Руководство пользователя

NNWizard

v0.3.13

Роботрек, 2021

Оглавление

О NNWizard

В современном мире широкое применение нашли нейронные сети. С уверенностью можно сказать, что они стали неотъемлемой частью нашей жизни. Разработка нейронной сети может быть как простой, так и очень сложной, но какой бы она ни была – почти всегда нейронные сети разрабатываются на одном из языков программирования.

Что же делать, если вы хотите разработать свою нейронную сеть, но не имеете навыков программирования? Или если вы не хотите тратить время на написание однотипного кода? Для решения этих проблем мы разработали специальную программу – NNWizard.

Разработка моделей нейронных сетей – нетривиальная задача и цель данного программного обеспечения – упростить процесс моделирования и обучения моделей нейронных сетей для их последующего использования на микроконтроллерных платах семейства "**Трекдуино**" с модулем искусственного интеллекта "**Артинтрек**".

Рисунок 1. Логотип NNWizard

NNWizard (Neural Network Wizard – англ. Волшебник нейронных сетей) – это среда визуального **моделирования** архитектуры сверточной нейронной сети, ее **обучения** и **экспорта** для последующего использования на аппаратном модуле **Артинтрек**.

Работа в **NNWizard** осуществляется в формате **проекта**.

Системные требования

Критически минимальные системные требования

Процессор	Архитектура: x64 Тактовая частота: ≥ 2GHz
Объем оперативной памяти	≥ 2GB
Операционная система	Microsoft Windows 7

Минимальные системные требования

Процессор	Архитектура: x64 Тактовая частота: ≥ 3GHz
Объем оперативной памяти	≥ 2GB
Видеокарта	NVIDIA CUDA-совместимая (Compute Capability ≥ 3.5) Не старше GeForce GTX 780*
Объем видеопамяти	≥ 1GB
Операционная система	Microsoft Windows 7

Рекомендуемые системные требования

Процессор	Архитектура: x64 Тактовая частота: ≥ 4GHz
Объем оперативной памяти	≥ 8GB
Видеокарта	NVIDIA CUDA-совместимая (Compute Capability ≥ 5.0) Не старше GeForce GTX 9**
Объем видеопамяти	≥ 4GB
Операционная система	Microsoft Windows 10

Интерфейс NNWizard

Главное окно

Главное окно – точка входа в программу.

Рисунок 2. Главное окно. 1 – главное меню; 2 – кнопка перехода к главному окну; 3 – кнопка перехода к окну моделирования; 4 – кнопка перехода к окну обучения; 5 – кнопка перехода к окну экспорта; 6 – кнопка перехода к окну настроек; 7 – кнопка создания нового проекта; 8 – кнопка открытия существующего проекта; 9 – Строка состояния; 10 – блок кнопок открытия недавно открытых проектов.

Вверху окна находится строка меню, отвечающая за быстрый доступ к важным функциям. Слева находится полоса управления, отвечающая за переход между окнами:

- главное окно;
- окно моделирования;
- окно обучения;
- окно экспорта;
- окно настроек.

При наведении указателя мыши на одно из них, появится подсказка, содержащая название окна, к которому будет совершен переход. Кнопка активного окна выделяется белым фоном.

Снизу находится строка состояния, отображающая информационные сообщения и сообщения об ошибках.

В основной части главного окна находятся кнопки, отвечающие за быстрый доступ к функциям, связанным с проектом: его создание и открытие.

Главное меню

Главное меню состоит из двух меню: «Файл» и «Помощь».

При нажатии на «Файл», откроется меню, содержащее подменю «Проект» и кнопку «Выйти».

Рисунок 3 . Главное меню.

При нажатии на подменю «Проект», откроется подменю, содержащее кнопки создания проекта («Новый»), его открытия («Открыть») и сохранения («Сохранить»).

Рисунок 4. Подменю «Проект».

При нажатии на кнопку «Выйти» (или при клавиатурном сочетании Ctrl+Q), произойдет выход из программы.

При нажатии на меню «Помощь», откроется меню, содержащее кнопку «О программе» для перехода к окну «О программе», содержащему информацию о версии программы, ее разработчиках и используемых компонентах.

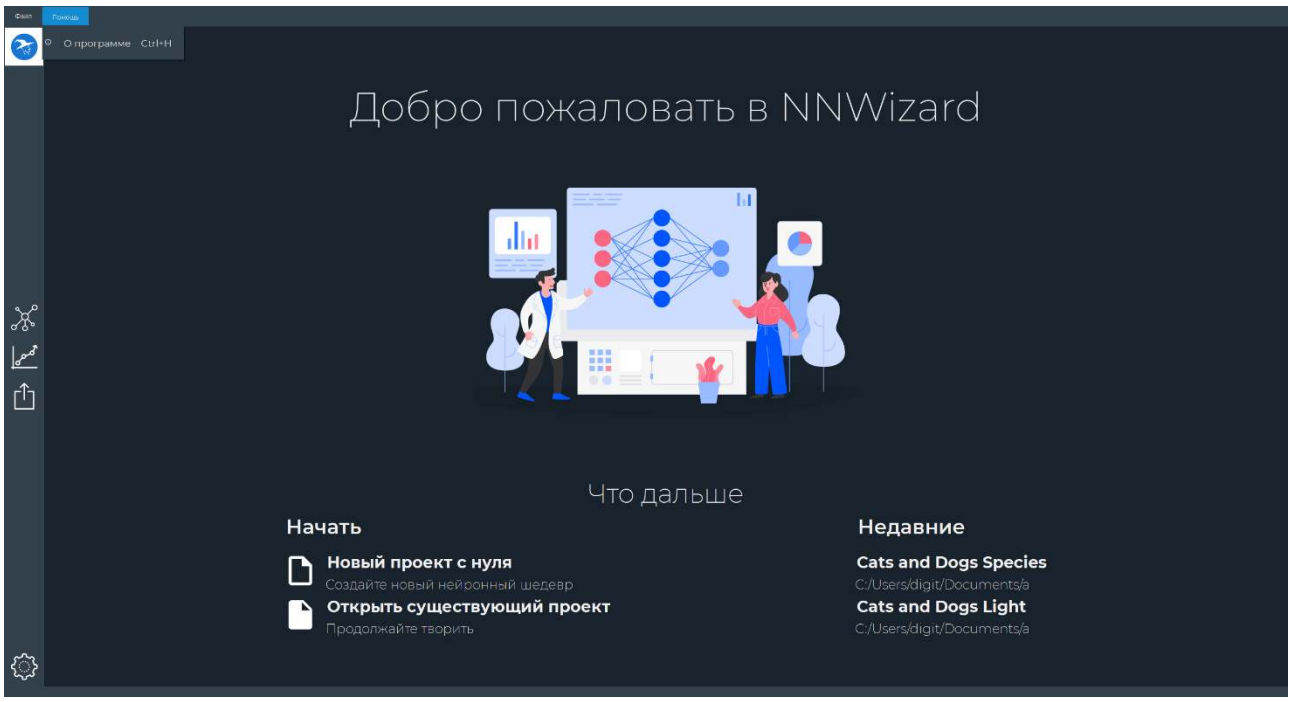

Рисунок 5. Меню «Помощь».

Окно моделирования

Окно моделирования содержит доску для редактирования архитектуры нейронной сети.

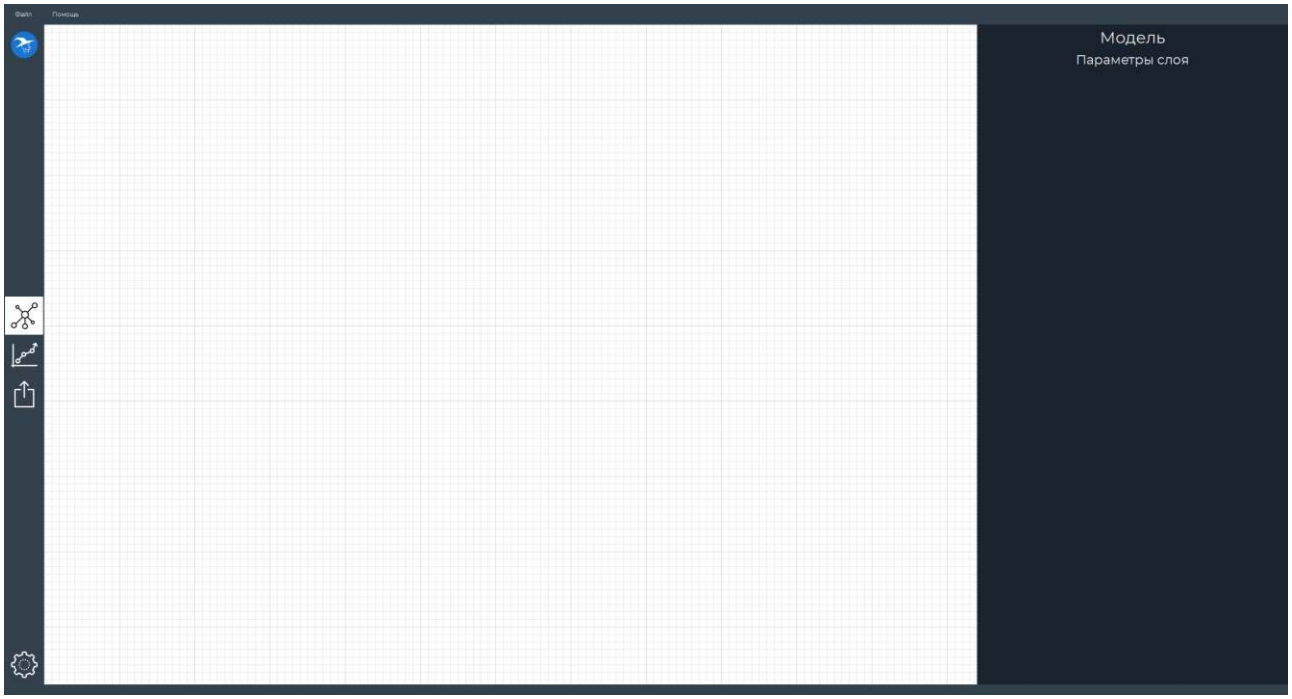

Рисунок 6. Окно моделирования архитектуры нейронной сети.

В правой части окна моделирования находится окно редактирования параметров выбранного слоя нейронной сети.

Окно обучения

Окно обучения содержит доску для редактирования параметров обучения.

Рисунок 7. Окно обучения нейронной сети.

В правой части окна обучения находятся:

- Графики функции потерь и правильности обученной модели;
- Индикаторы эпохи и пакета;
- Кнопка запуска процесса обучения модели.

Окно экспорта

Цель окна экспорта – сохранить нашу обученную модель нейронной сети, чтобы мы могли применить ее для управления роботом.

Окно экспорта содержит поле выбора переносного устройства для сохранения экспортированной модели и кнопку для запуска процесса экспорта.

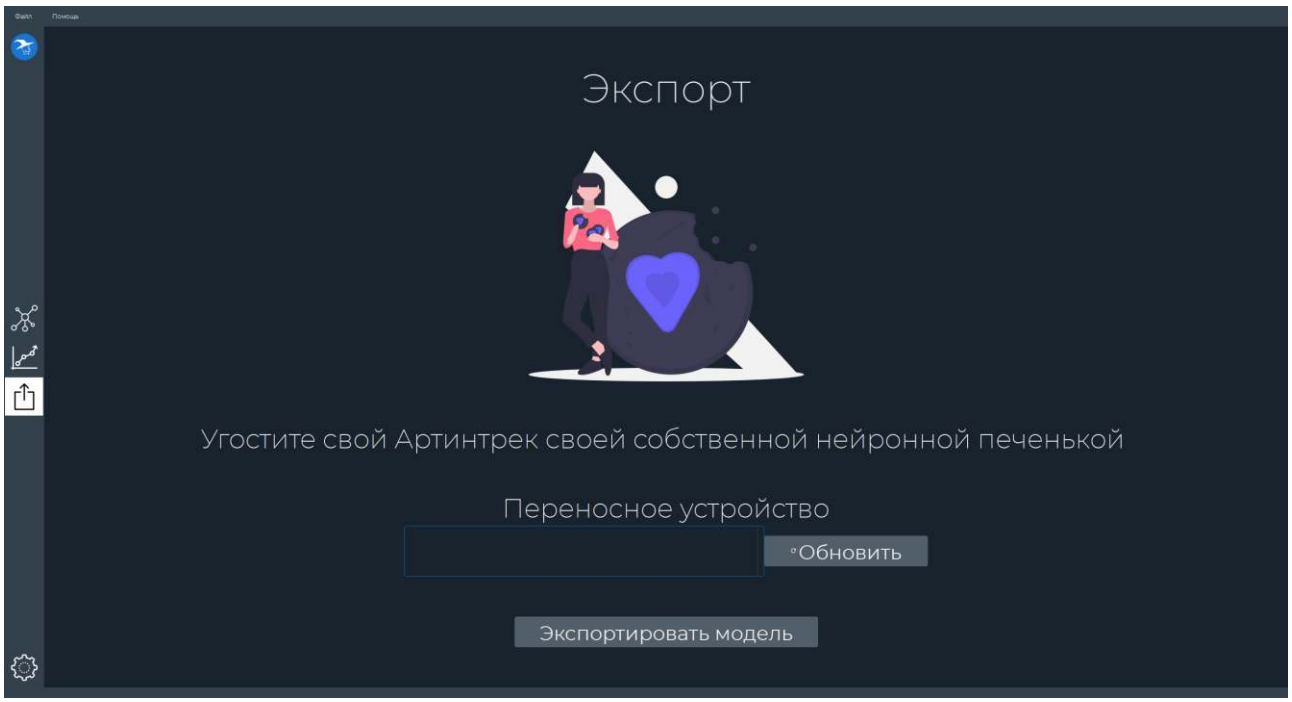

Рисунок 8. Окно экспорта обученной модели нейронной сети.

Окно настроек

Окно настроек содержит три вкладки: «Общие» настройки, «Помощь» и «О приложении».

Общие настройки

Вкладка «Общие» настройки содержит переключатель для выбора языка программы.

Смена языка осуществляется с помощью поля выбора языка нажатием на него левой кнопки мыши или прокручиванием колесика мыши при наведении на него.

