



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Московский государственный технический
университет имени Н.Э. Баумана (национальный исследовательский
университет)» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления и искусственный интеллект

КАФЕДРА Системы обработки информации и управления

**Домашнее задание №1 по курсу «Методы машинного
обучения в автоматизированных системах обработки
информации и управления»**

Подготовили:

У Жун

ИУ5И-25М

02.06.2024

Проверил:

Гапанюк Ю. Е.

2024 г.

Задание

Домашнее задание по дисциплине направлено на анализ современных методов машинного обучения и их применение для решения практических задач. Домашнее задание включает три основных этапа:

1. выбор задачи;
2. теоретический этап;
3. практический этап.

Этап выбора задачи предполагает анализ ресурса `paperswithcode`. Данный ресурс включает описание нескольких тысяч современных задач в области машинного обучения. Каждое описание задачи содержит ссылки на наиболее современные и актуальные научные статьи, предназначенные для решения задачи (список статей регулярно обновляется авторами ресурса). Каждое описание статьи содержит ссылку на репозиторий с открытым исходным кодом, реализующим представленные в статье эксперименты. На этапе выбора задачи обучающийся выбирает одну из задач машинного обучения, описание которой содержит ссылки на статьи и репозитории с исходным кодом.

Теоретический этап включает проработку как минимум двух статей, относящихся к выбранной задаче. Результаты проработки обучающийся излагает в теоретической части отчета по домашнему заданию, которая может включать:

- описание общих подходов к решению задачи;
- конкретные топологии нейронных сетей, нейросетевых ансамблей или других моделей машинного обучения, предназначенных для решения задачи;
- математическое описание, алгоритмы функционирования, особенности обучения используемых для решения задачи нейронных сетей, нейросетевых ансамблей или других моделей машинного обучения;
- описание наборов данных, используемых для обучения моделей;
- оценка качества решения задачи, описание метрик качества и их значений;
- предложения обучающегося по улучшению качества решения задачи.

Практический этап включает повторение экспериментов авторов статей на основе представленных авторами репозитория с исходным кодом и возможное улучшение

обучающимися полученных результатов. Результаты проработки обучающийся излагает в практической части отчета по домашнему заданию, которая может включать:

исходные коды программ, представленные авторами статей, результаты документирования программ обучающимися с использованием диаграмм UML, путем визуализации топологий нейронных сетей и другими способами;

результаты выполнения программ, вычисление значений для описанных в статьях метрик качества, выводы обучающегося о воспроизводимости экспериментов авторов статей и соответствии практических экспериментов теоретическим материалам статей;

предложения обучающегося по возможным улучшениям решения задачи, результаты практических экспериментов (исходные коды, документация) по возможному улучшению решения задачи.

1. выбора задачи

Название задачи: Компьютерное зрение

Источник ресурса: [Papers with Code]

(<https://paperswithcode.com/task/image-classification>)

Связанные статьи:

1. Karen Simonyan, et al. "Very Deep Convolutional Networks for Large-Scale Image Recognition." [Ссылка](<https://arxiv.org/pdf/1409.1556v6>)
2. Alexey Dosovitskiy, et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale"

[Ссылка](<https://arxiv.org/pdf/1804.02767v1>)

Репозитории с исходным кодом:

1. (<https://github.com/pytorch/vision>)
2. (https://github.com/google-research/vision_transformer)

2. Теоретический этап

Название задачи: Компьютерное зрение

Источник ресурса: [Papers with Code]

(<https://paperswithcode.com/task/image-classification>)

Связанные статьи:

1. Karen Simonyan, et al. "Very Deep Convolutional Networks for Large-Scale Image Recognition." [Ссылка](<https://arxiv.org/pdf/1409.1556v6>)
2. Alexey Dosovitskiy, et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale"

[Ссылка](<https://arxiv.org/pdf/1804.02767v1>)

Репозитории с исходным кодом:

1. (<https://github.com/pytorch/vision>)
2. (https://github.com/google-research/vision_transformer)

1. Общий подход к решению проблем:

В данной работе исследуется влияние глубины конволюционных нейронных сетей (ConvNets) на их точность при крупномасштабном распознавании изображений. Исследование сосредоточено на использовании архитектур с очень маленькими (3×3) сверточными фильтрами путем постепенного увеличения глубины сети, и обнаруживается, что увеличение глубины до 16-19 взвешенных слоев значительно улучшает производительность предыдущих конфигураций.

