­САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математико-механический факультет

Кафедра системного программирования

**Динамическая аутентификация на основе анализа клавиатурного почерка с использованием нейронных сетей**

Дипломная работа студента 545 группы

**Торегожина Арстана Едигеевича**

Научный руководитель ………….. Ю.А. Губанов

/ подпись /

Рецензент ………….. М. К. Грачёв

/ подпись /

«Допустить к защите»

Заведующий кафедрой,

д. ф.-м. н., профессор ………….. А. Н. Терехов

/ подпись /

Санкт-Петербург

2009

**Содержание**

[1. Введение 3](#_Toc228197959)

[2. Постановка задачи 4](#_Toc228197965)

[3. Обзор существующих решений 5](#_Toc228197966)

[4. Основная часть 8](#_Toc228197984)

[4.1. Основные понятия в области нейронных сетей 8](#_Toc228197985)

[4.2. Анализ параметров характеризующих клавиатурный почерк 15](#_Toc228197986)

[4.3. Сбор и хранение клавиатурных данных пользователя 16](#_Toc228197993)

[4.4. Обзор Neuron .NET 17](#_Toc228197994)

[4.5. Основные концепции проектируемой нейронной сети 17](#_Toc228197995)

[4.6. Метрика оценки качества нейронной сети 18](#_Toc228197996)

[4.7. Динамическая генерация нейронных сетей по входным данным 18](#_Toc228197997)

[4.8. Зависимость качества нейронной сети от слоёв и количества нейронов 19](#_Toc228197998)

[4.9. Алгоритм распознавания пользователя на основе голосования 20](#_Toc228197999)

[5. Выводы 20](#_Toc228198011)

[6. Результаты работы 21](#_Toc228198012)

[7. Список использованной литературы 22](#_Toc228198013)

1. Введение

## С каждым днем проблемы, связанные с защитой информации становятся все более актуальными. Информационные технологии являются неотъемлемой частью деятельности компании, и обеспечение безопасности этого ценнейшего актива является стратегическим приоритетом. Основу системы информационной безопасности образуют средства идентификации пользователей и управления их доступом к корпоративным информационным ресурсам.

## На сегодняшний день достаточно популярны решения, использующие для идентификации различные материальные носители — смарт-карты, токены (компактное устройство в виде USB-брелка, которое служит для авторизации пользователя, защиты электронной переписки, безопасного удаленного доступа к информационным ресурсам, а также надежного хранения любых персональных данных), «таблетки» Touch Memory (i Button). В некоторых из этих средств защита идентификационной информации обеспечивается шифрованием, генерацией одноразовых паролей, электронной подписью и цифровыми сертификатами. Однако недостаток данных подходов в том, что при потере материального носителя пользователь может не только потерять доступ к данным но и сами данные.

## Идентификация человека по персональным характеристикам, называемая биометрией, успешно использовалась задолго до появления компьютерных технологий. Безупречная точность такого вида идентификации послужила поводом для разработки автоматизированных систем контроля доступа, где необходим высокий уровень конфиденциальности информации. Таким примером может служить повсеместное внедрение биометрических паспортов с идентификационным чипом. Биометрические системы контроля доступа в настоящее время основываются на считывании и сравнении данных физиологического или поведенческого характера человека. Примерами физиологических характеристик являются отпечатки пальцев, форма руки, характеристика лица, радужная оболочка глаза. К поведенческим характеристикам относятся особенности или характерные черты, приобретенные либо появившиеся со временем, то есть динамика подписи, идентификация голоса, динамика нажатия на клавиши. Несомненно, удобство такого метода — все идентификаторы всегда при себе, в отличие от внешнего носителя-карты, тем более, что индивидуальные особенности человека абсолютно уникальны и не поддаются подделке.

## На сегодняшний день большинство разработок в рамках биометрии основаны на специальном оборудовании. Например, метод идентификации, основанный на отпечатке сетчатки глаза, требует дорогостоящей камеры, идентификация голоса, требует микрофона, а существующие методы анализа динамики нажатия на клавиши, специальной клавиатуры.

## Ввиду данных ограничений биометрия не получили заслуженного признания и мало распространена.

## Однако, есть возможность удешевить и упростить существующие подходы в построении вышеописанных систем. Такая биометрия будет использовать только самое распространённое оборудование, такое как клавиатура и мышка. Биометрическая система будет состоять только из программного обеспечения способного функционировать на большинстве компьютеров и работать быстро и незаметно для пользователя.

1. Постановка задачи

## Задачей работы является разработка подхода к построению системы сравнения биометрических характеристик пользователей на основе анализа клавиатурного почерка. Реализация системы использующей это решение, анализ эффективности и его тестирование.

## Задача разделяется на следующие подзадачи:

## Разработка архитектуры нейронной сети способной динамически определять пользователя по вводу с клавиатуры

## Создать систему способную успешно использовать данную нейронную сеть на практике

## Создать компоненту осуществляющую сбор информации

## Реализация нейронной сети для обработки информации

## Разработать и реализовать алгоритм принятия решений

## Провести тестирование системы

## Провести анализ полученного решения и возможность использования его в боевых условиях

1. Обзор существующих решений

Системы биометрической аутентификации на основе анализа клавиатурного почерка могут быть использоваться по двум назначениям:

- для идентификации оператора, претендующего на доступ к компьютерной системе;

- для проведения скрытного клавиатурного мониторинга работающих операторов.

Системы, решающие указанные две задачи, различаются тем, что в первом случае идентификация оператора осуществляется по относительно короткой парольной фразе, а во втором – по произвольному тексту.

* 1. Методы основанные на сравнении с эталоном.