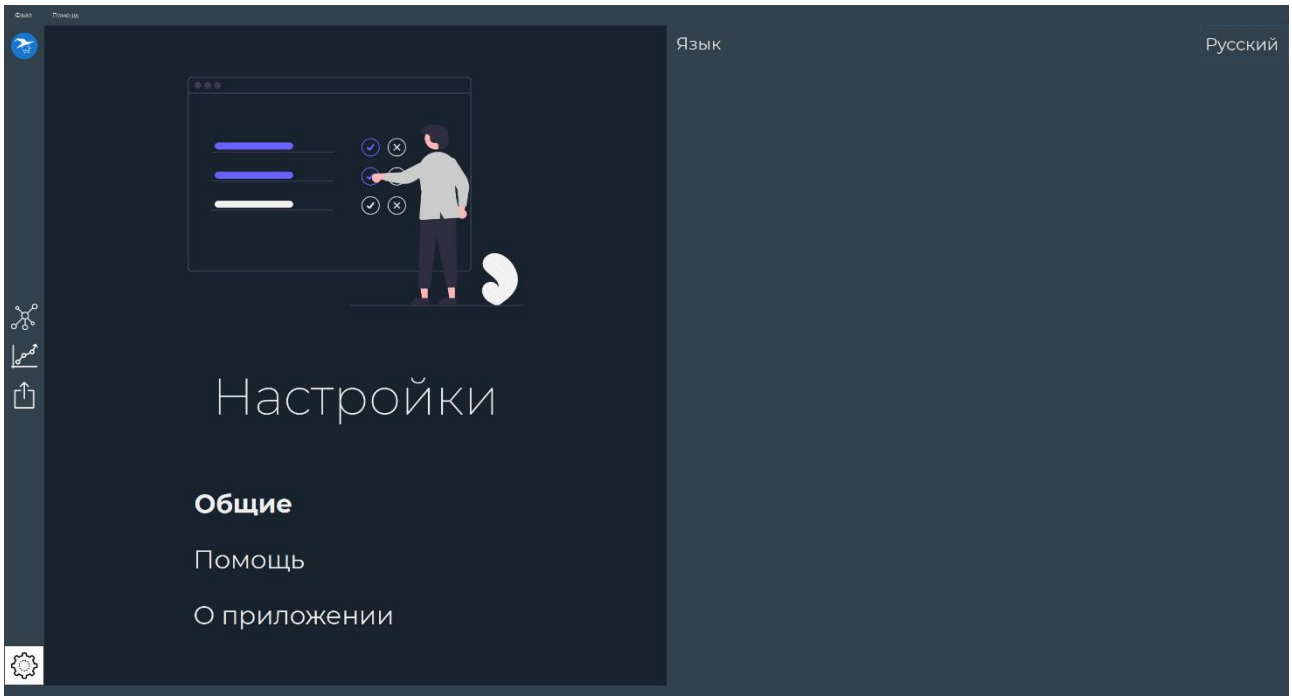

Рисунок 9. Вкладка «Общие» окна настроек.

Смена языка произойдет моментально без необходимости перезапуска приложения.

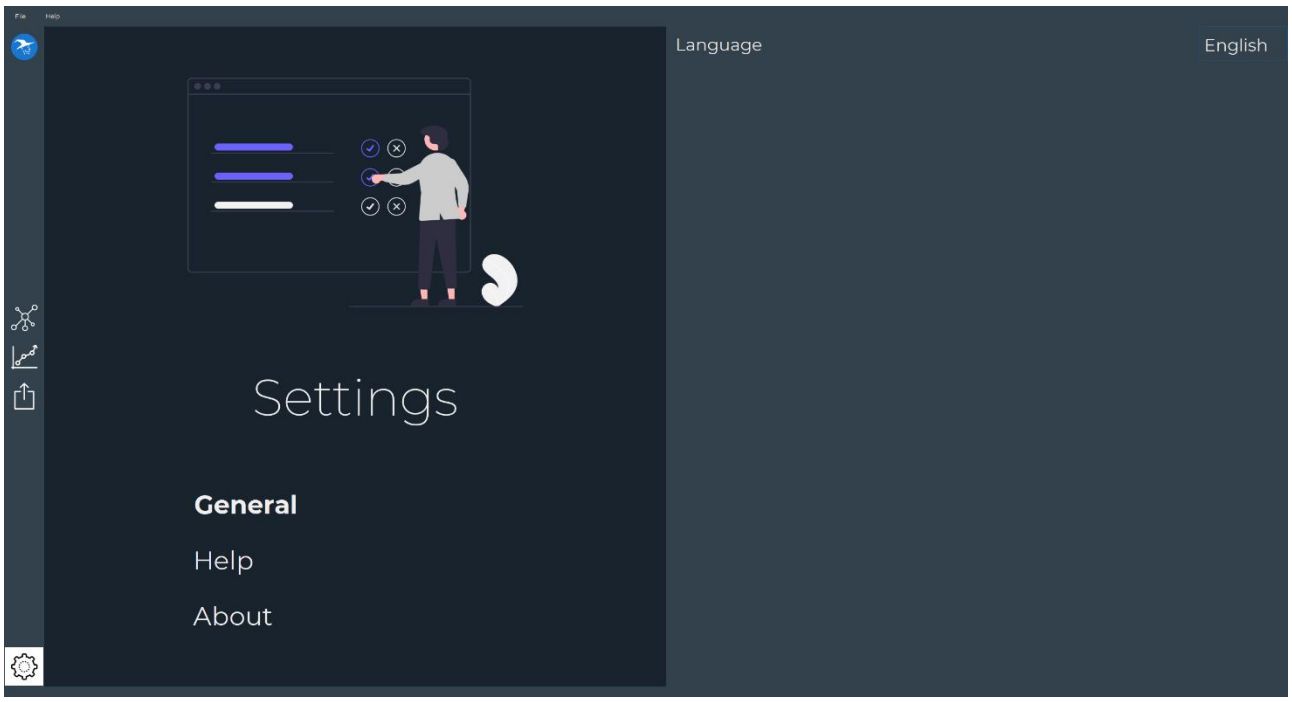

Рисунок 10. Вкладка «Общие» после смены языка на английский.

О приложении

Окно «О приложении» содержит сведения об используемой версии приложения, его разработчике и о версиях используемых компонентов.

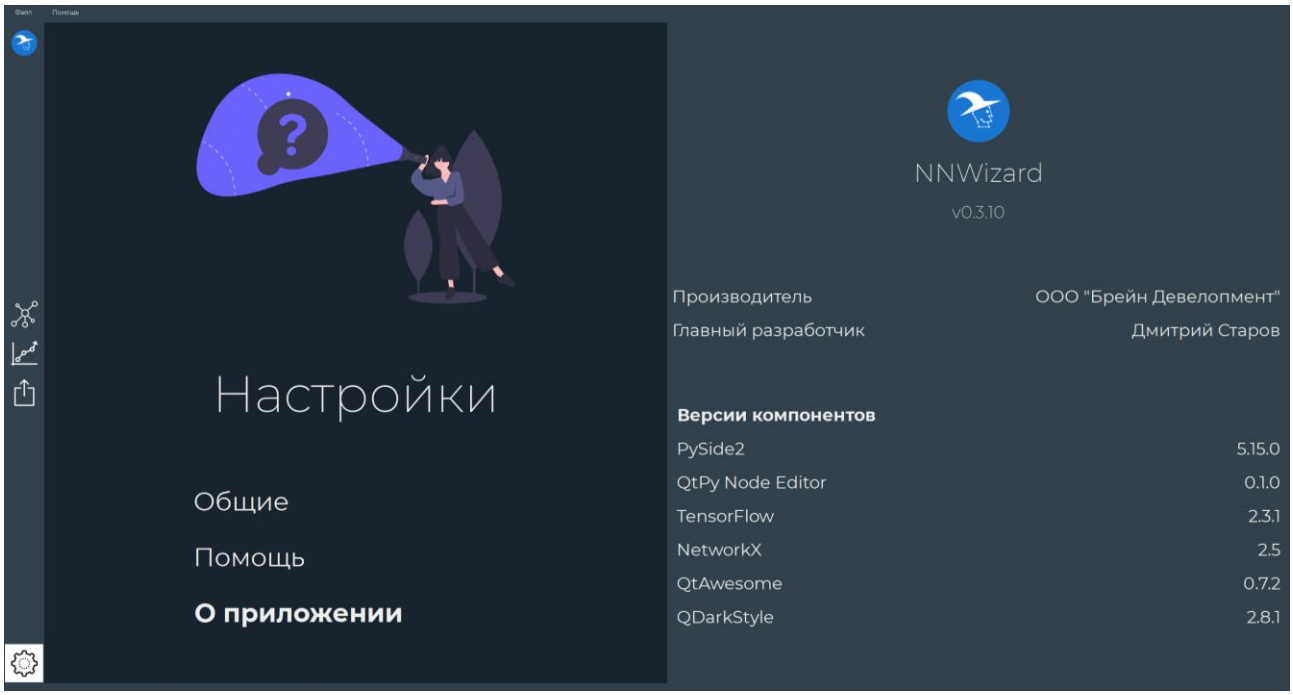

Рисунок 11. Вкладка «О приложении» окна настроек.

Работа с проектом

Работа с моделью нейронной сети ведется в формате проекта.

> Обратите внимание, что один проект содержит только одну модель нейронной сети.

Архитектура модели может быть составлена без создания проекта или до него, но процессы обучения и экспорта могут быть осуществлены только при созданном проекте.

Файл проекта содержит описания графа архитектуры модели нейронной сети и графа параметров ее обучения.

Создание проекта

Приступить к созданию проекта можно тремя способами:

1. Перейти к главному окну и нажать кнопку «Новый проект с нуля»;
2. Открыть меню «Файл»-«Проект» и нажать кнопку «Новый»;
3. Использовать клавиатурное сочетание Ctrl+N.

В результате будет открыто окно создания нового проекта.

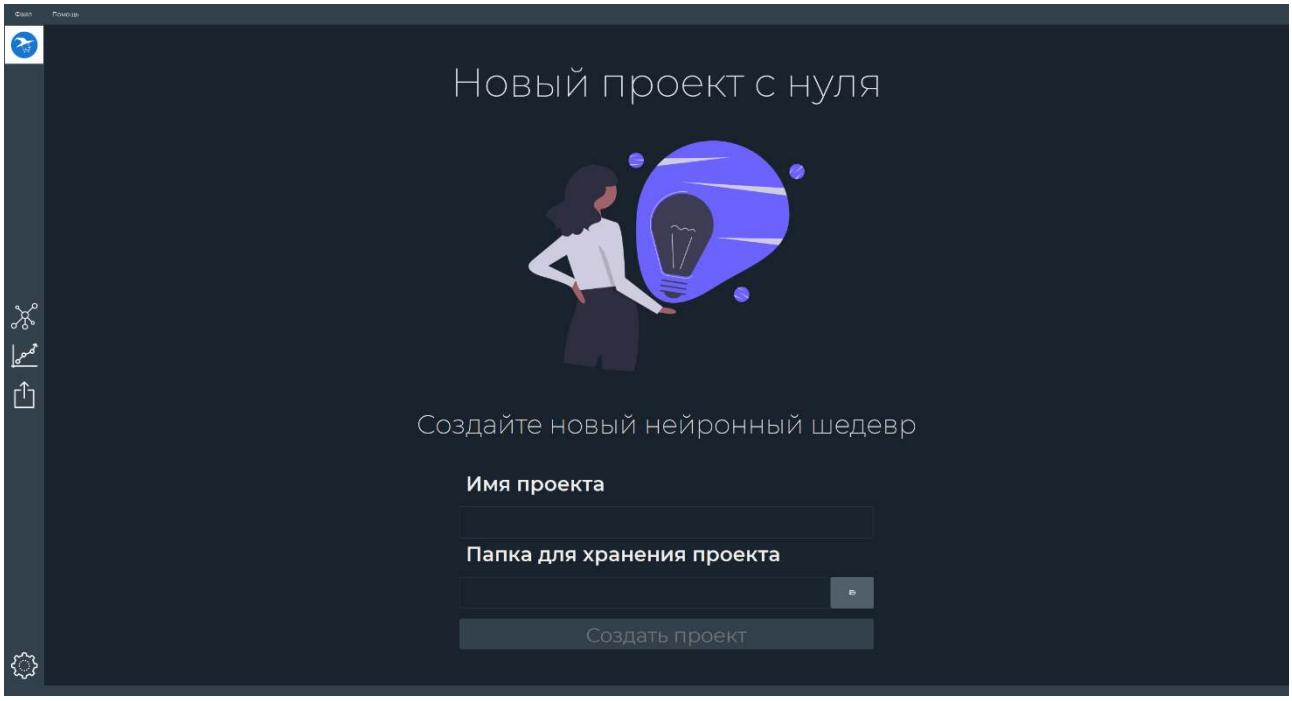

Рисунок 12. Окно создания нового проекта.

Это окно содержит поле ввода имени проекта, поле выбора папки для хранения проекта и кнопку создания проекта.

Кнопка «Создать проект» будет недоступна для нажатия, пока оба поля не будут заполнены.

Имя проекта может быть любым, за исключением некоторых ограничений, показанных в подсказке при наведении на поле имени проекта.

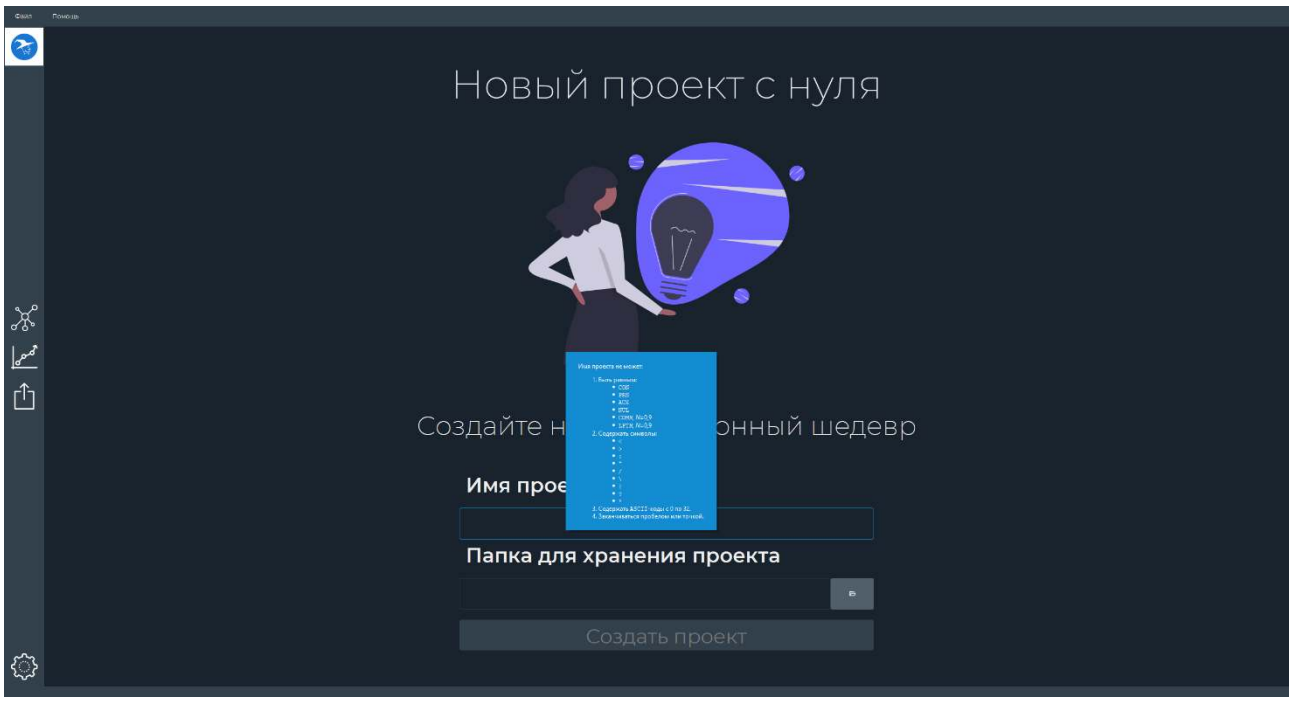

Рисунок 13. Подсказка при наведении на поле имени проекта.

Имя проекта не может:

- Быть равным:
  - CON;
  - PRN;
  - AUX;
  - NUL;
  - COM*N*, $N \in [0, 9]$;
  - LPT*N*, $N \in [0, 9]$;
- Содержать символы:
  - <;
  - >;
  - :;
  - ";
  - /;
  - \;
  - |;
  - ?;
  - *;
- Содержать ASCII-коды с 0 по 32;
- Заканчиваться пробелом или точкой.

Для выбора папки сохранения проекта нужно нажать на кнопку с иконкой папки справа от поля, выбрать папку и нажать «OK».

После заполнения этих полей активируется кнопка «Создать проект».

