KIPS Tr. Comp. and Comm. Sys.
Vol.7, No.1 pp.1~8 pISSN: 2287-5891

https://doi.org/10.3745/KTCCS.2018.7.1.1

Video Classification System Based on Similarity Representation Among Sequential Data

Hosuk Lee† · Jihoon Yang††

ABSTRACT

It is not easy to learn simple expressions of moving picture data since it contains noise and a lot of information in addition to time-based information. In this study, we propose a similarity representation method and a deep learning method between sequential data which can express such video data abstractly and simpler. This is to learn and obtain a function that allow them to have maximum information when interpreting the degree of similarity between image data vectors constituting a moving picture. Through the actual data, it is confirmed that the proposed method shows better classification performance than the existing moving image classification methods.

Keywords : Deep Learning, Video Classification, Similarity Measure, Representation Learning

순차 데이터간의 유사도 표현에 의한 동영상 분류

이 호 석† · 양 지 훈††

요 약

동영상 데이터는 시간에 따른 정보는 물론이고, 많은 정보량과 함께 잡음도 포함하고 있기 때문에 이에 대한 간단한 표현을 학습하는 것은 쉽지 않다. 본 연구에서는 이와 같은 동영상 데이터를 추상적이면서 보다 간단하게 표현할 수 있는 순차 데이터간의 유사도 표현 방법과 딥러닝 학습방법을 제안한다. 이는 동영상을 구성하는 이미지 데이터 벡터들 사이의 유사도를 내적으로 표현할 때 그것들이 서로 최대한의 정보를 가질 수 있도록 하는 함수를 구하고 학습하는 것이다. 실제 데이터를 통하여 제안된 방법이 기존의 동영상 분류 방법들보다도 뛰어난 분류 성능을 보임을 확인하였다.

키워드 : 딥 러닝, 비디오 분류, 유사도 측정, 표현 학습

1. 서 론

최근 데이터의 특징 벡터와 같은 추상적인 표현(Abstract Representation)을 학습할 수 있는 딥 러닝 구조들이 제안되었다. 딥 러닝은 여러 가지의 함수를 근사할 수 있다는 점 덕분에 이와 같은 역할을 하는 좋은 모델들을 학습하는 것이 하나의 연구 주제로 자리 잡게 되었다. 이런 모델들은 공통적으로 데이터로부터 좋은 표현, 즉, '데이터의 특징을 갖고 있는, 보다 간단한 형식의 데이터'를 추출하는 데에 그 목적이 있다. 이와 같이 데이터의 정보를 잃지 않고 간단하게 표현하는 방법들을 연구하는 기계학습의 한 분야를 표현 학습(Representation Learning)[1]이라 한다.

그림 데이터를 예로 들자면, 가로, 세로 각각 300 화소를 가진 컬러 그림이라 해도 최소 270000개의 화소에 대한 정보가 필요하다. 이를 현재 딥 러닝 분야에서 그림 데이터의 특징 추출 모델로 널리 쓰이고 있는 AlexNet[2]으로 특징 데이터(벡터)를 추출한다면, 이 데이터는 4096차원의 벡터로 보다 간단하게 나타낼 수 있다.

하지만 동영상 데이터를 대상으로 한다면 이 같은 모델을 찾는 일은 그림에 대한 모델을 찾는 일에 비해 간단하지 않다. 동영상 데이터는 여러 장의 그림 데이터와 연속적으로 이루어진 음성 데이터가 조합된 데이터로, 이 데이터가 가지고 있는

※ 이 논문은 2017년도 산업통상자원부 산업기술평가관리원의 지원을 받아 수행된 연구(NO: 10076752) 및 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원(2017-0-01772, (총괄) 비디오 튜링 테스트(Video Turing Test): 인간 수준의 비디오 이해 지능 및 검증 기술 개발)과 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터(IITP)의 지원을 받아 수행된 연구임(2017-0-1781, (3세부) 비디오 이해를 위한 데이터 수집 및 보정 자동화 시스템 개발).
† 비 회 원 : 서강대학교 컴퓨터공학과 석사
†† 종신회원 : 서강대학교 컴퓨터공학과 교수
Manuscript Received : May 25, 2017
First Revision : September 12, 2017
Second Revision : September 27, 2017
Accepted : November 18, 2017
* Corresponding Author : Jihoon Yang(yangjh@sogang.ac.kr)

정보량은 보통 그림 데이터보다 크다. 또한, 이 데이터는 시계에 따라 변하는 시계열 데이터로도 볼 수 있고, 따라서 이런 특성까지 고려하여 데이터가 가진 정보를 최대한 보존하면서 간단하게 표현할 수 있는 모델을 찾는 것은 쉽지 않다.

한편, 근래에 들어 휴대용 촬영기기나 편집기기 등 정보를 생산할 수 있는 도구가 많이 보급되면서 동영상 데이터나 글과 같은 순차 데이터(Sequential Data)의 축적량과 그 축적 속도가 점점 증가하고 있다. 데이터의 축적량이 커짐에 따라 이를 검색 등에 활용할 수 있도록 보다 간단하게 나타내는 작업 역시 중요해지고 있다. 그림 검색 시스템을 예로 들자면, 최근에는 사람이 직접 입력한 그림에 대한 정보 외에도, 그림 자체로부터 앞서 언급한 특징 벡터를 추출해 색인을 해두고 이를 검색에 활용하기도 한다.

동영상 데이터를 연속적인 그림의 나열이라 생각했을 때, 그림에서 특징을 추출할 수 있는 모델을 활용한다 하더라도 그림의 수가 많기 때문에 이를 간단하게 나타낼 수 있는 모델을 학습하는 것은 쉽지 않다. 그럼에도 불구하고 많은 연구들이 이루어져서 최근에는 이 같은 문제를 해결하기 위해 평균 풀링(Average Pooling)[3]과 같은 방법들이 제안되었고, 처음부터 순차 데이터를 학습하기 위해 고안된 딥 러닝 구조인 GRU unit[4]을 사용한 LSTM[5]과 같은 모델이 제안되었다.

본 연구에서는 간단한 신경망 구조로 순차 데이터 중 하나인 동영상 데이터의 특징을 반영하는, 한 개의 데이터를 추출할 수 있는 모델을 학습하는 방법을 제안한다. 이를 위해, 개별 이미지 벡터는 내적으로 유사도를 표현할 때 서로 최대한의 정보를 갖도록 도출된 모델로써 새로이 표현된다. 이 방법을 동영상 데이터 집합인 YLI-MED 데이터 집합[6]과 자체적으로 수집한 Youtube 데이터 집합에 적용하여 평균 풀링(Average Pooling)이나 GRU unit을 사용한 LSTM 신경망 구조 등 동영상 데이터를 학습하는 다른 방법들과 비교하여 더 좋은 성능을 보이는 것을 확인한다.

