ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

**Факультет информатики, математики и компьютерных наук**

**Программа подготовки бакалавров по направлению   
38.03.05 Бизнес-информатика**

*Тарасов Александр Вячеславович*

**КУРСОВАЯ РАБОТА**

Предсказание результатов баскетбольных матчей на основе нейронных сетей

|  |  |
| --- | --- |
|  | Научный руководитель  доцент кафедры ИСиТ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  О. Р. Набиуллин |

Нижний Новгород, 2017

Содержание

Введение………………………………………………………..….....................3

Глава 1. Введение в экономические сети……………………..…..……..........5

1.1.Понятие экономических сетей. Тенденции их развития……...……..5

1.2 Пример модели экономической сети.……………………………........8

Глава 2. Основные понятия……………………………………………………10

2.1.Фондовый рынок, ценная бумага, фондовая биржа……………….…10

2.2.Лондонская фондовая биржа ……………………………..…………...14

2.3.Франкфуртская фондовая биржа ………………………....…………...16

2.4 Построение графа рынка……………………………………………….18

Глава 3.Анализ Лондонской и Франкфуртской фондовых бирж. Их сравнение………………………………………………………………………..21

Заключение………………………………………….…………………………41

Список литературы………………………………………………......................42

**Введение**

На сегодняшний день применение нейронных сетей во всем мире очень разнообразно. Оно далеко не ограничивается комьютерным зрением, где эта математическая модель получила, наверное, самое широкое распространение, благодаря широкому спектру возможных дивидендов.

Нейронные сети применяются для разпознавания самых разных объектов при разработке электрокаров, постановки медицинского диагноза, фильтрации спама в почтовых клиентах.

Одним из направлений использования нейронных сетей является и прогнозирование. Например, курсов валют, цен на сырье. Свое развитие начали и разработки нейронных сетей с целью предсказания спортивных событий.

В своей работе я собираюсь применить модель нейронных сетей для предсказания победителей матчей Национальной Баскетбольной Ассоциации (НБА).

Этот спорт и эта лига выбраны не случайно.

Баскетбол является одним из самых «оцифрованных» на данный момент видов спорта. Количество всевозможной статистики, собираемой во время баскетбольных матчей, огромно. Лига НБА же является самой популярной баскетбольной лигой в мире, поэтому статистические показатели прошлого, которые так важны для обучения нейронных сетей, найти для нее намного проще. Более того, команды проводят значительно больше игр, чем в других видах спорта, что увеличивает выборку, упрощая прогноз.

**Цель работы** – исследовать понятие и различные способы построения нейронных сетей, получить информацию о результатах баскетбольных матчей, прошедших в выбранной лиге, применить полученные знания для построения собственной нейронной сети для прогнозирования результатов предстоящих матчей. В этой работе для построения нейронной сети я буду использовать алгоритм нейроэволюции нарастающих топологий. Применение данного алгоритма для построения нейронных сетей с целью предсказания результатов баскетбольных матчей ранее не рассматривалось в научных статьях, что говорит о новизне темы. Этот алгоритм позволит не только определить подходящие веса нейронов в сети, но и определить наиболее эффективную архитектуру нейронной сети.

**Задачи работы:**

1. исследовать понятие нейронных сетей;
2. изучить разнообразные алгоритмы построения нейронных сетей, в том числе примененные ранее для похожих исследований;
3. изучить возможные сложности, часто возникающие при попытках прогноза спортивных событий;
4. реализовать собственную нейронную сеть на основе алгоритма нейроэволюции нарастающих топологий для прогнозирования результатов предстоящих матчей;
5. оценить качество предоставленных прогнозов;
6. сделать выводы, определить пути улучшения модели для повышения качества прогнозов;

**Объектом исследования** является нейронная сеть, прогнозирующая результаты баскетбольных матчей.

**Предмет исследования** – оценка качества прогнозов, факторов влияющих на него.

**Глава 1.Нейронные сети и их использование для проблем предсказания.**

**1.1Понятие нейронных сетей.**

Иску́сственные нейро́нные се́ти (ИНС) — математические модели или программные их имплементации, которые построены в соответствии с методами работы биологических нейронных сетей —связанных между собой нервных клеток живых организмов. Определение впервые появилось при описании процессов, протекающих в мозге человека, и попытках воспроизвести эти процессы в моделях. В дальнейшем, когда были разработаны обучающие алгоритмы, симулирующие работу мозга, такие модели начали использоваться в практических задачах: при управлении организациями, различных прогнозах, для распознавания образов и других.

