

경사하강법과 뉴턴-랩슨법

- 스팸 메일 분류 로지스틱 회귀를 중심으로 -

대건고등학교 20928 정준영

I. 서론	2
II. 기계학습과 로지스틱 회귀	3
1. 기계학습	3
1.1. 지도학습	3
1.2. 비지도학습	3
1.3. 강화학습	3
2. 로지스틱 회귀	3
III. 경사하강법과 뉴턴-랩슨법	4
1. 경사하강법(steepest descent)	4
2. 뉴턴-랩슨법(Newton-Raphson method)	5
IV. 로지스틱 회귀를 이용한 스팸 메일 분류	6
1. 용어 정리	6
2. 코드	6
3. 결과	7
V. 결론	8
VI. 참고문헌	9

I. 서론

최근 인공지능에 관심을 가지게 되어 여러 학습법과 알고리즘에 관해 공부하던 중 로지스틱 회귀에 경사하강법과 뉴턴랩슨법이 사용되며 이 두 알고리즘의 원리가 미분이란 것을 알게 되어 이에 관해 탐구하고자 이번 탐구를 진행하게 되었다. 또한 이를 더욱 자세히 알아보고 직접 사용해보기 위해 요즘 문제가 되고있는 스팸 메일(문자)을 분류하는 인공지능을 경사하강법과 뉴턴랩슨법을 사용한 로지스틱 회귀를 통해 구현해 보았다.

II. 기계학습과 로지스틱 회귀

1. 기계학습

기계학습이란 다수의 파라미터로 구성된 모델을 주어진 데이터로 최적화하여 이 학습한 내용을 토대로 특정 작업을 수행하는 것이다. 학습 데이터의 형태에 따라 크게 지도학습, 비지도학습, 강화학습으로 구분된다.¹⁾²⁾

1.1. 지도학습

지도학습은 입력 데이터에 맞는 출력 정답이 미리 주어지는 학습 방법을 말한다. 컴퓨터는 이 입력 데이터와 그것의 출력(정답)을 모두 학습하여 훈련 데이터에 없던 새로운 상황도 처리할 수 있도록 일반화를 한다. 사람이 직접 각 데이터마다 정답을 달아 줘야 하므로 인건비와 시간이 많이 든다는 단점이 있다. 글자 인식이나 이미지 인식 등에 사용된다.³⁾ 로지스틱 회귀는 지도학습의 일종이다.

1.2. 비지도학습

비지도학습은 입력 데이터는 있지만, 정답은 알려주지 않는 학습 방법이다. 컴퓨터가 직접 데이터를 이용하여 정답을 유추하며 학습이 진행된다. 따라서 사람이 직접 정답을 입력해줄 필요가 없어 많은 데이터를 적은 인건비로 학습할 수 있지만, 컴퓨터가 직접 정답을 유추하므로 그 유추한 것이 실제 정답이 아닐 수 있다는 단점이 있다. 대표적인 비지도 학습으로 학습된 인공지능에는 ChatGPT가 있다.⁴⁾

1.3. 강화학습

강화학습은 현재의 상태에서 어떤 행동을 취하고 새로운 상태와 보상을 받으며 최종적으로 가장 큰 보상을 얻는 방법을 탐색하며 학습하는 방법이다. 체스 인공지능 등에 사용된다.⁵⁾

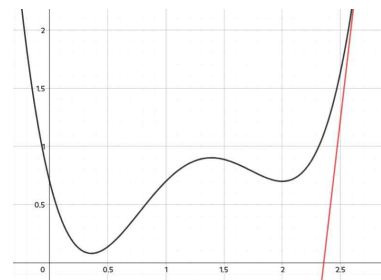
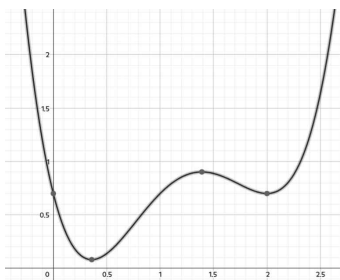
2. 로지스틱 회귀

