Experiments with a Kaggle Dataset: Cats versus Dogs Neural Network Classification

John Nguyen

April 2024

Abstract

This document reports on a series of experiments the author conducted on a dataset from a Kaggle competition [1] to re-acquaint himself with deep learning methods. This challenge includes a dataset of pre-processed images of cats and dogs. Some sample code was provided by the organizer of the competition, which was refactored and expanded upon for the purpose of experimentation. The primary goal of these experiments is to provided a library of models which can be used in future adversarial deep learning experiments. This document provides a quantitative analysis of this library of models including training statistics, methodology and model architecture.

1 Introduction

Within the past two decades, deep learning has demonstrated remarkable engineering feats. From AlphaGo [2], Stable Diffusion [citation needed], and ChatGPT [citation needed], we are finding seemingly transformational applications of AI (artificial intelligence) every few years. A well studied application of these tools is image classification. We can use deep neural networks to process images and attempt to classify them into some chosen categories.

Machine learning is a field of artificial intelligence where agents learn from data. Machine learning can be broken into three categories: supervised learning, unsupervised learning and reinforcement learning. Image classification is a form of supervised machine learning [citation needed] which learns from labelled data. The trained agent then attempts to label new, unseen data with a label appropriate to its training samples. In this experiment, our training data is images of cats and dogs. As you can expect, our labels are whether or not the image is of a cat or a dog. Our models aim to correctly classify if an (potentially previously unseen image) is of a cat or a dog.

The experiments in this project are to analyze the trade-offs with different deep learning architectures. Although this experiment features minor changes to architecture, this allows us to minimize other sources of noise and focus on the quantitative impact of these architectural changes. By varying hyperparameters, we obtain a better understanding of how seemingly innocuous choices of parameters may impact our overall performance and training. Architecture may impact training by enabling the model to better generalize data. In other words, the model is less likely to overfit to the data. Recall that the purpose of machine learning is not learn the training data, but rather learn the distribution of the training data.

Traditionally, the entire dataset is divided into three disjoint sets: the training set, validation set and testing set. The training set, as it's name suggests, is the part of data that is used to train the model. The validation set is used to compare different hyperparameters. Some experiments do not include a validation set as there may be insufficient data, or the experiment attempts to develop a suite of models, not just the singular best performing model. This experiment falls in the later category and will not include a validation dataset. Lastly the test set is used to evaluate the models. This data will be used to valuate the models. For this experiment, we use an 80/20 training/test split.

2 Machine Learning Taxonomy

This paragraph will discuss the hierarchical structure between artificial intelligence and deep learning. The following paragraph will discuss the various taxonomies of machine learning and where my experiment lies. Artificial intelligence (AI) refers to the study of computers and how they can think like humans. Machine learning (ML) is a class of artificial intelligence which refers to algorithms that learn from data. Machine learning encapsulates many different architectures and algorithms. One such architecture is deep neural networks. Deep learning refers to the subset of machine learning which utilizes deep neural networks. The purpose of this taxonomy is to develop basic competency in machine learning before we get into technical details of this report.

Machine learning can be broadly divided into three categories: supervised learning, unsupervised learning and reinforcement learning. Supervised learning refers to learning from labelled data. The goal is to look at new, unseen data (referred to as the test set) and determine a label based on previous seen data (called the training set). In unsupervised learning, the model is given unlabelled data and is asked to find patterns in this data. We analyze these patterns to make generalizations about the data. Lastly, reinforcement learning refers to a model learning from it's environment. Upon observing a state and taking an action, the model is given a reward/penalty. Through successive observations and actions, the model is able to map states to their optimal actions. For our experiments, we are firmly working in the domain of supervised learning.

Even in the subsection of machine learning which is classification, we can further categorize our experiments. We apply machine learning to processing images, a subject referred to as computer vision, or CV. CV can classically divided into two branches: detectors and classifiers. Detectors will attempt to find objects in an image. Classifiers will attempt to label images. They are typically combined in a multi-agent system where an image is passed in to the detector which crops out any detected objects and forwards them to the classifier. The classifier then labels the image, reinstalling the cropped object back into the input image, now with a label. For our experiments, we work exclusively with classifiers and assume all input images are of a singular cat or dog.

3 Methodology

We use a neural network as our classifier. A neural network is a data structure which has demonstrated tremendous ability to learn distributions [citation needed]. The name neural network comes from how this data structure was modelled after a human brain. A neural network is divided into layers of nodes. The first layer is called the input layer as it is where the raw data is first passed into. The final layer is correspondingly called the output layer, as it is from this layer that we derive conclusions from the neural network. For our case, the input layer takes in the RGB color values of the input image after a convolution is applied. Our output layer will output a number between 0 and 1, where a value closer to 0 indicates a 'cat' prediction, whereas a value between 0.5 and 1 indicates a 'dog' prediction. Data is passed into individual nodes which are transformed in some way before entering an activation function. This activation function will perturb the transformed data before passing it to the next layer of nodes.

Machine learning is defined as the branch of machine learning where learning agents learn from data. This idea is often implemented using a gradient descent based approach. For neural networks, we use an algorithm called back-propagation to optimize a neural network. Each layer has some number of parameters in their transformation and we want to compute the gradient of these parameters each time we use call back-propagation. By leveraging Leibniz's Chain Rule we are able to calculate the gradient of each parameter in a layer using the parameters of the previous layer. This type of calculation is already optimized by GPUs for graphics and animation. Consequently, we have a surprisingly fast algorithm to optimize neural networks.

