CS230

Improving Animal Face Classification: Exploring Data Augmentation and Group Normalization

Emma Spellman
Department of Computer Science
Stanford University
espell@stanford.edu

Abstract

Ecological research often depends on rare and low quality imagery and video. As the climate continues to change at an alarming pace, conservation efforts and ecological endeavors are of increasing importance. This paper analyzes possible improvements to animal classification efforts: data augmentation to battle sparsity of data and Group Normalization to increase accuracy. Initial findings show a positive effect when using these techniques.

1 Introduction

For this project I will be exploring the effects of data augmentation and group normalization on a CNN when applied to an image classification task. More specifically, the model will attempt to classify images into 3 different classes: cat, dog, and wild. This classification task is well motivated, as the use of computer vision is now permeating through the fields of ecology and conservation. Ecological research is commonly data sparse and computationally lacking. I believe computer vision solutions, like the one I will be exploring, can help ecologists and conservationists use the images and videos they collect from the field to more accurately study population density, migration patterns, and physiological characteristics.

2 Related work

Data augmentation is currently a well documented technique in the field [1][2] and consistently produces positive results. For this reason, I believe it is necessary to explore data augmentation within the context of ecological research (you can imagine the need for more images of rare or endangered species).

Group Normalization has been found to produce better results than Batch Normalization [3], however, it's not as widely researched. More specifically, graph normalization is more effective in cases of smaller batches (as batch normalization is susceptible to noise from small batches). Thus, in the ecological field where data can be rare, small batches may be necessary, meaning we should explore how group normalization can maintain metrics for image classification tasks. Finally, I believe exploring the combination of these two techniques is ultimately new and interesting.

3 Dataset and Features

I am using the *Animal Faces* dataset from Kaggle: https://www.kaggle.com/andrewmvd/animal-faces. This dataset contains 16,130 images, all of size $512x512$. These images are divided into three classes:

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

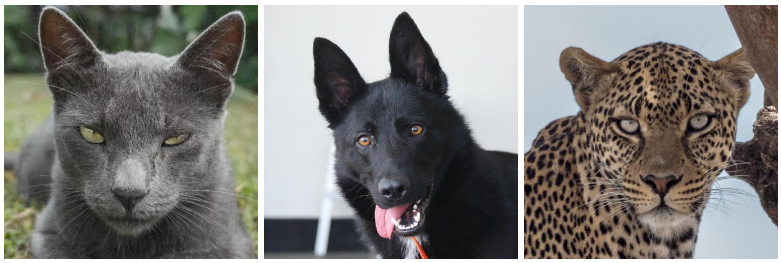


Figure 1: Images of class (a) "cat" (b) "dog" (c) "wild"

cat, dog, and wild (where "wild" is a variety of non-domesticated animals). The data is evenly distributed between classes and no pre-processing is required of the images. Examples of images from each class are shown in figure 1.

4 Methods

Broadly, I used a CNN model built using Keras. The general structure is defined as:

$$Conv2D \rightarrow MaxPooling \rightarrow Conv2D \rightarrow MaxPooling \rightarrow Conv2D \rightarrow$$
$$Normalization \rightarrow MaxPooling \rightarrow Conv2D \rightarrow Normalization \rightarrow MaxPooling$$
$$\rightarrow Conv2D \rightarrow MaxPooling$$

followed by some `dense` and `dropout` layers. Prior to any convolutional layers, the model pads the images with a padding size of 3. The 2D convloutional layers' (in network order) number of output filters are: 112, 72, 64, 32, and 16, respectively. The first convolutional layer has a kernel size of $(3, 3)$ and all other convolutional layers have a kernel size of $(2, 2)$. The activation function for all layers that accept one is `relu`. Each max-pooling layer has a size of $(2, 2)$ and strides vary between $(2, 2)$ and $(3, 3)$. All dropout layers have a dropout probability of 0.5. The final activation layer is a softmax function. The loss function is a "Sparse Categorical Cross Entropy" function with an rms optimizer. Finally, in the baseline model, the `normalization` component is Batch Normalization, whereas in my experimental model it is Group Normalization.

For the data augmentation in the experimental model, in addition to the original image (which is what the baseline model saw), I also add blurred, flipped, and grayed images to the training data. For example, for the original image below:

I produce the 3 augmented image shown in Figure 2. Thus, each image techincally appears 4 times in various forms within the training data set. This augmentation increases the training data size from 14630 examples to 58520 examples.

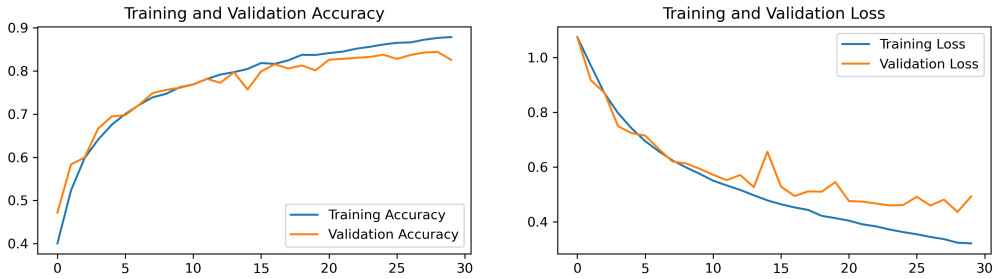
5 Experiments/Results/Discussion

The code can be found at: https://github.com/espell/animalfaces. To reiterate, the baseline model and the experimental model differ in two ways: the training data and the normalization method. The

Figure 2: Original image that is now (a) flipped (b) grayed (c) blurred

baseline's training data is comprised of only the original images, whereas the experimental model has both original and augmented images, as illustrated in the above section. The architecture differs slightly in that the baseline's normalization method is Batch Normalization whereas the experimental model uses Group Normalization.

