**Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая кибернетика и информационные технологии»

Отчет по лабораторной работе №3

по дисциплине «Системы машинного зрения»

на тему:

«Разработка нейросетевых функций. Операция Convolution Transpose»

Выполнил:

студент группы БВТ2003

Никитин Степан Романович

Москва, 2023 г.

Оглавление

[Цель работы 3](#_Toc153253044)

[Задание 3](#_Toc153253045)

[Теоретические сведения 4](#_Toc153253046)

[Выполнение работы 5](#_Toc153253047)

[Дополнительное задание 6](#_Toc153253048)

[Заключение 10](#_Toc153253049)

# Цель работы

Разработать алгоритм, используя язык python, реализующий работу операции транспонированной 2d свертки.

# Задание

При выполнении лабораторной работы необходимо:

* ознакомиться с описанием операции библиотеки PyTorch (<https://pytorch.org/docs/stable/generated/torch.nn.ConvTranspose2d>);
* используя язык программирования Python написать алгоритм, реализующий алгоритм свертки;
* составить отчет по лабораторной работе;
* дополнительное задание: реализовать алгоритм работы транспонированной свертки, через алгоритм двумерной свертки, реализованный в первой лабораторной. Необходимо перерассчитать входные параметры.

# Теоретические сведения

Операция свертки с транспозицией, также известная как транспонированная свертка, представляет собой обратную операцию к обычной свертке и находит применение в нейросетевых архитектурах для увеличения размерности входных данных. Эта операция часто используется в качестве первого слоя в некоторых нейронных сетях для увеличения размера изображений или добавления дополнительных каналов.

Принцип работы свертки с транспозицией заключается в применении фильтра к входным данным и создании нового массива с большим количеством элементов, чем исходный. Отличительной особенностью является транспонирование фильтра перед его применением к входным данным, что приводит к увеличению размерности входных данных. Также данная операция может быть использована для добавления дополнительных каналов в изображение, что полезно при обучении глубоких нейронных сетей.

Одним из ключевых преимуществ использования свертки с транспозицией является возможность увеличения количества параметров в сети без увеличения общего объема вычислений. Это особенно полезно в контексте обучения глубоких нейронных сетей, где увеличение числа параметров может содействовать повышению точности модели. Важно отметить, что данная операция также может вызвать размытие изображения, поэтому её следует использовать с осторожностью.

# Выполнение работы

В данной лабораторной работе разработан класс ConvTranspose, представляющий транспонированную сверточную операцию. Класс принимает параметры, такие как количество входных и выходных каналов, размер ядра свертки, шаг транспонированной свертки, паддинг, расширение, количество групп, использование параметра смещения и режим паддинга. Конструктор инициализирует обучаемые параметры, а метод forward выполняет транспонированную свертку с учетом указанных параметров, возвращая тензор результатов операции.



Рисунок 1 – Реализация класса ConvTranspose

Данный код выполняет тестирование транспонированной свертки с использованием метода test\_conv\_transpose. В нем создается экземпляр класса ConvTranspose с одним входным и одним выходным каналом, размером ядра свертки (3x3), шагом транспонированной свертки 2, паддингом 1 и режимом кругового паддинга. Затем создается входной тензор размером (1, 1, 4, 4) состоящий из единиц, и выполняется транспонированная свертка. Результат теста выводится в консоль, предоставляя пример входных и выходных данных операции транспонированной свертки.



Рисунок 2 – Вызов метода test\_conv\_transpose для вывода результата

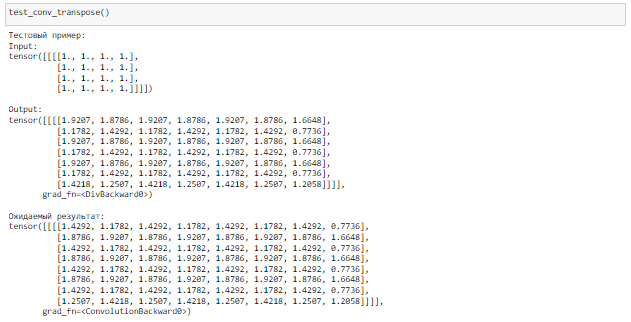


Рисунок 3 – Результат выполнение метода test\_conv\_transpose

## Дополнительное задание

Класс Conv2DTranspose реализует транспонированную двумерную свертку. В конструкторе устанавливаются параметры операции, такие как количество входных и выходных каналов, размер ядра и шаг транспонированной свертки. Веса (транспонированное ядро) и смещение инициализируются случайными значениями.

Метод forward выполняет транспонированную свертку на входном тензоре с учетом заданных параметров, создавая трехмерный тензор с результатами.

Рисунок 4 - Реализация класса Conv2DTranspose

Функция run\_transpose\_test предназначена для тестирования транспонированной свертки. Она создает экземпляр Conv2DTranspose, применяет его к случайному входному тензору, и сравнивает результаты с PyTorch. Параметры включают в себя характеристики операции, такие как размер ядра, шаг, и наличие смещения. Тест выводит информацию о совпадении результатов транспонированной свертки и ожидаемых результатов, обеспечивая проверку корректности работы операции с различными параметрами.

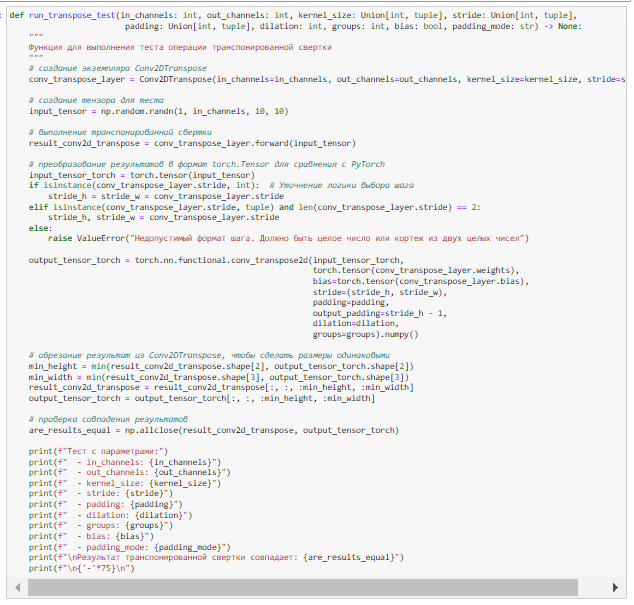


Рисунок 5 – Реализация метода run\_transposed\_conv\_test

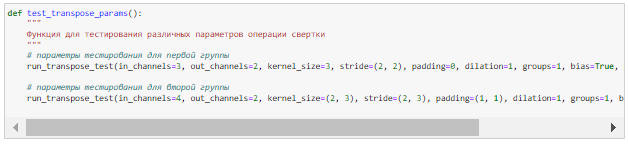


Рисунок 6 - Реализация метода test\_transpose\_params

Метод test\_transpose\_params создает и тестирует экземпляры класса Conv2DTranspose с различными параметрами для операции транспонированной свертки. Этот вызов функции предоставляет визуальное представление результатов работы класса, выводя в консоль тестовые примеры с разными параметрами входных и выходных каналов, размерами ядра свертки и шагами транспонированной свертки.



Рисунок 7 – Вывод результата

# Заключение

В ходе выполнения данной лабораторной работы я разработал алгоритм на языке Python, который реализует операцию транспонированной 2d свертки. Также я создал тесты для проверки корректности работы алгоритма.

Ссылка на git-репозиторий: <https://github.com/anothersnn/SMZ.git>