AI Coursework – EC201073

Introduction

In this Document, I will be outlining the various tests I did and discuss my findings. The key hyperparameters I will be testing are:

- Batch size
- Learning rate
- Weight decay
- Hidden size
- rDim

To keep results consistent, I will keep my no. of epochs at 75, This will be my constant. I will explore the hype parameters on 2 optimization models:

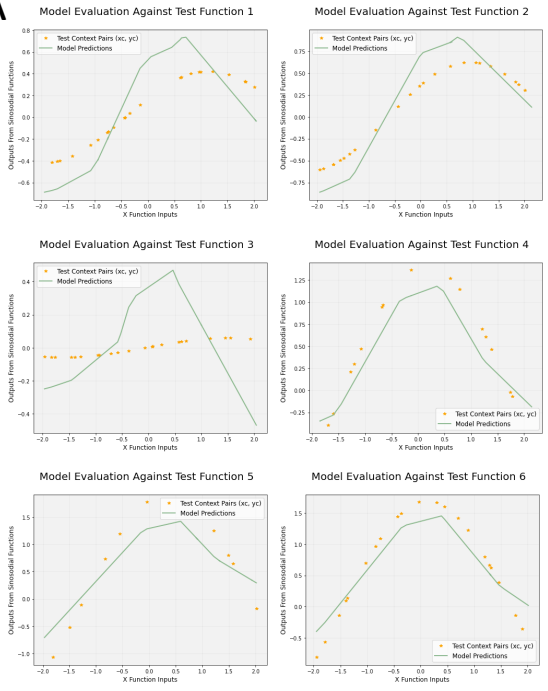
- Adam
- Stochastic Gradient Descent (SGD)

Visualizations based on hyperparameter tuning.

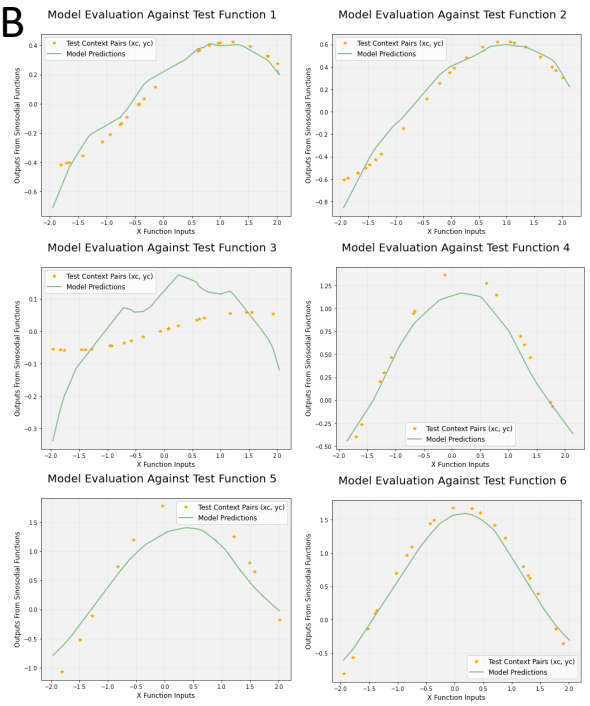
Below is a table of only SOME of the hyperparameter tuning I have performed. Each test will be assigned a letter, which corresponds to a set of graph results below. For each test, I will be using both the SGD and Adam optimizer, only the results from the better performing optimizer will be listed.

