핵심 머신러닝 - 4

생성자	㊊ 재환 김
태그	머신러닝

"학습 알고리즘 심층 분석"

- **손실 함수(Loss Function)**학습 알고리즘의 핵심 요소 중 하나입니다.모델의 예측값과 실제값 사이의 차이를 수치화하는 함수로, 이 차이를 최소화하는 것이 학습의 목표입니다.대표적인 손실 함수로는 평균 제곱 오차(MSE)와 교차 엔트로피(Cross-Entropy) 등이 있으며, 모델 유형에 따라 적절한 함수를 선택합니다.
- **최적화 기준(Optimization Criterion 또는 비용 함수(Cost Function))**손실 함수를 최소화하도록 모델 파라미터를 조정하는 방법을 정의하는 기준입니다.주로 경사 하강법(Gradient Descent) 같은 알고리즘을 사용하며, 이때 학습률(learning rate)이 중요한 역할을 합니다.최적화 과정은 손실 함수의 값을 반복적으로 줄여 모델의 데이터 학습 능력을 향상시키는 것입니다.
- **최적화 루틴(Optimization Routine)**학습 데이터의 패턴을 최대한 활용해 최적화 기준에 맞는 해답을 찾는 과정입니다.이 루틴은 반복 계산을 통해 모델 성능을 개선하며, 데이터 특성에 맞는 최적화 방법 선택이 중요합니다.대표적인 최적화 알고리즘으로는 확률적 경사 하강법(SGD)과 모멘텀 기법 등이 있습니다.

경사 하강법(Gradient Descent)

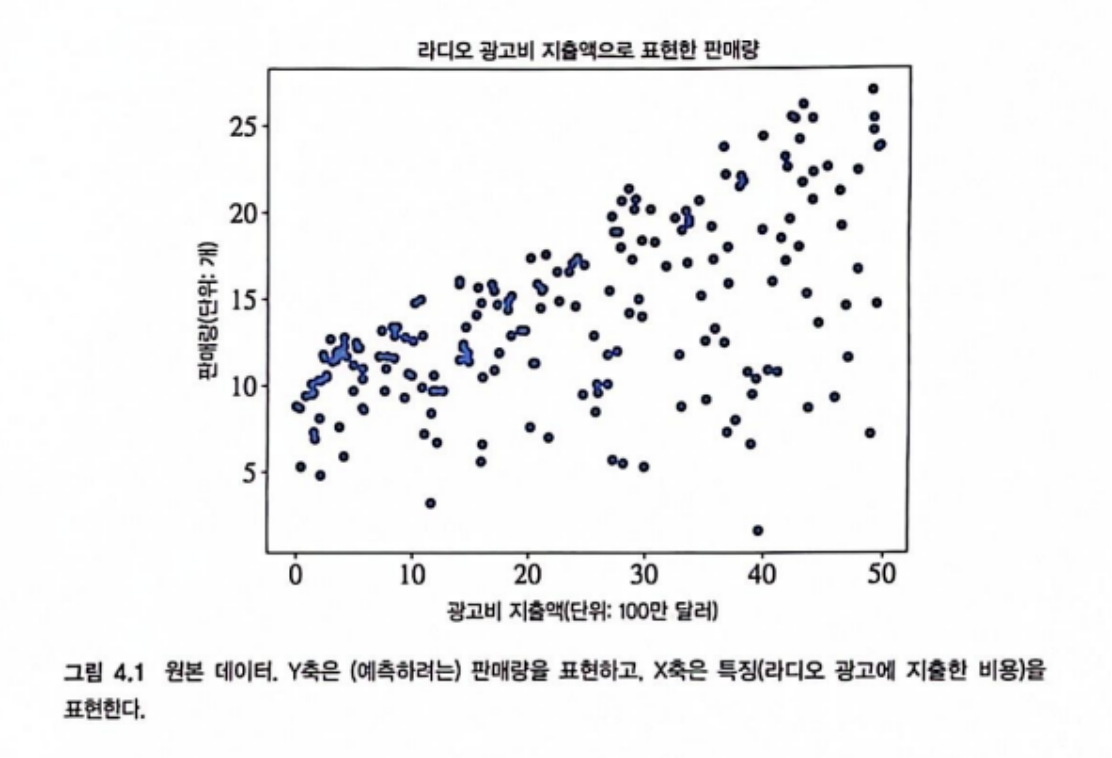

그림 4.1 원본 데이터. Y축은 (예측하려는) 판매량을 표현하고, X축은 특징(라디오 광고에 지출한 비용)을 표현한다.

