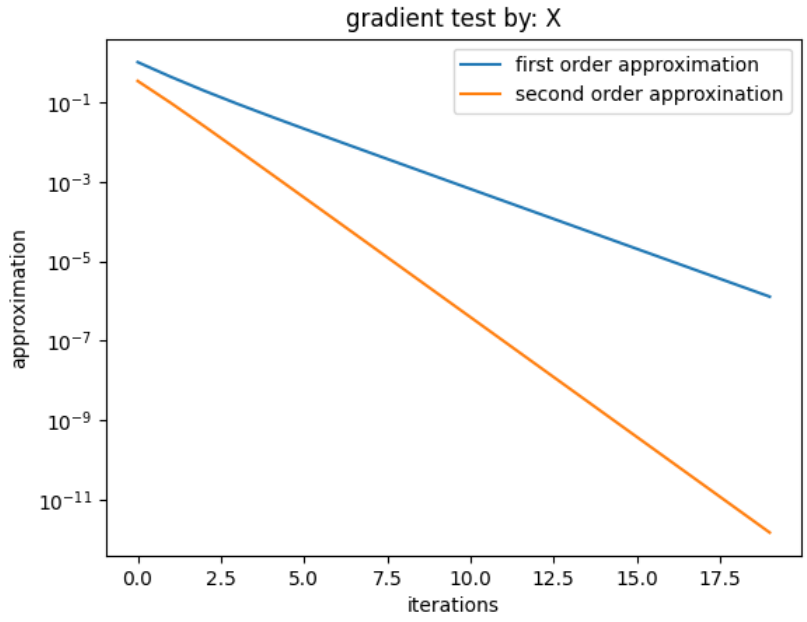
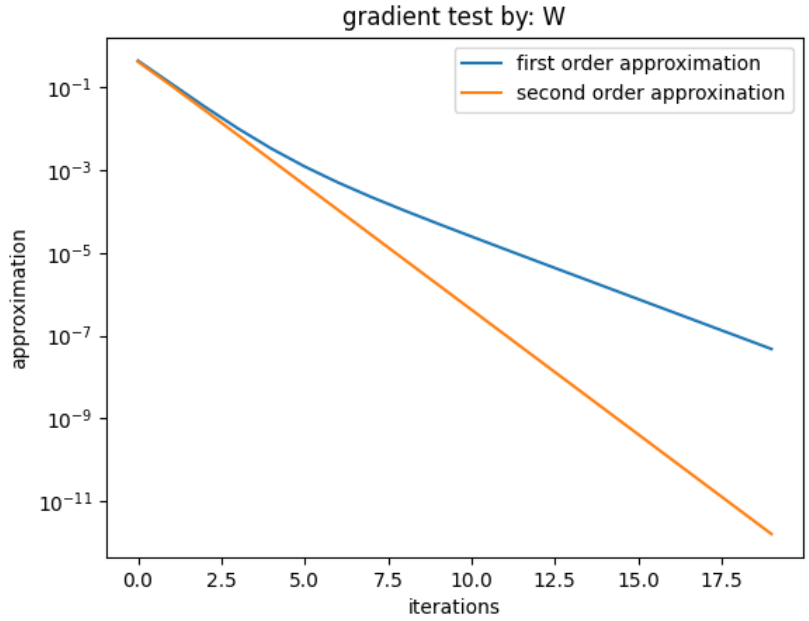
Assignment 1 – Report

