This program allows for the training and verification of a neural network for predicting soil heat conductivity based off of different parameters.

Use this section of code for training the neural network. Make sure the .txt file for rawdata contain nothing but msilt, organic matter % mass, Θqtz, $\rho$b dry, n, $\theta$w or $\theta$total, and is saved in the "Documents" folder.
Make sure the .txt file for dataresults contains nothing but the results, having one per line.

In[ ]:=
```
rows = 1200;

rawdata = Import["Parameters.txt", "List"];
dataresults = Import["Results.txt", "List"];
classifylist = {};
For[i = 1, i ≤ rows, i++, {
  AppendTo[classifylist, rawdata[[i]] → dataresults[[i]]];
 }]
heatConductivity = Predict[classifylist, Method → "NearestNeighbors"];
```

Use this bit of code for predicting $\lambda$ values by entering the following values: msilt, organic matter % mass, Θqtz, $\rho$b dry, n, $\theta$w or $\theta$total

```
heatConductivity[{}]
```

Use this section of code for get the RMSE values for error checking

In[ ]:=
```
checkdata = Drop[rawdata, rows];
checkresults = Drop[dataresults, rows];
checklist = {};
For[i = 1, i ≤ Length[checkresults], i++, {
  predictedresult = heatConductivity[checkdata[[i]]];
  error = (predictedresult - checkresults[[i]])^2;
  AppendTo[checklist, error];
 }]
rmse = Sqrt[Total[checklist] / Length[checkresults]]
```

Out[ ]=
```
0.47401
```

Use this section of code to get the RMSE values for error checking with data other than the default.

```
checkdata = Import["June 24 Check Data 2.txt", "Table"];
checkresults = Import["June 24 Check Results 2.txt", "List"];
checklist = {};
For[i = 1, i ≤ Length[checkresults], i++, {
  predictedresult = heatConductivity[checkdata[[i]]];
  error = (predictedresult - checkresults[[i]])^2;
  AppendTo[checklist, error];
 }]
rmse = Sqrt[Total[checklist] / Length[checkresults]]
```

```
0.493133
```