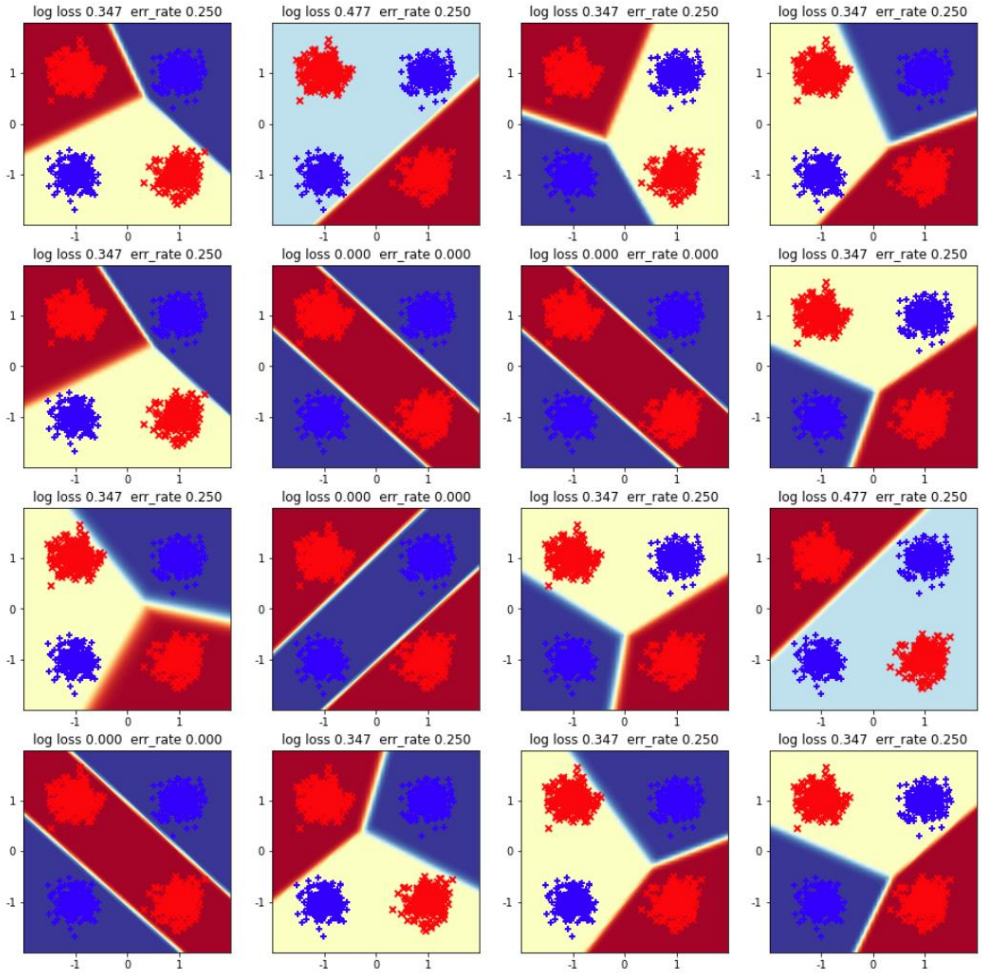
Question 1

```
finished LBFGS run  0/16 after    0.1 sec |  29 iters | converged     | loss 0.347
finished LBFGS run  1/16 after    0.1 sec |  29 iters | converged     | loss 0.477
finished LBFGS run  2/16 after    0.1 sec |  21 iters | converged     | loss 0.347
finished LBFGS run  3/16 after    0.1 sec |  35 iters | converged     | loss 0.347
finished LBFGS run  4/16 after    0.1 sec |  29 iters | converged     | loss 0.347
finished LBFGS run  5/16 after    0.1 sec |  29 iters | converged     | loss 0.000
finished LBFGS run  6/16 after    0.2 sec |  23 iters | converged     | loss 0.000
finished LBFGS run  7/16 after    0.2 sec |  37 iters | converged     | loss 0.347
finished LBFGS run  8/16 after    0.2 sec |  15 iters | converged     | loss 0.347
finished LBFGS run  9/16 after    0.2 sec |  26 iters | converged     | loss 0.000
finished LBFGS run 10/16 after    0.2 sec |  36 iters | converged     | loss 0.347
finished LBFGS run 11/16 after    0.3 sec |  28 iters | converged     | loss 0.477
finished LBFGS run 12/16 after    0.3 sec |  39 iters | converged     | loss 0.000
finished LBFGS run 13/16 after    0.3 sec |  29 iters | converged     | loss 0.347
finished LBFGS run 14/16 after    0.4 sec |  25 iters | converged     | loss 0.347
finished LBFGS run 15/16 after    0.4 sec |  30 iters | converged     | loss 0.347
```



