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 «Математической кибернетики и информационных технологий»

Отчет по лабораторной работе №3

по дисциплине «Системы машинного зрения»

Выполнила студент группы

БВТ2002

Быковская А.А.

Москва 2023

**Оглавление**

[Цель работы 3](#_Toc153584858)

[Выполнение 3](#_Toc153584859)

[Выводы 5](#_Toc153584860)

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.Conv2d);
* используя язык программирования Python написать алгоритм, реализующий алгоритм свертки;
* составить отчет по лабораторной работе;
* Доп. Задание: реализовать алгоритм работы транспонированной свертки, через алгоритм двумерной свертки, реализованный в первой лабораторной. Необходимо перерассчитать входные параметры.
* Выполнение данного задания засчитывается как выполнение лабораторной № 4.

**Теоретические сведения**

Операция свертки с транспозицией увеличивает размерность входных данных и находит применение в нейронных сетях для расширения изображений или каналов. Фильтр транспонируется перед применением, увеличивая количество элементов в созданном массиве. Это позволяет добавлять каналы в изображение без увеличения вычислений. Хотя такая свертка может улучшить точность модели, она также может вызвать размытие изображений, поэтому нужно применять её осторожно.

**Основные параметры свертки в этом примере:**

**in\_channels:** Количество каналов во входных данных.

**out\_channels:** Количество каналов в выходных данных или фильтрах.

**kernel\_size:** Размер ядра свертки. Это может быть целое число (для квадратного ядра) или кортеж из двух чисел (для прямоугольного ядра).

**stride:** Шаг свертки - количество пикселей, на которое смещается окно свертки на каждом шаге.

**padding:** Добавление нулей по краям входных данных для контроля размера выходных данных.

**output\_padding:** Дополнительный пэддинг для управления размером выходных данных при использовании транспонированной свертки.

**dilation:** Разреживание (dilation) определяет интервал между элементами ядра свертки.

Начало формы

Выполнение

Основное задание:

1. Ознакомление с функциями класса Generate:
2. Понимание основных параметров свертки и транспонированной свертки:

* Изучаем основные параметры, такие как in\_channels, out\_channels, kernel\_size, stride, padding, output\_padding, dilation, bias, и padding\_mode, используемые в методе ConvTranspose2d.

1. Создание тестов с использованием unittest:

* Написание тестов для каждого метода класса Generate, чтобы убедиться в их корректной работе.

1. Проверем возвращаемые значения, размеры тензоров, обработку различных входных данных и сценариев.
2. Запуск тестов и отладка:

* Запускаем тесты с использованием unittest и отслеживаем любые ошибки или несоответствия в возвращаемых значениях.

1. Исправление и повторный запуск тестов:
2. Проверка полного покрытия:

* Убедились, что тесты охватывают различные сценарии использования и краевые случаи, чтобы удостовериться в корректности работы методов.

Доп. Задание:

1. Тестирование generate\_bias:

* Создание тестов для проверки корректности возвращаемых значений смещения (bias\_values) при разных входных параметрах.

2. Тестирование generate\_filter:

* Написание тестов для проверки формы возвращаемого тензора фильтра (filter\_tensor) для различных размеров и типов ядер свертки.

3. Тестирование остальных методов:

* Создание тестов для остальных методов (apply\_dilation, update\_feature\_map, process\_padding) для проверки их корректности при обработке данных и вычислении результатов.

4. Тестирование ConvTranspose2d:

* Написание тестов для метода ConvTranspose2d, проверяющих корректность формы возвращаемых данных (сверток) и значений при различных параметрах.

5. Запуск тестов и отладка:

* Выполнение всех тестов для проверки корректности работы методов.
* Отслеживание и устранение ошибок, если тесты не проходят успешно.

6. Проверка полного покрытия:

* Убедиться, что тесты охватывают разные сценарии использования и проверяют граничные случаи для всех методов.

Выводы

В результате выполнения лабораторной работы, я реализовала свой собственный класс для вычисления транспонированной 2D свертки на языке Python. Далее он был протестирован на индивидуальных тестах, которые завершились успехом.

Реализация: https://github.com/xaptfy/MVS/blob/main/labs/lab3.ipynb