로지스틱 회귀(또는 로지스틱 회귀 분석)이란 특정 데이터가 어떠한 범주에 속할 확률을 0~1 사이의 값으로 예측한 뒤 특정 기준치에 따라 그 이상인 경우 특정 클래스로 분류하는 지도학습 알고리즘 중 하나이다. 이때 클래스는 ‘참/거짓’, ‘Yes/No’ 등 이진적이다. 컴퓨터는 데이터들의 특징(스팸 단어)을 추출하여 리스트화 하고 특징과 가중치를 곱하고 시그모이드 함수(또는 로지스틱 함수)(실수 전체를 정의역으로 가지고 함수값으로 유한한 구간 사이의 한정된 값(주로 0~1)만을 반환하는 함수. 함수식은 $S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$ 이다⁶⁾)에 넣어 각각의 데이터의 확률을 계산한다. 이때 가중치가 학습 성능에 큰 영향을 미치는데 처음에는 랜덤값(주로 0)으로 설정되지만 학습을 진행하며 손실에 맞춰 점차 조정된다. 이때 이를 조절하는 방법들을 최적화 알고리즘이라 하며 이 최적화 알고리즘 중 경사하강법과 뉴턴랩슨법이 있다.⁷⁾⁸⁾

III. 경사하강법과 뉴턴-랩슨법

1. 경사하강법(steepest descent)

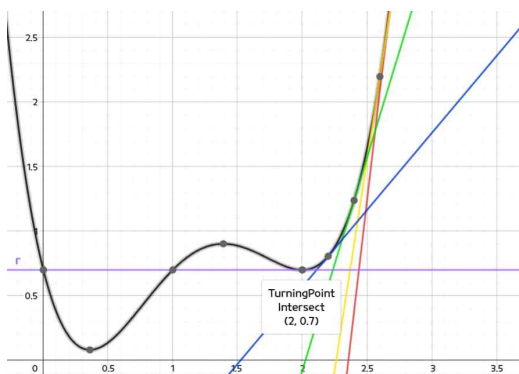
경사하강법은 함수의 최솟값을 빠르게 찾는 최적화 이론 기법이다. 경사하강법은 기본적으로 먼저 임의의 시작점과 기울기 보폭(학습률)을 설정한 다음 학습률 만큼 기울기가 낮아지는 방향으로 이동하여 함수의 지역최솟값(즉 극소값)을 찾는다. 이때 학습률이 너무 클 경우 최솟값을 넘어서는 과대적합이, 너무 작을 경우 시간이 오래걸리며 최솟값에 다가가지 못하는 과소적합 문제가 발생할 수 있다.⁹⁾ 따라서 더 많이 이동할수록 학습률을 점점 낮추게 된다. 이 과정을 알아보기 위해 간단한 사차함수를 만들어봤다. $f(x) = x^4 - 5x^3 + 8x^2 - 4x + 0.7$. 이 함수의 그래프를 지오지브라



(GeoGebra)를 사용하여 그려보았다. 이제 이 함수의 극소를 경사하강법으로 찾아보겠다. 시작점을 $x=2.7$ 로 잡으면 기울기는 $f'(x) = 4x^3 - 15x^2 + 16x - 4$, $f'(2.7) = 8.582$ 이다. $x=2.7$ 에서의 접선을 구해보면 $y - f'(2.7) = f'(2.7)(x - 2.7) = 8.582x - 20.2223$ 이다. 그래프는 위의 우측 그래프와 같다.

