

ПРИМЕНЕНИЕ НЕЙРОСЕТИ И ГЕНЕТИЧЕСКОГО АЛГОРИТМА ПРИ ПРОЕКТИРОВАНИИ МОНОЛИТНЫХ ПЛИТ НА ОСНОВАНИИ

А. Е. Желткович¹, В. В. Молош², К. Г. Пархоц³

¹ К. т. н., доцент, доцент кафедры прикладной механики
УО «Брестский государственный технический университет», Брест, Беларусь, e-mail: gelpek@mail.ru

² К. т. н., доцент, доцент кафедры прикладной механики
УО «Брестский государственный технический университет», Брест, Беларусь, e-mail: m.vic@rambler.ru

³ Инженер-программист, Беларусь, e-mail: konstantinparhoc@gmail.com

Реферат

В статье проиллюстрирована возможность конвергенции механики, нейротехнологии и биоподобных технологий. Показана возможность применения мягких вычислений (soft-computing) в задачах, связанных с проектированием. В работе представлены результаты самонапряжений в плите, на основании полученных данных с помощью нейросети, объединённой в систему с генетическим алгоритмом. Рассмотрена возможность оптимизации геометрических параметров плиты при заданных или изменяемых входных данных (прочность, самонапряжение, и др.) путем включения/отключения искусственных генетических признаков. Показано, что для описания состояния конструкции, где кинетика формирования бетонной структуры подчиняется нелинейным зависимостям, применение нейротехнологий и генетических алгоритмов наиболее оправдано.

В статье описан процесс разработки нейронной сети и генетического алгоритма, обсуждается вопрос качества полученных решений.

Ключевые слова: генетический алгоритм, мягкие вычисления, хромосомы, гены, функция приспособленности, самонапряженный бетон, нейронные сети.

APPLICATION OF A NEURAL NETWORK AND GENETIC ALGORITHM IN THE DESIGN OF MONOLITHIC SLABS ON A BASE

A. E. Zheltkovich, V. V. Molosh, K. G. Parchotz

Abstract

The article illustrates the possibility of convergence of mechanics, neurotechnology, and biosimilar technologies. The possibility of applying soft-computing to design-related tasks is shown. The work presents the results of self-stresses in the slab on ground obtained using a neural network combined into a system with a genetic algorithm. The possibility of optimizing the geometric parameters of the slab at the given or variable input parameters (strength, self-stress, etc.) by turning on/off artificial genetic features is considered. It has been shown that to describe the state of the structure, where the kinetics of the formation of the concrete structure obeys nonlinear behaviors, the use of neurotechnologies and genetic algorithms is most justified. The article describes the process of developing a neural network and a genetic algorithm, discusses the quality of the solutions obtained.

Keywords: genetic algorithm, soft computing, chromosomes, genes, fitness function, self-stressed concrete, neural networks.

Мягкие или жесткие решения?

На протяжении многих лет в инженерной науке жёсткие вычисления (hard-computing) находили самое широкое применение. Среди многих инженеров по сей день существует устойчивый стереотип, что метод классических вычислений столь надёжен и эффективен, что преждевременно даже обсуждать альтернативы. Но так ли он безальтернативен с позиции эффективности? В таблице 1 [1] приведен краткий перечень основных характеристик жестких и получающих всё более широкое применение мягких методов вычислений.

Таблица 1 – Перечень характеристик мягких и жёстких методов вычислений

Мягкие вычисления*	Жесткие вычисления
свободны от неточностей, неопределенностей, частичной истины и приближения	нуждаются в модели анализа состояний
опираются на формальную логику и вероятностные подходы	полагаются на двоичную логику «false-true» и четкую систему
имеют стохастический характер	детерминированы
работают с неоднозначными и зашумленными данными	работают с точными данными
могут выполнять параллельные вычисления	выполняют последовательные вычисления
дают приближительные результаты	дают точные результаты
вырабатывают свои собственные программы	требуют написания программ
*Мягкие вычисления (Soft Computing) – термин, введенный Лотфи Заде в 1994 году [2]	

«Жесткие вычисления – это тот древний подход, используемый в вычислениях, который требует от младшего [...] персонала точно заявленной аналитической модели. Результатом жесткого вычислительного подхода является гарантированный, установленный, правильный

результат (получаемый – ред.) [...] с использованием математической модели или алгоритмического правила [...]. Жесткие вычисления не способны найти решение реальной проблемы» [1]. Отметим, что применение соответствующих критериев соответствия обеспечит нахождение решения для некоторой абстрактной, математически идеализированной модели реальности, но не для самой реальности.

Традиционные методы вычислений, к которым, несомненно, относятся жесткие вычисления, подходят для решения многих прикладных задач, но их основной недостаток заключается в том, что они требуют большого количества времени и затрат на вычисления [3]. По этой причине мягкие вычисления являются лучшей альтернативой.

