

Министерство науки и высшего образования Российской Федерации  
Санкт-Петербургский политехнический университет Петра Великого  
Физико-механический институт

Работа допущена к защите

Руководитель ОП

К. Н. Козлов

«\_\_\_»\_\_\_\_\_ 2022 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА**  
**ПОСТРОЕНИЕ ЛАНДШАФТНЫХ КАРТ НА ОСНОВЕ**  
**ГЕНЕРАТИВНЫХ НЕЙРОННЫХ СЕТЕЙ**

по направлению подготовки 01.03.02 Прикладная математика и информатика  
Направленность (профиль) 01.03.02\_02 Системное программирование

Выполнил

студент гр. 5030102/80201

Игнатьев Д. Д.

Руководитель

Доцент кафедры «Прикладная математика»

кандидат физико-математических наук

Беляев С. Ю.

Консультант

Савчук Д. А.

Санкт-Петербург – 2022

**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ПЕТРА ВЕЛИКОГО**

**ФИЗИКО-МЕХАНИЧЕСКИЙ ИНСТИТУТ**

УТВЕРЖДАЮ

Руководитель ОП

\_\_\_\_\_ К. Н. Козлов

“\_\_” \_\_\_\_\_ 20\_\_ г.

**ЗАДАНИЕ**

**На выполнение выпускной квалификационной работы**

студенту Игнатьеву Даниилу Дмитриевичу 5030102/80201

1. Тема работы: Построение ландшафтных карт на основе генеративных нейронных сетей.
2. Срок сдачи студентом законченной работы: 01.06.2022
3. Исходные данные по работе: Карта высот поверхности Земли. Карта была разделена на квадратные изображения с разным разрешением от 64x64 до 256x256 пикселей с перекрытием и без. Таким образом было образовано 4 датасета для исследования генеративных нейросетей, размер датасетов от 3000 до 12000 изображений.
4. Содержание работы:
  - Разработка подхода генерации набора данных для обучения.
  - Сравнительный анализ работы двух архитектур генеративной нейросети по результатам обучения на исходном датасете.
  - Оценка и визуализация результатов.
5. Консультанты по работе: Даниил Александрович Савчук
6. Дата выдачи задания: 04.11.2022

Руководитель ВКР \_\_\_\_\_



С. Ю. Беляев

Задание принял к исполнению: 04.11.2022

Студент \_\_\_\_\_



Д. Д. Игнатьев

## РЕФЕРАТ

На 108 с., 20 рисунков, 3 таблицы, 5 приложений

КЛЮЧЕВЫЕ СЛОВА: HEIGHT MAP, TERRAIN GENERATION, CNN, GAN, КАРТА ВЫСОТ, КОНКУРИРУЮЩИЕ НЕЙРОННЫЕ СЕТИ, ГЕНЕРАТОР, ДИСКРИМИНАТОР, СИНТЕЗ ИЗОБРАЖЕНИЙ, ГЕНЕРАЦИЯ ЛАНДШАФТА.

Тема выпускной квалификационной работы: «Построение ландшафтных карт на основе генеративных нейронных сетей».

Данная работа посвящена исследованию применимости нейронных сетей в сфере генерации ландшафтов, подобных земным.

Область применения – синтез карт высот.

Методы исследования – анализ, эксперимент, тестирование, сравнение.

Задачи, которые решались в ходе исследования:

1. Поиск исходных данных и генерация обучающего набора.
2. Разработка двух различных архитектур нейронных сетей.
3. Обучение реализованных алгоритмов на различных наборах данных.
4. Оценка и визуализация результатов обучения нейронных сетей.

Работа основывалась на статьях из сети интернет.

Программирование и анализ проводился с помощью интерактивной облачной среды Google Colab. Задание и обучение нейросетей осуществлено на языке Python с использованием открытой библиотеки для машинного обучения Keras.

Программа, подготавливающая обучающий набор данных реализована на языке программирования C++ с использованием открытой библиотеки для работы с изображениями stb\_image.

Визуализация результатов выполнена в межплатформенной среде разработки компьютерных игр Unity.

В результате исследование были запрограммированы и обучены две нейронные сети.

