План введения:   
1.Описание ДМСД подхода и модели \\ стоит вынести в конец введения

2.Описание применения нейронной сети

Введение:

Математическое моделирование - удобный подход для описания комплексных экосистем. Описаный далее подход был разработан для исследования зоопланктона на определенном участке времени на озере Севан. Полное имя метода звучит следующим образом - дискретное моделирования динамического изменения системы. Моделирования с помощью данного метода позволяют определить влияние компонент системы друга на друга.

Данный метода описывает все возможные состояние межкомпонентного взаимодействия, а именно “плюс-плюс”, “минус - минус”, “плюс -минус”,  “плюс - ноль”, “минус - ноль”, “ноль - ноль”.

Основная суть метода состоит в том, что мы строим матрицу зависимости, которая отображает отношения каждой компоненты системы к каждой и на основе этой зависимости можем предсказывать следующее состояние системы, на основе наблюдения в текущий момент времени.

Для большего понимания рассмотрим пример исследования болезни у разных особей одного вида. Пускай имеются n особей, которые болеют одинаковой болезнью. Наблюдение за процессом болезни происходит через измерение нескольких характеристик организма особи, к примеру, пульс, температура, количество лейкоцитов в крови и т.п. - это и будет компонентами нашей системы. У разных особей болезнь может проходить по разному, в том плане, что один индивид прошел основную фазу за 2 дня и восстановительную фазу за 4, а другой за 3 и за 2 дня соответственно. При измерении показателей болезни мы знаем только “человеческое время”, но не биологическое, а для исследования необходимо много результатов, выстроенных в правильном порядке, а не просто много, не связанных друг с другом наблюдений. В данном случае метод помогает объединить  данные от многих особей в одну последовательность с корректной принадлежностью к временным промежуткам.

Идея заключается в выявлении матрицы зависимости компонент друг от друга .

В данной работе, для анализа данных и построение нужной нам матрицы было использовано программное воплощение математической модели -  
Искусственная нейронная сеть. Причины, которые побудили использовать соответствующую математическую модель описаны ниже.

В последние несколько лет мы наблюдаем взрыв интереса к нейронным сетям, которые успешно применяются в самых различных областях - бизнесе, медицине, технике, геологии , физике.

Нейронные сети возникли из исследований в области искусственного интеллекта, а именно, из попыток воспроизвести способность биологических нервных систем обучаться и исправлять ошибки, моделируя низкоуровневую структуру мозга (Patterson, 1996). Основной областью исследований по искусственному интеллекту в 60-е - 80-е годы были экспертные системы. Такие системы основывались на высокоуровневом моделировании процесса мышления (в частности, на представлении, что процесс нашего мышления построен на манипуляциях с символами). Скоро стало ясно, что подобные системы, хотя и могут принести пользу в некоторых областях, не ухватывают некоторые ключевые аспекты человеческого интеллекта. Согласно одной из точек зрения, причина этого состоит в том, что они не в состоянии воспроизвести структуру мозга. Чтобы создать искусственных интеллект, необходимо построить систему с похожей архитектурой.

Но, не смотря, на невозможность повторить полную структуру человеческого мозга, искусственные нейронные сети способны достичь замечательных результатов

Нейронные сети вошли в практику везде, где нужно решать задачи прогнозирования, классификации или управления. Такой впечатляющий успех определяется несколькими причинами:

Богатые возможности.  Нейронные сети - исключительно мощный метод моделирования, позволяющий воспроизводить чрезвычайно сложные зависимости. В частности, нейронные сети *нелинейны* по свой природе. На протяжение многих лет линейное моделирование было основным методом моделирования в большинстве областей, поскольку для него хорошо разработаны процедуры оптимизации. В задачах, где линейная аппроксимация неудовлетворительна (а таких достаточно много), линейные модели работают плохо. Кроме того, нейронные сети справляются с "*проклятием размерности*", которое не позволяет моделировать линейные зависимости в случае большого числа переменных

Простота в использовании.  Нейронные сети *учатся на примерах*. Пользователь нейронной сети подбирает представительные данные, а затем запускает *алгоритм обучения*, который автоматически воспринимает структуру данных. При этом от пользователя, конечно, требуется какой-то набор эвристических знаний о том, как следует отбирать и подготавливать данные, выбирать нужную архитектуру сети и интерпретировать результаты, однако уровень знаний, необходимый для успешного применения нейронных сетей, гораздо скромнее, чем, например, при использовании традиционных методов статистики.