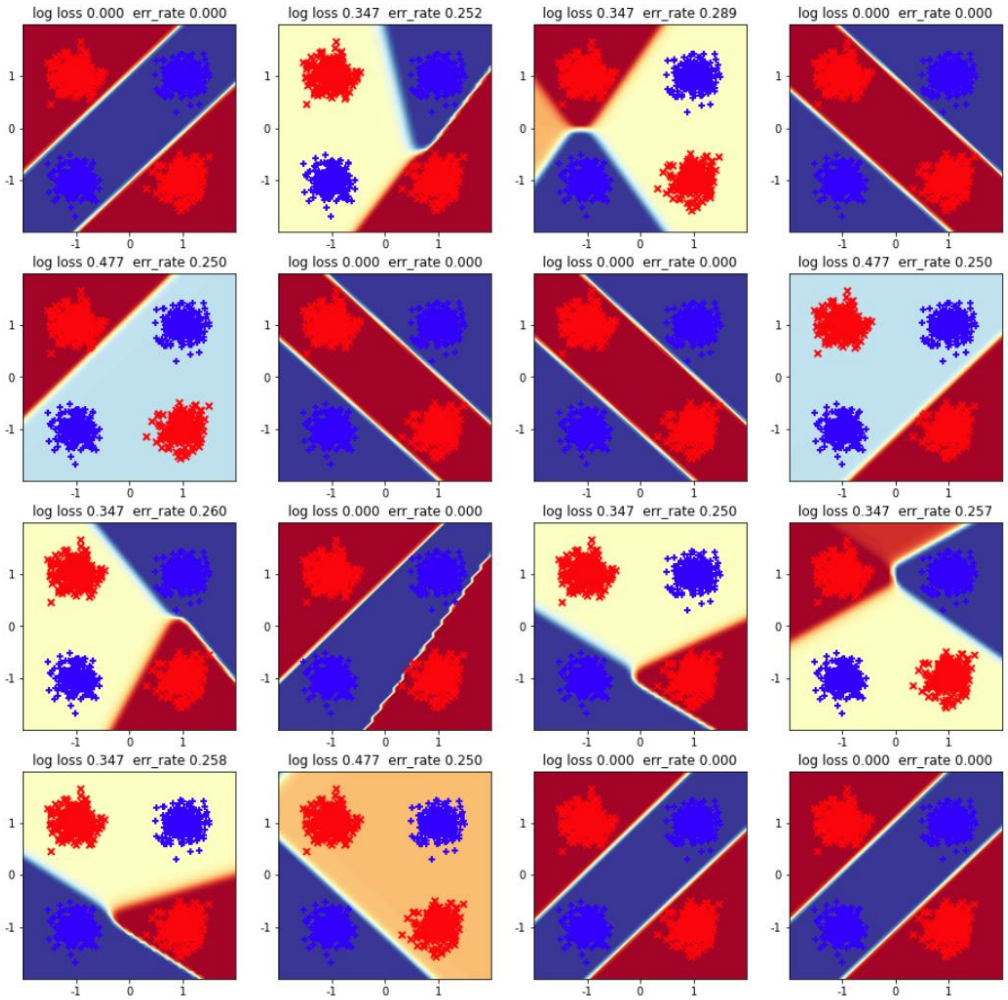
B

Answer: There is 1/4 reach the 0 trainning error. The remainings' error rate is all 0.25. All of them in the 16 iterations are converged and they are averaged converged in 0.025 seconds. Overall, it converged really fast.

