

Comp 307 Assignment 2 Report

Zhanghao Kong

Part 1.

1. With 200 epochs, accuracy is between 62-64 percents.

```
"C:\Program Files\Java\jdk-11.0.2\bin\java.exe"  
correct: 1000  
error num: 600  
Accuracy of correctness: 62.5%  
  
Process finished with exit code 0
```

2. For training, with huge amounts of data could be easier to achieve better result. Because the data set we used is too small, so the result is not very high.

Part 2.

1. The number of input nodes is 13, which is the total number of independent values in the wine training and testing set. The number of output nodes is the category classification of wine 1 or 2 or 3. Hidden layer is 6. Choosing a high hidden layer is for achieving better accuracy rate and when set it to 6, I achieve the highest result with 98 percent.

	precision	recall	f1-score	support
1	1.00	1.00	1.00	30
2	0.94	0.97	0.96	35
3	0.96	0.92	0.94	24
accuracy			0.97	89
macro avg	0.97	0.96	0.96	89
weighted avg	0.97	0.97	0.97	89

2.

parameters	value
learning rate	0.1
momentum	0.9
hidden layer	6
max iteration	300

Lower learning rate can make it learn more clearly without missing more datas.

Momentun equals 0.9 is default value.

Choosing a high hidden layer is for achieving a better accuracy rate . when i choose hidden layer to 6 and max iteration to 300, achieving best result 97percents.

3.For preventing neural network from overtraining resulting in overfitting, I set the max iteration to 200 and achieving a highest result. Finally i choose 200 as epochs for terminating.

4.

Epoch value					
100	precision		recall	f1-score	support
	1	0.97	0.97	0.97	30
	2	0.92	0.94	0.93	35
	3	0.96	0.92	0.94	24
	accuracy			0.94	89
	macro avg	0.95	0.94	0.94	89
	weighted avg	0.94	0.94	0.94	89
200	precision		recall	f1-score	support
	1	1.00	0.97	0.98	30
	2	0.94	0.97	0.96	35
	3	0.96	0.96	0.96	24
	accuracy			0.97	89
	macro avg	0.97	0.97	0.97	89
	weighted avg	0.97	0.97	0.97	89
300	precision		recall	f1-score	support
	1	0.97	1.00	0.98	30
	2	1.00	0.94	0.97	35
	3	0.96	1.00	0.98	24
	accuracy			0.98	89
	macro avg	0.98	0.98	0.98	89
	weighted avg	0.98	0.98	0.98	89
400	precision		recall	f1-score	support
	1	0.97	1.00	0.98	30
	2	0.97	0.94	0.96	35
	3	0.96	0.96	0.96	24
	accuracy			0.97	89
	macro avg	0.97	0.97	0.97	89
	weighted avg	0.97	0.97	0.97	89

500	precision	recall	f1-score	support	
	1	1.00	0.97	0.98	30
	2	0.92	0.97	0.94	35
	3	0.96	0.92	0.94	24
	accuracy			0.96	89
	macro avg	0.96	0.95	0.95	89
	weighted avg	0.96	0.96	0.96	89
	600	precision	recall	f1-score	support
1		0.97	0.97	0.97	30
2		0.97	0.89	0.93	35
3		0.89	1.00	0.94	24
accuracy				0.94	89
macro avg		0.94	0.95	0.94	89
weighted avg		0.95	0.94	0.94	89
700		precision	recall	f1-score	support
	1	0.97	1.00	0.98	30
	2	0.97	0.94	0.96	35
	3	0.96	0.96	0.96	24
	accuracy			0.97	89
	macro avg	0.97	0.97	0.97	89
	weighted avg	0.97	0.97	0.97	89
	800	precision	recall	f1-score	support
1		0.94	1.00	0.97	30
2		0.97	0.89	0.93	35
3		0.92	0.96	0.94	24
accuracy				0.94	89
macro avg		0.94	0.95	0.94	89
weighted avg		0.95	0.94	0.94	89
900		precision	recall	f1-score	support
	1	0.97	0.97	0.97	30
	2	0.94	0.94	0.94	35
	3	0.96	0.96	0.96	24
	accuracy			0.96	89
	macro avg	0.96	0.96	0.96	89
	weighted avg	0.96	0.96	0.96	89
	1000	precision	recall	f1-score	support
1		0.97	0.97	0.97	30
2		0.92	0.94	0.93	35
3		0.96	0.92	0.94	24
accuracy				0.94	89
macro avg		0.95	0.94	0.94	89
weighted avg		0.94	0.94	0.94	89

Analyse and Conclusion: Compare with these pictures, we can clearly find that when epoch reach 300, the output get highest value 98. This value is very good result because if the result reach 100 percents, it will overfitting, so for avoiding that, 300 is a good value for epoch.

5.K-nearest neighbor classifier is based on the value of K and the classification of data to get the accuracy. For small datasets, the nearest neighbor classifier may be a better choice, but not during large datasets, because it uses a lot of memory. However, the neural network does not need a lot of storage space to predict after training, so if the data set is large, it will be a better choice.

Part 3.

1.A good terminal set for this task is the input value like “x”.

```
new Add(config, CommandGene.DoubleClass),
new Multiply(config, CommandGene.DoubleClass),
new Multiply3(config, CommandGene.DoubleClass),
new Divide(config, CommandGene.DoubleClass),
new Subtract(config, CommandGene.DoubleClass),
new Exp(config, CommandGene.DoubleClass),
new Log(config, CommandGene.DoubleClass),
new Terminal(config, CommandGene.DoubleClass, a_minValue: -5.0, a_maxValue: 5.0, a_wholeNumbers: true) } };
```

2.

My function contains Add,Multiply,Multiply3, Exp,Log, Divide, Subtract, Terminal for basically math functions. I ignore Sine and Cosine cause sometimes it could be 0 then lead errors.

3.We need a have a sum of the absolute value which including the difference between the yield calculated by genotype and the actual yield of each generation, this is important for a good fitness function.

```
double result ;

long longResult = 0;
for (int i = 0; i < storeX.size(); i++) {
    x.set(storeX.get(i));
    double value = (double) arg0.execute_double( i: 0, NO_ARGS);
    longResult += Math.abs(value - storeY.get(i));
}

//root mean squared
double root = Math.sqrt(longResult);
result = longResult;
return result;
```

4.

Maximum initial depth: 6

- Population size: 2000

- Max Tree depth: 12

- Cross over probability: 95%
- Mutation probability: 5%
- Reproduction probability: 5%

Used 2000 maximum evolution of generation to stop my program. Besides, when fitness function return 0, then stop.

5

(1).

```
Best solution fitness: 0.0  
Best solution: log (log (log 1.0))
```

(2).

```
Best solution fitness: 0.0  
Best solution: log (4.0 - X)
```

(3).

```
Best solution fitness: 0.0  
Best solution: Exp((log ((log -1.0) - (Exp(X)))))
```