Exploring the Double Descent Curve in Deep Neural Networks

James Enouen
The Ohio State University
enouen.8@osu.edu

Abstract

This project intended to explore how the double descent curve from [1] is able to generalize to more complex models like deeper neural networks. Honestly, this work failed to get a perfectly comprehensive picture of exactly how the second descent interacts with multiple layers in a network; however, it manages to produce results manifesting the second descent and to give some future directions of further investigation.

I. Introduction

The goal of this project was to explore the double descent curve in its application to deep neural networks. The original goal was to thoroughly study how the double descent curve emerges throughout different datasets and different training regiments. This paper will walk through the journey of different approaches taken to attempt to find the double descent curve in this setting. Disappointingly, this paper spends more focus on 'how to train deep networks' than the originally desired 'investigating the second descent in deep networks.' In the end, there is no 'fairy tale ending' because none of the results are strongly conclusive; however, the final results indicate success in finding the second descent in the deep setting.

Let us begin with a review of what the double descent curve claims because it is the central focus of this paper. The paper [1] offers a new perspective on the classically believed bias-variance tradeoff. The new perspective suggests that we indeed get better results by adding more and more parameters/ hidden units to our model, despite the local maximum we approach as we reach the interpolation threshold of perfect fitting.

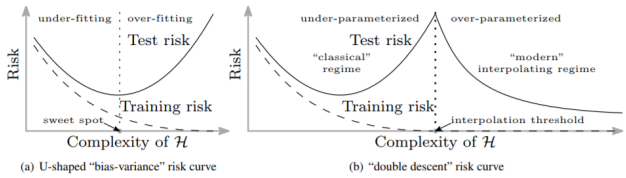


Fig. 1. Double descent curve from [1]

This goes against the traditionally believed idea that a model can overfit the training set and the only way to fix this problem is through regularization techniques. The paper suggests, moreover, that the minimum at the end of this curve generally tends to be lower than the local minimum attained in the bias-variance tradeoff curve. This advocates an overall different look on how to handle training models which is very interesting and consequently deserves study in applications like deep neural networks.

II. Inspirtation

This project began as a failure to easily replicate the results of the double descent curve in a deep neural network scenario. These original results were trained on a dataset similar to the one below. As can be seen in the tables below, the results are very far away from what one might expect for the double descent curve to react.

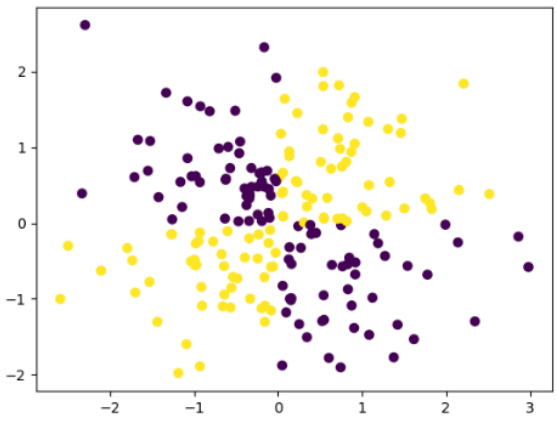
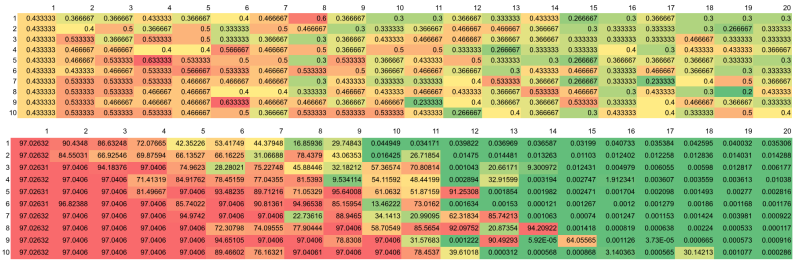


