

Abstract

This paper solves the problem of time-series classification using deep learning neural networks. The paper proposes to use a multilevel superposition of models belonging to the following classes of neural networks: two-layer neural networks, Boltzmann machines and autoencoders. Lower levels of superposition extract from noisy data of high dimensionality informative features, while the upper level of the superposition solves the problem of classification based on these extracted features. The proposed model has been tested on two samples of physical activity time series. The classification results obtained by proposed model in computational experiment were compared with the results which were obtained on the same datasets by foreign authors. The study showed the possibility of using deep learning neural networks for solving problems of time-series physical activity classification.

Keywords: *classification; time series; deep learning neural networks; model superposition; feature extraction.*

¹1 — Moscow Institute of Physics and Technology, maria_popova@phystech.edu
2 — Moscow Institute of Physics and Technology, strijov@ccas.com

Building superposition of deep learning neural networks for solving the problem of time series classification

M. Popova¹, V. Strijov²

May 24, 2015

References

- [1] *Längkvist, Martin and Karlsson, Lars and Loutfi, Amy* A Review of Unsupervised Feature Learning and Deep Learning for Time-Series Modeling // Pattern Recognition Letters, 2014. Vol. 42. No. 6. P. 11–24.
- [2] *Alex Nanopoulos and Rob Alcock and Yannis Manolopoulos* Feature-based Classification of Time-series Data // International Journal of Computer Research, 2001. Vol. 10. P. 49–61.
- [3] *Keogh, E., Pazzani, M* A simple dimensionality reduction technique for fast similarity search in large time series databases // In Proceedings of Pacific- Asia Conf. on Knowledge Discovery and Data Mining, P. 122–133.
- [4] *Weston J., Mukherjee S., Chapelle O., Pontil M., Poggio T., Vapnik V.* Feature selection for SVMs // In: Advances in neural information processing systems, MIT Press, 2013. PP. 668–674.
- [5] *Estévez, Pablo A. and Tesmer, Michel and Perez, Claudio A. and Zurada, Jacek M.* Normalized Mutual Information Feature Selection // IEEE Transactions on Neural Networks, 2009. Vol. 20. No. 2. P. 189–201.
- [6] *Fabian Mörchen.* Time series feature extraction for data mining using DWT and DFT // OAI-PMH server at citeseerx.ist.psu.edu, 2003. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.87.2037>.
- [7] *G. E. Hinton, and R. R. Salakhutdinov* Reducing the dimensionality of data with neural networks // Science, 2006. Vol. 313. No. 5786. P. 504–507.

[†]This paper was supported by RFBR grant 13-07-00709

- [8] *Andre Stuhlsatz, Jens Lippel, and Thomas Zielke* Feature Extraction With Deep Neural Networks by a Generalized Discriminant Analysis // IEEE Transactions on Neural Networks and Learning Systems, 2012. Vol. 23. No. 4. P. 596–608.
- [9] *Yuanfang Ren and Yan Wu* Convolutional deep belief networks for feature extraction of EEG signal // International Joint Conference on Neural Networks (IJCNN) 2014, Beijing, China, 6-11 July 2014. P. 2850–2853.
- [10] *Yan Xu, Tao Mo, Qiwei Feng, Peilin Zhong, Maode Lai, Chao Chang* Deep learning of feature representation with multiple instance learning for medical image analysis // In Proceedings of International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014. P. 1626–1630.
- [11] *G. E. Hinton, Li Deng, Dong Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, B. Kingsbury.* Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups // IEEE Signal Process. Mag., 2012. Vol. 29. No. 6. P. 82–97.
- [12] *Hüsken, M., Stagge, P.* Recurrent Neural Networks for Time Series Classification // Neurocomputing, 2003. Vol. 50. P. 223–235.
- [13] *Wulsin, D., Gupta, J., Mani, R., Blanco, J., Litt, B.* Modeling electroencephalography waveforms with semi-supervised deep belief nets: faster classification and anomaly measurement // Journal of Neural Engineering, 2011. Vol. 8. P. 1741–2552.
- [14] *Geoffrey E. Hinton.* A Practical Guide to Training Restricted Boltzmann Machines // Neural Networks: Tricks of the Trade – Second Edition. Springer, 2012. P. 599–619.
- [15] *Yoshua Bengio.* Learning Deep Architectures for AI // Foundations and Trends in Machine Learning, 2009. Vol. 2. No. 1. P. 1–127.
- [16] *D. Anguita, A. Ghio, L. Oneto, X. Parra, J. Luis Reyes-Ortiz* Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine // Ambient Assisted Living and Home Care – 4th International Workshop, IWAAL 2012, Vitoria-Gasteiz, Spain, December 3-5, 2012. Proceedings. Springer, 2012. Vol. 7657. P. 216–223.
- [17] *Kwapisz J. R., Weiss G. M., Moore S.* Activity recognition using cell phone accelerometers // SIGKDD Explorations, 2010. Vol. 12. No 2. P. 74–82.
- [18] *Ian Nabney* NETLAB: algorithms for pattern recognitions // Springer-Verlag, 2002.
- [19] <https://github.com/zellyn/deeplearning-class-2011/tree/master/ufdl/library>
- [20] <http://www.mathworks.com/matlabcentral/fileexchange/42853-deep-neural-network>
- [21] <https://github.com/rasmusbergpalm/DeepLearnToolbox>

Построение нейронных сетей глубокого обучения для классификации временных рядов*

М. С. Попова¹, В. В. Стрижов²

Аннотация: Статья посвящена решению задачи классификации временных рядов с использованием нейронных сетей глубокого обучения. Предлагается использовать многоуровневую суперпозицию моделей, относящихся к следующим классам нейронных сетей: двухслойные нейронные сети, машины Больцмана и автокодировщики. Нижние уровни суперпозиции выделяют из зашумленных данных высокой размерности информативные признаки, а верхний уровень по этим признакам решает задачу классификации. Предложенная модель была протестирована на двух выборках временных рядов физической активности человека. Результаты классификации, полученные предлагаемой моделью в ходе вычислительного эксперимента, сравнивались с результатами, которые были получены на этих же данных в работах зарубежных авторов. Исследование показало возможность применения нейронных сетей глубокого обучения к решению прикладных задач классификации физической активности человека.

