Regularization

모델의 과적합을 방지하기 위한 방법들

1) Regularization

각 피처가 갖는 영향을 줄여 (=파라미터들이 작은 값을 가지게 함) 모델의 복잡성을 줄이고자 함. 파라미터에 penalty를 부여하기 위해 기존의 loss function에 가중치를 더해서 사용

Ex) 100개의 피처가 존재하는 집값예측

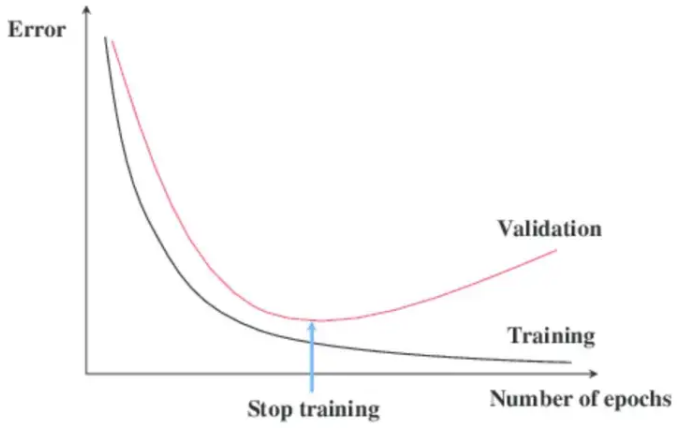
$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{100}\theta_j^2$$

- 값이 1인 θ_0을 제외한 나머지를 적절한 regularization parameter인 λ을 곱하여 작게 유지
- 선형회귀의 경우 RSS에 가중치를 추가하면 Lasso (L1), Ridge (L2)

이렇게 파라미터들의 합에 곱해져 가중치가 필요 이상으로 증가하는 것을 제한시키는 작은 값이 결국 **weight decay**임.

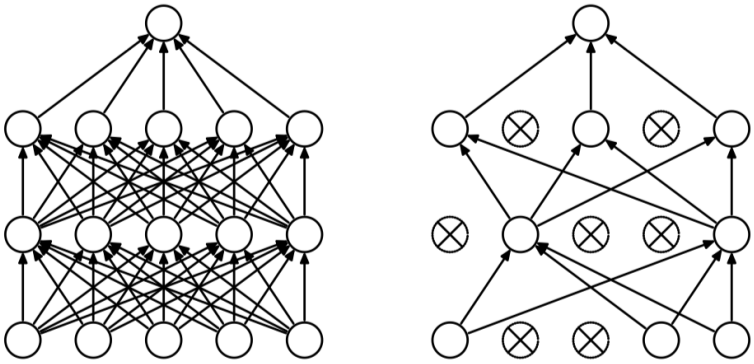
2) Early Stopping

validation error가 최소일 때 훈련을 중단시키는 방법



- 매우 간단하고 효율적이어서 Geoffrey Hinton이 “beautiful free lunch”라고 불렀다고..
- early stopping customizing: link

3) Dropout

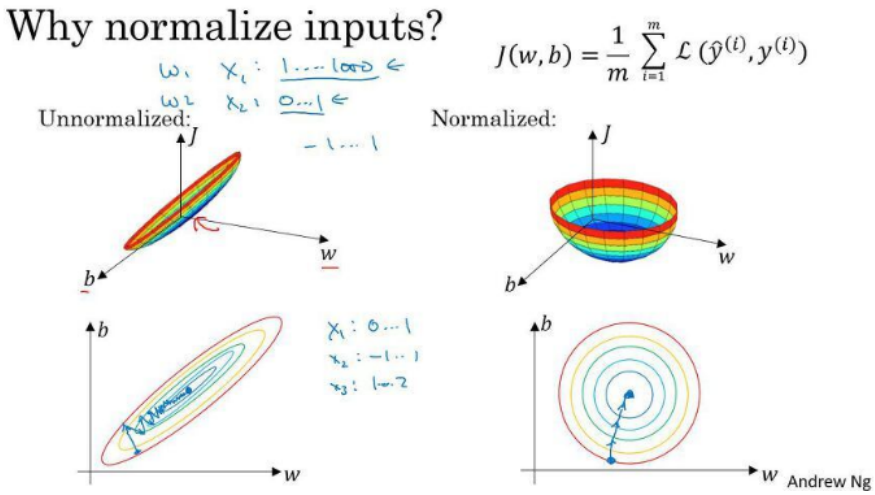


(a) Standard Neural Net (b) After applying dropout.

