



Deep Learning for Image Classification


Team 2

고준영 이우윤 이혜원

Assignment Goal



1) CIFAR-10 data set을 이용해 image classification을 위한 CNN 모델을 개발

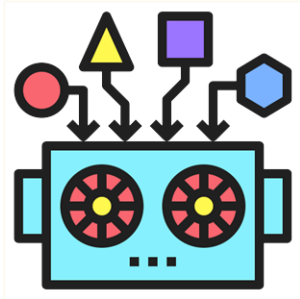


2) Intermediate feature space와 class activation map의 가시화



3) 최종 결과를 분석

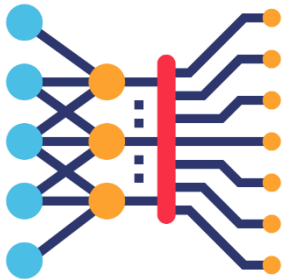
Introduction



**Deep
Learning**

: 문제에 대해 스스로 학습하고 답을 도출해내는 고차원의 머신러닝기법

: 인공 신경망은 인간의 뇌가 동작하는 것과 유사하게
데이터를 통해 패턴을 인식하고 데이터를 분류하며, 향후 사건을 예측하
도록 훈련시킬 수 있다



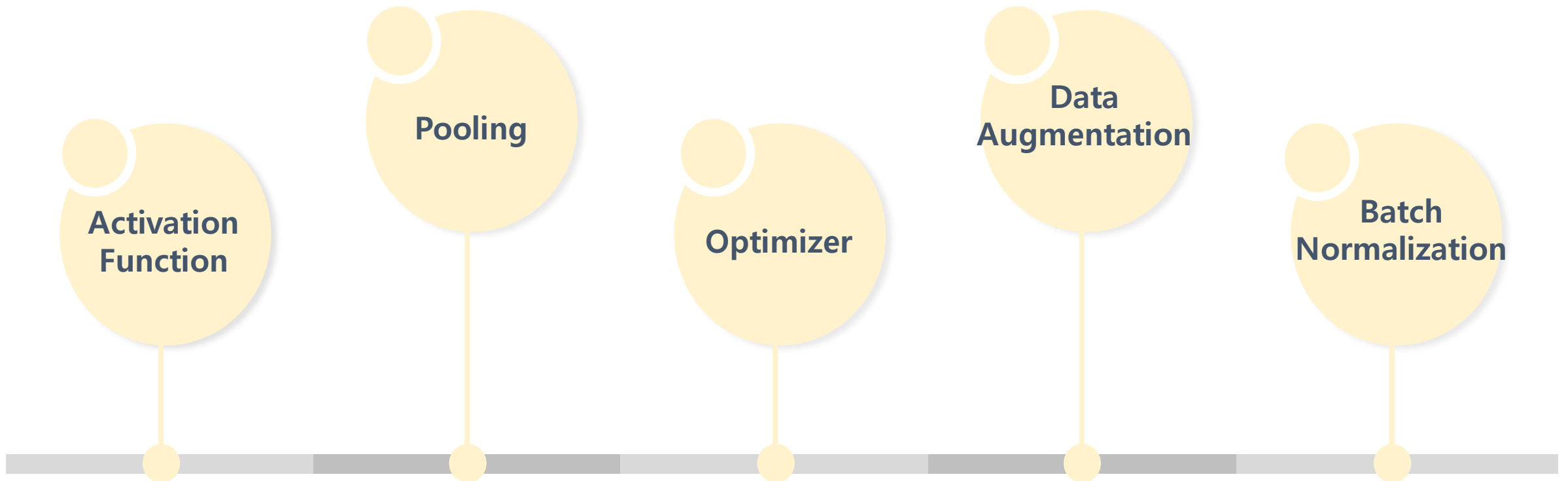
CNN

: Convolutional Neural Network

입력계층 -> 은닉계층 * n -> 출력계층

convolutional 계층, pooling 계층 등의 여러 계층을 거쳐
도출된 Feature Map 을 학습하여 합성하는 네트워크
Image Classification 에 유용하게 이용된다.

CNN Model Design



CNN 모델 디자인

Activation Function

Pooling

Optimizer

Data Augmentation

Batch Normalization

Activation Function

filters를 통해 feature map이 추출되면, CNN에서는 이 특징 맵에 활성화 함수를 적용하여 값을 활성화 시켜준다

Relu

네트워크가 효율적으로 구성된다는 장점

깊은 레이어에서도 효율적으로 작동한다는 장점

+ **Softmax**

Pooling

추출된 Activation map을 resizing 하여 새로운 layer를 생성하는 작업 :
각 커널 별 최대값 추출하는 Max Pooling,
평균값을 추출하는 Mean Pooling,
Average Pooling 등

