06.12.2018

CmpE 480 - Introduction to A.I. Programming Assignment 3

Deep Neural Networks

Deadline: 06.01.2019, 23:59

PART 1 - Deep Neural Networks (40 points)

Follow the steps below **incrementally** and after completing each step:

- report the total training and test times as well as training and test accuracies,
- explain the changes occured in your report.

1. Install tensorflow on your computer (the easiest option is using a docker image on a linux operating system https://www.tensorflow.org/install/docker). Learn how to save your changes as docker images to continue later. Else you may lose your changes. If your computer has a powerful GPU, you may use a GPU supported docker image.
2. Complete the tutorial on Fashion MNIST dataset using tensorflow and keras libraries as shown: https://www.tensorflow.org/tutorials/keras/basic_classification
   NOTE: Change evaluation metric to **categorical_accuracy**.
   ```
   model.compile(optimizer=tf.train.AdamOptimizer(),
                 loss='sparse_categorical_crossentropy',
                 metrics=[categorical_accuracy])
   ```
3. **Weight Initializer:** Compare using random_normal and random_uniform as your kernel_initializer parameters in your Dense layer.
4. **Preprocessing:** Subtract the mean value of training images from every pixel in every image in both train and test data to shift the total mean to 0.
5. **Convolutional Layer:** As the first layer add a new 3x3 convolutional layer with 128 filters. NOTE: For adding a convolutional layer you need to reshape your data to 4 dimensions (60000, 28,28) -> (60000, 28, 28, 1).

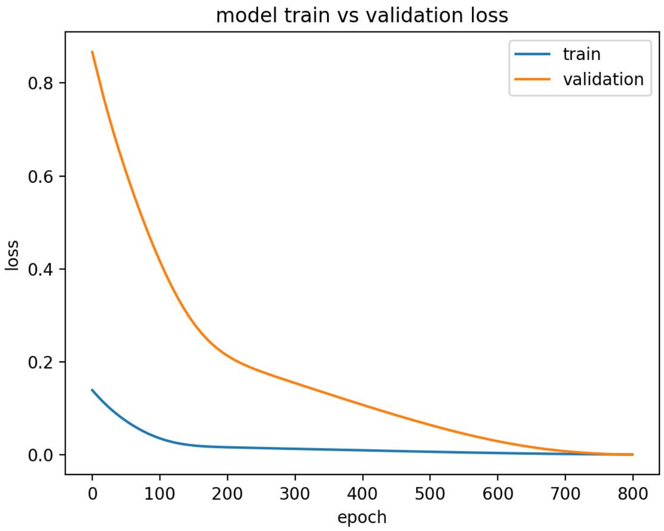
6. **Training & Validation**: Split the training data into 30% validation and 70% training. Add an EarlyStopping Keras callback with appropriate parameters to stop the training epochs if validation accuracy has not improved for 3 epochs (run your network training for 50 epochs if no early stop happens. Use the fit_generator function instead of fit).
7. **Data Augmentation:** Augment the training data by adding only horizontal flips of the training images. Also experiment with augmenting data with only vertical flips. Explain which one gives the best results. Explain your reasoning. You may use keras.preprocessing.image.ImageDataGenerator class. After this step continue with augmenting the data with horizontal flips only.
8. **Different Optimizer:** Change the optimizer to use Stochastic Gradient Descent (SGD) with the following parameters: (lr=0.1, momentum=0.7, decay=0.01, nesterov=True) Explain each of these parameters.

   NOTE: Sometimes SGD starts from a very bad location and the accuracy may stuck around 10%. If that happens, restart the training procedure. Since the algorithm is stochastic, after some restarts you will get a good starting point. If that will not happen, try to change the parameters.
9. **Regularization:** Add DropOut layer(s) for regularization where it makes sense.
10. **Batch Normalization:** Add BatchNormalization layer(s) where it makes sense.
11. **Loss Function:** Change sparse_categorical_crossentropy loss to categorical_crossentropy loss first. Experiment. Then change it to cosine_proximity loss. Experiment. Explain which loss function fits the data best and why?

PART 2 - Experiment (60 points)

Experiment with different layers, architecture, optimizers, activation functions (sigmoid, tanh, etc.), batch sizes (default batch_size is 32), augmentation and regularization techniques. Try to achieve test accuracy of 93%, or as close as possible. Top 3 students with the best test accuracy will receive 10 points bonus. The only restriction is not showing the test data when training your network. If you include test data into your training or validation set, your scores will not be valid. **Use the test data only for evaluation.**

Report your experiments in detail. Provide explanations of your choices. Also explain which experiments failed (or gave worse results) and which ones succeeded. Why? You should include at least 2 training/validation loss graphs (as shown below) in your report when comparing the experiments.



What to submit

- For each experiment use separate cells in your ipython notebook. Export it as html. Combine your ipython notebook and its html version into one zip file as <your_student_id>.zip. Send it to metehandoyran@gmail.com with the subject: "CMPE 480 Programming Assignment 3"
  (Note that your e-mails will be filtered by the subject)
- In your report explain your code and all the different steps you completed in detail.
- Write your report on your dokuwiki page at: http://robot.cmpe.boun.edu.tr/~cmpe480/doku.php?id=user:students