Empirically Analyzing Bayesian Neural Networks

Cameron Loewen

5/03/2024

1 Introduction

In general, neural networks (NNs) have been shown to approximate classes of functions arbitrarily well as their parameters are allowed to grow given some constraints to the domain of the data, the width of the layer, and the activation of the layer. This problem is well studied due to NNs not having intuitive parameterization and being somewhat of a 'black box'. Specifically, [1] states that a shallow neural network with infinite width can approximate any continuous function on a constrained domain if the activation function is nonlinear. [2], [3], and [4] explore different aspects of deep networks and the bounds on width necessary to approximate different classes of functions. [3] and [4] focus specifically on ReLU networks and show that it can be quite powerful despite the simplistic activation function.

With Bayesian neural networks (BNNs), the weights are no longer definite and are instead controlled by the parameters of a distribution. This paper focuses only on Gaussian weights and priors, so the parameters are mean and variance. Intuitively, a BNN with variance parameters set to 0 would simply be a NN trained for MLE, and consequently, the theorems regarding NNs would apply to BNNs. To put it more generally, the average output of a BNN has the power to approximate a function.

However, it is not very interesting to say that a model (BNN) can do what another model (NN) with half the number of parameters can do. [7] mentions that BNNs are powerful because they allow NNs to be able to capture uncertainty. The capturing of uncertainty is unarguably a powerful thing for a model to have, and analysis of the model shows that a BNN approximates a gaussian process. [5] shows in 2.1 that a single layered BNN with gaussian priors and a tanh activation (or step) function can approximate a smooth gaussian process (and Brownian and fractional Brownian). The author states that there are computational limitations to this proof, but most have been alleviated since the time of the writing. [6] Shows theoretical and empirical results on deep neural networks with limited width. They found a strong correlation between GP uncertainty and BNN error and that increasing width converged the error of the BNN to the GP. Further, their tests were on the MNIST and CIFAR-10 datasets.

For most cases in practice, the assumptions of the theories do hold and, typically, arbitrarily good models can be trained given enough complexity and data. However, these are existence theorems, and do not give any insight into what loss or algorithm will necessarily find these. For example, some loss functions result in local minima that may greatly hurt convergence and accuracy.

.

2 Methodology/Code

2.1 Code Implementation

The experiments were performed using the tensorflow library keras for optimizing the training and use of a neural network with hardware acceleration. Tensorflow-probability is used for the variational layers necessary for the BNNs weights. The interface designed allowed for the BNNs to take input of univariate functions for regression tasks. The interface also allowed width, depth, approximation function, and training time (epochs) to be varied. The neural networks were trained with RMSE, with a training step size of .05 that slowly degraded to 0 after threshold is reached. I use the sigmoid activation function in all tests

2.2 Metrics

The metrics useful were the difference in the mean between prediction and true, the KL divergence between the empirical Gaussian distributions of the model and the GP, and the variance of the model's output. Since the model gives out instances from a distribution, all of these metrics are gathered empirically by sampling from the trained model. Further, since each point is also a distribution, the difference, divergence, and variance are all averaged metrics over a discrete domain of the function tested. In addition to the averages, the max difference and max KL are also included to give an idea of the bound.

Difference from mean gives an idea of how well the model was able to find the mean function. The variance gives an idea of how spread/uncertain the model is about its predictions (lower is not necessarily better if the data is varied). The KL average gives an idea of how well the model is at capturing the true distribution at every point. Typically, if the variance is similar to the variance of the GP (which is 1), and the mean difference is small, the KL will also be small. However, there are models which are less accurate but have a far better KL, which shows that they are much closer to the actual gaussian process.

2.3 Tests

The main experiment ran 5 different functions, $y = 0$, $y = x$, $y = x^3$, $y = \sin(x)$, and $y = \sin(4\pi x) * \log(x + 1.1)$ on different BNN configurations. The main variances were number of epochs, width, and depth. These tests were ran with 10,000 data points, which is 100 points from 100 sampled functions from the GP.

Another experiment was ran to test the uncertainty quantification by running a subset of the first experiment on a smaller dataset of 100 points, 10 points from 10 sampled functions. This was ran just to be more certain on how the model changed its uncertainty quantification.

