Научная статья
Original article
УДК 004.89

ВЫБОР ОПТИМИЗАТОРА ДЛЯ ОБУЧЕНИЯ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ: ЗАДАЧА КЛАССИФИКАЦИИ ЛЕСНЫХ ПОЖАРОВ

CHOICE OPTIMIZER FOR TRAINING A CONVOLUTIONAL NEURAL NETWORK: FOREST FIRES CLASSIFICATION PROBLEM

Журавлева Наталья Александровна, студентка 2 курса магистратуры по направлению подготовки «Информатика и вычислительная техника» (Искусственный интеллект), ФГБОУ ВО «Иркутский национальный исследовательский технический университет» (664074 Россия, г. Иркутск, ул. Лермонтова, д. 83)

Zhuravleva Natalya Alexandrovna, 2nd year Master's student in the field of study «Informatics and Computer Engineering» (Artificial Intelligence), Federal State Budget Educational Institution of Higher Education «Irkutsk National Research Technical University» (83 Lermontova st., Irkutsk, 664074 Russia)

Аннотация. Статья посвящена сравнению и применению оптимизаторов в обучении нейронных сетей в решении задачи классификации лесных пожаров. За основу для модели сверточной нейронной сети была взята модель «Dogs vs. Cats». Сверточная нейронная сеть была обучена на наборе данных, который был создан для изучения различных моделей сверточных нейронных сетей и улучшения точности классификации лесных пожаров. В статье используются основные

оптимизаторы в обучении нейронных сетей для классификации объектов на изображении, такие как SGD, RMSprop, Adam, Adadelta, Adamax. Представлены графики точности и ошибки обучения моделей, а также структура модели нейронной сети.

Abstract. The article is devoted to the comparison and application of optimizers in the field of neural networks to solve the problem of forest fire detection. The “Dogs vs Cats” model was taken as the basis for the convolutional neural network model. A Convolutional Neural Network was developed on a dataset that was created to learn different Convolutional Neural Network models and wildfire filtering analysis. The article uses the main optimizers in neural network equipment to protect objects in images, such as SGD, RMSprop, Adam, Adadelta, and Adamax. High score sentences and model errors, and neural network model structure.

Ключевые слова: глубокое обучение, оптимизаторы, классификация объектов, точность классификации объектов, сверточная нейронная сеть, лесные пожары.

Keywords: deep learning, optimizers, object classification, object pursuit, convolutional neural network, forest fires.

Введение

Леса являются одним из незаменимых ресурсов планеты для социального развития и выживания человечества, которые защищают экологию Земли. К сожалению, при некоторых природных и антропогенных факторов возникают лесные пожары, которые наносят существенный вред экосистеме планеты и экономике различных стран. Частота лесных пожаров в последние годы значительно возросла из-за человеческой деятельности, изменения климата и иных факторов. Поэтому мониторинг и предотвращение лесных пожаров стало глобальной проблемой для всех стран мира.

Основная цель распознавания лесных пожаров с помощью сверточных нейронных сетей заключается в снижении временных, трудовых и финансовых ресурсов при обнаружении, мониторинге, тушении лесных пожаров и ликвидации последствий лесных пожаров за счет более быстрой идентификации пожаров в лесах. Заранее обнаруженные пожары также поможет сохранить экосистему всей планеты.

В машинном обучении задача классификации – задача разделения множества наблюдений (объектов) на группы, называемые классами, на основе анализа их формального описания. При классификации каждая единица наблюдения относится определенной группе или номинальной категории на основе некоторого качественного свойства.

Классифицировать объект – значит, указать номер или наименование класса, к которому относится данный объект. Классификация объекта – номер или наименование класса, выдаваемый алгоритмом классификации в результате его применения к данному конкретному объекту.

Точность классификации лесных пожаров зависит от используемой модели сверточной нейронной сети, а также от количества и качества самих данных. В этом исследовании используется измененная архитектура сверточной нейронной сети «Dogs vs. Cats» [1], которая обучалась на уже готовом наборе данных «Forest Fire Images» [2].

