VI Международный конкурс исследовательских работ школьников
«Research start» 2023-2024

Научно-исследовательский проект

«СОЗДАНИЕ НЕЙРОННОЙ СЕТИ И ЕЁ ОБУЧЕНИЕ ИГРЕ ПОСРЕДСТВОМ ГЕНЕТИЧЕСКОГО АЛГОРИТМА»

Выполнил учащийся
11 класса МБОУ СОШ № 103
муниципального образования
город Краснодар
Ташу
Мадин Юрьевич

Научный руководитель:
учитель информатики
МБОУ СОШ № 103
Попцова Алена Александровна

Краснодар
2024

Содержание

Введение

Проблема: можно ли обучить нейронную сеть игре?

Область: кибернетика.

Цель: создать нейронную сеть и проверить её обучаемость на примере обучения игре.

Задачи:

1) Анализ понятий, изучение литературы и интернет-ресурсов с целью получения сведений о принципах разработки нейронных сетей.
2) Разработка прикладного программного обеспечения.
3) Получение экспертной оценки.

Гипотеза: по нашему мнению, нейронную сеть можно обучить игре.

Актуальность

В эпоху цифрового общества всё большее значение обретают технологии искусственного интеллекта. На данный момент они используются для решения большого спектра задач, особенно тех, для которых алгоритмические решения либо неэффективны, либо невозможны. Результаты работы искусственного интеллекта становятся более совершенными, его использование обретает более широкий характер. Машинам даже стали под силу задачи, которыми раньше занимался только человек.

История искусственного интеллекта берёт начало со второй половины 20 века. За прошедшее время разработаны различные реализации ИИ, включая нейронные сети, а также множество подходов к их обучению. Н. Винер в своём труде «Кибернетика или управление и связь в животном и машине» исследует, как сделать работу нейронной сети доступной явному описанию, какие задачи

она способна решать. Однако никто не пытался ответить на эти вопросы на основе результатов обучения нейронной сети игре.

Между искусственным интеллектом и искусственным поведением существует большая разница. **Практическая значимость состоит в том, чтобы сделать нейронную сеть, результат работы которой имитирует действия человека.** Мы не хотим, чтобы наша нейронная сеть имела преимущество перед настоящими игроками. Требуется, чтобы она была настолько умной, насколько это необходимо, чтобы не выходить за пределы доступного поведения, как зачастую происходит.

План:

№	Вид деятельности	Сроки		Ожидаемый результат
		Начало	Окончание	
1	Дать определение понятиям: алгоритм, программное обеспечение, перцептрон, нейронная сеть, метод обратного распространения ошибки, генетический алгоритм	Сентябрь 2022	Октябрь 2022	Обзор основных понятий: алгоритм, программное обеспечение, перцептрон, нейронная сеть, метод обратного распространения ошибки, генетический алгоритм
2	Анализ существующих архитектур нейронных сетей	Октябрь 2022		Обзор существующих архитектур нейронных сетей
3	Анализ существующих методов селекции, мутации и кроссинговера для	Октябрь 2022	Ноябрь 2022	Обзор существующих методов селекции, мутации и

	генетического алгоритма			кроссинговера
4	Разработка нейронной сети прямого распространения	Ноябрь 2022	Декабрь 2022	Нейронная сеть прямого распространения
5	Разработка генетического алгоритма	Декабрь 2022	Февраль 2023	Генетический алгоритм
6	Интеграция нейронной сети и генетического алгоритма	Февраль 2023	Март 2023	Готовая к обучению нейросеть и программа, графически представляющая резальтат работы нейронной сети.
7	Обучение нейронной сети	Март 2023	Март 2023	Обученная нейронная сеть, информация о ходе обучения и оптимальных настройках обучения
8	Определение экспертной группы	Апрель 2023		Состав экспертной группы.
9	Создание экспертного листа	Апрель 2023		Экспертный лист.
10	Получение экспертной оценки проекта	Апрель 2023		Экспертная оценка проекта.

Глава 1. Теоретическая часть

1.1 Описание методов проекта

В ходе работы над проектом были определены следующие методы:

Обзор специальной литературы — анализ использованной литературы, формулировка основных идей и тенденций, использование материала для обоснования теоретической базы исследования. С помощью этого методы были даны определения основных понятий проекта.

