Stony Brook University
CSE512 – Machine Learning – Spring 18
Homework 3, Due: 3 April 2018

This homework contains 2 questions. The second question requires programming. The maximum number of points is 100 plus 20 bonus points.

1 Question 1 – Boosting (40 points)

We learned about boosting in lecture and the topic is covered in Murphy 16.4. On page 555 Murphy claims that "it was proved that one could boost the performance (on the training set) of any weak learner arbitrarily high, provided the weak learned could always perform slightly better than chance." We will now verify this in the AdaBoost framework.

1. (*10 points*) Given a set of N observations (x^j, y^j) where y^j is the label $y^j \in \{-1, 1\}$, let $h_t(x)$ be the weak classifier at step t and let α_t be its weight. First we note that the final classifier after T steps is defined as:

$$H(x) = sgn\left\{\sum_{t=1}^{T} \alpha_t h_t(x)\right\} = sgn\{f(x)\}$$

Where

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

Show that:

$$\epsilon_{\text{Training}} = \frac{1}{N}\sum_{j=1}^{N} \delta(H(x^j) \neq y^j) \leq \frac{1}{N}\sum_{j=1}^{N} \exp(-f(x^j)y^j)$$

Where $\delta(H(x^j) \neq y^j)$ is 1 if $H(x^j) \neq y^j$ and 0 otherwise.

2. (*10 points*) The weight for each data point j at step $t+1$ can be defined recursively by:

$$w_j^{(t+1)} = \frac{w_i^{(t)} \exp(-\alpha_t y^j h_t(x^j))}{Z_t}$$

Where Z_t is a normalizing constant ensuring the weights sum to 1:

$$Z_t = \sum_{j=1}^{N} w_j^t \exp(-\alpha_t y^j h_t(x^j))$$

Show that:

$$\frac{1}{N}\sum_{j=1}^{N} \exp(-f(x^j)y^j) = \prod_{t=1}^{T} Z_t$$

3. (*20 points*) We showed above that training error is bounded above by $\prod_{t=1}^{T} Z_t$. At step t the values $Z_1, Z_2, \ldots, Z_{t-1}$ are already fixed therefore at step t we can choose α_t to minimize Z_t. Let

$$\epsilon_t = \sum_{j=1}^{N} w_j^t \delta(h_t(x^j) \neq y^j)$$

be the weighted training error for weak classifier $h_t(x)$ then we can re-write the formula for Z_t as:

$$Z_t = (1 - \epsilon_t)\exp(-\alpha_t) + \epsilon_t \exp(\alpha_t)$$

   (a) First find the value of α_t that minimizes Z_t then show that

$$Z_t^{opt} = 2\sqrt{\epsilon_t(1 - \epsilon_t)}$$

   (b) Assume we choose Z_t this way. Then re-write $\epsilon_t = \frac{1}{2} - \gamma_t$ where $\gamma_t > 0$ implies better than random and $\gamma_t < 0$ implies worse than random. Then show that:

$$Z_t \leq \exp(-2\gamma_t^2)$$

   You may want to use the fact that $\log(1 - x) \leq -x$ for $0 \leq x < 1$

   Thus we have:

$$\epsilon_{\text{training}} \leq \prod_{t=1}^{T} Z_t \leq \exp(-2\sum_{t=1}^{T} \gamma_t^2)$$

   (c) Finally, show that if each classifier is better than random (e.g. $\gamma_t \geq \gamma$ for all t and $\gamma > 0$) that:

$$\epsilon_{\text{training}} \leq \exp(-2T\gamma^2)$$

Which shows that the training error can be made arbitrarily small with enough steps.

