1.9 다양한 정규화 기법들

- 정규화 기법은 성능 향상을 위한 것
- 신경망 개발 과정에서는 평가 단계가 정규화 단계에 대한 모의 실험 역할을 수행

L2 | L1 | Noise injection | Drop out | Batch normalization

정규화란?

빠르게 달려야 하는 육상 선수를 훈련시킬 때 일부러 모래주머니를 차게 하는 것처럼

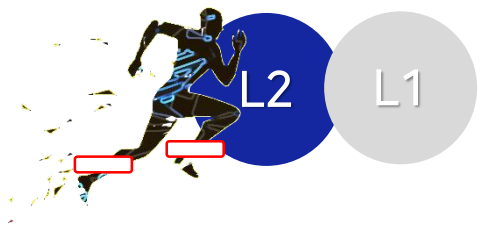
딥러닝 알고리즘에 **일부러 학습을 방해하는 제약**을 가해,

더 많은 실력을 쌓도록 유도하는 방법론

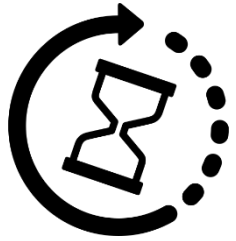
1.9 다양한 정규화 기법들

A $L' = L + \frac{1}{2}\lambda \sum_{i=1}^{n} w_i^2$

정규화항을 추가하여 복잡한 모델에서 손실함수가 최솟값인 경우에도 **과적합**을 막는다.



B $P_{L2} = \frac{1}{2}\lambda \sum_{i=1}^{n} w_i^2$

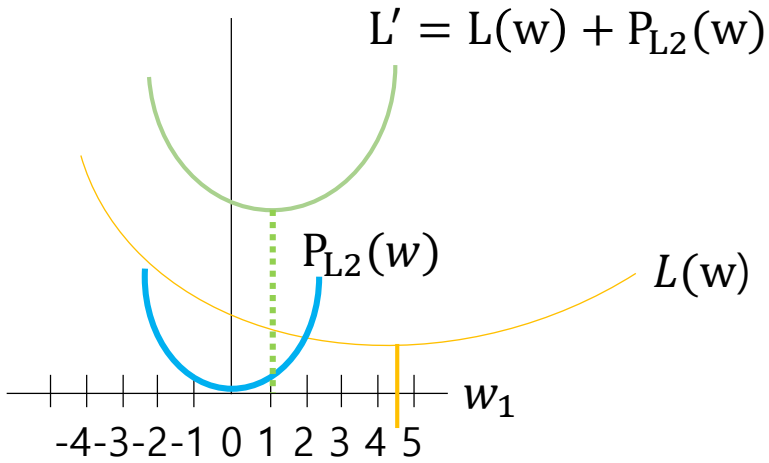


학습중...

가중치 파라미터 절댓값 중 일부 혹은 전부가 과도하게 커지는 방향으로 나아가는 경우 발생

- ✓ L2 손실, L1 손실은 **모두 절댓값이 큰 파라미터에 대해 불이익**을 할당.
- ✓ 값의 폭주를 막고 **작은 절댓값의 파라미터들로 문제를 풀도록 압박**
- ✓ 단, 이때 **편향**은 **일반적으로 포함하지 않는다.**
  편향을 적용하지 않았을 때, 더 L2 효과가 향상

※ λ(lambda) : 정칙화율, 손실율, ...
정규화 항의 영향을 정하는 양의 상수

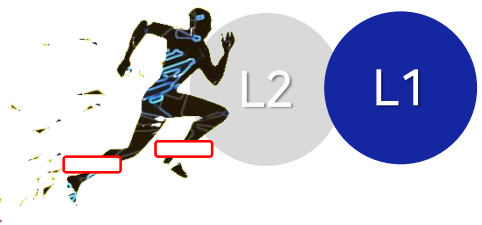


C *#임의의 손실함수 그래프*
최솟값은 4.5 부근

D *# 람다값 제외하고 정칙화항 그래프*
#$\frac{1}{2}w_1^2$ 의 모양인 0점을 지나는 단순 2차함수

E *#두 함수를 더한 그래프가 그려진 것을 확인할 수 있고,*
#일반 손실함수의 최소값보다 훨씬 작은 최소값을 구할 수가 있음.
#가중치가 너무 커지게 되는것을 방지해서 작은 값에 가까워 지게 함.

