CNN과 RNN의 기초 및 응용 연구

□ 이은주 / 계명대학교

Ⅰ. 서 론

2016년 3월, 전 세계적으로 굉장히 이슈가 되는 사건이 있었다. 다름 아닌, 구글 딥 마인드(Deep Mind)가 개발한 인공지능 바둑 프로그램인 알파고(AlphaGo)와 이세돌 9단의 바둑 대결에서 컴퓨터가 4대 1이라는 압승을 거둔 것이다. 이때, 일반 대중들에게 바둑에 대한 관심 못지않게 오래된 패러다임으로 생각되었던 인공지능에 대한 관심이 폭발적으로 증가하게 되었다[1]. 이미 컴퓨터에게 사람의 지능을 닮아가게끔 하는 연구는 끊임없이 진행되었지만 단순하게 입력된 정보를 추론하고 검색하는 기계에 불과했다. 이에 반해 알파고는 다양한 경우의 수를 추론하는 것뿐만 아니라 스스로 생각하고 더 나아가 인간이 생각할 수 있는 지능의 한계를 뛰어 넘을 수 있는 강력한 기술을 실현한 것이다. 이 기술이 바로 딥 러닝이다.

딥 러닝은 인간의 두뇌와 유사한 사고방식을 컴퓨터에 적용하여 홍수처럼 쏟아지는 데이터를 분석하는 인공지능 기술을 말한다. 많은 개발자들이 딥 러닝 기술을 추천하는 이유는 디지털 경제의 확산으로 학습 가능한 정보가 방대해졌으며 많은 정보를 실시간으로 처리할 수 있는 GPU 병렬처리 기술이 발달되었고 무엇보다 네트워크 구조가 간단하면서도 우수한 성능을 낼 수 있기 때문이다.

딥 러닝의 핵심 기술은 입력 데이터가 고차원이고 구조가 복잡한 경우 전 처리 과정에서 정보 손실이 될 수 있는 정보를 전체 학습과정에 포함시키면서 입력 데이터를 직접 학습하도록 통합하는 것이다. 이러한 딥 러닝 기술은 컴퓨터 비전 분야에서는 영상 인식(Video Recognition), 객체 추적(Object Tracking), 자율 주행 자동차(Self-driving Car) 플랫폼 등 이미 다양한 분야에 적용되어 실효성이 입증되었다.

따라서 본 논문에서는 딥 러닝을 학습하고자 하는 연구자들을 위해 실시간 영상 처리를 위한 최근의 딥 러닝 기술을 소개하고 효율적인 딥 러닝 시스템에 대한 연구 방향을 제시하고자 한다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 딥 러닝 기술을 대표하는 전통적인 CNN(Convolutional Neural Network) 및 CNN의 단점을 개선한 RNN(Recurrent Neural Network)의 개념과 구조에 대해 설명한다. Ⅲ장에서는 딥 러닝 모델을 이용한 응용 사례, 마지막으로 Ⅳ장에서는 결론으로써 향후 연구 방향을 제시한다.

Ⅱ. 딥 러닝 모델의 구조

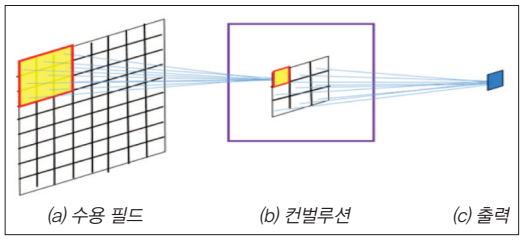
1. CNN의 구조

CNN은 이미지로부터 고수준의 추상화된 특징을 추출하거나 질감 정보를 처리하는 최적의 방법으로써 이미 2012년 ILSVRC(Imagenet Large Scale Visual Recognition Challenge)에서 기존 알고리즘들을 압도적으로 제치고 객체 인식에 대단히 뛰어난 성능이 검증되었다[2].