1. **기본 개념**

   경사 하강법은 손실 함수의 값을 최소화하기 위해 기울기가 가장 가파른 방향으로 반복적으로 파라미터를 조정하는 최적화 방법입니다. 기울기가 가파른 방향은 손실 함수의 값이 가장 빠르게 감소하는 방향으로, 이를 통해 최적의 파라미터를 찾을 수 있습니다. 경사 하강법은 선형 회귀, 로지스틱 회귀, SVM 등의 기계 학습 모델에서 널리 사용되며, 특히 손실 함수가 볼록(convex) 함수일 때 효율적으로 작동합니다. 이때 목표는 전역 최소값(global minimum) 또는 충분히 좋은 지역 최소값(local minimum)을 찾는 것입니다.

2. **구체적인 사례**

   위의 그래프는 라디오 광고비 지출액과 판매량 간의 관계를 보여줍니다. X축은 광고비 지출액(단위: 100만 달러), Y축은 판매량을 나타냅니다. 그래프는 라디오 광고비 지출이 판매량에 긍정적인 영향을 미치고 있음을 시사합니다.

3. **실제 데이터 설명**

   표는 네 개 회사의 라디오 광고 지출액과 그에 따른 판매량을 나타냅니다.

   - 첫 번째 회사: 378만 달러 지출, 221개 판매
   - 두 번째 회사: 393만 달러 지출, 10.4개 판매
   - 세 번째 회사: 459만 달러 지출, 9.3개 판매
   - 네 번째 회사: 413만 달러 지출, 18.5개 판매

회사	지출액(단위: 100만 달러)	판매량(단위: 개)
1	37.8	22.1
2	39.3	10.4
3	45.9	9.3
4	41.3	18.5
..	..	..

이 데이터는 경사 하강법을 적용할 수 있는 예시로, 광고비와 판매량 간의 관계를 모델링하기 위한 기초 자료입니다. 경사 하강법을 사용하여 이 데이터를 기반으로 최적의 광고비 지출액을 도출할 수 있습니다.

경사 하강법의 적용

경사 하강법을 통해 반복적으로 파라미터를 조정하여 최적의 값을 찾아가는 과정을 설명합니다.

각 변수는 경사 감소법을 통해 지속적으로 업데이트되며, 다음과 같은 방식으로 파라미터가 정의됩니다:

- 1차원 데이터에서는 $w^{(1)}$,
- 2차원 데이터에서는 $w^{(2)}$,
- 3차원 데이터에서는 $w^{(3)}$로 각각 나타납니다.

이 과정은 파라미터를 업데이트하는 방식이며, 손실 함수의 기울기를 계산하여 점진적으로 최적값에 수렴하게 됩니다.

경사 하강법(Gradient Descent)의 개념

경사 하강법은 손실 함수(Loss Function)를 최소화하기 위한 최적화 방법입니다. 이 방법은 기울기(Gradient)의 방향을 따라 점진적으로 최적의 파라미터를 찾아갑니다. 알고리즘은 주로 다음 단계로 구성됩니다:

1. **초기 파라미터 설정**

   모델의 파라미터 값을 임의로 초기화합니다. 이 파라미터들은 최적화하고자 하는 변수들로, 예를 들어 회귀 모델에서는 각 독립 변수에 대응하는 가중치가 될 수 있습니다.

2. **기울기 계산**

   현재 파라미터 값에서 손실 함수의 기울기를 계산합니다. 이 기울기는 손실 함수의 변화율을 나타내며, 양수일 때는 파라미터 값을 감소시키고, 음수일 때는 증가시킵니다.

