Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий

Кафедра «Информатика»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Рубан А.И.
подпись инициалы, фамилия
«16» июня 2016 г.

БАКАЛАВРСКАЯ РАБОТА

27.03.03 «Системный анализ и управление»

Методы и алгоритмы выявления зависимостей переменных в матрице процесса дискретно-непрерывного типа

Руководитель _____ 13.06.16 доцент, К.Т.Н. Даничев А.А.
подпись, дата должность, ученая степень инициалы, фамилия

Выпускник _____ 15.06.16 Потехин А.С.
подпись, дата инициалы, фамилия

Красноярск 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В нашу эпоху информации процесс моделирования играет важную роль в жизни человека. Этап построения моделей присутствует во всех сферах жизнедеятельности, без него не обходится ни одно исследование [1]. Поэтому на сегодняшний день существует целый ряд запрограммированных пакетов моделирования, основанных на твердом теоретическом фундаменте.

Но большинство алгоритмов моделирования основаны на построении параметрических моделей. Такие модели требуют достаточной априорной информации об исследуемом объекте, которая в реальности не всегда доступна. В данном исследовании ставятся задачи по выявлению дополнительной информации об объекте из не обработанных данных.

В частности, рассматривается задача выявления зависимостей переменных в матрице наблюдений процесса дискретно-непрерывного типа. В рамках данного исследования будет проведено исследование по сравнению разных методов выявления зависимостей, таких как факторный анализ, регрессионный анализ, дисперсионный анализ.

Цель: выявление зависимостей в данных, для более качественного моделирования технологических процессов.

Задачи:

- Постановка задачи выявления зависимости между переменными;
- Обзор классических методов выявления зависимостей;
- Разработка метода применения нейронной сети для задачи выявления зависимости между переменными;
- Апробация алгоритма и анализ полученных результатов.

Объект исследования: матрица наблюдений технологических процессов с ТЭЦ-1.

Предмет исследования: методы выявления зависимостей в матрице наблюдений.

Для решения поставленных задач предлагается использовать следующие методы:

- Построение и анализ структуры нейронной сети используя алгоритмы нейроэволюции;
- Оценка степени влияния входных переменных на выходные переменные путем анализа полученного графа;

Данное исследование позволит применить алгоритм нейроэволюции нейронных сете для выявления зависимостей, выявить сильные и слабые стороны данного алгоритма, а также применить его к реальным данным с целью выявления скрытых зависимостей в технологическом процессе на ТЭЦ-1.

Перспективным развитием данной работы будет сравнения алгоритма на основе нейронных сетей с методом анализа параметра размытости в методе непараметрической регрессии Надарая-Ватсона[2].

1 Обзор методов и алгоритмов выявления зависимостей

1.1 Общая постановка задачи выявления зависимостей

При исследовании различных процессов или объектов в технических, информационных и других сферах научной и промышленной деятельности одной из основных проблем является построение разнообразных математических моделей. Этот этап является очень важным в изучении того или иного процесса (или свойств объекта), т.к. в моделях отражены только те свойства, которые представляют интерес для исследования. Построение моделей помогает абстрагироваться от всего «ненужного» и сосредоточиться только на первостепенных признаках исследуемого объекта.

Для одного и того же объекта может быть построен целый ряд моделей, каждая из которых отражает свойства объекта: технические, экономические, социальные и др. Исследование, которое затрагивает все аспекты изучаемого объекта, считается полным. Такое исследование позволяет получить наиболее качественную и полную информацию об особенностях функционирования объекта, а также, приводит к цели с наименьшими затратами.

Математический язык моделей существенно облегчает изучение объекта, т.к. благодаря нему возможно проведение подробного анализа свойств и функционирования объекта. После проведения этапа моделирования задача становится формализованной. Строгость математического языка, основанная на обширной базе правил оформления и условных знаков, позволяет формализовать задачу.

Математическое представление объекта позволяет дать численную оценку качеству функционирования объекта. В свою очередь это дает возможность сравнить между собой различные объекты или состояния объекта по интересующим исследователя критериям.

При большом количестве параметров, для построения более качественной модели объекта необходимо выявить среди них только те, которые оказывают существенное влияние на исследуемый процесс. Для этого

необходимо провести исследование зависимостей между параметрами. Необходимо выявить для каждого выходного параметра системы такие входные параметры, которые оказывают на него наибольшее влияние.

Пусть объект имеет вид (рисунок 1.1.1).

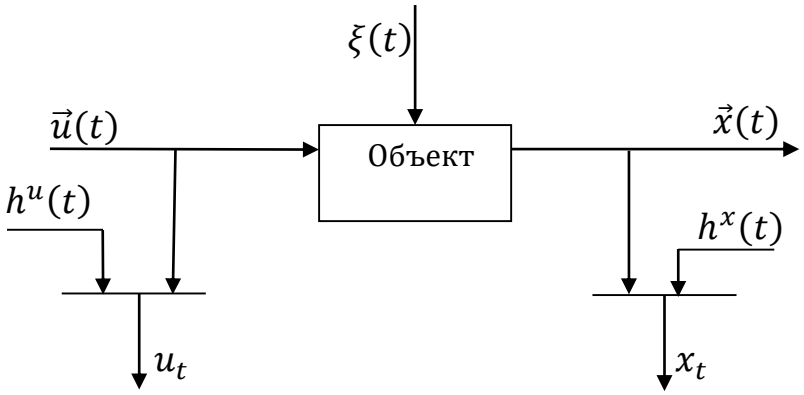

Рис. 1.1.1 – Общая схема моделирования объектов

Где $\vec{x}(t)$ – выходная векторная переменная объекта, $\vec{u}(t)$ – входная векторная переменная объекта. $\xi(t)$ – неизмеримая входная переменная объекта (случайное воздействие) $M(\xi) = 0,\ D(\xi) < \infty. u_t, x_t -$ помехи, присутствующие в каналах связи.

В действительности объекты функционируют в реальном времени t, но измерения переменных объекта производится через некоторый интервал времени Δt (поэтому исследуемые объекты и называются дискретно-непрерывные). Для различных переменных или их компонент этот интервал может быть своим, но в данном исследовании рассмотрен случай, когда интервал для всех переменных одинаков. Чем чаще производятся измерения, т.е. чем меньше интервал измерений Δt, тем выше точность построенных моделей. Но при бесконечно малом Δt потребуется бесконечное число экспериментов, а это физически невыполнимо. Поэтому перед исследователем всегда стоит задача поиска компромисса между критериями точности и затрат. Чаще всего для частоты измерений существует некоторый регламент, согласно которому величина Δt фиксирована. Иногда бывает, что эксперимент нельзя проводить чаще из-за каких-либо физических особенностей объекта. Так, например, положение звезд на небе возможно фиксировать только раз в сутки.

В общем случае объект можно представить в виде набора уравнения:

$$\vec{x}(t) = f\left(\vec{u}(t), \xi(t)\right)$$

Выявление зависимостей производится по наблюдениям – выборке статистически независимых измерений $x_i, u_i,\ i = \overline{1,s},$ где s – объем выборки. $f -$ набор уравнений, где каждому элементу вектора $\vec{x}(t)$ ставится в соответствие уравнение зависимости от определённого набора входных переменных.

1.2 Методы выявления зависимостей

Для выявления зависимостей или выявления информативности признаков (тут как-то в скобки, ну чтобы как-то указать что это не поиск зависимостей всё-таки) применяются следующие методы:

- Корреляционный анализ;
- Регрессионные анализ;
- Метод последовательного сокращения и добавления параметров модели.

Рассмотрим приведенные выше методы более подробно.

1.3 Корреляционный анализ

Метод обработки статистических данных, заключающийся в изучении коэффициентов (корреляции). Его применение возможно в случае наличия достаточного количества (для конкретного вида коэффициента корреляции) наблюдений из более чем одной переменной. При этом сравниваются коэффициенты корреляции между одной парой или множеством пар признаков, для установления между ними статистических взаимосвязей.

Значительная корреляция между двумя случайными величинами всегда является свидетельством существования некоторой статистической связи в данной выборке, но эта связь не обязательно должна наблюдаться для другой выборки и иметь причинно-следственный характер. Часто заманчивая простота корреляционного исследования подталкивает исследователя делать ложные интуитивные выводы о наличии причинно-следственной связи между парами признаков, в то время как коэффициенты корреляции устанавливают лишь статистические взаимосвязи.

Коэффициент корреляции используется наиболее часто, хотя его следует применять только при соблюдении следующих условий:

- Обе переменные являются количественными и непрерывными
- Как минимум один из признаков (а лучше оба) имеет нормальное распределение (поэтому расчет этого коэффициента является параметрическим методом оценки взаимосвязи признаков)
- Зависимость между переменными носит линейный характер
- Гомоскедастичность (вариабельность одной переменной не зависит от значений другой переменной)
- Независимость участников исследования друг от друга (признаки X и Y у одного участника исследования независимы от признаков X и Y у другого)
- Парность наблюдений (признак X и признак Y изучаются у одних и тех же участников исследования)
- Достаточно большой объем выборки
- Для адекватной проекции расчетов на генеральную совокупность выборка должна быть репрезентативной.

Пусть $(x_1, y_1), \dots, (x_S, y_S)$– выборка объема n из наблюдений случайной величины (ξ, η) , имеющей двумерное нормальное распределение. Изображая элементы выборки точками в декартовой системе координат, получим диаграмму рассеивания или корреляционное поле. Иногда по виду

корреляционного поля можно сделать предположение о наличии и характере связи между случайными величинами и ξ и η.

Выборочным коэффициентом корреляции называется число

$$r_{\text{выб}} = \frac{\frac{1}{S}\sum_{i=1}^{S} x_i y_i - \hat{x}\hat{y}}{\widetilde{S_x}\widetilde{S_y}}$$

На рисунке 1.3.1 приведены возможные формы корреляционного поля в зависимости от значения выборочного коэффициента корреляции.