Max Pooling

입력이 대폭 다운샘플링되어 계산 효율성을 향상시킨다는 장점

각 커널의 주요 가중치 갖는 value를 보존할 수 있다는 장점

최근 CNN에서 주로 사용한다는 점

CNN 모델 디자인

Activation Function

Pooling

Optimizer

Data Augmentation

Batch Normalization

+) Loss Function

Optimizer

: 학습 속도를 안정적이고 빠르게 하며, 오차 범위를 줄이기 위해 사용
: Momentum, RMSProp, Adam 등 다양한 종류



Adam

: 경사 하강법 + RMSProp

cnn에서 흔히 사용하는 최적화 함수이기도 하고, 간단하면서도 효율적이라는 장점이 있어서 선택함

Loss Function

: Loss (출력값과 정답간의 오차)를 계산하는 함수
: MSE, Cross Entropy Error 등



Cross Entropy

일반적으로 다중 클래스 분류 문제에서 교차 엔트로피를 많이 사용하기에,
우리 연구에서도 해당 loss function 을 채택함.

CNN 모델 디자인

Activation Function

Pooling

Optimizer

Data Augmentation

Batch Normalization

Data Augmentation : 수평/수직 이동, 수평/수직 flip, random 회전 확대, 임의 밝기 증가 등의 기법

본 연구 각도를 조정하여 이미지를 회전하는 방법과 수평/수직 이동, 수평/수직 flip을 사용

1) 임의의 초기값을 넣어 epoch를 5로 설정 2) 값을 바꿔가며 여러번 실험해 보면서 가장 적합한 augmentation 방법 도출

	Case 1	Case 2	Case 3
회전율	15도	15도	30도
Width 비율	0.1	0.3	0.3
Height 비율	0.1	0.3	0.3
Random Flip	수평 random Flip	수평 random Flip	수평 + 수직 random Flip
최고 정확도	0.7283	0.6608	0.5454

⇒ 데이터 보강을 가장 적게 적용했을 때
:정확도가 가장 높은 결과.

⇒ 가장 급격하게 적용했을 때
: 정확도가 제일 낮은 결과.

- 장점 : 일반화 기능을 향상시키고 overfitting 문제를 완화
 - 유의점 : 너무 과도하게 적용했을 경우에는 오히려 현실과 너무 동떨어지거나 기존 특징을 왜곡할 수 있음 학습 난이도와 성능에 좋지 않은 영향.
- ⇒ Data augmentation 기법이 무조건 좋은 성능만을 보여주는 건 아니고, 어떤 것을 분류하고 얼마나 데이터를 변형시키는 지에 따라 성능이 달라진다는 사실을 확인할 수 있음.

CNN 모델 디자인

Activation Function

Pooling

Optimizer

Data Augmentation

Batch Normalization

Batch Normalization: 학습에 불안정화가 일어나는 것을 방지하기 위해 평균을 빼서 분산으로 나누는 정규화를 진행
(활성화함수 전후단계 중 어디에 적용할 지에 대한 논의 중)

본 연구 -> 활성화 함수 이전에 적용 + epoch를 5로 설정하고 axis 값을 바꿔가면서 정확도를 비교

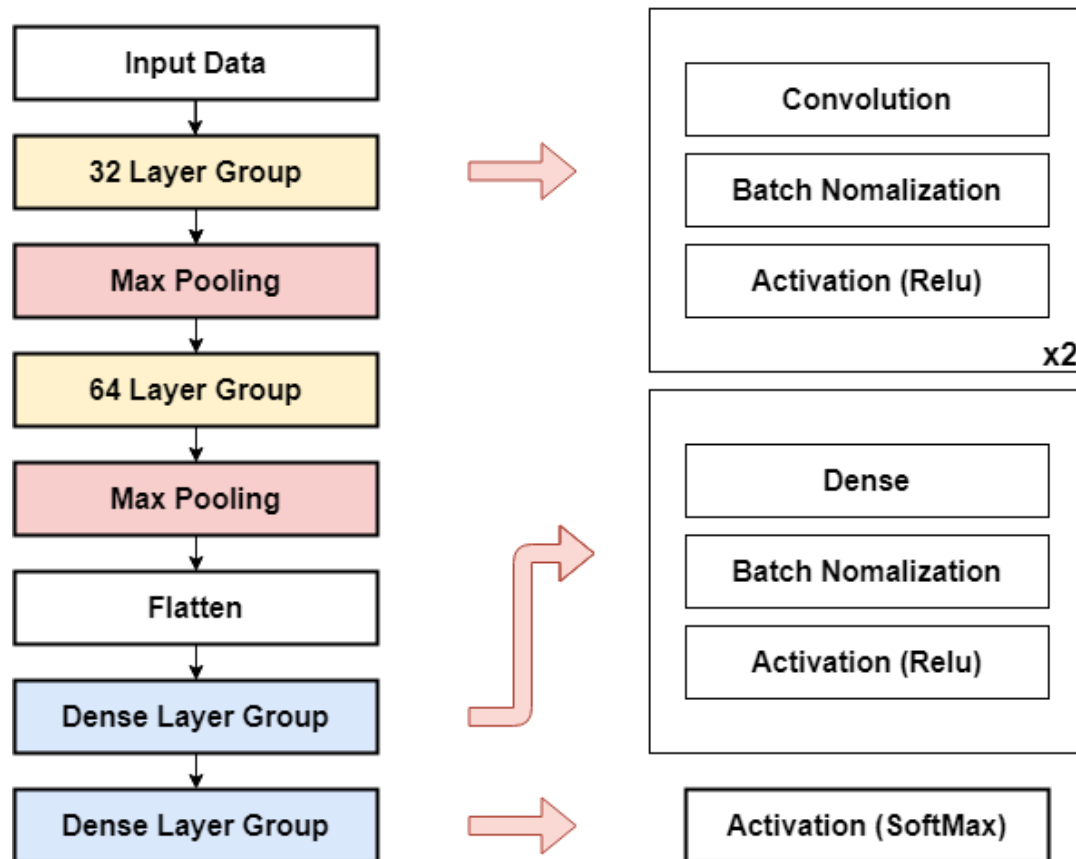
data augmentation 처럼 파라미터 값을 변경해 더 높은 성능을 보일 수 있는 방법 탐색 중,
배치 정규화의 수식에 쓰이는 momentum, epsilon, alpha, gamma 중에서
수학적으로 가장 쉽게 접근할 수 있다는 장점에 기반하여 정규화의 특성 축인 axis 변수를 조정하는 방법을 선택

