기계학습의 영상인식결과에 대한 입력영상의 영향도 분석 기법*

김도완† · 김우성† · 이은헌† · 김현철†
† 고려대학교 컴퓨터학과

Analysis Method of influence of input for Image recognition result of machine learning

Do-Wan Kim† · Woo-seong Kim† · Eun-hun Lee† · Hyeoncheol Kim†
† Dept. of Computer science and Engineering, Korea University, Seoul, South Korea

요 약

기계학습은 인공지능(AI, Artificial Intelligence)의 일종으로 다른 인공지능 알고리즘이 정해진 규칙을 기반으로 주어진 임무(Task)를 해결하는 것과는 달리, 기계학습은 수집된 Data를 기반으로 최적의 솔루션을 학습한 후 미래의 값들을 예측하거나 해석하는 방법을 사용하고 있다. 더욱이 인터넷을 통한 연결성의 확대와 컴퓨터의 연산능력 발전으로 가능하게 된 Big-Data를 기반으로 하고 있어 이전의 인공지능 알고리즘에 비해 월등한 성능을 보여주고 있다. 그러나 기계학습 알고리즘이 Data를 학습할 때 학습 결과를 사람이 해석하기에 너무 복잡하여 사람이 그 내부 구조를 이해하는 것은 사실상 불가능하고, 이에 따라 학습된 기계학습 모델의 단점 또는 한계 등을 알지 못하는 문제가 있다. 본 연구에서는 이러한 블랙박스화된 기계학습 알고리즘의 특성을 이해하기 위해, 기계학습 알고리즘이 특정 입력에 대한 결과를 예측할 때 어떤 입력들로부터 영향을 많이 받는지 그리고 어떤 입력으로부터 영향을 적게 받는지를 알아보는 방법을 소개하고 기존 연구의 단점을 개선하기 위한 방법을 제시한다.

1. 서 론

기계학습은 산업뿐만 아니라 교육, 언어, 예술 등 다양한 분야에서 괄목할만한 성장을 해오고 있다. 특히 2015년 세계의 관심사였던 알파고와 한국의 이세돌 9단과의 바둑대결은 알파고의 승리로 마무리되면서 기존 신경망 기법에서 진화한 딥러닝(Deep Learning)이 인공지능 분야에 선두주자로 자리매김을 하는 계기가 되었다.

하지만, 딥러닝을 포함한 대부분의 기계학습은 큰 단점이 있는데, 그것은 사람이 기계학습의 예측 결과에 대한 원인을 설명할 수 없다는 것이다. 사람의 경우 지식기반(Knowledge Base)의 규칙을 이용하여 특성들을 추출하고 그것을 바탕으로 예측 결과를 설명할 수 있는 반면, 기계학습은 수많은 내부 파라미터의 값들이 학습을 통해 최적화됨으로써 내부 파라미터 값의 의미를 사용자가 이해하지 못해 내부 알고리즘이 블랙박스처럼 되어 있어 그로 인해 기계학습의 예측결과에 영향을 주는 요인들을 사용자가 알지 못하고 그 원인을 설명을 할 수가 없다. 이는 기계학습의 예측 결과에 대한 잠재적 불확실성이 존재하며 그 결과를 검증없이 수용해야하는 위험한 상황을 직면하게 된다.

이 논문에서는 기존의 기계학습 알고리즘을 변경하지 않고 알고리즘의 입력 값과 결과 값의 상관관계를 직관적으로 보여주는 LIME(Local Interpretable Model-agnostic Explanation)[1]기법을 소개하고, 기존 LIME보다 사용자가 쉽게 이해할 수 있도록 향상된 시각화 방법을 함께 제시하고자 한다.

* 이 논문은 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2017R1A2B4003558).

2. 이론적 배경

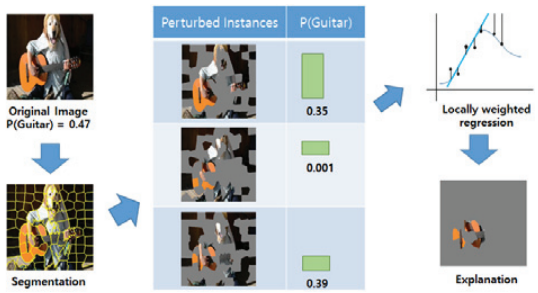
2.1 LIME(Local Interpretable Model-agnostic Explanation) 소개

LIME은 기계학습 알고리즘의 예측 결과가 어떠한 입력 값으로부터 긍정적인 영향을 받는지, 그리고 어떠한 입력 값들로부터 부정적인 영향을 받는지를 사용자가 직관적으로 이해할 수 있도록 해줌으로써, 기계학습이 출력한 결과의 신뢰성을 확인할 수 있는 방법 중 하나이다. LIME은 이미지 인식 분야 및 텍스트의 감성분석 분야 등에서도 활용되고 있는데, 본 논문에서는 이미지 인식분야에서 사용되는 LIME에 대해 다

루고자 한다.

