AI알고리즘

발견적 해법

수퍼스타를 키우는 곳
전주대학교

발견적 해법의 개념

다양한 발견적 해법

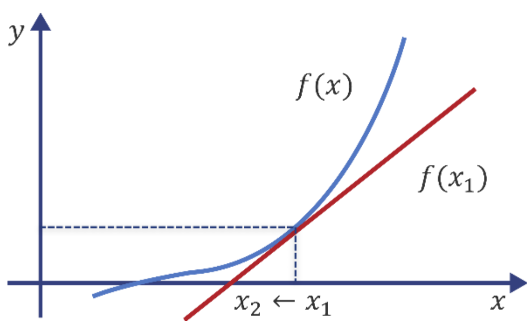
다양한 발견적 해법

❖ 발견적 해법(Heuristic Method)

- 문제자체의 복잡성 때문에 최적해를 구하는 의사결정 기법의 개발이 매우 어려움
- 최적해는 아니더라도 상대적으로 짧은 시간 내에 적절한 수준의 의사결정을 내릴 수 있는 기법

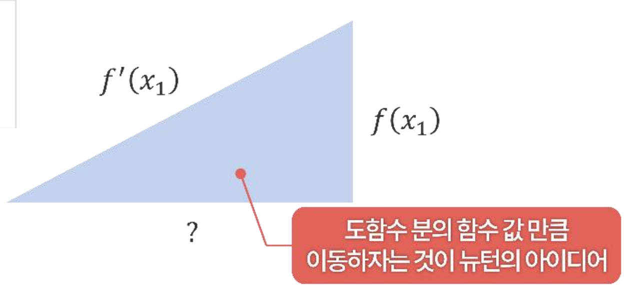
❖ 뉴턴법(Newton's Method)

- 방정식 f(x)=0의 해를 근사적으로 찾을 때 유용하게 사용되는 방법
  - 임의의 값 x=a를 넣고 f(a)의 값을 구함
  - 만일 f(a)>0이고 f'(a)>0라면 f(x)=0이 되는 x는 a보다 작은 값일 것
- 얼마나 값을 줄여야 하는가?
- 기울기를 구해보는 것
  - 미분값이 양수일 때 작은 쪽으로 움직임
- 현재 x값에서 접선을 그리고 접선이 x축과 만나는 지점으로 x를 이동시켜 가면서 점진적으로 해를 찾는 방법



- 알고리즘
  - 아무 값이나 초깃값 x1에서 시작
  - 다음 수식에 따라 수렴할 때까지 계속 x를 이동

$$x^{t+1} = x^t - \frac{f(x^t)}{f'(x^t)}$$



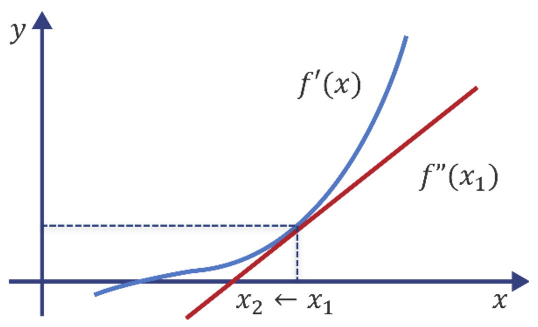
  - 종료
    - x값 변화가 거의 없을 때
    - 즉 $|x^{t+1} - x^t|$가 매우 작은 값이면 종료

다양한 발견적 해법

❖ 뉴턴법(Newton's Method)

- 특징
  - 해가 있더라도 뉴턴법으로 해를 못 찾을 수 있음
  - f(x)가 연속이고 미분가능해야 한다는 조건도 필요
  - 만일 f(x)=0인 해가 여러 개 있다면 뉴턴법은 그중 하나의 해를 찾아줄 뿐
  - 또한 해가 여러 개인 경우에는 초깃값 x1을 어떻게 주느냐에 따라서 뉴턴법으로 찾아지는 해가 달라질 수 있음
  - 초깃값 선택
    - 초깃값을 잘못 주면 시간이 오래 걸리거나 해를 찾지 못할 수도 있음
    - 먼저 일정한 간격으로 x 값을 변화시키면서 함숫값의 변화를 본 후 함숫값의 부호가 바뀌는 구간마다 보간법(Interpolation)으로 초깃값을 잡는 것
- 활용
  - f(x)의 최솟값 또는 최댓값을 구할 때 활용
    - 가장 좋은 후보군이 될 수 있는 극값
    - 기울기가 0인 점들을 찾을 때 활용 가능
    - 일반적으로 함수는 극점에서 최댓값 또는 최솟값을 가짐
    - f'(x)=0인 x를 뉴턴법으로 구한 후 f(x)에 대입하면 f(x)의 최댓값, 최솟값을 구할 수 있음

$$x^{n+1} = x^n - \frac{f'(x)}{f''(x)}$$



다양한 발견적 해법

❖ 테일러 급수 또는 테일러 전개

- 어떤 미지의 함수 f(x)를 근사 다항함수로 표현하는 것

$$f(x) \cong f(a) + f'(a)(x-a) + \frac{f''(x)}{2}(x-a)^2 + \cdots + \frac{f^n(a)}{n!}(x-a)^n$$

  - X=a 근처에서만 성립, x가 a에서 멀어지면 오차가 커짐
  - 잘 모르거나 복잡한 함수를 다루기 쉽고 이해하기 쉬운 다항함수로 대체시키기 위함