[실험 결과]	Axis = -1 (dafault)	Axis = 1	Axis = 3
최고 정확도	0.7464	0.7182	0.7425

⇒ axis=-1과 axis = 3일 때 가장 정확도가 좋게 나온 것을 확인할 수 있음.

⇒ Axis -1 과 3의 정확도가 비슷했던 이유에 대한 논의
: Keras 라이브러리의 사용 vs Theano 라이브러리

CNN 모델 중간 도출



기타 실험 변수

Epoch

Batch Size

Layers

Epoch

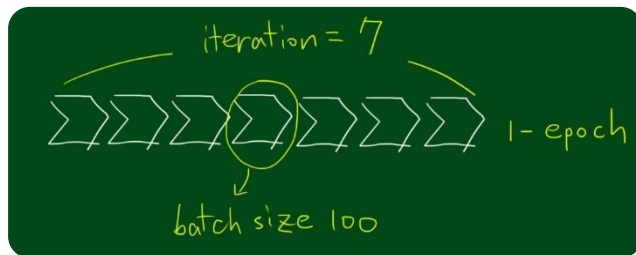
전체 트레이닝 셋이
신경망을 통과한 횟수

Batch Size

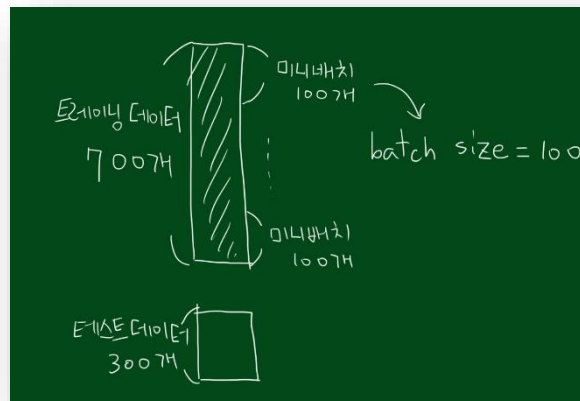
전체 트레이닝 데이터 셋을
여러 작은 그룹으로 나눌 때,
해당 소그룹의 속하는 데이터 수

