https://doi.org/10.2352/ISSN.2470-1173.2018.2.VIPC-153

CrossEncoders: A complex neural network compression framework

[a]ChiragAgarwal[1],[b]MehdiSharifzadeh[1],[c]DanSchonfeld[1];
[a]cagarw2@uic.edu,[b]mshari5@uic.edu,[c]dans@uic.edu;
[1]**Department of Electrical and Computer Engineering, University of Illinois at Chicago; 851 S Morgan St Chicago, Illinois 60607, USA**

Abstract

We propose a novel architecture based on the strucuture of AutoEncoders. The paper introduces CrossEncoders - an AutoEncoder architecture which uses cross-connections to connect layers (both adjacent and non-adjacent) in the encoder and decoder side of the network respectively. The network incorporates both global and local information in the lower dimension code. We aim for an image compression algorithm that has reduced training time and better generalization property. The use of cross-connections makes the training of our network significantly faster. The performance of the proposed framework has been evaluated using real-world data from highly competitive datasets like MNIST and CIFAR-10. Furthermore, we show that the proposed architecture provides high compression ratio and is robust as compared to previously proposed architectures and PCA. The results were validated using metrics, such as PSNR-HVS and PSNR-HVS-M respectively.

Introduction

Image compression plays an important role in the fields of telecommunication, and multimedia services. Traditionally, data compression has involved transformation and quantization. The pipeline of image compression specifically consists of three stages: pixel transformation, quantization, and entropy coding [1]. Image compression uses the fact that image data has more data redundancy as compared to other forms of data. This fact is exploited by most image codecs to achieve efficient image compression. Intuitively, one would expect these compressed representations to embody the global aspects of an image. In the study of data compression, there are mainly two types of compression, namely, lossy and lossless compression. Lossless compression includes the field of study in which the recovered data is exactly the same as the original input data. In this paper, we mainly focus on lossy compression where the reproduced image is not an exact replica of the original image. Some information is lost in the coding process. Evidently, a trade-off has to be maintained between the degree of compression and the degradation in image quality. Furthermore, compression of thumbnails, small scale images, is important both in terms of storage capacity and internet bandwidth. Hence, any improvement to small scale image compression will significantly improve the user experience, accessing content over low-bandwidth connections [2].

In recent years, neural networks have become useful to perform tasks that were accomplished by adhoc algorithms and heuristics [2]. With the current surge in the field of neural networks, we see them achieving state-of-the-art results in tasks such as image classification, object detection, and natural language processing. In the image compression task, we get a lower dimensional image representation by training a neural network with the image and then using the learned weights and coefficients from the hidden layer to recreate the image. We propose an architecture that is able to transform an image into a lower dimensional representation comprising of global conceptual aspects along with lower level details. The proposed framework is an encoder-decoder end-to-end architecture trained using backpropagation for image compression.

Related Work

Neural networks seems to be an ideal solution for the task of image compression due to their noise suppression and feature learning capabilities. Previously, various methods have been proposed for image compression using neural networks. Watanabe et al. [4] developed an algorithm on the basis of modular structured neural networks which consists of multiple neural networks with different block sizes. Abdel-Wahhab et al.[5] extended the two-layer network to multilayer networks. We can convert high-dimensional data to low-dimensional codes by training autoencoders, a special kind of feed-forward neural networks with a bottle-neck hidden layer [3]. Autoencoders used a hidden bottleneck layer to achieve better dimensionality reduction than principal component analysis (PCA). The training of these networks is typically performed using a greedy layer-wise pre-training, like Restricted Boltzmann machines (RBM), that is followed by a fine-tuning stage based on backpropagation.