CNN은 지역적 수용 필드(Local Receptive Field), 가중치 공유(Shared Weight), 서브샘플링의 세 가지 아이디어가 적용되었다. 지역적 수용 필드, 공유 가중치를 반영하는 컨벌루션 계층과 서브샘플링 계층은 여러 층으로 적재가 가능하며 마지막 단계에서 완전 연결 계층(Fully-Connected Layer)을 통해 분류 수행하는 계층 모델이다. CNN은 벡터 형태의 입력 데이터 대신 2차원 구조의 입력이 가능하므로 신경망이 영상을 잘 학습할 수 있도록 최적화시킨 알고리즘이다. 다음은 CNN의 각 계층에 대한 설명이다.

첫 번째, 컨벌루션 계층은 객체의 위치나 크기에 영향을 받지 않고 입력 정보에서 찾아내고자 하는 객체의 에지나 코너, 선의 끝과 같은 특징을 특징 맵(Feature Map)에 표현하기 위한 계층이다. 이 계층의 입력은 원본 영상이거나 이전 계층의 출력 즉 특징 맵이다. 다 계층 인공 신경망과는 달리 인접한 픽셀끼리의 지역(Local)적 특성을 반영하고자 2차원 배열 형태의 입력 영상에 N×N 크기의 필터를 슬라이딩 윈도우 방식으로 컨벌루션 연산을 한다[3]. 특징 맵에서 컨벌루션 필터의 가중치(Weight)는 공유되어지고 역전파(Back-propagation)를 통해 학습된다.

컨벌루션 연산은 영상 내의 모든 픽셀에 대해 반복적으로 처리하게 되는데 컨벌루션되는 필터의 개수가 많아지면 다양한 종류의 특징을 추출할 수도 있다. 2차원 입력 영상에 대해서 N×N 크기의 필터를 모든 가능한 위치에서 컨벌루션 연산이 수행될 수 있도록 이동시키므로 출력되는 특징 맵의 크기는 (입력 영상의 크기 − 필터 크기 +1)이 된다. 〈그림 1〉은 3×3 필터를 이용한 컨벌루션 연산 과정을 보여준다.



〈그림 1〉 필터를 이용한 컨벌루션 과정 (a) 지역적 수용필드 (b) 컨벌루션 (c) 출력

컨벌루션 필터링된 값들은 수식 (1)과 같이 비선형 변환함수인 Relu[4]를 이용하여 활성화된다.

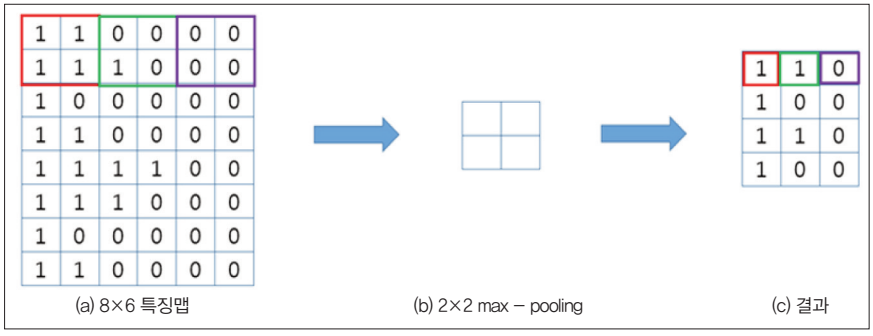
$$f(x) = \max(0, x) \quad (1)$$

이 때, x는 컨벌루션 연산을 통해 도출된 출력 값이며 뉴런이 선형적으로 활성화되어 큰 값을 가질 수 있게 함으로써 역전파를 해도 기울기(Gradient)가 사라지지 않도록 한다.