Layers

⇒ 레이어 수에
변화를 주며
추가 실험 진행



94.4% 2.56 100



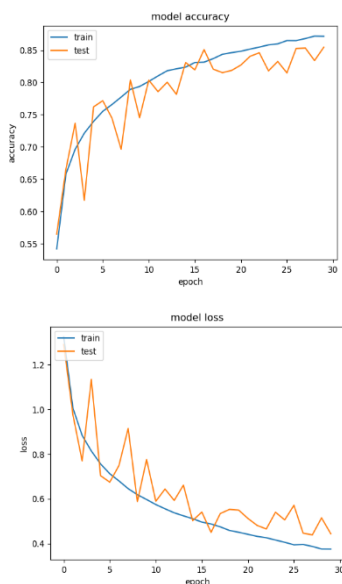
기타 실험 변수 | 실험 내용

Epoch

Batch Size

Layers

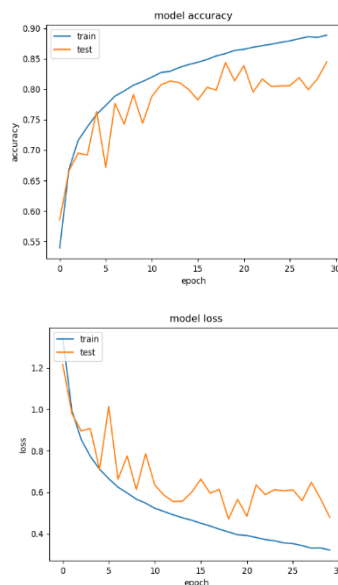
실험1) Epoch 30, Batch Size 32



test accuracy : 16 epoch(최고점)
이후, 감소와 증가를 반복

==
? Train model의 과 적합
~> Test model의 일반화를 저해?
==

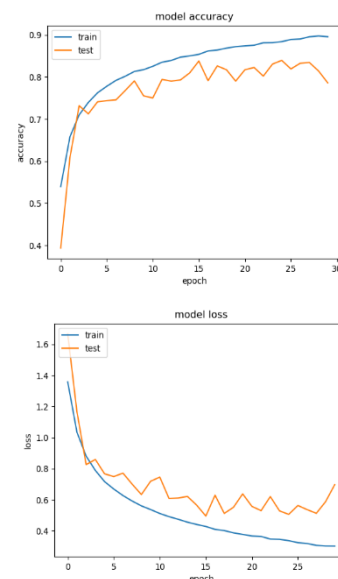
실험2) Epoch 30, Batch Size 50



Train과 Test 선의 간격이 커진
것을 확인

검증 데이터의 손실이 높고, 정
확도가 낮아짐

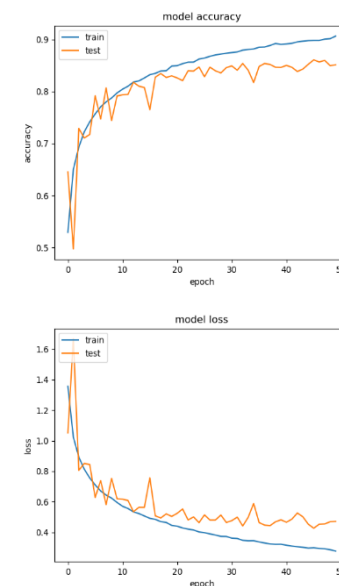
실험3) Epoch 30, Batch Size 128



데이터의 손실이 높고, 최종 정확
도는 차이가 더욱 벌어짐

!!
Batch Size의 크기가 클수록 비효
율적인 리소스 사용이 증가
==

실험4) Epoch 50, Batch Size 32



1) train accuracy (훈련 정확도)는 증가하지만
test accuracy(검증 정확도)는 계속된 증가를 보여
주지 않았음

2) 최종값에서의 accuracy와 loss 각각의 train 모
델과 test 모델의 차이가 점점 벌어짐

!!
train loss(훈련 손실)의 과 과 적합 요소가 Test에
대한 loss(검증 손실)을 높일 수 있다는 것 확인
==

기타 실험 변수 | 실험 내용

Epoch

Batch Size

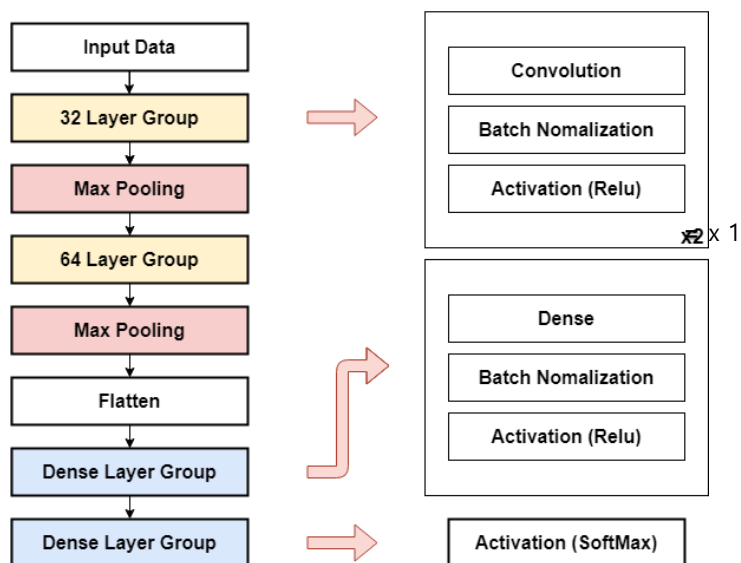
Layers

모든 Epoch는 30, Batchsize는 32라고 가정.

