Generative Modeling

Anna Dai

October 31, 2020

1 Introduction

The field of AI has seen an explosion since the turn of the century, prompting both excitement and alarm; underlying the quest for machines that can reason, learn, and act intelligently is the fear that we may render ourselves obsolete in the process (and perhaps initiate our very own robot apocalypse).

We can typically classify AI into one of two types:

(1) Discriminative modeling is used to teach machines how to categorize existing data. With a large enough dataset, a machine can learn to categorize a dataset of handwritten numbers by digit, to differentiate positive and negative reviews on Yelp, to detect the probability of an x-ray containing a broken bone. The model is trained on a dataset for which we already have the correct labels for, compared to the ground truth, and modified to improve its accuracy.

(2) Generative modeling is used to teach machines how to generate new examples of the observations in the provided dataset. For instance, given a large enough collection of faces, a generative model might be able to output images that might look like real people, but are actually faces that do not exist on the face of the Earth. If a discriminative model estimates the probability of an observation belonging to a certain category, a generative model estimates the probability that the observation occurs in the first place. In the context of the example, this translates to the probability that an arrangement of pixels might be called a face at all.

A majority of the progress made in the last twenty years has been on the discriminative front; not only does it have more applications in a business setting, it is easier to evaluate the quality of a discriminative model. While we can surely verify whether an existing image is a face or not, it is much more difficult to evaluate the quality of what is essentially a fake face. On top of that, out of the $255^{32\times 32}$ 32×32 pixel images that one can make, an extremely small subset of those would fall into the category of what we want to create, rendering its creation a very difficult task.

However, if we are to truly achieve artificial intelligence in a machine, its ability to create and generate must be a part of it. Current neuroscientific theory has even suggested that the human brain is not, in fact, a complex discriminative model operating off our sensory input, but actually a generative model that has been trained since birth to simulate our surroundings. It is clear that a deeper understanding of generative modeling will bring a new perspective to the idea of artificial intelligence. Below are two examples of applications of generative learning:

Figure 1: Above are images of faces generated by a generative adversarial network. The figure comes from [2].

Figure 2: Above are examples of face swapping generated by a generative adversarial network. The figure comes from [3].

2 Generative Modeling

Probability is the foundation of all modeling. The framework is as follows: essentially, the dataset can be described by a density function p_{data}, which we want to imitate as closely as possible with our model. We then want our model to generate other examples that appear to have been drawn from the dataset, without too closely resembling anything already in it. We can model structured data with low conditional dependence by parameterizing the attributes of the data, but oftentimes, we will not be looking at data with specific, finite attributes. Rather, artificial intelligence often aims to work with unstructured data such as images or audio files, whose unit pixels and soundbytes have a high level of conditional independence. In this case, it is much more difficult to construct a probabilistic model. We rely, instead, on an application of AI, deep learning.

3 Deep Learning

Deep learning is a type of machine learning that uses neural networks, and is particularly suited for learning unstructured, unlabeled data. A neural network consists of many layers, each layer containing a set of units and a set of weights that relate it to the next layer. For

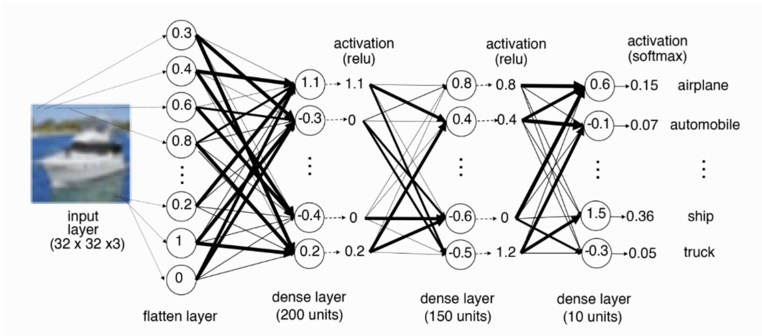

Figure 3: Above is a neural network of the CIFAR-10 dataset, consisting of three layers. Each layer is mapped to the next by a function; the 'relu' function is very commonly used in the intermediate layers. The last layer uses a special function, 'softmax,' that produces the probabilities that the image at hand is any of the ten labels. Its probabilities should sum to one. The figure comes from [1].

example, a type of layer called the dense layer is often used, where all units in the current layer are connected to all units in the next layer, with a weight attached to each connection. The conceptual driving force behind deep learning is representational learning. In other words, the model learns to distinguish the important features of the dataset: what makes a face a face, what makes audio classical music.

In a discriminative example, the goal is to have the deep learning model categorize data points with a set of labels. What we therefore want the final layer to output is a probability that the data point belongs to each of the labels; a model would return the label with the highest probability. In creating this model, we would first initialize all weights to some arbitrary value ϵ. With each data point in the training set, we would evaluate the accuracy of the model by comparing the expected and actual label of the data point, modify the weights at the final layer to decrease the model's error, and backpropagate that change throughout the entire network. Essentially, with the weights at each layer, we always want to travel in the direction of "lowest error" [1]. Below is an example of a neural network trained on CIFAR-10 data (a built in data set from Keras), which attempts to categorize images of various animals and vehicles.

4 GANs

GANs stands for generative adversarial networks, and is a technique derived from deep learning shown to be able to generate new data with the same statistics as the training set. The premise is as follows: GANs has two neural networks, one generative and one discriminative, play each other in a zero-sum game where the generative network creates candidates that the discriminative network then evaluates. The ultimate goal of the semester is to achieve an understanding of the theory behind GANs, as well as code a simple generative adversarial network.

[1] Note that this mirrors the idea of a gradient function in multivariable calculus.

5 Proposed Methodology

We will follow the text Generative Deep Learning by David Foster to understand the foundational concepts of deep learning and generative modeling. In addition, our goal is to supplement the theory covered by the textbook with:

- Research papers describing novel uses of generative learning and GANs, such as facial image generation and swapping. By examining current ways that modern society has made use of generative technology, we are able to 1) study the theory of generative modeling in real life examples, and 2) explore the social implications of the astronomical improvements of generative models in recent years.

- Practical coding examples in Python to give us a better understanding of deep learning. We will use the Keras and TensorFlow libraries, a popular artificial intelligence API for the Python language. We plan to begin by coding discriminative models, which will help us familiarize ourselves with TensorFlow as well as the more general concepts that underlie both discriminative and generative learning. Later in the semester we will progress by examining techniques such as convolutions and pooling that can improve their quality. Using what we have learned, the goal for the end of the semester is to shift gears to generative models and code an elementary GANs.

References

[1] D. Foster, *Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play*, O'Reilly Media, Inc, 2019.

[2] T. Karras, S. Laine, and T. Aila, *A style-based generator architecture for generative adversarial networks*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 4401–4410.

[3] Y. Nirkin, Y. Keller, and T. Hassner, *Fsgan: Subject agnostic face swapping and reenactment*, in Proceedings of the IEEE international conference on computer vision, 2019, pp. 7184–7193.