ece5300/au20 Default Project

Phil Schniter

November 4, 2020

This report describes the "default" ece5300 final project, i.e., the project that you should complete if you do not have an approved custom project.

1 Introduction

This project focuses on classification. The data comes from a real-world application, but the details are not important; it is possible to design good classifiers using standard machine-learning methods, without application knowledge. That's the power of machine learning.

The data will be made available to you in a starter GitHub repository, similar to how we distribute the labs. Some important parameters are:

- number of classes: $K = 20$
- samples in each class: $n/K = 5000$
- feature dimension: $d = 20$

Since the dataset is well balanced (i.e., equal # of samples per class), test accuracy is a good performance metric. There is a separate .csv file for each class. After loading that file, you will get a matrix whose rows correspond to the data samples, whose last column corresponds to the class label, and whose remaining columns correspond to features. The notebook `model_validation.ipynb` shows how to load the data.

2 Possible Approaches

Here are some ideas to try in a project. With a team of three people, several of these ideas could be tackled.

- Apply logistic regression to this classification task.
  - Does joint training (i.e., multinomial) work better than one-versus-rest?
  - Does regularization (e.g., L2 or L1) help?
  - What are the optimal regularization weights?
  - Does feature selection help?
  - What does the confusion matrix look like?
- Apply SVC/SVM to this classification task.

  - Similar questions as logistic regression, plus...
  - What are the jointly optimal regularization weight and kernel width?

- Apply a 2-layer neural network to this classification task.
  - What is a good choice of # hidden nodes?
  - What is a good choice of hidden activation functions?
  - What are good choices of learning rate and/or learning-rate schedule?
  - Do batch-norm and/or dropout help?
  - What does the confusion matrix look like?

- Apply a convolutional deep network to this classification task.
  - Similar questions as the shallow network, plus...
  - What is a good choice of # convolutional layers?
  - What is a good choice of # dense layers?
  - What is a good choice of # channels in each layer?
  - What is a good choice of kernel size in convolutional layers?

- Apply Random Forests to this classification task.
  - What are good parameter choices?
  - How does it compare to other methods in test accuracy and training time?

- Apply XGBoost to this classification task.
  - What are good parameter choices?
  - How does it compare to other methods in test accuracy and training time?

3 Validation Script

We will test your methods to confirm the results described in your written report. We will also test your methods on additional test data and give extra points to the groups that achieve the highest accuracy.

To help facilitate the model validation, you are required to modify the `model_validation.ipynb` notebook so that it validates the trained models designed by your various group members. In particular, this notebook

- loads the data from a specified directory
- loads each of your trained models (**from saved-model files that you provide**; we include some examples)
- applies your trained models to the data
- evaluates the accuracy of the model outputs.

You must modify the `model_validation.ipynb` notebook so that that generates the results described in your written report. This is a key project deliverable

For background on saving/loading sklearn models with pickle, see this page. For background on saving/loading PyTorch models, see this page and/or the `pytorch_saving_demo.ipynb` found in the default-project repo.