기본 : 각 레이어 훈련 2번씩, 2 레이어

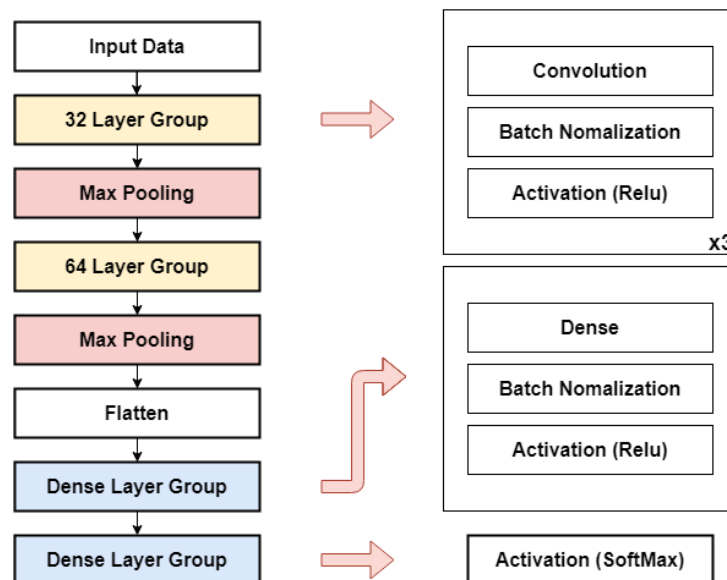
1

각 레이어 훈련 1번씩, 2레이어



2

각 레이어 훈련 3번씩, 2레이어



기타 실험 변수 | 실험 내용

Epoch

Batch Size

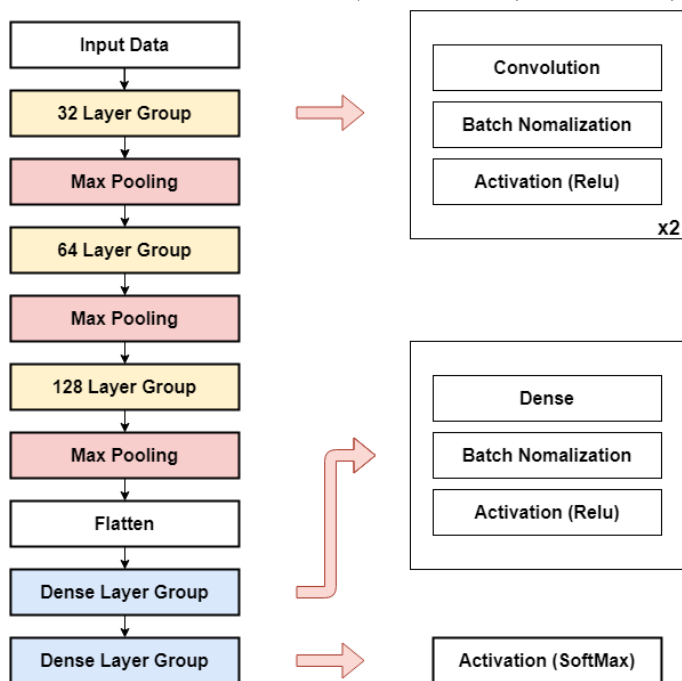
Layers

모든 Epoch는 30, Batchsize는 32라고 가정.

기본 : 각 레이어 훈련 2번씩, 2 레이어

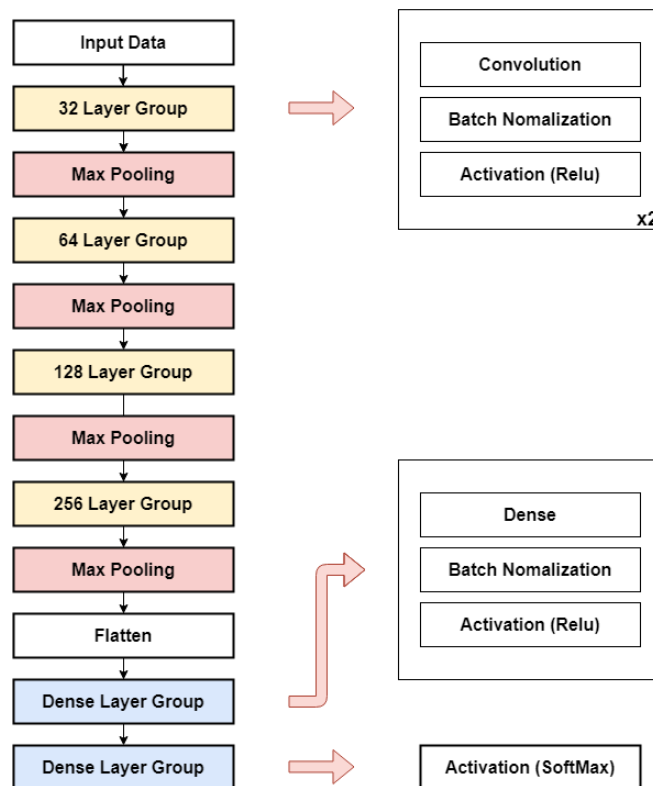
3

각 레이어 훈련 2번씩, 3레이어(128 추가)



4

각 레이어 훈련 2번씩, 4레이어 (128,256 추가)



