

Санкт-Петербургский государственный университет
Факультет прикладной математики - процессов управления
Кафедра математического моделирования энергетических систем

Букреев Евгений Михайлович

Магистерская диссертация
Прогнозирование потребления нефти с
использованием нейронной сети

Направление 01.04.02

Математическое и информационное обеспечение экономической
деятельности

Заведующий кафедрой,
доктор физ.-мат. наук,
профессор

Захаров В. В.

Научный руководитель,
кандидат физ.-мат. наук,
доцент

Свиркин М. В.

Санкт-Петербург

2017

Оглавление

Введение	3
FPNN-метод	7
FP-алгоритм	7
Нейронные сети	12
Прогнозирование потребления нефти	21
Набор данных о потреблении нефти странами ОПЕК	21
Характеристики эксперимента	23
Сравнение работы алгоритмов	26
Прогнозирование потребления основными странами-импортерами российской нефти	30
Заключение	34
Список литературы	36
Приложение	42

Введение

По данным Международного Энергетического Агентства(МЭА) общий объем потребления нефти в 2015 году составил примерно 4.3 млрд тонн, что на 55% больше чем в 1973 году [1]. Таким образом, средний рост потребления нефти за прошедшие с момента «нефтяного шока» годы составил приблизительно 1% в год. Вместе с тем, после завершения экономического кризиса 1973–1983 годов потребление нефти стабильно росло вплоть до начала кризиса 2008 года. Динамика потребления сырой нефти представлена ниже на Рис. 1

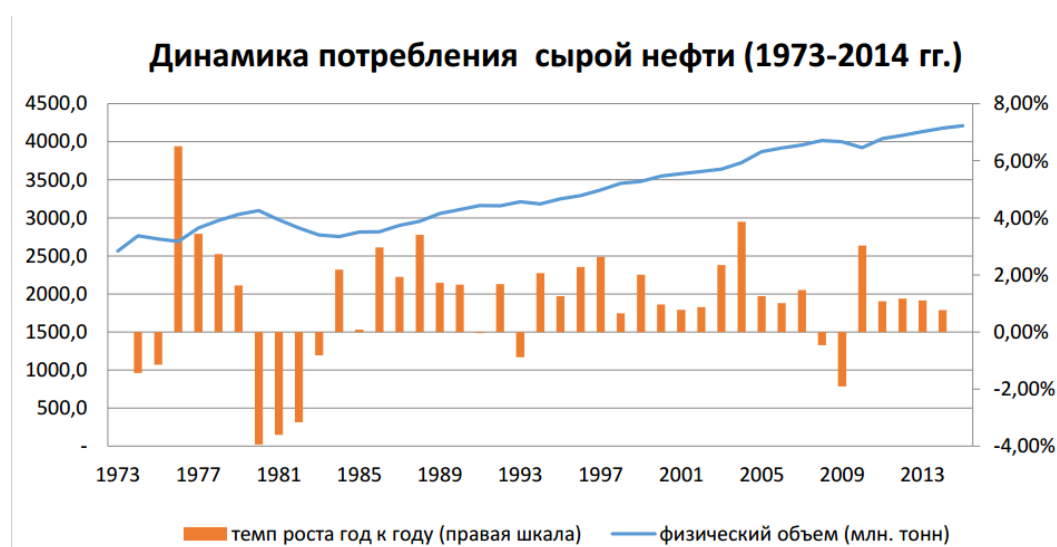


Рис. 1: Динамика потребления сырой нефти (1973–2014 гг.)

Данный рост был связан, прежде всего, с тем, что в странах «третьего мира» активно шла индустриализация и, несмотря на снижение физического потребления в развитых странах (за счет выноса производства в страны «третьего мира» и повышения энергоэффективности), общая потребность в энергоресурсах продолжала расти. Бурная индустриализация в странах «третьего мира» в настоящее время близка к завершению, однако, с точки зрения потребления нефти в долгосрочной перспективе это приведет не к падению, а к стабилизации спроса. Несмотря на то, что МЭА пересмотрел свой долгосрочный прогноз в сторону понижения, все равно по

обновленному прогнозу предполагается, что к 2040 г. спрос вырастет, хотя совокупный рост в период 2020–2040 годов не превысит 5%. Это означает, что к 2020 году объем годового потребления нефти составит примерно 4.35 млрд тонн, а к 2040 году – 4.65 млрд тонн. Следует отметить, что, несмотря на непрерывный рост спроса, предложение нефти в период с 1965 по 2014 год практически никогда не отставало от спроса более чем на 0.5%. Более того, на рынке в основном имел место излишек.

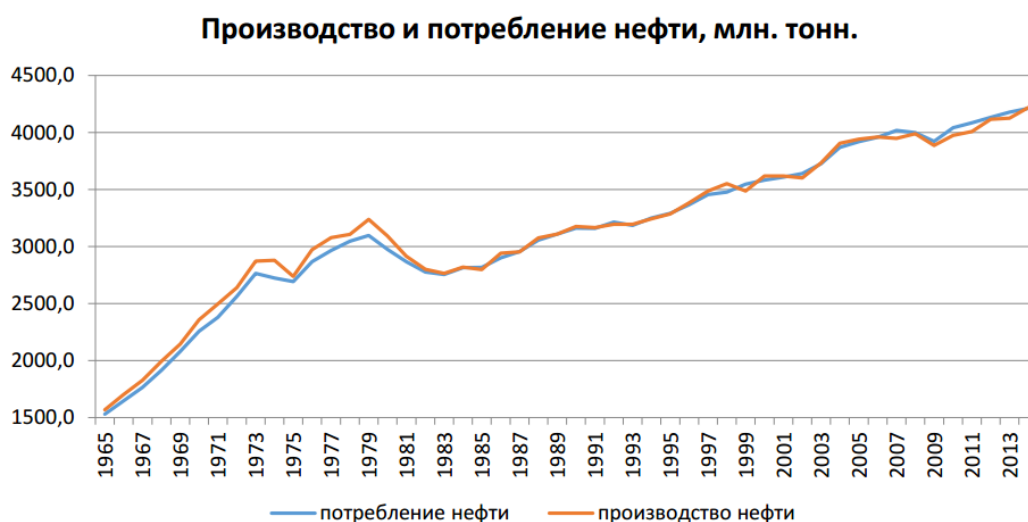


Рис. 2: Производство и потребление нефти, млн.тонн



Рис. 3: Излишек / дефицит нефти, млн.тонн

Таким образом, за исключением периода 2007–2013 гг. дефицит нефти на рынке носил эпизодический характер. Более типичной была ситуация

превышения предложения над спросом, что не мешало при этом нефти как драматично расти в период «нефтяного шока» 1970-х годов, так и столь же драматично снижаться в середине 1980-х годов.

В 2012 году был предложен новый метаэвристический алгоритм, названный алгоритмом опыления цветка (FP) [2]. Эффективность этого алгоритма можно объяснить следующими причинами: (1) Насекомые-опылители могут преодолевать большие расстояния, что позволяет алгоритму избегать особенности локального ландшафта и производить поиск на очень большом пространстве. (2) алгоритм гарантирует, что похожие виды цветов выбираются последовательно, что обеспечивает быструю сходимость к оптимальному решению. Для решения задач оптимизации алгоритм опыления цветка оказывается производительней чем генетический алгоритм (GA), метода роя частиц (PSO) [2], и алгоритма колонии пчел (ABC) [3].

Искусственные нейронные сети (NN) упоминаются как одно из мощнейших вычислительных средств [4], вследствие своей эффективности и надежности. Нейронная сеть допускает отсутствующие или ошибочные значения, и любая связь может быть смоделирована с произвольной точностью. Нейронная сеть считается одним из самых надежных и многообещающих вычислительных методов [5]. В данной работе алгоритм опыления цветка обеспечивает обучение нейронной сети в модели прогнозирования потребления нефти странами ОПЕК для сравнения этого алгоритма по показателям скорости сходимости и точности прогнозирования, а далее этот алгоритм применяется для прогнозирования потребления нефти основными странами-импортерами российской нефти.

Человеческую деятельность в современном мире невозможно представить без энергии. Таким образом, недостаток энергии может сделать жизнь человека очень трудной или даже невозможной [6]. Около 2/3 энер-

гетических потребностей в мире обеспечивается нефтью и газом [7]. Потребление нефти прямо связано с экономическим ростом [8]. Это означает, что более развитая экономика потребляет больше нефти. И наоборот, если экономическая деятельность не улучшается, то соответственно потребление нефти падает.

FPNN-метод

В этом разделе рассматриваются теоретические основы FP и NN, необходимые для понимания гибридного метода прогнозирования потребления нефти. Раздел содержит два основных подраздела: основные операции FP и его функционирование для достижения желаемой цели. Описывается краткое описание модели NN, а также даны ссылки на литературу для получения более подробной информации о нейронных сетях, поскольку нейронная сеть широко обсуждается в литературе. Подробно даются теоретические основы гибридизации FP и NN.

FP-алгоритм

Природа в течение миллионов и миллиардов лет решает сложные эволюционные задачи, и многие биологические системы развиваются удивительной эффективностью в достижении своих эволюционных целей, таких как воспроизводство. На основе биологических систем в течение нескольких последних десятилетий было разработано много успешных природных алгоритмов [9], [10]. Например, генетические алгоритмы были основаны на дарвиновской эволюции биологических систем [11], а оптимизация роя частиц была разработана на особенностях поведения птиц и рыб [12], [13]; алгоритм летучей мыши основывался на эхолокационном поведении летучих мышей [14], а алгоритм светлячков основан на проблесковом свете тропических светлячков [12], [15]. Все эти алгоритмы были применяются к широкому спектру задач.

Во многих задачах мы должны попытаться найти оптимальное решение заданной проблемы в условиях очень сложных ограничений. Такие задачи оптимизации зачастую сильно нелинейны, найти оптимальные решения является очень сложной задачей, если это не невозможно. Большин-

ство обычных способов оптимизации плохо работают для задач с нелинейностью и мультимодальностью. Нынешняя тенденция – использовать природные метаэвристические алгоритмы для решения таких сложных задач, и было показано, что метаэвристика очень эффективна. По этой причине литература метаэвристики значительно расширилась за последние два десятилетия [12], [13]. До сих пор исследователи использовали только очень ограниченные природные особенности, и на данный момент существует возможность для дальнейшего развития таких алгоритмов.

С точки зрения биологической эволюции, цель цветочного опыления – выживание наиболее приспособленного растения и оптимальное воспроизводство таких растений. Это на самом деле есть процесс оптимизации видов растений. Все перечисленные выше факторы и процессы цветочного опыления взаимодействуют, чтобы добиться оптимального размножения цветковых растений. Поэтому в 2012 году это вдохновило Янга на разработку нового алгоритма оптимизации.

По оценкам, в природе насчитывается более четверти миллиона видов цветковых растений, и около 80% всех видов растений являются цветущими. До сих пор остается загадкой, как цветущие растения стали доминировать в природе с мелового периода [16], [17]. Цветущее растение развивается уже более 125 миллионов лет, и цветы стали настолько влиятельными в эволюции, что мы не можем представить, как будет развиваться мир без цветов. Главной целью цветка является, в конечном счете, воспроизводство через опыление. Цветочное опыление обычно связано с переносом пыльцы, и такая передача часто связана с опылителями, такими как насекомые, птицы, летучие мыши и другие животные. Фактически, некоторые цветы и насекомые объединились в очень специализированное партнерство цветочных опылителей. Например, некоторые цветы могут опыляться только от определенных видов насекомых [15].

