STATISTICAL APPROACH TO IMAGE COMPRESSION BASED ON A RESTRICTED BOLTZMANN MACHINE

V.V. Krasnoproshin, V.V. Matskevich
Belarussian State University
Minsk, BELARUS
e-mail: `krasnoproshin@bsu.by`

Abstract

The article deals with the problem of color image compression. A statistical approach based on a restricted Boltzmann machine is investigated. An original algorithm for learning the neural network is proposed. The algorithm implements the annealing method. The effectiveness of training is based on the use of data parallelization technology.
Keywords: restricted Boltzmann machine, annealing method, data compression, training

1 Introduction

As a result of the rapid development of the information society and the transition to the era of the digital economy the problem of transferring large amounts of data arises. Therefore, for their effective transmission, the problem of information compression becomes urgent.

Compression can be performed without loss, for which different data coding algorithms are used. Or with losses, by constructing various types of informative statistics. The latter allows to achieve a higher compression ratio. In particular, lossy compression can be applied when minor data loss is not critical. For example, in television broadcasting, the blurring of an image or the complete loss of several frames is not significant.

One approach to solving the problem of compressing color images is associated with the use of a restricted Boltzmann machine. However, the effectiveness of the solution in this case essentially depends on the organization of the learning process.

The paper proposes an effective learning algorithm (which implements the ideology of the annealing method), based on the technology of data parallelization.

2 Problem analysis

Consider the problem of compressing colored rectangular images. In the most general form, it can be formulated as follows.

Let N color images be given. Each image is defined by a matrix of dimension nxm. The elements of the matrix i, j are the pixels that describe the brightness value of the image at a given point

$$\begin{cases} P = \{p_{ij}\}, i = \overline{1,n}, j = \overline{1,m} \\ p_{ij} = (x_{ij}, y_{ij}, z_{ij}) \in ([0, 255])^3 \end{cases} \tag{1}$$

where x_{ij}, y_{ij}, z_{ij} - the content of the red, green and blue colors in a pixel p_{ij}.

It is required to develop an algorithm that implements image compression process with minimal loss of information.

Consider the solution of the formulated problem using the restricted Boltzmann machine. It is one of the varieties of recurrent neural networks. Traditionally restricted Boltzmann machine is used to solve problems of compression, semantic search and data cleaning [1]. The process of training such a network requires a lot of time. Therefore, gradient methods (CD-n, PT-n-k and PCD) are usually used to train it. At the same time, the last two are modifications of the base CD method.

Sometimes during sampling, an annealing method is used to improve the accuracy of the gradient estimation [2]. However, the annealing method is not used directly as an algorithm for training a restricted Boltzmann machine. It is considered that the learning process in this case is time consuming. However, with increasing computational power of computers, this problem becomes less significant.

It can be noted that the annealing method has several advantages:

- with an infinitely large number of iterations (with a decrease in temperature exponentially), the method converges almost certainly to the optimal solution, i.e

$$P(\lim_{n \to +\infty} x_n = x^*) = 1 \tag{2}$$

where x_n - state at the moment of the n-th iteration, x* - optimal solution;

- does not require differentiability of the optimized functional;

- is simple to implement (only meta-parameters and set of transitions are configured);

- the problem of a local minimum is solved in a natural way with the help of high temperatures.

3 Architecture of the restricted Boltzmann machine

Let us consider in more detail the architecture and features of the functioning of the restricted Boltzmann machine. At the heart of the machine is the concept of a stochastic neuron. The result of the stochastic neuron is the implementation of a random variable with a certain distribution law. Depending on the distribution law in the neurons of the input and hidden layers, there are several types of restricted Boltzmann machines. In this case, the type Gauss-Bernoulli is considered. This choice was made on the basis of the theorem of the universal approximation. This variant of a restricted Boltzmann machine of Gauss – Bernoulli type is a universal approximator.

Formally, a restricted Boltzmann machine can be represented as a fully connected bipartite graph $G = (X, U)$, where X - vertex set - stochastic neurons, U - edges set -

synaptic connections.

$$\begin{cases} X = X_1 \cup X_2, X_1 \cap X_2 = \emptyset \\ U = \{u = (x_1, x_2) | \forall x_1 \in X_1, \forall x_2 \in X_2\} \end{cases} \tag{3}$$

vertices of subset X_1 - set the neurons of the input layer, X_2 - output layer neurons.