Ключевые слова: классификация; временные ряды; нейронные сети глубокого обучения; суперпозиция моделей; выделение признаков.

1 Введение

В данной работе рассматривается задача классификации временных рядов. Под временным рядом понимается упорядоченный набор измерений некоторой величины, в котором каждое измерение соответствует определенному моменту времени. Временные ряды обладают уникальными свойствами [1], которые усложняют работу с ними. Примерами таких свойств служат высокая размерность и зашумленность данных. Существуют методы снижения размерности и фильтрации шумов [2-4]. Однако при

*Работа выполнена при финансовой поддержке РФФИ проект 13-07-00709

¹Московский физико-технический институт, maria_popova@phystech.edu

²Московский физико-технический институт, strijov@ccas.com

снижении размерности пространства данных и выделении новых признаков снижается точность описания объектов. Ввиду этого возникает задача построения нового признакового пространства меньшей размерности, в котором признаки наиболее полно описывали бы исходные временные ряды.

Существует два основных способа построения признакового пространства – это экспертное и автоматическое выделение признаков. Первый способ заключается в экспертном назначении базовых функций и требует индивидуального подхода к каждой отдельной задаче, т. к. одни и те же базовые функции не могут достаточно точно описывать данные разной природы. Второй способ является более универсальным, и некоторые методы, успешно применяемые для предобработки временных рядов, описаны в [5-7]. В данной работе предлагается использовать нейронные сети глубокого обучения для выделения информативных признаков [7-9] и классификации временных рядов.

Нейронные сети глубокого обучения применяются для решения задач распознавания изображений [10] и речи [11], однако есть работы, показывающие возможность их применения к предобработке и классификации временных рядов [12, 13]. Предлагается использовать многоуровневую суперпозицию [7] ограниченных машин Больцмана [14, 15], автокодировщиков [15] и двухслойных нейронных сетей для машинного извлечения признаков и классификации временных рядов. Все уровни суперпозиции, кроме последнего, обучаются по принципу «обучение без учителя» и участвуют в построении признакового пространства. Последний уровень суперпозиции обучается «с учителем» и решает задачу классификации по признакам, выделенным на нижних уровнях суперпозиции. Такая конструкция позволяет снизить размерность пространства признаков, информативно описывающих исследуемое явление, а затем решить задачу классификации, основываясь на небольшом числе выделенных признаков. В данной работе поставлен вычислительный эксперимент на двух выборках – временных рядах акселерометра и временных рядах акселерометра и гироскопа мобильного телефона. Результаты классификации, полученные с помощью предложенной модели, сравнивались с результатами, полученными на тех же данных в других работах [16, 17]. Исследование показало возможность применения нейронных сетей глубокого обучения к решению прикладной задачи классификации физической активности человека.

2 Постановка задачи

Дана выборка $\mathcal{D} = \{(\mathbf{x}_i, t_i), i = 1, \dots, N\}$, состоящая из N пар объект–ответ. Объектами \mathbf{x}_i являются сегменты временного ряда – $\mathbf{x}_i \in \mathbb{R}^n$. Каждый объект принадлежит одному из M классов, метки классов $t_i \in \{1, \dots, M\}$. Выборка \mathcal{D} разделена на две подвыборки – обучающую \mathcal{L} и контрольную \mathcal{T} .

Моделью классификации назовем суперпозицию функций

$$\mathbf{f}(\mathbf{x}, \mathbf{w}) = \boldsymbol{\mu}_1(\boldsymbol{\mu}_2(\dots \boldsymbol{\mu}_K(\mathbf{x}))) : \mathbb{R}^n \rightarrow [0, 1]^M, \quad (1)$$

где μ_k , $k \in \{1, \dots, K\}$, – модели из класса нейронных сетей с соответствующими векторами параметров \mathbf{w}_k , $k \in \{1, \dots, K\}$. Под вектором параметров модели \mathbf{f} будем понимать вектор $\mathbf{w} = [\mathbf{w}_1, \dots, \mathbf{w}_K]^\top$.

Компоненты вектора $\mathbf{f}(\mathbf{x}, \mathbf{w})$ – это вероятности отнести объект \mathbf{x}_i к соответствующему классу:

$$\mathbf{f}(\mathbf{x}, \mathbf{w}) = \begin{bmatrix} p(y_1 = 1 | \mathbf{x}; \mathbf{w}) \\ p(y_2 = 2 | \mathbf{x}; \mathbf{w}) \\ \vdots \\ p(y_M = M | \mathbf{x}; \mathbf{w}) \end{bmatrix}. \quad (2)$$

Функцию ошибки S на некоторой подвыборке \mathcal{K} исходной выборки \mathcal{D} определим следующим образом:

$$S(\mathbf{w} | \mathcal{K}) = -\frac{1}{|\mathcal{K}|} \sum_{i=1}^{|\mathcal{K}|} \sum_{\xi=1}^M [y_\xi = \xi] \log p(y_\xi = \xi | \mathbf{x}_i, \mathbf{w}). \quad (3)$$

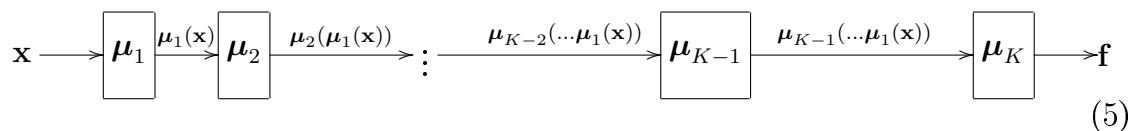
Для решения задачи классификации выборки \mathcal{D} с помощью модели нужно оптимизировать ее параметры \mathbf{w} . Для этого требуется решить задачу минимизации функции ошибки на обучающей выборке:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} S(\mathbf{w} | \mathcal{L}). \quad (4)$$

Для дополнительной оценки качества классификации будем вычислять значения функционала AUC (area under curve) на контрольной выборке для каждого класса по принципу один против всех и визуализировать полученные результаты с помощью ROC-кривых.

