CIFAR-10 Challenge

휴먼지능정보공학과

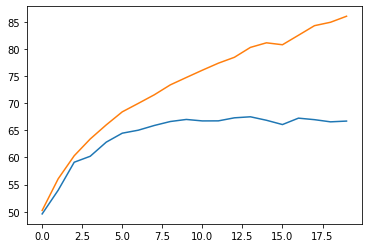
201710813 한현수

**Best Accuracy = 78.57572174072266**

**1. Experiments**

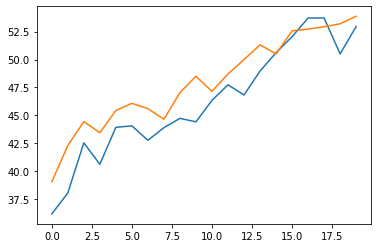
* Base Line

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| batch\_size | learning\_rate | num\_epoch | random\_seed | dropout | optimizer | train\_acc |
| 16 | 2.00E-03 | 20 | 42 | 0 | SGD | 86 |

  
먼저 베이스 라인의 실험 결과입니다. Dropout, l2 regularization, learning\_rate scheduling 없이, data augmentation은 random crop, random horizontal flip만 사용하여 진행한 결과입니다. 그래프의 개형을 보니 20epoch임에도 불구하고, overfitting현상이 일어남을 확인할 수 있었습니다.

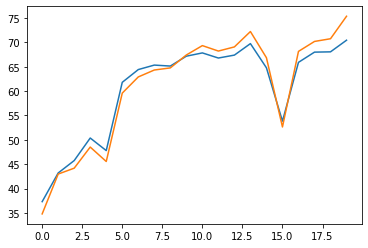
* Base Line + Dropout(0.5)

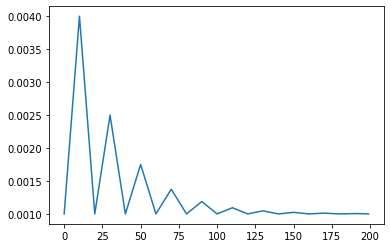
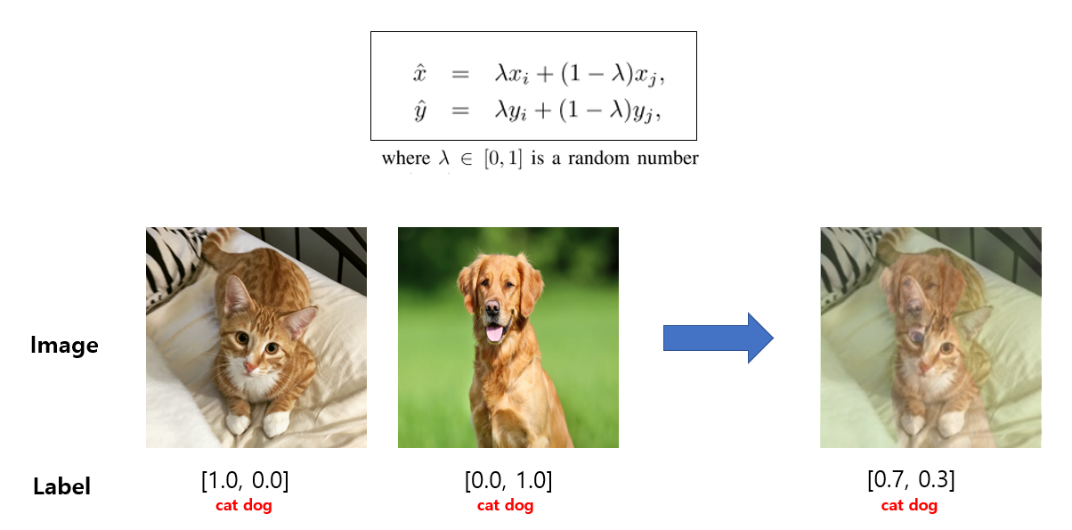
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| batch\_size | learning\_rate | num\_epoch | random\_seed | dropout | optimizer | train\_acc | test\_acc |
| 16 | 2.00E-03 | 20 | 42 | 0.5 | SGD | 53.9 | 52.9 |

  
 첫 실험에서 Dropout을 추가한 결과입니다. Dropout을 추가하니 overfitting이 일어나지 않음을 확인할 수 있었습니다. 하지만 epoch가 적어 underfitting인지는 확인할 수 없었습니다.

* Base Line + Dropout + Adam optimizer

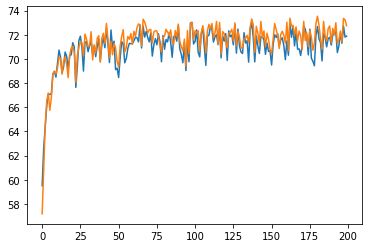
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| batch\_size | learning\_rate | num\_epoch | random\_seed | dropout | optimizer | train\_acc | test\_acc |
| 16 | 2.00E-03 | 20 | 42 | 0.5 | Adam | 75.4 | 70.5 |

  
이는 위의 실험에 SGD optimizer에서 Adam optimizer로 바꾼 결과입니다. 이를 보고 SGD optimizer에 비해 더 빠른 학습이 진행된다는 것을 확인하였습니다.

* Main Idea  
  제가 영감을 받은 Main idea는 아래의 논문에서 영감을 얻어 진행하였습니다.  
  He, Tong, et al. "Bag of tricks for image classification with convolutional neural networks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.
  + Learning Rate Scheduling  
      
    main idea에서 사용한 Learning Rate(lr) Scheduling입니다. 이는 200epoch동안의 lr으로 점점 0.001로 수렴하여 사용하였습니다. 그 이유는 참고한 논문에서의 추천하는 scheduling은 cosine annealing with warmup으로 초기 epoch은 점차 증가시키고 그 이후는 cosine 함수의 개형을 이용해 scheduling을 하는 것입니다. 이를 참고하여 유사하지만 pytorch에서 기본으로 제공하는 함수임 cyclicLR을 이용하여 위와 같은 learning rate scheduling을 하였습니다.
  + MixUp augmentation  
    MixUp이란 입력 이미지를 두개를 일정 비율()로 섞고 그에 따른 label 또한   
    a: (1-) 로 만들어 augmentation 하는 기법입니다. 이를 사용하면 아래와 같은 결과를 만들 수 있습니다.  
    

위의 두개의 main idea를 바탕으로 실험을 진행한 결과는 다음과 같습니다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Batch\_size | learning\_rate | num\_epoch | random\_seed | dropout | optimizer | L2 | MixUp\_Alpha | lr\_scheduler | train\_acc |
| 32 | 2.00E-03 | 200 | 42 | 0.5 | Adam | 1.00E-03 | 0.4 | cycliclr | 75.4 |

  
이를 보고 더 이상 train\_acc가 올라가지 않는다 판단하여 l2 regularization을 1e-4로 낮추고 200epoch을 2회 train을 한결과 accuracy = 78.58이라는 결과를 얻었습니다.

추가적으로 실험했던 것들: mixup augmentation을 배치마다 확률로 사용하자.(MixUp\_choice)  
batch\_size를 최대한 키워보자. -> 이는 배치를 키우니 training time은 확연히 줄지만 overfitting현상이 빨리 일어남을 확인할 수 있었습니다.