**\*Дописать\***

*Реферат выполняется на русском и английском языках (Приложение П1.3 к Положению).  
Реферат должен содержать: - предмет, тему, цель ВКР; - метод или методологию проведения ВКР; 28 - результаты ВКР; - область применения результатов ВКР; - выводы*

## ABSTRACT

# СОДЕРЖАНИЕ

## Оглавление

1. Введение.....	6
2. Основная часть .....	6
2.1 Искусственные нейронные сети: основные понятия, классификация.....	6
2.1.1 Функция ошибки .....	7
2.1.2 Функция активации.....	8
3. Заключение .....	10
4. Список использованных источников .....	11
5. Приложения .....	11

# 1. Введение

На сегодняшний день потребность в создании реалистичных ландшафтов в кинематографической и игровой промышленности становится все острее. Модели рельефа становятся больше, требования к их качеству выше. Однако наиболее популярными в использовании методы процедурной генерации являются строго настроенными и мало расширяемыми. Следовательно, алгоритмы нуждаются в частой доработке, а сами продукты их работы, по мере увеличения размеров, требуют больше вмешательства левел-дизайнеров. Таким образом, повышение качества, разнообразия и размера модели ландшафта влечет к большим затратам.

Решением данной проблемы должен стать новый инструмент, способный к расширению и слабо привязанный к типу ландшафта. Целью данной бакалаврской работы является исследование применимости нейронных сетей для генерации ландшафтных карт.

Искусственные нейронные сети впервые появились в середине двадцатого века. Их идея берет свое начало из нейроанатомии и заключается в подражании нейронным сетям мозга.

## 2. Основная часть

### 2.1 Искусственные нейронные сети: основные понятия, классификация

Под искусственными нейронными сетями (в дальнейшем просто «нейронная сеть») подразумеваются вычислительные структуры, которые моделируют простые биологические процессы, обычно ассоциируемые с процессами человеческого мозга. Они представляют собой распределенные и параллельные системы, способные к адаптивному обучению путем анализа положительных и отрицательных воздействий (Круглов В.В, 2002).

Нейронные сети являются одним из множества типов алгоритмов машинного обучения. Однако, за последние годы именно они приобрели **большую** популярность.

Основной структурной единицей нейронной сети является искусственный нейрон. Множество таких нейронов взаимодействует между собой. На рисунке ниже представлена схема искусственного нейрона:

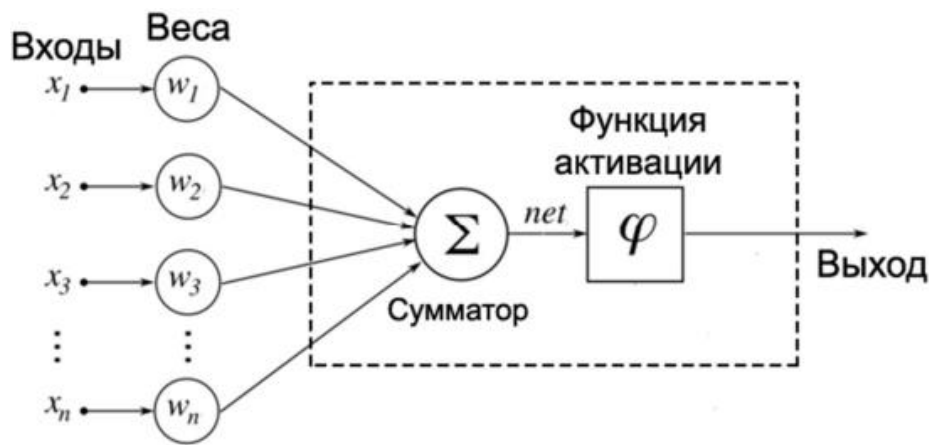


Рис. 1 Схема искусственного нейрона (Нейронные сети)

Как видно из схемы, у нейрона может быть несколько входных связей, через которые подаются различные сигналы. Он преобразует их посредством активационной функции и передает информацию другим нейронам.

Другими словами, искусственный нейрон — это функция  $R^n \rightarrow R$ , которая преобразует несколько входных параметров в один выходной. Своей эффективностью нейронные сети обязаны двум составляющим:

- 1) возможности параллельной обработки больших объемов информации.
- 2) способности обучаться, т. е. выделять признаки или создавать обобщение, или, иными словами, способности получать обоснованный результат для данных, не присутствовавших в процессе обучения (Хайкин, 2016)

Задача нейросети – найти достаточно хорошую функцию, аппроксимирующую обучающую выборку, при условии, чтобы эта функция обобщалась и на все остальные еще не встреченные входные данные. Отсюда следуют два ключевых понятия: функция ошибки и регуляризация.