2.4 Difficulties

I had difficulties with the library, specifically making the jump from NNs to BNNs. The use of tensorflow-probability required a change of versions with keras which did not utilize my computer very well and caused the GPU to take up huge amounts of memory with relatively small models. Ideally, it would only be 2x as much memory as a NN, and yet, while being able to run neural networks with 10,000 or so parameters (or more), I could only run a BNN with 2000 weights. Unfortunately, due to the complexity of running all the empirical tests, the memory demand was even higher, and my GPU could only handle a model of around 1000 weights before allocation errors occurred. This also made it impractical to test anything but small datasets in low dimensions.

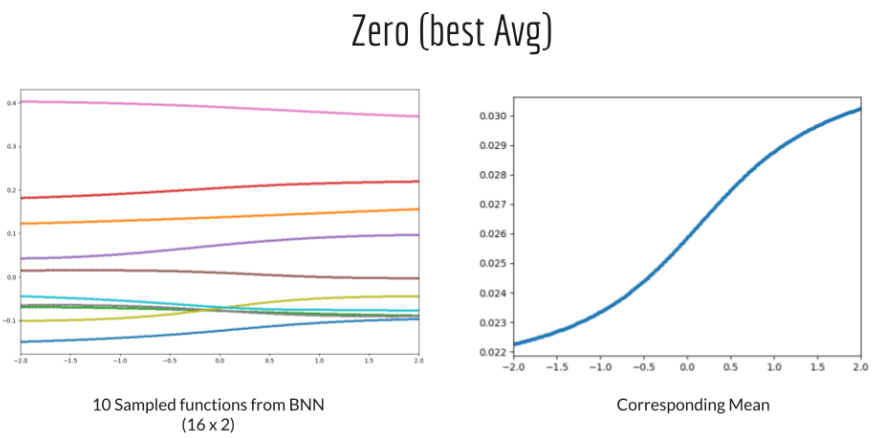
Another issue with the implementation was that the actual loss of the variational layers used to build the BNN in keras was not straightforward. As a result, I was unable to vary that for the experiment, as I likely would have had to change most of the code base. I wanted to vary it to see if changing the loss function helped with what appeared to be the model becoming overly certain. Essentially, with sufficient data and training time, the BNN essentially just became the MLE NN.

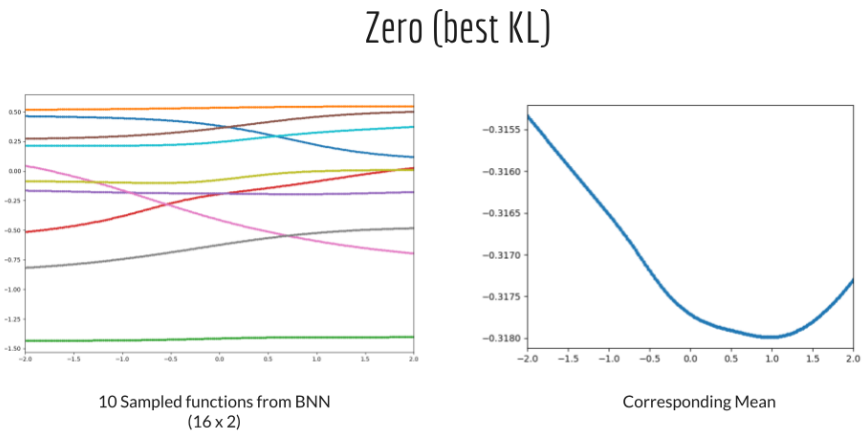
On a positive note, since gaussian distributions have a closed-form KL divergence, I was able to get estimated KL as a metric. Originally, I used bins to discretize the outputs and get an empirical pdf. However, this resulted in KL divergence to be infinite when the bins got too small and samples were lacking. The second approach was to do an empirical AUC using Riemann sums, but the results gathered were all very close and difficult to interpret. The last step, once I knew there was a closed function, was to fit the predicted data to a gaussian (since the prior and posterior is gaussian, this is valid), and then use KL between that and the test data's gaussian.

3 Experiments/Results

As discussed in the test subsection, function, width, depth, and epochs are varied. Further, a second experiment was done to act as a control for understanding data uncertainty in the model. The tables are quite large and rather tedious to get through, so I will discuss the trends prior with visual graphs of some entries in order to get the point of the tables across clearer. The addition of the tables is mostly for completeness and to fill in the gaps of the visuals. They will be put after the conclusion.