학습률을 0.1이라 정하면 다음 x 점은 기울기가 낮아지는 방향인 음의방향으로 0.1 이동하여 $x=2.6$ 이다. 이를 계속 반복해보겠다.(x 가 짝수인 경우만 울프람알파(WolframAlpha)로 계산해보았다)

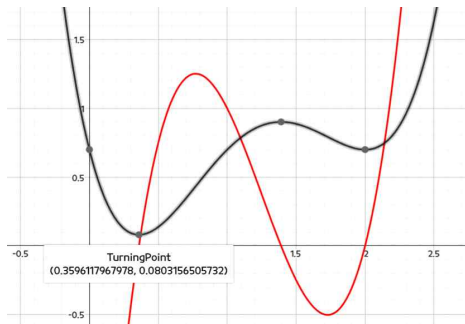
<p>Input interpretation</p> <p>tangent line to $y = x^4 - 5x^3 + 8x^2 - 4x + 0.7$ at $x = 2.6$</p> <p>Result</p> <p>$y = 6.504x - 14.7128$</p>	<p>Input interpretation</p> <p>tangent line to $y = x^4 - 5x^3 + 8x^2 - 4x + 0.7$ at $x = 2.4$</p> <p>Result</p> <p>$y = 3.296x - 6.6728$</p>	<p>Input interpretation</p> <p>tangent line to $y = x^4 - 5x^3 + 8x^2 - 4x + 0.7$ at $x = 2.2$</p> <p>Result</p> <p>$y = 1.192x - 1.8168$</p>	<p>Input interpretation</p> <p>tangent line to $y = x^4 - 5x^3 + 8x^2 - 4x + 0.7$ at $x = 2$</p> <p>Result</p> <p>$y = 0.7$</p>
---	--	--	--



사차함수 $f(x)$ 는 $x=2$ 에서 극솟값 0.7을 가진다는 것을 알 수 있다. 이렇게 경사하강법은 매우 복잡한 함수(가령 차수가 만단위를 넘어가는)에서도 극소를 찾을수 있게 해주어 매우 많은 파라미터가 있는 머신러닝 그중에서도 특히 딥러닝에서 사용된다.¹⁰⁾

2. 뉴턴-랩슨법(Newton-Raphson method)

뉴턴-랩슨법은 미분을 통하여 함수의 해의 근삿값을 빠르게 구하는 방법이다. 임의의 x_0 을 정하고 근사하여 $x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$ 로 식을 잡은 다음 $n=1$ 부터 n 을 1씩 늘려가면 $f(x_n)$ 은 0에, x_n 은 근에 수렴하게된다.¹¹⁾ 뉴턴랩슨법도 앞서서와 같이 예시를 들어보겠다. 함수는 앞서 사용한 함수 $F(x) = x^4 - 5x^3 + 8x^2 - 4x + 0.7$ 을 사용한다. 이번엔 $x=2$ 가 아닌 극소를 찾아보겠다. 먼저 극소를 해를 이용하여 찾기위해 $F(x)$ 를 미분하면 $f(x) = 4x^3 - 15x^2 + 16x - 4$ 이다. 함수의 그래프는 아래와 같다.



우선 초기값 x_0 을 찾기위해 사잇값정리를 사용해보았다. 연속함수 $f(x)$ 에서

Input interpretation	Input interpretation
$4x^{(3)} - 15x^{(2)} + 16x - 4$ where $x = 0$	$4x^{(3)} - 15x^{(2)} + 16x - 4$ where $x = 1$
Result	Result
-4	1

이 나오므로 열린구간 $(0, 1)$ 에서 해가 존

재한다. 초기값 x_0 을 해와 가장 가까운 정수인 1로 정해보자. 그리고 계산을 위해 미리 $f(x)$ 를 미분하면 $f'(x) = 2(6x^2 - 15x + 8)$ 이다. 이제 n 을 1부터 늘려가며 x_n 의 값을 관찰해보자. 답은 0.3596117967978이다. 값은 파이썬으로 코드를 작성하여 계산해보았다. 코드는 [깃허브\(new-reap.py\)](#)에 있다.