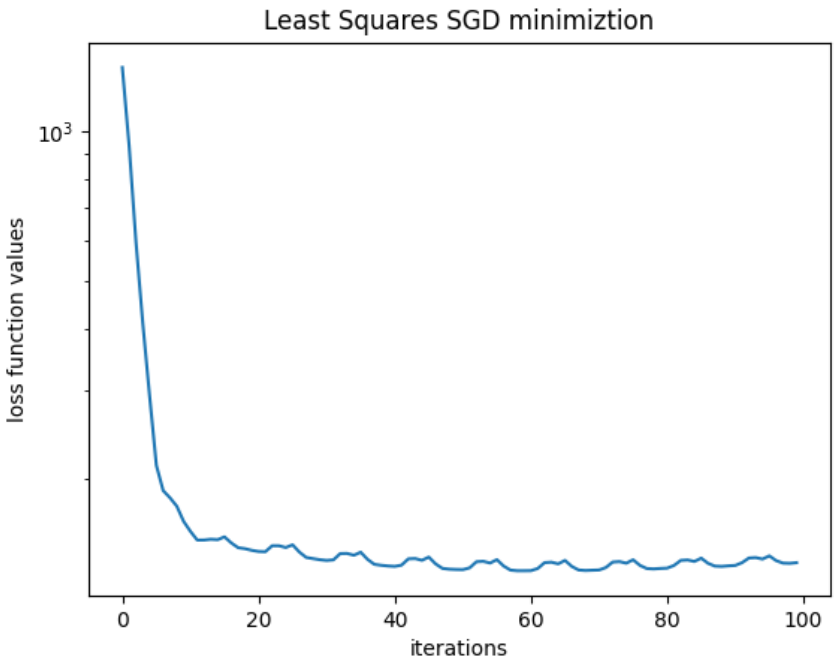
Part 1

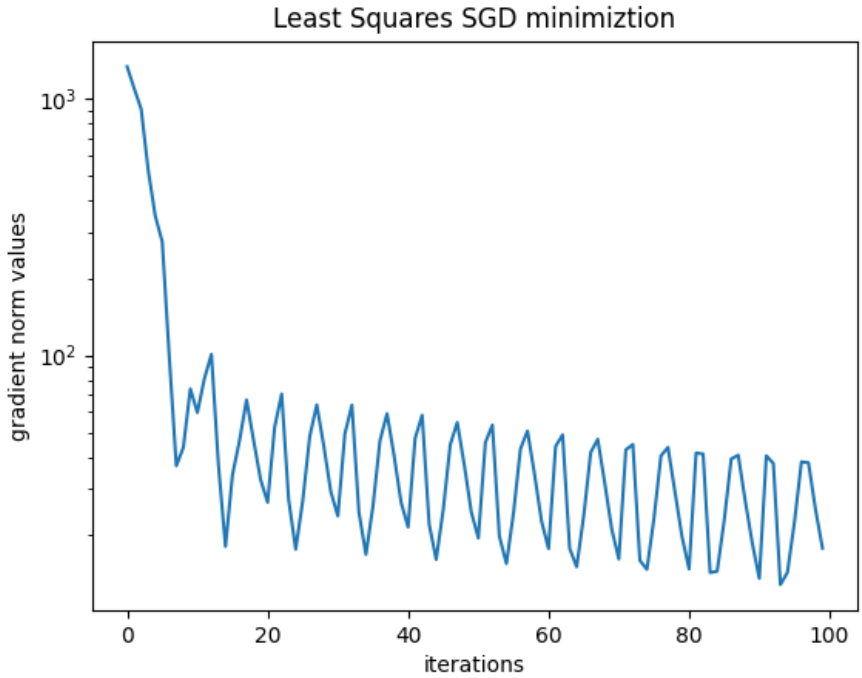
Q2.1.1- The following plot shows the result of gradient test on the derivatives of soft-max regression loss function, w.r.t x and W (bias is included in W):



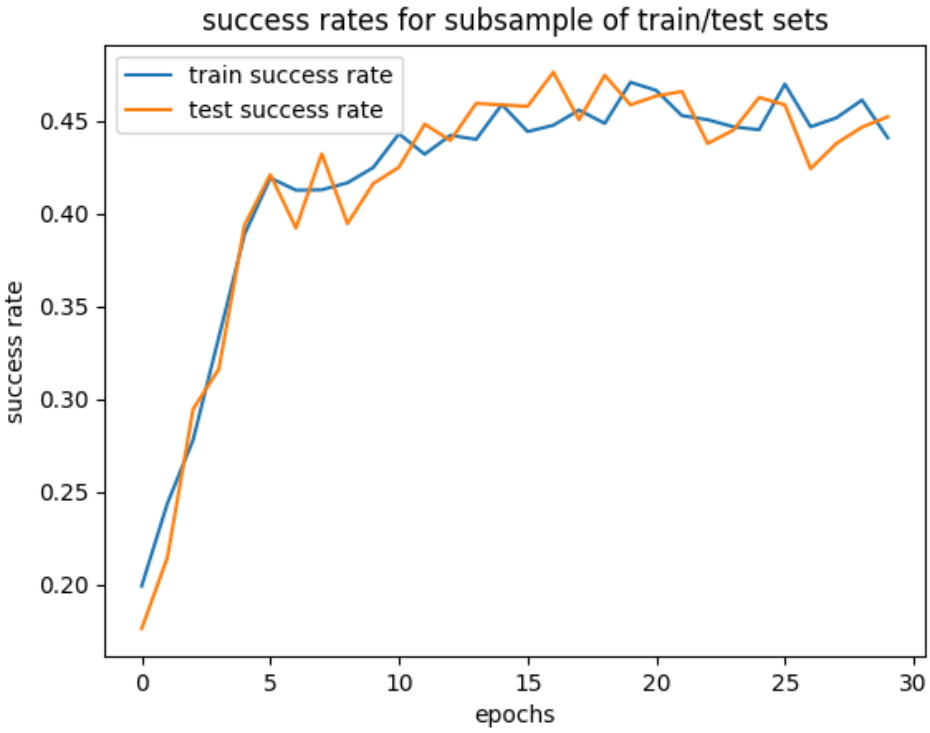


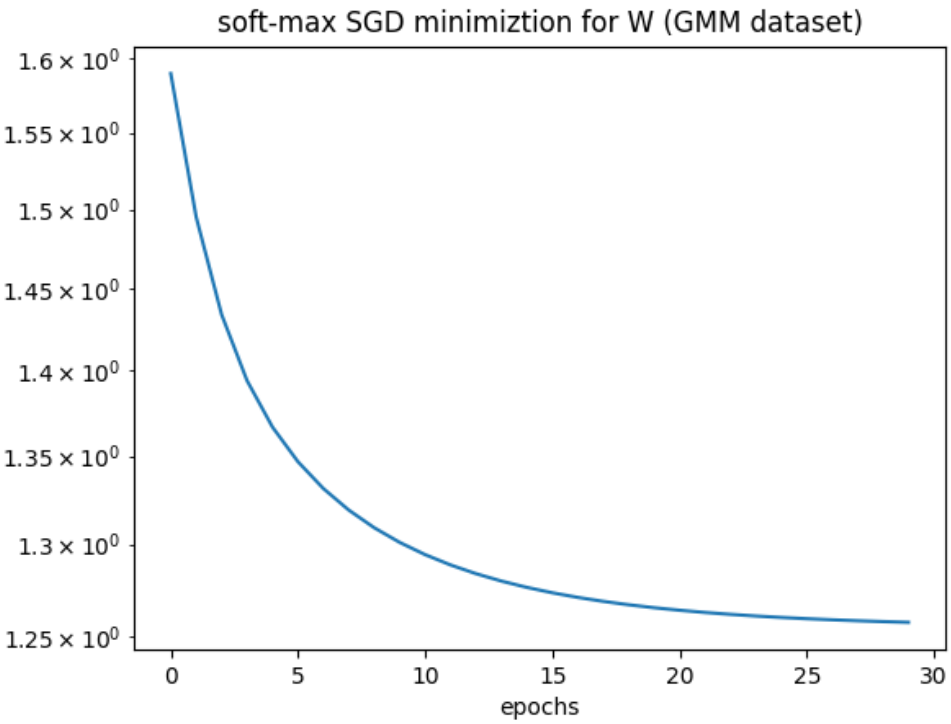
Q2.1.2 – The following plots show the minimization of a LS objective function using our implementation of SGD. With regards to a dataset we got from the internet:

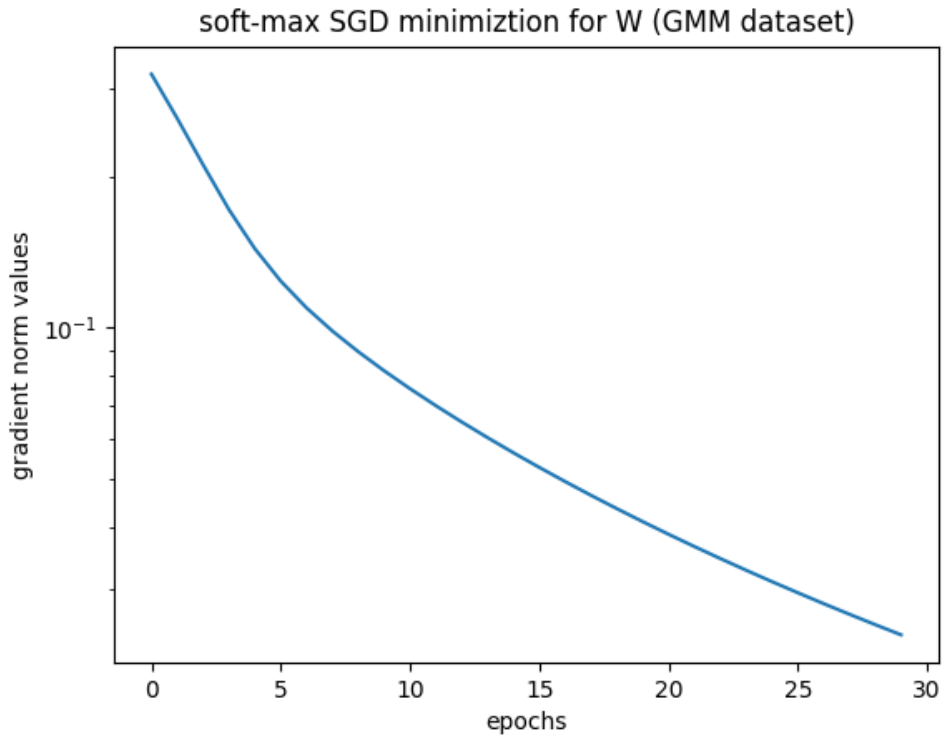




Q2.1.3 – The following plots show the success percentages of the data classification after each epoch for the test data during the minimization of soft-max regression objective function using our implementation of SGD:





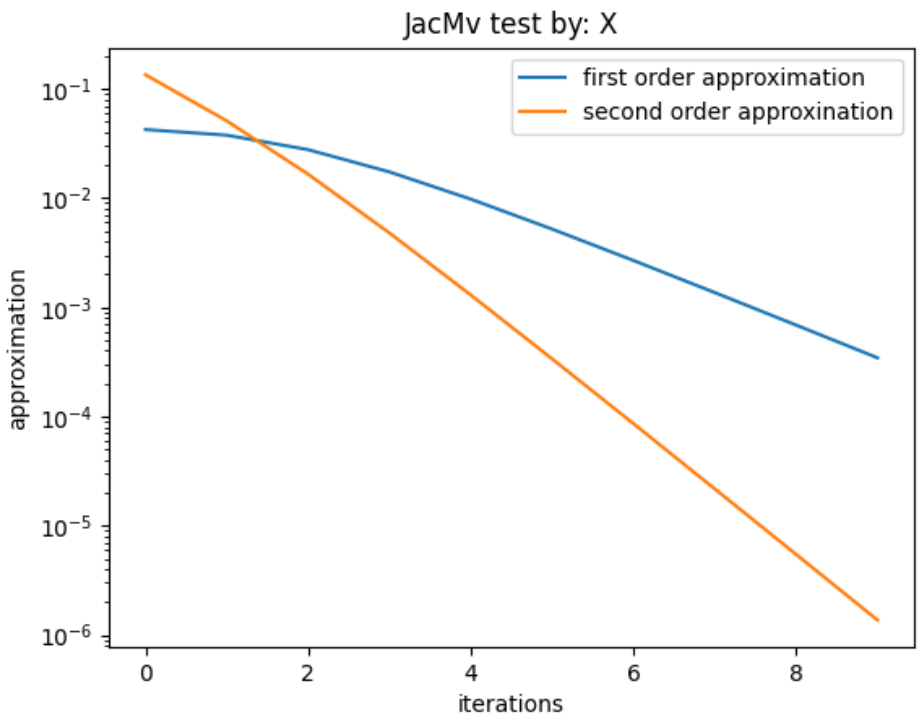


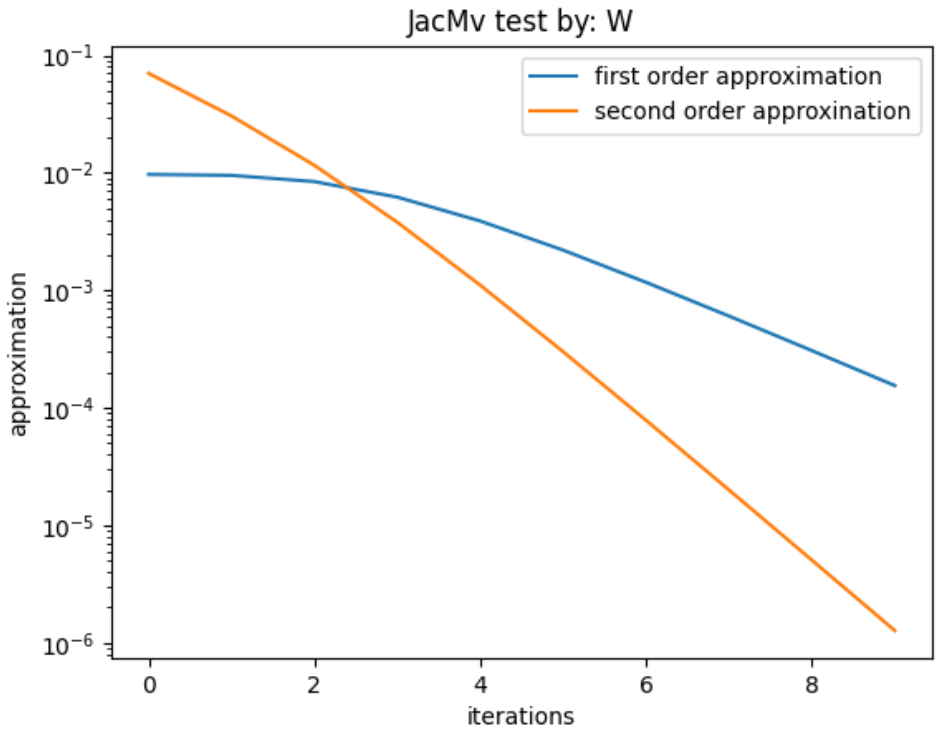
For this experiment we used a learning rate of 0.3 and mini-batch size of 200.

Other learning rates and mini-batch sizes we used during our tries were: (0.01, 200), (0.5, 200), (0.1, 200), (0.3,50), (0.3,100)

Part 2

Q2.2.1 – The following plots show the result of Jacobian tests w.r.t X and W for the standard neural network using the tanh activation function:

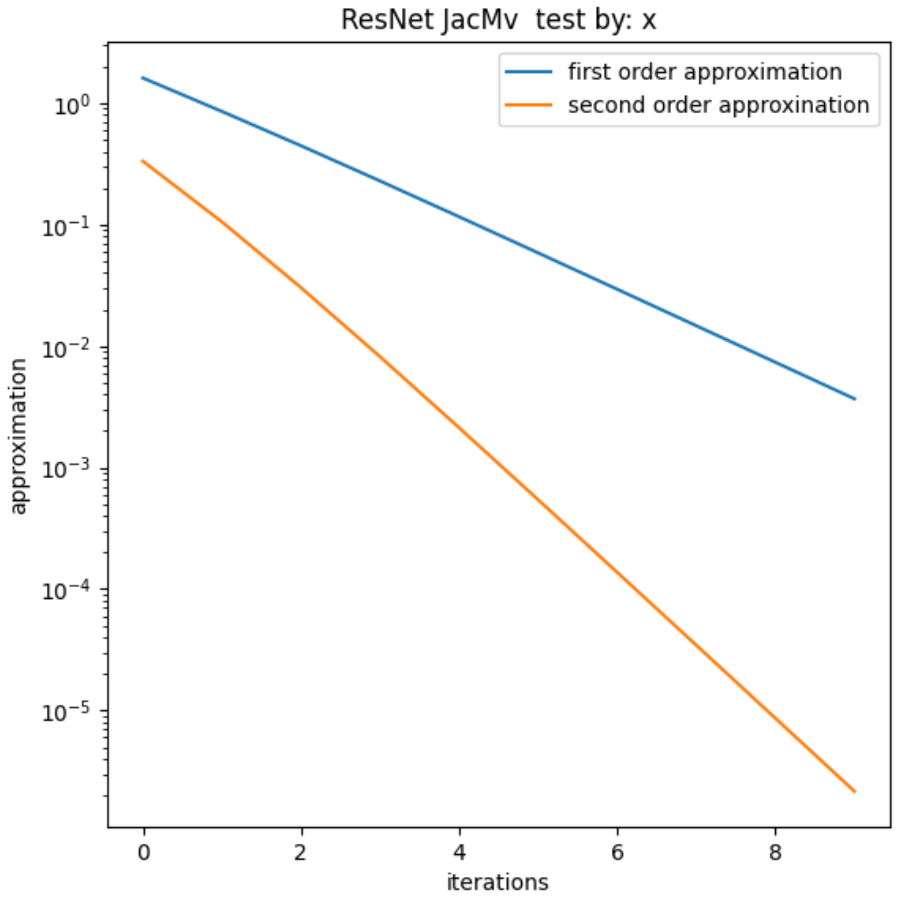


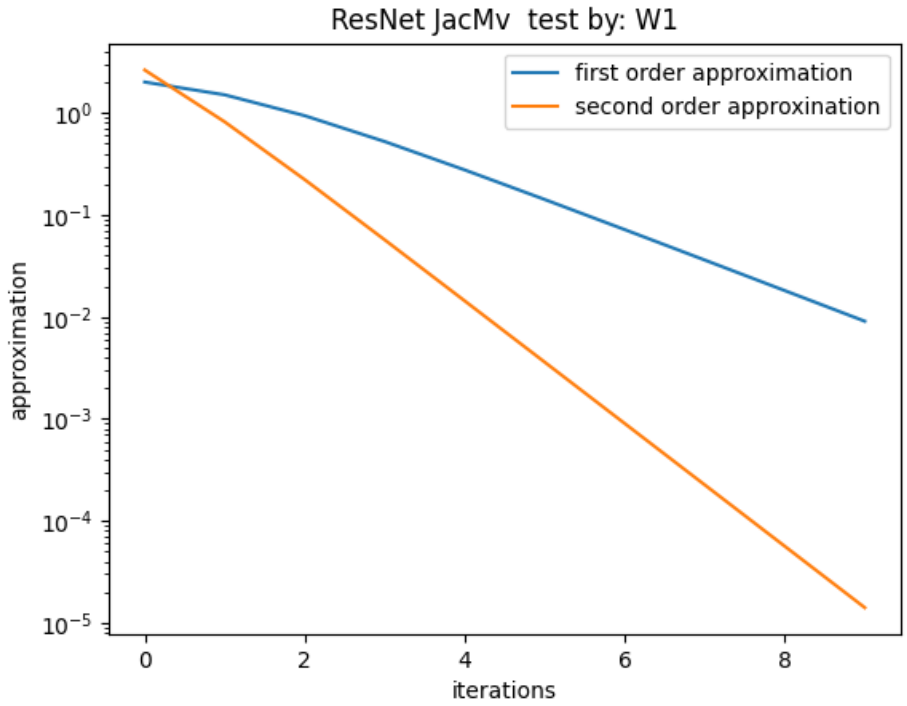


Remarks: 1. The biases are included in W.

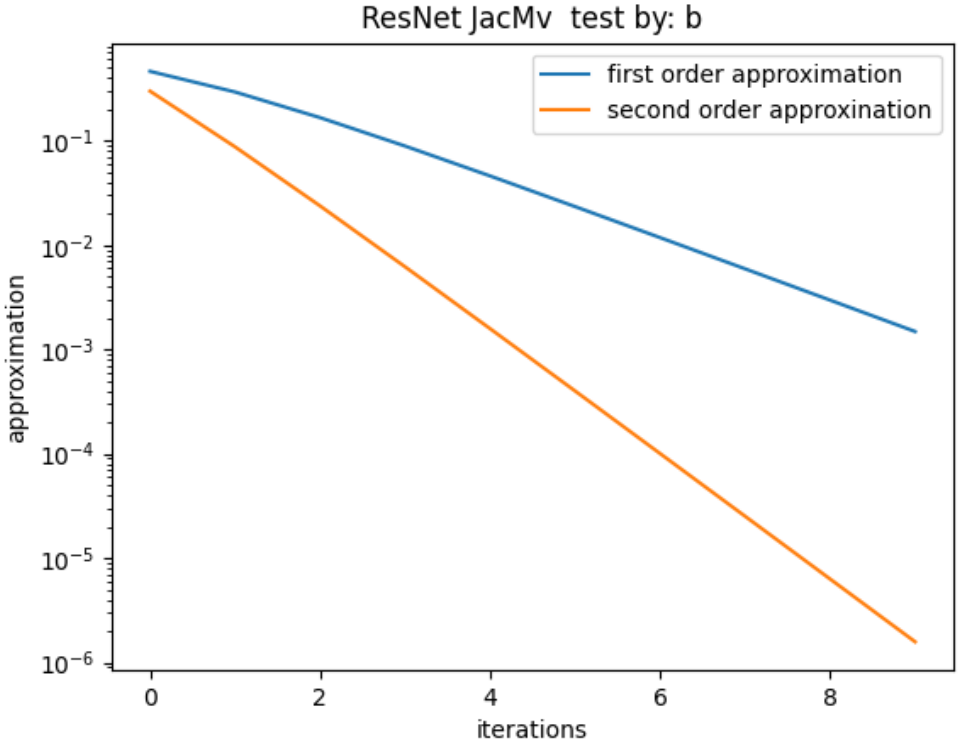
2. We confirmed that the Algebraic identity in which: $\forall u, v \; vectors, \;\; u^T J v = v^T J^T u$ holds.

