PCA
(Principle
Component Analysis)

2017010698
수학과 오서영

지도 학습 (Supervised Learning) : 답지 달린 시험 족보를 주고 학습

-〉 답을 찾기 위해 활용, labeling O

비지도 학습 (Unsupervised Learning) : 답을 맞히는 목적 X

-〉 labeling X

1) Clustering : k - means , DB-SCAN

2) PCA

PCA

1) 이미지와 같은 고차원 데이터에서 패턴을 찾는 도구

2) 데이터를 인공 신경망에 입력하기 전의 전처리 과정에서 PCA가 사용되곤 한다.

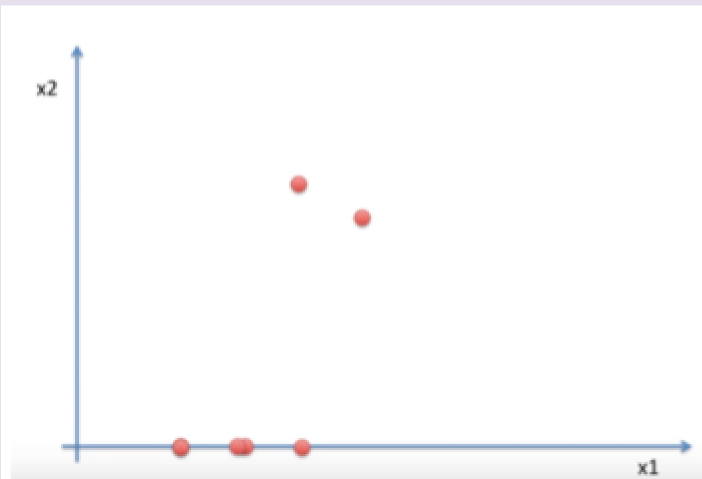
3) PCA를 통해 데이터의 범위를 재조정하고 데이터의 평균을 0으로 맞춰줌으로써 PCA는 고차원 데이터 중 중요한 차원을 골라 준다.

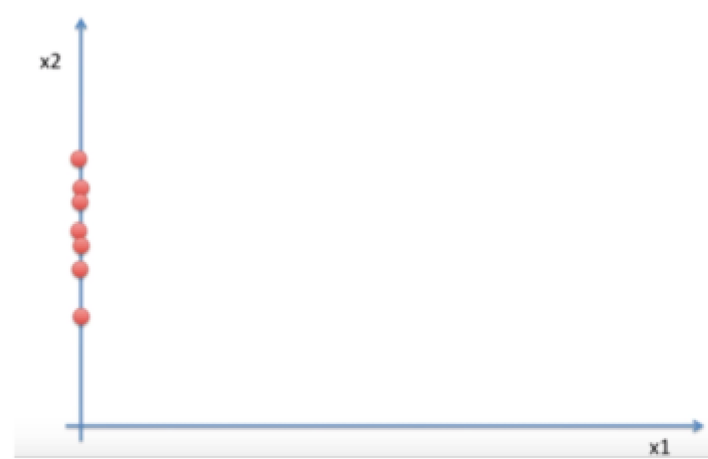
4) 신경망 학습의 수렴 속도와 성능 향상

PCA

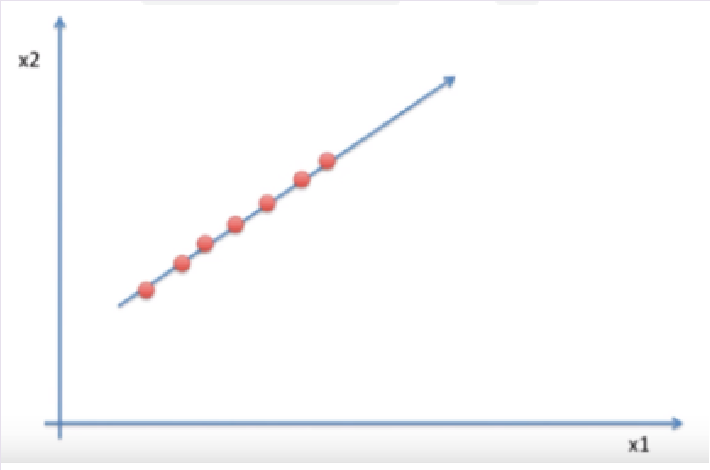
차원 축소
-〉 정사영

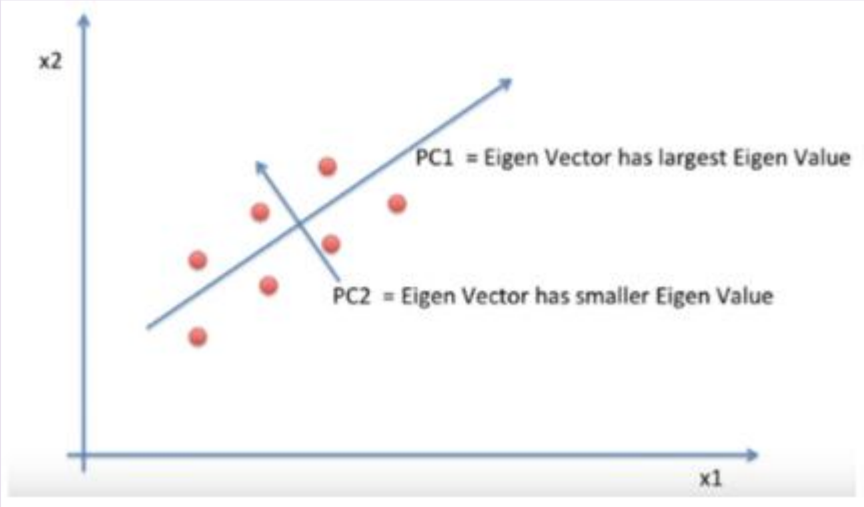
겹치는 점
-〉 정보유실





PCA





정보 유실이
가장 적게 일어나는 축
-> 데이터의 분산이
최대가 되는 축

과정

1) 학습 데이터 셋에서 **분산**이 최대인 축을 찾는다.

2) 첫 번째 축과 직교하면서 분산이 최대인 두 번째 축을 찾는다.

3) 첫 번째 축과 두 번째 축에 직교하고 분산을 최대한 보존하는 세 번째 축을 찾는다.

4) 1~3과 같은 방법으로 데이터셋의 차원(특성 수)만큼의 축을 찾는다.

-〉 i-번째 **주성분 (PC)** : i-번째 축을 정의하는 **단위 벡터**

계산

$$\begin{aligned}
Var\left[\mathbf{X}\vec{e}\right] &= \frac{1}{m-1}\sum_{i=1}^{m}\left[X\vec{e}-E\left(X\vec{e}\right)\right]^2 \\
&= \frac{1}{m-1}\sum_{i=1}^{m}\left[X\vec{e}-E(X)\vec{e}\right]^2, \quad (E(X)=0) \\
&= \frac{1}{m-1}\sum_{i=1}^{m}\left(X\vec{e}\right)^2 \\
&= \frac{1}{m-1}\left(\mathbf{X}\vec{e}\right)^T\left(\mathbf{X}\vec{e}\right) \\
&= \frac{1}{m-1}\vec{e}^T\mathbf{X}^T\mathbf{X}\vec{e} \\
&= \vec{e}^T\left(\frac{\mathbf{X}^T\mathbf{X}}{m-1}\right)\vec{e}, \quad \left(\frac{\mathbf{X}^T\mathbf{X}}{m-1}=\mathbf{C}\right) \\
&= \vec{e}^T\mathbf{C}\vec{e}
\end{aligned}$$

X : 평균을 0으로 조정한 데이터

e : 투영할 축의 단위벡터

계산

$$\text{maximize} \quad \vec{e}^T \mathbf{C} \vec{e}$$

$$\text{s.t.} \quad \|\vec{e}\|^2 = 1$$

$$L(\vec{e}, \lambda) = \vec{e}^T \mathbf{C} \vec{e} - \lambda \left(\vec{e}^T \vec{e} - 1 \right)$$

$$\begin{aligned} \frac{\partial L}{\partial \vec{e}} &= \left(\mathbf{C} + \mathbf{C}^T \right) \vec{e} - 2\lambda \vec{e} \\ &= 2\mathbf{C}\vec{e} - 2\lambda\vec{e} = 0 \end{aligned}$$

$$\therefore \mathbf{C}\vec{e} = \lambda\vec{e}$$

$$\therefore \mathbf{C} = \vec{e}\lambda\vec{e}^T$$

$$\mathbf{C}\vec{e} = \lambda\vec{e}$$

e : 공분산 C의 단위벡터

Lambda :
(C의 고유값)
= (eigenvector로 투영했을 때의 **분산)**

예시