

Mini Project 1

Due: April 28, 2023, 11:59PM PT

*Student Name: Andy Franck**Instructor Name: John Lipor*

Problem Description

The objective of this project is to first train a Convolutional Neural Network capable of accurately categorizing geothermal heat flow residuals into one of four distinct classes. After categorizing, the model will be studied using guided backpropagation to determine whether the trained model is utilizing certain structural settings to make its predictions (i.e. changes in elevation and topography).

The model will be trained on a detrended elevation map, partitioned into 222, 200x200 pixel images. For each patch, the label is the maximum residual value among all wells in the patch.

Although it would be convenient for the model to accurately predict a high percentage of labels, it is unlikely to happen with the limited data size. This report aims to develop a model that can accurately predict the labels with a reasonable degree of accuracy, and then analyze the model's behavior to determine whether it is utilizing the desired structural features to make its predictions.

Even without perfect model accuracy, the model can still be useful in determining what topographic features are most useful for predicting geothermal heat flow. This will lead to a better understanding of the relationship between topography and geothermal heat flow, and will help to inform future research.

Exploratory Data Analysis

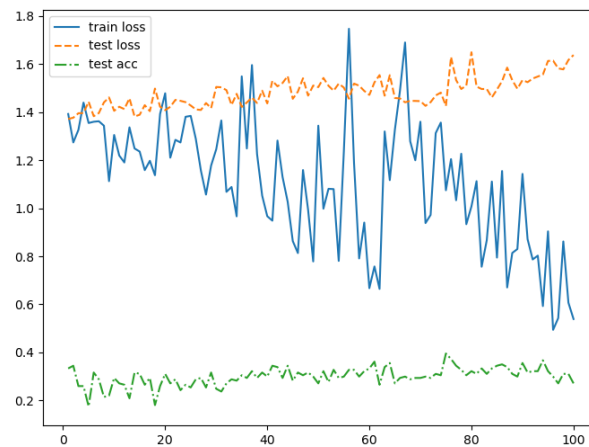
Because of the nature of the data, it was not necessary to perform much data analysis.

Challenges

The largest challenge of the project was the limited data size. With only 222 images, it was very difficult to make a model train properly and accurately predict labels. This was especially true when using a CNN, which requires a large amount of data.

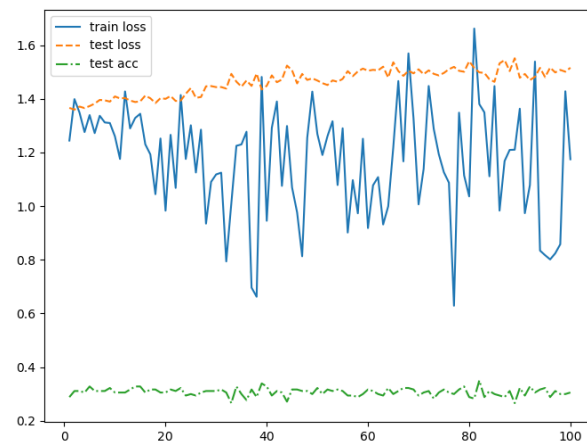
Approach

The initial step in development involved developing a baseline Network in Network model as described in the d2l textbook. This initial model was used to determine whether the data was being processed correctly, and to determine whether the model was capable of learning. This model was trained for 100 epochs, and overfit the data considerably:

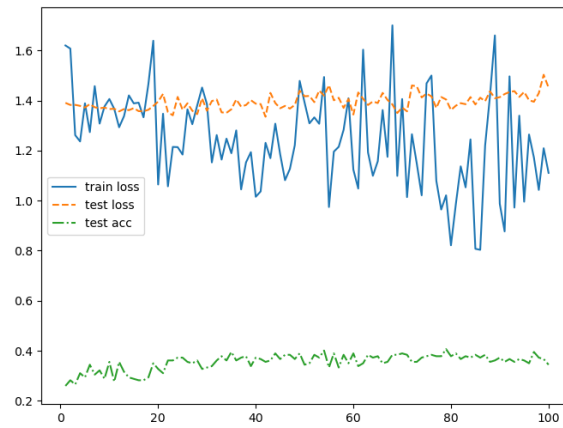


The model also did not appear to be learning very well at all. The loss function was not decreasing, and the accuracy was not increasing. This was likely due to the small data size.

To artificially increase the data size, image augmentation was utilized via the `torchvision.transforms` library. This library allows for the creation of a transform object that can be applied to a dataset. The transform object contained both random flips and rotations at random degrees. The transform was applied after loading each batch at the start of the training loop, so that each batch was augmented differently. After applying the transform, the model appeared to train slightly better, but still did not appear to be learning very well:



The model still appeared to be overfitting, so dropout was added after each Network in Network block. This helped to reduce the overfitting, however the model did not improve in accuracy much:



Evaluation and Summary

What I Learned