

УДК 656.07

АДАПТИВНОЕ УПРАВЛЕНИЕ ПЕРЕКРЕСТКАМИ С ИСПОЛЬЗОВАНИЕМ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ И ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Девятков В.В., Федотов М.В., Андреев Н.В., Маряшина Д.Н., Девятков Т.В.,
Шестюк В.М., Минниханов Р.Р. (Казань)

Введение

Многолетний опыт применения систем адаптивного управления движением на перекрестках по всему миру наглядно показал преимущества таких систем в условиях высокой интенсивности движения транспортных средств. За длительный период изучения и эксплуатации, выработано немало подходов к решению задачи эффективного или даже оптимального распределения потоков на перекрестке. Различие подходов проявляется не только в методике принятия решения, но и в части используемых источников объективных данных о текущей ситуации на перекрестке. В качестве типовых источников исходных данных выступают датчики движения и средства видеофиксации. Результатом работы алгоритмов адаптивного управления является определение длительности активной фазы движения транспортных средств.

Авторами статьи был проанализирован ряд работ исследователей [2-8], занимающихся разработкой новых и модификацией существующих алгоритмов управления транспортными потоками (далее ТП). Анализ работ показывает, что в последнее время классические подходы комбинируются с методами машинного обучения, глубокого анализа данных, искусственного интеллекта и т.п. Имитационное моделирование (далее ИМ) в таких системах используется редко и лишь для тестирования работоспособности предлагаемых методов.

Авторами данной статьи рассматривается подход к построению системы адаптивного управления ТП с использованием имитационного моделирования и нейронных сетей.

Предполагается, что система адаптивного управления будет интегрирована в общий контур платформы по управлению дорожным движением (далее ДД). На рисунке 1 представлена общая схема решения.

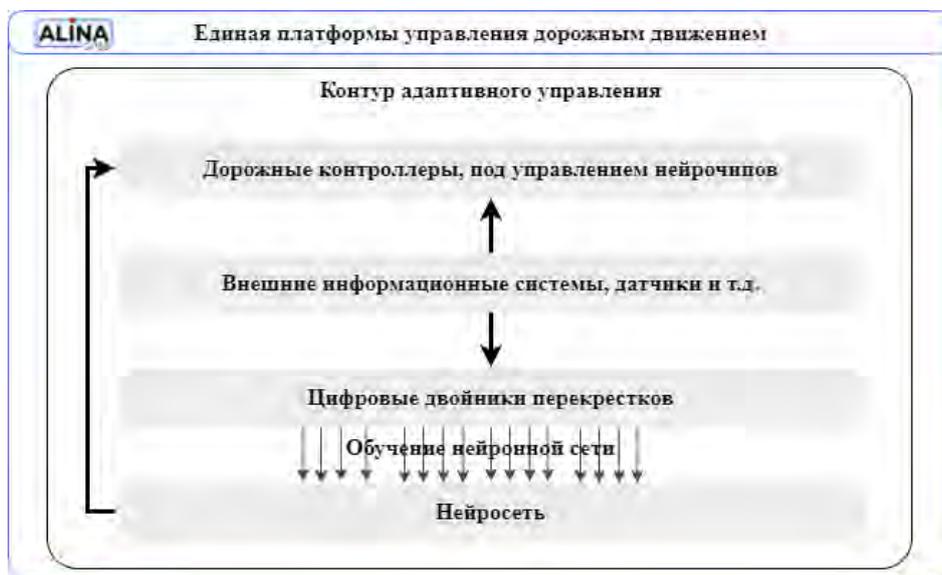


Рис. 1. Схема системы адаптивного управления дорожным движением

Имитационная модель перекрестка

Имитационная модель перекрестка позволяет воспроизвести заданную дорожную ситуацию для сбора статистики о характере движения транспортных средств и определения узких мест. Такой подход выгодно отличается от эксперимента на реальном перекрестке, т.к. позволяет отладить и подготовить компоненты без вмешательства в текущую дорожную инфраструктуру. А после, на этапе внедрения, скорректировать отклонения в работе имитационной модели и реального движения транспорта на основе реальной статистики и экспертных оценок.