1.9 다양한 정규화 기법들



$$L' = L + \alpha \sum_{i=1}^{n} |w_i|$$

$$P_{L1} = \alpha \sum_{i=1}^{n} |w_i|$$

✓ L1은 일률적으로 정해진 값을 덜어내는 방법

파라미터가 작은값이라면은 0 혹은 0에 가까운 파라미터 값을 양산.

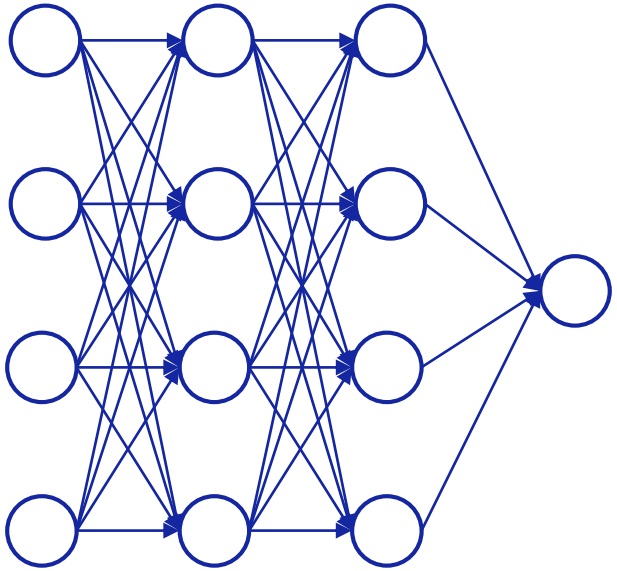
✓ L2와 마찬가지로 과적합을 억제하는 효과를 가지고 있기는 하지만,

그보다 가중치 분포에 미치는 영향이 더 크다.

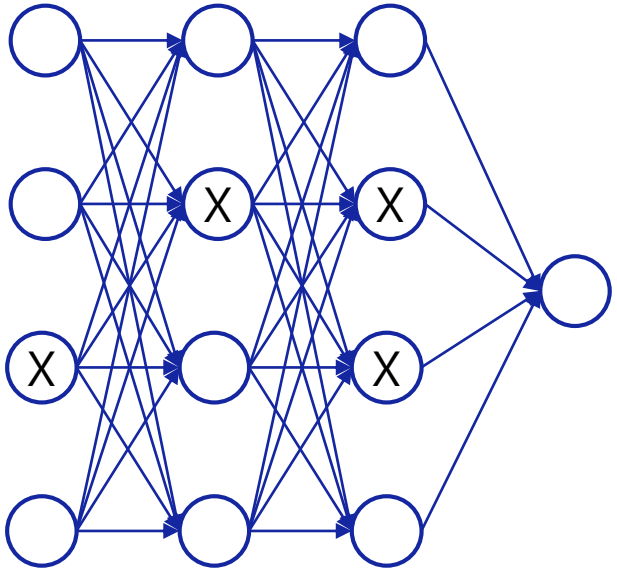
➢ 가중치들이 0 혹은 0에 가까운 원소를 많이 포함하는 일명 희소 텐서가 되는 것을 바란다면 L1손실을 이용.