The general purpose of these experiments is to isolate the effects of neural network architecture on this particular dataset [1]. Deep neural networks refer to neural networks with many layers and some number of nodes in each layer. In this experiment, we consider regular deep neural networks, narrow-deep neural networks and wide-shallow neural networks. We use a regular deep neural network as a benchmark. A narrow-deep neural network is one with relatively few nodes per layers and a relatively high number of layers in comparison to a regular deep neural network. In contrast, a shallow-wide neural network has many nodes per layer but relatively few layers. We are interested to see if these different architectures will have an effect on the learning and overall accuracy of models that use them. We document our controlled variables in the next section.

3.1 Experiments

Experiment Name	Layers	Convolutional Layers	Linear Layers
DefaultTest	6	3	3
SkinnyDeep	9	1	8
Wide3Layer	3	1	2
Wide4Layer	4	2	2
Wide4Layer_Linear	4	1	3

Table 1: The table contains the architectures for each experiment. Note the varying number of layers and the different types of layers.

In Table 1, we display the neural network paramters of each architectures and their labels. Each architecture was used to produce a 7 epoch mode and a 20 epoch mode. This is to analyze the change in training and test data as we increase the number of epochs. To distinguish between the 7 epoch and 20 epoch models, we will use append _20E to designate the 20 epoch model from the 7 epoch model. Each convolutional layer is two dimensional and has two dimensional max pooling post processing. All activation functions are ReLU except for the last layer, which is always linear.

The purpose of this experiment is to see how changes in architecture can effect performance. As I am writing this (9/3/2024), I realize it would have been better to have more drastic differences in architectures. Nevertheless, I am completing this write-up for completion.

4 Results

For each training epoch, we record the training loss, training accuracy, test loss and test accuracy. Data from each training session can be found in the

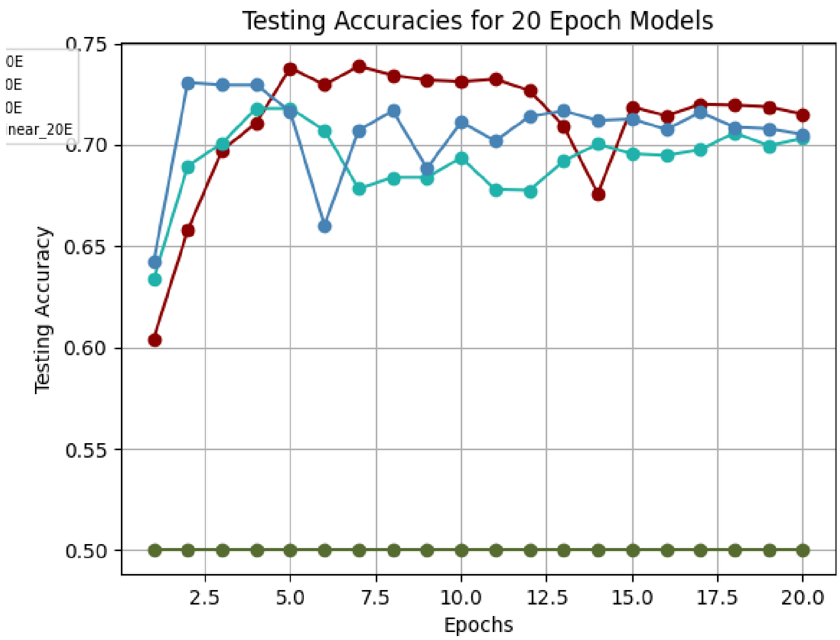

Figure 1: Figure of training accuracies for every architecture in 20 epoch setting.

experiment_data/ directory. Using this data. Diagrams of these results can be found in the diagrams/ directory. For example, Figure 1 is the diagram comparing training accuracies of all architectures with 20 epochs.

5 Analysis

Results for this experiment were largely inconclusive due to the aforementioned similarities in architecture. While we do notice difference in results, such as in Figure 1, we also acknowledge that these results are not as pronounced. In my personal study, I found recently discovered and read through [3] and found much better examples. Thus I will be cutting my analysis portion short.

6 Conclusions and Future Work

Ultimately, this project started off as a labor of love which was pushed aside once life events got in the way. I am returning to this project to simply finish this write-up to have a proper conclusion. I recommend studying [3] if this sort of experiment interests you – Dr. Raff provides much better examples.

Fundamentally, the differences in architecure do not have any general advantages that apply to all problems. As Raff describes in [3], differences in architecture are meant to model the structure of the data. Consequently, we would expect convolutional neural networks models to outperform linear neural networks models in image classification, such as in this example. However, this is not because convolutional neural networks are generally superior to linear networks, but instead due to convolutional models exploiting the spatial relationships of image data.

This project served as a good warmup for getting back into deep learning. Though I understand this write-up is not as satisfying nor well written as I initially hoped, I did not want this project to hang in my mind. I currently have neither the motivation nor interest in this topic to continue pursuing this line of effort. Nevertheless, I will leave this write-up with one resolution that this project has taught me: I enjoy the intersection of data analysis and machine learning. This includes analysis of deep learning metrics, as outlined in this paper, or using deep learning to perform advanced analytics. Thus I willbe focusing my efforts there next.

References

[1] Will Cukierski. Dogs vs. cats, 2013.

[2] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, jan 2016.

[3] Edward Raff. *Inside deep learning: Math, algorithms, Models.* Manning Publications Co. LLC, 2022.