На сегодняшний день наиболее известны две системы биометрической аутентификации на основе анализа клавиатурного почерка: система “ID – 007”, разработанная американской фирмой “Enigma Logic” и система “Кобра”[1], разработчиком которой выступает Академия ФСБ России. Системы такого типа реализуют три основных функции:

## 1) сбор информации;

## 2) обработка информации (механизмы сравнений с эталонными значениями);

## 3) принятие решений по результатам аутентификации.

## Первая и третья функции в системах “ID – 007” и “Кобра” реализуются алгоритмически одинаково (различие составляют некоторые коэффициенты), а вот вторая функция — обработка информации или механизмы сравнений с эталонными значениями — принципиально отличаются. Сравнение вновь полученных значений времен удержаний клавиш с эталонными значениями в обеих системах производятся по аддитивной характеристике. Сравнение межсимвольных интервалов с эталонными значениями в системе “ID – 007” выполняется по принципу аддитивного соотношения, а в системе “Кобра” – по принципу мультипликативного соотношения.

## Механизм аддитивного сравнения характеристик заключается в анализе разностей вновь образованных биометрических характеристик к соответствующим эталонным значениям, то есть, времен удержаний клавиш к своим эталонам, и времен – интервалов между нажатиями клавиш к соответствующим своим эталонным значениям.

## Например, в случае если данные представлены в виде матрицы, то сам анализ происходит по следующему алгоритму: Из интервалов между нажатиями клавиш одной матрицы, вычитаются соответствующие эталонные значения другой матрицы. Если значение - результат меньше нуля, то сравниваемое время меньше условно-эталонного, а если больше – то больше. Отклонение от эталонного значения принимается в процентах, причем отклонение является положительным, если отношение больше нуля и отрицательным в противном случае.

## Мультипликативный механизм сравнения характеристик отличается от аддитивного только тем, что анализирует не разность, а отношение вновь образованных биометрических характеристик.

## В обоих случаях все отклонения, которые лежат в пределах допустимых значений отклонений обнуляются, а те отклонения, которые остались за допустимыми пределами – остаются без изменений и выступают в качестве так называемых пиков аддитивной или мультипликативной характеристики [2].

## Недостаток аддитивной фильтрации заключается в том, что уверенное распознавание пользователя происходит лишь при сопоставлении утренних и вечерних характеристик, а при сопоставлении утренних и дневных характеристик одного и того же пользователя аддитивный подход оказывается мало пригодным.

## Недостаток мультипликативного подхода заключается в том, что уверенное распознавание пользователя происходит в промежутке утро-день, а при соотношении утренних и вечерних характеристик одного и того же пользователя мультипликативный подход является мало пригодным.

## Существует методы, основанные на обоих подходах. Использования обоих методов в некоторой степени компенсирует вышеописанные недостатки, но суть остаётся, приходится хранить и записывать эталонные значения для последующего использования.

## Недостаток способов основанных на сравнении с эталоном заключается в том, что эталонов может быть очень много (среднестатистический пользователь может использовать около 10 000 слов в повседневной работе с компьютером), что ведёт к тяжеловесности системы. К тому же поиск в большом количестве эталонов требует немалых вычислительных ресурсов.

* 1. Методы основанные на применения искусственных нейронных сетей

## В последние годы нашли широкое применение нейросетевые алгоритмы для решения задач классификации. Их отличает простота применения и хорошие характеристики. Наибольшее распространение для решения задач распознавания образов сейчас получили нейросети в которых для обучения используется алгоритм обратного распространения ошибки [6][7]. Сеть ВР реализует детерминированный алгоритм распознавания, "строя" разделяющую поверхность в n-мерном пространстве входных векторов. Было проведено исследование возможности применения ВР-сетей в задаче идентификации пользователя [3]. В проведенном эксперименте длина использованной ключевой фразы составила 72 символа. Замеряемыми параметрами выбраны времена удержания и интервалы между соседними нажатиями клавиш клавиатуры при наборе ключевой фразы. Таким образом, в проведенном эксперименте размерность исходного признакового пространства составила - 144. Выборка была составлена по итогам замера параметров клавиатурного почерка 14 человек. В нейросетевом эмуляторе была создана трехуровневая сеть, в которой:

## первый уровень - входной, состоит из 145 нейронных элементов с линейной передаточной функцией[5];

## второй уровень - скрытый, состоит из 145 нейроэлементов с сигмоидной передаточной функцией;

## третий уровень - выходной, состоит из 14 нейроэлементов (по числу операторов, участвовавших в эксперименте) с сигмоидной передаточной функцией.

## При обучении, каждому входному вектору ставился в соответствие двоичный вектор длины 14, в котором координата соответствующая номеру оператора была единичной, остальные координаты имели нулевое значение. Вся выборка векторов была разделена случайным образом на две части - обучающая и тестовая (по 5 векторов для каждого оператора). При обучении нейронной сети намеренно использовалась неочищенная выборка для оценки способности нейронной сети к фильтрации сигнала, поскольку ошибки в наборе ключевой фразы можно рассматривать как случайные помехи, наложенные на сигнал. После обучения нейронной сети было проведено несколько экспериментов по ее тестированию. В каждом эксперименте выбирался свой порог срабатывания выходных нейронов, т.е. все выходные нейроны с выходным значением меньшим значения порога исключались из рассмотрения, из оставшихся нейронов выбирался нейрон с максимальным значением и его номер считался номером опознанного оператора. Данная подход используется для решение следующего ряда задач: 1) Идентификация пользователь при входе в систему 2) Обнаружение смены оператора оператором. Для этого производится постоянное сравнение параметров клавиатурного почерка работающего в данный момент оператора с параметрами клавиатурного почерка оператора, осуществившего начало работы. При расхождении образцов клавиатурных почерков принятие каких-либо мер (напр. отключение от банка данных, информирование администратора системы, запрещение ввода информации и т.п.). 3) Составление портрета пользователя по его клавиатурному почерку. Происходит оценка ряда психофизических параметров пользователя, как то - степени профессионализма при работе с клавиатурой, выявление состояния стресса у пользователя и т.п.

