

3 장: 딥러닝 모델과 모델복잡도 이론

3.1 딥러닝 개념

3.2 딥러닝의 혁신점

3.3 딥러닝 아키텍처

3.4 모델복잡도 이론과 정규화

3.5 딥러닝 모델의 비교

3.1 딥러닝 개념

30 년 전에는 인공지능의 기초 연구 분야에 속하던 머신러닝이 최근 구글, 애플, 삼성 등 글로벌 기업들이 앞다투어 확보하려는 핵심 산업 기술로 발전하고 있다. 머신러닝 연구의 기원은 1959 년까지 거슬러 올라간다. 당시 IBM 연구소의 Arthur Samuel 은 체커 게임에서 기계가 경험을 통해서 스스로 성능을 향상하는 기계학습의 개념을 사용하였다[1]. 그리고 1986 년에 다층신경망(Multi-Layer Perceptron, MLP) 학습 알고리즘이 개발되면서 실제적인 머신러닝 연구가 시작되었으며, 당시까지 주류였던 기호논리 기반의 인공지능 기술의 한계를 극복하는데 기여하였다. 1990 년대부터는 인터넷 기반 산업이 발전함에 따라 정보검색, 데이터마이닝, 전자상거래, 추천 서비스 등을 위해 결정트리(Decision Trees), 베이즈안망(Bayesian Networks), 지지벡터머신(Support Vector Machines, SVM)과 같은 머신러닝 알고리즘들이 활용되기 시작하였다. 2000 년대 후반기에 들어서 머신러닝은 Apple Siri, IBM Watson, Google 무인자동차 등에 활용되면서 인공지능 산업을 고도화하는데 크게 기여하였다. 특히 최근에는 딥러닝(deep learning) 기술이 음성인식, 물체인식, 비디오게임 등에서 인간의 능력을 능가하는 성능을 보이면서 세간의 주목을 받고 있다[2,3]. 이는 그동안 축적된 빅데이터를 기반으로 고성능 컴퓨팅 능력을 이용하여 복잡한 구조의 딥러닝 모델을 학습하는 것이 가능해졌기 때문이다. 본 고에서는 딥러닝이 연구뿐 아니라 실제 산업 현장에서 사용되는 이유와 그 특징을 살펴보고, 딥러닝의 일종인 딥하이퍼넷(Deep Hypernetwork, DHN) 모델[4]의 구조와 학습 방법 및 응용 사례를 소개한다. DHN 은 기존의 Convolutional Neural Network (CNN)이나 Deep Belief Network (DBN)과 달리 스트림 형태의 순차적으로 관측되는 데이터로부터 실시간에 온라인 점진적 학습을 통해서 고차적 관계 구조의 지식을 자동으로 습득하는 평생학습 방법으로 개발되었다.

딥러닝은 신경망 기반의 복잡도가 큰 머신러닝 모델이다. 기존의 신경망 모델이 한 개의 은닉층을 사용한 비교적 단순한 모델인 것에 비해서 딥러닝 모델은 아주 많은 수의 은닉층을 사용한다. 인간 뇌의 시각피질에서도 V1, V2, V4 등 점차적으로 복잡한 정보를 추출하는 일련의 신경층들이 발견되며 딥러닝은 이러한 구조를 모사한 머신러닝 모델이다. 예를 들어, 하위층에서는 비교적 단순한 정보처리(예, 라인 탐지)를 수행하고, 상위층으로 갈수록 점차 복잡한 정보(예, 예지 검출과 물체 인식)를 추출하는 구조를 사용하는 것으로 알려져 있다. 최근 구글 연구팀은 백만 장의 이미지로 구성된 ImageNet 데이터베이스로부터 1000 가지 종류의 물체들을 인식하기 위해 22 층의 신경망 층을 가지는 딥러닝 모델을 개발하여 사용하였다[5].

