Jose Chavez Ramirez

EE104 Section 1

Professor Christopher Pham

April 9 2023

EE 104 Lab 6

**CNN – Baseline + Increasing Dropout + Data Augmentation + Batch Normalization**

After running the Baseline model and achieving a 72% accuracy, other methods were implemented including Increasing Dropout + Data Augmentation + Batch Normalization.

Increasing dropout – Regularization technique used to prevent overfitting. This technique works by randomly dropping out a certain percentage of neurons in the network while it is being trained. This results in the remaining neurons taking on more work. This improves the model’s ability to generalize.

Data Augmentation – Technique used to artificially increase the size of the training dataset by creating a modified version of the original set. Augmentation techniques include rotating, scaling, flipping, and flipping. Again, this allows the model to generalize and prevent overfitting.

Batch Normalization – Technique used to normalize the inputs of each layer in the network. It adjusts and scales the input to make them have zero mean and unit variance. This improves stability while training.

After implementing these methods, we are then able to manipulate the batch size and epoch value to get higher accuracy. The batch size is the hyperparameter that determines the number of samples that will be propagated through a neural network. By using a bigger batch size, the training is faster but causes a loss of accuracy and requires more memory. On the other hand, smaller batch size will result in better accuracy and will require less memory making it less demanding on hardware.

The epoch refers to one full iteration of the training data. For example, if there is a sample size of 20,000 with a batch size of 50, it would take 50 batches for one epoch to be completed. This means that the 20,000 sample would be passed through the network a total of 50 times.

Text

Description automatically generated

Model Results

**Image testing**

Graphical user interface

Description automatically generated

Text

Description automatically generated

After training the AI model, we proceed to test the accuracy of the model. We test it by giving it several images to see if the program can predict what the image is. In the code above we give the program an image link to a bird. We see in the console that the program makes a correct prediction.

**Happy Garden**

The happy garden game works by having the player water the flowers that become wilted. The goal is to keep the flowers alive until the time limit is reached. The game ends when a flower is left wilted for too long or the player touches a fangflower. Fangflowers are randomly generated from the regular flowers at a rate of 5 seconds. The image below shows that the game is over when the flowers were left wilted for two long.

Diagram

Description automatically generated with medium confidence

In the image below we see that the game has ended since a fangflower has touched the cow.

Chart

Description automatically generated with medium confidence

**Hello world to OpenAi**

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Hello world to ChatGTP**

Text

Description automatically generated

A computer screen capture

Description automatically generated with low confidence

The chatGTP program makes use of one of the many model engines in order to best respond to user’s input. The program can be changed by modifying the engine, prompt, max\_token value, n value, stop condition, and the temperature. The prompt parameter determines how the program will answer the users input. This can be set so that the response is always in capital letters or in a different language. The max\_tokens specify the number of tokens that the program can use to generate a response. N is the number of responses that the program will generate, meaning that the larger the number, the slower the response will be. The stop parameter can be used to set an end condition that will stop the response. For example, the program can be set to stop giving a response once it encounters a certain character. The temperature parameter controls the creativity of the response. The higher the value, the more unpredictable the response will be.

Demo Videos

Pt 1 training <https://youtu.be/AYr9tclF1qs>

Image testing and Happy Garden Game <https://youtu.be/M_OEbi32Kd8>

Hello world to OpenAi <https://youtu.be/u9lxQKdHrxA>

Hello world to ChatGTP <https://youtu.be/hlEOydNPUQY>