CS7020-Advances in Theory of Deep Learning

Analysis of Simplicity Bias

Name: Shivaram
Roll.No: BE20B032

1) Introduction:

Simplicity bias in neural networks refers to the tendency of these models to favor simpler explanations or representations of data, sometimes at the cost of capturing the full complexity and intricacies of the underlying patterns.

"Pitfalls of Simplicity Bias" paper discusses when and where neural networks exhibit simplicity bias using synthetic dataset where the neural network weighs more on the simple feature to make the classification than the other features.

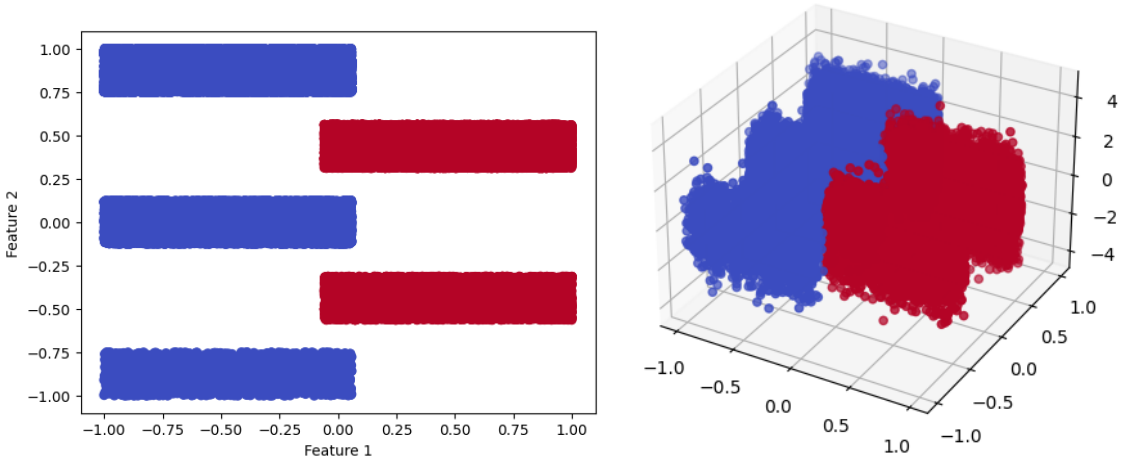
In this project we would be extending some of the experiments on simplicity bias using the same dataset(*"LSN"*).

2) Background:

As simplicity bias is observed when the neural network can classify / perform well on the data using simple features and weighing more on them. "*Pitfalls of Simplicity Bias* " paper have conducted experiments on their synthetic dataset called "LSN".

Dataset:

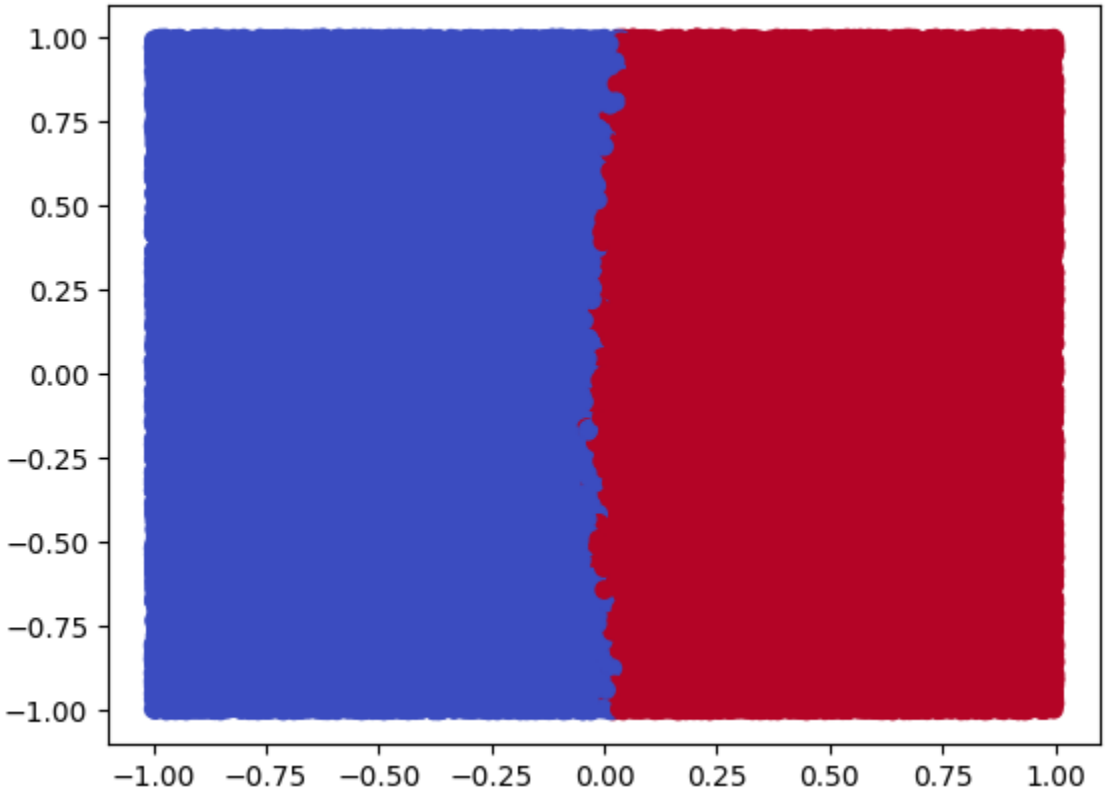
"LSN" dataset consists of 2 features out of which one being complex and the other is simple and linearly separable. Further the other dimensions consists of just Guassian noise as shown in the below picture.



