ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ “ВЫСШАЯ ШКОЛА ЭКОНОМИКИ”»

Московский институт электроники и математики им. А. Н. Тихонова

ОТЧЕТ

В РАМКАХ ДИСЦИПЛИНЫ

"НИС введение в специальность"

НА ТЕМУ

"Моя первая нейронная сеть"

Выполнил
Студент образовательной программы
«Прикладная математика»,
группы БПМ-223

Бутов Никита Дмитриевич

Москва 2023 г.

Содержание

1 Введение 2

2 Данные 4

3 Методы 5

4 Результаты 8

5 Заключение 9

6 Ссылки 9

1 Введение

Нейронные сети - это не просто модное явление, они становятся неотъемлемой частью нашего будущего. В нашем проекте мы сфокусированы на создании моей первой нейросети, и это не просто увлекательное занятие, но и актуальное исследование, учитывая огромный потенциал, который они несут. В мире, где данные становятся королем, нейронные сети открывают двери к анализу и использованию информации в ранее невиданных масштабах.

Наш проект будет настоящим путешествием в мир искусственного интеллекта, и его актуальность неоспорима. Мы не только создадим свою собственную нейросеть, но и рассмотрим ее применение в реальной жизни. От разгадывания загадок до создания музыки и анализа текстов - нейронные сети могут быть использованы в самых разных областях. Так что давайте вместе отправимся в путешествие в мир искусственного интеллекта и создания нашей собственной нейросети.

- Сперва необходимо проанализировать данные, с которыми мы собираемся работать и решить какую архитектуру нейронной сети мы используем. Например для работы с табличными даннымит нам будет достаточно ансамбля вроде градиентного бустинга, а для работы с изображениями мы будем работать со слоями двумерной свертки.

- После выбора архитектуры перейдем к обучению нашей нейросети. Нейронная сеть обучается путем инициализации случайных весов, прямого распространения входных данных через сеть с последующим вычислением ошибки, обратного распространения ошибки для коррекции весов, итеративного обновления весов через несколько эпох, оценки на тестовых данных и, наконец, достижения желаемой точности, что позволяет сети извлекать закономерности из данных и адаптироваться к разным задачам, таким как классификация, регрессия и другие.

- После завершения обучения нейронной сети необходимо провести тестирование, чтобы оценить ее способность к обобщению на новые, ранее не виденные данные. Это позволяет убедиться, что сеть действительно научилась извлекать обобщенные закономерности из обучающих данных, а не просто "запомнила"их. Тестирование помогает оценить точность и надежность модели, что важно при применении нейронных сетей в реальных задачах и приложе-

ниях.высокая, то сеть может использоваться для решения задач в реальном времени.

Заключительное замечание: внимательное обучение и тщательное тестирование позволяют достичь высокой эффективности и надежности сетей в широком спектре приложений, начиная от распознавания образов и автоматической обработки данных, и заканчивая генерацией контента и управлением системами. Этот процесс является ключевым фактором в успехе нейронных сетей в современном мире искусственного интеллекта.

Обучение нейронных сетей может сопровождаться рядом сложностей, включая проблемы с переобучением (когда сеть слишком хорошо подстраивается под обучающие данные и не способна обобщать на новые), недостаточным объемом данных (что может привести к недостаточной обобщающей способности сети), выбором оптимальной архитектуры сети и параметров обучения, а также вычислительной сложностью, требующей мощных вычислительных ресурсов. Кроме того, наличие шума в данных и несбалансированные классы также могут создавать проблемы при обучении нейросетей. Решение этих проблем требует глубокого понимания технических аспектов искусственного интеллекта и опыта в настройке и тестировании моделей.

Несмотря на эти проблемы, нейронные сети - мощный инструментом в области искусственного интеллекта. Они находят применение в широком спектре задач, от определения и прогнозирования тенденций до управления процессами и принятия решений. С развитием новых технологий и алгоритмов, возможности нейронных сетей будут только увеличиваться в будущем.

Цель данного исследования заключается в разработке классификатора человеческих эмоций, по изображению. Разработка будет вестись в среде программирования Google Colaboratory на языке Python 3 с приминением библиотеки keras tensorflow. В работе будет представлено описание данных, на основе которых обучается, настраивается и тестируется нейронная сеть, а также процесс подготовки этих данных. Дополнительно к этому будет дано более подробное объяснение о работе сверточных нейронных сетей и почему они лучше всего подходят для обработки изображений. В конце будет проведён анализ полученных результатов и подведены итоги работы.