I think the reason may be for the gradient, there are some local minimum and absolute minimum. Therefore, the error rate may vary, and when the gradient reaches the 0, this is the absolute minimum.

Question 2

```
finished LBFGS run  0/16 after    0.0 sec |  58 iters | converged     | loss 0.000
finished LBFGS run  1/16 after    0.1 sec | 126 iters | converged     | loss 0.347
finished LBFGS run  2/16 after    0.1 sec |  45 iters | converged     | loss 0.347
finished LBFGS run  3/16 after    0.2 sec |  80 iters | converged     | loss 0.000
finished LBFGS run  4/16 after    0.2 sec |  40 iters | converged     | loss 0.477
finished LBFGS run  5/16 after    0.2 sec |  42 iters | converged     | loss 0.000
finished LBFGS run  6/16 after    0.3 sec |  50 iters | converged     | loss 0.000
finished LBFGS run  7/16 after    0.5 sec |  42 iters | converged     | loss 0.477
finished LBFGS run  8/16 after    0.5 sec |  68 iters | converged     | loss 0.347
finished LBFGS run  9/16 after    0.6 sec | 114 iters | NOT converged | loss 0.001
finished LBFGS run 10/16 after    0.6 sec |  68 iters | converged     | loss 0.347
finished LBFGS run 11/16 after    0.7 sec | 109 iters | converged     | loss 0.347
finished LBFGS run 12/16 after    0.7 sec |  65 iters | converged     | loss 0.347
finished LBFGS run 13/16 after    0.8 sec |  33 iters | converged     | loss 0.478
finished LBFGS run 14/16 after    0.8 sec |  53 iters | converged     | loss 0.000
finished LBFGS run 15/16 after    0.8 sec |  61 iters | converged     | loss 0.000
```



