|  |  |  |  |
| --- | --- | --- | --- |
| **2023-1 머신러닝 및 실습** | | | |
| 학번 | 2021136150 | 이름 | 양동재 |

Logistic Regression은 분류 문제에서 가장 기본적으로 사용되는 알고리즘 중 하나로, 주어진 입력 데이터가 어떤 클래스에 속하는지 확률을 추정하는 이다. 이번 보고서에서는 Python과 NumPy를 이용하여 Logistic Regression을 구현하고, 이를 이용하여 classification problem을 해결하는 방법에 대해 상세하게 다룹니다. 구체적으로는 MNIST 데이터셋을 이용하여 Logistic Regression 모델을 학습하고, 테스트 데이터셋에서 모델의 성능을 평가하고 분류 결과를 분석한다. 이를 통해 Logistic Regression 알고리즘의 이론적인 부분뿐만 아니라 실제 구현과 평가 과정을 경험할 수 있다.

**[데이터]**

MNIST 데이터셋은 70,000개의 손글씨 숫자 이미지 데이터셋으로, 각 이미지는 해당 숫자와 함께 라벨링되어 이다. 이미지 분류 작업의 벤치마크 데이터셋으로 널리 사용됩니다. 데이터셋의 이미지는 28x28 크기의 그레이스케일로, 중앙에 위치해 있다. 이 데이터셋은 60,000개의 훈련 이미지와 10,000개의 테스트 이미지로 구성됩니다. MNIST 데이터셋은 로지스틱 회귀, 신경망, 서포트 벡터 머신을 포함한 다양한 머신러닝 알고리즘의 성능을 평가하는 데 사용된다.

Shape, arrow

Description automatically generated

**[문제1]**

1. **문제정의**

이번 실습 문제는 Logistic Regression 알고리즘을 구현하고, MNIST 데이터셋을 활용하여 0이 맞는지 아닌지를 판단하는 single class classification 문제를 해결하는 모델을 개발하는 것 이다. 해당 문제를 해결하기 위해서는 전반적인 기계학습 과정에 대한 이해와 Linear Regression에 대한 수학적 이해와 sigmoid 함수 등의 개념에 대한 이해, 그리고 classification과 regression의 명확한 구분이 필요하다. 이를 통해 MNIST 데이터셋의 이미지와 라벨을 이용하여 모델을 학습시키고, 새로운 이미지가 들어왔을 때 해당 이미지가 0인지 아닌지 정확하게 분류할 수 있도록 하는 것이 목표이다.

1. **학습전략**

**A. Hypothesis**

문제 1에서는 MNIST 데이터셋에서 입력 데이터가 0인지 아닌지를 구분하는 single class classifier를 logistic regression function을 사용하여 구현해야한다. 이를 위해서는 logistic regression의 동작 원리를 명확히 이해해야 한다.

기존의 linear regression에서 사용되는 cost function은 연속적인 값을 예측하는 문제를 해결하는데 적합한 cost function이다. 하지만 single class classification 문제에 적용하기에는 다소 문제가 있다. 이는 linear regression 모델의 결과값이 입력값과의 차이를 계산하는 방식으로, 음수부터 1보다 큰 값을 반환할 수 있기 때문이다. 반면 logistic regression은 activation function을 사용하여 0과 1과 같이 이산적인 값을 출력하므로 해당 문제를 해결하기 위해서는 logistic regression cost function을 사용해야 한다.

**B. EDA**

본 데이터셋의 특징은 2차원(Gray scaling Image) 28x28이미지 데이터이다. 단, 본 **문제는 데이터가 0의 손글씨에 해당하는 데이터만 사용한다.**

1. **자신이 구현한 함수 및 클래스에 대한 설명 또는 유도 과정 작성**

Hypothesis에서 설명한 바와 같이, classification을 진행하기 위해서는 loss function을 통해 나온 결과값이 probability 즉 0~1의 값을 갖는 discrete value여야 한다. 따라서 우리는 linear regression의 loss function에 결과값을 sigmoid function에 적용시켜-logistic regression- 0과 1 사이의 값을 결과로 반환받고, thresholding을 통해 특정 값을 넘기면 True, 그렇지 않으면 False라고 반환하여 classification문제를 해결한다.

Single class classifier Cost Function

Gradient Descent

Cross entropy

The result of single class classifier

**4. 결과 비교 및 분석**

Chart, line chart

Description automatically generatediteration을 반복할 수록 cost function의 값은 감소하며 감소함에 따라 Accuracy가 증가함을 확인할 수 있으며,

이미지에 대한 분류도 잘 해내는 것을 확인할 수 있다.

**[문제2]**

1. **문제정의**

이번 실습 문제는 Logistic Regression 알고리즘을 구현하고, MNIST 데이터셋을 활용하여 다양한 순자들을 분류하는 multi class classification 문제를 해결하는 모델을 개발하는 것 이다. 위에서 요구했던 역량과 더해 **ensemble** 모델에 대한 이해가 필요하다. 해당 배경 지식들을 통해 MNIST 데이터셋의 이미지와 라벨을 이용하여 모델을 학습시키고, 새로운 이미지가 들어왔을 때 해당 이미지가 어떤 순자인지 아닌지 정확하게 분류할 수 있도록 하는 것이 목표이다.