Искуственная нейронная сеть представляет из себя набор простых процессоров (искусственных нейронов), которые соединены и взаимодействуют между собой. Эти процессоры, как правило, достаточно просты, если сравнивать их с процессорами, которые используются в персональных компьютерах. Любой процессор подобной искуственной сети взаимодействует только посредством сигналов, которые он получает от других, окружающих его процессоров, и сигналами, которые он периодически отправляет другим процессорам. Однако, будучи соединёнными в обширную сеть с настраиваемым взаимодействием, эти отдельно простые нейроны вместе имеют способность выполнять совсем нетривиальные задачи.

Нейронные сети построены не на принципах простых программируемых алгоритмов, они самообучаются. Такая способность обучаться и есть то главное отличие и преимущество ИНС в сравнении с обычными повсеместно применяемыми алгоритмами. Весь процесс обучения технически завязан на изменении весов связей между различными нейронами, которые объединены в сеть. Обучаясь, сеть может выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Именно поэтому, когда обучение проходит успешно, выходные нейроны способны вывести правильный результат, основываясь на новых данных, не предоставленных ей на этапе обучения, а также неполных и/или «зашумленных», частично искаженных данных.

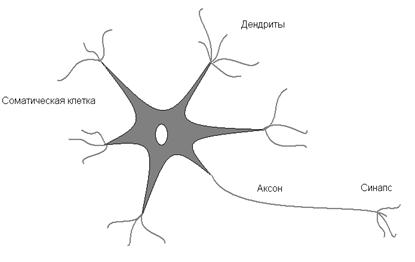


Рис.1 Схематическое изображение биологического нейрона (прообраза нейрона ИНС)

Основу каждой искусственной нейронной сети составляют относительно простые, в большинстве случаев - однотипные, элементы (ячейки), имитирующие работу нейронов мозга (далее под нейроном мы будем подразумевать искусственный нейрон, ячейку искусственной нейронной сети).

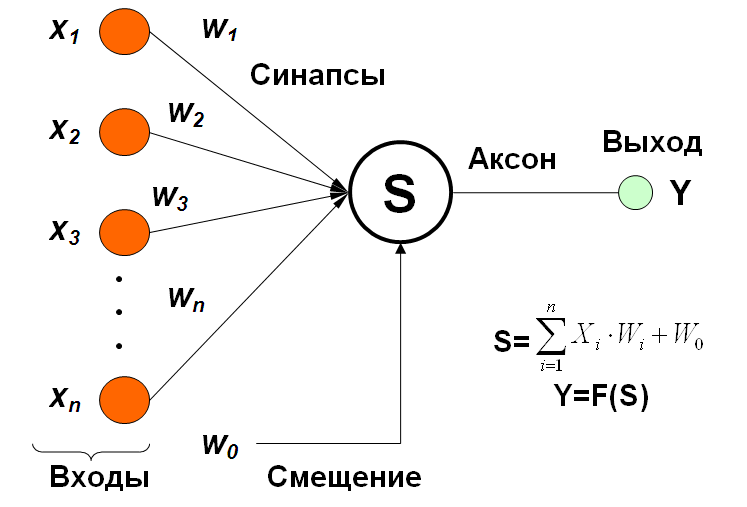


Рис. 2 - Искусственный нейрон

· Нейрон обладает группой синапсов - однонаправленных входных связей, соединенных с выходами других нейронов. Каждый синапс характеризуется величиной синоптической связи или ее весом wi.

· Каждый нейрон имеет текущее состояние, которое обычно определяется, как взвешенная сумма его входов:

· Нейрон имеет аксон - выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Выход нейрона есть функция его состояния:

y = f(s)

Функция f называется функцией активации.