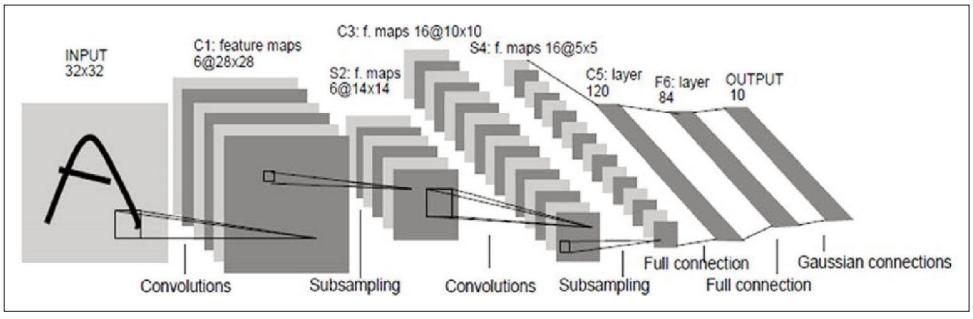
두 번째, 서브샘플링 계층은 입력 픽셀을 2×2 크기의 서브 영역으로 분할하고 각 서브 영역에서 픽셀의 최댓값, 최솟값 또는 평균값을 추출하여 해당 영역의 한 점으로 매핑(Mapping)한다. 이것은 뉴런이 가장 큰 신호에 반응하는 것과 유사하며 노이즈 감소 및 속도 향상, 영상의 분별력 또한 증가한다. 〈그림 2〉는 2×2 필터를 이용한 Max-Pooling 연산 과정을 보인다.

세 번째, 완전 연결 계층은 CNN의 마지막 계층으로 분류를 위한 과정을 수행한다. 서브샘플링 계층에서 나온 1차원 벡터 특징들을 이용해서 영상을 클래스별로 분류를 할 때 사용되는데, 일반적인 다계층 인공 신경망의 입력처럼 각각 하나씩 매핑한다. 〈그림 3〉은 전통적인 CNN 구조이다.

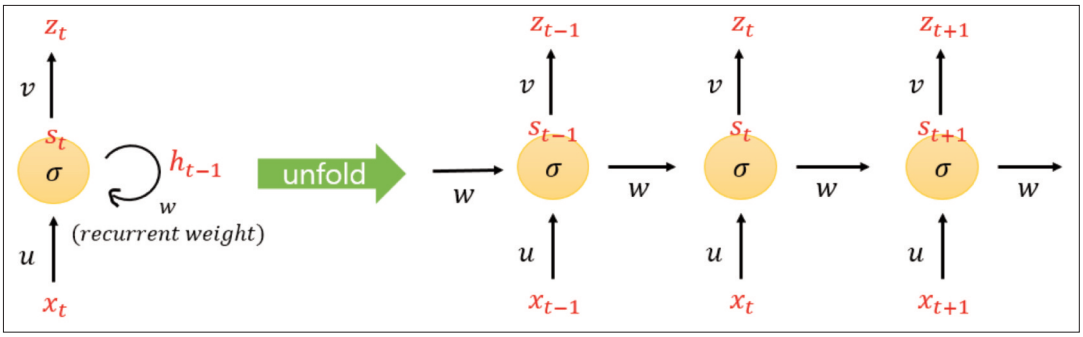
CNN을 기반으로 만들어진 대표적 딥 러닝 프레임워크 Caffe[6]는 현재 가장 대중적인 딥 러닝 애플리케이션으로 인정받고 있으며 사람이 손으로 쓴 필기체 숫자인 성능 테스트용 데이터 MNIST(Mixed National Institute of Standards and Technology Database)를 학습시키고 결과를 테스트해 볼 수 있도록 리눅스 혹은 윈도우 기반으로 설계되었다. Caffe기반 알렉스넷(AlexNet)[7]은 ImageNet에서



〈그림 2〉 2×2 필터를 이용한 Max-Pooling 과정



〈그림 3〉 전통적인 CNN 구조[5]



〈그림 4〉 RNN의 기본 구조 [8]

정확도 84.7%을 기록하며 10여 년 동안 깨지지 않은 기록 75%대의 벽을 허물었다.

하지만 CNN의 학습과정은 순서가 중요하지 않은 정보들이 공통으로 가지는 특징들만 관심이 있기 때문에 실시간으로 들어오는 정보들의 순서 관계를 처리할 수 없는 문제점이 있다. 이에 과거 및 현재 정보를 기반으로 미래 정보를 예측하는 시계열 특성을 반영한 RNN[8] 딥 러닝 모델을 소개하고자 한다.