2. 관련 연구

2.1 표현 학습

데이터의 표현(Representation)은 일반적으로 데이터의 특징 벡터(Feature Vector)와 같은 의미를 지닌다. 기계 학습에서의 데이터의 특징이란 '데이터로부터 파생된, 해당 데이터의 특징적인 정보를 갖고 있는 데이터'로 정의할 수 있다. 이런 데이터는 보통 한 데이터의 특징적인 정보를 갖고 있으므로, 원본 데이터의 총 정보량의 크기보다는 작아야 한다. 위와 같은 맥락에서 데이터가 노이즈를 비롯한 불필요한 정보들을 포함하고 있는 경우, 이와 같은 정보들을 가능한 한 배제한 채 데이터의 특징적인 정보들을 포함한 채 보다 간단한 형식의 데이터를 만드는 것도 '데이터의 표현을 학습 한다'고 할 수 있다.

표현 학습(Representation Learning)이란, 따라서 데이터로부터 그 특징들을 잘 반영하고 있는 데이터를 생성하는 것을 목적으로 한다. 이와 같은 정의들을 바탕으로 하여, 앞으로는 표현(Representation)과 특징(Feature)을 같은 의미로 사용하기로 한다. 즉, 본 연구에서의 '동영상 데이터의 표현(Video Data Representation)'이란, '동영상 데이터를 나타내는 특징 벡터'와 같은 의미를 지니고 있다고 할 수 있다.

2.2 풀링 기반 관련 연구

평균 풀링(Average Pooling)은 콘볼루션 신경망[8]에서 주로 쓰이고 있는 차원 축소 방법이다. 콘볼루션 신경망에서의 평균 풀링은 유한한 2차원 영역 내의 값들의 평균값을 구하여 입력 데이터의 차원을 축소하는 역할을 한다. 평균 풀링을 벡터들로 구성된 순차 데이터 $\boldsymbol{s}=\left\{\boldsymbol{s}_1,\boldsymbol{s}_2,...,\boldsymbol{s}_N|\boldsymbol{s}_i\in\mathbb{R}^n\right\}$ 에 적용한다면, 평균 풀링을 적용한 벡터 $\boldsymbol{s}_a$는 다음과 같이 Equation (1)의 식으로 계산할 수 있다.

$$\boldsymbol{s}_a=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{s}_i \tag{1}$$

2.3 LSTM 기반 관련 연구

Recurrent Neural Network(RNN)[7]은 순차 데이터를 학습하기 위한 신경망 구조이다. 최근에는, 이 구조의 확장으로 LSTM(Long Short Term Memory) 신경망 구조가 데이터를 학습하기 위한 방법으로 널리 쓰이고 있다. LSTM은 RNN이 가진 한계중 하나인 학습 과정에서의 경사 약화 문제(Vanishing Gradient Problem)[8] 을 극복한 구조로, 이 같은 특징 때문에 RNN에 비해 시계열 데이터 학습이 잘 이루어진다. 이 중에서도 GRU unit을 이용한 LSTM 신경망 구조는 현재 순차 데이터 중에서도 시계열 데이터를 학습하는 분야에서 좋은 성능을 보이고 있다. 본 연구에서는 비교 실험으로 이 신경망 구조를 쓰기로 하였다.

2.4 다른 딥 러닝 기반 관련 연구

[9]에서는 YLI-MED 데이터 집합에 대해 동영상 데이터를 간단하게 나타내는 대신에 이를 구성하는 각 데이터에 대한 분류기를 학습하고, 각각의 데이터를 분류하여 그 결과를 바탕으로 하여 다수결로 동영상을 분류하는 실험을 진행하였다. 동영상이 여러 개의 데이터가 모인 집합이라 생각한다면, 이처럼 데이터를 이루고 있는 각 원소들을 분류한 결과들을 취합하여 다수결을 통해 동영상이 가진 주제를 분류하는 방법이 한편으론 가장 합리적이고 기본적인 분류 방법이라 생각할 수 있다. 이는 비록 본 연구에서 제안하는 방법과는 차이가 있지만, 본 연구에서 제안한 알고리즘은 단 하나의 데이터를 생성하여 이를 바탕으로 동영상을 분류하므로, 비교하기에 적합하다고 판단되어 이 방법 역시 실험에 사용하였다.

3. 유사도 측정 기반의 동영상 데이터 표현

3.1 두 벡터의 유사도 측정법

데이터는 벡터로 표현할 수 있다. 따라서 두 벡터의 유사도를 측정하는 방법은 곧 데이터의 유사도를 측정하는 방법

이라 할 수 있다. 두 개의 벡터 $\boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^n$가 존재한다고 할 때, 두 벡터의 유사도는 두 벡터의 내적으로 정의할 수 있다. $\langle\bullet,\bullet\rangle$: $\mathbb{R}^n\times\mathbb{R}^n\rightarrow\mathbb{R}$ 을 $\mathbb{R}^n$에서의 일반적인 내적이라 하자. 그러면 내적이 주어진 공간 $\mathbb{R}^n$은 노름(Norm) $\sqrt{\langle\bullet,\bullet\rangle}$으로 유도한 거리 공간(Metric Space)이라 할 수 있다. 데이터가 속하는 공간은 유한 차원이므로, 유한 차원만 고려할 때 벡터의 내적에 대해 다음과 같은 명제가 성립한다.

$\langle\boldsymbol{x},\boldsymbol{y}\rangle=\langle\boldsymbol{x},\boldsymbol{y}'\rangle$ 이면 $\boldsymbol{y}=\boldsymbol{y}'$ 이거나 $\boldsymbol{x}\perp(\boldsymbol{y}-\boldsymbol{y}')$ 이다.

이 같은 성질로 인하여, 두 벡터는 특정 조건하에 = 또는 $\perp$ 에 대한 관계가 있다고 이야기 할 수 있다. 그러므로 두 벡터의 유사도를 두 벡터의 내적을 통해서도 측정할 수 있다고 볼 수 있다. 특히, 여러 쌍의 벡터의 내적 값이 동일하다면, 각 벡터들 역시 위와 같은 관계가 성립한다. 만약 L^2-노름으로 정규화한 벡터만 다룬다면 이 유사도 측정법은 두 벡터가 이루고 있는 각도만 고려하여 유사도를 측정하는 코사인 유사도가 된다. 이는 일반적인 두 벡터의 내적이 $\langle\boldsymbol{x},\boldsymbol{y}\rangle=|\boldsymbol{x}||\boldsymbol{y}|cos\theta$이고 L^2-노름으로 정규화한 벡터의 크기는 항상 1이므로 $\langle\boldsymbol{x},\boldsymbol{y}\rangle=cos\theta$ 이고, 따라서 두 벡터가 이루는 각도로 유사도를 측정한다고 볼 수 있다.

