

CS559 Project Proposal

Amie Roten

For my term project, I aim to implement a multilayer perceptron/neural network using backpropagation, and plan to construct it in a flexible manner such that the user can select the number of hidden layers and nodes/units for each layer, as well as the type of output layer (binary, softmax, etc.). Although this is not an especially novel project, after working with Python packages such as Keras to implement neural networks for previous projects, I think it is a great opportunity to get a good grasp on what is happening under the hood.

I'd like to specifically use this framework to do a simple speech recognition task, most likely attempting to classify speech based on the emotion being conveyed, using the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). I'm interested in speech signal processing, so I chose a speech-related task in order to not only get a chance to implement a neural network, but to also get experience preprocessing the audio files in order to create feature vectors for input to the model. I initially had hoped to evaluate the network on a phoneme recognition task, but most corpora I came across (that were free) did not include phonetic transcriptions. Additionally, considering how many phonemes there are in English, it seemed like it might be tricky to train a hand-built neural network on that many classes to a point where I could safely say that it was working. So, I decided on the eight-class emotion classification task instead!

I do have some fallback data (a binary classification task) that I've trained a Keras model on previously, in case the speech recognition task proves too complex for the project timeframe. I plan to begin by implementing the neural network, and testing it on this task, before moving on to the somewhat more complicated task. I'll consider the project a success (in my own eyes!) if I'm able to achieve roughly equivalent performance, in terms of accuracy, to the Keras implementation. I aim to keep efficiency in mind as well as I build the project, though I doubt I'd be able to achieve the same efficiency as Keras' version...we will find out!