Bull. Nov. Comp. Center, Comp. Science, 44 (2020), 31–35

Quantization of weights in capsule neural networks

M. Tarkov, M. Nevaev

Abstract. The influence of the weight quantization levels number on a capsule neural network functioning quality is studied. The network is tested on the MNIST dataset recognition. It is shown that to achieve the performance of the network with continuous weights it is enough to have 16 levels of uniform quantization and 8 levels of exponential quantization. The quantized weights of the capsule neural network can be successfully implemented based on multilevel memristors.

Keywords: capsule neural network, weights quantization, multilevel memristor.

Introduction

In the pattern recognition problems, the convolutional deep belief neural networks (NN) architecture is often used [1]. Using the convolution and the error back propagation, which are often applicable in the convolutional NN, it is possible to extract patterns from the data, due to which the trained model copes well with classification tasks. Using the data pooling operation, data complexity can be significantly reduced, which eliminates minor details and reduce the number of weights used. But this architecture is not without flaws.

In 2014, Jeffrey Hinton noted that pooling operations are a weak point of this architecture. It turns out that with the data dimensionality decrease, the mutual spatial relationship between separate objects in an image is lost. It is a negative factor for specific data (for example, the face on which the eyes and mouth are mixed up will be recognized like any other face). Another disadvantage is that when objects rotate in the image, the convolutional NN ceases to recognize them, which makes it difficult to simulate and increase the already large number of weights so that the trained model takes into account the patterns of rotating objects. In [2], the capsule NNs architecture is proposed, in which, instead of neurons, the main modules are the so-called capsules. Such architectures possess not only the advantages of the convolutional neural network architecture, but also lack its disadvantages.

The hardware NNs implementation requires a large volume of memory to store the neuron layer weight matrix and is expensive. This problem solution is simplified when using a device called a memristor (memory resistor) as a memory cell. The memristor was theoretically predicted in 1971 by Leon Chua [3]. The first physical implementation of the memristor was demonstrated in 2008 by the Hewlett Packard laboratory in the form of a thin-film TiO_2 structure [4]. The memristor behaves like a synapse: it "remembers" the full electric charge that has passed through it. The memristor-based memory can achieve a degree of integration of 100 Gbits/cm^2, several times higher than that based on the flash memory technology. These unique properties make the memristor a promising device for creating mass-parallel neuromorphic systems.

Binary are called memristors that implement two conductivity values. Multilevel are called memristors that implement many discrete levels of conductivity (the number of levels can reach tens or hundreds). Binary and multilevel memristors [5–7] are based on the filament switching mechanism and are more widespread than the analog memristors, in which the conductivity can be continuously changed, but for which materials are not nearly so common and which require a more complex implementation process. Multilevel memristors are tolerant to statistical fluctuations as compared to the analog memristors. The use of binary and multilevel memristors for setting the weighting coefficients of the capsule NNs makes it urgent to study the influence of the number of weights quantization levels on the pattern recognition quality by such networks.

1. The capsule neural network architecture

A capsule is a group of neurons whose activity vector represents the parameters for the implementation of a particular type of an object or a part of an object. The activity vector length is used to represent the probability of the existence of the entity and its orientation to represent the implementation parameters. Active capsules at the same level using transformation matrices predict the parameters for the implementation of higher level capsules. When several predictions match, a higher level capsule is activated.

The multilayer capsule system is much better than the convolutional network, it recognizes strongly overlapping numbers in the MNIST set. To achieve these results, an iterative matching routing mechanism is used: a lower-level capsule sends its output data to those higher-level capsules whose activity vectors have a large scalar product with a prediction coming from a lower-level capsule.

The CapsNet architecture [2] has two convolutional levels (layers) and one fully connected level. The first layer (Conv1) has 256 convolution cores with step 1 and ReLU activation. This layer converts the pixel intensity to the activity of local feature detectors, which are then used as input for the primary capsules. Primary capsules are the lowest level of multidimensional objects, and from the inverse graphical point of view, the activation of primary capsules corresponds to the inversion of the rendering process.

The second layer (PrimaryCapsules) is a convolutional capsule layer with 32 channels of convolutional 8D capsules (i.e., each primary capsule contains 8 convolutional modules with a core and step 2). Each primary capsule exit sees the outputs of all Conv1 units whose receptive fields overlap with the location of the capsule center. The total PrimaryCapsules have capsule exits (each out is an 8D vector), and all capsules in the grid share their weights with each other. One can consider PrimaryCapsules as a convolutional layer with a squashing nonlinear function at the output.

The final layer (DigitCaps) has one 16D capsule per digit class, and each of these capsules receives input from all the capsules in the layer below.