두 벡터의 유사도를 측정하는 방법에는 이와 같은 측도 외에도 다른 측도도 존재한다. 이런 측도들은 공통적으로 측도로 유도할 수 있는 노름으로 거리 공간을 생성할 수 있다.

3.2 내적을 기반으로 한 동영상 데이터의 표현

앞으로는 보다 수학적인 설명을 위해 우선 순차 데이터를 '$\mathbb{R}^n$에서 정의된 벡터 열, 즉 $\mathbb{R}^n$에 존재하는 N개의 벡터들의 집합 $\boldsymbol{d}=\{\boldsymbol{d}_1,\boldsymbol{d}_2,...,\boldsymbol{d}_N\}$'라 정의한다. 이와 같이 순차 데이터를 정의한다면, 순차 데이터를 이루는 각각의 원소 $\boldsymbol{d}_i$가 의미를 갖는다고 가정할 수 있다. 앞으로 다루게 될 동영상 데이터라면, 동영상으로부터 초당 1개의 정지 화면을 추출하였다고 가정했을 때, 이를 사람이 관찰한다면 각각의 정지 화면들로부터 유의미한 정보를 찾을 수 있을 것이다. 그리고 이들이 가진 정보를 취합한다면 동영상이 어떤 주제를 갖고 있는지 더 파악하기 수월해질 것이다.

유사도 표현을 이용한 동영상 분류 시스템에 사용할 학습 알고리즘은 바로 이와 같은 점에 착안하여 고안되었다. 본 연구는 내적으로 벡터들 사이의 유사도를 측도하였을 때, 동영상 데이터를 이루고 있는 각각의 벡터들이 다른 벡터들의 정보를 최대한 가질 수 있도록 만드는 모델(함수)을 찾는 것을 목적으로 한다. 즉, 동영상 데이터의 각 벡터들을 앞서 설명한 성질을 갖는 벡터가 되도록 변환하는 연속함수 f를 근사하고자 한다.

지금부터는 동영상 데이터를 $\boldsymbol{v}$라 하고, 순차 데이터 $\boldsymbol{d}$와 유사하게 나타내어 $\boldsymbol{v}=\{\boldsymbol{v}_1,\boldsymbol{v}_2,...,\boldsymbol{v}_N\}$이라 하자. $\boldsymbol{v}$를 이루는 각 벡터는 고정된 차원에 속하는 벡터들이며, $\mathbb{R}^n$에 존재한다고 하자. N은 동영상에 따라 달라질 수 있다. 다만, 지금은 다른 동영상은 고려하지 않도록 한다. 이를 바탕으로 동영상 $\boldsymbol{v}$가 주어져 있을 때, 본 연구에서 찾고자 하는 모델은 Equation (2)와 같이 목적 함수를 최소화 하는 모델이다.

$$\sum_{i=1}^{N}\left|\sum_{j=1}^{N}\langle\boldsymbol{v}_i,\boldsymbol{v}_i\rangle-\sum_{j=1}^{N}\langle\boldsymbol{v}_i,f(\boldsymbol{v}_j)\rangle\right| \tag{2}$$

(단, 여기서 $\boldsymbol{v}_i\in\{\boldsymbol{v}_1,\boldsymbol{v}_2,...,\boldsymbol{v}_N\}=\boldsymbol{v}$ 를 의미한다.)

함수 f는 또한 모든 $\boldsymbol{v}_i\in\{\boldsymbol{v}_1,\boldsymbol{v}_2,...,\boldsymbol{v}_N\}$에 대해 Equation (3)의 식과 같은 부등식을 만족해야 한다.

$$\left|\sum_{j=1}^{N}\langle\boldsymbol{v}_i,\boldsymbol{v}_i\rangle-\sum_{j=1}^{N}\langle\boldsymbol{v}_i,f(\boldsymbol{v}_j)\rangle\right| \leqq \left|\sum_{j=1}^{N}\langle\boldsymbol{v}_i,\boldsymbol{v}_i\rangle-\sum_{j=1}^{N}\langle\boldsymbol{v}_i,\boldsymbol{v}_j\rangle\right| \tag{3}$$

위 조건식을 만족하는 함수 f의 존재성은 보장된다. 우선 정의역과 치역이 모두 유한하므로 옹골 집합(Compact Set)이고, 따라서 두 집합 사이에 연속 함수가 적어도 한 개 존재한다. 만약 f를 항등함수라 정의한다면, 위 조건식의 등호가 성립하고, f의 값이 항상 $\boldsymbol{v}_i$가 나오도록 정의한다면, 위 조건식의 왼쪽 항은 0이 되어 부등호가 성립한다. 따라서 함수 f는 이런 함수들의 일차 결합으로 나타낼 수 있다. 그러므로 위 조건식을 만족하는 연속함수 f는 존재한다.

한편, 최소화하고자 하는 목적 함수를 (2)와 같이 정의한 이유는 다음과 같다. 먼저 $\boldsymbol{v}_i$와 가장 유사한 벡터는 자기 자신이다. 그러므로 두 벡터의 유사도를 내적으로 측도한다면, $\langle\boldsymbol{v}_i,\boldsymbol{v}_i\rangle$가 $\boldsymbol{v}_i$가 가질 수 있는 최대의 유사도라 생각할 수 있다. 또한, $\langle\boldsymbol{v}_i,f(\boldsymbol{v}_j)\rangle$는 $\boldsymbol{v}_j$에 대한 함수 f값과 $\boldsymbol{v}_i$의 유사도를 나타내고 있으므로, $\langle\boldsymbol{v}_i,\boldsymbol{v}_i\rangle$와 $\langle\boldsymbol{v}_i,f(\boldsymbol{v}_j)\rangle$의 차이는 $f(\boldsymbol{v}_j)$가 $\boldsymbol{v}_i$의 정보를 얼마나 잘 반영하고 있는지를 나타내고 있다고 생각할 수 있다.

본 연구에서는 위와 같은 역할을 하는 함수를 sigmoid 활성함수를 가진 하나의 층을 가진 신경망으로 근사하고자 한다. 신경망이 나타내고 있는 함수를 f라 했을 때, 이 함수는 신경망의 성질로 인해 미분 가능하다. 이런 점으로 인하여, 본 연구에서 제안하는 모델은 이미 다양한 방면으로 연구가 된 경사 기반(Gradient-Based)의 역전파 알고리즘(Backpropagation-Algorithm)[10]으로 학습할 수 있다.

