Image Compression over MNIST Dataset using Autoencoder

R.J.Poovaraghan [1], Ankush Rai [2], Ronit Dharmik [3], Siddharth Bhattacharjee [4]
[1] Assistant Professor (S.G), Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India.
[2, 3, 4] Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India

Abstract – Image compression has been the area of investigation for many years. A huge amount of images are being stored by Multinational companies and different organisations. For the last few year, deep learning has been very successful in computer vision tasks, and now its being used in image compression. Deep Autoencoder neural network trains on a large set of images to find similarities between the images in the set. Then those similarities are used to compress and present them using simpler and less codes then current image compression techniques. In this paper, we demonstrate how Deep Autoencoder neural network is used to train data and large sets of images are compressed. MNIST dataset is used for training and testing purpose. A 28 x 28 image is first converted to a vector of 784 x 1, which is then given to the network. Then we take this vector and compress it in each iterations of training phase of the network. After the vector is compressed, a compressed feature vector is hold by the middle hidden layer of the network to be stored or transmitted, Later, that vector is decoded back to obtain the original image.

Keywords—Image compression, Deep Auto encoder, MNIST Dataset, Vector.

1. INTRODUCTION

of images stored in a file system are almost identical for example, X-ray images in a medical database, large set of fingerprint images in a police database, CT scan images of brain, etc. These images have similar histograms, edge distributions and similar pixel intensities in same areas. These likenesses in an arrangement of pictures can be useful to decrease measure when putting away In many cases, group, transmitting and breaking down them by applying between picture encoding systems. Profound Autoencoders are one of the strategies that encode between picture pixels. (Karadimitriou, 1996) In advanced picture compression, there are four sorts of redundancies that can be utilized to pack pictures:

- Coding excess

- Inter pixel excess

- Psycho visual excess

- Set excess

Coding, entomb pixel and psychovisual repetition are abused by the present techniques for picture compressions. These three kinds of redundancies show up in monochrome pictures. Be that as it may, in an arrangement of comparable pictures, noteworthy measure of between picture repetition is available. The purported comparable pictures have following highlights:

- comparable pixel powers in similar zones,

- tantamount histograms,

- comparable edge appropriations and

- similar to appropriation of highlights.

The connection between pictures is estimated by item minute relationship coefficient, or Pearson's r. At the point when pictures having above highlights are estimated utilizing the coefficient, the relationship got is high. This measurable relationship is because of the nearness of between picture excess. This repetition is purported set excess. This set excess can be utilized to enhance picture compression proportion. (Karadimitriou, 1996)

2. RELATED WORK

Deep Autoencoders perform nonlinear dimensionality decrease on input pictures to diminish them to a packed portrayal. It has been utilized various occasions in the past to pack pictures. The outcomes got from this usage and the experiences picked up from their work are specified beneath.

2.1. Image compression using Huffman coding

Huffman Coding is a lossless compression algorithm which relegates codes to the pixels in view of the recurrence of event. Pixels happening all the more oftentimes will have shorter codes while pixels that happen less as often as possible will have a similar length codes. The outcome is variable length codes that contain vital number of bits. The codes wind up having interesting prefix property. The codes are put away on a parallel tree. The paired tree is constructed utilizing the codes, beginning from the surrenders and moving over to the foundation of the tree. An examination was performed to pack a grayscale picture utilizing Huffman coding algorithm. Huffman coding was utilized to diminish the quantity of bits required to speak to every pixel. Trial results demonstrated that up to a 0.8456 compression proportion was acquired on the picture. (D. Santa Clause Cruz, 2000)

2.2. Comparison with PCA and JPEG

Anand Atreya and Daniel O'Shea have introduced their utilization of Deep Autoencoders. Their initial step was to prepare the autoencoder. First the pictures were bolstered into the nonlinear encoding layer. This encoding layer progressively packed the information picture pixels into a compacted highlight vector. After the encoding was finished, the compacted highlight vector was again bolstered into the translating layer which at that point turned around the procedure of the encoding layer to acquire the recreation of information pictures. The distinction between the remaking and information pictures was viewed as blunder to additionally limit the distinction. Additionally, a sparsity parameter was utilized to guarantee sparsity of the initiation vectors in each layer with the exception of the yield layer. The following stage was to locate a perfect arrangement of quantization esteems utilizing k-implies grouping to encode the enactment esteems. At long last, they contrasted this compression algorithm and other regular algorithms like JPEG and PCA. They found that Deep Autoencoder picture compression algorithm beat JPEG in high compression and decreased quality plan. Additionally, Deep Autoencoder indicated better non-direct portrayal of the information picture than that of PCA and thus Deep Autoencoder would be advised to remaking quality. They saw that Deep Autoencoder could discover factual regularities in a particular area of pictures which was impractical by JPEG. (O'Shea, 2009)