As we can see, we have our data split into 5 slabs as shown in the figure.

Experiments conducted in the paper:

- Training a neural network which has hidden neurons less than the dimensions of the data and show the decision boundary.



Here is the replication of the experiment, where we have trained a neural network with one hidden layer consisting of 32 neurons on a dataset(as described in the above images) of dimension 50.

As expected we obtained the decision boundary, where the neural network relies more on feature 1 which is simple and linearly separable than the complex feature.

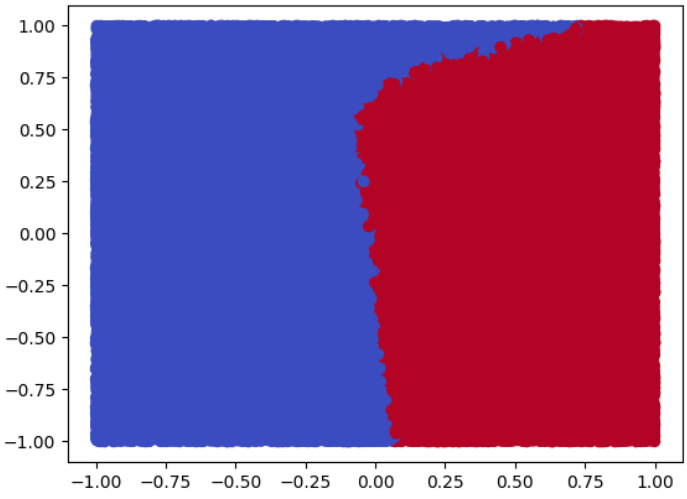
As further we conduct experiments where the SB(simplicity bias) occurs by varying the training conditions and network architecture.

3) Experiments:

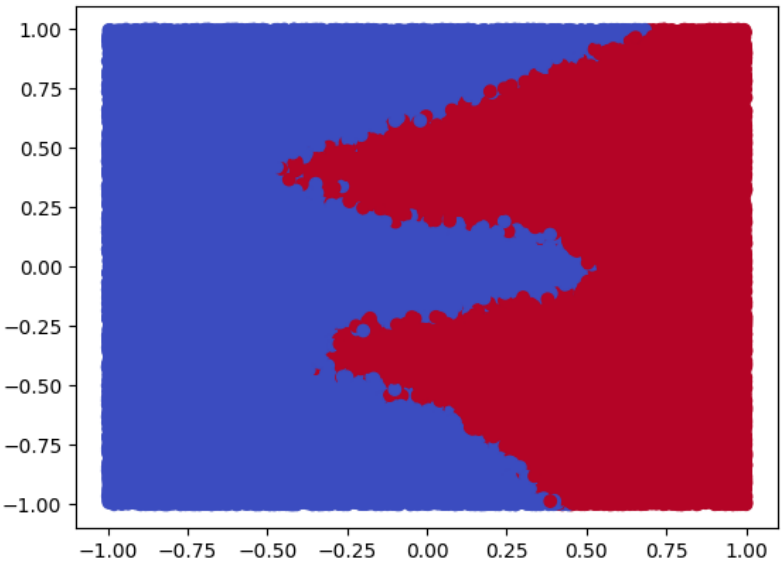
a) Depth of the network:

Conducted experiments by training neural networks with different architectures such that they vary in depth and visualized their decision boundaries. Here are the results:

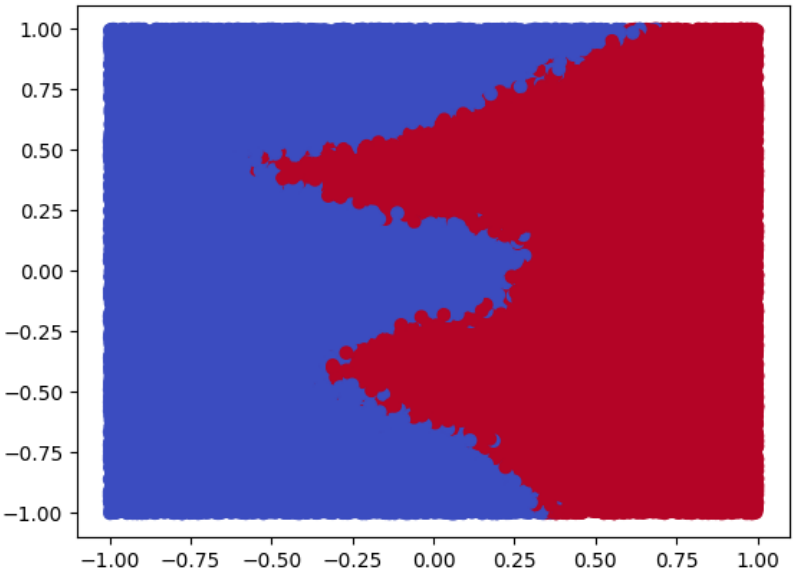
- Network with Architecture: 32



- Network with Architecture: 32 32 16 16



- Network with Architecture: 32 32 32 32 16 16

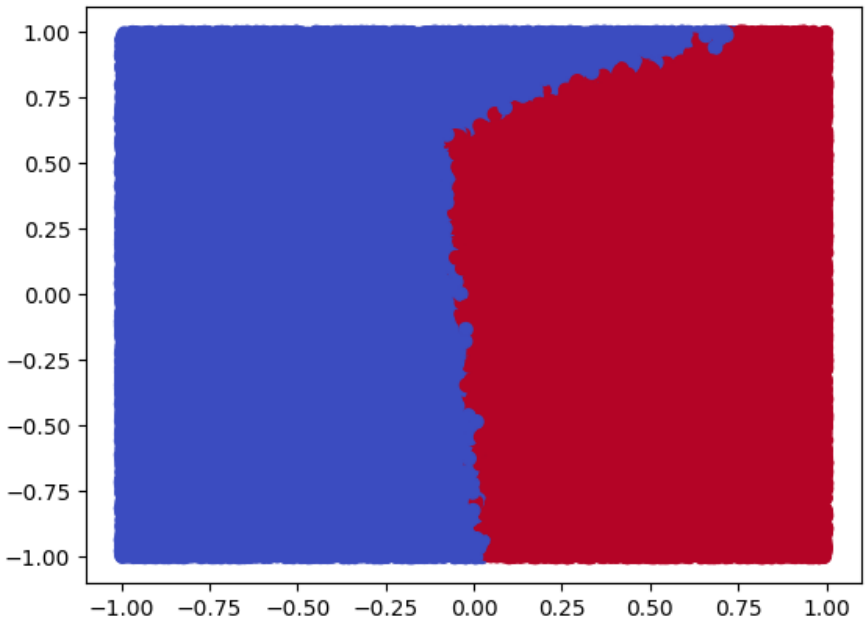


From the above decision boundaries of architectures it is clear that the network starts to focus on the complex feature for classifying. This shows that complex features become simpler with deeper networks.