На этапе подготовки к внедрению системы на перекресток имитационная модель предоставит специалисту инструменты для анализа показателей функционирования перекрестка и статистики движения транспортных средств при различных сценариях изменения интенсивности транспортных потоков, инфраструктуры и организации дорожного движения, а также анализа узких мест. А результаты моделирования будут являться исходными данными для обучения нейронной сети.

Предлагаемая имитационная модель и транспортная библиотека являются продолжением работ по моделированию участка УДС города Казани, которая выполнялась указанными выше организациями в 2022 году [1]. Подробно ознакомиться с ней можно в материалах конференций ИММОД 2023. Разработанное в ходе работ программное обеспечение является универсальным и может быть использовано для моделирования широкого спектра перекрестков.

В процессе работы имитационной модели симулируется движение транспортных средств и пешеходов в виртуальном пространстве и времени. Это позволяет довольно точно получить не только итоговые показатели движения и узкие места, но и увидеть причины их возникновения и характер развития. Виртуальное пространство обеспечивается логической организацией движения согласно расширенному представлению клеточного автомата. Корректное управление виртуальным временем и ресурсами гарантирует система моделирования – имитационная модель разработана на отечественном универсальном языке GPSS World Core и платформе Alina GPSS [2]. Данный язык на протяжении многих десятилетий используется для решения задач в логистике, промышленности и других областях науки и техники, и зарекомендовал себя как эффективный и надежный инструмент.

Исходными данными для имитационной модели могут выступать исторические данные о движении транспортных средств. При их наличии становится возможным проведение анализа движения на перекрестке за длительный период и выявление типовых интенсивностей появления транспортных средств на перекрестке. Эти данные особенно важны в случае отсутствия или отказа одного или нескольких датчиков движения. Наличие подобной статистики позволит сохранить адекватность принимаемых решений по адаптивному управлению, даже в ситуации отсутствия реальных данных.

Для определения данных о направлениях движения транспорта на перекрестке применяется несколько подходов. Первый – ручной сбор данных: методики проведения такого сбора в научном плане достаточно хорошо проработаны, но приводят к большим затратам времени и наличию специально обученных сотрудников. Второй – экспертное заключение: организации, занимающиеся управлением движением, обладают собственной статистикой и пониманием характера движения транспорта в различное время суток, дней недели и т.д. Третий – обработка записей с камер видеонаблюдения методами компьютерного зрения. Наличие большого количества записей позволяет сформировать достоверную статистику движения транспортных средств. Ни один из этих методов не может дать 100% точного распределения вероятностей движения на перекрестке, но позволяет отразить общий характер движения.

На основе результатов множественных прогонов имитационной модели для разных дорожных ситуаций можно сформировать таблицу базовых длительностей фаз. На основе этой таблицы, с учетом отклонений реального движения транспорта от смоделированного, принимать решение о задании фактической длительности фазы.

Формирование оптимизационных экспериментов позволит определять формально оптимальные длительности фаз для найденных шаблонных ситуаций.

Применение нейронных сетей для адаптивного управления

Адекватное воспроизведение дорожной ситуации в имитационной модели может потребовать довольно серьезных вычислительных ресурсов, особенно при управлении сложным перекрестком или моделировании сложной дорожной ситуации. Учитывая, что в качестве аппаратной платформы обычно используется мини-компьютер с ограниченными вычислительными мощностями, прямой подход с использованием имитационной модели (как основного инструмента адаптивного управления) может быть малоэффективен. Вместо этого предлагается использовать результаты моделирования для обучения нейронной сети, которая будет способна распознавать различные дорожные ситуации и выдавать решение в условиях частичного отсутствия объективных данных.

Для решения поставленной задачи возможно применение различных подходов в зависимости от исходных данных и сложности организации дорожного движения (далее ДД):

1. Рекуррентные нейронные сети (Recurrent neural network) и их модификации (LSTM и GRU) подходят для анализа временных рядов и последовательных данных, таких как состояние светофорных объектов на перекрестках, они способны учитывать зависимости между состояниями на разных временных шагах.

2. Сверточные нейронные сети (Convolutional Neural Network) подходят для анализа пространственных данных, связанных с перекрестком, могут эффективно обрабатывать изображения с камер видеонаблюдения, установленных на перекрестках.

3. Комбинированные архитектуры, объединяющие сверточные и рекуррентные слои, подходят для учета как пространственных характеристики ДД, так и изменения состояния светофорных объектов на протяжении времени.

