Neural Network Estimation of ALM

Zeke Abshire, Miles Bellaire, Ryan Field, Kohl Morris, Han Nguyen, Thao Nguyen

Introduction

Appendicular Lean Mass (ALM) is the amount of lean tissue (muscle) present in the arms and legs. Our project utilized data collected from the DXA and Size Stream SS20 scans from Pennington Biomedical Research Center's Metabolism and Body Composition Laboratory.

The DXA scan provides valuable body composition measurements, however it is an expensive and time-consuming process. In order to maintain the high accuracy of the DXA scan while using the cost-effective SS20 scan, we designed a neural network to predict ALM by training on 41 parameters (such as limb lengths, surface areas, and volumes) measured from 770 subjects of the SS20 scan. We evaluated the accuracy of our model based on the maximum, mean and standard deviation of the differences between predicted and actual ALM.

Method

Filling Missing Values:

The raw SS20 dataset had 584 missing values out of approximately 32,000 total values due to inconsistencies in the scans. Three methods were used to fill missing values:

1. Left-Right Method used right half of body measurements to fill left half body measurements, and vice versa.

2. Sum Method took into account the body's natural bilateral symmetry to calculate any missing values within the 12 parameters (left/right arm, left/right leg, torso, and total) that pertained to a subject's body surface area and volume.

$$Total\ SA = left\ arm_{SA} + right\ arm_{SA} + left\ leg_{SA} + right\ leg_{SA} + torso_{SA}$$

$$Total\ Volume = left\ arm_v + right\ arm_v + left\ leg_v + right\ leg_v + torso_v$$

3. Multivariate Imputation by Chained Equation (MICE) from the Scikit Learn Python package was used to impute the last remaining 383 missing values not filled by the previous 2 methods.

Standardize Inputs:

The filled SS20 dataset was standardized, using the Scikit Learn standard scalar function, to reduce the neural network training time for the training dataset to pass through our algorithms.

Method Flowchart
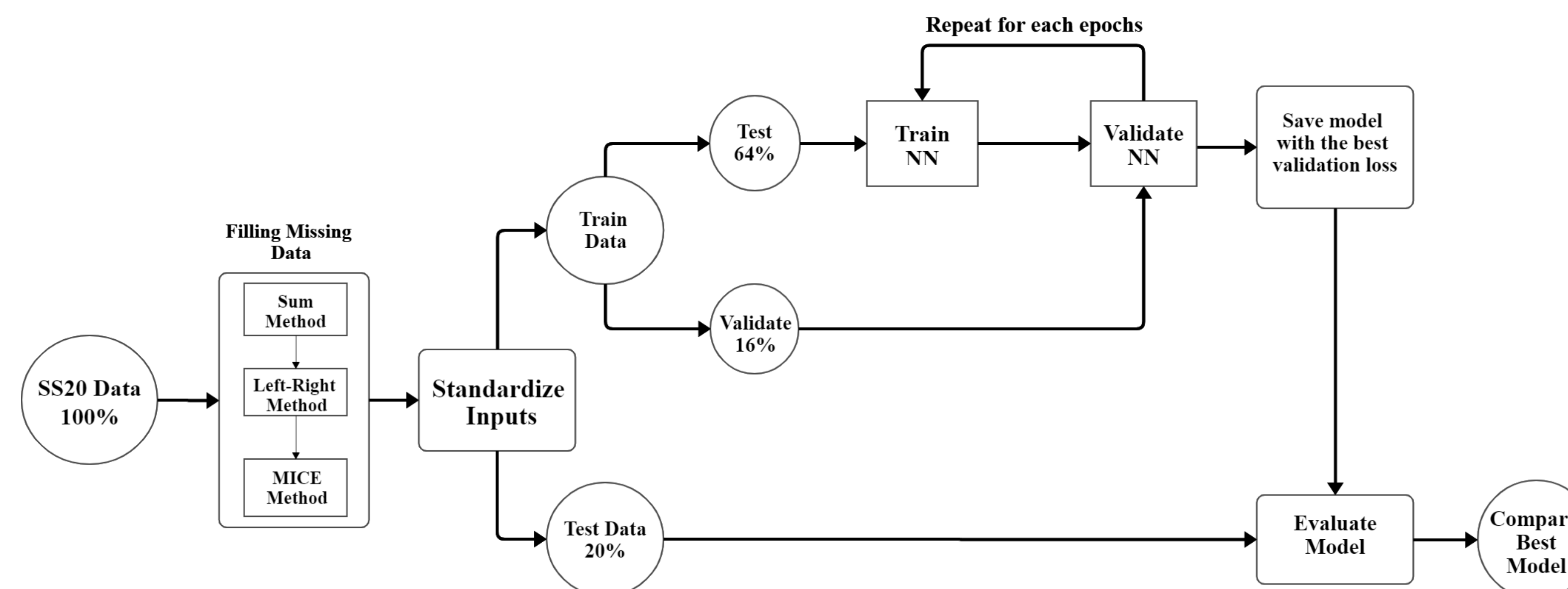
Determine the best model

The complete and standardized SS20 datasets were split into training data and test data. The training data was passed through a neural network for 500 epochs. After each epoch, the validation loss was computed to measure the performance of the neural network. The epoch with the lowest validation loss indicated the smallest mean square error. This model was saved for further statistical calculation as further training would have overfit the model.

