Overfitting 방지

Augmentation

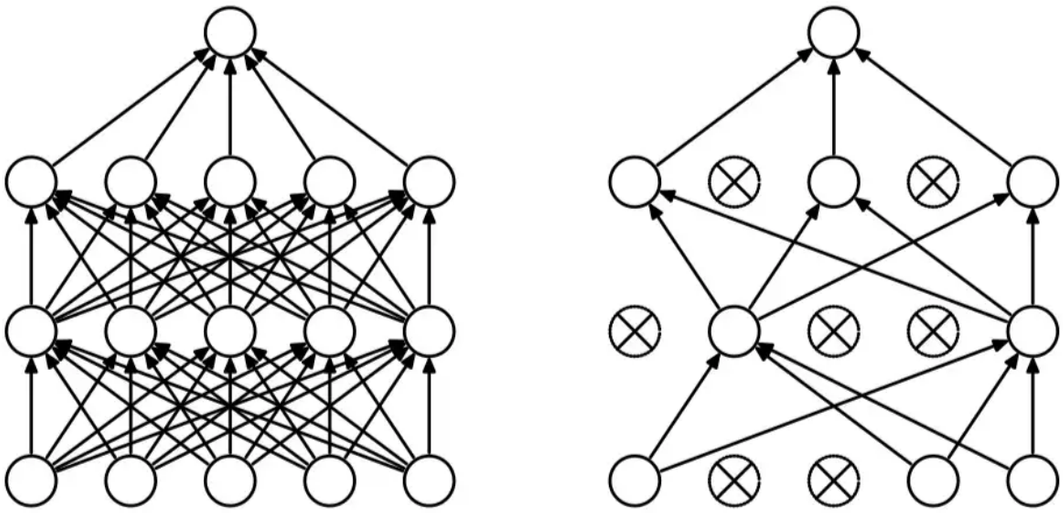
- 원본 이미지를 회전(rotate), 뒤집거나(flip), 자르는(cut) 등의 방법을 통해 새로운 이미지를 만드는 방법
- 이미지를 자르거나 섞는 방법으로 만들어진 새로운 이미지는 모델의 과적합(Overfitting)을 막아주는 중요한 역할
- 이미지 증강은 오직 학습 데이터셋에만 적용한다

tf.keras.preprocessing.image.ImageDataGenerator | TensorFlow v2.11.0

Generate batches of tensor image data with real-time data augmentation.

