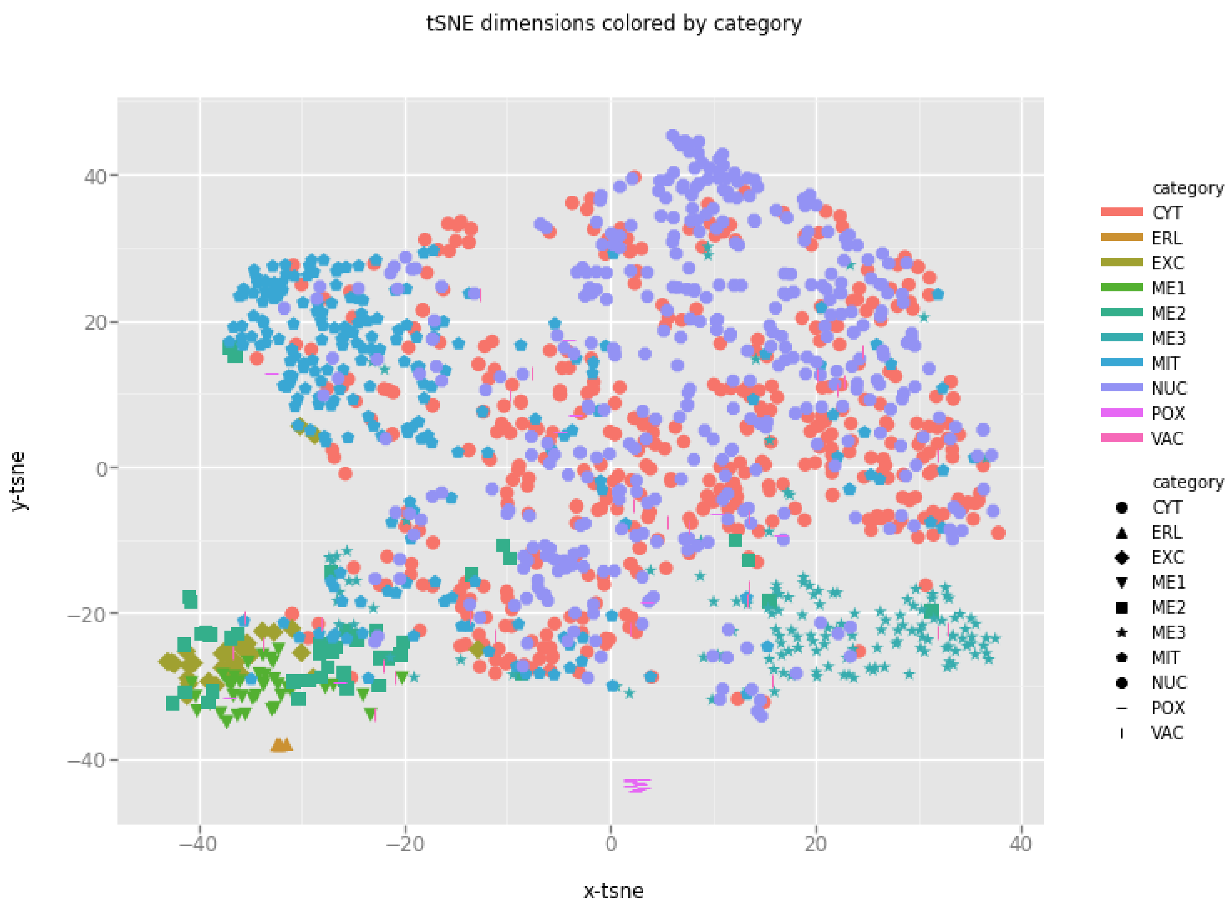
Goal: Build a neural network classifier that can find the localization site of a protein in yeast based on 8 attributes (features).