## Методы, основанные на применении обучаемых нейронных сетей, потенциально обладают большей точностью, но им присущи две группы принципиальных проблем: собственные проблемы искусственных нейронных сетей, связанны с возможностью возникновения неопределенно долгого процесса обучения, тупиков, состояния «паралича», а также проблемы, определяемые биометрической природой распознаваемых образов, главная из которых обучение - на всех возможных «чужих» пользователей (невозможность формирования представительной обучающей выборки для всех возможных «чужих») [4].

## Подводя итоги можно сделать вывод, что для динамической идентификации пользователя (анализ всех вводимых пользователем данных в реальном времени), рассмотренные способы малопривлекательны.

## В разработке кардинально нового подхода к построению подсистемы сравнения биометрических характеристик пользователей, может помочь новый взгляд на нейронные сети. Предлагается использовать нейронные сети не для классификации данных по пользователям, а для предсказания поведения пользователя по входным данным. Данный подход широко используется на рынках ценных бумаг [5].

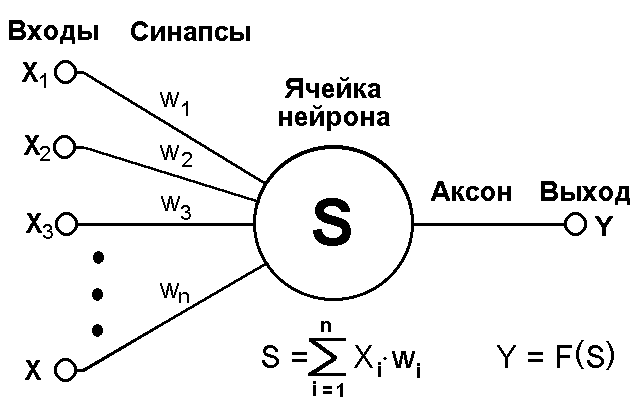
## Основная часть

## Основные понятия в области нейронных сетей

В последние десятилетия в мире бурно развивается новая прикладная область математики, специализирующаяся на искусственных нейронных сетях (НС). Актуальность исследований в этом направлении подтверждается массой различных применений НС. Это автоматизация процессов распознавания образов, адаптивное управление, аппроксимация функционалов, прогнозирование, создание экспертных систем, организация ассоциативной памяти и многие другие приложения. С помощью НС можно, например, предсказывать показатели биржевого рынка, выполнять распознавание оптических или звуковых сигналов, создавать самообучающиеся системы, способные управлять автомашиной при парковке или синтезировать речь по тексту.

Широкий круг задач, решаемый НС, не позволяет в настоящее время создавать универсальные, мощные сети, вынуждая разрабатывать специализированные НС, функционирующие по различным алгоритмам.

Несмотря на существенные различия, отдельные типы НС обладают несколькими общими чертами.



**Рис.1 Искусственный нейрон**

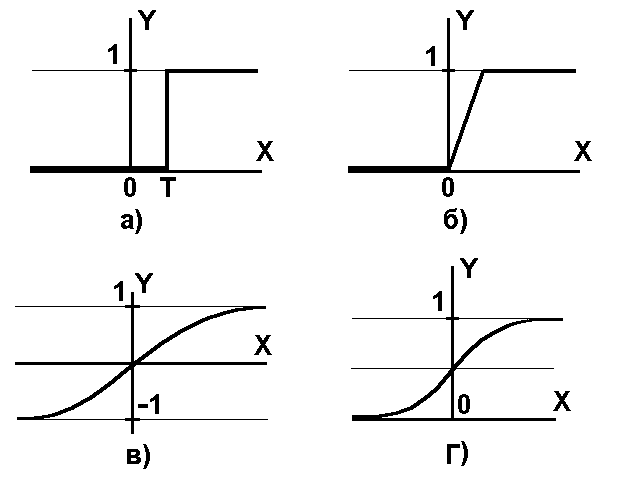
Во-первых, основу каждой НС составляют относительно простые, в большинстве случаев – однотипные, элементы (ячейки), имитирующие работу нейронов мозга. Далее под нейроном будет подразумеваться искусственный нейрон, то есть ячейка НС. Каждый нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками головного мозга, которые могут быть возбуждены или заторможены. Он обладает группой синапсов – однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон – выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Общий вид нейрона приведен на рисунке 1. Каждый синапс характеризуется величиной синоптической связи или ее весом wi, который по физическому смыслу эквивалентен электрической проводимости.

Текущее состояние нейрона определяется, как взвешенная сумма его входов:

 (1)

Выход нейрона есть функция его состояния:

*y = f(s)* (2)



**Рис.2 а) функция единичного скач­ка; б) линейный порог (гис­те­ре­зис); в) сигмоид – гипербо­ли­ческий тангенс; г) сигмоид – фор­мула (3)**

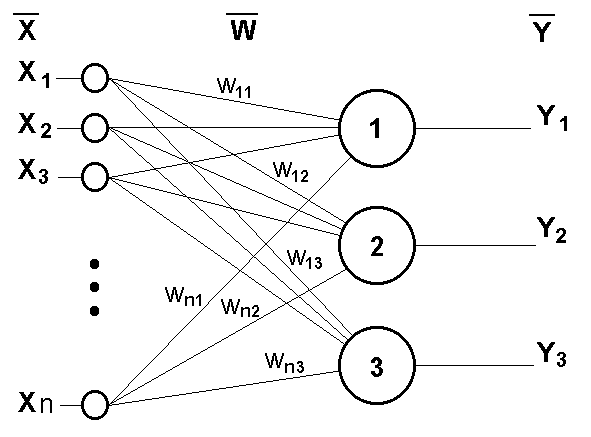
Нелинейная функция f называется активационной и может иметь различный вид, как показано на рисунке 2. Одной из наиболее распространенных является нелинейная функция с насыщением, так называемая логистическая функция или сигмоид (т.е. функция S-образного вида):

