Task 1

Synthetic Data Generation

- Mythical plants
  - Name (Str)
  - Height
    - Min (Int)
    - Max (Int)
  - Colours (Array)
  - Shape (Str)
  - Flowering (Bool)
  - Climate (Str)

- Plants are defined in plants.json
- Must adhere to the schema in plantSchema.json
- Allows for any number of new plants to be defined
- Additional features could be added to the schema
- Easily expandable and modular

Code reads in schema and plant list and generates a user defined number of plants for each type.

Command line arguments allow for the user to specify how many examples for the train test and validation sets, how many decimal places are wanted in the generated heights, and if they want the data to be pre-processed ready for machine learning.

Models

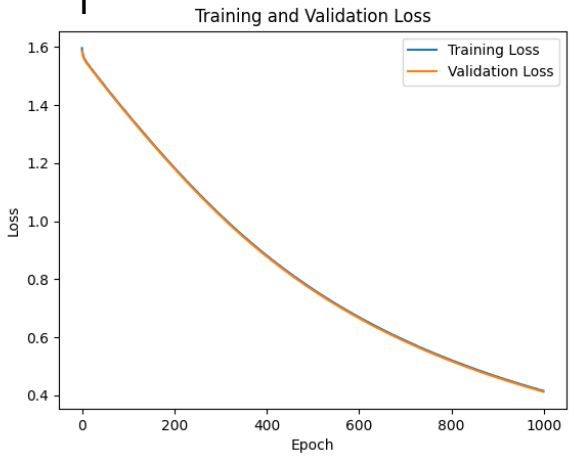
- Generic Neural Network class in Pytorch
- Allows for full customisation of :
  - Layer Size
  - Activation Function
  - Criterion Function
  - Optimiser
  - Name for model to be saved under
- Outputs trained model for later use
- Experiment:
  - How Hidden Size and Activation affects accuracy

Model	Hidden Size	Activation	Criterion	Optimiser
1	100	Sigmoid	Cross Entropy Loss	SGD
2	10	Sigmoid	Cross Entropy Loss	SGD
3	100	ReLU	Cross Entropy Loss	SGD
4	10	ReLU	Cross Entropy Loss	SGD

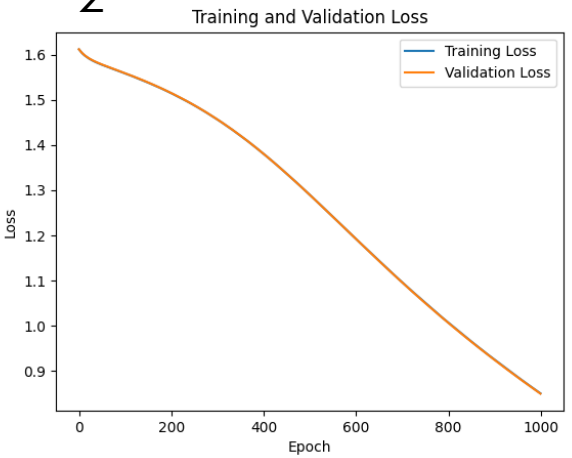
Training

- 1
  - Hidden : 100
  - Activation : Sigmoid
  - 100% on test set
- 2
  - Hidden : 10
  - Activation : Sigmoid
  - 89.25% on test set
- 3
  - Hidden : 100
  - Activation : ReLU
  - 100% on test set
- 4
  - Hidden : 10
  - Activation : ReLU
  - 100% on test set

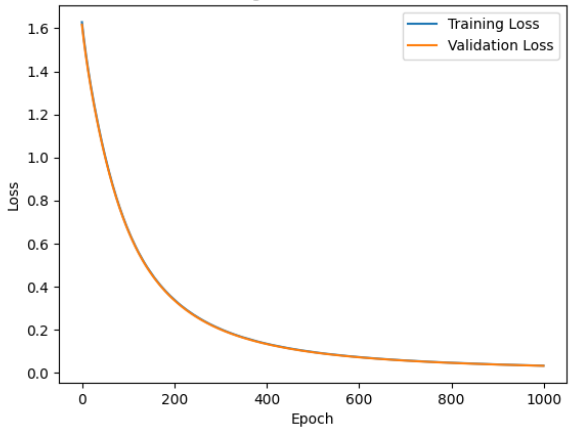
1



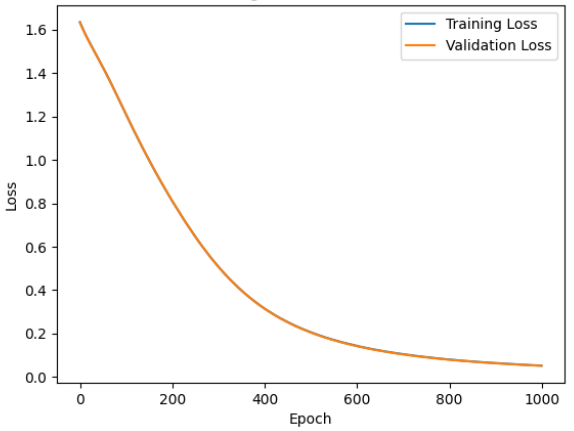
2



3



4



Conclusion

- Sigmoid based models fell short
  - One having 89.25% accuracy on the test set
- Having 100 Hidden Layers is better
  - Models converged faster with the same learning rate
- Datasets too easy to predict
  - Train and Validation Loss was too similar
  - 100% on test set from 3 of the models
  - Given customisability of generator, numbers could be tweaked to make the flowers harder to separate