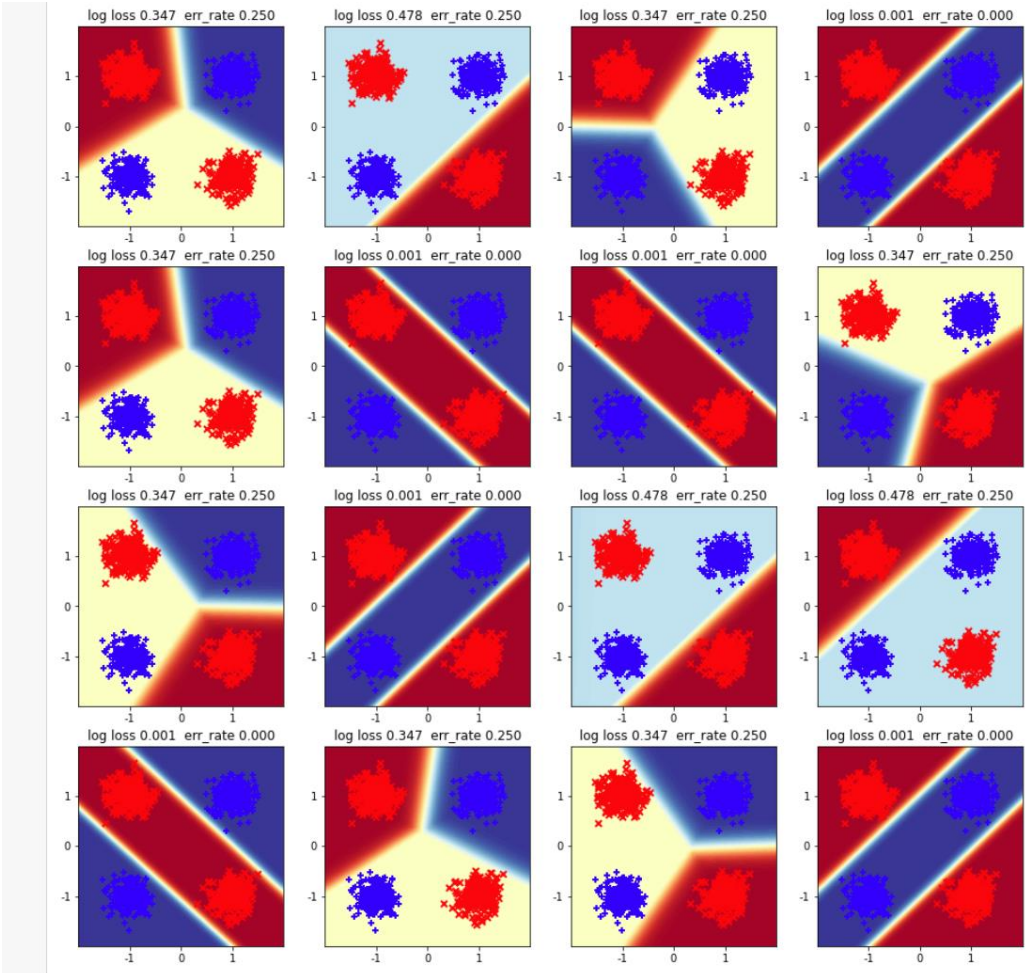
B

Answer: 7/16's runs reach 0 training error. Others are around 0.25 to 0.361. All of them in the 16 iterations are converged and they are averaged converged in 0.05 seconds. Overall, it converged really fast.

I think the reason may be for the gradient, there are some local minimum and absolute minimum. Therefore, the error rate may vary, and when the gradient reaches the 0, this is the absolute minimum.

Question 3

```
finished SGD run  0/16 after    5.5 sec | 267 epochs | converged     | loss 0.347
finished SGD run  1/16 after   11.8 sec | 307 epochs | converged     | loss 0.478
finished SGD run  2/16 after   16.8 sec | 239 epochs | converged     | loss 0.347
finished SGD run  3/16 after   25.0 sec | 400 epochs | NOT converged | loss 0.001
finished SGD run  4/16 after   30.7 sec | 275 epochs | converged     | loss 0.347
finished SGD run  5/16 after   38.2 sec | 400 epochs | NOT converged | loss 0.001
finished SGD run  6/16 after   45.6 sec | 400 epochs | NOT converged | loss 0.001
finished SGD run  7/16 after   50.5 sec | 273 epochs | converged     | loss 0.347
finished SGD run  8/16 after   54.4 sec | 219 epochs | converged     | loss 0.347
finished SGD run  9/16 after   61.6 sec | 400 epochs | NOT converged | loss 0.001
finished SGD run 10/16 after   68.6 sec | 394 epochs | converged     | loss 0.478
finished SGD run 11/16 after   76.5 sec | 400 epochs | NOT converged | loss 0.478
finished SGD run 12/16 after   84.3 sec | 400 epochs | NOT converged | loss 0.002
finished SGD run 13/16 after   89.7 sec | 304 epochs | converged     | loss 0.347
finished SGD run 14/16 after   96.1 sec | 331 epochs | converged     | loss 0.347
finished SGD run 15/16 after  103.6 sec | 400 epochs | NOT converged | loss 0.001
```



B

Answer: 3/8 of runs reach 0 training error, and others all have the same error rate, which is 0.25. In this time, only 9/16 of runs finally converged. For the converged one, the relatively fast converge is 4 seconds while the slow converge can be nearly 9 seconds. On the other hand, for the non-converged one, the speed is simliar.'