 (3)

При уменьшении α сигмоид становится более пологим, в пределе при α=0 вырождаясь в горизонтальную линию на уровне 0.5, при увеличении α сигмоид приближается по внешнему виду к функции единичного скачка с порогом T в точке x=0. Из выражения для сигмоида очевидно, что выходное значение нейрона лежит в диапазоне [0,1]. Одно из ценных свойств сигмоидной функции – простое выражение для ее производной

 (4)

Следует отметить, что сигмоидная функция дифференцируема на всей оси абсцисс, что используется в некоторых алгоритмах обучения. Кроме того она обладает свойством усиливать слабые сигналы лучше, чем большие, и предотвращает насыщение от больших сигналов, так как они соответствуют областям аргументов, где сигмоид имеет пологий наклон.



**Рис.3 Однослойный перцептрон**

Возвращаясь к общим чертам, присущим всем НС, отметим, во-вторых, принцип параллельной обработки сигналов, который достигается путем объединения большого числа нейронов в так называемые слои и соединения определенным образом нейронов различных слоев, а также, в некоторых конфигурациях, и нейронов одного слоя между собой, причем обработка взаимодействия всех нейронов ведется послойно.

В качестве примера простейшей НС рассмотрим трехней­ронный перцептрон (рис.3), то есть такую сеть, нейроны которой имеют активационную функцию в виде единичного скачка. На n входов поступают некие сигналы, проходящие по синапсам на 3 нейрона, образующие единственный слой этой НС и выдающие три выходных сигнала:

,  *j*=1...3 (5)

Очевидно, что все весовые коэффициенты синапсов одного слоя нейронов можно свести в матрицу **W**, в которой каждый элемент wij задает величину i-ой синаптической связи j-ого нейрона. Таким образом, процесс, происходящий в НС, может быть записан в матричной форме:

**Y=F(XW)** (6)

где **X** и **Y** – соответственно входной и выходной сигнальные векторы, **F( V)** – активационная функция, применяемая поэлементно к компонентам вектора **V**.

Теоретически число слоев и число нейронов в каждом слое может быть произвольным, однако фактически оно ограничено ресурсами компьютера или специализированной микросхемы, на которых обычно реализуется НС. Чем сложнее НС, тем масштабнее задачи, подвластные ей.

Выбор структуры НС осуществляется в соответствии с особенностями и сложностью задачи. Для решения некоторых отдельных типов задач уже существуют оптимальные, на сегодняшний день, конфигурации, описанные, например, в [6],[7],[8]. Если же задача не может быть сведена ни к одному из известных типов, разработчику приходится решать сложную проблему синтеза новой конфигурации. При этом он руководствуется несколькими основополагающими принципами:

* возможности сети возрастают с увеличением числа ячеек сети, плотности связей между ними и числом выделенных слоев;
* введение обратных связей наряду с увеличением возможностей сети поднимает вопрос о динамической устойчивости сети;
* сложность алгоритмов функционирования сети (в том числе, например, введение нескольких типов синапсов – возбуждающих, тромозящих и др.) также способствует усилению мощи НС.

Вопрос о необходимых и достаточных свойствах сети для решения того или иного рода задач представляет собой целое направление нейрокомпьютерной науки. Так как проблема синтеза НС сильно зависит от решаемой задачи, дать общие подробные рекомендации затруднительно. В большинстве случаев оптимальный вариант получается на основе интуитивного подбора.

Очевидно, что процесс функционирования НС, то есть сущность действий, которые она способна выполнять, зависит от величин синаптических связей, поэтому, задавшись определенной структурой НС, отвечающей какой-либо задаче, разработчик сети должен найти оптимальные значения всех переменных весовых коэффициентов (некоторые синаптические связи могут быть постоянными).

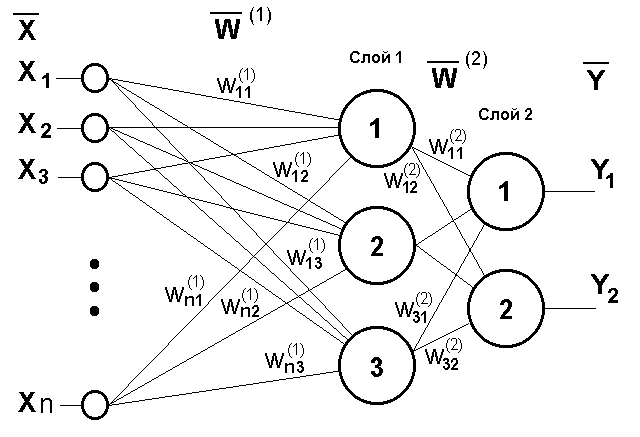
Этот этап называется обучением НС, и от того, насколько качественно он будет выполнен, зависит способность сети решать поставленные перед ней проблемы во время эксплуатации. На этапе обучения кроме параметра качества подбора весов важную роль играет время обучения. Как правило, эти два параметра связаны обратной зависимостью и их приходится выбирать на основе компромисса.

Обучение НС может вестись с учителем или без него. В первом случае сети предъявляются значения как входных, так и желательных выходных сигналов, и она по некоторому внутреннему алгоритму подстраивает веса своих синаптических связей. Во втором случае выходы НС формируются самостоятельно, а веса изменяются по алгоритму, учитывающему только входные и производные от них сигналы.

Существует великое множество различных алгоритмов обучения, которые однако делятся на два больших класса: детерминистские и стохастические. В первом из них подстройка весов представляет собой жесткую последовательность действий, во втором – она производится на основе действий, подчиняющихся некоторому случайному процессу.

