Systems and means of deep learning for classification problems [*]

O. Yu. Bakhteev[1], M. S. Popova[2], V. V. Strijov[3]

Annotation: The paper provides a guidance on deep learning net construction and optimization using GPU. The paper proposes to use GPU-instances on the cloud platform Amazon Web Services. The problem of time series classification is considered. The paper proposes to use a deep learning net, i.e. a multilevel superposition of models, belonging to the following classes: Restricted Boltzman Machines, autoencoders and neural nets with softmax-function in output. The proposed method was tested on a dataset containing time segments from mobile phone accelerometer. The analysis of relation between classification error, dataset size and superposition parameter amount is conducted.

Keywords: time series classification; deep learning; model superposition; autoencoder; restricted Boltzmann machine; cloud service

References

[1] Cho, K. 2014. Doctoral dissertation Foundations and Advances in Deep Learning. Espoo: Aalto University. 277 p.

[2] Langkvist, M., L. Karlsson and A. Loutfi. 2014. A review of unsupervised feature learning for time-series modelling. *Pattern Recognition Letters*. 42(1): 11-24.

[3] Desell, T., S. Clachar, J. Higgins and B. Wild. 2015. Evolving Deep Recurrent Neural Networks Using Ant Colony Optimization. *Evolutionary Computation in Combinatorial Optimization*. 9026: 86-98.

[4] Popova, M. S. and V. V. Strijov 2015. Building superposition of deep learning neural networks for solving the problem of time series classification. *Systems and Means of Informatics*. 25(3): 60-77.

[1]Moscow Institute of Physics and Technology, bakhteev@phystech.edu
[2]Московский физико-технический институт, maria_popova@phystech.edu
[3]Dorodnicyn Computing Centre, CSC IC RAS, strijov@ccas.ru

[5] Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research.* 15: 1929-1958.

[6] Wager, S., S. Wang and P. Liang. 2013. Dropout Training as Adaptive Regularization. *Advances in Neural Information Processing Systems.* 26: 351-359.

[7] Gal, Y. and Z. Ghahramani 2015. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *arXiv preprint arXiv:1506.02142.* Available at: http://arxiv.org/abs/1506.02142 (accessed November 25, 2015).

[8] Wan, L., M. Zeiler, S. Zhang, Y. LeCun and R. Fergus. 2013. Regularization of Neural Networks using DropConnect. *Proceedings of the 30th International Conference on Machine Learning (ICML-13).* 1058-1066.

[9] Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow and R. Fergus. 2014. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199.* Available at: http://arxiv.org/abs/1312.6199 (accessed November 25, 2015).

[10] Goodfellow, I. J., Q. V. Le, A. M. Saxe, H. Lee and A. Y. Ng. 2009. Measuring Invariances in Deep Networks. *Advances in Neural Information Processing Systems.* 22: 646–654.

[11] Raiko, T., H. Valpola and Y. LeCun. 2012. Deep Learning Made Easier by Linear Transformations in Perceptrons. *Journal of Machine Learning Research - Workshop and Conference Proceedings.* 22: 924-932.

[12] Bengio, Y., E. Laufer, G. Alain and G. Yosinski. 2014. Deep Generative Stochastic Networks Trainable by Backprop. *Proceedings of The 31st International Conference on Machine Learning.* 226–234.

[13] Ioffe, S. and C. Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv preprint arXiv:1502.03167.* Available at: http://arxiv.org/abs/1502.03167 (accessed November 25, 2015).

[14] Li, Z., C. Chang, F. Liang, T. S. Huang, C. Cao and J. R. Smith. 2013. Learning Locally-Adaptive Decision Functions for Person Verification. *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on.* 3610-3617.

[15] Sutskever, I., G. Hinton and G. Taylor. 2009. The Recurrent Temporal Restricted Boltzmann Machine. *Advances in Neural Information Processing Systems.* 21: 1601-1608.

[16] Fischer, A. and C. Igel. 2014. Training Restricted Boltzmann Machines: An Introduction. *Pattern Recognition.* 47: 25-39.