훈련 과정에서 뉴런을 제거하는 방법. 각 훈련 iteration마다 활성화되는 뉴런들이 다르다

4) Normalization

학습을 더 빠르게 하고, local optimum에 빠지는 가능성을 줄이는 등의 목적으로 데이터의 범위를 제한하는 것



정규화를 하지 않았을 때와 했을 때의 수렴속도 차이를 확인할 수 있다. 형태가 spherical 한 것이 gradient descent 방법으로 찾아갈 때 더 빠르다

Batch Normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

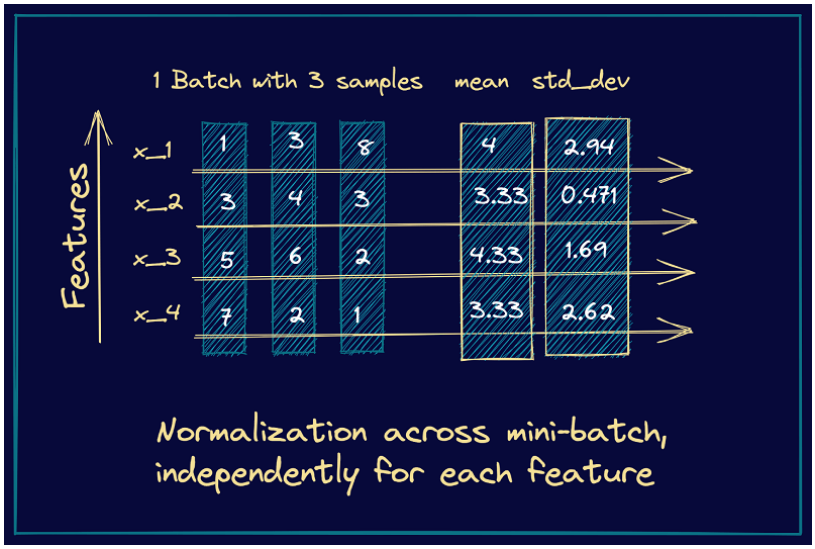
$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

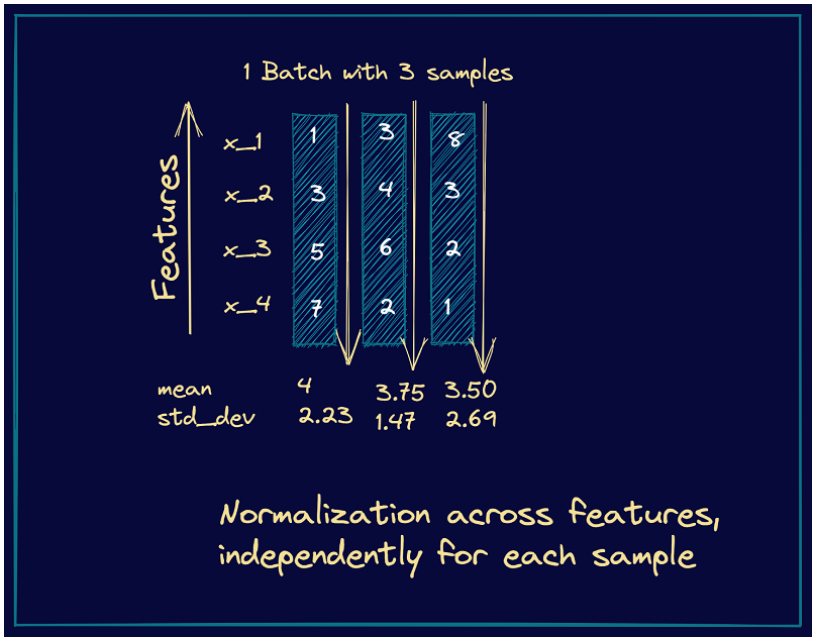
학습시 미니배치의 평균과 분산을 이용해서 정규화 후, γ와 β를 통해 scale 및 shift 실행 (활성화함수의 비선형 성질을 유지하기 위함). 정규화된 값을 활성화함수의 입력으로 사용 후 출력물을 다음층의 입력으로 사용