2.3. Representing and generalizing nonlinear structure in data

G. E. Hinton and R. R. Salakhutdinov have additionally displayed the utilization of autoencoder to lessen the dimensionality of information. They prepared a heap of Restricted Boltzmann machines (RBMs) to pre-prepare the neural network with each RBM comprising a layer of highlight locators. After the pretraining, the RBMs were unrolled to get full layers of the neural network which finally were calibrated utilizing Backpropagation algorithm to lessen the information remaking blunder. RBM is a two layer network in which the information twofold pixels are associated with the concealed layer utilizing symmetrical weights. The double pixels are the hubs in the unmistakable layer and the concealed layer enactments are refreshed utilizing sigmoid capacity on the whole of the considerable number of results of data sources and their individual weights. Again a sigmoid capacity is utilized on the total of the considerable number of results of concealed qualities and their particular weights. The shrouded layer here goes about as an element indicator. The quantity of hubs in the concealed layer is constantly kept littler than the quantity of pixels in the info picture with the goal that the shrouded layer does not wind up taking in a personality work. With steady diminishing in number of pixels in the shrouded layers, more RBMs are included onto the network with first layer of highlight locators as noticeable layer until the point when a coveted encoding state is come to. It is specified in the paper that each layer of the locators catches solid, high-arrange relationships and this is an effective method to continuously uncover low-dimensional, nonlinear structure. In the wake of pretraining the network, the encoder part of the neural network is then unrolled to get the decoder part where same weights of autoencoder are utilized. The weights are later balanced in the tweaking stage to get ideal reproduction. They as a rule presumes that backpropagation together with Image Compression Using Deep Autoencoder 6 Deep Autoencoder is an animal power towards productive speaking to and summing up nonlinear structure in information. (R. R. Salakhutdinov, 2006)

2.4. Deep Autoencoders as feature detectors

Andrew Y. Ng and others have shown the utilization of expansive scale unsupervised learning algorithms for building abnormal state highlights which incorporates the utilization of pile of RBMs to develop autoencoders. They have introduced the preparation utilizing unlabeled information to make abnormal state, class-particular component indicators. They have prepared a 9-layer privately associated meager autoencoder with pooling and nearby differentiation standardization on a substantial dataset of pictures (the model has 1 billion associations, the dataset has 10 million 200 x 200 pixel pictures). They prepared the network utilizing model parallelism and nonconcurrent stochastic slope drop (SGD) on a group of 1,000 machines (16000 centers) for 3 days. They demonstrated that it is conceivable to prepare a face locator without naming pictures as containing a face or not. They have utilized meager Deep Autoencoder with three imperative constituents: nearby open fields, pooling and neighborhood differentiate standardization. The nearby open field technique is utilized to scale the autoencoder to huge pictures. The thought behind this technique is that each component in the autoencoder can interface just to a little district of the lower layer. Likewise, to accomplish invariance to nearby mishappenings, they utilized neighborhood L2 pooling and neighborhood differentiate standardization. L2 pooling permitted to learn invariant highlights. The deep autoencoder was developed by reproducing three times a similar stage made out of nearby separating, neighborhood pooling and nearby differentiation standardization. The yield of one phase is contribution to the following one and the general model is finally deciphered as a nine-layered network. The first and second sublayers are known as separating and pooling individually. The third sublayer performs nearby subtractive and troublesome standardization and it is roused by natural and computational models. The weights are not shared which permit the learning of a larger number of invariances other than translational invariances. More or less, they demonstrate that Deep Autoencoders perform well as highlight indicators if there should be an occurrence of pictures. (Andrew Y. Ng, 2012)

3. PORPOSED MODELLING

The primary objective of this undertaking is to utilize Deep Autoencoder neural network to pack dim level pictures to get a 4:1 compression proportion on MNIST manually written digits dataset. This undertaking exhibits the utilization of Deep Autoencoder neural network to pack 28 x 28 pixel grayscale picture to a size of 14 x 14 picture. It can recreate just the estimation of the first picture.

3.1. System Architecture

- Data Collection

The handwritten digit pictures from MNIST database is utilized as the preparation and testing dataset. The pictures are of size 28 x 28 (784 aggregate pixels in each picture) and subsequently the span of information layer is 784. There are out and out 60000 pictures in the MNIST database. Initial 20000 pictures are utilized as preparing information and 1000 pictures are utilized as testing information. The information tests are appeared in Figure 1 and Figure 2.