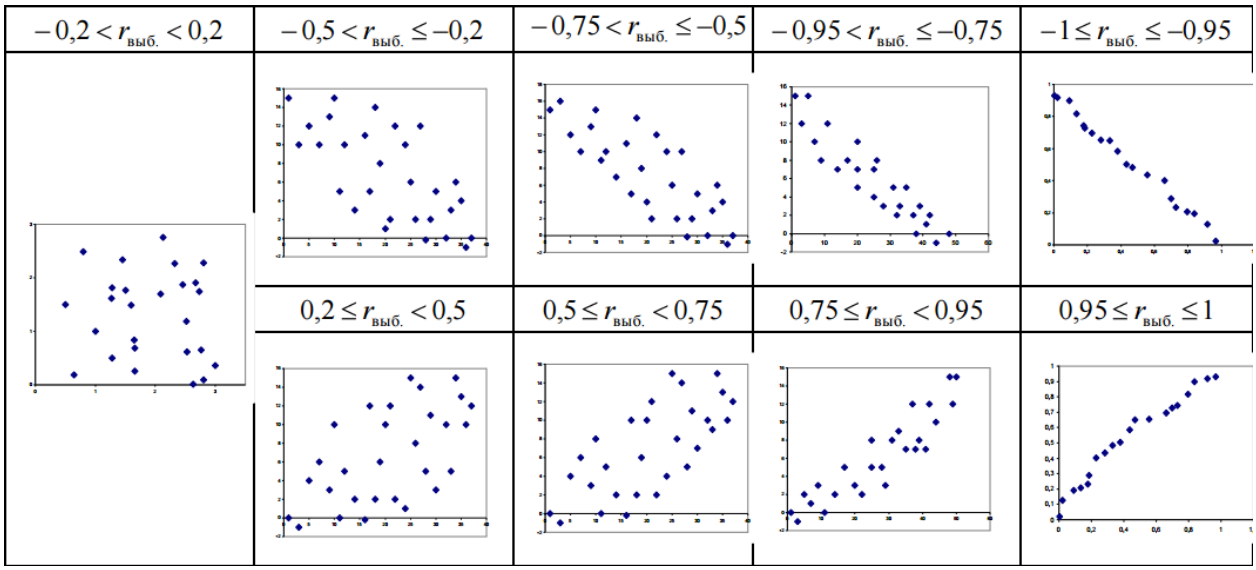

Рисунок 1.3.1 – Возможные формы корреляции поля

На практике большой интерес представляет задача проверки гипотезы о значимости корреляционной связи между случайными величинами, т. е. значимости отклонения коэффициента корреляции от нуля. Пусть $r_{\text{выб}}$ – выборочный коэффициент корреляции. При заданном уровне значимости проверяется гипотеза $H_0: r = 0$ о равенстве нулю теоретического коэффициента корреляции. Если нулевая гипотеза будет отвергнута, то говорят о значимости коэффициента корреляции, а значит о том, что случайные величины ξ и η коррелированы. Если нулевая гипотеза принимается, то коэффициент

корреляции незначим, и случайные величины и не коррелированы. Статистика критерия имеет вид:

$$t_{\text{набл}} = r_{\text{наб}} \sqrt{\frac{n-2}{1-r_{\text{выб}}^2}}$$

Находится значение точки $t_{\frac{\alpha}{2}}$ распределения Стьюдента с $n-2$ степенями свободы. Схема принятия решения выглядит следующим образом:

- если $|t_{\text{набл}}| = \left| r_{\text{выб}} = \sqrt{\frac{n-2}{1-r_{\text{выб}}^2}} \right| < t_{\frac{\alpha}{2}}$, то нет оснований отвергать нулевую гипотезу, коэффициент корреляции не значим, а ξ и η не коррелированы;
- если $|t_{\text{набл}}| = \left| r_{\text{выб}} = \sqrt{\frac{n-2}{1-r_{\text{выб}}^2}} \right| \geq t_{\frac{\alpha}{2}}$, то гипотеза отвергается, и коэффициент корреляции значимо отличается от нуля, а ξ и η коррелированы.

Корреляционный анализ дает возможность аналитику сделать предположение о возможно наличии связей между переменными, но так как с его помощью невозможно это доказать, то аналитик сможет получить некоторый набор переменных, между которыми возможна некоторая связь, что может послужить поводом к дальнейшему исследованию этих переменных. Также, с помощью корреляционного анализа невозможно выявить нелинейные взаимосвязи.

1.4 Регрессионный анализ

Регрессионный анализ – это статистический процесс выявления зависимостей между переменными. Он позволяет понять, как влияет на значение зависимой переменной независимая переменная.

Наблюдаются значения $(x_1, y_1), \dots, (x_S, y_S)$ двумерной случайной величины (ξ, η). Исследуется зависимость случайной величины ξ от случайной величины η. В общем случае регрессионная модель имеет вид:

$$y = f(x, \beta_0, \beta_1, \dots, \beta_k)$$

где параметры $\beta_0, \beta_1, \dots, \beta_k$ называются коэффициентами регрессии. Одна из задач регрессионного анализа – оценка коэффициентов регрессии. Для оценки коэффициентов регрессии, как правило, используется метод наименьших квадратов: в качестве оценок принимаются такие значения параметров, которые минимизируют сумму квадратов отклонений наблюдаемых значений y_i от $\widehat{y_i} = f(x, \beta_0, \beta_1, \dots, \beta_k), i = \overline{1, S}$. т. е. метод наименьших квадратов основан на минимизации суммы квадратов:

$$\sum_{i=1}^{S} \varepsilon_i^2 = \sum_{i=1}^{S} (y_i - \widehat{y_i})^2 \rightarrow min$$

Если предположить, что связь между переменными линейна, то соответствующая регрессионная модель имеет вид:

$$y = \beta_0 + \beta_1 x,$$

где β_0и β_1 – коэффициенты линейной регрессии.

1.5 Метод последовательного сокращения и метод последовательного добавления параметров

Суть данных методов состоит в том, чтобы строить модели объекта варьируя количество признаков.

1.5.1 Метод последовательного сокращения входных переменных модели (алгоритм Del)

Пусть задача состоит в том чтобы из S входных переменных выбрать K входных переменных, влияние которых на выходную переменную наибольшее.

Оценим ошибку модели при использовании всех S переменных α_0. Затем исключим из уравнения модели первую входную переменную и найдем ошибку модели α_{11}, которую дают оставшиеся $S-1$ входных переменных. Теперь из первоначального уравнения модели исключим вторую входную переменную и найдем ошибку модели α_{12}. Эту операцию поочередного исключения одной входной переменной проведем S раз. Среди полученных величин $\alpha_{11}, \dots, \alpha_{1j}, \dots, \alpha_{1S}$ найдем самую малую. Она укажет нам на входную переменную, исключение которого из уравнения модели было наименее ощутимым. Исключим это входную переменную из уравнения модели и испытаем оставшиеся $S-1$ входных переменных. Их поочередное исключение из уравнения модели позволит найти вторую входную переменную влияние которой на выход модели незначительно и снизить размерность пространства до $S-2$ входных переменных. Эти процедуры повторяются $S-K$ раз, то есть до тех пор в системе не останется заданное число входных переменных K.

Данный метод можно модифицировать, исключая входные переменные до тех пор, пока ошибка модели на следующем шаге метода не начнет возрастать.

1.5.2 Метод последовательного добавления входных переменных модели (алгоритм Add).

Данный алгоритм отличается от предыдущего лишь тем, что порядок проверки входных параметров модели начинается не с S мерного пространства а с одномерных пространств. Вначале все S входных переменных проверяются на

значимость. Для этого строится модели где в качестве входного параметра выступает каждый из S входных параметров в отдельности и в конечное уравнение модели включается входная переменная соответствующая модель которой показала наименьшую ошибку. Затем к нему по очереди добавляются все $S-1$ оставшихся входных переменных по одой. Выбирается модель от пары входных параметров, которой показала наименьшую ошибку. Так продолжается до получения модели с K входными переменными.

Результаты, получаемые алгоритмом Add, лучше, чем у Del. Объясняется этот факт влиянием малой представительности обучающей выборки: при одном и том же объеме выборки чем выше размерность признакового пространства, тем меньше обоснованность получаемых статистических выводов.

Оба описанных алгоритма дают оптимальное решение на каждом шаге, но это не обеспечивает глобального оптимума.

1.5.3 Алгоритм AddDel

Для ослабления влияния ошибок на первых шагах алгоритма применяется релаксационный метод. В алгоритме Add набирается некоторое количество S_1 информативных признаков и затем - часть из них $S_2 < S_1$ исключается методом Del. После этого алгоритмом Add размерность информативных признаков наращивается на величину S_1 и становится равной $2S_1 - S_2$. В этот момент снова включается алгоритм Del, который исключает из системы «наименее ценных» признаков. Такое чередование алгоритмов Add и Del, продолжается до достижения заданного количества признаков S.

2 Построение и обучение нейронной сети с помощью алгоритма NEAT

Для выявления зависимостей между переменными в матрице наблюдений мы будем применят многослойный персептрон, построенный и обученный по алгоритму NEAT.

2.2 Нейронные сети

Нейронная сеть – это система, состоящая из многих простых элементов, которые работают параллельно, функция которых зависит от структуры сети, силой взаимосвязанных связей, а вычисления производятся в самих элементах или нейронах.

Работу искусственного нейрона можно описать следующим образом (рисунок 1).

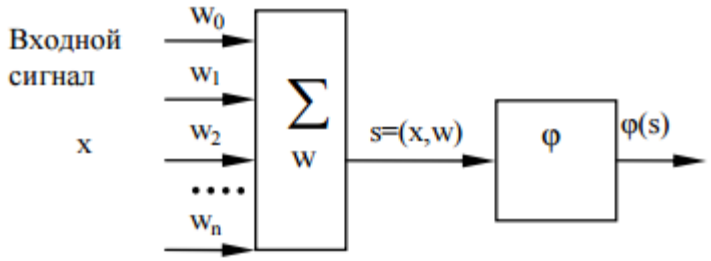

Рисунок 2.1.1 – Искусственный нейрон

Нейрон получает входные сигналы (исходные данные либо выходные сигналы других нейронов нейронной сети) через несколько входных каналов. Каждый входной сигнал проходит через соединение, имеющее определенный вес. С каждым нейроном связано определенное пороговое значение, которое вычитается из взвешенной суммы входов и в результате получается величина активации нейрона. Сигнал активации преобразуется с помощью функции активации (или передаточной функции). В результате получается выходной сигнал данного нейрона сети:

$$S = (x, w) = w_0 + \sum_{i=1}^{S} x_i w_i$$

где x_i - входные сигналы, совокупность которых формируют вектор;

w_i - весовые коэффициенты, совокупность которых образуют вектор весов;

S - взвешенная сумма входных сигналов, значение передается на нелинейный элемент.

