AI Assignment 6

2. In order to complete this assignment you will need to produce two versions of your code above, the existing one with 3 internal nodes, and a second with 10. You will then compare the performance of the two on the test set with different learning rates. Along with your extra code, submit a report answering the following questions. You may use figures or statistics as appropriate.

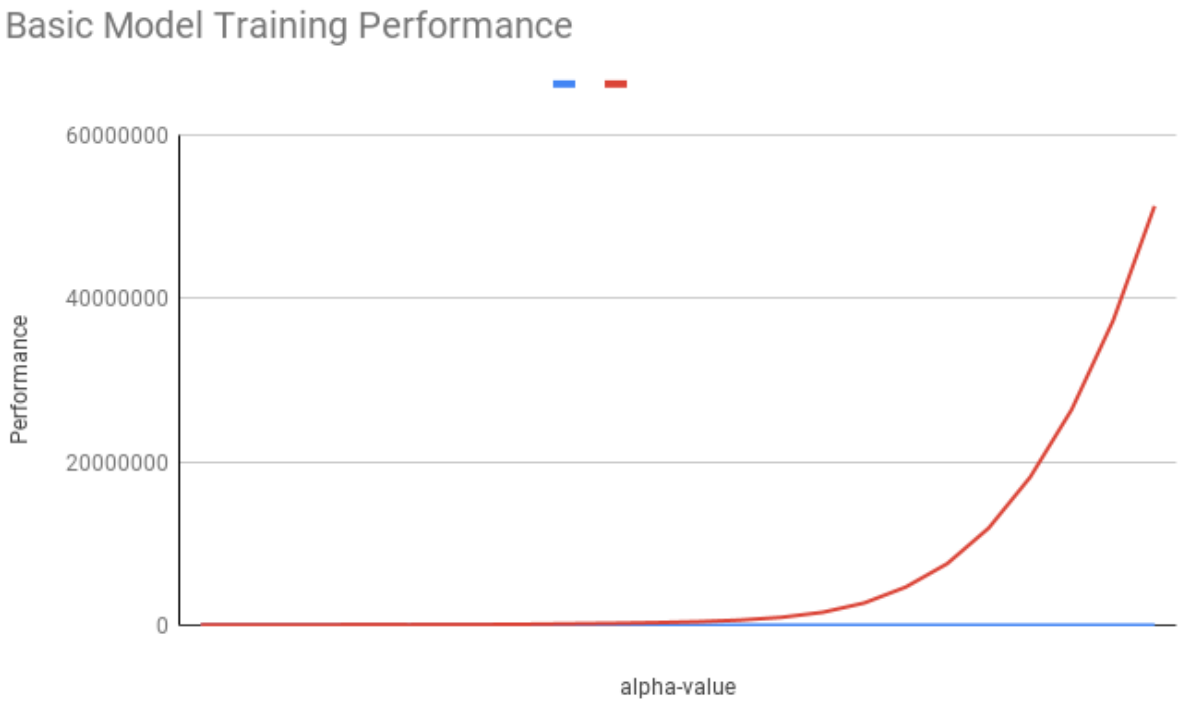
1. How did the two models compare on the training and testing sets?