2. RNN의 구조

RNN은 어떤 특정 부분이 반복되는 구조를 통해 순서를 학습하기에 효과적인 딥 러닝 기법이다. 〈그림 4〉는 은닉층(Hidden layer)에서 자기 자신으로의 *W*(Recurrent weight)를 가진 기본 구조로서 CNN과 동일한 방법으로 데이터에서 규칙적인 패턴을 인식하면서 *W*를 통해 과거의 정보를 통해 현재의 정보 파악에 도움을 받는 구조가 될 수 있다.

x_t는 현재 time step *t*에서의 입력 값, s_t는 메모리 역할을 하는 은닉층으로써 수식 (2)와 같이 s_{t-1}(이전 time step *t*)와 x_t에 의해 계산된다.

$$s_t = f(Ux_t + Ws_{t-1}) \quad (2)$$

이 때, *f*는 비선형 함수로서 tanh나 *ReLU*가 사용된다. RNN의 은닉층은 입력된 정보를 통해 계산된 결과에 대해서 저장할 수 있는 기능이 있어 짧은 시퀀스를 효과적으로 처리할 수 있다. z_t는 time step *t*에서의 출력 값으로 수식 (3)에 의해 계산되어 진다.

$$z_t = softmax(Vs_t) \quad (3)$$

RNN은 각 time step마다 가중치 *U*, *V*, *W*가 공유되어지기 때문에 역전파 알고리즘의 변형인 Back-propagation Trough Time(BPTT)을 통해 학습한다.

RNN은 각 출력 부분의 기울기는 현재 time step 이외에 이전 time steps에 매우 의존적이다. 많은 수의 뉴런 유닛이나 많은 수의 입력 유닛이 있는 경우 과거 학습 기능을 통해 반복적으로 곱해지는 가중치에 의해 에러값이 1보다 클 경우 누적에러가 기하급수적으로 증가(Gradient Exploding)하거나, 1보다 작을 경우 누적에러가 감소하여 빠르게 0으로 수렴(Gradient Vanishing)하는 문제가 발생할 수

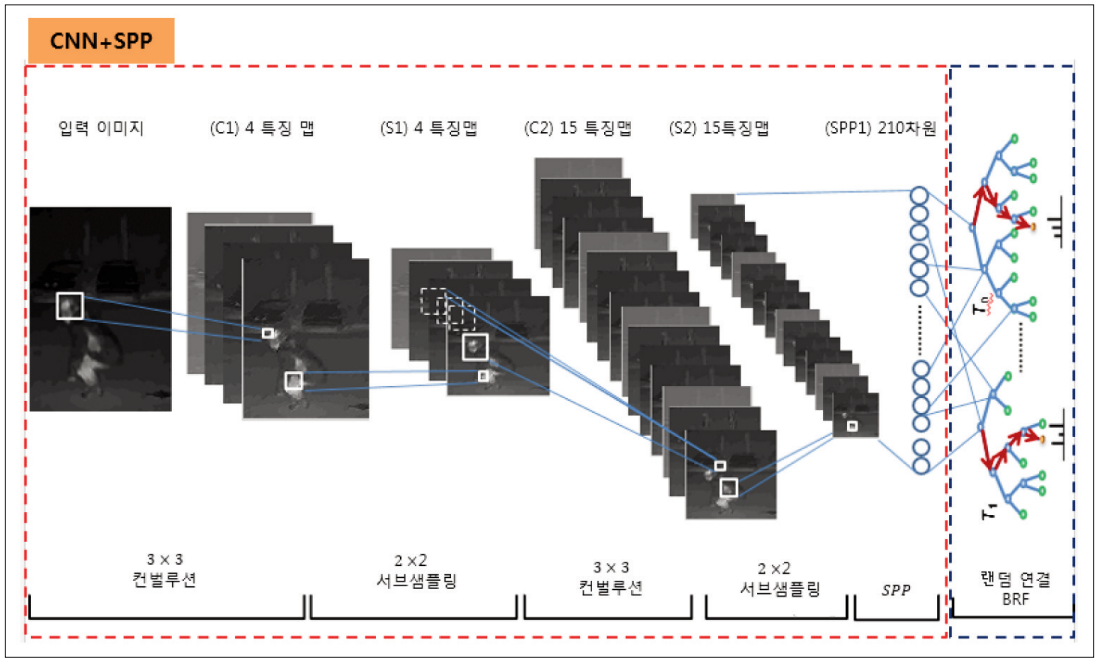
있다. 이러한 문제 해결을 위해서 LSTM(Long Short Term Memory)[9]가 제안되었다. LSTM 모델은 Ⅲ장 RNN의 응용 사례에서 간단히 기술한다.