Согласно сложившемуся подходу, можно выделить следующие направления мягких вычислений:

1. Задачи с нечёткой логикой.
2. Задачи, решаемые при помощи нейронных сетей.
3. Эволюционное моделирование – использование теории Дарвина для построения интеллектуальных систем.
4. Задачи, решаемые с использованием теории хаоса или аппарата, описывающего поведение некоторых нелинейных динамических систем, подверженных при определённых условиях явлению, известному как хаос. Поведение такой системы кажется случайным, даже если модель, описывающая систему, является детерминированной [4]. Примерами подобных систем являются турбулентность в атмосфере, потоки плазмы, биологические процессы и другие. Сложные системы чрезвычайно зависимы от первоначальных условий, и небольшие изменения приводят к бифуркациям и фрактальным ветвлениям. При применении классических вычислений это ведёт к потере контроля над процессом.

В исследовании [5] указывается, что сущность мягких вычислений состоит в том, что, в отличие от традиционных, они нацелены на приспособление к всеобъемлющей неточности реального мира. Руководящим принципом мягких вычислений является: «терпимость к неточности, неопределенности и частичной истинности для достижения [...] лучшего согласия с реальностью».

Исследователям, работающим с нагружающими бетонами, вообще с конструкциями из искусственных материалов, к коим без сомнения относится бетон, должна быть хорошо понятна фраза: «Поведение [...] системы кажется случайным, даже если модель, описывающая систему, является детерминированной». По этой причине при исследовании нагружающего бетона, а затем в процессе проектирования плиты на основании с оптимальной высотой бетонного слоя были применены некоторые «soft-computing» методы (нейронная сеть, объединённая в систему с генетическим алгоритмом).

Если говорить о выборе оптимизационных алгоритмов при решении нелинейных задач, к коим относятся задачи, связанные с определением напряжённо-деформированного состояния (НДС) в конструкциях на нагружающем бетоне, то по эффективности работы генетический алгоритм (ГА) является достаточно универсальным (рисунк 1).

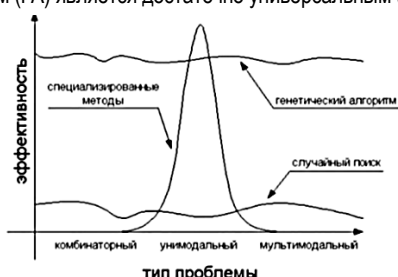


Рисунок 1 – Сравнительная эффективность алгоритмов оптимизации

1 Мягкие вычисления. Искусственная нейросеть

Подход, который использует вычислительные модели, имитирующие архитектурные, структурные или функциональные аспекты биологических нейронных сетей, таких как человеческий мозг, известны с середины XX века.

Искусственные нейронные сети (ИНС или НС), также известные как Neural Networks (NN) или Connection Models – алгоритмические математические модели, имитирующие поведенческие характеристики нервных сетей животных и выполняющие распределённую, параллельную обработку информации. Эти сети полагаются на особую систему обработки информации путем корректировки взаимосвязей между большим количеством внутренних узлов. НС, состоящая из нескольких слоёв связанных нейронов, называется глубокой. Глубокие нейронные сети включают в себя входной слой, скрытые слои и выходной слой; ИНС может соединять узлы нейронов левого уровня с узлами правого уровня [2, 11] (см. рисунок 2б).

Структурно в НС можно выделить два основных способа улучшения работоспособности нейронной сети, а именно горизонтальное расширение за счет увеличения числа нейронных узлов в каждом слое и вертикальное расширение за счет увеличения количества слоев нейронной сети [6].

Полносвязные НС имеют несколько основных особенностей:

1. Нейроны в одном слое не связаны друг с другом.
2. Каждый нейрон в слое N соединен со всеми нейронами в слое N-1 (при полном соединении), а выход нейронов в слое N-1 является входом в нейроны в слое N.
3. Каждое соединение (синапс) имеет вес.

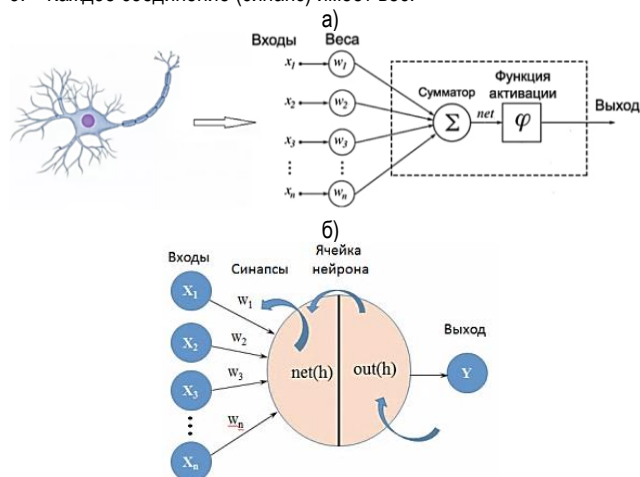


Рисунок 2 – Принципиальная схема нейрона (а), метод обратного распространения ошибки при обновлении весов синапсов (б)