딥러닝의 복잡한 학습 구조는 기존의 두 층짜리 SVM 이나 세 층짜리 MLP 신경망 구조와는 대조적이다. 학습 이론적으로 볼 때 딥러닝과 같은 복잡한 모델을 사용할 경우 과다학습(overfitting) 현상으로 인해 성능이 향상되지 않고 오히려 저하되는 것이 상식인데, 딥러닝은 이러한 기존의 상식을 위배하는 기이한 현상을 보인다. 모델 구조가 극단적으로 복잡해짐에도 불구하고 딥러닝의 성능이 좋은 이유는 무엇일까? 한 가지 근거는 최근 들어 스마트폰의 보급과 Youtube, Google Image 등을 통해서 데이터가 축적되어 가용한 학습 데이터가 무한히 많아진 것에서 찾을 수 있다. 즉 아주 많은 수의 학습 데이터를 사용할 경우 모델의 복잡도가 커져도 과다학습이 일어나지 않을 수 있다는 것을 실험적으로 발견한 것이다. 또한 학습 데이터가 많아지고 모델의 복잡도가 커지면 학습에 소요되는 시간이 비례하여 증가하는데도 불구하고 학습이 가능한 것은 컴퓨팅 파워의 비약적인 향상이 있었기

때문이다. 물론 가장 큰 이유는, 다음절에서 살펴보겠지만 최근 들어 딥러닝 구조를 학습하는데 필요한 여러 가지 테크닉들이 개발되었기 때문이다[6,7].

딥러닝이 산업 현장에서 선호되는 데는 몇 가지 이유가 있다. 일단은 어려운 문제를 잘 해결한다는 것이다. 예를 들어서, 물체인식과 음성인식 등 전통적인 패턴인식의 문제에서 딥러닝 기술은 신기록을 세웠다. 다른 이유는 현장의 문제를 과거보다 더욱 자동화된 방법으로 쉽게 풀 수 있다는 것이다. 데이터와 컴퓨팅 파워가 충분하다면 딥러닝이 사람이 코딩하는 것보다 더 좋은 성능을 낸다는 것이다. 특히 딥러닝은 인터넷과 웹에 널려 있는 무표지 데이터(unlabeled data)를 잘 활용할 수 있는 좋은 방법이다. 많은 딥러닝 방법들이 무감독 학습을 사용하여 정보를 자동으로 추출하는 기능을 포함하기 때문에 감독학습 문제를 풀더라도 이에 무표지 데이터를 추가로 활용함으로써 성능을 향상시킬 수 있다.

3.2 딥러닝의 혁신점

딥러닝의 핵심 아이디어는 기존에는 복잡한 문제를 풀기 위해서 특징 추출과 패턴 분류의 두 단계로 분리하여 문제를 해결하던 방식을 하나의 단계로 통합하여 해결하는 자동화로 볼 수 있다(그림 1). 기존에는 먼저 데이터 전처리 및 가공을 통해서 문제 해결에 적합한 특징들을 추출한 다음, 이를 학습 데이터로 하여 패턴 분류기를 훈련시키는 두 개의 단계로 문제를 해결하였다. 딥러닝 구조는 특징 추출을 위한 전처리 단계를 전체 학습 프로세스에 포함시킴으로써 가공되지 않은 원래 데이터를 직접 학습하도록 하는 통합된 문제해결 방식을 취한다. 딥러닝 구조는 특히 영상 데이터와 같이 차원수가 아주 크고 복잡한 데이터의 경우에 전처리 과정을 통해서 손실될 수도 있는 정보를 기계가 자동으로 추출해서 활용할 수 있다. 즉 기존의 전처리 방법이나 소위 feature engineering 을 통해 배제되었던 해의 영역조차도 딥러닝은 탐색함으로써 더욱 유용한 정보를 추출하여 활용할 수 있다.