Q2.2.2 – Same as 2.2.1 but for the residual network model:

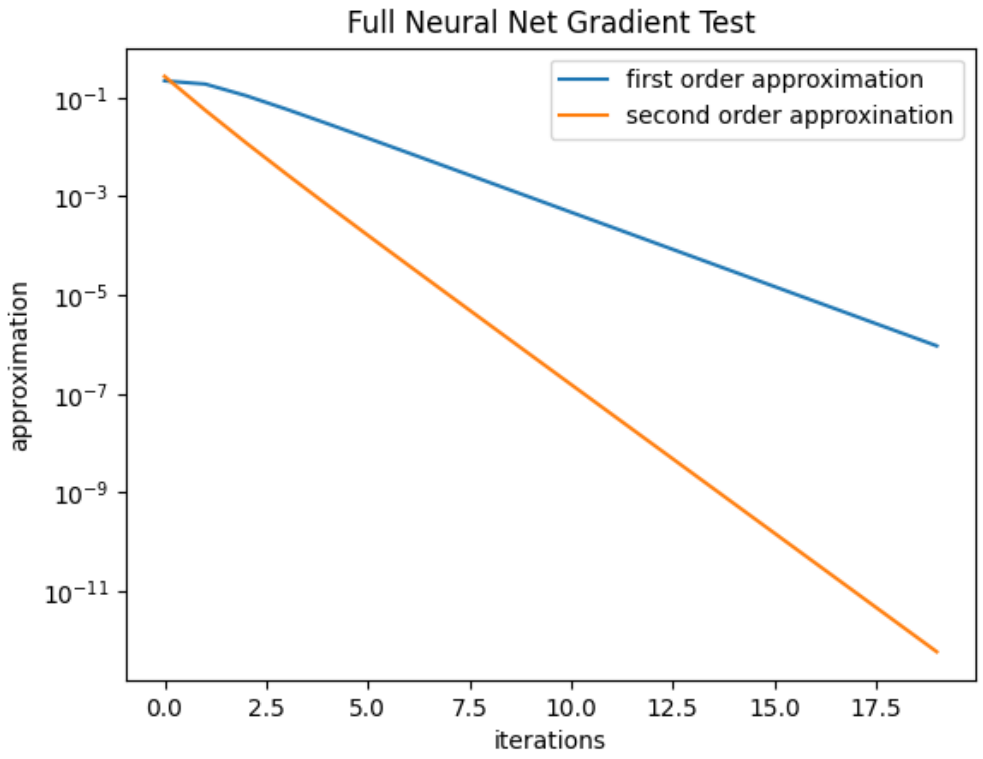




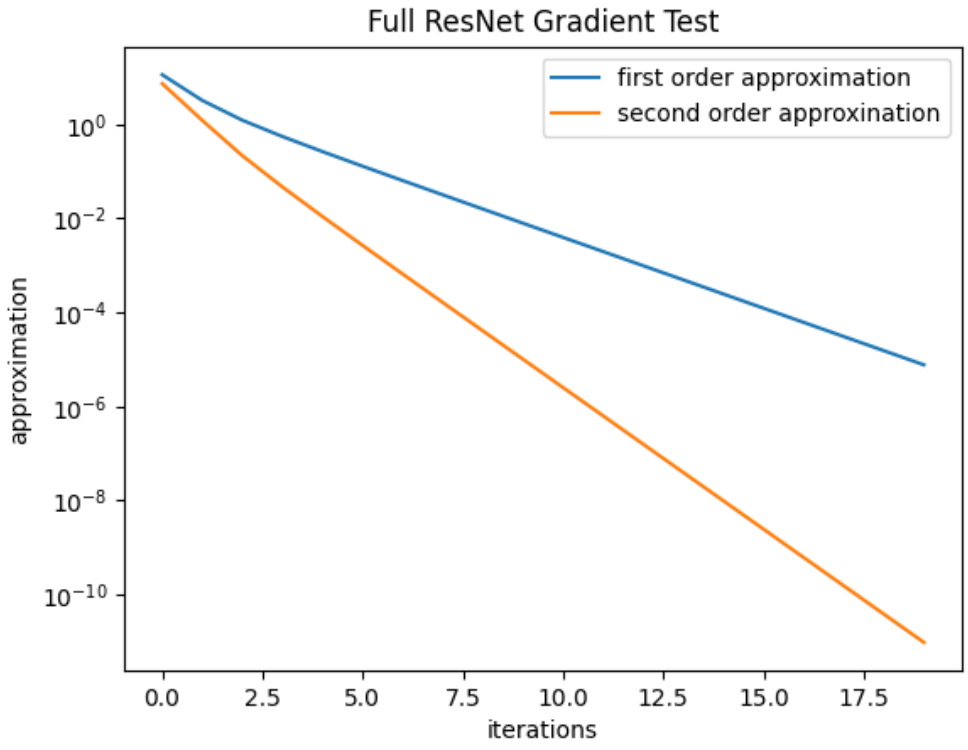

ResNet JacMv test by: W2
first order approximation
second order approxination
approximation
iterations
10^{1}
10^{-2}
10^{-5}
10^{-8}
10^{-11}
10^{-14}
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


ResNet JacMv test by: b
first order approximation
second order approxination
approximation
iterations
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
0
2
4
6
8

Q2.2.3 – The following is a plot of the gradient test for the entire standard neural network:

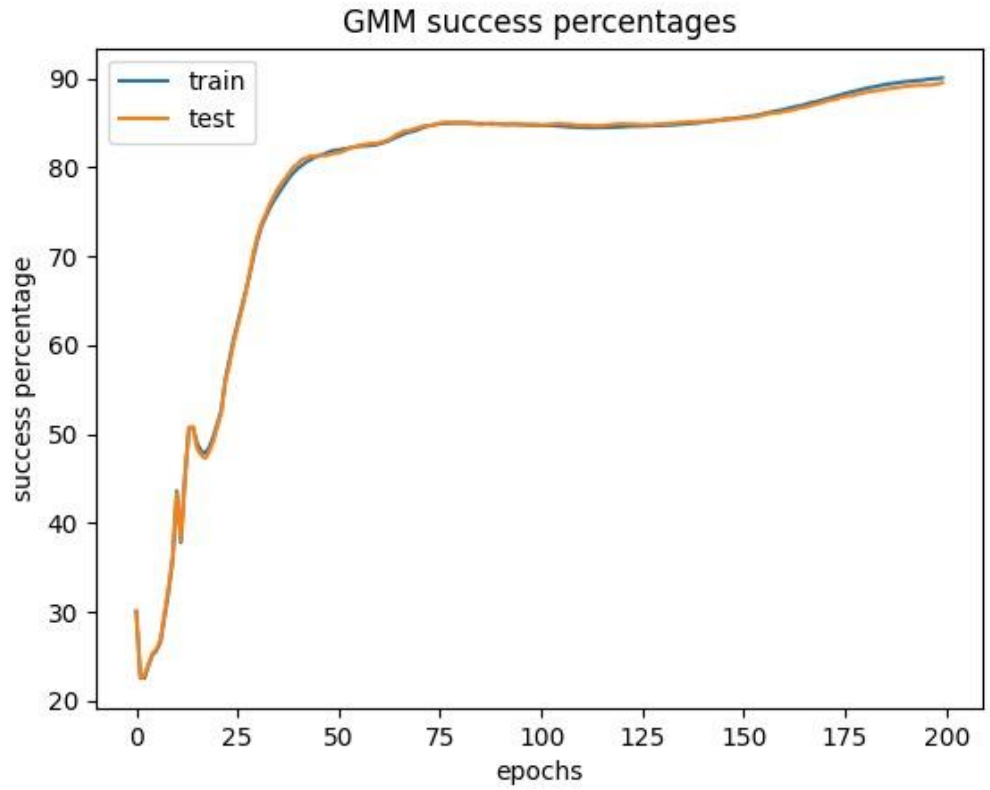


The following is a plot of the gradient test for the entire residual network:



Q2.2.4 – The following is a plot of the success rate of the neural network defined with two hidden layers of size 10 each, learning rate of 0.05, batch-size of 50 and 200 epochs.

For the GMM Dataset:



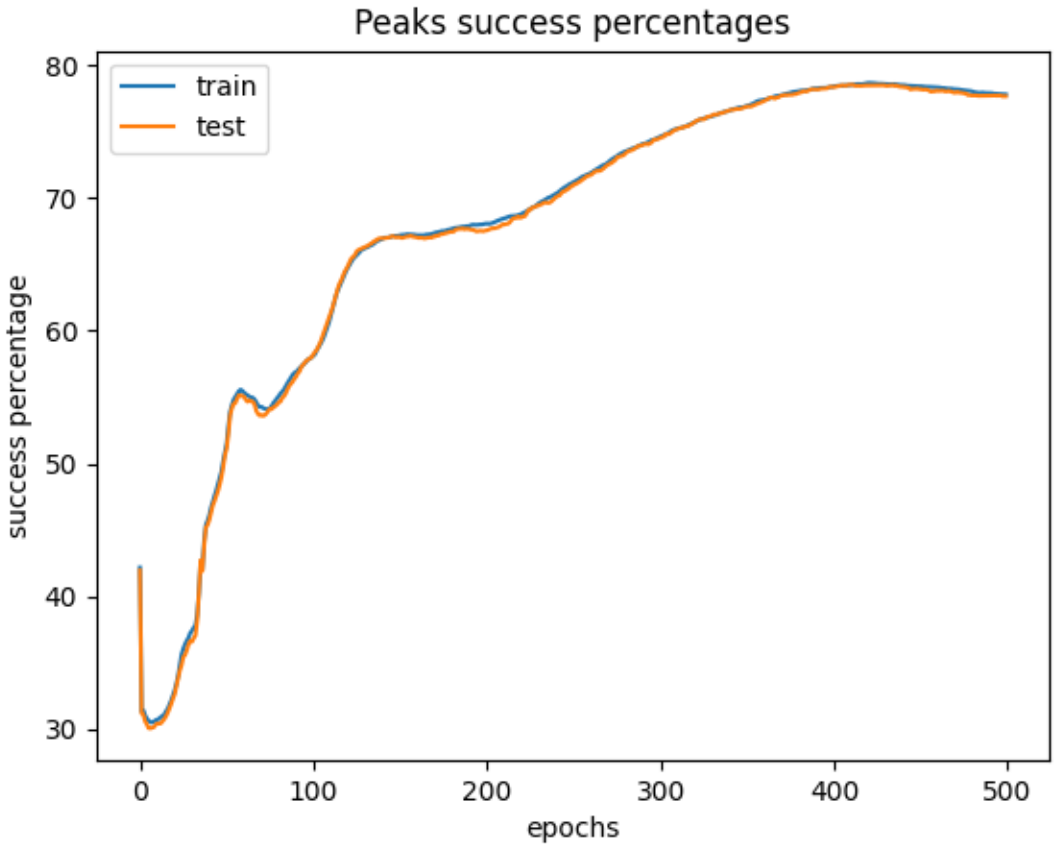
Additionally, we tried this experiment for the following hyperparameters and layer configurations:

Hidden Layer conf.	Mini-batch size	Learning Rate	Epochs	Validation success rate
[10,10]	200	0.0001	1000	27%
[10,10]	200	0.001	500	72%
[10,10]	50	0.01	300	94%
[10,10]	50	0.1	100	90%
[10,10]	50	0.05	1000	94.5%
[5, 5]	50	0.01	300	81%
[10,10,5]	50	0.01	300	87%

Our conclusions for this experiment are that we had the most success when the network was of the configuration – [10,10] and that the SGD minimization performs best with a learning rate of approximately 0.01 – 0.1.