Нейронные сети привлекательны с интуитивной точки зрения, ибо они основаны на примитивной биологической модели нервных систем. В будущем развитие таких нейро-биологических моделей может привести к созданию действительно мыслящих компьютеров.

Актуальность применения Искусственных нейронных сетей в контексте данной задачи обусловлена  тем, что решение задачи, базирующееся на применении генетического алгоритма (ссылка на работу Г.Н. Жолткевича), требует много времени и не может использовать потенциал много-поточных вычислений. А так же рассмотрения другого типа эмпирических алгоритмов, которые потенциально могут увеличить точность результатов. Соответственно применение нейронной сети призвано решить проблему невозможности расспаралеливания программной реализации и по возможности, исследуя различные архитектуры и алгоритмы обучения искусственной нейронной сети повысить точность результатов.

\\добавить описание, что такое цикл

\\НЕЙРОННЫЕ СЕТИ

<http://www.statsoft.ru/home/textbook/modules/stneunet.html>

https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F\_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F\_%D1%81%D0%B5%D1%82%D1%8C

Основная часть

1. Описание модели
2. Теория нейронной сети
3. Моделирование идеальной системы
4. Описание применения нейронной сети
   1. Архитектура
   2. Подготовка данных
   3. Реализация
5. Полученные результаты

**1.Описание модели**

Метод DMSD – описан в статье (ссылка на работу Г.Н. Жолткевича).  
У нас имеется некоторая биологическая система, которая описывается N компонентами … . Это может быть количество животных, в примере болезни – пульс, температура и т.д. Каждая компонента может принимать значение из конечного промежутка – 1,2...K. Система измеряется в определенные моменты времени, то есть измеряется значение каждой компоненты. Поэтому мы можем определить матрицу наблюдения за нашей системой, как бесконечную вправо матрицу.

Данная матрица описывет все возможные состояние системы на на временном промежутке t = 0,1 …   
Следовательно мы можем определить всю систему на момент времени t,

Как - , которая может единолично определится из состояния системы на момент (*t-*1).

Поскольку кажданя компонента может принимать строго определннае количество состояний, следовательно существует некоторое – , называемоем периодом системы, такое, что ***A*() –** примет все возможные состояние системы.

Так же справедливо утверждение:

, где – некое целое число > 0.

Соответственно мы можем выделить минор матрицы :

Который полностью описывает поведение системы.

Теперь рассмотрим матрицу отношения компонент, на основе которой мы можем определять поведения системы, используя лишь одно состояние.  
Пусть имеется множество – Ω = . Подробности взаимоотношений можно найти в статье (ссылка на статью Г.Н. Жолткевича).  
Соответственно мы можем определить матрицу взаимоотношений между компонентами –

D =

Где Ω и описывает влияние *i* компоненты на *j* –ю. Состояние компоненты в момент времени t+1 определяется из состояния компоненты t c учетом всех компонент, которые на нее влияют, используя матрицу D. Состояние компоненты может изменится только на целую величину из интервала - .

(t+1) = F((t+1))

В работе (ссылка на статью Г.Н. Жолткевича) рассматривается два подхода к функции F, в данной работе будет рассмотрен только один – «Weight Functions’ Approach» (стоит ли его описывать явно ?), соответственно моделирование будет использовать только данный подход. Моделирование идеальной системы(описано ниже), показало, что система может иметь полный цикл, в тех случаях, когда диагональные элементы равны нулю, что означает, что компонента не оказывает влияние на саму себя.

**2. Нейронная сеть**

Нейронная сеть ­- это громадный распределенный параллельный процессор, состоящий из элементарных единиц обработки информации, накапливающих эксперимен­тальные знания и предоставляющих их для последующей обработки. (Определение взято из книги С.Хайкина) **в ведение**

**Нейронная сеть** - это направленный граф, состоящий из узлов(нейронов), соединенных си­наптическими и активационными связями, который характеризуется следующими четырьмя свойствами.