3. **파라미터 업데이트**

기울기를 사용하여 파라미터 값을 업데이트합니다. 이때 업데이트되는 방향은 기울기가 작아지는 방향, 즉 손실 함수의 값이 줄어드는 방향입니다.

$$w_{\text{new}} = w_{\text{old}} - \eta \cdot \frac{\partial L}{\partial w}$$

여기서 w_{new}는 새로운 파라미터 값, w_{old}는 이전 파라미터 값, η는 학습률(Learning Rate), $\frac{\partial L}{\partial w}$는 손실 함수의 기울기입니다.

학습률 η는 파라미터가 업데이트되는 정도를 결정하며, 너무 크면 수렴하지 않고, 너무 작으면 수렴하는 데 시간이 오래 걸릴 수 있습니다.

4. **반복**

   위의 과정을 손실 함수가 수렴할 때까지 반복합니다. 수렴은 기울기가 0에 가까워져 파라미터 값이 더 이상 크게 변하지 않는 시점을 의미합니다.

이 과정은 손실 함수가 볼록 함수(convex function)일 때 특히 효과적입니다. 볼록 함수에는 하나의 전역 최소값(global minimum)이 존재합니다. 반면, 비볼록(non-convex) 함수에는 여러 개의 지역 최소값(local minimum)이 존재할 수 있어, 경사 하강법이 전역 최소값이 아닌 지역 최소값에 수렴할 수 있습니다.

경사 하강법의 종류

1. **배치 경사 하강법(Batch Gradient Descent)**전체 데이터셋을 한 번에 사용하여 손실 함수의 기울기를 계산하는 방식입니다.안정적으로 수렴할 수 있다는 장점이 있지만, 데이터가 많을 경우 계산 비용이 매우 커질 수 있습니다.
2. **확률적 경사 하강법(Stochastic Gradient Descent, SGD)**각 데이터 포인트에 대해 개별적으로 기울기를 계산하고, 그때마다 파라미터를 업데이트합니다.계산 비용이 적고 빠르게 수렴할 수 있지만, 최적값 주변에서 진동할 가능성이 큽니다.
3. **미니배치 경사 하강법(Mini-batch Gradient Descent)**배치 경사 하강법과 확률적 경사 하강법의 절충안입니다.데이터셋을 작은 배치로 나누어 각 배치마다 기울기를 계산하고 파라미터를 업데이트합니다.계산 효율성과 수렴 안정성을 동시에 얻을 수 있습니다.

선형 회귀 모델

주어진 데이터의 관계를 나타내기 위해 사용되는 선형 회귀 모델은 다음과 같은 식으로 표현됩니다:

$$f(x) = wx + b$$

여기서:

- w는 기울기(회귀 계수),

- b는 절편(바이어스 항),
- x는 입력 변수(여기서는 광고비 지출)를 의미합니다.

이 모델의 목표는 w와 b 값을 조정하여 예측 값과 실제 데이터 사이의 오차를 최소화하는 것입니다. 이 오차는 손실 함수(Loss Function)를 통해 측정됩니다. 여기서는 평균 제곱 오차(Mean Squared Error, MSE)를 손실 함수로 사용하고 있으며, 다음과 같이 정의됩니다:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - (wx_i + b))^2$$

여기서:

- y_i는 실제 값(판매량),
- x_i는 입력 값(광고비 지출),
- N은 데이터의 개수입니다.

경사 감소법을 사용한 파라미터 업데이트

경사 감소법의 핵심은 손실 함수의 기울기를 계산하고 이를 사용해 파라미터 w와 b를 반복적으로 업데이트하는 것입니다. 이를 위해 손실 함수에 대한 각 파라미터의 편미분을 구해야 합니다.

w에 대한 편미분

$$\frac{\partial L}{\partial w} = \frac{1}{N}\sum_{i=1}^{N} -2x_i(y_i - (wx_i + b))$$

이 식은 파라미터 w에 대한 손실 함수의 기울기를 나타냅니다. 이는 w를 얼마나 조정해야 손실 함수의 값이 줄어드는지를 나타내므로, 기울기의 반대 방향으로 파라미터를 업데이트해야 합니다.

b에 대한 편미분

$$\frac{\partial L}{\partial b} = \frac{1}{N}\sum_{i=1}^{N} -2(y_i - (wx_i + b))$$

이 식은 b에 대한 기울기로, 마찬가지로 손실 함수를 최소화하기 위해 b를 업데이트하는 데 사용됩니다.

