Data set description:

The data set I used is Wine Quality data set. The data set is downloaded from UCI Machine Learning Repository. There is a total of 1599 instances in this dataset.

The data set contains 12 attributes for each wine.

Input variables

- fixed acidity
- volatile acidity
- citric acid
- residual sugar
- chlorides
- free sulfur dioxide
- total sulfur dioxide
- density
- pH
- sulphates
- alcohol

Output variable

- quality (score between 0 and 10)

Problem description:

The task is to predict if the quality of the wine is greater than 5.

Inputs:

I took the first 11 columns in the data set which is the 11 attributes of the wine as the inputs. The input data are stored in a two dimensional area, and they are stored as double.

Outputs:

I took the last column which is the quality of the wine as the output. The output data are stored in a one dimensional area, and they are stored as integer. I modified the original output by changing the output to 0 if the score is less or equal to 5, and changing the output to 1 if the score is greater than 5.

3 deep learning architectures and Performance: (all 3 architectures are run at least 10 times, and pick the average for each architecture)

1. [11,15,15,15,2] : The program takes about 25 sec to train the data. And the result is 114/160.
    a. Accuracy: 114/160 = 71.25%
    b. TP: 69
    c. TN: 45
    d. FP: 27
    e. FN: 19

2. [11,30,30,30,2]: The program takes about 20 sec to train the data. And the result is 119/160.
    a. Accuracy: 119/160 = 74.375%
    b. TP: 68
    c. TN: 51
    d. FP: 20
    e. FN: 21
3. [11,30,30,30,30,30,30,2] : The program takes about 30 sec to train the data. And the result is 120/160.
    a. Accuracy: 121/160 = 75.625%
    b. TP: 64
    c. TN: 57
    d. FP: 16
    e. FN: 23

Discussion:

For both architecture 1 and architecture 2, I used 3 hidden layers. However, I increased the number of neurons for each layer. And as stated above, the result for architecture 2 is better than the result for architecture 1.

For both architecture 2 and architecture 3, I used the same number of neurons for each layer. However, I increased the number of layers for architecture 3. As stated above, the result for architecture 3 is slightly better than the result for architecture 2.

So, increasing the number of neurons and number of layers should be able to increase the accuracy.

So, I suggest to increase the number of neurons and number of layers.

Summary:

To sum up, the number of neurons and number of layers can affect the accuracy of deep learning solution. As the numbers increase, the accuracy of the solution also increases.