Наиболее распространенными функциями активации являются следующие: пороговая, sigmoid (или модифицированная пороговая функция), логистическая, гиперболический тангенс, линейная, линейная ограниченная, радиально- базисная и др. Как правило, функции активации всех нейронов в сети фиксированы, а веса являются параметрами сети и могут изменяться. Если сеть предполагается для, чего-то использовать, то у нее должны быть входы (принимающие значения интересующих нас переменных из внешнего мира) и выходы (прогнозы или управляющие сигналы). Кроме этого в сети может быть еще много скрытых нейронов, выполняющих внутренние функции. Входные, скрытые и выходные нейроны должны быть связаны между собой.

Наиболее распространенными являются многослойные сети, в которых нейроны объединены в слои. Слой – это совокупность нейронов, на которые в каждый такт времени параллельно поступает информация от других нейронов сети. После того, как определено число слоев и число элементов в каждом из них, нужно найти значения для весов и порогов сети, которые минимизировали бы ошибку прогноза, выдаваемого сетью. Именно для этого служат алгоритмы обучения. С использованием собранных исторических данных веса и пороговые значения автоматически корректируются с целью минимизации этой ошибки. По сути, этот процесс представляет собой подгонку модели, которая реализуется сетью, к имеющимся обучающим данным. Ошибка для конкретной конфигурации сети определяется путем прогона через сеть всех имеющихся наблюдений и сравнения реально выдаваемых выходных значений с желаемыми (целевыми) значениями. Все такие разности суммируются в функцию ошибок, значение которой и есть ошибка сети.

В качестве функции ошибок чаще всего берется сумма квадратов ошибок:

$$E(w) = \sum_{i=1}^{S} (d_i - y_i)^2$$

где d – желаемый выход сети, y – реальный выход сети. Самый известный вариант алгоритма обучения нейронной сети – так называемый алгоритм обратного распространения [2]. Существуют современные алгоритмы второго порядка, такие, как метод сопряженных градиентов и метод Левенберга-Маркара [3], которые на многих задачах работают существенно быстрее (иногда на порядок). Алгоритм обратного распространения в некоторых случаях имеет определенные преимущества.

2.3 Применение нейронных сетей

Нейронные сети широко применяются для различных задач в области обработки и анализа данных – распознавание и классификация образов, прогнозирование, управление и т.д [4].

Сети с прямой связью являются универсальным средством аппроксимации функций, что позволяет их использовать в решении задач классификации. Как правило, нейронные сети оказываются наиболее эффективным способом классификации, потому что генерируют фактически большое число регрессионных моделей (которые используются в решении задач классификации статистическими методами).

К сожалению, в применении нейронных сетей в практических задачах возникает ряд проблем. Во-первых, заранее не известно, какой сложности (размера) может потребоваться сеть для достаточно точной реализации отображения. Эта сложность может оказаться чрезмерно высокой, что потребует сложной архитектуры сетей. Простейшие однослойные нейронные сети способны решать только линейно разделимые задачи. Это ограничение преодолимо при использовании многослойных нейронных сетей. В общем виде можно сказать, что в сети с одним скрытым слоем вектор, соответствующий

входному образцу, преобразуется скрытым слоем в некоторое новое пространство, которое может иметь другую размерность, а затем гиперплоскости, соответствующие нейронам выходного слоя, разделяют его на классы. Таким образом сеть распознает не только характеристики исходных данных, но и "характеристики характеристик", сформированные скрытым слоем.

Задачи аппроксимации экспериментальных данных можно решать с помощью искусственных нейронных сетей следующих типов: многослойного персептрона, сетей с радиально-базисными функциями, вероятностных сетей, обобщенно-регрессионных сетей.

Задача аппроксимации функции для нейронной сети формируется как задача контролируемого обучения (обучение с учителем). Суть задачи состоит в следующем. Имеются значения функции в отдельных точках (узлах), система базисных функций и векторов регулируемых весовых коэффициентов. Необходимо обучить сеть, т. е. выбрать весовые коэффициенты при базисных функциях так, чтобы их комбинация давала аналогичную зависимость, которая наилучшим образом аппроксимирует множество значений функции отклика.

2.4 Алгоритм NEAT

Алгоритм NEAT - эволюционный алгоритм. Он позволяет использовать генетические алгоритмы для определения лучшей и минимально необходимой топологии нейронной сети [5]. В этом алгоритме нейронная сеть представляет собой ориентированный граф. Нейроны могут быть трех видов - сенсоры (входные), скрытые, и выходные. Связи содержат в себе номера нейронов, с которыми они связаны, веса и порядковый номер. Каждая связь может быть в двух состояниях: активном и неактивном.

Векторы нейронов и связей являют собой "Ген" (отдельную нейронную сеть), которым оперируют генетический алгоритм. В примере на рисунке прямого пути из 2 в 4 нет, поэтому эта связь обозначена неактивной. Такие связи не принимают участия в моделировании выхода модели, но могут снова стать активными в ходе эволюции данного гена или его мутации [3].

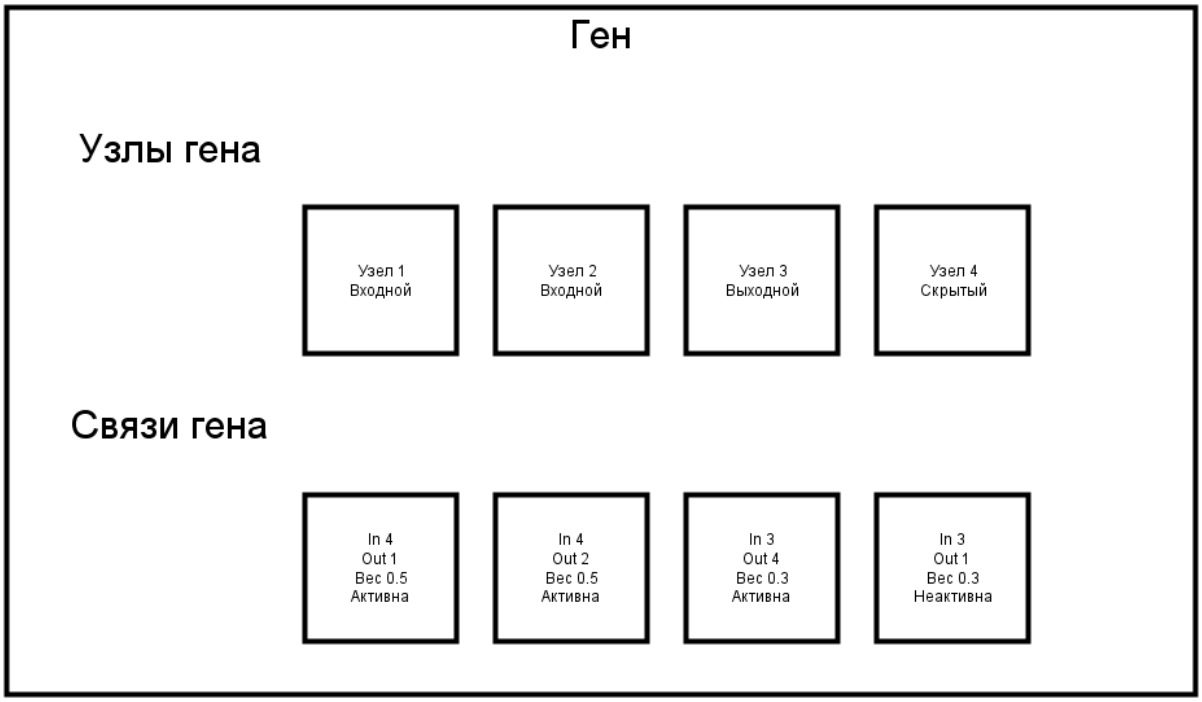

Рисунок 2.3.1 – Ген, текстовое представление.

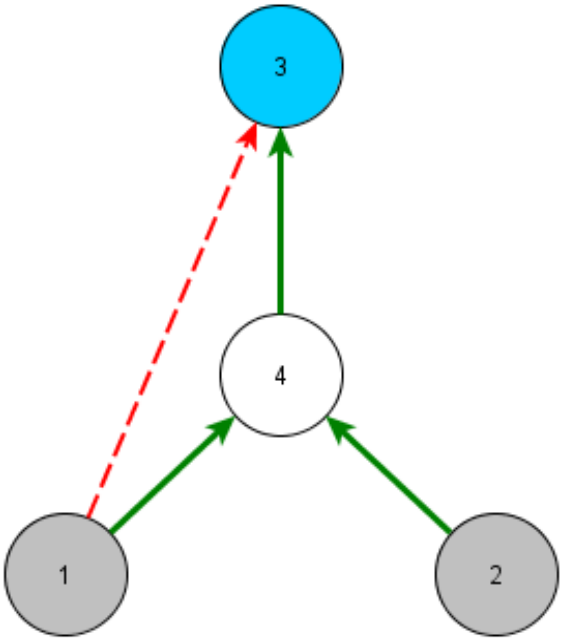

Рисунок 2.3.2 – Ген, графическое представление.

Каждая связь обладает своим уникальным маркером (innovation number). Новый маркер может быть создан только при мутации. Это по сути ID связи, но кроме всего прочего он позволяет отследить как "возраст" самой сети, так и

происходивших с ней мутаций. Кроме того, это позволяет значительно упростить скрещивание.

Мутации бывают двух видов. При первой мутации может добавится связь к уже существующим нейронам:

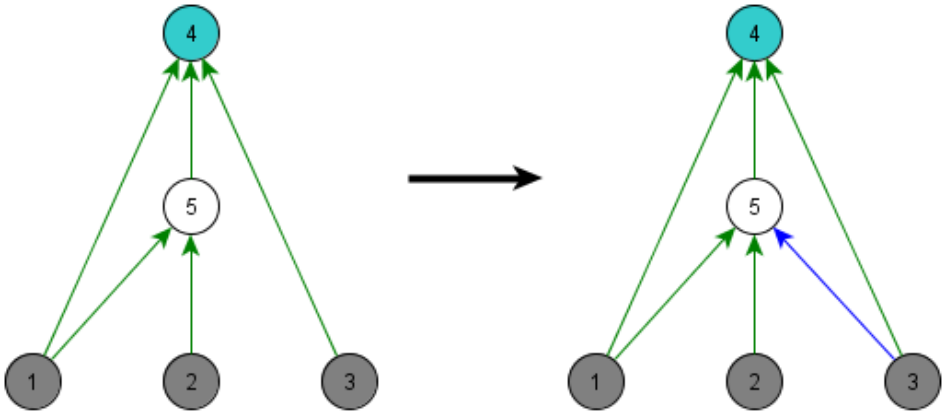

Рисунок 2.3.3 – Мутация первого типа.

