

## Исследование быстродействия нейросетевого распознавателя почерка.

77-30569/280351

# 12, декабрь 2011

Галкин В. А., Чернуха С. Н.

УДК 004.654

МГТУ им. Н.Э. Баумана

[chernen@bmstu.ru](mailto:chernen@bmstu.ru)

Введение. Применимость архитектурных программных решений оценивается в том числе путем выявления их производительности, обусловленной требованиями к физическим ресурсам ЭВМ. В данной статье рассматривается зависимость быстродействия on - line распознавателя почерка, построенного на базе нейронных сетей, от архитектуры его программной реализации — параметров нейронной сети, методов выделения и нормализации входных символов и пр.

Основные принципы. Как известно, при увеличении функциональных возможностей любого программного решения, будь то дополнительные функции, качество работы, интерактивность интерфейса и пр., неизбежно возрастают ресурсные потребности. Программа начинает отнимать большее количество процессорного времени, оперативной памяти, пространства на ПЗУ. Поэтому всегда разработчики и проектировщики вынуждены искать некий компромисс, чтобы позволить использовать программное решение на целевых аппаратных устройствах среднего (массового) в своем классе уровня, при условии обеспечения выполнения основных функций программного продукта на приемлемом уровне. Если же такого «компромисса» найти не удастся, и вместе с тем невозможно найти другое более рациональное архитектурное решение, частично функции разрабатываемого программного продукта могут быть реализованы аппаратно. Тогда разрабатывается специальное устройство, которое подключается к ЭВМ и выполняет алгоритмы программы аппаратно — быстрее, чем это может сделать встроенное вычислительное устройство ЭВМ (например, центральный процессор). Примером такого устройства может служить физический ускоритель Ageia PhysX [1].

В данной статье исследована зависимость быстродействия on - line распознавателя почерка, построенного на основе искусственных нейронных сетей, от архитектуры этих сетей и программного решения в целом. Под распознаванием почерка будем понимать процесс оцифровки рисуемой на экране последовательности связанных символов, каждый из которых должен быть ранее представлен в виде уникального эталонного начертания и соответствующего уникального цифрового кода распознавателю для обучения.

Процесс обучения распознавателя описан в статье [2] и здесь рассматриваться не будет. Он не требует своего протекания в реальном времени и в общем случае может быть выполнен на ЭВМ, отличной от той, на которой будет производиться распознавание, т.е. непосредственное использование продукта.

Выделение символов из слова. Как известно, нейронные сети в силу своего устройства не способны разделять объект на подобъекты, поэтому необходимо иметь алгоритм, позволяющий выделять отдельные символы (буквы) из слова. А уже с помощью нейронной сети будет происходить распознавание символов.

Из [2] можно выделить ограничение, накладывающихся на представление символов: начертание символа не может иметь разрывов. Иначе говоря, символ представляет собой непрерывный контур, который может быть нарисован одним движением манипулятора.

Как следствие из вышеизложенного — разрыв линии является признаком окончания начертания символа.

Метод автоматизированного выделения символов. Рассмотрим метод, предложенный авторами, для выделения символов из связанной последовательности с помощью нейронной сети. Суть метода заключается в итерационных попытках распознать наибольшую часть последовательности как неделимый символ. Под последовательностью символов (неразрывной последовательностью) будем понимать слитное начертание некоторого количества символов, которые например, могут представлять собой слово или часть слова. Так, распознавание начинается по окончании ввода связанной последовательности символов. Как указано выше, признаком окончания является разрыв линии контура. Алгоритм пытается выделить из начала последовательности фрагмент, протяженность которого равна максимальной длине обученного символа (естественно, с преобразованием пропорций). Затем алгоритм пытается распознать этот фрагмент. В случае неудачи отбрасываются последние k точек начертания фрагмента и нейронная сеть

вновь пытается его распознать. И так, пока на очередном шаге не будет получен ответ. Затем из последовательности удаляется начертание распознанной части (т.е. из начальной части контура всей последовательности удаляется контур распознанного символа) и операция продолжается вновь. Наконец, вся последовательность будет распознана, либо на очередном шаге будет выдан ответ о невозможности распознавания.