2 Данные

При выборе данных для обучения нейросети важно учесть, что качество и разнообразие данных оказывают прямое влияние на производительность модели. Данные должны быть точными и репрезентативными для задачи, которую вы решаете. Более того, обеспечение разнообразия данных позволяет сети извлекать обобщенные закономерности и делает ее устойчивой к различным входам.

Баланс классов также критически важен, чтобы сеть не сконцентрировалась только на наиболее представленных классах, игнорируя менее распространенные. Разделение данных на обучающую, валидационную и тестовую выборки позволяет оценить производительность и обобщающую способность модели на независимых наборах данных.

Препроцессинг данных может потребоваться для нормализации, удаления выбросов или изменения признаков с целью улучшения производительности модели. Все эти факторы совместно определяют эффективность обучения нейронных сетей, делая выбор данных критически важным шагом.

Для обучения нашей нейронной сети мы используем базу данных, содержащую изображения людей. Эта база данных не проходила предобработку, поэтому сделаем это сами. Для начала загрузим изображения используя модуль PIL в одном формате, с заданым размером 150 на 150 пикселей и сохраним их в формате numpy.array. Так как мы работаем с довольно маленьким датасетом, давайте попробуем увеличить его искуственно. Для этого снова воспользуемся методами библиотеки PIL и при загрузке исходных данных будем сохранять еще 35 изображений: Исходное изображение повернутое на 10°, 20°, 30°и т. д.

Наш исходный датасет разделен на три приблизительно равных по размеру класса: sad, angry и happy. Для представления этих классов и предсказания их нейросетью воспользуемся кодировкой OneHotEncoding и функцией библиотеки sklearn OneHotEncoder. Таким образом:

1 0 0 - angry

0 1 0 - happy

0 0 1 - sad

Теперь разделим данные на train и test. Будем использовать метод train_test_split из библиотеки sklearn.

Установим параметр shuffle = True, чтобы перемешать данные. Размер test выборки сделаем 20% от исходного датасета

3 Методы

Сверточные нейронные сети (Convolutional Neural Networks, CNN) - это специализированный класс глубоких нейронных сетей, разработанный для обработки данных, имеющих пространственную структуру, таких как изображения и видео. Они являются основой многих современных приложений в области компьютерного зрения, включая распознавание объектов, классификацию изображений, сегментацию, а также анализ и генерацию изображений.

Основной характеристикой сверточных нейронных сетей являются сверточные слои, которые позволяют автоматически извлекать важные признаки из входных данных. Эти слои применяют фильтры (ядра свертки) к различным частям входных данных, что позволяет обнаруживать различные текстурные и структурные характеристики. Затем результаты свертки объединяются и передаются через слои пулинга (подвыборки), чтобы уменьшить размер данных и извлечь наиболее информативные признаки.

Сверточные нейронные сети обычно имеют архитектуру, состоящую из нескольких сверточных слоев, слоев пулинга и полносвязных слоев. Они обучаются с использованием алгоритмов градиентного спуска, и при правильной настройке архитектуры и параметров они могут достичь выдающейся производительности в задачах компьютерного зрения. Сверточные нейронные сети стали основой для множества инновационных приложений, включая автоматическое распознавание лиц, самоуправляемые автомобили и медицинскую диагностику на основе изображений.

MaxPooling2D - это операция, широко используемая в сверточных нейронных сетях (CNN) для уменьшения размера пространственных измерений в данных. Она применяется к каждому каналу (или слою) входных данных независимо, чтобы уменьшить объем данных и сделать вычисления более эффективными. Эта операция особенно полезна для уменьшения количества параметров и вычислений в модели, а также для создания инвариантности к небольшим трансляциям объектов на изображениях.

Принцип MaxPooling2D заключается в разделении входной области на малень-

кие прямоугольные или квадратные блоки и выборе максимального значения из каждого блока. Результатом операции является уменьшенное изображение (или слой) с сохраненными самыми важными признаками. Например, при использовании MaxPooling2D с размером окна 2x2, каждый 2x2 блок входных данных будет заменен максимальным значением в этом блоке.

