compress.io

PRODUCT PIPELINE

1.Problem Analysis :

Data compression is used nearly everywhere on the internet - the videos you watch online, the images you share, the music you listen to , even the blog you're reading right now. Compression techniques make sharing the content you want quick and efficient. Without data compression, the time and bandwidth costs for getting the information you need, when you need it, would be exorbitant!

2.Designing the solution :

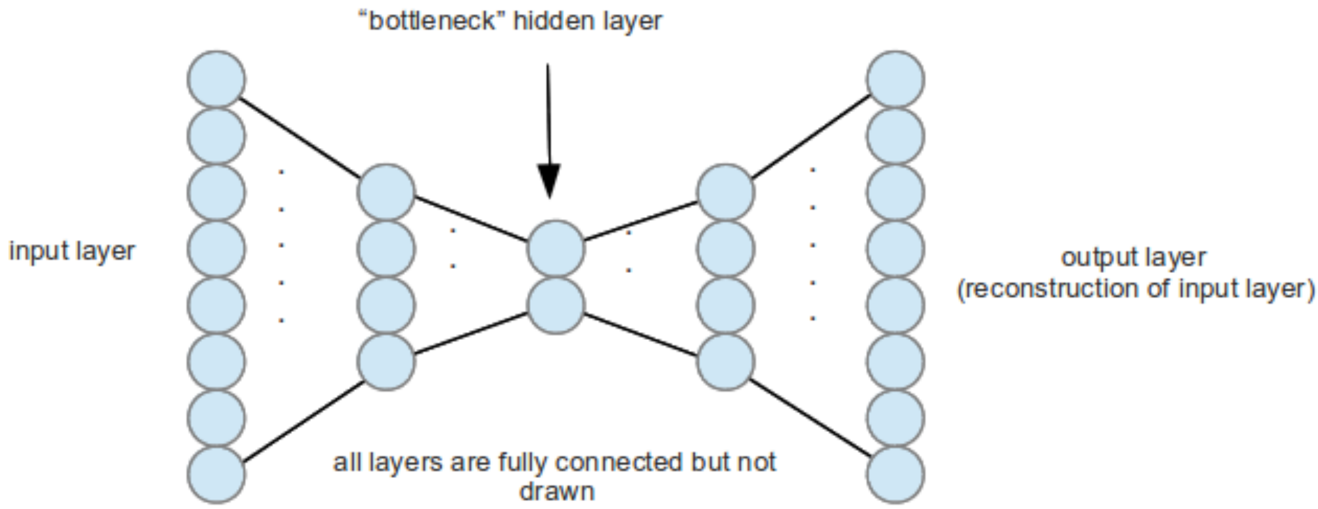
First, we design a novel CAE architecture to replace the conventional transforms and train this CAE using a rate-distortion loss function. Second, to generate a more energy compact representation, we utilise the principal components analysis (PCA) to rotate the feature maps produced by the CAE, and then apply the quantisation and entropy coder to generate the codes. Experimental results demonstrate that our method outperforms traditional image coding algorithms, by achieving a 13.7% BD-rate decrement on the Kodak database images compared to JPEG2000. Besides, our method maintains a moderate complexity similar to JPEG2000.

3.Implementing the decided solution :

— Data Collection

The handwritten digit pictures from MNIST database is utilised 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 utilised as preparing information and 1000 pictures are utilised as testing information.

— Network Architecture Setup



The neural network utilised here is a Deep Auto encoder 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.

Architecture functions :
-Activation functions : Relu , Sigmoid
-Optimizer function : adam
-Loss function : binary_crossentropy

— 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 organise. 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.
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.

— Testing

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.

— 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.