УДК 004.8 **Информационные технологии**

Черкасова Ирина Сергеевна, *студентка магистратуры 2 курс, МИРЭА-Российский технологический университет (РТУ МИРЭА),*
Россия, г. Москва, Институт кибербезопасности и цифровых технологий
Россия, г. Москва

ОПТИМИЗАЦИЯ ГИПЕРПАРАМЕТРОВ НЕЙРОННОЙ СЕТИ И СНИЖЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ ЗАТРАТ

Аннотация: В данной статье проводится оптимизация гиперпараметров нейронной сети, а точнее выбор наиболее подходящей скорости обучения, а также предлагается способ уменьшения вычислительных затрат. Данное исследование показало, что наилучший результат демонстрирует модель при скорости обучения 1e-4, вероятность верного распознавания на тестовом наборе составила 92,07%. При корректировке структуры модели было уменьшено время работы сети, но при этом качество работы упало несильно. Вероятность верного распознавания на тестовом наборе составила при использовании новой структуры составила 84,37%.

Ключевые слова: Сверточные Нейронные Сети, Оптимизация, Гиперпараметры, Структура модели.

Abstract: This article optimizes the hyperparameters of the neural network, or rather the choice of the most appropriate learning speed, and also proposes a way to reduce computational costs. This study showed that the best result is demonstrated by the model at a learning speed of 1e-4, the probability of correct recognition on the test set was 92.07%. When adjusting the structure of the model, the network operating time was reduced, but the quality of work fell slightly. The probability of correct recognition on the test set was 84.37% when using the new structure.

Keywords: Convolutional Neural Networks, Optimization, Hyperparameters,

Model Structure.

Введение

В работе [1] были рассмотрены различные модели – VGG16, ResNet50, Invision v3. Модели были включены в составную сверточную нейронную сеть, которая будет использоваться для классификации видео контента. В этой работе будет рассмотрен автоматический подбор гиперпараметров сети, а именно, скорость обучения, для улучшения эффективности работы, а также предложен метод для решения проблемы энергоэффективности с целью уменьшения затрат ресурсов.

Выбор скорости обучения нейронной сети

Гиперпараметры – это параметры, которые влияют на качество работы нейронной сети, но не определяются в процессе обучения. В первую очередь это архитектура нейронной сети, а именно количество слоев и количество нейронов в каждом слое. Функции активации, которые используются в различных слоях. А также особенности обучения нейронной сети – тип оптимизатора, который используется при обучении, количество эпох обучения и многое другое.

Все гиперпараметры влияют друг на друга, поэтому подобрать оптимальную комбинацию гиперпараметров, при которой используется максимальное качество работы нейронной сети вручную достаточно проблематично.

Цель оптимизации гиперпараметров в машинном обучении состоит в том, чтобы найти гиперпараметры данного алгоритма машинного обучения, которые обеспечивают наилучшую производительность, измеренную на валидационном наборе (рисунок 1).