### 2.1.1 Функция ошибки

Задача функции ошибки предельно проста. Оптимизация функции ошибки, а именно, минимизация её значения, влечет улучшение точности ответов нейросети. В качестве алгоритма оптимизации используется метод градиентного спуска (и его многочисленные варианты улучшения), суть которого заключается в поэтапном изменении значений параметров функции на значение, пропорциональное градиенту в точке.

Роль функции ошибки часто выполняет средняя квадратичная ошибка, в которой минимизируется среднее квадратов отклонений предсказанных значений от истинных:

$$RSS(\omega) = \frac{1}{N} \sum_{i=1}^N (y_i - x_i w)^2$$

Подобная функция ошибки используется в задачах регрессии. Так же в задачах регрессии могут использоваться различные её модификации, такие как средняя квадратичная логарифмическая ошибка

$MSLE(\omega) = \frac{1}{N} \sum_{i=1}^N (\log(1 + y_i) - \log(x_i w + 1))^2$ , которую можно использовать в случае, когда целевое значение обучающей выборки имеет большой разброс, и при обучении подобные случаи не требуют наложения слишком высоких штрафов. В задачах же классификации используется функция кросс-энтропии, которая бывает как бинарной, в случае выбора между двумя возможными классами, так и мульти-классовой:

$$J = -\frac{1}{N} \sum_{i=1}^N (L_i \log(S_i))$$

, где  $L_i$  - элемент вектора значения в one-hot кодировке, в которой длина вектора равна количеству классов; для правильного класса проставлена 1, для всех остальных - 0.  $S_i$  - результат работы алгоритма, а именно, вероятность, с которой алгоритм относит результат обработки входящих данных к  $i$ -ому классу. Вектор  $S_i$  в большинстве случаев - это результат работы функции softmax, которая используется в паре с кросс-энтропией в качестве функции ошибки.

Бинарная классификация является частным случаем кросс-энтропии и имеет вид:

$$J = -\frac{1}{N} \sum_{i=1}^N (y_i \log(y_i^*) + (1 - y_i) \log(1 - y_i^*)).$$

Альтернативой функции кросс-энтропии в случае бинарной классификации может быть так называемая hinge loss:  $\max(0, 1 - y_i)$ . Для ее использования целевые значения должны принадлежать множеству  $\{-1, 1\}$ . Однако, использование этой функции не всегда может привести к хорошему результату относительно кросс-энтропии.

### 2.1.2 Функция активации

Еще одним важным понятием нейронных сетей является функция активации - нелинейная функция, которая применяется нейроном для получения выходного значения. Существуют несколько функций, которые на данный момент используются в большей части решений на основе нейронных сетей:

Binary step:  $f(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases}$

Эта функция является пороговой. Она хорошо работает для бинарной классификации, но слабо применима к нейросетям с большим числом нейронов или небинарной классификацией.

linear:  $f(x) = x$

sigmoid:  $f(x) = \frac{1}{1 + e^{-x}}$

softmax:  $f(x)_i = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}}$  - аналог сигмоиды для задачи небинарной классификации

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

relu:  $f(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases}$



leakyRelu:  $f(x) = \begin{cases} ax, & x \leq 0 \\ x, & x > 0 \end{cases}$ , где  $a$  – настраиваемый коэффициент. Чаще всего  $a = 0.01$ .  
Графики этих функций показаны на Рис. 2

### 2.1.2 Обратное распространение ошибки

### 2.1.3 Генеративно-сопоставительная сеть

Одной из значимых проблем алгоритмов машинного обучения с учителем является нехватка данных. Для построения рельефа, похожего на земной, нужно обучать нейросеть на существующих данных поверхности земли, но их количество существенно ограничено. Поэтому за основу была выбрана генеративно-сопоставительная сеть (GAN), способная генерировать новые данные для обучения.

Основной их идеей GAN является наличие двух нейросетей, работающих в паре: генератор и дискриминатор. Дискриминатор на вход получает данные из двух источников: заранее подготовленной выборки с размеченными данными, а также вывод генератора. Основной задачей дискриминатора является определение того, является ли входящее сообщение реальным, либо оно поступило из генератора. Т.е., выходом дискриминатора будет являться булево значение. Соответственно, задача оптимизации дискриминатора – это задача минимизации его ошибки. Генератор же в свою очередь из случайного вектора генерирует данные, которые должны быть неправильно классифицированы дискриминатором. Данная схема изображена на рисунке.



