

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОГРАММНЫХ ПАКЕТОВ ДЛЯ РАБОТЫ С ИСКУССТВЕННЫМИ НЕЙРОННЫМИ СЕТЯМИ

Я. А. Туровский*, С. Д. Кургалин*, А. А. Адаменко**

*Воронежский государственный университет,

**Воронежский государственный университет инженерных технологий

Поступила в редакцию 16.03.2016 г.

Аннотация. в рамках разработки программного обеспечения для моделирования обучения нейрочипов был проведен сравнительный анализ трех программных пакетов для работы с искусственными нейронными сетями в задачах классификации: *ANNBuilder* – оригинальной разработки для целей автоматизации моделирования обучения нейрочипов, восстанавливающих поврежденную нервную ткань и пакетов *MatLab* и *Statistica*. Проведена серия из 10 вычислительных экспериментов по обучению искусственных нейронных сетей на сгенерированных обучающих выборках и проведен статистический анализ результатов этих экспериментов с целью выявления статистически значимых различий между ними для выбранных программных пакетов. В большинстве случаев наилучшие результаты обучения искусственных нейронных сетей показал программный пакет *ANNBuilder*. Сравнение *ANNBuilder* с вышеуказанными программными пакетами проводилось для оценки результатов выполняемого обучения и выбора базисных показателей обучения нейрочипов. Реализованные алгоритмы обучения искусственных нейронных сетей в пакете *ANNBuilder* могут быть использованы для создания базисных показателей обучения эволюционных алгоритмов в системах автоматизированного моделирования обучения нейрочипов, восстанавливающих нервную ткань.

Ключевые слова: искусственная нейронная сеть, программное обеспечение, программный пакет, алгоритм, информационная система, моделирование, сравнительный анализ.

Annotation. In the development of software for simulation training neurochips a comparative analysis of the three software packages to work with artificial neural networks in classification problems: *ANNBuilder* – originally developed for the purpose of automation simulation training neurochips restoring damaged nervous tissue and packet *MatLab* and *Statistica*. A series of computational experiments 10 training artificial neural networks generated learning samples and the statistical analysis of the results of these experiments to detect statistically significant differences between the selected software packages. In most cases, the best results of study of artificial neural networks showed a software package *ANNBuilder*. Compare *ANNBuilder* with the above software packages was carried out to assess the results, do the training and selection of training benchmarks neurochips. Implemented training algorithms of artificial neural networks in the package *ANNBuilder* can be used to create benchmarks teaching of evolutionary algorithms in computer-aided simulation training neurochips restoring nerve tissue.

Keywords: artificial neural network, software, software package, algorithm, information systems, simulation, comparative analysis.

ВВЕДЕНИЕ

В наши дни возрастает потребность в информационных системах, которые могут проводить аппроксимацию, классификацию и распознавание различных типов данных в широких диапазонах их изменения и способ-

ных к гибкому обучению в зависимости от изменения поставленных перед ними целей и задач. В таких системах широко используются искусственные нейронные сети (ИНС). В настоящее время имеется большое количество разнообразных программных пакетов для работы с ИНС, однако реализация методов и алгоритмов в большинстве программных решений (например, алгоритмы обучения ИНС или нормализации входных обучающих дан-

© Туровский Я. А., Кургалин С. Д., Адаменко А. А., 2016

ных) скрыта от пользователя. Пользователь в большинстве случаев не имеет возможности узнать, какие алгоритмы с какими характеристиками применили разработчики пакета с целью добиться наилучшего результата в обучении ИНС, вследствие чего трудно определить, за счет каких именно свойств алгоритма разные программные пакеты демонстрируют различные результаты на одних и тех же обучающих выборках (ОВ). Также, это вызывает сложности при применении пакета в собственных разработках и при проведении исследований. Как следствие, невозможно выбрать оптимальный программный пакет для работы с ИНС.

Современные искусственные нейронные сети часто используются не как отдельный инструмент, а интегрируются в информационные системы, для которых требуется гибкое обучение в зависимости от изменения поставленных задач. Вследствие этого для проведения большого объема вычислительных экспериментов ИНС должны быть интегрированы в программные комплексы, с использованием которых будут проводиться эти эксперименты.