Problem 1

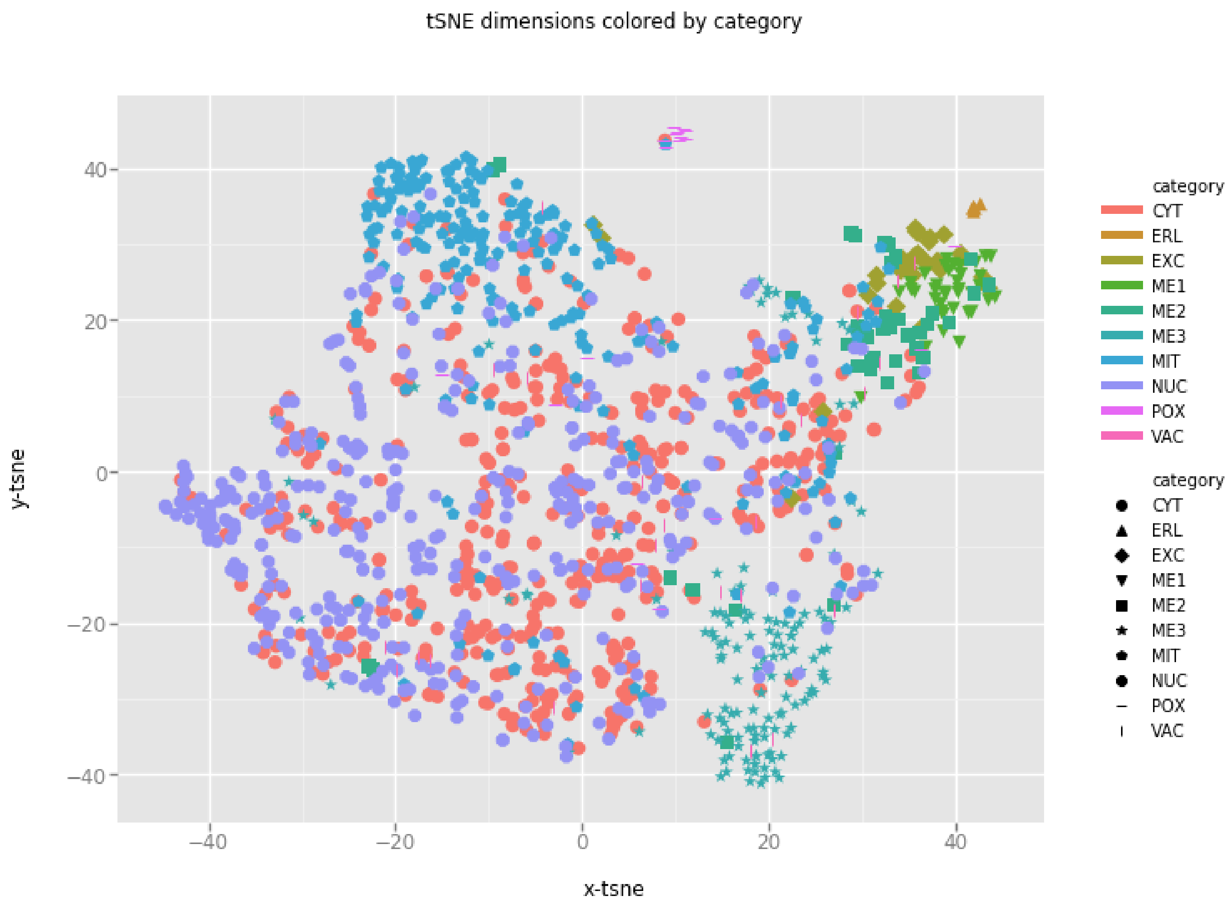
Check whether there are any outliers on the dataset using Local Outlier Factor and Isolation Forest.

The dataset is very imbalance in terms of number of observations per category. For example, there are only 5 observations for Category ERL. n_neighbors = 20 is the recommended value, however, the number of neighbors should not be greater than the samples. Therefore, we will do LocalOutlierFactor for each class to increase its accuracy. We see a similar situation in the isolation forest detection method. We modified the num_max_sample to get a better performance.

LocalOutlierFactor



IsolationForest



(a) Yes, there are outliers on the dataset.

(b) Both methods agree that the dataset contains outliers. However, LocalOutlierFactor removes more outliers than Isolation Forest.

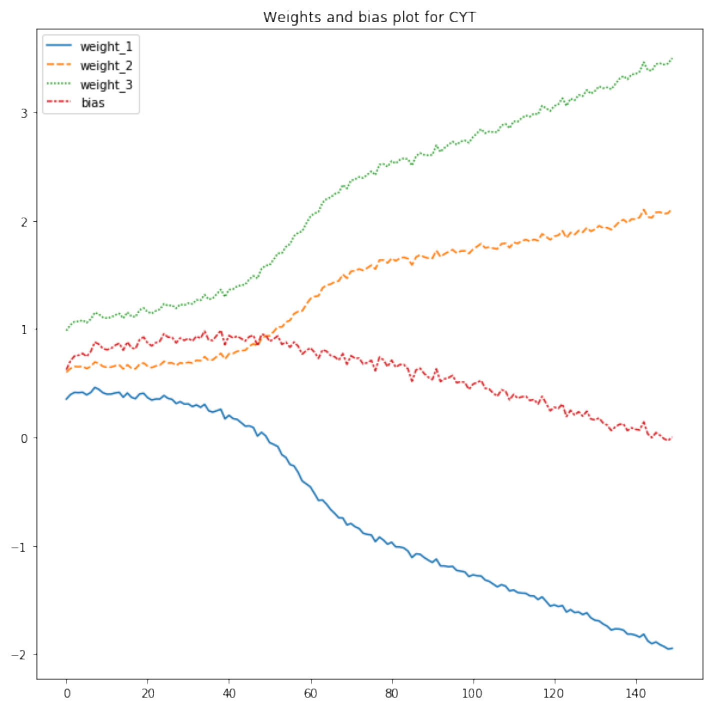
(c) LocalOutlierFactor relies on the assumption that normal data are more concentrated in a certain region while outliers locate in a sparse region.