Fig. 2. Depiction of the dataset used



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.433333	0.366667	0.366667	0.433333	0.366667	0.4	0.466667	0.6	0.366667	0.3	0.3	0.366667	0.333333	0.433333	0.266667	0.3	0.366667	0.3	0.3	0.3
2	0.433333	0.4	0.5	0.366667	0.5	0.333333	0.5	0.466667	0.3	0.333333	0.366667	0.466667	0.466667	0.366667	0.3	0.333333	0.333333	0.3	0.266667	0.333333
3	0.433333	0.533333	0.366667	0.533333	0.5	0.333333	0.366667	0.3	0.366667	0.433333	0.466667	0.466667	0.366667	0.366667	0.333333	0.333333	0.333333	0.466667	0.333333	0.333333
4	0.433333	0.466667	0.466667	0.4	0.4	0.566667	0.466667	0.5	0.366667	0.5	0.5	0.333333	0.266667	0.333333	0.333333	0.4	0.3	0.433333	0.433333	0.366667
5	0.433333	0.466667	0.533333	0.633333	0.533333	0.5	0.5	0.3	0.533333	0.366667	0.433333	0.5	0.333333	0.3	0.266667	0.366667	0.366667	0.366667	0.333333	0.3
6	0.433333	0.433333	0.466667	0.533333	0.566667	0.533333	0.466667	0.533333	0.5	0.366667	0.466667	0.366667	0.3	0.433333	0.466667	0.333333	0.466667	0.366667	0.3	0.333333
7	0.433333	0.533333	0.533333	0.533333	0.466667	0.466667	0.466667	0.3	0.433333	0.333333	0.333333	0.4	0.533333	0.366667	0.266667	0.333333	0.233333	0.4	0.5	0.366667
8	0.433333	0.533333	0.533333	0.466667	0.466667	0.4	0.4	0.333333	0.466667	0.433333	0.466667	0.433333	0.366667	0.533333	0.466667	0.3	0.433333	0.3	0.2	0.433333
9	0.433333	0.533333	0.466667	0.466667	0.466667	0.633333	0.466667	0.466667	0.366667	0.466667	0.233333	0.4	0.366667	0.366667	0.533333	0.333333	0.4	0.466667	0.366667	0.333333
10	0.433333	0.533333	0.533333	0.533333	0.466667	0.5	0.366667	0.533333	0.533333	0.533333	0.433333	0.266667	0.4	0.366667	0.3	0.433333	0.4	0.333333	0.5	0.4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	97.02632	90.4348	86.63248	72.07665	42.35226	53.41749	44.37948	16.85836	29.74843	0.044949	0.034171	0.039822	0.036969	0.036587	0.03199	0.040733	0.035354	0.042595	0.040032	0.035398
2	97.02632	84.55031	66.92546	69.87594	66.13527	66.16225	31.06888	78.4379	43.06353	0.016425	26.71854	0.01475	0.014481	0.013263	0.01103	0.012402	0.012258	0.012836	0.014031	0.014286
3	97.02631	97.0406	94.18376	97.0406	74.9623	28.28021	75.22748	45.88446	32.18212	57.36574	70.80814	0.001043	20.66171	9.300972	0.012431	0.004979	0.006055	0.00598	0.012817	0.006177
4	97.02632	97.0406	97.0406	71.41319	84.91762	78.45159	77.04355	81.5393	9.534114	54.11592	48.44199	0.002894	32.91599	0.003194	0.002747	1.912341	0.003607	0.003559	0.003613	0.01038
5	97.02631	97.0406	97.0406	81.49667	97.0406	93.48235	89.71216	71.05329	95.64008	61.0632	51.87159	91.25308	0.001854	0.001982	0.002471	0.001704	0.002098	0.001493	0.00277	0.002816
6	97.02631	96.82388	97.0406	97.0406	85.74022	97.0406	90.81361	94.96538	85.15954	13.46222	73.0162	0.001634	0.00153	0.000121	0.001267	0.0012	0.001279	0.00186	0.001168	0.001176
7	97.02632	97.0406	97.0406	97.0406	94.9742	97.0406	97.0406	22.73616	88.9465	34.1413	20.99095	62.31834	85.74213	0.001063	0.00074	0.001247	0.001153	0.001424	0.003981	0.000922
8	97.02632	97.0406	97.0406	97.0406	97.0406	72.30798	74.09555	77.90444	97.0406	58.70549	85.5654	92.09752	20.87354	94.20922	0.001418	0.000819	0.000638	0.00224	0.000533	0.000117
9	97.02632	97.0406	97.0406	97.0406	97.0406	94.65105	97.0406	97.0406	78.8308	97.0406	31.57683	0.001222	90.46293	5.92E-05	64.05565	0.001126	3.73E-05	0.000965	0.000573	0.000916
10	97.02632	97.0406	97.0406	97.0406	97.0406	89.46602	76.16321	97.04061	97.0406	97.0406	78.4537	39.61018	0.000312	0.000568	0.000968	3.140363	0.000565	30.14213	0.001077	0.000286

Fig. 3. Training and testing results for original dataset

This clearly does not depict the pattern we were looking for, Because increasing the length of the network also increases the number of parameters, it should increase the complexity of the model. Hence, the results need to be 'flipped' as can be seen in the following two graphics. This is clearly an issue of the results which were gathered.

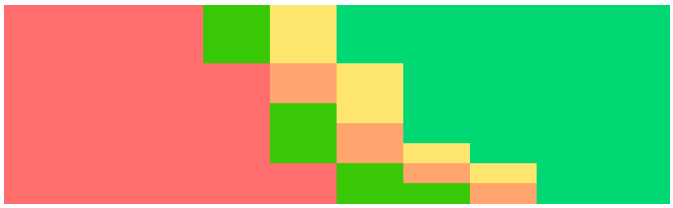


Fig. 4. Observed pattern of the validation errors

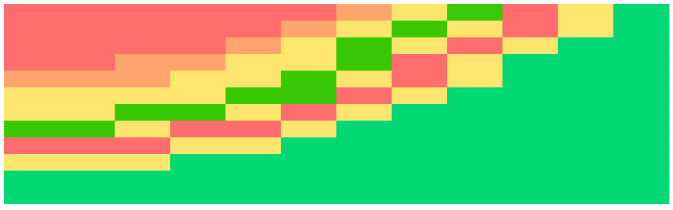


Fig. 5. Expected pattern of the validation errors from double descent perspective

A possibly even more fundamental issue is the fact that the more complex models had trouble scoring even better on the training loss function, which is fairly unreasonable. More advanced models have strictly greater functional complexity. This points to there being an issue in how these models are being trained. The rest of this paper are the steps taken to remedy this issue in light of the ultimate goal of finding the double descent curve in this setting.