Некоторые программные пакеты, такие как *MatLab* [1] или *Statistica* [2], предоставляют для работы с ИНС CASE-средства, позволяющие генерировать код на конкретном языке программирования и интегрировать его в различные информационные системы. Работа с таким программным кодом может вызвать у пользователя серьезные проблемы при попытке его масштабирования или изменения в своих целях (например, при изменении функции активации нейронов, добавлении нейронов в ИНС и др.), так как этот код «статичен»: топологию ИНС, весовые коэффициенты связей между нейронами и функции активации ИНС нельзя поменять без изменения результатов функционирования ИНС.

В итоге, исследователь не всегда может видеть детали функционирования применяемого программного пакета, что в случае, когда ИНС обучается на одной и той же ОВ при использовании разных программных пакетов, может привести к нарушению воспроизводимости результатов вычислительного экс-

перимента. Следовательно, сравнение результатов работы разных программных пакетов для обучения ИНС является важной задачей для оценки качества выполняемого обучения ИНС в программном пакете *ANNBuilder* и выбора базисных показателей обучения нейрончиков. Под базисным показателем мы будем подразумевать результат обучения ИНС с использованием любого программного пакета, который будет демонстрировать лучшие показатели по сравнению с другими программными пакетами из исследуемого пула.

Целью данной статьи является получение сравнительных характеристик работы ИНС, обучение которых проводилось с использованием разных программных пакетов в задачах классификации, включая разработанный программный пакет *ANNBuilder*.

Сравнение пакета *ANNBuilder* с пакетами *Statistica* и *MatLab* представляет интерес для оценки результатов проводимого обучения ИНС и выбора базисных показателей обучения нейрончиков, которые представляют интерес с точки зрения использования их в нейроинтерфейсах [3, 4]. Процесс обучения нейрончиков требует постановки большого количества вычислительных экспериментов, которые выходят за рамки возможностей отдельных программных пакетов для работы с ИНС. Как следствие, приходится применять несколько программных пакетов, реализация алгоритмов которых скрыта от пользователя. Это затрудняет воспроизводимость результатов, полученных при использовании разных пакетов. Также стоит отметить, что передача данных между программными пакетами в случае работы с ИНС представляет собой крайне трудоемкий процесс, если его осуществлять «вручную», т. е. без использования дополнительного программного обеспечения для получения и передачи информации об обучении ИНС и параметрах этого процесса между пакетами. В связи с отмеченными выше проблемами требуется существенная автоматизация процесса обучения нейрончика, что позволит выполнить за единицу времени значительно большее количество вычислительных экспериментов и итераций обучения, существенно ускорив получение результатов этих процессов.

ОБУЧЕНИЕ ИНС В ПРОГРАММНЫХ ПАКЕТАХ

Для проведения сравнительного анализа обучения ИНС были выбраны следующие программные пакеты: *ANNBuilder* – оригинальная разработка для целей автоматизации моделирования обучения нейронных сетей, восстанавливающих поврежденную нервную ткань; *MatLab* и *Statistica* – как одни из наиболее распространенных пакетов.

Для обучения ИНС было сгенерировано 10 ОВ, как линейно, так и нелинейно разделимых. Такого количества ОВ достаточно для решения поставленной задачи. Каждая ОВ содержит 2–6 классов, правильное распознавание которых, при соответствующем входном векторе, и является задачей обучения ИНС в задаче классификации.

В табл.1 приведены характеристики ОВ для обучения ИНС.

Для каждой из приведенных в табл. 1 ОВ было обучено несколько ИНС различных топологий с одним скрытым слоем в разных программных пакетах. Количество входов и нейронов в выходном слое выбиралось в зависимости от количества элементов во входном векторе и количества классов соответственно. Количество нейронов в скрытом слое было выбрано следующим: 4, 8, 12, 16, 20, 24, 28, 32, 36, 40. Таким образом, на каждой

ОВ обучалось по 10 ИНС с вышеприведенным количеством нейронов в скрытом слое. При этом 80 % от всей ОВ было отделено для обучения ИНС, а 20 % – для валидации ИНС.

После проведения экспериментов над группами ИНС для каждого программного пакета были определены результаты ошибочных распознаваний для каждой ИНС.

В табл. 2 приведены результаты ошибочных распознаваний ИНС, обученных при использовании программного пакета *ANNBuilder* со следующими параметрами: алгоритм – обратное распространение ошибки [5]; коэффициент обучения [6] – 0,5; момент [7] – 0,3. Каждая ИНС обучалась за 5000 итераций.