Ⅲ. 응용 사례

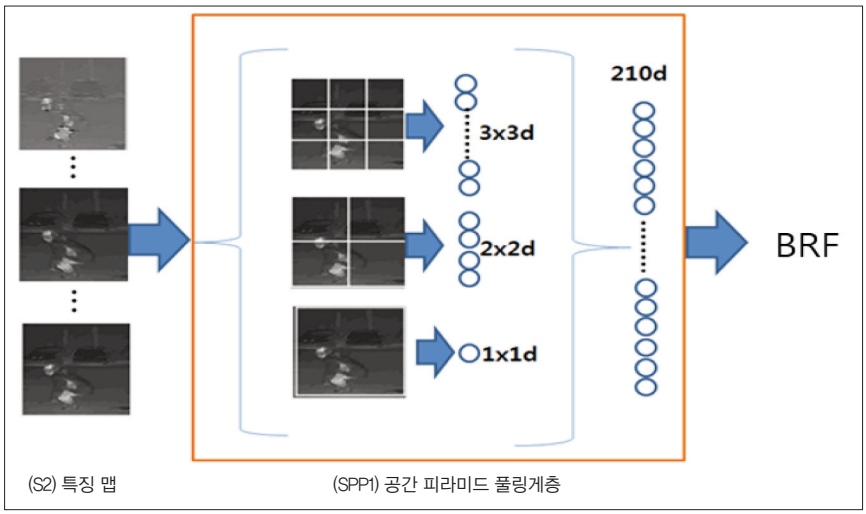
CNN과 같은 딥 러닝 기술은 대량의 데이터를 의미있는 데이터의 형태로 표현하고 학습하고자 하는 노력으로 영상이나 음성 인식과 같은 다양한 분야에 적용되었다. 하지만 이러한 최신 기술도 고차원 데이터에 대한 학습 시간이 많이 소요되며 과적합, 기울기 감소에 대한 문제점들이 발생한다. 이점을 해결하기 위하여 Ⅲ장에서는 최소한의 전처리를 하도록 설계된 다 계층 퍼셉트론으로 BRF를 결합하여 기울기 감소 문제 및 시간 단축의 효과를 거둘 수 있는 RC-CNN[10]을 소개한다. 〈그림 5〉는 자율 주행 차량 애플리케이션에서 보행자의 위험 행동을 인식하기 위한 경량의 RC-CNN 구조도이다.

보행자를 인식하는 대부분의 방법들은 모션(Motion) 정보를 이용한다. 하지만 차량의 진동, 차량의 속도, 보행자의 다양한 형태로 인해 보행자의 모션 정보를 정확하게 추정하기 어려우므로 '걷기', '서기', '달리기'와 같은 정적 이미지를 보행자의 의도를 추적하기 위한 단서로 삼는다.

RC-CNN은 1차 컨벌루션 계층(C1), 1차 서브샘플링 계층(S1), 2차 컨벌루션 계층(C2), 2차 서브샘플링 계층(S2), 공간 피라미드 풀링 계층(SPP1), BRF 계층으로 구성된다. 전통적인 CNN과의 차별화된 3가지를 소개한다.



〈그림 5〉 RC-CNN 네트워크 구조[10]