Структура НС для прогнозирования НДС

НС играет роль экстрактора признаков в алгоритме обнаружения цели. Эта схема позволяет НС автономно улавливать зависимости из огромных объемов данных. НС может преобразовывать сигналы от входа к выходу с помощью простой операции сложения и умножения. Таким образом, хорошо обученная сеть может быстро реагировать без громоздких операций вычислений, в отличие от программ, использующих методы жестких вычислений (например, метод конечных элементов). Чтобы уточнить характеристики НДС, была разработана и обучена искусственная нейронная сеть полносвязной архитектуры с семью входящими нейронами, двумя скрытыми слоями и десятью нейронами выходного слоя. Каждый из десяти нейронов выходного слоя учился «видеть» самонапряжение в соответствующей координате плиты. На входной слой НС подавались следующие **исходные данные**, полученные в ходе лабораторных экспериментов [7]: L, B, H – соответственно длина, ширина и высота плиты; $f_{c,cube}^m$ – средняя гарантированная прочность на сжатие бетона нормального веса; $\tau_{1,R(t)}$ – предельные касательные напряжения в контакте плиты и основания; $u_{1,R(t)}$ – смещение плиты, соответствующее максимальному касательному напряжению в контакте плиты и основания; ε_0 – относительная деформация свободного расширения бетона; σ_{CE} – самонапряжение в бетоне при однопроцентном ограничении.

Формирование матриц входных и выходных исходных данных

Поскольку количество параметров НС велико, необходим эффективный метод записи исходных данных (обычно поступающие на вход и извлекаемые на выходе данные представлены в виде матриц или векторов). Для этого создавалась матрица входных и выходных данных для 149 различных временных интервалов. Для данных был создан специальный файл с расширением .csv. Во второй части матрицы (данные для выходного слоя ИНС) помещены напряжения, предварительно сгенерированные при помощи базовой физ.-мат. модели (для десяти точек плиты) на базе экспериментальных наблюдений за перемещениями в соответствующих координатах.

Механизм работы ИНС

Каждый нейрон может рассматриваться как оператор, который изменяет входные данные [8]. На вход он получает некую взвешенную сумму от всех нейронов предыдущего слоя, сигнал же на выходе из нейрона может быть определен следующим образом:

$$y_i = \sum x_i \cdot w_i + b, \quad (1)$$

где x и y – входной и выходной сигналы НС,
 w – весовой параметр синапса,
 b – смещение (рисунок 2а).

Критерии качества работы ИНС

При тестировании нейросети на тестовой выборке использовалась среднеквадратическая (MSE метод) и относительная ошибки, поскольку эти метрики достаточно достоверно отражают точность результата прогнозирования как для пакета из нескольких образцов, так и для единичного образца. Функция потерь (losses) предлагается для оценки разрыва между истинным результатом (целью) и предсказанием, формируемым нейронной сетью. Таким образом, процесс обучения нейронной сети становится задачей оптимизации, целью которой является минимизация потерь. Функция потерь определяется как

$$E = \frac{1}{n} \sum (Y_{target} - Y_{predicted})^2, \quad (2)$$

где n – количество выборок,
 Y_{target} – фактические исходные данные,
 $Y_{predicted}$ – предсказанные величины самонапряжений.

Параметрическая оптимизация, алгоритм градиентного спуска

Целью оптимизации параметров являлось нахождение минимального значения функции E на гиперповерхности потерь. На каждой итерации алгоритм обновляет весовые параметры w в нейронах. Чтобы

найти минимум функции потерь, необходимо взять отрицательный градиент. В модели использовался оптимизатор Adam [9]. Минимум функции потерь определялся как частная производная потери, по весу:

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial out(h)} \cdot \frac{\partial out(h)}{\partial net(h)} \cdot \frac{\partial net(h)}{\partial w_i}, \quad (3)$$

где $net(h) = \sum_{i=1}^n x_i \cdot w_i$ – значение взвешенной суммы от нейронов слоя N-1 приходящей на вход некоторого нейрона в слое N (рисунок 2б),

$out(h) = f(net(h))$ – значение преобразованной (при помощи оператора «ReLU») функции (1) на выходе из нейрона,

$f(net(h)) = "Re Lu"$ – функция активации нейрона (это наиболее распространённая функция активации, в особенности, если решается не классификационная задача, а задача регрессии [10]).

После вычисления потерь информация антиградиента передается от выходного слоя к предыдущим, слой за слоем – обратно к входному слою, обновляя все веса синапсов. Этот метод поиска глобального минимума получил название – метод обратного распространения ошибки (back propagation algorithm – BP) [8, 11, 12]. Уравнение для итерации обновления веса формулируется следующим образом:

$$w_{i+1} = w_i - \alpha \cdot \frac{\partial E}{\partial w_i}, \quad (4)$$

где α – параметр скорости обучения.

Таким образом, с помощью градиентной информации НС может искать относительно оптимальные спуски на гиперповерхности потерь.