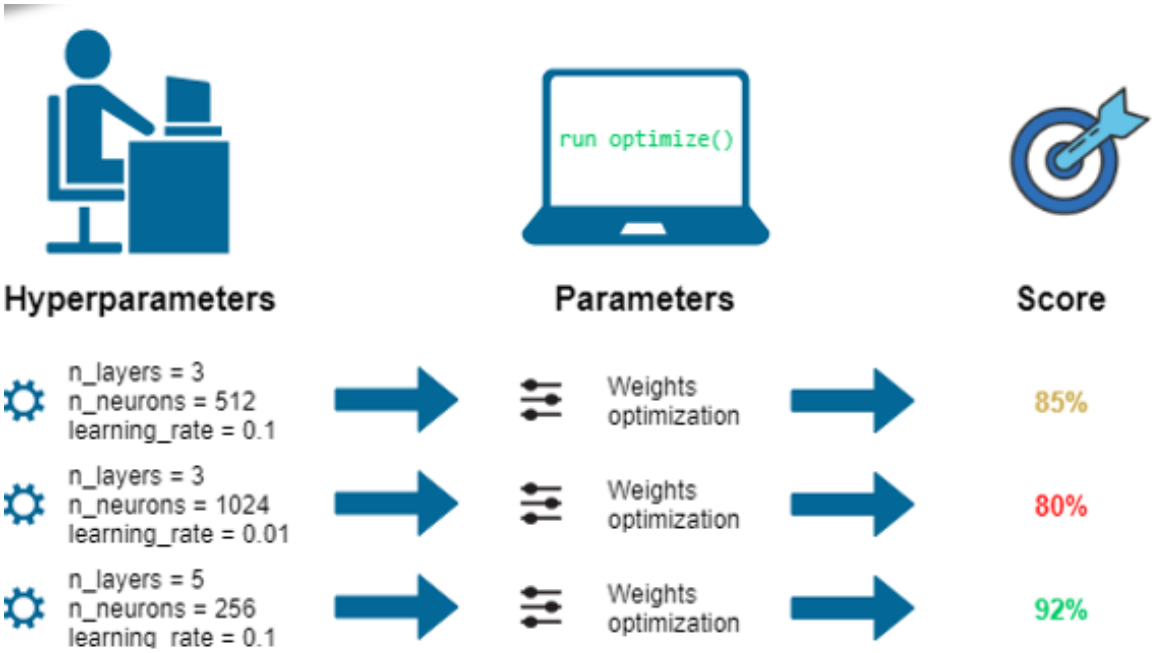

Рисунок 1 – Рабочий процесс оптимизации гиперпараметров

Проблема с оптимизацией гиперпараметров состоит в том, что оценка целевой функции для нахождения оценки чрезвычайно дорога. Каждый раз, когда используются новые гиперпараметры, мы должны обучать модель на данных обучения, делать прогнозы на данных валидации, а затем вычислять метрику валидации. С большим количеством гиперпараметров и сложных моделей, таких как ансамбли или глубокие нейронные сети, на обучение которых может уйти несколько дней, этот процесс быстро становится трудным для выполнения вручную, поэтому существуют различные алгоритмы для нахождения оптимальных гиперпараметров автоматически. Одним из таких алгоритмов является байесовская оптимизация.

В байесовской оптимизации значения гиперпараметров в текущей итерации выбираются с учётом результатов на предыдущем шаге. Основная идея алгоритма заключается в следующем – на каждой итерации подбора находится компромисс между исследованием регионов с самыми удачными из найденных комбинаций гиперпараметров и исследованием регионов с большой неопределённостью (где могут находиться ещё более удачные комбинации). Это позволяет во многих случаях найти лучшие значения параметров модели за

меньшее количество времени.

Применим цикл из следующих шагов: на основе байесовской оптимизации выбираем гиперпараметр, применяем этот гиперпараметр, оцениваем ошибку, переоцениваем и заново выбираем гиперпараметр. Этот цикл повторяем N раз. Таким образом оптимизируется скорость обучения сети, что приводит к ~~наи~~более точным результатам.

Для нахождения оптимальной скорости обучения была установлена утилита «Keras Tuner» - оптимизатор гиперпарамтеров, который был разработан командой Google в 2015 году специально для Keras в системе Tensorflow. С ее помощью можно легко настроить свое пространство поиска нужных гиперпараметров, также она имеет встроенные алгоритмы поиска, такие, как:

1) RandomSearch – алгоритм случайного поиска.
2) Hyperband –алгоритм оптимизации на основе многорукого бандита.
3) BayesianOptimization – байесовская оптимизация.

Использовать Keras Tuner достаточно просто. Необходимо написать функцию, которая будет создавать нейронную сеть с возможностью изменения гиперпараметров. В программе эта функция имеет название «build_model» – построение модели.

Было определено пространство поиска – предложены четыре скорости обучения 1e-2, 1e-3, 1e-4, 1e-5. Общее количество испытаний равно четырем. Количество эпох обучения осталось неизменным – пять. Выходной слой остается без изменений. В функции создания модели ее также необходимо скомпилировать, поэтому вызывается метод «compile» с оптимизатором «Adam», где и будет происходить подбор оптимальной скорости обучения.

На вход функция получает объект «hp» - движок гиперпараметра Keras Tuner. Именно этот движок используется для заполнения значений гиперпараметров из заданного диапазона.

Далее идет шаг создания тюнера, который и будет заниматься подбором гиперпараметров.

В тюнере прописываются:

1) Функция создания модели.

2) Метрика, по которой оценивается качество набора гиперпарметров – доля правильных ответов на проверочном наборе.