https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator

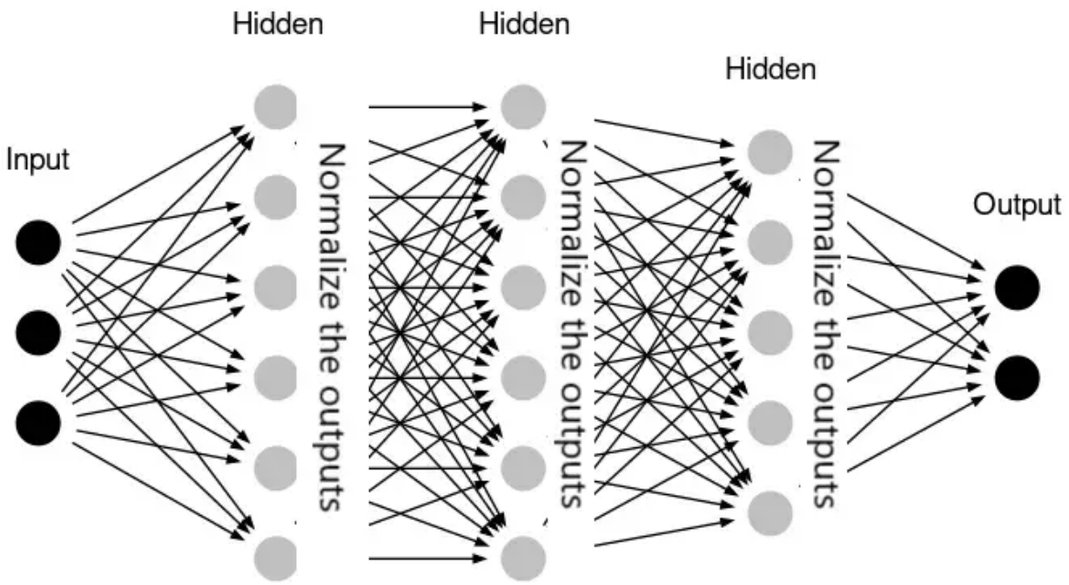
Dropout

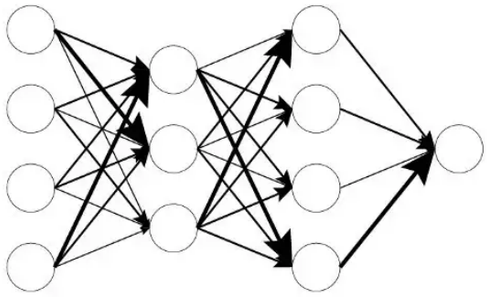


(a) Standard Neural Net (b) After applying dropout.

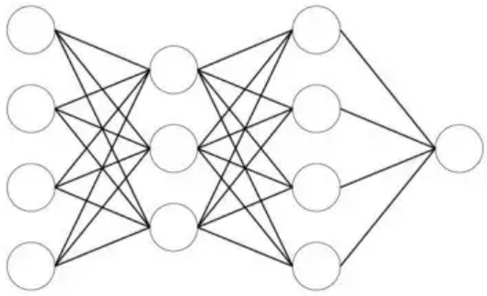
- 뉴런을 임의로 삭제하면서 Train하는 방법
- Train시 은닉층의 뉴련을 무작위로 골라 삭제
- Train시 데이터를 다음 Layer로 데이터를 전송할 때마다 삭제할 뉴런을 무작위로 선택하고, Test 시에는 모든 뉴런에 신호를 전달한다.

배치정규화





- **Raw** signal
- **High interdependancy** between distributions
- **Slow** and **unstable** training



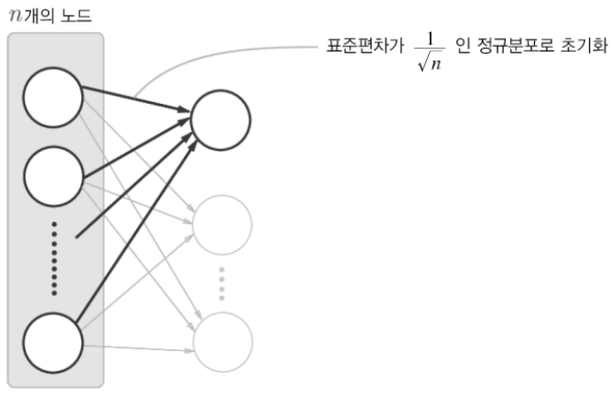
- **Normalized** signal
- **Mitigated interdependancy** between distributions
- **Fast** and **stable** training

- 드롭아웃과 마찬가지로 신경망 계층으로 구현되어 작동할 수 있는 정규화 기법이다.
- 장점
  - 신경망의 경사도 흐름 개선
  - 높은 학습률 허용
  - 규제와 유사한 동작, 드롭아웃 필요성 감소

Covariant Shift(공변량 변화)

- 배치정규화는 공변량 변화(covariate shift)문제를 해결하고자 제안
- Internal Covariant Shift
  - Internal Covariant Shift : **학습 과정에서 계층 별로 입력의 데이터 분포가 달라지는 현상**
  - Training이 진행되면서 i-Layer의 파라미터가 바뀜 -> i+1Layer 입장에서는 입력되는 데이터가 수시로 변화
  - 층이 깊어짐에 따라 영향이 더 커짐 -> 학습 방해 요인으로 작용