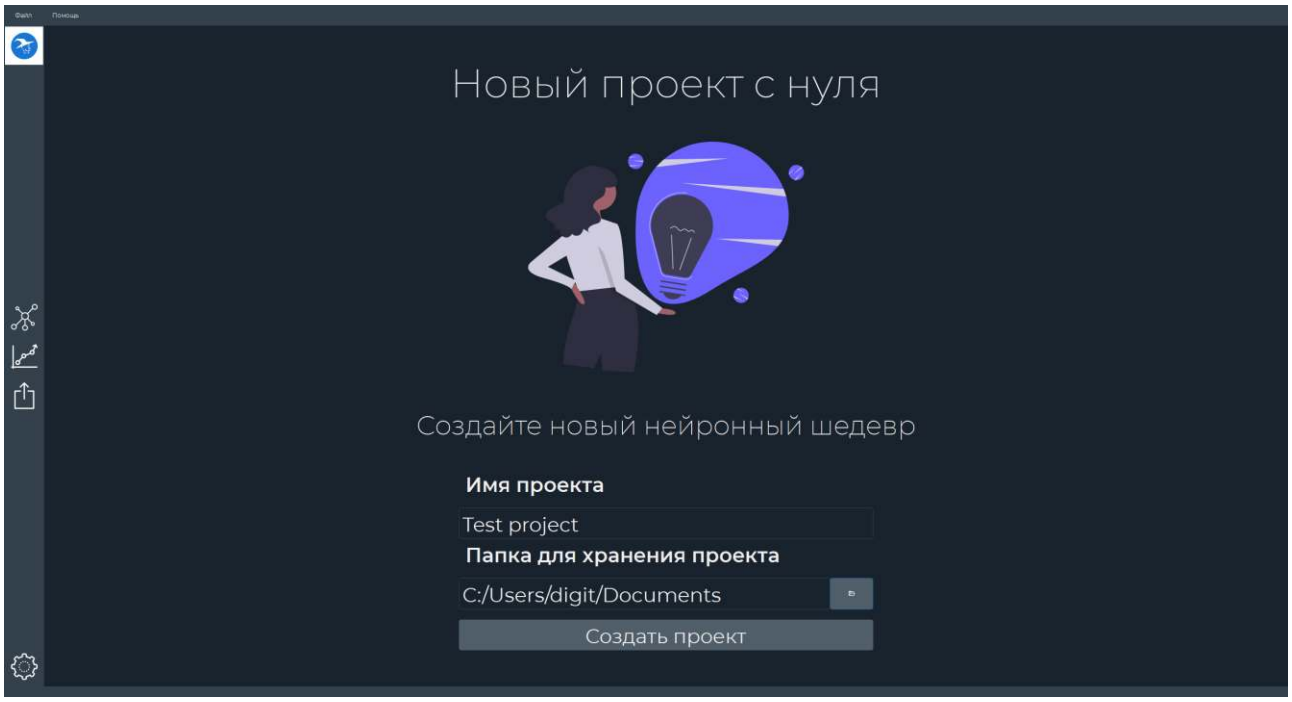

Рисунок 14. Пример заполненных полей окна создания проекта.

После нажатия на кнопку «Создать проект» произойдет переход к окну моделирования нейросети.

Сохранение проекта

Сохранить проект можно двумя способами:

1. Открыть меню «Файл»-«Проект» и нажать кнопку «Сохранить»;
2. Использовать клавиатурное сочетание Ctrl+S.

Открытие проекта

Открыть проект можно тремя способами:

1. Открыть меню «Файл»-«Проект» и нажать кнопку «Открыть»;
2. Перейти к главному окну и нажать «Открыть существующий проект»;
3. Использовать клавиатурное сочетание Ctrl+O.

> Если во время открытия проекта вы уже работаете с другим проектом, появится сообщение о том, что все несохраненные в текущем проекте изменения будут утрачены. Если вы хотите сохранить изменения, нажмите «Отмена», сохраните текущий проект и откройте желаемый, проигнорировав сообщение.

Дополнительно

В папке хранения проекта будет создана папка с именем проекта. В нее сохраняются:

- Дамп-файл с расширением .h5, содержащий веса обученной модели, показывающей минимальную за все время обучения функцию потерь;
- После экспорта:
  - Файлы модели в формате TensorFlow SavedModel;
  - Файлы модели в формате OpenVINO Model.

Управление доской

В окнах моделирования и обучения используется своеобразная «магнитная» доска. Можно представить, что мы смотрим на доску через видоискатель камеры. Мы можем двигать камеру, приближать или отдалять изображение.

Перемещение доски

Чтобы подвинуть камеру, нажмите левую кнопку мыши на свободном участке доски и, удерживая кнопку, переместите мышь.

Масштабирование доски

Для приближения доски прокрутите колесико мыши вперед. Для отдаления доски прокрутите колесико мыши назад.

Добавление карточки

Карточки – это основные кирпичики, с помощью которых строится модель нейронной сети или настраиваются параметры ее обучения.

Для добавления карточки на доску нажмите правой кнопкой мыши на свободный участок доски. Откроется окно выбора карточек, разделенных на категории.

Для прокручивания списка используйте колесико мыши или ползунок прокрутки справа от списка.

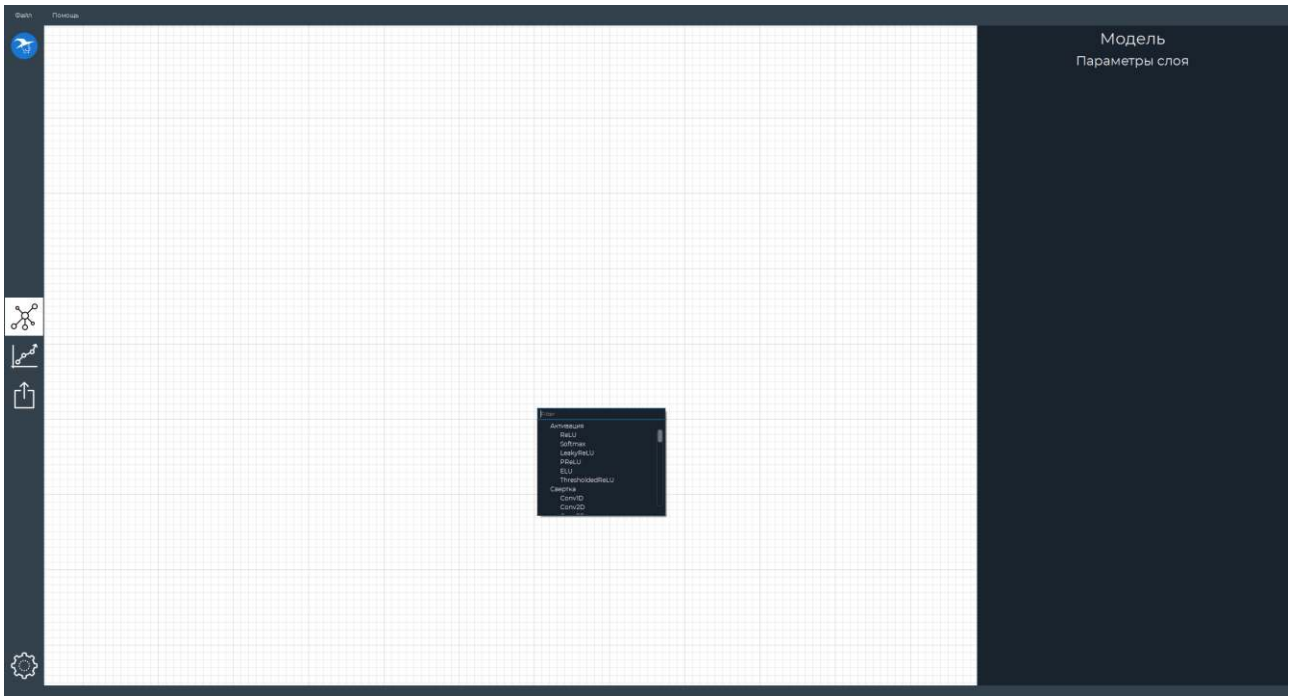

Рисунок 15. Окно выбора карточки.

Вверху списка карточек имеется поле фильтра. Начните вводить название карточки, чтобы исключить ненужные варианты.

Поле фильтра чувствительно к регистру. Будьте внимательны при наборе текста. Если вы напишите имя карточки не так, как она называется в списке, **NNWizard** не сможет найти эту карточку.

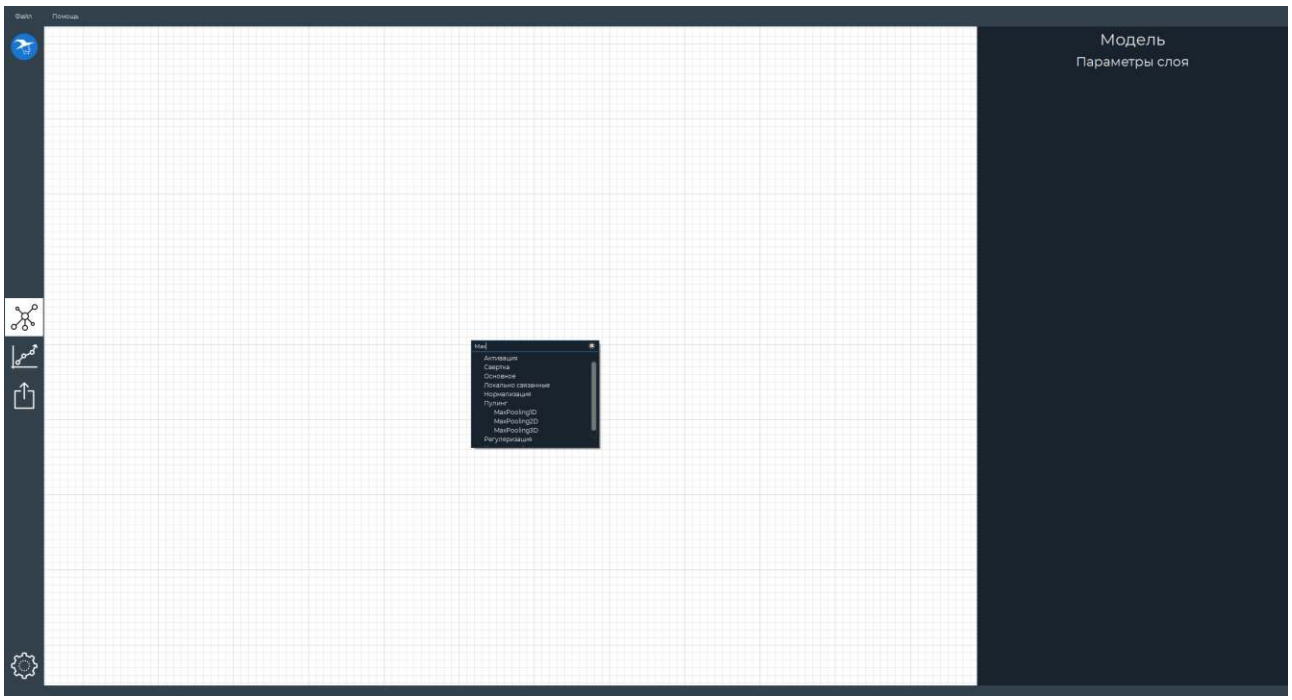

Рисунок 16. Фильтрация в окне выбора карточки.

Нажмите на имя карточки в списке, и она добавится на доску.

Действия с карточкой

После нажатия правой кнопкой мыши на карточке откроется меню действий. Оно состоит из двух кнопок: «Клонировать» и «Удалить».

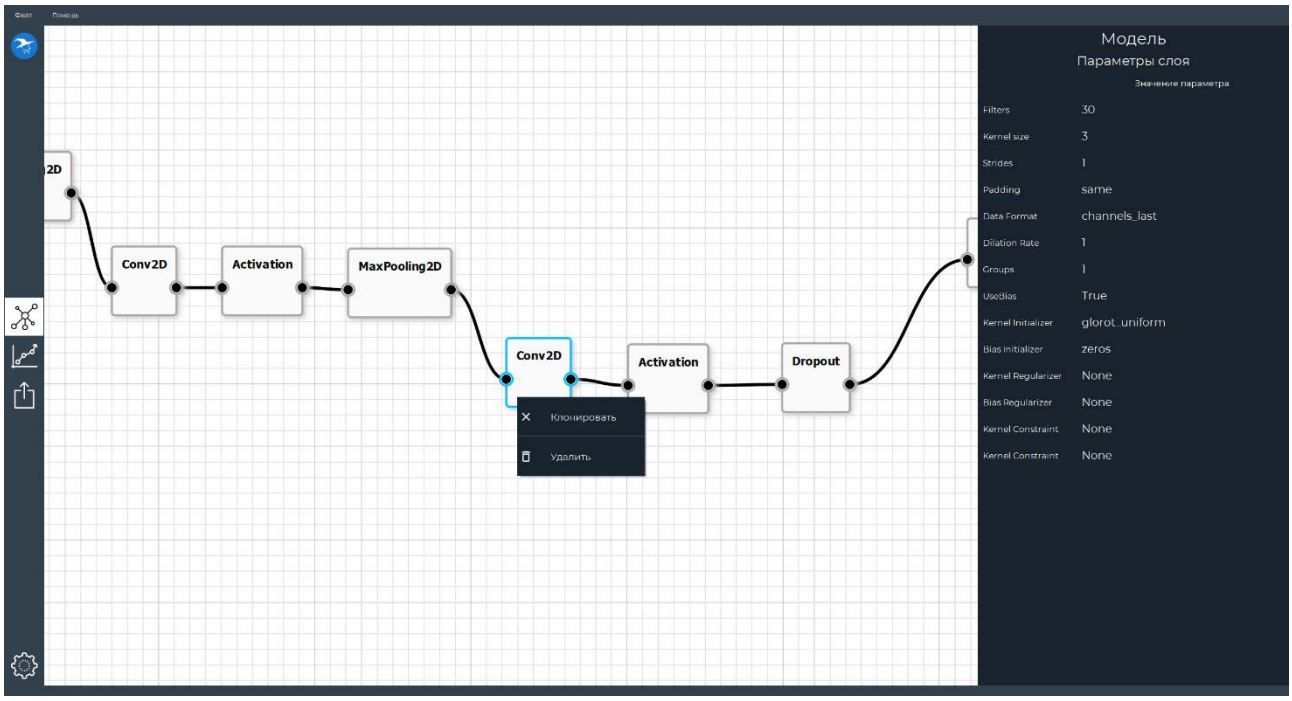

Рисунок 17. Меню карточки.

После нажатия на кнопку «Клонировать», снизу от выбранной карточки появится новая карточка с такими же параметрами, как у исходной.

