Action Recognition @ UCF101
CSE 527 HW 5

Drive Folder Link: https://drive.google.com/drive/folders/146gYfuyYZU0j4yXuEMDPdxOycbNeDYdX?usp=sharing

REPORT

For this homework, we were required to predict the actions happening in the videos from the video frames which were tagged as either training or testing samples.

I first started off by finding out these corresponding labels from the **videos_labels_subsets.txt** file and saved them for future use. Based on these labels, I divided my dataset into train data and test data categories so that I can add load them for use in model using Pytorch's **DataLoader.** But before loading these images, I applied the **FiveCrop** transform to get one center and 4 corner images, as well as normalised the images according to the measures on which the ImageNet data was trained on.

This was done so that we can pass our images using the **pre-trained VGG model**(which was trained on ImageNet) to get the features. But for this, we first removed the softmax layer as well as the last fully connected layer so as to only get **4096** features as the last layer mostly because the last layers of the network tend to be problem-specific.

I then passed all the images to the pre-trained VGG model to get their features and also saved them in a **.mat file** while taking averages for every 5 images because of FiveCrop as well as saving 25 features in each file as they represent 25 image frames of a particular video. I then loaded them back in a single variable using loadmat function and printed their shape **[2409, 25, 4096] for train and [951, 25, 4096] for test.**

I then again passed them onto the DataLoader to split them into batches and then passed them onto the LSTM model. I referred the LSTM network from this source: https://www.deeplearningwizard.com/deep_learning/practical_pytorch/pytorch_lstm_neuralnetwork/
in addition to the sources provided to us http://colah.github.io/posts/2015-08-Understanding-LSTMs/ as well as official PyTorch documentation. LSTMs handle sequential information well due to the sequential nature of passing frames as their foundational approach.

Lastly, I tested the SVM one vs all based classifier (**LinearSVC**) and had to reshape the data because it took only 2 dimensions as opposed to LSTM's 3. The shapes of train and test data for this were **[2409, 102400] and [951, 102400]** respectively.

Experiments:

1. I first tried to use CenterCrop instead of FiveCrop for getting the features and predicting using LSTM but the accuracy was very low, understandably, due to loss of critical information from the original 256x340 to 224x224 images.

2. I tried to train the vanilla LSTM network on 15 epochs but could only manage to get around 56 percent accuracy so decided to run the model for around 50 epochs

3. I also changed the learning rate from 0.1 to 0.01 which increased the accuracy manifolds

4. I also tried to create an LSTM model with 2 hidden layers to check for its accuracy but found out that single layered LSTM model performed better.

5. I also tried to fiddle around with LinearSVC's penalty parameter C but found out that c=1 and loss=square_hinged gave the best results.

Results:

1. Center Crop feature extraction, LSTM test accuracy: 41%

2. FiveCrop feature extraction, Vanilla LSTM test accuracy, Learning Rate 0.1, 50 epochs : 69%

3. FiveCrop feature extraction, Vanilla LSTM test accuracy, Learning Rate 0.01, 50 epochs: 79%

4. FiveCrop feature extraction, 2-hidden layer LSTM test accuracy, Learning Rate 0.01, 20 epochs: 59%

5. SVM Test Accuracy default params: 82.42%

6. SVM Test Accuracy C=1.0, loss=squared_hinged: 84.86%