Набор данных для классификации лесных пожаров

Для обучения и тестирования модели сверточной нейронной сети использовался уже готовый набор данных соревнования «Forest Fire Images» [2], проходящий на специальной платформе Kaggle. Kaggle был создан для проведения и организации соревнований по исследованию различных данных. Этот набор данных был создан путем объединения и слияния различных других небольших наборов данных.

Набор данных состоит из изображений лесных пожаров и изображений без лесных пожаров. Весь набор данных разделен на две папки с именами Test Data и Train Data. Тестовые данные содержат около 50 изображений с

лесными пожарами и без лесных пожаров. Данные для обучения содержат 5000 изображений с лесными пожарами и без лесных пожаров. Каждая из этих папок разделена на две подпапки с именами Fire и Non Fire, каждая из этих подпапок в Test Data содержит 25 изображений, а каждая из этих подпапок в Train Data содержит 2500 изображений. Изображения с лесным пожаром и без лесного пожара представлены на рисунке 1.

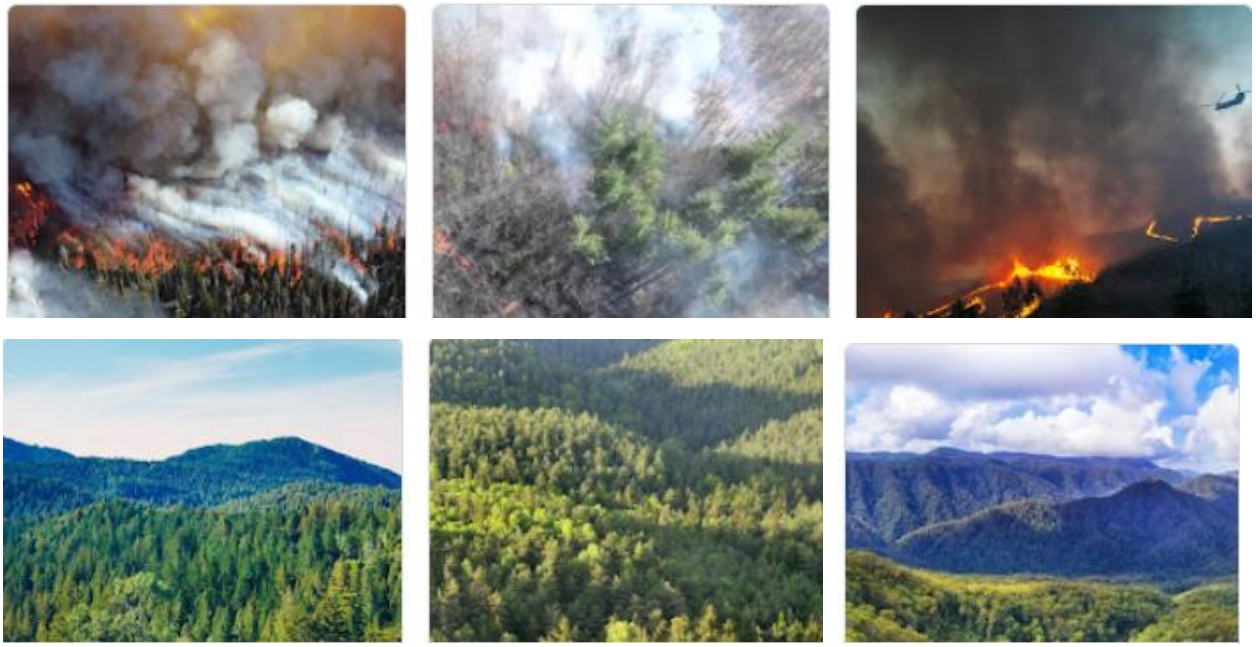


Рисунок 1 – Примеры изображений с лесным пожаром и без лесного пожара из набора данных

Модель сверточной нейронной сети

Лучшим подходом для обработки изображений считается применение сверточных нейронных сетей как разновидности моделей глубокого обучения [3,4]. Как отмечает разработчик Keras Франсуа Шолле, создание моделей глубокого обучения стало подобно конструктору LEGO, где важно эффективно собрать совместимые слои в конвейер для обработки данных. Библиотека Keras является достаточно простой в применении и на ее основе уже создано большое количество готовых моделей, которые позволяют использовать нейронные сети для решения практических задач.

