For my model, 4 layers in total(one input, 2 hidden, one output) I used a functional Keras API to build it and model.fit() to train it. I used 50 neurons in the hidden layer, 11 in the input, and one in the output layer. My activation functions for the first hidden layer, second hidden layer, and output layer were relu, tanh, and softmax respectively. I used the BinaryCrossEntropy loss function and set from\_logits to True so it would apply the Sigmoid activation function internally. For my optimization I used Stochastic Gradient Descent with a learning rate of 0.001. Despite experimenting with a wide range of values,my model’s accuracy sits between 23-25%.