딥러닝이란 무엇인가

초안: 2017.12, 수정: 2018.11

최희열 (한동대학교 전산전자공학부)

알파고의 충격 이후, 인공지능과 그 주요 기술인 딥러닝에 대한 관심이 매우 높다. 하지만 수학과 프로그래밍에 대한 배경 지식없이 딥러닝을 접하기가 쉽지않은 상황에서 초보자들도 쉽게 이해할 수 있도록 딥러닝을 소개한다.

1. 서 론

최근 '인공지능이 인류문명을 멸망시킬 것인가'에 대한 질문에 서로 다른 생각을 가진 주커버그(Mark Zuckerberg)와 머스크(Elon Musk)의 설전이 사회관계망을 뜨겁게 했다 [32]. 이러한 논쟁은 1956년 다트마우스 학회(Dartmouth Conference)에서 처음 정의된 인공지능이 최근 얼마나 급격하게 발전하는지를 잘 보여준다. 인간의 지능을 모사하려는 인공지능 기술은 오래전부터 점진적으로 발전해 왔고 1997년 딥블루(Deep Blue)가 당시 체스 세계 챔피언이였던 카스파로브(G. Kasparov)를 이기면서 잠시 대중의 관심을 끌었다. 딥블루는 체스 고수들의 지식을 바탕으로, 엄청난 양의 경우의 수를 계산하는 컴퓨팅 파워에 의존하기 때문에 알고리즘 측면으로는 높은 평가를 받기 어려웠다. 그에 비해, 아이비엠(IBM)의 왓슨(Watson)이나 애플(Apple)의 시리(Siri), 구글(Google) 나우(Now), 구글의 알파고 등의 인공지능 기술은 해당 분야의 전문가에 의존하는 대신 빅데이타에 기반하여 지식을 자동으로 축적하며, 컴퓨팅 파워에 의존한 경우의 수 계산만이 아니라 인간과 유사한 형태의 선택을 한다는 면에서 높은 평가를 받고 있다.

앞서 언급한 왓슨의 경우에서 처럼, 최근 패턴인식 성능의 비약적인 향상을 주도해온 것은 데이타 기반 인공지능이다. 1990년대 작은 양의 데이타에 기반한 기계학습 및 패턴인식은 이론적인 발전에도 불구하고, 그 성능이 인간 수준에는 미치지 못했다. 하지만, 2000년대부터 데이타의 대용량화와 빅데이타를 처리할 수 있는 클라우드 및 그래픽 프로세서(Graphical processing unit, GPU)를 활용한 컴퓨팅 파워의 폭발적 증가는, 빅데이타를 처리하지 못하는 기존의 커널 머신과 같은 패턴인식 기술의 한계를 극복하고 패턴인식의 새로운 패러다임을 예견했다 [2, 10]. 딥러닝(deep learning)은 이런 배경에서 음성인식과 영상인식을 비롯한 다양한 패턴인식 분야의 성능을 혁신하는 중요한 인공지능 기술이다.

딥러닝의 초기 개념은 1980년대에, 혹은 이미 그 이전부터 논의되었지만, 2006년 사이언스에 발표된 토론토 대학의 힌턴(G. Hinton) 교수 논문 이후 주목을 받기 시작했고 2012년즘을 거치면서 많은 사람들이 딥러닝을 체계적으로 연구하기 시작했다 [1, 28]. 딥러닝 연구의 성공적인 발전과 함께, 글로벌 IT 업체들이 다양한 서비스들을 제공하기 시작하면서, 딥러닝은 단순히 관련 연구자들 뿐만 아니라, 일반 대중의 관심을 받기 시작했고 주요 미디어에서도 관련 기사들을 쏟아내고 있다 [3, 4, 5]. 현실적으로 분석하고 이해할 방법이 없었던 여러 응용분야의 대용량 데이타에 대해 딥러닝이 분석 도구로 더욱 기대되고 있다.

딥러닝은 기존 천층신경망(shallow neural networks)의 계층수를 증가시켜 심층신경망(deep neural networks) 혹은 심층망(deep networks)을 구성하고, 빅데이타를 이용하여 심층망을 효과적으로 학습하여 패턴인식이나 추론에 활용하는 과정을 말한다. 이런 심층망의 장점은 기존의 천층망에 비해 더 많은 중간 계층을 사용함으로써 입출력 데이터에 대한 모델의 표현 능력을 크게 증가시킬 수 있다는 것이다. 이러한 심층망의 아이디어는 2006년 이전에는 심층망을 학습할 효과적인 방법이 없어서 주목받지 못했다. 2006년 제안된 사전학습(pre-training)은 심층망의 학습 가능성을 보여주었고, 그 이후 여러가지 다양한 학습 방법들이 제안되어 사용되고 있다 [6].

신경망에서 중간계층을 쌓는 것은 단순해 보이지만, 이러한 계층의 증가로부터 오는 양적 변화가 패턴인식 분야에서는 패러다임의 변화를 일으킬 만큼 대단한 혁신으로 이어졌다. 이러한 혁신은 크게 두 가지로 요약되는데, 하나는 해당 분야의 전문가의 지식 없이 데이타로부터 자동적으로 필요한 정보를 추출해낸다는 것이고, 또 하나는 기존에 독립적으로 학습되던 특징 추출기(feature extractor)와 분류기(classifier) 등의 모델들이 하나의 모델로 통합됨으로써 패턴인식의 성능이 극대화 된다는 점이다. 사실 이 두가지 측면은 동전의 양면과 같아서, 전문가의 지식이 필요없어진 것은 하나의 큰 모델을 통해 목적에 필요한 모든 지식을 데이타로부터 직접 추출할 수 있기 때문이고, 하나의 모델로 통합되었다는 것은 전문가의 지식을 대체할 부분이 모델로 흡수되어 더 큰 모델로 통합된 것을 의미한다. 이러한 변화는, 예를 들어, 의료 영상 분석에서 의사들의 사전 지식에 의존하던 기존의 복잡한 패턴인식 방법에서 심층망 학습 만으로 단순화 되면서도 더 정확해지는 것을 의미한다.

본 장에서는 우선 학습에 대한 기본적인 내용들을 살펴보고, 심층망의 발전 역사 및 다양한 개념들, 그리고 학습 원리 및 주요 모델들을 설명하고, 이들이 실제 응용 분야에서 어떻게 적용되는지를 사례 중심으로 설명함으로써 딥러닝에 대한 전체적인 이해를 돕는 것을 목적으로 한다.

2. 배 경

2.1. 학습이란 무엇인가

컴퓨터 프로그램이 학습한다는 것은 주어진 태스크에서 데이타를 볼 수록 성능이 좋아지는 과정을 말한다 [33]. 예를 들어, 음성인식 태스크에서 인식률로 성능을 측정하는 음성인식 시스템에서, 음성 데이타를 볼 수록 인식률이 올라간다면 시스템은 학습 중에 있다. 또다른 면으로 이야기하자면, 데이타를 보기 전에 가진 생각이 데이타를 보고나서 바뀔 때 (예를 들면, 딥러닝에 대해 그전에 가지고 있던 생각이 이 글을 읽고 나서 바뀔 때) 학습하고 있다고 한다. 학습은 잘 만들어진 지식을 전수받으면서 이루어 질 수도 있고, 데이타를 관측하면서 이루어질 수도 있는데, 기계학습이나 딥러닝에서 말하는 학습은 주로 데이타를 관측하면서 이루어지는 학습을 의미한다. 즉 데이타 기반의 인공지능이다.

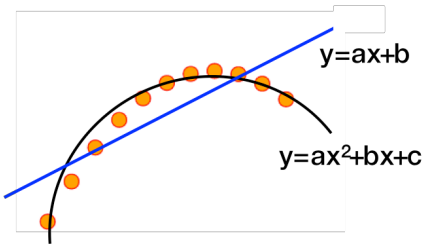

그림 2.1. 1,2차 다항식 모델에 기반한 회귀분석 예. 동그란 점들은 (x,y) 로 이루어진 데이터.

패턴인식과 같은 특정 임무에 관하여 데이터로부터 배운다는 것은 주어진 모델의 변수(parameters)를 조정하여 인식 정확도와 같은 성능의 최대화를 이루어 가는 과정이다. 즉, 성능이 목적함수로 정의되어 함수 최적화를 통해 최적의 변수를 찾아내는 과정으로 요약된다. 예를 들어, 그림 2.1 에서는, 선형모델의 경우 두 개의 변수 (a,b), 2차식에서는 3개의 변수 (a,b,c)를 조정하여 입력 값 x에 대해 출력 값 y를 예측할 수 있게 한다 (예를 들면, 수학점수 x를 보고 영어점수 y를 예측하는 모델). 그림에서처럼, 3개의 변수를 갖는 모델은 2개의 변수를 갖는 모델을 포함하는데, 2차식이 주어진 데이타 x와 y의 관계를 더 잘 설명하는 것처럼, 일반적으로 더 많은 변수를 사용하여 정의된 복잡한 모델일수록 데이타의 관계를 더 잘 설명하고 정의할 수 있다. 하지만, 변수가 많을 수록 학습이 어렵고 과적합(overfitting) 등의 문제까지 발생할 수 있어서 적절한 수준의 모델 복잡도를 찾는 것은 중요하다.