My major finding was that most of the models tended to collapse in certainty around the MLE. For the most part, this is acceptable since guessing the (approximate) true mean in regression is the goal. However, this collapse was very bad at approximating the true gaussian process that made the data. Further, if relying on the uncertainty quantification in BNNs, one may think that the data just has low variance when this is not the case. As can be seen in the two figures below, there is a major difference in the two "best" models for the zero mean GP. There is an order of magnitude in difference between distance to the average and the difference in the KL. On one hand, the one with the best average is a better estimator of the mean function (being ten times closer to the average), but the one with the best KL looks far closer to the original variance of the GP (having about 4x the range and spread).





When comparing experiment 1 to experiment 2, one can see that having more data lowered uncertainty. Further, within the experiments, training the model for longer also caused a collapse towards zero variance. While usually the presence of more data would lower uncertainty, when the data itself is uncertain, this should be captured by the model so as to not give overconfidence in the interval. For both problems, I believe this may be the training/optimization. Recall that the theorems only talk about the existence of an approximate gaussian process with a model of sufficient computation, but it does not guarantee that a certain loss or optimization algorithm would find it. For this reason, a good extension of this project would be to try different losses or regularizers to help maintain the uncertainty of the model.

Finally, due to limitations in the model, depth could not be varied substantially to draw conclusions from. There is a trend in growing the width of a shallow network, in that variance does seem to be better maintained. Unfortunately, I was unable to explore this trend to see if expanding width allowed the model to better capture the gaussian process.

4 Conclusion

Overall, the project did give insight into the theory and showed interesting trends that I would like to further explore. Unfortunately, issues with implementation were my biggest use of time and limitation to what experiments I was able to eventually run. The two major trends worth exploring based on my results are the collapse of uncertainty with data/training time despite the variance in the data and how increasing width in a shallow network causes variance to grow (which in this case was good due to the collapse). An interesting question is if this trend is only for the loss I used, and another question would be if the variance will converge or grow unbounded.

5 bibliography

[1] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, Shimon Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, Neural Networks 6 (1992) 861-867.

[2] G. Gripenberg, Approximation by neural networks with a bounded number of nodes at each level, Journal of Approximation Theory 122 (2003) 260-266.

[3] Dmitry Yarotsky, Error bounds for approximations with deep ReLU networks, Neural Networks 94 (2017) 103-114.

[4] Boris Hanin and Mark Sellke, Approximating Continuous Functions by ReLU Nets of Minimal Width, (2018)

[5] Neal, R.M, Priors for Infinite Networks. In: Bayesian Learning for Neural Networks. Lecture Notes in Statistics, vol 118. Springer, New York, NY. (1996)

[6] Lee, Jaehoon et al, Deep Neural Networks as Gaussian Processes. ArXiv abs/1711.00165 (2017)

[7] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra, Weight uncertainty in neural networks. ICML 37 (2015) 1613–1622.

6 Tables

Width	Depth	Epochs	Avg Mean Diff	Avg KL diff
16	2	10	0.130	0.042
4	3	50	0.003	2.187

Table 1: Configurations that achieved min KL and lowest min avg difference in experiment 1: Zero Function

Width	Depth	Epochs	Avg Mean Diff	Avg KL diff
64	1	10	0.202	0.473
8	1	50	0.091	1.895

Table 2: Configurations that achieved min KL and lowest min avg difference in experiment 1: Line Function

Width	Depth	Epochs	Avg Mean Diff	Avg KL diff
1024	1	50	0.46	0.345
4	1	300	0.196	2.945

Table 3: Configurations that achieved min KL and lowest min avg difference in experiment 1: Sine Function

Width	Depth	Epochs	Avg Mean Diff	Avg KL diff
16	2	10	0.265	0.246
8	1	50	0.145	1.068

Table 4: Configurations that achieved min KL and lowest min avg difference in experiment 1: Cube Function

Width	Depth	Epochs	Avg Mean Diff	Avg KL diff
64	1	10	0.452	0.704
8	2	300	0.276	2.370

Table 5: Configurations that achieved min KL and lowest min avg difference in experiment 1: Complex Function