Long short-term memory (LSTM) networks are a type of recurrent neural networks. Various extensions to the standard LSTM like incorporating spatial information have been proposed. This leads to convolutional LSTMs which were used by [2] for the task of image compression. Gregor et al. [6] implemented variational (recurrent) autoencoders for the problem of compression. Most of the work done in the field of image compression using neural networks used the traditional sequential feed-forward approach. We extend work on CrossNets [7] to the problem of image compression. CrossNets, built on the generalization of sequential feed-forward models like ResNets, uses cross-connections to connect both adjacent and non-adjacent layers. This results in a version of autoencoders which uses cross-connections, defined as *CrossEncoders*. Cross-connections in the network enable efficient reuse of features throughout the network, and thus helping to retain more information in the lower dimensional representation of the input data. Building on the success of CrossNets on the im-

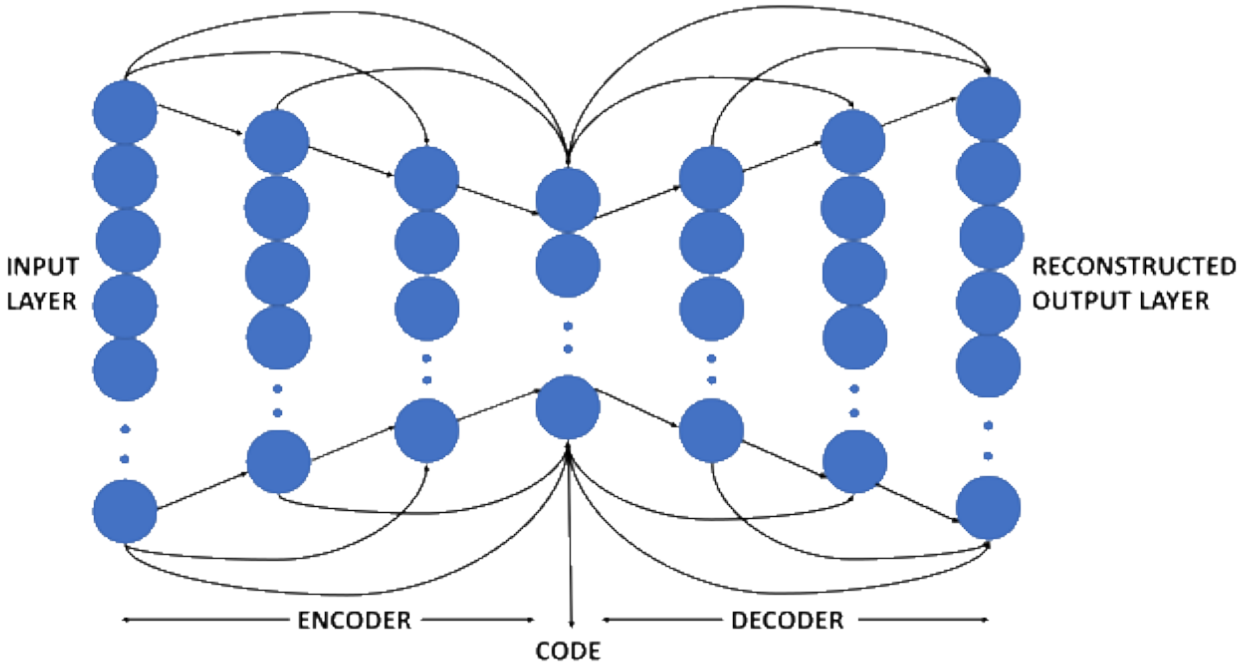

Figure 1. *Proposed Neural Network architecture. An arrow connecting two layers mean that every neuron in those layers are fully connected with one another. This architecture enables neurons in a layer j to connect with neurons in layers j', where $j' = j+1,\ldots,J$, and J is the total number of Hidden Layers in the network on the encoder and decoder side respectively.*

age classification task, we apply it to the MNIST and CIFAR-10 datasets for image compression problem. We compare our results using two commonly used metrics, PSNR-HVS [8] and PSNR-HVS-M[9].