Компьютерное моделирование — метод решения задачи анализа или синтеза сложной системы на основе изучения её компьютерной модели. Благодаря этому методу получены количественные и качественные данные о процессе и результатах обучения создаваемой нейронной сети.

Экспертная оценка — это метод поиска и результат его применения, полученный на основании использования персонального мнения экспертов или коллективного мнения группы экспертов. Путём экспертной оценки было проверено качество созданного продукта.

Также были определены теоретические методы:

Синтез — процесс практического или мысленного воссоединения целого из частей или различных элементов в единое целое. Использовался, чтобы продолжить разработку на основе уже полученных сведений из области информационных технологий.

Анализ — процедура мысленного разложения предмета (явления, процесса), его свойств или отношения между предметами на части. Использовался, чтобы разделить процесс реализации проекта на меньшие части.

Сравнение — метод сопоставления двух объектов, явлений, результатов и т.п. с целью выделения из них общих и различных качеств, их классификации и типологии. С помощью этого метода на основе данных разных этапов обучения нейронной сети выяснена динамика её развития.

Обобщение — мысленный переход от отдельных фактов и событий к более общим. Этот метод использовался, чтобы сформировать вывод по проекту.

1.2. Описание основных понятий

1.2.1. Алгоритм

Исторически сложилось так, что термин алгоритм не имеет точного значения и в различных областях математики используется либо в общем смысле (план действий), либо в некотором специальном смысле. В нашем случае алгоритм – последовательность команд, предназначенная исполнителю, в результате выполнения которой он должен решить поставленную задачу. Исполнитель должен уметь выполнять все команды, множество которых конечно и изначально строго задано[1].

Хотя у слова «алгоритм» существует несколько определений, они всегда удовлетворяют следующим общим требованиям[1]:

1) Дискретность — алгоритм представляет собой набор упорядоченных шагов, на выполнение каждого шага требуется определённое количество времени.
2) Детерминированность — результат работы алгоритма всегда одинаков для идентичных входных данных.
3) Понятность — алгоритм включает в себя только команды, доступные исполнителю.
4) Завершаемость — при правильно заданных входных данных алгоритм завершает свою работу за определённое количество шагов.
5) Массовость — алгоритм должен быть применим к разным наборам входных данных.
6) Результативность — завершение алгоритма даёт определённые результаты.

Также все алгоритмы могут быть поделены на вычислительные — те, которые путём некоторых вычислений преобразуют одни данные в другие, и управляющие, осуществляющие управляющие взаимодействия при неких условиях[1].

1.2.2. Программное обеспечение

Программа - данные, предназначенные для управления конкретными компонентами системы обработки данных в целях реализации определённого алгоритма. Конечная цель любой компьютерной программы – управление

аппаратными средствами. Даже если на первый взгляд программа не взаимодействует с оборудованием, не требует никакого ввода данных с устройств ввода и не осуществляет вывод данных на устройства вывода, все равно ее работа основана на управлении аппаратными устройствами компьютера. Работа компьютерной системы осуществляется в непрерывном взаимодействии аппаратных и программных средств[2].

Программное обеспечение - компьютерные программы, процедуры и, возможно, соответствующая документация и данные, относящиеся к функционированию компьютерной системы. Программное обеспечение выполняет следующие основные функции[3]:

1) обеспечивает работоспособность ЭВМ, так как без соответствующего ПО компьютеры не могут осуществлять никакие операции;
2) расширяет ресурсы вычислительной системы и повышает эффективность их использования;
3) облегчает взаимодействие пользователя с ЭВМ и повышает производительность его труда, т. е. обеспечивает пользовательский интерфейс.

Подходы к классификации ПО достаточно подробно формализованы в международном стандарте ISO/IEC 12182. В частности, первая версия стандарта предусматривала 16 критериев классификации программных средств:

1. по режиму эксплуатации;
2. по масштабу;
3. по стабильности;
4. по функции;
5. по требованию защиты;
6. по требованию надёжности;
7. по требуемым рабочим характеристикам;
8. по исходному языку;
9. по прикладной области;
10. по вычислительной системе и среде;

11. по классу пользователя;
12. по требованию к вычислительным ресурсам;
13. по критичности;
14. по готовности;
15. по представлению данных;
16. по использованию программных данных.