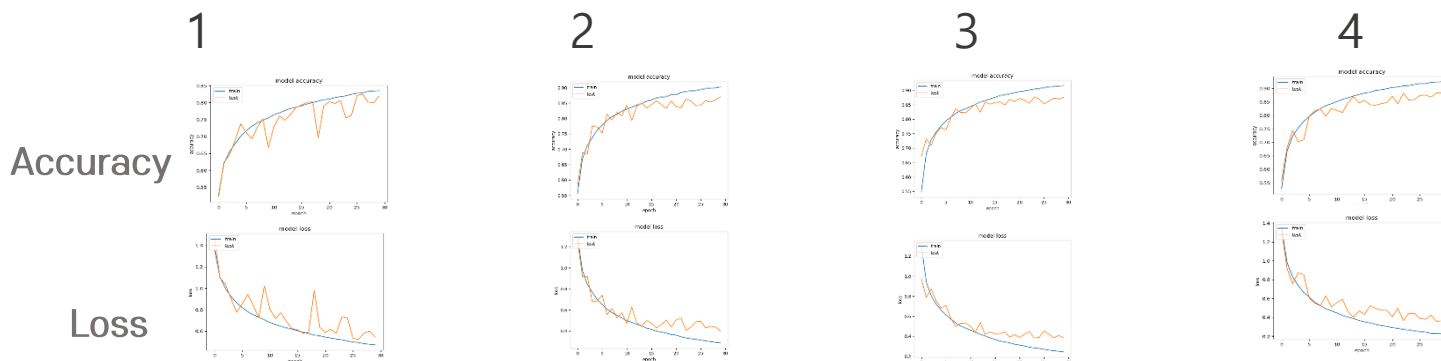
기타 실험 변수 | 실험 내용

Epoch

Batch Size

Layers

모든 Epoch는 30, Batchsize는 32라고 가정.



훈련을 1번씩 줄이니
기존의 실험보다
acc가 낮은 결과 확인
가능

훈련시간이 3배가 늘고
그만큼 acc도 증가
(과적합이 실험 1에 비해
더 늘어났는지 점검)

실험1에 비해 acc가 증가

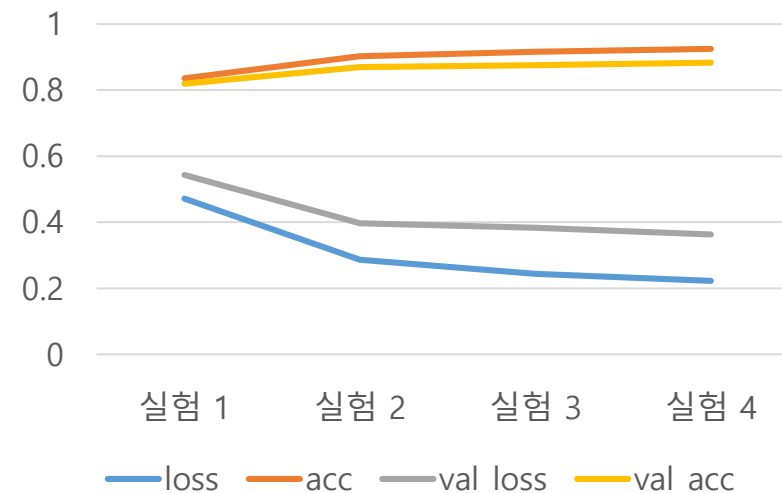
실험에 소요되는 시간이 상승
⇒ CUDA를 통해 실험하여
⇒ 시간을 단축.

acc 한 단계 더 향상
과적합이 생기는 듯 하였으나,
val_acc가 높아지고 있는 것을
확인

실험에 소요되는 시간이 상승
⇒ CUDA를 통해 실험하여
⇒ 시간을 단축.

- 기본 : 각 레이어 훈련 2번씩, 2 레이어
1. 각 레이어 훈련 1번씩, 2레이어
 2. 각 레이어 훈련 3번씩, 2레이어
 3. 각 레이어 훈련 2번씩, 3레이어(128 추가)
 4. 각 레이어 훈련 2번씩, 4레이어 (128,256 추가)

레이어 설정 별 변화



시각화

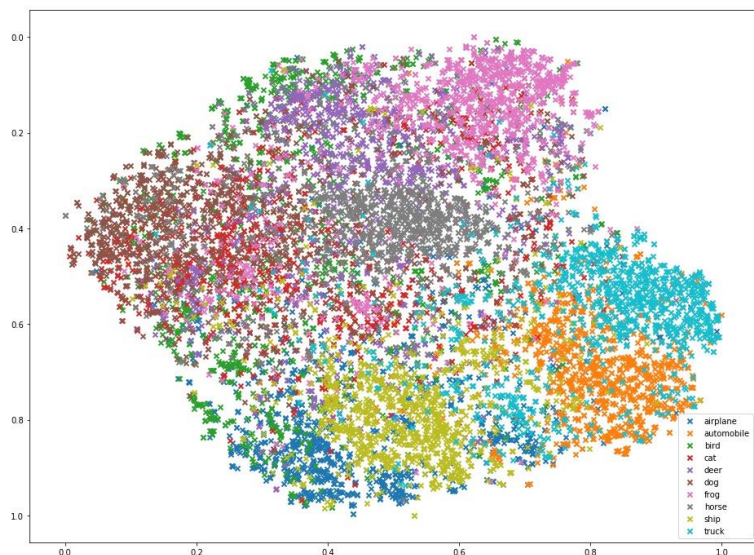
Intermediate Feature Space (TSN-E)