Развивая дальше вопрос о возможной классификации НС, важно отметить существование бинарных и аналоговых сетей. Первые из них оперируют с двоичными сигналами, и выход каждого нейрона может принимать только два значения: логический ноль ("заторможенное" состояние) и логическая единица ("возбужденное" состояние). К этому классу сетей относится и рассмотренный выше перцептрон, так как выходы его нейронов, формируемые функцией единичного скачка, равны либо 0, либо 1. В аналоговых сетях выходные значения нейронов спо­соб­ны принимать непрерывные значения, что могло бы иметь место после замены активационной функции нейронов перцептрона на сигмоид.

Еще одна классификация делит НС на синхронные и асинхронные[7]. В первом случае в каждый момент времени свое состояние меняет лишь один нейрон. Во втором – состояние меняется сразу у целой группы нейронов, как правило, у всего слоя. Алгоритмически ход времени в НС задается итерационным выполнением однотипных действий над нейронами



**Рис.4 Двухслойный перцептрон**

Сети также можно классифицировать по числу слоев. На рисунке 4 представлен двухслойный перцептрон, полученный из перцептрона с рисунка 3 путем добавления второго слоя, состоящего из двух нейронов. Здесь уместно отметить важную роль нелинейности активационной функции, так как, если бы она не обладала данным свойством или не входила в алгоритм работы каждого нейрона, результат функционирования любой p-слойной НС с весовыми матрицами **W**(i), i=1,2,...p для каждого слоя i сводился бы к перемножению входного вектора сигналов **X** на матрицу

**W**(Σ)=**W**(1)⋅**W**(2) ⋅...⋅**W**(p) (7)

то есть фактически такая p-слойная НС эквивалентна однослойной НС с весовой матрицей единственного слоя **W**(Σ):

**Y**=**XW**(Σ) (8)

Продолжая разговор о нелинейности, можно отметить, что она иногда вводится и в синаптические связи. Большинство известных на сегодняшний день НС используют для нахождения взвешенной суммы входов нейрона формулу (1), однако в некоторых приложениях НС полезно ввести другую запись, например:

 (9)

или даже

 (10)

Введение такого рода нелинейности, вообще говоря, увеличивает вычислительную мощь сети, то есть позволяет из меньшего числа нейронов с "нелинейными" синапсами сконструировать НС, выполняющую работу обычной НС с большим числом стандартных нейронов и более сложной конфигурации[8].

Из рисунка функции единичного скачка видно, что пороговое значение T, в общем случае, может принимать произвольное значение. Более того, оно должно принимать некое произвольное, неизвестное заранее значение, которое подбирается на стадии обучения вместе с весовыми коэффициентами. То же самое относится и к центральной точке сигмоидной зависимости, которая может сдвигаться вправо или влево по оси X, а также и ко всем другим активационным функциям. Это, однако, не отражено в формуле (1), которая должна была бы выглядеть так:

 (11)

Дело в том, что такое смещение обычно вводится путем добавления к слою нейронов еще одного входа, возбуждающего дополнительный синапс каждого из нейронов, значение которого всегда равняется 1. Присвоим этому входу номер 0. Тогда

 (12)

где w*0*= –T, x*0*= 1.

Очевидно, что различие формул (1) и (12) состоит лишь в способе нумерации входов.

Из всех активационных функций, изображенных на рисунке 2, одна выделяется особо. Это гиперболический тангенс, зависимость которого симметрична относительно оси X и лежит в диапазоне [-1,1]. Выбор области возможных значений выходов нейронов во многом зависит от конкретного типа НС и является вопросом реализации, так как манипуляции с ней влияют на различные показатели эффективности сети, зачастую не изменяя общую логику ее работы.

Какие задачи может решать НС? Грубо говоря, работа всех сетей сводится к классификации (обоб­щению) входных сигналов, принадлежащих n-мерному гипер­про­странству, по некоторому числу классов. С матема­ти­ческой точ­ки зрения это происходит путем разбиения гипер­про­стран­ства ги­пер­плоскостями (запись для случая однослой­ного пер­цеп­тро­на)

, k=1...m (13)

**Таблица 1**

|  |  |  |
| --- | --- | --- |
| x*1* x*2* | 0 | 1 |
| 0 | A | B |
| 1 | B | A |

Каждая полученная область является областью определения отдельного класса. Число таких классов для одной НС перцептронного типа не превышает 2m, где m – число выходов сети. Однако не все из них могут быть разделимы данной НС.

Например, однослойный перцептрон, состоящий из одного нейрона с дву­мя входами, не способен разделить плос­кость (двумерное гиперпространоство) на две полуплоскости так, чтобы осу­ще­ствить классификацию входных сигналов по классам A и B (см. таблицу 1).

Уравнение сети для этого случая

 (14)

является уравнением прямой (одномерной гиперплоскости), которая ни при каких условиях не может разделить плоскость так, чтобы точки из множества входных сигналов, принадлежащие разным классам, оказались по разные стороны от прямой.

Если присмотреться к таблице 1, можно заметить, что данное разбиение на классы реали­зу­ет логическую функцию исключающего ИЛИ для входных сигналов. Невозможность реализа­ции однослойным перцептроном этой функции получила название проблемы исключающего ИЛИ.

Функции, которые не реализуются однослойной сетью, называ­ют­ся линейно неразделимыми[6]. Решение задач, подпадающих под это ог­ра­ничение, заключается в применении 2-х и более слойных сетей или се­тей с нелинейными синапсами, однако и тогда существует вероят­ность, что корректное разделение некоторых входных сигналов на классы невозможно.

Наконец, мы можем более подробно рассмотреть вопрос обучения НС, для начала – на примере перцептрона с рисунка 3.