Опыление может принимать две основные формы: абиотические и биотические. Около 90% цветковых растений относятся к биотическому опылению, то есть пыльца переносится опылителем, например, насекомыми. Около 10% опыления занимает абиотическая форма, которая не требует каких-либо опылителей. Ветер, диффузия в воде помогают опылению таких цветущих растений, и хорошим примером такого опыления является трава [15]. Опылители могут быть очень разнообразными. По оценкам, существует по меньшей мере 200 000 разновидностей опылителей (различные насекомые, летучие мыши и птицы).

Янг смог эмулировать особенности цветочного опыления у цветковых растений, чтобы разработать алгоритм опыления цветка (FPA) на основе следующих правил [2]:

1. Процессы глобального опыления биотичны, т.е являются формой перекрестного опыления, в котором пыльца от одного цветка переносится на рыльце пестика другого цветка с помощью опылителя, который совершает полет Леви.
2. Локальное опыление рассматривается как форма самоопыления. При самоопылении пыльца переносится на рыльце пестика того же самого цветка или между цветками одного растения.
3. Постоянство цветка — это склонность отдельных опылителей посещать исключительно определенные виды цветов в обход других доступных видов, которые потенциально могут содержать больше нектара.
4. Локальное и глобальное опыление контролируются вероятностью $p \in [0, 1]$. Вследствие физической близости и других факторов (например, ветер) локальное опыление может иметь значительную долю p в общем процессе опыления.

Медоносные пчелы являются хорошим примером постоянства цветка [19]. То есть, эти опылители склонны посещать определенные виды цветов, минуя другие виды. Такое свойство может иметь эволюционные преимущества, поскольку это максимизирует перенос цветочной пыльцы на те же самые растения и таким образом максимизирует воспроизводство тех же видов цветов. Постоянство цветка может быть выгодным и для опылителей, поскольку они могут быть уверены, что поставка нектара доступна без дополнительных затрат на обучение или исследование. Вместо того, чтобы сосредотачиваться на некоторых потенциально более полезных новых видах цветков, постоянство цветка может потребовать минимальных инвестиционных затрат и более вероятного гарантированного потребления нектара [20].

Опыление может быть достигнуто путем самоопыления или перекрестного опыления. Перекрестное опыление или аллогамия означает, что опыление может происходить от пыльцы цветка другого растения, тогда как самоопыление является оплодотворением одного цветка, например персика, из пыльцы одного и того же цветка или разных цветов того же растения, что часто происходит, когда нет надежного опылителя.

Биотическое, перекрестное опыление может происходить на большом расстоянии, а опылители, такие как пчелы, летучие мыши, птицы и мухи, могут летать на большие расстояния, поэтому их рассматривают как глобальное опыление. Кроме того, пчелы и птицы могут вести себя по законам полета Леви [21], при этом направление и расстояние полета подчиняются распределению Леви.

В реальной жизни растение может иметь несколько цветов, и каждый цветок, как правило, выделяет миллионы или даже миллиарды гамет (репродуктивных клеток). Для упрощения разработки гибридного алгоритма предполагается, что каждое растение имеет один цветок и каж-

дый цветок испускает только одну пыльцу-гамету. Следовательно, гамета, цветок, растение, решение x_i не различаются. Все это одно и то же. Это означает, что решение x_i эквивалентно цветку и/или пыльцевой гамете. В будущих исследованиях можно легко распространить эти предположения на несколько цветочных гамет для каждого цветка и несколько цветов для задач многокритериальной оптимизации. В FR-алгоритме есть два ключевых этапа: глобальное и локальное опыления. В глобальном опылении цветочная пыльца переносится опылителями, например, насекомыми. Пыльца может перемещаться на большие расстояния, т.к. насекомые могут летать и перемещаться на дальние расстояния. Этот процесс гарантирует опыление и размножение наиболее приспособленного решения g_* . Правила 1 и 3 можно представить как:

$$x_i^{t+1} = x_i^t + L(g_* - x_i^t), \quad (1)$$

где x_i^t — пыльца i или вектор-решение x_i ; t — итерация; g_* — текущее самое лучшее решение, найденное среди всех решений в текущей популяции/итерации/эпохе; L — сила опыления(размер шага). Так как насекомые могут перемещаться на большие расстояния с различными шагами, можно использовать полет Леви, чтобы имитировать эту особенность, то есть полет Леви(Levy flight) представляет собой движение насекомых; $L > 0$ из распределения Леви:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\pi \lambda / 2)}{\pi} s^{\frac{1}{1+\lambda}}, (s \gg s_0 > 0), \quad (2)$$

где $\Gamma(\lambda)$ — стандартная гамма-функция, и это распределение справедливо для больших шагов $s > 0$. Символ \sim в (2) означает, что случайная величина подчиняется распределению в правой части. Правила 2 и 3 представлены как:

$$x_i^{t+1} = x_i^t + \epsilon(x_j^t - x_k^t), \quad (3)$$

где x_j^t и x_k^t – пыльца с разных цветов одного и того же вида растения, имитируя таким образом постоянство цветка в ограниченной окрестности; ϵ – параметр, который берется из равномерного распределения на $[0, 1]$. Выбор между глобальным и локальным опылениями происходит с помощью параметра p . На практике соседние цветы в не столь отдаленном районе, скорее всего, будут опыляться местной пылью. Для этого используется вероятность переключения (правило 4) для переключения между глобальным и локальным опылениями. Для начала можно использовать $p = 0.5$ как исходное значение, а затем провести параметрическое исследование, чтобы найти наиболее подходящий диапазон параметра. Из моделирования Янга было обнаружено, что $p = 0.8$ работает лучше всего для большинства задач. Эффективность FR является следствием следующих причин:

1. Насекомые-опылители могут перемещаться на большие расстояния, что позволяет избежать особенностей местного ландшафта и искать на очень большом пространстве.
2. Схожие виды цветов выбираются последовательно и часто, что гарантирует быструю сходимость к оптимальному решению.

Нейронные сети

Дадим следующее определение нейронных сетей, выступающих в роли адаптивной машины [23]:

Нейронная сеть — это громадный распределенный параллельный процессор, состоящий из элементарных единиц обработки информации, накапливающих экспериментальные знания и предоставляющих их для последующей обработки. Нейронная сеть сходна с мозгом с двух точек зрения.

- Знания поступают в нейронную сеть из окружающей среды и исполь-

зуются в процессе обучения.

- Для накопления знаний применяются связи между нейронами, называемые синаптическими весами.

Процедура, используемая для процесса обучения, называется алгоритмом обучения (learning algorithm). Эта процедура выстраивает в определенном порядке синаптические веса нейронной сети для обеспечения необходимой структуры взаимосвязей нейронов. Изменение синаптических весов представляет собой традиционный метод настройки нейронных сетей. Этот подход очень близок к теории линейных адаптивных фильтров, которая уже давно заявила о себе и применяется в различных областях деятельности человека [24], [25]. Однако нейронные сети могут изменять собственную топологию. Это обусловлено тем фактом, что нейроны в человеческом мозге постоянно отмирают, а новые синаптические связи постоянно создаются.

Многослойные сети прямого распространения – класс нейронных сетей прямого распространения, который характеризуется наличием одного или нескольких скрытых слоев (hidden layer), узлы которых называются скрытыми нейронами (hidden neuron), или скрытыми элементами (hidden unit). Функция последних заключается в посредничестве между внешним входным сигналом и выходом нейронной сети. Добавляя один или несколько скрытых слоев, мы можем выделить статистики высокого порядка. Такая сеть позволяет выделять глобальные свойства данных с помощью локальных соединений за счет наличия дополнительных синаптических связей и повышения уровня взаимодействия нейронов [26]. Способность скрытых нейронов выделять статистические зависимости высокого порядка особенно существенна, когда размер входного слоя достаточно велик. Узлы источника входного слоя сети формируют соответствующие элементы шаблона активации (входной вектор), которые составляют входной сигнал, посту-

пающий на нейроны (вычислительные элементы) второго слоя (т.е. первого скрытого слоя). Выходные сигналы второго слоя используются в качестве входных для третьего слоя и т.д. Обычно нейроны каждого из слоев сети используют в качестве входных сигналов выходные сигналы нейронов только предыдущего слоя. Набор выходных сигналов нейронов выходного (последнего) слоя сети определяет общий отклик сети на данный входной образ, сформированный узлами источника входного (первого) слоя. Сеть, показанная на рис. 4, называется сетью 10-4-2, так как она имеет 10 входных, 4 скрытых и 2 выходных нейрона. В общем случае сеть прямого распространения с m входами, h_1 нейронами первого скрытого слоя, h_2 нейронами второго скрытого слоя и q нейронами выходного слоя называется сетью $m - h_1 - h_2 - q$. Нейронная сеть, показанная на рис. 4, считается полносвязной (fully connected) в том смысле, что все узлы каждого конкретного слоя соединены со всеми узлами смежных слоев. Если некоторые из синаптических связей отсутствуют, такая сеть называется неполносвязной (partially connected).

Множество пар сигналов вход-выход, каждая из которых состоит из входного сигнала и соответствующего ему желаемого выхода, называют обучающими данными (training data) или обучающей выборкой (training sample). Для примера рассмотрим задачу распознавания цифр (digit recognition problem). В этой задаче входной сигнал (изображение) представляет собой матрицу, состоящую из черных и белых точек. Каждое изображение представляет одну из десяти рукописных цифр на белом фоне. Желаемым откликом сети является конкретная цифра, изображение которой подается в качестве входного сигнала. Обычно обучающая выборка состоит из большого числа рукописных цифр, что отражает ситуацию, которая может возникнуть в реальном мире. При наличии такого набора примеров нейронная сеть создается следующим образом.

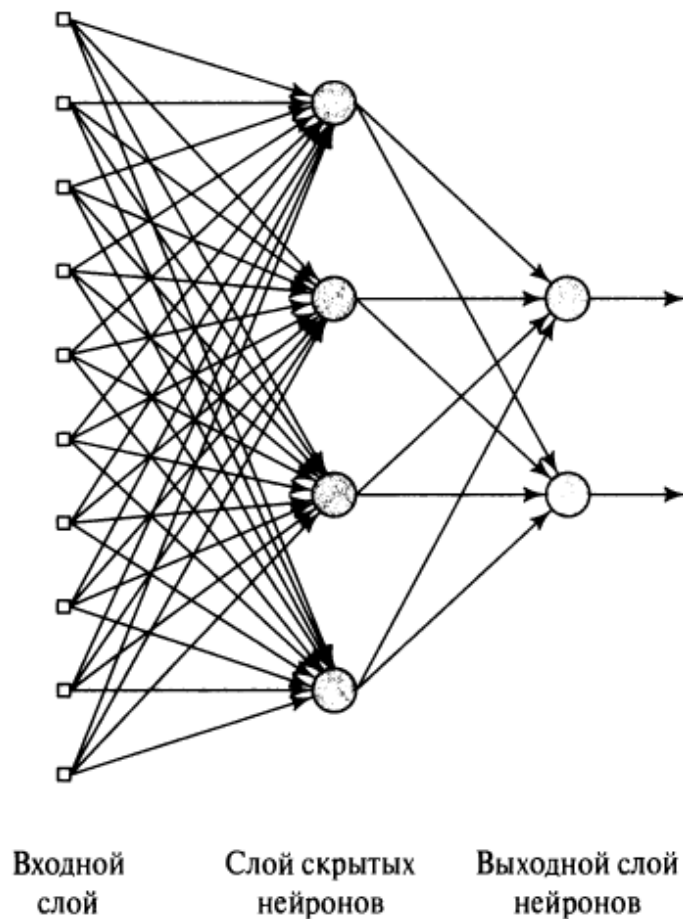


Рис. 4: Полносвязная сеть прямого распространения с одним скрытым и одним выходным слоем

- Во-первых, выбирается соответствующая нейросетевая архитектура, в которой размер входного слоя соответствует количеству пикселей на рисунке, а в выходном слое содержится десять нейронов, соответствующих цифрам. После этого выполняется настройка весовых коэффициентов сети на основе обучающего множества. Этот режим работы сети называется обучением.
- Во-вторых, эффективность обучения сети проверяется (тестируется) на множестве примеров, отличных от использованных при обучении. При этом на вход сети подается изображение, для которого известен целевой выход сети. Эффективность обучения сети проверяется путем сравнения результатов распознавания с реальными цифрами. Этот этап работы нейронной сети называют обобщением

(generalization) (данный термин взят из психологии).

Здесь и кроется фундаментальное отличие между созданием нейронной сети и разработкой классических методов обработки информации для задач классификации. В последнем случае мы в первую очередь формулируем математическую модель исследуемой среды, верифицируем ее на реальных данных, а затем разрабатываем классификатор на основе этой модели. Создание нейронной сети основывается непосредственно на реальных данных, которые говорят сами за себя. Таким образом, нейронные сети не только реализуют полноценную модель среды, но и обеспечивают обработку данных.

Набор данных, используемый для обучения сети, должен содержать как положительные, так и отрицательные примеры. Например, в задаче пассивной эхо-локации положительные примеры включают сигналы, отраженные от интересующего объекта (например, подводной лодки). Однако в реальной среде на отклик радара влияют и морские объекты, случайно попавшие в зону сигнала. Чтобы понизить вероятность неверной трактовки сигнала, в множество примеров добавляют сигналы, полученные при отсутствии искомого объекта.

В нейронной сети заданной архитектуры знания об окружающей среде представляются множеством свободных параметров (т.е. синаптических весов и порогов) сети. Такая форма представления знаний соответствует самой природе нейронных сетей. Именно в ней кроется ключ эффективности нейросетевых моделей.

Самым важным свойством нейронных сетей является их способность обучаться (learn) на основе данных окружающей среды и в результате обучения повышать свою производительность. Повышение производительности происходит со временем в соответствии с определенными правилами. Обучение нейронной сети происходит посредством интерактивного процес-

са корректировки синаптических весов и порогов. В идеальном случае нейронная сеть получает знания об окружающей среде на каждой итерации процесса обучения.

С понятием обучения ассоциируется довольно много видов деятельности, поэтому сложно дать этому процессу однозначное определение. Более того, процесс обучения зависит от точки зрения на него. Именно это делает практически невозможным появление какого-либо точного определения этого понятия. Например, процесс обучения с точки зрения психолога в корне отличается от обучения с точки зрения школьного учителя. Со своей точки зрения (с позиций нейронной сети) мы можем использовать следующее определение, приведенное в [27].

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.

Это определение процесса обучения предполагает следующую последовательность событий.

1. В нейронную сеть поступают стимулы из внешней среды.
2. В результате этого изменяются свободные параметры нейронной сети.
3. После изменения внутренней структуры нейронная сеть отвечает на возбуждения уже иным образом.

Вышеуказанный список четких правил решения проблемы обучения называется алгоритмом обучения (learning algorithm). Несложно догадаться, что не существует универсального алгоритма обучения, подходящего для всех архитектур нейронных сетей. Существует лишь набор средств, представленный множеством алгоритмов обучения, каждый из которых имеет свои достоинства. Алгоритмы обучения отличаются друг от дру-

га способом настройки синаптических весов нейронов. Еще одной отличительной характеристикой является способ связи обучаемой нейросети с внешним миром. В этом контексте говорят о парадигме обучения (learning paradigm), связанной с моделью окружающей среды, в которой функционирует данная нейронная сеть.

Нейронные сети (NN) широко упоминаются в литературе для решения нелинейных задач. Внимание к NN связано с возможностью моделировать сложные отношения в наборе данных, в то время как обычные статистические модели могут иметь трудности с таким моделированием. Нейронная сеть содержит, по меньшей мере, три (3) слоя с нейронами, распределенных по слоям. Как правило, скрытые и выходные нейроны имеют функции активации. Каждый нейрон в скрытом и выходном слоях помимо смещения связан с порогом и весами. Данные передаются в NN через входные нейроны и разворачиваются в сетевую структуру. Вычисление производится скрытыми и выходными нейронами, в то время как входные нейроны принимают только входные данные без вычислений. Веса и смещения итеративно изменяются во время обучения, чтобы свести к минимуму функцию ошибки, например, среднеквадратичную ошибку (MSE) [28]. Наиболее широко распространенным алгоритмом обучения является алгоритм обратного распространения ошибки (BP), который, по сути, является градиентным спуском, минимизирующим функцию ошибки [29]. Вербос [30] популяризовал алгоритм среди исследовательского сообщества. Синаптический вес (W) в BPNN обновляется по формуле

$$W_{k+1} = W_k + \Delta W_k, \quad (4)$$

где k — терация; ΔW_k — изменение текущего веса, вычисляется как:

$$\Delta W_k = -\eta \frac{\partial e_k}{\partial W_k}, \quad (5)$$

где η – скорость обучения; $\frac{\partial e_k}{\partial W_k}$ – градиент функции ошибки, которую надо минимизировать. Градиентный спуск BP до сих пор применяется во многих NN приложениях, хотя больше не считается оптимальным и эффективным алгоритмом обучения [31]. Нейронная сеть была использована при прогнозировании энергии [32]. Нейронные сети широко обсуждаются в литературе, более детальная информация дана в [28], [31].

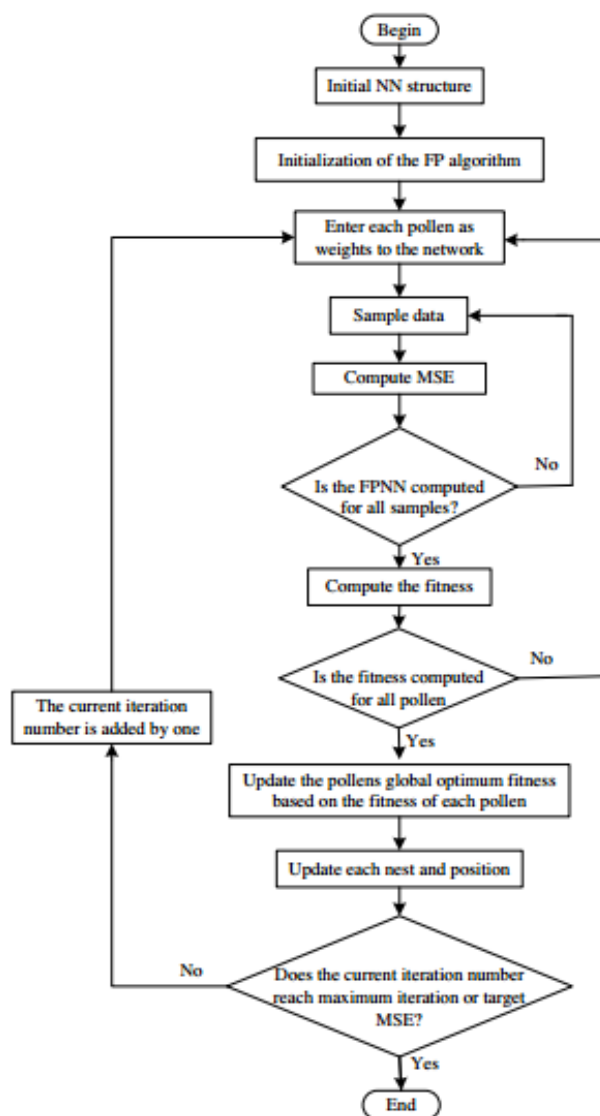


Рис. 5: Блок-схема FPNN

Блок-схема FPNN показана на Рис. 5. В FPNN алгоритме каждая пыльца представляет собой возможное решение(т.е. начальный вес и соответствующие смещения). Задача оптимизации веса и размера популяции представляют собой качество решения. На первой стадии лучшие началь-

ные веса и смещения инициализируются с помощью FP. Впоследствии, эти веса передаются в нейронную сеть. Веса в NN вычисляются и сравниваются с самым лучшим решением. На второй итерации FP обновляет веса с наилучшим возможным решением. FP продолжает искать лучшие веса пока не будет достигнута последняя итерация/эпоха или достигнут минимум целевой функции MSE. Ошибка вычисляется как

$$E = T_t - Y_t, \quad (6)$$

где Y_t — выход NN; T_t — фактические данные, когда представлен t -ый вход. Индекс производительности вычисляется как

$$V_F(x) = \frac{1}{2} \sum_{t=1}^R E^T E, \quad (7)$$

где $E = T_t - Y_t$, а R в уравнении 7 — общее количество входных примеров. MSE индекса производительности вычисляется как:

$$V_\mu(x) = \frac{\sum_{t=1}^N V_F(x)}{P_i} \quad (8)$$

Прогнозирование потребления нефти

В данной разделе алгоритм опыления цветка обеспечит обучение нейронной сети в модели прогнозирования потребления нефти странами ОПЕК для сравнения этого алгоритма по показателям скорости сходимости и точности прогнозирования. Далее этот алгоритм применяется для прогнозирования потребления нефти основными странами-импортерами российской нефти.

Набор данных о потреблении нефти странами ОПЕК

Данные о потреблении нефти странами ОПЕК в тысячах баррелей в день получены от Департамента Энергетики США, которые свободно доступны на официальном сайте организации [33]. Данные о потреблении нефти собираются для всех 12 стран ОПЕК (Алжир, Ангола, Эквадор, Иран, Ирак, Кувейт, Нигерия, Катар, Саудовская Аравия, ОАЭ, Венесуэла). Также вычисляется общее потребление нефти – это целевое значение прогнозирования. Данные ежегодного потребления, собранные с 1980 по 2012 год представлены на Рис. 6.

Набор данных нормирован для повышения точности и скорости сходимости [35], [36]. Хэир и др. [37] заявили, что для успешного прогноза требуется набор независимых положительно коррелирующих переменных. В результате, мы используем коэффициент корреляции чтобы изучить связь, которая существует в наборе данных, изображенной на Рис. 6. Коэффициенты корреляции представлены в таблице 1.