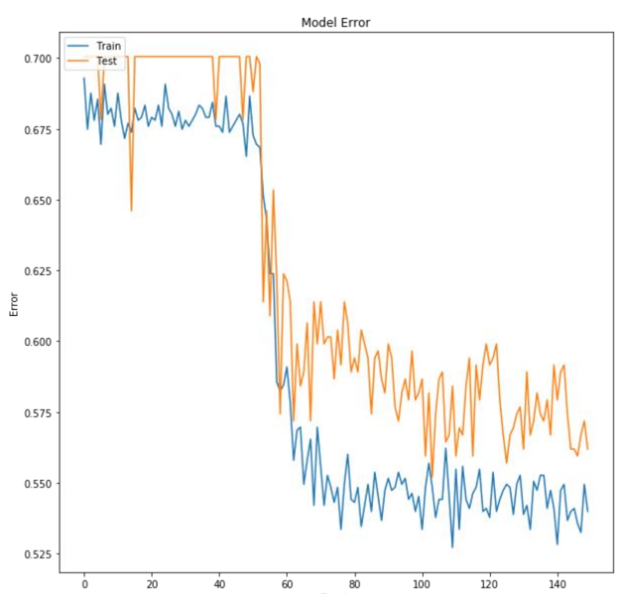
Isolation forest relies on the assumption that an outlier is easier to isolate by lines than normal data.

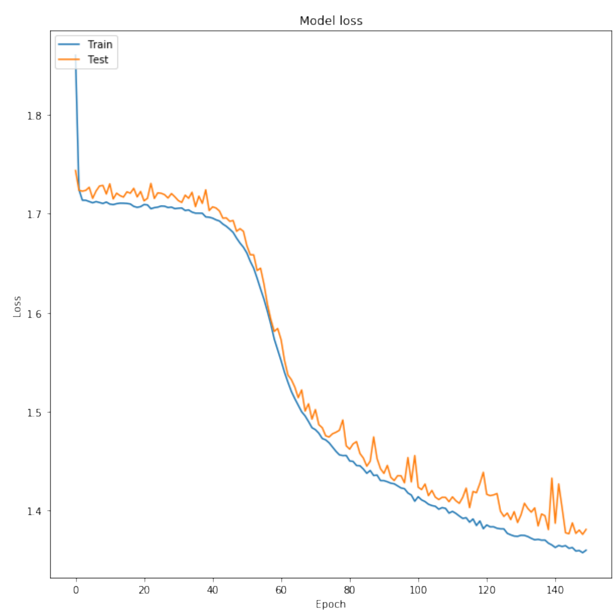
(d) Remove outliers using Isolation Forest.

Problem 2

Construct a 4-layer artificial neural network (ANN) and specifically a feed-forward multi-layer perceptron to perform multi-class classification.







Problem 3

Re-train the ANN with all the data.

After doing 150 epoches, the training error has decreased to 0.457547. Please refer to the jupyter notebook for the error_df.

Here is the final activation function formula for class "CYT":

$$W^{(2)} = \begin{bmatrix} W_{10}^{(2)} & W_{11}^{(2)} & W_{12}^{(2)} & W_{13}^{(2)} \\ W_{20}^{(2)} & W_{21}^{(2)} & W_{22}^{(2)} & W_{23}^{(2)} \\ W_{30}^{(2)} & W_{31}^{(2)} & W_{32}^{(2)} & W_{33}^{(2)} \\ W_{40}^{(2)} & W_{41}^{(2)} & W_{42}^{(2)} & W_{43}^{(2)} \\ W_{50}^{(2)} & W_{51}^{(2)} & W_{52}^{(2)} & W_{53}^{(2)} \\ W_{60}^{(2)} & W_{61}^{(2)} & W_{62}^{(2)} & W_{63}^{(2)} \\ W_{70}^{(2)} & W_{71}^{(2)} & W_{72}^{(2)} & W_{73}^{(2)} \\ W_{80}^{(2)} & W_{81}^{(2)} & W_{82}^{(2)} & W_{83}^{(2)} \\ W_{90}^{(2)} & W_{91}^{(2)} & W_{92}^{(2)} & W_{93}^{(2)} \\ W_{10_0}^{(2)} & W_{10_1}^{(2)} & W_{10_2}^{(2)} & W_{10_3}^{(2)} \end{bmatrix}$$

