Eric Foote
3502094
CS 4795

Survey Paper on Auto Encoders

Introduction

Auto encoders are a type of neural network artificial intelligence that is trained to reconstruct its input. They have a wide area of applications that are going to be highlighted in this paper. This paper is going to first explain the background and theory behind auto encoders and then highlight some of the applications of auto encoders with some additional points about some work that I did on machine translation that was the original topic for this paper before complications arose and I had to change plans. We are firstly going to begin with a basic overview of auto encoders.

Background Information

Overview

Auto encoders are a type of neural network that is trained to reconstruct its input. It consists of 3 portions, the encoder "a feed forward, fully connected neural network that compresses the input and encodes the input image as a compressed representation in a reduced dimension." The code "the reduced representation of the input that is fed into the decoder." Finally, the decoder which "is responsible for reconstructing the input back to the original dimensions from the code." [3] We can interpret this as the encoder takes some k dimensional data and transforms it to p dimensional data and the decoder does the opposite of it taking p dimensional data and transforms it to k dimensional data. Therefore it can be concluded that "the main objective of the auto encoder is to get an output identical to the input." [3] The formal mathematical definition of auto encoders is "to learn the functions

$$A : \mathbb{R}^n \rightarrow \mathbb{R}^p \text{ (encoder) and } B : \mathbb{R}^p \rightarrow \mathbb{R}^n \text{ (decoder) that satisfy}$$

$$\arg\min_{A,B} E[\Delta(\mathbf{x}, B \circ A(\mathbf{x})], \quad (1)$$

where E is the expectation over the distribution of x, and delta is the reconstruction loss function, which measures the distance between the output of the decoder and the input."[4] which restates what we have stated previously only with a more mathematical rigour. Using the above mathematical definition we can note a special case if A and B are linear operations, then our auto encoder is a linear auto encoder, when applying a linear auto encoder we "would achieve the same latent representation as Principle Component Analysis (PCA)" [4] where latent representation is defined as "The latent space representation of our data contains all the important information needed to represent our original data point."[7]. In other words this representation allows us to reduce the size (in terms of dimensions) of a particular data point down to a smaller representation without losing any necessary information. Now that we have a basic overview of the topic we can now start to briefly discuss some of the different types of auto encoders.

Types of Auto Encoders

There are 4 different types of auto encoders: convolutional, variational, denoising, and deep. We are going to begin our overview with convolutional auto encoders. CAE (convolutional auto encoders)

shave a simple structure they “learn to encode the input in a set of simple signals and then reconstruct the input from them”. [3] So in other words in a set of simple operations go from input to output and then the opposite from output to input. The next type is variational and they “tend to make strong assumptions related to the distribution of latent variables” [3] where this distribution “typically matches the training data much closer than a standard auto encoder” [3] and they can generate “new images”. [3] So another way to look at this is this particular auto encoder when it learns how to reduce the dimension of a particular data set also learns how to generate new values that satisfy the particular probability distribution of the data set. The next type of auto encoder is denoising and they “add some noise to the input image and learn how to remove it”. [3] This means that it is good for data that has some corruption in it since it will learn how to remove that corruption and get back to the original data. Finally, there is deep and they are “composed of two symmetrical deep-belief networks having four to five shallow layers.” [3] So we can see that this variation is simply more complex then the others we have stated so they are “able to learn more complex features” [3]. Now that we have an example of some of the various types of auto encoders we are now going to look at some applications of this particular neural network.

Methods

In the methods section we are going to outline a few applications of auto encoders and expand on a particular one that I was originally going to implement using Python. We begin first with the applications of auto encoders.

Applications

Some applications of Auto encoders are dimensionality reduction which is the branch of statistics concerned with the reducing the number of random variables needed to describe a particular data set. We can see this as the general process outlined for the encoder step, the transformation from p dimensional data to n dimensional data (using the same mathematical representation as previously). Another particular application is image colourization, being able to add or remove colour from an image, we can see this being a use for denoising, if we define the data and the process a certain way we can perform the process of adding or removing colour. Image generation is another application, “the idea is that given input images likes images of faces or scenery, the system will generate similar images”. [3] The idea is to be able to learn underlying features of the data set to be able to create new data points, this is an example of variational. Feature extraction is another one which is the ability “to learn important hidden features present in the input data”.[3] This is an example of denoising auto encoders, since they have to learn features of the data set in order to remove any possible corruption from it. Image denoising is an application and we have seen this is an example of denoising auto encoders, since thats how they are defined. The final application that I want to briefly touch upon is data compression, however, they are not used in place of traditional data compression algorithms since auto encoders are highly focused rather then the traditional algorithms being able to be applied to any data. One application that I want to expand further on is language translation and we are going to actually highlight some personal work trying to apply this technique to this particular application.

Language Translation

Language Translation is concerned with taking words from one particular language and translating them to another language. My original plan for this project was to create a German to English translator from scratch only taking advantage of packages like Tensor Flow or other helpful packages. I collected 2 different corpus’ one from the EU open data portal collected by the Federal office in Berlin

and the other from a tutorial on the Tensor Flow website. I was only able to get a word count from each sentence; I was unable to get the actual auto encoder created by myself, I was only able to use the example code from the tutorial. I didn't go any farther with that tutorial since I didn't feel I could create it on my own so I didn't proceed any further. I will expand on this further in the future work section.

Future Work

Some future work that I want to do as it relates to this type of model is to actually implement it on some of the applications that I have discussed and to continue my study of machine translation and hopefully be able to actually implement this model. Another thing I would like to do is dig deeper in to the theory behind some of the types of auto encoders, I only went very basic in my discussion of them and I would ideally like to learn some more on them. I am now going to draw some conclusions on the whole project.

Conclusion

In conclusion, I was very ambitious in my goals of creating a translator from scratch in the time frame given to do this project. However, auto encoders are a very interesting topic they are a fully connected neural network that is able to transform p dimensional data to n dimensional data and back to p dimensional data through 3 steps, the encoder, code and decoder. Furthermore, we have seen a few types of auto encoders such as convolutional, variational, denoising and deep with a variety of applications such as dimensionality reduction and image denoising.

Sources

[1] https://iq.opengenus.org/applications-of-autoencoders/
[2] https://sci2lab.github.io/ml_tutorial/autoencoder/
[3] https://www.mygreatlearning.com/blog/autoencoder/
[4] https://arxiv.org/pdf/2003.05991v2.pdf
[5] http://proceedings.mlr.press/v27/baldi12a/baldi12a.pdf
[6] https://en.wikipedia.org/wiki/Autoencoder
[7] https://towardsdatascience.com/understanding-latent-space-in-machine-learning-de5a7c687d8d