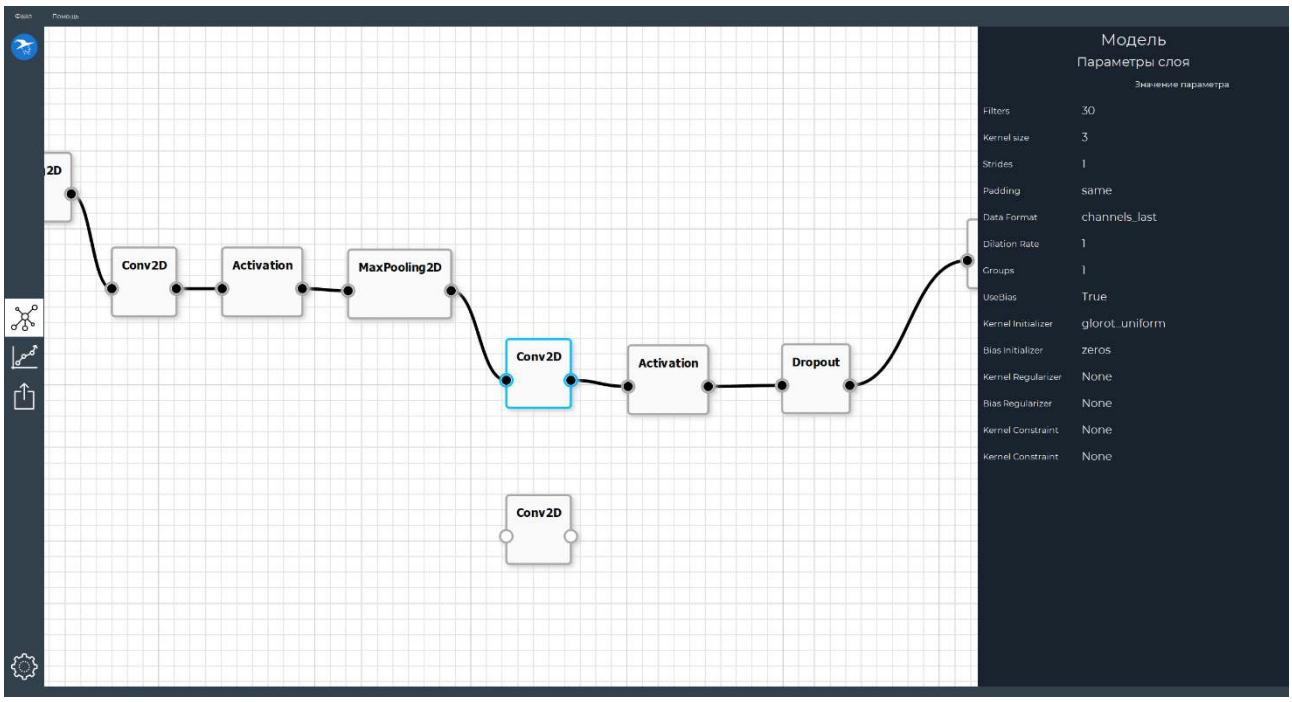

Рисунок 18. Клонирование карточки.

При нажатии на кнопку «Удалить», выбранная карточка будет удалена вместе с ее соединениями. Удалить карточку можно также с помощью клавиши «Delete».

Перемещение карточки

Для перемещения карточки зажмите левую кнопку мыши на карточке и переместите мышь. Отпустите левую кнопку мыши для сохранения нового положения. Карточка будет расположена на ближайшей к линиям сетки позиции.

Создание соединения

Наведите мышь на кружок в правой части карточки. Зажмите левую кнопку мыши и переместите ее. Появится пунктирная линия, ведущая из карточки в желаемое положение.

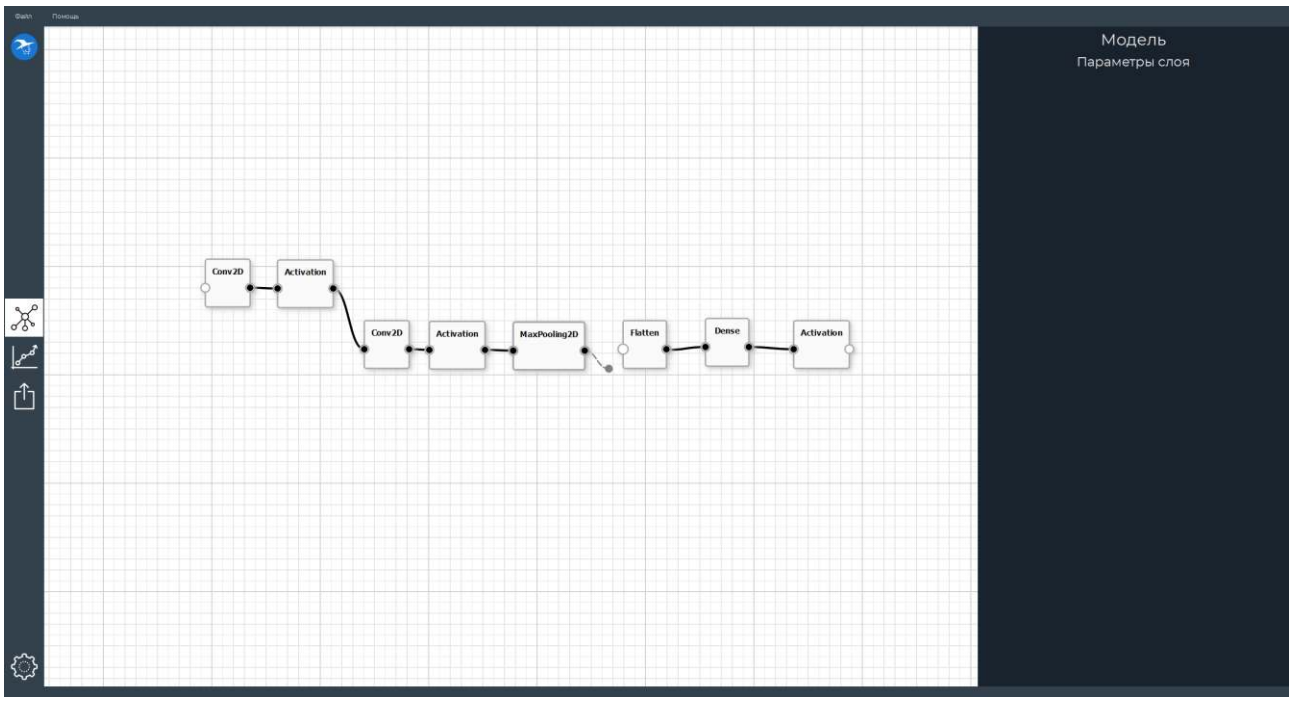

Рисунок 19. Создание соединения.

Не отпуская левой кнопки мыши, наведите ее на кружок в левой части другой карточки. При приближении к нему мыши, он будет увеличиваться в размере.

> Если кружок уменьшается, то соединение между двумя этими карточками не может быть установлено.

Переместите конец соединения во внутреннюю часть кружка желаемой карточки и отпустите левую кнопку мыши. Линия перестанет быть пунктирной – соединение создано!

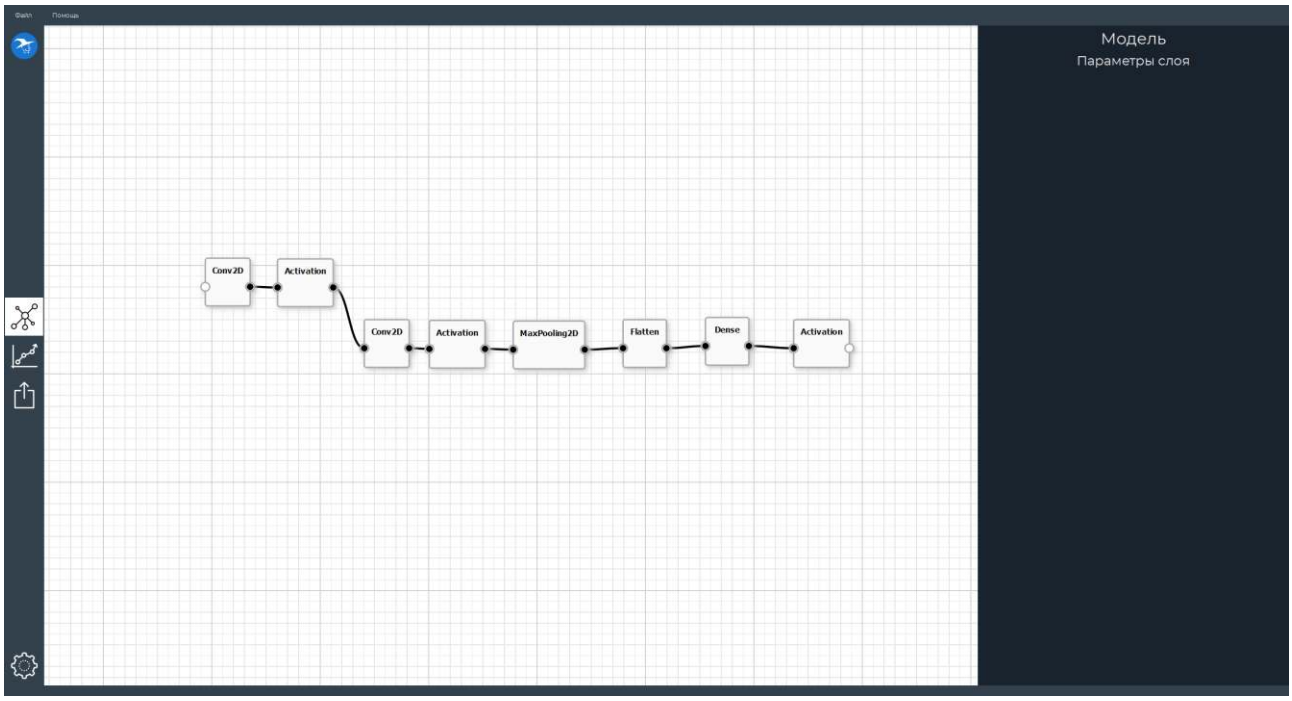

Рисунок 20. Соединение карточек.

Выделение

Чтобы удалить или переместить не одну карточку, а сразу несколько, вы можете выделить группу карточек. Для этого зажмите клавишу «Shift» и, удерживая нажатой левую кнопку мыши на свободной части доски, переместите мышь. На экране вы увидите область выделения.

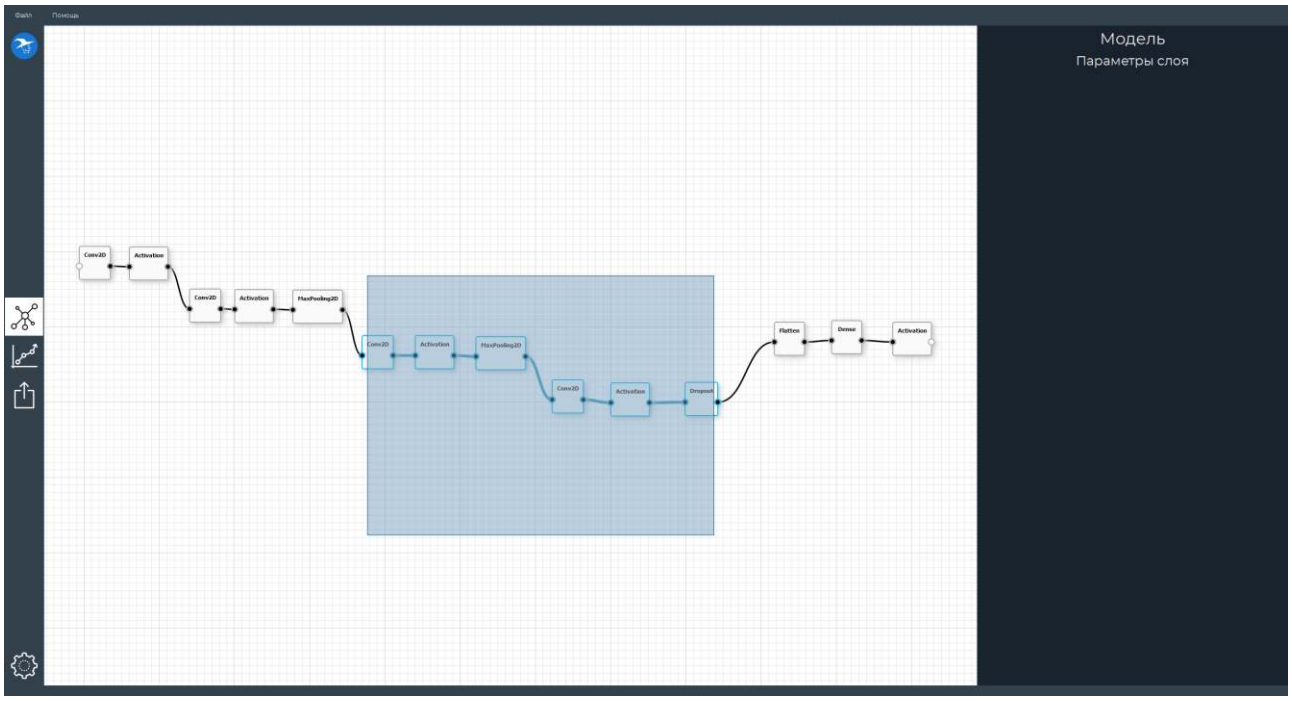

Рисунок 21. Выделение карточек.

Отпустите левую кнопку мыши. Все попавшие в область выделения карточки и соединения между ними будут выделены.

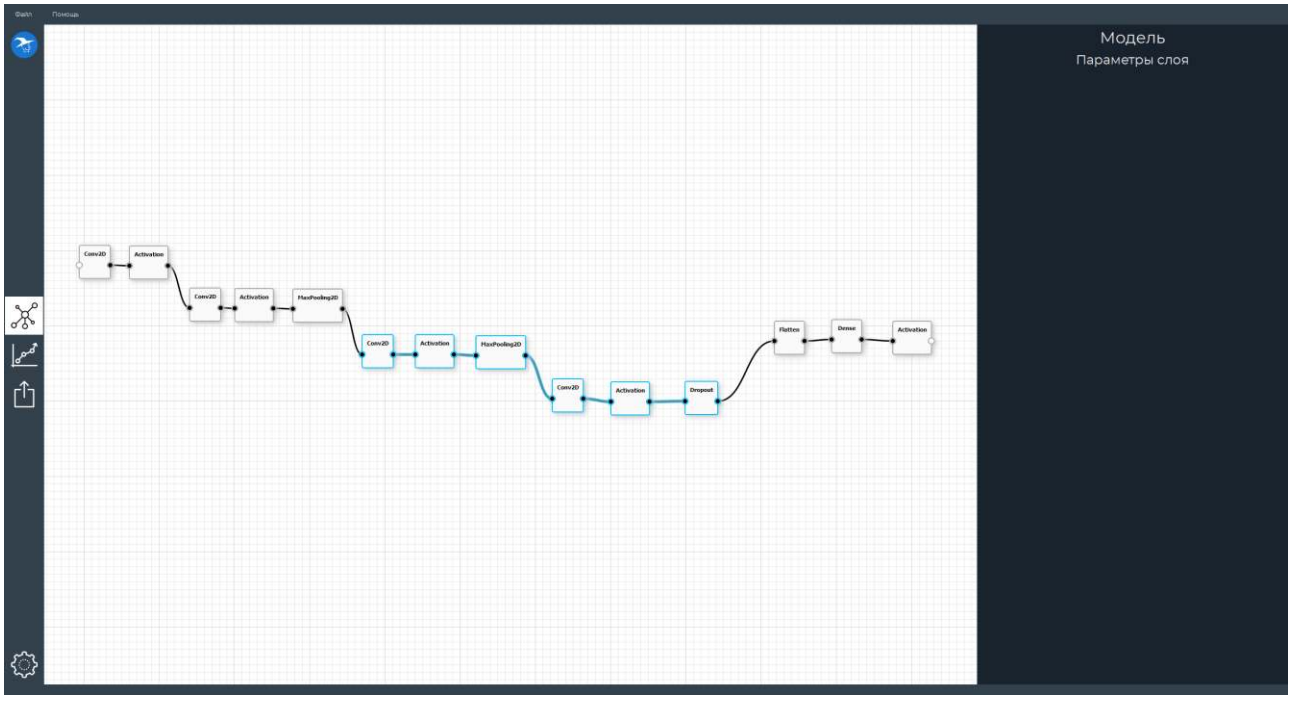

Рисунок 22. Выделенные карточки и соединения.

Перемещение группы

Для перемещения группы карточек и соединений зажмите левую кнопку мыши одной из карточек группы и переместите мышь. Отпустите левую кнопку мыши для сохранения нового положения.