1. **학습전략**

**A. Hypothesis**

문제 2에서는 MNIST 데이터셋에서 입력 데이터가 무슨 숫자 손글씨인지를 구분하는 multi class classifier를 logistic regression function을 사용하여 구현해야한다. 이를 위해서는 logistic regression의 동작 원리를 명확히 이해해야 한다.

기존의 linear regression에서 사용되는 cost function은 연속적인 값을 예측하는 문제를 해결하는데 적합한 cost function이다. 하지만 single class classification 문제에 적용하기에는 다소 문제가 있다. 이는 linear regression 모델의 결과값이 입력값과의 차이를 계산하는 방식으로, 음수부터 1보다 큰 값을 반환할 수 있기 때문이다. 반면 logistic regression은 activation function을 사용하여 0과 1과 같이 이산적인 값을 출력하므로 해당 문제를 해결하기 위해서는 logistic regression cost function을 사용해야 한다.

추가적으로 해당 문제를 해결하기 위해서는 logistic regression cost function의 결과가 어느 임계값을 넘느냐에 따라서 True, False를 결정해주는 hyper parameter threshold 값을 정해야한다. 우리는 본 과제 수행을 위해 0.5초과로 hyper parameter threshold값을 고정하여 문제를 해결했다.

**B. EDA**

본 데이터셋의 특징은 2차원(Gray scaling Image) 28x28이미지 데이터이다. 단, 본 **문제는 데이터가 0~9의 손글씨에 해당하는 데이터를사용한다.**

1. **자신이 구현한 함수 및 클래스에 대한 설명 또는 유도 과정 작성**

Hypothesis에서 설명한 바와 같이, classification을 진행하기 위해서는 loss function을 통해 나온 결과값이 probability 즉 0~1의 값을 갖는 discrete value여야 한다. 따라서 우리는 linear regression의 loss function에 결과값을 sigmoid function에 적용시켜-logistic regression- 0과 1 사이의 값을 결과로 반환받고, thresholding을 통해 특정 값을 넘기면 True, 그렇지 않으면 False라고 반환하여 classification문제를 해결한다.

Hypothesis Function

Cost Function for a single training example

Gradient descent update rule

Cross entropy

위의 방법을 통해 k개의 클래스를 분류하는 k개의 모델을 만들고, 새로운 입력데이터가 들어왔을 때, 모든 모델에 입력한 후 가장 높은 확률값을 갖는 모델의 클래스로 분류하여야 한다.

The one-vs-all logistic regression formula

**4. 결과 비교 및 분석**

|  |  |  |
| --- | --- | --- |
| “CLASS” | “ACCURACY” | ENSEMBLE MODEL ACCURACY |
| Classifier class 0 | 0.850400 | 0.6835 |
| Classifier class 1 | 0.850500 |
| Classifier class 2 | 0.850200 |
| Classifier class 3 | 0.850200 |
| Classifier class 4 | 0.850600 |
| Classifier class 5 | 0.850300 |
| Classifier class 6 | 0.850100 |
| Classifier class 7 | 0.850500 |
| Classifier class 8 | 0.832400 |
| Classifier class 9 | 0.850200 |

문제 1과 같은 방식으로 10개의 logistic regression models 개발하였고, 입력 데이터를 10개의 모델에 모두 돌려본 후 제일 큰 결과값을 반환하여 predict 하는 모델을 개발하였다. 모델 별 Accuracy는 위의 표와 같으며, ensemble 모델의 경우 Accuracy가 감소하는 것을 확인할 수 있다.

일반적인 ensemble학습의 경우 성능이 좋은 classifier들을 모아서 모델을 구성하면, 이 모델은 학습 데이터에서는 높은 성능을 보일 수 있다. 그러나, 이 모델이 새로운 데이터에 대해 예측을 할 때, single class classifier들이 각각 예측한 결과를 어떻게 결합하는가에 따라 성능이 크게 좌우된다. 예를 들어, ensemble모델에서 각 single class classifier가 예측한 결과를 평균을 내어 최종 예측을 하는 경우, 가장 높은 확률을 가진 클래스가 다른 클래스와의 차이가 크지 않은 경우, 평균값으로 인해 성능이 크게 떨어질 수 있다.

또한, single class classifier들의 예측이 서로 상관관계가 높은 경우, 앙상블 모델의 성능이 크게 개선되지 않을 수 있다.

따라서, 앙상블 학습에서는 각 classifier들의 예측 결과를 효과적으로 결합하는 방법을 찾는 것이 중요하다. 또한, single class classifier들이 과적합(overfitting)되지 않도록 하고, 다양한 유형의 classifier들을 결합하여 다양성을 추가하는 것도 중요하다.