1.2.3. Нейронная сеть

Нейронная сеть (также ИНС) — искусственные многослойные высокопараллельные логические структуры, составленные из формальных нейронов[5]. Воплощение ИНС обычно представляет собой систему соединённых и взаимодействующих простых процессоров (например, перцептронов), работающих с разного рода сигналами. Сами перцептроны представляют собой математическую или компьютерную модель восприятия информации мозгом (кибернетическую модель мозга). Предложенный Фрэнком Розенблаттом в 1958 году, он стал одной из первых моделей нейросетей. Состоит из трёх элементов: датчики, ассоциативные и реагирующие элементы (соответственно S, A и R). Подобная структура позволяет создать набор «ассоциаций» между входными стимулами и необходимой реакцией[4].

Главное отличие нейронных сетей от алгоритмов — возможность обучаться. Суть процесса обучения состоит в поиске оптимального решения поставленной задачи путём изменения взаимодействия одних нейронов на другие. Это значит, что в случае успеха ИНС найдет зависимости между входными и выходными данными и сможет работать даже с частично искажённой или отсутствовавшей в обучающей выборке информацией[6].

1.2.4. Понятия, связанные с обучением ИНС

Первый рассматриваемый алгоритм обучения — градиентный спуск — это один из методов обучения с учителем. Он представляет собой итеративный алгоритм оптимизации для нахождения локального минимума дифференцируемой функции. Идея состоит в том, чтобы делать повторные шаги в направлении, противоположном градиенту функции в текущей точке, так как

это направление самого крутого спуска[8].

Сам градиент – это вектор, своим направлением указывающий направление скорейшего роста некоторой скалярной величины φ, значение которой меняется от одной точки к другой, образуя скалярное поле. По модулю градиент равен скорости роста величины φ в направлении роста вектора. Пространство, на котором определена функция и её градиент, может быть как обычным трёхмерным пространством, так и пространством любой размерности[7]. Для его нахождения применяют метод обратного распространения ошибки — метод вычисления градиента, используемый при обновлении весов многослойного нейронной сети с целью минимизировать ошибку при работе ИНС. Однако, он применим только в случае возможности дифференцировать передаточную функцию нейронов[9].

Ещё один способ обучить ИНС — генетический алгоритм. Это один из методов обучения нейронной сети без учителя, являющийся адаптивным методом поиска, который в последнее время всё чаще используется для решения задач оптимизации. В основе его действия лежат как аналог механизма генетического наследования, так и аналог естественного отбора. При этом сохраняется биологическая терминология в упрощенном виде и основные понятия линейной алгебры. Генетические алгоритмы относят к области мягких вычислений. Это понятие объединяет такие области, как нечеткая логика, нейронные сети, вероятностные рассуждения, сети доверия и эволюционные алгоритмы, которые дополняют друг друга и используются в различных комбинациях или самостоятельно для создания гибридных интеллектуальных систем[10].

1.3. Обзор существующих классификаций нейронных сетей

Область искусственного интеллекта развивается уже несколько десятилетий, и за это время было создано множество подходов к построению и обучения нейронных сетей. Была разработана широкая классификация: ИНС различают по характеру обучения, связей, типам входных данных, настройке весов и другим характеристикам.

Классификация нейронных сетей по характеру обучения делит их на:

- нейронные сети, использующие обучение с учителем;
- нейронные сети, использующие обучение без учителя;
- нейронные сети, использующие обучение с подкреплением.

Обучение с учителем предполагает, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе они называются обучающей парой. Обычно сеть обучается на некотором числе таких обучающих пар. Предъявляется выходной вектор, вычисляется выход сети и сравнивается с соответствующим целевым вектором. Далее веса изменяются в соответствии с алгоритмом, стремящимся минимизировать ошибку. Векторы обучающего множества предъявляются последовательно, вычисляются ошибки и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемого уровня[11].

Обучение без учителя является намного более правдоподобной моделью обучения с точки зрения биологических корней искусственных нейронных сетей. Она не нуждается в целевом векторе для и, следовательно, не требует сравнения с предопределенными идеальными ответами. Обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса сети так, чтобы получались согласованные выходные векторы, т. е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. Процесс обучения, следовательно, выделяет статистические свойства обучающего множества и группирует сходные векторы в классы[11].