III. Exploratory Journey

A. Beginning Failures

The first steps taken to try to fix this issue was to try the same procedure on different datasets. Trying these on a sphere in two and four dimensions, the following results were generated:

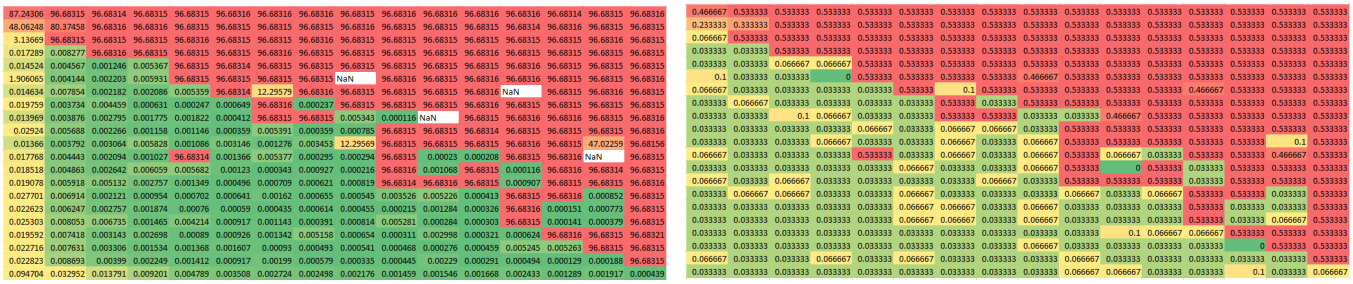


Fig. 6. 2D sphere training loss and testing error

Fig. 7. 4D sphere training loss and testing error

Both of these trials are not a huge step above the work started with. While the axis for these charts are now flipped from the previous images, they still have the issue of being in the wrong direction for the 'complexity' of the model. There are quite a few more failures of these first steps taken towards exemplifying the double descent curve. Notably, these charts have NaN as one of their values. This is due to numerical issues to do with logarithms and exponents. This happens to be because the loss function these training procedures are using is negative log likelihood. In retrospect, this was a little bit of an unreasonable loss function, considering the original paper was using mean squared error and zero-one loss as their functions. It is still interesting to see if this loss function has the potential to see the double descent curve itself, but this will be a future direction not included in this paper. Clearly, the training methods being used were insufficient to properly train a deep neural network which led to these results.

One of the first failures which unaligned the training procedure from the double descent paper was the size of the dataset. The goal of the second descent is to be able to perfectly fit the training set and the set being used was probably to large to feasibly do this with the number of parameters being used, hence the size of the training dataset was reduced and resulted in the following miniscule changes:

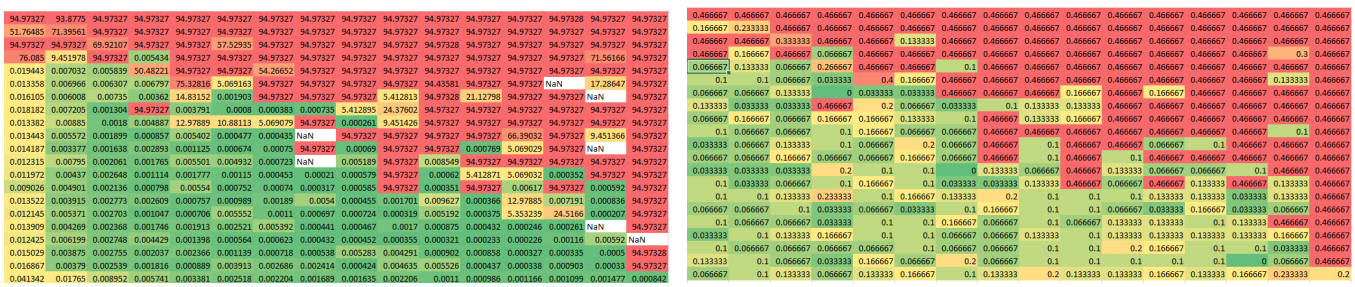


Fig. 8. Smaller dataset - 4D cylinder training loss and testing error

B. Minor Improvements (pretrain)

One of the characteristics that distinguishes the random Fourier features setting (where the double descent curve was discovered) from the neural network setting is the variance between the different layers of the model. The next step taken to fix this training procedure is an attempt to embrace this property that random Fourier features are fixing all previous layers and only training a final layer. For this, the models which got longer and longer by adding hidden layers would start off by using the original layers of the trained previous model. In this way, the goal was to reduce the variability by random initialization greatly and to smooth the results in the direction of longer and longer machines. This definitely made a notable improvement in the training losses of the models; however, it barely made a dent in the large scheme of our goal. This had nowhere near enough impact to 'flip' the results and barely added anything new to the boundary of our diagonal curve:

Fig. 9. First pretrained training losses and testing errors

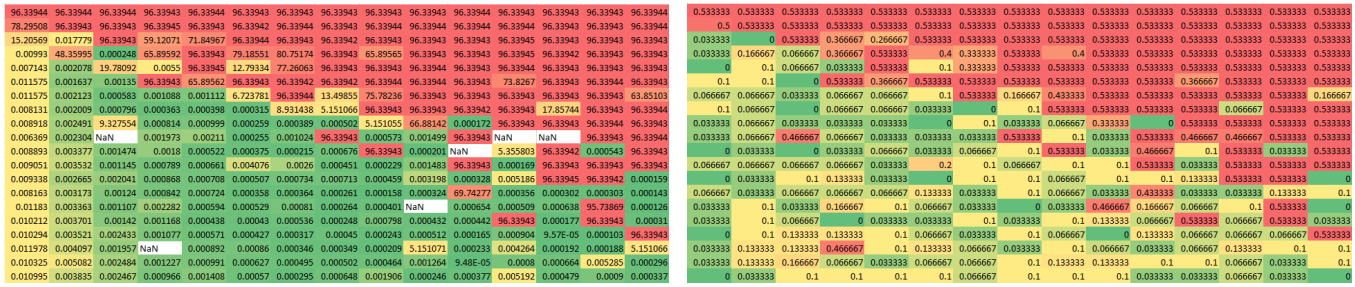


Fig. 10. Second pretrained training losses and testing errors

C. Fair Improvements (MSE)

The next alteration to the training procedure was to use MSE instead of NLL. This had a very noticable impact on the results. This was the first dataset where the 'flip' was acheived. This is a vast improvement over the previous sets of models for this reason. The training loss curve made sense given the complexities of the models. The results are provided below:

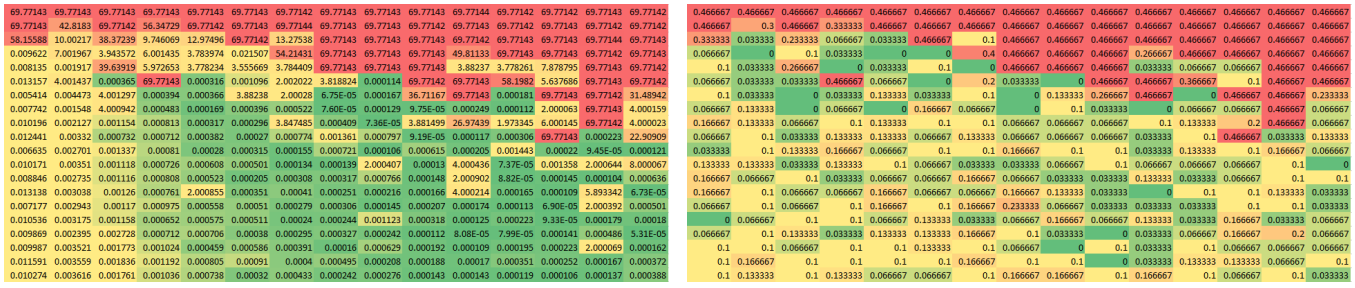


Fig. 11. First MSE training loss and testing error data

The shortcoming of this data is twofold. The first aspect is that neither are actually able to obtain the zero training loss which is required to hit the interpolation threshold. The second aspect which is in part a consequence of the first is how great the variance is in the testing accuracies of these models. There seems to be no clear cut pattern throughout these models. One could argue some small patterns like very good generalizability right after dropping off of the total inability to score well on the training set. This corresponds to the green ridge near the 'cliffs' of red and orange near the top and right of these

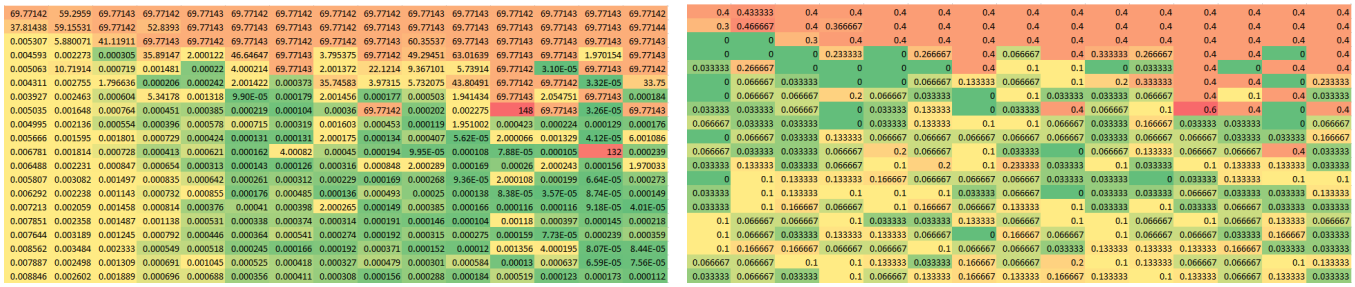


Fig. 12. Second MSE training loss and testing error data

depictions. It is likely that these correspond to the first descent in the double descent curve, but the goal of this project is to see the second descent.

Additionally, this training procedure was unable to perform well on more complex datasets. The following results were for a six dimensional version of the checkerboard pattern that was used in the first dataset used. Clearly the models were not able to properly learn the training set:

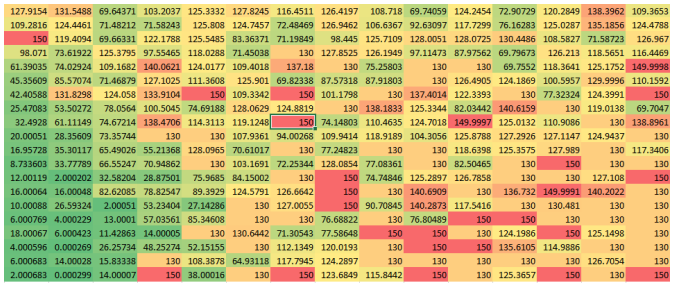


Fig. 13. Failure of MSE on six-dimensional dataset (training loss)

D. Learning Rate Schemes

The work of many others including those in [2] was sufficiently inspiring to alter the training of these models with a new, adaptive learning rate. This paper discusses more advanced learning rate procedures, but the one implemented for the next results is simply a decreasing function over time. It was, however, able to drastically outperform the same procedure with a fixed learning rate. On the same dataset as the last figure from the previous section, the new procedure was able to get the following results:

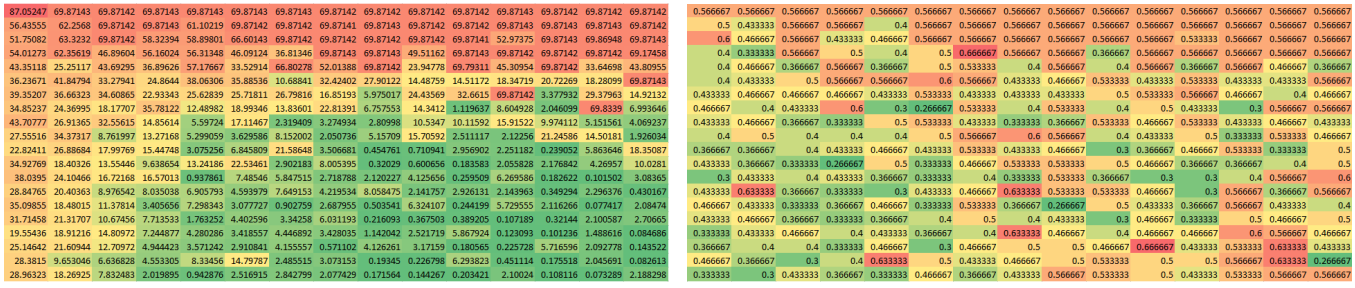


Fig. 14. Adapted learning rate training losses and testing errors

This data is clearly much more demonstrative of any type of second descent and achieved quite good training results. It seems almost that the valley of green possibly corresponds to the first descent. Hopefully extending the picture just a little would be sufficient to see the second descent. The next attempt was exactly this, running the same procedure for larger numbers of hidden layers and larger hidden unit sizes on a slightly simpler dataset. The results for this expanded chart are below:

E. Discussion

There is quite a lot of information in these two rectangles, so let us take this time to have a brief discussion about some of their features. A very prominent feature is the dark bands across the tops of these rectangles. This corresponds to the fact that,

