고유값 분해를 하는 이유

데이터가 2차원에서 1차원으로 바꾸려면 데이터가 손실이되는데

중요한 데이터(고유값)은 손실되면 안되기 때문에 중요한 데이터를 유지하기 위해 사용

데이터가 손실이 되는 이유는 예를 들어 공이라는 데이터를 한정된 주머니에 넣어야 하는데 한정된 공간이기 때문에

공이 찌그러져 넣어짐 --> 공이 찌그러지는 것은 데이터가 손실되는것, 공이라는 성질 자체는 고윳값

Feature selection

Frature extraction : 데이터에서 유의미한 데이터를 추출

Feature selection이 쓰이는 경우 - 의미있는 feature를 찾는 것(예를 들어 WEKA를 통해 결과값으로 의미있는 데이터 추출)

Feature extraction이 쓰이는 경우 - 데이터를 1차원으로 뿌렸을 때 하나의 선으로 표현이됨 1차원의 데이터를 가지고 사용

PCA의 목적 - 100차원을 2차원으로 줄일 때 분산(Varience)이 가장 큰 벡터를 찾고자 한다. (레이블을 축소하기 위해 사용됨)

Singular value decomposition (SVD) - 고윳값 행렬 분해와 같은 행렬 대각화 방법 중 하나

Optimizer 원래 state에서 최적점을 찾아 한칸씩 나아가는 것

Gradient descent 미분가능한 conven function의 optimum point를 찾는 방법 중 하나

Stochastic gradient descent(SGD)

데이터 양이 많은 경우 잘게 잘라서 돌리는 것(mini-batch)

장점 - 큰 데이터를 다루기 용이

단점 – local minima에 빠지기 쉬움, step size 기준을 정하기 어려움, 매번 초기화

Momentum

Gradient descent를 통해 해를 찾는 과정 중 관성을 부여

현재 gradient로 이동하는 것과 별개로 과거 이동방향을 이용

장점 – SGD가 oscillation 방황현상을 겪는 경우 좋은 효과를 얻을 수 있음

단점 – 진짜 해를 찾았을 때에도 관성 때문에 놓칠 수 있다.

Parameter에 대해 저장해야 할 양이 늘어남 🡪 필요한 메모리 커짐

Nesterov accelerated gradient(NAG)

현재 위치에서의 gradient와 momentum step을 독립적으로 계산 후 취합

Actual step 계산 시 momentum step이 영향을 끼침

Momentum에서 관성 작용시, 멈춰야 할 시점에 멈추지 못할 가능성을 인지하고 멈춰줌

Adaptive dradient(Adagrad)

변수 update 시점마다 step size를 다르게 설정

기울기가 변화가 많지 않은 변수들은 step size를 크게

기울기가 변화가 많은 변수들은 step size를 작게

장점 – learning rate를 수동으로 조절할 필요가 없음

단점 –분모에 제곱된 기울기를 축적🡪update되는 값이 점점 작아짐🡪결국 움직이지 않는 상황 발생

RMSProp

지금까지 축적된 기울기에 감마를 곱함 🡪 gradient값이 엄청 커지는 현상을 보완

Gradient 변화량의 변수간 상대적 크기 차이 유지 가능

Adadelta

Step size와 Gradient 둘다 고려

Adam(가장 좋음)

RMSProp + Momentum 동시에 2개를 고려

초반에 0에 가까워지는 bias를 없앰

Heuristic

가장 이상적인 해를 찾는 것이 아니라 현실적으로 만족할 만한 수준의 해답을 찾는 것

경험, 직관, 시행착오를 통해 만족스러운 설계로 발전시키는 과정

Metaheuristic

* Simple
* Local search process
* Combination optimization problem
* 복잡한 문제를 조금의 계산을 더해 최적 해를 찾음

Tabu search

금기를 학습시킴 (Tabu list를 만들고 list에 들어가 있는 정보는 피함)

Feasible solution :탐색 가능한 영역

Trial solutions : 시도했던 search

Tabu queue: 금기시 된 정보

Genetic algorithm 유전 알고리즘

생명체가 환경에 적응해가면서 진화하는 모습을 모방한 최적해 검색 방법

하나의 우수한 성질을 고르고 Crossover를 하고 다향성이 사라지면 안되기 때문에 Mutation을 한다.

여러 개의 chromosome 중 2개를 선택 🡪 2개를 crossover를 함 🡪 새끼 chromosome를 만듦(2개의 부모에서 우수한 성질을 가져와 만든 것) 🡪 우수한 성질만 가지고 있는 chromosome 세대 중 안 좋은 성질을 갖고 있는 것을 섞어 넣음(Mutation)(다양성을 확보하기 위해 안 좋은 성질을 섞어 넣는 것) 🡪 점점 우수한 성질을 가지고 있는 세대가 남게 된다.

\*매번 할 때 마다 성능이 다를 수 있다.

Traveling Salesman Problem(TSP)

Generic algorithm의 하나의 예시

출발점에서 시작해 모든 노드를 단 한번만 지나 출발점으로 돌아오는 최단 경로를 구하는 문제