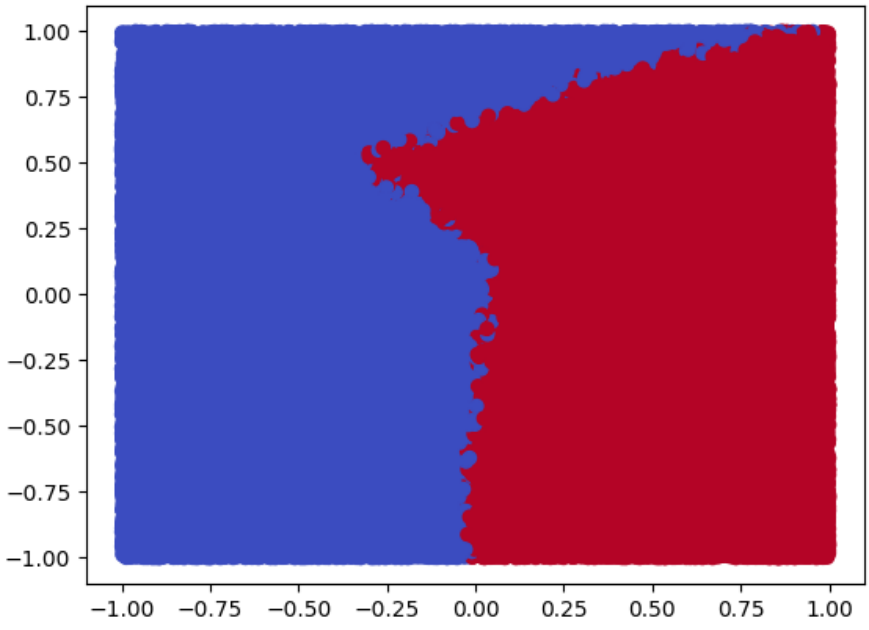
b) Number of Epochs:

In this experiment we test out how the neural network performs if it is trained for a longer period of time and here are the results:

- Epochs: 20



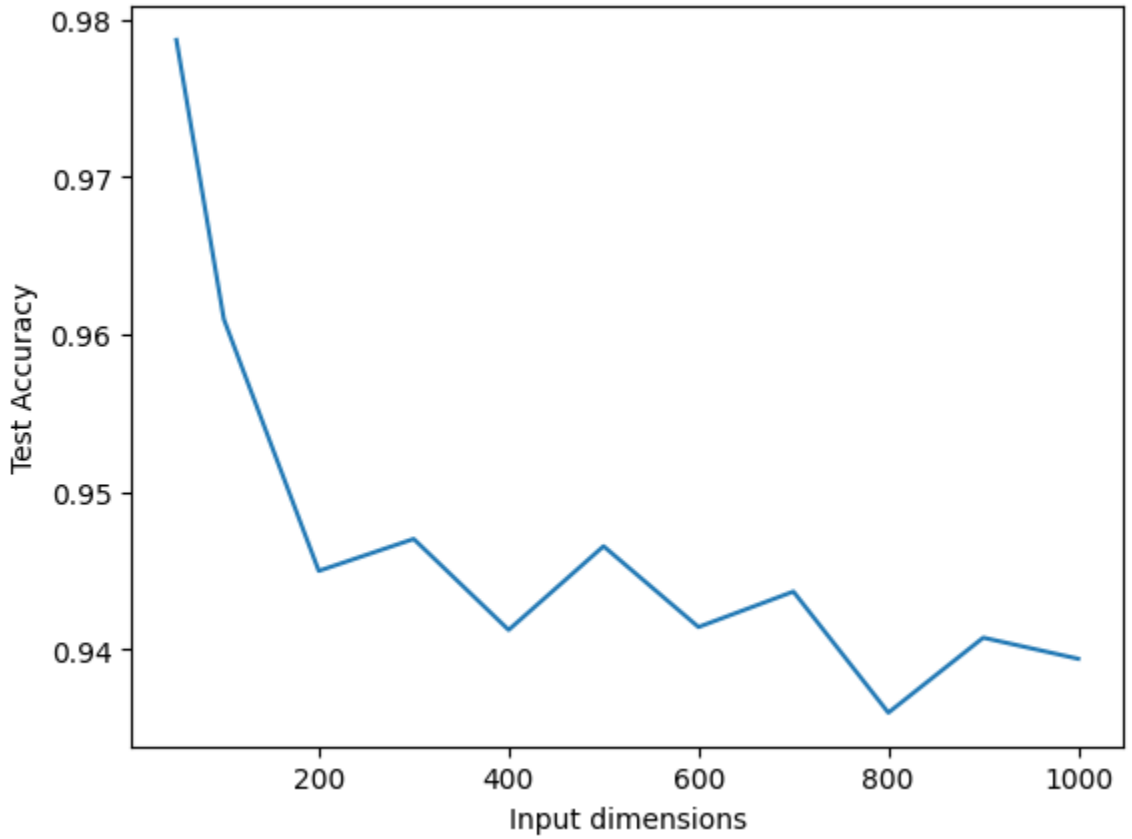
- Epochs: 50



From these two decision boundaries, it is visible that, by training the network for longer would get the network to cover the complex features as well.

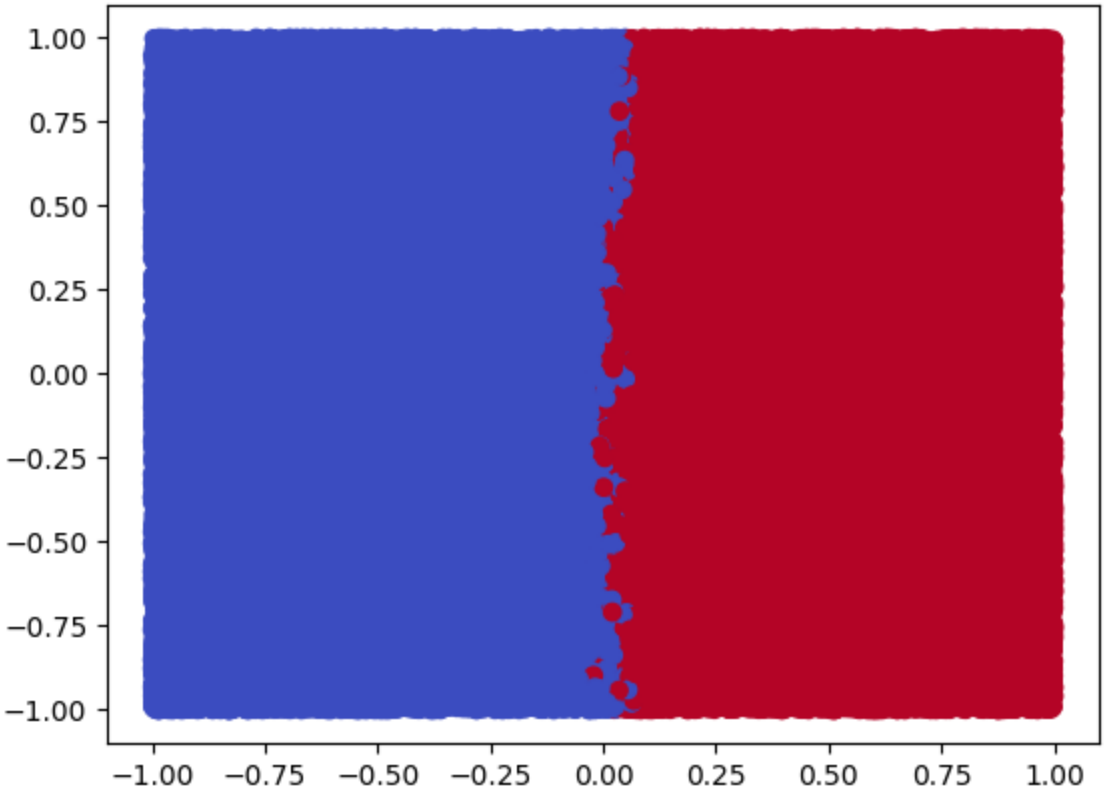
c) Dimension of the Data:

The number of input dimensions to the network also influences Simplicity bias. From the below plot of the accuracy to input dimension we could see that the accuracy decreases with the increase in input dimension. From this we could infer that simplicity bias arises in higher input dimensions and network weights the simple feature more.



d) Ensemble methods:

Ensemble methods, have multiple classifiers and voting of them is taken to decide on the final output. Ensemble methods can be used to avoid simplicity bias by training multiple networks on the same data. Here is the decision boundary obtained by ensembling 100 neural networks made up of 1 hidden layer with 32 neurons.