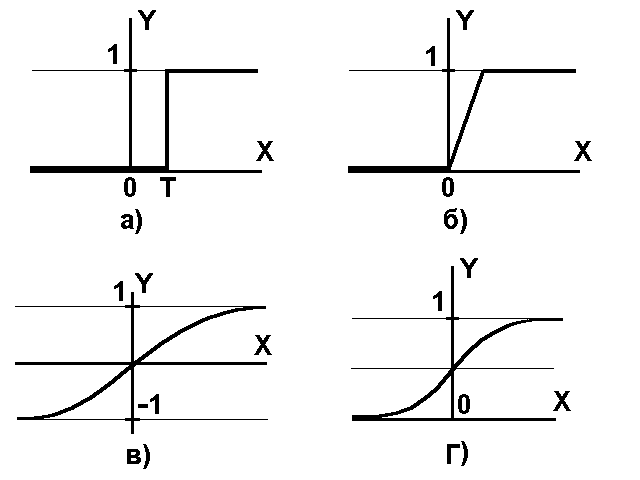


Рис. 3 - Функция активации

Функция активации может иметь разный вид :

· пороговый ( рис. 3.a),

· кусочно-линейный ( рис. 3.б),

· сигмоид( рис. 3.в, 3.г ).

Множество всех нейронов искусственной нейронной сети можно разделить на подмножества - т.н. слои. Взаимодействие нейронов происходит послойно.

Слой искусственной нейронной сети - это множество нейронов на которые в каждый такт времени параллельно поступают сигналы от других нейронов данной сети

Выбор архитектуры искусственной нейронной сети определяется задачей. Для некоторых классов задач уже существуют оптимальные конфигурации. Если же задача не может быть сведена ни к одному из известных классов, разработчику приходится решать задачу синтеза новой конфигурации. Проблема синтеза искусственной нейронной сети сильно зависит от задачи, дать общие подробные рекомендации затруднительно. В большинстве случаев оптимальный вариант искусственной нейронной сети получается опытным путем.

Искусственные нейронные сети могут быть программного и аппаратного исполнения. Реализация аппаратная обычно представляет собой параллельный вычислитель, состоящий из множества простых процессоров.

**1.2 Пример использования нейронных сетей для предсказания результатов баскетбольных матчей**

В научной статье **Predicting NBA Games Using Neural Networks** (Bernard Loeffelholz, Earl Bednar, and Kenneth W. Bauer, 2009) проверялось использование нейронных сетей как инструмента для предсказания успеха баскетбольных команд в Национальной Баскетбольной Организации (НБА). Статистика 620 матчей была собрана и использована для тренировки нескольких типов нейронных сетей, таких как нейронные сети прямого распространения (feed-forward), сеть радиально-базисных функций (radial basis), вероятностная неронная сеть (probabilistic), обобщённо-регрессионные нейронные сети (generalized regression neural networks).

Кроме того было исследовано, какой именно набор входных параметров для нейронных сетей обладает наиболее существенными признаками для прогнозирования. Для этого использовалось, как мнение экспертов – прогнозистов, так и степень объяснения признаками итогового результата. Результаты, полученные от нейронных сетей сравнили с предсказаниями нескольких баскетбольных аналитиков.

Лучшие нейронные сети были способны предсказать 74.33 процента результатов верно (в среднем), в то время как эксперты были правы в 68.67 процентах всех случаев. Более того, каждая нейронная сеть из четырех названных выше показала более высокое качество прогнозов, чем баскетбольные эксперты.

**2.1 Описание алгоритма нейроэволюции нарастающих топологий**

Описание данного алгоритма впервые было приведено в статье **Efficient Evolution of Neural Network Topologies (**Kenneth O. Stanley and Risto Miikkulainen).

Нейроэволюция – это искуственная эволюция нейронных сетей с использованием генетических алгоритмов.

Алгоритм нейроэволюции нарастающих топологий был разработан, чтобы соответствовать трем главным требованиям.