$$a^{(2)} = \begin{bmatrix} b_1^{(2)} \\ a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix}$$

$$
Z^{(2)} = \begin{bmatrix}
W_{10}^{(2)} * b_1^{(2)} + W_{11}^{(2)} * a_1^{(2)} + W_{12}^{(2)} * a_2^{(2)} + W_{13}^{(2)} * a_3^{(2)} \\
W_{20}^{(2)} * b_1^{(2)} + W_{21}^{(2)} * a_1^{(2)} + W_{22}^{(2)} * a_2^{(2)} + W_{23}^{(2)} * a_3^{(2)} \\
W_{30}^{(2)} * b_1^{(2)} + W_{31}^{(2)} * a_1^{(2)} + W_{32}^{(2)} * a_2^{(2)} + W_{33}^{(2)} * a_3^{(2)} \\
W_{40}^{(2)} * b_1^{(2)} + W_{41}^{(2)} * a_1^{(2)} + W_{42}^{(2)} * a_2^{(2)} + W_{43}^{(2)} * a_3^{(2)} \\
W_{50}^{(2)} * b_1^{(2)} + W_{51}^{(2)} * a_1^{(2)} + W_{52}^{(2)} * a_2^{(2)} + W_{53}^{(2)} * a_3^{(2)} \\
W_{60}^{(2)} * b_1^{(2)} + W_{61}^{(2)} * a_1^{(2)} + W_{62}^{(2)} * a_2^{(2)} + W_{63}^{(2)} * a_3^{(2)} \\
W_{70}^{(2)} * b_1^{(2)} + W_{71}^{(2)} * a_1^{(2)} + W_{72}^{(2)} * a_2^{(2)} + W_{73}^{(2)} * a_3^{(2)} \\
W_{80}^{(2)} * b_1^{(2)} + W_{81}^{(2)} * a_1^{(2)} + W_{82}^{(2)} * a_2^{(2)} + W_{83}^{(2)} * a_3^{(2)} \\
W_{90}^{(2)} * b_1^{(2)} + W_{91}^{(2)} * a_1^{(2)} + W_{92}^{(2)} * a_2^{(2)} + W_{93}^{(2)} * a_3^{(2)} \\
W_{10_0}^{(2)} * b_1^{(2)} + W_{10_1}^{(2)} * a_1^{(2)} + W_{10_2}^{(2)} * a_2^{(2)} + W_{10_3}^{(2)} * a_3^{(2)}
\end{bmatrix}
$$

$$t = e^{Z^{(2)}}$$

$$a^{(2)} = \frac{e^{Z^{(2)}}}{\sum_{j=1}^{n} t_i}$$

$$t = e^{Z_1^{(2)}}$$

$$\sum_{j=1}^{n} t_i = e^{Z_1^{(2)}} + e^{Z_2^{(2)}} + e^{Z_3^{(2)}} + e^{Z_4^{(2)}} + e^{Z_5^{(2)}} + e^{Z_6^{(2)}} + e^{Z_7^{(2)}} + e^{Z_8^{(2)}} + e^{Z_9^{(2)}} + e^{Z_{10}^{(2)}}$$

$$a_{CYT}^{(2)} = \frac{e^{Z_1^{(2)}}}{e^{Z_1^{(2)}} + e^{Z_2^{(2)}} + e^{Z_3^{(2)}} + e^{Z_4^{(2)}} + e^{Z_5^{(2)}} + e^{Z_6^{(2)}} + e^{Z_7^{(2)}} + e^{Z_8^{(2)}} + e^{Z_9^{(2)}} + e^{Z_{10}^{(2)}}}$$