39.9417	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286
31.47692	33.19711	34.74286	34.71488	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286
14.98706	16.58744	24.95724	34.74286	5.981618	34.74286	49.91169	32.78478	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286	34.74286
2.282926	5.94706	18.69151	7.781635	23.15639	18.75115	34.74286	24.64753	18.46542	34.74286	33.94915	24.28305	26.68446	34.74286	34.74286	34.74286	34.74286	34.74262	34.74286	
2.247073	1.593039	2.292514	3.803061	7.291332	5.63022	4.222106	21.03856	10.62554	4.506479	32.81013	10.03653	34.74286	15.50355	2.372525	34.74286	34.74286	7.677993	34.74286	
2.296677	6.267552	1.496338	1.315136	0.154206	5.100445	5.728635	4.937567	17.76291	4.864257	7.476812	1.758925	4.186625	2.528789	4.010234	14.46314	6.369112	28.7979	3.801031	
5.373813	7.192982	6.869622	0.185548	1.287167	0.154183	0.555633	4.267349	0.846209	28.11708	9.443418	0.205194	5.932609	7.068625	2.802536	4.557863	1.9718	10.43444	11.42129	
2.049911	3.266126	4.809151	0.242345	11.44112	2.21134	7.375144	0.207576	0.356377	1.099559	10.3798	2.137265	0.164463	0.196435	0.07413	1.938145	0.489154	5.23939	7.564335	
31.03781	1.237054	1.241145	7.422771	0.17964	0.580466	1.112597	2.10807	1.899871	0.081484	1.75126	0.112124	0.026047	2.013572	2.088425	4.120001	15.19349	30.21604	2.06243	
0.908437	2.802142	0.51364	0.370287	1.264288	2.078053	0.255236	0.162839	2.248136	0.153993	0.29939	0.42735	0.314579	3.522712	5.133423	0.099321	3.900885	0.06662	8.410292	
2.098577	0.684358	0.315468	0.208121	0.788801	1.076674	0.066442	2.055835	0.039216	2.026525	0.189585	0.11605	3.451273	1.573526	0.061823	0.014995	5.329741	2.124883	0.06361	
1.189087	1.265671	0.359632	0.308572	0.420412	0.153586	0.418844	0.71978	12.15858	0.654248	0.331832	5.89201	0.134371	2.184283	0.100736	5.355388	0.037253	0.121158	0.033781	
1.844171	0.678073	1.053086	0.435192	0.23183	0.138739	0.195829	0.434994	0.589932	2.064511	0.108165	0.06464	0.92434	0.059804	0.193243	2.020034	0.446719	0.063969	0.015934	
4.209053	2.342682	2.119465	0.510783	0.556164	0.350045	1.55069	0.161677	0.205248	0.052741	0.218655	1.77401	0.165593	3.956553	0.048979	2.050043	0.152669	0.040389	0.021217	
1.928752	1.558422	2.506073	0.114623	1.456679	4.540766	0.35756	2.220635	0.062739	0.120547	0.104166	1.172507	1.958015	0.134788	0.079202	0.259371	0.046829	0.132919	2.031734	
5.438932	1.513699	0.673597	0.442767	0.23942	0.153993	0.277395	2.129183	0.449645	0.112207	0.087891	0.357475	0.106922	0.146741	0.091046	0.073679	1.533473	0.158596	2.056636	
5.143455	1.745564	0.689625	1.203583	0.203316	2.501852	0.386905	2.096021	0.059385	0.341083	0.085737	1.067446	0.060121	0.049433	0.068514	0.017595	0.064978	0.123348	0.036471	
2.259663	0.86723	0.690636	1.312608	0.217436	1.793147	0.179106	1.826355	0.085773	2.426672	0.085666	0.114254	0.027877	0.025784	0.037866	0.056284	0.434645	0.072378	0.198752	
1.873895	1.030586	0.550946	0.338037	0.213402	0.194999	0.241752	1.116793	0.151986	0.325657	0.069592	0.063259	0.179947	0.076672	0.11008	0.04366	0.013054	0.036346	1.879442	
2.877115	1.010666	1.145105	1.980436	0.811049	0.659032	0.385791	0.06621	0.111129	0.080797	0.094188	0.054086	0.263589	0.056903	3.495325	1.950264	0.036949	0.027648	0.204878	
2.10755	0.942446	0.553375	0.422572	0.226875	0.201947	0.460034	0.093659	0.287971	0.103183	0.075673	0.0394	0.272913	0.040174	0.165893	0.026701	0.043485	0.10349	0.063559	
1.447685	3.067272	0.603279	0.30874	0.377736	0.268805	0.247923	0.265839	0.149849	0.056178	0.128143	0.042736	0.052183	2.311809	0.102786	0.105583	0.09213	0.069103	0.144502	
2.824826	1.510198	0.454975	1.261757	0.158276	0.266442	0.17051	2.141584	0.199813	0.12988	0.170626	0.197427	0.19284	0.228809	0.050354	0.048216	0.084863	0.051222	0.166556	
0.000297	2.670968	4.567015	0.941525	1.215613	3.355499	0.39357	0.271351	0.148173	0.152209	0.260518	0.068785	8.093704	0.130921	0.048387	0.123981	0.048483	0.546659	0.050921	
4.796639	2.761919	3.183969	2.185476	0.728974	1.366859	0.434789	0.637602	0.136494	0.141417	0.107077	0.039117	2.026295	0.18972	3.220253	0.065717	0.073408	0.055815	0.112957	
3.87012	3.809398	1.647767	1.283159	1.133328	0.590958	2.240679	0.504976	0.086182	0.526281	0.153265	0.046704	0.133403	0.285114	0.059932	0.085091	0.030363	0.052676	0.041732	
5.126562	1.851919	2.190931	1.567775	3.623433	0.255663	1.79723	0.130625	0.425317	0.196938	0.253232	0.215806	0.082268	0.070873	0.037717	0.299061	0.037353	0.087382	0.054459	
4.440878	2.414637	2.759219	0.584658	1.102695	0.377184	0.567491	0.409864	0.256514	0.183342	0.157168	0.099715	0.243858	0.068336	0.238571	0.05441	0.082036	0.032213	0.028427	
3.686091	3.763508	2.857501	1.023834	0.647066	0.358244	0.234957	0.226534	0.216786	0.085903	0.126898	0.104542	0.127862	0.164517	0.213745	8.076534	0.088079	0.247154	0.058188	
4.262071	2.05817	2.510567	1.995517	0.750161	1.509002	0.528978	0.377691	0.164282	0.098571	1.975671	0.066347	0.194716	0.058255	0.07315	0.072057	0.033195	0.095205	0.022615	
0.000306	2.152115	0.929466	0.977163	0.602972	0.352532	0.222015	0.254741	0.327054	0.115851	0.364626	0.206214	0.11394	0.081146	0.066085	0.201279	0.045758	0.039036	0.046646	
3.93426	3.388699	0.9211	0.764252	0.901613	0.43058	0.176854	0.205336	0.186896	0.274044	0.092932	0.041162	0.077522	0.10428	0.098562	2.03825	0.421858	0.088	0.047861	
2.676389	1.481096	1.180168	1.027325	0.417983	0.975241	0.373551	0.355219	0.15884	0.413191	0.051395	0.101291	0.074063	0.086451	0.105259	0.120222	0.028778	0.064582	0.042546	
2.620187	1.807732	0.752876	0.562303	0.244927	0.446827	0.488119	0.366506	0.164937	0.097363	0.064871	0.06697	0.095671	0.081471	0.137636	0.124231	0.042716	0.090031	0.072217	
3.397875	1.145579	1.24235	0.497581	0.352588	0.483908	0.286396	0.315066	0.149324	0.162142	0.141119	0.139981	0.064945	0.07934	0.042119	0.08076	0.068992	0.076972	0.067589	
3.903199	2.999649	1.090216	0.692039	0.524667	0.220282	0.366775	0.110329	0.128766	0.257819	0.10569	0.225388	2.101204	0.174206	0.050304	0.04643	0.109232	0.058927	0.021831	
5.555126	1.871929	1.458859	0.757569	0.821778	0.1835	0.56515	0.207468	0.138756	0.118212	0.257585	0.07798	0.06911	0.1317	0.081223	0.035982	0.034037	0.031133	0.032664	
4.699596	4.146275	1.806579	0.374469	0.593048	0.373389	0.259562	0.296512	0.449408	0.127126	0.298013	0.049891	0.07141	0.19392	0.047913	0.121584	0.031675	0.029674	0.045897	
3.348598	3.014014	1.633944	0.81119	0.180278	0.249446	0.240782	0.197854	0.138951	0.203074	0.238871	0.079958	0.082339	0.16024	0.076546	0.132377	0.150415	0.042049	0.036654	