Рассмотрим алгоритм обучения с учителем[6][7].

1. Проинициализировать элементы весовой матрицы (обычно небольшими случайными значениями).

2. Подать на входы один из входных векторов, которые сеть должна научиться различать, и вычислить ее выход.

3. Если выход правильный, перейти на шаг 4.

Иначе вычислить разницу между идеальным и полученным значениями выхода:



Модифицировать веса в соответствии с формулой:



где t и t+1 – номера соответственно текущей и следующей итераций; ν – коэффициент скорости обучения, 0<νj1; i – номер входа; j – номер нейрона в слое.

Очевидно, что если YI > Y весовые коэффициенты будут увеличены и тем самым уменьшат ошибку. В противном случае они будут уменьшены, и Y тоже уменьшится, приближаясь к YI.

4. Цикл с шага 2, пока сеть не перестанет ошибаться.

На втором шаге на разных итерациях поочередно в случайном порядке предъявляются все возможные входные вектора. К сожалению, нельзя заранее определить число итераций, которые потребуется выполнить, а в некоторых случаях и гарантировать полный успех.

## Анализ параметров характеризующих клавиатурный почерк

## Для исследований параметров характеризующих клавиатурный почерк проведен эксперимент. Эксперимент состоит в вводе с клавиатуры специально подобранного количества слов и словосочетаний с регистрацией времен удержания клавишей для покрытия всего вектора времен удержаний клавиш, а также с регистрацией времен межсимвольных интервалов, которые фиксируются в соответствующей матрице. Данный набор слов каждый пользователь набирает в различное время суток: утром, днем и вечером.

## Эксперимент выявил ряд интересных особенностей:

## существенная зависимость почерка от буквенных сочетаний в слове;

## существование глубоких связей между набором отдельных символов;

## наличие “задержек” при вводе символов;

## наиболее быстро абстрактный пользователь (т. е. некоторый образ усредненного пользователя) работает в середине рабочего дня (символическое название “День”), чуть медленнее утром, и гораздо медленнее – вечером.

## Перечислим, от каких параметров зависит печатный ритм пользователя [10]:

## Скорость, с которой пользователь находит буквы на клавиатуре. Даже если человек опытен и практикует слепую печать то он всё равно подбирает буквы с уникальной для него скоростью.

## Длина пальцев, что сказывается на том как быстро он дотягивается до дальних клавиш.

1. Психологическое состояние.

## Все эти параметры индивидуальны и влияют на ритм, с которым печатает пользователь, а это значит, что и ритм печати у всех пользователей тоже индивидуальный и его анализ может оказаться хорошим способом его аутентификации.

Подведём итоги и перечислим параметры, замер которых будет проведён в реализации данной работы:

1. Время между нажатиями.
2. Время удержания клавиши нажатой.
3. Ритм, который будет замеряться по алгоритму описанномы в следующей главе.
4. Время дня.
5. Последовательности букв.
6. Приложение, в котором пользователь печатает (в текстовых редакторах пользователь печатает не так как в играх или калькуляторе).
7. День недели. В зависимости от дня недели может изменится как ритм печатания, так и приложения которыми пользуется пользователь. Например, в выходные дни люди могут больше спать, а значит, мышцы пальцев и психологическое состояние могут отличаться от других дней недели.

## Сбор и хранение клавиатурных данных пользователя

Существует много приложений, которые работают в невидимом для пользователя режиме и следят за активностью клавиатуры, чтобы принимать какие-либо действия. Нам понадобится подобная функциональность для сбора данных нужных для работы нейронной сети.

Технология позволяющая следить за клавиатурным вводом пользователя вне зависимости от того каким приложением он в данный момент пользуется называется системными хуками (system hooks)

Вкратце изложим основные понятия и концепции данной технологии:

* Системные хуки позволяют установить специальную функцию отзыва которая позволяет перехватывать определённые послания в системе Windows (например, послания клавиатуры или мыши).
* Локальный системный хук - это системный хук, который вызывается только, когда определённое послание послано определённым потоком.
* Глобальный системный хук - это системный хук, который вызывается, когда определённое послание послано вне зависимости от потока.

На базе данной технологии была написана библиотека реализующая следующую функциональность:

* Слежение за всем клавиатурным вводом пользователя вне зависимости от активного в данный момент приложения
* Уведомление других компонент об активности клавиатуры в реальном времени
* Сохранение ввода в формате XML на жёстком диске
* Преобразование данных в удобный для обучения нейронной сети формат

Вся работы была проведена на Visual Studio 2008 на языке C#.

Полученная библиотека идеально подходит для сбора данных о клавиатурной активности пользователя и её сохранения.

## Обзор Neuron .NET

NeuronDotNet - это бесплатное программное обеспечение с открытым кодом, которое может быть использовано для создания разного рода нейронных сетей. Оно также предоставляет API для расширения существующей функциональности и внедрения новых алгоритмов в существующие нейронные сети. NeuronDotNet написана для платформы .NET на языке С#.

На сегодняшний день библиотека позволяет создавать два типа нейронных сетей:

1. Сети с обратным распространением ошибки, обучающиеся с учителем [8]
2. Самоорганизующиеся сети Кохонена [7]

Для сетей с учителем представлены пять видов слоёв со следующими активационными функциями:



Хотя NeuronDotNet позволяет создавать различного рода нейронные сети на любой вкус, у неё есть и много ограничений, одно из которых - скудный набор типов нейронных сетей.

Из всего вышеописанного, можно сделать вывод, что NeuronDotNet отлично подходит для выполнения поставленной задачи.