The number of neurons in the input layer is determined by the size of the input image. Since the input number is eight-bit, then in the hidden layer (for its lossless coding) 8 binary neurons are required. Therefore, the number of neurons in the hidden and the input layer to compress the image in k times was determined by the formula:

$$\begin{cases} |X_1| = nm \\ |X_2| = \frac{8nm}{k} \end{cases} \tag{4}$$

To each vertex of the input layer we assign a set of parameters $VB = \{b\}$ - vertex offsets and $\sigma = \{\sigma\}$ - vertex variances, and to the vertices of the output layer - set of parameters $HB = \{g\}$ - vertex offsets. The sizes of the sets are equal respectively

$$|VB| = |\sigma| = |X_1|, |HB| = |X_2| \tag{5}$$

Each edge connecting a pair of vertices of the input and output layers will be assigned a set of parameters $W = \{w\}$ - the weights of the edges.

The size of the set is equal to the following value

$$|W| = |X_1||X_2| \tag{6}$$

Thus, the described family of neural networks can be defined by four types of parameters (see Fig 1):

$$RBM = (W, VB, \sigma, HB) \tag{7}$$

In this case, a specific Boltzmann machine is specified by fixing the values of the parameters from the above sets. In fact, at the training stage, the neural network is set up for an objective task.

Thus, the restricted Boltzmann machine functions in two main modes. In the learning mode, the values of the parameters are calculated, which determine in the family of machines a specific instance for solving an applied problem. In the working mode, compression (encoding) of the image and its recovery (decoding) are performed.

4 Training a restricted Boltzmann machine

Consider the approach to the problem of learning a restricted Boltzmann machine using the annealing method.

The following algorithm is proposed that implements the ideology of the method.

At the preliminary stage, initialization (setting the initial values) of the parameters (W, VB, HB, σ), initial temperature T0 and cooling coefficient α is performed.

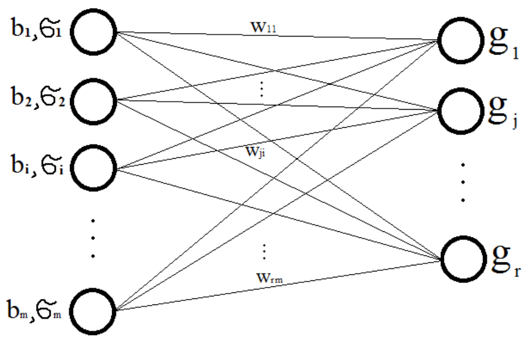

Figure 1: Gauss-Bernoulli RBM architecture

The main stage of the learning algorithm implements a procedure for sequentially updating the values of the specified parameters using a certain quality functional.

Describe the process to update settings in more detail. For simplicity, consider it on the example of the set of parameters W. For other sets, this procedure is identical.

To the set of parameters W, we associate a segment Lw of length l. After that, each element of the set W is sequentially placed in the center of the given segment. To determine the direction of change of parameter values, we generate a base random variable. If it is more than 0.5, then the value of the parameter increases, otherwise it decreases.

New parameter values are defined as follows. A random permutation is generated, the number of elements of which is equal to the number of elements of the set W. We order the elements of the set W in accordance with the permutation and change the values of the first Wp elements of the set. The new value of the parameter is determined as a result of the implementation of a uniformly distributed random variable on the segment, the ends of which are the current value of the parameter, and the end of the segment towards which the change is made.

Similarly, actions are sequentially performed for the sets VB, HB, σ.

For newly obtained parameter values, the quality functional is calculated.

As the latter, it is proposed to use the following function:

$$F(W, VB, HB, \sigma) = \frac{1}{Nnm} \sum_{i=1}^{N} \sum_{j=1}^{nm} |x_{ij} - f^{-1}(y_{ij})| \tag{8}$$

where y_{ij} - reconstituted input signal of restricted Boltzmann machine, f^{-1} - inverse function of the preliminary transformation of input data.

Then a decision is made to move to a new state:

- if the value of the functional is greater than the current one, then we move to the new state with probability:

$$P(y|x) = exp\left(-\frac{F(y) - F(x)}{T_i}\right) \tag{9}$$

where x - current state, y - state selected for transition, F - minimized objective function, T_i - temperature of i-th iteration

- else we always move to a new state.

