

УДК 004.8

5.2.2. Математические, статистические и инструментальные методы экономики (физико-математические науки, экономические науки)

СИСТЕМО-КОГНИТИВНЫЕ НЕЙРОСЕТИ КАК ПЕРСПЕКТИВА СИСТЕМЫ «ЭЙДОС» И ЕЕ ПРИМЕНЕНИЯ В ЭКОНОМИКЕ

Луценко Евгений Вениаминович
д.э.н., к.т.н., профессор
Web of Science ResearcherID S-8667-2018
Scopus Author ID: 57188763047
РИНЦ SPIN-код: 9523-7101
prof.lutsenko@gmail.com <http://lc.kubagro.ru>
https://www.researchgate.net/profile/Eugene_Lutsenko
*Кубанский Государственный Аграрный университет
имени И.Т.Трубилина, Краснодар, Россия*

Система «Эйдос» обеспечивает преобразование исходных данных в информацию, а ее в знания и решение путем применения этих знаний задач идентификации, прогнозирования, принятия решений и исследования моделируемой предметной области путем исследования ее модели. При этом генерируется 3 статистических и 7 системно-когнитивных моделей (СК-модели). В работах автора показано, что эти модели образуют конфигурацию моделей в смысле В.А.Лефевра, т.е. это все модели, которые можно построить на основе матрицы абсолютных частот, путем сравнения абсолютных и относительных частот с путем вычитания и деления и нормирования вычитанием 1 и логарифмированием. Это означает, что больше моделей на основе матрицы абсолютных частот таким путем получить невозможно. Однако, возможно получить новые модели на основе самих этих классических статистических и системно-когнитивных моделей системы «Эйдос». Это возможно потому, что классические модели системы «Эйдос», как и нейросетевые модели, количественно отражают силу и направление взаимосвязи между значениями входных и выходных параметров. Традиционно эти взаимосвязи выявляются путем анализа исходных данных. Но в данной работе предлагается рассматривать классические модели системы «Эйдос» как обучающую выборку для создания нейросетевых моделей. Таким образом, будут генерироваться модели, которые предлагается назвать системно-когнитивные нейросети (СК-нейросети). Результат обучения нейросети возможно представить в виде матричной модели, полностью аналогичной по структуре классическим моделям системы «Эйдос». Поэтому возможно использовать системно-когнитивные нейросетевые модели вместо классических моделей системы «Эйдос». Это позволяет задействовать весь функционал

UDC 004.8

5.2.2. Mathematical, statistical and instrumental methods of economics (physical and mathematical sciences, economic sciences)

SYSTEMIC COGNITIVE NEURAL NETWORKS AS A PERSPECTIVE OF THE EIDOS SYSTEM AND ITS APPLICATION IN THE ECONOMY

Lutsenko Evgeniy Veniaminovich
Doctor of Economics, Cand.Tech.Sci., Professor
Web of Science ResearcherID S-8667-2018
Scopus Author ID: 57188763047
RSCI SPIN code: 9523-7101
prof.lutsenko@gmail.com <http://lc.kubagro.ru>
https://www.researchgate.net/profile/Eugene_Lutsenko
*Kuban State Agrarian University named after I.T.
Trubilin, Krasnodar, Russia*

The Eidos system ensures the transformation of initial data into information, and its transformation into knowledge and solution by applying this knowledge to the tasks of identification, forecasting, decision-making and research of the simulated subject area by studying its model. At the same time, 3 statistical and 7 system-cognitive models (SC-models) are generated. The author's works show that these models form a configurator of models in the sense of V.A.Lefebvre, i.e. These are all models that can be built on the basis of an absolute frequency matrix, by comparing absolute and relative frequencies with subtraction and division and normalization by subtraction of 1 and logarithm. This means that it is impossible to obtain more models based on the absolute frequency matrix in this way. However, it is possible to obtain new models based on these classical statistical and system-cognitive models of the Eidos system themselves. This is possible because classical models of the Eidos system, like neural network models, quantitatively reflect the strength and direction of the relationship between the values of input and output parameters. Traditionally, these relationships are identified by analyzing the source data. But in this paper it is proposed to consider the classical models of the Eidos system as a training sample for creating neural network models. Thus, models will be generated, which are proposed to be called system-cognitive neural networks (SK-neural networks). The result of neural network training can be represented in the form of a matrix model, completely similar in structure to the classical models of the Eidos system. Therefore, it is possible to use system-cognitive neural network models instead of the classical models of the Eidos system. This allows you to use all the functionality of the Eidos system based on the use of not only classical models of the system, but also on the basis of system-cognitive

системы «Эйдос» уже основе применения не только классических моделей системы, но и на основе системно-когнитивных нейросетевых моделей. В этом состоит одно из перспективных направлений развития системы «Эйдос» и ее применения в экономике. Другие перспективы состоят в развитии системы на современных языках программирования, таких как Питон, C++ и других

neural network models. This is one of the promising directions for the development of the Eidos system and its application in the economy. Other prospects are the development of the system in modern programming languages such as Python, C++ and others

Ключевые слова: PYTHON, C++, СИСТЕМНО-КОГНИТИВНЫЕ МОДЕЛИ, АСК-АНАЛИЗ, АВТОМАТИЗИРОВАННЫЙ СИСТЕМНО-КОГНИТИВНЫЙ АНАЛИЗ, ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА «ЭЙДОС»

Keywords: PYTHON, C++, SYSTEM-COGNITIVE MODELS, ASK ANALYSIS, AUTOMATED SYSTEM-COGNITIVE ANALYSIS, INTELLIGENT EIDOS SYSTEM