255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255
255	255	110	0	44	224	255	255	255
255	223	18	2	3	184	255	255	255
255	244	80	2	3	184	255	255	255
255	255	111	2	3	184	255	255	255
255	239	64	2	3	184	255	255	255
255	229	34	2	3	131	224	255	255
255	255	130	2	3	3	147	255	255
255	255	255	2	3	3	147	255	255
255	255	255	0	2	2	147	255	255
255	255	255	2	3	3	147	255	255
255	255	255	2	3	3	147	255	255
255	255	255	2	3	3	147	255	255
255	255	255	0	2	2	85	255	255
255	255	255	2	3	3	3	213	255
255	255	255	106	3	3	3	111	255
255	255	255	146	3	3	3	111	255
255	255	255	255	37	2	2	0	220
255	255	255	255	80	3	3	2	220
255	255	255	255	182	3	3	2	220
255	255	255	255	224	44	3	2	220
255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255

Figure 1 - Data sample showing digit 1

255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	248	216	216	216	56	47	20	200	25	0	1	55
255	255	255	255	255	192	181	91	33	23	0	0	0	0	0	3	22	0	1	12	133
255	255	255	255	157	1	0	0	0	0	0	0	0	0	0	92	107	107	146	175	255
255	255	255	255	216	4	0	0	0	0	0	10	17	1	1	255	255	255	255	255	255
255	255	255	255	255	109	32	76	0	0	8	231	255	168	33	255	255	255	255	255	255
255	255	255	255	255	255	224	253	33	0	96	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	45	0	13	250	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	231	13	0	124	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	183	1	3	29	74	253	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	108	1	0	0	63	202	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	164	15	0	0	36	198	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	220	92	0	0	15	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	0	0	0	133	255	255	255
255	255	255	255	255	255	255	255	255	255	255	163	52	17	0	0	8	250	255	255	255
255	255	255	255	255	255	255	255	255	175	37	3	0	0	0	0	17	255	255	255	255
255	255	255	255	255	255	255	204	68	4	0	0	0	0	9	112	255	255	255	255	255
255	255	255	255	255	206	130	6	0	0	0	0	10	108	250	255	255	255	255	255	255
255	255	255	216	23	4	0	0	0	0	12	109	235	255	255	255	255	255	255	255	255
255	147	22	3	0	0	0	0	1	50	231	255	255	255	255	255	255	255	255	255	255
255	47	0	0	0	6	48	51	220	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

Figure 2 - Data sample showing matrix of digit 5

- Network architecture setup

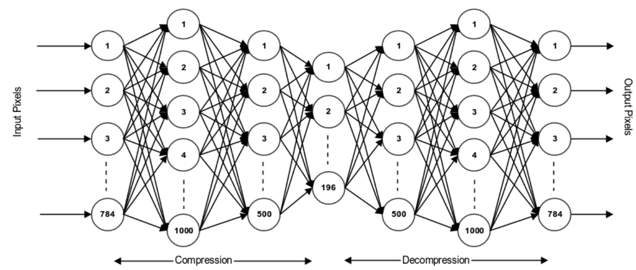

Figure 3 - Deep Autoencoder architecture

The neural network utilized here is a Deep Autoencoder which comprises of two symmetrical deep conviction networks that regularly have two to four layers of encoding shrouded layers and another two to four layers of disentangling concealed layers. Each match of layers in the deep conviction network is a piece of a heap of Restricted Boltzmann Machines (RBMs). Each combine comprises of two layers: an obvious layer and a shrouded layer (include identifying layer).

The encoding layer comprises of three shrouded layers with every one of size 1000, 500 and 196 individually. The primary concealed layer is estimated 1000 hubs in light of the fact that extending the size along these lines remunerates the fragmented portrayal given by the sigmoid conviction units. Consequently, the encoding layer packs the 784 pixels into 196 qualities.

- *Training*

The encoding layer is first pre-prepared by framing 3 RBMs: input layer and first concealed layer, first shrouded layer and second concealed layer and second shrouded layer and third shrouded layer. Each RBMs are pre-prepared at first to deliver a 196 size packed element vector. At that point, the RBMs are unrolled to shape the unraveling layer of the profound conviction organize. Subsequently, the interpreting layer comprises of two shrouded layers and one yield layer of size 500, 1000 and 784 separately. The interpreting layer at that point changes the compacted include vector by feed sending it through the layers to get the estimated recreation of the information picture.

