성균관대학교

SIOR

로봇학회

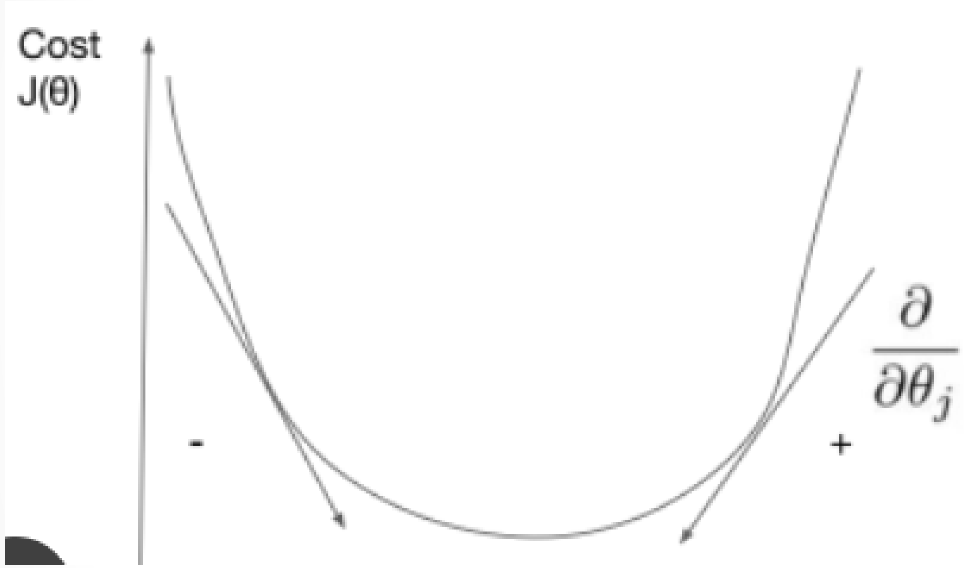
2022년 05월 00일

AI

n 주 차

목차

1. Learning rate

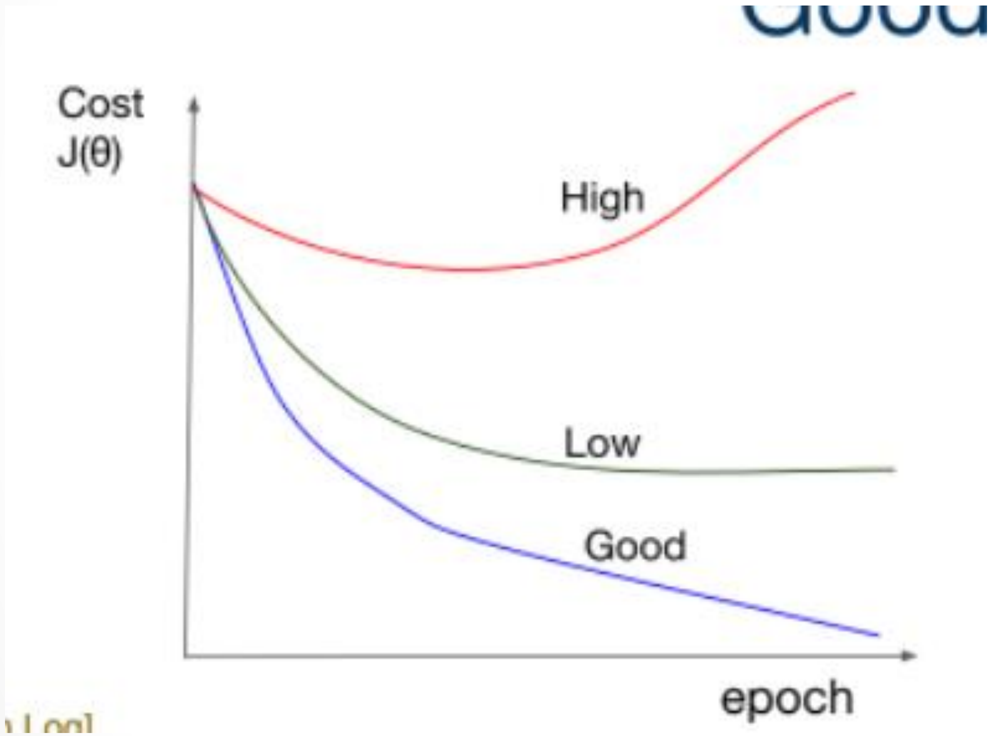


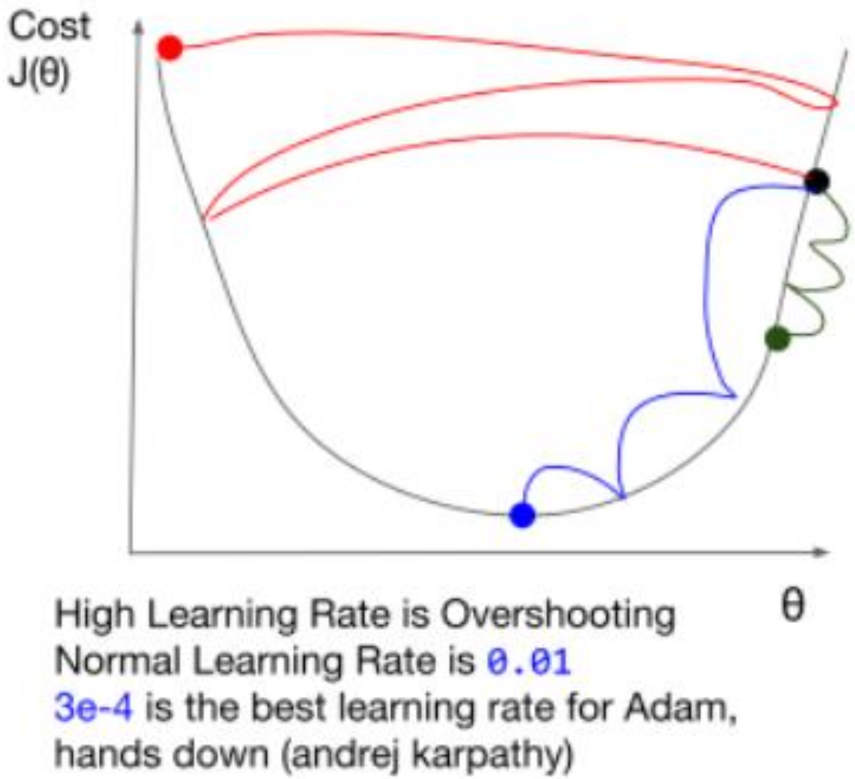