4. Глубокие усиленные модели (Deep Reinforcement Learning) подходят для нахождения оптимальных циклов светофорного регулирования с учетом изменяющихся дорожных условий и транспортных потоков.

Реализация нейронной сети

В рамках апробации данного подхода в качестве тестового перекрестка г. Казани было выбрано пересечение улиц Островского и Миннуллина (рисунок 2).



Рис. 2. Перекресток Островского – Миннуллина

Исходя из структуры перекрестка и схемы организации ДД на нем, было решено использовать полносвязную нейронную сеть с одним входным слоем, одним скрытым слоем обработки и одним выходным слоем (рис. 3).

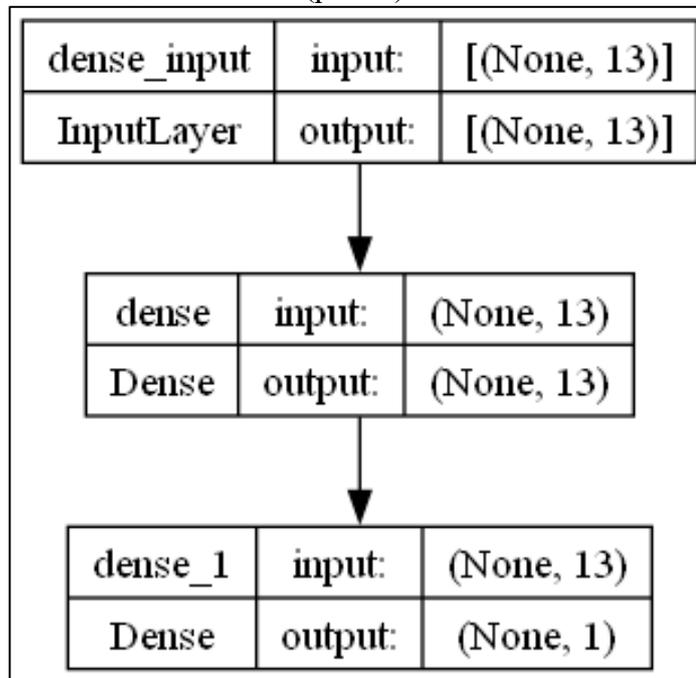


Рис. 3. Схема простой нейронной сети

Нейронная сеть состоит:

1. Входной слой (Input Layer):

- Размерность: 13 нейронов.
- Функция активации: отсутствует (линейная функция активации).
- Входные данные: $(X = [x_1, x_2, \dots, x_{13}])$

2. Скрытый слой (Hidden Layer):

- Тип слоя: полносвязный (Dense Layer).
- Нейроны: 13 нейронов.
- Функция активации: ReLU (Rectified Linear Activation).
- Выходы скрытого слоя: $(H = [h_1, h_2, \dots, h_{13}])$
- Формула для активации нейрона в скрытом слое:

$$h_i = \text{ReLU}\left(\sum_{j=1}^{13} w_{ij} x_j + b_i\right), \quad i = 1, 2, \dots, 13 \quad (1)$$

где (w_{ij}) – весовой коэффициент между входным нейроном (x_j) и скрытым нейроном (h_i) , (b_i) – смещение (bias) скрытого нейрона.

3. Выходной слой (Output Layer):

- Тип слоя: полносвязный (Dense Layer).
- Нейроны: 1 нейрон.
- Функция активации: отсутствует (линейная функция активации).
- Выход сети (предсказание): (\hat{y}) .
- Формула для выходного нейрона:

$$\hat{y} = \sum_{i=1}^{13} w_i h_i + b, \quad (2)$$

где w_i – весовой коэффициент выходного нейрона, b – смещение выходного нейрона.

4. Компиляция модели (Compilation):

- Оптимизатор: Adam.
- Функция потерь (Loss Function): Mean Squared Error (MSE).

В данной структуре нейронной сети каждый нейрон в скрытом слое связан с каждым нейроном входного слоя, что обеспечивает полное соединение (fully connected) между слоями. Функция активации ReLU применяется для внесения нелинейности в модель. Модель компилируется с помощью оптимизатора Adam, и функция потерь MSE вычисляет среднеквадратичную ошибку между предсказанными значениями (\hat{y}) и истинными целевыми значениями (y) .