Таблица 1 показывает, что отрицательной корреляции нет. Отрицательная корреляция означает, что переменные движутся в противоположных направлениях, вследствие чего невозможно определить направление колебания потребления нефти. Однако, признаки, указанные в таблице 1,

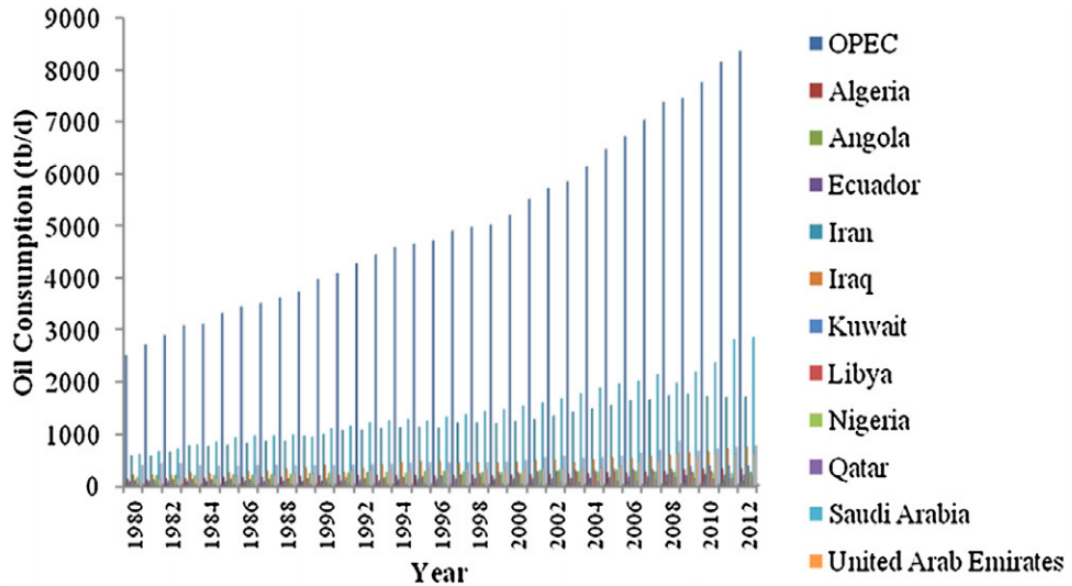


Рис. 6: Потребление нефти странами ОПЕК за период 1980–2012 и общее потребление

Страна	ОПЕК	Страна	ОПЕК
Алжир	0.976	Ливия	0.756
Ангола	0.940	Нигерия	0.580
Эквадор	0.976	Катар	0.937
Иран	0.983	Саудовская Аравия	0.988
Ирак	0.969	ОАЭ	0.968
Кувейт	0.952	Венесуэла	0.929

Таблица 1: Корреляция между общим потреблением нефти странами ОПЕК и каждой страной по отдельности

образуют значительные положительные значения между внутренним потреблением и общим потреблением нефти странами ОПЕК. В общем, они могут влиять на направление колебания, т.к. они движутся в том же направлении. Таким образом, потребление нефти каждой из стран ОПЕК является входами, а общее потребление – это целевое значение FPNN модели.

Характеристики эксперимента

В данном разделе описывается конфигурация ЭВМ, используемой для проведения эксперимента. С обоснованием объясняется функция, принятая для исследования. В настоящем исследовании для прогнозирования потребления нефти используется модель FPNN.

FPNN написана и реализована в Matlab2017a и C++ на компьютере со следующими характеристиками: Intel Core (TM) 2 Quad, CPU 2.33 GHz, RAM 2 GB, 64-битная операционная система. Все эксперименты проводятся на указанной машине. FPNN для запуска требует начальных настроек параметров. MSE используется как функция для измерения точности FPNN. MSE дает расчетную ошибку между фактическим потреблением нефти и прогнозируемым алгоритмом FPNN. Чем ближе MSE к нулю, тем выше точность. Если $MSE = 0$, то это совершенный прогноз. Выбор MSE обусловлен превосходством показателей (нормализация среднеквадратичной ошибки, среднеквадратичная ошибка, средняя абсолютная ошибка и т.д.) при оценке эффективности различных алгоритмов на том же наборе данных [38]. В наших экспериментах производительность FPNN сравнивается с другими основными биоалгоритмами, поэтому выбор MSE подходит для экспериментов.

Параметры контролируются с целью получения оптимального решения. Тем не менее, систематический способ изменения параметров является нерешенной проблемой [39]. Подобно другим метаэвристическим алгоритмам, FP чувствителен к начальным параметрам. В данном исследовании, параметры для FP устанавливаются равными $p = 0.85$, а количество гамет $n = 20$. NN также требует начальных параметров. Число входов и выходов нейронов равны 12 и 1 соответственно, так как ОПЕК состоит из 12 стран-участниц, а общее потребление нефти ОПЕК составляет одно значение. Количество скрытых слоев равно 1, в связи с тем, что одного скрытого

слоя достаточно, чтобы аппроксимировать любую функцию с произвольной точностью [38]. Функции активации в скрытом и выходном слоях являются нелинейными и линейными в соответствии с рекомендациями [40].

Количество нейронов скрытого слоя равно 7 в соответствии с [41]. Начальная структура нейронной сети (12–7–1 со смещением). Вес и смещение NN инициализируются алгоритмом опыления цветка. Имеем FPNN (12–7–1) с различными весами и смещением, которые инициализируются FR. Максимальное количество итераций $N = 1000$, а целевая ошибка устанавливается равной 0,0001. NN очень чувствительна к начальным значениям весов. Таким образом, чтобы дать значения весам нейронной сети, нейронные сети обучаются со значением веса алгоритма FR для каждой итерации. На каждой итерации NN запрашивает новые значения весов от FR. Чтобы гарантировать сходимость, останавливаем алгоритм при достижении целевой ошибки или при исчерпании максимального количества итераций.

FPNN запускается независимо 25 раз для каждого коэффициента разбиения данных, для того чтобы вычислить средние, лучшие и худшие результаты, т.к. метаэвристические алгоритмы недетерминированы [42]. С целью оценки эффективности FPNN, были выбраны GA, ABC, потому что они являются метаэвристическими алгоритмами, хорошо известные в литературе [43] и BP. Кроме того, ускоренный метод роя частиц (APSO) выбран вместо обычного метода роя частиц (PSO), потому что APSO имеет лучшую производительность [44]. Для прогнозирования потребления нефти используются NN, обученная ABC (ABCNN), GA (GANN), APSO (APSONN) и BP (BPNN). Считаем, что сравнение справедливо, т.к. предложенный метод FRNN сравнивается с эволюционными алгоритмами. Как правило, если NN обучается с помощью эволюционного алгоритма, справедливым сравнением является сравнение с NN, обученной другими эволюционными алгоритмами. Это упоминается в литературе [44], [45]. Таким образом, FR в

нашем исследовании сравнивается с GA, ABC и APPO нейронными сетями.

Настройки параметров алгоритмов сравнения:

- APPO: размер популяции $n = 20$, вероятность $p = 0.16$, $\gamma = 0.94$, $\alpha = 0.2$, $\beta = 0.5$, нижний и верхний диапазон = $[-5,5]$.
- GA: численность популяции $n = 20$, коэффициент скрещивания = 1, коэффициент мутации = 0,1, нижний и верхний диапазон = $[-5,5]$.
- ABC: размер колонии $n = 50$, максимальный цикл = 1000, размерность = 200, верхний диапазон = $[-5,5]$.

Результаты прогнозирования ABCNN, GANN, APSONN, и BPNN сравниваются с FPNN в обоих (обучающих и тестовых) наборах данных.

Сравнение работы алгоритмов

В этом разделе представлены результаты, полученные в результате экспериментов, описанных в предыдущем разделе. Для оценки производительности FPNN сравнивается с APSONN, ABCNN, GANN и BPNN. Произведено сравнение значений, предсказанных нейронной сетью FPNN и фактическим потреблением нефти странами ОПЕК.

Результаты работы предложенного алгоритма и алгоритмов сравнения представлены в таблицах 2, 3, 4, 5 в приложении. Первый столбец представляет собой коэффициент разбиения, используемый для оценки производительности алгоритмов. Разбиение данных влияет на производительность алгоритма [46]. В результате были использованы различные разбиения, чтобы обеспечить последовательное выполнение алгоритма и сохранение робастности, имея в виду, что поиск оптимальных решений является сложным из-за проблем, которые в основном сопровождается неопределенностью. Таким образом, акцент делается не только на оптимальное решение. Робастность решения также необходима в области инженерного проектирования и промышленности. Принятие решений в реальной жизни требует действий, которые являются оптимальными и надежными [47]. Второй столбец показывает алгоритмы с их соответствующими значениями производительности. Столбцы 3, 4 и 5 определяют средние, лучшие, и худшие значения времени сходимости в секундах и точности MSE, вычисленной в течение 25 независимых исполнений для каждого из алгоритмов.

Точность прогноза и время сходимости FPNN, описанные выше, сравнили с результатами ABCNN, GANN, APSONN, и BPNN на обучающих (таблица 2, 3) и тестовых (таблицы 4, 5) выборках. Результат алгоритма должен быть сравнен на обоих наборах данных, потому что необходимо учитывать более одного фактора и наиболее значимым является точность как в обучении, так и тестировании [48]. В таблицах 2, 3, 4, 5 ясно показано,

что FPNN предлагает многообещающие результаты (выделены жирным) в обучающем и тестирующем наборах данных. FPNN представляется лучше чем ABCNN, GANN, APSONN и BPNN, как в точности прогнозирования, так и времени сходимости. FP способен успешно избежать локальных минимумов, что повышает точность прогноза и скорость сходимости по сравнению с ABC, GA, APSO и BPNN. Есть два возможных объяснения вероятной причины, почему FPNN опережает ABCNN, GANN, APSONN и BPNN: (1) Способность FP исследовать очень большие пространства позволяет FP найти лучшие веса и смещения, следовательно, улучшить точность. (2) Способность FP последовательно и часто выбирать цветок того же вида помогает FA сходиться быстрее, чем ABC, GA, APSO и BPNN.

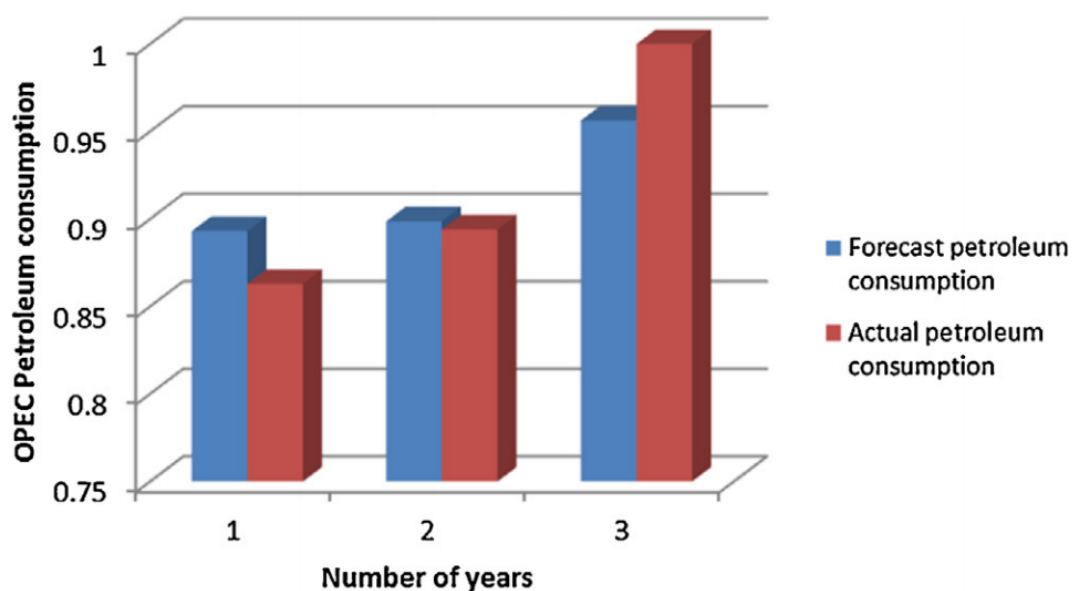


Рис. 7: FPNN прогнозирование потребления нефти странами ОПЕК за три года

Можно заметить из таблиц 2, 3, 4, 5, что обучение ABCNN и APSONN занимает очень много времени, чтобы сойтись к оптимальному решению по сравнению с GANN и BPNN. ABCNN и APSONN выполняются длительное время, вероятно, из-за медленной сходимости ABC и APSO. Производительность APSONN наиболее близко к FPNN, вероятно, из-за сходства между FP и APSO. Лучший алгоритм сходится к лучшему решению после

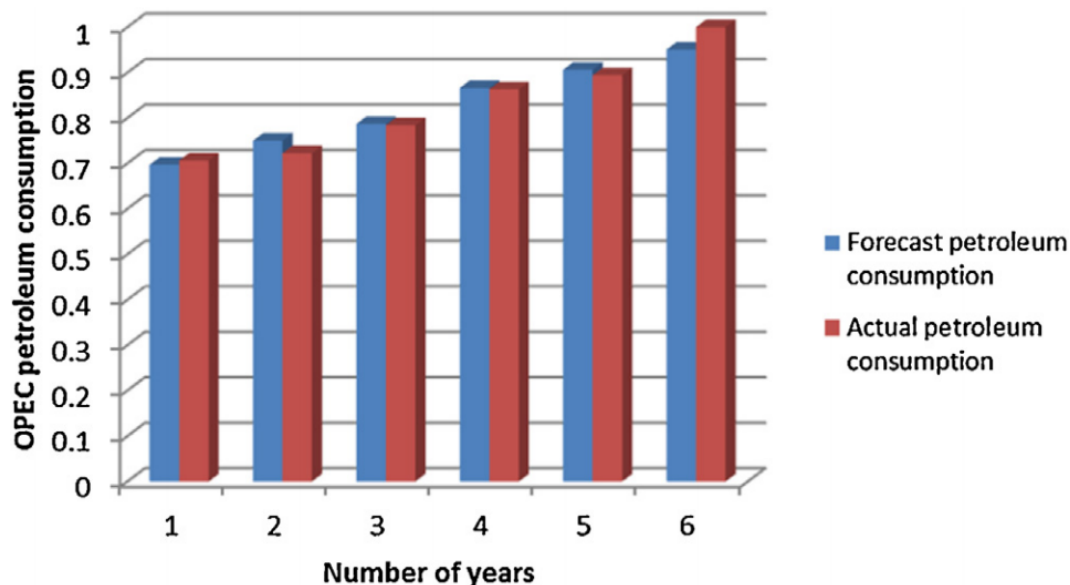


Рис. 8: FPNN прогнозирование потребления нефти странами ОПЕК за шесть лет

меньшего количества раз. Поэтому FPNN можно считать лучшим алгоритмом среди 5 сравниваемых. Кроме того, FPNN можно считать робастным, так как он способен демонстрировать одинаковую производительность во всех возможных соотношениях разбиения данных, а также поддерживать стабильную работу в течение экспериментов.

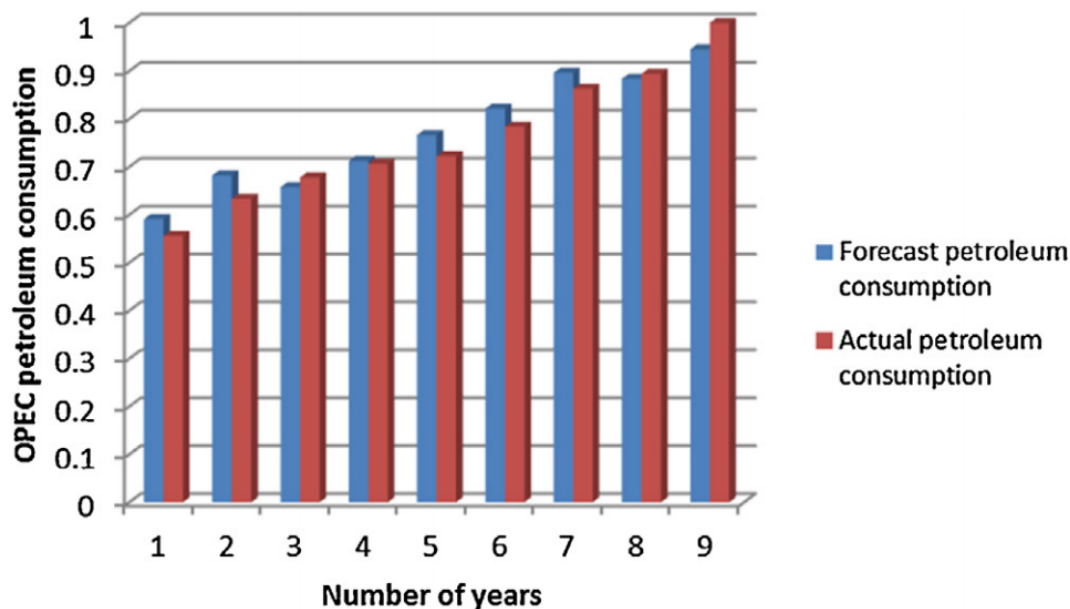


Рис. 9: FPNN прогнозирование потребления нефти странами ОПЕК за девять лет

Как правило, в литературе, предложенный алгоритм применяется

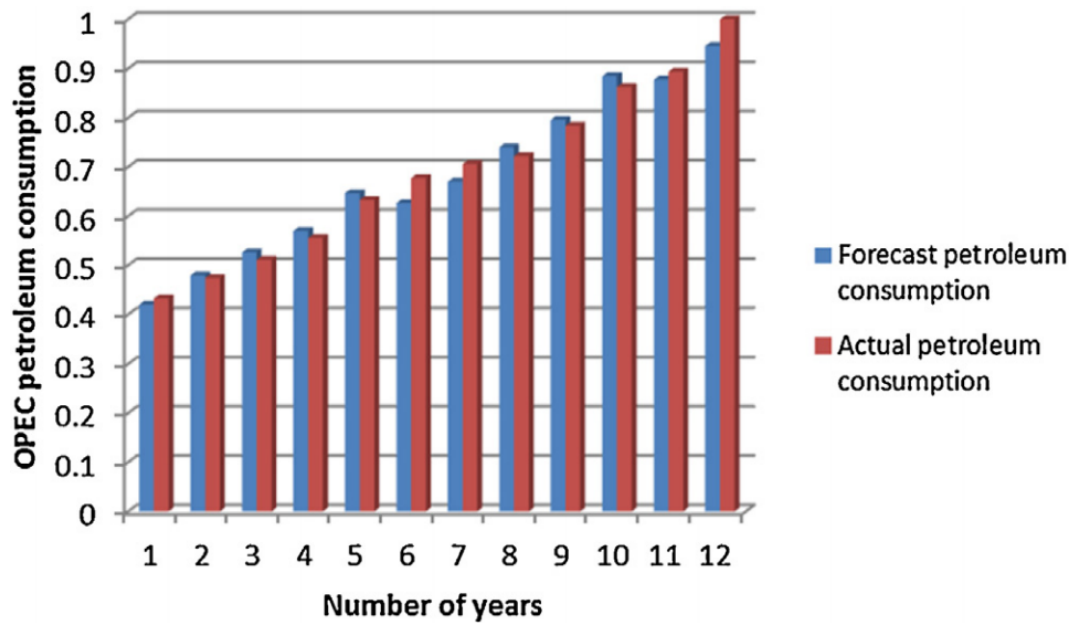


Рис. 10: FPNN прогнозирование потребления нефти странами ОПЕК за 12 лет

для прогнозирования фактических данных, основанных на исторических данных, с целью показать преимущества и эффективность алгоритма, как это сделано в [49], [50], [51]. Графики, изображенные на Рис. 7, 8, 9, 10, 11 показывают прогноз потребления нефти для различного количества лет. Потребление нефти странами ОПЕК, сделанное с помощью FPNN очень близко к реальным значениям. Это указывает на точность и надежность FPNN. Это означает, что FPNN является перспективным алгоритмом, который необходим для применения в реальных условиях.

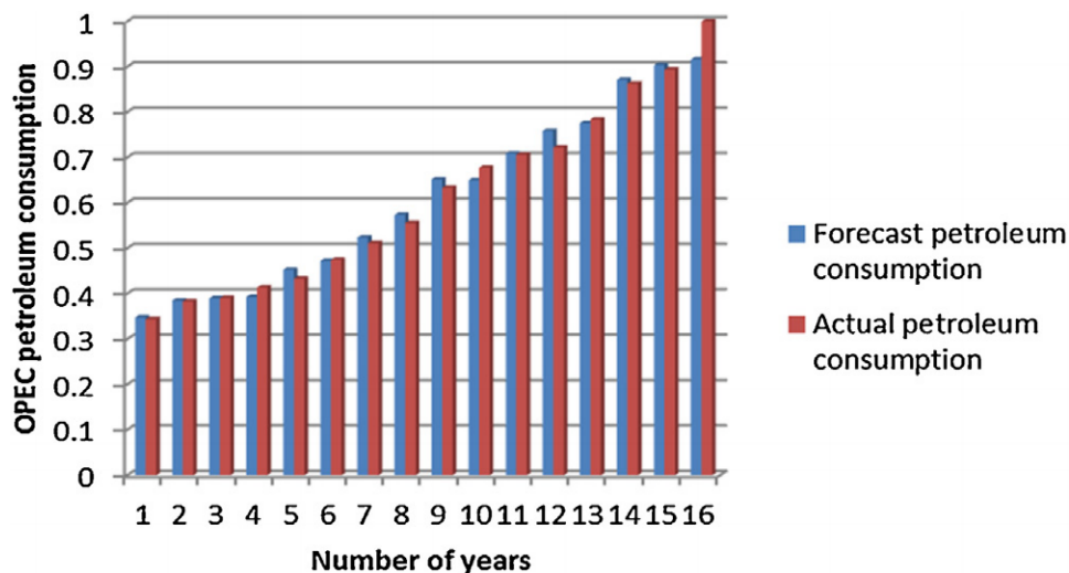


Рис. 11: FPNN прогнозирование потребления нефти странами ОПЕК за 16 лет

Прогнозирование потребления основными странами-импортерами российской нефти

В данном разделе рассматривается потребление крупнейшими странами-импортерами российской нефти за последние несколько лет. Импортеры нефти из России в 2015 году в процентном соотношении по странам указаны на Рис. 12, а их среднегодовое потребление в период с 1992–2015 гг. на Рис. 13.



Рис. 12: Крупнейшие импортёры нефти из России в 2015 году

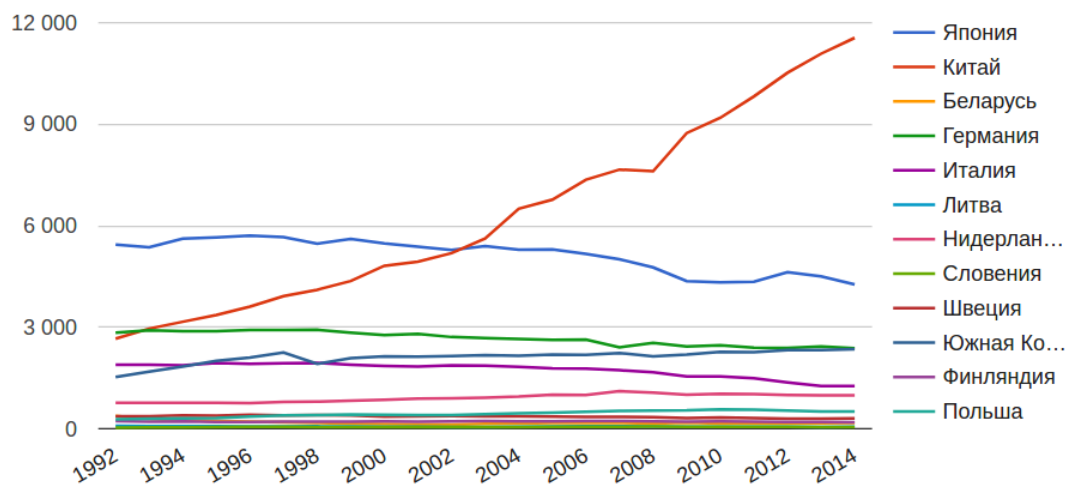


Рис. 13: Потребление нефти в период 1992–2015 гг., тыс.баррелей/день

Стоит отметить, что Китай, превратившийся в крупнейшую мировую экономику (по объему ВВП), с каждым годом импортирует все больше нефти, что не может не радовать Россию – одного из крупнейших экспортеров нефти и географического соседа Китая. Географическое расположение, а также ресурсная и технологическая взаимодополняемость сделали страны ЕС и Россию важнейшими торговыми партнерами, которых невозможно заменить, по крайней мере в краткосрочной перспективе.

ЕС по-прежнему крупнейший в мире регион по потреблению нефти, хотя спрос там снизился на 17% с 2005 года – до уровня ниже 12,5 млн баррелей в сутки (в 2015 году это снижение остановилось впервые за последние 10 лет). ЕС ожидает, что спрос продолжит снижаться на 0,5% в год в течение пяти лет. Тем не менее ЕС по-прежнему сильно зависит от импорта – в 2013 году было импортировано 83% потребляемой нефти.

Россия – крупнейший поставщик нефти в ЕС, на нее приходится около 29% общего импорта (на Норвегию – 12,6%, Нигерию – 9%, Саудовскую Аравию – 8,9%). Российская зависимость от европейского потребления нефти еще сильнее – ЕС покупает около 75% российского экспорта нефти, причем более 70% транспортируется по трубопроводам, а значит, не может без существенных затрат быть поставлено в другое место.

Снижение спроса на импортную нефть, европейские программы по-

вышениа эффективности и активная конкуренция со стороны США и стран Ближнего Востока – это очевидная угроза для российских экспортеров нефти. Ситуация становится все более сложной: начиная с 2018 года в России из-за нехватки инвестиций, нерациональной добычи и истощения дешевых месторождений ожидается постепенное снижение объема добычи нефти. По прогнозам, к 2035 году он может сократиться вдвое. Поэтому снижение продаж в Европе скорее будет результатом снижения предложения, чем сокращения спроса.

ЕС и Россия должны подготовиться к постепенной потере в товарообороте на этом рынке. Русские должны искать другие продукты, которые они смогут предложить ЕС, чтобы заменить падающие объемы поставок нефти, а европейские нефтеперерабатывающие заводы должны подготовиться к переходу с Urals на другие виды нефти.

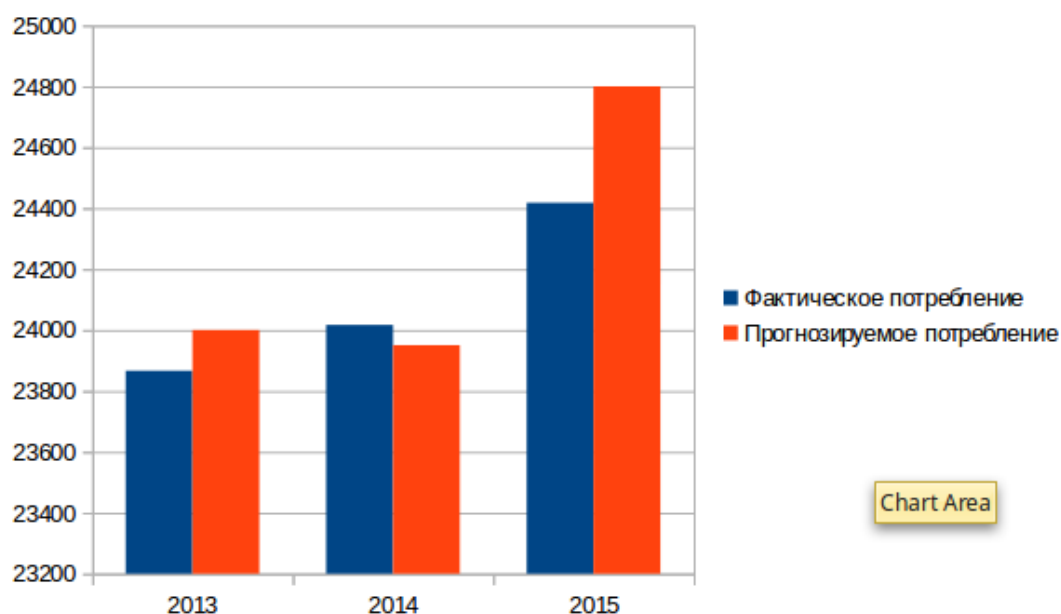


Рис. 14: FPNN прогнозирование потребления нефти крупнейшими странами-импортерами российской нефти за три года, тыс.баррелей/день

Для прогнозирования потребления нефти выберем модель, полученную в предыдущем разделе, а именно FPNN. Нейронная сеть состоит из трех слоев, один из которых является скрытым(12–7–1). Максимальное количество итераций $N = 1000$, а целевая ошибка устанавливается рав-

ной 0,0001. В прогнозе будет содержаться состояние об общем потреблении 12 крупнейшими странами-импортерами российской нефти. Прогнозируемые и фактические значения для трех(3) и шести(6) лет представлены на Рис. 14, 15.

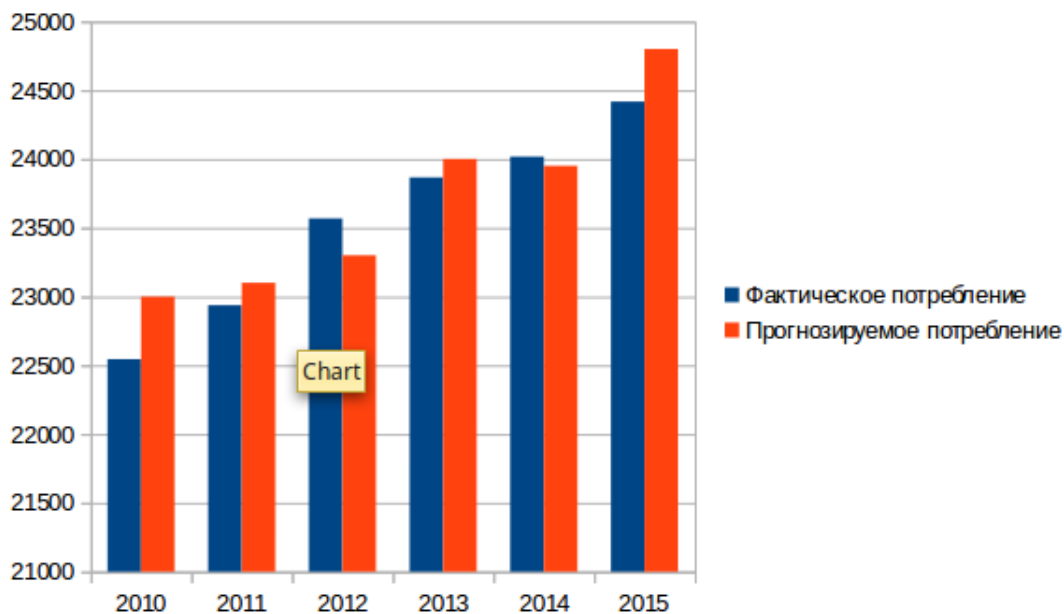


Рис. 15: FPNN прогнозирование потребления нефти крупнейшими странами-импортерами российской нефти за шесть лет, тыс.баррелей/день

Скорость сходимости и точность FPNN сравнимы с моделью из предыдущего раздела. Потребление нефти странами-импортерами российской нефти, сделанное с помощью FPNN близки к реальным значениям. Это указывает на надежность FPNN, как инструмента прогнозирования, что в свою очередь означает, что FPNN является перспективным алгоритмом, который применен в реальных условиях. В следующем разделе обсуждаются итоги проделанной работы и план дальнейшего исследования.

Заключение

В данном исследовании предлагается альтернативный подход к повышению точности прогнозирования потребления нефти и производительности такого прогнозирования. Применяемый в данной работе подход является одним из многих способов для надлежащего мониторинга и контроля потребления нефти. В данной работе NN обучается на основе алгоритма FP, вместо обычно используемых метаэвристических алгоритмов. Признанные метаэвристические алгоритмы, такие как GA, ABC, APSO и BP используются в работе для обучения NN с целью сравнения результатов. Было установлено, что метод, предложенный в исследовании работает лучше, чем GA, ABC, APSO и BP с точки зрения скорости сходимости, точности прогнозов и робастности.

Модель FPNN, показанная в исследовании не предназначена в качестве замены экспертов или существующих моделей. Тем не менее, FPNN может служить дополнением для других моделей, а также в качестве дополнительной модели для человека. FPNN имеет возможность прогнозировать потребление нефти, при условии, что потребление нефти известна.

Нейронная сеть написана на языке C++ стандарта C++11. В настройках нейронной сети можно выбирать количество слоев, функцию активации в каждом слое, количество нейронов в каждом слое. Принципы работы FPNN-метода представлены в виде псевдокода. FP-алгоритм также представлен в виде псевдокода, а также есть его реализация в Matlab2017a.

Предложенный подход прогнозирования потребления нефти может помочь в управлении эффективной разработки нефтяных компаний, адаптации существующей политики и осуществления политики в области потребления нефти. Это возможно, поскольку точное прогнозирование потребления энергии влияет на вложение инвестиций, качество окружающей среды, анализ доходов и управления маркетинговых исследований [52].

Рост экономики связан с увеличением потребления нефти, что означает, что чем больше экономика растет, тем больше нефти потребляется. Это может увеличить выбросы двуокиси углерода, способствуя тем самым глобальному потеплению. Нефтяные страны могут использовать предложенный подход, для точного измерения оценки потребления нефти; таким образом могут быть сформулированы необходимые команды, для тщательного регулирования потребления. Точность оценок в значительной степени зависит от точности модели, используемой для прогнозирования нефти.

Мониторинг колебания потребления нефти возможен с помощью прогнозирования значений потребления. Различные организации, связанные с нефтяной деятельностью, такие как Международное Энергетическое Агентство, Организация экономического сотрудничества и развития, Управление энергетической информации могут посчитать данную модель полезной для мониторинга потребления нефти. FPNN имеет хорошие шансы стать программным обеспечением с графическим пользовательским интерфейсом, который может служить в качестве важного инструмента в прогнозировании потребления нефти для вышеназванных организаций.

FPNN-метод, предложенный в исследовании, может быть полезным для администрации нефтяных компаний для эффективного мониторинга, контроля, разработки, чтобы адаптировать существующую политику и осуществлять будущую политику компании. В будущем, рекомендуется гибридизировать алгоритм опыления цветка и алгоритм поиска кукушки для обучения рекуррентной нейронной сети. Этот метод предложил Янг в одной из своих статей. Исходный код программы можно получить по <https://github.com/ishveik/forecasting-petroleum-consumption>

Литература

- [1] <http://www.bp.com/content/dam/bp/excel/energy-economics/statistical-review-2016/bp-statistical-review-of-world-energy-2016-workbook.xlsx>
- [2] X.S. Yang, Flower pollination algorithm for global optimization, *Unconventional Computation and Natural Computation* (2012) 240–249.
- [3] R. Wang, Y. Zhou, Flower pollination algorithm with dimension by dimension improvement, *Math. Probl. Eng.* (2014), <http://dx.doi.org/10.1155/2014/481791>, 9 pages.
- [4] H. Ma, Y. Wu, A grey forecasting model based on BP neural network for crude oil production and consumption in China, in: *Proceedings of IEEE Computer Society Third International Symposium on Information Processing, Qingdao, China, 2010*, pp. 105–109.
- [5] S.P. Niculescu, Artificial neural networks and genetic algorithms in QSAR, *J.Mol. Struct. THEOCHEM* 622 (1) (2003) 71–83.
- [6] L. Suganthi, A.A. Samuel, Energy models for demand forecasting—A review, *Renew. Sustain. Energy Rev.* 16 (2) (2012) 1223–1240.
- [7] B.N. Behmiri, J.R.P. Manso, How crude oil consumption impacts on economic growth of Sub-Saharan Africa, *Energy* 54 (2013) 74–83.
- [8] European Statistics. <http://ec.europa.eu/eurostat>, 2015.

- [9] Yang, X. S., Nature-Inspired Metaheuristic Algorithms, Luniver Press, (2008).
- [10] Yang, X. S., Engineering Optimization: An Introduction with Metaheuristic Applications, Wiley (2010).
- [11] Holland, J., Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, (1975).
- [12] Kennedy J. and Eberhart R. C.: Particle swarm optimization. Proc. of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942-1948 (1995).
- [13] Kennedy J., Eberhart R., Shi Y.: Swarm intelligence, Academic Press, (2001).
- [14] Yang, X. S., (2010c). A new metaheuristic bat-inspired algorithm, in: Nature-Inspired Cooperative Strategies for Optimization (NICSO 2010) (Eds. Gonzalez J. R. et al.), Springer, SCI 284, pp. 65–74.
- [15] Yang, X. S., Firefly algorithm, stochastic test functions and design optimisation, Int. J. Bio-Inspired Computation, 2(2), 78-84 (2010).
- [16] Walker, M., How flowers conquered the world, BBC Earth News, 10 July 2009. http://news.bbc.co.uk/earth/hi/earth_news/newsid_8143000/8143095.stm
- [17] Oily Fossils provide clues to the evolution of flowers, Science Daily, 5 April 2001. <http://www.sciencedaily.com/releases/2001/04/010403071438.htm>
- [18] Glover, B. J., Understanding Flowers and Flowering: An Integrated Approach, Oxford University Press, (2007).

- [19] Chittka, L., Thomson, J. D., and Waser, N. M., Flower constancy, insect psychology, and plant evolution, *Naturwissenschaften*, 86, 361-177 (1999).
- [20] Waser, N.M., Flower constancy: definition, cause and measurement. *The American Naturalist*, 127(5), 596-603 (1986)
- [21] Pavlyukevich I., Lévy flights, non-local search and simulated annealing, *J. Computational Physics*, 226, 1830-1844 (2007).
- [22] Reynolds A. M. and Frye M. A., Free-flight odor tracking in *Drosophila* is consistent with an optimal intermittent scale-free search, *PLoS One*, 2, e354 (2007)
- [23] Aleksander I. and H. Morton. *An Introduction to Neural Computing*, London: Chapman and Hall, 1990.
- [24] Haykin S. "Neural networks expand SP's horizons *IEEE Signal Processing Magazine*, 1996, vol. 13, no. 2, p. 24-29.
- [25] Widrow B. and S.D. Stearns. *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1985.
- [26] Churchland PS. and T.J. Sejnowski. *The Computational Brain*, Cambridge, MA: MIT Press, 1992.
- [27] Mendel J.M. and R.W. McLaren. "Reinforcement-learning control and pattern recognition systems in *Adaptive, Learning and Pattern Recognition Systems: Theory and Applications*, vol. 66, J.M. Mendel and K.S. Fu, eds., 1970, p. 287-318, New York: Academic Press.
- [28] S. Haykin, *Neural Networks*, 2nd ed., Prentice Hall, New Jersey, 1999.
- [29] D.E. Rumelhart, J.L. McClelland, *Parallel Distributed Processing*

Explorations in the Theory of Cognition, vol. 1, MIT Press, Cambridge, MA, 1986.

- [30] P. Werbos, *The Roots of the Backpropagation: From Ordered Derivatives to Neural Networks and Political Fore-Casting*, John Wiley and Sons, Inc, New York, 1993.
- [31] Y.H. Zweiri, J.F. Whidborne, L.D. Sceviratne, A three-term backpropagation algorithm, *Neurocomputing* 50 (2003) 305–318.
- [32] H.B. Azad, S. Mekhilef, V.G. Ganapathy, Long-term wind speed forecasting and general pattern recognition using neural networks, *IEEE Trans. Sustain. Energy* 5 (2) (2014) 546–553
- [33] Energy Information Administration of the US Department of Energy, 2015. [Online]. Available: <http://www.eia.gov/>.
- [34] M.O. Adetutu, Energy efficiency and capital-energy substitutability: evidence from four OPEC countries, *Appl. Energy* 119 (2014) 363–370
- [35] C. Quek, M. Pasquier, N. Kumar, Novel recurrent neural network-based prediction system for option trading and hedging, *Appl. Intell.* 29 (2008) 138–151.
- [36] K. Movagharnejad, B. Mehdizadeh, M. Banihashemi, M.K. Sheikhi, Forecasting the differences between various commercial oil prices in the Persian Gulf region by neural networks, *Energy* 36 (2011) 3979–3984
- [37] F.J. Hair, W.C. Black, J.B. Babin, R.E. Anderson, *Multivariate Data Analysis*, Pearson Prentice Hall, New Jersey, 2010.
- [38] G.Z. Peter, B.E. Patuwo, M.Y. Hu, A simulation study of artificial neural networks for nonlinear time-series forecasting, *Comput. Oper. Res.* 28 (2001) 381–396

- [39] X.-S. Yang, Cuckoo search and firefly algorithm: overview and analysis, in: X.S.I. Yang (Ed.), Cuckoo Search and Firefly Algorithm, Springer International Publishing, 2014, pp. 1–26.
- [40] M.H. Beale, M.T. Hagan, H.B. Demuth, Neural Network Toolbox™ User’s Guide, The MathWorks, Inc., Natick, 2013.
- [41] H. Chiroma, S. Abdul-Kareem, S.A. Muaz, A. Khan, E.N. Sari, T. Herawan, Neural network intelligent learning algorithm for inter-related energy products applications, in: Advances in Swarm Intelligence, Springer International Publishing, 2014, pp. 284–293.
- [42] M. Subotic, M. Tuba, N. Bacanin, D. Simian, Parallelized cuckoo search algorithm for unconstrained optimization, in: Proceedings of the 5th WSEAS Congress on Applied Computing Conference, and Proceedings of the 1st International Conference on Biologically Inspired Computation, World Scientific and Engineering Academy and Society (WSEAS), 2012, pp. 151–156
- [43] P. Civicioglu, E. Besdok, A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms, *Artif. Intell. Rev.* 39 (4) (2013) 315–346.
- [44] X.-S. Yang, S. Deb, S. Fong, Accelerated particle swarm optimization and support vector machine for business optimization and applications, in: Networked Digital Technologies, Springer, Berlin, Heidelberg, 2011, pp.53–66.
- [45] D. Karaboga, B. Akay, C. Ozturk, Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks, in: Modeling Decisions for Artificial Intelligence, Springer, Berlin, Heidelberg, 2007, pp. 318–329.

- [46] L. Yu, W. Shouyang, K.K. Lai, A neural-network-based nonlinear metamodeling approach to financial time series forecasting, *Appl. Soft Comput.* 9 (2009) 563–574.
- [47] X.-S. Yang, S. Deb, Cuckoo search: recent advances and applications, *Neural Comput. Appl.* 24 (1) (2014) 169–174.
- [48] Y. Jin, A comprehensive survey of fitness approximation in evolutionary computation, *Soft Comput.* 9 (2005) 3–12.
- [49] L. Yu, W. Dai, L. Tang, J. Wu, A hybrid grid-GA-based LSSVR learning paradigm for crude oil price forecasting, *Neural Comput. Appl.* (2015) 1–23, <http://dx.doi.org/10.1007/s00521-015-1999-4>.
- [50] A. Sanz-Garcia, J. Fernandez-Ceniceros, F. Antonanzas-Torres, A.V. Pernia-Espinoza, F.J. Martinez-de-Pison, GA-PARSIMONY: A GA-SVR approach with feature selection and parameter optimization to obtain parsimonious solutions for predicting temperature settings in a continuous annealing furnace, *Appl. Soft Comput.* 35 (2015) 13–28
- [51] S. Mollaiy-Berneti, Optimal design of adaptive neuro-fuzzy inference system using genetic algorithm for electricity demand forecasting in Iranian industry, *Soft Comput.* (2015) 1–10, <http://dx.doi.org/10.1007/s00500-015-1777-3>.
- [52] L. Ekonomou, Greek long-term energy consumption prediction using artificial neural networks, *Energy* 35 (2) (2010) 512–517.
- [53] D. Gately, N. Al-Yousef, H.M. Al-Sheikh, The rapid growth of OPEC's domestic oil consumption, *Energy Policy* 62 (2013) 844–859.

Приложение

Algorithm 1 FP algorithm

- 1: Задача \min или $\max f(x)$, $x = (x_1, x_2, \dots, x_d)$
 - 2: Инициализация начальной популяции n со случайными решениями
 - 3: Найти наилучшее решение g_* в начальной популяции
 - 4: Задать вероятность $p \in [0, 1]$
 - 5: **while** ($t < maxGen$) **do**
 - 6: **for** $1 : n$ **do**
 - 7: **if** $rand < p$ **then**
 - 8: Глобальное опыление $x_i^{t+1} = x_i^t + L(g_* - x_i^t)$
 - 9: **else**
 - 10: Определим $\epsilon \in [0, 1]$
 - 11: Случайным образом выбираем j и k
 - 12: Выполняем локальное опыление $x_i^{t+1} = x_i^t + \epsilon(x_j^t - x_k^t)$
 - 13: Вычисляем новые решения
 - 14: Если они лучше, то обновляем их в популяции
-

Algorithm 2 FPNN algorithm

```
1: Инициализация FP и NN структуры
2: Находим лучшее решение  $g_*$  в текущей популяции
3: Задать вероятность  $p \in [0, 1]$ 
4: Загружаем обучающую выборку
5: while  $MSE <$ 
6:   критерий остановки do
7:   Передаем веса в NN
8:   Вычисляем ошибку по формуле 6
9:   for  $1 : n$  do
10:    if  $rand < p$  then
11:      Глобальное опыление  $x_i^{t+1} = x_i^t + L(g_* - x_i^t)$ 
12:    else
13:      Определим  $\epsilon \in [0, 1]$ 
14:      Случайным образом выбираем  $j$  и  $k$  среди всех решений
15:      Выполняем локальное опыление  $x_i^{t+1} = x_i^t + \epsilon(x_j^t - x_k^t)$ 
16:    Строим новое решение
17:    Вычисляем новые решения
18:    Если они лучше, то обновляем их в популяции
19:    if  $X_j > X_i$  then
20:       $x_i \leftarrow x_j$ 
21:       $X_i \leftarrow X_j$ 
22:  Находим лучшее решение  $g_*$ 
```

```

% ----- %
% Flower pollination algorithm (FPA), or flower algorithm %
% Пример применения FPA для вычисления минимума функции Розенброка %
% ----- %

function [best ,fmin ,N_iter]=fpa_demo(para)
% Параметры по умолчанию
if nargin < 1,
    para=[20 0.85];
end

n=para(1);          % Размер популяции
p=para(2);          % вероятность выбора между опылениями

% Iteration parameters
N_iter=1000;        % Общее количество итераций

% Размерность данных
d=3;
Lb=-2*ones(1,d);
Ub=2*ones(1,d);

% Инициализация популяции/решений
for i=1:n,
    Sol(i,:)=Lb+(Ub-Lb).*rand(1,d);
    Fitness(i)=Fun(Sol(i,:));
end

% Найти текущее лучшее решение
[fmin ,I]=min(Fitness);
best=Sol(I,:);
S=Sol;

% Начало итераций — Flower Algorithm
for t=1:N_iter,
    for i=1:n,
        % Пыльца переносится насекомыми
        % и может переместиться на большое расстояние
        % L — Levy flights
        % Формула:  $x_{i,t+1}=x_{i,t}+L(x_{i,t}-gbest)$ 
        if rand>p,
            %% L=rand;
            L=Levy(d);
            dS=L.*(Sol(i,:)-best);
            S(i,:)=Sol(i,:)+dS;

            % Проверка на выход за границы/пределы
            S(i,:)=simplebounds(S(i,:),Lb,Ub);
        end
    end
end

```

```

% Иначе локальное опыление соседних цветов
else
    epsilon=rand;
    % Находим случайные цветы по соседству
    JK=randperm(n);
    % Если цветы одного и того же или похожего вида, то
    % они могут быть опылены, иначе, бездействие.
    % Формула:  $x_i^{t+1} + \text{epsilon} * (x_j^t - x_k^t)$ 
    S(i,:)=S(i,:)+epsilon*(Sol(JK(1),:)-Sol(JK(2),:));
    % Проверка на пределы/границы
    S(i,:)=simplebounds(S(i,:),Lb,Ub);
end

% Вычисление новых решений
Fnew=Fun(S(i,:));
% Если найдены лучшие решения, обновляем их
if (Fnew<=Fitness(i)),
    Sol(i,:)=S(i,:);
    Fitness(i)=Fnew;
end

% Обновляем текущее лучшее глобальное решение
if Fnew<=fmin,
    best=S(i,:);
    fmin=Fnew;
end
end
% Показывать результаты каждые 100 итераций
if round(t/100)==t/100,
    best
    fmin
end

end

% Применение простых ограничений
function s=simplebounds(s,Lb,Ub)
    % Нижняя граница
    ns_tmp=s;
    I=ns_tmp<Lb;
    ns_tmp(I)=Lb(I);

    % Верхняя граница
    J=ns_tmp>Ub;
    ns_tmp(J)=Ub(J);
    % Обновление
    s=ns_tmp;

```

```

% Levy flight
function L=Levy(d)
% Для деталей смотрите главу 11 книги:
% Xin-She Yang, Nature-Inspired Optimization Algorithms, Elsevier, (2014).
beta=3/2;
sigma=(gamma(1+beta)*sin(pi*beta/2)/(gamma((1+beta)/2)*...
beta*2^((beta-1)/2)))^(1/beta);
    u=randn(1,d)*sigma;
    v=randn(1,d);
    step=u./abs(v).^(1/beta);
L=0.01*step;

% Функция_Розенброка
function z=Fun(u)
z=(1-u(1))^2+100*(u(2)-u(1)^2)^2+100*(u(3)-u(2)^2)^2;

```

Разбиение	Алгоритм	Среднее	Лучшее	Худшее
90–10	FPNN	17.318478781	5.3441124	25.4477997
	GANN	21.599760985	5.6450988	32.4089472
	ABCNN	1078.348204784	245.2569203	1644.0717490
	APSONN	1147.936081480	1137.4864590	1160.9948420
	BPNN	71.13991031	70.1956293	72.0825834
80–20	FPNN	24.408422456	4.3132490	38.1623239
	GANN	21.853397747	5.5581175	32.9160964
	ABCNN	1098.291276980	240.1563437	1681.8133600
	APSONN	622.424642788	614.5872761	630.6467917
	BPNN	67.15956576	66.2133564	68.0796050
70–30	FPNN	23.729440302	4.4629256	36.9146311
	GANN	29.823770399	5.5102011	46.2545622
	ABCNN	1183.652207924	240.8519360	1877.0641990
	APSONN	627.689326384	620.1968321	635.1368825
	BPNN	68.34625147	67.4067120	69.2892311
60–40	FPNN	22.224060973	4.2621414	34.4074315
	GANN	23.541494887	5.6191962	35.7152157
	ABCNN	1120.666526924	242.0332384	1873.4761260
	APSONN	679.589133940	671.8437457	687.3790149
	BPNN	68.42942644	67.4332304	69.3962184
50–50	FPNN	17.607291640	4.2019747	26.6877923
	GANN	23.816470953	5.5257098	36.2400019
	ABCNN	1198.852470176	240.0067170	1873.4761260
	APSONN	752.682990404	744.8411809	760.8418091
	BPNN	78.95930054	77.8704024	80.1682141

Таблица 2: Время сходимости(сек) при обучении

Разбиение	Алгоритм	Среднее	Лучшее	Худшее
90–10	FPNN	0.000397234	0.0002137	0.0006857
	GANN	0.009676973	0.0045722	0.0141054
	ABCNN	0.002709705	0.0012946	0.0057576
	APSONN	0.003671513	0.0015747	0.0054657
	BPNN	0.001467548	0.0006741	0.0098920
80–20	FPNN	0.000271476	0.0001299	0.0003811
	GANN	0.043583819	0.0032442	0.0689826
	ABCNN	0.006194090	0.0016712	0.0099670
	APSONN	0.008613166	0.0080576	0.0090576
	BPNN	0.114863750	0.0619405	0.1739274
70–30	FPNN	0.003156222	0.0001577	0.0061697
	GANN	0.043021778	0.0064269	0.0736905
	ABCNN	0.004191528	0.0006319	0.0099670
	APSONN	0.003712283	0.0018223	0.0056822
	BPNN	0.149054708	0.0656022	0.2049988
60–40	FPNN	0.000207407	0.0001246	0.0002454
	GANN	0.055158568	0.0046934	0.0987970
	ABCNN	0.002105742	0.0006641	0.0057676
	APSONN	0.000834024	0.0006990	0.0009800
	BPNN	0.116267763	0.0635650	0.2016471
50–50	FPNN	0.001296049	0.0001615	0.0023688
	GANN	0.064654815	0.0030837	0.1123689
	ABCNN	0.003715591	0.0002860	0.0099280
	APSONN	0.000499143	0.0002328	0.0008333
	BPNN	0.157062536	0.0927028	0.2485662

Таблица 3: Точность обучения (MSE) в обучающей выборке

Разбиение	Алгоритм	Среднее	Лучшее	Худшее
90–10	FPNN	16.374652086	4.2851508	24.5865379
	GANN	21.546019869	5.5007970	32.4436726
	ABCNN	1083.564687316	244.6992528	1654.8122530
	APSONN	596.943101968	589.8458523	604.0199192
	BPNN	67.33254157	66.3757263	68.2773495
80–20	FPNN	17.586660387	4.3132490	26.6075585
	GANN	20.971776450	5.4818216	31.4890396
	ABCNN	1216.911415512	239.0465134	1880.4877670
	APSONN	842.041400112	825.7943131	855.6750413
	BPNN	67.84842447	66.7092589	68.9203720
70–30	FPNN	22.392703036	4.2923900	34.6760238
	GANN	41.764004765	5.4640657	66.4537831
	ABCNN	1147.526381320	243.4676544	1868.1281420
	APSONN	744.068250620	736.8487364	751.2289768
	BPNN	75.20721331	74.2544014	76.1364554
60–40	FPNN	22.234225182	4.3096664	34.3950287
	GANN	24.202904022	5.6057096	36.8271391
	ABCNN	1140.213973352	247.3804678	1877.0641990
	APSONN	734.756476676	726.1853597	744.0882833
	BPNN	70.92859406	69.8426321	72.0394524
50–50	FPNN	18.987609346	4.2095121	29.0942405
	GANN	22.312732609	5.5565065	33.6778900
	ABCNN	1084.688633140	243.1838017	1656.3609250
	APSONN	587.010445164	579.9539146	594.0554508
	BPNN	94.20479797	93.2177303	95.2055563

Таблица 4: Время сходимости(сек) тестовой выборки

Разбиение	Алгоритм	Среднее	Лучшее	Худшее
90–10	FPNN	0.000409957	0.0000301	0.0011717
	GANN	0.042849027	0.0202376	0.0771107
	ABCNN	0.002421257	0.0000816	0.0092400
	APSONN	0.000045900	0.0000459	0.0000459
	BPNN	0.017557917	0.0032995	0.0420978
80–20	FPNN	0.003152259	0.0001299	0.0060656
	GANN	0.117393978	0.0616088	0.1545841
	ABCNN	0.003678235	0.0009783	0.0057676
	APSONN	0.000738990	0.0006430	0.0008483
	BPNN	0.012365519	0.0062444	0.0314930
70–30	FPNN	0.000520163	0.0001113	0.0018554
	GANN	0.088926722	0.0520731	0.1134958
	ABCNN	0.004173302	0.0007348	0.0099670
	APSONN	0.002791996	0.0016316	0.0046316
	BPNN	0.038997484	0.0140656	0.1022774
60–40	FPNN	0.000556461	0.0001250	0.0029893
	GANN	0.063814055	0.0435508	0.0899280
	ABCNN	0.005073869	0.0008388	0.0099670
	APSONN	0.003782785	0.0025000	0.0057046
	BPNN	0.230197988	0.0320143	0.4416329
50–50	FPNN	0.001371345	0.0006851	0.0019121
	GANN	0.047616710	0.0354461	0.0656102
	ABCNN	0.003970256	0.0008388	0.0099670
	APSONN	0.003313023	0.0015770	0.0057770
	BPNN	0.051838101	0.0269026	0.1589041

Таблица 5: Точность обучения (MSE) для тестовой выборки