Details of the machine learning code

Data Cleaning:

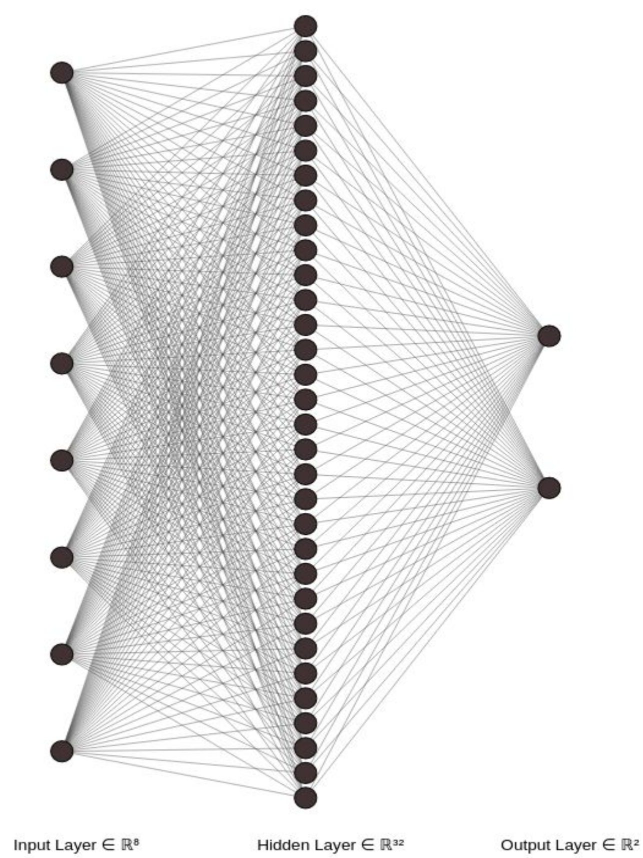
- For certain values of g1, g2, s1 and s2 any electromagnetic radiation having frequencies in the range 0 to 20 GHz gets absorbed. These dimensions are useless for the purpose of training hence need to be removed.
- For some other values there might be only one cutoff frequency in the range 0 to 20 GHz; these dimensions also cannot be used to train the model and hence need to be removed.
- For establishing the relationship between the cutoff frequencies and dimensions of FSS structures, isolation of these cutoffs is required because the remaining data plays no significant role in defining this relationship.
- Since one dimension can produce multiple pairs of cutoff frequencies, we can also choose to include other pairs . This requires further cleaning because for subsequent cutoffs data tends to have sharp peaks and for some of them lower and higher cutoff frequencies will be almost equal making the pair useless.

Data Scaling:

- All the inputs and outputs are bought between the range of -1 and 1 by using the formula $X' = X-\mu / (X_{max}-X_{min})$. Data scaling plays an important role in machine learning as unscaled data can lead to **exploding gradients** causing the learning process to fail. Also, it is necessary if we want to use the tanh activation function in the output layer because tanh has a range between -1 and 1.

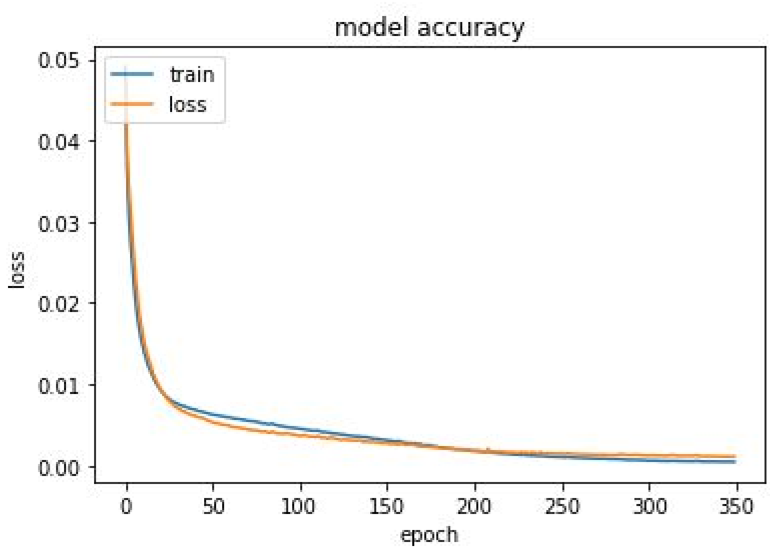
Feeding Data into DNN:

- Here we are using DNN to capture the relationship between FSS dimensions and cutoff frequencies so we have to map 4 inputs to 2 outputs. Structure of DNN is as follows:
    - Input layer has 8 neurons each accepting 4 inputs values i.e dimensions
    - A hidden layer having 32 neurons each accepting inputs from 8 neurons
    - An output layer having 2 neurons each accepting inputs from 32 neurons



- Input layer and hidden layer uses **Rectified Linear activation** function which because of its infinite i.e between 0 and infinity helps avoid **easy saturation**.
- To capture **non linearities** in the relationship while avoiding saturation of the neurons the output layer uses **Tanh** activation function.

- The number of neurons in each layer is in powers of 2 because that helps the GPU to take advantage of optimizations related to efficiencies in working with powers of two.
- Number of hidden layers and the number of neurons per layer are taken such that the most optimum accuracy in the final solution can be achieved.
- The data is divided into training, validation and test data in the ratio **3:1:1**. Such a ratio is chosen so that a large amount of data is available for training the DNN but enough still remains to evaluate the accuracy of the model.
- **Adam** optimizer is used for this model because it is the most suitable one for obtaining convergence with complex data.
- Around **350** epochs are used to train the model because any number more than that would lead to overfitting.



Generating Lookup Table:

- The reason for making a lookup table is two fold:
    - The relation from cutoff to dimension is hard to capture because it is highly complex
    - The Lookup table will be very small even if it includes all the possible combinations of dimensions.

- Around 40k combinations of dimensions are generated and fed in a neural network so that they can be mapped with cutoff frequencies and stored in the lookup table.
- This lookup table is used to find an entry having minimum average absolute error for the frequencies provided by the user. One important benefit of using such a technique is that multiple entries can have same errors which means **multiple combinations of dimensions** can give the same result hence giving the user the freedom to select the best suited one.