Для решения задачи используется модель сверточной нейронной сети, представленной на платформе Kaggle в соревновании «Dogs vs. Cats» [2] как шаблона для решения задачи по классификации лесных пожаров.

Основным блоком в нейронной сети является слой, который принимает и обрабатывает данные. Здесь используется класс Sequential, чтобы создать многослойную модель нейронной сети последовательного типа.

Для создания слоев применяются следующие модули:

- Conv2D - для создания сверточных слоев [5];
- BatchNormalization - для создания слоя пакетной нормализации, как метода ускорения и повышения стабильности нейронных сетей [6];
- MaxPooling2D - для создания слоя подвыборки, снижающего пространственную размерность данных [7];
- Dropout - метод регуляризации, который позволяет случайным образом исключить некоторые нейроны из обучения с указанной вероятностью [8];
- Dense - для создания полносвязных слоев [9];
- Flatten - для создания переходного слоя от Conv2D к Dense [10].

В модели нейронной сети выходное значение нейрона определяется функцией активации. Здесь используется функция активации Relu [11] во всех слоях, кроме последнего. На последнем слое – функция активации – softmax [11], которая позволяет определить вероятность принадлежности конкретного изображения к одному из классов. Была построена модель сверточной нейронной сети, состоящую из трех блоков со сверточными слоями Conv2D и 1 блока с полносвязными слоями Dense. Первые три блока состоят из сверточного слоя Conv2D, слоев BatchNormalization, MaxPooling2D и Dropout. Количество фильтров для каждого сверточного слоя удваивается, соответственно, 8, 16 и 32. Задача первых трех блоков - извлечение признаков из изображений. Последний блок выполняет роль классификатора.

После последнего сверточного слоя выходное изображение преобразуется в одномерный вектор слоем Flatten для передачи данных в полносвязный слой Dense. Последний слой Dense содержит два нейрона, так

как решается задача классификации на два класса – лесной пожар и не лесной пожар.

Далее происходит компиляция модели, где настраивается процесс обучения. Для этого необходимо определить следующие параметры:

- функция потерь (loss) или целевая функция оценивает отклонение результатов обучения нейронной сети от ожидаемых результатов. Для задачи классификации, как правило, используется функция потерь 'categorical_crossentropy', которая вычисляет расстояние между распределением вероятности на выходе сети и заданным распределением меток [3];
- оптимизатор (optimizer) реализует основной алгоритм глубокого обучения – алгоритм обратного распространения ошибки – механизм, с помощью которого нейронная сеть обновляет так называемые веса – параметры обучения – на основе данных, получаемых от функции потерь и настройки скорости обучения [3].
- метрика (metrics) служит для мониторинга качества предсказаний на этапе обучения и тестирования. Здесь использовалась метрика 'accuracy', которая определяется как доля правильно классифицированных изображений (точность предсказания) [3].

На рисунке 2 предоставлена структура модели сверточной нейронной сети.