[17] Socher, R., E. H. Huang, J. Pennington, A. Y. Ng and C. D. Manning. 2011. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. *Advances in Neural Information Processing Systems 24*. 801-809.

[18] Shu, M. and F. Fyshe. 2013. Sparse Autoencoders for Word Decoding from Magnetoencephalography. *Proceedings of the third NIPS Workshop on Machine Learning and Interpretation in NeuroImaging (MLINI)*. Available at: http://www.cs.cmu.edu/~afyshe/papers/SparseAE.pdf (accessed November 25, 2015).

[19] Vincent, P., H. Larochelle, I. Lajoie, Y. Bengio and P. Manzagol. 2013. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *The Journal of Machine Learning Research*. 11: 3371-3408.

[20] Kwapisz, J. R., G. M. Weiss and S. Moore. 2010. Activity recognition using cell phone accelerometers. *SIGKDD Explorations*. 12(2): 74-82.

[21] Bastien, F., P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley and Y. Bengio. 2012. Theano: new features and speed improvements. *Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop*. Available at: http://arxiv.org/pdf/1211.5590v1.pdf (accessed November 25, 2015).

[22] Bergstra, J., O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley and Y. Bengio. 2010. Theano: a CPU and GPU Math Expression Compiler. *Proceedings of the Python for Scientific Computing Conference (SciPy)*. 3-11.

[23] Goodfellow, I. J., D. Warde-Farley, P. Lamblin, V. Dumoulin, M. Mirza, R. Pascanu, J. Bergstra, F. Bastien and Y. Bengio. 2013. Pylearn2: a machine learning research library. *arXiv preprint arXiv:1308.4214*. Available at: http://arxiv.org/abs/1308.4214 (accessed November 25, 2015).

[24] Dieleman, S., J. Schl?ter, C. Raffel et al. 2015. Lasagne: First release. Aug. 2015. Available at: http://dx.doi.org/10.5281/zenodo.27878 (accessed November 25, 2015).

[25] Nickolls, J., I. Buck, M. Garland and K. Skadron. 2008. Scalable Parallel Programming with CUDA. *ACM Queue*. 6(2): 40-53.

[26] Stone, J. E., D. Gohara and G. Shi. 2010. OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems. *Journal IEEE Design & Test*. 12(10): 66-73.

[27] Erhan, D., Y. Bengio, A. Courville, P. Manzagol and P. Vincent. 2010. Why Does Unsupervised Pre-training Help Deep Learning? *The Journal of Machine Learning Research*. 11: 625-660.

[28] Duan, K., S. S. Keerthi, W. Chu, S. K. Shevade and A. N. Poo. 2003. Multi-Category Classification by Soft-Max Combination of Binary Classifiers. *The fourth international workshop on multiple classifier systems*. 125-134.

[29] Cho, K., T. Raiko and A. Ilin. 2013. Gaussian-Bernoulli Deep Boltzmann Machine. *The 2013 International Joint Conference on Neural Networks (IJCNN)*. 1-7.

[30] Hinton, G. E., S. Osindero and Y. Teh. 2006. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*. 18: 1527–1554.

[31] Available at: http://stackoverflow.com/questions/2549035/do-you-get-charged-for-a-stopped-instance-on-ec2 (accessed November 25, 2015).

[32] Available at: https://us-west-2.console.aws.amazon.com/console/ (accessed November 25, 2015).

[33] Bishop, C. M., eds. 2006. Pattern Recognition and Machine Learning. Springer-Verlag New York. 738 p.

[34] Cassioli, A., D. D. Lorenzo, M. Locatelli, F. Schoen and Sciandrone, M. 2012. Machine Learning for Global Optimization. *Computational Optimization and Applications*. 1(1): 279-303.

[35] Bakhteev, O. Yu. 2016. Deep learning software. Available at: https://svn.code.sf.net/p/mlalgorithms/code/Group074/Bakhteev2015TheanoCuda/code/ (accessed November 25, 2015).

[36] Popova, M. S. 2015. Deep learning software. Available at: https://svn.code.sf.net/p/mlalgorithms/code/Group174/Popova2015DeepLearning/ (accessed November 25, 2015).