이제 Equation (2)를 만족하는 최적의 f를 f^*라 하자. f^*는 최적이므로, Equation (2)를 만족하는 모든 f에 대해 다음 Equation (4)와 같은 부등식을 만족한다.

$$\sum_{i=1}^{N}\left|\sum_{j=1}^{N}\langle\boldsymbol{v}_i,\boldsymbol{v}_i\rangle-\sum_{j=1}^{N}\langle\boldsymbol{v}_i,f^*(\boldsymbol{v}_j)\rangle\right| \leqq \sum_{i=1}^{N}\left|\sum_{j=1}^{N}\langle\boldsymbol{v}_i,\boldsymbol{v}_i\rangle-\sum_{j=1}^{N}\langle\boldsymbol{v}_i,f(\boldsymbol{v}_j)\rangle\right| \tag{4}$$

동영상 데이터를 하나의 벡터로 표현하기 위해서는 $\{\boldsymbol{v}_1,\boldsymbol{v}_2,...,\boldsymbol{v}_N\}$의 f^*에 대한 출력 $\{f^*(\boldsymbol{v}_1),f^*(\boldsymbol{v}_2),...,f^*(\boldsymbol{v}_N)\}$에 대해 평균 풀링(Average Pooling)을 적용한다. 즉, 동영상 데이터에 대한 하나의 벡터 표현을 $\frac{1}{N}\sum_{i=1}^{N}f^*(\boldsymbol{v}_j)$로 한다. 동영상 데이터를 이처럼 표현해야 하는 이유는 Equation (5)와 같다.

$$\left|\sum_{j=1}^{N}\langle \boldsymbol{v}_i, \boldsymbol{v}_i\rangle - \sum_{j=1}^{N}\langle \boldsymbol{v}_i, f^*(\boldsymbol{v}_j)\rangle\right| \quad (5)$$

위와 같은 수식이 주어졌을 때, 이 수식은 몇 가지 조작을 통해 Equation (6)과 같이 나타낼 수 있다.

$$\begin{aligned}&\left|\sum_{j=1}^{N}\langle \boldsymbol{v}_i, \boldsymbol{v}_i\rangle - \sum_{j=1}^{N}\langle \boldsymbol{v}_i, f^*(\boldsymbol{v}_j)\rangle\right| \quad (6)\\ &= \left|N\langle \boldsymbol{v}_i, \boldsymbol{v}_i\rangle - \sum_{j=1}^{N}\langle \boldsymbol{v}_i, f^*(\boldsymbol{v}_j)\rangle\right|\\ &= \left|N\langle \boldsymbol{v}_i, \boldsymbol{v}_i\rangle - \left\langle \boldsymbol{v}_i, \sum_{j=1}^{N} f^*(\boldsymbol{v}_j)\right\rangle\right|\\ &= N\left|\langle \boldsymbol{v}_i, \boldsymbol{v}_i\rangle - \left\langle \boldsymbol{v}_i, \frac{1}{N}\sum_{j=1}^{N} f^*(\boldsymbol{v}_j)\right\rangle\right|\end{aligned}$$

따라서 위와 같은 수식에 의해, 동영상 데이터를 표현하는 하나의 특징 벡터를 $\frac{1}{N}\sum_{i=1}^{N} f^*(\boldsymbol{v}_j)$로 나타낼 수 있다. 위의 경우는 1개의 동영상만을 고려한 수식이나, 여러 개의 동영상이 존재하더라도 유한하게 존재하므로 위와 같은 수식은 여전히 성립하고, 따라서 각 동영상에 대해 f^*가 주어질 경우 동영상 데이터를 $\frac{1}{N}\sum_{i=1}^{N} f^*(\boldsymbol{v}_j)$으로 나타낼 수 있다.

지금까지 나열한 학습 과정들은 [Algorithm 1]을 통해 설명한다.

Algorithm 1

학습 동영상(v_{TR})과 실험 동영상(v_{TE})들을 모아놓은 집합을 $\{v_{TR_1}, v_{TR_2},\ldots,v_{TR_N}, v_{TE_1}, v_{TE_2},\ldots,v_{TE_M}\}$이라 하고, 이 집합에 속한 동영상들을 단순히 v_i로 나타내었을 때, $l(i)$를 동영상 v_i가 가진 특징 벡터의 개수라 하자. 이 정의를 바탕으로 각 동영상 v_i를 벡터들의 집합 $\{v_{i1}, v_{i2},\ldots,v_{il(i)}\}$이라 하자. 그리고 W를 신경망의 가중치(weight), b를 신경망의 바이어스(bias)라 하고, E를 신경망을 학습할 총 반복 횟수(epoch)이라 하자. 참고로, 이를 바탕으로 하였을 때, $f(x)=\sigma(Wx+b)$이다.(단, 여기서 σ는 sigmoid 함수이다.)

입력값: 동영상 $\{v_{TR_1}, v_{TR_2},\ldots,v_{TR_N}, v_{TE_1}, v_{TE_2},\ldots,v_{TE_M}\}$
반환값: 유사도 측정 기반 표현법을 이용하여 구한 각 동영상에 대한 특징 벡터 집합
$\{v_{TR_1}{}^*, v_{TR_2}{}^*,\ldots,v_{TR_N}{}^*, v_{TE_1}{}^*, v_{TE_2}{}^*,\ldots,v_{TE_M}{}^*\}$

1: 신경망의 매개변수 W, b와 learning rate η를 초기화
2: **for** each epoch e from 1 to E **do**
3: **for** each $v_i \in \{v_{TR_1}, v_{TR_2},\ldots,v_{TR_N}\}$ **do**
4: **for** each vector $v_{TR_{ij}} \in \{v_{TR_{i1}}, v_{TR_{2i}},\ldots,v_{TR_{il(i)}}\}$ **do**
5: Calculate $\triangle_{W_{ij}} = \frac{\partial}{\partial W}\sum_{k=1}^{l(i)}\langle v_{TR_{ij}}, f(v_{TR_{ik}})\rangle$
6: Calculate $\triangle_{b_{ij}} = \frac{\partial}{\partial b}\sum_{k=1}^{l(i)}\langle v_{TR_{ij}}, f(v_{TR_{ik}})\rangle$
7: **if** $\sum_{k=1}^{l(i)}\langle v_j, v_j\rangle - \sum_{k=1}^{l(i)}\langle v_{TR_{ij}}, f(v_{TR_{ik}})\rangle \geq 0$
Update $W \leftarrow W+\eta\triangle_{W_{ij}}$
else
Update $W \leftarrow W-\eta\triangle_{W_{ij}}$
8: **if** $\sum_{k=1}^{l(i)}\langle v_j, v_j\rangle - \sum_{k=1}^{l(i)}\langle v_{TR_{ij}}, f(v_{TR_{ik}})\rangle \geq 0$
Update $b \leftarrow b+\eta\triangle_{W_{ij}}$
else
Update $b \leftarrow b-\eta\triangle_{W_{ij}}$
9: **end for**
10: **end for**
11: **end for**
12: **for** each $v_i \in \{v_{TR_1}, v_{TR_2},\ldots,v_{TR_N}, v_{TE_1}, v_{TE_2},\ldots,v_{TE_M}\}$ **do**
13: Calculate $v_i^* = \frac{1}{l(i)}\sum_{j=1}^{l(i)} f(v_{ij})$
14: **end for**
15: return $\{v_{TR_1}{}^*, v_{TR_2}{}^*,\ldots,v_{TR_N}{}^*, v_{TE_1}{}^*, v_{TE_2}{}^*,\ldots,v_{TE_M}{}^*\}$