#### Преимущества:

1. Метод позволяет использовать в одной последовательности (слове) символы (буквы) разных размеров. Отсутствует ограничение на ширину и высоту букв, а также на наклон самой последовательности относительно границ области ввода.
2. Метод позволяет осуществлять «слепой» ввод данных. Пользователю не обязательно смотреть на экран при рисовании последовательности символов.

#### Недостатки:

1. Метод является очень ресурсоемким, т.к. преобладает итеративный подход. Если вводимая неразрывная последовательность (слово) содержит много символов (букв), то это потенциально может привести к тому, что распознаватель не будет успевать выполнять оцифровку в режиме реального времени.
2. Если в последовательности будет присутствовать ошибочный символ (символ, начертание которого не было представлено распознавателю при обучении), то все оставшиеся в данной последовательности символы не смогут быть распознаны, т.к. нельзя будет выделить эти символы из последовательности — найти их границы.
3. Ввиду аппроксимированного представления символа [2] в виде значимых отрезков, возможен случай, когда некоторая последовательность символов, представленная в виде этих отрезков, будет достаточно точно повторять представление одного из символов обучающей выборки. Тогда последовательность будет распознана неверно.

Тогда как влияние недостатка п.3 может быть значительно снижено различными алгоритмическими проверками, недостатки из п.1 и 2 устранить невозможно.

Метод ручного разделения символов. Рассмотрим метод ручного разделения символов, который является менее ресурсоемким по сравнению с описанным выше. Согласно этому методу поле для ввода символов делится на ячейки (клетки). Пользователь при вводе последовательности символов должен следить за тем, чтобы каждый символ попадал в отдельную клетку. Пересечения линий ячеек с начертанием введенной последовательности дают границы символов этой последовательности. При таком подходе распознавание можно начинать сразу после пересечения контура

последовательности с линией ячейки, не дожидаясь окончания ввода всей неразрывной последовательности символов.

Отметим недостатки и преимущества метода ручного разделения символов.

Преимущества:

1. Метод обеспечивает высокое быстродействие, как при вводе символов по отдельности.
2. Метод вынуждает пользователя более рационально использовать пространство области распознавания, что повышает аккуратность начертания вводимых символов и как следствие точность распознавания.

Недостатки. Пользователь вынужден максимально точно вводить символы в рамках границ ячеек, что вызывает некоторые неудобства. В процессе одного цикла ввода нельзя будет использовать различный масштаб ввода символов.

Недостаток этого метода является очень существенным с точки зрения применимости и удобства работы с распознавателем, так как делает практически невозможным «слепой ввод».

Оценка производительности. Чтобы сравнить методы выделения символов из последовательности, а также выявить ограничения на допустимые параметры архитектуры нейронной сети и распознавателя в целом, произведем оценку производительности. Для этого разработаем имитационную модель распознавателя.

Будем считать, что распознаватель планируется использовать преимущественно для русского и английского алфавитов и арабских цифр. Такое предположение не накладывает существенных ограничений на возможность распознавания созданных пользователем собственных алфавитов и языков.

Согласно исследованиям Н.Н. Соколова [4] максимально возможная для человека скорость письма составляет 15–20 слов в минуту. Это теоретические данные, полученные на основании поэлементного анализа букв русского языка с учетом предельной скорости движения руки человека. Экспериментальное определение скорости общего письма, (письма, содержащего словесный текст нетехнического характера), выполненное во ВНИИМ, показало, что скорость письма составляет  $(14 \pm 3)$  слов в минуту, т.е. максимальная скорость письма составляет 17 слов в минуту, что хорошо согласуется с теоретическими оценками. [6]

Из исследований С. А. Шарова и на основании его частотного словаря русского языка [6] средняя длина слова составляет 5,28 символа.

Для английского языка средняя длина слова составляет 5,10 символов [7, 8].

Из вышеизложенного следует, что человек способен писать до 100 букв русского и английского алфавита в минуту.

Моделировать будем только использование ресурсов ЦП, т.к. использование ОЗУ очевидно, не будет значительным, и минимальными требованиями к объему ОЗУ могут быть стандартные требования платформы разработки изделия, как например Microsoft .NET Framework 3.5 или Microsoft .NET CompactFramework [9].