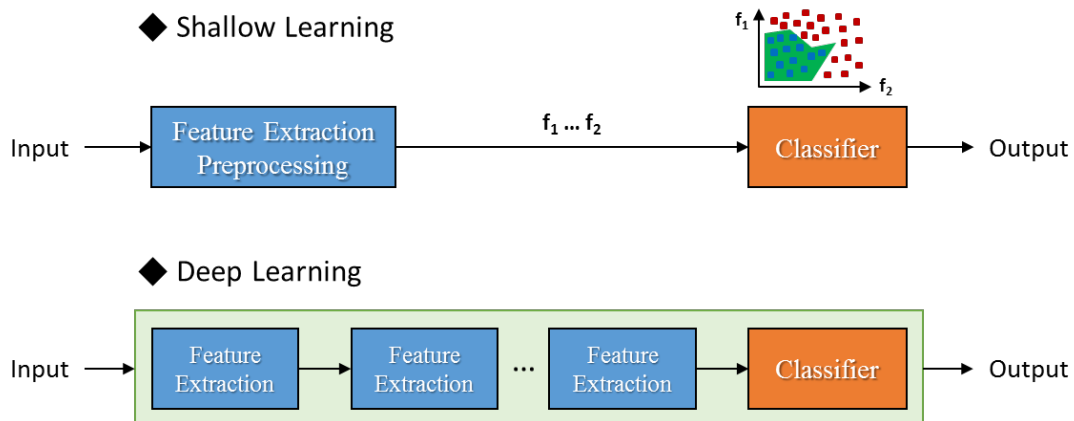


그림 1. 딥러닝 개념

딥러닝과 같이 다층구조의 복잡한 신경망이 유용할 것이라는 것은 과거에도 알고 있었다. 뉴런의 수를 증가하거나 층의 수를 증가시킴으로써 보다 복잡한 패턴 분류 경계면을 생성해 낼 수 있기 때문이다(그림 2). 과거에도 다층구조의 신경망을 활용하려는 시도가 없었던 것은 아니다[8]. 그러나 실험적으로 층의 수를 늘림으로써 학습 시간은 늘어나는데 반해서 성능향상은 얻지 못하였었다. 또한 이론적으로 한 개의 은닉층만을 사용하여도 무한히 많은 수의 뉴런을 사용하면 임의의 복잡한 함수도 근사할 수 있다는 Universal Function Approximator 정리가 1989 년에 증명되었다[9,10]. 또한 1990 년대에 SVM 이 등장하여 아주 빠른 학습이 가능한 shallow network 으로 많은 문제를 해결할 수 있었다.

그렇다면 예전에는 성과를 내지 못하던 딥러닝 모델이 최근 들어 성능 향상을 이룩할 수 있는 이유는 무엇인가? 한 가지 이유는 컴퓨팅 파워가 좋아져서 예전에 할 수 없었던 아주 고난도의 학습 실험을 수행할 수 있기 때문이다. 또한 가용한 학습 데이터가 무한히 많아져서 아주 많은 데이터를 학습시킴으로써 아무리 복잡한 모델구조도 과다학습을 하지 않게 만들 수 있기 때문이다. 여기에, 무엇보다도 대규모 데이터로 대규모 모델을 학습시키는 효율을 향상할 수 있는 여러 가지 학습 구조와 학습 알고리즘적 테크닉들이 개발되었다. 이러한

새로운 기술들이 차원수의 저주 문제, 과다학습 문제, Vanishing Gradient 문제, Non-convex 최적화 문제, 느린 학습 속도 등의 이슈를 일부 해결하였다.


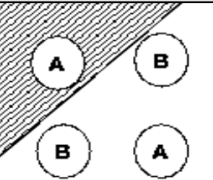
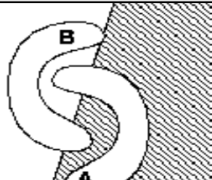
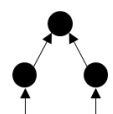
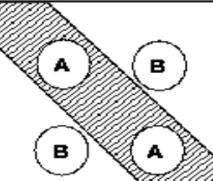
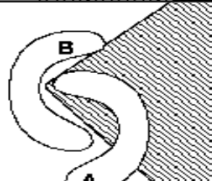
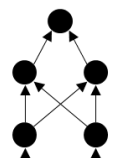
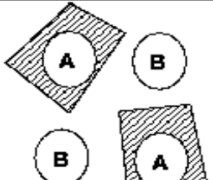
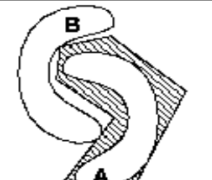
| Structure | Region | XOR | Meshed Regions |
|--|-----------------------------------|--|--|
| Single Layer  | Half plane bounded by hyper-plane |  |  |
| Two Layers  | Convex open or closed regions |  |  |
| Three Layers  | Arbitrary (limited by # of nodes) |  |  |