2. Специфическая топология нейронной сети:

Архитектура сети состоит из нескольких конволюционных слоев, которые используют конволюционные фильтры 3×3 и конволюционные фильтры 1×1 (в некоторых конфигурациях). Сеть также включает слой максимального объединения и полностью связанный слой. В экспериментах использовались сетевые архитектуры различной глубины от 11 взвешенных слоев до 19 взвешенных слоев.

3. Математическое описание, функциональный алгоритм:

В процессе обучения использовался мелкосерийный градиентный спуск (на основе обратного распространения) с использованием импульса и затухания веса для упорядочивания обучения. Скорость обучения снижается в 10 раз, когда точность валидационного набора перестает повышаться. При инициализации весов сети использовалась стратегия обучения более мелких сетей с последующим переходом к более глубоким сетям. Использовалась функция активации ReLU и случайная инициализация весов.

4. Описание набора данных, использованного для обучения модели:

В качестве основного набора данных использовался набор ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), содержащий изображения 1000 категорий и разделенный на обучающий набор (1,3 млн изображений), проверочный набор (50 000 изображений) и тестовый набор (100 000 изображений, метки категорий не раскрываются).

5. Оценка качества решения задач:

Для оценки эффективности классификации использовались коэффициенты ошибок топ-1 и топ-5. Коэффициент ошибок топ-1 - это доля ошибок классификации нескольких категорий, а коэффициент ошибок топ-5 рассчитывается как доля предсказанных пяти лучших категорий, которые не содержат истинной категории. Кроме того, для повышения эффективности использовались многомасштабные тесты и тесты с несколькими культурами.

6. Студенческие предложения по улучшению качества решения

задач:

В тексте нет прямых рекомендаций для студентов, но некоторые предложения можно предположить, основываясь на результатах экспериментов:

Увеличить глубину сети: эксперименты показали, что более глубокие сетевые структуры могут повысить точность классификации.

Использовать многомасштабное обучение и тестирование: использование различных размеров изображений при обучении и тестировании позволяет улучшить способность модели распознавать объекты разных размеров.

Объединение нескольких моделей: производительность можно еще больше повысить, если объединить результаты нескольких моделей.

Оптимизация стратегий инициализации и регуляризации сети: соответствующие методы инициализации и регуляризации сети помогут повысить эффективность обучения и обобщенность модели.

Изучить более сложные сетевые архитектуры: хотя в данной работе используется традиционная архитектура ConvNet, для дальнейшего улучшения производительности можно попробовать более сложные архитектуры.