2.2 이미지 인식분야에서의 LIME

LIME은 다음과 같은 3단계 프로세스로 이루어진다. 첫 단계는 Image segmentation(분할)단계로, 객체 형태를 기반으로 입력되는 이미지를 작은 unit으로 구분하여 다음 단계인 상관관계 분석의 입력 값으로 사용한다. 두 번째 단계는 Correlation analysis (상관관계 분석)단계로, Linear regression을 기반으로 입력 값과 출력 값의 상관관계를 도출한다. 마지막은 Explanation (표현) 단계로 입력 값과 출력 값의 상관관계를 사용자가 이해할 수 있도록 시각화 하는 단계이다. [그림 1]은 LIME의 3단계 프로세스를 간략하게 설명하고 있다.



[그림 1] LIME의 기본 개념

2.2.1 Image segmentation (이미지 영역 분할)

Image segmentation은 LIME에 있어서 가장 중요한 단계로서 입력 영상을 의미를 가지는 작은 unit으로 분할하는 단계이다. LIME은 segmentation을 통해 분할된 작은 unit들을 다음 단계인 상관관계 분석단계의 입력으로 사용한다. Image segmentation은 평면 이미지에서 의미 있는 객체(Object)를 배경과 분리함으로써 이미지를 좀 더 의미 있고 해석하기 쉬운 형태로 표현하기 위해 사용되는데, 현재 Quick-Shift, Felzenszwalb[2], SLIC[3], Watershed 등과 같은 여러 가지 방법이 소개되어 다양한 분야에서 사용되고 있다. 이 논문에서는 기존 LIME에서 사용한 Quick-Shift 방법 이외에 Felzenszwalb와 SLIC 방법을 추가로 비교 분석하여 사용자가 좀 더 직관적으로 이해할 수 있는 방법을 제안하고자 한다.

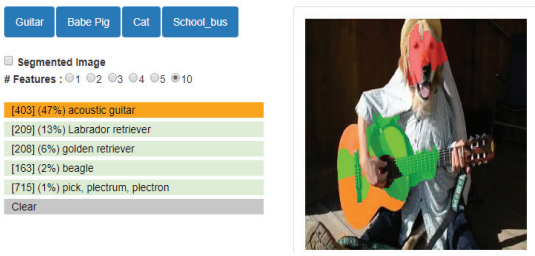
2.2.2 Correlation analysis (상관관계 분석)

Correlation analysis 단계에서는 segmentation단계에서 만들어진 이미지의 분해된 unit들을 입력으로 사용하게 된다. 앞서 입력된 이미지는 이미 분석하고자 하는 기계학습 모델에 의해 식별된 결과 값(예 : acoustic guitar : 47%)이 존재하며, 이 예측된 결과 값이 이 이미지의 어느 unit으로부터 영향을 받는지에 대해 분석하는 것이 상관관계 분석의 주된 목표이다.

Correlation analysis단계에는 3가지 과정이 존재한다. 첫 번째 과정으로 개별로 나누어진 unit들을 조합하여 이미지의 부분집합들을 형성한다. 두 번째 과정으로 이 부분집합들을 기계학습 모델의 입력으로 사용하여 각 부분집합들에 대한 결과 값을 얻는다. 세 번째 과정으로 여러 개의 부분집합들을 반복 측정하여 입력 unit들과 얻어낸 결과들의 상관관계를 분석한다. 이때의 상관관계는 선형 회귀(Linear regression)를 사용하여 선형적인 모델을 사용한다. 만일 어떠한 unit을 부분집합에서 제거하였을 때 결과 값의 감소량이 크다면 해당 unit은 예측 결과에 매우 중요한 요소임을 판단할 수 있으며 해당 unit은 긍정적 영향도가 매우 높다는 것을 알 수 있다. 반면 unit을 제거하였는데도 불구하고 별다른 변화가 보이지 않는다면 영향도가 낮다는 것을 알 수 있다. 또한 unit을 제거하였을 때 반환된 결과 값이 크게 증가한다면 부정적 영향도가 높아 결과 예측 시 방해가 되는 unit임을 알 수 있다. 이러한 과정을 거쳐 각 결과 값 판정에 영향도가 높은 입력 unit들을 찾아낼 수 있다.

2.2.3 Explanation (표현)

LIME의 마지막 단계는 도출된 입력과 출력의 상관관계를 사용자가 이해할 수 있도록 시각화를 통해 표현(Explanation)하는 단계이다. LIME은 [그림 2]와 같이 segmentation된 입력 unit들 중에서 예측한 결과에 긍정적 영향도가 높은 unit들을 녹색으로, 부정적 영향도가 큰 unit들을 적색으로 표시함으로써, 사용자가 기계학습이 예측한 결과에 영향을 주는 unit들을 직관적으로 이해할 수 있게 한다.



