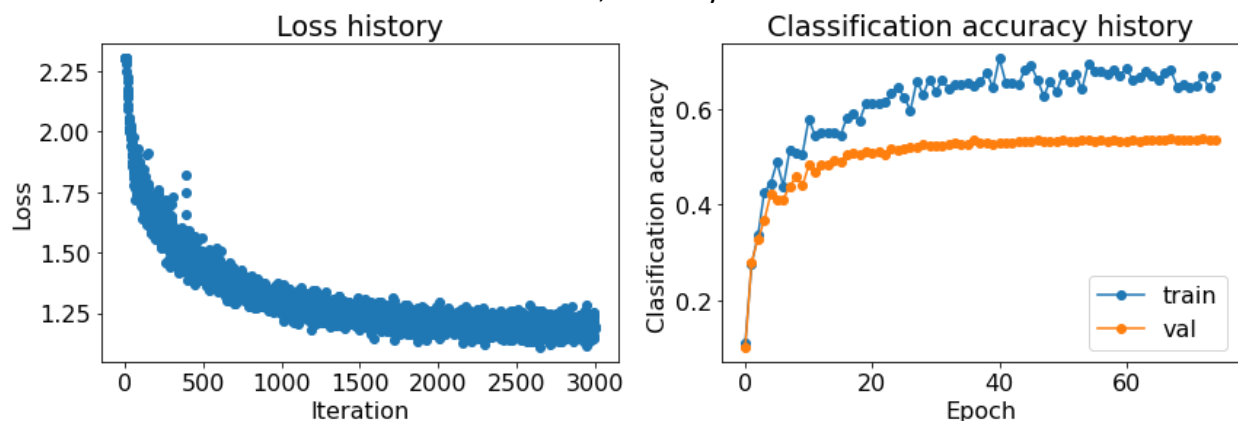


- In the first section of this notebook, you'll train an SVM classifier. The last cell in this section will print out the weights that your model has learned for each category. This shows you which parts of the image the classifier has found most important for deciding an image belongs to that category. **Please note that you need to fill in the code that is highlighted by "YOUR TURN:" in the comments.** Ours looks like this; yours should look similar but may not be identical.



Part 2: Two-Layer Neural Network

- Open **two_layer_net.ipynb** and follow the instructions in the notebook.
- This notebook will walk you through training a two-layer neural network to classify images. At the end, you'll train it using a range of hyperparameters to find the combination with the best results, and plot the loss and accuracy over training of your best model. Since this is the best of several models, you should see a loss curve that is clearly decreasing over time and an accuracy that is clearly increasing over time. Ours look like this. Yours should look similar, but may not be identical.



Project Submission

- Download your work from either Google Drive or the Palmetto Cluster and zip together **linear_classifier.ipynb**, **linear_classifier.py**, **two_layer_net.ipynb**, **two_layer_net.py**, **svm_best_model.pt** and **nn_best_model.pt**.
- Name your zip file by Firstname_Lastname_P4.zip
- Submit your zip file to Canvas on the assignment page **by the due date**.