При втором типе мутации создается новый нейрон, на месте уже существующей связи между двумя нейронами. При этом старая связь становится неактивной, и создаются две новые:

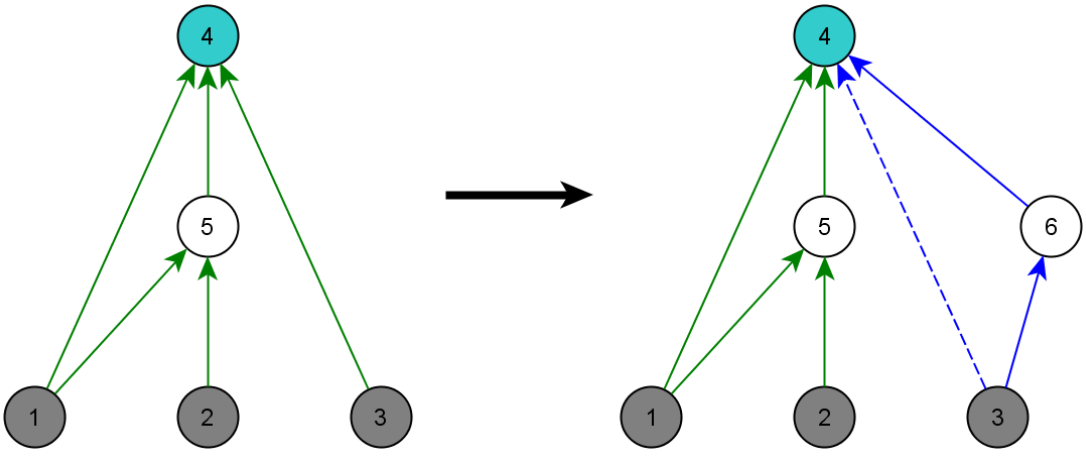

Рисунок 2.3.4 – Мутация второго типа.

В обоих случаях новым связям будет присвоен новый исторический маркер (уникальное число).

Эти-же исторические маркеры используются при скрещивании чтобы понять, как смешать два гена нужно в ген потомок внести все уникальные исторические маркеры обоих родителей. Если связь неактивна у одного из родителей, то у потомка она также будет не активна. Если связь неактивна у обоих родителей, то у нее есть шанс мутировать и стать активной у потомка.

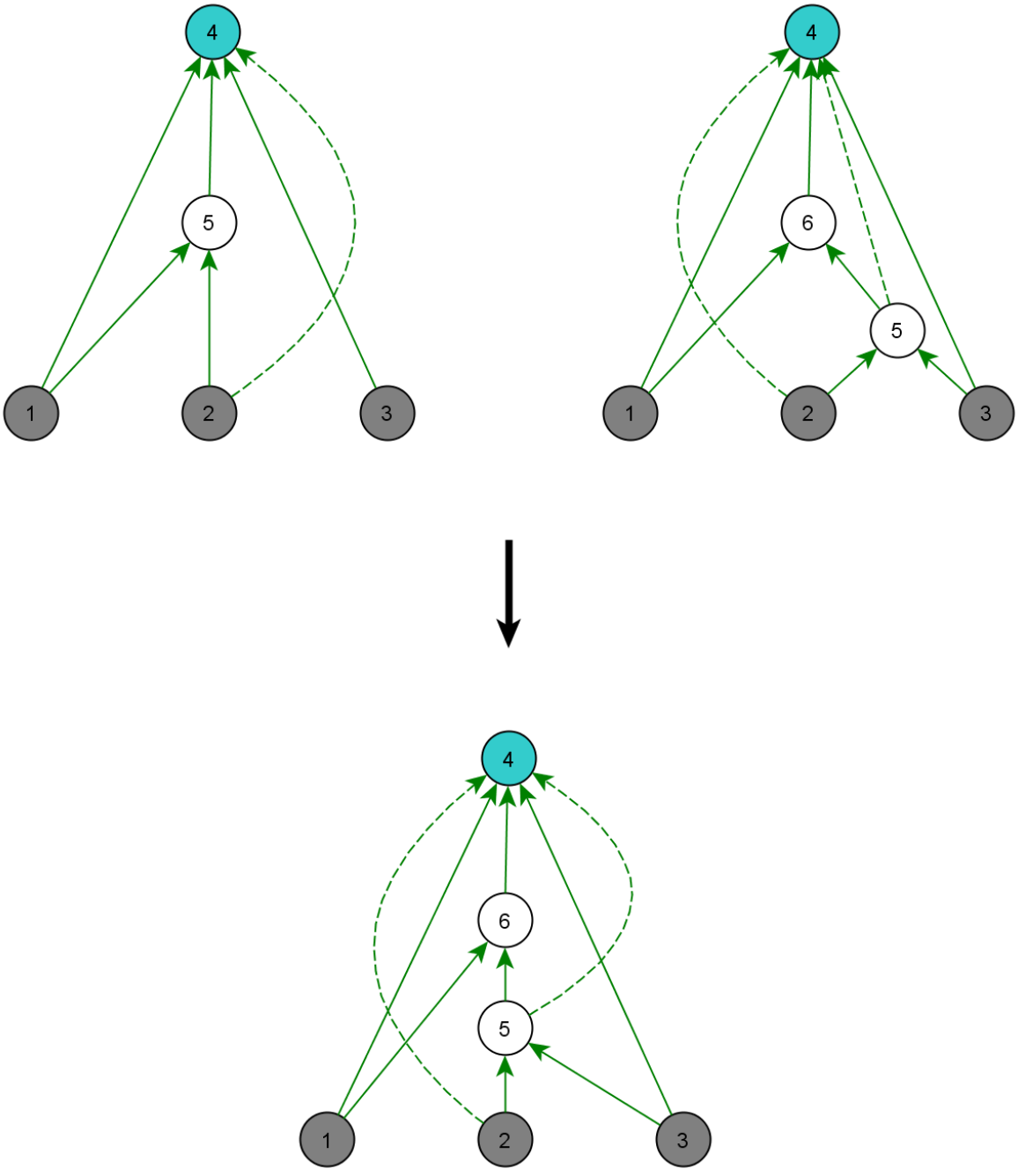

Рисунок 2.3.5 – Скрещивание генов.

Далее, как и в любом генетическом алгоритме создается начальная популяция.

Изначально каждая особь состоит только из пронумерованных входных и выходных нейронов. Каждый выходной нейрон связан с выходным. То есть перед нами классический однослойный персептрон:

Далее к каждой особи применяется фитнес-функция. Она определяет то, насколько данная особь подходит для выполнения возложенных на нее функций. На входы подаются данные, и сравниваются с ожидаемыми выходами.

Эта фитнес функция ранжирует всю популяцию. Те, особи, которые оказались лучше других будут иметь большие вероятности на "продолжение рода", но даже худшие имеют небольшой статистический шанс. Это помогает избежать скатыванию популяции в локальный минимум/максимум.

Далее алгоритм отбирает особей с учетом полученных вероятностей и происходит "скрещивание". Именно тут алгоритм NEAT хорош, ибо имеет простые правила "двуполого" скрещивания генов. Чаще всего именно скрещивание - самая сложная часть генетических алгоритмов.

С помощью скрещивания генерируется полностью новая популяция (в некоторых версиях берутся несколько особей прежнего поколения и добавляются в новое поколение). После чего происходит мутация - либо значения весов с определенной вероятностью изменяются, либо, как в алгоритме NEAT, случайным образом добавляются новые связи/нейроны.

После чего цикл повторяется. Таких циклов/популяций может быть очень много.

Плюсом получившейся нейронной сети является то, что ее топология генерируется на лету, и скорее всего будет значительно сложнее, чем статически заданные.

Из минусов - такая сеть будет заточена на определенный вид задач, и не будет иметь избыточности. Это может уменьшить ее пластичность, что негативно скажется на восприятии неизвестных входных данных.

2.5 Применение алгоритма Neat, для выявления зависимостей

Нейронная сеть на выходе алгоритма NEAT, будет иметь структуру, исследуя которую, можно будет выдвинуть предположение о наличии связей между входом и выходом.

При настройке программы применяются следующие настройки:

- Тип начальных соединений – имеет три возможных варианта:
  - полное соединение – каждое с каждым. Все нейроны сети будут соединены с другими нейронами которое не находятся с ними на одном уровне. Пример на рисунке.
  - Не соединены – изначально какие-либо связи отсутствуют. Пример на рисунке.
  - FS-NEAT – в случайном порядке выбирается один входной нейрон который будет связан со всеми нейронами скрытого слоя или если таковые ответствуют то с выходным слоем нейронов. Пример на рисунке
- Максимальный вес – определяет максимальное значение для веса связей между нейронами.
- Минимальный вес – определяет минимальное значение для веса между нейронами.
- Функции активации – определяет набор функций активации нейронов. Доступные следующие значения: *abs, exp, sigmoid, sin, log*.
- Размер популяции – определяет количество особей в каждом поколении
- Максимальная граница совпадения модели и объекта – когда хотя бы одна особь достигнет данного уровня совпадения с объектом работа алгоритма будет завершена
- Вероятность возникновения новой связи – определяет вероятность того, что при мутации возникнет новая связь между двумя нейронами.
- Вероятность возникновения нового нейрона – определяет вероятность того, что при мутации возникнет новый нейрон в скрытом слое.

- Вероятность удаления связи – определяет вероятность того, что при мутации связь между двумя нейронами будет удалена.
- Вероятность удаления нейрона – определяет вероятность того, что при мутации будет удалена связь между двумя нейронами.

В данной реализации алгоритм будет работать до тех пор, пока не будет достигнута максимальное граница совпадения выхода модели с выходом объекта. Максимальное совпадение рассчитывается по следующей формуле:

$$\varepsilon = 1 - \frac{\sum_{i=1}^{S}(\hat{x}_i - x_i)^2}{S}$$

где ε – величина определяющая совпадение модели и объекта, $\hat{x}_i$ – выход модели, x_i – выход объекта, S – объем выборки.

После получения алгоритмом выборок входа объекта и соответствующего им выхода, начинают генерироваться нейронные сети.