Так как мы исследуем возможность работы с системой на КПК (карманном персональном компьютере) и ПК (персональном компьютере), моделирование будем проводить для процессоров ARM920T 250МГц [10] (КПК) и IntelSLACR [11] с выделением только одного ядра (ПК).

Схема имитационной модели представлена на рис. 1.

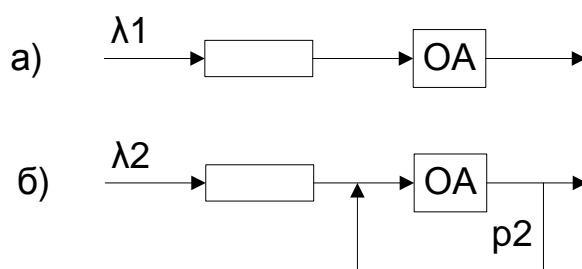


Рис. 1. Схема модели при а) – ручном выделении символов, б) – автоматизированном.

При использовании алгоритма ручного выделения символов на вход системы поступает поток символов с частотой  $\lambda_1 = 100 \frac{1}{\text{мин}}$ . В случае же применения алгоритма автоматизированного выделения символов на вход поступает поток последовательностей символов (слов) с частотой  $\lambda_2 = 20 \frac{1}{\text{мин}}$ . Далее рассчитаем среднее число повторов цикла распознавания последовательностей. Пусть каждый символ последовательности в среднем состоит из  $n$  точек, а самый протяженный контур одного символа в  $m$  раз длиннее самого короткого контура символа из тех, которые могут использоваться в неразрывных последовательностях (т.о. исключаем символы знаков препинания и цифр). Тогда средняя разность длин контуров символов может быть вычислена по формуле:

$$\Delta l = \frac{n(m-1)}{1+m}. \quad (1)$$

С учетом того, что алгоритм автоматизированного выделения символов проходит этот путь шагами, равными  $\frac{MinBorder}{3}$  (рис. 2), где *MinBorder* — минимальное число точек символов, которые могут встречаться в неразрывных последовательностях; среднее число попыток распознавания последовательности или ее части можно вычислить по формуле:

$$N = 5 * \frac{n(m-1)}{1+m} * \frac{3}{MinBorder}. \quad (2)$$

Максимальное отношение протяженности контуров русских рукописных символов наблюдается между буквами «с» и «ж» и составляет  $m=5$ . Средняя длина рукописного символа русского алфавита на экране КПК в удобном для пользователя масштабе составляет  $n=55$  точек; а минимально допустимая длина рукописного символа неразрывной последовательности составляет  $MinBorder=15$  точек. Подставляя эти значения в формулу 2, получим  $N=37$ .

Из теоретических расчетов следует, что алгоритм автоматизированного выделения символов требует процессорного времени в  $\frac{37\lambda_2}{\lambda_1} = \frac{740}{100} = 7,4$  раза больше, чем алгоритм ручного разделения символов.

Рассмотрим работу центрального процессора, являющегося обслуживающим аппаратом, как показано на рис. 1, с точки зрения временных затрат. В этом контексте блок-схема алгоритма одного цикла работы ОА (одной попытки распознавания) представлена на рис. 2. Функции F1–F4 описывают зависимость затрачиваемого времени на выполнение конкретной операции от ее параметров. Выражения для зависимостей F1 – F4 определены эмпирическим путем и сведены в таблицу 1. Таблица 2 содержит минимально допустимые значения исследуемых параметров архитектуры нейронной сети, выбранные согласно рекомендациям [12] и [13], и будут являться начальными при моделировании.