학습을 하려면 데이타를 설명하는 모델을 결정하고 (위 예에서는 1차 혹은 2차식, 다음 장에서는 신경망) 그 모델의 변수 w를 ({a,b} 혹은 {a,b,c}, 신경망에서는 연결강도) 찾아가는 과정으로 요약할 수 있는데, 이때 변수를 찾아가려면 어떤 변수가 더 좋은 변수인지 측정할 수 있어야 한다. 주로 예측값과 실제 정답값 사이의 차이 혹은 오차를 최소화 하는 변수를 좋은 변수라고 생각할 수 있고 이러한 오차를 목적함수 f(w)라고 한다. 이렇게 정의된 오차와 같은 목적함수를 최소화 하려고 변수들을 찾아가는 방법은 주로 경사강하법(gradient descent method)를 사용한다.

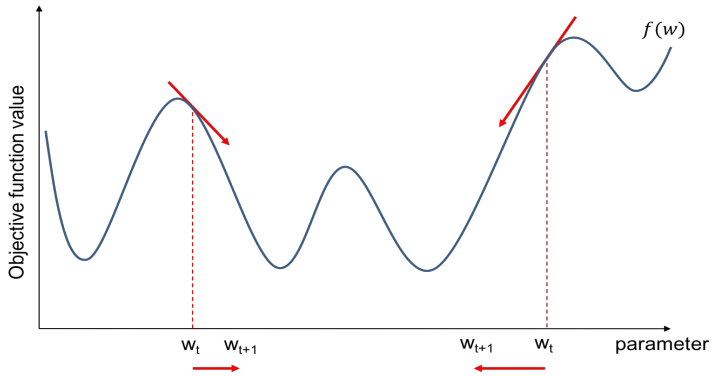

그림 2.2. 경사강하법의 적용 예: 현재시점 t 에서, 왼쪽의 w_t의 경우 기울기가 음수이므로 변수를 오른쪽으로 이동하고 오른쪽의 w_t의 경우 기울기가 양수이므로 변수를 왼쪽으로 이동한다.

경사하강법의 원리는 간단하고 이를 통해 대부분의 딥러닝 알고리즘들이 학습되고 있다. 그림 2.2 에서와 같이, 현재 변수의 값 w_t에서 목적함수 $f(w_t)$의 기울기가 음수인 경우 w_t를 오른쪽으로 이동하여 $f(w_{t+1})$의 값은 줄어든다. 반대로, 기울기가 양수인 경우 w_t를 왼쪽으로 이동하여 $f(w_{t+1})$의 값은 줄어든다. 대부분의 딥러닝 알고리즘의 경우, 변수의 개수는 수백만, 수억개로 구성될 수 있는데, 단순히 위 과정을 변수의 개수만큼, 즉 수백만번, 혹은 수억번, 반복하는 것으로 학습이 이루어진다.

2.2. 학습의 종류

데이터로부터 지능을 얻는, 즉, 학습하는 방법은 크게 감독학습, 무감독학습 및 강화학습으로 나눠진다. 이렇게 나눠지긴 하지만 목적함수를 설정하고 주어진 데이타에서 최적화 과정을 통해 변수를 찾아낸다는 면에서 모두 동일한 과정을 따른다.

감독학습은 입력과 출력데이타가 모두 주어진 경우 입력으로 출력을 맞추는 문제로 이해할 수 있다. 주로 음성인식, 얼굴인식 등의 패턴인식이나 회귀분석(regression)과 같은 경우에 사용된다. 인식 문제는 주어진 데이타를 보고 여러개의 클래스 중에 정답 클래스를 맞추는 문제이고 회귀분석은 주어진 데이타의 정답값을 맞추는 문제이다. 무감독학습은 입력데이타는 있지만 출력데이타는 없는 경우 입력데이타를 분석하는 문제이다. 주로 군집화(clustering)나 차원축소(dimension reduction)등에 사용되며, 데이타를 분석하고 이해하는데 도움이 된다. 강화학습은 입력은 주어지지만 정답에 해당하는 출력값은 주어지지 않고 보상값(reward) 만 주어지는 경우에 사용하는 학습 방법으로, 알파고나 로봇 등의 학습에 활용된다.

강화학습이 감독학습과 다른점은 예측값이 다음 시간의 입력값에 영향을 미친다는 점과 정답이 아닌 보상값이 주어진다는 점이다. 다음 시간의 입력값에 영향을 미친다는 뜻은 예를 들어 자율주행의 경우 손잡이를 오른쪽으로 돌릴 경우 보게될 영상 입력과 왼쪽으로 돌릴 경우 보게될 영상입력이 달라진다는 뜻으로, 시스템이 내리는 결정이 외부 환경에 영향을 미치고 결국 영향을

받은 환경에 의해 시스템이 보게될 다음 입력이 결정된다는 의미이다. 정답이 없이 보상이 주어진다는 것은 알파고에서 예를 들면 정확히 어디에 두어야 하는지 정답이 있는 경우는 감독학습이고, 정답은 모르지만 현재 선택한 곳에서 이길 수 있을 확률이 얼마인지를 예측해서 알려줄 때의 값을 보상이라고 한다.

3. 신경망의 역사와 학습 원리

3.1 역사

딥러닝은 심층망에서의 학습과 추론에 대한 연구이며, 심층망은 기존 신경망의 계층을 확장한 형태이므로, 딥러닝을 이해하려면 신경망의 발전을 이해할 필요가 있다. 최근 인공지능의 역사를 신경망의 역사로 해석하는 현상은 딥러닝이 인공지능에서 차지하는 비중을 잘 보여준다.

최초의 신경망은 1949년 헵(D. Hebb)에 의해 시작되었다고 여겨진다. 헵은 신경망을 학습시키려고 헤비안 학습(Hebbian learning)을 제안했는데, 그 원리는 같이 행동하는 뉴런들을 더 단단히 연결하는 것으로 요약된다 (“Fire together, wire together”). 단순하지만 아직도 많은 경우에 사용되는 학습 원리이다. 이후 1958년 로젠블랫(F. Rosenblatt)이 단층신경망인 퍼셉트론(Perceptrons)을 제안하고 알파벳 인식에 적용했다. 퍼셉트론을 지켜보면서 사람들은 인간 수준의 인공지능이 곧 가능할 것으로 믿었지만, 1969년 메사추세츠공대(MIT)의 민스키(M. Minsky) 교수가 퍼셉트론의 한계를 증명함으로써 신경망에 대한 사람들의 기대는 사라졌다 [7]. 이때 이미 신경망의 계층을 늘려 계산 능력을 키우면 어떨까 하는 질문들이 있었지만, 민스키는 여전히 한계를 극복할 수 없을 것이라고 단정했다.

이후 신경망은 1986년 럼멜하트(D. Rumelhart), 힌턴, 그리고 윌리암(R. Williams)이 발표한 역전파(backpropagation) 알고리즘의 등장으로 다시 주목을 받게 된다 [8]. 사실 역전파 알고리즘은 그전에 제안되었지만, 럼멜하트와 그 동료들의 연구로부터 주목받기 시작했고, 신경망은 또다시 낙관적인 전망으로 사람들의 관심을 끌었다. 이 역전파 알고리즘은 단층신경망 뿐만 아니라 한두개의 은닉층을 가지는 다단계퍼셉트론(multi-layered perceptron, MLP)도 학습가능하게 만들었고, 관련 연구들이 많이 이루어졌다. 하지만, 1995년 배프닉(V. Vapnik)과 코테스(C. Cortes)에 의해 서포트벡터머신(support vector machines, SVMs)이 소개되고, 신경망보다 더 쉽게 학습이 가능하면서도 좋은 성능을 보이자, 사람들은 다시 신경망을 버리고 SVM을 쓰기 시작했다. 신경망은 지금도 어느정도 그러하지만 학습이 쉽지않다는 문제가 있다.

이후 10여년간 신경망은 연구자들의 무관심을 받았지만, 힌턴은 여전히 신경망의 가능성을 믿고 연구를 계속했고, 2006년 사이언스 저널에 논문을 발표하면서 신경망의 가능성을 증명함으로써 패턴인식의 패러다임을 바꾸고, 음성인식, 영상인식등의 분야에서 성공적으로 적용함으로써 딥러닝이라는 이름으로 신경망 연구의 부활을 가져왔다. 딥러닝은 언어이해와 같은

분야에서도 성과를 내면서 인공지능의 수준을 한단계 성숙시키는 기술로 인정받고 있다. 몇몇 기계학습 전문가들은 신경망의 성공에 대해 지나친 열광을 우려하기도 하는데 [30], 이는 이미 몇차례 신경망에 대한 기대와 좌절을 경험했기 때문에 신중하자는 뜻으로 여겨진다.

3.2. 신경망의 구조와 학습

딥러닝은 인공신경망이라는 특수한 모델에 기반한다. 인공신경망은 뇌신경에서 발생하는 정보처리 과정을 매우 단순화한 계산 모델로써, 생물학적 신경망에서의 연산단위(혹은 뉴런)와 연결(혹은 시냅스)을 노드(node)와 그들 사이의 연결강도(weight)로 구현한다. 그림 3.1은 입력 x와 은닉계층 h 그리고 출력 y 가 있는 간단한 신경망 구조의 예이다. 이때 x와 h 그리고 y는 노드가 되고 이들을 연결하는 W^1 과 W^2 는 연결강도가 된다.