На рис. 1 представлен результат процесса обучения ИНС в программном пакете *ANNBuilder* на обучающей выборке № 7 (см. табл. 2) с одним скрытым слоем, в котором находится 40 нейронов.

Для работы с ИНС в программном пакете *Statistica* был выбран инструмент *Statistica Automated Neural Networks* [8]. При обучении ИНС использовались следующие параметры: алгоритм – Gradient Descent [9], в основе которого лежит алгоритм обратного распространения ошибки; коэффициент обучения – 0,5; момент – 0,3. Каждая ИНС обучалась за 5000 итераций.

Для работы с ИНС в программном пакете *MatLab* использовался инструмент *MatLab*

Таблица 1

Сведения об ОВ, используемых для обучения ИНС

№	Количество элементов во входном векторе	Количество входящих векторов	Количество классов
1*	3	10000	4
2*	4	10000	4
3*	4	10000	4
4*	4	10000	4
5*	4	10000	4
6*	3	10000	6
7	27	7783	2
8	27	8005	2
9	27	7784	2
10	27	14266	2

* – обозначены линейно разделимые ОВ

Таблица 2

Результаты ошибочных распознаваний для обученных ИНС в программном пакете ANNBuilder, в процентах от абсолютного количества ошибочных распознаваний

№ обучающей выборки / количество нейронов в скрытом слое	4	8	12	16	20	24	28	32	36	40
1	1,4	1,3	1,3	1,3	1,3	1,3	1,3	1,3	1,4	1,4
2	1,4	1,4	1,4	1,4	1,4	1,4	1,4	1,4	1,4	1,4
3	1,4	1,5	1,4	1,3	1,4	1,5	1,5	1,5	1,5	1,7
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	5,2	4,1	3,8	2,6	2,1	1,7	1,3	1,1	1,1	0,9
8	9,7	7,1	6,1	5	3,8	4,4	3,5	3,1	2,1	2,3
9	7,2	7,0	5,8	5,6	4,4	3,8	2,8	2,3	2,4	2,1
10	17,2	9,7	9,0	8,1	8,1	8,1	7,3	6,1	7,0	6,0