Repeat $\{ \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \}$

Learning rate is a hyper-parameter that controls how much we are adjusting the weights with respect the loss gradient

1. Learning rate

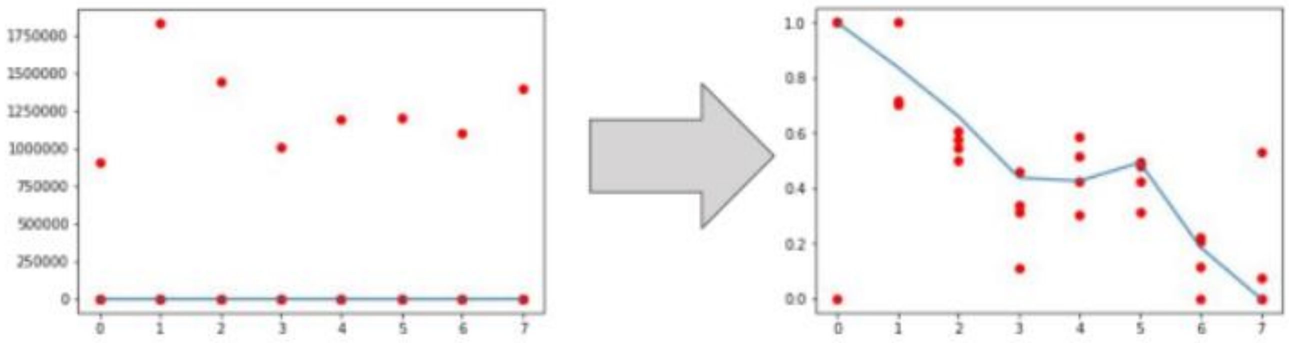
...





