
Milestone 1 8420

Will Sherrer
Clemson University
wsherre@g.clemson.edu

Ashton Sobeck
Clemson University
asobeck@g.clemson.edu

1 Introduction

CNN's take images and use filters to classify them into different categories. While training a CNN is a novel concept today, tweaking the parameters and structure of a CNN is a problem that depends on different variables. What we want to focus on is also the data that is fed into a CNN. Since these networks are so powerful we want to test the limit of their abilities. Image recognition systems have been in place for a while and will only get better. As the amount of images on the internet and devices will only grow storing them could pose as a challenge. With this project we hope to see if training CNN's on compressed images can be as effective as normal ones. This could help reduce data if compressed images can be regenerated, and classified as well as uncompressed ones.

2 Method

We are reducing these images via PCA. We plan to train 2 models with the regular dataset, and the CIFAR-10 dataset reduced by 20%, 40%, 60%, and 80%. We will compare the Training and Testing Loss and Accuracy.

3 Current Progress

Currently we have designed 2 models that are trainable and testable. These models are both CNN's that will be trained in the CIFAR dataset. We wanted to do 1 smaller model and 1 larger model to see if the added parameters help identify images.

4 Future Work

Now we will work on reducing the CIFAR dataset efficiently. Then we will focus on tweaking the hyper-parameters for efficiency and recording the data

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, None, None, 3)]	0
conv2d (Conv2D)	(None, None, None, 32)	416
max_pooling2d (MaxPooling2D)	(None, None, None, 32)	0
conv2d_1 (Conv2D)	(None, None, None, 64)	8256
max_pooling2d_1 (MaxPooling2D)	(None, None, None, 64)	0
conv2d_2 (Conv2D)	(None, None, None, 128)	32896
max_pooling2d_2 (MaxPooling2D)	(None, None, None, 128)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 128)	0
dense (Dense)	(None, 256)	33024
dense_1 (Dense)	(None, 10)	2570

=====
Total params: 77,162
Trainable params: 77,162
Non-trainable params: 0

Figure 1: Model 1 Parameters

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 32, 32, 32)	896
dropout (Dropout)	(None, 32, 32, 32)	0
conv2d_4 (Conv2D)	(None, 32, 32, 32)	9248
max_pooling2d_3 (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_5 (Conv2D)	(None, 16, 16, 64)	18496
dropout_1 (Dropout)	(None, 16, 16, 64)	0
conv2d_6 (Conv2D)	(None, 16, 16, 64)	36928
max_pooling2d_4 (MaxPooling2D)	(None, 8, 8, 64)	0
conv2d_7 (Conv2D)	(None, 8, 8, 128)	73856
dropout_2 (Dropout)	(None, 8, 8, 128)	0
conv2d_8 (Conv2D)	(None, 8, 8, 128)	147584
max_pooling2d_5 (MaxPooling2D)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dropout_3 (Dropout)	(None, 2048)	0
dense_2 (Dense)	(None, 256)	524544
dropout_4 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 10)	2570

=====
Total params: 814,122
Trainable params: 814,122
Non-trainable params: 0

Figure 2: Model 2 Parameters