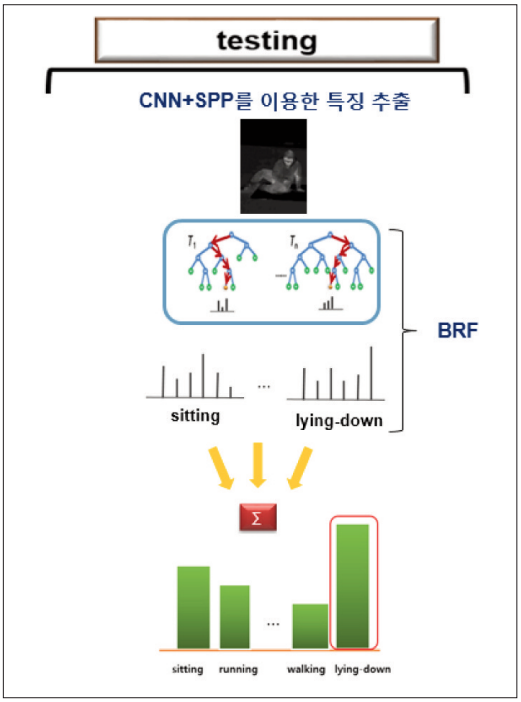
〈그림 6〉 SPP를 이용한 출력 특징 벡터 생성[10]

첫 번째, 전통적인 CNN은 항상 고정된 크기가 입력되어지는 반면 RC-CNN은 지역 및 전역적 공간 특성을 잘 반영하여 입력 이미지의 크기에 구애받지 않도록 공간 피라미드 풀링(Spatial Pyramid Pooling)[11] 방식을 사용하여 세분화된 특징을 추출한다.

두 번째, 역 전파 학습 알고리즘을 사용하는 대신 반복학습 과정을 줄이고 응용시스템에 적합한 최적의 필터를 생성하기 위하여 GA(Genetic Algorithm) 필터뱅크 업데이트를 수행한다. 필터는 9개의 값으로 이루어져 있으며 LeCun 등[3]이 제시한 방법과 동일한 방법인 9개의 랜덤 값으로부터 최적의 필터 계수 추정법을 사용하였다. 먼저 0~255의 범위를 9개의 영역으로 나눈다. 9개의 영역에서 임의의 값을 추출한 후 그 값의 음수와 양수 값을 이용하여 하나의 필터 값(4쌍)을 구성한다. 같은 방법으로 100번의 필터 생성 과정을 거쳐 만들어진 필터들 중에서 성능이 우수한 필터를 선택한다. 전통적인 CNN의 경우 End-to-End 방식의 오류 역 전파 학습인 반면 제안된 필터뱅크 방식의 학습은 입력 계층에서 GA 알고리즘으로 생성된 필터의 값은 단지 마지막 계층의 부스티드 랜덤 포레스트를 거쳐 나온 정확률(적합도)만을 가지고 학습 여부를 판단하게 된다. 추가 학습이 필요할 경우 입력 계층에서 GA 연산을 통해 이전에 생성된 필터 값에서 진화된 필터로 다시 학습을 진행하는 작업을 반복한다.

세 번째, 추출된 특징 벡터를 완전 연결 방식의 멀티 레이어 퍼셉트론 방식 대신에 랜덤 특징을 선출하여 분류에 적용하는 BRF(Boosted Random Forest)[12]와 결합하여 처리 속도를 향상시키고 성능을 높이는 알고리즘을 적용하였다. C1~SPP1의 단계를 거쳐 특징벡터를 생성하고 각 트리에 값을 입력 시켜 나온 확률 값들을 〈그림 7〉과 같이 누적하여 최종 분류 클래스를 결정하도록 한다.