파라미터 업데이트 공식

경사 감소법을 사용하여 파라미터를 업데이트하는 과정은 다음과 같은 공식에 따라 이루어집니다:

$$w = w - \alpha\frac{\partial L}{\partial w}$$
$$b = b - \alpha\frac{\partial L}{\partial b}$$

여기서 α는 학습률(Learning Rate)로, 기울기를 얼마나 크게 반영할지를 결정합니다. 학습률이 너무 크면 수렴하지 않을 수 있고, 너무 작으면 학습 속도가 매우 느려질 수 있습니다.

에포크(Epoch)와 반복 과정

경사 하강법은 데이터를 여러 번 학습하면서 반복적으로 파라미터를 업데이트합니다. 각 반복을 "에포크(Epoch)"라고 합니다. 한 에포크는 전체 데이터셋을 사용하여 손실 함수의 기울기를 계산하고 파라미터를 한 번 업데이트하는 과정입니다.

파라미터가 업데이트될 때마다 손실 함수의 값은 점진적으로 감소해야 합니다. 이 과정은 손실 함수 값이 충분히 작아질 때까지 반복됩니다.

경사 하강법의 변형 및 개선 방법

이 절에서는 경사 하강법을 변형하거나 개선하는 다양한 방법을 설명합니다.

- **미니배치 확률적 경사 하강법(SGD)**: 기본 경사 하강법의 변형으로, 전체 데이터 대신 미니배치(일부 데이터)를 사용해 계산 속도를 높이고 근사치를 구합니다.
    - 미니배치 사용으로 메모리 효율성이 향상되고, 계산 속도가 빨라지며, 대규모 데이터셋에도 적합합니다.
- **Adagrad**: 각 파라미터의 학습률을 개별적으로 조정하여 점진적으로 줄여가는 방식입니다. 학습 초기에는 빠르게 진행되다가 점차 속도가 감소하며, 과도하게 큰 학습률을 방지합니다.
- **Momentum**: 기울기 변화 방향을 반영하여 학습 수렴 속도를 높입니다. 기울기 변동을 완화하고 오버슈팅을 방지하여 더 안정적인 학습을 가능케 합니다.
- **RMSprop과 Adam**: 최근 널리 사용되는 경사 하강법 개선 알고리즘입니다.
    - **RMSprop**은 Adagrad의 학습률 감소 문제를 해결하기 위해 학습률을 적절히 조정합니다.
    - **Adam**은 Momentum과 RMSprop의 장점을 결합하여 경사 하강법의 속도와 정확도를 동시에 개선합니다.

결론적으로, 이러한 변형 방법들은 경사 하강법의 수렴 속도를 개선하고 더 안정적인 학습을 가능하게 합니다.

머신 러닝 엔지니어의 작업 방식

이 절에서는 머신 러닝 엔지니어가 일반적으로 사용하는 작업 방식과 도구를 설명합니다.

- **사이킷런(Scikit-learn)**: 파이썬 기반 머신 러닝 엔지니어들 사이에서 널리 쓰이는 대표적인 라이브러리입니다. 사이킷런은 높은 안정성과 효율성을 자랑하며, 선형 회귀, 로지스틱 회귀, SVM, KNN 등 다양한 알고리즘을 지원합니다.

- **간단한 선형 회귀 예시**: 예제 코드는 사이킷런을 활용해 선형 회귀 모델을 간단히 구현하는 방법을 보여줍니다.

학습 알고리즘 사용 시 주의사항

이 섹션에서는 학습 알고리즘 사용 시 주의해야 할 점과 특정 상황에서의 적용 방법을 설명합니다.

