OS 4118 **Example of Neural Network Computations** Spring 2019

Suppose we are building a neural network to predict the Species column from the other four measurements in the famous iris data (called, as you remember, iris in R). We will use a fully connected network with one hidden layer having of two nodes, like this:

1

1

Setosa

Sepal.Length

Hidden 1

Versicolor

Sepal.Width

Hidden 2

Petal.Length

Virginica

Petal.Width

Let us label the input nodes IB (“input bias node”), SL, SW, PL, PW, the hidden nodes HB (“hidden bias node”), H1, H2, and the output nodes SE, VE, and VI. The entire model requires 19 weights to be specified, and each weight can be identified by the two nodes it joins. So, for example, SL -> H1 is the weight from the Sepal.Length node to the Hidden 1 node.

Now suppose the weights are the ones given in the table. (Normally we would scale our inputs, and then the weights in the network would usually be closer to zero.)

At the hidden nodes, we compute the linear combination – the weighted sum – of the inputs with the weights – call it *z* – and then use the **sigmoid** activation function

*f*(*z*) = 1/(1 + exp (–*z*)).

|  |  |  |  |
| --- | --- | --- | --- |
| Weight Name | Weight Value | Weight Name | Weight Value |
| IB -> H1 | 19.8 | HB -> SE | 148.8 |
| IB -> H2 | –215.7 | HB -> VE | –27.9 |
| SL -> H1 | –10.4 | HB -> VI | –120.6 |
| SL -> H2 | –93.0 | H1 -> SE | –500.8 |
| SW -> H1 | –37.5 | H1 -> VE | 259.0 |
| SW -> H2 | –91.8 | H1 -> VI | 241.7 |
| PL -> H1 | 83.5 | H2 -> SE | –177.1 |
| PL -> H2 | 166.3 | H2 -> VE | –145.2 |
| PW -> H1 | –30.5 | H2 -> VI | –31.4 |
| PW -> H2 | 145.4 |  |  |

At the output nodes, we compute the linear combination of the hidden node values with the weights, but instead of using the activation function, we convert the three values into probabilities with **softmax**. That is, we convert the (SE, VE, VI) values by

PSE = exp (SE) / (exp (SE) + exp (VE) + exp (VI)

PVE = exp (VE) / (exp (SE) + exp (VE) + exp (VI)

PVI = exp (VI) / (exp (SE) + exp (VE) + exp (VI).

Question 1: What are the three predicted probabilities the network makes for row 84 of the data set?

Question 2: How should the model classify iris #84?

Question 3: In order to improve the network we measure how it’s performing compared to the “true” values in the training set. For a numeric output we might use the usual sum of squares, . For the categorical case, as here, a common choice is the multinomial deviance, just as we use in logistic regression. In logistic regression (remember?) the loss for one observation is –2 (*y* log () + (1 – *y*) log (1 – ) ), where *y* is 0 or 1 and 0 log 0 ≡ 0. We could also have written that as –2 , where *j* indexes the two classes. Here, when the true class of the observation is 1, we have *y*1 = 1 and *y*2 = 0, while when the true class is 2, we have *y*1 = 0 and *y*2 = 1. This extends instantly to the multi-class case. In our case, with three classes, the loss is

For a multi-class response, the loss is –2 .

What is the loss associated with our predictions for observation 84 from question 1?