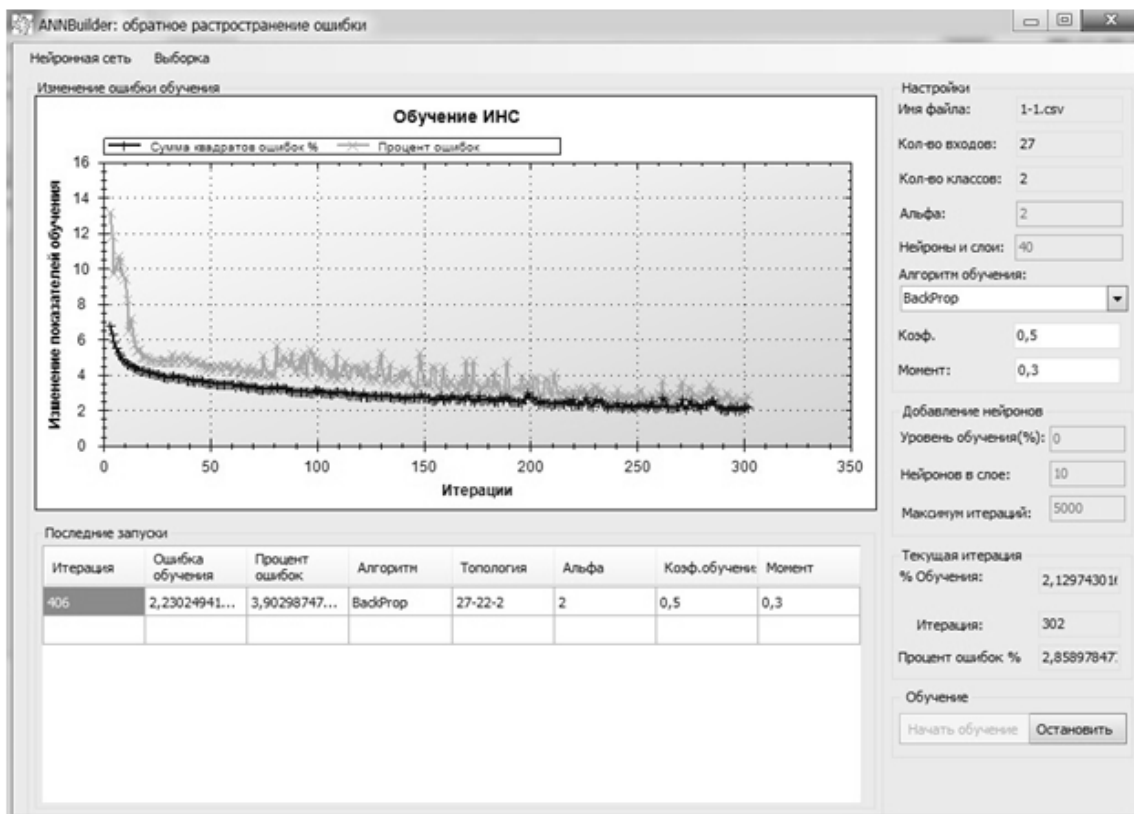


Рис. 1. Интерфейс программы ANNBuilder, демонстрирующий результат обучения ИНС на выборке № 7 с одним скрытым слоем, в котором находится 40 нейронов

Neural Network ToolBox [10]. При обучении ИНС в программном пакете *MatLab* были выбраны следующие параметры: алгоритм – Gradient Descent with momentum [11], в основе которого лежит алгоритм обратного распространения ошибки; коэффициент обу-

чения – 0,5; момент – 0,3. Каждая ИНС обучалась за 5000 итераций.

При обучении были использованы: алгоритм обратного распространения ошибки и алгоритмы Gradient Descent и Gradient Descent With Momentum. Алгоритм обратного

Таблица 3

Результаты ошибочных распознаваний для обученных ИНС в программном пакете *Statistica*, в процентах от абсолютного количества ошибочных распознаваний

№ ОВ / количество нейронов в скрытом слое	4	8	12	16	20	24	28	32	36	40
1	1,5	1,5	1,4	1,5	1,4	1,5	1,4	1,5	1,4	1,3
2	1,5	1,6	1,7	1,7	1,7	1,8	1,7	1,7	2,0	1,9
3	1,1	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	7,6	6,9	6,0	6,5	5,5	6,4	6,0	6,7	5,5	5,8
8	10,8	10,1	10,5	10,6	11,1	10,1	10,6	10,3	9,6	10,6
9	9,6	8,7	9,9	9,1	9,3	8,5	9,3	8,9	8,5	9,0
10	10,6	9,7	10,0	10,2	10,3	10,2	10,3	9,8	9,9	10,0

Таблица 4

Результаты ошибочных распознаваний для обученных ИНС в программном пакете *MatLab*, в процентах от абсолютного количества ошибочных распознаваний

№ обучающей выборки / количество нейронов в скрытом слое	4	8	12	16	20	24	28	32	36	40
1	0,2	0,6	0,3	0,4	0,2	0,4	0,2	0,1	0,3	0,2
2	1,6	1,5	1,7	1,5	1,4	1,3	1,7	1,4	1,7	2,0
3	1,2	1,2	1,4	1,1	1,2	1,2	1,1	1,1	0,7	1,1
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0,1	0	0	0	0	0
6	5,0	2,5	0	0	0	0	0	0	0	0
7	8,4	7,7	6,7	6,6	5,9	5,9	6,1	6,4	5,4	5,5
8	14,7	12,3	11,5	11,8	11	10,7	10,8	10,8	11,3	10,6
9	11,1	10,3	9,2	8,6	8,8	8,9	8,2	8,2	7,9	8,0
10	12,8	11,9	12,1	11,8	11,3	11,2	11,1	10,7	10,9	10,8

распространения ошибки, минимизирующий ошибки в пространстве весов, сам по себе является усовершенствованным вариантом алгоритма метода градиентного спуска [7, 12–16]. В случае использования пакета *MatLab* алгоритм Gradient Descent With Momentum [11] также является разновидностью алгоритма обратного распространения ошибки. Алгоритм обучения Gradient Descent [9] в пакете *Statistica*, как и алгоритмы обучения в пакетах *ANNBuilder* и *MatLab*, также использует момент и коэффициент обучения и изменяет ве-

совые коэффициенты связей между нейронами при помощи метода градиентного спуска. Коэффициент скорости обучения определяет степень изменений, которые претерпевают весовые коэффициенты ИНС за одну итерацию обучения. Использование момента помогает функции ошибки выйти из локального минимума в пространстве ошибок. Это особенно полезно, когда поверхность ошибок имеет различную кривизну в разных направлениях, приводящую к образованию на этой поверхности длинных узких «долин».