Чтобы определить распределение данных, выдаваемых генератором  $p_G(x)$ , сперва вводится вектор шумов  $p_Z(z)$ , затем вводится дифференцируемая функция отображения  $G(z, \theta_G)$ , представляющая собой многоуровневый перцептрон с параметрами  $\theta_G$ . Так же определяется второй перцептрон  $D(z, \theta_D)$ , областью значений которого является скаляр.  $D(x)$  представляет собой вероятность того, что  $x$  принадлежит реальным данным, а не порожденным генератором  $p_G$ . Таким образом, выведем функцию ошибки:

$$\min_G \max_D V(D, G) = E[\log D(x)] + E[\log(1 - D(G(z)))]$$

Где  $E[\log D(x)]$  - кросс-энтропия ответов дискриминатора в случае реальных данных, а  $E[\log(1 - D(G(z)))]$  - в случае сгенерированных.

Другими словами, задача дискриминатора – это максимизировать данную функцию ошибки, генератора – минимизировать. Можно заметить, что в случае минимизации выражения по функции генератора, первое слагаемое никак от генератора не зависит; таким образом оно может быть опущено. Однако, как показывает практика, минимизации  $E[\log(1 - D(G(z)))]$  недостаточно для успешного обучения генератора в виду того, что на ранних этапах обучения, когда дискриминатор с довольно большой уверенностью может отличить изображения от произвольного шума,  $\log(1 - D(G(z)))$  будет насыщаться. Решением будет обучение генератора путем максимизации  $\log(D(G(z)))$  вместо минимизации  $\log(1 - D(G(z)))$ . Данная функция позволяет получить гораздо больший градиент на начальных этапах обучения.

Суммируя все вышесказанное, итоговый алгоритм обучения генеративно-сопоставительной сети по минибатчам будем иметь следующий вид:

- На каждой итерации выбирается  $k$  элементов выборки  $(x_1, x_2, \dots, x_k)$ , а также генерируется  $k$  произвольных векторов определенной длины  $(z_1, z_2, \dots, z_k)$ , которые будут использованы в качестве входных параметров сети генератора.
- Для всех  $x_i$ , а так же  $z_i$  вычисляются  $D(x_i)$ , а так же  $D(G(z_i))$
- Обновляется  $D$ , учитывая полученное значение градиента:

$$\nabla \frac{1}{k} \sum_{i=1}^k (\log D(x_i) + \log(1 - D(G(z_i))))$$

- Данные операции, обновляющие дискриминатор, в большинстве случаев производятся несколько раз, прежде чем обновить генератор.
- Обновляем генератор по вновь сгенерированным  $k$  векторам шумов:

$$\nabla \frac{1}{k} \sum_{i=1}^k \log(1 - D(G(z_i)))$$

Несмотря на относительную простоту алгоритма, обучение генеративных сетей – крайне сложный и нестабильный процесс. Задача нахождения точки равновесия в данной минимакс игре усложняется с усложнением конкретного случая применения GAN. Ян Гудфеллоу писал, что ему на самом деле крайне повезло, что первые запуски генеративной сети, написанной им, принесли довольно осмысленные результаты. Потому что, учитывая всю нестабильность обучения, выбор иных гиперпараметров или структуры сети привел бы к невозможности получить достаточные для продолжения исследований результаты. Таким образом, успешный синтез изображений в высоком разрешении по-прежнему остается неуловимой целью. В рамках данной работы была поставлена задача реализации генеративно-сопоставительных сетей, позволяющих синтезировать достоверные изображения в достаточно больших разрешениях: 64x64 и 128x128 пикселей. Будет проведен анализ ряда существующих успешных подходов и рекомендаций, на основе всего этого будет построен агрегированный алгоритм и проанализированы полученные результаты. Решение должно представлять собой разумный компромисс между качеством полученных изображений и скоростью обучения, однако первостепенной целью оптимизации будет являться качество изображений, получаемых на выходе работы сети.

### 3. Заключение

#### **4. Список использованных источников**

#### **5. Приложения**