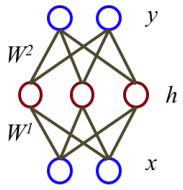

그림 3.1. 하나의 은닉계층을 가진 신경망의 예. 원들은 노드, 선들은 연결강도. 그림에서 x는 2차원 벡터.

신경망에서 입력값 $x = [x_1, x_2, \ldots x_d]$는 벡터로 주어지는데, 예를들어 크기가 10x20인 흑백 이미지가 주어질때 x 는 200차원의 벡터가 된다. 즉 d 가 200이 된다. x는 h를 거치면서 출력값 y로 변환된다. 이때 h와 y도 벡터가 되고 각각의 차원은 구조 설계시에 결정해준다. x 에서 h를 거쳐 y가 되는 과정은 다음과 같이 연결강도 행렬의 곱과 비선형 함수 σ(주로 sigmoid 함수)를 적용하는 것으로 이루어진다.

$$h = \sigma(W^1 x),$$

$$y = W^2 h.$$

위 식에서 W^k는 k번째 계층의 연결강도를 의미한다. 이러한 신경망은 입력 x가 주어질 때 출력 y를 계산하는 함수로 볼 수 있다. y를 구하는 식에서 비선형 함수를 사용하기도 하고 그렇지 않은 경우도 있는데, 이는 문제의 특성을 따라 결정한다. 여기서는 사용하지 않는 경우의 예를 보여준다.

앞서 기계학습에서 모델의 변수를 수정해가는 과정이 학습이라고 했었는데, 인공신경망에서 변수는 바로 연결강도이고, 신경망의 학습 원리는 기계학습의 경사강하법에 기반한 학습 원리와 동일하다. 예를 들어, 인공신경망에서의 감독학습은 일반적인 기계학습의 감독학습과 동일한 방식으로 작동하는데, 단지 모델과 변수가 다를 뿐이다. 신경망의 학습은 현재의 연결강도로 정의되는 신경망에 학습 데이타의 입력값을 대입했을

때의 출력값과 동일 입력에 대한 정답값 사이의 차이(오차)를 감소시키는 방향으로 경사강하법을 통해 연결강도를 조정해가는 과정이다.

출력값과 정답 사이의 차이를 표현하는 비용함수(목적함수)와 비용함수값을 감소시키려는 연결강도의 개선 방법에 따라 경사하강법의 다양한 학습 알고리즘이 있다. 비용함수 E는 주로 출력값 y와 목표값 t사이의 차이, 즉, $E=(y-t)^2$로 정의하고, 연결강도의 개선은 위에서 설명한 것처럼 현재 변수에서 함수의 기울기를 사용하여 이루어지는데, 단순한 경사하강법을 적용한 경우 구체적으로는 다음과 같이 정의된다.

$$w = w - \eta \frac{\partial E}{\partial w} \ .$$

여기서 η는 학습률(learning rate)을 의미하며 한번에 어느 정도를 학습할지 학습양을 결정한다. 신경망에 변수가 되는 연결강도는 여러 계층에서 행렬의 요소값으로 표현되어 매우 많지만, 요소값의 개수만큼(그림3.1의 경우에는 12번) 그림2.2에서와 같이 경사하강법을 적용한다. 오차에 대한 낮은 계층의 연결강도의 편미분은 연쇄법칙(chain rule)을 통해 계산되고 이때 연결강도를 수정하는 수식(update rule)은 출력값과 정답값의 차이에서부터 시작된 에러의 역전파(backpropagation)된 형태로 나타나게 된다. 즉, 위 계층에서의 에러값이 연쇄법칙을 따라서 아래 계층으로 전파되는 방식으로 편미분이 이루어진다. 이런 학습 과정은 매우 단순하고, 초기에 제안되었을 때에도 하나 혹은 두개의 계층을 포함하는 천층망에서는 잘 작동했다.

데이타를 통해 비용함수를 감소시키는 방향으로 연결강도를 여러번 수정하다보면 더이상 비용함수값이 줄어들지 않게 되는데, 이때 모델이 수렴했다고 하고, 학습을 멈춘다. 학습이 끝나면 연결강도를 더이상 수정하지 않는다. 고정된 연결강도를 갖는 인공신경망은 입력값에 대한 출력값을 계산하는 함수가 되는데, 이미지분류와 같은 경우, 이미지를 입력하면 분류된 결과(예를들면 '사자' 나 '자동차' 등)를 출력하게 된다.

주어진 인공신경망의 네트워크에 대해 깊이를 계층의 수로 정의할 수 있는데, 딥러닝은 쉽게 말해서 기존의 신경망에서 다루는 네트워크들 보다 더 큰 깊이(더 많은 계층)를 갖는 인공신경망이라고 할 수 있다.

4. 딥러닝의 학습 원리

4.1. 심층망의 어려움

패턴인식 등에서의 성능을 향상하려고 여러 계층을 쌓으려는 심층망이 최근까지 활발히 연구되지 않은 이유는, 위에서 언급한 것처럼 신경망은 학습이 어려운데 심층망은 특별히 더 어렵다는 것이다. 즉, 신경망을 학습하는데 사용되는 역전파 알고리즘이 3개 이상의 계층으로

쌓은 심층망에서는 에러의 역전파에 어려움을 겪는 것인데, 이는 사라지는 경사(vanishing gradient)라는 현상 때문이다. 에러 정보가 출력노드에서 입력노드 방향으로 전달되면서 점점 사라지는 것을 말하는데, 에러 정보가 낮은 계층까지 잘 전해지지 않으면서 낮은 계층의 연결강도는 학습 정도가 거의 이루어지지 않아 초기의 랜덤 값에서 크게 벗어나지 못하게 된다.

앞서, 인공신경망이 연결강도를 조정하여 다양한 함수를 표현할 수 있다는 점을 언급했다. 하지만, 사라지는 경사 현상으로 인해 상대적으로 낮은 층은 학습의 양이 극히 작아지므로, 결국 학습 과정에서 상위 몇 개 층의 연결강도만을 조정하게 된다. 이는 결국 낮은 계층의 연결강도는 초기의 랜덤 값에서 크게 변하지 않음으로써 전체적인 성능을 떨어뜨리게 된다. 하지만, 2006년 힌턴이 사전학습을 제시함으로써 심층망에서 학습이 가능한 방법을 보여줬고, 이후 다양한 방법들이 제안되고 있다.

4.2. 심층망의 학습 방법

심층망의 학습을 가능하게 하는 방법들 중에 일반적으로 사용될 수 있는 대표적인 방법들 몇가지를 살펴보자.

4.2.1. 사전학습(pre-training)

사전학습은 이름 그대로 심층망에 역전파 알고리즘을 적용하기 전에 각 계층별로 사전학습을 진행하는 것이다. 즉, 역전파 알고리즘을 임의의 값(random value)에서 시작하는 것이 아니라, 사전학습을 통해 심층망의 연결을 학습에 도움이 되는 값으로 미리 변형해 놓는 것을 의미한다. 입력값이 주어지면, 첫번째 계층을 무감독 학습으로 먼저 학습하고, 그 출력값을 두번째 계층의 입력으로 사용하여 두번째 계층을 학습한다. 이러한 과정을 모든 계층에 순서대로 진행한다. 즉, 전체 신경망을 층별로 분해해서 학습하는 것이다. 이후 역전파 알고리즘으로 전체 신경망을 학습하는데 이를 미세조정(fine-tuning)이라고 한다. 이름 그대로 미세조정을 통해서는 연결강도가 조금 조정된다.

이런 사전학습은 초기값을 최적해 근처로 옮겨 놓는다는 점에서 최적화(optimization) 문제의 좋은 초기해를 찾는 방법으로 해석할 수 있다. 뿐만 아니라, 무감독학습이 $p(x)$로 표현되는 데이타의 분포를 학습하고, 감독학습에 기반한 미세조정은 $p(y/x)$ 로 표현되는 분류성능을 최대화 하는데, 베이즈 룰(Bayes rule)에 따라, 좋은 사전 분포 $p(x)$ 는 분류 문제 $p(y/x)$에 대한 좋은 사전 지식이 된다. 사전학습의 또 다른 장점은 무감독학습이기 때문에 레이블 없는 빅데이타를 학습에 사용할 수 있다는 점이다. 감독학습에 필요한 레이블이 많지 않은 데이타들도 있고, 또 레이블을 만드는데 드는 비용이 매우 큰 경우 무감독 학습은 유용하다.

또한 사전학습은 변수들의 사전 분포(prior distribution)를 지정하는 것으로 이해될 수 있는데, 이는 곧 목적함수에 제약조건(regularizer)를 설정하는 것과 같다. 최근에는 사전학습 없이도 드랍아웃(dropout)이나 새로운 비선형함수(ReLU)와 같은 다양한 제약조건을 설정함으로

효과적으로 학습하는 추세이다.

4.2.2. 드랍아웃

드랍아웃은 학습하는 중에 노드들의 일부(주로 절반)를 임의로 끄는데, 매 학습 회수마다 임의의 선택을 새로 한다. 학습이 끝난 후 새로운 데이타에 대해서는 절반의 노드를 끄는 대신 모든 노드들의 출력값을 절반으로 나눈다. 이러한 방법은 기계학습의 배깅(bagging) 방법과 비슷한 효과를 만드는데, 안정성과 정확도를 향상시킨다 [11]. 그리고 중요한 것은 드랍아웃은 상호적응(coadaptation) 문제를 해소하는 것으로 이해된다. 두개의 노드가 한번 비슷한 연결강도를 가지게 되면, 그 두 노드는 비슷한 방식으로 업데이트 되면서 마치 하나의 노드처럼 작동하고, 이것은 컴퓨팅 파워와 메모리의 낭비로 이어진다. 드랍아웃이 임의로 노드들을 끌때 이러한 두개의 노드가 나눠지게되면 상호 적응 문제를 회피할 수 있게 된다.