Fig. 15. Larger model space - adapted learning rate training losses

0.607143	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	
0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	
0.5	0.5	0.285714	0.392857	0.25	0.392857	0.464286	0.25	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	
0.214286	0.321429	0.357143	0.178571	0.464286	0.464286	0.392857	0.535714	0.5	0.392857	0.392857	0.25	0.428571	0.392857	0.392857	0.392857	0.392857	0.392857	0.392857	
0.178571	0.214286	0.25	0.428571	0.25	0.357143	0.178571	0.535714	0.321429	0.178571	0.392857	0.321429	0.392857	0.607143	0.428571	0.392857	0.392857	0.25	0.392857	
0.25	0.321429	0.392857	0.357143	0.178571	0.357143	0.357143	0.285714	0.535714	0.357143	0.5	0.357143	0.321429	0.178571	0.392857	0.285714	0.464286	0.642857	0.392857	
0.25	0.25	0.428571	0.25	0.214286	0.214286	0.321429	0.25	0.5	0.321429	0.392857	0.25	0.321429	0.357143	0.285714	0.392857	0.357143	0.25	0.464286	
0.178571	0.214286	0.392857	0.321429	0.464286	0.25	0.25	0.321429	0.321429	0.357143	0.392857	0.214286	0.25	0.25	0.142857	0.285714	0.392857	0.535714	0.321429	
0.5	0.178571	0.25	0.285714	0.25	0.285714	0.392857	0.214286	0.285714	0.392857	0.464286	0.5	0.25	0.392857	0.321429	0.285714	0.428571	0.464286	0.321429	
0.214286	0.285714	0.214286	0.178571	0.357143	0.178571	0.285714	0.428571	0.25	0.535714	0.285714	0.571429	0.25	0.392857	0.25	0.428571	0.5	0.285714	0.392857	
0.142857	0.142857	0.321429	0.142857	0.428571	0.285714	0.214286	0.214286	0.178571	0.5	0.392857	0.392857	0.357143	0.214286	0.5	0.214286	0.321429	0.642857	0.25	
0.107143	0.178571	0.285714	0.25	0.357143	0.357143	0.357143	0.25	0.285714	0.285714	0.357143	0.5	0.285714	0.25	0.357143	0.321429	0.357143	0.178571	0.357143	
0.178571	0.321429	0.285714	0.214286	0.321429	0.285714	0.285714	0.25	0.285714	0.357143	0.25	0.464286	0.464286	0.178571	0.25	0.321429	0.321429	0.357143	0.178571	
0.285714	0.214286	0.285714	0.178571	0.214286	0.285714	0.357143	0.178571	0.392857	0.178571	0.321429	0.392857	0.321429	0.428571	0.357143	0.392857	0.357143	0.5	0.464286	
0.142857	0.25	0.25	0.285714	0.214286	0.392857	0.5	0.25	0.285714	0.464286	0.321429	0.214286	0.357143	0.25	0.25	0.5	0.464286	0.428571	0.392857	
0.214286	0.214286	0.178571	0.214286	0.214286	0.25	0.214286	0.321429	0.285714	0.464286	0.428571	0.464286	0.464286	0.321429	0.321429	0.321429	0.285714	0.25	0.107143	
0.214286	0.107143	0.25	0.25	0.214286	0.285714	0.285714	0.142857	0.25	0.357143	0.285714	0.357143	0.321429	0.285714	0.25	0.428571	0.357143	0.357143	0.392857	
0.178571	0.178571	0.178571	0.25	0.285714	0.178571	0.214286	0.428571	0.321429	0.25	0.285714	0.321429	0.321429	0.392857	0.321429	0.285714	0.25	0.357143	0.25	
0.142857	0.178571	0.178571	0.214286	0.285714	0.214286	0.25	0.214286	0.214286	0.214286	0.178571	0.392857	0.25	0.214286	0.428571	0.25	0.142857	0.392857	0.285714	
0.142857	0.285714	0.25	0.357143	0.25	0.321429	0.321429	0.321429	0.214286	0.357143	0.25	0.357143	0.535714	0.357143	0.142857	0.25	0.25	0.357143	0.607143	
0.107143	0.214286	0.142857	0.321429	0.214286	0.357143	0.321429	0.285714	0.357143	0.428571	0.178571	0.214286	0.392857	0.25	0.25	0.321429	0.285714	0.5	0.285714	
0.178571	0.142857	0.142857	0.285714	0.285714	0.25	0.357143	0.214286	0.5	0.214286	0.392857	0.285714	0.357143	0.428571	0.321429	0.392857	0.428571	0.321429	0.285714	
0.25	0.25	0.25	0.321429	0.285714	0.25	0.25	0.25	0.142857	0.428571	0.178571	0.25	0.535714	0.392857	0.214286	0.178571	0.428571	0.214286	0.285714	
0.071429	0.142857	0.214286	0.178571	0.142857	0.392857	0.214286	0.107143	0.214286	0.178571	0.178571	0.25	0.392857	0.285714	0.285714	0.357143	0.535714	0.357143	0.357143	
0.071429	0.178571	0.178571	0.142857	0.178571	0.178571	0.214286	0.321429	0.142857	0.285714	0.25	0.321429	0.214286	0.285714	0.178571	0.392857	0.428571	0.321429	0.321429	
0.107143	0.107143	0.178571	0.107143	0.178571	0.142857	0.071429	0.285714	0.107143	0.25	0.321429	0.071429	0.321429	0.285714	0.285714	0.25	0.214286	0.107143	0.321429	
0.142857	0.071429	0.107143	0.285714	0.214286	0.178571	0.357143	0.214286	0.178571	0.214286	0.25	0.214286	0.142857	0.178571	0.285714	0.214286	0.178571	0.535714	0.25	
0.107143	0.25	0.214286	0.142857	0.107143	0.142857	0.071429	0.357143	0.142857	0.285714	0.285714	0.428571	0.357143	0.214286	0.428571	0.357143	0.321429	0.107143	0.285714	
0.214286	0.107143	0.178571	0.071429	0.178571	0.214286	0.285714	0.178571	0.142857	0.107143	0.107143	0.178571	0.214286	0.285714	0.178571	0.428571	0.392857	0.285714	0.321429	
0.142857	0.142857	0.214286	0.25	0.142857	0.214286	0.214286	0.142857	0.142857	0.107143	0.25	0.142857	0.214286	0.392857	0.25	0.392857	0.25	0.357143	0.214286	
0.142857	0.142857	0.214286	0.142857	0.178571	0.285714	0.142857	0.392857	0.25	0.285714	0.285714	0.357143	0.214286	0.321429	0.428571	0.25	0.357143	0.214286	0.392857	
0.107143	0.25	0.107143	0.178571	0.178571	0.214286	0.071429	0.321429	0.357143	0.428571	0.25	0.214286	0.357143	0.285714	0.285714	0.285714	0.25	0.285714	0.285714	
0.178571	0.178571	0.035714	0.142857	0.25	0.25	0.321429	0.142857	0.321429	0.142857	0.214286	0.214286	0.25	0.285714	0.25	0.285714	0.285714	0.392857	0.357143	
0.142857	0.178571	0.178571	0.25	0.214286	0.178571	0.321429	0.357143	0.178571	0.285714	0.25	0.214286	0.392857	0.107143	0.285714	0.357143	0.428571	0.357143	0.392857	
0.214286	0.142857	0.178571	0.107143	0.178571	0.25	0.25	0.5	0.321429	0.178571	0.285714	0.321429	0.285714	0.357143	0.214286	0.214286	0.285714	0.142857	0.285714	
0.142857	0.214286	0.142857	0.035714	0.25	0.178571	0.178571	0.321429	0.357143	0.178571	0.357143	0.392857	0.25	0.321429	0.357143	0.357143	0.428571	0.321429	0.142857	
0.178571	0.214286	0.25	0.25	0.071429	0.107143	0.285714	0.178571	0.142857	0.392857	0.321429	0.25	0.392857	0.5	0.285714	0.142857	0.285714	0.321429	0.178571	
0.285714	0.285714	0.142857	0.178571	0.071429	0.178571	0.142857	0.25	0.25	0.214286	0.178571	0.285714	0.178571	0.392857	0.214286	0.285714	0.285714	0.25	0.357143	
0.035714	0.142857	0.214286	0.214286	0.178571	0.25	0.107143	0.142857	0.178571	0.285714	0.142857	0.285714	0.321429	0.25	0.285714	0.321429	0.25	0.321429	0.142857	

