

ЗАДАНИЕ №4

Тема: Сравнение предобученных сверточных архитектур CNN

Цель работы — познакомиться с различными архитектурами свёрточных нейросетей, понять их отличия, сильные и слабые стороны, и научиться применять предобученные модели на практике.

Описание задания

В этом задании вы **не будете обучать модели**, а будете использовать предобученные на ImageNet архитектуры из `torchvision.models`. Вам предстоит сравнить **ResNet-18**, **MobileNetV2** и **GoogLeNet** по качеству предсказаний и скорости на валидационных выборках.

Что нужно сделать:

1. Загрузите предобученные модели:

- `ResNet-18`
- `MobileNetV2`
- `GoogLeNet`

2. Подготовьте валидационный датасет (на выбор):

- CIFAR-10 (по умолчанию),
- или SVHN / STL10 / Caltech101.

3. Примените модели к данным:

- выполните инференс,
- измерьте **точность** (accuracy),
- измерьте **время инференса** (по желанию).

4. Сравните архитектуры:

- по точности,
- по скорости,
- по особенностям архитектурного устройства.

5. Ответьте письменно на вопросы.

Формат сдачи:

- Один `ipynb`-файл с кодом и вашими текстовыми ответами внизу ноутбука.
 - Файл должен запускаться и проверяться автономно.
-

Вопросы для анализа:

1. Какая модель показала наилучшую точность на выбранной выборке? Почему?
2. Как влияет структура архитектуры (глубина, тип свёрток, количество параметров) на точность?

Пример кода:

```
# Импорты
import torch
import torchvision.transforms as transforms
import torchvision.models as models
import torchvision.datasets as datasets
import matplotlib.pyplot as plt
from torch.utils.data import DataLoader
import time

# Загрузка предобученных моделей
model_resnet = models.resnet18(pretrained=True).eval()
model_mobilenet = models.mobilenet_v2(pretrained=True).eval()
model_googlenet = models.googlenet(pretrained=True).eval()

models_dict = {
    'ResNet-18': model_resnet,
    'MobileNetV2': model_mobilenet,
    'GoogLeNet': model_googlenet
}

# Предобработка изображений
transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406],
                          std=[0.229, 0.224, 0.225])
])

# ◇ Валидационная выборка (пример: CIFAR10)
val_dataset = datasets.CIFAR10(root="./data", train=False, download=True,
                                transform=transform)
val_loader = DataLoader(val_dataset, batch_size=32, shuffle=False)
```

Подсказка: Допиши функцию подсчёта ассурасы и времени инференса для каждой модели и выведи результаты сравнения.

Срок сдачи: 8 апреля. Удачи!