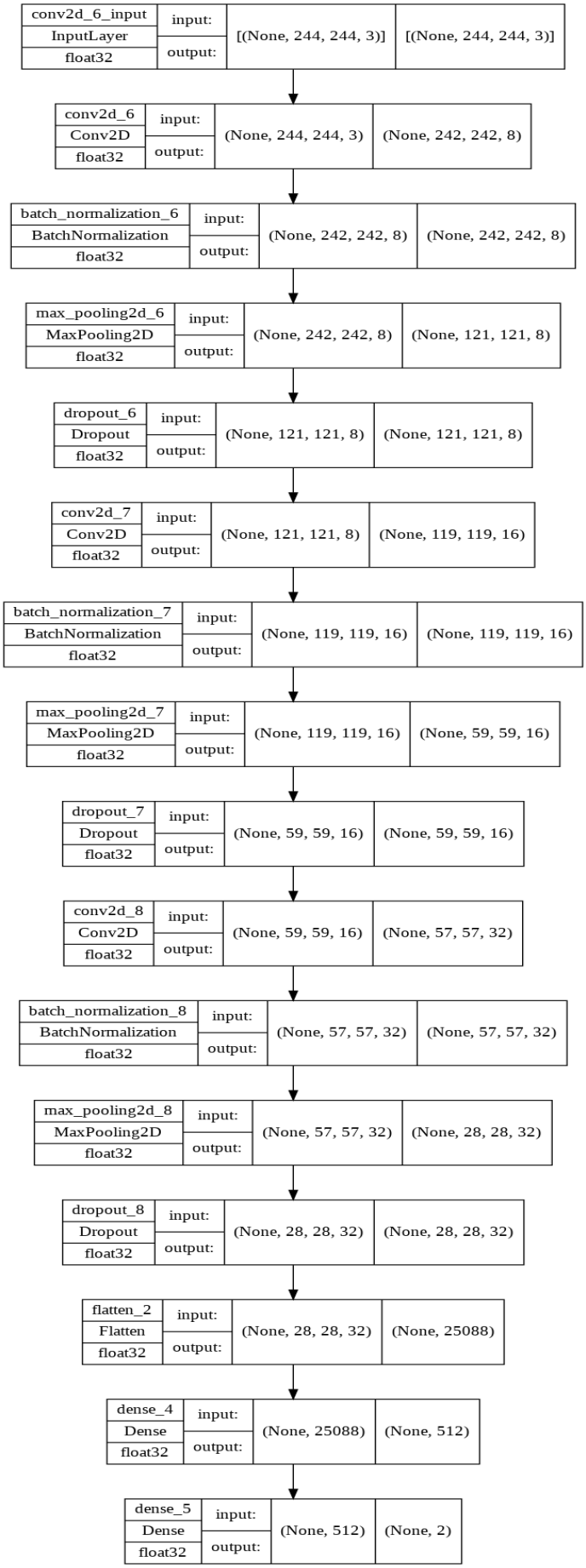


Рисунок 2 – Структура модели сверточной нейронной сети

Описание оптимизаторов

Оптимизатор – это метод достижения лучших результатов, помощь в ускорении обучения [12]. Другими словами, это алгоритм, используемый для незначительного изменения параметров, таких как веса и скорость обучения, чтобы модель работала правильно и быстро. Для обучения построенной модели сверточной нейронной сети были выбраны оптимизаторы: SGD, RMSprop, Adam, Adadelta, Adamax.

Стохастический градиентный спуск (SGD) – это мини-пакетный градиентный спуск стремится сбалансировать устойчивость стохастического градиентного спуска и эффективность пакетного градиентного спуска [12]. Это наиболее распространенная реализация градиентного спуска, используемая в области глубокого обучения. Когда мини-пакеты выбираются случайным образом, то при застревании в локальных минимумах некоторые шумные шаги могут привести к выходу из этих минимумов.

Среднеквадратичное распространение корня (RMSprop) – это экспоненциально затухающее среднее значение. Существенным свойством RMSprop является то, что данный оптимизатор не ограничивается только суммой прошлых градиентов, но RMSprop более ограничен градиентами последних временных шагов [12]. RMSprop вносит свой вклад в экспоненциально затухающее среднее значение прошлых «квадратичных градиентов». В RMSProp уменьшается вертикальное движение, используя среднее значение, потому что они суммируются приблизительно до 0, принимая среднее значение. RMSprop предоставляет среднее значение для обновления.

Адаптивная оценка момента (Adam) – один из самых эффективных алгоритмов оптимизации в обучении нейронных сетей. Adam сочетает в себе идеи RMSProp и оптимизатора импульса. Вместо того чтобы адаптировать скорость обучения параметров на основе среднего первого момента (среднего значения), как в RMSProp, Adam также использует среднее значение вторых моментов градиентов. В частности, алгоритм вычисляет

экспоненциальное скользящее среднее градиента и квадратичный градиент, а параметры beta1 и beta2 управляют скоростью затухания этих скользящих средних.

Adadelta – это метод стохастического градиентного спуска, основанный на адаптивной скорости обучения по измерению для устранения двух недостатков:

- постоянного снижения скорости обучения на протяжении всего обучения.
- необходимости в выбранной вручную глобальной скорости обучения.