3. Практический этап

Повторение эксперимента

```
# Copyright 2024 Google LLC.
```

```
# Licensed under the Apache License, Version 2.0 (the "License");
```

```
# you may not use this file except in compliance with the License.
```

```
# You may obtain a copy of the License at
```

```
#
```

```
# http://www.apache.org/licenses/LICENSE-2.0
```

```
#
```

```
# Unless required by applicable law or agreed to in writing, software
```

```
# distributed under the License is distributed on an "AS IS" BASIS,
```

```
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or  
implied.
```

```
# See the License for the specific language governing permissions and
```

```
# limitations under the License.
```

```
"""setup.py for vision_transformer repo, vit_jax package."""
```

```
import os
```

```
from setuptools import find_packages
```

```
from setuptools import setup
```

```
here = os.path.abspath(os.path.dirname(__file__))
```

```

try:
    README = open(os.path.join(here, 'README.md'), encoding='utf-8').read()
except IOError:
    README = "

install_requires = [
    'absl-py',
    'aqtp!=0.1.1', # https://github.com/google/aqt/issues/196
    'clu',
    'einops',
    'flax',
    'flaxformer @ git+https://github.com/google/flaxformer',
    'jax',
    'ml-collections',
    'numpy',
    'packaging',
    'pandas',
    'scipy',
    'tensorflow_datasets',
    'tensorflow_probability',
    'tensorflow',
    'tensorflow_text',
    'tqdm',
]

tests_require = [
    'pytest',
]

__version__ = None

with open(os.path.join(here, 'version.py')) as f:
    exec(f.read(), globals()) # pylint: disable=exec-used

setup(
    name='vit_jax',
    version=__version__,
    description='Original JAX implementation of Vision Transformer models.',
    long_description=README,
    long_description_content_type='text/markdown',
    classifiers=[
        'Development Status :: 3 - Alpha',
        'Intended Audience :: Developers',
        'Intended Audience :: Science/Research',

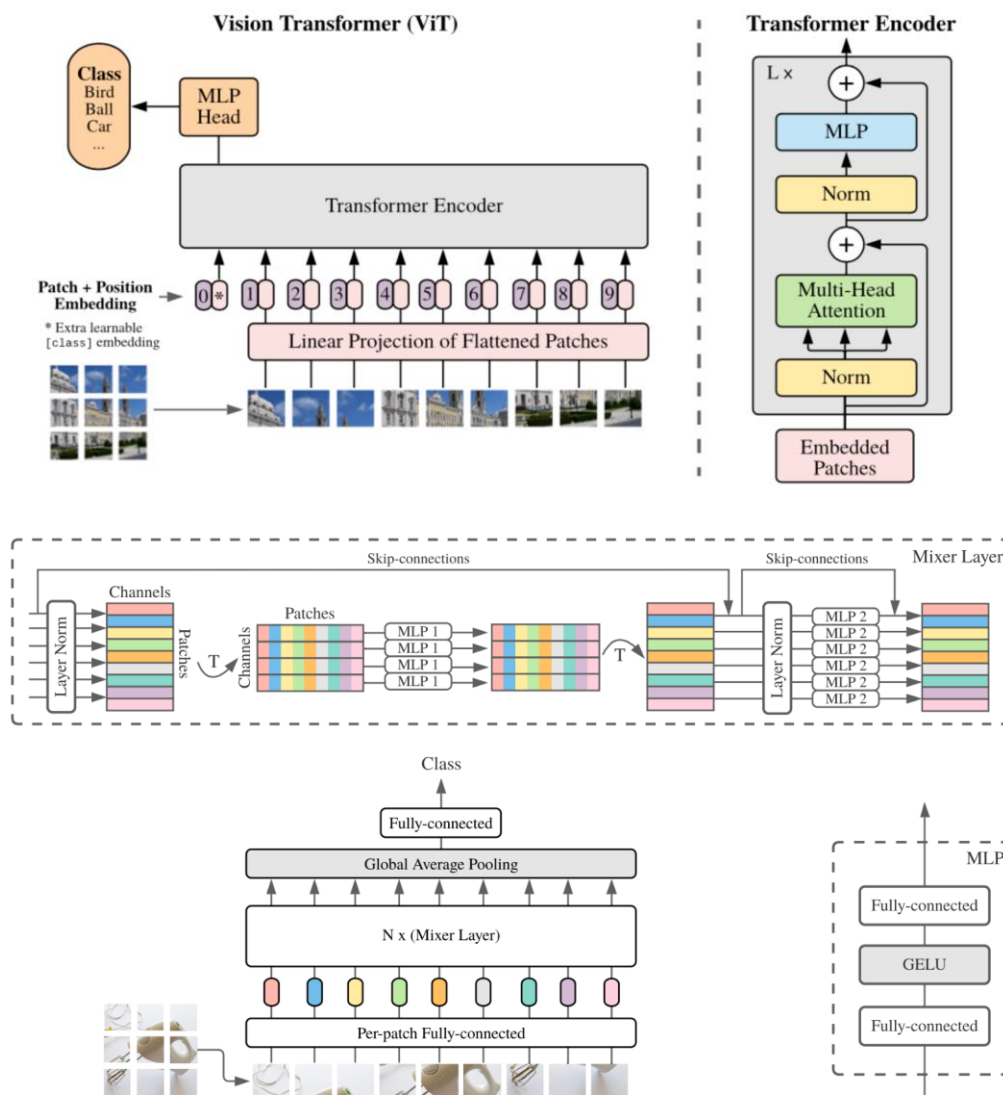
```