Fig. 16. Larger model space - adapted learning rate testing errors

especially in the one hidden unit case, the model is not becoming more complex despite having more parameters being thrown at it. A chain of multiple single nodes is unable to do anything better than a linear separation boundary if its activation is monotone. Consequently, this should not be able to do any better in classifying a training set and the same sort of phenomena holds while the hidden layers are quite small. Generally, as we go down and right we are getting better training losses which is what we expect, but the function is not as smooth as it should be. Optimally, the loss would strictly decrease when any step was made to the right or down. This is clearly not the case in our empirical setting and indicates that our validation dataset will be similarly bumpy. In the testing errors, we see the darkest red spots are at the top left corner (with the worst model) and scattered slightly after the boundary of the 'learning cliff' we can see in the training losses.

Beyond this, there is a mixture of greens and oranges with no immediately obvious patterns. If we recall that this is a continuous function with some sort of noise added upon it, we can start to make out some patterns in this data. There is a green sort of streak coming from the middle left of the data, corresponding to some models which generalized particularly well. We can also see a sort of valley in the top right quadrant near the middle right. It seems this valley could correspond to the first descent of the double descent curve. To see this one has to keep in mind that complexity is not a linear function of layer number and hidden unit size. Consequently, the second descent will not map out a linear curve in this plane and it will be hard to determine exactly what the 'peaks' which correspond to the interpolation maximum will manifest itself as in this plane. Nonetheless, we do start to see the double descent curve on models which are only a few layers deep (from the green section in the bottom left)

IV. Conclusion

Overall, this project turned out to be more about how to tune a deep neural network than it did about discovering the double descent curve. This is slightly disappointing, however, the last results produced are clearly the beginning of uncovering the double descent curve in this deep neural network setting. It is likely that other methodologies to help train deep neural networks could be useful in furthering this study so long as they do not inhibit the network's learning by regularization techniques. An example of an unhelpful technique is penalizing the network's weights. The double descent paradigm intends to achieve zero training loss which isn't feasible with a regularizing term on the size of the network's weights. Other schemes like an adapted learning rate which is able to get even lower scores on the training loss function is an example of a technique which would be useful.

As this project was limited by computational resources, it may also be true that for real applications as well, it becomes too difficult to actually go beyond the point of interpolation. Hence, these models would want to fit themselves snugly within the first descent. Ultimately, the double descent curve is a very promising outlook which seems like it will generalize to deeper network models. Unfortunately, the details are not worked out today by this project, but this should only inspire further work on the fascinating topic.

References

[1] Mikhail Belkin et al. "Reconciling modern machine learning and the bias-variance trade-off". In: *arXiv preprint arXiv:1812.11118* (2018).

[2] Noam Shazeer and Mitchell Stern. "Adafactor: Adaptive Learning Rates with Sublinear Memory Cost". In: *arXiv preprint arXiv:1804.04235* (2018).