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.107	0.046	0.009	2.029	2.029
4	2	0.086	0.084	0.018	1.559	1.559
4	3	0.077	0.075	0.016	1.585	1.585
4	4	0.042	0.042	0.026	1.353	1.353
4	5	0.042	0.042	0.018	1.529	1.529
8	1	0.112	0.046	0.015	1.8	1.8
8	2	0.168	0.161	0.088	0.772	0.772
8	3	0.024	0.019	0.175	0.47	0.47
16	1	0.152	0.071	0.02	1.541	1.541
16	2	0.133	0.13	0.815	0.081	0.081
64	1	0.101	0.042	0.21	0.473	0.473
256	1	0.614	0.567	0.31	0.427	0.427
1024	1	1.067	1.028	0.57	1.059	1.059

Table 6: Mean function: zero, epochs = 10

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.154	0.072	0.002	2.844	2.844
4	2	0.053	0.036	0.003	2.488	2.488
4	3	0.003	0.003	0.004	2.253	2.253
4	4	0.037	0.037	0.003	2.478	2.478
4	5	0.04	0.04	0.006	2.129	2.129
8	1	0.153	0.07	0.002	2.815	2.815
8	2	0.027	0.017	0.006	2.053	2.053
8	3	0.015	0.012	0.017	1.572	1.572
16	1	0.157	0.073	0.003	2.423	2.423
16	2	0.009	0.009	0.0	3.368	3.368
64	1	0.111	0.054	0.01	1.853	1.853
256	1	0.122	0.053	0.006	2.239	2.239
1024	1	0.341	0.332	0.205	0.62	0.62

Table 7: Mean function: zero, epochs = 50

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.128	0.053	0.0	3.964	3.964
4	2	0.125	0.051	0.001	3.742	3.742
4	3	0.024	0.024	0.0	4.839	4.839
4	4	0.022	0.022	0.0	5.541	5.541
4	5	0.018	0.018	0.0	3.823	3.823
8	1	0.136	0.055	0.0	3.505	3.505
8	2	0.021	0.021	0.0	4.871	4.871
8	3	0.025	0.025	0.0	4.707	4.707
16	1	0.14	0.056	0.0	3.617	3.617
16	2	0.025	0.024	0.0	4.843	4.843
64	1	0.142	0.051	0.001	3.179	3.179
256	1	0.077	0.032	0.004	2.346	2.346
1024	1	0.095	0.042	0.078	1.145	1.145

Table 8: Mean function: zero, epochs = 300

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.941	0.452	0.007	2.197	2.197
4	2	1.1	0.509	0.014	2.191	2.191
4	3	1.047	0.507	0.042	1.661	1.661
4	4	1.028	0.505	0.03	1.834	1.834
4	5	1.052	0.506	0.036	1.72	1.72
8	1	0.66	0.281	0.038	1.429	1.429
8	2	1.198	0.524	0.243	1.017	1.017
8	3	1.275	0.543	0.213	1.163	1.163
16	1	0.573	0.247	0.039	1.358	1.358
16	2	1.018	0.506	0.147	1.134	1.134
64	1	0.515	0.202	0.179	0.586	0.586
256	1	1.253	0.529	0.234	1.204	1.204
1024	1	1.258	0.51	0.589	0.901	0.901

Table 9: Mean function: line, epochs = 10

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.237	0.11	0.008	2.207	2.207
4	2	0.968	0.421	0.018	1.822	1.822
4	3	1.107	0.51	0.002	3.2	3.2
4	4	1.106	0.511	0.004	2.852	2.852
4	5	1.122	0.512	0.0	3.885	3.885
8	1	0.189	0.091	0.008	2.16	2.16
8	2	0.767	0.281	0.045	1.399	1.399
8	3	1.11	0.511	0.003	3.046	3.046
16	1	0.202	0.096	0.007	2.07	2.07
16	2	1.105	0.51	0.003	2.938	2.938
64	1	0.293	0.095	0.017	1.775	1.775
256	1	0.423	0.123	0.051	1.396	1.396
1024	1	1.089	0.448	0.297	1.165	1.165

Table 10: Mean function: line, epochs = 50

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.247	0.192	0.001	3.394	3.394
4	2	0.303	0.195	0.003	2.845	2.845
4	3	1.191	0.522	0.0	4.171	4.171
4	4	1.194	0.524	0.0	4.21	4.21
4	5	1.193	0.523	0.0	4.185	4.185
8	1	0.253	0.198	0.001	3.596	3.596
8	2	0.39	0.224	0.006	2.992	2.992
8	3	0.501	0.204	0.017	2.349	2.349
16	1	0.24	0.19	0.002	3.144	3.144
16	2	0.539	0.239	0.028	1.9	1.9
64	1	0.315	0.204	0.003	2.898	2.898
256	1	0.466	0.235	0.013	2.339	2.339
1024	1	0.587	0.199	0.132	1.153	1.153