3 Описание алгоритма

Согласно (1) предлагаемая модель представляет из себя K -уровневую суперпозицию нейронных сетей (5). В данной работе будем рассматривать нейронные сети следующих типов: ограниченная машина Больцмана, автокодировщик и двухслойная нейронная сеть. Далее будет описан каждый из вышеперечисленных типов нейронных сетей.



3.1.1 Ограниченная машина Больцмана

Ограниченная машина Больцмана – это двухслойная нейронная сеть. Структура ограниченной машины Больцмана представляет из себя двудольный граф. Нейроны первого слоя называются видимыми и соответствуют значениям признаков объекта. Вектор

состояний нейронов первого слоя – это вектор $\mathbf{v} = \{v_l \in \mathbb{R}, l \in \text{vis}\}$, а $\text{vis} = \{1, \dots, L\}$ – множество индексов видимых нейронов. Нейроны второго слоя называются скрытыми и участвуют в выделении признаков. В данной работе нейроны скрытого слоя могут находиться в одном из двух состояний – 0 или 1. Вектор состояний – это вектор $\mathbf{h} = \{h_j \in [0, 1], j \in \text{hid}\}$, $\text{hid} = \{1, \dots, J\}$ – множество индексов скрытых нейронов. Каждая компонента h_j вектора состояний – это вероятность того, что j -й нейрон скрытого слоя находится в состоянии 1:

$$h_j = p(j\text{-й нейрон в состоянии 1}). \quad (6)$$

Таким образом, ограниченная машина Больцмана – это вероятностная модель. Вероятность пары векторов состояний нейронов (\mathbf{v}, \mathbf{h}) определяется следующим образом:

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h})), \quad (7)$$

где $E(\mathbf{v}, \mathbf{h})$ – это значение энергии пары (\mathbf{v}, \mathbf{h}) , а Z – нормировочная величина, которая определяется следующим образом:

$$Z = \sum_{\mathbf{v}, \mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h})).$$

Выражение для энергии пары $E(\mathbf{v}, \mathbf{h})$ зависит от типа объектов, которые требуется моделировать ограниченной машинной Больцмана. В данной работе используется энергия для моделирования бинарных данных:

$$E(\mathbf{v}, \mathbf{h}) = - \sum_{l \in \text{vis}} b_l^v v_l - \sum_{j \in \text{hid}} b_j^h h_j - \sum_{l, j} v_l h_j w_{lj} \quad (8)$$

или, как вариант, энергия для моделирования вещественных данных:

$$E(\mathbf{v}, \mathbf{h}) = \sum_{l \in \text{vis}} \frac{(v_l - b_l^v)^2}{2\sigma_l^2} - \sum_{j \in \text{hid}} b_j^h h_j - \sum_{l, j} \frac{v_l}{\sigma_l} h_j w_{lj}, \quad (9)$$

где σ_l – стандартное нормальное отклонение шума l -го признака, параметры b_l^v , b_j^h , $l \in \text{vis}$, $j \in \text{hid}$, – смещения нейронов видимого и скрытого слоев и $\mathbf{W} = [w_{lj}]$, $l \in \text{vis}$, $j \in \text{hid}$ – матрица весовых коэффициентов между нейронами видимого и скрытого слоев.

Вероятность входного вектора состояний \mathbf{v} , или вероятность того, что входной вектор \mathbf{v} описывается моделью (7), выражается как сумма по всем скрытым состояниям:

$$p(\mathbf{v}) = \sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h})).$$

Приведем выражения для условных вероятностей [15], которые понадобятся далее для оптимизации параметров:

$$p(\mathbf{h}|\mathbf{v}) = \frac{p(\mathbf{v}, \mathbf{h})}{p(\mathbf{v})} = \frac{\exp(-E(\mathbf{v}, \mathbf{h}))}{\sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}))} = \prod_{j \in \text{hid}} p(h_j|\mathbf{v}),$$

$$p(\mathbf{v}|\mathbf{h}) = \frac{p(\mathbf{v}, \mathbf{h})}{p(\mathbf{h})} = \frac{\exp(-E(\mathbf{v}, \mathbf{h}))}{\sum_{\mathbf{v}} \exp(-E(\mathbf{v}, \mathbf{h}))} = \prod_{l \in \text{vis}} p(v_l|\mathbf{h}).$$

Оптимизация параметров ограниченной машины Больцмана заключается в нахождении таких значений параметров $\Theta = \{w_{lj}, b_l^v, b_j^h, l \in \text{vis}, j \in \text{hid}\}$, при которых величина вероятности элементов обучающей выборки имеет наибольшее значение:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} p(\mathcal{L}; \Theta) = \prod_{\mathbf{x} \in \mathcal{L}} \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}.$$

Выражение для энергии (9) используется, если соответствующая машина Больцмана является первым уровнем суперпозиции (1). В остальных случаях используется выражение (8).

3.1.2 Алгоритм оптимизации параметров ограниченной машины Больцмана

Приведем краткое описание алгоритма оптимизации параметров ограниченной машины Больцмана с одношаговым сэмплированием Гиббса. С подробным описанием и обоснованием можно ознакомиться в [15].

Ниже приведен псевдокод одного шага алгоритма. Один цикл оптимизации ограниченной машины Больцмана состоит в повторении этого шага для всех объектов обучающей выборки. Предлагается выполнять некоторое заданное количество циклов.

Исходные параметры: \mathbf{x} – входной вектор признаков, \mathbf{W} , \mathbf{b}^v , \mathbf{b}^h – начальные значения параметров, см. (??).

Результат: \mathbf{W} , \mathbf{b}^v , \mathbf{b}^h – значения параметров после одного шага оптимизации.