В результате первичного тестирования нейронной сети результаты получились неудовлетворительными. Для получения удовлетворительных результатов и дальнейшей возможности применения нейронной модели на реальном перекрестке было принято решения перейти от полносвязной нейронной сети, описанной выше, к сверточной нейронной сети.

Сверточная нейронная сеть, схема которой представлена на рисунке 4, предназначена для решения задачи регрессии.

Архитектура нейронной сети предназначена для работы с временными рядами и может использоваться для предсказания времени фаз работы светофора.

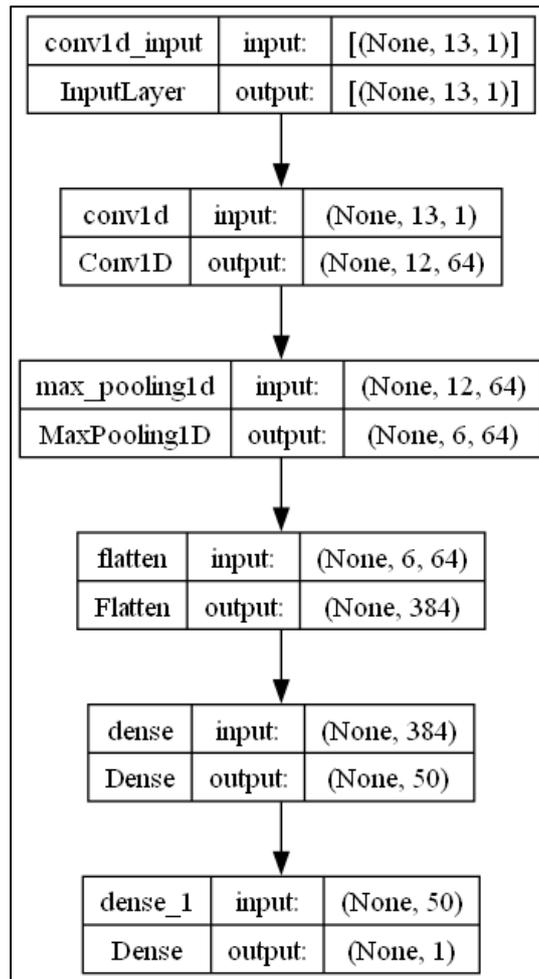


Рис. 4. Сверточная нейронная сеть

Структура сети:

1. Входной слой (Input Layer):

- Входные данные представлены как временные ряды, где n_steps – количество временных шагов, а $n_features$ – количество признаков на каждом временном шаге:

$$X \in R^{n_steps \times n_features} \quad (3)$$

2. Сверточный слой (Convolutional Layer):

- В данном слое используется одномерная свертка (1D Convolution) с $filters$ фильтрами (ядрами) размера $kernel_size$:

$$Conv1D_output = ReLU(Conv1D(X, filters, kernel_size)) \quad (4)$$

3. Пулинг слой (Pooling Layer):

- После свертки применяется операция пулинга (пулинг по максимуму - Max Pooling) с окном размера $pool_size$:

$$Pooling_output = MaxPooling1D(Conv1D_output, pool_size) \quad (5)$$

4. Выравнивающий слой (Flatten Layer):

- Выход из пулинг слоя выравнивается в одномерный вектор:

$$Flatten_output = Flatten(Pooling_output) \quad (6)$$

5. Полносвязный слой (Dense Layer):

- Следующий слой - полносвязный слой (Dense) с 50 нейронами и функцией активации ReLU:

$$Dense_output = ReLU(Dense(Flatten_output, 50)) \quad (7)$$

6. Выходной слой (Output Layer):

- В конце сети находится выходной слой с одним нейроном, который предсказывает некоторое численное значение (регрессия):

$$Output = Dense(Dense_output, 1) \quad (8)$$

7. Компиляция (Compilation):

- Сеть компилируется с оптимизатором Adam и функцией потерь Mean Squared Error (MSE):

$$Loss = MSE(y, Output) \quad (9)$$

где y - целевые значения для обучения.

Для обучения нейронной сети была определена следующая структура исходных данных (рис. 5):

- 1 Наименование направления (преобразованы в числовые значения).
- 2 Номер первой дороги.
- 3 Номер полосы первой дороги.
- 4 Кол-во авто на полосе.
- 5 Номер первой дороги.
- 6 Номер второй полосы первой дороги.
- 7 Кол-во авто на полосе.
- 8 Номер второй дороги.
- 9 Номер полосы второй дороги.
- 10 Кол-во авто на полосе.
- 11 Номер второй дороги.
- 12 Номер другой полосы второй дороги.
- 13 Кол-во авто на полосе.

