CS671 - Deep Learning and its Applications

Course Instructor : Aditya Nigam (Session : Feb-May 2018)

Assignment-01

Date : March 16, 2018

Note:

1. Answer all questions.
2. You are free to make any reasonable assumption that you may need to logically answer a question.
3. Code has to be well commented and as general as possible.
4. You are expected to submit a README file containing how to run your codes.
5. submit your codes with README file as .zip file. The file name will be: $assignment01\_ < groupid > .zip$
6. Assignment needs to be solved in groups.
7. Use Tensorflow to solve this tutorial work.
8. You can install via pip for ubuntu or via anaconda for windows. (We prefer to use ubuntu).

The goal of this assignment is to learn the usage of session api of tensorflow as discussed in the tutorial on 10 March, 2018

1. **To run a a simple computation using tensorflow session api.**
   (a) You will be given input in a file. The file will contain 10 rows having 2 values in each row seperated by 'space' character.
   (b) Each row corresponds to a datapoint. The first value in a row is 'a' value, and second is 'b' value.
   (c) For each data point you have to evaluate the given computation.

$$e = (a + b) * (b + 1)$$

   (d) The output will be stored in a file where each row will contain final value of the computation of each data point.
   (e) You can only use tensorflow session api for this task.
2. **Making a CNN on cifar-10 dataset.**
   (a) In this problem, we're going to specify a model for you to construct. The goal here isn't to get good performance (that'll be next), but instead to get comfortable with understanding the TensorFlow documentation and configuring your own model.
   (b) Using the code provided in Tutorail-01 as guidance, and using the TensorFlow documentation, specify a model with the following architecture:
   - 7x7 Convolutional Layer with 32 filters and stride of 1
   - ReLU Activation Layer
   - Batch Normalization Layer
   - 2x2 Max Pooling layer with a stride of 2
   - fully connected layer with 1024 output units.
   - ReLU Activation Layer

- Fully connected layer from 1024input units to 10 outputs.
- Softmax Activation Layer for classification.

(c) Use Adam optimizer with categorical crossentropy loss.

(d) You can download the data in pickle format from `https://students.iitmandi.ac.in/~s16007/cifar-10-pyth tar.gz`

(e) It is a 10 class classifictaion dataset. Read the readme file in the data for more details.

(f) The data is already split in training batches and testing batch. Use 80% of training data for training the models, 20% to validate the parameters of the model. Once the model is trained, finally testing it using testing data.

3. **Train a great model on CIFAR-10!**

(a) Now it's your job to experiment with architectures, hyperparameters, loss functions, and optimizers to train a model that achieves $>= 70\%$ accuracy on the validation set of CIFAR-10.

(b) **Things you should try:**

- **Filter Size:** Above we used 7x7; this makes pretty pictures but smaller filters may be more efficient
- **Number of Filters:** Above we used 32 filters. Do more or fewer do better?
- **Pooling vs Strided Convolution:** Do you use max pooling or just stride convolutions?
- **Network architecture:** The network above has two layers of trainable parameters. Can you do better with a deep network?
- **Network architecture:** The network above has two layers of trainable parameters. Can you do better with a deep network?
- **Use Learning Rate Decay:** Decaying the learning rate might help the model converge. Feel free to decay every epoch, when loss doesn't change over an entire epoch, or any other heuristic you find appropriate.
- **Regularization:** Use dropouts or l2 weight regularization.

(c) **What we expect**: At the very least, you should be able to train a ConvNet that gets at $>= 70\%$ accuracy on the validation set. This is just a lower bound - if you are careful it should be possible to get accuracies much higher than that! Extra credit points will be awarded for particularly high-scoring models or unique approaches.