3) Максимальное количество запусков процесса обучения нейронной сети и с разным набором гиперпараметров.

Для подбора используется функция tuner.search. Она вызывается также как функция fit у модели. Ей необходимо указать данные для обучения, размер мини-выборки, количество эпох и данные, которые будет использоваться для проверки.

Результаты вывода программного кода представлены ниже, также для более удобного анализа они занесены в таблицу.

Таблица 1 – результат обучения при подборе гиперпарамтеров

Номер испытания	Кол-во эпох, шт.	Скорость обучения	Время обучения, минуты	Вероятность верного распознавания на тестовом наборе данных, %
1	5	1e-2	122	91,20
2	5	1e-4	78	92,07
3	5	1e-5	85	91,11
4	5	1e-3	87	91,86

Из таблицы видно, что лучшие результаты обучения нейросетевой модели достигаются при скорости обучения 1e-4, при которой обеспечивается вероятность верного обнаружения на тестовом наборе данных 92,07%.

Важно, что при повышении эффективности обучения модели, затрачиваемое время увеличивается не сильно, что способствует решению проблемы энергоэффективности.

Решение проблемы энергоэффективности

Для снижения вычислительных затрат в данной статье было предложено использовать в начале сети дополнительные слои для уменьшения размера

входного изображения. Структура предложенной модели представлена на рисунке 2.

На нем можно увидеть, что было добавлено 4 слоя:

1. Conv2D(3, (7, 7)
2. Activation('relu')
3. Conv2D(3, (3, 3),strides=(2, 2)
4. Activation('relu')

Представленные слои позволяют выделить основные признаки и значительно сократить размер входящего изображения. Так на рисунке видно, что параметры изображения уменьшились в два раза. При этом слои классификации обучаются на тренировочной выборке изображений небольшого размера, а в дальнейшем, при использовании модели на устройствах с высоким разрешением дополнительно обучаются только входные слои, изменяющие масштаб.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 300, 300, 3)       444
_________________________________________________________________
activation_6 (Activation)    (None, 300, 300, 3)       0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 150, 150, 3)       84
_________________________________________________________________
activation_7 (Activation)    (None, 150, 150, 3)       0
_________________________________________________________________
module_wrapper_3 (ModuleWrap (None, 1)                 22982561
=================================================================
Total params: 22,983,089
Trainable params: 528
Non-trainable params: 22,982,561
```

Рисунок 2 – Архитектура новой нейросетевой модели

После добавления слоев идет загрузка ранее обученной модели, но с отключением процесса обучения по всем слоям. Таким образом новая модель будет обучена только на 528 параметрах, что должно уменьшить время обучения,

но при этом позволить сохранить качество обучения сети.

Стоит обратить внимание, что в подгруженной модели – слой «module_wrapper_3» заранее была прописана скорость обучения 1e-4, так как именно эта скорость была определена на предыдущем шаге – оптимизации гиперпараметров.

Вероятность верного распознавания составила 84,37% на тестовом наборе данных. Время обучения составило 48 минут в облачной среде для работы с Python – Google Colab. На рисунках 3 и 4 представлены графики работы нейронной сети, а именно, график вероятности верного распознавания и вероятность ошибки.

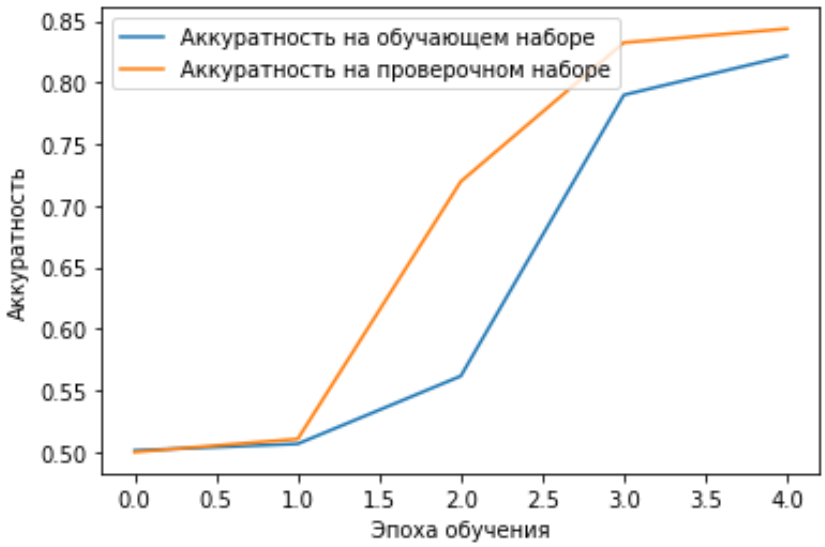

Рисунок 3 – График вероятности правильного распознавания

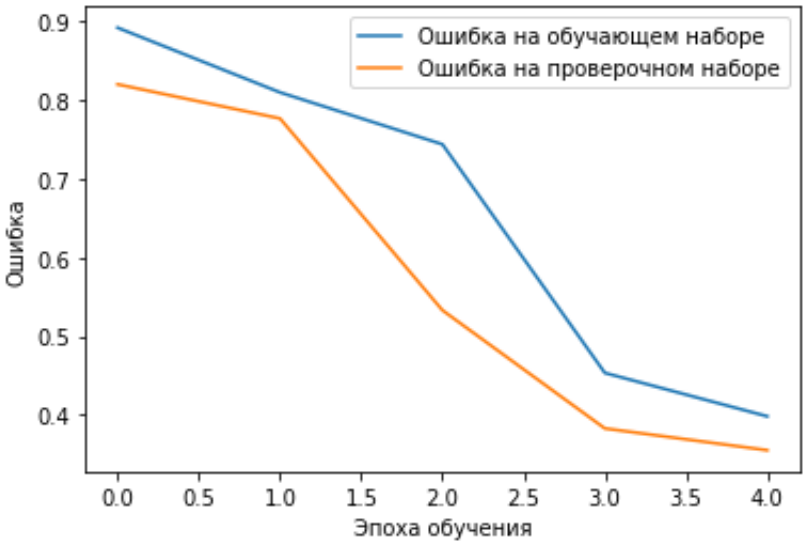

Рисунок 4 - График вероятности правильного распознавания и вероятности ошибки

Заключение

В ходе представленного исследования были проанализированы алгоритмы оптимизации гиперпараметров, что позволило улучшить ранее исследованную модель нейронной сети. Таким образом была определена наиболее подходящая скорость обучения модели – 1e-4. При использовании этой скорости вероятность верного распознавания составила 92,07%.

Далее был предложен метод по оптимизации модели для уменьшения времени обучения, чтобы сократить потребление ресурсов. При помощи создания новой модели удалось уменьшить время обучения нейронной сети до 48 минут, при этом не потерпеть значительные потери в качестве обучения. Вероятность верного распознавания сети в конечном счете составила 84,37%.

Библиографический список:

1. Черкасова И. С. Классификация видео контента на основе сверточных нейронных сетей // E-Scio [Электронный ресурс]: Электронное периодическое издание «E-Scio.ru» — Эл № ФС77-66730 — Режим доступа: http://e-scio.ru/wp-content/uploads/2021/12/Черкасова-И.-С.pdf.

2. Исследуем архитектуры сверточных нейронных сетей, Режим доступа: свободный [Электронный ресурс] – URL - https://proglib.io/p/issleduem-arhitektury-svertochnyh-neyronnyh-setey-s-pomoshchyu-fast-ai-2020-12-28 (Дата обращения: 30.11.2021).

3. Пучков А. Ю., Дли М. И. Алгоритм настройки гиперпараметров сверточной нейронной сети в задаче классификации объектов //Математические методы в технике и технологиях-ММТТ. – 2018. – Т. 4. – С. 47-50.

4. Wikipedia. Оптимизация гиперпарамтеров, Режим доступа: свободный [Электронный ресурс] – URL - https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%B3%D0%B8%D0%BF%D0%B5%D1%80%D0%BF%D0%B0%D1%80%D0%B0%D0%BC%D0%B5%

D1%82%D1%80%D0%BE%D0%B2 (Дата обращения: 20.02.2022).

5. Пятакович В. А., Василенко А. М., Хотинский О. В. Аналитическая конструкция и исходные структуры искусственной нейронной сети, техническая реализация модели математического нейрона //Вестник евразийской науки. – 2017. – Т. 9. – №. 3 (40). – С. 89.