- 정규화를 층 단위로 적용



- 내부 공변량 변화 현상 해결
- 정규화 적용 위치가 중요 -> 일반적으로 FC층, Conv층 이후 or 비선형 함수 이전에 적용
- 미니배치에 적용하는 것이 유리

배치정규화 동작

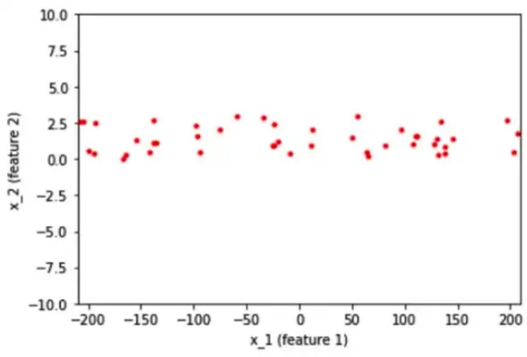
Batch Normalization (BN)

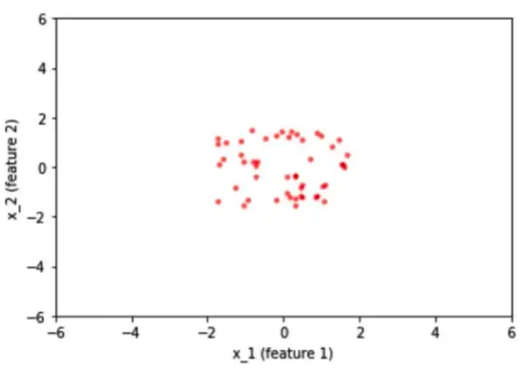
➡ layer ➡ x ➡ $\hat{x} = \frac{x - \mu}{\sigma}$ ➡ $y = \gamma\hat{x} + \beta$

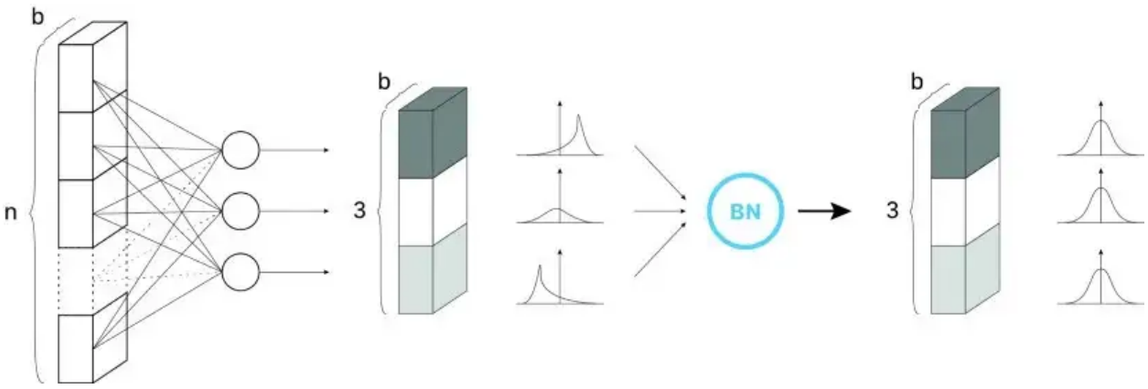
- μ: mean of x in mini-batch
- σ: std of x in mini-batch
- γ: scale
- β: shift

- μ, σ: functions of x, analogous to responses
- γ, β: parameters to be learned, analogous to weights

- 배치 정규화에 따른 Feature 의 분포







Batch normalization in 3 levels of understanding

Batch normalization is computed differently during the training and the testing phase. B.1.1) Training At each hidden layer, Batch Normalization transforms the signal as follow : The BN layer first

https://towardsdatascience.com/batch-normalization-in-3-levels-of-understanding-14c2da90a338

배치 정규화(Batch Normalization)

gaussian37's blog

https://gaussian37.github.io/dl-concept-batchnorm/

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

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

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

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

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