<http://dx.doi.org/10.21515/1990-4665-199-012>

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ	3
2. МЕТОД	4
2.1. КРАТКОЕ ОПИСАНИЕ МЕТОДА РЕШЕНИЯ ПРОБЛЕМЫ	4
2.2. ОБОСНОВАНИЕ ВЫБОРА ЯЗЫКА ПРОГРАММИРОВАНИЯ ДЛЯ РАЗРАБОТКИ ДЕЙСТВУЮЩИХ ПРОТОТИПОВ РЕШЕНИЯ ПРОБЛЕМЫ.....	6
3. РЕЗУЛЬТАТЫ	8
3.1. КАК УСТРОЕНЫ МАТРИЦЫ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ В БИБЛИОТЕКАХ NUMPY, TENSORFLOW И PYTORCH? ..	8
3.2. ВИДЫ ФУНКЦИЙ АКТИВАЦИИ НЕЙРОНОВ.....	9
3.3. МАТРИЧНОЕ ПРЕДСТАВЛЕНИЕ ОДНОГО СЛОЯ НЕЙРОННОЙ СЕТИ: КОЛОНКИ - Y_j , СТРОКИ - X_i , ЗНАЧЕНИЯ (ВЕСА) - W_{ij}	11
3.4. ВЗАИМОСВЯЗЬ МАТРИЦ СЛОЕВ НЕЙРОННОЙ СЕТИ: В СЛЕДУЮЩЕМ СЛОЕ РОЛЬ X_i ВЫПОЛНЯЕТ Y_j ПРЕДЫДУЩЕГО СЛОЯ.....	12
3.5. РЕДУКЦИЯ МНОГОСЛОЙНОЙ СЕТИ К ОДНОСЛОЙНОЙ: РАСЧЕТ МАТРИЦЫ ВЗАИМОСВЯЗИ ВХОДОВ 1-ГО СЛОЯ X_i С ВЫХОДОМ ПОСЛЕДНЕГО СЛОЯ Y_j	14
3.6. РАСЧЕТ В ПИТОНЕ МАТРИЦ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ W_{ij} ДЛЯ НАИБОЛЕЕ ПОПУЛЯРНЫХ НЕЙРОННЫХ СЕТЕЙ	15
4. ОБСУЖДЕНИЕ	18
4.1. ИСПОЛЬЗОВАНИЕ МАТРИЧНЫХ СИСТЕМНО-КОГНИТИВНЫХ МОДЕЛЕЙ В КАЧЕСТВЕ ОБУЧАЮЩЕЙ ВЫБОРКИ ДЛЯ НЕЙРОСЕТЕЙ	18
4.2. ТИПЫ НЕЙРОСЕТЕЙ, ДОПУСКАЮЩИЕ ПРЕДСТАВЛЕНИЕ КОЭФФИЦИЕНТОВ ВЗАИМОСВЯЗИ МЕЖДУ ЗНАЧЕНИЯМИ РЕЦЕПТОРОВ И НЕЙРОНАМИ В ВИДЕ МАТРИЦ, АНАЛОГИЧНЫХ СИСТЕМНО-КОГНИТИВНЫМ МОДЕЛЯМ.....	21
4.3. РАЗРАБОТКА ПРОГРАММЫ НА ЯЗЫКЕ ПИТОН, УЛУЧШАЮЩЕЙ СИСТЕМНО-КОГНИТИВНЫЕ МОДЕЛИ СИСТЕМЫ «ЭЙДОС» С ПОМОЩЬЮ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ	22
5. ВЫВОДЫ	23
ЛИТЕРАТУРА	24

<http://ej.kubagro.ru/2024/05/pdf/12.pdf>

1. Введение

Система «Эйдос» обеспечивает преобразование исходных данных в информацию, а ее в знания и решение путем применения этих знаний задач идентификации, прогнозирования, принятия решений и исследования моделируемой предметной области путем исследования ее модели. При этом генерируется 3 статистических и 7 системно-когнитивных моделей (СК-модели). В работах автора [1-8] и других показано, что эти модели образуют конфигуратор моделей в смысле В.А.Лефевра, т.е. это все модели, которые можно построить на основе матрицы абсолютных частот, путем сравнения абсолютных и относительных частот с путем вычитания и деления и нормирования вычитанием 1 и логарифмированием. Это означает, что больше моделей на основе матрицы абсолютных частот таким путем получить невозможно.

Однако, возможно получить новые модели на основе самих этих классических статистических и системно-когнитивных моделей системы «Эйдос». Это возможно потому, что классические модели системы «Эйдос», как и нейросетевые модели, количественно отражают силу и направление взаимосвязи между значениями входных и выходных параметров. Традиционно эти взаимосвязи выявляются путем анализа исходных данных. Но в данной работе предлагается рассматривать классические модели системы «Эйдос» как обучающую выборку для создания нейросетевых моделей. Таким образом, будут генерироваться модели, которые предлагается назвать системно-когнитивные нейросети (СК-нейросети).

Результат обучения нейросети возможно представить в виде матричной модели, полностью аналогичной по структуре классическим моделям системы «Эйдос». Поэтому возможно использовать системно-когнитивные нейросетевые модели вместо классических моделей системы «Эйдос».

Для ряда нейросетей такое представление не является проблемой. Но для многослойных и некоторых других нейронных сетей еще нужно разработать способ такого представления их моделей. В данной статье предлагается вариант решения этой проблемы.

Это позволяет задействовать весь функционал системы «Эйдос» уже основе применения не только классических моделей системы, но и на основе системно-когнитивных нейросетевых моделей.

В этом состоит одно из перспективных направлений развития системы «Эйдос» и ее применения в экономике. Другие перспективы состоят в развитии системы на современных языках программирования, таких как Питон, С++ и других.

2. Метод

2.1. Краткое описание метода решения проблемы

Для реализации предложенного подхода к созданию системно-когнитивных нейросетевых моделей (СК-нейросетей) на основе классических моделей системы "Эйдос" используется следующий метод.