CAM

학습 과정에서 생성되는 feature space를 저장하여 영상 분류가 어떤 식으로 이루어지고 정확도가 어떻게 향상되는지를 시각화를 통해 검토

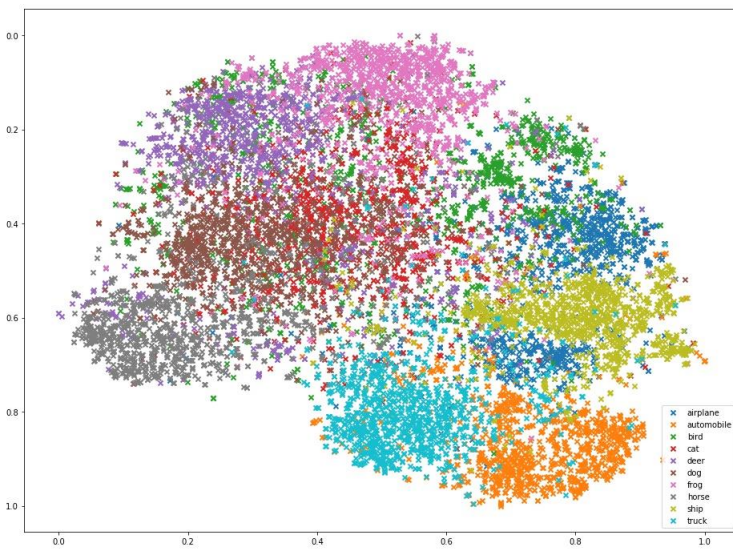
Step 1) 각 epoch에서 최고 정확도가 갱신될 때마다 모델을 저장

Step 2) t-SNE를 이용하여 그 모델에서 분류가 어떤 식으로 진행되었는지를 산점도 그래프로 확인

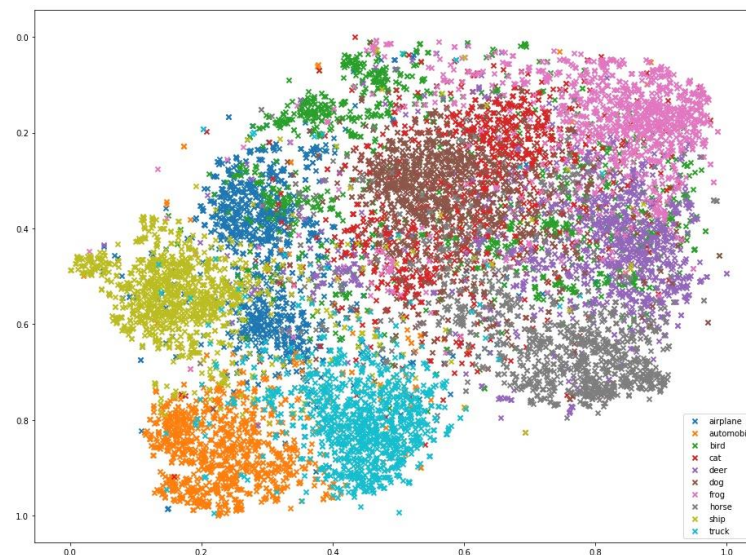


Epoch = 1일 때, accuracy는 0.5791

정확도가 낮아서 이미지의 특징이 유사한 것들은 제대로 분류하지 못하고 있음.



Epoch = 6으로 설정
=> 정확도 70%대로 상승

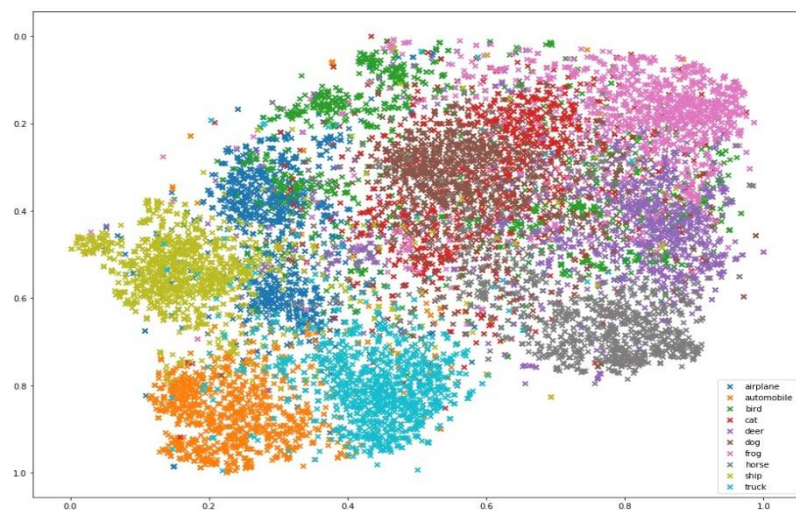


Epoch = 24로 설정
=> 정확도 80%대로 상승

최종 실험 결과

Intermediate Feature Space (TSN-E)