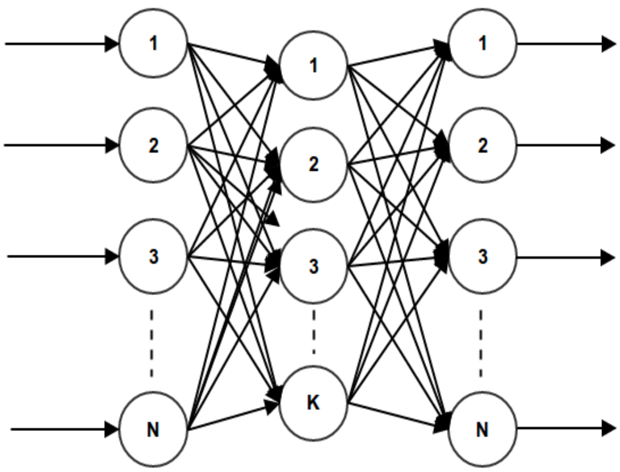

Figure 4 - Restricted Boltzmann Machine

K is the degree of the inside covered layer. This covered layer takes in the features of the data layer which is of size N. These machines when set intelligently reducing the K's regard outline an encoding layer which a little bit at a time entirety up over the main information.

The inclination for the sigmoid capacity of the hubs in the system was registered as:

$$g'(z) = \frac{d}{dz}g(z) = g(z)(1 - g(z)) \quad (1)$$

where sigmoid is given by:

$$\text{sigmoid}(z) = g(z) = \frac{1}{1 + e^{-z}}. \quad (2)$$

The cost function used for the network was:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[-y_k^{(i)}\log((h_\theta(x^{(i)}))_k) - (1 - y_k^{(i)})\log(1 - (h_\theta(x^{(i)}))_k)\right] \quad (3)$$

J - cost function

m - no. of images

y - target images(in this project, input images)

h - network output

4. RESULTS AND DISCUSSIONS

MNIST dataset is used for training and testing purpose. A 28 x 28 image is first converted to a vector of 784 x 1, which is then given to the network. Then we take this vector and compress it in each iterations of training phase of the network. After the vector is compressed, a compressed feature vector is hold by the middle hidden layer of the network to be stored or transmitted, Later, that vector is decoded back to obtain the original image.

5. CONCLUSION

This application can pack the MNIST handwritten digits pictures up to 4:1 compression proportion. The yield acquired changes the 0-255 qualities in unique picture to just two dark scale esteems 0 or 255 in the reproduced picture. This is because of the threshold activity performed in the preparation stage.

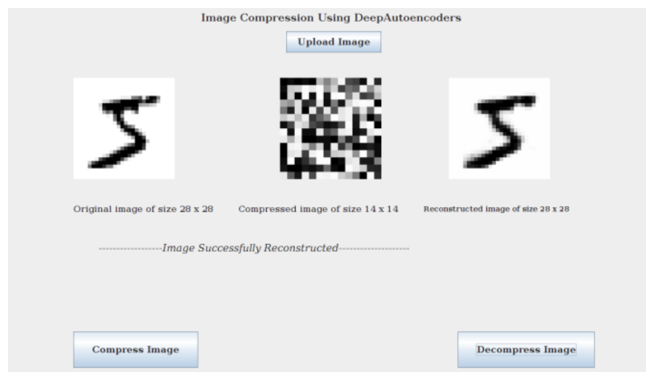

Figure. 5 - After image is reconstructed from compressed form

REFERENCES

[1] Andrew Y. Ng, Q. V. (2012). Building High-level Features Using Large Scale Unsupervised Learning. 29th International Conference on Machine Learning, (pp. 2-5). Edinburgh, Scotland, UK.

[2] D. Santa-Cruz, T. E. (2000). JPEG 2000 still image coding versus other standards.

[3] Daniel B. Russakoff. Carlo Tomasi, T. R. (2004). Image Similarity Using Mutual Information of Regions. In J. M. Tomas Pajdla, Berlin: Springer Berlin Heidelberg.

[4] Karadimitriou, K. (1996). Louisiana: Department of Computer Science, Louisiana State University.

[5] O'Shea, A. A. (2009). Novel Lossy Compression Algorithms with Stacked Autoencoders. Stanford University.

[6] R. R. Salakhutdinov, G. E. (2006, July 28). Reducing the Dimensionality of Data with Neural Networks. SCIENCE VOL 313, pp. 505-507.