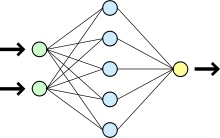
1. Каждый нейрон представляется множеством линейных синаптических связей, внешним порогами , возможно, нелинейной связью активации. порог, представляемый входной синаптической связью, считается равным + 1.

2. Синаптические связи нейрона используются для взвешивания соответствующих входных сигналов.

3. Взвешенная сумма входных сигналов определяет индуцированное локальное поле каждого конкретного нейрона.

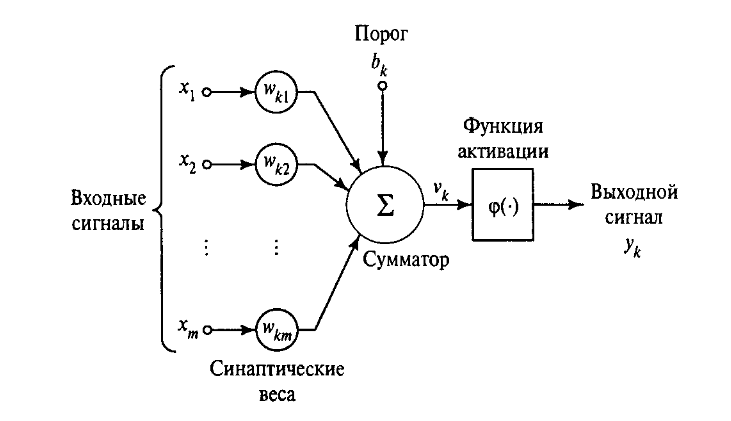
4. Активационные связи модифицируют индуцированное локальное поле нейрона, создавая выходной сигнал. (С.Хайкин)

На рисунке приведена схема простой нейронной сети.



Под архитектурой нейронной сети в работе имеется ввиду следующее понятие – структура графа, который представляет сеть.

Исскуственный нейрон (математический нейрон Маккалока — Питтса) — узел искусственной нейронной сети, являющийся упрощённой моделью естественного нейрона. Математически, искусственный нейрон обычно представляют как некоторую нелинейную функцию от единственного аргумента — линейной комбинации всех входных сигналов. Данную функцию называют функцией активации. Полученный результат посылается на единственный выход. Такие искусственные нейроны объединяют в сети — соединяют выходы одних нейронов с входами других. Искусственные нейроны и сети являются основными элементами идеального нейрокомпьютера.

Ниже приведен рисунок модели нейрона, взятый из книги С. Хайкина

**Обучение** ­- это процесс, в котором свободные параметры нейронной сети настраиваются посредством моделирования среды, в которую эта сеть встроена. Тип обу­чения определяется способом подстройки этих параметров.

В контексте задачи нейронная сеть используется для определение одного элемента матрицы *D.* Соответственно, для определения всей матрицы необходимо построить и запустить N^2 различных сетей. Данный подход был применен, что бы позволить алгоритм нахождения матрицы разбить на параллельные потоки.

В работе рассматривается три алгоритма обучения, а именно – BackProp, QuickProp, RPropMinus. Ниже будет приведена сравнительная характеристика алгоритмов, а так же характеристика применения каждого из алгоритмов. Все эксперименты проводились с использование языка программирования – Python, с применение библиотеки – PyBrain.

**3. Моделирование идеальной системы**

Идеальная система – система, в которой присутствуют все состояния системы, образующие цикл.

Поскольку для работы нейронной сети необходимо данные, на которых сеть может обучаться – следовательно, необходимо смоделировать идеальную систему, в которой мы можем гарантировать наличие всех состояний из полного цикла системы. Зная характеристики системы можно построить алгоритм, по которому мы сможем построить данные о системе. Алгоритм подходит для любого количества компоненты системы, поскольку никак не зависит от этого параметра. Так же при построении следующего состояния требуется выбрать параметр - , который определяет изменения компоненты. Выбирать его следует исходя из количества компонент в системе, а так же стоит учесть количество значений, которые может принимать одна компоненты. Используя ниже описанный алгоритм, проведем моделирование системы для количества компонент:

* Четыре компоненты, количество значений, принимаемое одной компоненты принадлежит множеству целых чисел из промежутка  
  [1,5], .
* Четыре компоненты, количество значений, принимаемое одной компоненты принадлежит множеству целых чисел из промежутка  
  [1,6], .