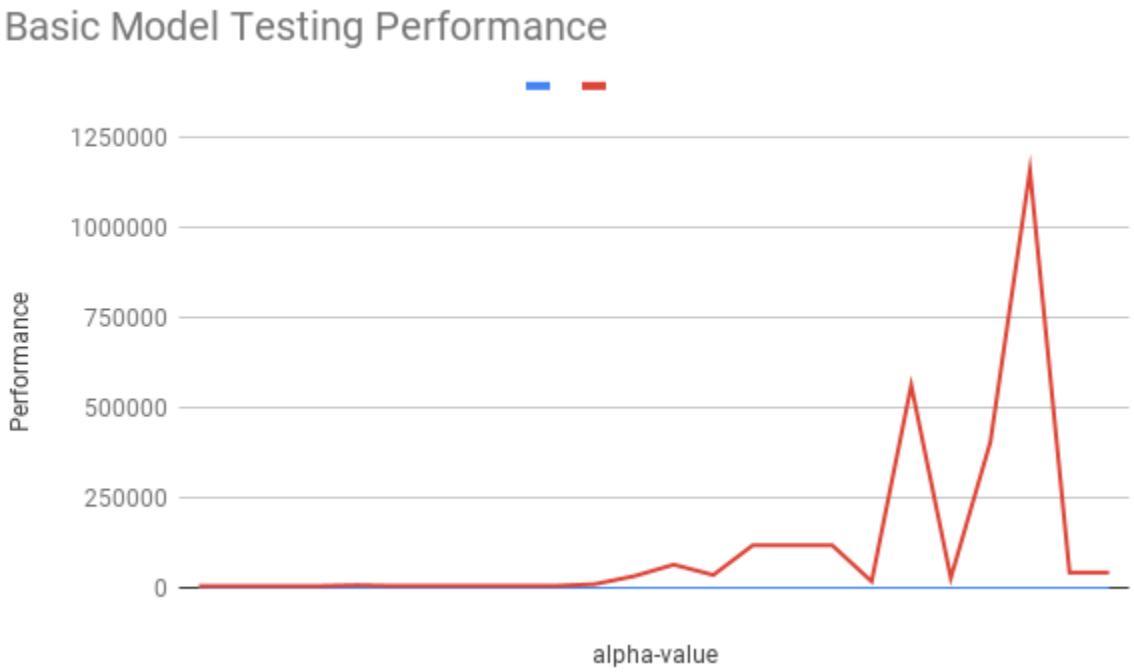
a. Basic Model Analysis

For different values of alpha we have obtained the following values.

alpha-value	Training Performance	Testing Performance
0.00001	72993.66481	7192.278223
0.0001	65275.2103	7192.094358
0.001	64577.66354	7210.459908
0.01	65738.06631	7407.770944
0.05	109336.7174	9894.6341
0.1	81161.0613	7702.871374
0.15	92760.18607	7704.499412
0.2	108719.621	7774.717614
0.25	139839.8849	7936.781896
0.3	193483.5712	7948.162177
0.35	250102.6462	12664.2583
0.4	331380.279	34915.04737
0.45	446654.7662	66754.90403
0.5	635972.5992	38082.33484
0.55	972568.5527	120745.3988
0.6	1594266.733	120745.3988
0.65	2733031.748	120745.3988
0.7	4682081.719	20308.35843
0.75	7544903.269	564099.5265
0.8	11897107.66	29973.16092
0.85	18120422.27	406643.6241
0.9	26382248.44	1159524.772
0.95	37269149.66	45125.70581
1	51294501.53	44605.5827

Below are the graphs plotted for the above data both for training and testing data.



