Project Proposal

STAT 5600 - Methods in Statistical Learning

Aaron Davis

October 23rd, 2022

Neural networks of every sort are composed of some linear combination of features which is then transformed by some non-linear function (called an activation function) to learn a more complex non-linear function that relates some feature set (X) to some output (Y) such that

$$Y \approx f(X)$$

Two of the most common activation functions to use in the hidden layers of a neural network are ReLU (rectified linear unit) and Sigmoid activations where

$$ReLU(z) = \max\{0, z\}$$

$$Sigmoid(z) = \frac{1}{1 + e^{-z}}$$

But how do we choose better activation functions? What if a different activation function would allow me to estimate f(X) just as accurately, but with fewer computations during inference? How would we go about finding such an activation function?

The focus of this project is training and compressing better activation functions for neural networks, such that we can achieve the same accuracy with fewer parameters and fewer computations at inference time.

An activation function is just some non-linear function that is applied to a linear combination of the past layer's outputs (or the inputs to the model, depending on where you're at in the model). Now I want to learn an activation function instead of assume one, and the best way I know of to approximate some unknown function of unknown complexity is neural networks.

So essentially we'll be training an activation function using a portion of our neural network with one input and one output, and then we'll take that activation function and approximate it using splines or some other similar function estimation technique. The goal of the project is to determine if some metrics proposed for model evaluation can be better achieved using learned activation functions than assumed activation functions over the same number of training epochs.

The data used for this project is of little importance, but we'll focus on using CIFAR-10, mnist, Titanic, and other well known datasets for our project benchmarks.

When testing on classification problems, we'll use the Adam optimizer, softmax activation (for final layer), and categorical_crossentropy loss. All of the neural network parts of the project will be done in keras in R, and all the compression portions will be done in tidy models.