Batch Size	Learning Rate	Weight Decay	Hidden Size	rDim	Best Optimizer	Graph Reference
32	0.001	0.0005	8	2	Adam	A
32	0.001	0.005	16	3	Adam	B
32	0.01	0.0005	12	2	Adam	C
32	0.01	0.005	16	4	Adam	D
32	0.1	0.0005	24	5	SGD	E
32	0.1	0.005	24	5	SGD	F
20	0.001	0.00005	30	3	Adam	H

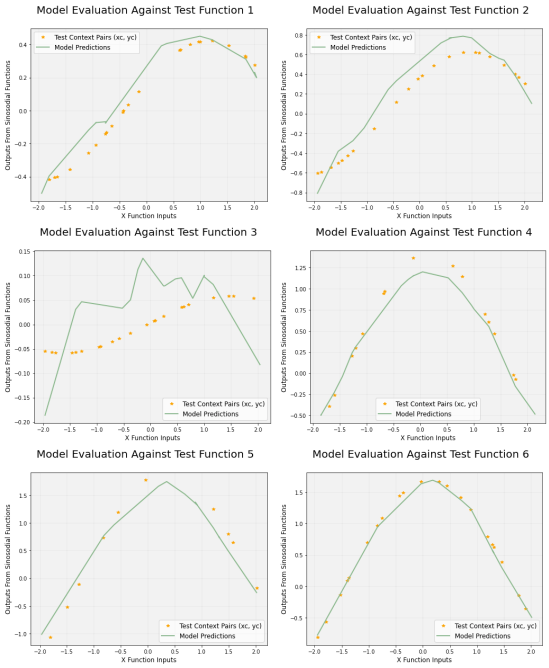
A



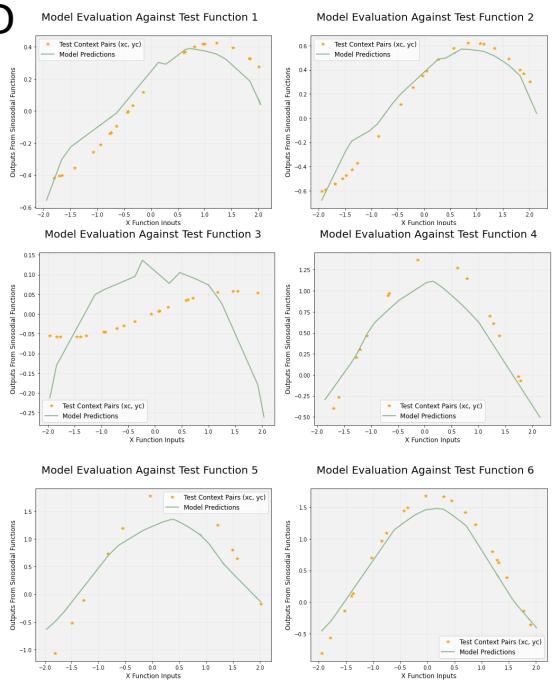
B



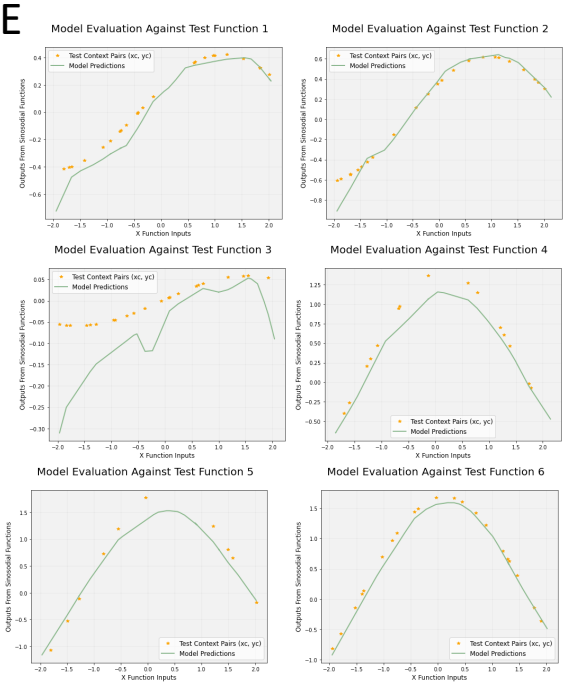
C


Model Evaluation Against Test Function 1
Model Evaluation Against Test Function 2
Model Evaluation Against Test Function 3
Model Evaluation Against Test Function 4
Model Evaluation Against Test Function 5
Model Evaluation Against Test Function 6
Test Context Pairs (xc, yc)
Model Predictions
Outputs From Sinosodial Functions
X Function Inputs

