Соревнование по классификации дорожных знаков

Public score: 0.97417

Private score: 0.97130

AI Masters
Сергей Руднев

Baseline

В качестве бейзлайна была использована нейронная сеть, состоящая следующих блоков

```
def forward(self, x):
    x = self.conv1(x)
    x = self.bn1(x)
    x = self.relu(x)
    x = self.conv2(x)
    x = self.bn2(x)
    x = self.relu(x)
    x = self.maxpool(x)
    return x
```

Также она включала в себя слой пулинга и линейный слой

```
cnn_baseline = nn.Sequential(
    CNNBlock(3, 32),
    CNNBlock(32, 64),
    CNNBlock(64, 128),
    CNNBlock(128, 256),
    CNNBlock(256, 512),

    nn.AdaptiveAvgPool2d((1, 1)),

    nn.Flatten(),
    nn.Linear(512, 67)
)
```

Аугментация

Для уменьшения переобучения и увеличения разнообразия обучающей выборки была использована аугментация из библиотеки albumentations, а также была немного увеличена сложность модели

```
aug = A.Compose([
    A.ShiftScaleRotate(),
    A.AdvancedBlur(),
    A.OpticalDistortion(),
    A.RandomBrightnessContrast(),
    A.RandomGamma(),
    A.Sharpen(),
    A.MotionBlur(),
    A.ChannelDropout(),
    A.Emboss(),
    A.Solarize()
])
```

Получилось добиться уменьшения переобучения и повысить скор на тестовой выборке

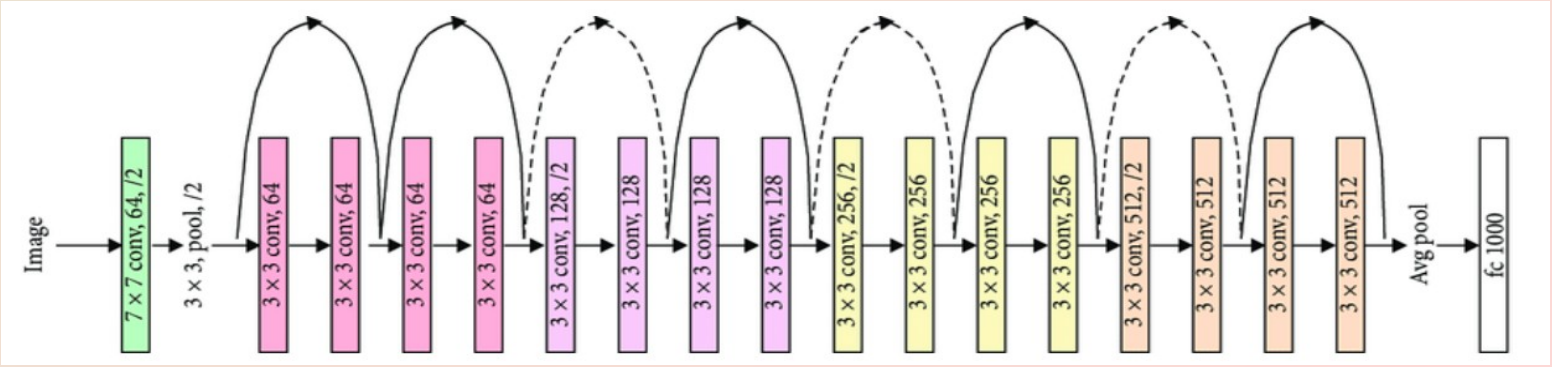
```
cnn_aug = nn.Sequential(
    CNNBlock(3, 64),
    CNNBlock(64, 128),
    CNNBlock(128, 256),
    CNNBlock(256, 512),
    CNNBlock(512, 1024),

    nn.AdaptiveAvgPool2d((1, 1)),
    nn.Flatten(),
    nn.Linear(1024, 67)
)
```

Предобученная модель

Затем была использована предобученная модель resnet18 из библиотеки torchvision

Были взяты два первых её слоя, а дальше использован нужный линейный слой.

Сначала небольшое количество эпох происходило обучение только последнего слоя, чтобы избавиться от шума и не испортить уже обученные веса resnet18. Качество модели оказалось даже хуже, чем у использованной ранее. Затем аналогичным образом была использована предобученная resnet152, которая показала небольшой прирост в качестве на тестовой выборке по сравнению с resnet18, но всё равно хуже, чем при использовании своей модели



LR scheduling

Также был использован подход адаптивного выбора lr.

Если лосс какое-то количество эпох не падает, то lr уменьшался, это помогало оптимизатору выйти с плато, в которое он попадал. Также был использован подход CosineAnnealingLR, при котором lr изменялся по определённому заданному закону.

Это помогло добиться прироста в качестве