Обозначения в полученных нейронный сетях:

- Серый квадрат – входной нейрон сети;
- Синий круг – выходной нейрон сети;
- Белый круг – скрытый нейрон;
- Сплошная стрелка – активная связь между нейронами, может быть положительной(зеленая) и отрицательно (красная);
- Прерывистая стрелка – неактивная связь между нейронами.

Разберем работу алгоритма на примере пример. Пусть имеется вектор наблюдений $\vec{U} = (u_1, u_2, u_3)$ все компоненты которого распределены от 0 до 10 и выход объекта который построен по следующему принципу: $x_1(u_1, u_2) = 5u_1 + 3u_2 + 10$ и шум в размере 5% от максимального значения функции.

Таким образом алгоритму необходимо по предоставленной выборке входа и выхода объекта построить модель, которая бы отражала зависимость

входной переменой x_1 от выходных переменных u_1 и u_2. Так как входная переменная u_3 не применялась при построении выхода объекта, то она должна быть исключена из модели.

При построении этой модели были использованный параметры, описанные в приложении А. В первом поколении лучшим индивидом была сеть следующего вида (рисунок 2.4.1).

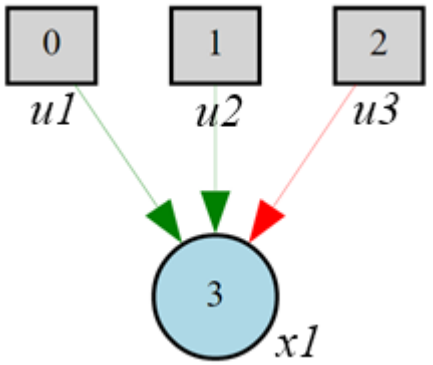

Рисунок 2.4.1 – Лучшая сеть в первом поколении.

Так как в данном примере начальные тип начальных связей был «полное соединение», то из-за отсутствия мутаций не было добавления и удаления связей и узлов, могли измениться только веса связей.

Дальнейшие рисунки нейронных сетей будут показывать уже мутировавшие вариации сетей, полученных в первом поколении.

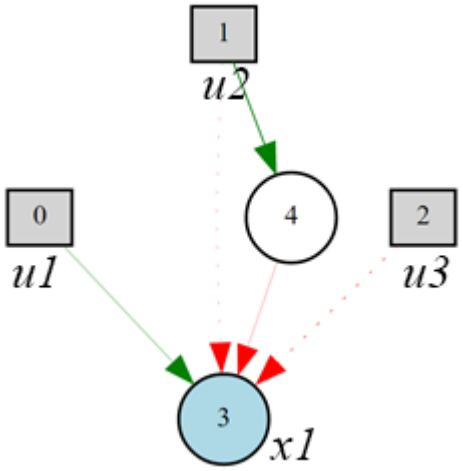

Рисунок 2.4.2 – Лучшая сеть во втором поколении.

На рисунке 2.4.2 видно, что вторая предложенная нейронная сеть существенно отличается от первой. Здесь видно, что входная переменная u_3 не оказывает влияния на выход x. Другие же переменные u_1 и u_2, воздействуют на x. Однако, между u_3и x есть неактивная связь, она может быть активирована в будущих поколениях. Посмотрим на следующую модификацию сети, предложенную алгоритмом (рисунок 2.4.3).

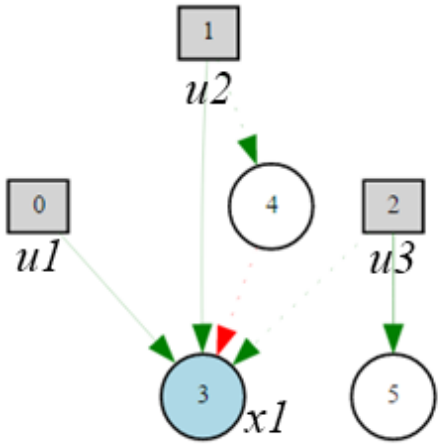

Рисунок 2.4.3 – Лучшая сеть в третьем поколении.

Здесь видно, что алгоритм пытается найти применение u_3, но не соотносит ее с x.

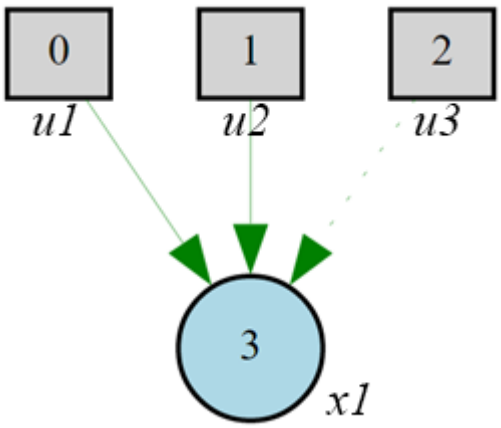

Рисунок 2.4.4 – Финальная полученная сеть.

На рисунке 2.4.4 финальная сеть, предложенная алгоритмом. Анализирую структуру сети можно сделать вывод, что в итоге наилучшей оказалась сеть где

отключена связь между u_3 и x, о чем свидетельствует пунктирная линия от u_3 к x_1. Связь между u_1 и x_1 и связь между u_2 и x_1 присутствует, о чем свидетельствуют зеленые сплошные линии. Соответственно, после исследования сети мы выявили в данном наборе данных.

3 Численные эксперименты

3.1 Апробация алгоритма на искусственных данных

Попробуем выявить линейные зависимости. При построении этих моделей были использованный параметры из приложения Б. Пусть на вход мы подаем $x_1(u_1, u_2) = 5u_1 + 10u_2 + 10$ и независимую переменную u_3. Каждая из переменных определенна на интервале от 0 до 10. $D(x_1): 10 \leq x_1 \leq 160$. Помеха в размере 5% от максимума выхода объекта.

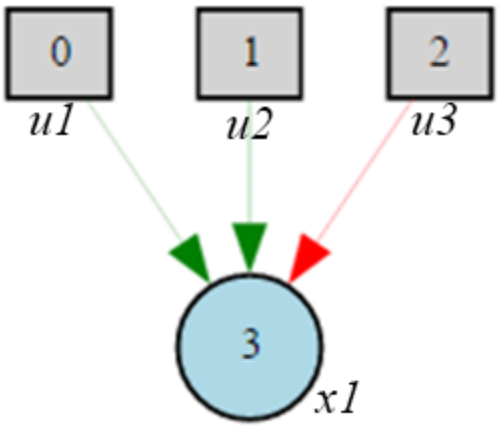

Рисунок 3.1.1 – Лучшая сеть в первом поколении.

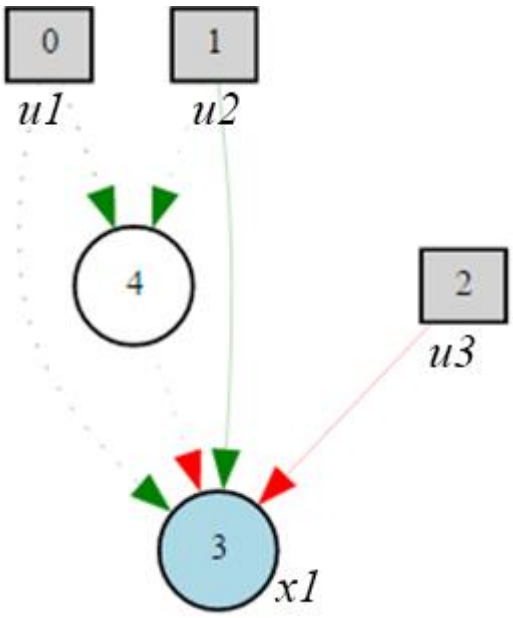

Рисунок 3.1.2 – Лучшая сеть в пятом поколении.

В ходе эволюции нейронной сети (рисунок 3.1.2) был добавлен еще один нейрон между u_1 и x_1. Связи между ними отключены, но могут быть активированы в процессе эволюции.

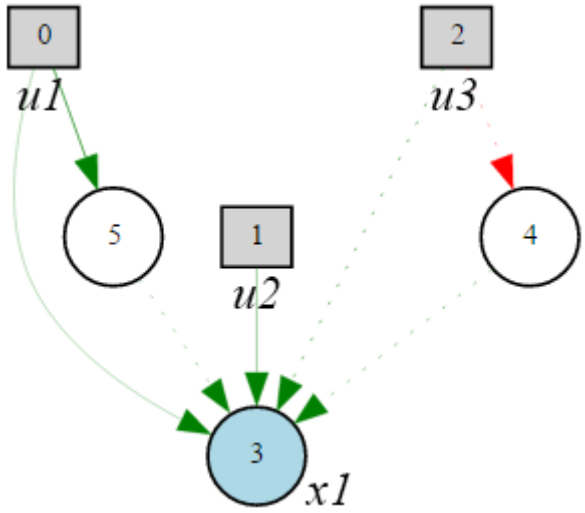

Рисунок 3.1.3 – Финальная полученная сеть.

На рисунке 3.1.3 в полученной нейронной сети алгоритмом был достигнут порог совпадения модели и объекта. Можно сделать вывод, что x зависит от u_1 и u_2 и не зависит от u_3, так как анализирую сеть можно увидеть, что между выходной переменной x_1 и входными u_1 и u_2 есть связи обозначенные зелеными сплошными линиями, что свидетельствует о влиянии данных входов на выход. В то же время, входная переменная u_3 имеет связь с x_1 через зеленую пунктирную линию, что означает отключенную связь, соответственно в данной модели u_3 не влияет на выход модели.

Добавим еще 3 входных переменных и 1 выходную, $x_2(u_4, u_5) = 8u_4 + 9u_5$ и независимая переменная u_6. Каждая из переменных определенна на интервале от 0 до 10. Помеха в размере 5% от максимума выхода объекта. Таким образом алгоритм должен будет разделить входные переменные и соотнести их с соответствующими выходным.

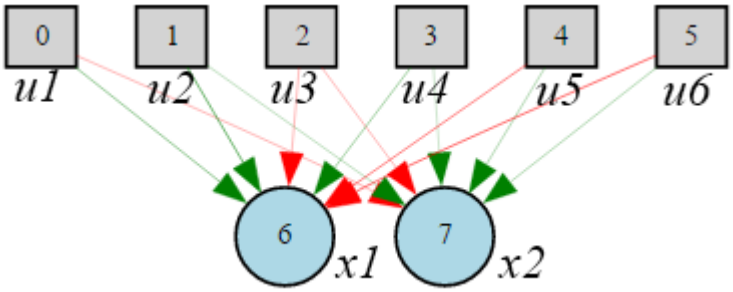

Рисунок 3.1.4 – Лучшая сеть в первом поколении.