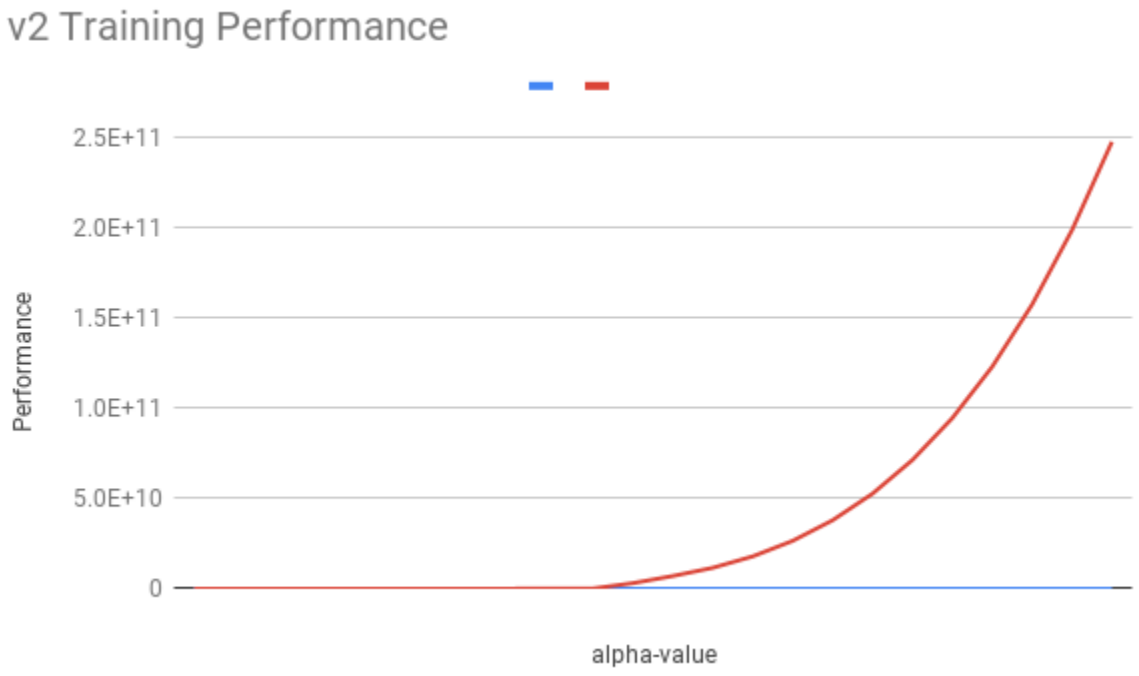


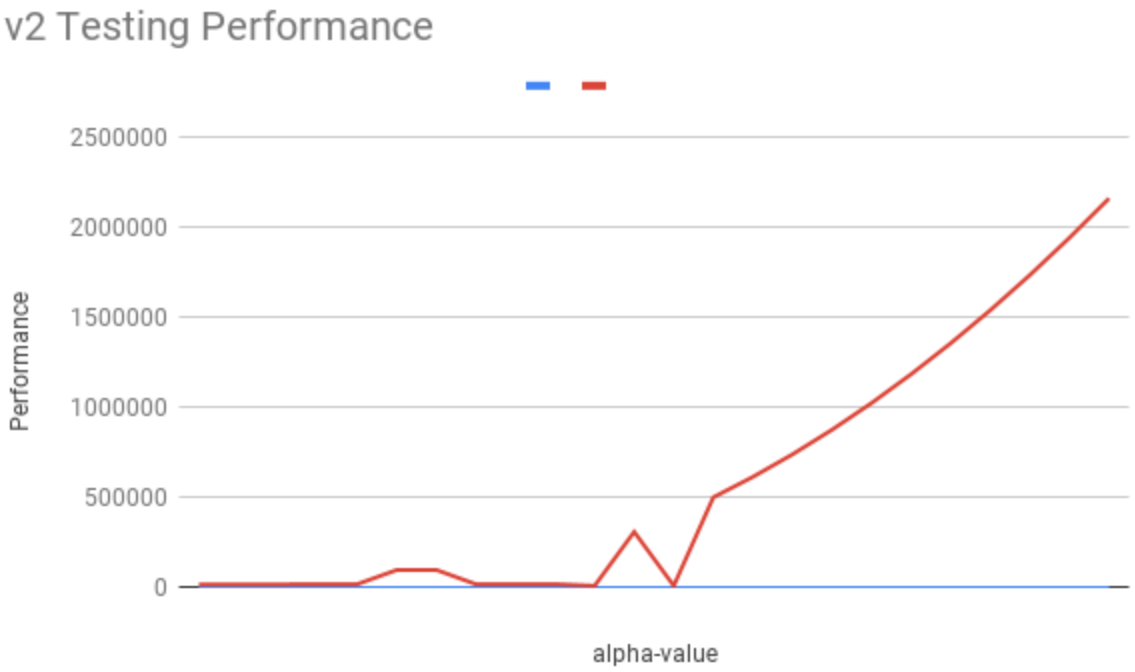
b. Second Version Model Analysis

For different values of alpha we have obtained the following values.

alpha-value	Training Performance	Testing Performance
0.00001	111592.6554	19154.37719
0.0001	71159.67166	19191.73894
0.001	67215.58729	19416.599
0.01	68245.02276	19934.02906
0.05	81663.7234	20273.20294
0.1	331439.1218	99342.82264
0.15	1536135.658	99722.89663
0.2	5062166.387	20397.93408
0.25	13045880.15	20531.93476
0.3	28613595.31	20542.13078
0.35	124153478.8	11576.10875
0.4	3091996588	311022.6928
0.45	7054559094	11576.10875
0.5	11598058318	502728.2432
0.55	17924474082	616107.6041
0.6	26541041744	741171.3554
0.65	37945592246	877919.4971
0.7	52683193355	1026352.029
0.75	71347617828	1186468.952
0.8	94581343416	1358270.265
0.85	123075552860	1541755.969
0.9	157570133894	1736926.063
0.95	198853679241	1943780.547
1	247763486618	2162319.422

Below are the graphs plotted for the above data both for training and testing data.





Training Data Comparison: From the above analysis for both models as the alpha value increases the squared error keeps on increasing. This is because we are only performing a single epoch to update the weights in the back propagation. Behaviour of both the models while training is same but for the version 2 as we have more number of nodes in the hidden layer the error rate is so high when compared to the basic model as the alpha-value keeps on increasing. So in conclusion the basic model is getting trained better than the version 2 model.

Testing Data Comparison: From the above analysis as the basic model got trained better than the version 2 the performance of the basic model is better than the version 2 model. Along with that, adding more nodes in the hidden layer will always lead to more error which is happening in the case of version 2 model. Even though we are observing some uneven spikes in the error rate as the alpha value keeps on increasing for both models, relatively basic model is better than the version 2 model.

2. Was there a clear relationship between the alpha values and the performance on the data?

From the above attached analysis figures for both the models, as the alpha value increases both the models were performing randomly with some spikes in between. With that behaviour, we can infer that alpha value should neither be too small or too large. If it does so it will diverge away from the global minimum. The alpha value should

always be fit in such a way that it starts converging towards the global minimum and the increment/decrement operations should be performed in small steps such that it gradually converges closer to the global minimum.