1. Выбор и подготовка данных:

Исходные данные, используемые для построения классических моделей системы "Эйдос", должны быть тщательно отобраны и подготовлены.

Этап подготовки данных включает в себя очистку от выбросов, обработку пропущенных значений, а также возможно преобразование и нормализацию данных в соответствии с требованиями конкретной нейросетевой архитектуры.

2. Построение классических моделей:

На основе подготовленных данных строятся классические статистические и системно-когнитивные модели с помощью системы "Эйдос".

Эти модели количественно отражают взаимосвязи между значениями входных и выходных параметров и используются как обучающая выборка для нейросетевых моделей.

3. Обучение нейросетевых моделей:

На основе классических моделей системы "Эйдос" происходит обучение нейросетевых моделей.

Для этого используются современные методы машинного обучения и глубокого обучения, такие как нейронные сети с различными архитектурами (например, полносвязные, сверточные, рекуррентные нейронные сети), а также соответствующие алгоритмы оптимизации и регуляризации.

4. Оценка и тестирование моделей:

После завершения обучения моделей производится их оценка и тестирование на отложенной выборке данных или с использованием кросс-валидации.

Оценка моделей проводится с учетом метрик качества, таких как точность, полнота, F1-мера и другие, а также анализа их поведения на различных типах входных данных и сценариях использования.

5. Интеграция и применение моделей:

Полученные системно-когнитивные нейросетевые модели (СК-нейросети) могут быть интегрированы в систему "Эйдос" для использования в различных задачах и исследованиях.

Это позволяет использовать весь функционал системы "Эйдос" на основе применения не только классических моделей, но и новых системно-когнитивных нейросетевых моделей.

Таким образом, предложенный метод позволяет эффективно использовать современные методы машинного обучения для расширения функциональности системы "Эйдос" и повышения ее эффективности в

различных областях применения, включая экономику и другие научные и прикладные задачи.

2.2. Обоснование выбора языка программирования для разработки действующих прототипов решения проблемы

С целью изучения на реально действующих прототипах эффективности различных вариантов использования матриц системно-когнитивных моделей системы «Эйдос» в качестве обучающей выборки для генерации нейронных сетей могут быть применены различные языки программирования.

Для обоснования выбора наиболее рационального языка программирования учитывались следующие требования к нему:

1. Язык программирования и IDE среда разработки должны быть просты и удобны для разработки реально действующих прототипов.

2. Язык программирования должен иметь большое количество библиотек для создания нейронных сетей.

3. Язык программирования должен быть современным, популярным и перспективным.

1. Простота и удобство для разработки реально действующих прототипов:

- Python: Python известен своей простотой и читаемостью кода, что делает его отличным выбором для быстрой разработки прототипов. Есть множество IDE, таких как PyCharm, Jupyter Notebook, и VS Code, которые обеспечивают удобную среду разработки.

- JavaScript: JavaScript также широко используется для быстрой разработки прототипов веб-приложений благодаря своей простоте и доступности. IDE, такие как Visual Studio Code, WebStorm и Atom, обеспечивают хорошую поддержку.

- C++: C++ может быть более сложным для быстрой разработки прототипов из-за своей строгой типизации и низкоуровневых особенностей, но его все еще можно использовать для этой цели с помощью современных IDE, таких как Visual Studio или CLion.

- C#: C# с богатой средой разработки в Visual Studio хорошо подходит для создания прототипов благодаря своей простоте и хорошей интеграции с платформой .NET.

- R: R является отличным выбором для прототипирования статистических и аналитических приложений. IDE, такие как RStudio, предоставляют удобную среду для работы с R.

2. Наличие библиотек для создания нейронных сетей:

- Python: Python имеет огромное сообщество и множество библиотек для машинного обучения и нейронных сетей, таких как TensorFlow, PyTorch, Keras, и scikit-learn.

- JavaScript: JavaScript также имеет некоторые библиотеки для нейронных сетей, такие как TensorFlow.js и Brain.js, но выбор не такой обширный, как в Python.

- C++ и C#: В этих языках также есть некоторые библиотеки для машинного обучения, но они не так распространены и развиты, как в Python.

- R: R имеет хорошие библиотеки для статистического анализа и машинного обучения, но возможностей для работы с нейронными сетями не так много, как в Python.

3. Популярность и перспективы:

- Python: Python является одним из самых популярных и востребованных языков программирования в мире благодаря своей простоте, мощности и обширной экосистеме библиотек.

- JavaScript: JavaScript также очень популярен, особенно в веб-разработке, и его популярность продолжает расти.

- C++ и C#: Оба языка остаются популярными в некоторых областях, таких как игровая индустрия (C++) и разработка приложений для Windows (C#), но их популярность не так высока, как у Python и JavaScript.

- R: R широко используется в области статистического анализа и аналитики данных, и его популярность остается стабильной в этой области.

Исходя из этих критериев, язык Python выделяется как наиболее удобный и перспективный язык программирования для разработки прототипов использования классических системно-когнитивных моделей системы «Эйдос» в качестве обучающей выборки для генерации нейронных сетей и когнитивных неросетевых моделей (СК-нейросетей). JavaScript также имеет свои преимущества, особенно в веб-разработке, но выбор библиотек для нейронных сетей не так велик, как в Python. Поэтому по всем обоснованным и рассмотренным критериям для разработки прототипов решения проблемы был выбран язык Питон.

3. Результаты

3.1. Как устроены матрицы нейросетевых моделей в библиотеках NumPy, TensorFlow и PyTorch?

В Python матрицы в нейросетевых моделях обычно реализованы с использованием библиотеки NumPy или аналогичных инструментов, таких как TensorFlow или PyTorch. В основе этих матриц лежат массивы данных, которые представляют собой основные структуры данных для работы с тензорами и многомерными массивами.

Вот как обычно устроены матрицы в нейросетевых моделях на Python:

1. NumPy Arrays: NumPy - это библиотека Python, предоставляющая поддержку больших многомерных массивов и матриц, вместе с большим

набором математических функций для операций с этими массивами. В нейросетевых моделях, как правило, используются массивы NumPy для представления входных данных, весов, смещений и выходных данных.

2. TensorFlow Tensors: TensorFlow - это открытая библиотека для машинного обучения, разработанная компанией Google. В TensorFlow тензоры являются основными структурами данных. Они могут быть многомерными массивами, аналогичными NumPy, но с дополнительными возможностями для эффективных вычислений на графах вычислений.

3. PyTorch Tensors: PyTorch - еще один популярный фреймворк для глубокого обучения, который также использует тензоры в качестве основных структур данных.

3.2. Виды функций активации нейронов

В контексте нейронных сетей, смещения (biases) являются параметрами, которые добавляются к взвешенной сумме входных данных для каждого нейрона в слое. Они позволяют управлять сдвигом активации нейрона и обеспечивают более гибкую модель.

Каждый нейрон в слое принимает входные данные, умножает их на соответствующие веса и добавляет смещение перед применением функции активации.

Функция активации нейрона - это нелинейная функция, которая применяется к выходу нейрона после вычисления его взвешенной суммы входных данных и весов. Она определяет активационное состояние нейрона, т.е., в зависимости от входных данных, нейрон может быть "включен" (активирован) или "выключен" (неактивирован).

Существует множество различных функций активации, каждая из которых имеет свои уникальные свойства и применение. Ниже приведены некоторые из наиболее распространенных функций активации:

1. Сигмоидальная функция (Логистическая функция):

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Эта функция сжимает входные данные в диапазоне (0, 1), что делает ее полезной для бинарной классификации.

2. Гиперболический тангенс (tanh):

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Данная функция похожа на сигмоидальную функцию, но сжимает входные данные в диапазоне (-1, 1).

3. Rectified Linear Unit (ReLU):

$$\text{ReLU}(x) = \max(0, x)$$

Это простая нелинейная функция, которая возвращает входное значение, если оно положительное, и 0 в противном случае. Она часто используется в глубоком обучении из-за своей простоты и эффективности.

4. Leaky ReLU:

$$\text{Leaky ReLU}(x) = \begin{cases} x, & \text{если } x > 0 \\ \alpha x, & \text{если } x \leq 0 \end{cases}$$

Это вариант ReLU, который позволяет небольшому, постоянному отрицательному наклону (обычно $\alpha = 0.01$) для отрицательных значений, чтобы избежать "умирания" нейронов.

5. Софтмакс (Softmax):

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}}$$

Эта функция преобразует входные значения в вероятностное распределение, где сумма всех выходных значений равна 1. Она часто используется в выходном слое многоклассовой классификации.

Каждая из этих функций активации имеет свои особенности и применения, и выбор конкретной функции может зависеть от конкретной задачи и архитектуры нейронной сети.

Итак, выход нейрона y связан с его входами x_i с помощью функции активации f . Аргументом же функции активации f обычно является взвешенное весовыми коэффициентами w_i произведение входов нейрона плюс смещение b :

$$y = f\left(\sum_{i=1}^n (x_i \cdot w_i) + b\right)$$

Где:

- x_i - i -й входной признак нейрона,
- w_i - вес, соответствующий i -му входному признаку,
- b - смещение (bias),
- n - количество входных признаков нейрона,
- f - функция активации нейрона.

Смещение b добавляется к взвешенной сумме входных данных перед применением функции активации. Это позволяет модели управлять сдвигом активации нейрона и обеспечивает гибкость в настройке поведения нейрона. Без смещения нейрон может ограничиться простой линейной моделью, а с добавлением смещения он может обучаться адаптировать свои выходы к желаемым значениям в более гибким образом.

3.3. Матричное представление одного слоя нейронной сети:

колонки - y_j , строки - x_i , значения (веса) - W_{ij}

Да, можно представить матрицу слоя нейронной сети в виде, где строки представляют диапазоны x_i , столбцы представляют диапазоны y_j , а значения ячеек этой матрицы будут весами W_{ij} .

Такое представление удобно для понимания взаимосвязей между входами и выходами нейронов в слое. Предположим, что у нас есть m входов (x_1, x_2, \dots, x_m) и n выходов (y_1, y_2, \dots, y_n) . Тогда матрица весов W будет иметь размерность $m \times n$, где элемент W_{ij} будет представлять вес, связывающий i -й вход и j -й выход.

Вот как можно представить такую матрицу:

Таким образом, каждый столбец этой матрицы будет содержать веса, применяемые к соответствующему выходу y_j , а каждая строка будет представлять веса, применяемые ко всем выходам для соответствующего входа x_i .

3.4. Взаимосвязь матриц слоев нейронной сети: в следующем слое роль x_i выполняет y_j предыдущего слоя

Да, в следующем слое роль x_i может выполнять y_j из предыдущего слоя. В глубоких нейронных сетях каждый слой, за исключением входного, получает данные с выхода предыдущего слоя в качестве своего входа. Таким образом, выходные данные y_j предыдущего слоя (колонки матрицы) являются входными данными x_i (строки матрицы) для следующего слоя.

Предположим, что выходной слой имеет p нейронов (z_1, z_2, \dots, z_p) . Тогда веса, связывающие j -й выход из предыдущего слоя с i -м входом текущего слоя, могут быть представлены в матрице весов W размерности $p \times r$, где каждый элемент W_{ij} будет соответствовать весу, применяемому к связи между j -м выходом предыдущего слоя и i -м входом текущего слоя.

Взаимосвязь матриц слоев нейронной сети в аналитической форме имеет вид:

Пусть:

- X - матрица входных данных текущего слоя размерности $m \times n$, где m - количество примеров в обучающем наборе, а n - количество входных нейронов текущего слоя.

- W - матрица весов размерности $n \times r$, где r - количество нейронов в следующем слое.

- Y - матрица выходных данных следующего слоя размерности $m \times r$, где m - количество примеров в обучающем наборе, а r - количество нейронов в следующем слое.

Тогда взаимосвязь между матрицами слоев нейронной сети может быть представлена следующим образом:

$$Y = XW$$

Где:

- Каждый элемент Y_{ij} матрицы Y представляет собой выходное значение i -го примера для j -го нейрона следующего слоя.

- Каждый элемент X_{ik} матрицы X представляет собой значение k -го входного нейрона для i -го примера текущего слоя.

- Каждый элемент W_{kj} матрицы W представляет собой вес, применяемый к связи между k -м входным нейроном текущего слоя и j -м выходным нейроном следующего слоя.

Таким образом, *каждый элемент Y_{ij} матрицы Y вычисляется как сумма произведений соответствующих элементов X и W по всем k входным нейронам текущего слоя:*

$$Y_{ij} = \sum_{k=1}^n X_{ik} \cdot W_{kj}$$

Это позволяет нам представить взаимосвязь между слоями нейронной сети в аналитической форме, где каждый входной нейрон следующего слоя

связан с каждым выходным нейроном предыдущего слоя через соответствующие веса.

3.5. Редукция многослойной сети к однослойной: расчет матрицы взаимосвязи входов 1-го слоя x_i с выходом последнего слоя y_j

Учитывая, что *матрицы весовых коэффициентов каждого слоя нейронной сети связаны друг с другом с помощью операции умножения матриц*, возможно расчетным путем получить матрицу связей входов 1-го слоя нейронной сети и выходов последнего ее слоя. Это позволяет свести (редуцировать) многослойную нейронную сеть к эквивалентной ей однослойной сети.

Если у нас есть матрицы весов $W1$, связывающие вход 1-го слоя x_{1i} с выходом 1-го слоя y_{1j} , и матрицы весов $W2$, связывающие вход 2-го слоя x_{2i} с выходом 2-го слоя y_{2j} , то матрица, связывающая вход 1-го слоя x_{1i} с выходом 2-го слоя y_{2j} , может быть вычислена перемножением матриц $W1$ и $W2$.

Пусть $W1$ - матрица весов, связывающая вход 1-го слоя x_{1i} с выходом 1-го слоя y_{1j} , размерности $m \times n$, где m - количество входов, а n - количество нейронов в 1-м слое.

Пусть $W2$ - матрица весов, связывающая вход 2-го слоя x_{2i} с его выходом y_{2j} , размерности $n \times r$, где n - количество нейронов в последнем слое, а r - количество выходов.

При этом выход 1-го слоя y_{1j} подается на вход 2-го слоя x_{2i} :

$$x_{2i} = y_{1j}$$

Тогда матрица, связывающая вход 1-го слоя x_{1i} с выходом 2-го слоя y_{2j} , будет представлена следующим образом:

$$W12 = W1 \times W2$$

Где W_{12} - матрица размерности $m \times r$, представляющая связь между входом 1-го слоя x_{1i} и выходом 2-го слоя y_{2j} .

Итак, используя данный подход, мы можем объединить любые два смежных слоя нейронной сети в один слой. Если в нейронной сети больше 2-х слоев, то данный подход позволяет последовательно рассчитать матрицы взаимосвязей всех пар смежных слоев и, в конце концов, получить однослойную нейронную сеть.

Это действительно интересный подход, который может быть использован, например, в случае анализа влияния входных переменных на выход нейронной сети, если известны только веса связей внутри сети. Особенно ценным он является потому, что позволяет свести многослойные нейронные сети к виду классических системно-когнитивных моделей системы «Эйдос» и использовать их в этом качестве, используя весь функционал данной системы для решения различных задач с использованием этих системно-когнитивных нейросетевых моделей.

Однако, стоит отметить, что в реальных многослойных сетях такой подход может быть сложно применить из-за сложной структуры сети и различных функций активации, используемых в разных слоях.

3.6. Расчет в Питоне матриц нейросетевых моделей W_{ij} для наиболее популярных нейронных сетей

Чтобы вычислить матрицу W_{ij} для наиболее популярных нейронных сетей, сначала нужно получить архитектуры и матрицы весов этих сетей. Архитектуры нейронных сетей доступны в их официальных документации или репозиториях. Понятно, что непустые матрицы весов могут быть только у предварительно обученных моделей.

Вот 10 наиболее популярных нейронных сетей краткая характеристика каждой:

<http://ej.kubagro.ru/2024/05/pdf/12.pdf>

1. Нейронная сеть прямого распространения: Простой тип нейронной сети, в которой информация передается только в одном направлении, от входных узлов к выходным.

2. Сверточная нейронная сеть: Специально разработана для анализа изображений, использует операцию свертки для обнаружения локальных шаблонов.

3. Рекуррентная нейронная сеть: Способна учитывать последовательность данных благодаря внутренней памяти, которая сохраняет информацию о предыдущих состояниях.

4. Сеть LSTM: Тип рекуррентной нейронной сети, способный эффективно обрабатывать и моделировать долгосрочные зависимости в данных.

5. Сеть GRU: Обновленная версия рекуррентной нейронной сети с воротными механизмами, более эффективна в обучении на длинных последовательностях данных.

6. Сеть с архитектурой ResNet: Глубокая сверточная нейронная сеть с особым акцентом на передаче информации 'в глубину', использует блоки с остаточным соединением.

7. Сеть с архитектурой Inception: Имеет множество параллельных подсетей, каждая из которых предназначена для обнаружения шаблонов разного масштаба и уровня детализации.

8. Сеть с архитектурой VGG: Глубокая сверточная нейронная сеть с простой и однородной структурой, широко используется в обработке изображений и классификации.

9. Сеть с архитектурой MobileNet: Легковесная сверточная нейронная сеть, оптимизированная для работы на мобильных устройствах с ограниченными вычислительными ресурсами.

10. Сеть с архитектурой DenseNet: Каждый слой принимает входные данные от всех предыдущих слоев в блоке, способствует повышению градиентного потока и облегчает обучение.

Для загрузки и извлечения матрицы весов предварительно обученных моделей, можно использовать такие библиотеки как TensorFlow или PyTorch, чтобы загрузить модели и извлечь матрицы весов.

Вот пример того, как это можно сделать для библиотеки TensorFlow:

```
import tensorflow as tf

# Загрузка предварительно обученной модели (например, VGG16)
model = tf.keras.applications.VGG16(weights='imagenet', include_top=True)

# Получение матрицы весов
weights_matrix = model.get_weights()[0]

# Вывод размерности матрицы весов
print("Размерность матрицы весов:", weights_matrix.shape)

# Вывод всей матрицы весов
print("Матрица весов:")
print(weights_matrix)
```

Аналогично можно использовать библиотеку PyTorch для загрузки модели и извлечения матрицы весов:

```
import torch
import torchvision.models as models

# Загрузка предварительно обученной модели (например, ResNet50)
model = models.resnet50(pretrained=True)

# Получение матрицы весов
weights_matrix = model.state_dict()["conv1.weight"]

# Вывод размерности матрицы весов
print("Размерность матрицы весов:", weights_matrix.shape)

# Вывод всей матрицы весов
print("Матрица весов:")
print(weights_matrix)
```

Для извлечения матриц весов других моделей и фреймворков нужно провести аналогичные действия, используя соответствующие инструменты и библиотеки. После извлечения матриц весов для всех наиболее популярных

нейронных сетей, можно провести анализ и решение различных задач на основе этих данных.

Вы можете визуализировать все слои нейронной сети с их весовыми коэффициентами на одном графическом файле, используя различные методы визуализации. Одним из таких методов является использование графовых библиотек, таких как Graphviz, для отображения структуры нейронной сети, а также представления весов в виде цветовой карты или других визуальных атрибутов.

4. Обсуждение

Система «Эйдос» имеет 6 автоматизированных программных интерфейсов для ввода в систему исходных данных различных типов: табличных числовых и текстовых, текстовых файлов, графических файлов. На основе этих исходных данных она формирует системно-когнитивные модели, отражающие влияние значений факторов на переход объекта моделирования в будущие состояния, соответствующие классам. Эти модели имеют вид матриц, в которых значения факторов соответствуют строкам, классы - колонкам, в клетках на их пересечении находятся коэффициенты взаимосвязи между ними.

4.1. Использование матричных системно-когнитивных моделей в качестве обучающей выборки для нейросетей

Матричные системно-когнитивные модели создаются системой «Эйдос» непосредственно на основе эмпирических данных и представляют собой методы анализа данных, в результате которых данные преобразуются в информацию, а она в знания. Эти модели представляют собой матрицы, где колонки соответствуют классам, а строки признакам. Такие модели могут использоваться для представления сложных систем и процессов, а также для

анализа и прогнозирования. Использование матричных системно-когнитивных моделей в качестве обучающей выборки для нейронных сетей вполне возможно и может быть полезным.

Нейронные сети, с другой стороны, являются мощным инструментом машинного обучения, который может обрабатывать большие объемы данных и находить сложные зависимости между входными и выходными данными.

Используя матричные системно-когнитивные модели в качестве обучающей выборки для нейронных сетей, вы можете:

1. Преобразовывать данные в формат, понятный нейронным сетям. Например, вы можете использовать выделенные из матриц признаки как входные данные для нейронной сети.

2. Использовать результаты матричных моделей как признаки для обучения нейронной сети. Например, вы можете использовать выходные значения матрицы как целевую переменную для обучения нейронной сети.

3. Объединять матричные модели с нейронными сетями для создания более мощных и точных моделей. Например, можно использовать результаты матричного анализа как входные данные для нейронной сети, которая затем улучшает предсказания с помощью обучения на этих данных.

Важно правильно подготовить данные и выбрать подходящую архитектуру нейронной сети для вашей задачи. При этом необходимо учитывать размер и структуру данных, чтобы эффективно использовать матричные системно-когнитивные модели в качестве обучающей выборки для нейронных сетей.

В результате использования системно-когнитивных моделей в качестве обучающей выборки для нейронных сетей с последующим извлечением матриц весов этих нейронных сетей может осуществлено обновление

коэффициентов взаимосвязи в исходных матрицах на основе результатов, полученных от нейронной сети.

Но возникает закономерный вопрос о том, для каждой ли нейронной сети возможно представление их весовых матриц в виде матриц системно-когнитивных моделей? Не для всех типов нейронных сетей будет возможно представить результаты в виде матриц такой же структуры, как системно-когнитивные модели. Однако, с учетом определенных подходов и техник, вы можете приблизить результаты работы нейронной сети к матричным форматам моделей системы «Эйдос».

Вот несколько способов, как это можно сделать:

1. Использование нейронных сетей с фиксированным числом выходов: Можно использовать нейронную сеть, у которой число выходов соответствует числу классов модели системы «Эйдос». В этом случае выходы нейронной сети могут быть представлены в виде вектора, который можно интерпретировать как строку матрицы.