C

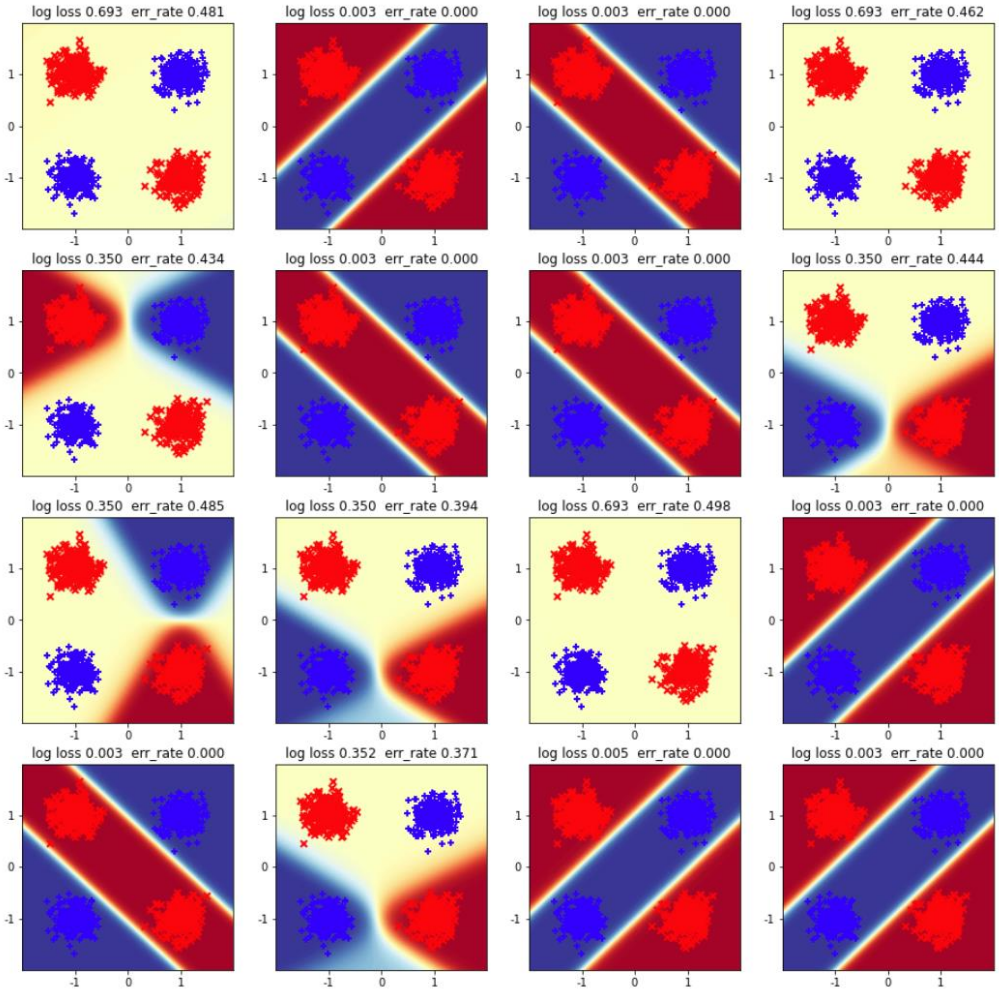
Answer: The SGD solver takes much longer time to calculate, and just about

half of the runs successfully converge. I think this may because the SGD does not do the regularization to its input, which may cause it to be slow.

In addition, there is some local minimum, which makes the SGD think that it is already get the best solution. Therefore, the SGD cannot learn well.