Таким образом, для достоверного сравнения результатов были выбраны наиболее близкие друг к другу по параметрам и алгоритмам методы обучения ИНС.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ИНС

После проведения обучения ИНС был проведен анализ результатов этого процесса, полученных в разных программных пакетах. При сравнении результатов обучения для всех рассматриваемых программных пакетов относительно друг друга был проведен тест этих результатов по критерию Краскела – Уоллиса [17] для проверки близости медиан нескольких ОВ. Были получены сведения о статистической значимости или незначимости различий результатов обучения ИНС. При этом выбиралась одна ОВ, на ней обучались ИНС разных топологий в представленных программных пакетах и сравнивались результаты ошибочных распознаваний этих ИНС. Таким образом, было проведено сравнение результатов обучения ИНС различных топологий на десяти ОВ при использовании каждого из указанных программных пакетов. Обучение ИНС разной топологии в каждом из трёх программных пакетов на одной ОВ считалось отдельным экспериментом. Так как всего было взято 10 ОВ, то было проведено 10 экспериментов.

На рис. 2 и рис. 3 в качестве примера изображены значения медиан, полученных при проведении теста Краскела–Уоллиса результатов обучения ИНС на обучающих выборках № 2 и № 9 (см. табл. 1). Как видно из этих рисунков, программный пакет *ANNBuilder* показал существенно меньшее (от 0,01 % до 1,5 % ошибочных распознаваний) количество ошибок распознавания, чем два других программных пакета.

Для попарного сравнения результатов обучения ИНС с использованием программных пакетов был выбран тест Уилкоксона [18]. Он является традиционно используемым непараметрическим тестом для сравнения двух зависимых выборок. Из каждого эксперимента выбирались результаты не для всех, а лишь

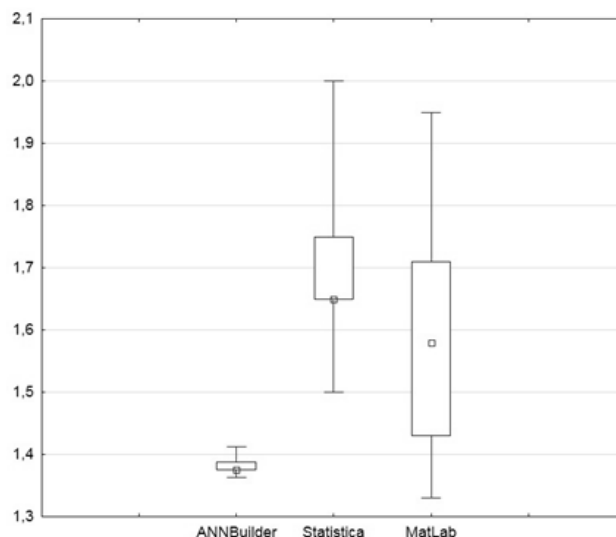


Рис. 2. Значения медианы при проведении теста Краскела – Уоллиса результатов обучения ИНС на ОВ № 2. По оси Y – процент ошибочных распознаваний ИНС

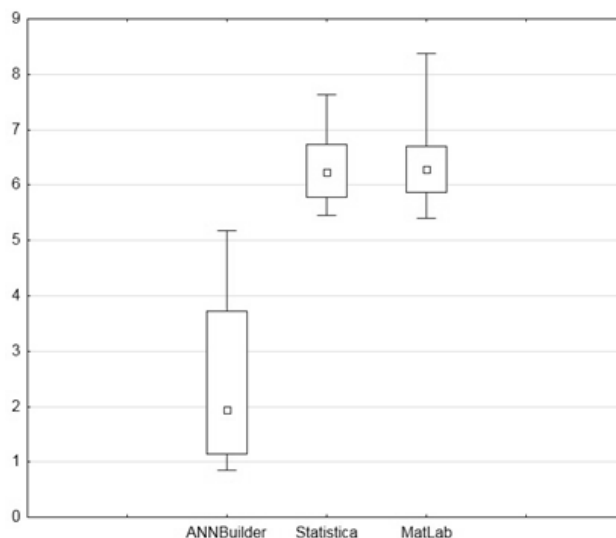


Рис. 3. Значения медианы при проведении теста Краскела – Уоллиса для результатов обучения ИНС на ОВ № 9. По оси Y – процент ошибочных распознаваний ИНС