Обучение ИНС

Следующим шагом являлось обучение нейросети. Все этапы создания НС, обучения и проверки реализованы на программном языке Python в рамках среды Tensorflow [13]. Количество эпох устанавливалось равным 250. В процессе регуляризации случайным образом отбрасывались 90 % исходных данных назначались в качестве тренировочной выборки (training dataset) и 10 % данных было оставлено для проверки (validation) качества работы модели. Поскольку разработанная нейросеть легкая, ей требовалось примерно 10–15 секунд для обучения.

2 Мягкие вычисления. Генетический алгоритм

Методы оптимизации в общих чертах можно разделить на две категории: метод на основе градиентного спуска и метод без градиента. Метод на основе градиента, такой как метод Ньютона [14] (метод обратного распространения ошибки) или метод сопряженного градиента [15], трудно кодировать в сложной проблеме, в частности связанной с поиском оптимальных топологий, геометрий, и т. д. Генетический алгоритм (ГА), использующий метод оптимизации без градиента, может быть наиболее рационален.

Проектирование плиты на основании является, по сути, задачей выбора оптимальных геометрических параметров плиты при необходимости обеспечения определённого уровня самонапряжения (например, в центре плиты/заданных координатах). С другой стороны, это и задача подбора оптимального состава бетона при известных (или «плавающих») параметрах, описывающих контактные характеристики плиты и основания. На первом этапе работы нейросети была получена реакция плиты (НДС) в зависимости от воздействия входных параметров, которые указаны выше. На этапе 2 данные весов для наиболее оптимальной архитектуры (в частности, это была сеть с 7x10x10x10 нейронами) помещались в код ГА для решения уже обратной оптимизационной задачи. При оптимальных весовых значениях (синапсов) подобранных НС, для определения напряжений в плите размером 4x4 м ставилась задача получить оптимальную толщину плиты H , прочность бетона, значение самонапряжения в кондукторе (при однопроцентном ограничении расширению), контактные характеристики плиты с основанием, свободное расширение. Уровень самонапряжения назначался – 0,55 МПа, для удобства сравнения с результатами расчётов полученных по базовой физ.-мат. модели.

Механизм ГА

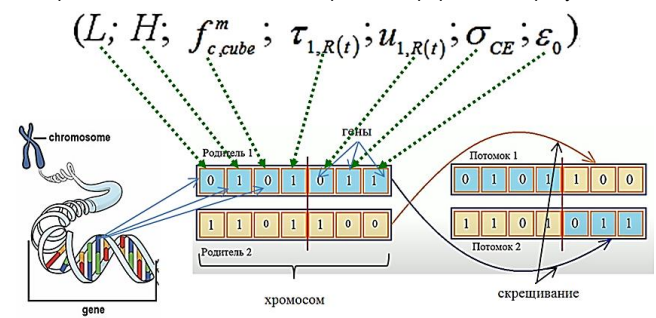
Генетический алгоритм – это своего рода эвристический подход к поиску наиболее оптимального решения, имитирующий процесс эволюции в природе [16]. Поскольку ГА был построен по аналогии эволюции в живой природе, он использует биологические термины [17]. ГА использует принцип «генетического» хранения информации, принцип «размножения» (crossing over), принцип «мутации» (mutation) и «естественного/искусственного отбора» (natural selection).

Основная идея ГА состоит в том, чтобы генерировать несколько решений случайным образом, а затем оценивать оптимальность решений, используя функцию приспособленности (fitness function) [16].

Назначение параметров при проектировании с использованием ГА

В ГА группа параметров с определёнными признаками записывается в виде вектора, называемого хромосомой. Как и в биологии, в искусственной хромосоме содержится «генетическая» информация. В данной работе хромосома представлена вектором параметров, который сохраняет информацию о новом решении в процессе работы оптимизационного алгоритма. Каждая хромосома содержит семь ген-параметров ($L; H; f_{c.cube}^m; \tau_{1,R(t)}; u_{1,R(t)}; \sigma_{CE}; \varepsilon_0$), что означает, что вектор параметров имеет размерность 1x7 исходных значений. Генетический алгоритм генерирует множество хромосом за один раз, используя алгоритм случайных чисел и назначенный диапазон вакантных значений для параметров. Интервал возможной изменчивости гена, например, отвечавшего за среднюю прочность на сжатие бетона нормального веса, назначался в пределах от нуля до 40 МПа, толщина проектируемой плиты от нуля до 0,3 м, и т. д. Таким образом, каждая хромосома с ген-параметрами отличалась одна от другой. Группа параметров в одном поколении носит название популяции. В данном исследовании в силу небольшого числа ген-параметров в ГА устанавливалось 60 поколений-эволюции.

Чтобы генерировать следующую популяцию хромосом, группа параметров взаимодействует друг с другом определённым образом. При моделировании создавалось 40 хромосом, из которых 2 пары наделялись возможностью взаимодействовать (crossing over) [17]. Не взаимодействовавшие хромосомы переходили в следующее поколение без изменений. При небинарном кодировании новый ген (значение физического параметра) случайным образом отбирался из определённого интервала значений и передавался от поколения к поколению. После взаимодействия (crossing over) получался соответствующий ген родительской генерации. Упрощённая схема получения хромосомы нового поколения проиллюстрирована на рисунок 3.



