Alex Howell

Seismic Image classification using Convolutional Neural Networks

Deep machine learning simulates how the human brain learns. The algorithms have multiple layers of hidden nodes to process the data given. Each node has a given weight that is represented as a number that changes as it learns. This method uses a hierarchy of features to determine what is similar and what is different. It starts from the input and checking distinct pixels to being able to differentiate higher-level features. From those weights, it can make predictions on new data. Seismic analysis has significance because it shows underground features without actually having to dig into the ground to save time and money. Depending on what signals are received when scanning the earth, the results can represent faults, oil or even gas deposits. Convolutional neural networks (CNN) are a way to decipher the deference in features. Convolutions are a mathematical of two functions to make a third, and with that it creates a specified number of feature maps to tell differences apart.

In the convolution algorithms around two thousand pictures were used, half were non-faulted pictures and the rest had faults. The data was retrieved from the directory specified and depending on the class, a label array was created to match the two different classes. After the data was retrieved, it was normalized by multiplying every value by 1/255 to get a value between 0 and 1. The first algorithm only had one convolution with 32 kernels created for feature mapping and the input size of the original pictures which were 150x300x3. After the convolution layer, the pooling layer is used with 2x2 parameter. This function is used to reduce the size of the feature maps to prevent overfitting which means the algorithm finds to many features. The next layer is flatten which then takes the two-dimensional array and turns it into one-dimension. The dense function adds a hidden layer of neurons to process the current data. The next layer is the dropout function to reduce the number of neurons used by the percent given. For binary analyzing to work, the algorithm has to use the dense function again with one neuron to get the decision between two classes. After training, the algorithm had 100% accuracy and almost no loss.

The two convolution algorithm has a similar style except a couple of extra layers. It starts with convolution of the original pictures, and the pooling layer. Next the model has the second convolution with another pooling function. The rest is the same as the one convolution algorithm with the flatten, dense, dropout and dense for one neuron layers. The accuracy was high and loss was low showing the algorithm was successful. The prediction on this model showed 97.5% accuracy. The next algorithm has three convolutions all having thirty-two feature maps and pooling before the dense and dropout functions. After the learning process, the model was generally successful again with high accuracy and low loss. The prediction based off the model was 95% accuracy.

The number of convolutions should increase as the data gets more complicated. The data set used was simpler than most other real-world examples which could explain why one convolution had the highest prediction rate. With more convolutions, the machine could have been overfitted and found excess features that made it predict incorrectly. Activation methods could have been another way to help accuracy.