- **범주형 변수 처리**: 학습 알고리즘에서 범주형 변수는 핵심 요소입니다. 이러한 변수를 처리할 때는 '원핫 인코딩(one-hot encoding)' 같은 기법을 사용해 숫자형 데이터로 변환한 후 학습에 적용해야 합니다.
    - 예를 들어, 색상이나 날씨 같은 범주형 변수는 숫자로 변환해야 머신 러닝 모델이 처리할 수 있습니다.
- **SVM과 KNN의 특성**: SVM(서포트 벡터 머신)은 클래스 간 경계를 명확히 구분하고, 새로운 데이터의 클래스를 예측하는 데 효과적입니다. 반면, KNN(k-최근접 이웃 알고리즘)은 데이터를 군집화하고, 새로운 데이터와 주변 데이터의 유사성을 기반으로 분류합니다.
- **트리 기반 모델의 강점**: 의사결정 트리와 같은 모델은 데이터 분류에 매우 효과적입니다. 특히 데이터의 특정 특성에 따라 명확한 분류 기준을 설정할 수 있어 유용합니다.

SVM과 KNN 분류 모델

- **SVM(서포트 벡터 머신)**: 클래스 간 경계를 구분하는 알고리즘입니다.
    - SVM은 고차원 공간에서 데이터를 분리하는 최적의 초평면을 찾습니다.
    - 이 알고리즘은 경계에 가장 가까운 데이터 포인트(서포트 벡터)를 중심으로 학습하여 새로운 데이터의 클래스를 예측합니다.
- **KNN(k-최근접 이웃 알고리즘)**: 새로운 데이터 포인트의 가장 가까운 'k'개 이웃을 참조하여 분류하는 알고리즘입니다.
    - KNN은 주변 데이터 포인트들의 클래스를 기반으로 새로운 데이터를 분류합니다.
    - 이 방법은 별도의 학습 과정 없이 예측 시 전체 데이터셋을 탐색하므로, 대규모 데이터셋에서는 계산 비용이 높을 수 있습니다.

로지스틱 회귀와 결정 트리

- **로지스틱 회귀(Logistic Regression)**: 이진 분류 문제에 주로 사용되는 알고리즘입니다.
    - 입력 데이터를 바탕으로 특정 클래스에 속할 확률(0과 1 사이의 값)을 예측합니다.

    - SVM이나 결정 트리와 달리, 클래스 소속 확률을 직접 제공하여 결과 해석이 용이합니다.
- **결정 트리(Decision Tree)**: 트리 구조를 사용하여 데이터를 분류하는 알고리즘입니다.
    - 각 노드에서 특정 특성을 기준으로 데이터를 분할하여, 의사 결정 과정을 직관적으로 이해할 수 있습니다.
    - 복잡한 비선형 관계를 학습하는 데 효과적이며, 데이터의 특성을 명확히 보여줍니다.

모델 학습 방식

- **모델 학습 방식**: 이 절에서는 모델의 학습 과정을 설명합니다.
    - **로지스틱 회귀와 결정 트리**는 데이터셋 전체를 한 번에 학습하여 모델을 생성합니다. 이후 새로운 데이터 예측 시 이 모델을 활용합니다.
    - 반면, **SVM과 KNN** 알고리즘은 예측 시마다 데이터를 새로 분석합니다. 따라서 새로운 데이터 추가 시 기존 모델을 재구성해야 할 수 있습니다.

반복 학습 알고리즘

- **SGD(SGDClassifier, SGDRegressor)**: 경사 하강법을 사용하는 이 알고리즘은 데이터를 반복적으로 처리하여 모델을 학습합니다.
    - 이 방식은 데이터 추가 시 모델을 쉽게 업데이트할 수 있습니다.
    - **PassiveAggressiveClassifier/Regressor** 역시 데이터 추가 시 모델을 반복적으로 학습하는 유사한 방식으로 작동합니다.

일반적인 라이브러리와 모델 종류

- 다양한 머신 러닝 라이브러리는 각기 다른 방식으로 작동하는 알고리즘을 제공합니다. 데이터 특성에 따라 적절한 알고리즘을 선택할 수 있습니다.
    - 예를 들어, **나이브 베이즈(Naive Bayes)**, **멀티레이어 퍼셉트론(Multilayer Perceptron, MLP)**, **SGD**, **Passive-Aggressive 모델** 등이 있으며, 각 알고리즘은 특정 데이터 유형에 더 적합할 수 있습니다.