В обучении с подкреплением нет учителя, обеспечивающего целевые сигналы для сети, вместо этого иногда используется функция годности или функция оценки, по которой проводится оценка качества работы сети, при этом значения на выходе оказывает влияние на поведение сети на входе. В частности, если сеть реализует игру, на выходе измеряется количество пунктов выигрыша или оценки позиции. Каждая цепочка вычисляет ошибку как суммарную девиацию по выходным сигналам сети. Если имеется набор образцов обучения, ошибка вычисляется с учётом ошибок каждого отдельного образца[11].

Классификация нейронных сетей характеру связей делит их на:

1) Многослойные — ИНС, состоящая из входного, выходного и расположенного(ых) между ними одного (нескольких) скрытых слоёв нейронов[12].
2) Рекуррентные — ИНС, в которых выход нейрона может вновь подаваться не его вход. В более общем случае это означает возможность распространения сигналов от выходов к входам[12].
3) Сеть радиально-базисных функций — ИНС, использующая радиально-базисные функции как функции активации[13].
4) Самоорганизующиеся карты — ИНС с обучением без учителя, выполняющая задачу визуализации и кластеризации. Является методом проецирования многомерного массива в пространство меньшей размерности (зачастую двумерное)[14].

Прочие

Классификация нейронных сетей по типу настройки весов делит их на:

• сети с фиксированными связями – весовые коэффициенты нейронной сети выбираются сразу, исходя из условий задачи;

• сети с динамическими связями – для них в процессе обучения происходит настройка синаптических весов[11].

Классификация нейронных сетей по типу входных данных делит их на:

• аналоговые – входная информация представлена в форме действительных чисел;

• двоичные – вся входная информация в таких сетях представляется в виде нулей и единиц[11].

В ходе работы над теоретической частью были описаны методы проекта и определена область их применения. Для уточнения основного содержания теоритической части были описаны основные понятия по теме проекта, а также с целью выяснения свойств создаваемой нейронной сети была описана классификация ИНС.

Глава 2. Практическая часть

2.1. Разработка нейронной сети прямого распространения

В работе над проектом будем разрабатывать нейронную сеть прямого распространения, так как данная архитектура является относительно простой в исполнении и способна удовлетворить требованиям к желаемому результату.

«Прямое распространение» подразумевает собой то, что нейроны образуют собой отдельные слои. Внутри одного слоя нейроны никак не взаимодействуют, однако они обмениваются информацией с каждым нейроном из соседнего слоя. В этой структуре сигнал, подающийся на вход, проходит через каждый нейрон, претерпевает изменения, и возвращается в виде результата работы ИНС.

Так как никакие современные компьютеры не способны в точности смоделировать поведение человеческого мозга, то работа нейронной сети заключается в простом преобразовании данных. Вместо электрических сигналов (хотя они используются в работе компьютера, их поведение не задается программно) используются вещественные числа в десятичной системе счисления, а также 1 и 0 в двоичной. Вместо синапсов используются множители, называемые весами, на которые умножается значение сигнала при переходе из одного слоя ИНС в другой. Обобщив всё это, можем сказать, что работа ИНС сводиться к математической формуле.

Во-первых, каждый искусственный нейрон хранит некоторое значение x (в нашем случае — вещественное число). Во-вторых, перед отправлением сигнала в нейрон последующего слоя x умножается на значение веса — w. Также во многих реализациях встречается такой параметр как смещение — значение, которое прибавляется к значению нейрона после передачи в него сигналов из всех нейронов предыдущих слоев (как если бы в слое присутствовал нейрон, который имеет $w = 1$ и хранимое значение которого для разных нейронов другого слоя будет разным), обозначим как b. Так как значения нейронов нормализуются в пределах $[0; 1)$, чтобы лучше отразить их степень активности, используются специальные функции активации, чьё множество значений лежит в этом

промежутке. В нашей реализации будем использовать функцию tanh (гиперболический тангенс):

$\tanh(x) = (e^{2x} — 1) / (e^{2x} + 1)$

Тогда конечная формула для вычисления значения произвольного нейрона будет выглядеть так:

$x' = \tanh(b + \sum_{i=1}^{n} x_i w_i)$, где x' – значение произвольного нейрона из не входного слоя, n – количество нейронной в слое, идущем перед слоем, в котором расположен нейрон со значением x'.