- $n = 1 : x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0 - \frac{-4}{16} = 0.25$
- $n = 2 : x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 0.3445945945945946$
- $n = 3 : x_3 = 0.3592679811662522$
- $n = 4 : x_4 = 0.3596116102109729$
- $n = 6 : x_6 = 0.3596117967977924$
- $n = 7 : x_7 = 0.3596117967977924$

5번만에 해에 완전히 수렴하였다. 이렇듯 뉴턴랩슨법을 사용하면 차수가 매우 큰 함수에서도 빠르게 해의 근삿값을 구할 수 있다.¹²⁾

IV. 로지스틱 회귀를 이용한 스팸 메일 분류

1. 용어 정리

들어가기에 앞서 로지스틱 회귀에 사용할 여러 개념과 용어들에 대해 간단히 설명하겠다.

- 유니그램, 바이그램: 각각 한 개의 단어와 연속된 두 개의 토큰(단어)로 이루어진 시퀀스이다¹³⁾
- L2정규화: 벡터의 각 요소들의 제곱합의 제곱근으로 벡터의 크기를 조절하여 가중치를 조절하여 과적합을 방지하는 방법¹⁴⁾
- 에포크(Epoch): 기계학습에서 학습을 돈 데이터 사이클 횟수
- 손실: 실제값과 예측값의 차이
- 조기 종료: 손실이 개선되지 않을 시 즉 너무 많이 학습하여 과대적합(극소(가장 손실이 낮을 때)를 넘어서서 다시 손실이 증가하는 현상)이 일어났을 때 학습을 종료하는 방법
- 정확성: 전체 예측값 중 실제 값과 일치하는 비율
- 정밀도: 예측한 결과 클래스가 실제 정답 클래스인 비율
- 재현율: 실제 정답 중 정답으로 올바르게 예측한 비율
- F1점수: 정밀도와 재현율의 조화 평균으로 학습된 모델의 성능을 평가¹⁵⁾

2. 코드

경사하강법을 이용한 것과 뉴턴랩슨법을 사용한 코드를 따로 만들었다. 설명을 달아놓은 주석은 뉴턴랩슨법 코드에 달아놓았으며 경사하강법에만 있는 코드에 대해서는 따로 경사하강법 코드에 주석을 달아놓았다. 전체코드는 [깃허브](#)에 업로드 해두었다.

- [경사하강법\(spam_steepest.py\)](#)
- [뉴턴랩슨법\(spam_newton.py\)](#)

3. 결과

출력 결과는 아래와 같다. (상: 경사하강법 / 하: 뉴턴랩슨법)

```
Epoch 0, Loss: 0.6929790937064584
Epoch 1000, Loss: 0.6289814942844518
Epoch 2000, Loss: 0.6188518565955138
Epoch 3000, Loss: 0.616556203799559
Epoch 4000, Loss: 0.6159738548270719
Epoch 5000, Loss: 0.6158155362937997
Epoch 6000, Loss: 0.6157703703063092
Epoch 7000, Loss: 0.6157568731686157
Early stopping at epoch 7965
정확성 : 0.7949
정밀도 : 0.6145
재현율 : 1.0000
F1 점수 : 0.7612
1. "win big prizes": Spam
2. "your today schedule": Normal
3. "고객센터 문의 결과": Normal
4. "무료 투자 강의": Spam
5. "hello?": Normal
6. "안녕하세요": Normal
7. "무료 선물": Spam
8. "당신의 컴퓨터가 해킹되었습니다": Normal
9. "지금 시작 시 무료 코인 지급": Spam
10. "무료 생방송 수업을 예약하세요": Spam
11. "채팅방 입장 시 코스닥 상장 확정 종목 공개": Spam
12. "70% 할인 마감 1일 전": Spam
13. "Your computer has been hacked": Spam
14. "Report for exam": Normal
15. "Hello teacher": Normal
16. "new clothes discount only today": Spam
```

```
Iteration 0, Loss: 0.5035487986148134
Iteration 10, Loss: 0.4536221020172711
Iteration 20, Loss: 0.453377414246838
Iteration 30, Loss: 0.45331479997592655
Iteration 40, Loss: 0.4532863916023145
Iteration 50, Loss: 0.4532702120412736
Iteration 60, Loss: 0.45325977423664837
Iteration 70, Loss: 0.4532524861264506
Iteration 80, Loss: 0.45324711054324557
Iteration 90, Loss: 0.4532429827522426
정확성 : 0.7949
정밀도 : 0.6145
재현율 : 1.0000
F1 점수 : 0.7612
1. "win big prizes": Spam
2. "your today schedule": Normal
3. "고객센터 문의 결과": Normal
4. "무료 투자 강의": Spam
5. "hello?": Normal
6. "안녕하세요": Normal
7. "무료 선물": Spam
8. "당신의 컴퓨터가 해킹되었습니다": Normal
9. "지금 시작 시 무료 코인 지급": Spam
10. "무료 생방송 수업을 예약하세요": Spam
11. "채팅방 입장 시 코스닥 상장 확정 종목 공개": Spam
12. "70% 할인 마감 1일 전": Spam
13. "Your computer has been hacked": Spam
14. "Report for exam": Normal
15. "Hello teacher": Normal
16. "new clothes discount only today": Spam
```