MaxPooling2D помогает сети фокусироваться на самых важных признаках и уменьшает чувствительность к небольшим изменениям в данных. Это уменьшение пространственных размеров также уменьшает вычислительную нагрузку и предотвращает переобучение. MaxPooling2D часто применяется после сверточных слоев в архитектуре CNN, чтобы последовательно уменьшать размеры данных и постепенно агрегировать признаки, ведущие к финальным предсказаниям модели.

Batch Normalization (BatchNorm) - это техника регуляризации и нормализации данных, широко используемая в глубоких нейронных сетях, особенно в сверточных и полносвязных слоях. Ее цель заключается в стабилизации распределения активаций (выходов) между слоями сети во время обучения, что способствует более быстрой и стабильной сходимости модели. BatchNorm делает обучение более эффективным и позволяет использовать более высокие скорости обучения, что ускоряет процесс обучения.

Основной принцип Batch Normalization состоит в том, что для каждой батча (набора данных, обрабатываемого одновременно) вычисляются среднее и стандартное отклонение активаций внутри слоя. Затем данные активации нормализуются путем центрирования и масштабирования по этим средним и стандартным отклонениям. Это выравнивает распределение активаций и делает их более устойчивыми к изменениям весов и сдвигам.

Помимо нормализации, BatchNorm включает в себя два дополнительных параметра - гамма (scale) и бета (shift), которые позволяют сети "выучивать"оптимальные сдвиги и масштабы для каждой активации. Это делает BatchNorm более мощным инструментом и помогает сети лучше адаптироваться к данным.

Batch Normalization также имеет положительное воздействие на обобщающую способность модели и помогает уменьшить эффект переобучения. Эта техника стала стандартной практикой в глубоком обучении и позволила обучать более глубокие и более эффективные нейронные сети.

Flatten - это операция в глубоких нейронных сетях, которая преобразует мно-

гомерные данные, такие как изображения или трехмерные тензоры, в одномерный вектор. Это позволяет использовать выходы слоев сверточных нейронных сетей или рекуррентных нейронных сетей в полносвязных слоях или других слоях с одномерными входами. Например, после извлечения признаков из изображения с помощью сверточных слоев, Flatten представляет эти признаки как одномерный вектор и передает его для классификации или другой задачи. Эта операция часто используется в конце архитектур нейронных сетей, чтобы свести многомерные данные к формату, который подходит для финальных вычислений и вывода.

Слой Dense (полносвязный слой) в нейронной сети представляет собой стандартный слой, в котором каждый нейрон входного слоя соединен с каждым нейроном выходного слоя. Это означает, что каждый входной признак взаимодействует с каждым нейроном в слое Dense. Dense слой выполняет линейную комбинацию входных признаков, взвешивая их с соответствующими весами, и затем применяет функцию активации к полученным суммам.

Слой Dense может иметь разное количество нейронов в выходном слое, что позволяет модели извлекать различные степени абстракции и обучаться на разных уровнях признаков. Так, в задаче классификации, последний слой Dense часто используется для вывода классификационных результатов, а промежуточные слои могут служить для извлечения и агрегации признаков из данных.

Для улучшения способности модели к аппроксимации сложных функций обычно используются нелинейные функции активации, такие как ReLU (Rectified Linear Activation) или sigmoid. Эти функции добавляют нелинейность в слой Dense, что позволяет модели обучаться сложным взаимосвязям между признаками и более точно моделировать разнообразные данные.

Последний слой в нашей сетке model.add(Dense(3, activation='softmax')) возвращает вектор в формате OneHotEncoding. Активационная функция 'softmax' - классическое решение в задачах классификации.

В данной задаче классификации мы будем использовать последовательную модель нейронной сети. Последовательная модель - это одна из наиболее простых и распространенных архитектур нейронных сетей. Эта модель представляет собой стек слоев нейронов, где каждый следующий слой принимает данные от предыдущего и передает их следующему. Основное преимущество последовательной модели заключается в ее простоте и удобстве использования. Для создания такой модели с использованием библиотеки Keras, достаточно создать экземпляр

класса Sequential и последовательно добавлять слои, указывая их параметры с помощью метода add().