В первом поколении все входные переменные связанны с выходными. При типе связей «полное соединение» в ходе эволюции связи между элементами нейронной сети будут удаляться и упрощаться.

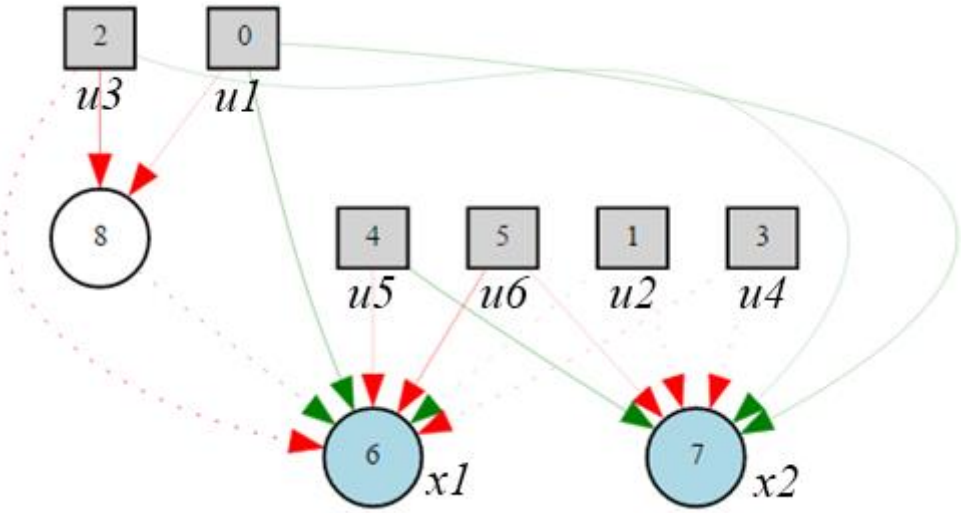

Рисунок 3.1.5 – Лучшая сеть в пятом поколении.

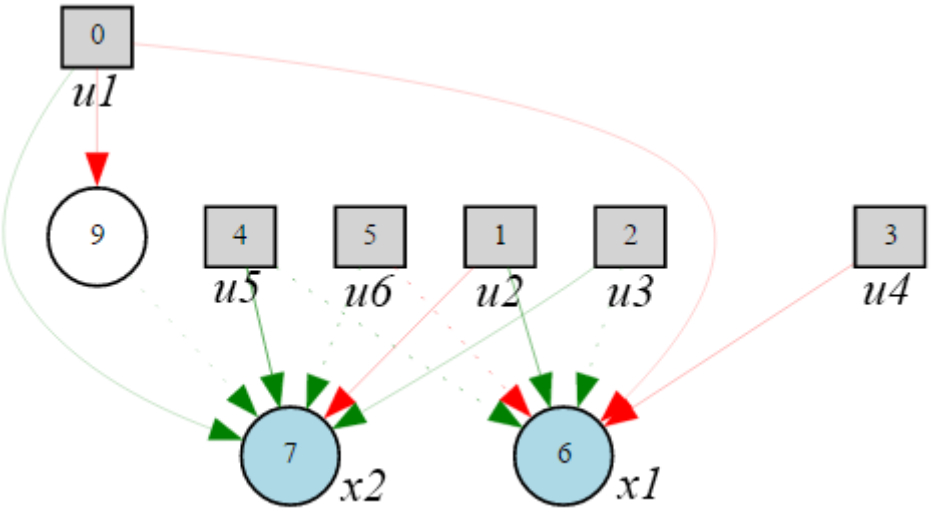

Рисунок 3.1.6 – Лучшая сеть в 10 поколении.

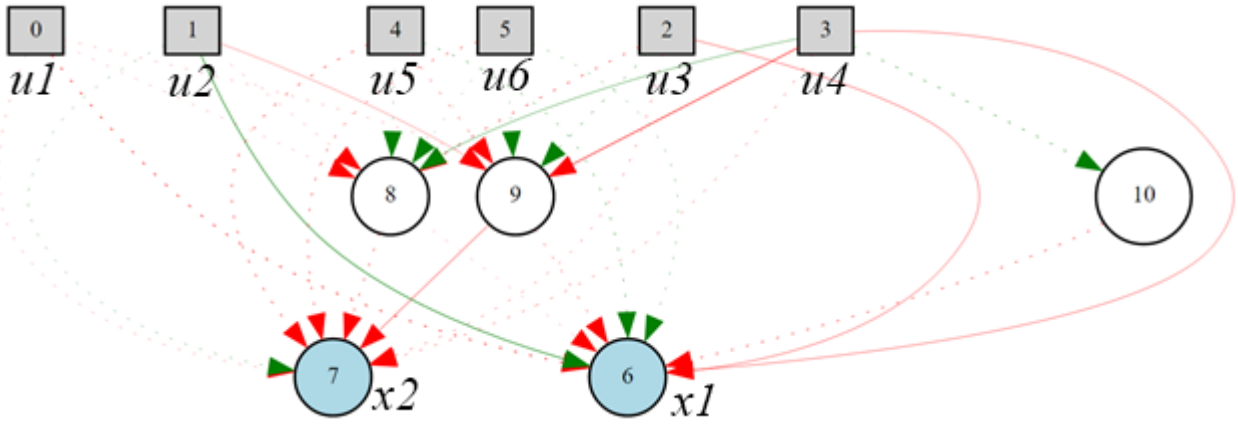

Рисунок 3.1.7 – Лучшая сеть в 15 поколении

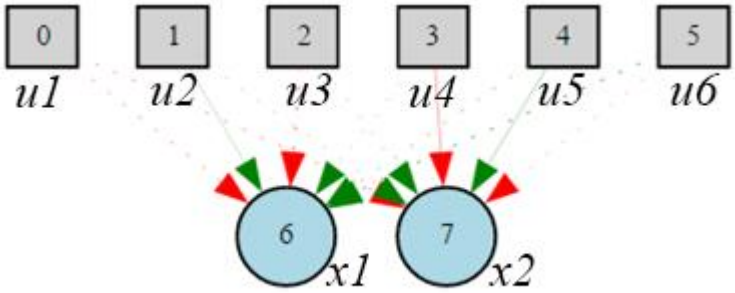

Рисунок 3.1.8 – Финальная полученная сеть.

В ходе эволюции была полученная нейронная сеть (рисунок 3.1.8). Можно выявить связи между u_2 и x_1, и между u_4, u_5 и x_2. Однако связь между u_1 и x_1 не была выявлена. Связь не была выявлена, так как влияние оказываемое u_1 на x_1 незначительно.

Уберем выходы по очереди, но оставим прежнее количество входных переменных.

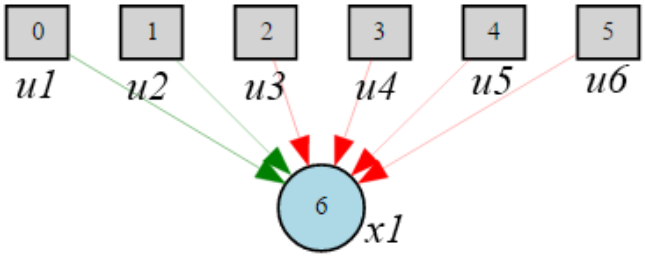

Рисунок 3.1.9 – Лучшая сеть в первом поколении.

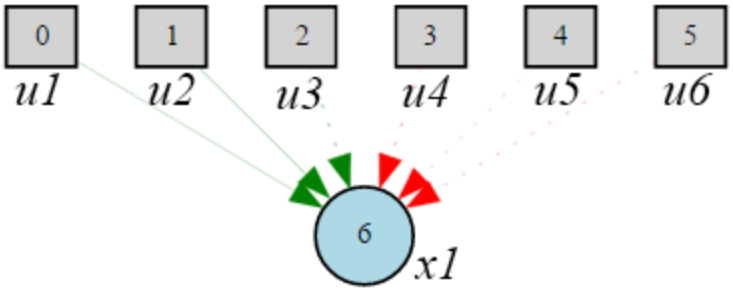

Рисунок 3.1.10 – Финальная полученная сеть.

При устранении выходных переменных, качество построения модели существенно увеличивается. Зависимости определенный верно.

Повторим эксперимент оставив только x_2.

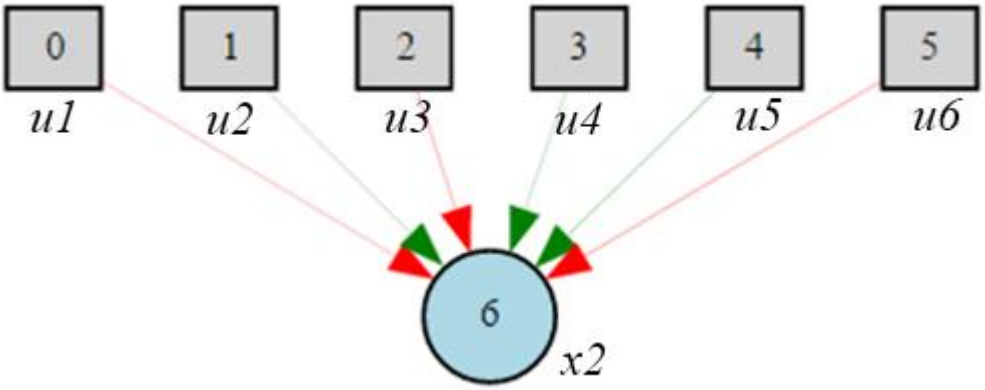

Рисунок 3.1.11 – Лучшая сеть в первом поколении.

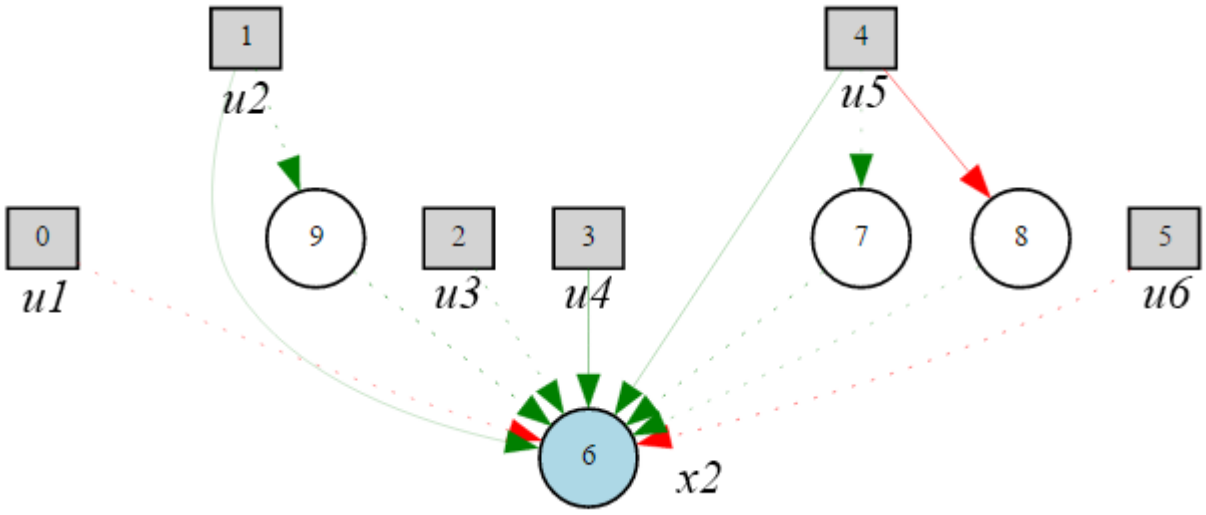

Рисунок 3.1.12 – Финальная полученная сеть.

Алгоритмом ошибочно была указана зависимость между u_2 и x_2. Если исключить все переменные у которых отсутствует связь с выходом, то u_2 отфильтруется (рисунок 3.1.13)

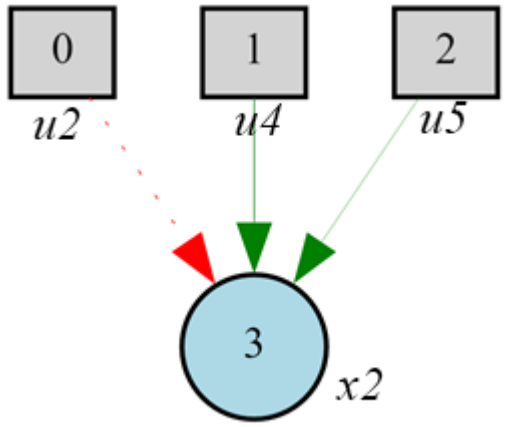

Рисунок 3.1.13 – Сеть после повторной проверки.

Попробуем выявить нелинейные зависимости. При построении этих моделей были использованный параметры из приложения В. Пусть на вход мы подаем $x_1 = 3\sin(u_1) + \sqrt{u_2}$ и независимую переменную u_3. Помеха в размере 5% от максимума выхода объекта. При построении этой модели были использованный параметры из приложения 3.

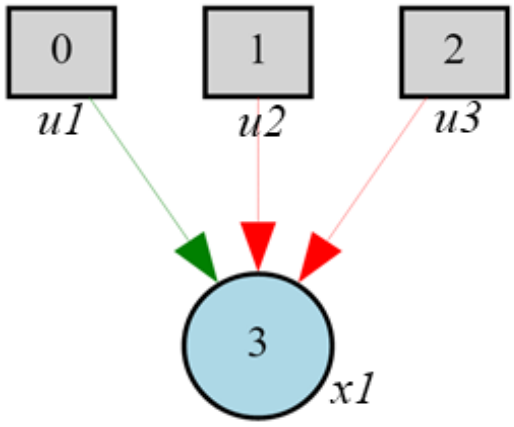

Рисунок 3.1.14 – Лучшая сеть в первом поколении.

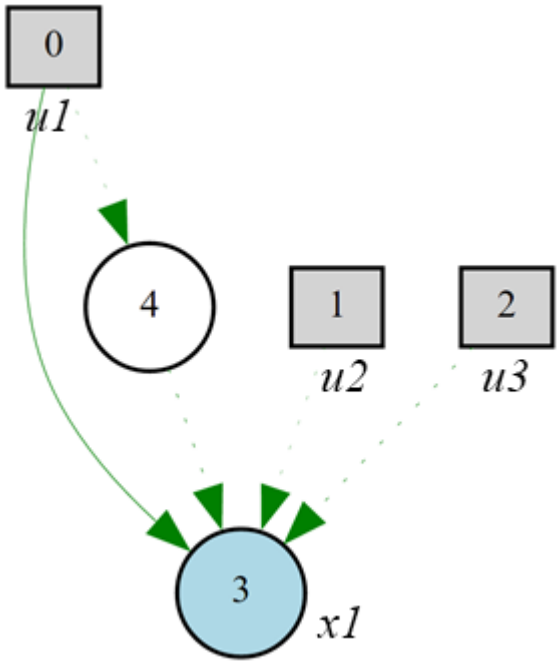

Рисунок 3.1.15 – Лучшая сеть в пятом поколении.

3.2 Апробация алгоритма на данных с ТЭЦ-1

С ТЭЦ-1 была получена матрица наблюдений процесса дискретно-непрерывного типа за период с 1 марта по 1 апреля. Требуется вывить зависимости мелу переменными и сравнить результаты с технологической схемой (рисунок 3.2.1). Вид данных представлен на рисунке 3.2.2.

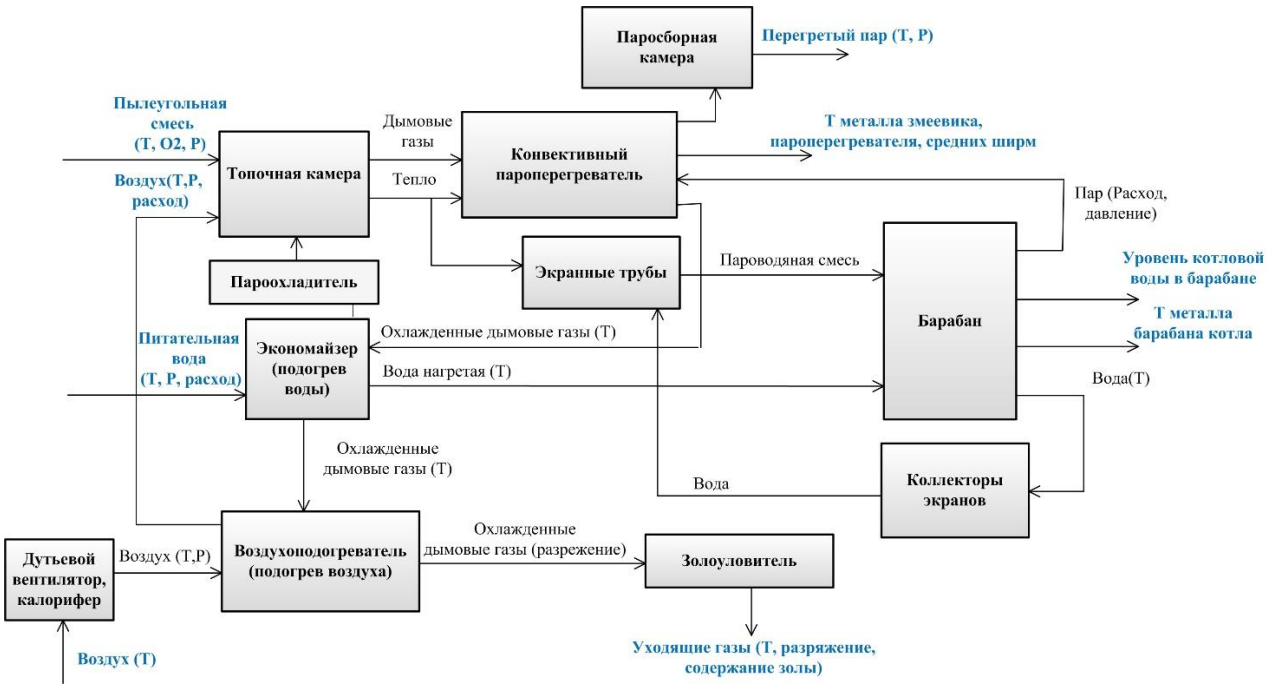

Рисунок 3.2.1 – Технологическая схема процесса

	60HAH12CP001_XQ01 Давление пара в паросборной камере, справа, нитка Б	60LAB00CF001_XQ01 Расход питательной воды к котлу	60LAB00CP001_XQ01 Давление питательной воды к котлу	60LAB00CT001_XQ01 Температура питательной воды к котлу	60LBA02CF001_XQ01 Расход перегретого пара, справа, нитка Б	60HLA12CP001_XQ01 Давление воздуха за дутьевым вентилятором Б	60HLA22CP001_XQ01 Давление воздуха до воздухоподогревателя, справа	60HLA31CP001_XQ01 Давление воздуха за воздухоподогревателем к горелкам Б, В слева	60HLA32CP001_XQ01 Давление воздуха за воздухоподогревателем к горелкам А, Г справа	60HLA41CP001_XQ01 Давление воздуха за воздухоподогревателем к горелкам А, Г слева	60HLA42CP001_XQ01 Давление воздуха за воздухоподогревателем к горелкам Б, В справа
1	137,086	414,078	190,141	225,625	207,992	273,734	235,688	147,008	-19,129	36,627	154,969
2	137,125	433,375	189,625	225,625	203,695	273,797	227,172	150,664	-19,251	33,146	155,062
3	137,242	428,125	188,789	225,625	205,461	267,297	234,836	151,492	-19,098	38,092	146,734
4	137,422	426,297	190,273	225,688	203,789	275,812	230,227	156,711	-19,129	39,738	147,281
5	137,523	408,641	190,523	225,688	204,43	274,016	222,812	148,438	-19,373	34,918	144,016
6	137,523	421,047	189,859	225,688	205,117	272,453	241,125	147,25	-19,342	33,086	149,078
7	137,703	421,625	190,414	225,688	205,469	275,484	231,109	150,211	-19,129	32,965	152,344
8	137,922	421,781	189,555	225,812	205,07	272,953	234,102	152,008	-18,945	37,236	151,859
9	137,961	400,484	189,648	225,812	206,422	271,031	239,195	155,703	-19,007	38,152	147,891
10	138,133	441,875	189,344	225,938	207,133	275,641	234,039	157,289	-18,885	40,289	152,344
11	138,156	439,203	188,633	225,938	207,5	274,109	225,438	156,711	-18,763	37,359	149,633
12	138,117	455,344	187,875	225,938	207,352	275,234	229,406	159,82	-19,251	41,418	154,148
13	138,195	428,922	188,391	225,938	208,625	272,641	234,742	150,914	-19,129	35,742	149,781
14	138,234	411,781	188,945	225,938	206,906	278,406	229,492	150,914	-19,251	37,389	149,352
15	138,219	392,938	188,977	225,938	204,672	275,516	229,406	156,469	-19,037	37,695	147,859
16	138,18	412,281	187,742	225,938	206,633	272,547	236,758	152,406	-19,159	32,902	151,305
17	138,156	371,594	189,898	225,938	208,148	272,547	239,078	153,289	-19,067	40,318	152,93
18	138	418,625	190,031	225,938	206,703	282,281	218,047	154,234	-19,037	35,01	151,188
19	137,938	455,5	188,945	225,812	206,938	270,75	241,031	155,703	-19,129	37,848	143,344
20	137,898	394,281	189,531	225,812	209,594	269,312	230,164	149,477	-19,159	37,572	147,375
21	137,82	424,688	189,227	225,938	209,18	275,641	238,891	155,641	-18,915	42,211	157,047
22	137,82	446,578	188,711	225,938	209,523	269,859	223,453	141,359	-19,129	33,482	143,773
23	137,742	407,562	189,078	225,938	208,43	274,688	224,062	153,016	-18,731	36,84	146,367
24	137,82	422,484	188,695	225,938	208,188	277	237,945	148,289	-19,007	38,121	151,219