Our group repeated this process 50,000 times to determine the best model based on the mean, standard deviation, and maximum differences between predicted and actual ALM.
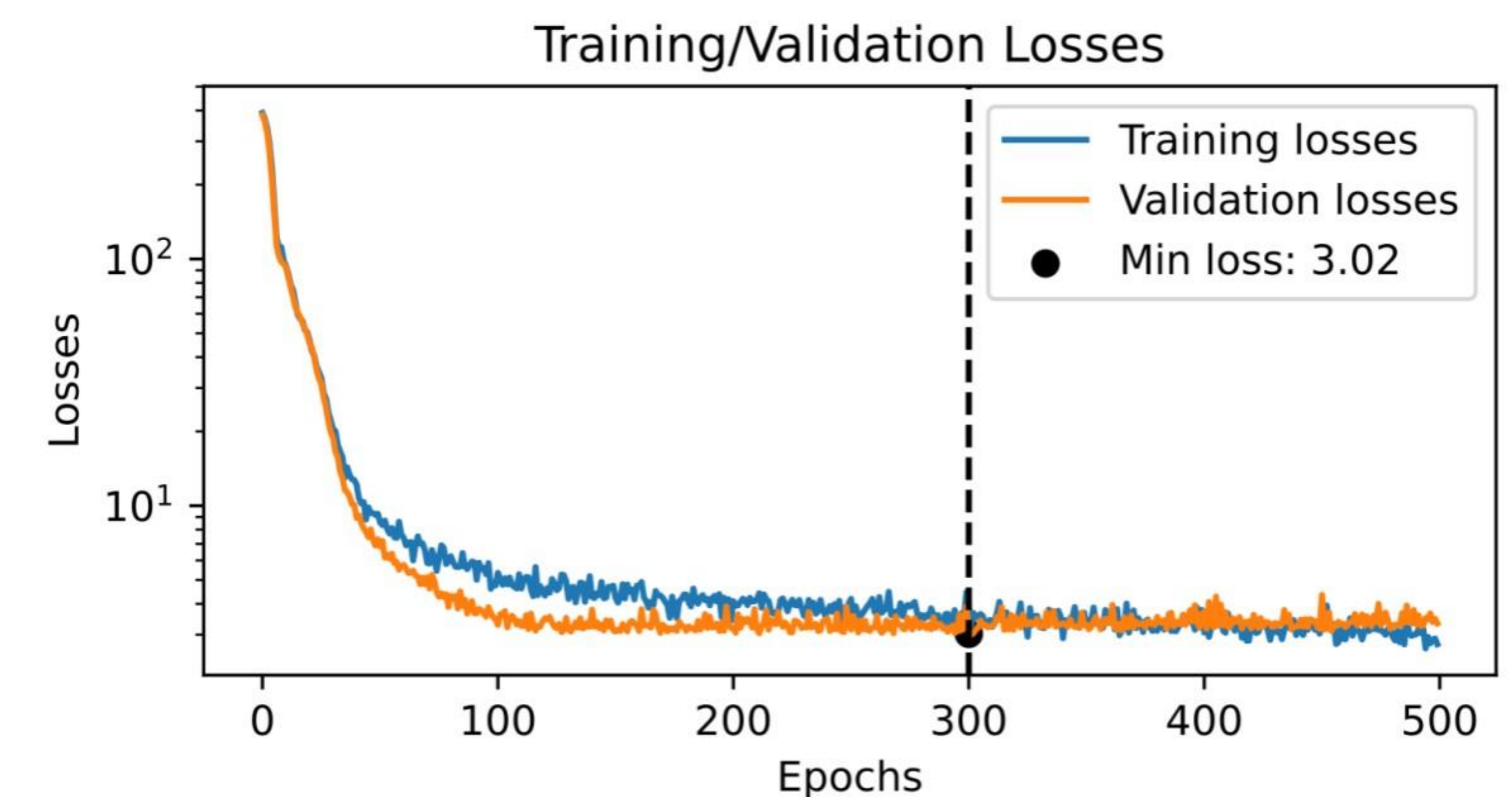
Model