Сама последовательная модель будет иметь следующую архитектуру:

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 batch_normalization_1 (Bat  (None, 150, 150, 3)       12
 chNormalization)

 conv2d_3 (Conv2D)           (None, 150, 150, 32)      896

 max_pooling2d_3 (MaxPoolin  (None, 75, 75, 32)        0
 g2D)

 conv2d_4 (Conv2D)           (None, 75, 75, 64)        18496

 max_pooling2d_4 (MaxPoolin  (None, 37, 37, 64)        0
 g2D)

 conv2d_5 (Conv2D)           (None, 37, 37, 128)       73856

 max_pooling2d_5 (MaxPoolin  (None, 18, 18, 128)       0
 g2D)

 flatten_1 (Flatten)         (None, 41472)             0

 dense_2 (Dense)             (None, 128)               5308544

 dense_3 (Dense)             (None, 3)                 387

=================================================================
Total params: 5402191 (20.61 MB)
Trainable params: 5402185 (20.61 MB)
Non-trainable params: 6 (24.00 Byte)
_________________________________________________________________
```

4 Результаты

Процесс обучения модели занимает некоторое время, и в данном исследовании требовалось около получаса для завершения этого этапа. Для удобства дальнейшего использования, библиотека Keras предоставляет функциональность для сохранения обученной модели в файл и ее последующей загрузки. После завершения обучения, на тестовых данных была достигнута точность на уровне 92%. Вся информация о процессе обучения модели сохраняется в истории обучения (переменная "history"). Результаты обучения были визуализированы на графиках для наглядности. На рисунке 1 показана динамика изменения точности предсказаний модели.

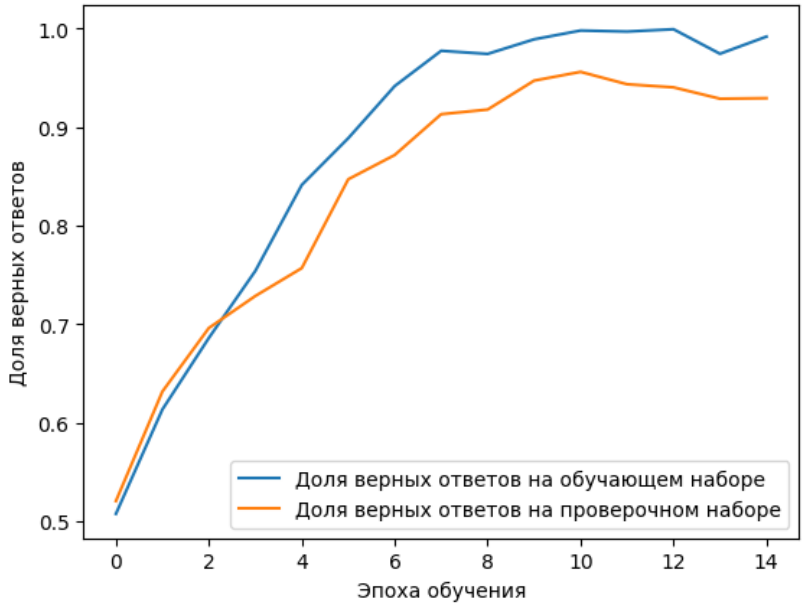

Рис. 1: Изменение точности ответов на данных для обучения (синий) и на валидационных данные (оранж.)

5 Заключение

Нейронные сети представляют собой мощный инструмент в области машинного обучения, который может успешно применяться в широком спектре задач, начиная с распознавания образов и заканчивая прогнозированием временных рядов. Сверточные нейронные сети, особенно, отлично справляются с обработкой изображений и видео, так как они способны выделять ключевые характеристики изображений и использовать их для классификации или сегментации. Keras, как одна из наиболее популярных библиотек, предоставляет простой и гибкий подход к созданию и обучению нейронных сетей. С её помощью можно легко разрабатывать сложные модели нейронных сетей и настраивать параметры для достижения высокой точности в прогнозировании.

6 Ссылки

- Русскоязычная документация Keras : сайт. – URL: https://ru-keras.com/
- Свёрточная нейронная сеть // Wikipedia : сайт. – URL: https://ru.wikipedia.org/wiki/Свёрточная_нейронная_сеть)
- Функция активации // Wikipedia : сайт. – URL: https://ru.wikipedia.org/

wiki/Функция_активации