그림 2. 딥구조 학습 모델의 필요성

이 문제를 해결하는데 핵심적으로 기여한 혁신은 크게 세 가지를 들 수 있다. 첫 번째는, 많은 수의 층으로 구성된 다층신경망을 학습할 수 있는 기술을 개발한 것이다. 다층망을 학습시키는 오류역전과 알고리즘은 층이 많이 쌓으면 학습이 잘 되지 않았다. 출력에 가까운 층에서는 오류의 값이 커서 교정이 되지만 아래 층으로 오류가 역전과 되면서 에러의 값이 줄어들어 변경 효과가 희석되는 vanishing gradient 문제가 발생한다. 따라서 아주 많은 수의 층을 쓰는 딥네트워크는 오류역전과 알고리즘으로 학습이 어렵다. 최근에서야 이 문제를 극복하는 방안으로 층별 선훈련(layerwise pre-training) 방법이 제안되었다[11]. 이는 상위층을 학습하기 전에 먼저 하위층의 시냅스를 학습시켜 둔다(그림 3). 이렇게 순차적으로 하위층부터 학습시킴으로써 Vanishing Gradient 문제로 인해서 하위층의 시냅스 학습이 잘 되지 않는 문제점을 해결한다. 이 방법은 DBN 에서 사용한다.

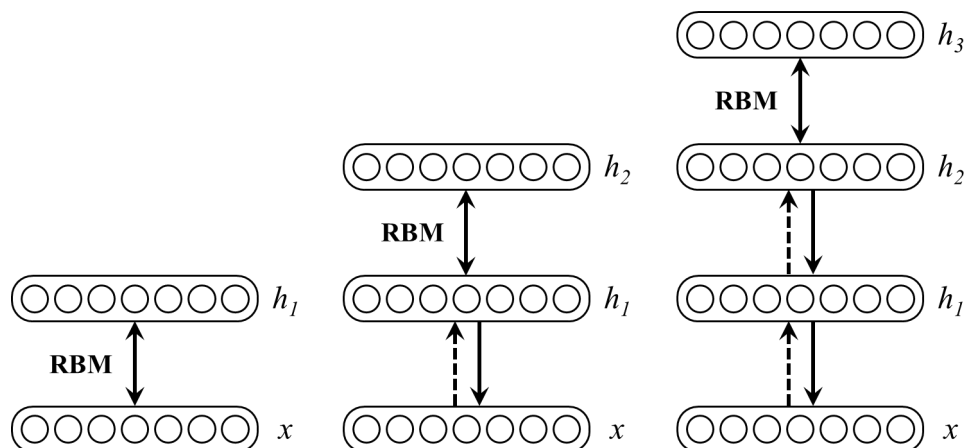


그림 3. 층별 순차적 선트레이닝의 개념

두 번째는 영상과 같이 차원수가 아주 높은 데이터로부터 유용한 특징과 표현을 자동으로 추출하기 위해 컨볼루션 커널(convolution kernels)을 도입한 것이다(그림 4). 이를 통해서 위치가 달라도 같은 파라미터값을 갖도록 함으로써 파라미터의 수를 줄임으로써 학습해야 하는 차원의 수를 줄인다. 이 방법은 CNN 에서 사용하는 방법이다. 이 방법은 과다학습을 방지하면서 유용한 특징을 추출할 수 있는 장점이 있다.

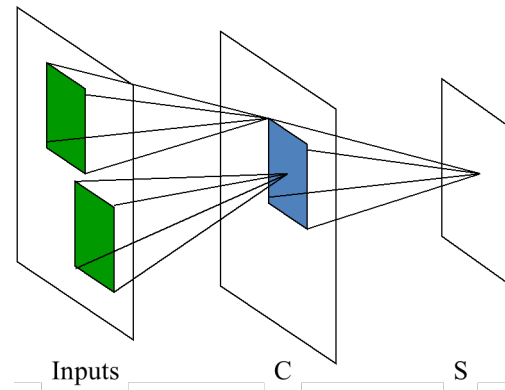


그림 4. 컨볼루션 네트워크

세 번째로, 학습 방법을 변경하는 대신 새로운 뉴런 활성화 함수를 가진 유닛을 도입한 것이다. ReLU 유닛(rectified linear units), 즉, 정류선형유닛은 뉴런이 선형적으로 활성화되어 큰 값을 가질 수 있게 하여 경사도가 상수가 되도록 함으로써 오류 역전파를 해도 경사도가 사라지지 않도록 하는 효과가 있다[12]. 그림 5 는 시그모이드 유닛과 ReLU 유닛의 특성을 비교하고 있다.