**Алгоритм моделирования идеальной системы:**

**Шаг 0:** Инициализировать матрицу D случайными значениями из множества – . Инициализировать количество итераций - IC , нулем.   
Перейти к шагу 2.

**Шаг 1:** Если IC < 10000, то перейти к шагу 2, иначе конец выполнения.

**Шаг 2:**Инциализивароть вектор – *A*(0) случайным образом. Присвоить t значение – нуль. Присвоить IC =IC+1.  
Перейти к шагу 3.

**Шаг 3:**Построить состояние (*t* +1).   
Перейти к шагу 4.

**Шаг 4:** Найти k , такое, что A(t) = A(k), где k = 1..t-1.  
Перейти к шагу 5.

**Шаг 5:** Если k то перейти к шагу 6,иначе перейти к шагу 3.

**Шаг 6:** Если k == t-1 , то перейти к шагу 1, иначе конец выполнения всего алгоритма.

Циклом системы будет все состояния системы от (A(k),A(k+1),…A(t-1))

**Доказательство конечности алгоритма:**

Алгоритм будет завершен за конечное количество шагов, поскольку, количество состояний системы конечно, значит мы на определенном t получим состояние, которое уже было.

**Результаты моделирования**

При моделировании системы, состоящих из 4-х компонент было получено 101 различных циклов системы. При моделировании системы, состоящей из 5-ти компонет было получено 153 различных циклов системы. В обоих случаях количество состояний в одном цикле системе равно 18.

**4.Применение ИНС**

Работа с нейронной сетью в работе делится на два этапа. Этап обучения нейронной сети и этап тестирования. Этап обучения предполагает прогонку сети на различных данных, полученных путем моделирования идеальной системы и корректировка весов, для уменьшения ошибки. Этап обучения будет более подробно раскрыт ниже, при описании применения каждого из алгоритмов.  
Этап тестирования представляет собой получения ответа на данных, не рассмотренных на этапе обучения, но так же полученных путем моделирования идеальной системы, для того, что бы была возможность определить процент правильных ответов, полученных их обученной нейронной сети.

В качестве данных, используемых на этапе обучения, мы будем использовать одинаковые наборы данных для всех трёх алгоритмов, для того, что бы наиболее точно сравнить данные алгоритмы.

Виды архитектуры, рассмотренные в работе различаются лишь в количестве внутренних слоёв и различие в количестве нейронов, входящих в скрытый слой. Архитектура входного и выходного слоя всегда будет одинаковой. Входной слой состоит из *T*^2 нейронов, выходной слой состоит из 3-х нейронов. Количество нейронов выходного слоя обусловлено тем, что множество значений, которые может принимать каждый элемента матрицы *D* , состоит из множества мощности – 3. Количество нейронов входного слоя, зависит от выбора данных, на которых будет обучаться сеть. Данные следует выбрать такими, которые оказывают непосредственное влияние на результат. В большинстве случаев выбор входных данных производится интуитивно, на основе неких предположений.  
В данном случае мы хотим получить результат, который характеризует влияние одной компоненты системы на другую компоненту системы. В работе в качестве обучающих данных будет использоваться матрица, элементы которой являются коэффициентами корреляции Пирсона, для всех состояний системы из цикла. Это обусловлено тем, что согласно условию задачи информация о правильном временном порядке состояний системы отсутствует, а с помощью матрицы корреляции мы можем определить связь между отдельными состояниями системы , без учета исходного порядка.

R = ,

где -Коэффициент корреляции Пирсона, между *A(i), A(j)*.

,

где – выборочная дисперсия вектора , – соответственно для вектора .

Использование скрытых слоёв будет рассмотрено ниже в описании применения каждого из алгоритмов, поскольку определенная структура обусловлена текущим алгоритмом обучения.

В работе рассматривается процесс обучения с учителем, где в качестве «учителя» будут использоваться данные, полученные при моделировании идеальной модели. В процессе обучения нейронной сети с учителем, необходимо минимизировать ошибку между полученными выходными данными и правильными данными. Под ошибкой в большинстве случаев, как и в данной работе, понимается сумма квадратов ошибок для всех элементов обучающегося множества. Каждый из ниже приведённых алгоритмов по-своему старается минимизировать ошибку. Преимущества и недостатки, а так же данные точности предсказания на тестируемых данных, длительности обучения будут предоставлены ниже.