For Peaks Dataset:

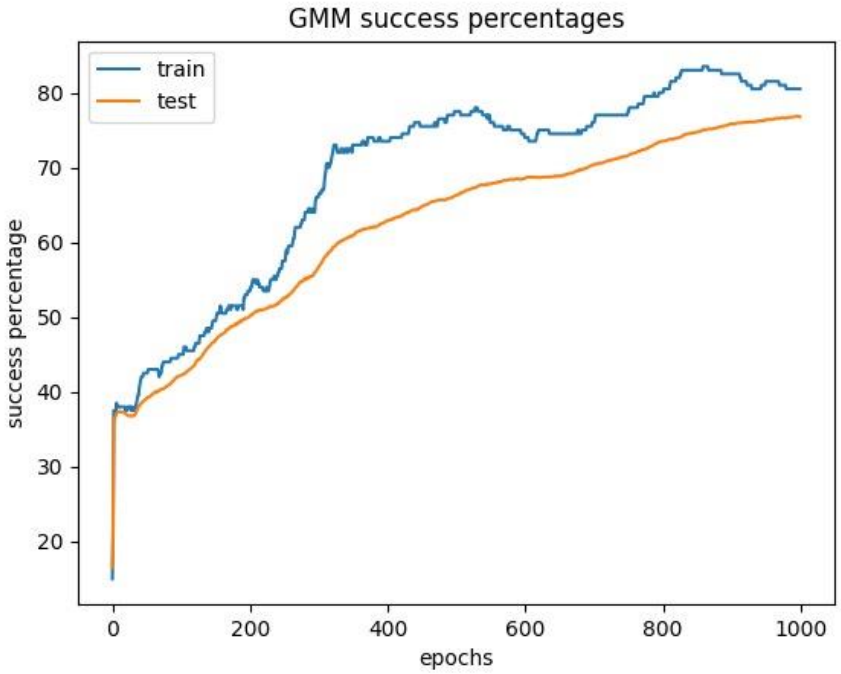
Hidden Layer conf.	Mini-batch size	Learning Rate	Epochs	Validation success rate
[20]	50	0.01	200	60%
[50]	50	0.01	200	47.5%
[]	50	0.01	200	45%
[15,10]	50	0.01	200	70%
[30,20]	50	0.01	1000	78%
[30,20]	100	0.01	500	65%
[30,20]	200	0.01	500	63%
[30,20,10]	50	0.01	200	65%
[30,20,10]	200	0.01	1000	68%



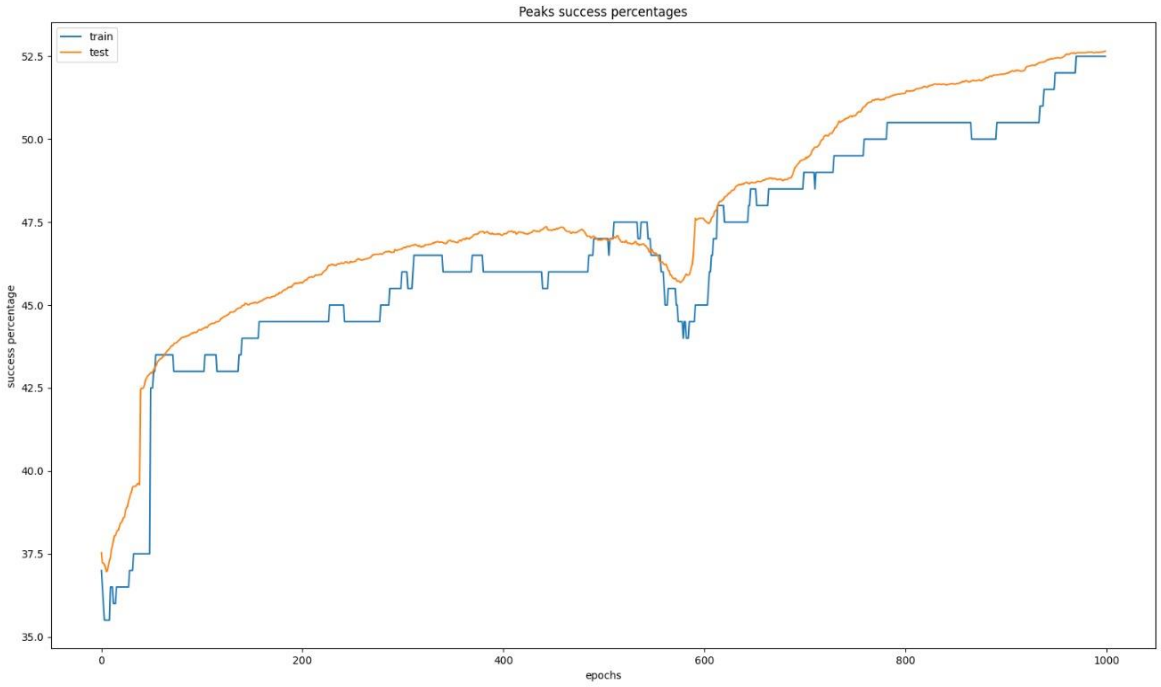
For this experiment we found that a Hidden layer configuration of [30,20] works relatively well, combined with a mini-batch size of 50.

Q2.2.5 – We conducted the same experiment as the most successful one for each data set but now using only 200 data points, the results are demonstrated in the following graphs:

For GMM:



For Peaks:



We received worse results for using only 200 data points, we see that it is not enough data in order to train the model in the optimal way possible.