

How to use the demo program.

1) Working environment

It is checked only with Windows10 and Eclipse Java EE IDE for Web Developers. Version: Neon.3 Release (4.6.3). No other environment was checked to work.

2) Preparation

Unzip src.zip to a directory.

The src directory should be extracted where our java code existed.

Make a working directory for the java code (let's say it 'workspace')

Launch the Eclipse for Java with selecting the 'workspace' as a workspace.

File -> Import -> general -> import project from folders or archives

Select the src directory, then next

The src directory should be imported to the project explorer

Now, you are ready to run

3) The programs

There should be 6 classes, those are MLPLMT21.java, MLPLMT21App.java, MLPLMT20.java, MLPLMT20App.java, MLPLMT0.java, and MLPLMT0App.java. These programs are a part of our Stacked Denoised Auto-encoder and an example of complete implementation of our new algorithms. The TL2 in our manuscript correspond to MLPLMT20java, MLPLMT20App.java, TL1 correspond to MLPLMT21.java, MLPLMT21App.java, and SDA or FNN correspond to MLPLMT0.java, and MLPLMT0App.java.

4) How to run

Open MLPLMT20App.java, MLPLMT21App.java or MLPLMT0App.java, and run it as a java application.

5) How to see the results

You see the results in the console window in the Eclipse. The last five sentences are results. For example,

```
-----  
00: 64 out of 80.0 ratio: 0.8  
01: 60 out of 80.0 ratio: 0.75  
10: 60 out of 80.0 ratio: 0.75  
11: 64 out of 80.0 ratio: 0.8  
correct all: 60 out of 80.0 ratio: 0.75  
-----
```

Four XOR tests are run simultaneously, then you will get four results every run. The demo run 20 runs. So, you will get 80 results for each input 00, 01, 10, and 11. The demo program counts how many time you get the expected results 00 ->0, 01 -> 1, 10 -> 1 and 11 -> 0. These are the first four sentences. For example, in 64 times out of 80 trials, your get the correct answer for 00-input, and so on. The last sentence show that how many time you get exact answer for all four inputs in a single run. For example, you get 60 times correct answers for 00, 01, 10 and 11 inputs over 80 trials. We should say the results may not be exact same as in the number of manuscript. It is because our results are the average number of many runs, but you see only one trial.

#### 6) Results you see

For TL2: MLPLMT20App.java

00: 76 out of 80.0 ratio: 0.95  
01: 80 out of 80.0 ratio: 1.0  
10: 76 out of 80.0 ratio: 0.95  
11: 76 out of 80.0 ratio: 0.95  
correct all: 76 out of 80.0 ratio: 0.95

For TL1: MLPLMT21App.java,

00: 64 out of 80.0 ratio: 0.8  
01: 60 out of 80.0 ratio: 0.75  
10: 60 out of 80.0 ratio: 0.75  
11: 64 out of 80.0 ratio: 0.8  
correct all: 60 out of 80.0 ratio: 0.75

For FNN(SDA): MLPLMT0App.java

00: 56 out of 80.0 ratio: 0.7  
01: 55 out of 80.0 ratio: 0.6875  
10: 53 out of 80.0 ratio: 0.6625  
11: 51 out of 80.0 ratio: 0.6375  
correct all: 49 out of 80.0 ratio: 0.6125

I hope you enjoy the difference.