**Структура сети**:

Как было сказано ранее структура различных вариантов нейронной сети различается лишь в количество внутренних слоев и в количество нейронов, их которых состоит внутренний слой. Рассмотри 7 типов архитектуры:

* Два скрытых слоя.
  + Количество нейронов в первом слое – 200, во втором – 20.
  + Количество нейронов в первом слое – 200, во втором – 40.
  + Количество нейронов в первом слое – 100, во втором – 20.
  + Количество нейронов в первом слое – 100, во втором – 40
* Один скрытый слой.
  + Количество нейронов в скрытом слое – 200.
  + Количество нейронов в скрытом слое – 250.
  + Количество нейронов в скрытом слое – 300.

**Данные, на которых будет проводить эксперимент:**

Рассмотри работу алгоритма на данных, которые были получены путём моделирования идеальной системы для системы, состоящих из четырёх компонент. Множество данных состоит из сто одного различного цикла. Для каждого цикла существует матрица D, которая описывает поведение системы. Как было сказано ранее, для определения полной матрицы не обходимо построить N^2 искусственных нейронах сетей. Отличия при построении заключаются лишь в множестве обучающих данный. Так для определения элемента матрицы D с индексом (*i,j*) в качестве примеров выходных сигналов при обучении необходимо использовать только элементы матрицы D для обучающих циклов под индексом(*i,j*) . Разобьем это множество случайным образом на 90 циклов и на 11. Первая часть будет использована для обучения. Вторая часть для тестирования.

Для работы с системой, состоящих из пяти компонент множество данных, полученных путем моделирования идеальной системы имеет 153 различных циклов. Для обучения возьмем случайным образом 130 циклов из множества, для тестирования – все оставшиеся.

Алгоритм обучения для нейронной сети для получения элемента матрицы D c индексом *i,j*.

**Шаг 1**: Посчитать корреляционные матрицы для всех циклов из обучающей выборки

**Шаг 2**: Обучать нейронную сеть на множестве обучающих примеров с помощью одного из ниже приведенных алгоритмов обучения. Каждый из алгоритмов имеет свою специфику, поэтому количество итераций будет определятся непосредственно при описании конкретного алгоритма.

После обучения нейронной сети, происходит тестирование на данных из множества для тестирования. В процессе тестирования происходит сравнения полученных данных с данными, которые являются правильными.

Процент правильных ответов вычисляется следующим образом.

Для того , что бы построить полную матрицу D, необходимо проделать выше описанные действия для всех индексов (*i,j*). Общий процент ошибки вычисляется, как среднее арифметическое всех .

**4.1 Алгоритм BackProp**

Описание алгоритма

Основная идея этого метода состоит в распространении сигналов ошибки от выходов сети к её входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы.

Метод вычисляет вектор градиента поверхности ошибок и продвигается в этом направлении на заданный шаг. Последовательность таких шагов, постепенно приводит к минимуму. Алгоритм действует итеративно, на каждом шаге продвигаясь в направлении вектора градиента.

Проблемы с которыми можно столкнутся, используя данных метод следующие:

* При выборе большой длине шага, можно «перескочить» через минимум.
* При выборе маленького шага, время затрачиваемое на обучения может быть очень большим, что делает неактуальным применение данного метода в контексте данной задачи.

Достоинства метода:

* Просто в реализации
* При выборе маленького шага, с большой вероятностью будет достигнут нужный минимум

В качестве функции активации нейрона была выбрана традиционная сигмоидальная функция:

Условие окончания: Количество итераций больше 2000, или значение ошибки меньше, чем 0.002.

**4.2 Алгоритм QuickProp**

Рассматриваемый алгоритм является улучшенной версией алгоритма обучения рассмотренного раннее, призванный решить проблему, связанную с большим временем обучения.

Алгоритм Qiuck Propagation содержит элементы, предотвращающие зацикливание в точке неглубокого локального минимума, где из-за близости к нулю производной функции активации процесс в алгоритме Back Propagation прекращается, такая ситуация может возникнуть из-за больших значений весовых коэффициентов, в сравнении со значениями других весовых коэффициентов. Данный алгоритм использует модифицируемую формулу для изменения весовых коэффициентов. Формула для каждой связи строится индивидуально на основе дополнительных параметров, за счет чего достигается увеличении скорости сходимости.