Adadelta – это более надежное расширение Adagrad, которое адаптирует скорость обучения на основе движущегося окна обновлений градиента, а не накапливает все прошлые градиенты [12]. Таким образом, Adadelta продолжает обучение, даже когда было сделано много обновлений. По сравнению с Adagrad, в оригинальной версии Adadelta нет необходимости в установлении начальной скорости обучения нейронной сети.

AdaMax является адаптацией оптимизатора Adam теми же авторами, использующими нормы бесконечности (отсюда «max») [12]. Этот оптимизатор развивает идею Adam на один шаг дальше и для текущего градиента принимает максимум beta2 прошлых градиентов и текущий градиент. Это приводит к меньшему и более плавному движению к минимумам.

Результаты обучения модели сверточной нейронной сети

Для сравнения результатов обучения модели нейронной сети с использованием всех выше перечисленных оптимизаторов были выбраны критерии: точность предсказания, функция потерь, количество правильно распознанных изображений с лесным пожаром, количество правильно распознанных изображений без лесного пожара, время обучения модели сверточной нейронной сети (Таблица 1).

Таблица 1 – Критерии для сравнения результатов обучения модели нейронной сети с использованием различных оптимизаторов

№	Критерий для оценки	Способ оценивания
1.	Точность предсказания	Чем выше результат, тем лучше обученная модель будет предсказывать данные
2.	Функция потерь	Чем ниже результат, тем лучше обученная модель будет предсказывать данные
3.	Количество правильно распознанных изображений с лесным пожаром	По количеству правильно предсказанных изображений с лесным пожаром
4.	Количество правильно распознанных изображений без лесного пожара	По количеству правильно предсказанных изображений без лесного пожара
5.	Время обучения	По времени обучения модели сверточной нейронной сети

В таблице 2 представлены результаты обучения сверточной нейронной сети, используя все перечисленные оптимизаторы.

Таблица 2 – Результаты обучения сверточной нейронной сети

Оптимизатор	Точность предсказания (%)	Функция потерь (%)	Правильно распознано изображений без лесного пожара (шт.)	Правильно распознано изображений с лесным пожаром (шт.)	Время обучения
SGD	82,143	2,119	17	25	20 мин.
RMSprop	91,643	0,186	21	22	15 мин.
Adam	94,389	0,116	23	25	13 мин.
Adadelta	89,643	0,270	22	17	16 мин.
Adamax	90,357	0.752	22	18	10 мин.

Модель обучались с различными оптимизаторами без добавления новых изображений, на 20 итерациях. Обучающая выборка содержала 80%, а проверяющая 20% от тренеровочных данных.

Самый точный результат смогла предсказать модель, обучаясь с использованием оптимизатора Adam. Функция потерь составила 0,116, а

точность предсказания составила 94,389%, т.е. данная модель сможет распознать класс изображения верно в 94 случаях из 100. Модель обучалась 13 минут и правильно предсказала 25 из 25 изображений с лесным пожаром правильно, а изображения без лесного пожара 23 из 25. Эти 2 изображения содержали много оранжевого цвета на фоне, поэтому модель присвоила им класс лесного пожара.

На рисунке 3 представлен график функции потерь во время обучения модели сверточной нейронной, обучаясь с использованием оптимизатора Adam. Синим цветом, показана функция потерь обучающей выборки, а красным – проверяющей.