The rest of the paper is organized as follows. CrossEncoder section gives an overview of the architecture concerning image compression. Experiment section illustrates the evaluation results. Finally, we discuss our contributions and findings and draw conclusion and future work in the last section.

CrossEncoder

The proposed architecture (illustrated in Figure 1) is a combination of the traditional sequential feed-forward network along with higher order cross-layer connections among the neurons in different layers. This enables the local features learned in the early layers to be preserved in the later layers along with the global features. Figure 1 gives a general overview of the proposed architecture. It follows a normal autoencoder structure, with the addition of the cross-connections. Hence the name *CrossEncoders*. Given some input data, the network will try to reconstruct it as best as it can on the output. In such problems the input and the output layers have the same size. The input to the network is a N element vector, and then the subsequent hidden layers break down the input data to lower dimension, and finally we get the lower dimensional input code. Clearly given any input data, the compression ratio is controlled by the number of neurons in the code layer. In order to implement cross-connections, we simply concatenate the outputs of all previous layers before inputting them to a new layer. After getting the coded input data, we simply form a symmetrical network in the decoder stage starting with the input as the coded data. The whole network supports an end-to-end training using backpropagation. Common improvements like momentum and early termination to speed up training can be easily added to the network. More architectural details, and results are provided in the following section.

Cross Connections

For any multiple layer perceptron network with layers $0,...,L$, we consider the input layer as layer 0 with I input neurons as $\xi_1,\ldots,\xi_I$. For each layer i from 1 through $L-1$, we define $v_{(i,j)}^{l,m}$ as the weight between neuron l in layer i and neuron m in layer j, where $i<j$. Let $v_{(i,j)}$ denote the matrix of weights from layer i to j. The activation function used is defined by the function g. The explicit output values of any neuron i in layer j, H_j^i, and final output $H_L = y$, are defined recursively as

$$H_0 = \xi, \qquad H_1 = g\left(H_0 v_{(0,1)}\right), \tag{1}$$

$$H_j = g\left(\sum_{i<j} H_i v_{(i,j)}\right), \qquad y = g\left(\sum_{i<L} H_i v_{(i,L)}\right). \tag{2}$$

From equation (2) it is observed that each neuron in any hidden layer $j(>1)$ is recursively computed using the activated outputs from previous hidden layer neurons plus the usual weighted sum of the input from the input layer neurons. The weights in the weight matrix $v_{(i,j)}$ are updated using the backpropagation algorithm. In addition to the backpropagated error through the output-hidden neurons, we have a weight update through the cross-layer weights (weights connecting neurons in layer j with the neurons in layers j', where $(j' > j+1)$) as well. The cross layer weights

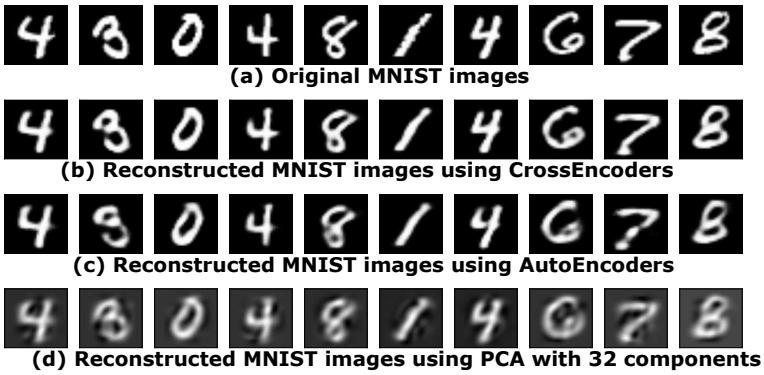

Figure 2. *Reconstruction of a set of 10 random images using CrossEncoders, AutoEncoders, and Principal Component Analysis (PCA)*

