Machine Learning Capstone Project

1. Source of CNN implementation:

The keras library was used for our CNN implementation with the tensorflow framework as our back end.

1. Experiment Information:
   1. Input image size: Our input image size is 101\*101. This was discovered to be a size that will comfortably hold a worm within its boundaries. We made use of what we called supercropping i.e. when an object was discovered we crop the object and its surroundings into the image. An image is seen as a worm image if most of the worm is within the image.
   2. Data statistics: We decided on two classes, worm and not worm. There are 7000 images in each class; 5500 for training and 1500 for testing. This gives 11000 training images, 3000 testing and a total of 14000 images.
   3. Hyperparameters:
      1. Architecture:

Conv2D(filter=32, kernel\_size=3, activation='relu')

MaxPooling2D(pool\_size=(3,3))

Conv2D(filter=64, kernel\_size=5, activation='relu')

MaxPooling2D(pool\_size=(10,10))

Flatten()

Dense(number\_of\_units=2, activation='softmax')

* + 1. Number of epochs: 6
    2. Optimizer: RMSprop
    3. Loss function: binary\_crossentropy

1. Classifier performance: We end up with the following results:
   * 1. Training accuracy: 96%
     2. Test accuracy: 98%

This was a little strange, test being higher than training but we believe that this is

as a result of there being a lot more training data than test data.

1. Train and test time: The training time for the algorithm is 678 seconds. Also the worst test time we observed for the model is 5 seconds.
2. Comparison with Logistic regression

|  |  |  |
| --- | --- | --- |
|  | Logistic Regression | CNN |
| Training accuracy | 81% | 96% |
| Testing accuracy | 81% | 98% |
| Training time | 300 seconds | 678 seconds |
| Testing time | 75 seconds | 5 seconds |

There is a clear improvement from a single logistic regression to CNN’s in both accuracy and test time. However this is not just as a result of the methods themselves. Measures were taken to improve the data set from the last attempt at solving this problem and there are much less mistakes (i believe none now) in the dataset and so the current implementation should not struggle with its convergence. Also, there were also efforts made to add more data to the data set than last time so this algorithm had more to train on.