Table 11: Mean function: line, epochs = 300

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.882	0.521	0.02	1.99	1.99
4	2	1.049	0.621	0.059	1.57	1.57
4	3	1.114	0.637	0.058	1.679	1.679
4	4	1.016	0.63	0.016	2.197	2.197
4	5	1.103	0.635	0.06	1.648	1.648
8	1	0.83	0.484	0.018	2.06	2.06
8	2	1.072	0.632	0.023	2.066	2.066
8	3	1.11	0.634	0.103	1.485	1.485
16	1	0.822	0.439	0.037	1.777	1.777
16	2	1.072	0.629	0.182	1.117	1.117
64	1	0.816	0.43	0.213	0.737	0.737
256	1	0.858	0.513	0.219	0.836	0.836
1024	1	0.902	0.539	1.123	0.599	0.599

Table 12: Mean function: sine, epochs = 10

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.985	0.27	0.012	2.545	2.545
4	2	0.854	0.454	0.029	1.794	1.794
4	3	1.108	0.635	0.005	2.682	2.682
4	4	1.112	0.635	0.004	2.836	2.836
4	5	1.075	0.633	0.005	2.69	2.69
8	1	0.97	0.292	0.019	2.85	2.85
8	2	0.87	0.495	0.035	1.6	1.6
8	3	1.093	0.634	0.014	2.146	2.146
16	1	1.02	0.296	0.015	2.334	2.334
16	2	1.088	0.633	0.003	2.88	2.88
64	1	0.957	0.27	0.02	2.03	2.03
256	1	1.066	0.336	0.056	2.089	2.089
1024	1	0.838	0.46	0.377	0.482	0.482

Table 13: Mean function: sine, epochs = 50

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.722	0.196	0.001	3.646	3.646
4	2	0.722	0.2	0.002	3.076	3.076
4	3	0.718	0.212	0.003	3.149	3.149
4	4	1.021	0.631	0.0	4.954	4.954
4	5	1.021	0.631	0.0	4.043	4.043
8	1	0.73	0.202	0.001	3.595	3.595
8	2	0.735	0.21	0.002	2.989	2.989
8	3	1.032	0.631	0.0	4.17	4.17
16	1	0.735	0.206	0.0	4.554	4.554
16	2	0.706	0.22	0.022	2.403	2.403
64	1	0.73	0.209	0.001	3.567	3.567
256	1	0.71	0.209	0.008	2.699	2.699
1024	1	0.665	0.254	0.101	1.127	1.127

Table 14: Mean function: sine, epochs = 300

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.853	0.185	0.06	1.258	1.258
4	2	1.053	0.273	0.028	1.88	1.88
4	3	1.025	0.263	0.055	1.616	1.616
4	4	1.008	0.259	0.02	2.146	2.146
4	5	1.008	0.259	0.009	2.539	2.539
8	1	0.66	0.145	0.047	1.226	1.226
8	2	1.028	0.263	0.119	1.247	1.247
8	3	1.063	0.277	0.074	1.453	1.453
16	1	0.704	0.15	0.028	1.49	1.49
16	2	1.029	0.265	0.396	0.889	0.889
64	1	0.824	0.205	0.13	1.086	1.086
256	1	1.329	0.484	0.255	1.537	1.537
1024	1	1.41	0.503	0.66	0.934	0.934

Table 15: Mean function: cube, epochs = 10

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.498	0.154	0.004	2.594	2.594
4	2	0.854	0.188	0.015	2.069	2.069
4	3	1.045	0.27	0.002	3.315	3.315
4	4	1.06	0.275	0.003	3.11	3.11
4	5	1.041	0.268	0.003	3.067	3.067
8	1	0.471	0.162	0.003	2.63	2.63
8	2	0.994	0.244	0.021	2.015	2.015
8	3	1.04	0.268	0.007	2.683	2.683
16	1	0.491	0.162	0.004	2.368	2.368
16	2	1.048	0.271	0.003	3.229	3.229
64	1	0.482	0.154	0.02	1.747	1.747
256	1	0.718	0.189	0.025	1.876	1.876
1024	1	1.1	0.298	0.194	1.27	1.27

