Exercise 3
Jonathan Khashper
212758163

I took the code from the github repo. The implementation of the layers is pretty straightforward.

We will do the exact same thing as in the original, but replace the values where instead of using the first input -> first layer -> output, we will now need input -> first layer -> second layer -> output

1. For the new layer, were going to need new weights and biases. I added them (changed to h1 and h2 respectively)
2. For the forward propagation, add the following two lines:

   ```
   z_h2 = np.dot(a_h1, self.weight_h2.T) + self.bias_h2
   a_h2 = sigmoid(z_h2)
   ```

   This is the calculation for the next layer, exactly like we did for the first.
   Still use the sigmoid function (probably not optimal, but it's the simplest for the changes were making to the original.
3. The back propagation is a bit more complex, but its also not that hard:
   Similar changes to the forward propagation, but instead of output -> first layer, we change the code so it does the same but output -> second, second -> first.
   The only not straightforward change here is using delta_out for the first layer, we now use delta_h2, which uses the result of the second layer back propagation to correct the first layer.
4. Slightly change the return values:

   Forward now returns `a_h1, a_h2, a_out`

   Back now returns
   ```
   (d_loss__dw_out, d_loss__db_out,
    d_loss__d_w_h2, d_loss__d_b_h2,
    d_loss__d_w_h1, d_loss__d_b_h1)
   ```

   That means we also need to correct the mse and acc calculations, to account for the new return values.

Those are all the changes I did, we now have a two layer ANN, and it is slightly better: (Also changed AUC Calculation, cause it didn't exist)

Model	Test Accuracy	Macro AUC
Original (single layer)	Test accuracy: 94.96%	Test macro auc: 0.99462
Two layer	Test accuracy: 95.07%	Test macro auc: 0.99503
Pytorch implementation	90.43%	0.982803

I ran all models for 80 epochs. They had 64 nodes in the first layer, and the two layered ANNs had 32 nodes in the second. I believe that increasing the number of nodes and number of epochs will increase the benefits of the second layer, however this does pose the risk of potentially overfitting the model.