Problem 4

Do back-propagation on the ANN using Keras and by hand.

```
model_back.add(Dense(3, input_dim = X_matrix.shape[1], activation = 'sigmoid', kernel_initializer = 'zeros', bias_initi
model_back.layers[0].get_weights()
```

```
Out[41]: [array([[0., 0., 0.],
                 [0., 0., 0.],
                 [0., 0., 0.],
                 [0., 0., 0.],
                 [0., 0., 0.],
                 [0., 0., 0.],
                 [0., 0., 0.],
                 [0., 0., 0.]], dtype=float32), array([1., 1., 1.], dtype=float32)]
```

```
In [42]: # Hidden Layer 2
         model_back.add(Dense(3, activation = 'sigmoid', bias_initializer= 'ones'))
         model_back.layers[1].get_weights()
```

```
Out[42]: [array([[ 0.2885157 ,  0.15007329, -0.2697848 ],
                 [-0.84490645,  0.51890874, -0.47144252],
                 [-0.33753812,  0.9407184 ,  0.54724324]], dtype=float32),
          array([1., 1., 1.], dtype=float32)]
```

```
In [43]: # Output Layer
         # input_dim = # of output
         model_back.add(Dense(Y_matrix.shape[1], activation = 'softmax', bias_initializer= 'ones'))
         model_back.layers[2].get_weights()
```

```
Out[43]: [array([[ 0.5371748 , -0.26527536,  0.08623469, -0.29283202, -0.5771761 ,
                   0.0364961 , -0.15968442, -0.19208124,  0.54288864,  0.59075224],
                 [ 0.03545719, -0.0664953 , -0.4742814 , -0.03014451, -0.1657862 ,
                  -0.03368115, -0.05236948, -0.03004026, -0.2806594 , -0.29026085],
                 [-0.58669925,  0.33548915, -0.47667387, -0.6716246 ,  0.4851929 ,
                  -0.05356944, -0.23507911,  0.22114581, -0.46402645, -0.40377453]],
               dtype=float32),
          array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.], dtype=float32)]
```

```python
# output
layer_1
```

```
array([[0.73099977, 0.7310842 , 0.73117375]], dtype=float32)
```

```python
# output
layer_2
```

```
array([[0.5850712, 0.8981969, 0.7030989]], dtype=float32)
```

```python
# output
layer_3
```

```
array([[0.11438229, 0.12475157, 0.06022668, 0.06268091, 0.10573442,
        0.11665708, 0.0901494 , 0.13088317, 0.09428091, 0.10025359]],
      dtype=float32)
```

In [175]:

```python
# z1^(3)
layer_activation(4, 5, layer_2)
```

Out[175]:

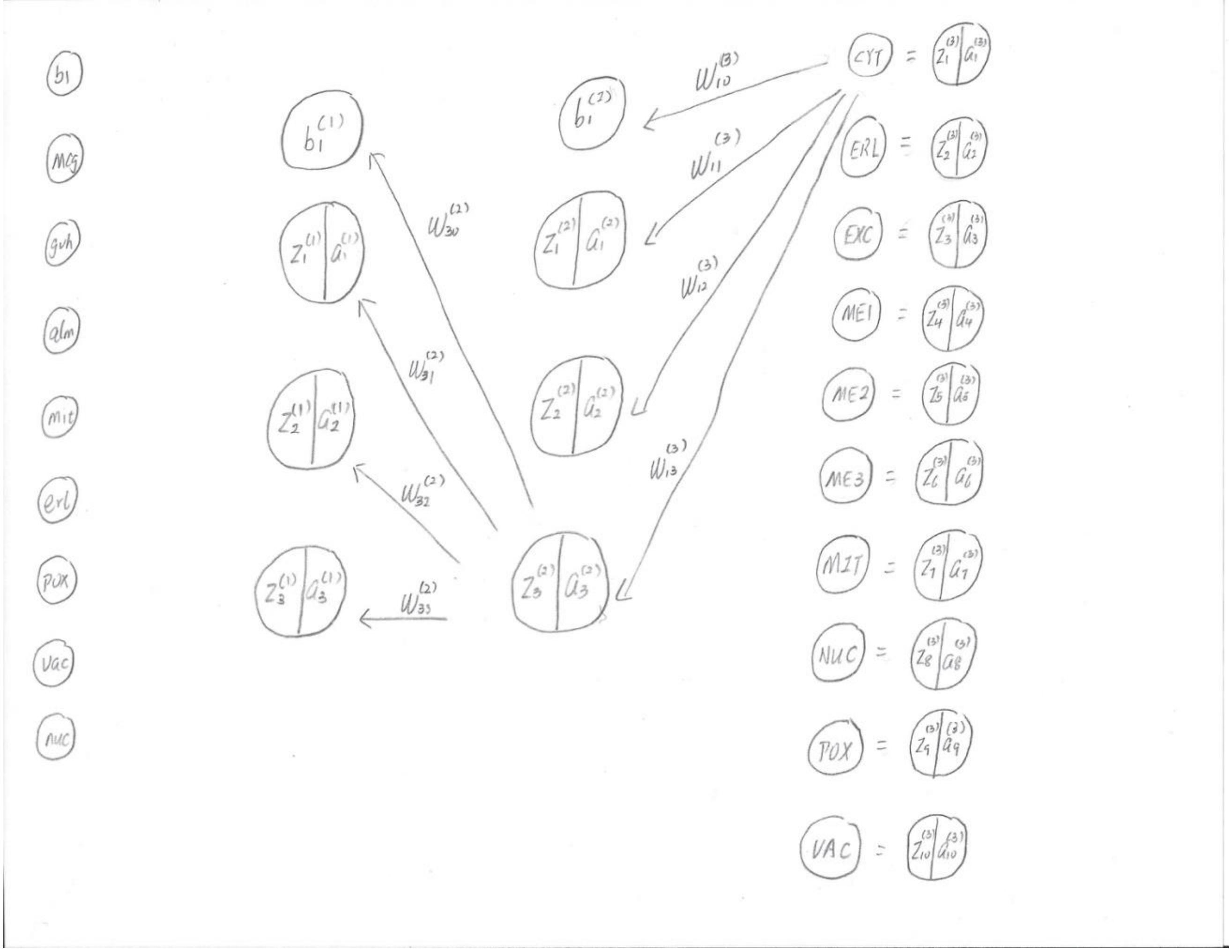
```
array([0.93362547, 1.02095121, 0.28930649, 0.3293782 , 0.85454082,
       0.9534359 , 0.69425128, 1.0161241 , 0.73928463, 0.80102731])
```