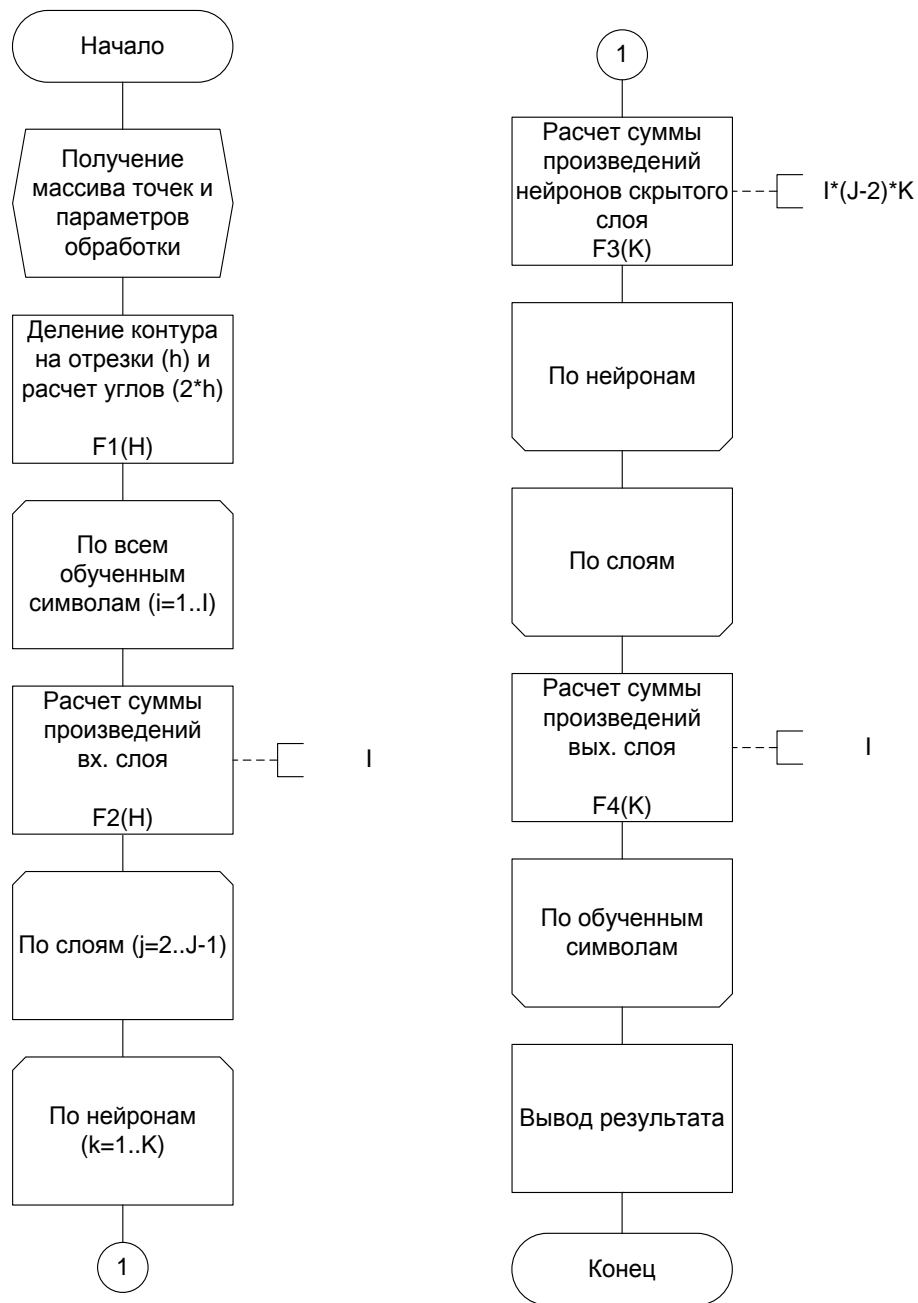


Рис. 2. Блок - схема алгоритма работы ОА.

Таблица 1. Сводная таблица параметров и выражений.

Обозначение	Значение ПК/КПК	Расшифровка
$\lambda_1$	100	Частота входного потока символов (1/мин)
$\lambda_2$	20	Частота входного потока последовательностей символов (1/мин)
m	5	Максимальное отношение протяженности контуров символов
n	55	Средняя длина контура одного символа (точек)
MinBorder	15	Минимально допустимая длина рукописного символа неразрывной последовательности (точек)
F1(H)	0,0009H	Затраты на деление контура на H отрезков и расчет направляющих углов (мс)
	0,2529H	
F2(H)	0,0004H	Затраты на просчет входного слоя нейронной сети от количества составляющих отрезков контура H (мс)
	0,0784H	
F3(K)	0,0003K	Затраты на просчет одного нейрона скрытого слоя нейронной сети от количества нейронов K в предыдущем слое (мс)
	0,0430K	
F4(K)	0,0004K	Затраты на просчет выхода сети от количества нейронов K в предыдущем слое (мс)
	0,0568K	

Таблица 2. Начальные значения исследуемых параметров.