Для реализации структуры нейронной сети используют вектора (динамические массивы данных) или матрицы. В нашем исполнении будут использоваться вектора, так как они включены в стандартные средства языка программирования C++.

Сама нейронная сеть представляет собой класс, в котором хранятся данные о топологии (количестве слоев и нейронов в них), значения отдельных нейронов, весов и смещений. Пользователю доступны функции для инициализации объекта нейронной сети, создания случайных входных данных, случайных весов и смещений, проведения прямого распространения и получения данных для отладки.

2.2. Разработка генетического алгоритма

Так как игра, в которой действует наш алгоритм, представляет собой изменяющуюся среду и способна иметь огромное количество возможных состояний, то создание обучающей выборки становится сложной или практически невозможной задачей (в зависимости от выбранной среды). Поэтому использование генетического алгоритма будет хороши методом корректировки синапсов созданной нейронной сети при отсутствии данных для обучения.

Принцип работы генетического алгоритма схож с принципом действия естественного отбора в природе: создаётся поколение агентов со случайными значениями, после выполнения ими определённых действий вычисляется их приспособленность к условиям среды, на основе результатов этих вычислений

создаётся новое поколение агентов, значения которых формируются путём смешивания значений наиболее приспособленных (далее кроссинговер). На некотором этапе работы подобного алгоритма все особи будут представлять собой агентов с одинаковыми значениями, поэтому к созданным путём смешивания агентам применяется какая-либо мутация — случайное изменение одного или нескольких значений.

Также, хорошей практикой является использование в процессе кроссинговера не только наиболее приспособленных особей, но подбор случайной особи при приоритете наиболее приспособленных. Этот процесс получил название «отбор по правилу рулетки». Действительно, наблюдается существенное сходство: чем больше размер слота рулетки, тем выше вероятность попасть в него. В случае генетического алгоритма размер «слота» определяется приспособленностью агента.

Для реализации генетического алгоритма возьмём достаточно простого агента и столь же простые условия отбора. Агент будет представлять с собой две строки, содержащие десять и пять вещественных чисел в промежутке [0;1] соответственно. Формула приспособленности будет следующая:

$f = (\sum_{a=1}^{10} x_a) + (1 - \sum_{b=1}^{5} x_b)$, где f – значение приспособленности, x – одно из хранимых агентов значений (x_a – в первой строке, x_b – во второй).

Для нас очевидно, что максимальное значение приспособленности будет равно пятнадцати (все числа в первой строке — единицы, во второй строке — нули). Однако, чтобы достичь значения тринадцать, алгоритм в среднем обрабатывает три тысячи поколений по 400 особей, что примерно равно четырем минутам ожидания. Хотя числа кажутся неоправданно большими, для программы, не имеющей чёткого алгоритма для достижения результата, это весьма поразительно.

Итак, удостоверившись в том, что алгоритм работает, включим в его состав ранее созданную ИНС. Теперь агентом будет выступать объект нейронной сети, а изменяемыми значениями — веса и смещения. Среда будет представлять собой двумерное пространство, бесконечное по абсциссе и ограниченное по ординате.

Пространство заполнено «стенами с отверстиями» - статичными структурами из двух прямоугольников, чья длина и ширина ограничены и меньше значения ограничения оси ординат. Агент — тоже прямоугольник, имеющий фиксированную скорость по оси абсцисс (больше нуля) и регулирующий свою скорость по оси ординат. Чем больше агент существует, не врезаясь в элементы «стен с отверстиями», тем более он считается приспособленным.

2.3. Разработка интерфейса для взаимодействия пользователя с агентами и средой

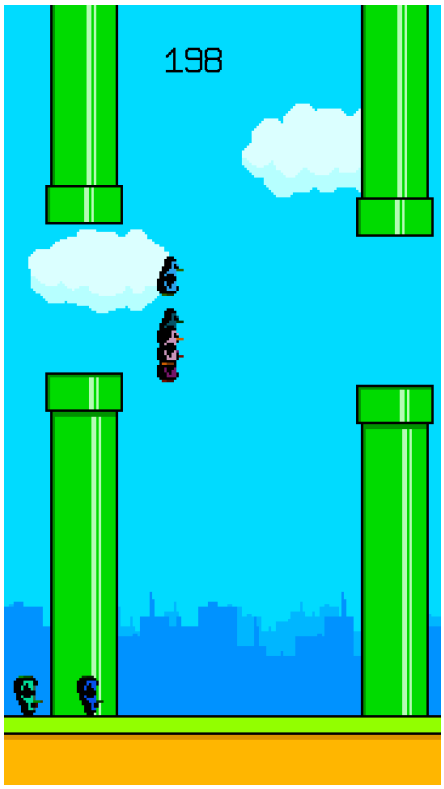
На данный момент имеется некоторая среда, в которой происходит обучение, но которая не имеет никакого графического представления. Это затрудняет возможность пользователя отслеживать процесс обучения, выявлять его тенденции и устранять программные ошибки.