для пары программных пакетов и проводился тест для того, чтобы выяснить, какой из программных пакетов дает меньшее количество ошибок классификаций в случаях, если тест Краскела-Уоллиса показывает, что в эксперименте присутствуют статистически значимые расхождения результатов. В итоге для каждого эксперимента проводилось два теста Уилкоксона (для пакетов *ANNBuilder* и *MatLab*, *ANNBuilder* и *Statistica*). Вводилась поправка на эффект множественных сравнений.

В табл. 5 приведены данные по сравнению результатов обучения для разных программных пакетов при работе с ИНС с помощью теста Краскела – Уоллиса и Уилкоксона. Видно, что показатели обучения ИНС на выборках № 5 и № 6 демонстрируют отсутствие статистически значимых различий между ошибочными распознаваниями ИНС, обученных в разных программных пакетах. Уровень p [19, 20] показывает статистическую значимость различий показателей обучения относительно пакета *ANNBuilder*, а числа 1, 2 и 3 отражают занятое программным пакетом место по числу правильно распознанных классов среди других пакетов. На ОВ № 4 пакеты *ANNBuilder* и *Statistica* показали одинаковые или близкие результаты ошибочных распознаваний ИНС.

ЗАКЛЮЧЕНИЕ

После проведения серии из 10 вычислительных экспериментов по обучению ИНС в программных пакетах *ANNBuilder*, *Statistica* и *MatLab* были получены статистически значимые результаты сравнения обучения ИНС. В большинстве случаев наименьшие показатели ошибочных распознаваний (от 0% до 5,95%) показал программный пакет *ANNBuilder*. Реализованные алгоритмы обучения ИНС в программном пакете *ANNBuilder* не уступают по результатам ошибочных распознава-

ний пакетам *MatLab* и *Statistica*. Поэтому они могут использоваться для расчетов базисных показателей для применения эволюционных алгоритмов в системах автоматизированного моделирования обучения нейронных сетей, восстанавливающих нервную ткань [21].

Использовать иные программные пакеты для работы с ИНС в рамках конструирования и проектирования нейронных сетей трудоемко, т.к. вычислительные эксперименты обычно выходят за рамки возможностей этих пакетов. Пакет *ANNBuilder* интегрирован в систему цифрового проектирования нейроинтерфейсов и нейронных сетей, не уступает по точности распознавания классов одним из наиболее распространенных программных пакетов для работы с ИНС, снижает временные затраты на перенос, обработку и загрузку параметров и результатов обучения ИНС между пакетами. Таким образом *ANNBuilder* можно применять для автоматизации проведения вычислительных экспериментов по моделированию обучения нейронных сетей. Так же стоит подчеркнуть, что пакет *ANNBuilder* имеет встроенные инструменты (использующие оригинальные алгоритмы) для работы с ИНС: анализ и повреждение ИНС, корректировка ИНС, обучение ИНС, включающее в себя, помимо алгоритма обратного распространения, алгоритм перебора весовых коэффициентов связей между нейронами и эволюционный алгоритм.

Таблица 5

Сравнение алгоритмов обучения ИНС в разных программных пакетах для работы с ИНС по их результатам в тестах Краскела – Уоллиса и Уилкоксона (p – уровень) и по количеству правильно распознанных классов (занимаемое место 1–3)

№ обучающей выборки	пакет <i>ANNBuilder</i>	пакет <i>Statistica</i>	пакет <i>MatLab</i>
1	2	3 ($p = 0,002$)	1 ($p = 0,005$)
2	1	2 ($p = 0,005062$)	3 ($p = 0,012516$)
3	3	1 ($p = 0,005062$)	2 ($p = 0,005062$)
4	1	1	2 ($p = 0,005062$)
5	–	–	–
6	–	–	–
7	1	2 ($p = 0,005062$)	3 ($p = 0,005062$)
8	1	2 ($p = 0,005062$)	3 ($p = 0,005062$)
9	1	2 ($p = 0,005062$)	3 ($p = 0,005062$)
10	1	2 ($p = 0,092602$)	3 ($p = 0,046854$)

