Final Project

ECE 5450: Machine Learning

2020

The main goal of the project is to classify natural images in the CIFAR-10 dataset. The dataset and it's description can be found at https://www.cs.toronto.edu/~kriz/cifar.html.

The project consists of two parts as described below. It is recommended that you use the following resources:

- Scikit-learn and Scikit-image libraries for the support vector machine implementation in Part I
- Pytorch for the implementation of the deep learning components in Part II
- Google Colab: Free GPU resources and popular deep learning libraries (Pytorch) installed. See Python resources module on ICON to setup and use Google Colab.

Part 1: Classification using support vector machines (40% of grade)

(Due on Dec 6th 2020)

An example of SVM based multi-class classification is provided in svm_faces.ipynb. Here, a simple face recognition task with 6 classes is considered. The principal component analysis features are used to classify the objects.

You will extend the classifier to CIFAR 10 (https://www.cs.toronto.edu/~kriz/cifar.html) dataset with 60000 32x32 color images with 10 classes. You need to select the best classifier using model selection and report its accuracy. The main tasks are to experiment with different parameters of the model to improve classification accuracy. You should experiment and determine the best choices for each of the following.

1. Best set of image features for color images. You can use skimage library to try different feature extraction strategies.
2. Best kernel function (e.g. polynomial, radial basis function) for support vector classification.
3. Good normalization strategies for the data to yield improved performance.

Deliverables

Ipython notebook with clear documentation (within comments) on what you have tried and what you have observed, including plots.

Grade allocation:

30% for implementation

5% for presentation of code and data.

Well commented code with adequate details for easy understanding.

3% extra credit will be allocated for clear explanation on the observations (e.g. why some features, normalization strategies, kernel functions work better than others for this dataset).

5% for performance

Top 30% (15 students) will receive the 5% credit for the classification with highest accuracy

Top 10% (5 students) will receive 5% extra credit.

Part II: Classification using deep neural networks (60% of grade):

(Due on Dec 18th 2020)

An example of CNN based multi-class classification is provided in cnn_mnist.ipynb. You will use the MNIST dataset to do tasks 1-5 since it is a smaller dataset, which will translate to faster training and testing. Once you have have identified the “best” set of design parameters, you will translate it to the CIFAR 10 dataset.

1. Learning rate (LR) and Optimizer: Adam or SGD
    a. Reading Material:
        i. https://medium.com/octavian-ai/which-optimizer-and-learning-rate-should-i-use-for-deep-learning-5acb418f9b2
        ii. https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c
        iii. https://shaoanlu.wordpress.com/2017/05/29/sgd-all-which-one-is-the-best-optimizer-dogs-vs-cats-toy-experiment/
    b. Find proper LR for Adam and SGD, plot training loss vs training epoch number, and compare the convergence speed of the two optimizers and their respective test classification accuracies.
    c. Describe the lessons you learn from the experiments

2. Activation functions
    a. Reading Material:
        i. https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/
    b. Train two networks with Sigmoid and Relu as respective activation functions

c. Test and compare the training convergence speeds and classification accuracies on the test dataset. Give your observation.

3. <u>Early stopping strategy</u>
   a. Reading Material:
      i. https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/
      ii. https://towardsdatascience.com/preventing-deep-neural-network-from-overfitting-953458db800a
   b. Develop your early stopping strategy
   c. Test the classification accuracies with or without early stopping

4. <u>Data augmentation</u>
   a. Reading Material:
      i. https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/
      ii. https://www.aiworkbox.com/lessons/augment-the-cifar10-dataset-using-the-randomhorizontalflip-and-randomcrop-transforms
   b. Augment the training data and train the network
   c. Test the classification accuracy and compare it to that without using augmentation

5. <u>Network depth vs network width</u>
   a. Design two networks with different depths, but similar total number of parameters
   b. Test the classification accuracy and give your observation.

<u>Grade allocation:</u>

50% for implementation

5% for presentation of code and data.

Well commented code with adequate details for easy understanding.

3% extra credit will be allocated for clear explanation on the observations (e.g. why some features, normalization strategies, kernel functions work better than others for this dataset).

5% for performance

Top 30% (15 students) will receive the 5% credit for the classification with highest accuracy

Top 10% (5 students) will receive 5% extra credit.