Question 4

```
finished SGD run  0/16 after    3.2 sec | 161 epochs | converged     | loss 0.693
finished SGD run  1/16 after   11.1 sec | 400 epochs | NOT converged | loss 0.005
finished SGD run  2/16 after   18.1 sec | 400 epochs | NOT converged | loss 0.005
finished SGD run  3/16 after   22.1 sec | 215 epochs | converged     | loss 0.693
finished SGD run  4/16 after   29.9 sec | 400 epochs | NOT converged | loss 0.351
finished SGD run  5/16 after   37.9 sec | 400 epochs | NOT converged | loss 0.005
finished SGD run  6/16 after   45.0 sec | 400 epochs | NOT converged | loss 0.005
finished SGD run  7/16 after   52.6 sec | 400 epochs | NOT converged | loss 0.351
finished SGD run  8/16 after   60.2 sec | 400 epochs | NOT converged | loss 0.351
finished SGD run  9/16 after   67.8 sec | 400 epochs | NOT converged | loss 0.351
finished SGD run 10/16 after   70.2 sec | 124 epochs | converged     | loss 0.693
finished SGD run 11/16 after   77.8 sec | 400 epochs | NOT converged | loss 0.005
finished SGD run 12/16 after   85.4 sec | 400 epochs | NOT converged | loss 0.005
finished SGD run 13/16 after   92.8 sec | 400 epochs | NOT converged | loss 0.353
finished SGD run 14/16 after  100.6 sec | 400 epochs | NOT converged | loss 0.007
finished SGD run 15/16 after  108.8 sec | 400 epochs | NOT converged | loss 0.005
```



B

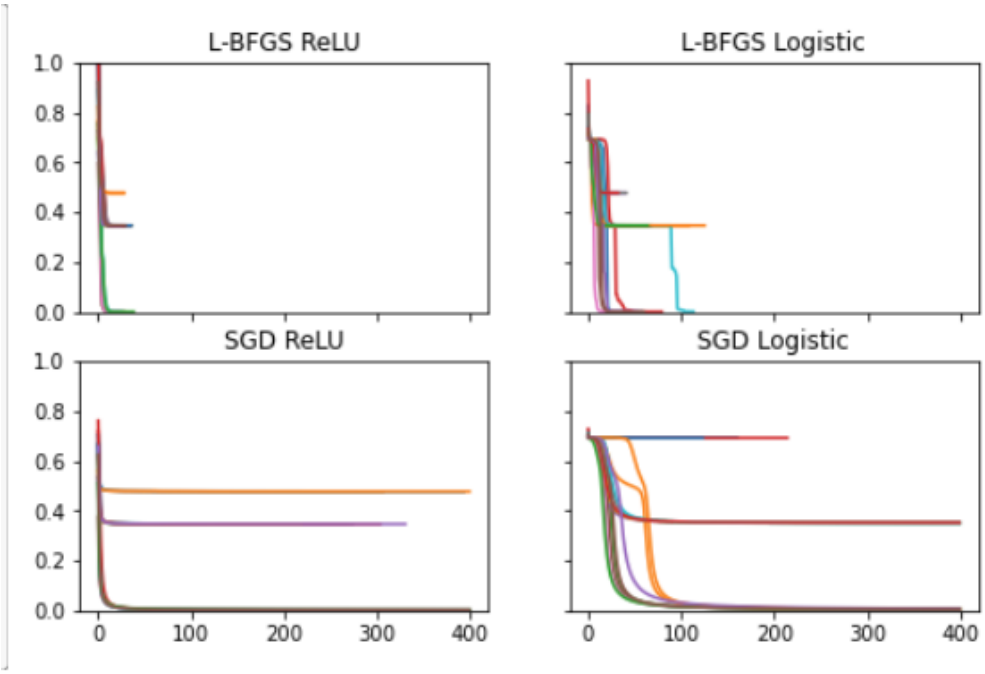
Answer: 1/2 of runs reach 0 training error, and the others are about 0.371 to 0.498. In this time, only 3/16 of runs finally converged. For the converged one, the relatively fast converge is 3 seconds while the slow converge can be nearly 6 seconds. On the other hand, for the non-converged one, the speed is simliar.

C

Answer: The SGD solver takes much longer time to calculate, and just little runs successfully converge. I think this may because the SGD does not do the regularization to its input, which may cause it to be slow.

In addition, there is some local minimum, which makes the SGD think that it is already get the best solution. Therefore, the SGD cannot learn well.

Question 5



B

Answer All ReLU seems easier to optimize, and all Logistic Sigmoid requires more iterations.

C

Answer: 1. compare two functions with other different classfications

2. compare two functions with other different solvers

3.compare two functions with other different hypeparameters

(All do the compare experiment for the same data)