- γ와 β는 학습되는 파라미터. 꼭 평균이 0, 분산이 1이 되지 않아도 된다.
- 입력층을 지나서 만나는 레이어들의 입력은 정규화하기 쉽지 않음. BN은 이를 해결함
- regularization 효과가 있어서 dropout 등 사용하지 않아도 됨



Batch Normalization

Layer Normalization

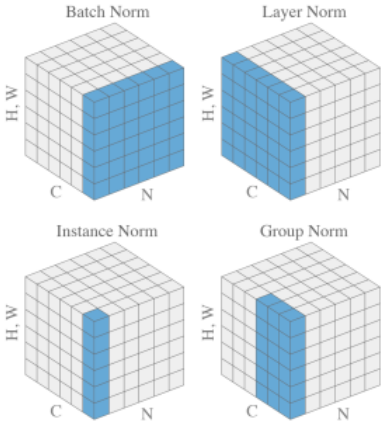


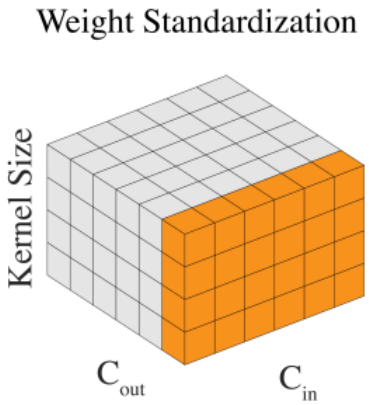
Layer Normalization

- https://arxiv.org/abs/1607.06450
- 미니배치에 의존하지 않는 정규화

Group Normalization

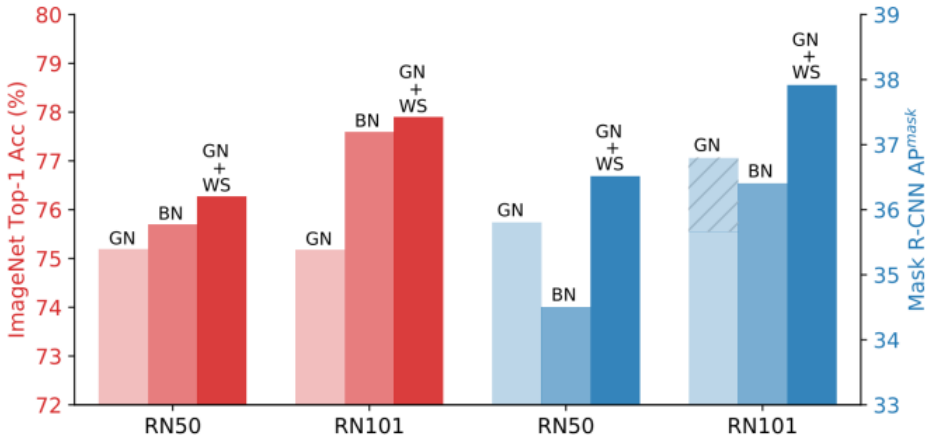
LN와 정규화 방향은 같지만 피처를 그룹으로 세분화하여 각 그룹별로 정규화





Weight Standardization

레이어의 가중치가 0의 평균, 1의 분산을 갖도록 바꾸어 정규화된 weight로 학습. 마찬가지로 미니배치에 대한 dependency가 없고, 배치 사이즈에 상관 없이 좋은 성능을 보임



Imagenet 분류 ResNet모델별 정규화 기법의 성능 평가 결과. GN과 WS를 같이 사용한 기법이 BN보다 우수한 성능을 보인다

참고

https://junstar92.tistory.com/24

https://ai-pool.com/a/s/dropout-in-deep-learning

https://light-tree.tistory.com/132

https://www.pinecone.io/learn/batch-layer-normalization/

https://ai-pool.com/a/s/normalization-in-deep-learning

https://towardsdatascience.com/different-normalization-layers-in-deep-learning-1a7214ff71d6