## Основные концепции проектируемой нейронной сети

Первое что надо сделать при создании приложений на основе нейронных сетей – это выбрать общую архитектуру. В данной работе нейронная сеть построена по следующим принципам:

1. Обучение методом “обучение с учителем”
2. Входной слой состоит из параметров характеризующих клавиатурный почерк описанных в разделе 4.2
3. Нейронная сеть не имеет рекурсий и возвратов
4. На каждом слое присутствует одна активационная функция
5. На выходе нейронной сети ожидается нормализованный параметр из раздела 4.2.

Таким образом нейронная сеть будет не классифицировать пользователей, а предсказывать характеризующие их признаки.

Например, пользователь ввёл слово “курс” мы подаём на вход нейронной сети буквы “К”, “У” и “Р” и если нейронная сеть на выходе выдала букву “С” то узнавание прошло успешно.

Данные сценарий приблизительный, на вход проектируемой сети будут подавать много других параметров, как и ожидаться на выходе.

Данный подход к построению нейронной сети позволяет:

* не заботиться о всевозможных пользователях, имеющих доступ к компьютеру
* настроить процесс обучения в реальном времени
* процесс аутентификации становится быстрым.

## Метрика оценки качества нейронной сети

Построить нейронную сеть мало, надо определить насколько хорошо она удовлетворяет поставленной цели. Так как нейронная сеть будет идентифицировать пользователя, то её качество будет измеряться в том, насколько хорошо она это делает. Для того чтобы это определить нужны три набора тестовых данных:

1. Обучающие примеры – примеры, предназначенные для обучения нейронной сети записанные пользователем.
2. Хорошие тестовые примеры - примеры, предназначенные для тестового прогона нейронной сети, эти примеры тоже записаны пользователем, но не входят в обучающие примеры.
3. Отрицательные тестовые примеры – примеры предназначены для тестового прогона нейронной сети, записанные другим человеком, ожидается, что нейронная сеть не узнает эти примеры.

Нейронная сеть будет обучаться на обучающих примерах, после этого результаты прогонов на хороших и плохих примерах подут проанализированы с использованием ниже описанной метрики.

Сама метрика называется RMSE(Root Mean Squared Error) - общая ошибка нейронной сети и вычисляется по следующей формуле:

RMSE = *,* где O множество фактических значений нейронной сети, P множество ожидаемых значений нейронной сети, N количество примеров.

Общая ошибка нейронной сети на хороших примерах должна быть меньше ошибки на плохих примерах.

## Динамическая генерация нейронных сетей по входным данным

Для построения нейронной сети был разработан и реализован механизм генерации нейронных сетей в зависимости от входных данных.

Основные особенности:

* генерация нейронной архитектуры;
* создание и обучение нейронной сети по предоставленной архитектуре;
* замер качества;
* сохранение сгенерированной сети на жёсткий диск.

Результатом реализации была программа, которая работала по схеме показанной на рисунке 6. На вход подаётся три набора тестовых данных. Итерация начинается с двухслойной нейронной сети, оба слоя с линейной активационной функцией.

Для удобства хранения, всё множество архитектур нейронных сетей было представлено в виде множества натуральных чисел в четверичной системе исчисления (В NeuronDotNet всего 4 активационные функции, не считая линейную функцию). Это означает, что каждая не линейная активационная функция была закодирована числом от 0 до 3, таким образом, количество слоёв в закодированной нейронной сети равно разрядности числа плюс один.



Например, как показано на рисунке 7, число 123 означает, что в закодированной нейронной сети 4 слоя: линейный входной слой, два скрытых слоя (1 и 2), и один выходной слой (3).

Генерация схемы новой нейронной сети достигается прибавлением единицы к предыдущей схеме нейронной сети. Таким образом, охватывается всё множество нейронных архитектур, которые могут быть созданы с использованием NeuronDotNet.

На каждом слое количество нейронов в скрытых слоях изменяется от одного до десяти. Каждый вариант нейронной архитектуры обучается и тестируется. Для удобства использования, результаты записываются в файл и сохраняются на диске.

Программа осуществляет поиск наилучшей архитектуры нейронной сети для предоставленных тестовых данных. Это позволяет создать нейронную сеть, наиболее приспособленную для решения конкретной задачи.

Разработанный механизм позволяет создавать нейронные сети для большого количества разнообразных задач.

## Зависимость качества нейронной сети от слоёв и количества нейронов

Для создания нейронной сети для распознавания клавиатурного почерка описанная в предыдущем разделе программа была запущена на 40 часов. Более длительный поиск представляется малоэффективным, в виду того, что обучение и тестирование сетей с большим количеством слоёв, занимает значительный промежуток времени, что не приемлемо в аутентификации в реальном времени. За данный промежуток времени было проанализировано около 5000 нейронных архитектур, самая большая из которых состояла из 7 слоёв.

Проведённый эксперимент выявил, что количество нейронов в скрытых слоях приводит, после определённого момента, к ухудшению качества обучения. Увеличение количества слоёв за редким исключением не приводит к улучшению качества.

Данные результаты совпадает с мнением экспертов в области нейронных сетей о том, что оптимальная работа большинства успешных приложений достигается при использовании от 3 до 5 слоёв. Например, нейронная сеть разработанная в университете Карнеги Меллон (Carnegie Mellon University) для управления движением машины по территории университета, использовала только 3 слоя [13].

В статье [14] эксперт по нейронным сетям Морин Кодилл (Maureen Caudill) утверждает: "Более того, старайтесь по возможности использовать нейросети с тремя слоями, если только у Вас не возникнет действительно насущная необходимость воспользоваться четвертым слоем. (Если вы не знаете, какая причина может заставить вас сделать это, то вам, вероятно, это и не требуется.) И даже не рассматривайте возможность перехода к 5-слойным системам. Каждый раз, когда ошибка распространяется от выходного слоя к предыдущему, она становится всё менее и менее значимой"

Хотя по итогам эксперимента были найдены нейронные архитектуры, показавшие качество лучшее, чем у нейронной архитектуры выбранной автором, но они обладают двумя существенными недостатками:

1. Слишком много слоёв в нейронной сети. С большим количеством скрытых слоёв и нейронов, приводит к увеличению времени обучения в реальном времени.
2. Отсутствие стабильности. Нейронная сеть теряет качество при обучении в реальном времени.