2. Data preprocessing

Feature Scaling



Standardization
(Mean Distance)

$$x_{new} = \frac{x - \mu}{\sigma}$$

[Python Code(numpy)]

```
Standardization = (data - np.mean(data)) / sqrt(np.sum(
(data - np.mean(data))^2 ) / np.count(data))
```

Normalization
(0~1)

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

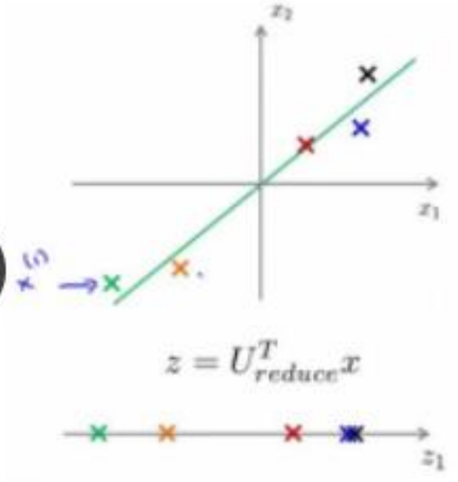
```
Normalization = (data - np.min(data, 0)) / (np.max(data, 0)
- np.min(data, 0))
```

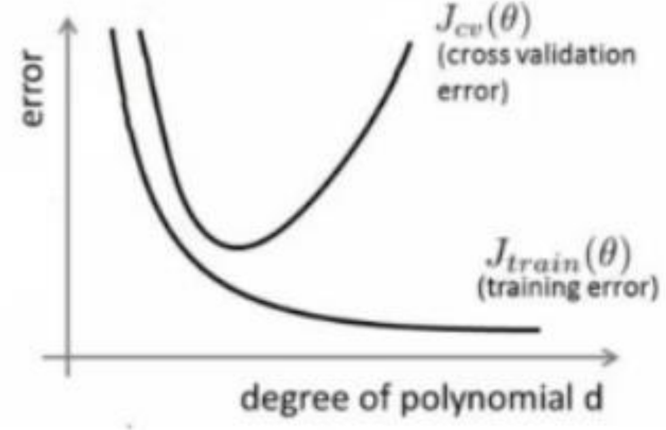
3. Overfitting

1.

Set a features

- Get more training data - more data will actually make a difference, (helps to fix high variance)
- Smaller set of features - dimensionality reduction(PCA) (fixes high variance)
- Add additional features - hypothesis is too simple, make hypothesis more specific (fixes high bias)





1. $h_\theta(x) = \theta_0 + \theta_1 x$
2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$
3. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3$

$\vdots$

10. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$

[sklearn Code]

```
from sklearn.decomposition import PCA
pca = decomposition.PCA(n_components=3)
pca.fit(X)
X = pca.transform(X)
```

3. Overfitting

2.

Regularization (Add term to loss)

Large λ
High bias (underfit)
$\lambda = 10000.\ \theta_1 \approx 0, \theta_2 \approx 0, \ldots$
$h_\theta(x) \approx \theta_0$

Intermediate λ
"Just right"

Small λ
High variance (overfit)
$\lambda \approx 0$

λ-- : fixes high bias (Under fitting)
λ ++ : fixes high variance (overfitting)

Linear regression with regularization

Model: $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

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

[Tensorflow Code]

```
L2_loss = tf.nn.l2_loss(w) # output = sum(t ** 2) / 2
```

성 균 관 대 학 교

Thank You

로 봇 동 아 리