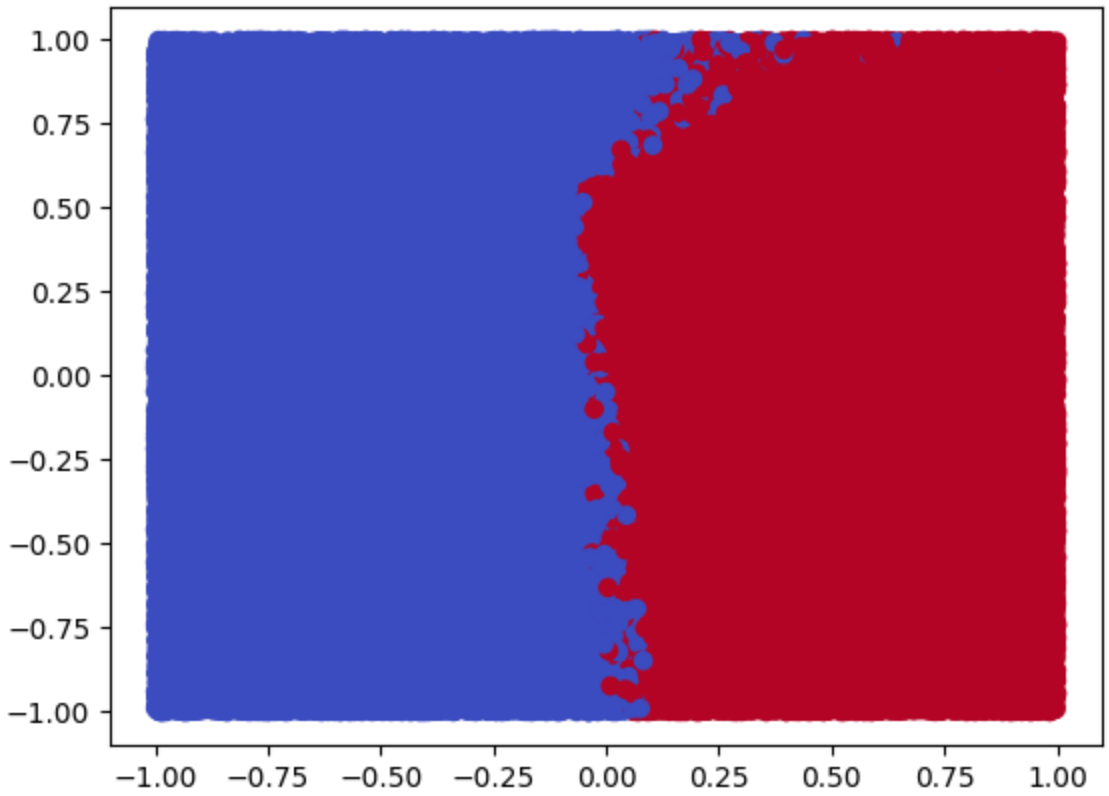


From the above decision boundary, its clear that ensemble method on the dataset didn't overcome the Simplicity bias as expected.Here all the neural networks are trained on the same features.

e) Dropout layer

The dropout layer is a regularization technique commonly used in neural networks. It is designed to prevent overfitting and improve the generalization ability of the model. During training, the dropout layer randomly sets a fraction of the input units to zero at each update, effectively "dropping out" those units. This random dropout forces the network to learn more robust and independent features, as it cannot rely too heavily on any single input unit.

By using Dropout layer after the hidden layer, we trained the neural network to make it learn features individually.



From the above decision boundary, we infer that addition of dropout layer didn't help to overcome the simplicity bias

Link to Colab Notebook: https://colab.research.google.com/drive/1QJtEFEXJGMsTYraLtzPSiaF184iz3lqB?usp=sharing

4) References:

a) Paper: https://arxiv.org/pdf/2006.07710.pdf
b) Explanation: https://youtu.be/xqpNv-0FqZg
c) Critic: https://www.youtube.com/watch?v=h0uA0y1DRWI
d) Github: https://github.com/harshays/simplicitybiaspitfalls