Инициализация

$\mathbf{v}_1 = \mathbf{x}$;

для каждого $j \in \text{hid}$ выполнять

 вычислить вероятность $p(h_{1j} = 1 | \mathbf{v}_1)$;
 выбрать значение h_{1j} из множества $\{0, 1\}$ в зависимости от величины $p(h_{1j} | \mathbf{v}_1)$;

конец цикла

для каждого $k \in \text{vis}$ выполнять

 вычислить вероятность $p(v_{2k} = 1 | \mathbf{h}_1)$;
 выбрать значение v_{2k} из множества $\{0, 1\}$ в зависимости от величины вероятности $p(v_{2i} | \mathbf{h}_1)$;

конец цикла

для каждого $j \in \text{hid}$ выполнять

 вычислить $P(h_{2j} = 1 | \mathbf{v}_2)$;

конец цикла

$\mathbf{W} \leftarrow \mathbf{W} + \eta(\mathbf{h}_1 \mathbf{v}_1^\top - p(\mathbf{h}_{2*} = 1 | \mathbf{v}_1) \mathbf{v}_2^\top)$;

$\mathbf{b}^v \leftarrow \mathbf{a} + \eta(\mathbf{v}_1 - \mathbf{v}_2)$;

$\mathbf{b}^h \leftarrow \mathbf{b} + \eta(\mathbf{h}_1 - p(\mathbf{h}_{2*} = 1 | \mathbf{v}_2))$.

3.2 Автокодировщик

Автокодировщик представляет собой следующую суперпозицию блоков:

$$\boldsymbol{\mu} = \varphi(\mathbf{g}(\mathbf{x})),$$

где

$$\mathbf{g}(\mathbf{x}) = \sigma(\mathbf{W}_g \mathbf{x} + \mathbf{b}_g) - \text{кодирующий блок, или encoder,} \quad (10)$$

$$\varphi(\mathbf{g}(\mathbf{x})) = \sigma(\mathbf{W}_h \mathbf{g}(\mathbf{x}) + \mathbf{b}_h) - \text{декодированный блок, или decoder,} \quad (11)$$

а \mathbf{W}_g , \mathbf{W}_φ , \mathbf{b}_g , \mathbf{b}_φ – параметры автокодировщика, $\sigma(\mathbf{t}) = \frac{1}{1 + \exp(-\mathbf{t})}$ – сигмоидная функция.

Под образом вектора \mathbf{x} будем понимать вектор $\mathbf{g}(\mathbf{x}) = \sigma(\mathbf{W}_g \mathbf{x} + \mathbf{b}_g)$.

Будем предполагать, что в данной модели матрицы \mathbf{W}_g и \mathbf{W}_φ ортогональны:

$$\mathbf{W}_\varphi = \mathbf{W}_g^\top.$$

Оптимизация параметров автокодировщика $\Theta = (\mathbf{W}_g, \mathbf{W}_h, \mathbf{b}_g, \mathbf{b}_h)$ проводится так, чтобы по образу $\mathbf{g}(\mathbf{x})$ можно было восстановить образ \mathbf{x} с помощью преобразования (11) или, другими словами, чтобы выходной вектор \mathbf{f} был как можно больше похож на

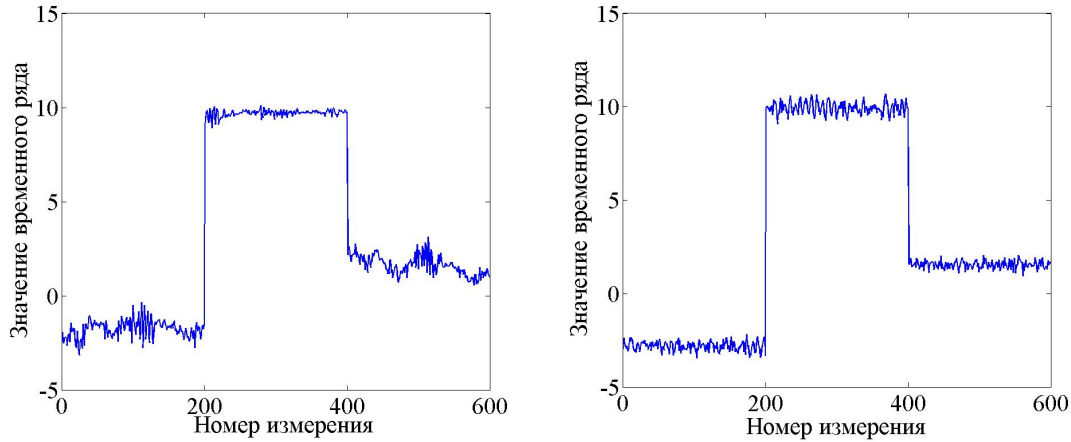


Рис. 1 Прообраз (слева) и восстановленный прообраз (справа) сегмента временного ряда

входной вектор \mathbf{x} для всех элементов обучающей выборки. Мерой сходства в данной работе выступает следующая функция:

$$S(\Theta, \mathbf{x}) = \|\mathbf{f}(\mathbf{x}|\Theta) - \mathbf{x}\|_2^2.$$

Таким образом,

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}} \frac{1}{2|\mathcal{L}|} \sum_{\mathbf{x} \in \mathcal{L}} S(\Theta, \mathbf{x}). \quad (12)$$

Для оптимизации параметров автокодировщика нужно выбрать начальное приближение для параметров каждого блока в отдельности [14], а затем настроить параметры всей модели как целого методом обратного распространения ошибки [7].

На рис. 1 изображены прообраз и восстановленный прообраз сегмента временного ряда с помощью автокодировщика.

3.3 Двухслойная нейронная сеть

Двухслойная нейронная сеть – это отображение вида

$$\mathbf{a}(\mathbf{x}) = \mathbf{W}_2^\top \tanh(\mathbf{W}_1^\top \mathbf{x}), \quad (13)$$

$$\boldsymbol{\mu}(\mathbf{x}) = \frac{\exp(\mathbf{a}(\mathbf{x}))}{\sum_{j=1}^n \exp(a_j(\mathbf{x}))}.$$

Вектор $\boldsymbol{\mu}$ интерпретируется как вектор вероятностей: μ_ξ есть вероятность того, что вектор \mathbf{x} принадлежит классу с номером ξ :

$$\boldsymbol{\mu}(\mathbf{x}) = \{\mu_\xi\}, \quad 0 \leq \mu_\xi \leq 1, \quad \sum \mu_\xi = 1, \quad \xi \in \{1, \dots, M\}.$$

Под вектором параметров двухслойной нейронной сети будем понимать $\mathbf{w} = \text{vec}(\mathbf{W}_1^T | \mathbf{W}_2^T)$, где $\mathbf{W}_1, \mathbf{W}_2$ — матрицы весов первого и второго слоя нейронной сети (13). Вектор меток классов $\mathbf{y} = [y_1, \dots, y_\xi, \dots, y_M]^T$ определим следующим образом:

$$y_\xi = \begin{cases} 1, & \text{если } \xi = \underset{\xi \in \{1, \dots, M\}}{\text{argmax}}(f_\xi), \\ 0 & \text{иначе.} \end{cases}$$

В качестве функции ошибки выберем функцию

$$S(\mathbf{w} | \mathcal{R}) = - \sum_{\mathbf{x} \in \mathcal{R}} \sum_{\xi=1}^M [y_t = 1] \ln(f_\xi(\mathbf{x}, \mathbf{w})), \quad (14)$$

максимизирующую логарифм правдоподобия мультиномиально распределенной случайной величины \mathbf{y} и заданную на подвыборке \mathcal{R} исходной выборки \mathcal{D} , t — истинная метка класса.