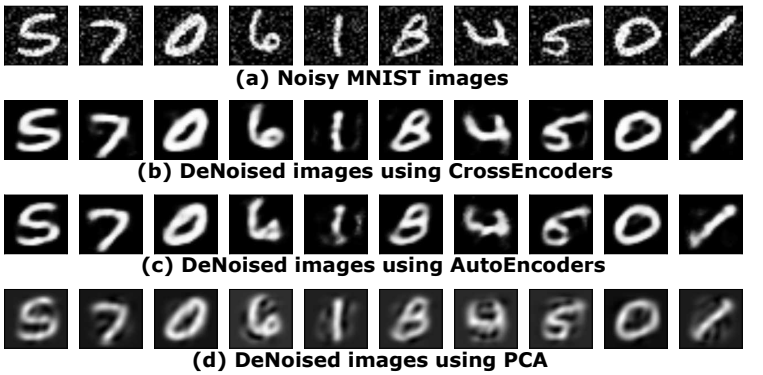

Figure 3. *Reconstruction of a set of 10 random noisy images using CrossEncoders, AutoEncoders, and Principal Component Analysis (PCA). The noise introduced to the MNIST testing images was a standard gaussian distribution noise with zero mean and unit standard deviation.*

determine the effect of neurons in one layer on all its non-adjacent layers.

Experimental Results

We evaluate the effectiveness of our proposed architecture by implementing the cross-layer connections on an autoencoder neural network framework. Using the results from our experiments, we show the superior performance of CrossEncoders as compared to its traditional counterpart. We compare the results both on their visual appearance and evaluation metrics, such as PSNR-HVS, and PSNR-HVS-M.

MNIST

MNIST [10] consists of hand-written digit images. Each sample is a 28×28 binary pixel image. The train and test sets contains 60,000 and 10,000 images respectively. Each image in the dataset was converted into a (1×784) element vector by lexicogrpahically vectorizing the two-dimensional image. The CrossEncoder architecture used for MNIST experiments was based on the classical multiple layer perceoptron network. The corresponding network configuration we used for MNIST was - $784 - 128 - 64 - 32 - 64 - 128 - 784$. Observing the encoder and decoder stage separately, we see the symmetrical nature of the network. The encoder reduces the (1×784) vector to a (1×32) vector, thus reducing the input dimension by a factor of ≈ 24. The compression is controlled mainly by the middle layer containing 32 neurons, in this case. The decoder stage then converts the (1×32) compressed data to the original (1×784) input data. We compared the reconstructed input data results for CrossEncoders, with those of a sequential AutoEncoder network without any cross-connections and PCA. ReLU activations were used for all the layers except the output layer, which uses sigmoid activation. The whole network was trained for 100 epochs. AdaDelta optimizer was used for training the network with MNIST.

Figure 4 illustrates the training difference between the two architectures in terms of cross entropy loss with the increase in the number of epochs. We see better and faster training in the case of CrossEncoders. Additionally, in Figure 2, 3 we can observe the

PSNR-HVS and PSNR-HVS-M metric values between the reconstructed and the original images for the respective datasets. For the case of MNIST it is the mean value of all the testing images, whereas, for CIFAR-10 they are mentioned individually for each class because the contextual information in CIFAR-10 is very complex as compared to MNIST.

	CrossEncoder		AutoEncoder	
Dataset	PSNR-HVS	PSNR-HVS-M	PSNR-HVS	PSNR-HVS-M
MNIST	**68.1779**	**74.0189**	63.5185	66.8311
CIFAR_Airplane	**69.4176**	**71.6445**	66.7032	68.3178
CIFAR_Automobile	**71.9259**	**75.3031**	69.1339	71.7502
CIFAR_Bird	**72.4740**	**75.2403**	70.4905	72.7060
CIFAR_Cat	**70.8002**	**73.5446**	69.9665	72.3577
CIFAR_Deer	**70.5044**	**72.9241**	69.9925	72.1435
CIFAR_Dog	**68.5469**	**70.9579**	68.0573	70.2097
CIFAR_Frog	**68.6494**	**71.0496**	67.8620	69.9140
CIFAR_Horse	**67.3306**	**69.7039**	66.1475	68.0416
CIFAR_Ship	**68.1964**	**70.2893**	66.6365	68.2483
CIFAR_Truck	**65.9498**	**68.1995**	64.6065	66.3831