Further cooling takes place according to the rule:

$$T_{i+1} = \alpha T_i \tag{10}$$

After cooling, the obtained solution is checked for optimality:

the solution is optimal if during the last S iterations there has not been a transition to a new state. If the solution obtained is optimal, then:

- stop algorithm,

- otherwise go to the next iteration.

Thus, as a result of the learning process in the parameter space, describing the family of restricted Boltzmann machines, their specific values are fixed. This builds a specific instance of the machine that best suits the objective task. And in this case, you can go directly to the process of compressing images.

5 Experiments and results

The effectiveness of the approach was tested experimentally. For the compression problem, STL-10 dataset from the repository of Stanford University [4] was used. This is one hundred thousand unmarked color images of 96x96 pixels. Each image is described by 27648 integers (in the range from 0 to 255), specifying the content of red, green and blue colors [5].

The experiments were carried out on a computer with a 4 core processor and a nvidia 1060 3gb video card.

Measurement of operations time was performed using the function gettimeofday.

Before training, each input image was represented by three components: Red, Green, Blue.

Received three data array dimension nxmxN, where nxm – image size, N - number of objects in the dataset. Arrays were divided into blocks of lower dimension. The blocks described the image from left to right, top to bottom.

Each block was processed by a separate neural network. This gave a number of advantages:

- learning machines are completely independent. Each machine has its own parameters and its own training data. This allowed parallelizing the learning process;

- decomposition reduced the total number of tunable parameters, which reduced the amount of computation.

For each i,j-th block element (i=1,n;j=1,m) arithmetic average was calculated. Then it was subtracted from the initial values of the i, j-th element. As a result, the data preprocessing was as follows:

$$f(x_{lij}) = \frac{1}{3}(x_{lij} - \mu_{ij}), l = \overline{1, N} \tag{11}$$

where μ_{ij} - arithmetic average of coordinate (i;j).

For the experiments, the images were divided into blocks of 12x12. The experiments were carried out within the framework of one block by splitting it into mini blocks of

16 elements. To calculate the training time, the experimentally obtained time was multiplied by the number of blocks of 12x12 in the original images.

Each mini-block was processed on a separate Boltzmann machine. The input layer of the machine contained the number of neurons equal to the size of the mini-block.

As a result of the experiments, the following main results were obtained.

At 32-fold compression, the loss was 15%, and the estimated training time was 2.2 days. These results are comparable to similar results obtained using gradient methods: 2 days and 20% loss [7]. This suggests the possibility of practical use of the approach described in this paper for solving the problem of color image compression.

6 Conclusion

The article deals with the problem of color image compression. A statistical approach based on a restricted Boltzmann machine is investigated. An original algorithm for learning the neural network is proposed. The algorithm implements the annealing method. The effectiveness of training is based on the use of data parallelization technology.

It was experimentally shown that with proper selection of parameters in the learning algorithm and the use of parallelization technology, the learning process is carried out in a reasonable time [7]. This allows us to make an assumption about the prospects of using the idea of random search in the development of deep learning algorithms. With the rapid development of computer technology, such algorithms can be effectively used to train neural networks and, in particular, for a restricted Boltzmann machine.

References

[1] MohanaPriya P., Shalinie S. M. (2017). Restricted Boltzmann machine - based cognitive protocol for secure routing in software defined wireless networks. *IET Networks*. Vol. 6, No. 6, pp. 162–168.

[2] Lattner S., Grachten M., Widmer G. (2016). Imposing Higher-Level Structure in Polyphonic Music Generation using Convolutional Restricted Boltzmann Machines and Constraints. *J. Creative Music Systems*.

[3] Adachi S.H., Henderson M.P. (2015). Application of Quantum Annealing to Training of Deep Neural Networks. *arXiv*.

[4] STL-10 dataset link: academictorrents.com/details/a799a2845ac29a66c07cf74e2a2838b6c5698a6a

[5] STL-10 dataset description link: stanford.edu/ acoates//stl10/

[6] CUDA toolkit link: developer.nvidia.com/cuda-downloads

[7] Zhu H., Akrout M., Zheng B., Pelegris A., Jayarajan A., Phanishayee A., Schroeder B., Pekhimenko G. (2018). Benchmarking and Analyzing Deep Neural Network Training. *arXiv*.