2. Преобразование выходов нейронной сети в матрицу: В случае, если нейронная сеть имеет переменное количество выходов или выходы, которые не соответствуют прямому представлению в виде матрицы, можно использовать некоторые алгоритмы преобразования. Например, можно преобразовать вектор выходов нейронной сети в матрицу с использованием кластерного анализа или других методов машинного обучения.

3. Использование специальных архитектур нейронных сетей: Некоторые архитектуры нейронных сетей, такие как рекуррентные нейронные сети с выходным слоем, способны работать с данными в матричной форме. Целесообразно исследовать возможность применения таких архитектур для решения проблемы, рассматриваемой в статье.

4. Разработка собственной архитектуры: Автором разработана и в 2003 году предложена собственная архитектура нейронной сети, которая полностью совместима с матричными системно-когнитивными моделями.

4.2. Типы нейросетей, допускающие представление коэффициентов взаимосвязи между значениями рецепторов и нейронами в виде матриц, аналогичных системно-когнитивным моделям

Для представления коэффициентов взаимосвязи между значениями рецепторов и нейронами в виде матриц, аналогичных системно-когнитивным моделям, в наибольшей степени подходят следующие типы нейронных сетей:

1. Многослойный перцептрон (MLP): MLP состоит из одного или нескольких скрытых слоев, каждый из которых полностью связан с предыдущим и последующим слоями. Каждая связь между нейронами в сети имеет свой вес, который можно представить в виде матрицы. Веса определяют коэффициенты взаимосвязи между нейронами.

2. Сверточные нейронные сети (CNN): В CNN используются фильтры, которые применяются к различным областям входных данных. Веса этих фильтров формируются в процессе обучения и могут быть представлены в виде матриц. Каждый фильтр может рассматриваться как набор коэффициентов, описывающих взаимосвязь между рецепторами и нейронами в скрытых слоях.

3. Рекуррентные нейронные сети (RNN): В RNN каждый нейрон в слое принимает входные данные и состояние из предыдущего временного шага. Веса связей между нейронами формируются в процессе обучения и могут быть представлены в виде матрицы. Коэффициенты взаимосвязи между значениями рецепторов и нейронами хранятся в этих матрицах.

4. Long Short-Term Memory (LSTM) и Gated Recurrent Unit (GRU): Это специальные типы рекуррентных нейронных сетей, которые способны лучше

улавливать долгосрочные зависимости в данных. Веса связей между нейронами в LSTM и GRU также могут быть представлены в виде матриц.

Эти типы нейронных сетей обладают свойством иметь параметры, которые могут быть представлены в матричной форме, что делает их подходящими для моделирования системно-когнитивных взаимосвязей аналогично вашим системно-когнитивным моделям.

4.3. Разработка программы на языке Питон, улучшающей системно-когнитивные модели системы «Эйдос» с помощью нейросетевых технологий

На Python можно написать программу для улучшения системно-когнитивных моделей с использованием нейросетевых технологий. Вот общий план того, как это можно сделать:

1. Подготовка данных:

- Преобразование существующих системно-когнитивных моделей в формат данных, который можно использовать для обучения нейронной сети. Это может включать в себя создание тренировочного набора данных, где каждая строка представляет значения факторов, а каждый столбец - класс объекта, а значение в ячейке - коэффициент взаимосвязи.

- Разделение данных на обучающий и тестовый наборы.

2. Выбор модели нейронной сети:

- Выбор архитектуры нейронной сети, которая будет обучаться на ваших данных. Это может быть, например, многослойный персептрон (MLP), сверточная нейронная сеть (CNN) или рекуррентная нейронная сеть (RNN), в зависимости от структуры и особенностей ваших данных.

3. Обучение нейронной сети:

- Использование обучающего набора данных для обучения выбранной модели нейронной сети.

- Определение оптимальных параметров обучения, таких как скорость обучения, количество эпох и размер пакета.

4. Оценка и тестирование модели:

- Оценка производительности обученной модели с использованием тестового набора данных.

- Анализ результатов и, если необходимо, доработка модели.

5. Использование модели для улучшения существующих моделей:

- Использование обученной нейронной сети для улучшения существующих системно-когнитивных моделей.

- Это может включать в себя обновление коэффициентов взаимосвязи в исходных матрицах на основе результатов, полученных от нейронной сети.

6. Тестирование и валидация:

- Проверка улучшенных моделей на тестовых данных.

- Валидация результатов и анализ их эффективности.

Вот как выглядит общий алгоритм. Конечно, реализация каждого из этих шагов может потребовать дополнительной детализации и настройки в зависимости от конкретных характеристик ваших данных и целей моделирования.

5. Выводы

В настоящей статье рассмотрены перспективы использования системно-когнитивных нейросетей (СК-нейросетей) в рамках системы "Эйдос" и их применение в экономике. Основываясь на классических моделях системы "Эйдос", автор предлагает использовать их в качестве обучающей выборки для создания нейросетевых моделей. Такой подход позволяет генерировать СК-нейросети, которые могут заменить классические модели "Эйдос". Результат обучения нейросетей может быть представлен в виде матричной модели, аналогичной структуре классических моделей, что

позволяет использовать функционал системы "Эйдос" на основе СК-нейросетевых моделей. Это открывает новые перспективы для развития системы "Эйдос" и ее применения в экономике.

Дополнительные перспективы развития включают развитие системы на современных языках программирования, таких как Python и C++, что даст возможность более эффективного и гибкого использования системы "Эйдос" в практических задачах.

Таким образом, системно-когнитивные нейросети представляют собой важное направление развития системы "Эйдос" и могут значительно расширить ее функционал, улучшив эффективность и точность решений в экономических задачах.

