GANs

Latent space is a hidden representation of the data that captures its underlying patterns. In this space, similar data points are situated closely together.

First of all, I’m going to introduce the concept of a Generative Adversarial Network. This idea was first introduced by Ian Goodfellow in 2014, a Google scientisit. A GAN consists of two parts, a generator and a discriminator, both of which take the form of Neural Networks. These are models with trainable parameters whose structure loosely resemble that of the human brain. The goal of the GAN is that it takes training set of data and it is hoped that the generator will learn to create data that resembles the distribution of the training data as closely as possible. The discriminator’s job is to take an input of an item of data, either from the real data or the generator’s output and output a probability that the data came from the real set. A training process ensues, the discriminator learns to distinguish between real and generator data with more examples. The generator’s goal is to fool the discriminator, and it learns to create samples which become more and more alike to the actual training data. In this process, the generator is analogous to a team of counterfeiters trying to produce fake currency and use it without detection while the discriminator is like the police trying to detect the counterfeit currency. Both teams improve their methods until the counterfeits are indistinguishable from the genuine articles. Coming back to this diagram, it is hoped that over the training process, the system will converge to a point where the discriminator outputs a probability of ½ for any given input, that is, it cannot distinguish between the training data and the generator’s output and the generator learns the distribution of the training data.

BiGANs

In order for the GAN to produce realistic images, it must really develop an understanding of the underlying structure of the training data, which is that the feature space distribution of the data represents. Unfortunately, the standard GAN model only provides a means of mapping from the latent distribution to the data distribution, by the generator model. However, it does not provide a means of learning the inverse mapping from a given piece of generator data back into the feature space. In the context, of my project, this is what is required for feature learning. A new framework, called a bi-directional GAN, or BiGAN for short, enables the learning of this inverse a mapping. The BiGAN framework requires the addition of an additional Neural Net called the encoder which it is hoped will learn the reverse mapping of the generator. The discriminator in the BiGAN not only discriminates in the data space but also in the latent space, and it takes pairs of inputs. In order for this discriminator to be fooled, not only must the generator learn the distribution of the training data but the encoder must also learn the generator’s inverse mapping.

Motivation

So now that the background knowledge has been covered, I can try to explain what the motivation behind my project is. The main aim is that the learning of the underlying features of a data distribution will enabled improved classification of data of that type. Classification is a classic problem in Machine Learning. It essentially is the task of taking an input and assigning it a class label. A simple example would be a binary classifier for images of cats. Such a classifier could take an image as input and assigns it a label of either cat or not a cat, based on its confidence level. If the classifier is then tested on images it has never seen before, the percentage of images that it labels correctly is known as its classification accuracy. Building a good classifier requires a lot of data to train on and the data must also be manually labelled which is a very big job. The BiGAN provides a method of learning the really important underlying characteristic data of what makes up a cat image. So a classfier can then be built which looks for these essential features and outputs cat if it finds them and not cat if it doesn’t find them. In this simple case, learning features can negate the need for labelling. And it is hoped also that the classification accuracy will improve and also the computational resources required by the classifier will reduce. In larger, more complex data sets a small number of labelled examples may be required but it will nonetheless be much less intensive than a fully labelled example.

The BiGAN provides a means of unsupervised feature learning. This means that the data in question is never labelled as being a certain type. The GAN process we saw earlier simply takes the training data as input, unlabelled and learns its underlying features. The data fed to the discriminator is of course labelled as being from the training data or generator but this is a generic approach and it does not tell anything about the characteristics of the actual data. If we were radically change the domain of the data, the generator and discriminator would be trained in much the same way. So the final point on the merit of the BiGAN is that it is not domain specific, the framework that I’ve already presented is largely a universal approach and should be transferrable to a vast range of images and other domains.

The pictures at the bottom of the slide.

Feature Learning

So I’ve included this side to try and give more of an visual representation as to what kind of features can be learned by these networks. A Neural consists a number of layers which consist of linear transformation of weights and biases and a non-linear activation function. The example I have here is that of data set of face images. This is a very simple network only showing three layers but the idea of the feature learning comes across. At the lower layers, it’s been shown that the layers pick up low-level features such as edges. As we progress the layers, the mid-level layers start to pick up recognizable features of an actual face such as eyes and noses. In the higher levels, what resemble actual faces begin to appear. So this data set has a very hierarchical set of features which makes it quite visually demonstratable. So as my project progresses and I try different image sets, I will hopefully be able to show obvious learning patterns such as this.

Goals

So that’s more or less the background to my project. I’m now going to run through what I hope to achieve for the remainder of my project.

First of all, I hope to learn the features of various different image data sets. At first, I will look at small, custom built image data sets and try to really understand how to discover their essential features. This may be something as simple as a data set of cats or dogs. With each new data set, the BiGAN may have to be slightly tweaked in order to ensure that realistic images are produced by the generator and that good feature learning is taking place. I will try to adapt the BiGAN, to large, more complex data sets, the kind of mainstream industry data sets that are used in research papers. Examples of this are MNIST, a handwritten digit data set form US censory data. This is a rather simple one. A more complex would be the CIFAR data sets which contain natural images according to classes like bird and airplane. Then the benchmark data set would be ImageNet which contains natural images with 1000 classes as complex as specific as a bald eagle.

I also hope to identify a domain in which I can learn really disctinct features that I can focus my work on. This may be a specific set of images, for example, maybe faces, or it could even to extend to a different domain entirely, such as audio, depending on progress.

Once this had been identified I will have to build a data set for this domain, learn the features and hopefully use these to show impressive classification results.

Work completed

To date, a lot of my time has been spent getting up to speed with the considerable material of my project. This involved a state of art literature survey investigating the closest methods for feature learning.

I then began to implement simple neural networks for image classification tasks as this knowledge is what underpins my project.

I then moved on to implementing an actual GAN. It’s written in Python, it uses TensorFlow which is framework for Deep Learning from Google. It’s trained on the MNIST data set shown earlier using a Titan X GPU provided by my supervisors which I share time on with some of my fellow masters colleagues. This gives satisfactory results that I showed earlier.

And up to now, I’ve begun implementing the additional networks to enable the BiGAN, which as of now is yet to be completed.

Work Plan

So up to now I haven’t focused on the feature learning part of my project. So my work plan from now is that, by the time I come back in January I hope to have finalised a general BiGAN implementation, have experimented with various forms of image data sets and find how to really learn the underlying features and with the knowledge gained from this process to be able to identify a domain of specialisation for my project. From then on, I hope to build a comprehensive data set for that domain, learns its features and show their classification capabilities