

(1) Source code is in the folder named src (path: cl1074 -> Q1 -> src).

class name	location	language	description
Main.java	cl1074/Q1/src	javja	To start the program.
Neuron.java	cl1074/Q1/src	java	To generate neuron for network. Every neuron will keep its net and output value.
Backpropagation.java	cl1074/Q1/src	java	To run the main algorithm.

(2) Screen shots

a. Learning rate = 0.5, Target error = 0.1

```
input the learning rate:
0.5
input the target error:
0.1
|
(1) initial weights:
w1:-0.23977664676352162
w2:0.6267086491450733
w3:0.3114587644067748
w4:-0.11261319029834804
w5:0.9922016447054602
w6:0.3205333367581973
=====end of initial weights=====

(2) first-batch error:0.10985231467363132
(3) final weight:
w1:-0.2722183199654997
w2:0.597989317850317
w3:0.3000258193821176
w4:-0.12220902134156857
w5:0.9224603567938995
w6:0.25823064207272056
=====end of final weights=====
(4) final error:0.05829131722436501
(5) total number of batches run through in the training:4
```

b. Learning rate = 1, Target error = 0.1

input the learning rate:

1

input the target error:

0.1

(1) initial weights:

w1:0.8414811426859854

w2:-0.7167330144002562

w3:0.36329868286699396

w4:-0.7808890667768216

w5:0.02657312916076049

w6:0.44215417500792964

=====end of initial weights=====

(2) first-batch error:0.11657683899672161

(3) final weight:

w1:0.7096120570662272

w2:-0.9189719974966616

w3:0.31807134690855593

w4:-1.0627254694463852

w5:0.14398908460457216

w6:0.5613954734514978

=====end of final weights=====

(4) final error:0.09481021537718025

(5) total number of batches run through in the training:118

c. Learning rate = 0.5, Target error = 0.02

input the learning rate:

0.5

input the target error:

0.02

(1) initial weights:

w1:-0.6877218930137787

w2:-0.26579518668284163

w3:0.720017922807545

w4:-0.7274422060572774

w5:0.7341655050058866

w6:0.32826283160954506

=====end of initial weights=====

(2) first-batch error:0.15032790641786692

(3) final weight:

w1:-446.4531046029754

w2:-440.50048105165223

w3:7.468281215048335

w4:-9.843609098244206

w5:-1.9462600131233547

w6:1.4695387189965927

=====end of final weights=====

(4) final error:0.017510737281637483

(5) total number of batches run through in the training:185334

- d. Learning rate = 1, Target error = 0.02

```
input the learning rate:
1
input the target error:
0.02
|
(1) initial weights:
w1:0.8587108391723226
w2:-0.6821686709767776
w3:0.04427585065574946
w4:-0.7745520692226662
w5:0.7743092340417312
w6:-0.6207935648040506
=====end of initial weights=====

(2) first-batch error:0.12300061324857242
(3) final weight:
w1:2.6841455484691137
w2:-5.258943402493675
w3:-25.32014769175034
w4:-24.827114742178303
w5:0.2905228452593708
w6:-3.17907779021097
=====end of final weights=====
(4) final error:0.018218318519376744
(5) total number of batches run through in the training:4855
```

- (3) Try a few different learning rates and include the best one I can find.

In this part, I fixed target error to 0.1 and try learning rate from 0.1, 0.2, ... to 1.

For each learning rate, I run 10 times of the program and take the average of the batches run to find the best one.

Learning rate	Numbers of run for 10 times trial	Average error
0.1	6, 80, 15, 4, 1544, 2390, 795, 13, 1, 107	495.5
0.2	411, 6, 12, 5, 1101, 10, 9, 512, 48, 7	212.1
0.3	114, 1, 414, 11, 114, 12, 3, 31, 383, 12	109.5
0.4	276, 5, 98, 2, 20, 4, 35, 13, 1, 39	49.3
0.5	36, 6, 6, 14, 7, 6, 9, 7, 229, 237	55.7
0.6	6, 13, 78, 49, 1, 22, 134, 27, 35, 57	42.2
0.7	3, 29, 15, 1, 36, 5, 26, 142, 248, 3	50.8
0.8	36, 98, 8, 2, 62, 44, 35, 35, 20	34.0
0.9	12, 27, 84, 10, 14, 15, 61, 116, 8, 47	39.4
1.0	15, 5, 12, 3, 17, 124, 37, 5, 2, 2	22.2

Based on the table above, I find that when target error = 0.1, learning rate = 1 is the best one.