(1) Создать такое генетическое представление, которое позволит разрозненным топологиям скрещиваться таким образом, чтобы это имело наибольший смысл. Для этого в алгоритме применяется историческая маркировка для выявления генов с одинаковым происхождением.

(2) Найти способ защиты топологических инноваций, которые нуждаются в нескольких поколениях для оптимизации, чтобы не исчезнуть из популяции преждевременно. Для этого в алгоритме каждая инновация выделяется в отдельный вид, чтобы ей не приходилось конкурировать с более оптимизированными топологиями.

(3) Минимизировать топологии во время эволюции без использования специальных функций, измеряющих сложность. Представленное решение состоит в том, что изначально любая структура минимальна и растет только в случае необходимости.

Каждый геном (рис.4) включает в себя список *соединительных генов*, которые ссылаются на два *узловых гена*, которые они соединяют. Каждый соединительный ген включает в себя входной узел, выходной узел, вес соединения, активировано ли данное соединение (бит активации) и номер инновации, который позволяет найти соответствующие гены во время скрещивания (объясняется в дальнейшем).

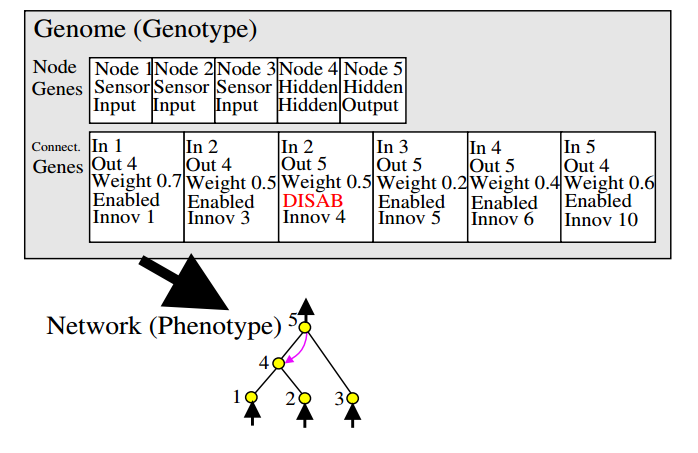


Рис.4 Пример преобразования генотипа (способ записи сети) в фенотип (визуальное представление сети). Третий ген исключен, поэтому соответствующая связь (между узлами 2 и 5) в фенотипе отсутствует.

Мутации согласно алгоритму могут изменять как веса соединений, так и структуру сети. Веса соединений мутируют случайным образом путем генерации действительных чисел. Структурные мутации, расширяющие геном бывают двух видов (рис.5). При мутации добавления связи один новый ген связи добавляется в геном, соединяя два, ранее не соединенных, узла. При мутации добавления узла существующее соединения разбивается, и новый узел помещается вместо старого соединения. Ген старого соединения исключается, два новых соединительных гена добавляются в геном. Такой метод добавления узлов был выбран, чтобы немедленно интегрировать узлы в сеть, но при этом сохранить возможность проследить происхождение узлов.

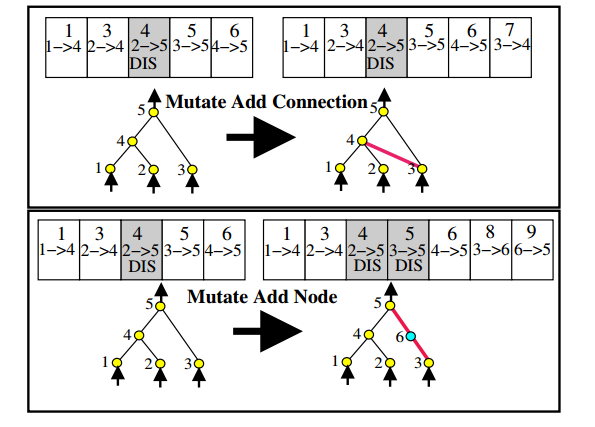


Рис.5. Два вида структурных мутаций в алгоритме нейроэволюции нарастающих топологий. Оба типа, добавления соединения и добавления узла, проиллюстрированы генотипом и фенотипом. Верхнее число каждого гена – это его инновационное число. Эти цифры идентифицируют первоначального исторического предка каждого гена, что позволяет находить совпадающие гены во время скрещивания. Новые гены обозначаются новыми, все более высокими числами.