1.9 다양한 정규화 기법들

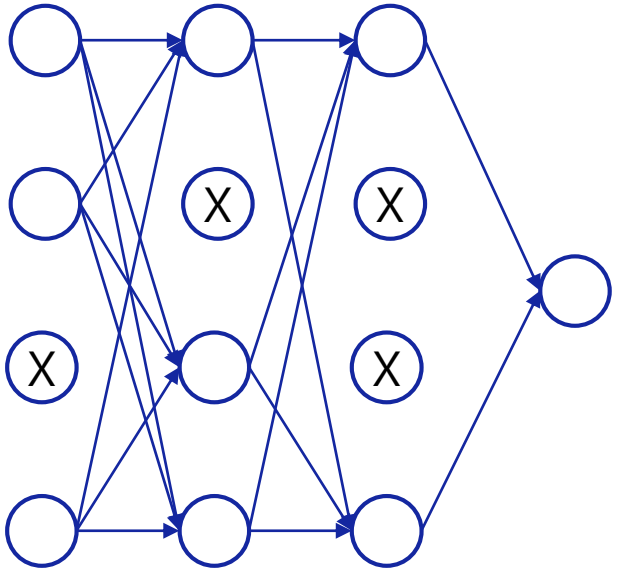
- ✓ 드롭아웃은 과적합 방지를 위해 도입되는 정규화 기법.
- ✓ 계산량을 줄이기 위한 기법은 아니다.
- ✓ 성능 향상을 위해 계산량을 늘려 투자하는 기법.



A. Dropout 적용 전



B. Dropout 처리
(난수함수 처리)



C. Dropout의 실질적 효과
(계산량은 오히려 늘어남)

1.9 다양한 정규화 기법들


```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 11, 11, 64)        18496
_________________________________________________________________
average_pooling2d_1 (Average (None, 5, 5, 64)          0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 3, 3, 128)         73856
_________________________________________________________________
flatten_2 (Flatten)          (None, 1152)              0
_________________________________________________________________
dense_4 (Dense)              (None, 128)               147584
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_5 (Dense)              (None, 10)                1290
=================================================================
Total params: 241,546
Trainable params: 241,546
Non-trainable params: 0
_________________________________________________________________
```

← dropout_계층 삽입
keep ratio = 60%

↑
40%의 확률로
dropout

- 드롭아웃 기법은 **학습시점에서만 적용**되야 하고,
  학습 이후에는 **모든 퍼셉트론들이 다시 투입**되어 문제를 해결
- 같은 문제를 반복해서 풀더라도, **그 처리를 담당하는 퍼셉트론 그룹을 매번 다르게 배당**함으로써 문제 자체를 일정한 패턴형태로
  단순 암기해버리기 어렵게 만드는데서 얻을 수 있습니다.
- 드롭아웃의 무작위 처리는 **지속해서 학습과정에 교란**을 가져오는데
  이 덕분에 학습과정이 일종의 **매너리즘에 빠지지 않고 학습**
- 드롭아웃 기법은 실제로 처리에 반영되는 퍼셉트론 수를 줄여버림으로서
  학습을 어렵게 만드는 장애요인으로 작용하고, **학습속도를 떨어뜨리는 반작용 발생**
- 과적합 현상이 줄어들고, 학습과정에서 성능과 평가 단계에서 성능의 차이가 줄어들면서
  실전에서 우수한 품질을 기대할 수 있음.

1.9 다양한 정규화 기법들

- ✓ **미니배치 내의 데이터**들에 대해 벡터 성분별로 **정규화를 수행**하는 방식.
- ✓ 정규화는 **대상 값들에 동일한 선형 변환**을 가해줌으로써 **평균 0, 표준편차 1의 분포**로 만들어 주는 처리.
- ➢ 입력 성분 간의 **분포 차이로 인한 가중치 학습의 불균형을 방지**하기 위해 도입

