Московский государственный технический университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»

Отчет
Лабораторная работа № 2
По курсу «Проектирование интеллектуальных систем»

Вариант 9

ИСПОЛНИТЕЛЬ:
Попов Илья Андреевич
Группа ИУ5-23М

_____

"__"_____2022 г.

ПРЕПОДАВАТЕЛЬ:
Канев А.И.

_____

"__"_____2022 г.

Москва 2022

Задание

По заданию выбрать свои классы и обучить сверточную нейронную сеть до 70% точности на тестовой выборке используя GPU. Провести три обучения для 3 разных тактик пуллинга: пуллинг с помощью шага свёртки stride, макс пуллинг, усредняющий пуллинг. Сравнить достигнутое качество, время обучения и степень переобучения. Выбрать лучшую конфигурацию. Сохранить модель. Перезапустить среду выполнения - теряются все текующие переменные.

Загрузить в colab готовую уже обученную на cifar100 модель. Преобразовать в onnx и сохранить локально.

Скачать каталог с html-файлом и встроить в него два файла моделей - обученную на ЛР1 и на ЛР2.

Скачать картинки из интернета согласно варианту и открыть их в html по кнопке. Автоматически в скрипте масштабируется изображение.

Выбрать в js нужные классы для готовой модели. Проверить на устойчивость обе модели, полносвязную и свёрточную, двигая картинку, убедиться в наличии свойства инвариантности сверточного слоя.

Варианты для CIFAR100

1. 23
2. 9
3. 39

Выполнение

```
!nvidia-smi
```

```
Fri Apr  1 07:45:29 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla K80           Off  | 00000000:00:04.0 Off |                    0 |
| N/A   33C    P8    25W / 149W |      0MiB / 11441MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Будем использовать GPU для обучения нейросети

Загружаем набор данных CIFAR100

```
class Cifar100_MLP(nn.Module):
    def __init__(self, hidden_size=32, classes=100):
        super(Cifar100_MLP, self).__init__()
        # https://blog.jovian.ai/image-classification-of-cifar100-dataset-using-pytorch-8b7145242df1
        self.seq = nn.Sequential(
            Normalize([0.5074,0.4867,0.4411],[0.2011,0.1987,0.2025]),
            # первый способ уменьшения размерности картинки - через stride
            nn.Conv2d(3, HIDDEN_SIZE, 5, stride=4, padding=2),
            nn.ReLU(),
            # второй способ уменьшения размерности картинки - через слой пуллинг
            nn.Conv2d(HIDDEN_SIZE, HIDDEN_SIZE*2, 5, stride=3, padding=0),
            nn.ReLU(),
            #nn.AvgPool2d(4),
            #nn.MaxPool2d(4),
            nn.Flatten(),
            nn.Linear(HIDDEN_SIZE*8, classes),
        )

    def forward(self, input):
        return self.seq(input)

HIDDEN_SIZE = 32
model = Cifar100_MLP(hidden_size=HIDDEN_SIZE, classes=len(CLASSES))
# NEW
model.to(device)
print(model(torch.rand(1, 32, 32, 3).to(device)))
summary(model, input_size=(32, 32, 3))
model
```

```
tensor([[0.0258, 0.1505, 0.0682]], device='cuda:0', grad_fn=<AddmmBackward0>)
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
         Normalize-1           [-1, 3, 32, 32]               0
            Conv2d-2            [-1, 32, 8, 8]           2,432
              ReLU-3            [-1, 32, 8, 8]               0
            Conv2d-4            [-1, 64, 2, 2]          51,264
              ReLU-5            [-1, 64, 2, 2]               0
           Flatten-6                 [-1, 256]               0
            Linear-7                   [-1, 3]             771
```

Создаём модель нейронной сети и проводим её обучение

Структура сети:

- Входной слой: 32*32*3 нейронов
- Первый скрытый свёрточный слой: 64 нейрона, 32 канала
- Второй скрытый свёрточный слой: 4 нейрона 64 канала
- Третий скрытый полносвязный слой: 256 нейрона
- Выходной слой: 3 нейрона

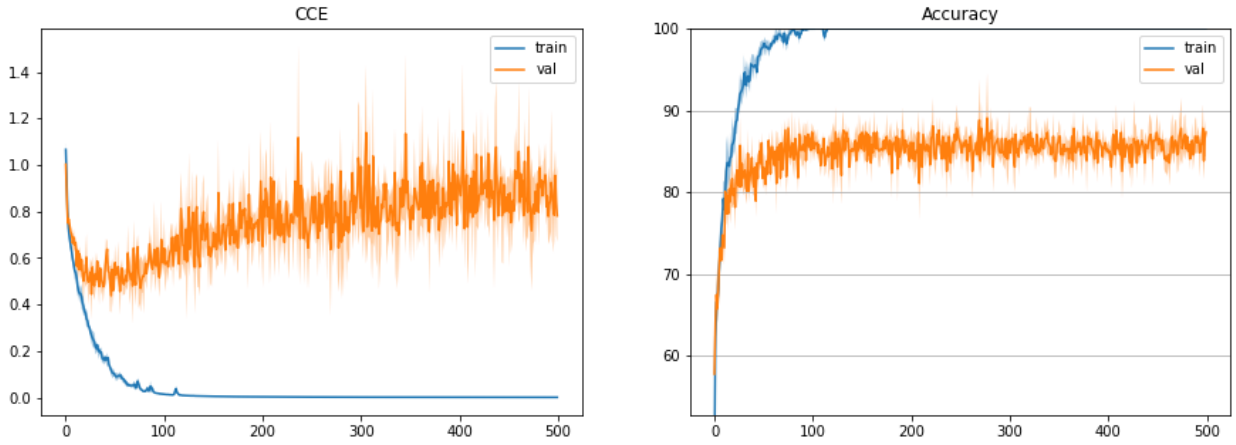
64*4=256 нейронов

64*(32*25+1) = 51264 Параметров

204800 связей