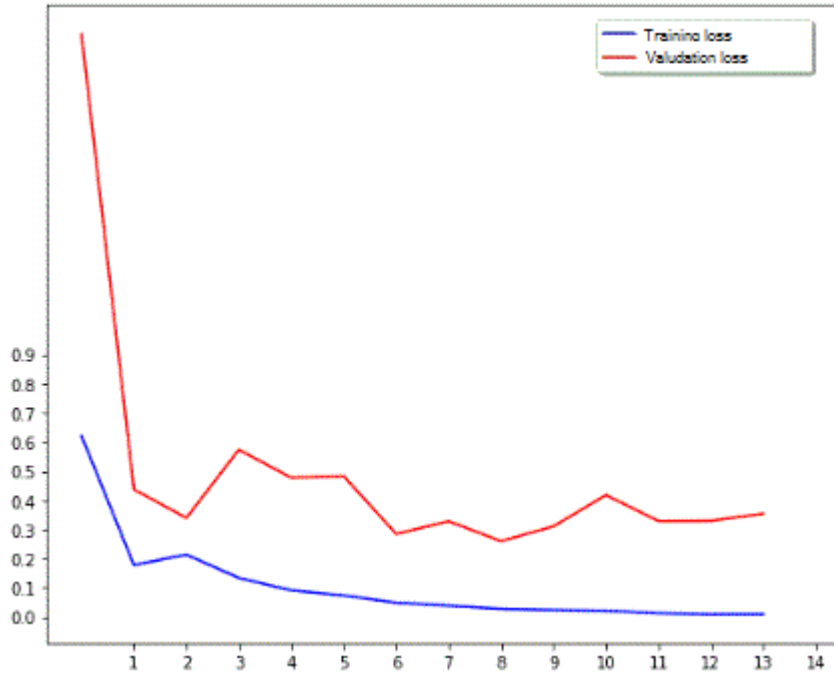


Рисунок 3 – График функции потери обучающей (синим цветом) и проверяющей (красным цветом) выборок

На рисунке 4 представлен график предсказания во время обучения модели сверточной нейронной, обучаясь с использованием оптимизатора Adam. Синим цветом, показана точность предсказания обучающей выборки, а красным – проверяющей.

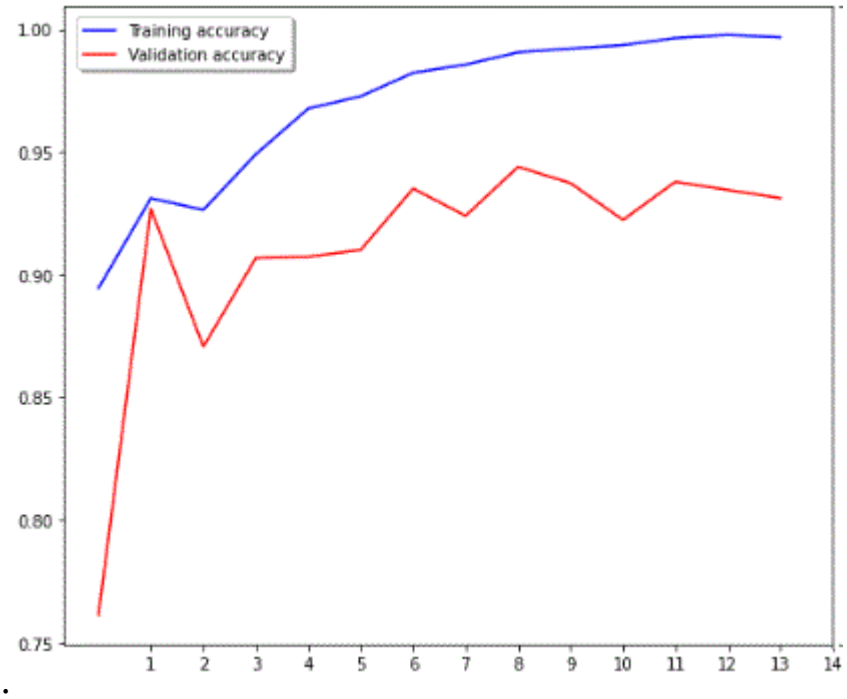


Рисунок 4 – График точности предсказания обучающей (синим цветом) и проверяющей (красным цветом) выборок

Заключение

В результате проведенных исследований было выявлено, что оптимальным оптимизатором для решения задачи классификации лесных пожаров с помощью сверточных нейронных сетей является Adam.

При обучении модели сверточной нейронной сети было произведено сравнение эффективности оптимизаторов по следующим критериям: точность предсказания, функция потерь, количество правильно распознанных изображений с лесным пожаром, количество правильно распознанных изображений без лесного пожара, время обучения модели сверточной нейронной сети.

Результаты обучения модели сверточной нейронной сети позволило определить наиболее эффективный оптимизатор прирешении задачи классификации лесных пожаров – Adam. При обучении модели с использованием оптимизатора Adam модель имела скорость обучения 13 минут и может предсказывать данные с точностью в 94,389%. Также функция потерь составила 0,116 %. Модель правильно предсказала 25 из 25 изображений с лесным пожаром, а изображения без лесного пожара 23 из 25.