Исходя из структуры исходных данных и было выбрано количество нейронов на входном слое равное 13. И 1 нейрон на выходном слое – выходной параметр с временем проезда всех авто в секундах.



Рис. 5. Пример обучающей выборки

Оценка работоспособности нейронной сети

Размер обучающей и тестовой выборки составил 7371 строк и 821 строку соответственно. В результате тестирования получен результат по среднему времени предсказания, средней абсолютной ошибке и средней абсолютной процентной ошибке (рис.6).

```
Средняя абсолютная ошибка (MAE): 2.2679658952496955
Среднее время предсказания (сек): 0.03409774904564731
Средняя абсолютная процентная ошибка (MAPE): 5.2566333508474665 %
```

Рис. 6. Результаты тестирования

В контексте текущей задачи не вычислялась точность модели, так как «точность» (ассигасу) – это метрика оценки качества модели, которая измеряет, насколько точно модель классифицирует данные. В данном случае речь идет о задаче регрессии (предсказание численных значений), а не классификации, поэтому термин «точность»

не является наилучшим выбором. Метрика точности в области регрессии не имеет смысла [9].

Вместо этого для оценки качества модели в задаче регрессии можно использовать другие метрики, такие как:

1. Средняя абсолютная ошибка (далее MAE) – средняя абсолютная разница между предсказанными и фактическими значениями.
2. Средняя квадратичная ошибка (MSE) – средняя квадратичная разница между предсказанными и фактическими значениями.
3. Коэффициент детерминации (R-squared) – мера, которая оценивает объяснительную способность модели.

Для текущей модели была выбрана средняя абсолютная ошибка. MAE обычно измеряется в тех же единицах, что и целевая переменная, и не выражается в процентах. MAE представляет собой среднюю абсолютную разницу между прогнозами модели и фактическими значениями, и эта разница измеряется в тех же единицах, что и сами данные.

Средняя абсолютная процентная ошибка (далее MAPE), которая выражает ошибку в процентах относительно фактических значений. Формула MAPE выглядит следующим образом:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left(\frac{|y_i - \hat{y}_i|}{|y_i|} \right) \times 100\% \quad (10)$$

где: (n) – количество примеров в вашем тестовом наборе данных, (y_i) – фактическое значение для i -го примера, (\hat{y}_i) – прогнозное значение для i -го примера.

MAPE измеряется в процентах и предоставляет информацию о средней относительной ошибке прогнозов к фактическим значениям.

Также была построена схема, на которой отображены результаты сравнения эталонных значений с предсказанными (рис. 7). За эталонные значения были приняты значения, полученные с помощью имитационного моделирования.