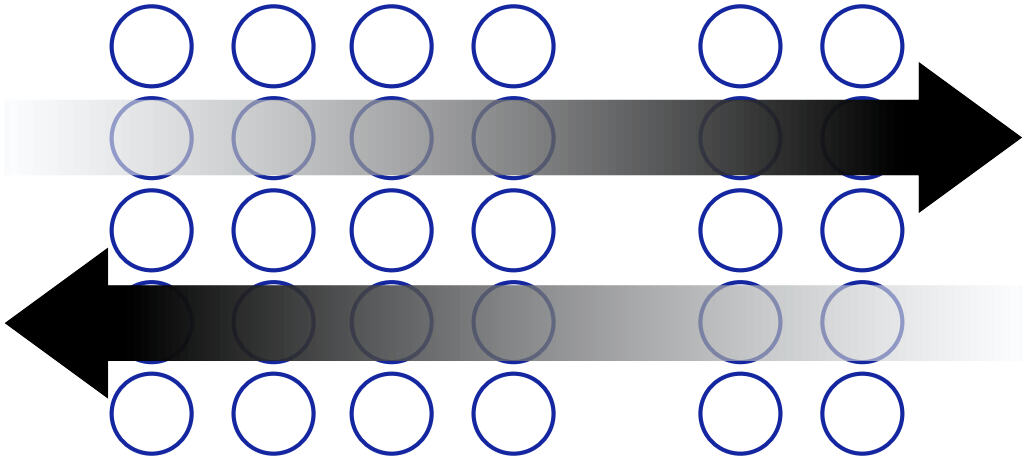
신경망의 성능을 올리기 위해서는 신경망의 구조를

특이하게! 난해하게 쌓아야 한다는데?!

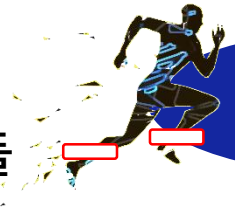
드롭아웃과 배치 정규화가 등장했으니

이제 그럴필요 없어!

『깊은 신경망』



- ✓ 신경망을 깊게 쌓다보면은 **역전파 처리 중 층을 거듭할 수록** 파라미터 성분에 따라 **기울기가 급격히 소멸 혹은 폭주**하면서 **학습이 제대로 이뤄지지 못하는 경우 발생**
- ➢ 네트워크각 층의 입력을 구성하는 **성분별 분포가 심하게 달라서 발생**



Batch normalization

1.9 다양한 정규화 기법들

- ✓ **미니배치 데이터**를 정규화 대상으로 삼는다.
- ✓ **전체 데이터셋 정규화**는 최초의 입력 데이터 즉, **처음에만 가능한 기법**.
- ➢ 그렇기 때문에 **은닉계층에서 생산하는 은닉 벡터에 대해 적용 불가**
- ✓ **미니배치 데이터 기준** 평균과 표준편차를 구한 후 각각의 데이터 값에서 평균을 뺀 후 표준편차로 나누는 **정규화 과정**을 거치게 되면은 **성분별 불균형은 우선 해결**.

- ✓ 그리고 이 값에다가 **크기요소(scale factor)라는 파라미터 값을 곱**하고, **이동요소(shift factor)라는 파라미터 값**을 더해서 성분별 불균형을 갖는 새로운 값들을 만들어 내줍니다.

- ✓ 매번 달라지는 미니배치 평균과 분산의 **이용으로 일종의 잡음 주입** 즉, **노이즈 효과 발생.**
- ➢ 신경망은 **더 어렵게 학습**을 진행, 미니배치 구성에 에포크마다 무작위로 바뀐다면 **배치 정규화는 학습 과정을 끊임없이 교란.**
(★ 드롭아웃과도 같은 효과)

이 작은 표본집단에 불과한 **미니 배치 데이터**에 정규화를 적용한다는 것이 정말 효과가 있다고?

오히려 Dropout 보다 효과가 좋아!

작은 미니배치는 배치 정규화 효과가 극단적으로 나타날 수 있기 때문에, 배치 정규화를 이용할 때 너무 작은 미니배치보다는 **약간 크기를 조금 키운 미니배치에서 학습 결과가 더 좋게 나와.**

같은 수의 학습 데이터를 처리할 때 **미니배치 크기를 키워주게 되면**은 자연스럽게 미니배치 처리 횟수가 줄어들면서 **결과적으로 학습 속도를 높이는 부수적인 효과** 까지 얻는다고!

그럼 이 배치정규화는 **어디에 배치**해야 하는거야?

합성곱 계층이나 **완전연결계층 바로 다음** 혹은 활성화 함수 통과 직전에 배치시켜 주는 것이 좋아!