2 Question 2 – Action recognition with CNN (60 points+20 bonus)

In this section, you will train convolutional neural networks (CNN) to classify images and videos using Pytorch. Similar to homework 2, we use the UCF101 data (see http://crcv.ucf.edu/data/UCF101.php). There are also 10 classes of data in this homework but the actual classes and data are different from the homework 2. Each clip has 3 frames and each frame is $64 * 64$ pixels. The label of clips are in $q3\_2\_data.mat$. $trLb$ are labels for training clips and $valLb$ are labels for validation clips.

You will first train a CNN for action classification for each image. Then try to improve the network architecture and submit the classification results on the test data to Kaggle. Then, you will train a CNN using 3D convolution to classify each clip as a video rather than a image, and submit your results to Kaggle.

The detail instructions and questions are in the jupyter notebook $Action\_CNN.ipynb$. In this file,there are 8 'To Do' spots for you to fill. The score of each 'To Do' is specified at the spot. For the 5th and 8th 'TO DO', you need to submit results csv files to Kaggle. The results would be evaluated by Categorization Accuracy.

For the 5th 'TO DO' submit the result *.csv* file to `http://www.kaggle.com/c/cse512springhw3`. For the 8th 'To Do' submit the result *.csv* file to `http://www.kaggle.com/c/cse512springhw3video`.

We will maintain a leader board for each Kaggle competition, and the top three entries at the end of the competition (assignment due date) will receive 10 bonus points. Any submission that rises to top three after the assignment deadline is not eligible for bonus points. The ranking will be based on the Categorization

Accuracy. To prevent exploiting test data, you are allowed to make a maximum of 2 submissions per 24 hours. Your submission will be evaluated immediately and the leader board will be updated.

Environment setting:

Please make a $./data$ folder under the same directory with the *Action_CNN.ipynb* file. Put data $./trainClips$, $./valClips$, $./testClips$ and *q3_2_data.mat* under $./data$.

We recommend using virtual environment for the project. If you choose not to use a virtual environment, it is up to you to make sure that all dependencies for the code are installed globally on your machine. To set up a virtual environment, run the following in the command-line innterface:

```
cd your_hw3_folder
sudo pip install virtualenv      # This may already be installed
virtualenv .env                    # Create a virtual environment
source .env/bin/activate         # Activate the virtual environment
pip install -r requirements.txt  # Install dependencies
# Note that this does NOT install TensorFlow or PyTorch,
# which you need to do yourself.

# Work on the assignment for a while ...
# ... and when you're done:
deactivate                       # Exit the virtual environment
```

Note that every time you want to work on the assignment, you should run 'source .env/bin/activate' (from within your hw3 folder) to re-activate the virtual environment, and deactivate again whenever you are done.

3 What to submit?

3.1 Blackboard submission

For question 1, please put everything in one single pdf file and submit it on Blackboard, please include your name and student ID in the first page of the pdf file. For question 2, submit the jupyter notebook files *Action_CNN.ipynb* with your answers filled at the 'To Do' spots. Put the pdf file and your jupyter notebook file in a folder named: SUBID_FirstName_LastName (e.g., *10947XXXX_heeyoung_kwon*). Zip this folder and submit the zip file on Blackboard. Your submission must be a zip file, i.e, SUBID_FirstName_LastName.zip.

3.2 Kaggle submission

For Question 2, you must submit a $.csv$ file to for each Kaggle competition to get the Categorization Accuracy. A submission file should contain two columns: Id and Class. The file should contain a header and have the following format.

$$\begin{array}{cc} Id, & Class \\ 0, & 3 \\ 1, & 7 \\ 2, & 2 \\ ... & ... \end{array}$$

A sample submission file is available from the competition site and our handout. You MUST use your Stony Brook CS email account to submit. A submission file can be automatically generated by $predict\_on\_test()$ and $predict_on_test_3d()$in *Action_CNN.ipynb*.

4 Cheating warnings

Don’t cheat. You must do the homework yourself, otherwise you won’t learn. You must use your real name to register on Kaggle. Do not create multiple accounts to bypass the submission limitation per 24 hours. Doing so will be considered cheating.