УДК 004.896; 004.514.64 doi: 10.15827/0236-235X.151.459-467 2025. Т. 38. № 3. С. 459–467

Использование многослойной нейронной сети для построения адаптивного интерфейса системы автоматизированного проектирования КОМПАС-3D

А.В. Субботин [1]✉, Т.М. Зубкова [2]

[1] Оренбургский государственный университет, Аэрокосмический институт, г. Оренбург, 460018, Россия
[2] Оренбургский государственный университет, Институт математики и информационных технологий, г. Оренбург, 460018, Россия

Ссылка для цитирования
Субботин А.В., Зубкова Т.М. Использование многослойной нейронной сети для построения адаптивного интерфейса системы автоматизированного проектирования КОМПАС-3D // Программные продукты и системы. 2025. Т. 38. № 3. С. 459–467. doi: 10.15827/0236-235X.151.459-467
Информация о статье
Группа специальностей ВАК: 2.3.7
Поступила в редакцию: 21.10.2024 После доработки: 25.12.2024 Принята к публикации: 16.01.2025

Аннотация. Исследование посвящается проблеме адаптации интерфейсов в САПР с целью повышения эффективности работы инженера-конструктора при проектировании машиностроительных изделий на производстве. В данной статье рассматривается подход по адаптации интерфейса САПР КОМПАС-3D под задачи проектировщика. В качестве задачи выступает набор данных, описывающих будущую деталь. Новизной предложенного подхода является применение многослойной нейросети для анализа взаимосвязей между входными переменными, описывающими геометрические характеристики детали, и командами, используемыми в процессе построения ее трехмерной модели. Обучение нейросети проводилось на основе трехмерных моделей из библиотеки стандартных изделий. Результатом работы нейросети является набор рекомендаций, состоящих из необходимых команд, которые потребуются проектировщику для построения модели детали. Разработано программное средство, позволяющее на основе рекомендаций нейросети сформировать интерфейс с инструментальной панелью, который располагается поверх окна системы КОМПАС-3D. Отличительной особенностью данного интерфейса является гибкость в настройке. Программное средство позволяет расширять базу готовых проектов деталей и переобучать нейронную сеть для повышения точности классификации. Взаимодействие программного средства и системы КОМПАС-3D реализовано с помощью набора для разработки ПО Software Development Kit и языка программирования Python. Внедрение разработанного программного средства в рабочую среду инженера-конструктора позволит оптимизировать его труд и повысить производительность с помощью адаптивного интерфейса можно уменьшить количество команд и время их поиска в интерфейсной части САПР. Таким образом, использование нейросети в адаптации интерфейса позволит повысить эффективность взаимодействия с системой КОМПАС-3D. Кроме того, обеспечит более простое и интуитивно понятное рабочее пространство для проектировщика.
Ключевые слова: многослойная нейронная сеть, интерфейс, адаптивный интерфейс, рабочее пространство проектирования, САПР, КОМПАС-3D

Введение. С возрастанием сложности и многофункциональности ПО усложняется его интерфейсная часть. Часто для программного продукта разрабатывается стандартный интерфейс с ограниченными возможностями настройки компонентов. Это не позволяет пользователю адаптировать его под свои нужды и особенности.

Упростить взаимодействие с программными средствами пытаются с помощью разработки различных методик и подходов на базе ИИ. Предлагается проектирование адаптивных пользовательских интерфейсов, основанных на знаниях и онтологических подходах. В таких исследованиях учитывают специфику предметной области и изменения в пользовательском контексте [1, 2], в том числе для людей с ограниченными возможностями [3]. Также на основе онтологического подхода ведутся исследования для персонализации интерфейса мобильных устройств на основе четырех контекстов: пользователь, устройство, задача и среда [4]. В работе [5] использовалось машинное обучение для определения полезности и эффективности адаптивных интерфейсов, а также сделана попытка выявления точной модели пользовательских предпочтений.

В разработке адаптивных интерфейсов применяются концепции интеллектуального пространства, основанного на потребностях пользователя и на специфике задачи [6]. В качестве решения проблемы изменчивости контекста в реальном времени предлагается гибкая настройка отдельных виджетов интерфейса с помощью внедрения адаптивных идентификаторов [7]. Помимо ситуационного контекста, для адаптации интерфейса разрабатываются модели пользователя, которые учитывают его индивидуальные характеристики [8]. Нейронные сети используются и для формирования адаптивных компонентов интерфейса на основе пользовательских характеристик [9].

В сложных системах, таких как САПР, тоже ведутся исследования по адаптации интерфейса. Например, в работе [10] интерфейс адаптируется под характеристики каждого проектировщика, а в программе [11] формируются персональные рекомендации по использованию команд на основе накопленного опыта.

Недостатком всех имеющихся инструментов адаптации интерфейсов является низкая производительность пользователя при выполнении конкретных задач в определенной программной системе. Этот недостаток характерен и для интерфейса САПР КОМПАС-3D [12]. Интерфейс и необходимые команды регулируются по виду задачи: деталь, сборка, чертеж, фрагмент, текстовый документ, спецификация. Все предоставляемые интерфейсные элементы являются стандартными, и для их изменения требуется ручная настройка.

Постановка задачи

При работе с системой КОМПАС-3D были выделены основные элементы интерфейса: таблица с переменными, дерево построения модели, инструментальная панель, содержащая команды для построения детали.

Перед началом моделирования различные переменные, которые характеризуют будущую модель, создаются или загружаются в документ проекта. Например, документ PLib_Arm_GOST_17375_30753_712.m3d (рис. 1) из библиотеки стандартных изделий имеет четыре переменные (наружный диаметр торцов углов, радиус кривизны осевой линии, толщина стенки, угол) и три операции в дереве построения (начало координат, эскиз, операция вращения). Для построения данной детали система КОМПАС-3D предоставляет стандартный интерфейс с наборами и панелями команд (инструментальная панель). При применении трех операций проектировщику необходимо тратить время на поиск этих команд среди множества предоставленных. Так, в стандартном наборе «Твердотельное моделирование» насчитывается более 50 команд-элементов, многие из которых являются составными, то есть включают в себя еще ряд команд.

Таким образом, основной целью исследования является построение интерфейса с отсутствием лишних команд до того момента, как проектировщик приступает к моделированию детали.

Исходными данными для исследования выступают модели деталей из библиотеки стандартных изделий КОМПАС-3D.

Разработка модели нейронной сети для определения необходимых команд на инструментальной панели ПО

Поскольку исследование предполагает выявление скрытых зависимостей между переменными и командами построения трехмерной модели изделия, то наиболее подходящим методом для реализации математического аппарата будет использование нейронных сетей и глубокого обучения. Авторы выбрали многослойную нейронную сеть прямого распространения на основе персептрона, так как данный вид сетей часто используют для анализа данных и построения классификаторов. Архитектура таких сетей (рис. 2) состоит из входного и выходного слоев, неограниченного количества скрытых слоев [13].

Сеть получает входные сигналы из множества X, которые затем обрабатываются путем

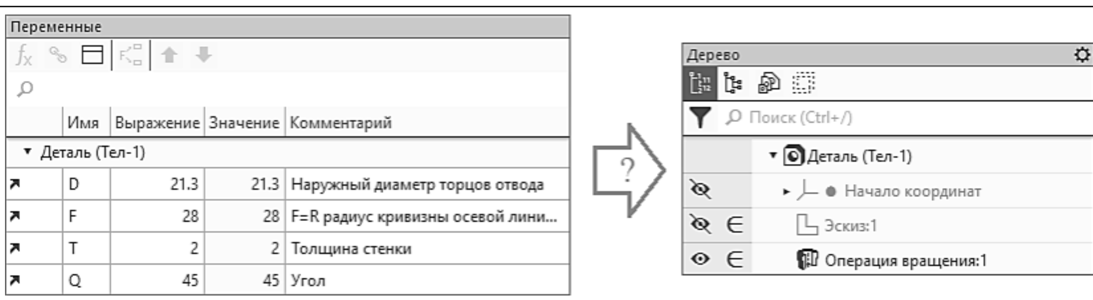

Рис. 1. Переменные и дерево построения для детали в КОМПАС-3D

Fig. 1. Variables and a construction tree for a part in KOMPAS-3D

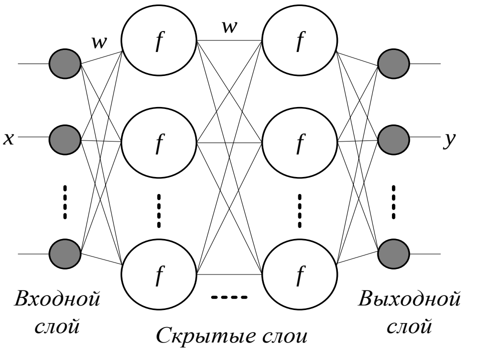

Рис. 2. Общая схема многослойной нейронной сети

Fig. 2. General scheme of a multilayer neural network

умножения каждого сигнала на соответствующий синаптический вес и добавления порогового значения. Все результаты выполненных операций для каждого x попадают в нейрон, где обрабатываются функцией активации f, а результатом является выходное значение Y. Описанные действия проводятся для каждого входного значения и всех нейронов. Математически выглядит это следующим образом:

$$u_{mk} = \sum_{i}^{n} w_{mki} x_i, \quad (1)$$

$$y_k = f(u_{mk} + b_{mk}), \quad (2)$$

где m – количество скрытых слоев сети (1, .., M); k – количество нейронов в слое (1, .., K); n – количество сигналов, входящих в нейрон (1, .., N); w – синаптический вес; x – входной сигнал (1, .., X), f – функция активации нейрона; b – пороговое значение; y – выходной сигнал (1, .., Y). Если скрытых слоев больше 1, то для второго и последующих слоев в формуле 1 вместо x_i используется y_k из предыдущих слоев.

Функция активации рассчитывает значение выходного нейрона. Для решения конкретно поставленной задачи подбирается необходимая функция активации, которая зависит от структуры нейронной сети и обрабатываемых данных. В рамках исследования были применены функции *relu* и *softmax*:

$$relu(x) = \max(0, x), \quad (3)$$

$$softmax(x) = \frac{e^{x_i}}{\sum_{i=1}^{X} e^{x_i}}. \quad (4)$$

Функция *relu* выбрана из-за того, что она позволяет избавиться от отрицательных чисел. На графике она выглядит как линейная, но наличие производной функции дает возможность использовать обратное распространение. Функция *softmax* нормализует входной вектор значений в распределение вероятностей, пропорциональных экспонентам входных значений. Данная функция получила широкое применение в выходных слоях для расчета конечных результатов. Учитывая многовариантную возможность системы КОМПАС-3D при построении одной и той же детали, функция *softmax* позволит определить вероятностный характер использования команд (выходных значений) для построения детали. Важным фактором выбора функций активации являются экспериментальные данные, полученные в ходе обучения и анализа работы сети.

Структура разработанной нейронной сети представлена на рисунке 3.

Обучение нейронной сети подразумевает процесс настройки весовых коэффициентов. Каждый нейрон, включенный в сеть, вносит вклад в обучение, так как от него зависит, будет ли активирован сигнал. Поэтому при решении различных задач требуется тонкая настрой-

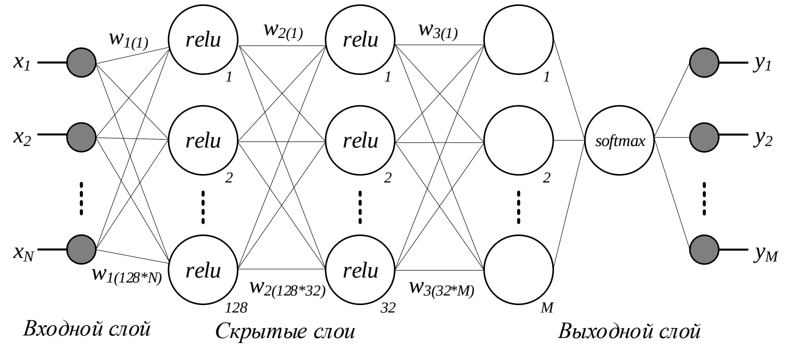

Рис. 3. Структура нейронной сети системы подбора команд для построения детали в КОМПАС-3D

Fig. 3. Structure of the neural network for team selection system in KOMPAS-3D part modeling

ка структуры сети (количество слоев, нейронов).

В рамках исследования применяется метод обучения с учителем, так как имеются эталонные выходные данные, а корректировка синаптических весов производилась с использованием метода обратного распространения ошибки.

Таким образом, для решения поставленной задачи исследования требуется обучить сеть получать настроенную матрицу весовых коэффициентов с целью их применения в модели сети для определения необходимых команд при построении детали машиностроительного изделия в системе КОМПАС-3D. Процесс обучения нейронной сети представлен в виде диаграммы деятельности (рис. 4).

Как видно из диаграммы, сначала требуется подготовить данные обучающей выборки, для этого был разработан скрипт, позволяющий считать из файлов с расширением *.m3d все переменные и команды из дерева построения. Вычисляются массив уникальных переменных (uX) и массив команд (uY) по всем файлам. Далее все входные (X) и выходные данные (Y) представляются в виде структуры, которая приемлема для обучения сети. То есть для переменных формируется массив на основе uX, где 1 – наличие определенной переменной, 0 – от-

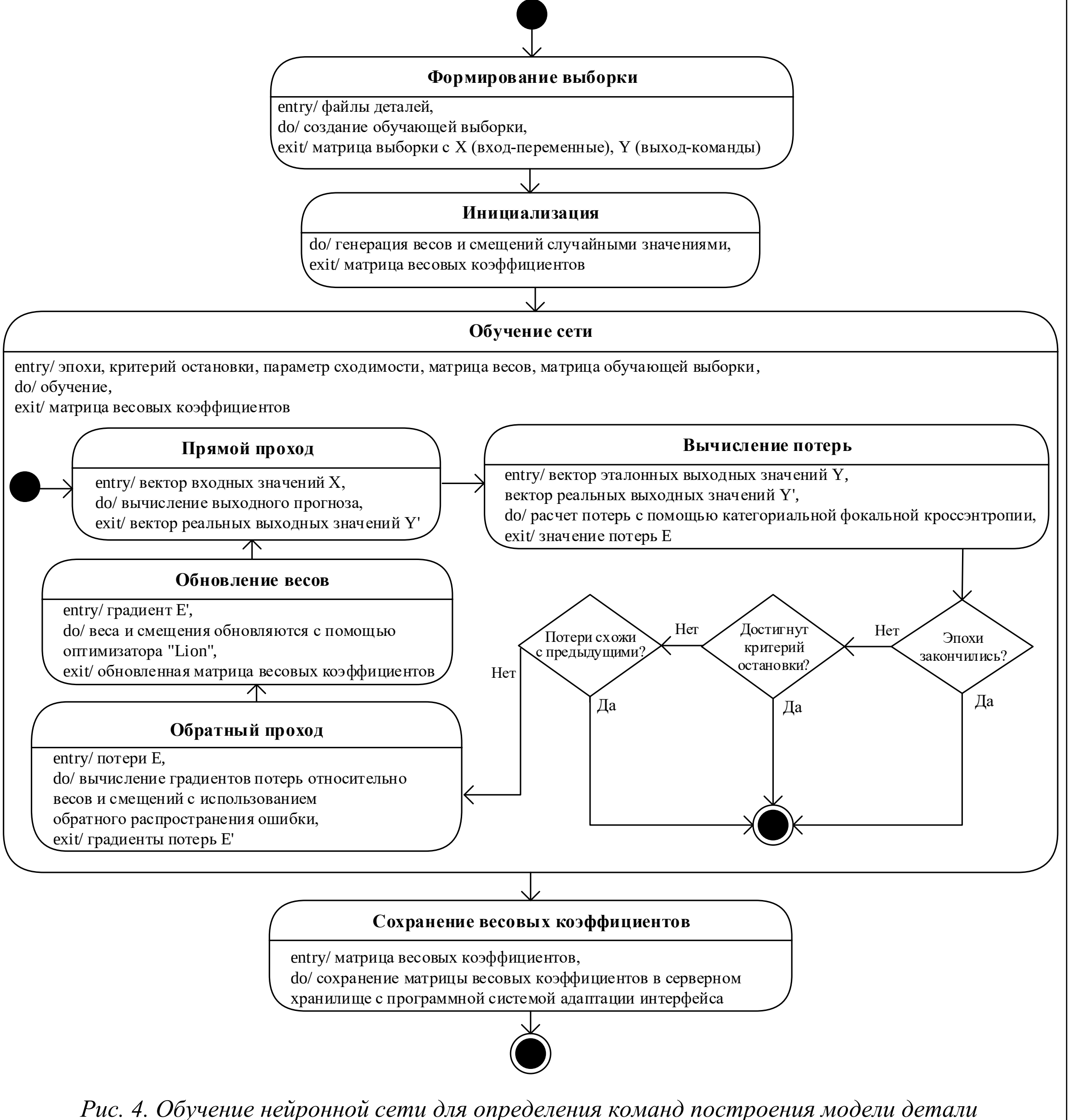

Рис. 4. Обучение нейронной сети для определения команд построения модели детали

Fig. 4. Training a neural network to determine commands for building a part model

сутствие (рис. 5). Аналогично формируется и выходная структура данных.

x_i			var3	var1	var7	var8		
uX	var1	var2	var3	var4	var5	var6	var7	var8
x_i	1	0	1	0	0	0	1	1

Рис. 5. Пример преобразования входных данных во входной вектор для нейросети

Fig. 5. Example of converting input data into an input vector for a neural network

После подготовки данных инициализируется матрица весовых коэффициентов, запускается процесс обучения. Входной вектор делает прямой проход по сети, получая вектор с реальными значениями, которые сравниваются с вектором эталонных значений, и высчитывается ошибка сети (потери). Далее с помощью алгоритма обратного распространения ошибки вычисляются градиенты потерь для правильной оптимизации весов (уменьшение или увеличение). Обучение заканчивается по истечении некоторого количества эпох, если определен критерий остановки (процент допустимой ошибки) или если сеть из эпохи в эпоху выдает одинаковые потери. Отличительными элементами при обучении являются:

– вычисление потерь с помощью категориальной фокальной кроссэнтропии. Она применяется к плохо классифицированным обучающим выборкам и игнорирует хорошо классифицированные выборки;

– обновление весов осуществляется с использованием оптимизатора Lion – метода стохастического градиентного спуска, где для управления величиной обновления используется оператор знака. Этот оптимизатор экономит память, поэтому является наиболее эффективным.

После обучения сети рекомендуется удостовериться в ее точности на проверочной выборке, то есть на данных, которые не входили в обучающую выборку, но имеют схожую структуру и среду происхождения (детали).

При работе сети выходными данными являются вероятности использования команд, и суммой всех вероятностей является 1 (100 %). Эталонные величины представлены как 1 (наличие) или 0 (отсутствие). Для более точного определения потерь сеть использовала преобразование, подразумевающее изменение всех выходов по выражению

$$Y_i^{'} = \begin{cases} 0, при Y_i^{'} < 0,01, \\ 1, остальные\_случаи. \end{cases}$$

Оценка работы обученной нейронной сети показала, что точность правильной классификации равна 92 % (ошибка сети 8 %). При программном анализе результатов был установлен факт наличия таких потерь. Аргументируется это многовариантностью построения одной и той же детали (рис. 6) или схожестью построения разнородных деталей. Нейронная сеть давала правильные результаты для всех примеров, соответствующих реальным и эталонным значениям векторов, а также в некоторых слу-

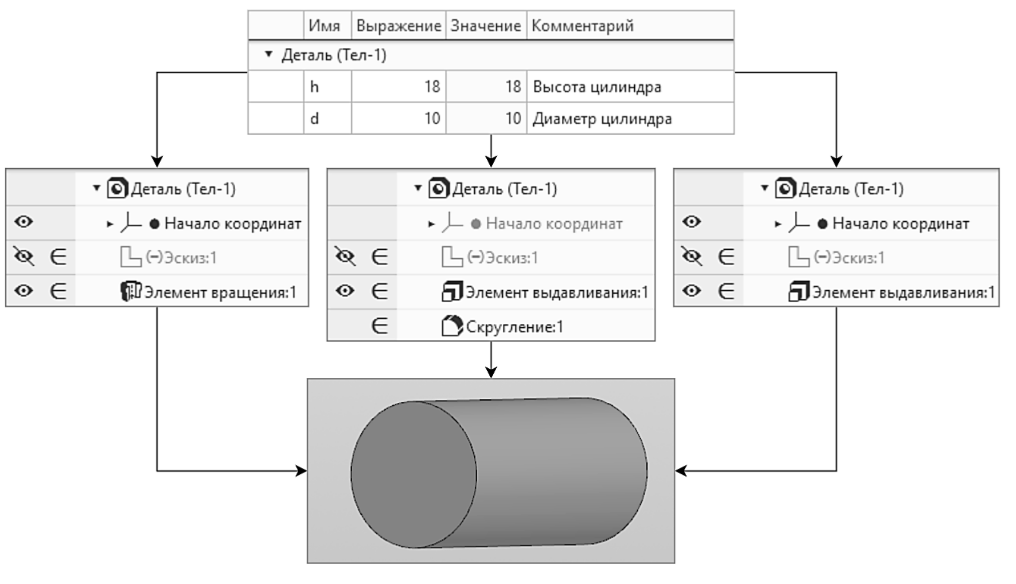

Рис. 6. Пример многовариантности построения цилиндра

Fig. 6. Example of the cylinder construction multivariance

чаях добавляла несколько лишних команд (в среднем 2–3), отличных от эталонных значений. Поэтому построенную модель и результаты ее обучения можно считать успешными.

Разработка нейросетевой системы определения команд для построения детали

На основе разработанной нейронной сети по определению необходимых команд для моделирования детали создано программное средство по формированию адаптивного интерфейса для системы КОМПАС-3D.

Для разработки выбран набор инструментов SDK (*Software Development Kit*) и язык программирования Python, поскольку он поддерживает API КОМПАС-3D и является стандартным для создания макросов в этой системе проектирования.

Разработанное ПО имеет уровни доступа для проектировщика и для администратора, которые отличается назначением и функционалом. Администратор может работать с программой независимо от КОМПАС-3D, а проектировщик должен работать в рамках уже активной системы трехмерного моделирования.

Проектировщику доступны следующие функции:

– создание новых переменных (характеризующих деталь);

– выбор существующих переменных;

– загрузка переменных в текущий документ, открытый в системе КОМПАС-3D, или выгрузка из него;

– получение рекомендаций по использованию команд для моделирования детали;

– формирование интерфейса на основе рекомендаций;

– оформление сравнительного отчета по рекомендуемым и использованным командам для отправки администратору.

При входе в систему с уровнем доступа «проектировщик» (при успешной идентификации и авторизации) загружается главное окно ПО, которое подгружает и заполняет данные из открытого и активного документа «Деталь». Если активный документ не имеет переменных, проектировщик заполняет их, пользуясь имеющейся базой переменных. После этого проектировщик может получить рекомендации на основе результатов работы нейросети по использованию команд и сформировать интерфейс. Если переменная, созданная проектировщиком, отсутствует в базе, то она не будет участвовать в формировании рекомендаций, и проектировщик получит предупреждение.

Далее можно приступать к моделированию трехмерной детали. После решения поставленной задачи проектировщик возвращается в программу для загрузки команд, которые были использованы (рис. 7), и для формирования отчета администратору.

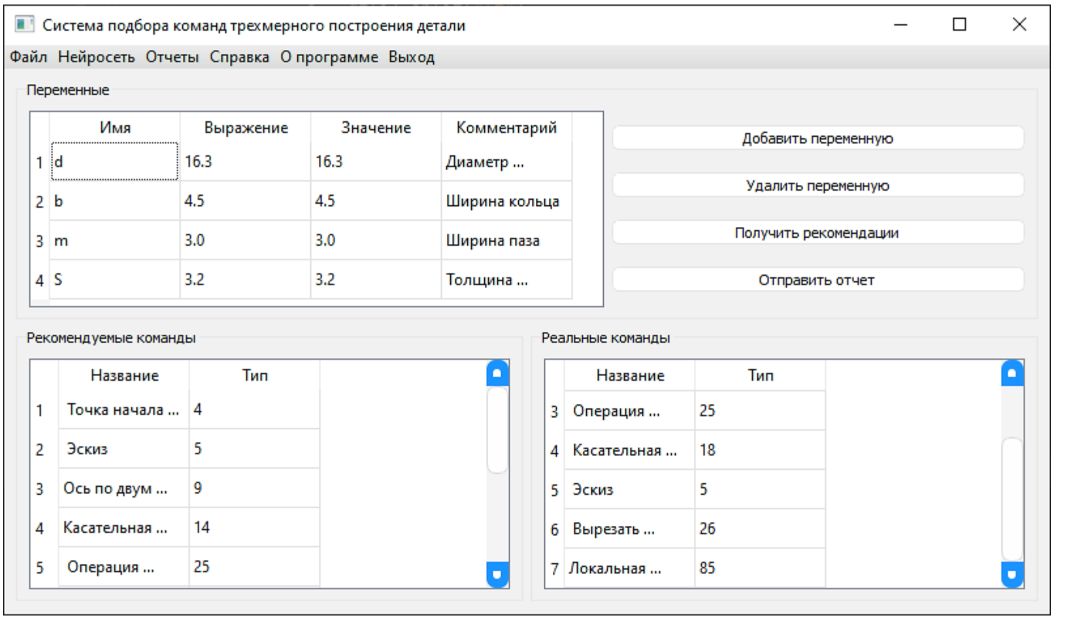

Рис. 7. Основное окно работы проектировщика

Fig. 7. Designer's main window

У сформированного адаптивного интерфейса есть существенные преимущества перед стандартным:

– уменьшение времени на поиск необходимой команды за счет сокращения их общего количества;

– более гибкая настройка, позволяющая применить другие настройки (например, изменение свойств компонентов [10]);

– отсутствие привязки к системе трехмерного моделирования, что позволяет размещать элементы интерфейса в любой части экрана (http://www.swsys.ru/uploaded/image/2025-3/16.jpg).

Преимущественным фактором использования данного ПО для инженера-конструктора является формирование определенной структуры построения деталей. С каждой новой деталью уменьшается вероятность появления новых переменных, а это повышает точность работы нейронной сети. В результате рекомендации по использованию определенного набора команд помогут стандартизировать процесс создания деталей. Это упростит работу проектировщика и повысит производительность его труда.

Администратор осуществляет контроль и занимается обновлением данных программы в условиях предприятия.

Работа в программе под уровнем доступа «администратор» позволяет обращаться к данным (файлам *.m3d) для анализа и формирования обучающей выборки (http://www.swsys.ru/uploaded/image/2025-3/3.jpg). Администратор может настраивать структуру и формировать процесс обучения нейронной сети, анализировать отчеты по результатам работы проектировщика с ПО. При работе с данными администратору доступны документы с деталями и параметры с деревом построения.

У администратора также имеется возможность контролировать количество уникальных переменных и используемых команд, от которых зависит формирование структуры данных для обучения нейронной сети. Администратор может переобучить сеть (если это необходимо), настраивая при этом параметры обучения (эпохи, потери). Если новая обученная сеть имеет худшие показатели, то администратор может вернуться к предыдущей версии. Для этого в БД программы хранятся все предыдущие версии обученной сети, и администратор может выбирать ту, которой будут пользоваться проектировщики (http://www.swsys.ru/uploaded/image/2025-3/4.jpg).

Администратору приходят отчеты о новых построенных деталях, он проверяет, как отработала нейронная сеть и какие данные при этом использовались проектировщиком. Если вводилась новая переменная, администратор проверяет ее на уникальность. Например, если проектировщик использовал существующую переменную, но определил ее по-другому, администратор может изменить ее на предыдущую версию. Изменения будут автоматически внесены в файл *.m3d. Такой подход позволяет поддерживать нормализованную структуру данных, повышая точность результатов работы нейросети.

Заключение

В данной статье описана разработка системы адаптации интерфейса КОМПАС-3D. Программная система основана на многослойной нейронной сети, позволяющей с помощью переменных, характеризующих будущую деталь, определять необходимые команды для построения трехмерной модели. Уникальной особенностью разработанной системы является адаптация интерфейса под конкретные задачи проектировщика, занимающегося трехмерным моделированием машиностроительных изделий.

В результате исследования представлена методика формирования интерфейса САПР, а также математическая модель для реализации. Алгоритм по определению необходимых команд представлен в виде нейронной сети, которая включает следующие этапы: подготовка обучающей выборки из файлов деталей формата *.m3d, настройка и обучение сети, проверка адекватности обученной сети. Получение исходных данных для обучения сети проводилось из библиотеки стандартных изделий с помощью написанного скрипта на языке Python и SDK КОМПАС-3D. В статье приведены примеры работы системы по формированию адаптивного интерфейса.

Основным преимуществом использования методики является уменьшение времени поиска необходимых команд на инструментальной панели. Интерфейс становится более персонализированным и служит подсказкой по использованию команд.

Перспективы дальнейшего развития предложенной методики и программного средства заключаются в охвате других видов документов КОМПАС-3D (чертеж, сборка) и в расширении функциональности для поддержки более сложных задач проектирования. Внедрение подобных решений в рабочие процессы инженеров-конструкторов может значительно повысить их эффективность и качество выполненной работы, обеспечив наиболее интуитивное и персонализированное взаимодействие с ПО. Таким образом, предложенная методика открывает новые возможности для оптимизации процессов проектирования и улучшения пользовательского опыта в инженерной среде.

Список литературы

1. Трегубов А.С. Разработка адаптивных контекстозависимых интерфейсов с использованием онтологических моделей // Кибернетика и программирование. 2017. № 6. С. 50–56. doi: 10.25136/2306-4196.2017.6.24747.
2. Грибова В.В., Паршкова С.В., Федорищев Л.А. Онтологии для разработки и генерации адаптивных пользовательских интерфейсов редакторов баз знаний // Онтология проектирования. 2022. Т. 12. № 2. С. 200–217. doi: 10.18287/2223-9537-2022-12-2-200-217.
3. Поцелуйко А.С. Персонализация интерфейсов мобильных приложений на основе онтологической модели для людей с ограниченными возможностями // Системный анализ в науке и образовании. 2020. № 4. С. 48–56. doi: 10.37005/2071-9612-2020-4-48-56.
4. Iqbal M.W., Naqvi M.R., Khan M.A., Khan F. Mobile devices interface adaptivity using ontologies. Comput., Materials & Continua, 2022, vol. 71, no. 3, pp. 4767–4784. doi: 10.32604/cmc.2022.023239.
5. Miraz M.H., Ali M., Excell P.S. Adaptive user interfaces and universal usability through plasticity of user interface design. Computer Sci. Review, 2021, vol. 40, art. 100363. doi: 10.1016/j.cosrev.2021.100363.
6. Миронов М.Д., Кашевник А.М. Адаптивный интерфейс на базе смартфона для системы управления шестиколесным мобильным роботом // Докл. ТУСУР. 2017. Т. 20. № 1. С. 110–118. doi: 10.21293/1818-0442-2017-20-1-110-118.
7. Yigitbas E., Josifovska K., Jovanovikj I. et al. Component-based development of adaptive user interfaces. Proc. ACM SIGCHI Symposium on EICS, 2019, no. 13, pp. 1–7. doi: 10.1145/3319499.3328229.
8. Фомина Т.А., Новикова Г.М. Проектирование адаптивного интерфейса ИС для поддержки деятельности образовательного учреждения // Вестн. Алтайской академии экономики и права. 2020. № 6. Ч. 1. С. 125–133. doi: 10.17513/vaael.1174.
9. Зубкова, Т.М., Тагирова Л.Ф. Нейросетевая инструментальная среда для создания персонализированных интерфейсов прикладных программ // Программные продукты и системы. 2023. Т. 36. № 2. С. 212–222. doi: 10.15827/0236-235X.142.212-222.
10. Тагирова Л.Ф., Субботин А.В., Зубкова Т.М. Программная инструментальная система создания адаптивных пользовательских интерфейсов // Науч.-технич. вестн. ИТМО. 2022. Т. 22. № 4. С. 751–759. doi: 10.17586/ 2226-1494-2022-22-4-751-759.
11. Куликов Д.Д. Программа адаптивного пользовательского интерфейса САПР: Свид. о регистр. ПрЭВМ № 2023619441. Рос. Федерация, 2023.
12. АСКОН. Системы проектирования КОМПАС-3D v22 (Система трехмерного моделирования КОМПАС-3D v22): Свид. о регистр. ПрЭВМ № 2023665204. Рос. Федерация, 2023.
13. Тормозов В.С., Василенко К.А., Золкин А.Л.. Настройка и обучение многослойного персептрона для задачи выделения дорожного покрытия на космических снимках города // Программные продукты и системы. 2020. Т. 33. № 2. С. 343–348. doi: 10.15827/0236-235X.130.343-348.

Software & Systems doi: 10.15827/0236-235X.151.459-467 2025, 38(3), pp. 459–467

Building an adaptive KOMPAS-3D CAD interface: A multilayer neural network

Andrey V. Subbotin [1]✉, Tatyana M. Zubkova [2]

[1] Orenburg State University, Aerospace Institute, Orenburg, 460018, Russian Federation
[2] Orenburg State University, Institute of Mathematics and Information Technology, Orenburg, 460018, Russian Federation

For citation
Subbotin, A.V., Zubkova, T.M. (2025) ‘Building an adaptive KOMPAS-3D CAD interface: A multilayer neural network’, *Software & Systems*, 38(3), pp. 459–467 (in Russ.). doi: 10.15827/0236-235X.151.459-467
Article info
Received: 21.10.2024 After revision: 25.12.2024 Accepted: 16.01.2025

Abstract. The paper focuses on interface adaptation in CAD systems to enhance design efficiency for mechanical engineers in manufacturing environments. It presents a CAD interface adaptation approach for designer-specific tasks in KOMPAS-3D.

The task is a set of data describing a future detail. The proposed approach's novelty is in employing a multilayer neural network to analyze relationships between input variables that describe part geometry and commands used during 3D model construction. Authors trained the neural network on 3D models from a standard parts library. The neural network generates a set of recommended commands that assist designers in creating part models. The developed software tool generates a neural network-driven toolbar interface that overlays the KOMPAS-3D CAD system window. A distinctive feature of this interface is the flexibility in configuration. The software tool enables expansion of the standard part database and retraining of the neural network to improve classification accuracy. The software tool integrates with KOMPAS-3D through its Software Development Kit (SDK) using Python. Deploying the developed software tool in mechanical engineers' workflows will optimize labor efficiency and boost productivity. The adaptive interface allows to reduce the number of commands and the time to search for them in the CAD interface. Thus, using a neural network to adapt the interface will increase the efficiency of interaction with KOMPAS-3D and provide a simpler and more intuitive workspace for a designer.
Keywords: multilayer neural network, interface, adaptive interface, design workspace, CAD, KOMPAS-3D

References

1. Tregubov, A.S. (2017) 'Development of adaptive context-dependent interfaces using ontological models', *Cybernetics and Programming*, (6), pp. 50–56 (in Russ.). doi: 10.25136/2306-4196.2017.6.24747.

2. Gribova, V.V., Parshkova, S.V., Fedorischev, L.A. (2022) 'Ontologies for development and generation adaptive user interfaces of knowledge base editors', *Ontology of Designing*, 12(2), pp. 200–217 (in Russ.). doi: 10.18287/2223-9537-2022- 12-2-200-217.

3. Potseluiko, A. S. (2020) 'Personalization of mobile application interfaces based on an ontological model for people with disabilities', *System Analysis in Sci. and Education*, (4), pp. 48–56 (in Russ.). doi: 10.37005/2071-9612-2020-4-48-56.

4. Iqbal, M.W., Naqvi, M.R., Khan, M.A., Khan, F. (2022) 'Mobile devices interface adaptivity using ontologies', *Computer, Materials & Continua*, 71(3), pp. 4767–4784. doi: 10.32604/cmc.2022.023239.

5. Miraz, M.H., Ali, M., Excell, P.S. (2021) 'Adaptive user interfaces and universal usability through plasticity of user interface design', *Computer Sci. Review*, 40, art. 100363. doi: 10.1016/j.cosrev.2021.100363.

6. Mironov, M.D., Kashevnik, A.M. (2017) 'Adaptive remote control interface based on smartphone application for six-wheeled mobile robot', *Proc. of TUSUR Univ.*, 20(1), pp. 110–118 (in Russ.). doi: 10.21293/1818-0442-2017-20-1-110-118.

7. Yigitbas, E., Josifovska, K., Jovanovikj, I. et al. (2019) 'Component-based development of adaptive user interfaces', *Proc. ACM SIGCHI Symposium on EICS*, (13), pp. 1–7. doi: 10.1145/3319499.3328229.

8. Fomina, T.A., Novikova, G.M. (2020) 'Designing an adaptive interface of an information system for support of educational institution's activities', *Bull. of the Altai Academy of Economics and Law*, (6), pt. 1, pp. 125–133 (in Russ.). doi: 10.17513/vaael.1174.

9. Zubkova, T.M., Tagirova, L.F. (2023) 'Neural network tool environment for creating adaptive application program interfaces', *Software & Systems*, 36(2), pp. 212–222 (in Russ.). doi: 10.15827/0236-235X.142.212-222.

10. Tagirova, L.F., Subbotin, A.V., Zubkova, T.M. (2022) 'Software development system for creation adaptive user interfaces', *Sci. Tech. J. Inf. Technol. Mech. Opt.*, 22(4), pp. 751–759 (in Russ.). doi: 10.17586/2226-1494-2022-22-4-751-759.

11. Kulikov, D.D. (2023) *Adaptive User Interface CAD Program*, Pat. RF, № 2023619441.

12. ASCON (2023) *Design Systems KOMPAS-3D v22 (Three-Dimensional Modeling System KOMPAS-3D v22)*, Pat. RF, № 2023665204.

13. Tormozov, V.S., Vasilenko, K.A., Zolkin, A.L. (2020) 'Setting up and training a multilayer perceptron for the problem of highlighting the road surface in the city space images', *Software & Systems*, 33(2), pp. 343–348 (in Russ.). doi: 10.15827/0236-235X.130.

Авторы
Субботин Андрей Владимирович [1], аспирант, aws1998@ya.ru
Зубкова Татьяна Михайловна [2], д.т.н, профессор, bars87@mail.ru

[1] Оренбургский государственный университет, Аэрокосмический институт, г. Оренбург,460018, Россия
[2] Оренбургский государственный университет, Институт математики и информационных технологий, г. Оренбург, 460018, Россия

Authors
Andrey V. Subbotin [1], Postgraduate Student, aws1998@ya.ru
Tatyana M. Zubkova [2], Dr.Sci. (Engineering), Professor, bars87@mail.ru

[1] Orenburg State University, Aerospace Institute, Orenburg, 460018, Russian Federation
[2] Orenburg State University, Institute of Mathematics and Information Technology, Orenburg, 460018, Russian Federation