RC-CNN은 전통적인 CNN에서 고정된 영상 사이즈를 입력 받아야 하는 제약을 넘어서 다양한 입력 크기에도 불구하고 처리속도 개선뿐만 아니라

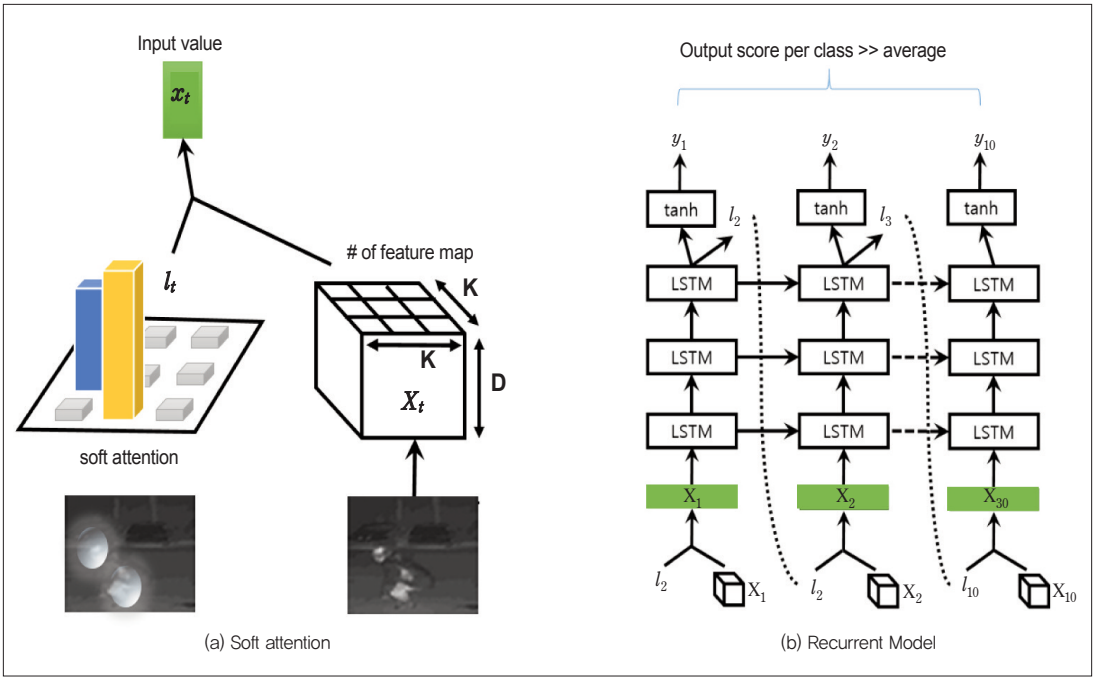


〈그림 7〉 보행자 위험 행동 분류를 위한 테스트 과정[10]

SPP와의 융합으로 인한 견고한 인식이 가능함을 알 수 있었다.

다음으로 소개할 응용 사례는 Sharma 등[13]이 제안한 행동인식을 위한 Recurrent Model이다. 실제 사람은 객체를 인식할 때 배경을 포함한 모든 정보를 사용하여 인식하는 것이 아니라 자신이 관심(Attention)있는 부분을 시간 흐름에 따라 포커싱하면서 그 sequence들을 종합하여 결론 내리는 것을 알 수 있다. 이 논문은 CNN처럼 전체 이미지를 한 번에 처리하지 않고 각 time step마다 관심 있는 영역을 감지하여 처리한다. 〈그림 8〉은 시각적 관심(Visual Attention)을 이용한 RNN 구조도이다.

〈그림 8〉(a)의 결과는 수식 (4)의 soft attention($l_{t,i}$)과 CNN 메커니즘에 의해 생성된 X_T 큐브(D개의 특징맵)에 의해 현재 time step t의 입력(X_t)이 된다(수식 (5) 참조).



〈그림 8〉 Recurrent 네트워크 구조[13]

$$l_{t,i} = p(L_t = i \mid h_{t-1}) = \frac{\exp(w_i^T h_{t-1})}{\sum_{j=1}^{k\times k} \exp(w_j^T h_{t-1})} \quad i \in 1 \ldots K^2 \quad (4)$$

$$X_t = E_{p(L_t \mid h_{t-1})}[X_t] = \sum_{i=1}^{k^2} l_{t,i} X_{t,i} \quad (5)$$

전통적인 RNN과 차별화 된 점은 은닉층이 LSTM(Long Short Term Memory)으로 구성되어 있다. LSTM의 핵심은 셀 상태(Cell State)에 있다. 셀은 여러 개의 게이트(Gate)라는 요소를 활용하여 정보를 불러오거나 정보를 기억하거나 더 이상 필요없는 기억된 정보의 제거 기능을 통하여 선택적 정보 수집 및 유지가 가능하다. 각 게이트가 갖는 가중치는 이전 모델의 은닉층과 같은 원리로 역전파 알고리즘을 통한 학습이 이루어진다.

〈그림 8〉(b)의 마지막 LSTM을 통해 나온 출력값 l_2 중 하나는 다음 time step의 입력으로 나머지 하나는 softmax를 적용하여 현재 time step의 클래스별 스코어를 계산한다.