최종적으로는 epoch = 30으로 코드를 수행하였고,
그에 따른 Feature Space 확인



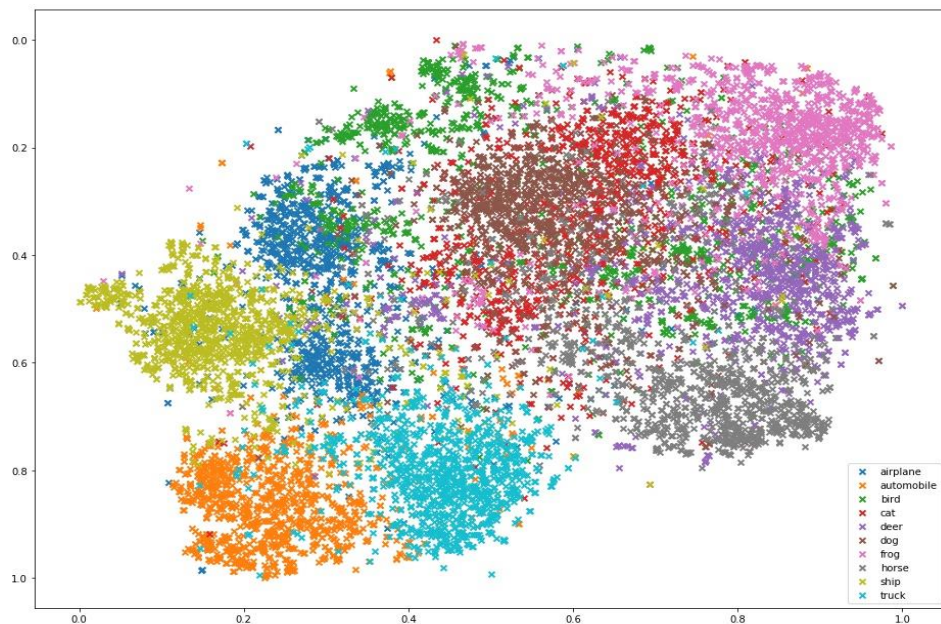
CAM



=> 128x128 픽셀에서 32 사이즈의 이미지로 추출한 Heat Map
해당 사진을 통해,
이미지의 특징적인 부분에서 높은 값을 가지는 것을 확인

최종 실험 결과

최종적으로는 epoch = 30으로 코드를 수행하였고, 그에 따른 Feature Space 확인 성공.
epoch = 24일 때, 정확도가 0.8497로 가장 우수하게 나온 것을 확인할 수 있었다.



[시사점 / 결론 토의 내용]

발견 :
정확도(epoch:30) < 정확도(epoch:24)

고찰 :
Epoch가 낮을 때는 모델이 학습 데이터의 기본 패턴을 학습하지 못한다.
반면, 결과에서 검증 데이터로 일반화하지 못하는 부분이 확인되어 훈련 과정에서 과적합이 있을 수 있었을 것이다. 라고 판단하였다.

결론 : 따라서 모델의 학습을 위해서 Epoch를 적절히 조율해가며 코드를 수행할 필요가 있다는 결론에 도달했다.

References

- "CONVOLUTIONAL NEURAL NETWORK", MATHWORKS, ACCESSED JUNE 5, 2021, <HTTPS://KR.MATHWORKS.COM/DISCOVERY/CONVOLUTIONAL-NEURAL-NETWORK-MATLAB.HTML>
- "NEURAL NETWORKS", MATHWORKS, ACCESSED JUNE 5, 2021, <HTTPS://KR.MATHWORKS.COM/DISCOVERY/NEURAL-NETWORK.HTML>
- YAQUB, M., JINCHAO, F., ZIA, M. S., ARSHID, K., JIA, K., REHMAN, Z. U., & MEHMOOD, A. (2020). STATE-OF-THE-ART CNN OPTIMIZER FOR BRAIN TUMOR SEGMENTATION IN MAGNETIC RESONANCE IMAGES. BRAIN SCIENCES, 10(7), 427. <HTTPS://DOI.ORG/10.3390/BRAINSCI10070427>
- IOFFE, S., & SZEGEDY, C. (2015, JUNE). BATCH NORMALIZATION: ACCELERATING DEEP NETWORK TRAINING BY REDUCING INTERNAL COVARIATE SHIFT. IN INTERNATIONAL CONFERENCE ON MACHINE LEARNING (PP. 448-456). PMLR.