

# Assignment 3 보고서

2017-11621 전기정보공학부 배병욱

이번 inception module을 사용한 CNN에서 저는 Google-Net을 참고하여 만들었습니다. 조정한 요소로는 batch size, optimizer, Inception layer, learning rate 등이 있습니다.

## 1. Batch size

Skeleton으로 주어진 batch size는 epoch이 2일 경우에는 조금 부족할 것이라고 예상했습니다. 그렇기에 처음에는 batch size를 10, 20, 등으로 증가시켜갔으며 batch size가 50일 경우에 mini batch로 사용하기에 적당하다고 생각했습니다.

## 2. Optimizer

원래의 SGD가 아닌 RMSProp, Adam, Adagrad 등의 다양한 optimizer를 사용해 봤지만 이렇다 할 성능의 변화는 나오지 않았습니다. 이는 적은 epoch으로 인한 것이라 생각하여 optimizer는 SGD로 유지시키기로 하였습니다. 하지만 SGD만 사용할 경우에 학습속도가 초반에 너무 느리거나 후반에 loss가 증가하는 경향을 보이기에 원래의 momentum에서 Nestreov momentum을 사용하였으며, weight decay도 사용하기로 하였습니다.

## 3. Inception layer

Google-Net을 참고하여 kernel을 제 임의대로 만들어 사용하여 봤지만 성능이 60%대에서 잘 증가하지 못하는 경향을 보였습니다. 가장 먼저 생각한 것은 2 epoch만 학습을 할 때에는 drop out은 오히려 역효과를 보일 것이라 생각하여 삭제하였습니다. FC layer에서 3중 FC layer를 사용하며 ReLU를 사용하거나, CNN에서 1번과 같이 ReLU를 사용하고, kernel size를 다양하게 바꿔가며 시도해 보았지만 이렇다 할 성과는 얻지 못했기에 해당 코드들은 주석처리 하였고, 논문의 Google-Net을 토대로 kernel size를 최대한 유사하게 바꿔졌습니다.

## 4. Learning rate

교수님께서 수업시간에 가장 강조한 hyper parameter이기에, 최대한 다양한 learning rate를 사용해봤습니다. Learning rate에 따라서 2.에서 언급한 것 과 같이 weight decay, momentum 값을 바꿔가며 최대한 learning rate들을 사용해 봤습니다.

코드를 바꿨을 때에 따른 accruacy를 작성한 메모들의 요약본을 밑의 부록에 첨부하였습니다.

## 부록

1. RMSProp (lr 0.001, momentum 0.9) Linear 3층 (w. Relu) Batch = 10 26%	2. SGD (lr 0.001, momentum 0.9) Linear 3층 (w. Relu) Batch = 10 65%	3. Adam(lr 0.001) Linear 3층 (w. Relu) Batch = 10 47%
4. SGD (lr 0.001, momentum 0.9) Linear 3층 (w. Relu) Batch = 10 First layer w. ReLU fn 61%	5. SGD (lr 0.001, momentum 0.9) Linear 3층 (w. Relu) Batch = 10 w/o dropout 63%	8. SGD (lr 0.001, momentum 0.9) Linear 3층 (w. Relu) Batch = 10 First layer w. ReLU fn w/o dropout 64%
9. SGD (lr 0.002, momentum 0.9) Linear 3층 (w. Relu) Batch = 10 First layer w.o ReLU fn w/o dropout 64%	10. SGD (lr 0.001, momentum 0.9) Linear 3층 (w. Relu) Batch = 5 First layer w.o ReLU fn w/o dropout 60%	12. SGD (lr 0.001, momentum 0.9) Linear 3층 (w. Relu) Batch = 20 First layer w.o ReLU fn w/o dropout 68%
13. SGD (lr 0.001, momentum 0.9) Linear 3층 (w. Relu) Batch = 100 First layer w.o ReLU fn w/o dropout 60%	15. SGD (lr 0.001, momentum 0.9) Linear 2층 (w.o Relu) Batch = 20 First layer w.o ReLU fn w dropout 69%	17. SGD (lr 0.001, momentum 0.9) Linear 2층 (w.o Relu) One less inception for lLayer_2 68%
18. SGD (lr 0.001, momentum 0.9) Linear 2층 (w.o Relu) Change of the inception layer filter sizes 71%	26. SGD (lr 0.0095, momentum 0.9, Nestreov true, weight decay 0.000001) Linear 2층 (wo Relu) Change of the inception layer filter Batch 20 70%	28. SGD (lr 0.001, momentum 0.9, Nestreov true, weight decay 0.0001) Linear 2층 (wo Relu) Change of the inception layer filter Batch 50 68%
37. SGD (lr 0.00153, momentum 0.9, Nestreov true, weight decay 0.0001) Linear 2층 (wo Relu) Change of the inception layer filter Batch 50 71%	42. change to Google Net SGD (lr 0.00153, momentum 0.9, Nestreov true, weight decay 0.0001) Linear 2층 (wo Relu) Batch 50 69%	46. SGD (lr 0.003, momentum 0.9, Nestreov true, weight decay 0.0001) Linear 2층 (wo Relu) Batch 50 70%
51. SGD (lr 0.0055, momentum 0.9, Nestreov true, weight decay 0.001) Linear 2층 (wo Relu) Batch 50 71%	53. SGD (lr 0.007, momentum 0.9, Nestreov true, weight decay 0.001) Linear 2층 (wo Relu) Batch 50 70%	54. SGD (lr 0.0069, momentum 0.9, Nestreov true, weight decay 0.001) Linear 2층 (wo Relu) Change of the inception layer filter Batch 50 70%
57. SGD (lr 0.0065, momentum 0.9, Nestreov true, weight decay 0.001) Linear 2층 (wo Relu) Change of the inception layer filter Batch 50 71%	71. SGD (lr 0.0055, momentum 0.9, Nestreov true, weight decay 0.001) Linear 2층 (wo Relu) Change of the inception layer filter Batch 50 71%	72. SGD (lr 0.0050, momentum 0.9, Nestreov true, weight decay 0.001) Linear 2층 (wo Relu) Change of the inception layer filter Batch 50 72%

## 참고문헌

[1] Szegedy et al., "Going deeper with convolutions", CVPR 2015.