Оптимизация параметров двухслойной нейронной сети заключается в том, чтобы найти вектор параметров \mathbf{w} , минимизирующий функцию ошибки S по обучающей выборке. В данной работе оптимизация параметров проводится методом обратного распространения ошибки.

4 Вычислительный эксперимент

В вычислительном эксперименте использовались два набора временных рядов с датчиков мобильного устройства — WISDM [17] и HAR [16]. Цель вычислительного эксперимента заключалась в повышении точности классификации для различных типов суперпозиции моделей и сравнении их со значениями точности из работ [17, 16]. В вычислительном эксперименте также получены значения AUC для каждого из классов и построены соответствующие ROC-кривые.

4.1 Программное обеспечение

Для построения модели как суперпозиции блоков, описанных в предыдущем разделе, было реализовано программное обеспечение на языке MatLab. Для этого были использованы инструментарии [18-21]. На рис. 2 и 3 изображены схемы в стандарте IDEF0, описывающие структуру проекта.

4.2 Эксперимент на наборе данных WISDM

Первый набор состоял из сегментов временных рядов акселерометра мобильного телефона, каждый из которых описывал один из четырех типов физической активности человека — ходьбу, бег, стояние и сидение. Сегмент представлял из себя 10 секундный отрезок исходного временного ряда и состоял из 600 измерений — по 200 измерений