reconstructed MNIST outputs of CrossEncoders, AutoEncoders, and PCA using a 32-element representation of the original input image. Qualitatively, we see better reconstructed results for CrossEncoders. The superior output is a result of better training using cross-connections. Furthermore, we experimented with noisy input images and performed reconstruction of compressed inputs with the same training model and no prior information of the noise. We introduced a gaussian noise $N(0,1)$ with noise factor of 0.2 to the testing images, and compressed them using the network trained on the 60,000 training images. On reconstructing the input images using the compressed code, we see CrossEncoders performing better denoising. It is to be noted that AutoEncoders lead to false positives, reconstructing digits to a different class, something which is not seen in the case of CrossEncoders. The images denoised by CrossEncoders not only has less noise as compared to PCA and AutoEncoders, but also show superior digit representations. Quantitatively, we compare the performances of CrossEncoders and AutoEncoders in Table 1 using metrics such as PSNR-HVS and PSNR-HVS-M.

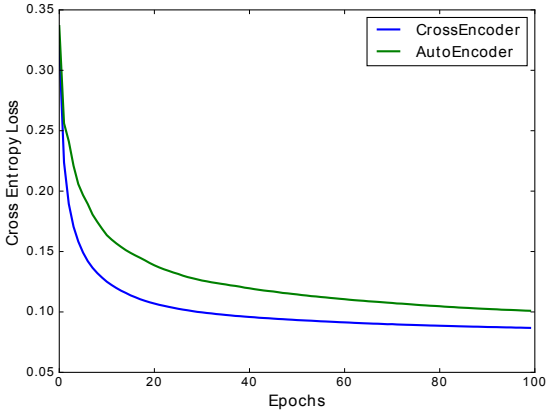

Figure 4. *Plot of cross entropy loss with increase in the number of epochs.*

CIFAR-10

The CIFAR-10 dataset [11] consists of 32×32 color images drawn from 10 different categories. The whole dataset is divided into 50,000 training, and 10,000 testing images. We converted the color images into grayscale for our experiments. No further pre-processing was done on the dataset. The CrossEncoder architecture used for CIFAR-10 experiments was based on the multiple layer perceptron network with a configuration of $1024-512-128-32-128-512-1024$. Each CIFAR image was reshaped into a 1×1024 vector and was provided as an input to the network. We follow a similar training procedure as done for the MNIST experiments. Again, each 32×32 image was encoded into a 1×32 code vector, which was used to reconstruct the original input. For the CIFAR dataset, we reduced the input dimension by a factor of 32. Notably, the only difference between the MNIST and CIFAR-10 experiments is the architectural configuration. MNIST is comparatively an easier task as it consists of binary images, whereas CIFAR-10 contains more variation. Hence, the need for a bigger network for CIFAR. A quantitative comparison is performed on the basis of PSNR-HVS and PSNR-HVS-M as detailed in Table 1. We see CrossEncoders perform no worse, and often better than AutoEncoders. The results in some of the classes are way superior when compared to other classes. We believe using a deeper network or a CrossEncoder based on convolutional neural network for encoding and decoding will result in better performance both qualitatively and quantitatively.

Discussion

Using our proposed network, we aimed to increase the performance of traditional sequential feed-forward networks by adding cross-connections. As seen in the previous sections, CrossEncoders increase the performance of AutoEncoders by in-

troducing cross-connections in their framework. More experiments using bigger networks and diverse data sets would shed more light on the performance of our proposed architecture. This has excellent potential as a future project.

Some contributions of the paper can be listed as follows:

- CrossEncoders have better learning capabilities. They lead to improved reconstruction of images from both normal, and noisy images. They train better and faster.