а) – хромосом и ген в ГА, б) – скрещивание между хромосомами
Рисунок 3 – Упрощённая схема получения хромосомы нового поколения

Оптимизация при помощи ГА

В ГА известно несколько подходов отбора оптимального решения: колесо рулетки, турнирный метод, ранговый метод, комбинированный метод, др. От этого зависит с какой скоростью и будет ли вообще популяция двигаться к глобальному оптимуму или останется в локальном экстремуме.

Хромосом с высоким уровнем приспособленности имеет большую вероятность быть выбранным. Хромосом с наименьшей приспособленностью отсеивается с наибольшей вероятностью. В данной работе использовался алгоритм наиболее устойчивого состояния (steady-state selection) для выбора особи, переходящей в новое поколение [18]. Блок-схема генетического алгоритма показана на рисунке 4а.

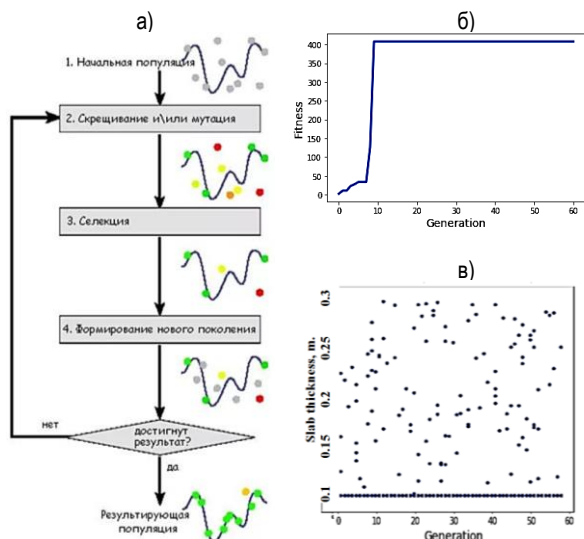
При вычислении оптимальных признаков хромосомы следующей популяции, чтобы избежать преждевременной сходимости алгоритма, вводилась мутация, вероятность которой принималась равной 0,05.

Включение/выключение генетических признаков

Важным преимуществом ГА является возможность управлять вариативностью требуемых ген. При необходимости определённые ген-параметры могут быть заблокированы для эволюционной изменчивости. Например, в данной работе блокировались ген-параметры, отвечавшие за длину плиты и за свободное расширение бетона. И наоборот, другим ген-параметрам обеспечивалась возможность изменяться, например, параметрам, ответственным за прочность бетона, самоупрежение, толщину плиты и др.

Функция приспособленности

Функция приспособленности (fitness function) [16] важная при запуске оптимизатора ГА (рисунок 4б). Функция приспособленности представляет собой разрыв (степень приближения) между реакцией на входное воздействие хромосомы предыдущего поколения к реакции хромосомы текущего поколения на то же воздействие. На основе оценки приспособленности алгоритм выбирает некоторое выдающееся решение. Когда (после определённого количества итераций) показатель fitness function стабилизируется (выходит на минимальную изменчивость), считается, что популяция достигла оптимальности.



а) блок-схема генетического алгоритма, б) уровень приспособленности, достигнутый в процессе оптимизации, в) эволюция ген-параметра ответственного за толщину плиты

Рисунок 4 – Блок-схема ГА и результаты эволюционной оптимизации

Нормализация

Для того чтобы скрещивать особи между собой или манипулировать генами (нейронами) любым другим образом, гены (нейроны) должны быть выражены через безразмерный статистический показатель (используемый, как правило, для сравнения значений разной размерности). В математической статистике известен ряд методов: десятичное масштабирование, минимальная нормализация, нормализация средним (Z-нормализация). В работе была применена нормализация средним (в том числе и на первом этапе, для работы нейросети). Z-нормализация устанавливает среднее значение (математическое ожидание) и дисперсию для каждого гена (нейрона) и представляется формулой

$$z = (x - \mu) / \sigma, \tag{7}$$

где μ и σ – математическое ожидание (mean) и стандартное отклонение (standard deviation) соответственно.

Z-оценка (Z-scores) соответствует количеству стандартных отклонений конкретного параметра до среднего значения [19]. Нормализация для любого параметра z может быть представлена графически, как показано на рисунке 5.