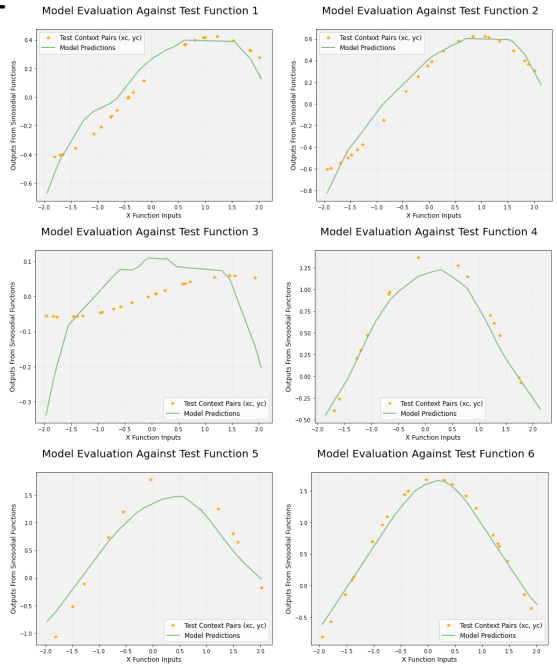
D


Model Evaluation Against Test Function 1
Model Evaluation Against Test Function 2
Model Evaluation Against Test Function 3
Model Evaluation Against Test Function 4
Model Evaluation Against Test Function 5
Model Evaluation Against Test Function 6
Test Context Pairs (xc, yc)
Model Predictions
Outputs From Sinosodial Functions
X Function Inputs