Посредством мутаций создаются геномы разных размеров, иногда с абсолютно разными соединениями в одинаковых позициях.

Чтобы производить скрещивания, система должна быть способна определить совпадающие гены в популяции генотипов. Ключевое наблюдение состоит в том, что два гена, которые имеют одинаковое историческое происхождение, представляют одну и ту же структуру (возможно, с разными весами), поскольку оба они были получены из одного и того же наследственного гена однажды в прошлом. Таким образом, все, что системе нужно делать, чтобы знать, какие гены соединяются с какими, это проследить историческое происхождение каждого гена в системе.

Отслеживание исторического происхождения требует совсем немного вычислений. Каждый раз, когда появляется новый ген (путем структурных мутаций), глобальный номер инновации инкрементируется и присваивается этому гену. Номер инновации, таким образом, представляет хронологию каждого гена системы. Например, предположим, что две мутации на рис.5 появились одна за другой в системе. Первому соединительному гену, произошедшему во время первой мутации добавления связи, назначается номер инновации 7, и двум новым соединительным генам, добавленным во время следующей мутации добавления узла, назначаются номера 8 и 9В будущем, как бы эти геномы не скрещивались, потомки будут наследовать те же номера инноваций на каждом гене; номера инноваций никогда не меняются. Таким образом, историческое происхождение любого гена системы известно на протяжении всей эволюции.

Историческая маркировка дает алгоритму нейроэволюции нарастающих топологий новую мощную возможность. Система теперь точно знает, какие гены с какими совпадают (рис.6). Гены, которые не совпадают либо дизъюнктивные, либо избыточные, в зависимости от того, появляются они в пределах или вне диапазона других родительских номеров инноваций. Во время скрещивания, гены, которые присутствуют в обоих родителях с одинаковыми номерами инновации, наследуются. Гены, которые не совпадают, наследуются от более подходящего родителя, или, если они одинаково подходят, то от любого родителя случайным образом. Так историческая маркировка дает алгоритму нейроэволюции нарастающих топологий возможность производить скрещивание без сложного топологического анализа.

Представленный метод скрещивания отличается своей простотой. Любые две структуры могут быть объединены вместе без необходимости в каком-либо топологическом анализе.

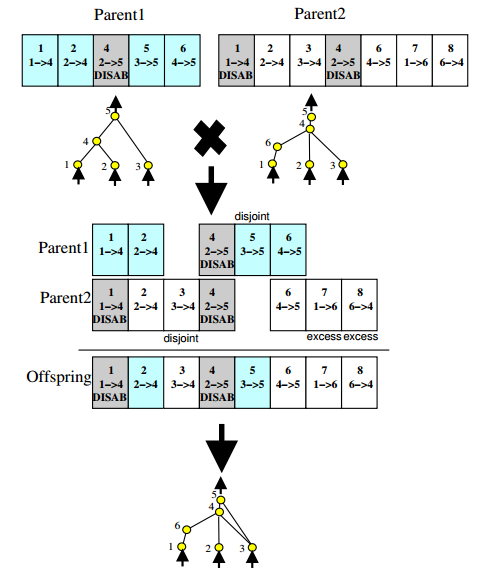


Рис.6. Объединение геномов разных сетевых топологий с использованием номеров инновации. Хоть оба родителя выглядят непохожими, их номера инноваций (обозначенные вверху каждого гена) указывают, какие гены следует объединить.

Добавление в нейронную сеть новых структур, как правило, изначально снижает приспособленность (fitness) сети. Однако алгоритм нейроэволюции нарастающих топологий определяет популяции таким образом, что отдельные генотипы конкурируют в пределах собственных ниш, а не всей популяции в целом. Таким образом, инновации в топологии защищены и имеют время на оптимизацию, прежде чем им придется конкурировать с остальными нишами из популяции.