Удаление группы

Чтобы удалить группу карточек и соединений, нажмите клавишу «Delete». Все выделенные карточки и соединения будут удалены.

> Также будут удалены соединения, затрагивающие карточки, но не входящие в группу.

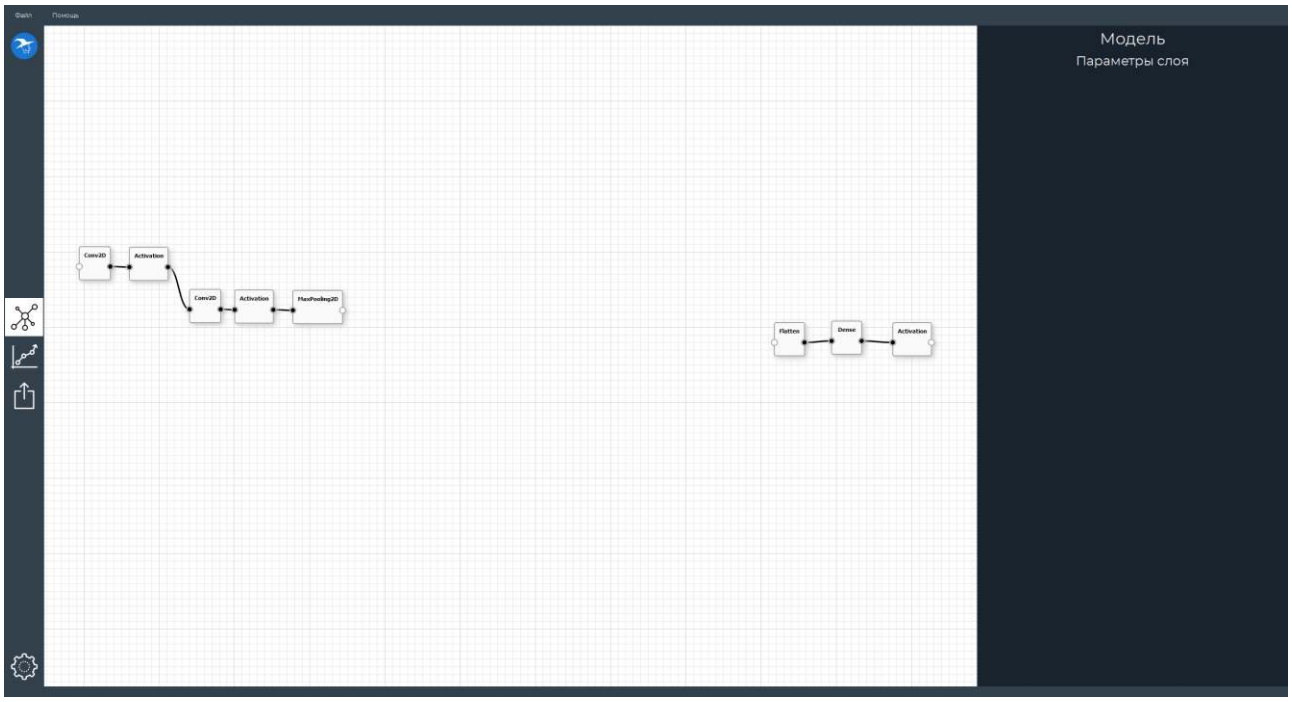

Рисунок 23. Граф после удаления группы.

Моделирование нейронной сети

Архитектура модели нейронной сети представлена в виде последовательного графа. Граф строится на доске из карточек слоев, соединенных друг с другом в желаемой последовательности. Каждый слой может иметь:

- 1 соединение, если он является первым или последним слоем в модели;
- 2 соединения, если он является промежуточным.

> Стоит заметить, что **NNWizard** позволяет строить только последовательные модели нейронных сетей. В них у каждого слоя может быть не больше одного предшествующего слоя и не больше одного последующего. Поэтому, модель нейросети, построенная в **NNWizard** будет представлять собой цепочку карточек и не может состоять из нескольких несвязанных друг с другом графов.

Слои архитектуры нейронной сети

Почему мы так сосредоточены именно на сверточных нейронных сетях? Потому что это очень распространенный вид нейросетей, с которым легко работать. Основная задача **NNWizard** – это быстрая разработка нейронных сетей без навыков программирования. В **NNWizard** очень легко строить и обучать сверточные сети, а если вы захотите проверить их работу, то легко сможете сделать это, например, с помощью модуля искусственного интеллекта **Артинтрек**.

Этот модуль получит изображение с помощью камеры, обработает его с помощью вашей нейросети и вернет ответ на робототехнический контроллер. С помощью него вы сможете управлять роботом.

Слой свертки

Главная составляющая слоя свертки – **детектор признаков**, также известный как **ядро свертки** или **фильтр**, который перемещается по изображению и проверяет есть ли внутри участка изображения, на котором он находится, искомый признак. Такой участок называется **полем восприятия**, а сам процесс называется **сверткой**.

Детектор признаков – это двумерная сетка, хранящая веса, представляющие участок изображения. Сами изображения могут быть различного размера, но размер фильтра обычно составляет 3×3 пикселя. Размер фильтра также задает размер поля восприятия. Фильтр применяется к полю и вычисляет произведения между значением в поле и значением в фильтре. Эти произведения суммируются и сохраняются в массиве.

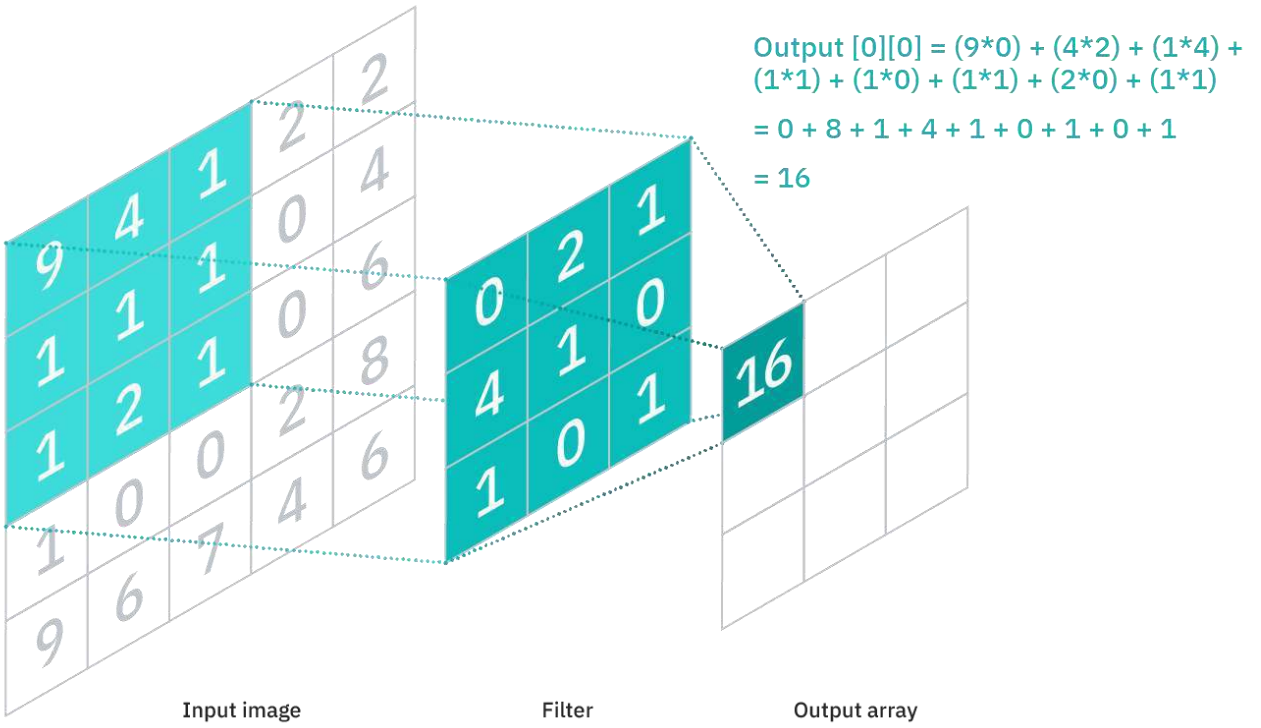

Рисунок 24. Принцип работы слоя свертки.

После этого фильтр смещается на определенный шаг, также называемый **страйдом**, и повторяет процесс для нового положения поля восприятия. Так происходит, пока свертка не будет применена ко всему изображению.

Финальный массив сумм произведений называется **картой признаков**.

> Заметьте, что веса детектора признаков остаются нетронутыми во время движения поля восприятия по изображению, то есть являются общими для всей свертки. Веса свертки подстраиваются во время обучения нейросети.

Как можно увидеть, каждое выходное значение карты признаков зависит не от конкретного пикселя, а от всех пикселей внутри поля восприятия. Таким образом, обнаружение пикселей применяется ко входному изображению не полностью, а частично. Поэтому слой свертки (как и слой пулинга) часто называют **частично связанными** слоями.

Слой свертки в NNWizard

Слой свертки представлен в **NNWizard** в виде карточки «Conv2D» в окне моделирования нейросети.

Параметры слоя свертки

Размер фильтра задается параметром «Kernel size» в панели редактирования карточки слоя. Он устанавливает размер одной из сторон квадратного фильтра.

Параметры, которые должны быть установлены до обучения нейросети, называются **гиперпараметрами**.

Для слоя свертки гиперпараметрами являются:

- **Количество фильтров**. Например, три отдельных фильтра будут возвращать три разных карты признаков.

Количество фильтров задается параметром «filters» в панели редактирования карточки слоя.

- **Шаг**. Расстояния, или количество пикселей, на которое сдвигается поле восприятия. Шаг задается параметром «strides» в панели редактирования карточки слоя.

Слой пулинга

Слой пулинга предназначен для сокращения размерности входных данных (или другими словами, уплотнения карт признаков).

Пусть мы уже имеем вычисленные карты признаков для подаваемого изображения, полученные с помощью слоя свертки. Тогда для дальнейшей работы нет необходимости использовать эту карту целиком. Мы можем уплотнить карту признаков, тем самым повысив обобщающие способности нейронной сети и увеличив скорость ее обучения и выполнения. Такой прирост производительности достигается за счет сокращения количества параметров.

Принцип работы слоя пулинга очень схож с тем, как работает слой свертки. У слоя пулинга так же имеется фильтр, применяемый к полю восприятия. Поле восприятия так же перемещается по изображению с определенным шагом.

Однако, есть принципиальное различие между слоем пулинга и слоем свертки – элементы фильтра слоя пулинга не имеют весов. Вместо этого используется функция, вычисляющая совокупное значение всех пикселей изображения, попадающих в поле восприятия.

Обычно применяется два вида такой функции:

Усредняющий пулинг

Усредняющий пулинг вычисляет среднее значение всех пикселей внутри поля восприятия.

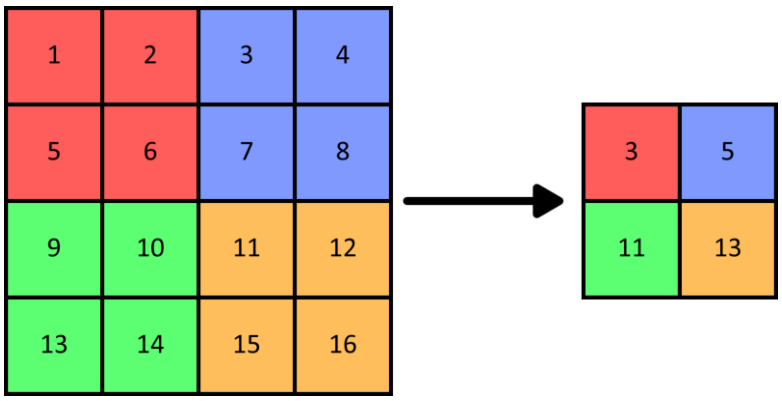

Рисунок 25. Визуализация усредняющего двумерного пулинга.

Максимизирующий пулинг

Максимизирующий пулинг выбирает максимальное значение среди всех пикселей внутри поля восприятия.

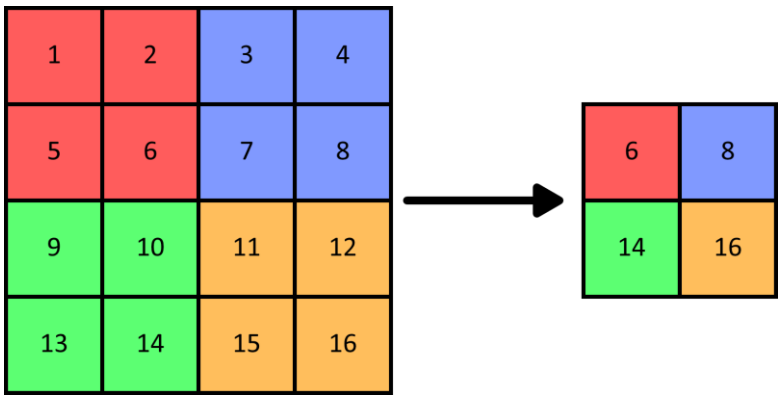

Рисунок 26. Визуализация двумерного пулинга максимального значения.

Обычно данный вид пулинга применяется чаще всего.

> Хотя в результате сокращения размерности мы теряем много информации, слой пулинга имеет свои преимущества для использования в сверточной нейронной сети. Он помогает уменьшить сложность данных, улучшить эффективность модели нейросети и уменьшить риск переобучения. Что такое переобучение вы узнаете в одном из следующих видео.

Слой пулинга в NNWizard

Слой усредняющего пулинга представлен в **NNWizard** в виде карточки «AveragePooling2D» в окне моделирования нейросети.

Слой максимизирующего пулинга представлен в **NNWizard** в виде карточки «MaxPooling2D» в окне моделирования нейросети.

Параметры слоя пулинга

Размер фильтра задается параметром «Pool size» в панели редактирования карточки слоя. Он устанавливает размер одной из сторон квадратного фильтра.

Для слоя пулинга гиперпараметром является значение **шага.** Шаг задается параметром «strides» в панели редактирования карточки слоя.

Слой активации

Функция активации получает входное значение и возвращает обработанный результат. Функция активации часто описывается как такая, которая определяет передаст ли каждый нейрон данные дальше по нейронной сети. На практике же функция активации позволяет избавиться от отрицательных данных или привести данные к некоторому отрезку, например, $[-1, 1]$ или $[0, 1]$.

Нелинейные функции

Нелинейная функция возвращает линейный вход с кривизной, называемой **нелинейностью**. Такая нелинейность позволяет усиливать слабые входные сигналы.

Сигмоида

Сигмоидная функция широко распространена в машинном обучении и простейших реализациях нейронных сетей. Она ограничивает входные данные до отрезка $[0, 1]$.

Сигмоидная функция активации – это возрастающая нелинейная функция, имеющая форму латинской буквы «S».