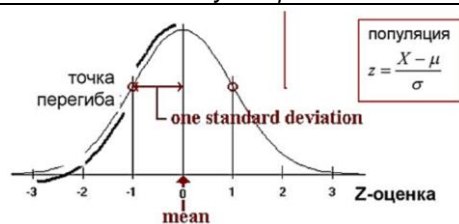
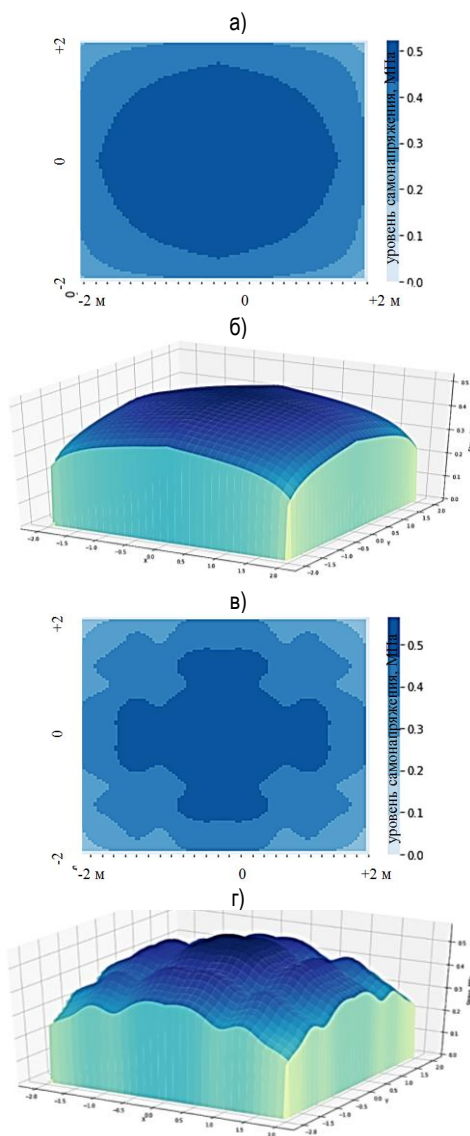


Рисунок 5 – Нормализация данных

Результаты проектирования

На рисунке 6 представлены результаты самоупрежений, полученные при помощи НС (а, б) и ГА (в, г), в плите 4x4x0,1 м, для времени – 80 часов от затворения бетона.

Затем ГА сгенерировал плиту 4x4 м, толщиной 0,100286 м (при этом геометрические размеры реальной [7] полосы-плиты составляли – 4x0,1 м). В процессе проектирования полосы-плиты ГА подобрал самоупрежение при одноосном ограничении – 1,17 МПа (при этом в образцах-кондукторах самоупрежение на 80 час от затворения составляло – 0,81 МПа). Прочность бетона на сжатие при ГА проектировании – 12,9 МПа (в экспериментах на сжатие на 80 час – 10,6 МПа). Пиковое напряжение на контакте – 0,7 МПа (экспериментальное значение – 0,545 МПа), перемещение, соответствующее этому напряжению – 19 мкм (в эксперименте – 18 мкм).



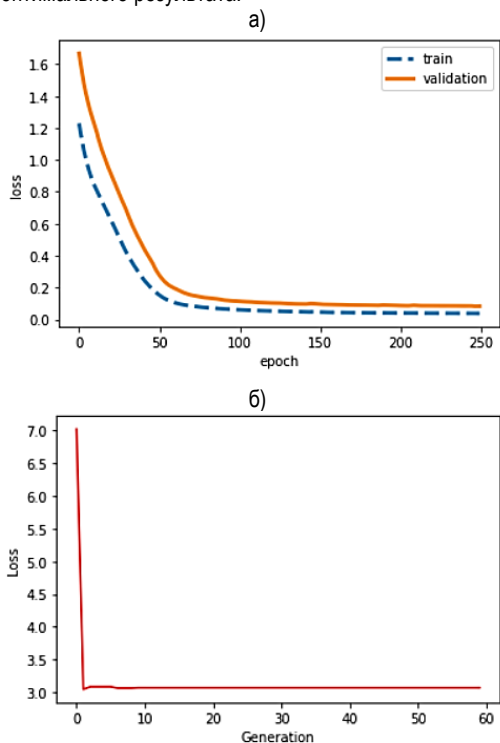
а, б) – полученное при помощи нейросети; в, г) – полученное при помощи генетического алгоритма

Рисунок 6 – Самоупрежение в плите 4x4x0,1 м

Оценка точности

В ходе работы с нейросетью формировался отдельный файл .csv для тестовой выборки данных. Было установлено, что наибольшая сходимость достигается при обучении нейросети с архитектурой 7x10x10x10 нейронов (функции потерь на тренировочной и валидационной выборках представлены на рисунке 7а). По мере увеличения нейронов в промежуточных слоях потери в тестовой выборке непрерывно уменьшались и для сети 7x10x10x10 достигли значения $E = 0,011$. Более того, для оценки точности прогнозирования напряжений с использованием НС дополнительно рассчитывалась относительная ошибка тестовой выборки. Относительная ошибка составила – 1,9 %.

В процессе ГА-оптимизации потери от поколения к поколению уменьшались, приближаясь к нулю (рисунок 7б). Из графика потерь можно сделать вывод, что уже к третьему поколению модель достигла оптимального результата.



а) при обучении НС, б) при оптимизации с помощью ГА
Рисунок 7 – Функции минимизации ошибок

На рисунке 8 представлены результаты моделирования напряжений в плите 4x4x0,1 м при помощи базовой физ.-мат. модели (на базе экспериментальных данных), нейросети с архитектурой 7x10x10x10 нейронов и с использованием генетического алгоритма.