Table 16: Mean function: cube, epochs = 50

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.594	0.19	0.001	3.519	3.519
4	2	0.672	0.192	0.003	2.754	2.754
4	3	0.851	0.2	0.006	2.702	2.702
4	4	1.175	0.33	0.0	4.524	4.524
4	5	1.171	0.328	0.001	3.917	3.917
8	1	0.605	0.189	0.001	3.502	3.502
8	2	0.788	0.207	0.005	2.868	2.868
8	3	1.171	0.328	0.0	5.924	5.924
16	1	0.597	0.183	0.001	3.357	3.357
16	2	0.865	0.205	0.014	2.547	2.547
64	1	0.675	0.19	0.003	2.767	2.767
256	1	0.783	0.209	0.017	2.193	2.193
1024	1	1.049	0.378	0.118	1.341	1.341

Table 17: Mean function: cube, epochs = 300

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	1.262	0.383	0.008	2.764	2.764
4	2	1.296	0.39	0.002	3.389	3.389
4	3	1.295	0.39	0.023	2.293	2.293
4	4	1.284	0.394	0.025	2.242	2.242
4	5	1.218	0.42	0.025	2.167	2.167
8	1	1.132	0.401	0.021	2.031	2.031
8	2	1.398	0.362	0.094	1.709	1.709
8	3	1.317	0.383	0.066	1.806	1.806
16	1	1.134	0.408	0.02	2.161	2.161
16	2	1.382	0.367	0.188	1.423	1.423
64	1	1.253	0.452	0.161	1.161	1.161
256	1	1.312	0.529	0.109	1.426	1.426
1024	1	2.756	1.859	1.248	3.248	3.248

Table 18: Mean function: Product of sinx and logx, epochs = 10

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	1.336	0.357	0.002	3.426	3.426
4	2	1.445	0.355	0.003	3.18	3.18
4	3	1.453	0.355	0.005	2.985	2.985
4	4	1.466	0.354	0.002	3.354	3.354
4	5	1.442	0.356	0.001	3.73	3.73
8	1	1.329	0.357	0.001	3.537	3.537
8	2	1.44	0.356	0.003	3.286	3.286
8	3	1.433	0.358	0.011	2.558	2.558
16	1	1.318	0.359	0.005	2.634	2.634
16	2	1.43	0.358	0.001	4.009	4.009
64	1	1.28	0.37	0.009	2.59	2.59
256	1	1.509	0.351	0.008	2.91	2.91
1024	1	1.469	0.352	0.172	1.459	1.459

Table 19: Mean function: Product of sinx and logx, epochs = 50

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	1.373	0.305	0.002	4.089	4.089
4	2	1.353	0.289	0.001	4.569	4.569
4	3	1.323	0.289	0.001	3.892	3.892
4	4	1.418	0.36	0.001	4.06	4.06
4	5	1.411	0.361	0.0	5.559	5.559
8	1	1.243	0.318	0.008	2.878	2.878
8	2	1.253	0.276	0.004	3.304	3.304
8	3	1.414	0.361	0.0	5.237	5.237
16	1	1.317	0.312	0.005	3.012	3.012
16	2	1.402	0.362	0.001	3.839	3.839
64	1	1.322	0.287	0.002	3.483	3.483
256	1	1.137	0.318	0.011	2.415	2.415
1024	1	1.35	0.368	0.079	1.639	1.639

Table 20: Mean function: Product of sinx and logx, epochs = 300

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.396	0.374	0.021	1.785	1.785
4	2	0.385	0.383	0.006	2.357	2.357
4	3	0.35	0.343	0.021	1.469	1.469
4	4	0.504	0.503	0.014	1.718	1.718
4	5	0.286	0.285	0.065	0.969	0.969
8	1	0.522	0.431	0.021	1.564	1.564
8	2	0.683	0.649	0.252	0.52	0.52
8	3	0.581	0.571	0.06	1.043	1.043
16	1	0.436	0.401	0.026	1.877	1.877
16	2	0.337	0.303	0.342	0.194	0.194
64	1	0.15	0.125	0.189	0.404	0.404
256	1	0.126	0.073	0.138	0.606	0.606
1024	1	0.303	0.229	0.498	0.142	0.142