최종적으로 관심 행동 예측은 모든 time step을 거쳐 나온 확률 값의 평균을 이용하여 최종 행동 예측을 하게 된다.

Ⅳ. 결 론

본 논문은 최근 들어 인식 및 예측 분야에서 많은 각광을 받고 있는 딥 러닝 모델 CNN(Convolutional Neural Network)과 RNN(Recurrent Neural Network) 알고리즘을 자세히 살펴보았다. 특히, 무인 자동차에서 보행자의 위험 행동을 인식하는데 CNN을 실시간으로 사용할 수 있도록 구현하였으며 또한 데이터에서 규칙적인 패턴을 인식하면서 Recurrent weight를 통해 현재의 정보 파악에 도움을 받는 구조로써 공간적 특성뿐만 아니라 시계열 특성을 고려하기 위한 관심(Attention) 기반 RNN 구조에 대해 알아보았다.

관심(Attention) 기반 RNN 모델은 아주 오랜 시간으로부터의 장기 보존 여부 및 정보 메모라이징 문제를 야기했던 전통적인 RNN의 대안으로 LSTM (Long Short Term Memory) 모델을 사용하고 있다. 현재 Window버전 Caffe를 활용하여 보행자의 위험 행동을 예측하는 방법으로 기존의 RC-CNN의 공간적 특성과 LSTM의 RNN화를 통한 시간적 특성을 접목하여 time step마다 출력된 확률 값의 결합으로 더 높은 성능 향상을 위한 연구가 진행 중에 있다.

참 고 문 헌

[1] 연합뉴스, "<알파고 충격>①인공지능,마침내 인간을 넘어서다," http://www.yonhapnews.co.kr/bulletin/2016/03/14/0200000000AKR20160314161200017.html, 2016.03.15.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, et al, "Imagenet large scale visual recognition challenge," International Journal of Computer Vision, Vol 15, No 3, pp. 211-252, 2015.

[3] P. Sermanet, Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," Proc. of The International Joint Conference on Neural Networks, pp. 2809-2813, July, 2011.

[4] X. Glorot, A. Bordes, Y. Bengio, "Deep sparse rectifier networks," Proc. of the 14th International Conference on Artificial Intelligence and Statistics, Vol. 15, pp. 315-323, 2011.

[5] Y. LeCun, et al, "Gradient-based learning applied to document recognition," Proc. of the IEEE, 1998.

[6] Y. ia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, et al, "Caffe: Convolutional architecture for fast feature embedding," Proc. of The International Conference on Multimedia, pp. 675-678, November, 2014.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classication with deep convolutional neural networks," In Proc. Neural Information Processing System, 2012.

[8] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural Networks, Vol 61, pp. 85-117, 2015.

[9] M. Baccouche, et al, "Sequential deep learning for human action recognition." International Workshop on Human Behavior Understanding. pp. 29-39, 2011.

[10] E. J. Lee, B. C. Ko, J. Y. Nam, "Recognizing pedestrian's unsafe behaviors in far-infrared imagery at night," Infrared Physics & Technology, Vol 76, pp. 261-270, 2016.

[11] K. He, X. Zhang, S. Ren, J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 37, No 9, pp. 1904-1916, 2015.

[12] Y. Mishina, M. Tsuchiya, H. Fujiyoshi, "Boosted Random Fores," Proc. of the International Conference on Computer Vision Theory and Applications, pp. 594-598, January, 2014.

[13] S. Sharma, R. Kiros, R. Salakhutdinov, "Action recognition using visual attention," arXiv preprint arXiv:1511.04119, 2015.

필 자 소 개

이 은 주

– 2000년 : 계명대학교 컴퓨터공학과 졸업(공학사)
– 2004년 : 계명대학교 대학원 컴퓨터공학과 졸업(공학석사)
– 2016년 : 계명대학교 대학원 컴퓨터공학과 졸업(공학박사)
– 2016년 9월 ~ 현재 : 계명대학교 산업기술연구소 연구원
– 주관심분야 : 컴퓨터 비전 및 패턴인식