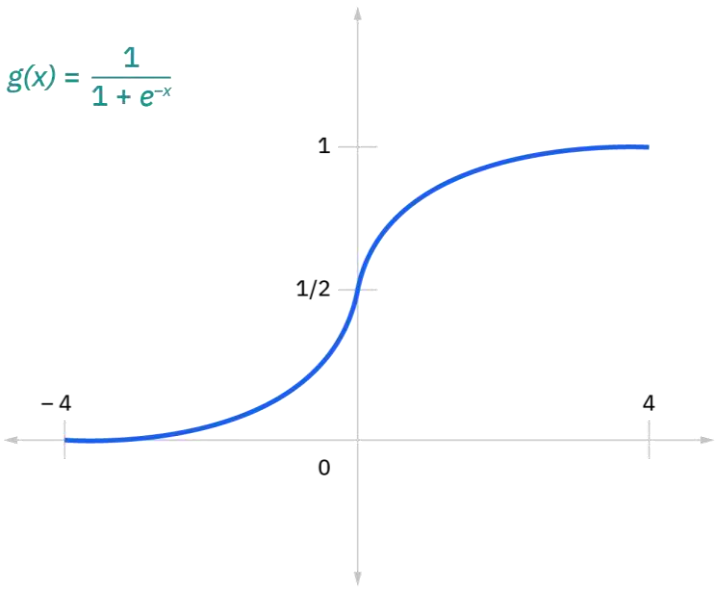

Рисунок 27. Сигмоидная функция активации

Эта функция часто применяется для того, чтобы выдать ответ «да» (1) или «нет» (0), так же, как и для решения других задач, где необходимо выдать принадлежность объекта одному из двух вариантов ответа.

Сигмоидная функция активации может быть добавлена на доску с помощью карточки «Activation» через задание параметра «Функция», равного «sigmoid».

Гиперболический тангенс

Функция гиперболического тангенса очень похожа на сигмоидальную функцию, но она преобразует данные в отрезок $[-1, 1]$.

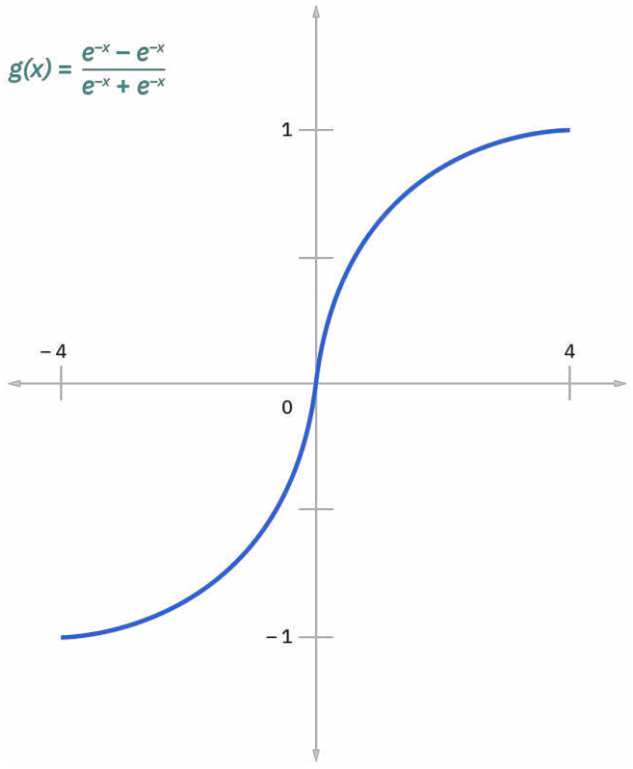

Рисунок 28. Функция активации «гиперболический тангенс»

Функция активации «гиперболический тангенс» может быть добавлена на доску с помощью карточки «Activation» через задание параметра «Функция», равного «tanh».

Линейные функции

ReLU

ReLU – это функция усеченного линейного преобразования. Это одна из наиболее распространенных функций, применяемых в нейронных сетях. Она позволяет избавиться от отрицательных значений во входных данных.

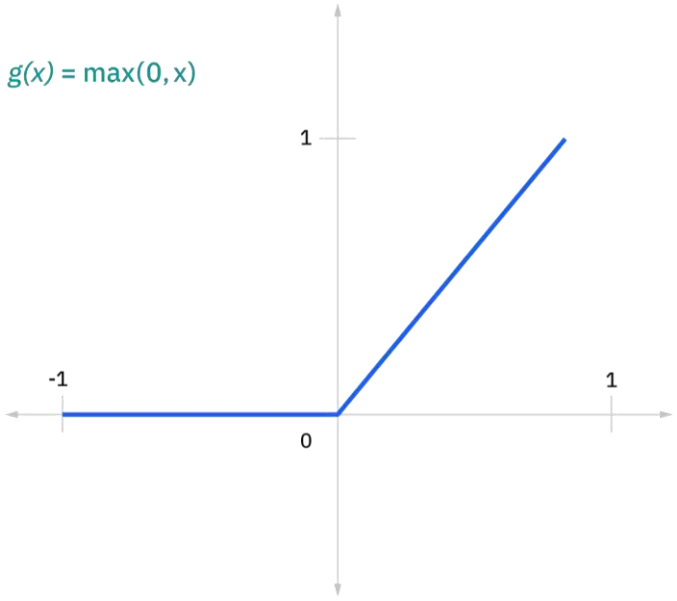

Рисунок 29. Функция активации «ReLU».

ReLU часто используется в сверточных нейронных сетях после каждого слоя свертки. Благодаря этому мы можем гарантировать, что яркость пикселей на карте признаков не будет меньше нуля.

Функция активации **ReLU** может быть добавлена на доску с помощью карточки «ReLU» или с помощью карточки «Activation» через задание параметра «Функция», равного «relu».

Softmax

Функция активации **Softmax** (или мягкий максимум) применяется для того, чтобы для каждого входа определить процент объема, занимаемого этим входом среди всех полученных. Сумма всех выходов после активации **Softmax** = 1.

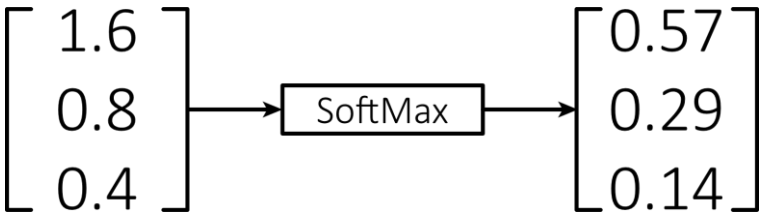

Рисунок 30. Принцип работы функции активации «Softmax».

Softmax часто применяется для того, чтобы получить вероятности принадлежности объекта каждому из возможных вариантов ответа.

Функция активации **Softmax** может быть добавлена на доску с помощью карточки «Softmax» или с помощью карточки «Activation» через задание параметра «Функция», равного «softmax».

Полносвязный слой

Независимо от вида нейронной сети, один из главных ее слоев – это полносвязный слой. Название этого слоя полностью описывает его суть.

В частично связанных слоях входные данные связаны с выходами не напрямую. В полносвязном слое же ситуация обстоит иначе. Он состоит из множества нейронов. Каждый из этих нейронов связан со всеми выходами предыдущего слоя.

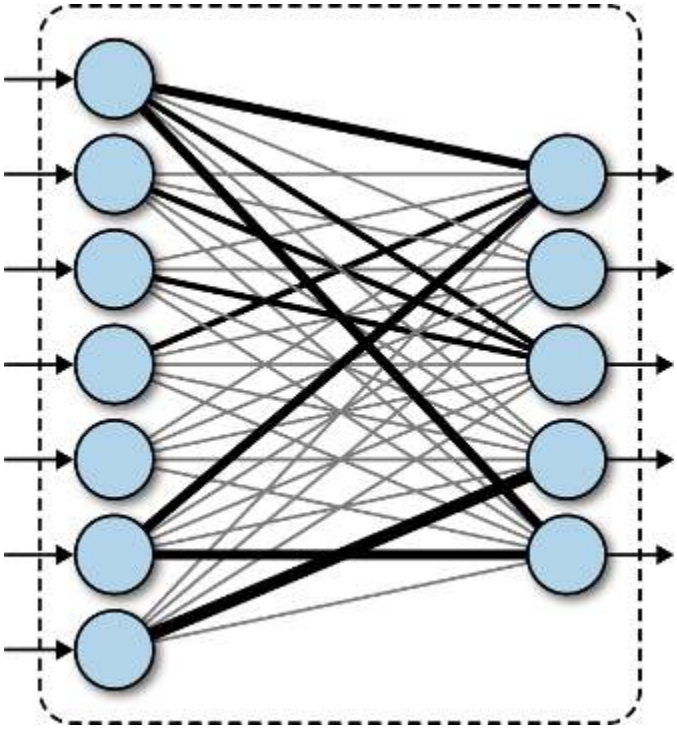

Рисунок 31. Полносвязный слой.

Основная задача полносвязного слоя – аппроксимация сложной функции (или, другими словами, попытка вести себя как другая функция без знания о том какой она является). Можно считать, что слои сверточной сети, идущие до полносвязного, являются средствами предобработки изображения. То есть, все, что идет до полносвязного слоя, используется для выделения различных признаков, которые затем используются для решения поставленной перед нейросетью задачи.

Классификация

В нейронной сети может быть несколько полносвязных слоев. Более ранние слои служат для нахождения зависимостей между данными. Последний полносвязный слой служит для осуществления финального прогноза.

Полносвязный слой является предпоследним слоем нейронной сети и решает задачу **классификации**.

Классификация – это задача определения принадлежности данных одному из заранее заданных классов.

Бинарная классификация

Если классов всего два, то такая классификация называется **бинарной** или **двоичной**.

Последний полносвязный связный слой в таком случае содержит лишь один нейрон. После полносвязного слоя используется одна из нелинейных активаций.

Категориальная классификация

Если классов больше двух, то такая классификация называется **категориальной** или **многоклассовой**.

Последний полносвязный связный слой в таком случае содержит количество нейронов, равное количеству классов в наборе данных. Последним слоем нейросети является функция активации **Softmax**.

Полносвязный слой в NNWizard

Полносвязный слой представлен в **NNWizard** в виде карточки `Dense` в окне моделирования нейросети.

Параметры полносвязного слоя

Количество нейронов в полносвязном слое задается параметром «Units» в панели редактирования карточки слоя.

Слой распрямления

Каждый из нейронов полносвязного слоя соединяется со всеми выходными слоями предыдущего слоя. Однако, слои свертки и пулинга возвращают карты признаков, являющиеся матрицами.

Чтобы иметь возможность использовать выход этих слоев в качестве входа полносвязного слоя, мы должны превратить матрицы – двумерные сетки данных – в вектор – колонку данных. Для этого используется слой, решающий только эту задачу, – слой распрямления.

Слой распрямления в NNWizard

Слой распрямления представлен в **NNWizard** в виде карточки «Flatten» в окне моделирования нейросети.

Слой нормализации

Слои нормализации служат для масштабирования данных, отправляемых в слои активации. В результате работы этих слоев мы получаем нормализованные данные, приведенные к одному диапазону, что позволяет ускорить обучение нейронной сети.

Нормализация смягчает эффект исчезающего градиента на различных значениях входных слоев. Нормализуя выход нейронов, мы будем получать на входах только значения, близкие к нулю.

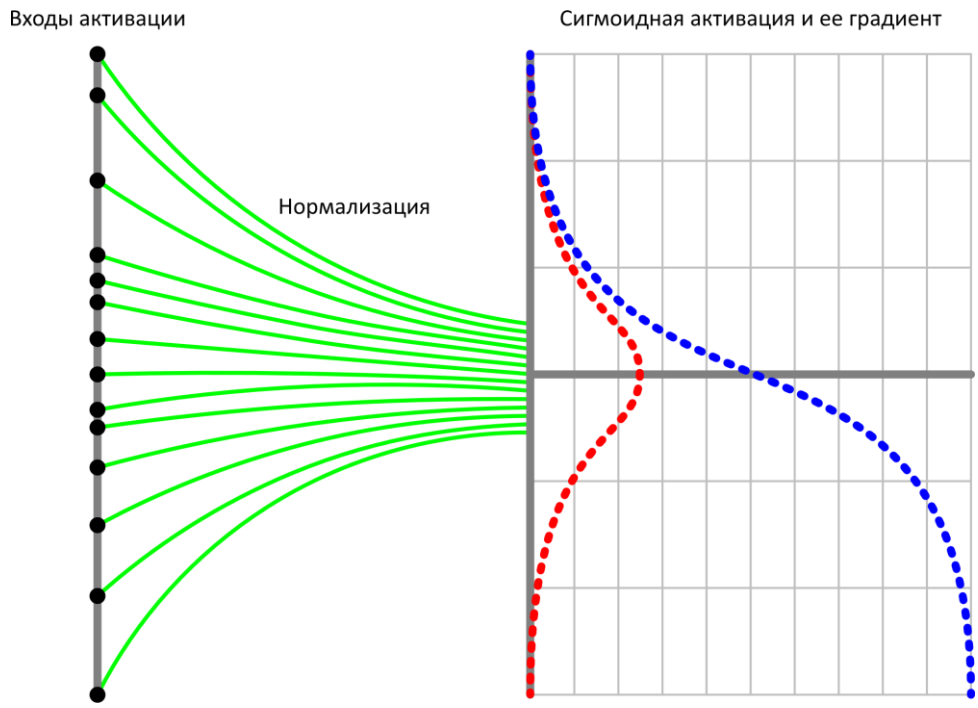

Рисунок 32. Нормализация.

BatchNormalization

Слой **BatchNormalization** (или же «пакетная нормализация») представляет собой выполнение алгоритма для нормализации данных – приведения их к виду случайной величины с нулевым математическим ожиданием и единичной дисперсией.

Пакетная нормализация называется «пакетной», потому что мы проводим вычисления статистик и изменения только на определенном подмножестве (пакете) всего набора данных.

LayerNormalization

Слой **LayerNormalization** нормализует активации предыдущего слоя для каждого примера в пакете независимо, в отличие от **BatchNormalization**, где нормализация применяется ко всему пакету в целом.

Слой регуляризации

Регуляризация – это метод добавления некоторых дополнительных ограничений с целью предотвратить переобучение нейронной сети.

Во время обучения может возникнуть ситуация, когда нейронная сеть обучится так хорошо, что при найденных значениях весов функция потерь будет почти равняться нулю.

Dropout

Слой исключения нейронов – это вычислительно дешевый способ регуляризации нейронной сети. Он случайным образом выбирает нейроны предыдущего слоя и выключает их выходы. Это позволяет имитировать большое количество нейросетей с разной структурой. Такой подход делает отдельные нейроны более устойчивыми к различным входам.

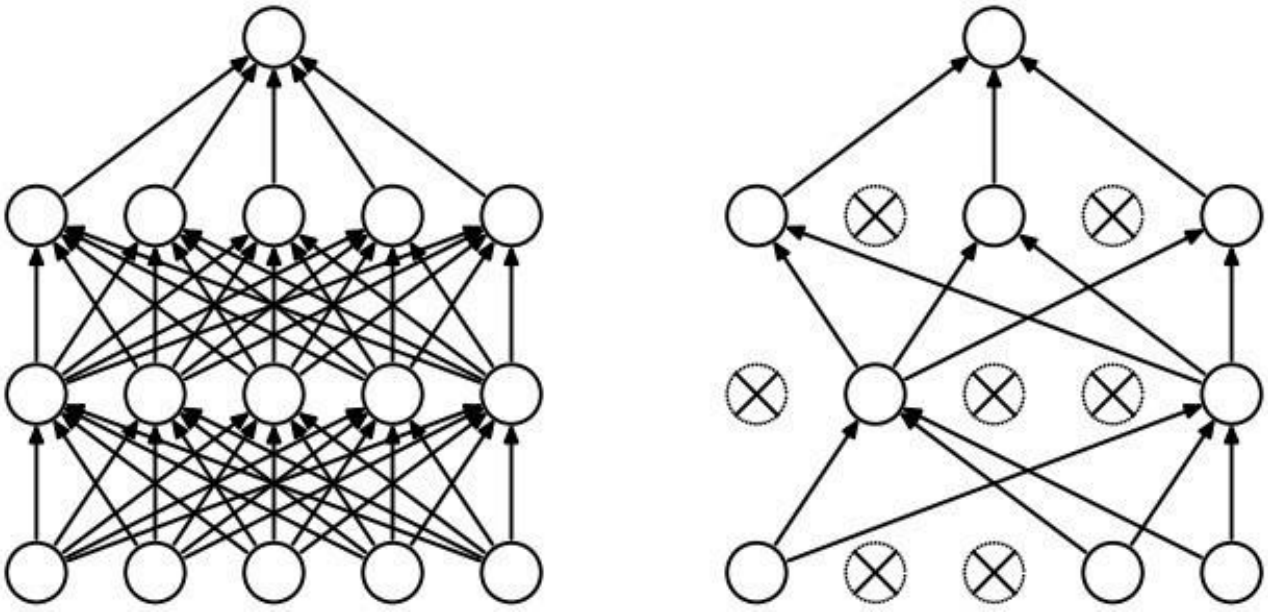

Рисунок 33. Скрытые слои нейронной сети до и после применения дропаута.

