CS 461
Program 4
Due Friday night, May 10.

For the last program of the semester, you’ll be using TensorFlow to build a neural network to analyze a data set from a health study. This is a small data set—only about 300 cases—so we’re not expecting extremely high accuracy. This is a subset of data from a large general-health study carried out by the Cleveland Clinic. You will be using various health measures to classify and predict the presence or absence of heart disease.

The data file itself is a CSV file; the data dictionary is an attached text file. Note that the data dictionary says the target variable (‘target’, indicating the presence or absence of heart disease) is coded 1-4; this data file has been reduced further and is coded 0-1.

There is no specific guidance provided for how big or complex your network should be, but here are some general principles:

- Don’t get carried away. For a data set this small, there’s probably no need for more than 1 or 2 hidden layers. There is no need for convolutional layers or time-lag data (in fact, the data won’t support those transformations).
- For data involving category classification (sex, blood sugar elevated yes/no, etc), use one-hot coding. For data involving true numbers (age, cholesterol, resting heart rate, etc), normalize the data. This puts everything onto similar scales and reduces the amount of learning the network has to do. Use the TensorFlow documentation for guidance on how to do this—it’s a line or 2 of code.
- Use softmax on the output (classification) layer to estimate the probability that each case has heart disease.
- Use 70% of the data as a training set, 15% as a test set, and 15% as a validation set. The training strategy you use (k-fold cross-validation, leave-out-one, etc) is up to you.
- Try varying the configuration of your network—Rectified Linear v. Sigmoids v. Linear, more or fewer hidden layers, hidden layers being smaller or larger, whatever. Play around with it a little.

Submit:

- Your TensorFlow source code.
- A short report describing the configurations of your networks and the results you got. Whichever of your networks got the best results, discuss why you think it got the best results and what changes might work even better.