The neural network contained an input layer (41 nodes), first hidden layer (41 nodes), second hidden layer (82 nodes), and an output layer (1 node), with each hidden layer activated by the ReLU function.
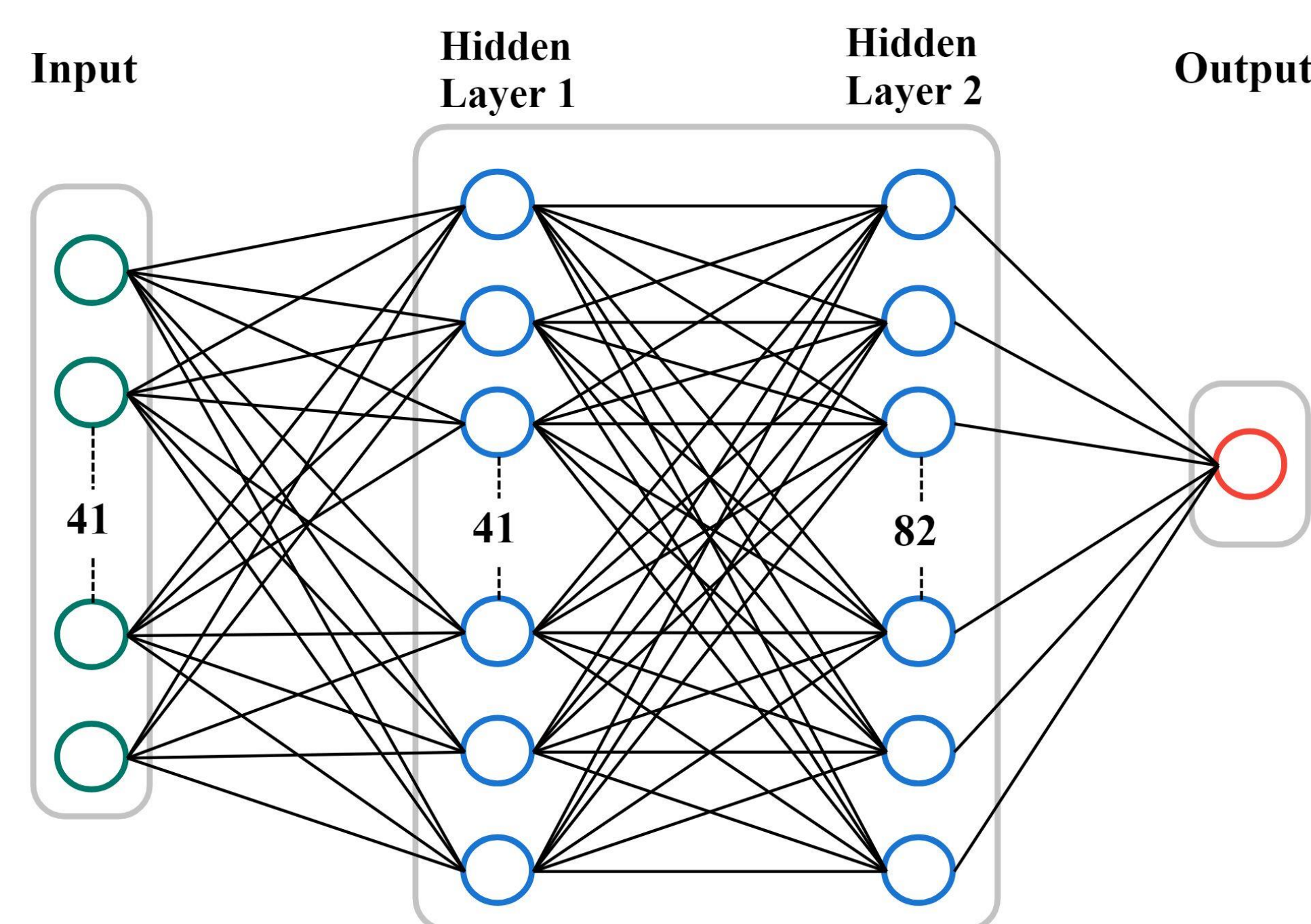
Using PyTorch, we trained the model with the mean square error loss function (MSELoss) as the criterion, and the Adam optimizer, a gradient descent algorithm that adjusts the training parameters, with 10^{-5} weight decay.

Results

The neural network was evaluated by comparing training loss and validation loss. The lowest validation loss occurs at 301 epochs.

The best neural network produced the following results.

	Absolute Error (kg)	Relative Error (%)
Mean	1.2108	6.4200
Standard Deviation	0.8969	4.7557
Median	0.9944	5.2724
Maximum	3.9143	20.7542

Conclusion

Our model was able to estimate ALM with an average of 93.6% accuracy, with the worst case having a 21% relative error. In the future, increasing the amount of training data would improve the accuracy of our model.

Acknowledgement