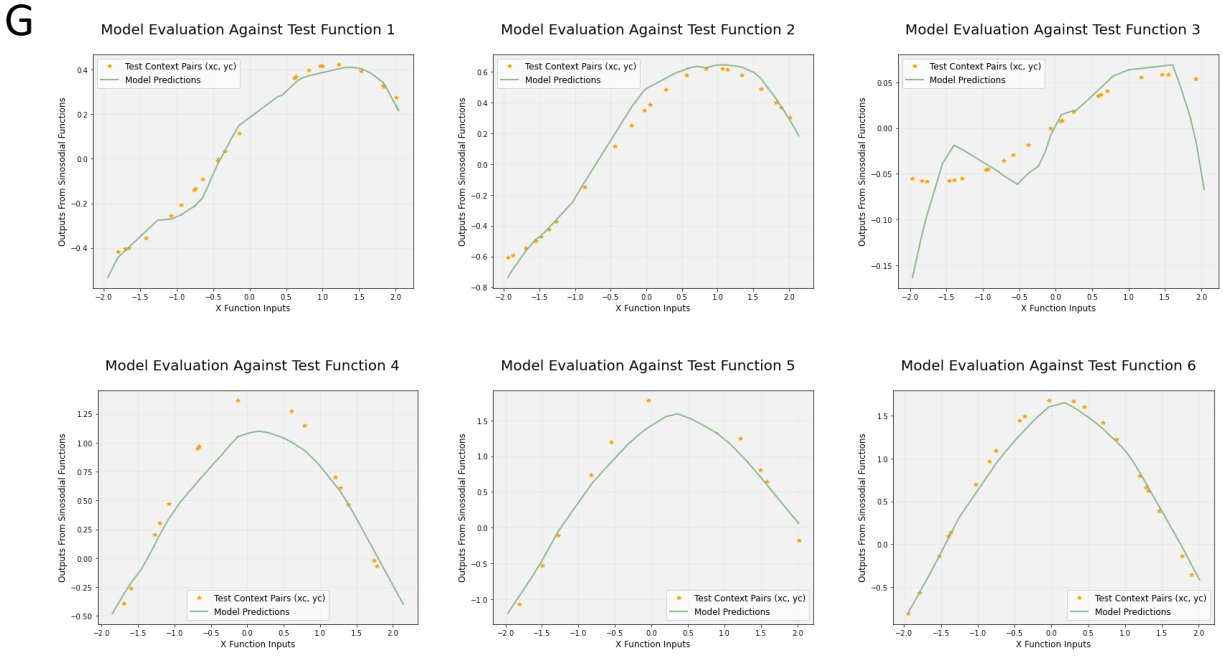
E


Model Evaluation Against Test Function 1
Model Evaluation Against Test Function 2
Model Evaluation Against Test Function 3
Model Evaluation Against Test Function 4
Model Evaluation Against Test Function 5
Model Evaluation Against Test Function 6
Test Context Pairs (xc, yc)
Model Predictions
Outputs From Sinosodial Functions
X Function Inputs

F


Model Evaluation Against Test Function 1
Model Evaluation Against Test Function 2
Model Evaluation Against Test Function 3
Model Evaluation Against Test Function 4
Model Evaluation Against Test Function 5
Model Evaluation Against Test Function 6
Test Context Pairs (xc, yc)
Model Predictions
Outputs From Sinosodial Functions
X Function Inputs

G


Model Evaluation Against Test Function 1
Model Evaluation Against Test Function 2
Model Evaluation Against Test Function 3
Model Evaluation Against Test Function 4
Model Evaluation Against Test Function 5
Model Evaluation Against Test Function 6
Test Context Pairs (xc, yc)
Model Predictions
Outputs From Sinosodial Functions
X Function Inputs

Findings

Changing the Batch Size

- A smaller batch size generally does improve the performance of the model, however not as significant as I thought.
- I think during training, it mainly makes the loss more stable.
- I find it doesn't go up and down as much when working with a smaller batch size.
- A smaller batch seems to make the model overfit to particular batches and does not generalize well, This is observed after using the pickle test data provided. A balance between 15-35 seems to be best.

Changing the Learning Rate

- Learning rate depends on the optimizer.
- I found that a high learning rate (bigger number) is more suitable for SGD as it doesn't seem as adaptable as Adam.
- A Smaller Learning rate (smaller number) is more suitable for Adam as it seems to be a lot more sensitive.
- If Learning Rate is high when using Adam optimizer, the model seems to escape local minima quite a lot and results are inconsistent.
- A good learning rate for SGD would be 0.1 – 0.01
- A good learning rate for Adam would be 0.001 – 0.0001

Changing the Weight Decay

- Weight decay helps improve results slightly when approaching the minimum loss. However, if weight decay is high, the model does not seem to learn from the functions at all.
- A good weight decay is either 0.0001 or 0.0005

Changing the Hidden Size

- A high hidden size seems to work well, between 25 and 40.
- Having a high hidden size requires a low rDim, if rDim is similar to the Hidden Size or anything above 6 the model stops working well.

Changing the rDim

- rDim is good when it is relatively low, I found 2-4 rDim size to work well, anything beyond that and the model stops learning the underlying pattern in the functions.
- rDim makes loss decrease more consistent. The model does not jump as much with a low rDim.

Performance on Optimizers

- Generally, I found Adam to perform much better than SGD. What I have found through the various tests I have done is that:
- Adam, as an optimizer does not generalize as well as SGD, but is highly adaptable.
- Adam seems to learn from individual functions a lot better without requiring a high learning rate.
- Adam is a lot more sensitive. It does not require a high learning rate to work. In fact, I found that increasing the learning rate lead the Adam optimizer to not learning from the model at all.
- I would think SGD would be better when larger datasets are provided, and the model has to give a general picture rather than fitting to every function, whereas Adam can be a lot more particular.
- As Learning Rate and hidden size is increased, SGD becomes better than Adam.
- When using SGD, I found you obtain better results when having a high Hidden size with a relatively low rDim. SGD is does not learn quickly, so increasing the Learning rate as well as Weight decay does help drastically improve the results.

Best Parameters I found:

Optimizer = Adam, Batch Size= 2, Learning Rate = 0.001, Weight Decay = 0.00005, rDim = 3, Hidden Size = 30

Best model can be found in the .ipnyb file