4. 실험 환경

4.1 실험에 사용한 데이터 집합

4.1.1 YLI-MED

YLI-MED 데이터 집합은 Yahoo Flickr Creative Commons 100 Million(YFCC100M) 데이터 집합의 부분 집합으로, 10개의 이벤트로 구성된 동영상으로 되어 있으며, 각 이벤트당 100개의 학습 데이터로 총 1000개의 학습 데이터와 823개의 실험 데이터로 이루어져 있다. Table 1은 YLI-MED 데이터 집합의 통계치를 나타낸 것이다.

Table 1. Statistics of YLI-MED Dataset

Event ID	Event Name	Count
Ev101	Birthday Party	237
Ev102	Flash Mob	150
Ev103	Getting a Vehicle Unstuck	141
Ev104	Parade	230
Ev105	Person Attempting a Board Trick	194
Ev106	Person Grooming an Animal	139
Ev107	Person Hand-Feeding an Animal	220
Ev108	Person Landing a Fish	143
Ev109	Wedding Ceremony	219
Ev110	Working on a Woodworking Project	150
	Total	1823

4.1.2 Youtube

실험을 위해 따로 수집한 Youtube 데이터 집합을 실험에 사용하였다. 이 집합은 YLI-MED 데이터 집합과 유사하나, 총 16개의 이벤트로 구성되어 있으며, 906개의 학습 데이터와 463개의 실험 데이터로 이루어져 있다. 직접 수집에는 한계가

Table 2. Statistics of Youtube Dataset

Event ID	Event Name	Training	Test
Ev101	First Birthday Party	64	33
Ev102	Camping	28	15
Ev103	Climbing	18	10
Ev104	Amusement Park	63	33
Ev105	Pet	64	32
Ev106	Entrance Ceremony	22	11
Ev107	Birthday Party	56	29
Ev108	Wedding	64	32
Ev109	Cute Festival	66	33
Ev110	Bicycle Stunt	66	34
Ev111	Graduation Ceremony	65	33
Ev112	Fishing	66	34
Ev113	Toddle	66	34
Ev114	Golf	66	33
Ev115	Ramen Cook	66	34
Ev116	Makeup	66	33
	Total	906	463

있어 데이터 분포나 수가 YLI-MED 데이터 집합보다는 고르지 못하고 그 수도 적지만, 본 연구에서 제안한 방법에 대한 성능을 확인하기에는 적합하다고 판단되어 사용했다. Table 2는 Youtube 데이터 집합의 통계치를 나타낸 것이다.

4.2 데이터 전처리

YLI-MED 데이터 집합에는 원본 동영상은 물론 영상과 음성 모달리티에 대해 추출한 특징 벡터들도 제공된다. 하지만 제공되는 특징 벡터들을 분석하였을 때 누락되거나 잘못 추출된 특징 벡터들이 있다. 이에 따라서, 깨진 데이터를 제외하고 나머지 998개의 학습 데이터와 823개의 실험 데이터로부터 특징 벡터를 직접 추출하였다. 영상 모달리티에 대해서는 동영상으로부터 초당 1개의 정지화면을 저장하여 ILSVRC 2012[11] 데이터로 선 학습(Pre-training)한 Google의 Inception-v3[12] 모델의 'pool3' 계층으로부터 특징 벡터를 추출하였다. 음성 모달리티에 대해서는 openSMILE[13]을 이용하여 매 0.010초당 0.025초의 구간에 해당하는 음성 데이터에 대해 20차원의 MFCC 특징 벡터를 추출하였다. 각 모달리티 별로 추출한 특징 벡터는 시간에 대해 동기화가 되어 있지 않으므로, 음성 모달리티 특징 벡터 100개를 이어 붙여서 1초의 구간에 해당하도록 총 2000차원의 특징 벡터를 생성하여 영상 모달리티 특징 벡터와 동기화를 하였다.

4.3 구현

본 연구에서 제안한 모델의 학습은 GeForce GTX1070 GPU, Intel i7-3770k CPU, 32GB RAM의 사양을 가진 PC에서 진행되었다. 모든 알고리즘은 Ubuntu 16.04 LTS 환경에서 Google의 Tensorflow[14] 딥 러닝 라이브러리와 MATLAB을 이용하여 구현하였다.

5. 실험 방법

5.1 투표를 이용한 동영상 분류

각 동영상에 대하여 하나의 특징 벡터를 사용하는 방법 대신에, 동영상 $\boldsymbol{v}=\{\boldsymbol{v}_1,\boldsymbol{v}_2,...,\boldsymbol{v}_N\}$를 이루는 각각의 특징 벡터들 $\boldsymbol{v}_i$에 대하여 softmax 분류기를 설계하고 학습하여 이를 실험에 이용하였다. 즉, 각 특징 벡터 $\boldsymbol{v}_i$를 입력으로 하고 클래스를 나타내는 10차원 벡터를 출력으로 하는 1개의 계층으로 구성되어 있는 신경망을 설계하고, 학습 데이터에 대해 이 신경망을 학습하고 실험 데이터에 대해 분류를 진행하였다. 하나의 동영상은 여러 개의 특징 벡터로 구성되어 있으므로, 동영상을 기준으로 분류를 할 때에는 가장 많은 특징 벡터가 예측한 클래스를 해당 동영상의 클래스라 정의하였다. 즉, 투표와 같은 방식으로 다수결로 한 동영상의 클래스를 예측하였다. 하나의 동영상을 예측한 클래스를 c, 동영상을 $\boldsymbol{v}=\{\boldsymbol{v}_1,\boldsymbol{v}_2,...,\boldsymbol{v}_N\}$, 그리고 동영상을 이루는 각 특징 벡터의 예측 클래스를 c_i라 할 때, c는 Equation (7)과 같이 정의할 수 있다.

