K Shravan kumar	12544520 (skk7n5@mst.edu)
V Meghana Reddy	12577075 (mvgby@mst.edu)

Gradient Descent Convergence

1. We made changes in get_dataset_args() to obtain a dataset with 500 samples and each sample is normalized to have unit length.
   - Modified MLPBase class and created a fully connected neural network with a single hidden layer (1000 hidden neurons) with relu activation and standard normal initialization.

   $$\textit{initialize } \mathbf{w}_r \sim N(\mathbf{0}, \mathbf{I}),\ a_r \sim \text{unif}\left[\{-1, 1\}\right] \textit{ for } r \in [m],$$

   - No of parameters = 51010
   - width of the hidden layer > number of input instances

2. Constructed H_infinity and H_0 matrices using input samples and initial weights of neural network. And calculated the minimum eigenvalues of both matrices and they are positive.

   H_0:

   $$\mathbf{H}_{ij}(t) = \frac{1}{m}\mathbf{x}_i^\top \mathbf{x}_j \sum^{m} \mathbb{I}\left\{\mathbf{x}_i^\top \mathbf{w}_r(t) \geq 0, \mathbf{x}_j^\top \mathbf{w}_r(t) \geq 0\right\}$$

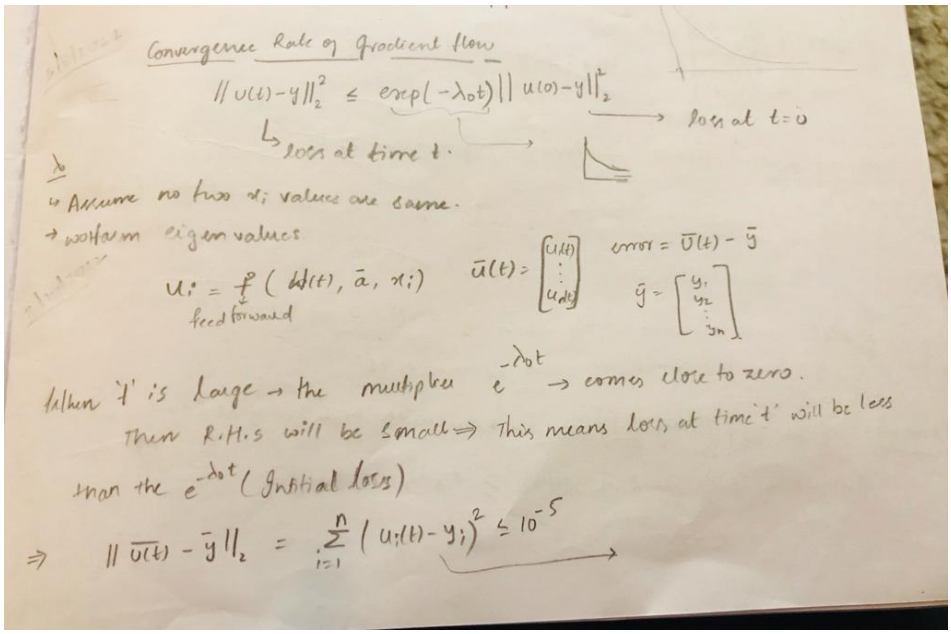
   H_Infinity:

   $$\begin{aligned}
   &\mathbf{H}_{i,j}^{\infty} \\
   &= \mathbb{E}_{\mathbf{w}\sim N(\mathbf{0},I)}\left[\mathbf{x}_i^T\mathbf{x}_j \mathbb{I}\{\mathbf{w}^T\mathbf{x}_i \geq 0, \mathbf{w}^T\mathbf{x}_j \geq 0\}\right] \\
   &= \mathbf{x}_i^T\mathbf{x}_j \mathbb{E}_{\mathbf{w}\sim N(\mathbf{0},I)}\left[\mathbb{I}\{\mathbf{w}^T\mathbf{x}_i \geq 0, \mathbf{w}^T\mathbf{x}_j \geq 0\}\right] \\
   &= \mathbf{x}_i^T\mathbf{x}_j \frac{\pi - \text{angle}(\mathbf{x}_i, \mathbf{x}_j)}{2\pi}.
   \end{aligned}$$

   H_Infinity = 0.10984
   H_0=0.04648147

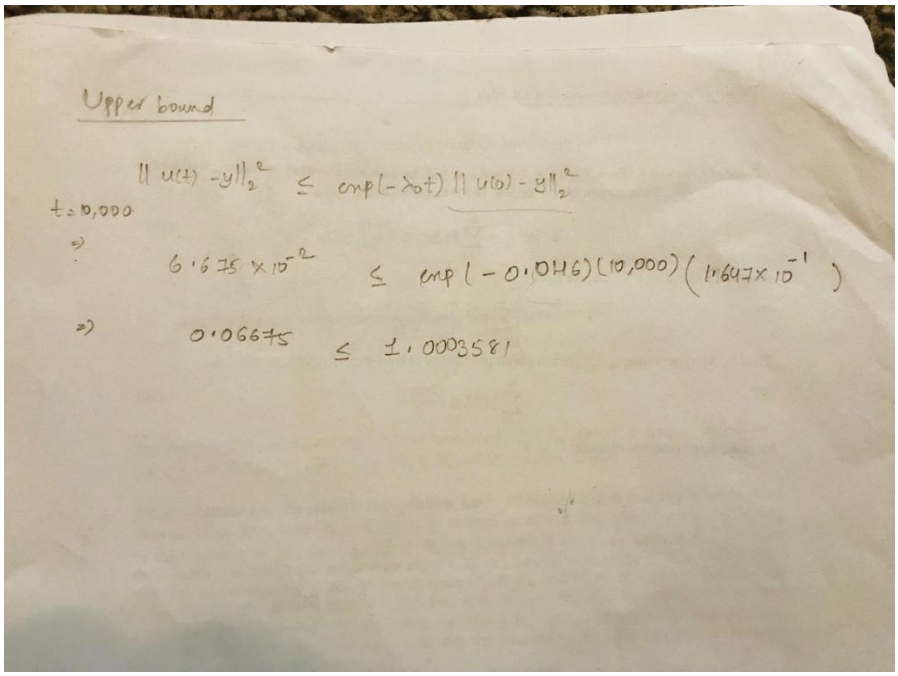
   H_t ~ H_0 ~ H_infinity → Statistical average over random initialization

3. Train_model function is modified to use mean squared error loss function and SGD optimizer.
   - We used batch size of 500 samples (took entire dataset for training)
   - Used one hot encoding to get the shape of (500,10) for target (actual output) as we are using MSE loss function
   - Training loss values are plotted and they are exponentially decaying



-Comparison with theoretical upper bound:

$$\|\mathbf{u}(t) - \mathbf{y}\|_2^2 \leq \exp(-\lambda_0 t)\, \|\mathbf{u}(0) - \mathbf{y}\|_2^2 .$$



4. Initially we took all 50 samples of digit ‘0’ images and trained the model for 1000 steps. And we appended all 50 samples of digit ‘1’ imaged and trained the model for 2000 steps and repeated this process till digit ‘9’ images in a stage wise manner.
   -And compared these stagewise training loss values with previous training loss values and which turned out to be equal with training accuracy of 96.6%