2. The weights quantization algorithms

Methods for quantizing the weights of a trained NN are described below. The uniform quantization algorithm:

1. Find the maximum $W_{\max}$ of the weight coefficient module of the trained capsule network.
2. Find the quantum (jump) of the weight $\Delta = \frac{W_{\max}}{L-1}$, where $L \geq 2$ is the number of quantization levels.
3. Convert all W_{ij} weights (i, j are the identifiers of connected capsules) according to the rule: if $(k-1)\Delta < |W_{ij}| < k\Delta$, then W_{ij} gets the value $(k-1)\Delta\,\mathrm{sign}(W_{ij})$, $k = 1, \ldots, L$, where

$$\mathrm{sign}(a) = \begin{cases} 1, & a > 0 \\ -1, & a \leq 0. \end{cases}$$

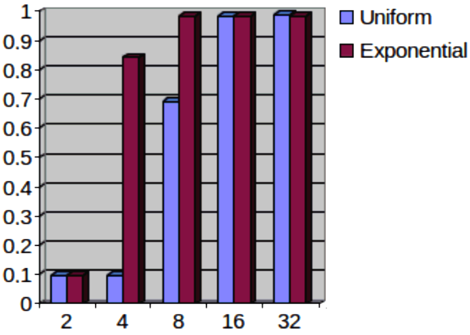
The exponential quantization algorithm:

1. Find the maximum $W_{\max}$ of the weight coefficient module of the trained capsule network.
2. Find the quantum (jump) of the weight $\Delta = \frac{W_{\max}}{2^L - 1}$, where $L \geq 2$ is the number of nonzero quantization levels.
3. Convert all the weights W_{ij}, according to the rule:
   a) if $|W_{ij}| < \Delta$, then $|W_{ij}| \leftarrow 0$;
   b) if $(2^k - 1)\Delta < |W_{ij}| < 2^k\Delta$, then $|W_{ij}| \leftarrow (k-1)\Delta\,\mathrm{sign}(W_{ij})$, $k = 1, \ldots, L$.

As an example, for $L = 4$ we obtain the quantization levels 0, $\frac{W_{\max}}{8}$, $\frac{W_{\max}}{4}$, $\frac{W_{\max}}{2}$, $W_{\max}$.

3. Experiments

In order to determine the NN accuracy loss during quantization the weights of the capsule NN trained on the set of handwritten digits MNIST [8] were quantized in all layers using the above algorithms for various numbers of quantization levels. The quantization algorithms are implemented in Python using the Pytorch machine learning library.



The figure shows a diagram of the dependence of the capsule network recognition quality on the quantization levels number and the quantization algorithm (uniform and exponential). From the diagram it follows that:

- The recognition quality of the capsule network with continuous weights (about 98 %) is achieved with the network with uniform weights quantization when the quantization levels number is equal to 16 or more.
- With the exponential quantization, this requires a smaller number of levels (8 or more).

Conclusion

In this paper, the influence of the number of quantization levels of a capsule neural network weights on the NN quality functioning during the recognition of the MNIST dataset digits images is studied. Two methods of weights quantization are considered — uniform and exponential. It is shown that to achieve the quality of the network with continuous weights (98 %), it is enough to have 16 levels of the uniform quantization, and 8 levels of the exponential quantization. Thus, the weights of the capsule neural network can be successfully implemented based on the multilevel memristors with the above levels numbers.

References

[1] LeCun Y., Bengio Y., Hinton G. Deep Learning // Nature. — 2015. — Vol. 521. — P. 436–444.

[2] Sabour S., Frosst N., Hinton G. Dynamic routing between capsules // Proc. 31st Conf. on Neural Information Processing Systems (NIPS 2017). — Long Beach, CA, USA, 2017.

[3] Chua L. Memristor — the missing circuit element // IEEE Trans. Circuit Theory. — 1971. — Vol. 18. — P. 507–519.

[4] Strukov D.B., Snider G. S., Stewart D.R., Williams R. S. The missing memristor found // Nature. — 2008. — Vol. 453. — P. 80–83.

[5] He W., Sun H., Zhou Y., et al. Customized binary and multi-level HfO_2 x-based memristors tuned by oxidation conditions // Scientific Reports. — 2017. — Vol. 7. — 10070.

[6] Yu S., Gao B., Fang Z., et al. A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation // Adv. Mater. — 2013. — Vol. 25. — P. 1774–1779.

[7] Tarkov M. S. Construction of a Hamming network based on a crossbar with binary memristors // Prikladnaya Diskretnaya Matematika. — 2018. — No. 40. — P. 105–113 (In Russian).

[8] URL: http://yann.lecun.com/exdb/mnist/.