Минусы алгоритма заключаются в том, что при определённых условиях возможна значительная потеря точности, но зависит от структуры нейронной сети.

В качестве плюсов алгоритма следует учесть значительно возрастающую скорость обучения, что делает его применения актуальным в контексте задачи, рассматриваемой в данной работе.

В качестве активационной функции используется сигмоидальная функция, которая была рассмотрена выше, при описании алгоритма Back Propagation.

* **4.3 Алгоритм RProp(**M. Riedmiller, H.Braun. "A direct adaptive method for faster backpropagation learning: The RPROP algorithm". San Francisco,1993.

**)**

Алгоритм RProp – является модификацией алгоритма обратного распространения ошибки, призванный решить проблему медленной сходимости.

Алгоритм RProp использует только знаки частных производных, для изменения значения весовых коэффициентов нейронной сети. Структура алгоритма следующая: если на текущем шаге производная, соответствующая весу , изменила свой знак, то это говорит о том, что последнее изменение было большим и алгоритм перепрыгнул локальный минимум, из чего следует , что величину изменения нужно уменьшить на величину заданного при инициализации алгоритма – и присвоить весовому коифициенту его же значение на предыдущей операции. Если же знак производной не изменился, тогда следует увеличить значение коррекции весовых коэффициентов на . С помощью данного приема достигается увеличение скорости сходимости. Параметры и выбираются при инициализации алгоритма. При использовании алгоритма для решения задачи, рассматриваемой в данной работе были использованы следующие значения:

Для того, что бы не попасть в ситуацию, когда значения коррекции весов слишком велико, или равняется слишком малому значению, данную величину следует ограничить сверху и снизу. В работе для ограничения были использованы следующие параметры:

Алгоритм Resilient Propagation в среднем сходится в 4-5 раз быстрее и не показывает ощутимых потерь в точности, что делает его применение полностью оправданным.

Минусом данного алгоритма является сложность в реализации в сравнении с алгоритмом Back Propagation.

**5.Анализ полученных результатов:**

При моделировании системы, состоящей из 4-х компонент были использованы данные, полученные путём моделирования идеальной системы. Рассмотренные ранее различные структуры нейронных сетей не показали значимых отличий. Ниже приведены характеристики для структуры состоящей из одного скрытого слоя, состоящего из 250 нейронов.

5.1 Система, состоящая из 4 компонент

Результаты (см. Рис. 3) отображают среднее время, затрачиваемое на обучение нейронной сети и восстановление матрицы . На этапе тестирования было восстановлено 500 матриц, что позволяет полученные результаты, как статистически правильные.

Рис. 3

Рис. 4

Результаты, характеризующие ошибки при тестировании нейронных сетей отображены ниже на рисунке (см. Рис. 4) .

5.2 Система, состоящая из 5 компонент.

При тестировании нейронных сетей на системе, состоящей их пяти компонент процент ошибки не менялся на ощутимые значения, но ожидаемо изменилось время (см. Рис. 5).

Рис. 5

Согласно результатам тестирования, алгоритм Resilient Propagation показал себя лучше остальных. Он ощутимо не проигрывает по времени алгоритму обучения Quick Propagation, но в тоже время наиболее точно восстанавливает матрицу системы.

Эксперименты проводились на компьютере с двухъядерным процессором. Для вычислений программа использовала два потока. Работа с нейронными сетями осуществлялась с помощью языка программирования – Python и использование библиотеки – PyBrain.

Выводы:

В работе было рассмотрено применение нейронных сетей для описания динамической системы на основе данных об этой системе. В реальном мире данные о системе не будут такими точными, как данные полученные при моделировании идеальной системы. Потому для развития идеи анализа динамической системы с помощью искусственных нейронных сетей

Что было сделано :  
1.Смоделирована идеальная система  
2.Определен подход, как испольсовать ИНС, так что бы была решена проблема расспаралеливания   
3.Протестированы различные алгоритмы обучения сети  
4.Протестированы различные архитектуры ИНС

Точноть больше 75 процентов

Решена проблема многопоточности

Литература: