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

Softmax Regression은 다중 클래스 분류(multi-class classification) 문제에서 가장 기본적으로 사용되는 알고리즘 중 하나입니다. 이 알고리즘은 주어진 입력 데이터가 어떤 클래스에 속하는지 추정하는데 사용됩니다. 이번 보고서에서는 Python과 NumPy를 이용하여 Softmax Regression을 구현하고, 다중 클래스 분류 문제를 해결하는 방법을 상세하게 다룹니다. 구체적으로는 MNIST 데이터셋을 사용하여 Softmax Regression 모델을 학습하고, 테스트 데이터셋에서 모델의 성능을 평가하고 분류 결과를 분석합니다. 이를 통해 Softmax Regression 알고리즘의 이론적인 부분 뿐만 아니라 실제 구현과 평가 과정을 경험할 수 있습니다. 보고서는 전문적인 방식으로 작성되었으며, Softmax Regression 알고리즘의 이론과 응용에 대한 깊은 이해를 제공합니다.

**[데이터]**

MNIST 데이터셋은 70,000개의 손글씨 숫자 이미지 데이터셋으로, 각 이미지는 해당 숫자와 함께 라벨링되어 있으며, 이미지 분류 작업의 벤치마크 데이터셋으로 광범위하게 활용되고 있습니다. 데이터셋의 이미지는 28x28 크기의 그레이스케일로, 중앙에 위치하며, 이 데이터셋은 60,000개의 훈련 이미지와 10,000개의 테스트 이미지로 구성되어 있습니다. MNIST 데이터셋은 머신러닝 분야에서 많은 알고리즘, 특히 로지스틱 회귀, 신경망, 서포트 벡터 머신 등의 성능 평가에 사용됩니다. 이러한 알고리즘은 MNIST 데이터셋을 통해 이미지 분류와 같은 다양한 문제를 해결하는 데 활용됩니다.

Shape, arrow

Description automatically generated

**[문제1]**

1. **문제정의**

이번 문제는 손으로 쓴 숫자 이미지를 인식하는 Multi class classification 모델을 Softmax Regression 알고리즘을 사용하여 개발하는 것입니다. 이를 위해 기계학습 과정과 Softmax Regression의 수학적 개념과 softmax 함수 등을 이해해야 하며 Softmax 함수가 어떻게 Multi class classification 문제를 해결하는지에 대한 알고리즘을 이해하고 있어야합니다.

이 과제를 수행하면 MNIST 데이터셋에서 숫자 이미지와 라벨을 사용하여 모델을 생성하고 학습하며, 새로운 이미지가 입력될 때 해당 이미지가 어떤 숫자인지 정확하게 분류할 수 있게 됩니다.

1. **학습전략**

**A. Hypothesis**

문제 1에서는 MNIST 데이터셋에서 입력 데이터가 어떤 숫자 손글씨인지를 구분하는 multi class classifier를 softmax regression function을 사용하여 구현해야합니다. 이를 위해서는 softmax regression의 동작 원리를 명확히 이해해야 합니다.

지난 실습에서 사용된 Logistic regression의 cost function은 하나의 클래스를 구분하는 문제를 해결하는 데 적합합니다. 그러나 multi class classification 문제에는 적합하지 않습니다. 이는 logistic regression 모델의 결과값이 이진 형태로 나타나기 때문입니다. 반면 softmax regression은 다양한 클래스에 대해 각각의 확률을 구할 수 있으므로, 해당 문제를 해결하기 위해서는 softmax regression cost function을 사용해야 합니다. 이를 통해 각 클래스별 확률을 계산하고, 가장 높은 확률을 가진 클래스를 예측값으로 선택할 수 있습니다.

**B. EDA**

본 데이터셋의 특징은 2차원(Gray scaling Image) 28x28이미지 데이터입니다.

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

Hypothesis에서 설명한 바와 같이, Multi class classification을 효율적으로 진행하기 위해서는 loss function을 통해 나온 결과값이 probability 즉 0~1의 값을 갖는 discrete value여야 합니다. 따라서 linear regression의 loss function에 결과값을 softmax function에 적용시켜-softmax regression- multi class에 대해서 0과 1 사이의 값을 결과로 반환받고, thresholding을 통해 특정 값을 넘기면 True, 그렇지 않으면 False라고 반환하여 classification문제를 해결해야합니다.

Softmax function

Gradient Descent

Cross entropy

Chain rule

**4. 결과 비교 및 분석**

Chart, histogram

Description automatically generatediteration을 반복할 수록 cost function의 값은 감소하며 감소함에 따라 Accuracy가 증가함을 확인할 수 있으며,이미지에 대한 분류도 잘 해내는 것을 확인할 수 있었습니다. 특히 이번에 새롭게 적용된 Data separate을 통해 Model의 overfitting과 underfitting 여부를 확인할 수 있었습니다.

모델의 경우 Over, under fitting없이 문제를 잘 해결한 것을 확인할 수 있었으며 iteration의 증가에 따른 loss의 감소로 보아 학습 또한 문제없이 잘 수행된 것을 확인할 수 있었습니다.

모델의 성능 ACC

training set : 0.7922

test set은 0.7993

**[문제2]**

1. **문제정의**

이번 과제는 Softmax Regression 알고리즘을 사용하여 손으로 쓴 숫자 이미지를 인식하는 Multi class classification 모델을 개발하는 것입니다. 이를 위해 기계학습 과정과 Softmax Regression의 수학적 개념과 softmax 함수 등을 이해해야 하며, Softmax 함수가 어떻게 Multi class classification 문제를 해결하는지에 대한 알고리즘을 이해하고 있어야합니다. 이 과제를 수행하면 MNIST 데이터셋에서 숫자 이미지와 라벨을 사용하여 모델을 생성하고 학습하며, 새로운 이미지가 입력될 때 해당 이미지가 어떤 숫자인지 정확하게 분류할 수 있게 됩니다.

**“2번 문제에서는 L2 규제를 적용하여 모델의 과적합을 방지하는 것이 목표입니다. 이를 위해 L2 규제가 무엇인지 이해하고, 모델의 손실 함수에 L2 규제 항을 추가하는 방법을 알아야합니다.”** 이 과제를 수행하면 Softmax Regression 모델을 개선하여, 더욱 정확한 숫자 이미지 분류를 수행할 수 있게 됩니다.

1. **학습전략**

**A. Hypothesis**

기존의 linear regression과 logistic regression에서 사용되는 cost function은 연속적인 값을 예측하는 문제를 해결하는데 적합한 cost function이지만, multi class classification 문제에 적용하기에는 다소 문제가 있습니다. 이는 linear regression 모델의 결과값이 입력값과의 차이를 계산하는 방식으로, 음수부터 1보다 큰 값을 반환할 수 있기 때문입니다. 따라서, 이러한 문제를 해결하기 위해 softmax regression을 사용하는데, 이는 activation function을 사용하여 각 클래스에 대한 확률값을 출력하므로 해당 문제를 적합하게 해결할 수 있습니다.

**“추가적으로, L2 정규화를 softmax regression 모델에 적용하여 overfitting 문제를 해결할 수 있습니다. L2 정규화는 모델의 가중치를 일부러 작게 만들어서 overfitting을 방지하는 방법으로, 이를 위해 cost function에 가중치의 제곱합에 비례하는 항을 추가합니다. 이 항은 모델이 큰 가중치를 가질수록 커지기 때문에, 모델이 가장 작은 가중치를 가지는 방향으로 학습됩니다.”**

**B. EDA**

본 데이터셋의 특징은 2차원(Gray scaling Image) 28x28이미지 데이터입니다.

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

L2 regularization은 과적합을 방지하기 위해 모델의 가중치(weight)가 너무 커지지 않도록 제한하는 방법입니다. 이를 Softmax Regression 모델에 적용하면, Loss function에 가중치들의 L2 norm을 더해주게 됩니다. 따라서, 모델의 손실 함수는 다음과 같이 표현됩니다.

Hypothesis Function

Cross entropy

Gradient descent update rule

Loss function

여기서 lambda는 L2 regularization의 강도를 제어하는 하이퍼파라미터입니다. 따라서, 모델의 학습 과정에서는 이 L2 regularization term이 gradient 계산에 추가되어 모델의 가중치가 너무 커지지 않도록 합니다.