Рисунок 3.2.2 – Пример данных

Для выявления зависимостей необходимо выделить переменные, для которых необходимо найти такие входные переменные, которые оказывают на них наибольшее влияние. При построении этих моделей были использованный параметры из приложения Г. В качестве выхода выберем Расход перегретого пара(x_1), а в качестве входа подадим все остальные параметры. Полученная нейронная сеть отразит те переменные, влияние которых было максимально (рисунок 3.2.3).

Рисунок 3.2.3 – Финальная сеть нейронной сети

Из структуры нейронной сети можно выделить, что переменные, которые оказывают влияние на выход следующе:

- Давление питательной воды к котлу(u_1);
- Давление воздуха до воздухоподогревателя(u_2);
- Давление воздуха за воздухоподогревателем к горелкам(u_3);
- Температура дымовых газов за электрофильтрами(u_4);
- Температура пылеугольной смеси после мельницы-вентилятора(u_5).

Для того, чтобы выделить из этих 5 переменных, наиболее влиятельную, оставим только эти 5 переменных в качестве входных параметров нейронной сети.

Результатом серии экспериментов стали несколько нейронных сетей, которые показали наилучшее сходство с модель. Рисунок 3.2.4 и рисунок 3.2.5 показывают две наилучших сети из всех выборок, однако остальные сети имели схожую структуру.

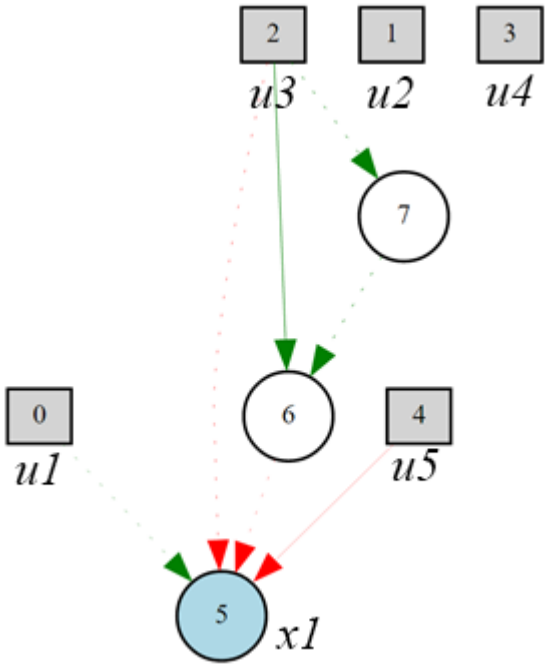

Рисунок 3.2.4 – Финальная сеть

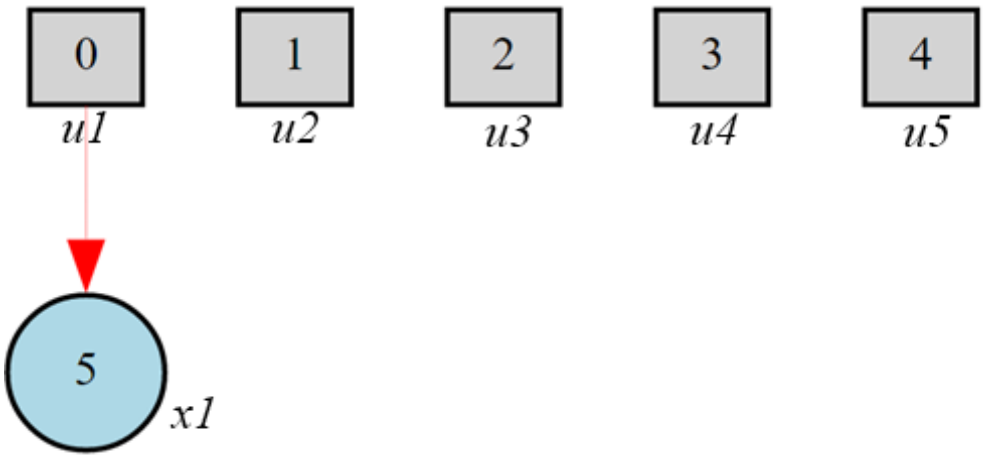

Рисунок 3.2.5 – Финальная сеть

Из приведенных сетей видно, что существенное влияние на выходную переменную оказывают следующие входные переменные:

- Давление питательной воды к котлу;
- Температура пылеугольной смеси после мельницы вентилятора.

Если соотнести данные результаты со схемой на рисунке 3.2.1, то можно сделать вывод, что данные зависимости не противоречат логике процесса.

3.3 Вывод

Алгоритм выявления зависимостей между данными с использованием анализа нейронных сетей, построенных при помощи алгоритма нейроэволюции, был апробирован на искусственных данных и показал не плохой результат в выявлении зависимостей. Результаты, полученные при применении алгоритма на реальных данных, показал результаты, которые не противоречат технологической схеме, таким образом можно утверждать, что алгоритм способен выявлять сильные зависимости между переменными.

ЗАКЛЮЧЕНИЕ

В итоге работы стоит отметить, что поставленные задачи достигнуты, и по результатам исследования был сделан ряд выводов.

Было проведено исследование различных классических методов выявления зависимостей между переменными, а также представлено описание и работа метода основанного на анализе структуры нейронной сети построенной с применением алгоритма нейроэволюции.

В работе представлено описание работы алгоритма построения нейронных сетей, а также пример анализа полученных результатов. Данный алгоритм был реализован для ПК.

При проверке работы алгоритма на искусственных данных, он показал хорошие результаты при нахождении в ситуации, когда модель строится для одного выхода. При моделировании нейронной сети с несколькими выходами алгоритм допускал незначительные ошибки и мог не обнаруживать связи, между переменными, влияние которых не значительно велико.

В дальнейшем предполагается разработать алгоритм выявления зависимостей, основанный на методе анализа параметра размытости в методе непараметрической регрессии Надарая-Ватсона.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Тарасенко, Ф. П. Прикладной системный анализ (Наука и искусство решения проблем) : учебник / Ф. П. Тарасенко. – Томск: Изд-во Том. ун-та –2004. –224 с.

2. Надарая, Э. А. Непараметрические оценки кривой регрессии // Труды ВУ АН ГрССР.- Тбилиси: –1965. –№5. –С. 56-68.

3. Дармаев, Т. Г. Об одном эволюционном алгоритме настройки искусственных нейронных сетей / Т.Г. Дармаев, Ф.В. Хандаров // Вестник Бурятского Государственного Университета – Томск: –2012. –№SB. –С. 211-217

4. Филатова Т. В. Применение нейронных сетей для аппроксимации функций / Т. В. Филатова // Вестник: Вестник Томского государственного университета – Томск: –2004. –№284. – С. 121-125.

5. Kenneth, O. S. Efficient Evolution of Neural Networks Through Complexification / O. S. Kenneth. – Texas: The University of Texas at Austin, 2004. –180p.

6. Гмурман, В. Е. Теория вероятностей и математическая статистика: учебное пособие / В. Е. Гмурман. – Москва : 2003. – 479 с.

ПРИЛОЖЕНИЕ А

- Тип начальных соединений – полное соединение
- Максимальный вес – 50
- Минимальный вес – (-50)
- Функции активации – sigmoid
- Размер популяции – 100
- Максимальная граница совпадения модели и объекта – 5
- Вероятность возникновения новой связи – 0.9
- Вероятность возникновения нового нейрона – 0.9
- Вероятность удаления связи – 0.2
- Вероятность удаления нейрона – 0.2

ПРИЛОЖЕНИЕ Б

- Тип начальных соединений – полное соединение
- Максимальный вес – 50
- Минимальный вес – (-50)
- Функции активации – sigmoid
- Размер популяции – 1000
- Максимальная граница совпадения модели и объекта – 0.97
- Вероятность возникновения новой связи – 0.9
- Вероятность возникновения нового нейрона – 0.9
- Вероятность удаления связи – 0.3
- Вероятность удаления нейрона – 0.3

ПРИЛОЖЕНИЕ В

- Тип начальных соединений – полное соединение
- Максимальный вес – 50
- Минимальный вес – (-50)
- Функции активации – sigmoid, sin
- Размер популяции – 500
- Максимальная граница совпадения модели и объекта – 0.97
- Вероятность возникновения новой связи – 0.9
- Вероятность возникновения нового нейрона – 0.9
- Вероятность удаления связи – 0.3
- Вероятность удаления нейрона – 0.3

ПРИЛОЖЕНИЕ Г

- Тип начальных соединений – полное соединение
- Максимальный вес – 150
- Минимальный вес – (-150)
- Функции активации – sigmoid
- Размер популяции – 150
- Максимальная граница совпадения модели и объекта – 0.97
- Вероятность возникновения новой связи – 0.95
- Вероятность возникновения нового нейрона – 0.95
- Вероятность удаления связи – 0.1
- Вероятность удаления нейрона – 0.1