$$\arg\max_c \left|\{c_i=c | i\in\{1,2,...,N\}\}\right| \tag{7}$$

이 분류 방법의 장점은 동영상 분류에 사용할 수 있는 증거가 여러 개가 있다는 것이다. 사람의 경우 동영상을 구분하는 데에 있어서 영상 모달리티의 역할이 크다. 때로는 한 화면만 보고도 분류가 가능하기에, 동영상으로부터 추출한 영상 모달리티 벡터 집합이 주어져 있을 경우, 이 방법을 이용한 분류 정확도가 다소 높을 것으로 예상할 수 있다. 본 연구에서 제안하는 학습 모델은 여러 개의 벡터들로부터 이들의 정보를 담고 있는 하나의 특징 벡터를 생성하는 것을 목적으로 하므로, 이 같은 점에서 위 실험 방법은 제안한 모델과의 동영상 분류 정확도 비교에 적합하다고 볼 수 있다. 이 실험은 영상 모달리티와 음성 모달리티 각각에 대해 독립적으로 진행하였다.

5.2 특징 벡터의 평균 폴링을 이용한 분류

이 실험은 각 동영상을 나타내는 특징 벡터의 각 성분 값을 동영상을 이루는 특징 벡터들의 각 성분에 대한 평균값으로 하는 벡터라 정의하고 진행하였다. 즉, 동영상을 $\boldsymbol{v}=\{\boldsymbol{v}_1,\boldsymbol{v}_2,...,\boldsymbol{v}_N\}$라 할 때, 동영상을 나타내는 특징 벡터를 $\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{v}_i$라 정의하였다. 동영상을 이루고 있는 영상, 음성 모달리티 각각에 대하여 위와 같은 특징 벡터를 생성하여 앞서 설명한 실험 방법들과 마찬가지로 각 모달리티에 대해 독립적으로 실험을 진행하였다.

5.3 GRU 유닛을 사용한 LSTM 구조를 이용한 분류

동영상은 일종의 시계열 데이터라 볼 수 있다. $\boldsymbol{v}=\{\boldsymbol{v}_1,\boldsymbol{v}_2,...,\boldsymbol{v}_N\}$라 하고, $\boldsymbol{v}_i$에서 i가 시간을 나타낸다고

도 생각할 수 있다. 특히, 데이터의 전처리를 4.2절과 같이 하였으므로, 동영상을 이루는 특징 벡터는 시간에 대하여 순차적인 특성을 갖고 있다고 이야기 할 수 있다. 그러므로 동영상을 분류하는 작업에 있어서도 LSTM 신경망 구조를 사용하는 것을 고려해 볼 수 있다. 본 연구에서는 동영상에 대하여 각 모달리티를 이루는 특징 벡터 집합에 대하여 LSTM 중에서도 최근에 제시되어 좋은 성능을 보이고 있는 GRU unit을 사용한 LSTM 신경망 구조를 학습하여 동영상 분류 실험을 진행하였다. 이 신경망에 대해 설정할 수 있는 값 중에서는 은닉 계층이 갖고 있는 뉴런의 수가 있는데, 이는 입력 벡터의 차원과 동일하게 설정하여 실험을 진행하였다. 즉, 영상 모달리티에 대해서는 뉴런의 수를 2048로, 음성 모달리티에 대해서는 뉴런의 수를 2000으로 설정하여 실험을 진행하였다. 실험은 각 모달리티에 대하여 독립적으로 진행하였다.

5.4 유사도 측정 기반의 동영상 데이터 표현을 이용한 분류

이 실험은 본 연구에서 제안한 학습 알고리즘을 이용하여 각 동영상에 대해 하나의 특징 벡터를 학습하여 진행하였다. 또한, 본 연구에서 제안한 학습 알고리즘은 비지도 학습 기반 알고리즘인데, 이 알고리즘으로 학습한 특징 벡터의 분류 성능을 높이기 위하여 앞서 학습이 된 신경망의 끝단에 분류 층을 추가하여 역전파 알고리즘을 이용해 미세조정(Fine-tuning)을 진행하고 이로부터 특징을 추출하였다. 실험 결과 설명 시에는 두 실험을 구분하기 위해 미세조정을 하지 않은 실험을 'snet-u', 미세조정을 한 실험을 'snet-t'라 명명하였다.

6. 실험 결과 및 분석

6.1 YLI-MED 데이터 집합 실험 결과 및 분석

5장에서 설명한 여러 실험 방법들을 4.1장에서 소개한 데이터 집합에 적용하여 실험하였다. 편의상 각 방법에 대한 명칭으로 5.1장에서 소개한 실험 방법은 'Vote', 5.2장에서 소개한 방법은 'Avg', 5.3장에서 소개한 방법은 'GRU', 5.4장에서 소개한 방법들은 각각 'snet-u', 'snet-t'로 한다.

YLI-MED 데이터 집합을 이용한 각 실험 방법에 대한 클래스별 정확도 및 총 정확도는 Table 3, Table 4와 같다. Table 3은 영상 모달리티 벡터만을 사용하여 실험한 결과이고, Table 4는 음성 모달리티 벡터만을 사용하여 실험한 결과이다.

영상 모달리티 특징 벡터만을 사용해 YLI-MED 데이터 집합을 분류한 경우, Table 3에서 'snet-t'의 분류 정확도가 대부분의 클래스에서 다른 방법들보다 높은 것을 확인할 수 있다. 대부분의 클래스에서 'snet-t'의 분류 정확도가 더 높음에도 불구하고 총 분류 정확도는 'Vote'와 같은데, 이는 'Ev104' 클래스의 데이터 개수가 많기 때문이다. 이를 MAP 수치로 환산한다면, 'snet-t'의 분류 정확도가 79.12%로 가장 높다. 'Vote' 방식은 동영상이 갖고 있는 여러 개의 데이터를 분류하고 이 결과를 바탕으로 하여 투표로 결정하기 때문에 단 하나의 벡터만을 사용하여 분류를 진행하는 'snet-t'보다 더 증거가 많다고 볼 수 있다. 그럼에도 불구하고 'snet-t'의 분류 정확도가 가장 높으므로, 제안한 알고리즘이 동영상을 이루고 있는 각 데이터의 정보를 잘 반영하고 있다고 볼 수 있다.

