**Week3**

**2024\_spring\_vision\_tutorial**

**Goal : Wandb,DDP 구현 및 Timm을 활용해 Resnet50으로 82% 이상 성능 도달**

**[주요 실험 및 결과]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Default Setting** | **Experiment&Results1** | **Experiment&Results2** |
| **Batch\_size** | | 512 | 512 | 512 |
| **Optimizer** | Opt. | SGD | SGD | SGD |
| Momentum | 0.4 | 0.9 | 0.9 |
| Weight-decay | 1e-9 | 1e-3 | 1e-3 |
| **Scheduler** | Sched | CosineAnnealingLR | CosineAnnealingLR | CosineAnnealingLR |
| Learning rate | 0.1 | 0.25 | 0.25 |
| Warmup-lr | 0.01 | 0.01 | 0.01 |
| Min-lr | 1e-6 | 1e-6 | 1e-6 |
| Warmup-epochs | 5 | 5 | 5 |
| **Augmentation** | Hflip | 0.7 | 0.7 | 0.7 |
| Color-jitter | 0.2 | 0.2 | X |
| Auto Augment | None | None | original |
| Cut mix | 0 | 0 | 0 |
| Mix-up | 0 | 0 | 0 |
| **Best Accuracy** | | 70.72 | 78.19 | 80.31 |
| **Time** | | 48m (epoch:200) | 49m 45s (epoch:200) | 47m 25s (epoch:200) |
| **Loss History plot** | |  |  |  |
| **Analyze** | | Learning rate, weight decay, Momemtum값이 작아서 local minimum에 빠져 특정 정확도 값에 수렴하는 것으로 의심   * Learning rate,Momemtum, weight\_deacy 값을 키워보자 | 정확도가 오르긴 했지만, train\_loss는 epoch이 진행됨에 따라 계속 감소하는 것과는 반대로 eval\_loss는 일정 epoch이상에서는 더 이상 감소를 보이지 않았다.  -> Overfitting이 의심되어, 데이터 셋에 적절한 Augment를 찾아 적용해 데이터의 복잡도를 늘려보자 | 진한 초록색 부분  이전보다 train\_loss의 감소에 따라 eval\_loss가 감소하는 것을 볼 수 있었지만, 여진히 Overfitting이 의심되어, Cutmix를 통해 데이터 복잡도를 늘려보고자 함 |

**[사용한 기법]**

* **Momemtum: 이전 단계에서의 파라미터 업데이트 방향으로 힘을 받아 현재 단계에서 더 나은 방향으로 파라미터를 조정 (수렴 속도 향상 및 이상치에 대한 저항성)**
* **Weight\_decay: 학습과정에서 모델의 파라미터에 대한 규제를 가하는 방법. 파라미터에 대한 규제를 가하면 모델이 좀 더 일반화 되어 훈련 데이터에 대해 과적합을 예방하는데 도움이 된다.**
* **Color Jitter: Data augmentation의 기법 중 하나로, 이미지의 Lightness, Hue 그리고 saturation등이 임의로 변하는 것을 의미**
* **AutoAugment: 모델의 일반화 성능을 향상시키기 위해 데이터셋에 자동으로 증강을 적용시키는 알고리즘. 다양한 변형을 진행하면서 어떤 변형을 얼마나 적용할지를 강화학습을 통해 최적화 한다. 이번 실험에서는 timm의 autoaugment를 사용했기에, Policy가 ‘v0’와 ‘original’이 있었는데 Imagenet 데이터셋 기반 policy인 original을 적용해 실험을 진행하였다.(cifar10 또는 cifar100 policy를 찾아보았는데 timm에서는 지원해주지 않는 것 같았다..)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Experiment&Results3** | **Experiment&Results4** | **Experiment&Results5** |
| **Batch\_size** | | 512 | 512 | 512 |
| **Optimizer** | Opt. | SGD | SGD | SGD |
| Momentum | 0.9 | 0.9 | 0.9 |
| Weight-decay | 1e-3 | 1e-3 | 1e-3 |
| **Scheduler** | Sched | CosineAnnealingLR | CosineAnnealingLR | CosineAnnealingLR |
| Learning rate | 0.25 | 0.25 | 0.25 |
| Warmup-lr | 0.01 | 0.01 | 0.01 |
| Min-lr | 1e-6 | 1e-6 | 1e-6 |
| Warmup-epochs | 5 | 5 | 5 |
| **Augmentation** | Hflip | 0.7 | 0.7 | 0.7 |
| Color-jitter | X | X | X |
| Auto Augment | Original | original | Original |
| Cut mix | 1.0 | 1.0 | 0.5 |
| Mix-up | 0 | 0 | 0.5 |
| **Best Accuracy** | | 80.91 | 82.13 | 81.28 |
| **Time** | | 49m 31s (epoch: 200) | 1h 1m (epoch : 250) | 1h 1m (epoch: 250) |
| **Loss History plot** | |  |  |  |
| **Analyze** | | Cutmix를 1.0으로 설정하니 Train\_loss가 이전 실험과 다르게 값이 높아 학습자체를 어려워 하는 것 같았음. Eval\_loss를 확인했을 때 계속해서 감소하는 추세를 보여 epoch를 더 늘려서 특정 값에 수렴할 수 있도록 확인해보고자 함. | 갈색 부분  Epoch값을 증가해서 확인한 결과 eval\_loss값이 이전 실험값들에 비해 더 적은 값으로 수렴하는 것을 확인 | Mixup을 함께 적용해보았지만, Train과정에서 더욱 불안정해지는 모습과 이전 실험들과 eval\_loss를 비교했을 때 큰 차이가 없었다. 결과적으로 성능 향상에 있어 유의미한 결과를 보기는 어려웠다. |