4.2.3. 조기멈춤(Early stopping)

심층망과 같은 패턴인식 모델을 학습할 때는 보통 두 가지 목표를 동시에 달성하기를 원한다. 하나는 비용함수를 최소화하는 모델을 찾는 것이고, 다른 하나는 찾은 모델이 학습에 사용되지 않은 데이터에 대해서도 인식을 잘하기를 원하는 것, 즉 과적합을 피하는 것이다. 과적합 문제는 근본적으로 학습 데이터에 대해서만 비용함수를 최소화하기 때문에 발생한다. 즉 모델이 지나치게 학습 데이터에만 최적화되어 학습에서 보지 못한 새로운 데이터에 대해서는 오히려 큰 에러를 발생시키는 문제이다. 이를 해결하려고 주로 쓰는 방법은 조기멈춤(early stopping)이다. 이 방법은 학습 데이타 중 일부를 검증 데이타로 따로 떼어놓고, 남은 데이타로만 학습을 진행한다. 학습 중 검증 데이타로 성능을 검증해서 검증에러가 떨어지다가 올라가기 시작하면 학습을 멈춘다. 이 방법은 매우 간단해 보이지만, 잘 작동한다.

그외에도 최대값출력(Maxout)이나 선형정류기(linear rectifier. ReLU), 나머지연결(residual connection) 등의 모델 관련 기법들과, GPU(graphic processing unit)와 여러대의 서버들에 기반한 병렬연산(parallel computing), 심층망의 모델 크기와 계산양을 줄이려는 시도, 즉 천층망으로 심층망의 성능을 모방하는 모델 압축(model compression) 등 많은 기법들이 있다.

5. 딥러닝의 주요 개념들

본 장에서는 이해를 도우려고 딥러닝에서 주로 이야기되는 개념들에 대해 간략히 정리한다. 특별히, 딥러닝은 표현 학습(representation learning)의 일종으로써, 표현 학습의 중요한 개념들을 살펴본다.

5.1 계층화(hierarchy)와 추상화(abstraction)

딥러닝은 여러개의 계층을 쌓음으로써 표현력을 증가시키는데, 계층을 증가시킬수록 더 추상적인 표현을 찾게 된다 (그림 5.1 참고). 입력되는 데이타 (얼굴 인식의 경우 얼굴 이미지) 는 가공되지 않은 그대로 이고, 출력되는 데이타는 가장 추상적인 표현 (얼굴 인식의 경우 얼굴 ID) 일때, 계층이 증가될 수록 그 추상화의 단계는 세분화되고, 상위층으로 올라갈 수록 추상화의 정도가 점진적으로 높아진다. 계층을 올라갈 수록 비선형 함수에 의해 아래 계층의 표현을 조금더 출력값을 닮아 가는 방향으로 표현은 변화되기 때문이다.

이러한 추상적인 표현은 데이타에서 일어나는 작은 변화들에 강건한 특징(invariance to local variations)을 갖게 된다. 이러한 추상화 과정은 출력이 이산적인 군집화나 실수와 같은 회귀분석 모두에 동일한데, 입력값중에 출력값에 영향을 미치는 표현들만 찾아내는 과정으로 이해할 수 있다.

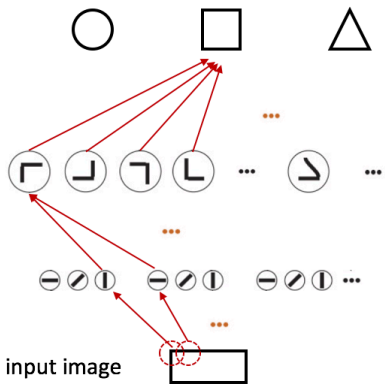

그림 5.1. 입력 값으로부터 추상적인 표현을 찾아가는 과정. 계층이 올라갈 수록 추상화의 정도는 높아진다.

5.2 재사용성(reusability, transfer learning)

계층화에의한 추가적인 장점은 재사용성 이다. 주어진 특정 임무에 필요한 정보를 추출하는 신경망을 학습했을 때, 이와 유사한 다른 임무가 주어질 경우 이미 학습된 신경망을 사용할 수 있다는 뜻이다. 이를 지식전달(knowledge transfer)이라고 하는데, 그림 5.2 에서 처럼 각 임무(task) 별로 다른 신경망이 구성되지만, 아래 계층은 공유 할 수 있다. 이러한 가능성은 위에서 설명한 추상화 과정에서 아래 계층은 매우 원초적인 특징들만을 표현하기 때문에 비슷한 임무인 경우 아래 계층에서는 달라질 부분이 거의 없기 때문이다.

이러한 특징은 데이타의 양이 많지 않은 문제를 해결해야할 때, 비슷하지만 데이타가 많은 문제로부터 학습된 신경망을 재사용할 수 있게 해줌으로써 더 많은 응용이 가능해진다.

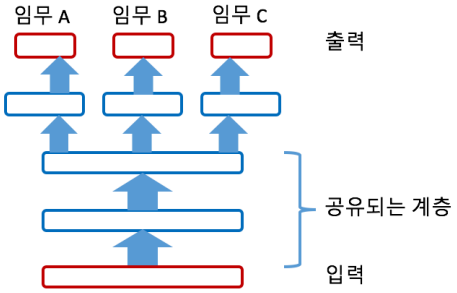

그림 5.2. 임무별로 각각의 신경망 전체를 학습하는 것이 아니라 일부를 공유/재사용 할 수 있다. 즉 임무A를 해결하려고 학습된 신경망의 아래 부분은 임무B의 모델을 학습할 때 재사용될 수 있다.

5.3 지역적(local) vs. 분산(distributed) 표현

데이타를 표현하는 방법이 여러가지가 있는데, k최근접이웃(k nearest neighbor, kNN) 등의 방법들은 데이타 각각을 기준으로 표현 (지역적 표현) 하고, 주요성분분석(principal component analysis, PCA) 같은 방법들은 데이타를 여러개의 구성요소로 나눠서 표현 (분산표현) 할 수 있다. 지역적 표현과 분산표현 방법중에 어느 것이 좋은 지는 경우에 따라 달라지지만, 데이타가 여러개의 구성요소들의 집합으로 이루어진 경우 분산표현이 효과적이다. 예를 들어, 얼굴 이미지는 눈, 코, 입 등으로 이루어지는데, 얼굴을 각 구성요소로 표현할 경우 분산표현을 사용하는 예가 된다. 이때, 지역적 표현은 특정 얼굴 이미지가 다른 이미지들 중에 어떤 몇몇 이미지들과 얼마나 닮았는지로 표현하게 된다. 여기서 지역적이라는 말은 오해되기 쉬운데, 데이타 전체 샘플들중에 지역적으로 가까운 샘플들만 사용된다는 뜻이고, 분산표현은 여러개의 변수들에 의해 정보가 분산되어 표현된다는 뜻이다. 분산표현은 더 작은 변수로 더 많은 구역을 표현할 수 있고, 전혀 새로운 데이타에 대한 표현도 가능하다.

5.4 희소표현(sparse representation)

희소표현은 분산표현 만큼이나 딥러닝 관련 논문에서 자주 등장하는 개념이지만, 지역적 표현과 혼동되기 쉬운 개념이다. 지역적 표현은 희소표현과 관련없다. 굳이 따지자면 지역적 표현은 굉장히 희소한 표현을 가지는 것이지만, 일반적으로 희소표현이라는 개념은 데이타 샘플들 중에 작은 개수의 샘플들에 의해서 표현되는 것을 의미하는 것이 아니라, 분산표현 중에 작은 개수의 변수에 의해서만 표현되는 것을 의미하므로 다른 이야기다. 예를 들면 그림 5.3 에서처럼 PCA 는 분산표현이지만 압축표현 이고, 비음수행렬분해(non-negative matrix factorization, NMF)의 경우는 분산표현 중에 희소표현이다. 즉 희소표현은 분산표현 방법들 중에서 압축표현의 반대되는 개념이다.

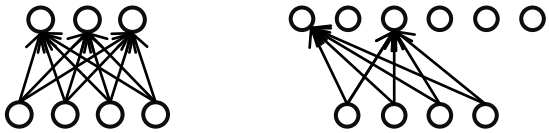

그림 5.3. 압축표현 (왼쪽) 과 희소표현 (오른쪽). 연결된 선이 없는 경우는 연결강도가 0 임을 의미.

희소표현은 신경망을 효율적으로 사용할 수 있게 한다. 신경망의 규모가 클 경우, 연결선들이 많아 데이타를 표현하는 데 있어서 낭비되는 경우가 있고 학습 데이타에 너무 치우치는 과적합 문제가 발생한다. 이때 희소표현을 제약으로 사용하게 되면 신경망을 보다 효과적으로 표현하려고 학습 데이타에 너무 치우치지 않게 된다. 주로 학습할 때 L1 norm을 희소표현 제약으로 사용하는 경우가 많은데, 합성곱신경망(convolutional neural networks, CNNs)에서처럼 처음부터 신경망의 연결선들을 희소하게 구성하는, 구조적인 접근법도 있다.

이러한 희소표현은 뇌신경망에서도 관찰되고 있는데, 뇌의 전체 뉴런들 중에 매우 작은 부분만이 동시에 활성화(spatial sparseness)되고 각 뉴런은 자주 활성화 되지 않는다(temporal sparseness).

5.5 다양체(manifold)

다양체학습(manifold learning)은 2000년부터 활발히 연구되기 시작된 주제이고 [34], 최근 그 연구가 줄어드는 경향이 있지만, 그 개념은 딥러닝에서도 여전히 중요하게 사용되고 있고, 실제 많은 논문들에서 다양체라는 개념을 사용해서 데이타 표현을 설명한다.

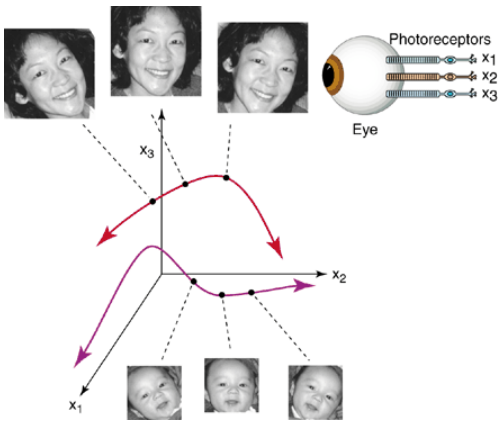

그림 5.4. 고차원의 이미지는 낮은차원의 다양체를 이룬다. 이러한 다양체는 동일한 이미지의 다양한 변화(회전, 위치이동 등)에 의해 만들어진다. 즉 데이타의 변화가 다양체를 만든다. [34] 에서 발췌.

데이타의 다양체에 대한 개념은 그림 5.4 에서 처럼 고차원의 데이타라도 훨씬 더 낮은 차원의 다양체에 집중되어 있고 이러한 낮은 차원의 다양체가 고차원의 공간에 포함되어 있다는 가정에서 시작된다. 이러한 가정은 이미지나 음성 등 대부분의 데이타에서 적절하다. 즉, 많은 경우, 데이타가 주어지면 데이타의 다양체를 생각할 수 있고, 이러한 다양체는 데이타의 변화를

효과적으로 표현하게 된다.

이러한 다양체를 찾는 연구를 다양체학습이라고 하는데, 딥러닝에서 다양체를 찾는 방법은 다소 다르다. 기존의 다양체학습은 변수를 두지 않고 데이타 간의 거리를 가장 잘 보존하는 낮은 차원의 다양체를 찾았지만, 딥러닝에서는 신경망의 연결선을 변수로 두고 에너지 혹은 그와 유사한 목적함수를 최소화하는 방법으로 다양체를 찾아낸다. 사실 딥러닝에서는 다양체를 찾으려고 하는 것은 아니지만, 결론적으로 좋은 다양체를 찾게 되는 셈이다.

5.6 풀기(disentanglement)

딥러닝을 가능하게했던 사전학습이 분류 문제에 있어서 도움이 되는 이유를 설명하는 다양한 방법이 있겠지만, 설득력있는 설명이 바로 풀기이다.

분류의 경우 주어진 데이타에대한 클래스를 찾아내는 건데, 각 클래스를 설명하는 요소들이 데이타 곳곳에 녹아있다고 볼 수 있다. 사전학습의 경우 섞여있는 요소들, 즉 클래스를 설명하는 요소들을 풀어내어 클래스를 찾기 쉽게 해주는 것이다. 사전학습에서는 어떤 요소가 분류문제에 도움이 되는지 알 수 없기 때문에 가능한 모든 요소들을 풀어내는 것이 중요하다. 다양체는 데이타의 변화 중 중요한 부분들만 표현하는 것이라면, 풀기는 모든 요소들을 추출하여 표현하는 것이 목표다. 하나의 임무만 주어진 경우라면 노이즈에 해당하는 부분들을 버리는 다양체가 도움이 되겠지만, 다음 임무에서는 어느 것이 노이즈 인지 알 수 없고, 따라서 모든 요소들을 다 추출하여 표현하는 것이 도움이 된다.

5.7 숨은 지식(dark knowledge)

숨은 지식은 가장 최근에 나온 개념이다. 심층망을 통해 성능이 향상되었지만, 학습된 심층망의 많은 부분이 주어진 임무에 큰 도움이 안된다는 가정에서 시작된다. 따라서 주어진 임무 해결에 좀 더 작은 신경망이 충분할 수 있다. 하지만, 작은 신경망을 처음부터 학습시키면 그만한 성능이 나오지 않는다는게 문제인데, 이때 숨은 지식을 활용하면 작은 신경망도 학습이 가능해진다. 그림 5.5 처럼, 심층망을 학습하고 나면, 입력에 대해서 나오는 출력값은 분류에 적합한 정보 뿐만아니라 더 많은 정보를 가지고 있는데, 이를 숨은 지식 이라고 한다. 입력에 대한 클래스 정보 뿐만 아니라, 그 분포자체가 더 많은 지식을 표현하고 있는데, 이를 활용할 수 있다는 것이다. 이 숨은 지식을 천층망을 학습하는 타겟 정보로 활용할 수도 있다. 이렇게 학습된 천층망은 심층망과 비슷한 수준의 성능을 가지게 되는데 결국 비슷한 수준의 성능을 만들면서도 모델의 크기를 줄인 셈이 된다. 이를 모델 압축(model compression) 이라고 한다.

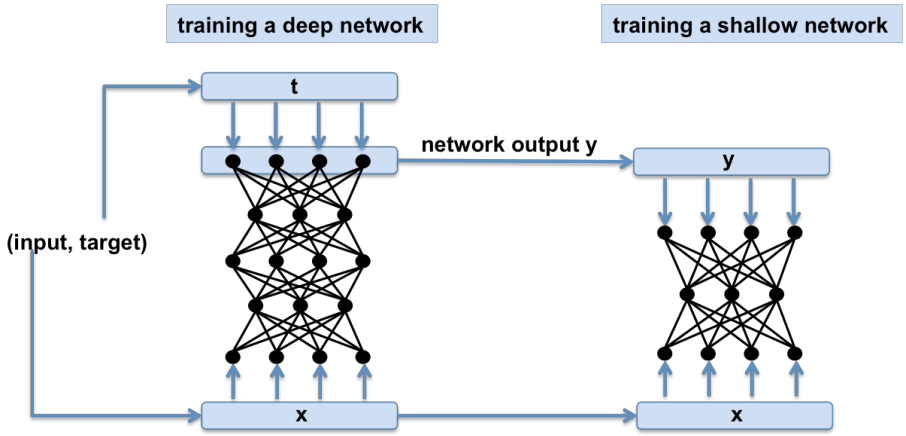

그림 5.5. 심층망을 학습한 뒤, 입력에 대해서 나오는 출력값은 분류에 필요한 정보 뿐만 아니라 더 많은 정보를 가지고 있는데, 이를 숨은 지식 이라고 한다.

5.8 딥러닝의 필요성

신경망이 하나의 은닉 계층만 가지고 있어도, 보편 근사기(universal function approximators)로 작동한다는 것은 잘 알려진 사실이다. 이 말은 적절한 인공신경망의 구조를 디자인하고 연결강도를 결정하면 어떠한 함수라도 근사적으로 표현할 수 있다는 것이다 [12]. 그렇다면 하나보다 더 많은 여러개의 계층을 쌓는 것이 어떤 이점이 있는지 설명이 필요하다. 주어진 데이타를 표현하거나 입력과 출력간의 관계를 충분히 표현하려면 그만큼 모델이 복잡해야 하는데, 천층망에서는 노드의 개수를 증가시는 것이 유일한 방법이다. 이러한 방법은 계층을 쌓아 올리는 것에 비해 효과적이지 못하다. 즉, 어떠한 신경망이 더 효과적이냐의 질문이 중요하다. 또한 심층신뢰신경망(deep belief networks, DBNs)의 경우 계층을 쌓을 수록 모델의 정확도가 좋아지는 것은 이론적으로도 증명된다.

천층망과 심층망의 차이를 설명하기 적절한 간단한 신경망을 예로 들어보자. 그림 5.6 에서 동일한 수의 연결을 가지는 두 가지 신경망 구조의 예를 보여준다. 두개의 신경망은 변수 (연결선의 수) 가 같은, 즉 비슷한 복잡도를 가지고 있다고 할 수 있지만, 심층망은 보다 높은 표현능력을 가진다고 할 수 있다. 이는 x_1에서 y_1으로 가는 길의 개수를 세는 것으로 설명할 수 있다. 천층망에서는 x_1에서 y_1으로 8개의 길이 있고 심층망에는 32개의 길이 있는데, 이는 심층망이 입력과 출력 사이를 더 많은 방법으로 모델링 할 수 있다는 것을 의미한다.

심층망에 대한 생물학적인 연관성도 심층망에 대한 기대를 높인다. SVMs 이나 MLPs 등의 천층망이 많은 패턴인식 문제에 성공적으로 적용되어 왔지만, 음성인식이나 영상인식에서는 여전히 인간 두뇌의 성능에 미치지 못했다. 따라서, 인간의 두뇌에 있는 생물학적 신경망의 원리들을 이해하는 것이 인공신경망의 성능을 향상하는데 도움이 될 것으로 기대해 왔다. 인간두뇌는 영상인식에 있어서 기본적으로 5~10개의 계층을 통해 연산을 수행한다 [15]. 즉,

어려운 문제에 대해 심층망의 성공적인 예가 이미 우리 두뇌에서 작동하고 있다는 뜻이다. 또한 메사츄세츠공대의 포지오(T. Poggio) 교수나 구글의 미래학자 커즈웨일(R. Kurzweil)은 계층적 모델은 인간수준의 지능을 확보하는데에 필수적인 원리라고 주장한다 [13, 14]. 이는 신경망을 더욱 깊이 만드는 것이 인공신경망을 패턴인식등의 여러가지 지능적인 태스크에 필수적이라는 뜻이다.

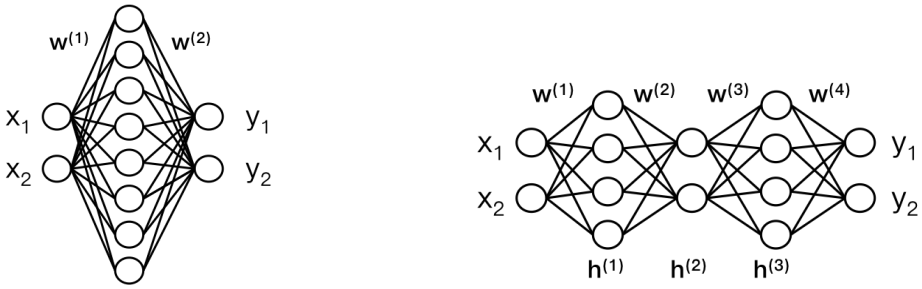

그림 5.6. 총 32개의 연결을 가지는 천층망 (왼쪽) 과 심층망 (오른쪽) 구조의 예. 심층망은 같은 수의 변수를 가지는 천층망에 비해 입력과 출력 사이의 더 복잡한 관계를 모델링 할 수 있다.

6. 주요 알고리즘

딥러닝에는 다양한 모델과 알고리즘들이 있다. 여기서는 그러한 알고리즘들을 세가지 그룹으로 나누고 (구성요소, 생성모델, 그리고 판별모델), 대표적인 알고리즘들 중심으로 간략히 설명한다.

6.1 구성요소

심층망을 구성하는 구성요소들은 여러가지가 있는데, 주로 제한볼츠만기계(restricted Boltzmann machines, RBMs)이나 오토인코더(auto-encoders, AEs)가 사용된다. 두 모델의 구조는 그림 6.1 과 같다.

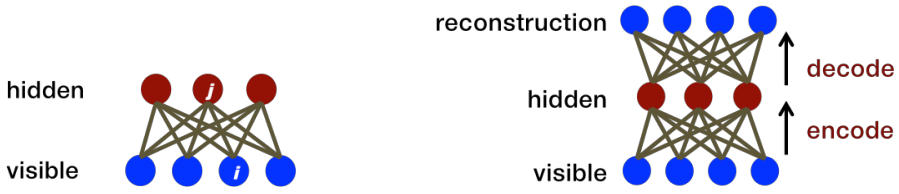

그림 6.1. RBM 구조 예 (왼쪽) 와 오토인코더 구조 예 (오른쪽).

RBM은 1986년에 소개된 생성모델이고 최근 딥러닝에 구성요소로 사용되면서 주목을 끌게 되었다 [18]. 그림 6.1 의 네트웍 구조에서 관측(visible) 노드들은 은닉(hidden) 노드들과만 연결되어있고, 관측 노드 계층과 은닉 노드 계층 사이의 관계에 기반한 확률모델을 아래와 같이

정의한다. 그리고, 학습은 확률을 최대화 하는 방향으로 이루어지는데, 경사하강법(gradient descent method)과 같은 방식으로 유도하지만 정확한 식이 아닌 근사식에 기반한 CD (contrastive divergence)라는 방법을 사용한다 [6].

$$p(v,h) = \frac{1}{z} e^{-E(v,h)},$$

$$E(v,h) = -v^T W h.$$

오토인코더는 그림 6.1 (오른쪽) 에서처럼 간단한 신경망인데, 입력과 출력은 동일하다. 즉, 인코딩후 디코딩 했을때 원래 입력과 같아져야한다는 것으로, 인코딩으로 정보의 손실이 최소화되기를 기대한다. 이때, 인코딩 행렬 W 에 대해 디코딩은 W의 전치(transpose)로 표현할 수 있어서 동일한 변수를 사용할 수 있다. 학습은 복구에러의 역전파에 기반한다. 최근에는 제한된(restricted) 오토인코더가 제안되어 다양한 형태로 사용되고 있다 [17].

6.2 판별모델

신경망의 부흥을 이끈 것은 여러가지 패턴인식에서 기존의 최고 기록들을 경신해 온 판별모델이라고 할 수 있다. 힌턴의 초기 모델은 RBM으로 초기화한뒤 역전파로 미세조정하는 모델이였지만, 현재 영상인식에서 가장 많이 사용되는 모델은 합성곱신경망이고 [2, 19], 음성인식등의 시계열 데이타인식에는 순환신경망(recurrent neural networks, RNNs)이 가장 많이 사용된다. 합성곱신경망과 순환신경망은 1980년대부터 사용되던 모델로써, 사전학습과 같은 기법들을 사용하진 않고, 바로 감독학습으로 학습할 수 있다.

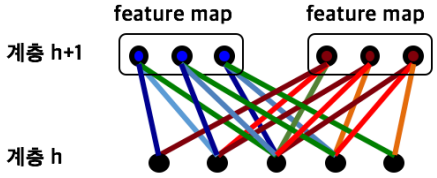

그림 6.2. 각 특징맵(feature map)은 아래 계층과 연결되지만 지역적인 특징을 가진다. 그림에서 계층 h+1 의 첫번째 노드는 아래 계층의 3개 노드와만 연결된다. 또한 이 연결선은 계층 h+1 의 두번째 노드와 공유된다. 같은 색의 연결 선은 같은 변수를 사용하는 즉 공유됨을 의미한다.

합성곱신경망의 중요한 특징중에 하나는 뇌신경과학적인 발견들에 기초한 모델이라는 점인데, 특히 후벨과 비젤(Hubel-Wiesel)의 단수-복잡 세포라던가, 지역적 감각수용장(local receptive fields), 뽑기(pooling) 같은 개념들은 뇌과학으로부터 왔다. 합성곱신경망은 이러한 개념들에 기초하여 지역적 감각수용장을 전체 이미지에 합성곱(convolution)하여 신경망의 연결강도를 공유하는 방법으로 변수의 수를 대폭 축소한다 (그림 6.2 참고). 이를 활용하여 후쿠시마(Fukushima)가 1980년대에 네오코그니트론(Neocognitron)을 발표하고 [9], 1989년 르쿤(Y. LeCun)이 네오코그니트론에 역전파 알고리즘을 결합하여 합성곱신경망을 완성했다.

합성곱신경망이 사전학습 없이도 학습 가능한 이유는 이러한 지역적 연결과 공유된 연결이 역전파되는 에러정보를 사라지지 않게 하고 에러의 분산을 줄이기 때문이다.

합성곱신경망이 영상인식에 최적화된 구조를 가지고 있는 반면, 시계열 데이타에 대해서는 순환신경망이 적절한 구조를 가지고 있다. 순환신경망은 그림 6.3 에서와 같이 일반 신경망의 각 계층에서 상위 계층으로 연결 뿐만아니라 자기 자신 계층에도 연결을 만드는데, 이러한 돌아오는 연결은 마치 메모리와 같은 역할을 함으로써 데이타의 시간적인 변화를 모델링할 수 있게 만든다. 반면, 이러한 돌아오는 연결로 학습은 더욱 어려워진다 [20]. 최근에는 긴 단기기억 모델(Long short-term memory, LSTM) 등을 사용하면서 이러한 학습 문제들이 해소되었고, 이를 사용한 순환신경망이 필기체 인식이나 음성인식 분야에서 우수한 성능을 내고 있다 [10].

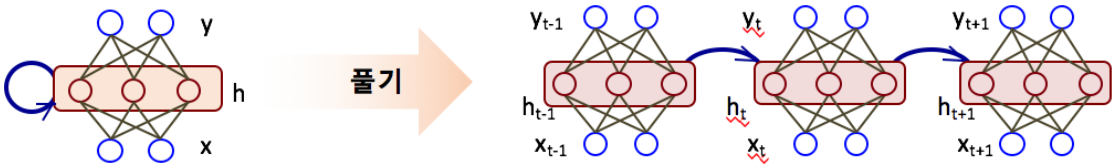

그림 6.3. RNN은 돌아오는 연결로 메모리의 기능을 추가했다. 학습은 돌아오는 연결을 '풀기' 의 과정을 거친 뒤 일반 신경망의 학습과 같은 방식으로 수행된다.