In [189]:

```python
# z3^(2)
layer_activation(2, 3, layer_1)[2]
```

Out[189]:

```
0.8582530690265031
```



b1
mcg
gvh
alm
mit
erl
pox
vac
nuc
CYT
ERL
EXC
ME1
ME2
ME3
MIT
NUC
POX
VAC

Backpropagation : Output Layer → Second hidden Layer

$$\frac{\partial E_1}{\partial a_1^{(3)}} = -1\left(\left(y_1 \times \frac{1}{a_1^{(3)}}\right) + (1-y_1) \times \left(\frac{1}{(1-a_1^{(3)})}\right)\right)$$

$$\frac{\partial a_1^{(3)}}{\partial Z_1^{(3)}} = \frac{(e^{Z_1^{(3)}} \times (e^{Z_2^{(3)}} + e^{Z_3^{(3)}} + e^{Z_4^{(3)}} + e^{Z_5^{(3)}} + e^{Z_6^{(3)}} + e^{Z_7^{(3)}} + e^{Z_8^{(3)}} + e^{Z_9^{(3)}} + e^{Z_{10}^{(3)}})}{(e^{Z_1^{(3)}} + e^{Z_2^{(3)}} + e^{Z_3^{(3)}} + e^{Z_4^{(3)}} + e^{Z_5^{(3)}} + e^{Z_6^{(3)}} + e^{Z_7^{(3)}} + e^{Z_8^{(3)}} + e^{Z_9^{(3)}} + e^{Z_{10}^{(3)}})^2}$$

$$\frac{\partial Z_1^{(3)}}{\partial W_{10}^{(3)}} = b_1^{(2)} \qquad \frac{\partial Z_1^{(3)}}{\partial W_{11}^{(3)}} = a_1^{(2)} \qquad \frac{\partial Z_1^{(3)}}{\partial W_{12}^{(3)}} = a_2^{(2)} \qquad \frac{\partial Z_1^{(3)}}{\partial W_{13}^{(3)}} = a_3^{(2)}$$

$$\delta W_{jk}^{(3)} = \begin{bmatrix} \frac{\partial E_1}{\partial a_1^{(3)}} \times \frac{\partial a_1^{(3)}}{\partial Z_1^{(3)}} \times \frac{\partial Z_1^{(3)}}{\partial W_{10}^{(3)}} \\ \frac{\partial E_1}{\partial a_1^{(3)}} \times \frac{\partial a_1^{(3)}}{\partial Z_1^{(3)}} \times \frac{\partial Z_1^{(3)}}{\partial W_{11}^{(3)}} \\ \frac{\partial E_1}{\partial a_1^{(3)}} \times \frac{\partial a_1^{(3)}}{\partial Z_1^{(3)}} \times \frac{\partial Z_1^{(3)}}{\partial W_{12}^{(3)}} \\ \frac{\partial E_1}{\partial a_1^{(3)}} \times \frac{\partial a_1^{(3)}}{\partial Z_1^{(3)}} \times \frac{\partial Z_1^{(3)}}{\partial W_{13}^{(3)}} \end{bmatrix} = \begin{bmatrix} \delta W_{10}^{(3)} \\ \delta W_{11}^{(3)} \\ \delta W_{12}^{(3)} \\ \delta W_{13}^{(3)} \end{bmatrix}$$

lr = learning rate

$$W^{(3)\prime} = \begin{bmatrix} W_{10}^{(3)} - lr \times \delta W_{10}^{(3)} \\ W_{11}^{(3)} - lr \times \delta W_{11}^{(3)} \\ W_{12}^{(3)} - lr \times \delta W_{12}^{(3)} \\ W_{13}^{(3)} - lr \times \delta W_{13}^{(3)} \end{bmatrix}$$

$$\frac{\partial E_1}{\partial a_1^{(3)}} = -1\left(\left(y_1 \times \frac{1}{a_1^{(3)}}\right) + (1-y_1) \times \left(\frac{1}{(1-a_1^{(3)})}\right)\right)$$