Для реализации графического интерфейса была использована библиотека SFML 2.5.1. Она предоставляет возможности по созданию графических представлений разной сложности на языке программирования C++, и, согласно названию («Simple and Fast Multimedia Library» - англ. «Простая и Быстрая Мультимедийная Библиотека»), проста в освоении и реализует производительные решения.

В SFML несущим элементом графического интерфейса является некоторое «окно» - ограниченная область устройства визуальной связи, в которой происходит отрисовка всех элементов интерфейса. Также «окно» в SFML выполняет задачи по отслеживанию вводимых пользователем данных. Создаваемое «окно» будет состоять из двух частей: визуального представления среды и агентов, а также панели для изменения параметров среды и получения информации о ходе обучения нейронной сети.

Для создания первой части нам необходимо разработать дизайн визуального представления отдельных её элементов. Мы могли бы использовать разноцветные прямоугольники для того, чтобы различать, к примеру, границы пространства и «стены с отверстиями», однако это не самый наглядный способ. Поэтому, элементы визуального представления будут выполнены с помощью

спрайтов — совокупности прямоугольника и приложенного к нему изображения. Выводя на экран элементы через цикл отрисовок «окна» и регулируя их положения, получаем подобный результат:

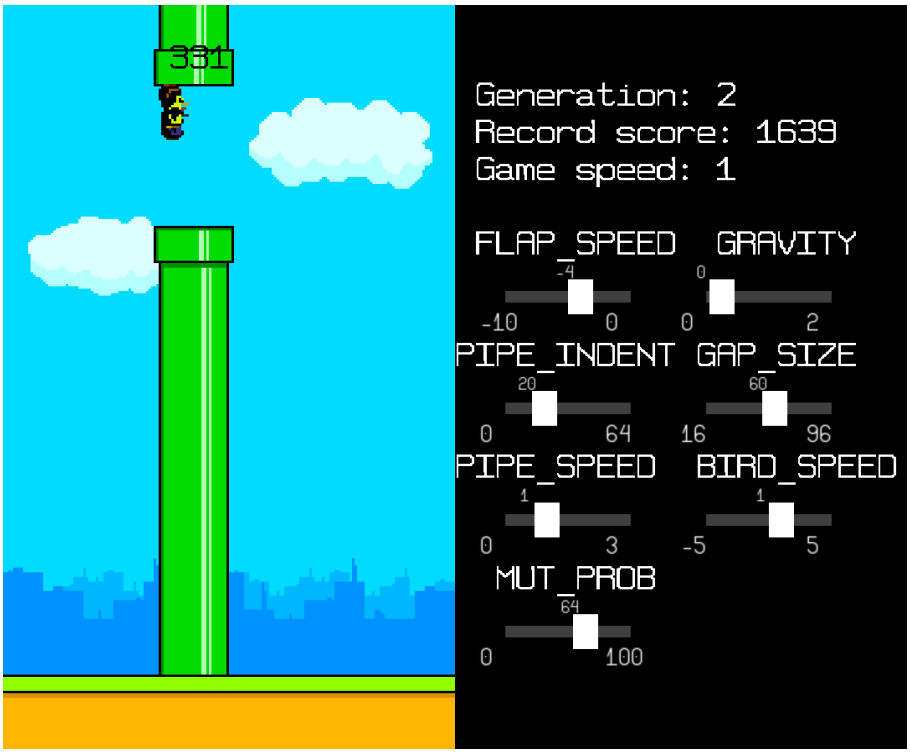


Для создания второй части требуется загрузить или создать шрифт, так как SFML не обладает предустановленным набором шрифтов, но поддерживает их использование. Наш шрифт представляет собой изображение - непрерывную полосу различных символов. Во время работы программа разделяет эту полосу на более мелкие изображения, содержащие отдельные символы. По запросу на вывод текста программа «достаёт» из памяти нужные символы и формирует некоторый текст.

