

보고서: 코드설명

Line 별로 Models과 layers에 대해서 구조를 그림으로 그리고 설명할 것 • 학습 관련한 loss 및 optimizers의 파라미터와 선택한 이유를 설명할 것

모델 설명

아래는 PyTorch를 사용하여 수정된 ResNet-18 모델의 구조에 대한 설명과 함께 구조를 도식화한 내용입니다.

ResNet-18 모델 구조 설명

1. 모델 로드:

```
models.resnet18(pretrained=True)
```

- 사전 학습된 ResNet-18 모델을 로드합니다. 이 모델은 이미지넷 데이터셋을 사용하여 사전 학습되었으며, 다양한 이미지 관련 태스크에 적용될 수 있는 강력한 특성 추출 능력을 가지고 있습니다.

2. 첫 번째 컨볼루션 레이어 수정:

```
model.conv1 = nn.Conv2d(1, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
```

- 원래 ResNet-18 모델은 3 채널(RGB) 이미지를 입력으로 받습니다. 하지만 여기서는 1 채널 입력(흑백 이미지)을 받도록 첫 번째 컨볼루션 레이어를 수정했습니다. 이 레이어는 7x7 크기의 커널을 사용하여 입력 이미지에서 특성을 추출합니다.

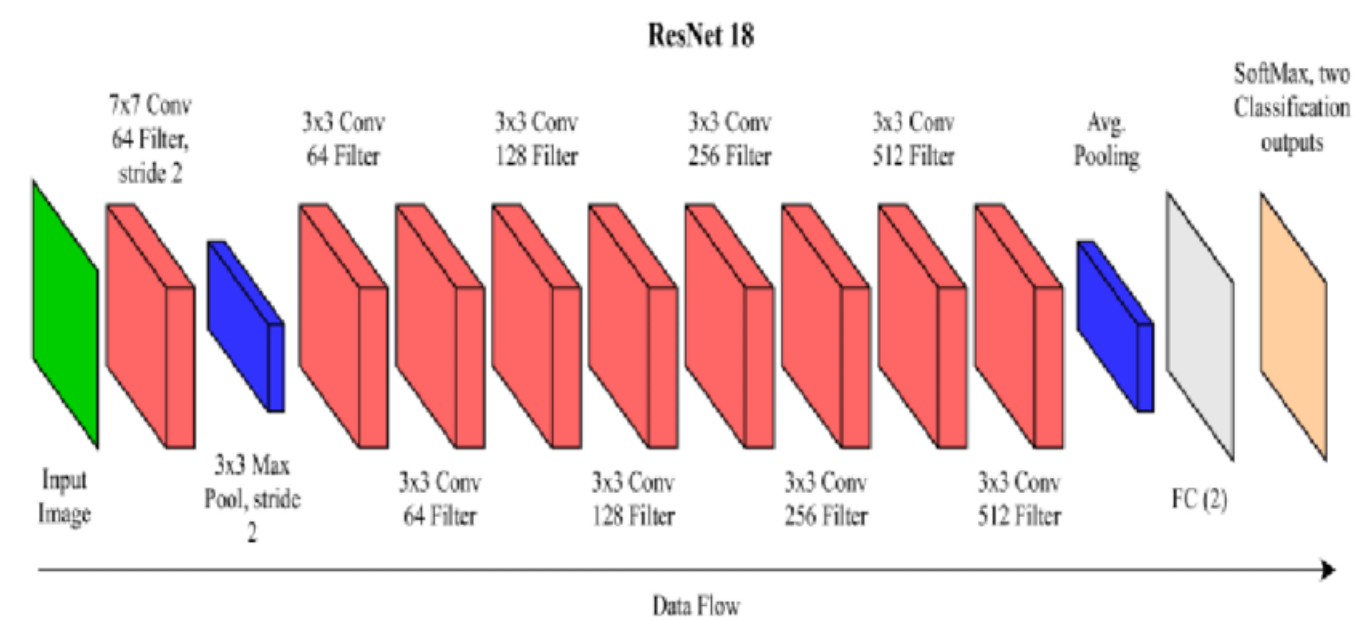
3. 최종 연결 레이어 수정:

```
model.fc = nn.Linear(model.fc.in_features, 26)
```

- ResNet-18의 마지막 완전 연결 레이어(fully connected layer)를 수정하여, 26개 클래스(알파벳 A-Z)에 대한 출력을 생성합니다. 이 변경은 모델의 출력 차원을 알파벳 분류 문제에 맞게 조정합니다.

이 구조는 기존 ResNet-18의 강력한 특성 추출 능력을 유지하면서, 손글씨 알파벳 인식과 같은 특정 문제에 맞게 조정되었습니다.

모델 다이어그램



학습 관련 Loss 및 Optimizers의 파라미터와 선정 이유

Loss Function

- **선택한 Loss Function:** `nn.CrossEntropyLoss`
- **선정 이유:** 이 Loss Function은 LogSoftmax와 NLLLoss를 결합한 것으로, 다중 클래스 분류 문제에 적합합니다. 각 클래스에 대한 모델의 확률 추정치를 계산하여, 실제 레이블과의 차이를 효율적으로 측정합니다. 또한, 이 함수는 수치적 안정성이 높고, 모델의 로짓(logit, 모델 출력의 원시 점수)을 직접 처리할 수 있어 사용하기 쉽습니다.

Optimizers

- **선택한 Optimizers:** `Adam`과 `SGD` (momentum 포함)
- **선정 이유:** - **Adam:** 적응형 학습률을 사용하여, 다양한 데이터와 네트워크 구조에 대해 좋은 성능을 보이는 것으로 알려져 있습니다. 특히, 빠른 수렴 속도와 효율적인 계산이 장점입니다.
- **SGD with Momentum:** 기본적인 SGD에 비해 모멘텀(momentum)이 추가되어, 학습 과정에서 방향성을 유지하며 더 빠르고 안정적으로 최적점에 도달합니다.