$a_1^{(3)}$ = layer 3

$$\frac{\partial E_1}{\partial a_1^{(3)}} = -1\left(\frac{1}{1-0.11438229}\right)$$

$$= -1.1291554$$

$$\frac{\partial a_1^{(3)}}{\partial z_1^{(3)}} = \frac{\left(e^{z_1^{(3)}} \times \left(e^{z_2^{(3)}} + e^{z_3^{(3)}} + e^{z_4^{(3)}} + e^{z_5^{(3)}} + e^{z_6^{(3)}} + e^{z_7^{(3)}} + e^{z_8^{(3)}} + e^{z_9^{(3)}} + e^{z_{10}^{(3)}}\right)\right)}{\left(e^{z_1^{(3)}} + e^{z_2^{(3)}} + e^{z_3^{(3)}} + e^{z_4^{(3)}} + e^{z_5^{(3)}} + e^{z_6^{(3)}} + e^{z_7^{(3)}} + e^{z_8^{(3)}} + e^{z_9^{(3)}} + e^{z_{10}^{(3)}}\right)^2}$$

$z_1^{(3)}$ = layer 2 × model_back.layer[2].get_weights()

= [0.93362547, 1.02095121, 0.28930649, 0.3293782, 0.85454082, 0.9534359, 0.69425128, 1.0161241, 0.73928463, 0.80102731]

$$\frac{\partial a_1^{(3)}}{\partial z_1^{(3)}} = \frac{\left[\left(e^{0.93362547} \times \left(e^{1.02095121} + e^{0.28930649} + e^{0.3293782} + e^{0.85454082} + e^{0.9534359} + e^{0.69425128} + e^{1.0161241} + e^{0.73928463} + e^{0.80102731}\right)\right]\right.}{\left(\sum_{z=1}^{z=10} e^{z_i^{(3)}}\right)^2}$$

$$= \frac{2.54371464 \cdot 19.533284}{(22.077)^2} = 0.10194$$

$$\frac{\partial Z_1^{(3)}}{\partial W_{10}^{(3)}} = 1 \qquad \frac{\partial Z_1^{(3)}}{\partial W_{11}^{(3)}} = 0.5850712 \qquad \frac{\partial Z_1^{(3)}}{\partial W_{12}^{(3)}} = 0.8981969 \qquad \frac{\partial Z_1^{(3)}}{\partial W_{13}^{(3)}} = 0.7030989$$

$$\delta W_{jk}^{(3)} = \begin{bmatrix} -1.129 \times 0.10194 \times 1 \\ -1.129 \times 0.10194 \times 0.585 \\ -1.129 \times 0.10194 \times 0.898 \\ -1.129 \times 0.10194 \times 0.703 \end{bmatrix} = \begin{bmatrix} -0.115 \\ -0.067 \\ -0.1034 \\ -0.081 \end{bmatrix}$$

$lr = 1$

$$W^{(3)'} = \begin{bmatrix} 1 + 1 \times (-0.115) \\ 0.5372 + 1 \times (-0.067) \\ 0.0355 + 1 \times (-0.1034) \\ -0.5867 + 1 \times (-0.081) \end{bmatrix}$$

$$= \begin{bmatrix} 0.885 \\ 0.4702 \\ -0.0679 \\ -0.6677 \end{bmatrix}$$

Backpropagation Second Hidden Layer → First hidden Layer

$$\frac{\partial E_1}{\partial a_3^{(2)}} = \frac{\partial E_1}{\partial a_1^{(3)}} \cdot \frac{\partial a_1^{(3)}}{\partial Z_1^{(3)}} \cdot W_{13}^{(3)}$$

$$\frac{\partial a_3^{(2)}}{\partial Z_3^{(2)}} = \text{Sigmoid}\,(Z_3^{(2)}) \times (1 - \text{Sigmoid}\,(Z_3^{(2)})$$

$$\frac{\partial Z_3^{(2)}}{\partial W_{30}^{(1)}} = b_1^{(1)} \qquad \frac{\partial Z_3^{(2)}}{\partial W_{31}^{(2)}} = a_1^{(1)} \qquad \frac{\partial Z_3^{(2)}}{\partial W_{32}^{(2)}} = a_2^{(1)} \qquad \frac{\partial Z_3}{\partial W_{33}^{(2)}} = a_3^{(1)}$$