I ran the baseline model for 30 epochs. I plotted the loss and accuracy for both train and validation:

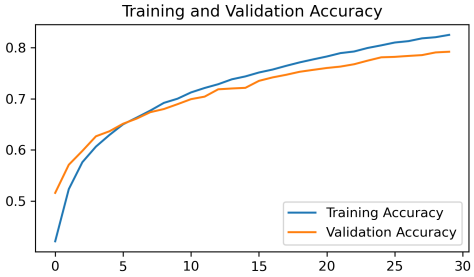


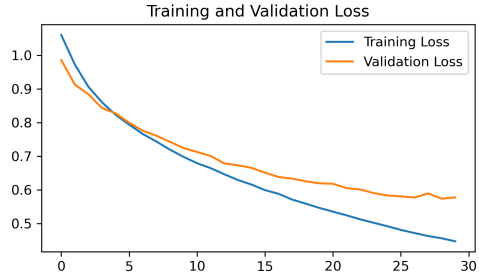
It's clear from these plots that there was still room to grow in accuracy. It doesn't seem that it had approached overfitting yet either. In the next section, I will explain this choice of epochs. In addition to these plots, I output a variety of metrics to measure the performance of the model:

	precision	recall	f1-score	support
cat (Class 0)	0.75	0.86	0.80	500
dog (Class 1)	0.79	0.86	0.83	500
wild (Class 2)	0.94	0.72	0.81	500
accuracy			0.81	1500
macro avg	0.83	0.81	0.81	1500
weighted avg	0.83	0.81	0.81	1500

Something interesting of note here is the "wild" classes precision and recall, which are much higher and lower, respectively. My hypothesis for this is that wild animals (for example, cheetahs) have very salient features (like their spots), that would make true positives easier to accomplish in relation to false positives. However, the f1-scores are fairly consistent across all 3 classes. The final accuracy achieved for the baseline model is 0.81.

For consistency, I also ran the experimental model for 30 epochs and produced the same plots and metrics:





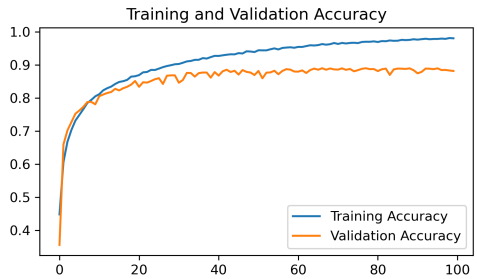
	precision	recall	f1-score	support
cat (Class 0)	0.80	0.84	0.82	500
dog (Class 1)	0.85	0.77	0.81	500
wild (Class 2)	0.81	0.84	0.82	500
accuracy			0.82	1500
macro avg	0.82	0.82	0.82	1500
weighted avg	0.82	0.82	0.82	1500

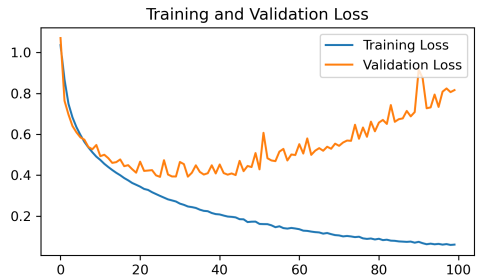
In general, the trend in the plots and metrics were very similar. Interestingly, the "wild" class scores leveled out in the experimental model. I believed the data augmentation aided in this result. Overall accuracy was slightly higher for the experimental model. I believe increasing the number of epochs would widen this difference (as the accuracy was trending steeper than the baseline model).

6 Conclusion/Future Work

As mentioned above, I believe increasing the number of epochs will only emphasize the positive effect of data augmentation and group normalization on this task. Thus, with more time and resources, I would play with increasing the number of epochs as well as tuning the learning rate more closely.

Before the baseline outlined above, I ran a model (for 100 epochs) with the same baseline architecture but all grayed data (for both test and train). This approached produced these plots:





It's clear here (by the stalling in accuracy and increasing validation loss) that when the data is *entirely gray*, the model overfits by around epoch 25 or 30. This is the reason I chose 30 epochs. However, full color imagery of course requires more complexity and learning. Thus, the choice of stopping early at around 30 epochs did not translate to the experiments I ran.

Ultimately, I believe data augmentation and group normalization accomplished what I wanted: they increased the amount of data available (which is necessary in the ecological field) and slightly increased accuracy. In the future, I believe playing with training time and further tuning hyperparamters will highlight these positive effects.

References

[1] Improving Deep Learning Using Generic Data Augmentation (Taylor and Nitschke): https://arxiv.org/pdf/1708.06020.pdf

[2] The Effectiveness of Data Augmentation in Image Classification using Deep Learning (Wang and Perez): https://arxiv.org/pdf/1712.04621.pdf

[3] Group Normalization (Wu and He): https://openaccess.thecvf.com/content_ECCV_2018/html/Yuxin_Wu_Group_Normalization_ECCV_2018_paper.html