음성 모달리티 특징 벡터만을 사용해 YLI-MED 데이터 집합을 분류한 경우, Table 4에서 'snet-u'와 'snet-t' 학습 알고리즘을 사용한 분류 정확도가 'Ev105'를 제외한 나머지 모든 클래스에서 가장 높은 분류 정확도를 보이고 있다. 본 연구에서 제안한 'snet-u'와 'snet-t'를 제외한 나머지 방법들과 'snet-t'를 비교했을 때, 최소 10% 이상의 분류 정확도가 차이가 나는 것으로 보아, 음성 모달리티 특징 벡터를 사용한 경우 'snet-t' 알고리즘이 동영상 데이터가 가진 정보를 더 잘 반영하고 있다고 볼 수 있다.

Table 3. Results of Image Modality Experiments on YLI-MED Dataset

Event ID	Vote	Avg	GRU	snet-u	snet-t
Ev101	85.40%	82.48%	81.02%	86.13%	87.59%
Ev102	70.00%	64.00%	48.00%	74.00%	62.00%
Ev103	75.61%	80.49%	70.73%	73.17%	80.49%
Ev104	80.00%	73.08%	80.77%	67.69%	73.08%
Ev105	81.91%	84.04%	82.98%	82.98%	84.04%
Ev106	79.49%	71.79%	79.49%	76.92%	82.05%
Ev107	60.83%	70.00%	55.00%	59.17%	61.67%
Ev108	79.07%	81.40%	81.40%	86.05%	83.72%
Ev109	84.87%	85.71%	84.03%	78.99%	86.55%
Ev110	90.00%	74.00%	62.00%	78.00%	90.00%
Accuracy (%)	78.74%	77.52%	74.12%	75.82%	78.74%
MAP	78.71%	76.70%	72.54%	76.31%	79.12%

Table 4. Results of Audio Modality Experiments on YLI-MED Dataset

Event ID	Vote	Avg	GRU	snet-u	snet-t
Ev101	24.09%	10.81%	16.79%	54.01%	64.96%
Ev102	26.00%	16.00%	22.00%	26.00%	16.00%
Ev103	4.88%	19.51%	7.32%	9.76%	26.83%
Ev104	10.00%	7.69%	11.54%	18.46%	20.00%
Ev105	2.13%	13.83%	20.21%	7.45%	12.77%
Ev106	10.26%	7.69%	15.38%	20.51%	25.64%
Ev107	12.50%	9.17%	18.33%	20.00%	27.50%
Ev108	9.30%	9.30%	13.95%	16.28%	16.28%
Ev109	26.05%	10.08%	11.76%	32.77%	31.09%
Ev110	16.00%	10.00%	8.00%	24.00%	24.00%
Accuracy (%)	15.19%	10.81%	14.95%	25.76%	29.77%
MAP	14.12%	11.40%	14.53%	22.92%	26.51%

6.2 Youtube 데이터 집합 실험 결과 및 분석

Youtube 데이터 집합을 이용한 각 실험 방법에 대한 클래스별 정확도 및 총 정확도는 Table 5, Table 6과 같다. Table 5는 이 데이터 집합의 영상 모달리티 벡터만을 사용하여 실험한 결과이고, Table 6은 이 데이터 집합의 음성 모달리티 벡터만을 사용하여 실험한 결과이다.

영상 모달리티 특징 벡터만을 사용해 Youtube 데이터 집합을 분류한 경우, Table 5에서 'snet-t'의 MAP 수치가 가장 높은 확인할 수 있다. 이 동영상은 YLI-MED 데이터 집합보다 클래스가 많지만 그 숫자는 적어서 분류 정확도가 YLI-MED 데이터 집합보다 높은 편이다. 하지만 그럼에도 불구하고, 'snet-t'로 학습한 특징 벡터로 분류한 정확도가 대부분의 클래스에서 다른 방법들로 분류한 결과보다 높다. 따라서 본 연구에서 제안한 알고리즘으로 학습한 특징 벡터가 동영상 데이터를 이루고 있는 각 데이터의 정보를 잘 반영하고 있다고 볼 수 있다.

Table 5. Results of Image Modality Experiments on Youtube Dataset

Event ID	Vote	Avg	GRU	snet-u	snet-t
Ev101	90.91%	90.91%	69.70%	78.79%	93.94%
Ev102	73.33%	80.00%	53.33%	86.67%	73.33%
Ev103	100.0%	100.0%	80.00%	100.0%	100.0%
Ev104	100.0%	100.0%	93.94%	100.0%	100.0%
Ev105	93.75%	96.88%	81.25%	93.75%	93.75%
Ev106	36.36%	36.36%	18.18%	54.55%	45.45%
Ev107	79.31%	86.21%	27.59%	86.21%	79.31%
Ev108	96.88%	96.88%	90.63%	100.0%	93.75%
Ev109	93.94%	87.88%	72.73%	81.82%	90.91%
Ev110	94.12%	94.12%	85.29%	97.06%	97.06%
Ev111	72.73%	69.70%	69.70%	69.70%	72.73%
Ev112	97.06%	97.06%	91.18%	97.06%	94.12%
Ev113	97.06%	94.12%	91.18%	91.18%	94.12%
Ev114	96.97%	96.97%	93.94%	93.94%	100.0%
Ev115	100.0%	100.0%	88.24%	97.06%	100.0%
Ev116	100.0%	100.0%	84.85%	100.0%	100.0%
Accuracy (%)	91.58%	91.58%	78.19%	90.50%	91.58%
MAP	88.90%	89.19%	74.48%	89.24%	89.28%

Table 6. Results of Audio Modality Experiments on Youtube Dataset

Event ID	Vote	Avg	GRU	snet-u	snet-t
Ev101	27.27%	3.03%	9.09%	27.27%	24.24%
Ev102	0.00%	6.67%	0.00%	0.00%	0.00%
Ev103	0.00%	0.00%	0.00%	20.00%	0.00%
Ev104	3.03%	9.09%	18.18%	30.30%	30.30%
Ev105	15.63%	12.50%	21.88%	40.63%	34.38%
Ev106	0.00%	0.00%	0.00%	9.09%	0.00%
Ev107	17.24%	6.90%	17.24%	17.24%	24.14%
Ev108	3.13%	18.75%	6.25%	37.50%	43.75%
Ev109	39.39%	12.12%	21.21%	27.27%	48.48%
Ev110	11.76%	23.53%	14.71%	5.88%	14.71%
Ev111	24.24%	9.09%	6.06%	36.36%	39.39%
Ev112	2.94%	8.82%	11.76%	32.35%	41.18%
Ev113	41.18%	14.71%	14.71%	50.00%	50.00%
Ev114	0.00%	21.21%	18.18%	36.36%	36.36%
Ev115	0.00%	0.00%	17.65%	17.65%	23.53%
Ev116	12.12%	15.15%	9.09%	42.42%	30.30%
Accuracy (%)	14.04%	11.23%	13.17%	29.16%	31.32%
MAP	12.37%	10.10%	11.63%	26.90%	27.55%

한편, 음성 모달리티의 경우 'snet-t'의 단순 분류 정확도 및 MAP 수치가 가장 높은 것을 볼 수 있다. 음성 데이터의 경우, 짧은 구간만으로는 판별을 하기 힘드므로 보다 긴 구간의 정보가 필요하다. 이 점에서 볼 때, 'snet-t'로 학습한 특징 벡터가 모든 특징 벡터들의 정보를 잘 반영하고 있다고 볼 수 있다.