Литература

1. Keras CNN Dog or Cat Classification [Электронный ресурс]. – Режим доступа: https://www.kaggle.com/uysimty/keras-cnn-dog-or-cat-classification – (дата обращения: 22.05.22).
2. Forest Fire Images dataset [Электронный ресурс]. – Режим доступа: https://www.kaggle.com/datasets/mohnishsaiprasad/forest-fire-images – (дата обращения: 22.05.22)
3. Глубокое обучение на Python / Ф. Шолле - СПБ.: Питер, 2018. – 400 с.
4. Сверточная нейронная сеть, часть 1: структура, топология, функции активации и обучающее множество [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/348000/ – (дата обращения: 22.05.22)
5. tf.keras.layers.Conv2D [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv2D – (дата обращения: 22.05.22)
6. tf.keras.layers.BatchNormalization [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/api_docs/python/tf/keras/layers/BatchNormalization – (дата обращения: 22.05.22)
7. tf.keras.layers.MaxPool2D [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/api_docs/python/tf/keras/layers/MaxPool2D – (дата обращения: 22.05.22)
8. tf.keras.layers.Dropout [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout – (дата обращения: 22.05.22)
9. tf.keras.layers.Dense [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense – (дата обращения: 22.05.22)

10. tf.keras.layers.Flatten [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Flatten – (дата обращения: 22.05.22)
11. Полное руководство по функциям активации [Электронный ресурс]. – Режим доступа: https://www.machinelearningmastery.ru/complete-guide-of-activation-functions-34076e95d044/ – (дата обращения: 22.05.22)
12. Обзор оптимизаторов [Электронный ресурс]. – Режим доступа: https://digitrain.ru/articles/437861/ – (дата обращения: 22.05.22)

References

1. Keras CNN Dog or Cat Classification [Electronic resource]. – Access mode: https://www.kaggle.com/uysimty/keras-cnn-dog-or-cat-classification – (date of access: 22.05.22).
2. Forest Fire Images dataset [Electronic resource]. – Access mode: https://www.kaggle.com/datasets/mohnishsaiprasad/forest-fire-images – (date of access: 22.05.22)
3. Deep learning with Python / F. Chollet – St. Petersburg: Peter, 2018. – 400 c.
4. Convolutional Neural Network, Part 1: Structure, Topology, Activation, and Trainning Set [Electronic resource]. – Access mode: https://habr.com/ru/post/348000/ – (date of access: 22.05.22)
5. tf.keras.layers.Conv2D [Electronic resource]. – Access mode: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv2D – (date of access: 22.05.22)
6. tf.keras.layers.BatchNormalization [Electronic resource]. – Access mode: https://www.tensorflow.org/api_docs/python/tf/keras/layers/BatchNormalization – (date of access: 22.05.22)
7. tf.keras.layers.MaxPool2D [Electronic resource]. – Access mode: https://www.tensorflow.org/api_docs/python/tf/keras/layers/MaxPool2D – (дата обращения: 22.05.22)

8. tf.keras.layers.Dropout [Electronic resource]. – Access mode: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout – (date of access: 22.05.22)
9. tf.keras.layers.Dense [Electronic resource]. – Access mode: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense – (date of access: 22.05.22)
10. tf.keras.layers.Flatten [Electronic resource]. – Access mode: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Flatten – (date of access: 22.05.22)
11. A Complete Guide to Activation Functions [Electronic resource]. – Access mode: https://www.machinelearningmastery.ru/complete-guide-of-activation-functions-34076e95d044/ – (date of access: 22.05.22)
12. Overview of optimizers [Electronic resource]. – Access mode: https://digitrain.ru/articles/437861/ – (date of access: 22.05.22)

©Журавлева Н.А., 2022Научно-образовательный журнал для студентов и преподавателей «StudNet» №5/2022.

Для цитирования: Журавлева Н.А. ВЫБОР ОПТИМИЗАТОРА ДЛЯ ОБУЧЕНИЯ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ: ЗАДАЧА КЛАССИФИКАЦИИ ЛЕСНЫХ ПОЖАРОВ// Научно-образовательный журнал для студентов и преподавателей «StudNet» №5/2022.