6.3 생성모델

딥러닝 연구 성과의 상당부분은 감독학습에 기반한 판별모델로부터 나온 결과들이지만, 무감독학습의 생성모델도 딥러닝 연구의 큰 축을 이루고 있다. 뿐만아니라, 생성모델은 인공지능에 있어서도 굉장히 중요한데, 모델이 데이타를 생성해낼 수 있다는 말은 그 모델이 그 데이타를 잘 이해했다고 판단할 수 있는 근거가 되기 때문이다. 초기에는 심층신뢰신경망 모델이 주요 연구 대상이였지만 최근에는 순환신경망, 생성대립신경망(generative adversarial networks, GANs)이나 변이오토인코더(variational auto encoders, VAEs) 등이 많이 연구되고 있다 [16, 48, 49]. 여기서는 심층망중 가장 기본적인 생성모델인 심층신뢰신경망(DBN) 모델을 간략히 살펴보고, GAN의 개념만 간략히 살펴본다.

DBN은 신경망과 같은 네트워크 구조인데 최상위 층은 RBM 형태의 무향그래프(undirected graph), 그 아래 계층들은 모두 위에서 아래로 내려오는 유향그래프(directed graph)로 구성된다. DBN의 학습은 먼저, 사전학습된 RBM들을 쌓은 후, 제일 위층은 RBM 그대로 두고 나머지는 유향그래프로 두고 업다운 알고리즘을 사용하여 미세조정 하는 방법을 사용한다 [6]. 학습 후 DBN은 학습에 사용된 데이타와 같은 종류의 데이타를 생성해낼 수 있다. 제일 위층에서 노이즈를 생성한 다음, RBM내에서 반복적인 샘플링을 통해 개념이 생성되면 아래로 내려오는 방법, 혹은 제일 아래에 실제 데이타를 넣고 (혹은 노이즈를 넣고) 제일 위층까지 올라간 다음 반복샘플링을 한 뒤 다시 내려오는 방법이 있다. 최상위 계층은 데이타의 개념을 표현한다.

GAN을 설명하는 쉬운 예로 위조지폐조작범과 위조지폐감별사간의 싸움을 들 수 있다. 조작범이 위조지폐를 “생성”하고 감별사는 위조지폐를 “판별”해내는 게임에서 조작범이 감별사를 속이려고 위조지폐를 더 정교하게 만들면 감별사도 함께 더 정확히 감별해내려고 노력한다. 이런 일련의 반복적인 작업이 진행되면 조작범의 위조지폐는 실제지폐와 구분하기 힘들만큼 유사할 것으로 기대된다. 즉, 그림 6.5 와 같이 생성모델과 판별모델이 서로 최적화의 과정을 반복하다보면 평형상태에 도달하게 되고 이때의 생성모델은 데이타를 생성할 수 있게 된다.

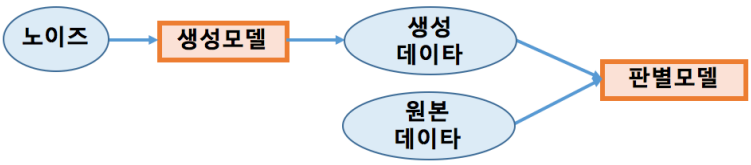

그림 6.5. 생성모델이 임의의 노이즈로부터 데이타를 생성하여 판별모델을 속이도록 학습하고, 판별모델은 원본데이타로부터 생성데이타를 구분해내도록 학습한다. 학습후 최종 결과물은 생성모델이다.

7. 응용 사례

딥러닝을 이용한 사례는 매우 많이 있지만, 여기서는 몇가지 예만 살펴본다. 가장 대표적인 예는 2012년 ImageNet 데이타 (http://www.image-net.org/) 에서 이미지 인식이였는데, 이미지가 주어지면 1000개의 클래스 중에 이미지에 정답 클래스를 맞추는 문제이다. 기존의 최고성능은 21.9% 였지만 합성곱신경망을 적용하여 그 성능을 32.6% 로 대폭 향상시켰다 [2, 21]. 기존의 수십년동안 진행된 컴퓨터 비전 기술들의 성능을 딥러닝을 적용함으로 획기적으로 뛰어넘은 것이다.

객체 인식과 비슷하지만 좀 더 복잡한 문제인 객체 검출의 경우도 합성곱신경망의 적용으로 성능이 개선되었다. 객체 검출은 이미지의 어느부분에 무슨 객체가 있는지를 알아맞추는 문제로써 객체 인식을 포함하는 문제이다. 최근엔 객체검출의 정확도가 개선됨과 동시에 1초에 90장을 처리할 수 있을 정도로 속도도 많이 개선되었다 [35] (그림 7.1 참고)

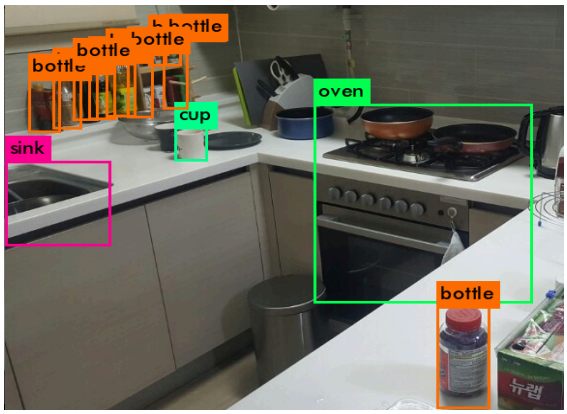

그림 7.1. 객체검출의 예. 한장의 이미지에 여러 객체를 동시에 실시간 검출

음성인식에 관하여는 2000년 이후 별다른 성능개선이 없다가 2010년 전후로 딥러닝에 의해 대폭적인 개선이 있었다. 하지만, 이미지 인식에서 합성곱신경망은 이미지로부터 클래스 정보까지 처리를 하지만 음성인식에서는 아직 전체 시스템에서 일부 (음향모델) 만을 딥러닝이 담당하고 있다 [31]. 초기에는 심층신뢰신경망 기반의 방식이 주로 사용되다가 최근에는 LSTM 기반 순환신경망 방식이 주목받고 있고, 합성곱신경망을 음성인식에 적용하려는 연구도 계속되고 있다.

다른 기계학습 알고리즘들과 달리, 심층망에 기반한 딥러닝은 이미지나 음성 인식 등의 패턴인식 뿐만 아니라 다양한 문제들에 잘 적용될 수 있는데, 언어이해(language understanding), 다양한 종류의 데이타 처리(multimodal learning), 지식 전달(knowledge transfer or transfer learning), 데이타 생성(data generation) 등이 그러한 예가 된다 [22, 23, 24, 25]. 그중에 대표적인 응용분야가 기계번역인데, 최근 딥러닝 기반의 기계번역을 신경기계번역(neural machine translation)이라고 한다 [36, 37].

신경기계번역은 입력 언어의 문장을 인코딩하고, 인코딩 결과를 출력 언어의 문장으로 디코딩하는 것으로 번역을 수행한다. 이때 인코딩과 디코딩은 순환신경망을 주로 사용하지만, 최근엔 순환신경망이 아닌 신경망을 사용하는 결과들도 나오고 있다 [38, 39]. 번역에서 사용한 기법은 이미지 캡션 생성에도 비슷한 방식으로 적용된다. 캡션 문장의 매 단어를 생성할 때마다 이미지의 특정 영역을 보면서 단어를 하나씩 순서대로 생성한다 [40].

그외에도 예술적인 분야로 여겨지는 응용사례들도 있다. 문학상에 근접한 수준의 소설을 쓰기도 하고 [42], 영화 대본을 쓰기도 한다 [43]. 또한 사진을 주면 유명한 화가들의 스타일로 변경해주기도 한다 [41] (그림 7.2 참고).

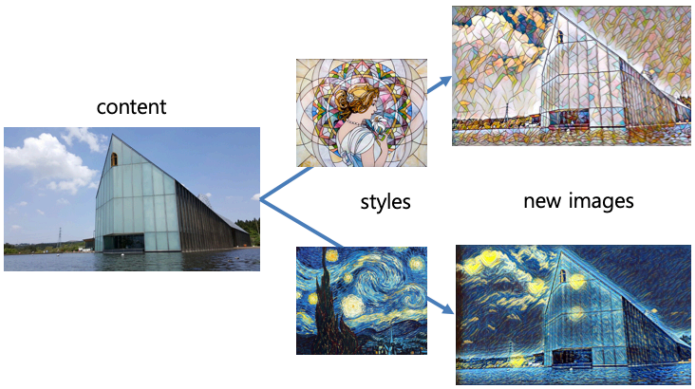

그림 7.2. 주어진 이미지에 대해 특정 그림 스타일로 변경할 수 있다.

마지막으로 자율주행에서도 활발한 연구가 수행되고 있다. 현재의 상용 모델들은 보행자 인식,

차선인식, 신호등 인식 등의 여러 인식 모델들이 합성곱신경망 기반으로 수행되고 결과를 해석하는 인공지능 엔진이 차량을 제어한다 [44]. 하지만 자율주행 알고리즘에서도 딥러닝의 영역이 점차 확대되고 있고 강화학습이 함께 사용되어 정확도를 향상시키고 있다 [45, 44].