В нейронной сети переобучение может появляться из-за совокупной адаптации. Во время обновления веса отдельного нейрона учитывается деятельность остальных нейронов. Из-за этого изменение веса одного нейрона может влиять на веса других нейронов, приводя к переобучению.

Метод выключения нейронов позволяет предотвратить такую адаптацию.

> Заметьте, что слой исключения применяется только во время обучения. Во время работы нейросети используются все нейроны.

Слой исключения в NNWizard

Слой исключения представлен в **NNWizard** в виде карточки «Dropout» в окне моделирования нейросети.

Параметры слоя исключения

Единственным параметром слоя исключения является процент выключаемых нейронов. Он задается параметром «Rate» в панели редактирования карточки слоя.

> Обратите внимание, что процент задается в виде вещественного числа от 0 до 1, где 0.2 соответствует 20%.

Как изменить гиперпараметр слоя

Чтобы открыть параметры слоя, дважды нажмите на него левой кнопкой мыши. После этого в правой части окна моделирования появятся параметры выбранного слоя.

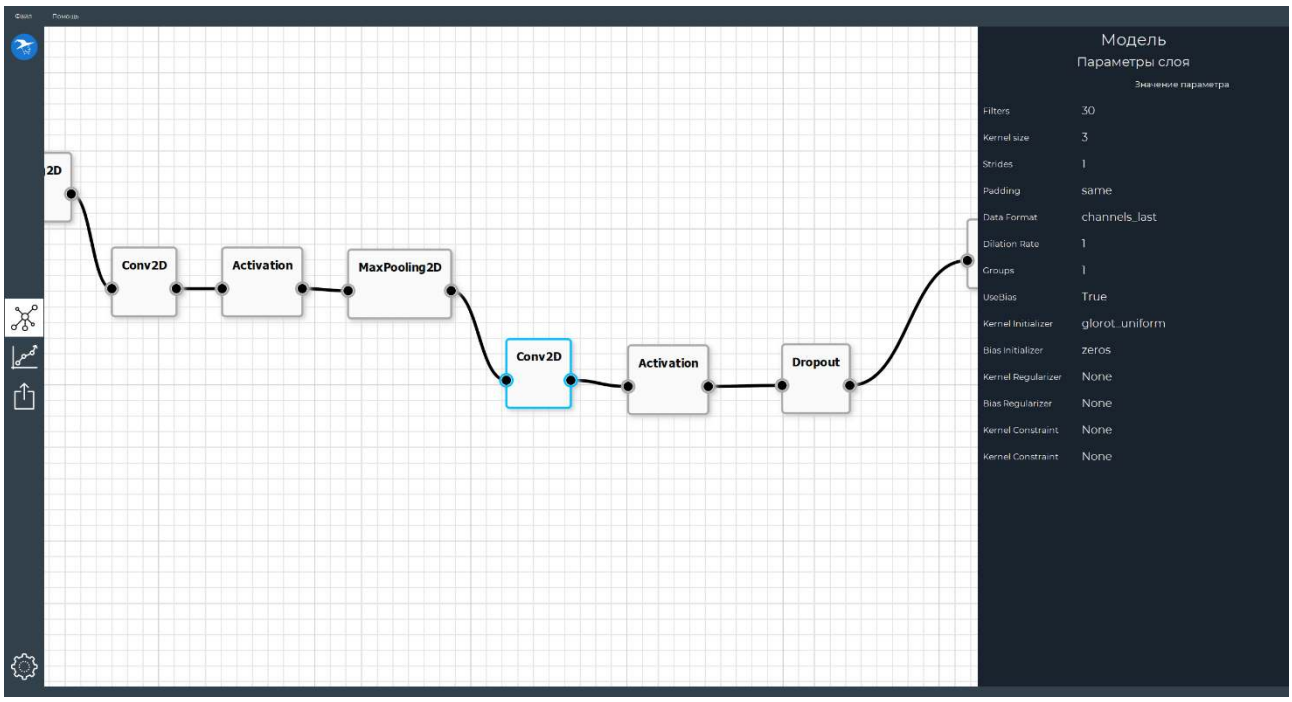

Рисунок 34. Параметры слоя.

Каждый параметр находится на новой строке, состоящей из двух колонок: первая – имя параметра, вторая – его значение. Все параметры инициализированы значением по умолчанию.

Для изменения значения параметра дважды нажмите на значение параметра левой кнопкой мыши, обновите значение и нажмите клавишу Enter.

Рисунок 35. Редактирование значения параметра.

Если вы не знаете предназначение параметра и его допустимые значения, наведите мышь на поле значения параметра, после чего рядом появится подсказка.

Рисунок 36. Подсказка значения параметра.

Обучение нейронной сети

Настройка параметров обучения так же, как и моделирование, представлена в виде доски, на которой создается граф. Работа с доской аналогична работе с доской на этапе моделирования архитектуры. Однако, граф обучения не последовательный и имеет тип «многие к одному».

На данный момент единственный доступный тип модели – классифицирующая, то есть такая, которая по анализу целого изображения возвращает его принадлежность какому-либо классу.

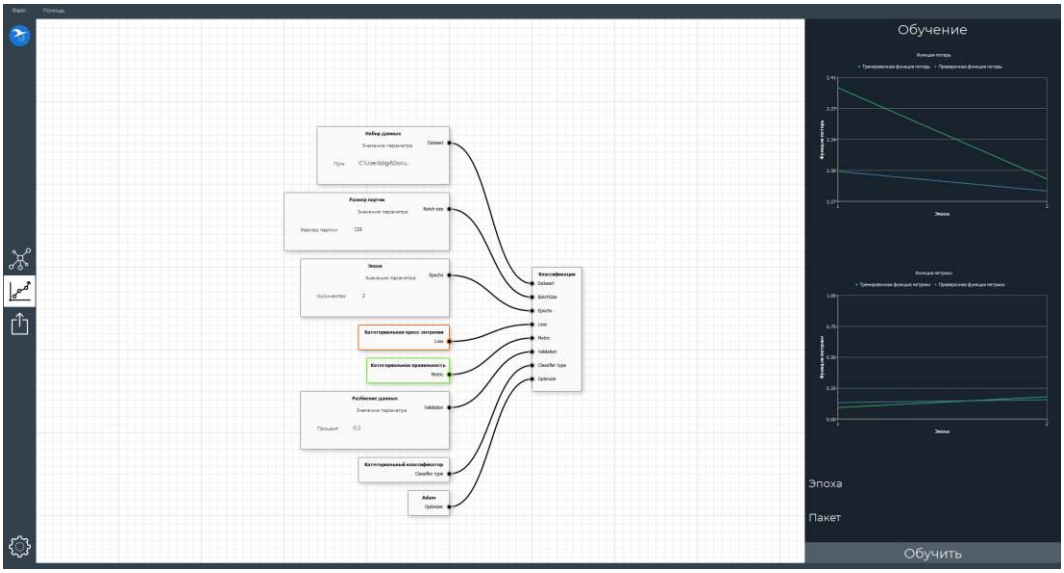

Рисунок 37. Пример графа обучения категориальной модели классификатора.

Параметры классификатора

Данный классификатор принимает несколько параметров.

Dataset

Вход «Dataset» отвечает за то, откуда нейронная сеть будет брать данные. То есть, где лежит набор данных для обучения и проверки.

В качестве карточки, идущей ко входу «Dataset», используется карточка «Набор данных» («Dataset»). В ней есть поле «Путь», в которое необходимо вписать абсолютный путь к набору данных.

> Заметьте, что изображения должны быть разделены по папкам, носящим названия классов, на которые нейронная сеть будет учиться их разделять. Названия папок должны содержать только латинские буквы и цифры.

Например, пусть имеется набор данных объектов разных цветов: белых, синих и красных. Тогда нужно создать папку «Colors» и расположить их в следующей иерархии:

Colors

- Blue – изображения с синими объектами
- Red – изображения с красными объектами
- White – изображения с белыми объектами

Тогда в качестве значения поля «Путь» карточки «Набор данных» будет использоваться абсолютный путь к папке «Colors».

Batch size

Вход «Batch size» отвечает за размер пакета, то есть, за количество изображений, помещаемых в один пакет.

Чем меньше размер пакета, тем больше сеть адаптируется под конкретные образцы изображений и тем больше потребуется времени для получения точной модели.

> 💡 Старайтесь использовать большие размеры партии (например, ≥ 256).

В качестве карточки, идущей ко входу «Batch size», используется карточка «Размер партии» («BatchSize»). Размер пакета указывается в поле «Размер партии»

> ❗ Большой размер пакета требует большое количество оперативной/видео памяти. Если процесс обучения долгое время не начинается после нажатия кнопки «Обучить», попробуйте сократить размер партии до более маленьких значений (≈ 16).

Epochs

Вход «Epochs» отвечает за количество эпох обучения – количество раз, за которое весь набор данных пройдет нейронную сеть от начала до конца. Чем больше количество эпох, тем более точно обучится сеть.

> ❗ При слишком большом значении количества эпох сеть может переобучиться (запомнить набор данных и ответы к нему, потеряв возможность правильно работать на новых данных).

В качестве карточки, идущей ко входу «Epochs», используется карточка «Эпохи» («Epochs»). Количество эпох указывается в поле «Количество».

Loss

Вход «Loss» отвечает за используемую функцию потерь.

Предназначение функции потерь – вычислить некоторое число, которое модель нейронной сети должна стремиться минимизировать во время процесса обучения.

Для классификации используется функция потерь «**кросс-энтропия**». Если классификация бинарная (когда мы имеем всего 2 класса), то используется карточка «Бинарная кросс-энтропия» («BinaryCrossentropyLoss»). Если же классификация категориальная, то используется карточка «Категориальная кросс-энтропия» («CategoricalCrossentropyLoss»).

Metric

Вход «Metric» отвечает за используемую функцию метрики.

Функция метрики используется для оценки обученности модели.

> 💡 Функция метрики похожа на функцию потерь, но ее значения не используются для обучения.

Наиболее часто для задачи классификации используется функция метрики «**правильность**». Если классификация бинарная, то используется карточка «Бинарная правильность» («BinaryAccuracyMetric»). Если же классификация категориальная, то используется карточка «Категориальная правильность» («CategoricalAccuracyMetric»).

Validation

Вход «Validation» отвечает за валидационное разбиение, то есть, какой процент данных из датасета будет использован для проверки нейросети.

Для проверки правильности обученной модели исходный набор данных делится на два множества: тренировочные и проверочные. На тренировочном наборе сеть обучается. На тестовом – проверяется то, насколько хорошо обученная сеть работает на данных, которых она никогда не видела.

В качестве карточки, идущей ко входу «Validation», используется карточка «Разбиение данных» («ValidationSplit»). Процент валидационных данных указывается в поле «Процент».

Classifier type

Вход «Classifier type» отвечает за тип классификатора. Если классификация бинарная, то типом классификатора будет карточка «Бинарный классификатор» («BinaryClassifier»). Если же классификация категориальная, то типом классификатора будет карточка «Категориальный классификатор» («CategoricalClassifier»).

Optimizer

Вход «Optimizer» отвечает за используемый оптимизатор.

Оптимизатор используется для нахождения глобального минимума функции потерь.

SGD

Оптимизатор, основанный на стохастическом градиентном спуске.

RMSprop

Оптимизатор, реализующий алгоритм RMSprop.

RMSprop:

- Поддерживает скользящее среднее квадратов градиентов;
- Делит градиент на корень этого среднего.

Adam

Оптимизатор стохастического градиентного спуска, основанного на адаптивной оценке импульсов первого и второго порядка.

Вычислительно эффективный, не требующий много оперативной памяти и отлично приспособленный для больших задач (в терминах данных/параметров).

Adagrad

Оптимизатор с параметрически-заданной скоростью обучения, которая адаптируется относительно того, как часто параметр обновляется в течение процесса обучения.

Adadelta

Более устойчивая версия оптимизатора Adagrad.

Adamax

Вариант оптимизатора Adam, основанный на бесконечной норме. В некоторых случаях превосходит Adam.

Nadam

Вариант оптимизатора Adam с импульсом Нестерова.

Ftrl

Оптимизатор, реализующий алгоритм Follow the (Proximal) Regularized Leader.

Обучение

При нажатии на кнопку «Обучить», начнется процесс обучения, состоящий из нескольких этапов.

Подготовка данных

На данном этапе набор данных загружается в генератор и разделяется на два подмножества: для обучения и валидации;

Обработка данных

На данном этапе изображения проверяются на целостность и предпринимается попытка восстановить поврежденные изображения.

> ! Во время работы данного этапа возможно «замораживание» интерфейса – в таком случае нужно дождаться начала следующего этапа. Это связано с выделением оперативной памяти под набор данных.

Как только процесс загрузки в оперативную память будет завершен, полосы прогресса обновятся и значения в них будут расти.

Обучение

Данные начинают обрабатываться нейронной сетью и происходит процесс ее обучения.

Становятся активными счетчики прогресса эпох и пакетов. Кнопка «Обучить» меняет состояние на «Остановить обучение».

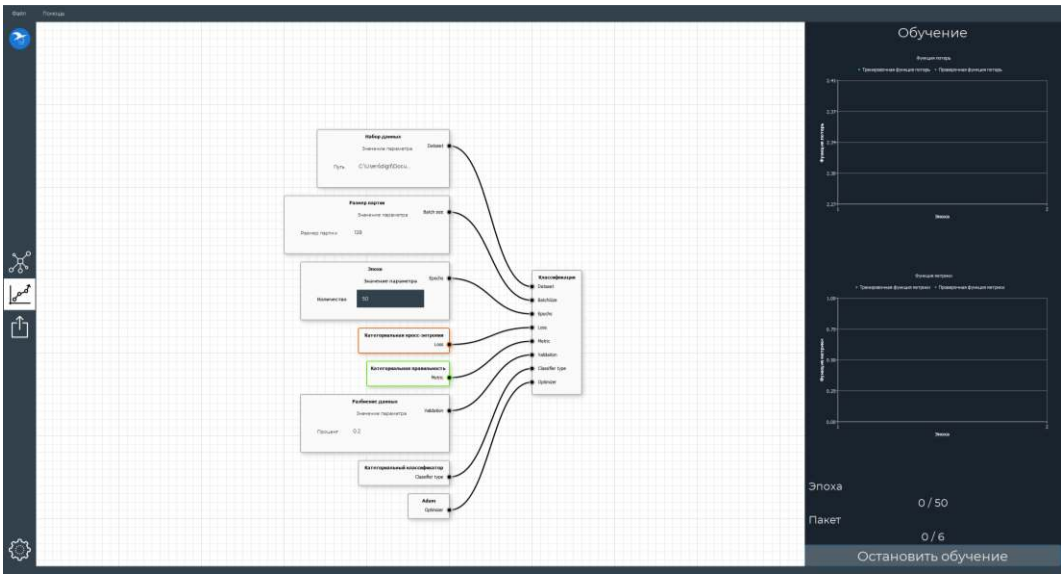

Рисунок 38. Начало процесса обучения.