Обозначение	Значение	Расшифровка
H	5	Число значимых отрезков символа
I	100	Число обученных символов
J	3	Число слоев нейронной сети
K	3	Число нейронов в скрытом слое нейронной сети



Моделирование будем проводить на языке GPSS [14]. Установим максимально допустимую длину очереди транзактов к устройству ЦП равной 250 символов или 50 последовательностей. Если в процессе моделирования окажется, что очередь переполнена, значит, выбранные параметры недопустимы при данном уровне производительности.

Наиболее значимые результаты моделирования сведены в таблицу 3. В столбце «№ п/п» в скобках указан тип применяемого алгоритма: а — ручное выделение символов, б — автоматизированное. В столбце «Время» указано среднее время в миллисекундах, затрачиваемое ЦП на распознавание одного символа последовательности. Если в процессе моделирования при заданных параметрах очередь транзактов к устройству ЦП не переполнялась, то в столбце «Пригодность» ставится «да», иначе «нет». Символ «\*» в графе пригодности означает, что в процессе моделирования заполнение очереди транзактов иногда достигало максимально возможного значения, но не вызывало стабильного переполнения. Параметры подбирались с учетом их важности на основании рекомендаций [12].

Таблица 3. Результаты исследования быстродействия.

№ п/п	Н	І	Ј	К	Время, мс ПК/КПК	Пригодность ПК/КПК
1(а)	5	100	3	3	0,59	да
					96,20	да
2(б)	5	100	3	3	4,76	да
					769,64	нет
3(а)	7	100	4	4	1,41	да
					216,97	да
4(б)	7	35	4	4	3,97	да
					616,71	да*
5(а)	7	100	5	5	3,81	да
					553,12	да
6(а)	7	100	3	7	2,03	да
					307,11	да

7(a)	7	100	3	10	3,69	да
					543,45	да*
8(б)	7	50	3	4	3,81	да
					597,06	да*
9(б)	7	100	3	3	5,41	да
					899,12	нет
10(б)	7	1000	4	11	622,85	да*
					9361,30	нет

Вначале выберем оптимальные архитектурные параметры нейронной сети. Будем руководствоваться тем, что в общем случае распознаватель должен предоставлять возможность в рамках одного сеанса распознавать символы и неразрывные последовательности алфавита одного языка, арабских цифр, знаков препинания и служебных знаков — всего 50 символов (33 буквы, 10 цифр, 7 дополнительных знаков). Для некоторых символов необходимо иметь возможность задания нескольких вариантов начертания (например, пользователь может по-разному писать букву в середине слова в соединении с другими буквами и на конце или вначале слова). Также нужно иметь некоторый запас для ввода служебных символов. Поэтому суммарное число символов (параметр I), с которыми распознаватель должен работать в рамках одного сеанса, положим равным 100.

Для преодоления ограничений перцептронной представляемости, характерных для распознавания символов (проблема линейной делимости), будет достаточным наличие всего трех слоев нейронной сети [12]. Поэтому число слоев, задаваемое параметром J, выберем равным трем.

Выбирая оптимальное число отрезков деления символа (параметр H), рассмотрим рис. 3.



Рис. 3. Начертание буквы «ж» в пяти и семи отрезках.

Контур буквы «ж» является самым протяженным среди всех рукописных начертаний букв русского алфавита. Как видно из рис. 3, пяти отрезков недостаточно, чтобы нейронная сеть смогла отличить рукописное начертание буквы «ж» от буквы «т». Чтобы отличить эти буквы, нужно разбивать символ на 7 отрезков. Выделение большего числа значимых отрезков не приносит пользы при использовании русского и английского алфавитов, т.к. в этом случае результат распознавания будет чрезмерно сильно зависеть от степени соответствия введенного начертания эталонному, сильное влияние будет иметь «шум» от дрожания руки и т.п. Поэтому выберем число отрезков деления символа (параметр  $N$ ), равное семи.