СПИСОК ЛИТЕРАТУРЫ

1. <http://www.mathworks.com/products/matlab/>
 2. http://www.statsoft.ru/products/STATISTICA_Base/
 3. Туровский Я. А. Опыт моделирования альтернативных человеко-машинных интерфейсов / Я. А. Туровский, С. Д. Кургалин, А. А. Вахтин, С. В. Борзунов, А. В. Алексеев, В. А. Белобродский, И. Ю. Чурсин // Актуальные направления научных исследований XXI века: теория и практика. – 2014. – № 5-1(10-1). – С. 249–252.
 4. Туровский Я. А. Оценка скорости работы нейрокомпьютерного интерфейса, реализованного с использованием гибридного интеллекта / Я. А. Туровский, С. Д. Кургалин // Биомедицинская радиоэлектроника. – 2015. – № 3. – С. 61–70.
 5. Istook E. Improved backpropagation learning in neural networks with windowed momentum / E. Istook, T. Martinez // International Journal Of Neural Systems. – № 12. – С. 303–318.
 6. Heung B. K. Fast learning method for back-propagation neural network by evolutionary adaptation of learning rates // B. K. Heung, H. J. Sung, G. K. Tag, H. P. Kyu // Neurocomputing. – 1996. – № 11. – С. 101–106.
 7. Qian N. On the momentum term in gradient descent learning algorithms / N. Qian // Neural Networks. – 1999. – № 12. – С. 145–151.
 8. http://www.statsoft.ru/products/STATISTICA_Neural_Networks/
 9. <http://documentation.statsoft.com/STATISTICAHelp.aspx?path=SANN/Dialogs/SANNCustomNeuralNetworkMLPtab>
 10. <http://www.mathworks.com/products/neural-network/>
 11. <http://www.mathworks.com/help/nnet/ref/traingdm.html>
 12. https://ru.wikipedia.org/wiki/Метод_обратного_распространения_ошибки
 13. Хайкин С. Нейронные сети. Полный курс / С. Хайкин. – М. : Вильямс, 2006. – С. 274–275.
 14. Nazari J. Implementation of back-propagation neural networks with MatLab / J. Nazari, O. K. Ersoy // Electrical and Computer Engineering. – 1992. – № 9-1. – P. 7–8.
 15. Petrini M. Extended network for back-propagation algorithm / M. Petrini // Annals of the University of Petroşani, Economics. – 2012. – 12(4). – P. 177–184.
 16. Rojas. R. Neural Networks / R. Rojas. – Springer-Verlag, Berlin. – 1996. – С. 151–184.
 17. Kruskal W. H. Use of ranks in one-criterion variance analysis / W. H. Kruskal, W. A. Wallis // Journal of the American Statistical Association. – 1952. – № 260. – С. 583–621.
 18. Лапач С. Н. Статистика в науке и бизнесе // С. Н. Лапач, А. В. Чубенко, П. Н. Бабич. – Киев : Морион, 2002. – С. 164–166.
 19. Кобзарь А. И. Прикладная математическая статистика. Справочник для инженеров и научных работников / А. И. Кобзарь. – Физматлит, 2006. – 816 с.
 20. <http://www.machinelearning.ru/wiki/index.php?title=P-Value>
 21. Туровский Я. А. Автоматизирование обучения нейронных сетей / Я. А. Туровский, С. Д. Кургалин, А. А. Адаменко // Актуальные направления научных исследований века: теория и практика. – 2015. – № 5, Ч. 2(16-2). – С. 191–196.
- Работа выполнена при поддержке гранта РФФИ №АААА-А16-116051110239-2*
- Туровский Я. А.** – к.м.н., доцент, Воронежский государственный университет.
E-mail: yaroslav_turovsk@mail.ru
- Кургалин С. Д.** – д.ф.-м.н., профессор, Воронежский государственный университет.
E-mail: kurgalin@bk.ru
- Адаменко А. А.** – аспирант, Воронежский государственный университет инженерных технологий.
E-mail: adamenko.artem@gmail.com
- Turovsky Yaroslav Aleksandrovich** – PhD, docent, Voronezh State University.
E-mail: yaroslav_turovsk@mail.ru
- Kurgalin Sergey** – Dr., Professor, Voronezh State University.
E-mail: kurgalin@bk.ru
- Adamenko Artem Alexandrovich** – a graduate student, Voronezh State University of Engineering Technology.
E-mail: adamenko.artem@gmail.com