Eoin Mag Ualghairg

Handmade Neural Network Generator, Trainer, and Data pre-processor.

# Pre-processor

The pre-processor is split into 3 separate scripts written in Python 3.

The first script is a scraper(scraper.py). This script uses Google’s google\_image\_download library to search Google Images using keywords and download those images.

The second script is an image scaler(scaler.py). It uses the Python Image Library(PIL) to scale the images ready for the Neural Network. It opens all images in a folder, scales them to a specified size and overwrites the original files.

The final pre-processing converts the images to greyscale, extracts the 8-bit pixel brightness and saves it to a txt file. Pixelvalue.py uses Python Image Library(PIL) to read, convert to grayscale, extract brightness values and save the images.

scaler.py:



pixelvalue.py:



# Neural network

The Neural Network generation and evaluation is written in C++. The reason for C++ is its speed due to the code being singlethreaded. The main file is called trainer.cpp and with a header file named io.hpp that handles reading pixelvalue(s).txt and converting them to a vector. Writing strings to a text file. Reading text files. Printing the model, a 3D vector. Generating a random double between min and max(inclusive) and splitting a string along a delimiter and returning a Vector.

Firstly, ‘generatemodel()’ is called. It has 4 parameters: input dimensions, number of hidden layers, hidden layers dimension and output layer dimensions. Generate returns a 3D vector of the model:

Model data:



Model generation:



Whiteboard maths:



Training:

A random double is generated(a). The code loops over each node’s weight and generate a new a. If a > threshold it ‘mutates’ and generates a new weight. This is done a custom amount of times and stores each model in a vector. The code runs all the training data on all the models and stores the best performing one. The same mutation code is ran over the new model. This means that a new generation has been formed. This process is done a certain amount of time until a desirable model is reached. After all the generations have been simulated, the best model is saved into a text file for later use.

Running:

The text file, above, is read and the model is loaded. A shared header file is used to reuse code from the training script to run the model on new data.