

10

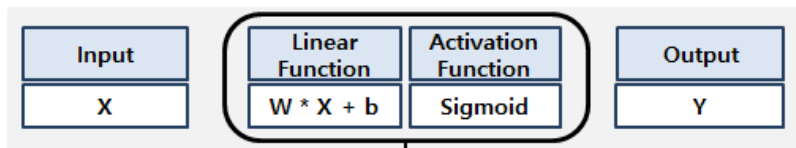
인공신경망 튜닝

1. 튜닝 I



인공신경망 튜닝 I 개요

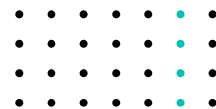
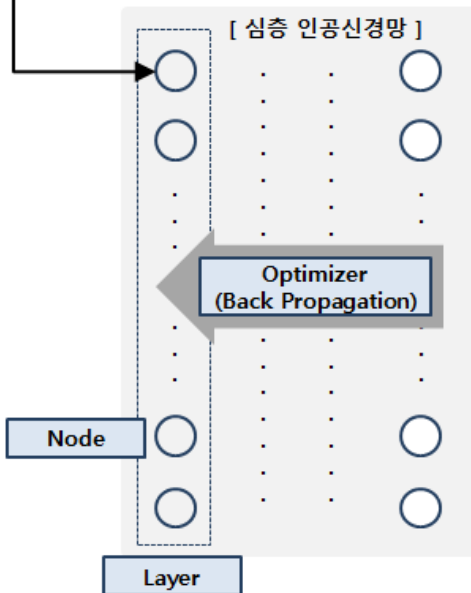
[단일 인공신경망]



[Tuning Point]

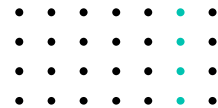
- | | | |
|---|---------------------|----------|
| ① | Input | 입력 값 전처리 |
| ② | Cost Function | 비용함수 |
| ③ | Activation Function | 알고리즘 |
| ④ | Initializer | 가중치 초기화 |
| ⑤ | Optimizer | 알고리즘 |
| ⑥ | Node | 노드 개수 |
| ⑦ | Layer | 레이어 개수 |

[심층 인공신경망]

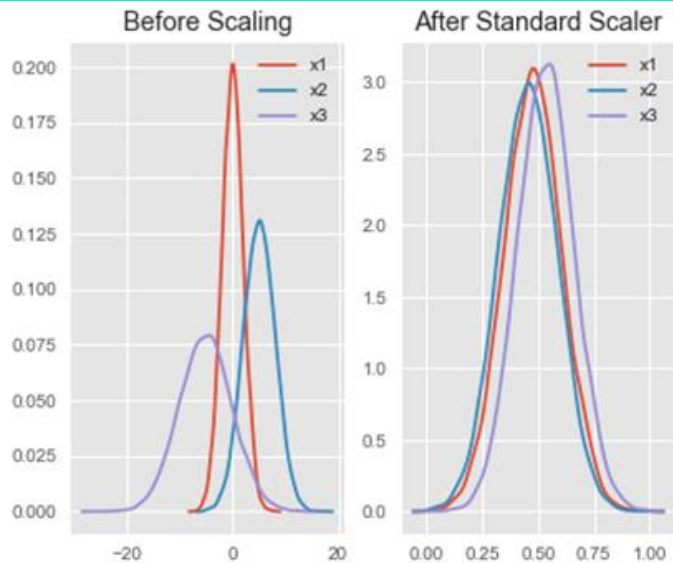


인공신경망 튜닝 I

입력데이터 전처리



표준화(Standardization)

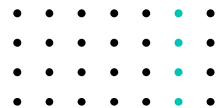


Scaling | Pic credits: Google

- 표준화는 표준정규분포를 구해서 데이터를 변환하는 기법
가우시안 분포를 가진 데이터에 일반적으로 표준화 기법을 적용



인공신경망 튜닝 I 입력데이터 전처리



정규화(Normalization)

노름
(Norm)

- 벡터의 크기(혹은 길이)를 측정하는 방법(혹은 함수)

$$\|x\|_p := (\sum_{i=1}^n |x_i|^p)^{1/p}$$

- n : 원소의 개수 • x : 두 벡터 사이의 거리
- L1 Norm : p가 1 • L2 Norm : p가 2 (default)

- 정규화는 벡터의 각 원소를 노름으로 나누어준 것
- 원소의 절대 값이 1보다 작은 값으로 모두 변경된다

정규화
(Normalization)

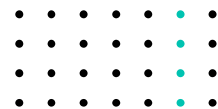
- 크기를 1로 규격화하기 위해 벡터를 노름으로 나눈 것

$$\text{Normalization}(x) = x / \|x\|$$



인공신경망 튜닝 I

입력데이터 전처리



```
import numpy as np
import matplotlib.pyplot as plt
```

```
def normalize(x):
    norm = np.linalg.norm(x)
    if norm == 0:
        return x
    return x / norm
```

① 벡터의 노름 구하기

② 벡터의 정규화

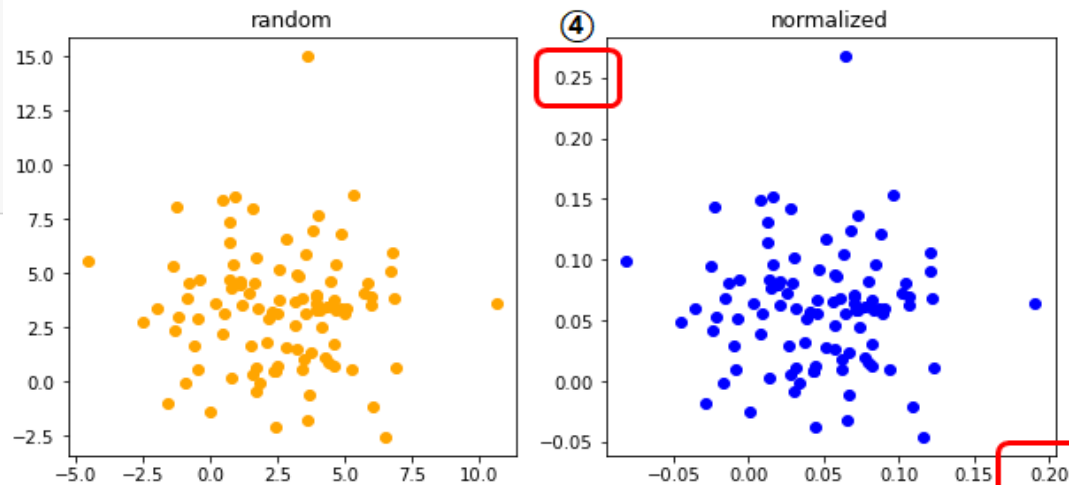
```
x = np.random.normal(3, 2.5, size=(2, 100))
```

③ 가우시안 분포 데이터 생성

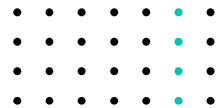
```
x_norm = normalize(x) # 데이터 정규화
```

```
plt.figure(figsize=(10,10))
plt.subplot(221)
plt.scatter(x[0], x[1], color="orange")
plt.title("random")
plt.subplot(222)
plt.scatter(x_norm[0], x_norm[1], color="blue")
plt.title("normalized")
```

정규화(Normalization)



인공신경망 튜닝 I 비용함수



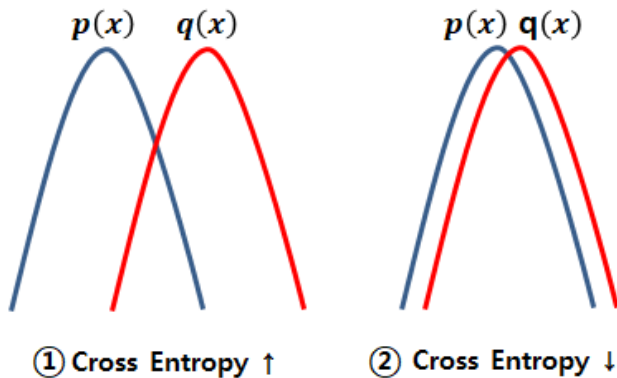
교차 엔트로피

교차 엔트로피
(Cross Entropy)

