Machine Learning (CS 419/519)
Assignment 04 (40 points)
Due 5:00 PM, Friday, 16 November 2018

In this assignment, you will be using the Keras and TensorFlow systems installed on the CS Volta machine to do some exploratory work with neural networks. You will be handing in two completed Python notebooks, produced using Jupyter.

First-time set-up: To use the existing tools installed on Volta, you will be starting your interactions from the command line in each session, although once you have your notebook up and running, there is generally no need to do any work directly from the command line. Before you begin, you must do a bit of configuration, so that you can properly use the Jupyter notebook (you will only need to do this the first time, after that it will be configured for use):

1. Establish an `ssh` connection with the server at `volta.cs.uwlax.edu`, using your UWL log-in and password. (This will work from any standard SSH client, so long as you are either on the campus network, or using the UWL VPN.)

2. To set up your configuration for Jupyter, execute the following two commands, from the home directory to which you log in:

   ```
   mkdir .jupyter
   nano .jupyter/jupyter_notebook_config.py
   ```

   (Note: be sure to include the dot before `jupyter`). This will create a hidden directory containing a new Jupyter configuration file. You can then paste in the contents of the matching configuration file that has been supplied to you with this assignment. After pasting into the file, exit by choosing `Control-X` and hitting return to confirm saving its contents.

Starting a Jupyter session: Once your account is configured, you will begin a work-session by using SSH, as before, to log into `volta.cs.uwlax.edu`. Once you have done so:

1. Launch Jupyter with the command:

   ```
   jupyter notebook &
   ```

   (note that the use of the ampersand allows you to continue to access the command line if necessary while the Jupyter server is running).

2. When Jupyter launches, you will get a message terminating in a URL of a form similar to:

   ```
   http://volta:8892/?token=be6912...
   ```

This indicates the port number on which Volta is communicating, and includes a long token value used for authentication. Copy and paste this URL into a web browser; edit it so that the full server, `volta.cs.uwlax.edu` is used, instead of simply `volta`. This will cause your browser to open a window giving you a view onto your CS account file-system; this is the same file-system that is mounted if you log into a machine in the CS lab in 16 Wing. While any files you create will live in that file-system, any compute work you do will be on the more powerful Volta machine itself.

3. You will now do all your work from within the Jupyter system in your browser. See the video posted to D2L with this assignment for basic use. (There are many short tutorial posts and videos available for Jupyter online as well, if you want more information about its use.)

Assignment: You will be completing two distinct exercises in Jupyter. These will involve doing some basic and more complex machine learning tasks, and documenting some of what you saw. All of your work will end up in two different Jupyter notebook files, which you will turn in for credit.

1. (*5 pts.*) Choose a working directory on your file system. Upload the two files contained in the `diabetes_exercise` folder in the assignment materials. Opening the notebook file (ending in suffix `.ipynb`) will open a new browser tab containing a complete, executable Jupyter notebook. Read through the material in the notebook, executing the code as you go to build a simple feed-forward classifier for the diabetes test-data supplied.

   As you go, note the following:

   (a) The code contains a section that limits the memory footprint of TensorFlow. This code restricts anything you execute to use only 5% of the memory associated with each of the GPUs on Volta. This means that each GPU will only allot you 808 MB for use. For these exercises, this should be sufficient, and should allow reasonable numbers of users to be on the system at the same time. If this turns out to be less memory than is required, we can talk about it.

   Note: If you write your own code on the system, you need to remember to include the same restrictions on memory use. If you do not, you will monopolize the entire memory of the system, and nobody else will be able to use it. Thus, please don't do it. If our system monitors indicate this is happening regularly, we will restrict the system to users who behave more kindly to others.

   (b) The notebook contains sections written in Markdown syntax. "Running" such a section simply renders the Markdown, making it look pretty. You can edit both code and Markdown by clicking into the relevant cells and making changes.

   For this part of the assignment, you are directed to make some changes to the model that is built and trained. As you do, you will be editing the Markdown section, replacing the instructions with an explanation of what you saw. Your grade for this part will be based on the quality and accuracy of your explanation.

   If you have not used Markdown before, it is quite simple. The examples given show much of its basic syntax. More information can be found at numerous online sources, as for example: https://www.markdownguide.org/cheat-sheet/

2. (*35 pt.*) For the second part of the assignment, you will create a new Jupyter notebook. In that notebook, you will be working on a convolutional neural network for the CIFAR-10 object-recognition task. The database for this well-known benchmark is available directly in Keras. For this part of the assignment:

   (a) Begin a new notebook. As you start to build your code, remember to include the code from the previous notebook that restricts memory usage.

   (b) There are a number of online tutorials to get started with CIFAR-10. Start by working through all of the steps in the following: `http://tinyurl.com/cifar10-exercise`

   (c) As you work through the exercise, add Markdown cells between the various major pieces of code (as in the notebook supplied you for the first part of this assignment). Each Markdown cell should contain some explanatory text, indicating what each piece of the code is doing and why. Your grade will depend upon the quality of these explanations.

   (d) The tutorial, as given, works through building a classifier that gets approximately 80% accuracy. At the end of the exercise are some suggestions on how it might be improved. Work through some of these suggestions, or try other techniques of your own, and see if you can improve the classifier performance. In the notebook Markdown sections, document things you have tried, and the results (good or bad). For this part of the assignment, you will be graded not based upon the final result (i.e., classifier accuracy improvements), but upon the clarity of your explanations and evidence of having tried something genuinely interesting.

You will hand in a single compressed archive containing the two completed notebook files.