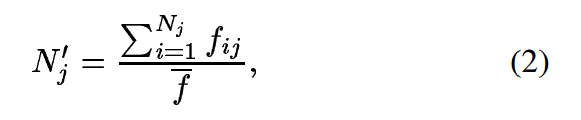
Исторические маркировки позволяют системе разделить популяцию на виды, базируясь на топологической схожести. Количество избыточных и дизъюнктивных генов между парой геномов это естественный показатель уровня их совместимости. Чем более дизъюнктивны два генома, тем менее схожи их эволюционные истории, тем, соответственно, они менее совместимы. Поэтому, мы можем выразить отличие разных структур в алгоритме нейроэволюции нарастающих топологий как простую линейную комбинацию количества их избыточных (E) и дизъюнктивных (D) генов и средней разницы в весах совпадающих генов (Wa):

a  *(1)*

Коэффициенты с1, с2, с3 регулируют важность трех факторов, а N обозначает количество генов в более крупном геноме и нормализует размер генома.

Мера отличия нам разделять геномы по видам, используя пороговое значение t . Геномы тестируются по одному за раз. Если мера отличия генома со случайно выбранным представителем вида меньше t, то геном помещается в этот же вид. Каждый геном помещается в первый вид, где условие попадания выполняется, поэтому каждый геном не может принадлежать более, чем одному виду.

В качестве механизма репродукции в алгоритме используется *явное разделение приспособленности*, то есть организмы одного вида должны быть схожи по уровню приспособленности в своей нише, что затруднит захват всей популяции одним отдельным видом. Поэтому исходные уровни приспособленности корректируются делением на число организмов в виде. Виды со временем расширяются или сокращаются в зависимости от того, выше их скорректированная приспособленность или ниже, чем средняя приспособленность всей популяции:



, где Nj и N’j это старое и новое количество организмов в виде j, fij это скорректированная приспособленность генома i из вида j, а это средняя скорректированная приспособленность всей популяции. Заранее выбранный процент представителей каждого вида скрещиваются случайным образом, чтобы сгенерировать N’j число потомков, которые заменяют собой весь вид.

В редких случаях, когда приспособленность всей популяции не растет на протяжении более чем 20 поколений, только два лучших вида производят репродукцию, тем самым поиск оптимальной структуры фокусируется в самых многообещающих местах.

D. Minimizing Dimensionality TWEANN algorithms typically start with an initial population of random topologies [2, 7, 21, 22]. Such topological diversity must be introduced from the start because new structure frequently does not survive in these methods, which do not protect innovation. However, it is not clear that such diversity is necessary or useful. A population of random topologies has a great deal of structure that has not withstood a single fitness evaluation. Therefore, there is no way to know if any of such structure is necessary. It is costly though because the more connections a network contains, the higher the number of dimensions that need to be searched to optimize the network. Therefore, with random topologies the algorithm may waste a lot of effort by optimizing unnecessarily complex structures.

In contrast, NEAT begins with a uniform population of networks with no hidden nodes. Because NEAT protects innovation using speciation, it can start this way, minimally, and grow new structure only as necessary. New structure is introduced incrementally as structural mutations occur, and only those structures survive that are found to be useful through fitness evaluations. This way, NEAT searches through a minimal number of weight dimensions, significantly reducing the number of generations necessary to find a solution.

In the current implementation of NEAT-Python, a population of individual genomes is maintained. Each genome contains two sets of genes that describe how to build an artificial neural network:

1. Node genes, each of which specifies a single neuron.
2. Connection genes, each of which specifies a single connection between neurons.

To evolve a solution to a problem, the user must provide a fitness function which computes a single real number indicating the quality of an individual genome: better ability to solve the problem means a higher score. The algorithm progresses through a user-specified number of generations, with each generation being produced by reproduction (either sexual or asexual) and mutation of the most fit individuals of the previous generation.

The reproduction and mutation operations may add nodes and/or connections to genomes, so as the algorithm proceeds genomes (and the neural networks they produce) may become more and more complex. When the preset number of generations is reached, or when at least one individual exceeds the user-specified fitness threshold, the algorithm terminates.