KYUNGPOOK NATIONAL UNIVERSITY

CIFAR-10 Classification

Using Resnet-18

전자전기공학부 석사과정 배찬호

01 개선한 사항

- model_num = 5 (ensemble 모델 5개)
- Batch_size 변경(128 -> 512), multi-gpu 사용
- Early Stopping 적용(patience = 50)
- Image augmentation에 Random Rotation, Color Jitter적용
- 더 많은 학습데이터로 pretrained된 모델 사용

Pretrained Model

- Larger source data gives better accuracy.

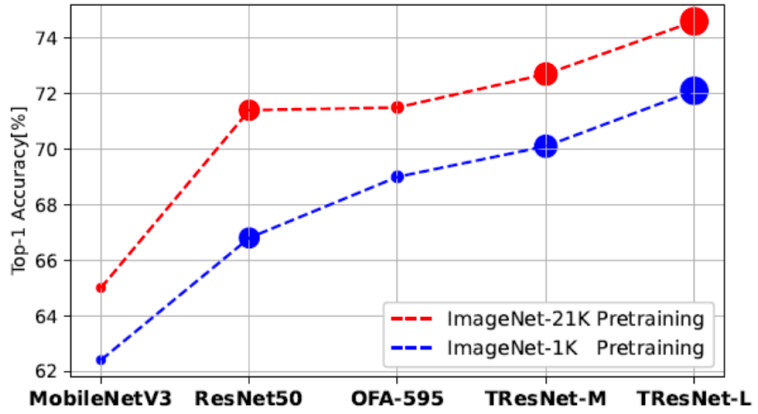

Figure 1. **Transfer learning results**: comparing transfer learning results, for different models and pretraining schemes, on iNaturalist dataset. Markers sizes are proportional to the models' memory footprint.

Recipes.pdf, p.10

- 같은 Pretrained 된 모델이라도
  어떤 데이터셋으로 학습한지에 따라
  Transfer-learning에서의
  성능이 달라짐을 알 수 있다.

02 Pretrained Model

Model card for resnet18.fb_swsl_ig1b_ft_in1k(Hugging Face)

Pretrained on Instagram-1B hashtags dataset using semi-weakly supervised learning and fine-tuned on ImageNet-1k by paper authors.

- Instagram-1B으로 사전학습을 한 뒤, 이를 다시 ImageNet-1K로 Fine-tuned한 pretrained model
- Instagram-1B hashtags 데이터셋은 수백만 개의 해시태그로 구성되어 있으며, 다양한 주제와 관련된 태그들이 포함되어 있는 데이터셋
- ImageNet-1k는 1000개의 다양한 클래스로 구성된 대규모 이미지 데이터셋
- 기존 pretrained 모델과 비교했을 때 CIFAR-10에서 약 2% 정도의 성능 향상

03 Image Augmentation

Baseline

```python
# Define the data transforms
transform_train = transforms.Compose([
    transforms.Resize(256),
    transforms.RandomCrop(224),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
])
```

Modified

```python
# Define the data transforms
transform_train = transforms.Compose([
    transforms.Resize(256),
    transforms.RandomCrop(224),
    transforms.RandomHorizontalFlip(),
    transforms.RandomRotation(10),  # 회전: 최대 10도 회전
    transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1),  #
    transforms.ToTensor(),
    transforms.Normalize((0.4914, 0.4822, 0.4465), (0.2023, 0.1994, 0.2010))
])
```

- Normalize 값 변경
- RandomRotation 적용

  지정한 범위 내에서 random하게 rotation 적용
- Color Jitter 적용

  위의 코드에서 brightness, contrast, saturation, hue는 각각 밝기, 대비, 채도, 색상

  밝기, 대비, 채도, 색상을 지정한 범위 내에서 random하게 변경

04 Early Stopping

```python
# Train the model
for epoch in range(total_epoch):
    train()
    test()
    scheduler.step()

print('Finished Training')

# Save the checkpoint of the last model
PATH = './resnet18_cifar10_%f_%d.pth' % (lr, seed_number)
torch.save(model.state_dict(), PATH)
```

```python
# Train the model
for epoch in range(total_epoch):
    train()
    val_acc = test()

    # Update the learning rate
    scheduler.step()

    if val_acc > max_val_acc:
        epochs_no_improve = 0
        max_val_acc = val_acc
        best_model = model.state_dict()
    else:
        epochs_no_improve += 1
        if epochs_no_improve == n_epochs_stop:
            print('Early stopping!')
            break

print('Finished Training')

# Save the checkpoint of the last model
PATH = './resnet18_cifar10_%f_%d.pth' % (lr, seed_number)
torch.save(model.state_dict(), PATH)
```

- Test_accuracy에서 50 epochs 동안 accuracy가 올라가지 않을 경우 가장 정확도가 높았던 모델을 저장하고 학습을 종료

05 Result

```
(yolov8) pi@pi-System-Product-Name:~$ /home/pi/anaconda3/envs/yolov8/bin/python /home/pi/chbae/cifar10/ensemble.py
cuda
Files already downloaded and verified
Accuracy of the ensemble on the 10000 test images: 97.610000 %
```

- 그 후 저장된 5개의 모델을 ensemble 코드를 통해 load 후 ensemble
- 최종 결과 : 97.61%