Чтобы добавить возможность изменения среды без повторной сборки программы, а также доступную человеку, не разбирающемуся в языке программирования C++, были использованы ползунки. Всего присутствуют семь ползунков, регулирующие скорость агента по оси ординат (FLAP_SPEED), воздействие среды на агента по оси ординат (GRAVITY), размер «отверстий» (GAP_SIZE), их максимальное отклонение (PIPE_INDENT), скорость отклонения (PIPE_SPEED), скорость агента по оси абсцисс (BIRD_SPEED) и

шанс мутации (MUT_PROB).

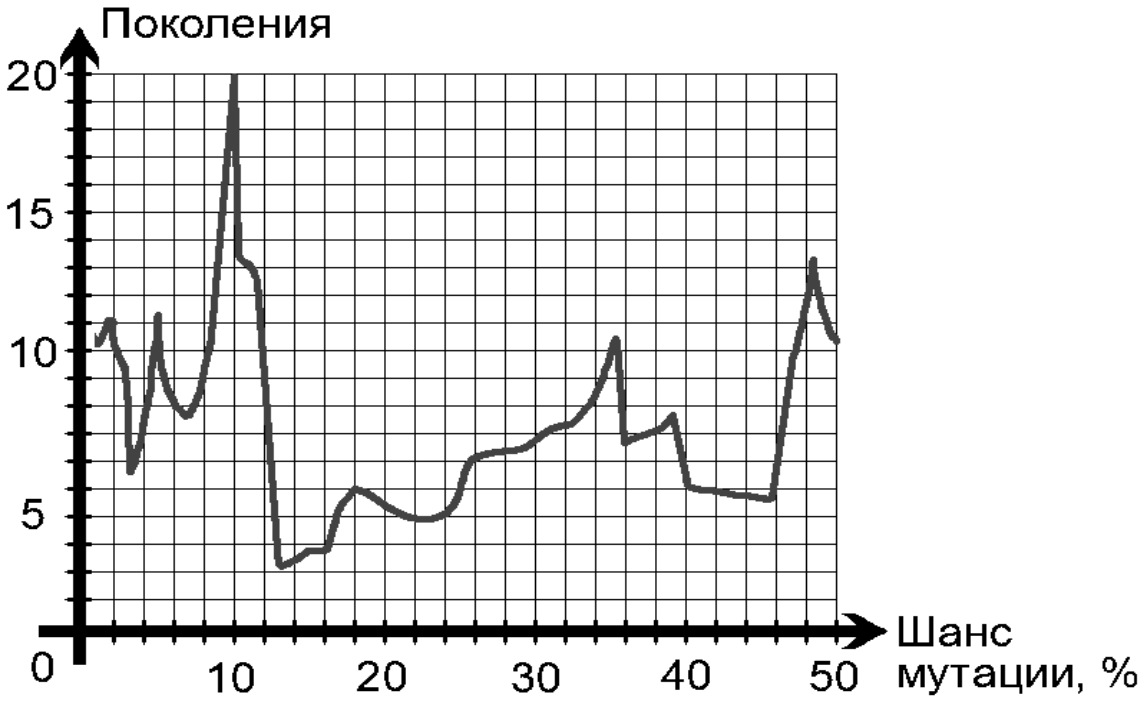
Объединённые вместе части интерфейса выглядят следующим образом:



2.4. Обучение нейронной сети и его результаты

После создания всех модулей приложения, мы наконец способны оценить, насколько успешно нейронная сеть будет обучаться и обучается ли вообще.

Ниже приведён график зависимости среднего количества поколений, необходимых нейронной сети для формирования агента, способного существовать внутри среды не менее 65535 обновлений экрана (при стандартной скорости это время приблизительно равно 11 минутам, однако поддерживается ускорение до 64 раз, что сводит максимальное время на обработку одного поколения до 10,6 секунд):



Из графика видно, что обучение нейронной сети действительно происходит, так как в большинстве случаев количество поколений до окончания сеанса работы было равно конечному числу. Более того, наблюдаются заметные изменения в работе агента при разных значениях шанса мутации.