```
    passed += pbar.format_dict['elapsed']
    pbar = tqdm(total=EPOCHS*steps_per_epoch, miniters=5)
    pbar.update((epoch+1)*steps_per_epoch)
    x_vals = np.arange(epoch+1)
    _, ax = plt.subplots(1, 2, figsize=(15, 5))
    stats = np.array(losses)
    stats_val = np.array(losses_val)
    ax[1].set_ylim(stats_val[:, 0, 1].min()-5, 100)
    ax[1].grid(axis='y')
    for i, title in enumerate(['CCE', 'Accuracy']):
        ax[i].plot(x_vals, stats[:, 0, i], label='train')
        ax[i].fill_between(x_vals, stats[:, 1, i],
                           stats[:, 2, i], alpha=0.4)
        ax[i].plot(x_vals, stats_val[:, 0, i], label='val')
        ax[i].fill_between(x_vals,
                           stats_val[:, 1, i],
                           stats_val[:, 2, i], alpha=0.4)
        ax[i].legend()
        ax[i].set_title(title)
    plt.show()
print('Обучение закончено за %s секунд' % passed)
```

```
100% 6000/6000 [00:18<00:00, 324.31it/s]
```




```
Обучение закончено за 53.44697070121765 секунд
```

Обучение

```
for part in ['train', 'test']:
    y_pred = []
    y_true = []
    with torch.no_grad(): # отключение автоматического дифференцирования
        for i, data in enumerate(dataloader[part], 0):
            inputs, labels = data
             # на GPU
            inputs, labels = inputs.to(device), labels.to(device)

            outputs = model(inputs).detach().cpu().numpy()
            y_pred.append(outputs)
            y_true.append(labels.cpu().numpy())
        y_true = np.concatenate(y_true)
        y_pred = np.concatenate(y_pred)
        print(part)
        print(classification_report(y_true.argmax(axis=-1), y_pred.argmax(axis=-1),
                                    digits=4, target_names=list(map(str, CLASSES))))
        print('-'*50)
```

```
train
              precision    recall  f1-score   support

          23     1.0000    1.0000    1.0000       500
           9     1.0000    1.0000    1.0000       500
          39     1.0000    1.0000    1.0000       500

    accuracy                         1.0000      1500
   macro avg     1.0000    1.0000    1.0000      1500
weighted avg     1.0000    1.0000    1.0000      1500

--------------------------------------------------
test
              precision    recall  f1-score   support

          23     0.8558    0.8900    0.8725       100
           9     0.8750    0.8400    0.8571       100
          39     0.8400    0.8400    0.8400       100

    accuracy                         0.8567       300
   macro avg     0.8569    0.8567    0.8566       300
weighted avg     0.8569    0.8567    0.8566       300

--------------------------------------------------
```

Результат обучения

Всего было проведено обучение в 4 вариациях сети

№	Ядро (1; 2 св. слой)	Шаг	Паддинг	Пуллинг	t обучения	Точность
1	5; 3	4; 1	2; 1	avg(4)	48,6 с	86,33%
2	5; 5	4; 2	2; 0	-	52,8 с	84,33%
3	5; 3	4; 1	2; 1	max4	52 с	87,33%
4	5; 5	4; 3	2; 0	-	53,4 с	85,67%

Проверка модели на картинках из интернета

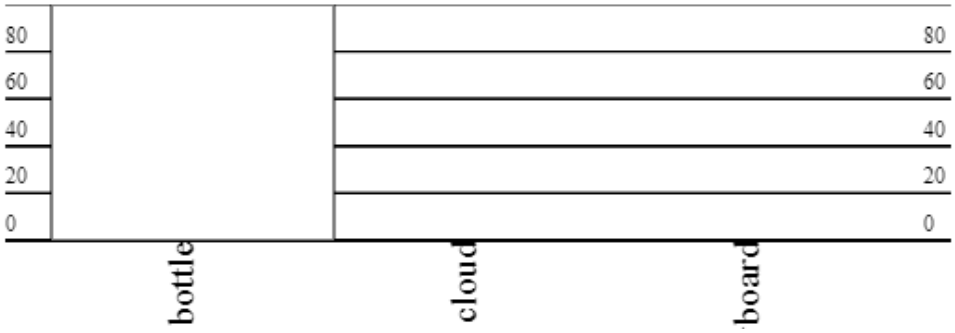
Выберите файл cifar100_...2_p0.onnx Select ONNX file

Выберите файл butilka-vin...0x800.jpg

Class label 39 Add 10 Random Undo Reset

9,23,39

← ↑ → ↓



Успешное распознавание даже с сильным сдвигом

9,23,39