두 방법 모두 예측 결과와 정확성(0.7949), 정밀도(0.6145), 재현율(1.0000), F1점수(0.7612)은 동일하며 손실(loss)는 경사하강법이 0.615..., 뉴턴랩슨법이 0.453...으로 다르다. 또한 경사하강법에선 7965번째 에포크(Epoch)에서 조기종료가 일어난 것을 알 수 있다. 학습속도는 뉴턴랩슨법(약2초)이 경사하강법(약7초)보다 약3.5배 빨랐다.

V. 결론

로지스틱 회귀에서 최솟값(극소)을 찾기위해 사용되는 알고리즘인 경사하강법과 뉴턴-랩슨법에 대해 알아보고 이를 사용하여 스팸 메일 분류 인공지능을 만들어본 후 경사하강법과 뉴턴랩슨법 끼리 비교를 해보았다. 뉴턴랩슨법은 빠르게, 경사하강법은 비교적 느리지만 정확하게 극소를 찾을 수 있다는 것을 조사하였고 실제로도 속도면에서 뉴턴랩슨법이 경사하강법에 비해 더 빠르단 것을 직접 확인하였다.

어려운 개념들이 많아 이해를 하는데 꽤 힘들었지만 직접 함수식을 만들고 이의 극소를 경사하강법과 뉴턴랩슨법을 통해 구해보는 과정을 통해 이해를 빠르게 하고 이를 실제로 활용하여 로지스틱 회귀에서 어떻게 사용되는지를 스팸 메일 분류기를 만들어보며 알게되어 의미있고 흥미로운 탐구였다 생각한다.

이번 탐구를 통해 배운 것을 토대로 더욱 심화된 수학적/컴퓨터공학적 주제를 탐구하고 더 많은 인공지능을 경사하강법과 뉴턴랩슨법으로 빠르고 효율적으로 학습해보고싶다.

IV. 참고문헌(각주와 미주로 대체)

- 1) 기계학습, 위키피디아
- 2) 기계 학습, 네이버 지식백과, IT용어사전
- 3) 지도학습, 위키피디아
- 4) 비지도 학습, 나무위키
- 5) 강화학습, 위키피디아
- 6) 시그모이드 함수, AI용어사전
- 7) 로지스틱 회귀, AI용어사전
- 8) 로지스틱 회귀분석, 데이터 사이언스 스쿨
- 9) 경사하강법, AI용어사전
- 10) 경사하강법, 신경망 학습 방법 | 2장, 딥러닝, 3Blue1Brown, Youtube, 1:50 ~ 3:00, 2017. 10.
- 11) 뉴턴의 방법, 수학백과
- 12) Newton's Method, The Organic Chemistry Tutor, Youtube, 2018. 03.
- 13) n-그램, 위키피디아
- 14) L2 normalization (L2 정규화, L2 노멀라이제이션), 위키독스
- 15) F1 점수: 머신러닝 모델의 성능을 평가하는 강력한 지표, MozenWorld, 티스토리