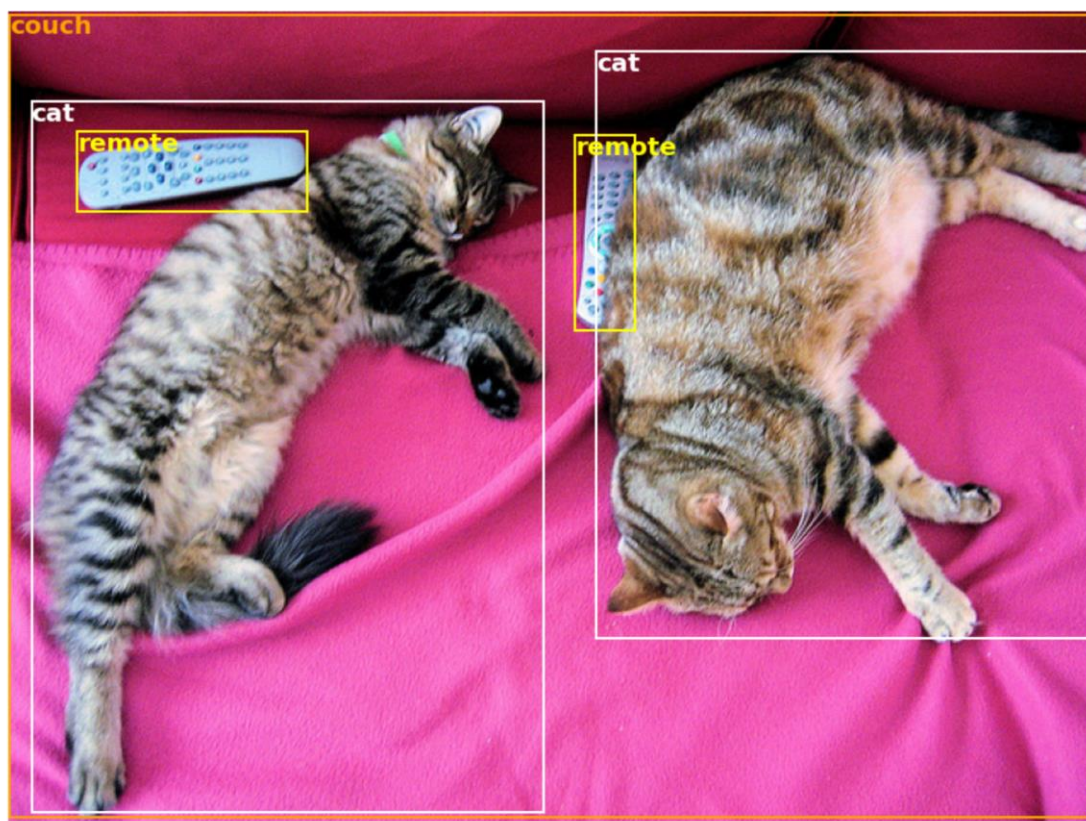
```

'License :: OSI Approved :: Apache Software License',
'Programming Language :: Python :: 3.7',
'Topic :: Scientific/Engineering :: Artificial Intelligence',
],
keywords="",
author='Vision Transformer Authors',
author_email='no-reply@google.com',
url='https://github.com/google-research/vision_transformer',
packages=find_packages(),
zip_safe=False,
install_requires=install_requires,
tests_require=tests_require,
extras_require=dict(test=tests_require),
)

```

результат работы:





вынести вердикт:

В статье подробно описаны конфигурации сетей, процедуры обучения, протоколы тестирования и детали реализации, что дает достаточно информации для воспроизведения экспериментов другими исследователями. Использование общедоступных наборов данных (например, ImageNet) и общедоступных весов моделей еще больше повышает воспроизводимость экспериментов.

Экспериментальные результаты в статье согласуются с теоретическим анализом. Авторы демонстрируют рост производительности сети при постепенном увеличении ее глубины, что соответствует теоретическим ожиданиям относительно влияния глубины сети на производительность в глубоком обучении.

Прозрачность деталей эксперимента:

Авторы подробно описывают в статье экспериментальные установки и конфигурации гиперпараметров, включая архитектуру сети, стратегии обучения и методы тестирования, что помогает другим исследователям понять каждый шаг эксперимента и проверить результаты.

Надежность экспериментальных результатов:

Эксперименты в статье оценивались на разных наборах данных и при разных условиях тестирования, что свидетельствует о робастности модели. Это говорит о том, что экспериментальные результаты не являются продуктом случайности в конкретных условиях, а поддаются обобщению.

Потенциальные возможности для улучшения:

Хотя экспериментальные результаты, приведенные в статье, согласуются с теоретическим

анализом, я обнаружил, что все еще есть возможности для улучшения. Например, более высокую производительность можно получить путем дальнейшей настройки гиперпараметров, изучения различных сетевых архитектур или использования дополнительных методов улучшения данных.