8. 결 론

지난 10여년간 딥러닝은 컴퓨팅 파워와 빅데이타에 기반하여 많은 개선과 새로운 시도들을 경험해왔고, 지금도 다양한 연구들이 진행되고 있는데, 이러한 연구들 중에 주요한 방향은 크게 몇가지 형태로 요약될 수 있다. 하나는 대규모 모델을 빠르고 효과적으로 학습하는 방법 혹은 모델의 크기를 줄이는 방법 등에 관한 최적화 연구이고 [46], 또 하나는 다양한 사례들에 맞게 신경망 구조를 변형하고 적용해서 성능을 개선하는 연구이다 [29, 17]. 여러개의 GPU를 활용하여 속도를 개선하는 방법이나, 학습 알고리즘 자체를 개선하는 연구들이 활발히 진행되고 있고, 번역이나 이미지로부터 자막을 자동생성 하는 것과 같은 기존에는 불가능해 보였던 영역들에서 딥러닝이 기존 방식들의 성능들을 넘어서기 시작했다 [37, 40]. 영화대본 작성이나 소설 쓰기등과 같이 예전에 미쳐 생각하지도 못했던 분야까지 시도되고 있다.

딥러닝은 패턴인식 분야 뿐만아니라 더 넓게는 인공지능 분야를 혁신하고 있다. 하지만, 단순히 계층을 깊이 쌓는 것 만으로는 마르(D. Marr)에 의해 주장된 수동적 정보처리의 한계를 뛰어넘지는 못한다. 브룩스(R. Brooks)의 주장처럼 인간수준의 인공지능을 확보하는데에 환경과 상호 작용하는 로봇이 필요하다 [26, 27]. 결국, 인공지능 분야에서 더 많은 발전을 이루려면 강화학습은 반드시 필요한 기술이고, 최근 강화학습과 심층망을 결합하는 시도들은 중요한 연구 방향이다.

원래 신경망이 뇌신경망으로부터 영감을 얻어 시작되었지만, 뇌신경망과 딥러닝의 신경망은 여러면에서 다르다. 최근 주목받는 차이는 신경-상징 결합 문제(neural symbolic integration problem)로 설명된다 [47]. 뇌는 신경망으로 만들어져 있지만 상징에 기반한 추론이 가능하다. 예를 들어, '사람은 죽는다', '소크라테스는 사람이다' 에 대해 '소크라테스는 죽는다' 라는 추론이 가능한데, 이러한 추론이 뇌속의 신경망에서 이루어지고 있다. 하지만 현재의 딥러닝에서는 상징에 기반한 추론이 아직은 불가능해보인다. 인간 수준의 인공지능, 즉 일반 인공지능(general AI)이 가능해지려면 뇌과학에 대한 지식이 축적되고 딥러닝의 기술이 더욱 발전할 필요가 있다.

마지막으로, 인간수준의 일반 인공지능이 가능해지면 인류를 위협할 것이라며 불안해하는 사람들이 많이 있다. 하지만, 현재 딥러닝을 연구하는 사람들의 대부분은 인공지능이 인류를 공격하는 것보다는 인공지능의 결함이나 인공지능의 오남용이 현재로서는 더욱 위협적이라고 우려하고 있다. 이러한 시점에서 인공지능의 핵심기술인 딥러닝의 배경과 개념들을 살펴보고 이해하는 것은 중요하다.

참고문헌

[1] G. Hinton, R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, Vol. 313, No. 5786, pp. 504–507, Jul. 2006.

[2] A. Krizhevsky, I. Sutskever, G. Hinton, "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing (NIPS), Lake Taho, NV, 2012.

[3] J. Markoff, "How Many Computers to Identify a Cat? 16,000," *New York Times*. June 25, 2012.

[4] J. Markoff, "Scientists See Promise in Deep-Learning Programs," *New York Times*. November 24, 2012.

[5] G. Marcus, "Is 'Deep Learning' a Revolution in Artificial Intelligence?" *The New Yorker*, November 25, 2012.

[6] G. Hinton, S. Osindero, Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation* Vol.18, pp. 1527-1554, 2006.

[7] M. Minsky, S. Papert, *Perceptrons,* Cambridge, MA: MIT Press, 1969.

[8] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning internal representations by error propagation" in *Parallel Distributed Processing*, MIT Press, 1986, pp. 318–362.

[9] K. Fukushima, "Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, Vol. 36, No. 4, pp.193–202, 1980.

[10] A. Graves, J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, Vol. 18, No. 5-6, pp.602–610, 2005.

[11] P. Baldi, P. J. Sadowski, "Understanding dropout," Advances in Neural Information Processing Systems (NIPS), 2013, pp.2814-2822.

[12] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford: Oxford University Press, 1995.

[13] M. Riesenhuber, T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscie*nce, Vol. 2, No. 11, pp.1019–1025, 1999.

[14] R. Kurzweil, *How to Create a Mind: The Secret of Human Thought Revealed*. Penguin Books, 2012.

[15] S. J. Thorpe, M. Fabre-Thorpe, "Seeking Categories in the Brain," *Science*. Vol. 291, No. 5502, pp.260-262, Jan. 2001.

[16] A. Graves, A. Mohamed, G. Hinton. "Speech Recognition with Deep Recurrent Neural Networks," International Conf. on Acoustics, Speech, and Signal Processing (ICASSP), 2013, Vancouver, Canada.

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

[18] P. Smolensky, "Information Processing in Dynamical Systems: Foundations of Harmony Theory," in *Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 1*, MIT Press, Cambridge, MA, 1986, pp. 194-281.

[19] D. C. Ciresan, U. Meier, J. Masci, J. Schmidhuber, "A committee of neural networks for traffic sign classification," In Proc. of International Joint Conference on Neural Networks (IJCNN), 2011, pp.1918-1921.

[20] Y. Bengio, P. Simard, P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, Vol. 5, No. 2, pp.157-166, 1994.

[21] Y. Jia, "Caffe: An Open Source Convolutional Architecture for Fast Feature Embedding," 2013, http://caffe.berkeleyvision.org/.

[22] T. Mikolov, W.-T. Yih, G. Zweig, "Linguistic Regularities in Continuous Space Word Representations," In Proc. of NAACL HLT, 2013.

[23] R. Socher, M. Ganjoo, H. Sridhar, O. Bastani, C. D. Manning, A. Y. Ng, "Zero-shot learning through cross-modal transfer," In Proc. of International Conference on Learning Representations (ICLR), Scottsdale, AZ, 2013.

[24] Y. Bengio, É. Thibodeau-Laufer, G. Alain, J. Yosinski, "Deep Generative Stochastic Networks Trainable by Backprop," In Proc. of International Conference on Machine Learning (ICML), 2014.

[25] A. Graves. "Generating Sequences With Recurrent Neural Networks," 2014.

[26] D. Marr, "Vision: A Computational Investigation into Human Representation and Processing of Visual Information," Freeman, San Francisco, 1982.

[27] R. A. Brooks, "Elephants Don't Play Chess," *Robotics and Autonomous Systems,* Vol. 6, pp.3-15, 1990.

[28] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," Technical Report IDSIA-03-14, 2014.

[29] C. Farabet, B. Martini, B. Corda, P. Akselrod, E. Culurciello, Y. LeCun, "NeuFlow: A Runtime Reconfigurable Dataflow Processor for Vision", in Proc. of the Fifth IEEE Workshop on Embedded Computer Vision (ECV), Colorado Springs, 2011.

[30] http://spectrum.ieee.org/robotics/artificial-intelligence/machinelearning-maestro-michael-jordan-on-the-delusions-of-big-data-and-other-huge-engineering-efforts

[31] L. Deng, "Three classes of deep learning architectures and their applications: a tutorial survey," *APSIPA Transactions on Signal and Information Processing*, 2012.

[32] A. Petroff, "Elon Musk says Mark Zuckerberg's understanding of AI is 'limited'," CNN. July 25, 2017.

[33] T. Mitchell, "Machine Learning," McGraw-Hill Education, 1997

[34] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," Science, vol. 290, pp. 2319-2323, 2000.

[35] J. Redmon, A. Farhadi, "YOLO9000: Better, Faster, Stronger," arXiv:1612.08242, 2016.

[36] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," Advances in Neural Information Processing Systems (NIPS) 2014.

[37] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," Proc. Int'l Conf. on Learning Representations (ICLR), 2015.

[38] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, and K. Kavukcuoglu, "Neural Machine Translation in Linear Time," arXiv:1610.10099, 2016.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, and I. Polosukhin,"Attention Is All You Need," arXiv:1706.03762, 2017.

[40] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zimel, and Y. Bengio, "Show, Attend and Tell : Neural Image Caption Generation with Visual Attention," arXiv:1502.03044, 2015.

[41] L. A. Gatys, A. S. Ecker, and M. Bethge. "A neural algorithm of artistic style." arXiv:1508.06576 2015.

[42] https://www.digitaltrends.com/cool-tech/japanese-ai-writes-novel-passes-first-round-nationanl-literary-prize/

[43] https://www.youtube.com/watch?v=LY7x2Ihqjmc

[44] C. Chen, A. Seff, A. L. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," In ICCV, 2015.

[45] M. Bojarski et al., "End to End Learning for Self-Driving Cars," arXiv:1604.07316, 2016.

[46] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding", arXiv: 1510.00149, 2016

[47] H. Jaeger, "Deep neural reasoning". Nature, 538, 467-468, 2016

[48] I. Goodfellow, J. Pouget-Abadie, and M. Mirza, “Generative Adversarial Networks,” arXiv:1406.2661, 2014.

[49] D. P. Kingma, and M. Welling, “Auto-Encoding Variational Bayes,” arXiv: 1312.6114v10, 2014.