CSCC11: HW3 Due by 11:59pm Sunday, December 1, 2019

University of Toronto Scarborough

November 18, 2019

Please submit separate files for a) write-up (named write_up_hw3.pdf) in PDF (You can convert word doc to PDF or preferably use LaTeX (https://www.overleaf.com/)), b) Python files and c) figures (if you choose to include them separately from the write-up). All files to be submitted on MarkUs. Do not turn in a hard copy of the write-up.

1. **Short questions** (no programming required for this question)

   (a) Describe the following loss functions.
   - Hinge Loss
   - Squared Hinge Loss

   (b) Briefly describe the following neural network.
   - Recurrent neural network (RNN)
   - Long-short term memory (LSTM)
   - Auto-encoder (AE)

   (c) What is Vapnik–Chervonenkis (VC) dimension? What is the VC dimension of a
   - Straight line
   - Triangle

2. **Image classifier using convolution neural network**

   - **Dataset:** CIFAR10 dataset
     (download https://www.cs.toronto.edu/ kriz/cifar-10-python.tar.gz)
     The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.
     The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.
   - **Classes:** The dataset consists of 10 classes {'airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck'}

- **Starter code:** Available in the file cnn_hw3.py
- **Steps:**
  (a) Load and normalize the CIFAR10 training and test datasets using *torchvision* (code provided in the starter code)
  (b) Define a Convolutional Neural Network (3 convolution layers, max pooling, and 3 layers for feed-forward network. You will decide the rest of the parameters yourself.)
  (c) Define a loss function. Try
    – Cross-Entropy
    – Hinge Loss
    – Squared Hinge Loss
  (d) Define an optimizer. Try
    – Stochastic gradient descent with momentum
    – Adam
  (e) Train the network on the training data and save the model
  (f) Test the network on the test data
- **Input:** trainset, testset (mentioned in the starter code)
- **Output:** Loss function value, Confusion matrix for test dataset (for each loss function and optimizer).
- **Libraries:** Pytorch
- **Functions:** Provided in the starter code.
- **Python file name:** cnn_hw3.py

NOTE: You can use GPU instead of CPU for efficiency.

3. **Linear support vector machine**

Implement a linear SVM **using any Python built-in functions** for the breast cancer data (80% train and 20% test).

- **Input:** Dataframe (starter code provided linear_svm_hw3.py)
- **Output:**
  – Visualize the relationship between pair of features (use built-in function)
  – Check correlation between all features using heat-map (use built-in function)
  – Classification report (use built-in function)
- **Libraries:** Any library, any function
- **Functions:** No restriction
- **Python file name:** linear_svm_hw3.py