Table 21: Mean function: zero, epochs = 10

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.903	0.465	0.037	1.779	1.779
4	2	0.477	0.442	0.017	1.863	1.863
4	3	0.518	0.505	0.011	1.994	1.994
4	4	0.525	0.523	0.011	1.934	1.934
4	5	0.494	0.493	0.03	1.458	1.458
8	1	0.982	0.516	0.021	1.842	1.842
8	2	0.448	0.442	0.027	1.514	1.514
8	3	0.512	0.51	0.043	1.319	1.319
16	1	1.033	0.573	0.032	2.099	2.099
16	2	0.549	0.542	0.014	1.86	1.86
64	1	0.83	0.479	0.062	1.181	1.181
256	1	0.934	0.729	0.091	1.112	1.112
1024	1	0.56	0.409	0.212	0.669	0.669

Table 22: Mean function: zero, epochs = 50

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.228	0.227	0.009	2.042	2.042
4	2	0.214	0.213	0.002	2.854	2.854
4	3	0.2	0.2	0.0	3.938	3.938
4	4	0.169	0.167	0.004	2.421	2.421
4	5	0.22	0.22	0.008	2.116	2.116
8	1	0.242	0.24	0.006	2.281	2.281
8	2	0.201	0.2	0.001	3.425	3.425
8	3	0.239	0.238	0.001	3.376	3.376
16	1	0.202	0.2	0.005	2.357	2.357
16	2	0.24	0.238	0.001	3.462	3.462
64	1	0.224	0.223	0.005	2.296	2.296
256	1	0.305	0.273	0.026	1.556	1.556
1024	1	0.468	0.396	0.199	1.232	1.232

Table 23: Mean function: zero, epochs = 300

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	1.09	0.491	0.036	2.273	2.273
4	2	1.143	0.555	0.048	2.369	2.369
4	3	1.286	0.589	0.032	2.396	2.396
4	4	1.272	0.588	0.025	2.505	2.505
4	5	1.262	0.585	0.017	2.714	2.714
8	1	0.908	0.405	0.062	1.132	1.132
8	2	1.378	0.612	0.25	1.676	1.676
8	3	1.014	0.547	0.271	2.333	2.333
16	1	1.079	0.479	0.05	1.386	1.386
16	2	1.768	0.829	0.439	2.968	2.968
64	1	0.703	0.325	0.351	0.922	0.922
256	1	0.969	0.486	0.142	1.882	1.882
1024	1	2.195	1.21	0.358	7.235	7.235

Table 24: Mean function: line, epochs = 10

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.583	0.335	0.019	2.301	2.301
4	2	1.297	0.598	0.016	2.553	2.553
4	3	1.174	0.57	0.012	2.541	2.541
4	4	1.303	0.592	0.009	2.916	2.916
4	5	1.303	0.594	0.013	2.707	2.707
8	1	0.629	0.294	0.017	2.396	2.396
8	2	1.309	0.599	0.039	2.206	2.206
8	3	1.256	0.584	0.046	1.974	1.974
16	1	0.518	0.302	0.04	1.871	1.871
16	2	1.288	0.595	0.007	2.872	2.872
64	1	0.598	0.359	0.065	1.497	1.497
256	1	0.861	0.338	0.106	1.211	1.211
1024	1	1.432	0.64	0.448	1.316	1.316

Table 25: Mean function: line, epochs = 50

Width	Depth	Max mean Diff	Avg Mean Diff	avg var	Max KL diff	Avg KL diff
4	1	0.423	0.215	0.012	1.997	1.997
4	2	0.904	0.417	0.036	1.385	1.385
4	3	1.21	0.575	0.001	3.39	3.39
4	4	1.236	0.581	0.0	3.929	3.929
4	5	1.231	0.579	0.004	2.644	2.644
8	1	0.514	0.254	0.024	2.185	2.185
8	2	0.769	0.349	0.076	0.952	0.952
8	3	1.214	0.575	0.001	3.23	3.23
16	1	0.549	0.303	0.027	1.715	1.715
16	2	1.19	0.568	0.002	3.022	3.022
64	1	0.618	0.284	0.042	1.432	1.432
256	1	0.71	0.307	0.051	1.16	1.16
1024	1	0.668	0.267	0.178	0.634	0.634

Table 26: Mean function: line, epochs = 300