- CrossEncoders helps in preserving input information efficiently. The PSNR-HVS and PSNR-HVS-M metric values from Table 1 substantiates this point.

Conclusion

In this paper, we introduced an architecture which introduces cross-layer connections. The network was trained using the traditional back-propagation algorithm. As seen in the experimental evaluations, the cross-layer connections enable the neurons to learn efficiently. This was validated in the performance results for both the MNSIT and CIFAR datasets. Building on the results of this paper we aim to experiment CrossEncoders on convolution neural network style architectures.

References

[1] Jiang, J. "Image compression with neural networksa survey." Signal Processing: Image Communication 14.9 (1999): 737-760.

[2] Toderici, George, Sean M. O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. "Variable rate image compression with recurrent neural networks." arXiv preprint arXiv:1511.06085 (2015).

[3] Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." science 313, no. 5786 (2006): 504-507.

[4] Watanabe, Eiji, and Katsumi Mori. "Lossy image compression using a modular structured neural network." In Neural Networks for Signal Processing XI, 2001. Proceedings of the 2001 IEEE Signal Processing Society Workshop, pp. 403-412. IEEE, 2001.

[5] Abdel-Wahhab, Osama, and Moustafa M. Fahmy. "Image compression using multilayer neural networks." IEE Proceedings-Vision, Image and Signal Processing 144, no. 5 (1997): 307-312.

[6] Gregor, Karol, Frederic Besse, Danilo Jimenez Rezende, Ivo Danihelka, and Daan Wierstra. "Towards conceptual compression." In Advances In Neural Information Processing Systems, pp. 3549-3557. 2016.

[7] Agarwal, Chirag, Mehdi Sharifzhadeh, Joe Klobusicky, and Dan Schonfeld. "CrossNets: A New Approach to Complex Learning." arXiv preprint arXiv:1705.07404 (2017).

[8] Egiazarian, Karen, Jaakko Astola, Nikolay Ponomarenko, Vladimir Lukin, Federica Battisti, and Marco Carli. "New full-reference quality metrics based on HVS." In Proceedings of the Second International Workshop on Video Processing and Quality Metrics, vol. 4. 2006.

[9] Ponomarenko, Nikolay, Flavia Silvestri, Karen Egiazarian, Marco Carli, Jaakko Astola, and Vladimir Lukin. "On between-coefficient contrast masking of DCT basis functions." In Proceedings of the third international workshop on video processing and quality metrics, vol. 4. 2007.

[10] LeCun, Yann. "The MNIST database of handwritten digits." http://yann. lecun. com/exdb/mnist/ (1998).

[11] Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009).

Author Biography

Chirag Agarwal *received his B.Tech in Electronics and Communication engineering from Future Institute of Engineering and Management, Kolkata, India in 2012. Currently, he is a PhD student in the department of Electrical and Computer engineering at Univeristy of Illinois at Chicago. His current research interests are in computer vision, and neural networks. Along with his current research he works on the application of deep learning in the field of medical Imaging.*

Mehdi Sharifzadeh *received his BS in electrical engineering from Sharif University of Technology in 2012. Currently, he is a PhD student and researcher in the Department of Electrical and Computer Engineering at University of Illinois at Chicago. His current research are in machine learning, and problems in Image Processing and Computer Vision.*

Dan Schonfeld *received the B.S. degree in electrical engineering and computer science from the University of California at Berkeley in 1986 and the M.S. and Ph.D. degrees in electrical and computer engineering from the Johns Hopkins University, Baltimore, MD, in 1988 and 1990, respectively. In 1990, he joined the University of Illinois at Chicago, where he is currently a Professor in the Department of Electrical and Computer Engineering. He has authored over 120 technical papers in various journals and conferences. His current research interests are in multi-dimensional signal processing, image and video analysis, computer vision, and genomic signal processing.*