Solutions - Tutorial Week 8

1.

A) $TSS = \sum_o \quad \sum_{i=1}^{n} (y_i - \overline{y})^2$

where y_i is the desired output of node *i* and $\overline{y}$ the actual output.
For these 3 instances, The TSS (Total Sum Squared Error) = 2.044931736

B) $MSE = \frac{TSS}{n}$,

Where the TSS = 2.044931736 and n = 3 then,
the MSE (Mean Squared Error) = 0.681643912

C) Classification Error is the number of incorrectly classified instances divided by the total number of instances.
The classification error 2/3 = 66%

D) Example 1 is setosa correctly classified as setosa.
Example 2 is versicolor incorrectly classified as setosa.
Example 3 is versicolour incorrectly classified as virginica.

The confusion matrix

```
 a b c    <-- classified as
 1 0 0 | a = Iris-setosa
 1 0 1 | b = Iris-versicolor
 0 0 0 | c = Iris-virginica
```

2.

<u>Using the 402040 strategy</u>

[0 – 0.4] = 0
(0.4 – 0.6) = unknown
[0.6 – 1] = 1

#1.1
0.6 <= 0.90041 <= 1.0 thus treated as 1
0 <= 0.27903 <=0.4 thus treated as 0
0 <= 0.00001 <= 0.4 thus treated as 0
Actual output = 1 0 0
Desired output = 1 0 0
Correct Classification

#2.1

0 <= 0.22013 <=0.4 thus treated as 0

0 <= 0.28581 <=0.4 thus treated as 0

0 <= 0.00019 <=0.4 thus treated as 0

Actual output = 0 0 0

Desired output = 0 1 0

Incorrect Classification

#3.1

0 <= 0.02983 <= 0.4 thus treated as 0

0 <= 0.29019 <= 0.4 thus treated as 0

0 <= 0.00105 <= 0.4 thus treated as 0

Actual output = 0 0 0

Desired output = 0 1 0

Incorrect Classification

<u>Using the 206020 strategy</u>

[0 – 0.2] = 0

(0.2 – 0.8) = unknown

[0.8 – 1] =1

#1.1

0.8 <= 0.90041 <= 1.0 thus treated as 1

0.2 < 0.27903 < 0.8 thus treated as unknown

0 <= 0.00001 <= 0.2 thus treated as 0

Actual output = 1 unknown 0

Desired output = 1 0 0

Report unknown

#2.1

0.2 < 0.22013 < 0.8 thus treated as unknown

0.2 < 0.28581 < 0.8 thus treated as unknown

0 <= 0.00019 <= 0.2 thus treated as 0

Actual output = unknown unknown 0

Desired output = 0 1 0

Report unknown

#3.1

0 <= 0.02983 <= 0.2 thus treated as 0

0.2 < 0.29019 < 0.8 thus treated as unknown

0 <= 0.00105 <= 0.2 thus treated as 0

Actual output = 0 unknown 0

Desired output = 0 1 0

Report Unknown

Using the 500050 strategy

[0 – 0.5] = 0

(0.5 – 1] = 1

#1.1

0.5 < 0.90041<=1.0 thus treated as 1

0 <= 0.27903 <= 0.5 thus treated as 0

0 <= 0.00001 <= 0.5 thus treated as 0

Actual output = 1 0 0

Desired output = 1 0 0

Correct Classification

#2.1

0 <= 0. 22013 <= 0.5 thus treated as 0

0 <= 0. 28581 <= 0.5 thus treated as 0

0 <= 0. 00019 <= 0.5 thus treated as 0

Actual output = 0 0 0

Desired output = 1 0 0

Incorrect classification

#3.1

0 <= 0.02983 <=0.5 thus treated as 0

0 <= 0.29019 <= 0.5 thus treated as 0

0 <= 0.00105 <=0.5 thus treated as 0

Actual output = 0 0 0

Desired output = 0 1 0

Incorrect Classification

Using the Winner Takes All (WTA) strategy

Highest node = 1

The rest will be 0

#1.1

0.90041 is the highest thus treated as 1

Actual output = 1 0 0

Desired output = 1 0 0

Correct Classification

#2.1

0. 28581 is the highest thus treated as 1

Actual output = 0 1 0

Desired output = 1 0 0

Incorrect Classification

#3.1
0.02983 is the highest thus treated as 1
Actual output = 1 0 0
Desired output = 0 1 0
Incorrect Classification

3. It's application dependent. If it's an application where the machine will make the easy decisions, but the difficult ones are to be referred to a person, then 402040. If the task is to compare the accuracy of classifiers then, we don't want any 'Unknown' so 500050. Or WTA.

4. a) For any nominal attribute, use a 1-out-of-m coding. For example, the attribute cap-shape has six possible values so 6 input nodes are needed. The coding will be
b → 1 0 0 0 0 0
c → 0 1 0 0 0 0
f → 0 0 1 0 0 0
k → 0 0 0 1 0 0
s → 0 0 0 0 1 0
x → 0 0 0 0 0 1

b) 6 + 4 + 10 + 2 + 9 + 4 = 35 nodes

c) 1 for 'e' and 0 for 'p' (or the other way around)

d) '1 0' for 'e' and '0 1' for 'p' (or the other way around)

e) Using 1 node: If applying 402040, then [0,0.4) as 'e', (0.4,0.6) as unknown, and [0.6, 1] as 'p'. If applying 500050, then [0,0.5] as 'e', [0.5,1] as 'p'.

Using 2 nodes: Find the largest output first, If applying 402040, then if largestOutput>=0.6, treat this one as 1 and all others 0; if largestOutput<0.4 then treat all outputs as 0; otherwise no decision. Note though, only the first condition will potentially lead to correct classification, the latter two will both be "unknown", as valid output patterns are "0 1" and "1 0".

f) There is no way to accurately predict the best number of hidden units without training several networks and looking at their error rates. After sufficient training, if both your training error and test error are too high then you may need to increase the number of hidden nodes, because the network seems to be underfitting the data due to insufficient complexity caused by insufficient number of hidden nodes. On the other hand, if your train error is low but overfitting is severe, then you might need to reduce the number of hidden nodes.

Some references offer "rules of thumb" for getting started: "A rule of thumb is for the size of this layer to be somewhere between the input layer size and the output layer size". Detailed discussion on this problem, as well as referenced papers, can be found on the following page, http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-10.html#b