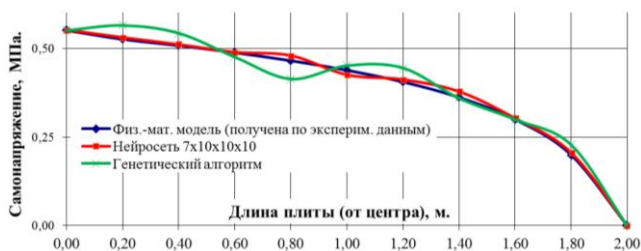


Рисунок 8 – Самонапряжение на 80 час от момента затворения

Заключение

1. В настоящий момент жесткие вычисления являются практически безальтернативным средством описания процессов и явлений, они проверены практикой и чрезвычайно эффективны и, тем не менее, они не описывают непосредственно саму реальность. В случае несложных и линейных процессов жесткие вычисления

достигают порой почти 100 % сходимостью с экспериментом, однако сама реальность для классических моделей продолжает оставаться «вещью в себе». Мягкие вычисления (основанные на методах нечеткой логики, нейросетевых моделях, ГА), хоть и не раскрывают сути физических явлений, но воспринимая поток данных поступающих непосредственно из окружающей среды (при помощи датчиков, сенсоров, и т. д.), как бы нащупывают связь между ними. При этом, если увеличивать плотность потока данных или иметь достаточно времени для наблюдений, то мягкие вычисления (НС, ГА) могут быть подстроены практически под любую задачу (имитируя работу нейронов и/или эволюцию в природе), при этом точность прогнозирования может превысить точность традиционных алгоритмов (особенно в области моделирования нелинейных процессов). Вместе с тем, на определенном этапе (при недостатке экспериментальных данных, этапе предварительной настройки параметров), нейросеть и ГА могут тестироваться и на данных, сгенерированных при помощи традиционных физико-механических моделей.

- Для определения самонапряжения в плитах на основании была разработана и обучена полносвязная нейросеть с архитектурой 7x10x10x10 нейронов в слоях. На тестирующей выборке для плиты 4x4x0,1 м. была достигнута высокая сходимость с физ.-мат. моделью. Среднеквадратичная ошибка составила – 0,011. Относительная ошибка составила – 1,9 %.
- При проектировании плиты, при заявленном требовании обеспечения самонапряжения в центре плиты на уровне 0,55 МПа (как в плите, из тестовой выборки НС) при помощи ГА была сгенерирована плита размерами 4x4x0,100286 м. Самонапряжение в центре плиты составило – 0,549 МПа, при этом максимальная приспособленность генетических признаков была достигнута уже к десятому поколению. Это свидетельствует, что для данного класса задач, изначально назначенные 60 поколений для оптимизации решения несколько избыточны и в дальнейшем могут быть пересмотрены в сторону уменьшения без существенной потери точности для решения.

Список цитированных источников

- Разница между мягкими и жесткими вычислениями. Progler. [Электронный ресурс]. – Режим доступа: <https://progler.ru/blog>. – Дата доступа: 10.11.2022.
- Zadeh, Lotfi A. Fuzzy Logic, Neural Networks, and Soft Computing / Lotfi A. Zadeh – Communications of the ACM. – March 1994. – Vol. 37, No. 3. –P 77–84.
- Разница между мягкими и жесткими вычислениями. Surveillance-packages. [Электронный ресурс]. – Режим доступа: <https://ru1.surveillancpackages.com/difference-between-soft-computing-and-hard-computing-4b2a>. – Дата доступа: 11.11.2022.
- Игорь Н. фон Бекман / Нелинейная динамика сложных систем: теория и практика. Метанаука. Эволюция систем. Материалы к курсу лекций и учебнику / Игорь Н. фон Бекман. – Москва : МГУ. – 2018. – С. 613
- Применение интеллектуальной обработки информации Soft Computing в физике. [Электронный ресурс] – Режим доступа: <http://spacelab.mininuniver.ru>. – Дата доступа: 12.11.2022.
- Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning / Shin H. C. [et al.] // IEEE transactions on medical imaging. – 2016. – 35(5). – P. 1285–1298.
- Желткович, А.Е. Расчёт вынужденных перемещений и напряжений от усадки в монолитных бетонных плитах, взаимодействующих с основанием / А. Е. Желткович, В. В. Тур // Строительная наука и техника. – 2011. – № 2 (35) : – С. 120–125.
- Chen H, Lu F, He B. Topographic property of backpropagation artificial neural network: From human functional connectivity network to artificial neural network [J]. Neurocomputing. – 2020. – 418. – P. 200–210.
- Adam: A Method for Stochastic Optimization [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/1412.6980>. – Дата доступа: 14.11.2022.