$$\delta W_{jk}^{(2)} = \begin{bmatrix} \frac{\partial E_1}{\partial a_3^{(2)}} \cdot \frac{\partial a_3^{(2)}}{\partial Z_3^{(2)}} \cdot \frac{\partial Z_3^{(2)}}{\partial W_{30}^{(2)}} \\ \frac{\partial E_1}{\partial a_3^{(2)}} \cdot \frac{\partial a_3^{(2)}}{\partial Z_3^{(2)}} \cdot \frac{\partial Z_3^{(2)}}{\partial W_{31}^{(2)}} \\ \frac{\partial E_1}{\partial a_3^{(2)}} \cdot \frac{\partial a_3^{(2)}}{\partial Z_3^{(2)}} \cdot \frac{\partial Z_3^{(2)}}{\partial W_{32}^{(2)}} \\ \frac{\partial E_1}{\partial a_3^{(2)}} \cdot \frac{\partial a_3^{(2)}}{\partial Z_3^{(2)}} \cdot \frac{\partial Z_3^{(2)}}{\partial W_{33}^{(2)}} \end{bmatrix} = \begin{bmatrix} \delta W_{30}^{(2)} \\ \delta W_{31}^{(2)} \\ \delta W_{32}^{(2)} \\ \delta W_{33}^{(2)} \end{bmatrix}$$

$$W^{(2)'} = \begin{bmatrix} W_{30}^{(2)} - lr \times \delta W_{30}^{(2)} \\ W_{31}^{(2)} - lr \times \delta W_{31}^{(2)} \\ W_{32}^{(2)} - lr \times \delta W_{32}^{(2)} \\ W_{33}^{(2)} - lr \times \delta W_{33}^{(2)} \end{bmatrix}$$

$$\frac{\partial E_1}{\partial a_3^{(2)}} = \frac{\partial E_1}{\partial a_1^{(3)}} \cdot \frac{\partial a_1^{(3)}}{\partial z_1^{(3)}} \cdot W_{13}^{(3)}$$

$$= -1.129 \cdot 0.10194 \cdot (-0.5883)$$

$$= 0.0677$$

$$\frac{\partial a_3^{(2)}}{\partial z_3^{(2)}} = \text{Sigmoid}(Z_3^{(2)}) \times (1 - \text{Sigmoid}(Z_3^{(2)}))$$

$$= \text{Sigmoid}(0.85825307) \times (1 - \text{Sigmoid}(0.85825307))$$

$$= 0.2091$$

$$\frac{\partial Z_3^{(2)}}{\partial W_{30}^{(1)}} = 1 \qquad \frac{\partial Z_3^{(2)}}{\partial W_{31}^{(2)}} = 0.73099977 \qquad \frac{\partial Z_3^{(2)}}{\partial W_{32}^{(2)}} = 0.7310842 \qquad \frac{\partial Z_3^{(2)}}{\partial W_{33}^{(2)}} = 0.73117375$$

$$\delta W_{jk}^{(2)} = \begin{bmatrix} 0.0677 \times 0.2091 \times 1 \\ 0.0677 \times 0.2091 \times 0.7309 \\ 0.0677 \times 0.2091 \times 0.7311 \\ 0.0677 \times 0.2091 \times 0.7312 \end{bmatrix} = \begin{bmatrix} 0.0142 \\ 0.0103 \\ 0.0103 \\ 0.0104 \end{bmatrix}$$

$$W^{(2)'} = \begin{bmatrix} 1 - 0.0142 \\ 0.2885 - 0.0103 \\ -0.8449 - 0.0103 \\ -0.3375 - 0.0104 \end{bmatrix} = \begin{bmatrix} 0.9858 \\ 0.2782 \\ -0.8552 \\ -0.3479 \end{bmatrix}$$

Problem 5

Perform a parameter sweep (grid search) on the number of hidden layers (investigate layer 1, 2 or 3) and number of nodes in each hidden layer (investigate node 3, 6, 9, 12).

The optimal configuration is {'num_node': 9, 'num_layers': 1} using the RandomizedSearchCV method.

As we can from the matrix below,

	Node 3	Node 6	Node 9	Node 12
Layer 1	0.428	0.436	0.418	0.437
Layer 2	0.551	0.454	0.491	0.434
Layer 3	0.584	0.514	0.496	0.569

The error generally decreases as we add more nodes to the layer. However, it starts to increase again when we add too many nodes to the layer.

Problem 6

The following sample belong to NUC.

```
# Predict
sample = np.array([[0.52,0.47,0.52,0.23,0.55,0.03,0.52,0.39]])
```

```
pred = np.argmax(model.predict(sample))
```

```
category[pred]
```

```
'NUC'
```