- 테일러 정리(Taylor's Theorem)
  - n번 미분가능한 함수 f(x)에 대해, 다음을 만족하는 실함수 h(X)가 반드시 존재

$$f(x) = f(a) + f'(a)(x-a) + \cdots + \frac{f^n(a)}{n!}(x-a)^n + h_n(x)(x-a)^n$$

$$\lim_{x \to a} h_n(x) = 0$$

    - 테일러 정리(Taylor's Theorem)는 어떤 함수를 유한 차수(n차)의 다항함수로 근사 시킬 수 있는 수학적 근거를 제공

❖ 최소자승법(Least Square Method)

- 어떤 모델의 파라미터를 구하는 한 방법으로서 데이터와의 $residual^2$의 합을 최소화하도록 모델의 파라미터를 구하는 방법
- 추측의 에러를 적게 하는 것
- $\sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n}\left(y_i - f(x_i)\right)^2$를 최소화
- F(x)가 직선 ax+b인 경우, $\sum_{i=1}^{n}(y_i - ax_i - b)^2$를 최소화하는 a, b 결정
- 어떤 기준을 가지고 모델의 파라미터를 구하는가를 말해줄 뿐 실제로 어떻게 계산하는가는 별개의 문제
  - 대수적 방법
    - 행렬식 형태로 표현한 후에 선형대수학을 적용하는 방법
  - 해석학적 방법
    - 식을 각각의 모델 파라미터들로 편미분한 후에 그 결과를 0으로 놓고 연립 방정식을 푸는 것

다양한 발견적 해법

❖ Gradient Descent 방법(Steepest Descent 방법)

- 미분의 개념을 최적화 문제에 적용한 대표적 방법 중 하나
- 함수의 Local Minimum(Maximum)을 찾는 방법 중 하나
- Gradient Ascent 방법
  - 어떤 함수의 극대점을 찾기 위해 현재 위치에서의 Gradient 방향으로 이동해 가는 방법
- Gradient Descent 방법
  - 극소점을 찾기 위해 Gradient 반대 방향으로 이동해 가는 방법

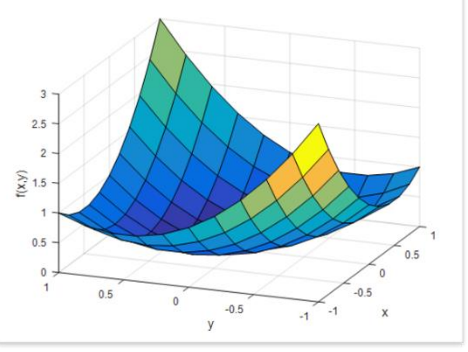
❖ 그레디언트(Gradient)

- 다변수 함수 f(x1,x2,...,xn)이 있을 때 f의 그레디언트(Gradient)

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n}\right)$$

- 각 변수로의 일차 편미분 값으로 구성되는 벡터
- $f(x,y) = x^2 + xy + y^2$
  - f의 그레디언트(Gradient)
  - (1,1)에서 f값이 최대로 증가하는 방향은 (3,3), 기울기는 ‖(3,3)‖=*sqrt*(18)

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) = (2x + y, 2y + x)$$



다양한 발견적 해법

❖ Gradient Descent 방법(Steepest Descent 방법)

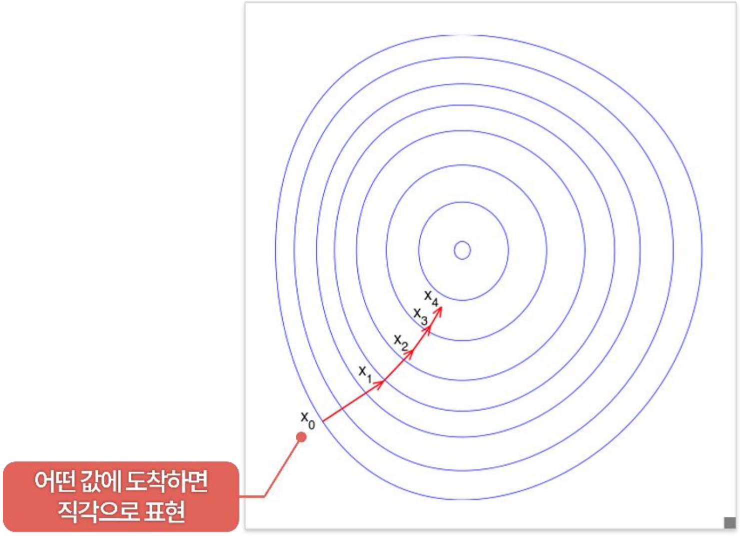
- 그레디언트의 특성을 이용하여 어떤 비용함수의 값을 최소화시키기 위한 파라미터 값을 아래와 같이 점진적으로 찾는 방법

$$x_{k+1} = x_k - \lambda_k \nabla f(x_k), \qquad k \geq 0$$

- 마이너스를 붙이고, 편미분 방향으로 작게 이동
- 즉, 어떤 초깃값 x_0부터 시작하여 위 식에 따라 Gradient 반대 방향으로 x를 조금씩 이동시키면 f(x)가 극소가 되는 x를 찾을 수 있다는 방법

❖ 그레디언트(Gradient)

- λ_k는 알고리즘의 수렴 속도를 조절하는 파라미터로서 Step Size 또는 Learning Rate



다양한 발견적 해법

❖ Gradient Descent 방법(Steepest Descent 방법)

- 문제점
  - Local Minimum에 빠지는 것
  - 해에 근접할수록 |∇f|가 0에 가까워지기 때문에 수렴 속도가 느려진다는 것
  - λ의 크기를 키움 → 수렴 속도에 문제
  - 알고리즘이 발산할 수 있는 문제점이 있음
- Gradient Descent 방법
- 뉴턴법(Newton's Method)
- 가우스-뉴턴법