Поэтому было принято решение, остановится на трёхслойной нейронной сети с сигмоидной активационной функцией ().

## Алгоритм распознавания пользователя на основе голосования

Спроектированная в разделе 4.8 нейронная сеть может предсказывать только один параметр, а на вход принимает все остальные параметры, характеризующие пользователя. В ходе ряда экспериментов было определено, что не существует такого параметра, который нейронная сеть предсказывала бы с наибольшей вероятностью. В связи с этим было решено сделать несколько нейронных сетей, каждая из которых предсказывает один из параметров.  Сам процесс распознавания представлен на рисунке 8 и состоит в следующем:

1. пользователь начинает печатать на клавиатуре;
2. механизм сбора данных присылает образец;
3. образец приводится к виду принимаемым нейронными сетями;
4. образцы подаются нейронным сетям;
5. полученный от каждой нейронной сети результат анализируются, и с учётом точности, выдаётся бинарный результат (свой или чужой);
6. при помощи голосования нейронных сетей (своих больше чем чужих) делается вывод о правомерности данного пользователя;
7. все результаты передаются “менеджеру” (специальная компонента, принимающая решения о валидности пользователя). В данной реализации менеджер ведёт статистику ошибок пользователя, записывает её в файл и показывает пользователю. В перспективе можно реализовать механизм блокирующий дальнейший ввод до тех пор, пока пользователь не введёт специальный пароль.

## Выводы

В работе была предпринята попытка применить нейронные сети, для создания биометрического приложения, применимое для нужд аутентификации. При этом возник ряд трудностей, как специфичных для выбранных технологий, так и с выбором итоговой архитектуры приложения.

Одним из основных результатов работы является разработанное дополнение для Neuron.NET позволяющее генерировать, обучать и тестировать нейронные сети.

Технология Neuron.NET обеспечила простую работу с нейронными сетями, а системные хуки, XML сериализация и WCF (для обеспечения коммуникации между разработанными компонентами) дали возможность легко собрать нужные данные для их работы и обучения. Но оказались жёстко привязаны к платформе .Net и операционной системе Windows.

Выбранная архитектура нейронных сетей позволила не заботиться о выявлении всех возможных пользователей и предоставила лёгкий и быстрый способ динамического обучения и аутентификации. С другой стороны, точность такого подхода не так велика, как хотелось (эксперименты дали 70% узнаваний).

Ввиду вышеописанного можно сделать вывод, что данный подход к аутентификации способен решить ряд проблем, связанных с разграничением доступа, как к компьютеру, так и к отдельным программам. Использование разработанной архитектуры в купе с решениями на основе манипулятора “мышь” или анализатора запускаемых приложений, позволяет предположить большие показатели точности и успешность подхода в целом.

Полученное решение построено таким образом, что позволяет использовать его обширному кругу пользователей без особого понимания теоретической основы или знаний в области компьютерных технологий.

## Результаты работы

1. Разработано решение, позволяющее эффективно собирать информацию о клавиатурном вводе пользователя и сохранять его на диске.
2. Предложен алгоритм генерации нейронных сетей по входным данным и его реализация.
3. Предложен алгоритм распознавания пользователя по клавиатурному почерку и его реализация.
4. Разработано решение, позволяющее проводить сравнительный анализ распознавания.
5. Разработано решение в виде сервиса, реализующего распознавание на основе нейронных сетей.
6. Сделаны выводы о возможности применимости результатов данной работы.

## 7. Список использованной литературы

1. Password alternatives [Network Security](http://www.sciencedirect.com/science/journal/13534858) [Volume 1995, Issue 2](http://www.sciencedirect.com/science?_ob=PublicationURL&_tockey=%23TOC%236094%231995%23980049997%23228738%23FLP%23&_cdi=6094&_pubType=J&view=c&_auth=y&_acct=C000050221&_version=1&_urlVersion=0&_userid=10&md5=bbf0c05c7afe19d11f05fc4ad9fc46f1), February 1995, Pages 11-15.
2. Фор А. Восприятие и распознавание образов. М.: Машиностроение, 1989.
3. Трушина Е.А. Идентификация пользователя ЭВМ по клавиатурному почерку, как метод защиты от несанкционированного доступа. 1997. <http://www.securityclub.ru/>
4. О.М. Лепёшкин, А.В. Скубицкий Разработка подхода к распознаванию биометрического портрета пользователя по клавиатурному почерку на основе методов нелинейной динамики.
5. Нейронные сети в финансовых прогнозах. http://icmm.ru
6. Ф.Уоссермен, Нейрокомпьютерная техника, М.,Мир.
7. Итоги науки и техники: физические и математические модели нейронных сетей, том 1, М., изд. ВИНИТИ.
8. Artificial Neural Networks: Concepts and Theory, IEEE Computer Society Press.
9. С.Короткий Нейронные сети: основные положения
10. Брюхомицкий Ю.А., Казарин М.Н. Исследование биометрических систем динамической аутентификации пользователей ПК по рукописному и клавиатурному почеркам. Учебно-методическое пособие. –Таганрог: Изд-во ТРТУ
11. Кобиелус Джеймс. Информационная безопасность: идентификация и аутентификация. 1997. <http://www.old.jet.msk.su/>
12. Широчин В.П., Кулик А.В., Марченко В.В. Динамическая аутентификация на основе анализа клавиатурного почерка.
13. журнал Byte, август 1989 года, стр. 227 -233
14. журнал AI Expert в январе 1991 года