Так, наиболее стабильным был диапазон 15-25%. Результат при значении от 25 до 50% несколько хуже, но также приемлем. А при значении шанса мутации менее 15% наблюдается резкое увеличение средней длительности сеанса.

Заключение

В ходе работы над проектом, используя научные методы, был разработан и протестирован итоговый продукт — реализация нейронной сети, обучающейся посредством генетического алгоритма, а также описан процесс разработки и возможности созданного продукта.

Для подтверждения достижения предоставленного в проектной работе результата была проведена экспертная оценка в лице учителя информатики МБОУ СОШ №103 Моисеевой Оксаны Викторовны. Работа получила положительную оценку по всем критериям (качество разработанного продукта, его соответствие теме и поставленным требованиям, действительность предоставленных в проектной работе данных).

Ответ на проблемный вопрос получен: нейронная сеть оказалась способна к обучению игре. Таким образом, также подтверждена гипотеза.

Была достигнута цель проекта: разработана и обучена нейронная сеть. Поставленные задачи решены.

Перспективы развития данной темы видим в создании более «дружелюбного» к пользователю интерфейса, добавлении поддержки большего количества топологий нейронной сети, методов мутации, селекции и кроссинговера, а также в проведении более углублённого исследования зависимости результатов обучения нейронной от различных параметров сре

Список литературы

1. Ворожцев А. В., Винокуров Н. А. Алгоритмы: построение, анализ и реализация на языке программирования Си // язык Си, алгоритмы, методы построения и анализа алгоритмов (Московский физико-технический институт, 2007) — М., 2007 — 452 с.

2. ГОСТ 19781-90. ЕСПД. Термины и определения.

3. И. Соммервилл. Инженерия программного обеспечения. - М.: Вильямс, 2002

4. Фрэнк Розенблатт. Принципы нейродинамики. Перцептроны и теория механизмов мозга. - М.: Мир, 1965

5. Научно редакционный совет: председатель — Осипов Ю. С. и др. Большая Российская энциклопедия — М.: Большая Российская энциклопедия, 2004. - 26787 с.

6. Горбань А. Н. Нейронные сети на персональном компьютере. - Новосибирск: Наука, 1996

7. Купцов Л. П. Математическая энциклопедия (в 5 томах). - М.: Советская Энциклопедия, 1977. - 1152 с.

8. Оптимизация в ML / Тяпкин Д. https://academy.yandex.ru/handbook/ml/article/optimizaciya-v-ml

9. Галушкин А. И. Синтез многослойных систем распознавания образов. - М.: Энергия, 1974

10. Генетические алгоритмы / Панченко Т. В. // Тарасевич Ю. Ю. - Астрахань: Издательский дом «Астраханский университет», 2007. — 87 [3] с.

11. Искусственные нейронные сети и их приложения / Ф. М. Гафанов, А. Ф. Галимянов // Миннегалиева Ч.Б. - Казань: Издательство Казанкского университета, 2018. - 121 с.

12. Основы ИНС / Радько П. https://neural.radkopeter.ru/chapter/основы-инс/#Сети-с-обратными-связями

13. Брумхед Д. С., Дэвид Лоу. Multivariable functional interpolation and adaptive networks // Complex systems. - 1988. - 2. - 321-325 с.

Туево Кохонен. Self-Organaizing Maps. - Берлин: Springer Berlin Heidelberg

«13» апреля 2023 года

Бланк экспертной оценки нейронной сети, разработанной в ходе реализации проектной работы «Создание нейронной сети и её обучение игре посредством генетического алгоритма», учеником 10Г класса МАОУ СОШ №103 Ташу Мадином

ФИО эксперта, должность: Моисеева Оксана Викторовна, учитель информатики МАОУ СОШ №103.

№ п/п	Критерий оценки	Соответствует	Условно соответствует	Не соответствует	Примечание
1	Соответствие созданного продукта теме проекта.	+			
2	Использование исключительно приведённых в тексте проектной работы средств.	+			
3	Соответствие результата работы созданного продукта поставленным требованиям.	+			
4	Достоверность приведённых в ходе проектной работы числовых критериев оценки эффективности работы созданного продукта при различных настройках.	+			

Моис (Моисеева О.В)

рис.1 — заполненный экспертный лист.