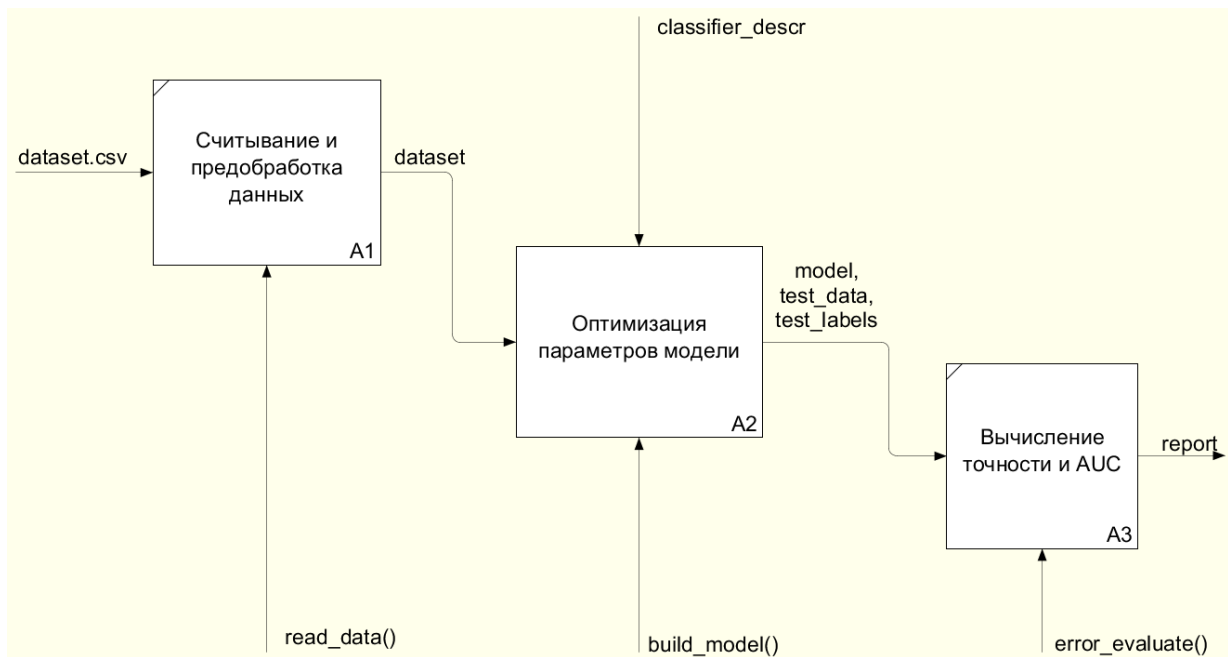


Рис. 2 Общая структура проекта

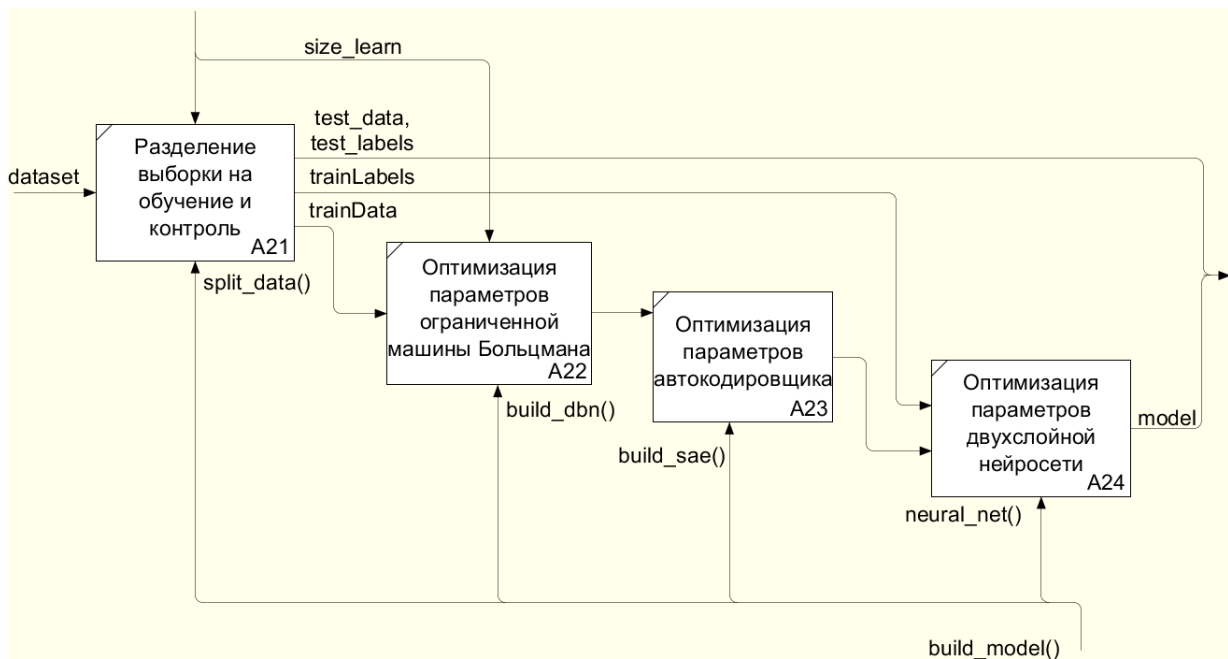


Рис. 3 Структура блока A2 Оптимизация параметров модели

проекции ускорения на каждую из координатных осей. С более подробным описанием данных можно ознакомиться в работе [17].

Выборка состояла из 4219 объектов и была неоднородной по числу элементов в раз-

ных классах. Для того чтобы избежать такого эффекта, как игнорирование моделью малочисленных классов, в обучающую выборку добавлялись повторяющие объекты таким образом, чтобы сбалансировать число представителей каждого класса. Разделение на обучающую и контрольную выборку проводилось случайным образом в соотношении 3:1. В эксперименте использовались следующие суперпозиции: трехслойная суперпозиция, состоящая из машины Больцмана, автокодировщика и двухслойной нейросети с 300, 200 и 100 нейронами в каждом слое соответственно; четырехслойная суперпозиция, состоящая из машины Больцмана, еще одной машины Больцмана, автокодировщика и двухслойной нейросети с 400, 300, 200 и 100 нейронами в каждом слое соответственно; а также пятислойная суперпозиция, состоящая из двух машин Больцмана, двух автокодировщиков и двухслойной нейросети с 500, 400, 300, 200 и 100 нейронами в каждом слое соответственно. В табл. 1 приведены результаты точности классификации для описанных выше моделей, а также результаты из работы [17]. В табл. 2 приведены значения функционала AUC, а на рис. 4 – соответствующие ROC-кривые.

Таблица 1 Сравнительные результаты

Класс	3 слоя	4 слоя	5 слоев	Kwapisz et. al.
Бег	98%	95%	97%	98%
Ходьба	95%	94%	96%	92%
Сидение	100%	100%	100%	95%
Стояние	89%	82%	84%	92%

Таблица 2 значения функционала AUC

Класс	Бег	Ходьба	Сидение	Стояние
3 слоя	0,985	0,964	0,999	0,902
4 слоя	0,983	0,964	0,990	0,960
5 слоев	0,981	0,939	1,000	0,822

4.3 Эксперимент на наборе данных HAR

Второй набор состоял из векторов признаков, полученных предварительной обработкой сегментов временных рядов с акселерометра и гироскопа мобильного телефона Samsung Galaxy S II. Временные ряды сегментировались на 2,56-секундные отрезки, из которых затем получали вектор из 561 признака. Более подробное описание данных приведено в [16]. В этой части вычислительного эксперимента, как и в предыдущей,