[그림 2] Explanation 화면

3. 실험 및 개선

기존 LIME에서 Image segmentation을 위해 사용한 Quick-Shift방법은 객체의 경계선이 복잡하여 사용자가 객체를 구분하기가 어렵고, 15.5초라는 긴 실행 시간이 필요하였다. 본 실험에서는 Quick-Shift방법 이외에 Image segmentation분야에서 잘 알려진

Felzenszwalb방법과 SLIC방법을 비교 실험하고, segmenation에 필요한 입력 parameter를 최적화함으로써 사용자의 인지용이성을 향상시키고자 하였다.

실험을 위해서 사용한 이미지 Data set은 ILSVRC-2012 를 이용하였으며, 영상 인식을 위한 기계학습 모델은 2012년 영상 인식 분야에서 3.45%의 오차율로 우승을 차지했던 CNN 기반의 Inception V3 모델을 사용하였다. 또한 실험을 위해 Python을 기반으로 Tensorflow 와 Matplotlib 라이브러리를 이용하여 LIME을 실행하였다.

Data	c	N	Instance size
ILSVRC -2012	1000	1.2M	Various Image sizes

[표 1] Overview of dataset.
c : Number of target classes, N : Dataset size

첫 번째 실험은 기존 LIME에서 사용한 Quick-Shift 이외에 Felzenszwalb, SLIC segmentation방법을 비교 실험하였다. [그림 3]과 [표2]에서 보는바와 같이 Felzenszwalb방법은 Quick-Shift방법보다 실행시간이 빠르지만, 분할된 객체에 대한 경계값이 Quick-Shift보다 복잡하여 사용자 시인성이 저하되는 것을 알 수 있다.

SLIC방법은 실행속도가 Felzenszwalb방법보다는 느리지만 Quick-Shift방법보다 약 4배가 빠르고, segmentation된 결과도 이미지의 객체뿐만 아니라 객체 내에서의 부분영역까지 분할하여 Quick-shift의 결과에 비해 사용자가 직관적으로 객체를 쉽게 인식할 수 있는 출력을 가져왔다.

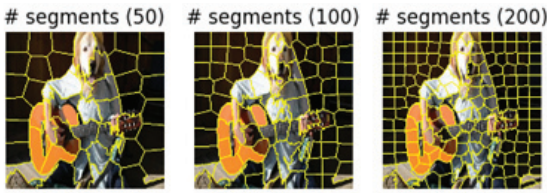
Quick-Shift Felzenszwalb SLIC

[그림 3] Segmentation방법에 따른 결과 비교

방법	소요시간 (sec)	Segments 개수 (pieces)
Quick-Shift	**15.5**	58
Felzenszwalb	0.587	306
SLIC	**3.7**	90

[표 2] Segmentation방법에 따른 소요시간 비교

두 번째 실험으로는 SLIC segmentation을 실행 시 함께 입력되는 parameter값들 중 n-segments 옵션을 최적화하여 사용자의 인지용이성을 최대화시키고자 하였다. [그림 4]에서 보는바와 같이 n-segments옵션의 값을 50, 100, 200으로 변경하며 실험한 결과 100으로 설정했을 때 가장 양호한 결과가 나옴을 알 수 있었다.



[그림 4] SLIC의 n-segments 옵션에 따른 결과 비교

4. 결론 및 향후 계획

본 연구는 이미지 인식 과정에 있어서 LIME기법을 적용하여 기계학습 알고리즘을 변경하지 않으면서 결과 값에 영향을 준 입력 unit에 대해 사용자에게 설명할 수 있는 LIME기법을 소개하였으며, 기존 LIME보다 사용자의 인지용이성을 향상시키기 위해 LIME에서 사용한 Quick-Shift 방법과 image segmentation분야에서 잘 알려진 Felzenszwalb, SLIC segmentation방법을 비교 분석함으로써 SLIC 방법(n-segments 옵션값 : 100)이 가장 뛰어난 성능을 보여주는 것을 확인하였다.

향후 연구에서는 이미지 분야에서 범위를 확장하여 동영상 속 객체(Object)인식에 대한 근거를 설명하는 연구를 진행해 보고자 한다.

참 고 문 헌

[1] 1.Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why Should I Trust You?: Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135-1144). ACM.

[2] Felzenszwalb, P., Huttenlocher, D.(2004). Efficient graph-based image segmentation. IJCV.

[3] Radhakrishna Achanta, et al (2012) SLIC Superpixels Compared to State-of-the-Art Superpixel Methods.