Полоса «Эпоха» показывает сколько эпох обучения пройдено. Полоса «Пакет» показывает сколько пакетов прошло через нейронную сеть.

Когда полоса «Пакет» приблизится к концу, произойдет замедление – во время него вычисляются функция потерь и функция метрики на тренировочной и валидационной выборках. После окончания каждой эпохи будут показаны значения обучающих и

валидационных функций потерь и метрики. Начиная со второй эпохи, графики будут обновляться и показывать визуальную информацию о том, какие значения на функциях мы получаем.

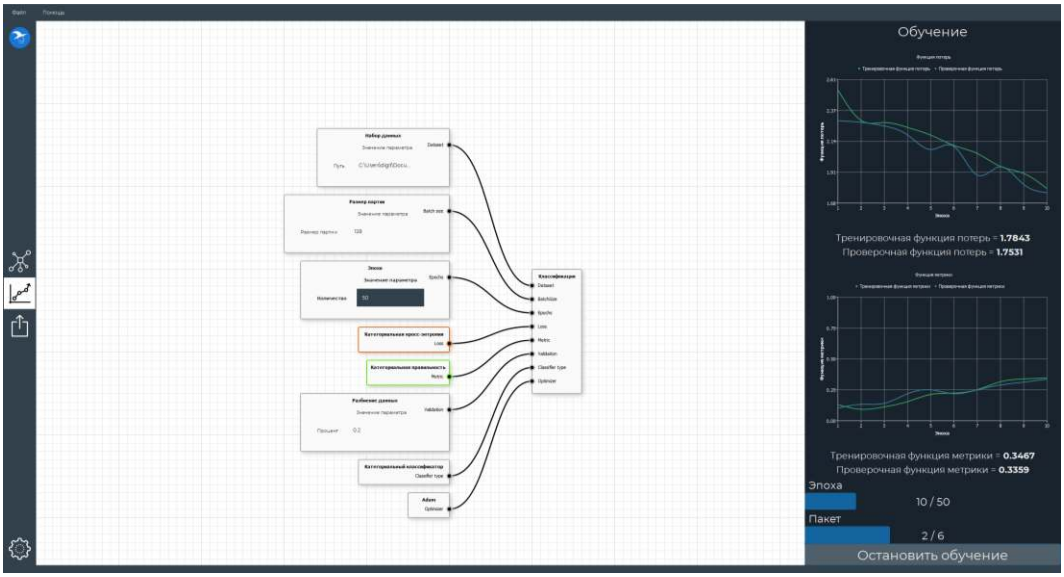

Рисунок 39. Графики функции потерь и метрики после десяти эпох обучения.

Синим цветом отображается функция, посчитанная на тренировочном наборе данных, зеленым – на проверочном. По различиям в графиках тренировочного и проверочного наборов можно судить о переобучении модели.

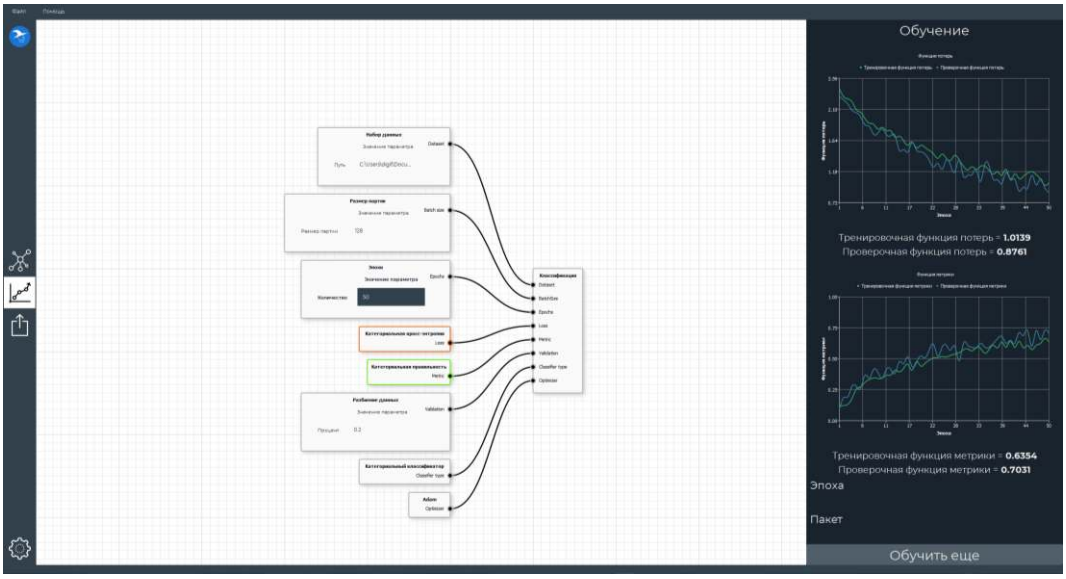

Рисунок 40. Графики функций потерь и метрики после 50 эпох обучения.

Остановка обучения

Обучение можно остановить принудительно. Для этого необходимо нажать кнопку «Остановить обучение». Обучение остановится, когда текущая эпоха подойдет к концу.

Дообучение

После принудительной остановки или планового окончания обучения модель может быть дообучена. Для этого необходимо нажать кнопку «Обучить еще».

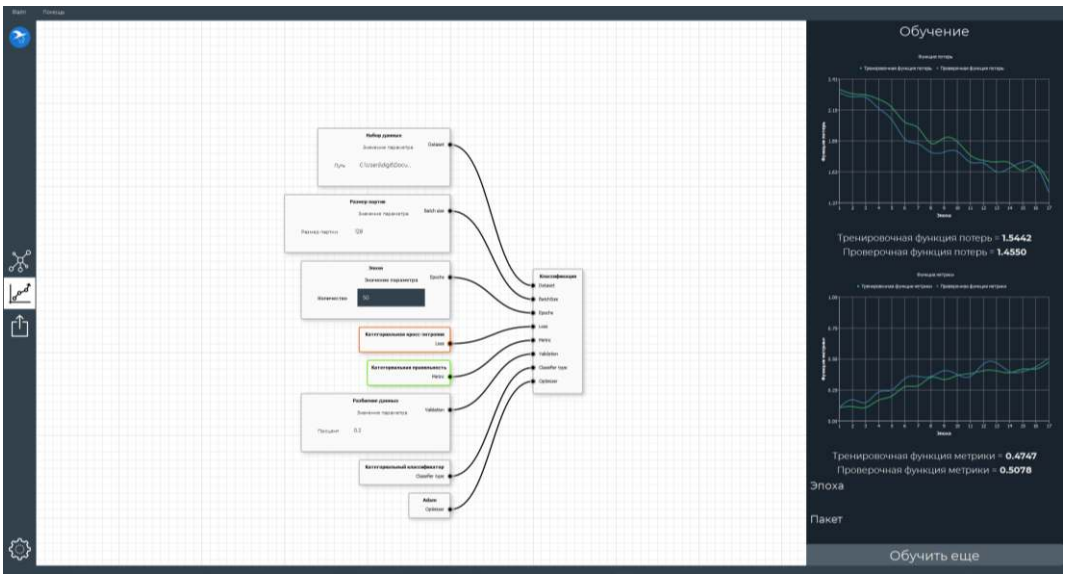

Рисунок 41. Кнопка "Обучить еще".

Чтобы сохранить возможность дообучения между сессиями, сохраните проект после окончания обучения. Графики обучения также будут сохранены и возобновлены при следующем открытии проекта.

При изменении узлов модели, их параметров и соединений между ними, возможность дообучения в текущей сессии пропадает. В результате таких действий кнопка «Обучить еще» сменит свое состояние на «Обучить», что приведет к обучению с нуля и очистке графиков.

! Если провести дообучение и не сохранить проект, при следующем его открытии возможность дообучения пропадет. Чтобы избежать такого поведения, не забывайте сохранять проект.

Обучение на GPU

NNWizard поддерживает возможность обучения нейронной сети на графическом адаптере, что заметно увеличивает скорость обучения.

Для обучения на **GPU** необходимо:

- Наличие подключенного к компьютеру дискретного графического адаптера от **NVIDIA®**;
- Графический адаптер должен иметь поддержку **CUDA®** (Compute capability ≥ 3.0);
- На компьютере должны быть установлены:
  - **Драйвер графического адаптера NVIDIA®** версии не ниже 450.x;
  - **CUDA® Toolkit v10.1**;
  - **cuDNN SDK v7.6.5**.

Если эти условия не выполняются, то обучение нейронной сети будет происходить на **CPU**.

Установка CUDA® Toolkit

Инструкцию по установке **CUDA® Toolkit** см. **https://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/**

Установка cuDNN SDK

Распакуйте скачанный архив в папку «C:\tools\cuda».

Добавление путей в переменную PATH

Чтобы **NNWizard** мог найти установленную версию **CUDA® Toolkit**, необходимо добавить путь установки в системную переменную среды PATH. Для этого:

1. Откройте интерфейс командной строки;
   1. Нажмите клавиатурное сочетание Win+R;
   2. Введите «cmd»;
   3. Нажмите клавишу «Enter».
2. Поочередно введите следующие команды, завершая каждую нажатием клавиши «Enter»:
   1. SET PATH={Путь установки CUDA® Toolkit}\bin;%PATH%
   2. SET PATH={Путь установки CUDA® Toolkit}\ extras\CUPTI\lib64;%PATH%
   3. SET PATH={Путь установки CUDA® Toolkit}\include;%PATH%
   4. SET PATH={Путь установки cuDNN SDK}\bin;%PATH%
3. Закройте интерфейс командной строки.

Экспорт нейронной сети

После процесса обучения обученную модель можно экспортировать для использования на модуле «**Артинтрек**».

> ! Процесс экспорта должен быть запущен при активном проекте на модели, проходившей обучение хотя бы один раз (в течение текущей рабочей сессии или в одной из предыдущих).

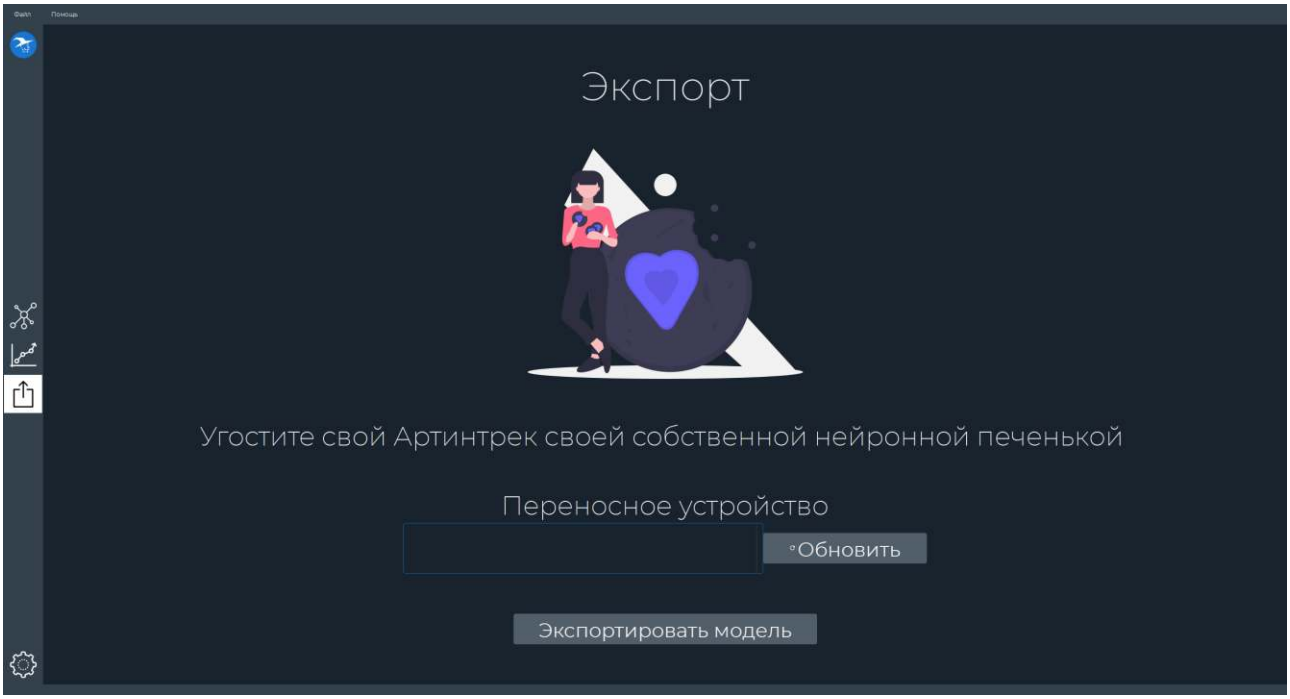

Рисунок 42. Окно экспорта обученной модели нейронной сети.

Для экспорта необходимо подготовить устройство – **USB** Flash-накопитель с именем «NNWIZARD».

Подготовка носителя

Если у вас нет такого устройства, выполните следующие действия:

1. Возьмите USB Flash-накопитель и вставьте его в USB-порт компьютера;
2. Откройте «Проводник» и перейдите к окну «Мой компьютер»;
3. Найдите вставленный накопитель в списке устройств;
4. Нажмите на нем правой кнопкой мыши и выберите «Форматировать...»;
5. В селекторе «Файловая система» выберите «FAT32»;
6. В селекторе «Размер единицы распределения» выберите «Стандартный размер кластера»;
7. В поле «Метка тома» введите «NNWIZARD»;
8. Поставьте галочку «Быстрое (очистка оглавления)» в группе «Способы форматирования»;
9. Нажмите кнопку «Начать»;
10. Дождитесь окончания форматирования.

Экспортирование

1. После того, как вы вставили в компьютер Flash-носитель, нажмите кнопку «Обновить»;
2. В списке выбора носителя появятся все носители, названные «NNWIZARD», с их буквенными обозначениями;
   Если в списке не появилось ни одного элемента, попробуйте повторить поиск, снова нажав кнопку «Обновить». Если после повторения процедуры список по-прежнему пуст, возможно, носитель был неправильно подготовлен – попробуйте повторить **процедуру подготовки носителя**.
3. Выберите в списке нужный носитель;
4. Нажмите кнопку «Экспортировать модель»;
5. На экране появится окно с непрерывной полосой прогресса;
6. Когда обучение будет завершено, появится информационное окно, сигнализирующее об окончании экспорта. Нажмите кнопку «OK».

После выполнения данной процедуры на носителе должны появиться:

- Файл «config.json» – файл конфигурации модели;
- Папка «model», содержащая три файла:
  - «labels.txt» – файл, содержащий метки классов;
  - «custom_*_classifier.xml» – файл с описанием архитектуры нейронной сети;
  - «custom_*_classifier.bin» – файл с весами обученной нейронной сети;

Импорт обученной модели на Артинтрек

Для импорта обученной модели на модуль «**Артинтрек**», выполните следующую процедуру:

1. Включите модуль «**Артинтрек**»;
2. Дождитесь загрузки рабочего стола;
3. Вставьте накопитель с экспортированной моделью в один из USB-портов модуля;
4. Дождитесь появления окна выбора действий с накопителем и подождите около одной минуты;
   После данного этапа все необходимые файлы будут автоматически скопированы на модуль «**Артинтрек**» и вы сможете использовать вашу модель.

   ! Операция копирования перезапишет предыдущую модель.

5. Отсоедините накопитель от модуля «**Артинтрек**».