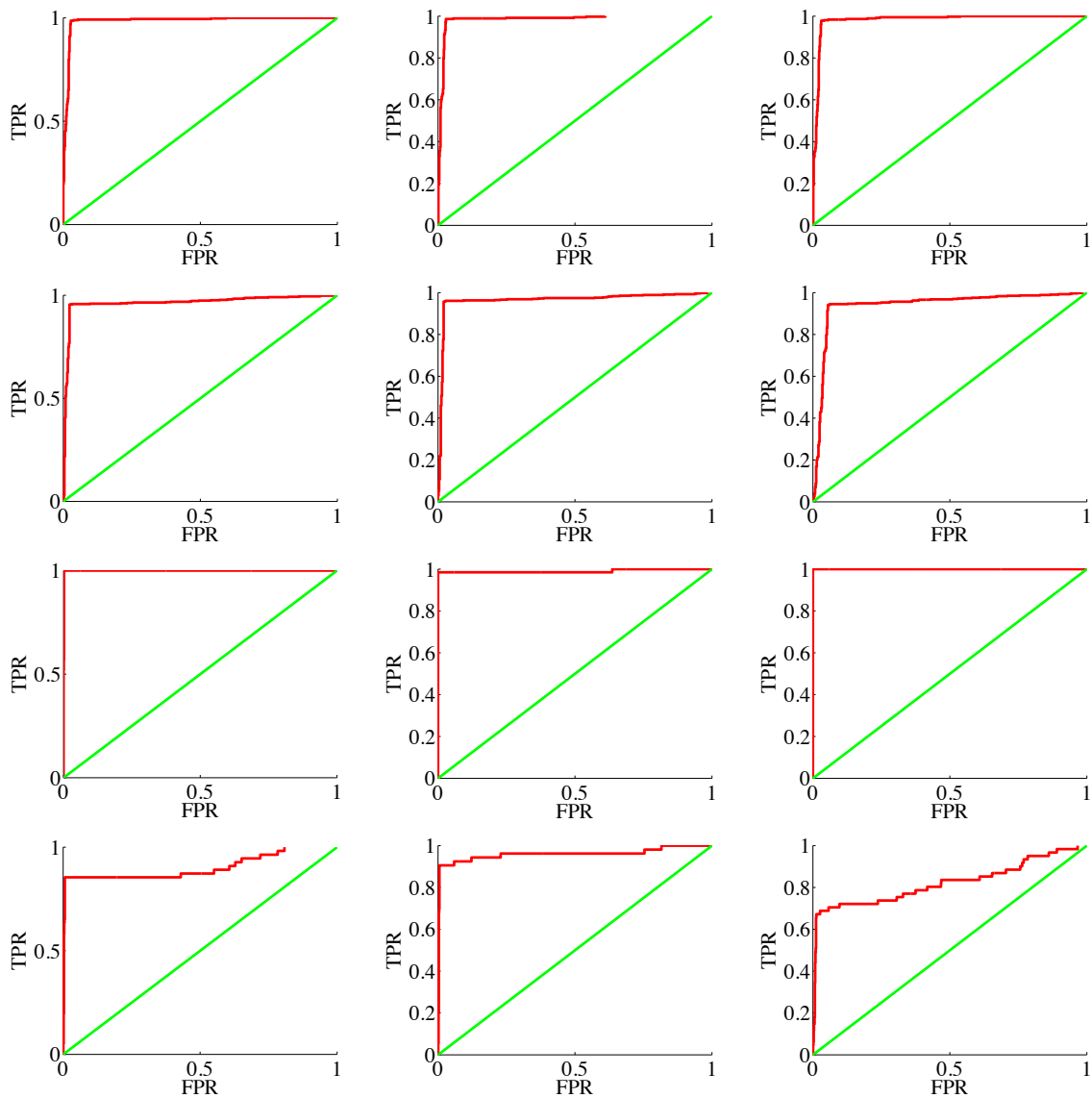


Рис. 4 ROC-кривые для каждого класса (по строчкам): суперпозиция из 3 слоев (слева), из 4 слоев (в центре), из 5 слоев (справа)

использовались следующие суперпозиции: трехслойная суперпозиция, состоящая из машины Больцмана, автокодировщика и двухслойной нейросети с 300, 200 и 100 нейронами в каждом слое соответственно, четырехслойная суперпозиция, состоящая из машины Больцмана, машины Больцмана, автокодировщика и двухслойной нейросети с 400, 300, 200 и 100 нейронами в каждом слое соответственно, а также пятислойная суперпозиция, состоящая из двух машин Больцмана, двух автокодировщиков и двухслойной нейросети с 500, 400, 300, 200 и 100 нейронами в каждом слое соответственно. В табл. 3 приведены результаты точности классификации для описанных выше моделей, а также результаты из работы [16]. В табл. 4 приведены значения функционала

AUC, а на рис. 5 – соответствующие ROC-кривые.

Таблица 3 Сравнительные результаты

Класс	3 слоя	4 слоя	5 слоев	Anguita et. al.
Ходьба	96%	98%	96%	97%
Подъём	94%	93%	80%	87%
Спуск	98%	92%	96%	72%
Сидение	91%	65%	84%	95%
Стояние	75%	80%	71%	97%
Лежание	99,7%	98%	92%	100%

Таблица 4 Значения функционала AUC

Класс	Бег	Ходьба	Сидение	Стояние	Сидение	Стояние
3 слоя	0,986	0,969	0,974	0,934	0,866	0,995
4 слоя	0,982	0,989	0,984	0,767	0,903	0,987
5 слоев	0,9672	0,958	0,969	0,828	0,858	0,981

5 Заключение

В данной работе решалась задача классификации временных рядов. В качестве модели классификации была предложена суперпозиция нейронных сетей глубокого обучения. В вычислительном эксперименте рассматривалась задача многоклассовой классификации физической активности по измерениям с датчиков мобильного телефона. Для выполнения вычислительного эксперимента было реализовано программное обеспечение на языке MatLab, агрегирующее в себе несколько инструментариев. Точность классификации предложенной моделью вычислялась на наборах данных из [16, 17] и сравнивалась с точностью, полученной в этих работах. Полученные результаты оказались сравнимы с результатами из соответствующих работ, что говорит о возможности применения суперпозиции нейронных сетей глубокого обучения к решению задачи классификации временных рядов.

Список литературы

- [1] *Längkvist M., Karlsson L., Loutfi A.* A Review of Unsupervised Feature Learning and Deep Learning for Time-Series Modeling // *Pattern Recognition Letters*, 2014. Vol. 42. No. 6. P. 11–24.
- [2] *Nanopoulos A., Alcock R., Manolopoulos Y.* Feature-based Classification of Time-series Data // *International Journal of Computer Research*, 2001. Vol. 10. P. 49–61.
- [3] *Keogh E., Pazzani M.* A simple dimensionality reduction technique for fast similarity search in large time series databases // *Proceedings of Pacific-Asia Conf. on Knowledge Discovery and Data Mining*. – Kyoto, Japan: Springer, 2000. P. 122–133.
- [4] *Weston J., Mukherjee S., Chapelle O., Pontil M., Poggio T., Vapnik V.* Feature selection for SVMs // *Advances in neural information processing systems*. – Denver, USA: MIT Press, 2000. P. 668–674.
- [5] *Estévez P. A., Tesmer M., Perez C. A., Zurada J. M.* Normalized Mutual Information Feature Selection // *IEEE Transactions on Neural Networks*, 2009. Vol. 20. No. 2. P. 189–201.
- [6] *Mörchen F.* Time series feature extraction for data mining using DWT and DFT // OAI-PMH server at citeseerx.ist.psu.edu, 2003. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.87.2037>.
- [7] *Hinton G. E., Salakhutdinov R. R.* Reducing the dimensionality of data with neural networks // *Science*, 2006. Vol. 313. No. 5786. P. 504–507.
- [8] *Stuhlsatz A., Lippel J., Zielke T.* Feature Extraction With Deep Neural Networks by a Generalized Discriminant Analysis // *IEEE Transactions on Neural Networks and Learning Systems*, 2012. Vol. 23. No. 4. P. 596–608.
- [9] *Ren Y., Wu Y.* Convolutional deep belief networks for feature extraction of EEG signal // *Proceedings of International Joint Conference on Neural Networks (IJCNN 2014)*. – Beijing, China: IEEE, 2014. P. 2850–2853.
- [10] *Xu Y., Mo T., Feng Q., Zhong P., Lai M., Chang C.* Deep learning of feature representation with multiple instance learning for medical image analysis // *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP 2014)*. – Florence, Italy: IEEE, 2014. P. 1626–1630.
- [11] *Hinton G. E., Li Deng, Dong Yu, Dahl G. E., Mohamed A., Jaitly N., Senior A., Vanhoucke V., Nguyen P., Sainath T. N., Kingsbury B.* Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups // *IEEE Signal Process. Mag.*, 2012. Vol. 29. No. 6. P. 82–97.