모든 결과들을 종합하여 봤을 때, 본 연구에서 제안한 알고리즘은 영상 모달리티에 비해 음성 모달리티의 분류 정확도를 더 크게 향상시키고 있다고 볼 수 있다. 이는 퍼스널 미디어라는 데이터의 특성에 기반한 결과라고 볼 수 있다, 퍼스널 미디어는 우선 방송용 동영상과는 달리 재생 시간이 짧고 화면의 전환이 많지 않다. 하지만 음성의 경우는 이와 다르다. 음성은 길이가 짧다 하여도 시간에 따른 신호의 변화가 크다. 본 연구에서 제안한 알고리즘이 동영상 데이터와 같은 복잡한 신호를 가능한 한 정보를 잃지 않고 하나의 특징 벡터로 나타내는 것을 목적으로 한다는 점을 상기한다면, 음성 신호와 같이 신호의 변화가 크더라도 이와 같은 목적을 잘 달성하고 있다고 볼 수 있다.

7. 결론 및 향후 과제

7.1 결론 및 성과

본 연구에서는 동영상 데이터, 그 중에서도 퍼스널 미디어 데이터에 대해 이를 이루고 있는 특징 벡터 열이 주어져 있을 때, 이를 하나의 벡터로 나타낼 수 있도록 하는 신경망 구조와 이를 학습하는 방법을 제안하였다. 특징 벡터들 사이의 유사도를 내적으로 측정하여 이 차이를 줄인다는 간단한 아이디어를 통해 다른 방법들보다도 정보를 잃지 않은 채 보다 간단한 형식으로 동영상 데이터를 나타낼 수 있음을 확인하였다. 유사도를 측정하는 방법은 여러 가지가 있으므로, 내적 기반이 아닌 거리 공간을 정의할 수 있는 다른 측도를 사용한다면, 더 좋은 특징 벡터를 생성할 수도 있을 것이라 기대된다. 본 연구에서 제안한 방법의 이런 특징들은 동영상과 같은 큰 데이터들을 보다 간단한 형태로도 색인하여 보관이 가능할 수 있음을 보여주고 있다.

7.2 향후 과제

본 연구에서 제안한 학습 알고리즘은 퍼스널 미디어 데이터를 이루고 있는 각각의 특징 벡터에 대해 학습을 진행하므로, 하나의 동영상을 학습하는 시간 복잡도가 동영상이 가진 특징 벡터의 개수의 제곱이다. 따라서 학습 시간을 개선하기 위해서는 이를 병렬화 할 수 있는 방법이 필요하다. 또한, 보다 좋은 특징 벡터를 학습하기 위해 다른 유사도 측정 방법을 생각해 볼 수 있을 것이다. 마지막으로, 동영상 데이터의 특성을 고려할 때 멀티 모달 학습을 진행한다면 단일 모달 학습보다 좋은 성능을 낼 수 있으리라 기대된다. 하지만 본 연구에서는 단일 모달리티만을 고려하였으므로, 이를 멀티 모달 학습 알고리즘으로 확장시키는 것에 대한 연구가 필요하다.

References

[1] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE transactions on Pattern Analysis and Machine Intelligence*, Vol.35, No.8, pp.1798-1828, 2013.

[2] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” In Advances in Neural Information Processing Systems (pp. 1097-1105), 2012.

[3] Y. L. Boureau, and Y. L, Cun, “Sparse feature learning for deep belief networks,” *Proc. of Advances in Neural Information Processing Systems*, pp. 1185-1192, 2008.

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” arXiv preprint arXiv:1412.3555, 2014.

[5] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, Vol.9, No.8, pp.1735-1780, 1997.

[6] J. Bernd, D. Borth, B. Elizalde, G. Friedland, H. Gallagher, L. Gottlieb, A. Janin, S. Karabashlieva, J. Takahashi, and J. Won, “The YLI-MED corpus: Characteristics, procedures, and plans,” arXiv preprint arXiv:1503.04250, 2015.

[7] C. Goller, and A. Kuchler, “Learning task-dependent distributed representations by backpropagation through structure,” *Proc. of IEEE International Conference on Neural Networks*, pp.347-352, 1996.

[8] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol.6, No.2, pp.107-116, 1998.

[9] K. Ashraf, B. Elizalde, F. Iandola, M. Moskewicz, J. Bernd, G. Friedland, and K. Keutzer, “Audio-based multimedia event detection with DNNs and sparse sampling,” *Proc. of the 5th ACM on International Conference on Multimedia Retrieval*, pp.611-614, 2015.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, Vol.323, No.9, pp.533-536, 1986.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Reconition Challenge,” *International Journal of Computer Vision*, Vol.115, No.3, pp.211-252, 2015.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” arXiv preprint arXiv:1512.00567, 2015.

[13] F. Eyben, M. Wöllmer, and B. Schuller, “Opensmile: the munich versatile and fast open-source audio feature extractor,” *Proc. of the 18th ACM International Conference on Multimedia*, pp.1459-1462, 2010.

[14] Abadi, M., Agarwal, A., et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” Available: http://tensorflow.org (retrieved 2016, Feb. 2)

이 호 석

http://orcid.org/0000-0003-0377-5293

e-mail : jaerom89@gmail.com

2014년 서강대학교 수학과(학사)

2017년 서강대학교 컴퓨터공학과(석사)

2017년~현 재 쿠팡(주) 재직

관심분야 : 기계학습, 패턴인식, 인공지능

양 지 훈

http://orcid.org/0000-0002-0300-7705

e-mail : yangjh@sogang.ac.kr

1983년 서강대학교 전자계산학과(학사)

1989년 Iowa State University Computer Science(석사)

1999년 Iowa State University Computer Science(박사)

2002년~현 재 서강대학교 컴퓨터공학과 교수

관심분야 : 기계학습, 인공지능, 패턴인식, 데이터마이닝, 생물정보학