Chart

Description automatically generated**4. 결과 비교 및 분석**

iteration을 반복할 수록 cost function의 값은 감소하며 감소함에 따라 Accuracy가 증가함을 확인할 수 있으며,이미지에 대한 분류도 잘 해내는 것을 확인할 수 있었습니다. 특히 이번에 새롭게 적용된 Data separate을 통해 Model의 overfitting과 underfitting 여부를 확인할 수 있었습니다.

모델의 경우 Over, under fitting없이 문제를 잘 해결한 것을 확인할 수 있었으며 iteration의 증가에 따른 loss의 감소로 보아 학습 또한 문제없이 잘 수행된 것을 확인할 수 있었습니다.

모델의 경우 L2 정규화를 통해 feature를 update하기 위핸 weight의 값을 조정하였으며 이에따라 parameter의 크기가 조정됨을 확인할 수 있었습니다.

모델의 성능 ACC

training set : 0.7922

test set은 0.7993

|  |  |
| --- | --- |
| Comparison Weight size between L2 regularization or not | |
| Chart  Description automatically generated | Chart  Description automatically generated |
| Soft max 손실 함수와 L2 regularization softmax 손실함수의 차이점은 L2 정규화 항인 (λ/2) \* ||w||^2입니다.  이 항은 가중치 벡터 w의 크기에 비례하는 값이므로, 가중치의 크기가 커질수록 손실 함수의 값도 커지게 됩니다. 이는 가중치의 크기가 너무 커지지 않도록 제한하는 효과를 가지며, overfitting을 방지하는 역할을 합니다.  여기서 ||w||^2는 가중치 벡터 w의 제곱합(norm squared)입니다. 이를 계산하기 위해서는 각 가중치를 제곱하고 합한 뒤, 2로 나누어주면 됩니다. 따라서 L2 정규화를 적용한 손실 함수에서 가중치 벡터 w의 크기는 다음과 같이 계산됩니다: ||w||^2 = ∑(i=1~n)(w\_i^2)  여기서 n은 가중치 벡터의 크기입니다.  따라서 L2 정규화를 적용한 소프트맥스 회귀 모델의 손실 함수는 가중치의 크기를 제한하는 효과를 가지므로, overfitting을 방지할 수 있으며 본 그래프는 L2 regularization이 적용된 weight의 값과 적용되지 않은 weight의 값을 iteration별로 더한 값 입니다. 전체 feature에 대해서 모두 덧셈연산을 진행했음에도 불구하고 L2 regularization 의 weight 크기가 상대적으로 작은것을 확인할 수 있습니다. | |

**5. 고찰 및 느낀점**

* Model evaluation performance trust reliability는 confusion matrix 즉, model evaluation method를 적용하기 전에 bias-variance trade-off에 대해서 고려해야한다. 여기서 bias는 model이 training data에 대해 만들어낸 prediction value와 GT의 평균 차이를 이야기하며 high bias인 경우 model training data feature를 반영하지않고 model을 oversimplification 시킨 것 이다. Variance의 경우 Trained model이 test data에 대해 만들어낸 prediction value와 GT의 차이에 대한 variability를 의미하며 high variance의 경우 학습 데이터의 locality를 지나치게 반영하여 새로운 데이터에 대한 일반화의 어려움을 겪는다.
* 따라서 우리는 total error를 minimization 하는 bias와 variance간의 balance를 찾아내는 것 이며 total error는 보통 variance + bias^2 + irreducible error로 계산한다.
* 우리는 해당 문제를 해결하기 위해서 Rgularization이라는 기법을 주로 사용하는데, L2 regularization의 경우 feature selection절차를 진행할 수 없으나, feature importance에 대해서 알 수 있으며, EDA 과정에서 우리가 활용할 수 있는 피쳐들이 모두 결과값에 유의미한 경우에 활용될 수 있다.
* Linear regression과 artificial neural netowrk의 method차이를 보면 아래와 같다.
  + Linear regression : data 🡪 y = mx + b 🡪 error function 🡪 partial differentiation to find optimal m and b 🡪 coefficient of determination
  + AI : training data 🡪 network topology 🡪 loss function 🡪 partial differentiation to find optimal w 🡪 confusion matrix with training set