- [12] *Hüsken M., Stagge P.* Recurrent Neural Networks for Time Series Classification // *Neurocomputing*, 2003. Vol. 50. P. 223–235.
- [13] *Wulsin D., Gupta J., Mani R., Blanco J., Litt B.* Modeling electroencephalography waveforms with semi-supervised deep belief nets: faster classification and anomaly measurement // *Journal of Neural Engineering*, 2011. Vol. 8. P. 1741–2552.
- [14] *Hinton G. E.* A Practical Guide to Training Restricted Boltzmann Machines // *Neural Networks: Tricks of the Trade*. 2nd ed. – Springer, 2012. P. 599–619.
- [15] *Bengio Y.* Learning Deep Architectures for AI // *Foundations and Trends in Machine Learning*, 2009. Vol. 2. No. 1. P. 1–127.
- [16] *Anguita D., Ghio A., Oneto L., Parra X., Luis Reyes-Ortiz J.* Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine // *Ambient Assisted Living and Home Care: Proceedings of the 4th International Workshop (IWAAL 2012)*. – Springer, 2012. Vol. 7657. P. 216–223.
- [17] *Kwapisz J. R., Weiss G. M., Moore S.* Activity recognition using cell phone accelerometers // *SIGKDD Explorations*, 2010. Vol. 12. No 2. P. 74–82.
- [18] *Nabney I.* NETLAB: algorithms for pattern recognitions. – Springer-Verlag, 2002.
- [19] Deep Learning Toolbox <https://github.com/zelyyn/deeplearning-class-2011/tree/master/ufdl/library>
- [20] Deep Neural Network <http://www.mathworks.com/matlabcentral/fileexchange/42853-deep-neural-network>
- [21] Deep Learning Toolbox <https://github.com/rasmusbergpalm/DeepLearnToolbox>

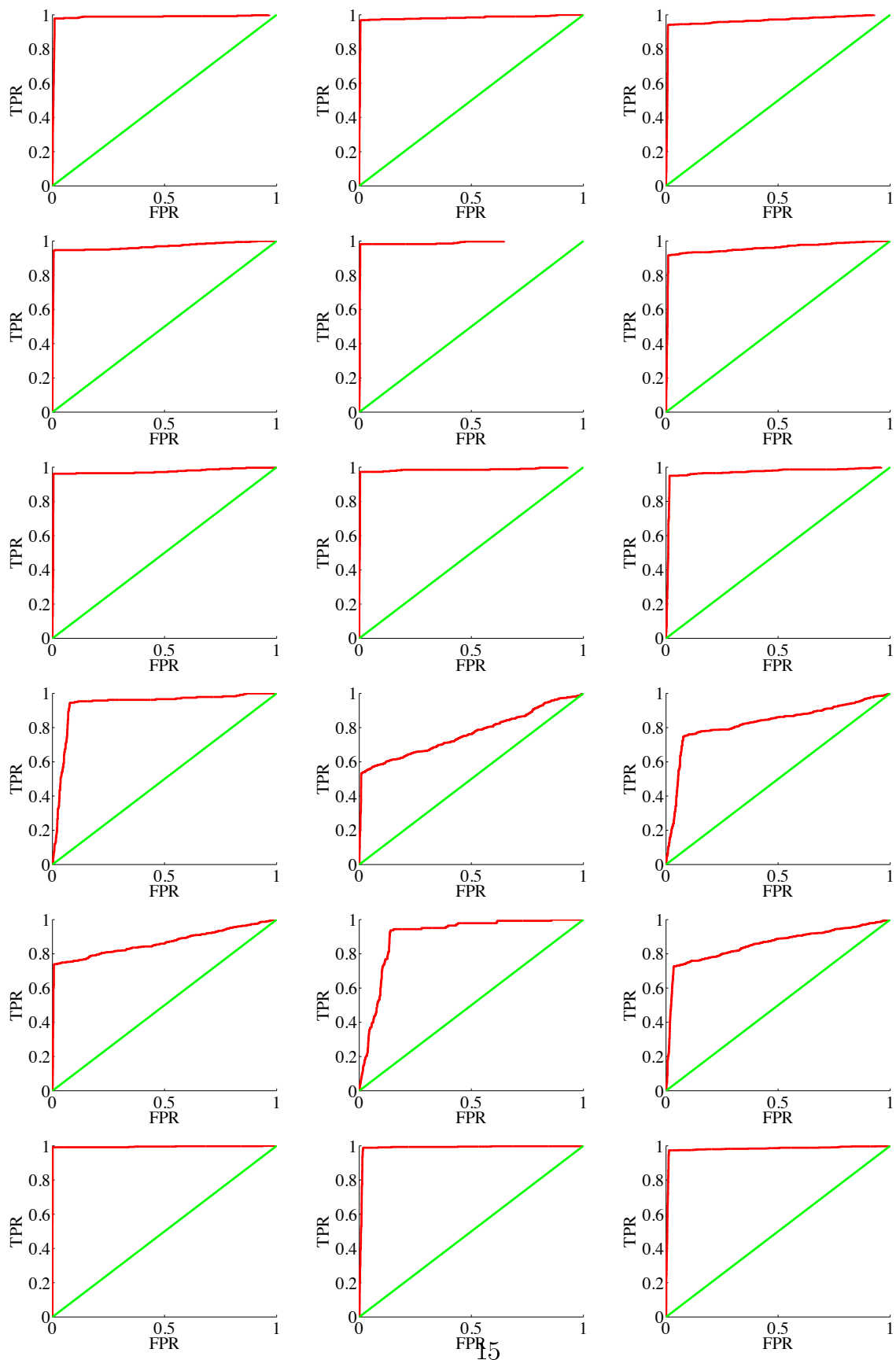


Рис. 5 ROC-кривые для каждого класса (по строчкам): суперпозиция из 3 слоев (слева), из 4 слоев (в центре), из 5 слоев (справа)