МАТЕМАТИЧЕСКОЕ
И ИМИТАЦИОННОЕ
МОДЕЛИРОВАНИЕ
Имитационное моделирование

I. V. Tsimokhin
(F. Scorina GSU, Gomel)

APPLICATION OF CONVOLUTIONAL NEURAL NETWORKS FOR PLANKTON CLASSIFICATION

Convolutional neural network (CNN) is a type of feed-forwarding artificial neural network. CNNs are biologically inspired: it is known that cats have a complex cell structure in their visual cortex with two types of neural cells. The idea of CNNs is to use two similar types of layers: convolutional and pooling layers. Convolutional layers apply several learnable filters, known as kernels, to whole input with overlapping. By doing this CNNs can detect kernel specified pattern in input. Pooling layers apply specified nonlinear function to not overlapping sets of input. CNNs are known to be more accurate for image recognition problems [1], so CNNs was chosen for the problem of plankton classification.

The problem of plankton classification was to classify black-and-white images in 121 classes. There were given train dataset with images labeled with classes. Most of those classes match with biological classification of plankton; but there were also some classes for different types of image artefacts. As images had different resolution data preprocessing needed. At first stages it only included simple image resizing. Later more complex preprocessing were added, which included image rotation around mass centre, applying Canny edge detector.

Neural networks were created with PyBrain [2] library, which allows to create complex connections between layers, including connections with shared weight, which can be used as convolutional kernels for CNNs. Images were processed with Python Image Library (PIL). For preventing code duplication some additional modules were written.

For this problem different network configurations and different image preprocessing methods were tested. As learning CNNs requires complex calculations to be done, only some networks were learnt to give result better than

random value would give. That networks have a convolutional layer with 5 kernels, a pooling layer, a convolutional layer with 4 kernels, a pooling layer; the output of the last layer was connected into fully connected layer, which then connected to output. For fully connected and convolutional layers sigmoid function was used, and for pooling layers maximum function was used.

BIBLIOGRAPHY

1 Gradient-based learning applied to document recognition / Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner // Proceedings of the IEEE. – 1998. – Vol. 11. – P. 2278–2324.

2 PyBrain / Tom Schaul [et al.] // Journal of Machine Learning Research. – 2010. – Vol. 11. – P. 743–746.