• 이산 확률 분포 $H(p, q) = - \sum_{x \in X} p(x) \log q(x)$

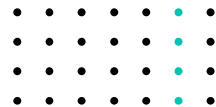
• 연속 확률 분포 $H(p, q) = - \int_x p(x) \log q(x)$

- 엔트로피: 정보를 최적으로 인코딩하기 위해 필요한 비트 수
- 교차 엔트로피: 두 확률 분포가 틀릴 수 있는 정보 양을 고려해서 최적으로 표현되기 위해 필요한 정보량

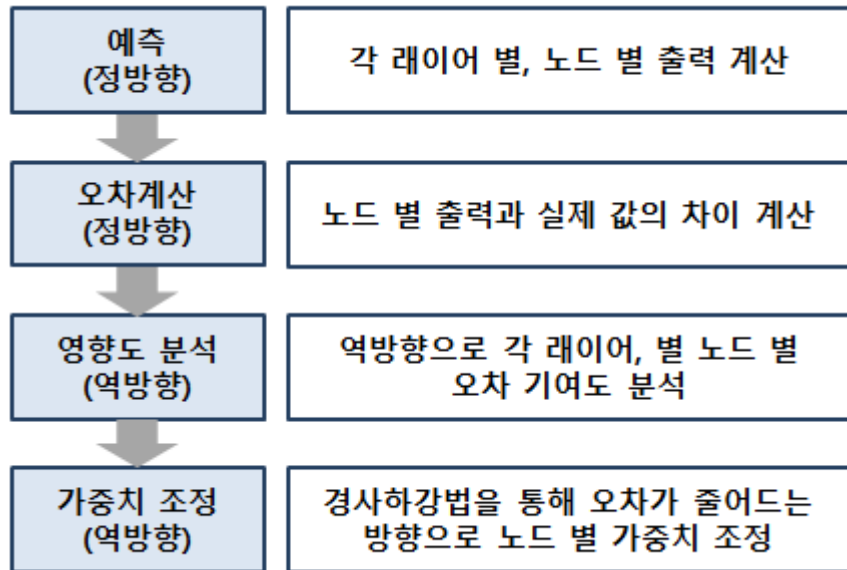


인공신경망 튜닝 I

활성화 알고리즘



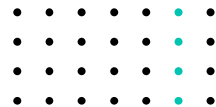
오차 역전파 과정



오차 역전파 (Error Backpropagation): 오차를 계산해서 오차가 줄어드는 방향으로 네트워크를 거꾸로 올라가면서 가중치를 조정하는 과정



인공신경망 튜닝 I 활성화 알고리즘



그래디언트 소실과 폭주

- 초기 설정된 가중치가 너무 작게 설정되거나 활성화 함수의 출력 구간이 입력 값보다 너무 작게 구성되어 있으면 기울기 소실 현상이 발생
- 가중치 초기 값이 너무 크게 설정되어 있거나 활성화 함수의 출력 구간이 너무 크게 구성되어 있으면 기울기 폭주 현상이 발생

기울기 소실
(Vanishing Gradient)

입력 층으로 갈수록 기울기가 급격히 감소하는 현상

기울기 폭주
(Exploding Gradient)

입력 층으로 갈수록 기울기가 급격히 폭증하는 현상



인공신경망 튜닝 I

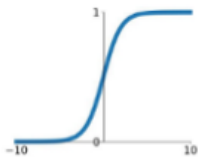
활성화 알고리즘



다양한 활성화 함수

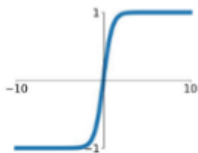
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



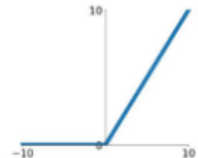
tanh

$$\tanh(x)$$



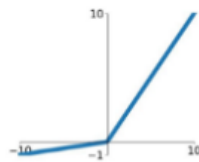
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

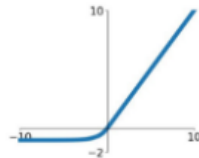


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



오차 역전파 (Error Backpropagation): 오차를 계산해서 오차가 줄어드는 방향으로 네트워크를 거꾸로 올라가면서 가중치를 조정하는 과정