Сравнивая результаты экспериментов из таблицы 3, видно, что именно уровень быстродействия КПК будет накладывать ограничения на параметры распознавателя. При использовании алгоритма автоматизированного выделения символов ЦП КПК не способен обеспечить требуемый уровень быстродействия для алфавита из 100 символов при любых допустимых архитектурных параметрах нейронной сети. Чтобы можно было использовать этот более функциональный алгоритм, алфавит распознавателя нужно снижать до 50 символов, что, согласно расчетам, приведенным выше, в общем случае не будет являться достаточным. При использовании алгоритма ручного разделения символов, КПК способен обеспечить требуемый уровень производительности без функциональных ограничений, допуская наличие до 10 нейронов в каждом из скрытых слоев.

Таким образом, процессор КПК не всегда справляется с задачей распознавания в режиме реального времени при использовании алгоритма автоматизированного выделения символов. В то же время, алгоритм ручного разделения символов имеет существенное функциональное ограничение — практическую невозможность «слепого» ввода, что обязывает пользователя постоянно смотреть на экран используемого устройства. Поэтому, при выборе алгоритма выделения символов из неразрывных последовательностей для КПК следует уделить пристальное внимание характеристикам производительности этого устройства.

Заключение. В данной статье была обозначена проблема выделения символов (букв) из связанной последовательности (слова), для решения которой было предложено два различных метода — алгоритмы ручного и автоматизированного разделения символов.

Для выбора оптимальных методов, алгоритмов и параметров как всего распознавателя в целом, так и нейронной сети в частности, на которой он построен, было проведено моделирование процесса его работы. По результатам исследования можно выделить ограничения, обусловленные затратами вычислительных ресурсов, на допустимые значения архитектурных параметров on-line распознавателя почерка, построенного на основе нейронных сетей. Нужно заметить, что исследования проводились на платформе Microsoft Windows Mobile 5.0 (КПК) и Microsoft Windows 2008 Server (ПК), при этом алгоритмы были реализованы на языке Microsoft C# [3].

Так, при использовании алгоритма автоматизированного выделения символов для КПК с центральным процессором ARM920T 250МГц [10] максимально допустимыми значениями параметров являются:

1. число отрезков разбиения символа: 7;
2. число элементов алфавита распознавателя 50;
3. число слоев нейронной сети: 3;
4. число нейронов в каждом скрытом слое: 4.

При использовании менее функционального алгоритма ручного разделения символов максимально допустимыми значениями параметров являются:

1. число отрезков разбиения символа: 7;
2. число элементов алфавита распознавателя: 100;
3. число слоев нейронной сети: 3;
4. число нейронов в каждом скрытом слое: 10.

Персональный компьютер, в свою очередь, не накладывает значимых ограничений на параметры распознавателя. Так, ПК с процессором Intel SLACR [11], выделяя лишь одно ядро для выполнения программы-распознавателя, допускает следующие максимальные значения параметров при использовании наиболее функционального алгоритма автоматизированного выделения символов:

1. число отрезков разбиения символа: 7;
2. число элементов алфавита распознавателя: 1000;
3. число слоев нейронной сети: 4;
4. число нейронов в каждом скрытом слое: 11.

Таким образом, сегодняшний средний уровень производительности бытовых вычислительных устройств накладывает серьезные ограничения на допустимые значения параметров распознавателя почерка, построенного на основе нейронных сетей, что

существенно сказывается на его функциональных возможностях. Однако, характеристики быстродействия даже самых малопроизводительных устройств, таких как карманные персональные компьютеры, позволяют выполнить программную реализацию распознавателя почерка, способную предоставить пользователю полезную функциональность на приемлемом уровне.

