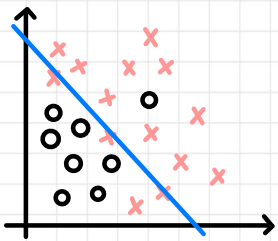
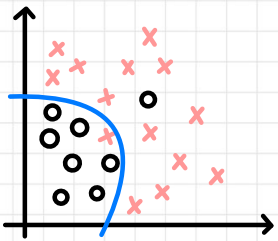
Khyeyoon

Bias and Variance

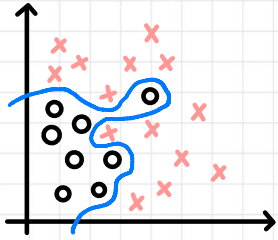


high bias

Underfitting



just right



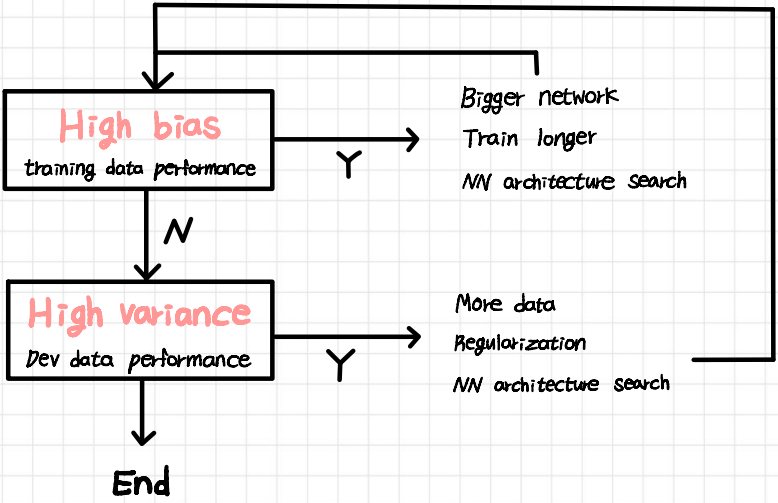
high variance

Overfitting

→ Training Set와 Dev Set의 관계로 모델이 적절하게 학습되고 있는지 알 수 있음

Khyeyoon

Basic recipe for machine learning



Khyeyoon

Regularization : Logistic regression

→ data의 scale을 학습하기 좋은 형태로 조정하는 것

$W \in \mathbb{R}^{n_x}, \; b \in \mathbb{R}$

$$\min_{w,b} J(w,b) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^i, y^i) + \frac{\lambda}{2m}\|w\|_2^2 + \cancel{\frac{\lambda}{2m} b^2}$$

regularization parameter

w는 많은 매개변수로 구성된 벡터이기 때문에 b는 하나의 숫자이기 때문에 큰 영향력을 가지지 못함

L_2 regularization $\quad \|w\|_2^2 = \sum_{j=1}^{n_x} w_j^2 = w^T w$

L_1 regularization $\quad \frac{\lambda}{2m}\sum_{j=1}^{n_x} |w_j| = \frac{\lambda}{2m}\|w\|_1$

Khyeyoon

Regularization : Neural Network

$$J(w^{[1]}, b^{[1]}, \cdots, w^{[L]}, b^{[L]}) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^i, y^i) + \frac{\lambda}{2m}\sum_{l=1}^{L} \|w^{[l]}\|_F^2$$

"Frobenius Norm" $w : (n^{[l-1]}, n^{[l]})$

$$\|w^{[l]}\|_F^2 = \sum_{i=1}^{n^{[l-1]}} \sum_{j=1}^{n^{[l]}} (w_{ij}^{[l]})^2$$

$$w^{[l]} = \underbrace{\left(1 - \frac{\alpha\lambda}{m}\right) w^{[l]}}_{\text{Weigh decay}} - \alpha(\text{Back propagation 값})$$

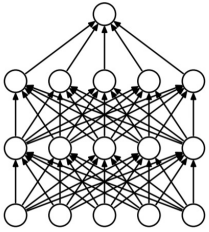
w에 1보다 작은 값을 곱하기 때문에 L2 Norm은 weigh decay라고 불리기도 함

λ 값이 커지면, 많은 hidden unit을 0에 가깝게 만들면서 각 unit의 영향력이 감소

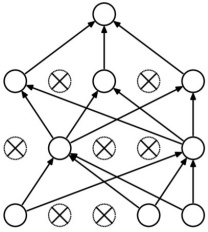
→ logistic regression과 같은 단순한 신경망

Khyeyoon

Dropout regularization



(a) Standard Neural Net



(b) After applying dropout.

특정 확률 p로 node를 활성화하거나 비활성화 함

→ overfitting 방지

Inverted dropout

```
# Layer 3
```

→ dropout vector

```
d3 = np.random.rand(a3.shape[0], a3.shape[1]) < keep-prob
a3 = np.multiply(a3, d3)
a3 /= keep-prob
```

$Z^{[4]} = W^{[4]} a^{[3]} + b^{[4]}$

→ 20%는 0

0.8

Z의 값이 줄어들지 않으려면 a3을 keep-prob로 나누어줘야함

??