**[문제3]**

1. **문제정의**

해당 실습 문제는 이전에 구현한 멀티클래스 분류 앙상블 모델에 정규화(regularization) 개념을 적용해 성능을 개선하는 것이다. 이를 통해 머신러닝 개념 중 하나인 정규화가 어떤 의미를 가지고 어떤 기능을 수행하는지 이해 해야한다.

1. **학습전략**

기존에 개발했던 Multi class classifier 모델에 정규화 적용

머신러닝에서 정규화(regularization)는 모델이 과적합(overfitting)되는 것을 방지하기 위한 기법으로, 모델의 복잡도를 제한하거나 모델 파라미터의 값을 조절함으로써 수행된다. 일반적으로, 모델의 복잡도가 높아질수록, 훈련 데이터에 대한 예측 정확도가 향상되지만, 테스트 데이터나 새로운 데이터에 대한 예측 성능은 떨어지는 경우가 많다.

가장 일반적인 정규화 방법으로는 L1 정규화(Lasso regularization)와 L2 정규화(Ridge regularization)가 있다.

L1 정규화는 모델의 가중치(weight) 중 일부를 0으로 만드는 특성이 있어 특성 선택(feature selection)의 역할을 할 수 있다.

L2 정규화는 가중치의 크기를 제한하고 과적합을 방지하는 데 효과적이다.

또한, 드롭아웃(Dropout)이라는 정규화 방법도 있다. 드롭아웃은 신경망에서 무작위로 일부 뉴런을 제거하여 과적합을 방지한다.

정규화의 장점은 과적합을 방지하고 일반화 성능을 향상시킨다는 것이다. 그러나 정규화를 적용하면 모델의 복잡도가 낮아지기 때문에, 모델이 학습할 수 있는 것의 범위가 제한되어 일부 중요한 특성이 누락될 수 있다.

또한, **정규화 하이퍼파라미터를 적절하게 조절하지 않으면 모델의 성능이 저하될 수 있어 본 문제 해결을 통해 적합한 하이퍼 파라미터를 찾아내는 것 또한 중요한 과제중 하나이다.**

1. **자신이 구현한 함수 및 클래스에 대한 설명 또는 유도 과정 작성**

Regularization

L1 regularization

L2 regularization

Cost function applied in regularization

L1 regularization method

L2 regularization method

1. **결과 비교 및 분석**

L2 정규화에서 하이퍼 파라미터 람다가 커질수록 가중치(weight)의 크기를 작게 만드는 효과가 강해진다. 이는 모델이 더 일반화(generalization)될 수 있도록 도와준다. 일반화란 모델이 이전에 본 적 없는 새로운 데이터에 대해 더 잘 예측할 수 있는 능력을 의미한다. 따라서 L2 정규화에서는 람다의 크기를 조정하여 모델의 복잡도를 조절할 수 있다. 람다가 커질수록 모델의 가중치 값이 작아지기 때문에, 모델이 과적합(overfitting)을 방지할 수 있어 일반화 능력이 향상된다.

1. **고찰 및 느낀점**

* None Linear problem을 해결하기 위해서는 activation function을 통해 0~1의 확률값으로 전환할 필요가 있다.
* Regularization은 가중치의 값이 변화함에 따라 모델이 더 일반화되어 다양한 새로운 데이터에 대해 더 잘 예측할 수 있도록 돕는다. 이는 중요한 의미를 갖습니다. 일반적으로 머신러닝 문제에서는 데이터 확보가 어렵기 때문에 cross-validation 방법을 통해 모델의 성능을 평가한다. 하지만 이 방식은 이미 보유한 데이터에 대해서만 성능평가를 진행하기 때문에, 실험적 데이터의 성능평가 결과와 실제 서비스에서의 성능이 동일하지 않을 수 있다. 따라서 regularization을 통해 가중치를 세밀하게 조정함으로써, 일반화된 모델을 구축하여 실 서비스에서도 잘 적용될 수 있도록 하는 것이 기계학습의 궁극적 목표이다.
* Signle classifier를 여러게 합쳐서 만든OvH(One-vs-ALL)방식은 아래의 문제점이 있음을 확인할 수 있었다
  + 클래스가 많아질수록 이진 분류 문제의 개수가 급격히 증가하여 계산 비용이 증가
  + 클래스를 구분하기 위한 경계선이 서로 다를 수 있어서, 이를 결합했을 때 클래스 간 경계가 부정확하거나 불연속적이어서 분류 성능이 저하됨
  + 클래스의 불균형 문제가 발생할 수 있다. 예를 들어, 특정 클래스에 대한 데이터가 적으면, 해당 클래스에 대한 이진 분류 문제에서 정확한 학습이 이루어지지 않을 수 있다.
* 실험 결과 Logistic Function에 고질적인 문제를 발견하였다. Logistic Function의 경우 sigmoid 함수를 따라가는데 이 때 sigmoid함수의 경우 함수에 데이터를 입력할 때 함수의 결과값이 0과 1에 가까우면 가까울수록 다음과 같이 변화 폭이 기하급수적으로 줄어든다. 즉 gradient가 점점 소실된다는 의미이다.