#### Литература.

1. Свободная энциклопедия – Википедия [Электронный ресурс] : PhysX (чип) – Режим доступа: [http://ru.wikipedia.org/wiki/PhysX\\_\(%D1%87%D0%B8%D0%BF\)](http://ru.wikipedia.org/wiki/PhysX_(%D1%87%D0%B8%D0%BF)), свободный. – Загл. с экрана.
2. Галкин В.А., Чернуха С. Н. Обучение нейросетевого распознавателя рукописных символов //Проблемы построения и эксплуатации систем обработки информации и управления, Сб. статей, МГТУ им. Н. Э. Баумана; – М., 2011. Вып.8, С. 36-48.
3. Галисеев Г.В. Программирование на языке С# 2005. – М.: Диалектика, 2006. – 368с.
4. Соколов Н.Н. Курс стенографии по единой государственной системе. Ч. 1. – М.:УЧПЕДГИЗ, 1948. – 103с.
5. Тарковский Д. Как это делалось в процессе. Записки защитника [Электронный ресурс] – 2008.
6. Шаров С.А. Частотный словарь [Электронный ресурс] : РосНИИ ИИ, 2001.
7. Languages by Average word length [Электронный ресурс] – Режим доступа: <http://blogamundo.net/lab/wordlengths/>, свободный. – Загл. с экрана.
8. Petr Sojka Notes on Compound Word Hyphenation in TEX : TUGboat, Volume 16 (1995), No. 3 – Proceedings of the 1995 Annual Meeting.
9. System Requirements for Version 3.5 [Электронный ресурс] : MSDN – Режим доступа: <http://msdn.microsoft.com/en-us/library/bb882520.aspx>, свободный. – Загл. с экрана.
10. ARM920T [Электронный ресурс] – Режим доступа: <http://www.arm.com/products/CPUs/ARM920T.html>, свободный. – Загл. с экрана.
11. Intel® Core™2 Quad Processor Q6600 [Электронный ресурс] – Режим доступа: <http://processorfinder.intel.com/details.aspx?sSpec=SLACR>, свободный.
12. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. Пер. с англ. – М.: Мир, 1992. – 240с.

13. Rumelhart D. E., Hinton G. E., Williams R. J. 1986. Learning internal representations by error propagation. In *Parallel distributed processing*, vol. 1, pp. 318-362. Cambridge, MA: MIT Press.
14. В.Д.Боев Моделирование систем. Инструментальные средства GPSS World. – СПб.: БХВ-Петербург, 2004 г. – 368с.

## Study of operating speed of neuro-network handwriting parser

**77-30569/280351**

# 12, December 2011

Galkin V.A., Chernuha S.N.

Bauman Moscow State Technical University

[chernen@bmstu.ru](mailto:chernen@bmstu.ru)

The authors studied dependence of speed of on-line handwriting parser created on the basis of artificial neuron networks, on the architecture of these networks and program solution as a whole for manual and automated methods of symbols (letters) selection from a connected sequence (words). For each method the authors developed a simulation model which is realized in GPSS language. The simulation results showed that personal pocket computers (PDA) make it possible to carry out handwriting parsing software capable of granting the user useful functionality at an acceptable level.

---

Publications with keywords: [handwriting recognizer](#), [processing speed](#)

Publications with words: [handwriting recognizer](#), [processing speed](#)

---

### Reference

1. Galkin V.A., Chernukha S. N., in: Collection of articles “Problems of construction and operation of systems of information processing and management”, Moscow, MGTU im. N. E. Baumana – BMSTU, Is.8, 2011, pp. 36-48.
2. Galiseev G.V., Programming in C # 2005, Moscow, Dialektika, 2006, 368 p.
3. Sokolov N.N., Course of shorthand for the unified state system, Ch. 1, Moscow, UChPEDGIZ, 1948, 103 p.
4. LanguagesbyAveragewordlength, <<http://blogamundo.net/lab/wordlengths/>>.
5. Petr Sojka, Notes on Compound Word Hyphenation in TEX, in: Proceedings of the 1995 Annual Meeting, TUGboat, Volume 16 (3), 1995.
6. SystemRequirementsforVersion 3.5, <<http://msdn.microsoft.com/en-us/library/bb882520.aspx>>.
7. ARM920T, <<http://www.arm.com/products/CPU/ARM920T.html>>.

8. Intel® Core™2 QuadProcessorQ6600,  
<<http://processorfinder.intel.com/details.aspx?sSpec=SLACR>>.
9. Uossermen F., Neurocomputer technology: Theory and practice, Moscow, Mir, 1992,  
240 p.
10. Rumelhart D. E., Hinton G. E., Williams R. J., Learning internal representations by error  
propagation. In Parallel distributed processing, Vol. 1, Cambridge, MA, MIT Press.,  
1986, pp. 318-362.
11. V.D. Boev, Modeling of systems. Tools GPSS World, SPb., BKhV-Peterburg, 2004,  
368 p.