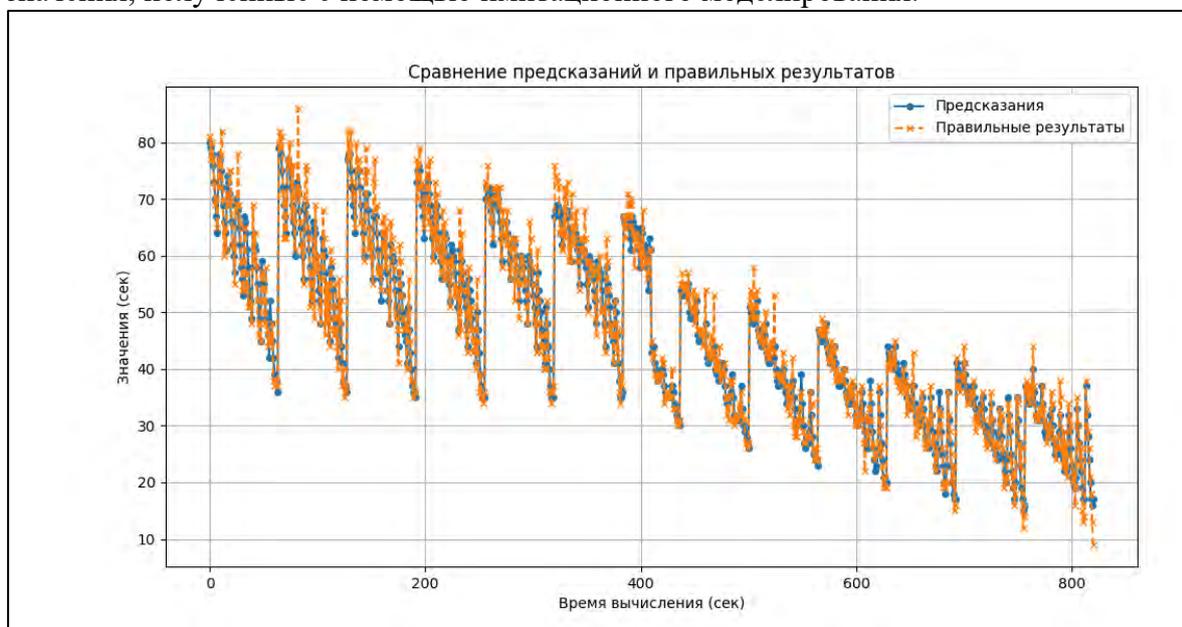


Рис. 7. Схема сравнения предсказаний и эталонных значений

Заключение

В рамках данной работы закладывается базис создания отечественного программно-аппаратного комплекса по адаптивному управлению дорожным

движением. Отличительной особенностью комплекса является реализация адаптивных алгоритмов управления транспортными потоками на перекрестках с применением имитационного моделирования и нейронных сетей.

Разработанная имитационная модель хорошо показала себя при моделировании движения на основе исторических данных. В настоящее время выполняются работы по повышению адекватности модели путем усложнения алгоритмов поведения виртуальных водителей.

Разработанная нейронная модель представляет собой свёрточную нейронную сеть, спроектированную для анализа временных рядов. Результаты экспериментов показали, что модель обладает средней абсолютной ошибкой (MAE) в размере 2.27 секунд относительно данных по результатам моделирования. Среднее время предсказания составляет 0.03 секунды на одно предсказание (одно направление движения). Средняя абсолютная процентная ошибка (MAPE) оценена на уровне 5.26%.

В перспективе планируется перейти от локального адаптивного управления изолированными перекрестками к сетевому адаптивному управлению всей улично-дорожной сетью города.

Литература

1. **Аникеев Е. А.** Информационные системы управления дорожным движением // Моделирование систем и процессов. – 2016. – № 4. – С. 5-11.
2. **Cai Yanguang and Cai Hao.** A Dynamic Traffic Signal Timing Model and Its Algorithm for Junction of Urban Road // Future Wireless Networks and Information Systems, LNEE 144. – P. 55.
3. **P. Balaji and D. Srinivasan.** Type-2 fuzzy logic based urban traffic management // Engineering Applications of Artificial Intelligence. – 2010. – 24. – P. 11.
4. **Saman Lawe and Ruili Wang.** Optimization of Traffic Signals Using Deep Learning Neural Networks // Springer International Conference on Artificial Intelligence, LNAI 9992. – 2016. – P. 03–415.
5. **Monireh Abdoos, Nasser Mozayani and Ana LC Bazzan.** Hierarchical control of traffic signals using Q-learning with tile coding // Applied intelligence. – 2014. – 40, 2. – P. 201–213.
6. **Guanjie Zheng, Yuanhao Xiong, Xinshi Zang, Jie Feng and Hua Wei.** Learning Phase Competition for Traffic Signal Control. ACM ISBN 978-1-4503-6976-3/19/11.
7. **De Oliveira D., Bazzan A.L.C., da Silva B.C. et. al.** Reinforcement learning-based control of traffic lights in non stationary environments: A case study in a microscopic simulator». EUMAS06, 2006. Dec. 4–7.
8. **Ayad M. Turky, M.S. Ahmad, M.Z.M. Yusoff and Baraa T. Hammad** Using Genetic Algorithm for Traffic Light Control System with a Pedestrian Crossing. / P. Wen et al. (Eds.): RSKT 2009, LNCS 5589. – P. 512–519. 2009.
9. **Васенков Д.В.** Методы обучения искусственных нейронных сетей // Компьютерные инструменты в образовании. – СПб.: Изд-во ЦПО Информатизация образования. – 2007 – № 1. – С. 20–29.