10. Функции активации нейронов сети: сигмоида, линейная, ступенчатая, ReLu [Электронный ресурс]. – Режим доступа: <https://neurohive.io/ru/osnovy-data-science/activation-functions>. – Дата доступа: 14.11.2022.
11. Галушкин А. И. Синтез многослойных систем распознавания образов / А. И. Галушкин. – М.: «Энергия», 1974.
12. Werbos, P. J. Beyond regression: New tools for prediction and analysis in the behavioral sciences / P. J. Werbos. – Cambridge: Harvard University, 1974.
13. Citing TensorFlow [Электронный ресурс]. – Режим доступа: <https://www.tensorflow.org/about/bib>. – Дата доступа: 16.11.2022.
14. Polyak B T. Newton's method and its use in optimization [J]. European Journal of Operational Research, 2007, 181(3): pp. 1.086–1.096.
15. Nazareth, J. L. Conjugate gradient method / J. L. Nazareth. – Wiley Interdisciplinary Reviews: Computational Statistics, 2009. – Vol 1(3). – P. 348–353.
16. Katoch, S., Chauhan, S.S., Kumar, V.A. Review on genetic algorithm: past, present, and future / S. Katoch, S.S. Chauhan, V. A. Kumar // Multimedia Tools and Applications. – 2021. – Vol 80(5). – P. 8091–8126.
17. Safitri, A. R., Muslim, M. A. Improved accuracy of naive bayes classifier for determination of customer churn uses smote and genetic algorithms / A .R. Safitri, M. A. Muslim // Journal of Soft Computing Exploration. – 2020. – Vol 1(1). – P. 70–75.
18. Differential evolution algorithm with multiple mutation strategies based on roulette wheel selection / Qian W, Chai J, Xu Z [et al.] // Applied Intelligence. – 2018. – Vol 48(10). – P. 3612–3629.
19. Нормализация данных (Data normalization) [Электронный ресурс]. – Режим доступа: <https://wiki.loginom.ru/articles/data-normalization>. – Дата доступа: 15.11.2022.
6. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning / Shin H. C. [et al.] // IEEE transactions on medical imaging. – 2016. – 35(5). – R. 1285–1298.
7. Zheltkovich, A.E. Raschyot vyzhizhennykh peremeshchenij i napryazhenij ot usadki v monolitnykh betonnykh plitah, vzaimodejstvuyushchih s osnovaniem / A. E. Zheltkovich, V. V. Tur // Stroitel'naya nauka i tekhnika. – 2011. – № 2 (35) : – S. 120–125.
8. Chen H, Lu F, He B. Topographic property of backpropagation artificial neural network: From human functional connectivity network to artificial neural network [J]. Neurocomputing. – 2020. – 418. – R. 200–210.
9. Adam: A Method for Stochastic Optimization [Elektronnyj resurs]. – Rezhim dostupa: <https://arxiv.org/abs/1412.6980>. – Data dostupa: 14.11.2022.
10. Funkcii aktivacii neironov seti: sigmoida, linejnaya, stupenchataya, ReLu [Elektronnyj resurs]. – Rezhim dostupa: <https://neurohive.io/ru/osnovy-data-science/activation-functions>. – Data dostupa: 14.11.2022.
11. Galushkin A. I. Sintez mnogoslujnyh sistem raspoznavaniya obrazov / A. I. Galushkin. – M.: «Energija», 1974.
12. Werbos, P. J. Beyond regression: New tools for prediction and analysis in the behavioral sciences / P. J. Werbos. – Cambridge: Harvard University, 1974.
13. Citing TensorFlow [Elektronnyj resurs]. – Rezhim dostupa: <https://www.tensorflow.org/about/bib>. – Data dostupa: 16.11.2022.
14. Polyak B T. Newton's method and its use in optimization [J]. European Journal of Operational Research, 2007, 181(3): pp. 1.086–1.096.
15. Nazareth, J. L. Conjugate gradient method / J. L. Nazareth. – Wiley Interdisciplinary Reviews: Computational Statistics, 2009. – Vol 1(3). – R. 348–353.
16. Katoch, S., Chauhan, S.S., Kumar, V.A. Review on genetic algorithm: past, present, and future / S. Katoch, S.S. Chauhan, V. A. Kumar // Multimedia Tools and Applications. – 2021. – Vol 80(5). – R. 8091–8126.
17. Safitri, A. R., Muslim, M. A. Improved accuracy of naive bayes classifier for determination of customer churn uses smote and genetic algorithms / A .R. Safitri, M. A. Muslim // Journal of Soft Computing Exploration. – 2020. – Vol 1(1). – R. 70–75.
18. Differential evolution algorithm with multiple mutation strategies based on roulette wheel selection / Qian W, Chai J, Xu Z [et al.] // Applied Intelligence. – 2018. – Vol 48(10). – R. 3612–3629.
19. Normalizaciya dannyh (Data normalization) [Elektronnyj resurs]. – Rezhim dostupa: <https://wiki.loginom.ru/articles/data-normalization>. – Data dostupa: 15.11.2022.

References

Материал поступил 18.01.2023, одобрен 27.02.2023, принят к публикации 28.02.2023