Литература

1. Луценко Е.В. Автоматизированный системно-когнитивный анализ влияния состава бетона на его физико-механические свойства и стоимость / Е.В. Луценко // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета (Научный журнал КубГАУ) [Электронный ресурс]. – Краснодар: КубГАУ, 2023. – №07(191). С. 31 – 50. – IDA [article ID]: 1912307009. – Режим доступа: <http://ej.kubagro.ru/2023/07/pdf/09.pdf>, 1,25 у.п.л. EDN: AEFBLJ
2. Луценко, Е. В. Когнитивные функции как обобщение классического понятия функциональной зависимости на основе теории информации в АСК-анализе и системной нечеткой интервальной математике / Е. В. Луценко, А. И. Орлов // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. – 2014. – № 95. – С. 58-81. – EDN RVEYDF.
3. Луценко, Е. В. Хиршамания при оценке результатов научной деятельности, ее негативные последствия и попытка их преодоления с применением многокритериального подхода и теории информации / Е. В. Луценко // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. – 2015. – № 108. – С. 1-29. – EDN TROLXF.
4. Луценко, Е. В. Интенсивные технологии возделывания плодовых культур / Е. А. Егоров, А. Н. Фисенко, Ж. А. Шадрин [и др.]. – Краснодар, 2004. – 394 с. – ISBN 5-98272-008-9. – EDN QCGJDD.
5. Луценко, Е. В. Инвариантное относительно объемов данных нечеткое мультиклассовое обобщение F-меры достоверности моделей Ван Ризбергена в АСК-анализе и системе "Эйдос" / Е. В. Луценко // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. – 2017. – № 126. – С. 1-32. – DOI 10.21515/1990-4665-126-001. – EDN XXXBDV.

6. Луценко, Е. В. Синтез адаптивных интеллектуальных измерительных систем с применением АСК-анализа и системы "Эйдос", системная идентификация в эконометрике, биометрии, экологии, педагогике, психологии и медицине / Е. В. Луценко // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. – 2016. – № 116. – С. 1-60. – EDN VQUVHJ.

7. Лойко, В. И. Инвестиционно-ресурсное управление сельскохозяйственным производством / В. И. Лойко, Т. П. Барановская, Е. В. Луценко // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. – 2012. – № 83. – С. 563-595. – EDN PJVOPF.

8. Луценко, Е. В. Критерии реальности и принцип эквивалентности виртуальной и "истинной" реальности / Е. В. Луценко // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. – 2004. – № 8. – С. 9-27. – EDN JWXNGV.

References

1. Lucenko E.V. Avtomatizirovannyj sistemno-kognitivnyj analiz vliyaniya sostava betona na ego fiziko-mexanicheskie svojstva i stoimost' / E.V. Lucenko // Politematicheskij setевой e`lektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta (Nauchnyj zhurnal KubGAU) [E`lektronnyj resurs]. – Krasnodar: KubGAU, 2023. – №07(191). S. 31 – 50. – IDA [article ID]: 1912307009. – Rezhim dostupa: <http://ej.kubagro.ru/2023/07/pdf/09.pdf>, 1,25 u.p.l. EDN: AEFBLJ

2. Lucenko, E. V. Kognitivny`e funkcii kak obobshhenie klassicheskogo ponyatiya funkcional`noj zavisimosti na osnove teorii informacii v ASK-analize i sistemnoj nechetkoj interval`noj matematike / E. V. Lucenko, A. I. Orlov // Politematicheskij setевой e`lektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta. – 2014. – № 95. – S. 58-81. – EDN RVEYDF.

3. Lucenko, E. V. Xirshamaniya pri ocenke rezul'tatov nauchnoj deyatel`nosti, ee negativny`e posledstviya i popytka ix preodoleniya s primeneniem mnogokriterial`nogo podxoda i teorii informacii / E. V. Lucenko // Politematicheskij setевой e`lektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta. – 2015. – № 108. – S. 1-29. – EDN TROLXF.

4. Lucenko, E. V. Intensivny`e texnologii vozdely`vaniya plodovy`x kul`tur / E. A. Egorov, A. N. Fisenko, Zh. A. Shadrina [i dr.]. – Krasnodar, 2004. – 394 s. – ISBN 5-98272-008-9. – EDN QCGJDD.

5. Lucenko, E. V. Invariantnoe otnositel`no ob`emov danny`x nechetkoe mul`tiklassovoe obobshhenie F-mery` dostovernosti modelej Van Rizbergena v ASK-analize i sisteme "E`jdos" / E. V. Lucenko // Politematicheskij setевой e`lektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta. – 2017. – № 126. – S. 1-32. – DOI 10.21515/1990-4665-126-001. – EDN XXXBDV.

6. Lucenko, E. V. Sintez adaptivny`x intellektual`ny`x izmeritel`ny`x sistem s primeneniem ASK-analiza i sistemy` "E`jdos", sistemnaya identifikaciya v e`konometrike, biometrii, e`kologii, pedagogike, psixologii i medicine / E. V. Lucenko // Politematicheskij setевой e`lektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta. – 2016. – № 116. – S. 1-60. – EDN VQUVHJ.

7. Lojko, V. I. Investicionno-resursnoe upravlenie sel`skoxozyajstvenny`m proizvodstvom / V. I. Lojko, T. P. Baranovskaya, E. V. Lucenko // Politematicheskij setевой e`lektronnyj

nauchny`j zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta. – 2012. – № 83. – S. 563-595. – EDN PJVOPF.

8. Lucenko, E. V. Kriterii real`nosti i princip e`kvivalentnosti virtual`noj i "istinnoj" real`nosti / E. V. Lucenko // Politematicheskij setevoj e`lektronny`j nauchny`j zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta. – 2004. – № 8. – S. 9-27. – EDN JWXNGV.