$$f(x) = \frac{1}{1 + \exp(-x)} \text{ vs. } f(x) = \max(0, x)$$

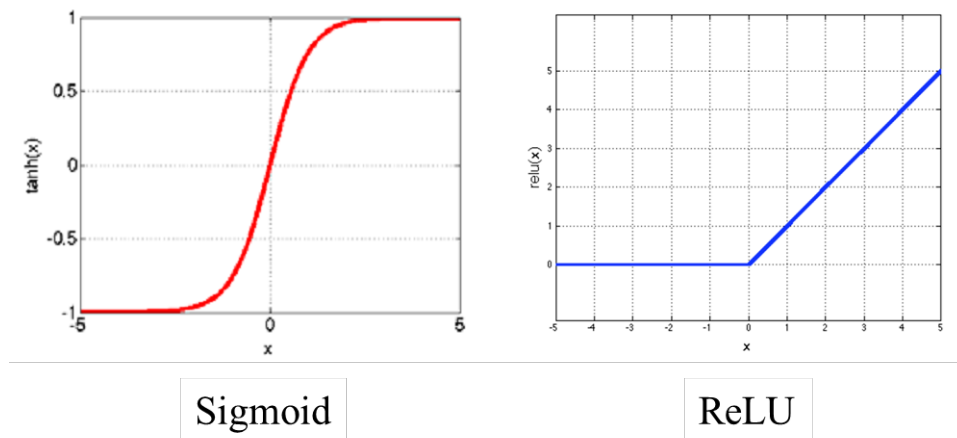


그림 5. 시그모이드 함수와 ReLU 의 비교

시그모이드 유닛은 0 과 1 사이의 값으로 압축됨으로써 Vanishing Gradient 문제를 유발한다. 이에 반해서 정류선형유닛은 포화가 되지 않고 빠르게 수렴하는 특성이 있다.

3.3 딥러닝 아키텍처

현재 가장 많이 사용되는 딥러닝 모델은 CNN 과 DBN 이다[7,11]. 이 두 가지 모델은 다수의 뉴런층을 사용한다는 점에서 있어서 유사하다. 그러나 이 두 모델은 여러 가지 면에서 차이가 있다. CNN 은 기본적으로 감독학습 문제를 풀도록 설계되어 있는 반면, DBN 은 무감독학습 문제를 목표로 한다(그림 6, 그림 7). 그러나, CNN 도 전단에서는 무감독 학습으로 특징을 추출하고, 또한 DBN 도 마지막 단계에서 감독학습을 적용할 수는 있어 두 모델 모두 무감독학습과 감독학습이 결합된 형태로 해석할 수는 있다. 또 다른 차이점은 CNN 은 입력데이터를 분류하기 위한 변별적 학습에 초점이 맞추어져 있는 반면에 DBN 은 입력데이터를 재생성하는 생성적 학습에 초점이 있다. 전자의 장점은 패턴 분류 성능이 좋다는 것이고 단점은 모델로부터 샘플을 생성해 낼 수 없다는 것이다. 후자는 반대로 패턴 분류에 적용할 경우 분류 성능은 최적은 아닐 수도 있으나 모델로부터 새로운 샘플을 생성해 낼 수 있다는 장점이 있다. 이는 일반적으로 변별적, 생성적 머신러닝 방법들이 갖는 장단점과 같다. 다만 딥러닝은 아주 많은 수의 뉴런층을 사용하여 아주 복잡한 특징과 표현을 스스로 구축함으로써 아주 복잡한 문제를 풀수 있다는 것이 큰 장점이다. 최근의 CNN 기반 딥러닝 모델의 경우 22 층짜리가 등장하였다(그림 8).

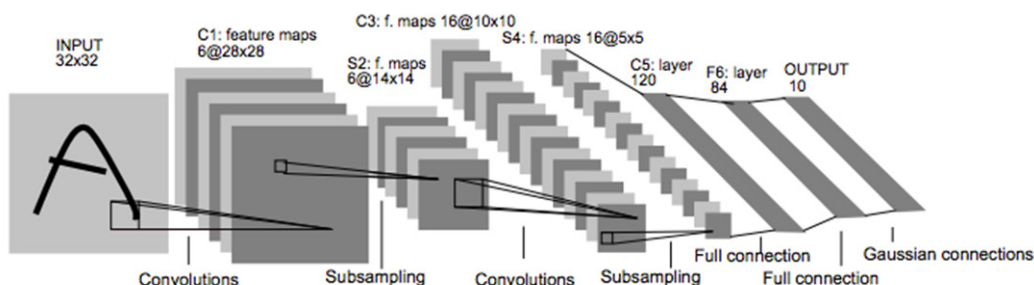


그림 6. Convolutional Neural Network (CNN)의 구조

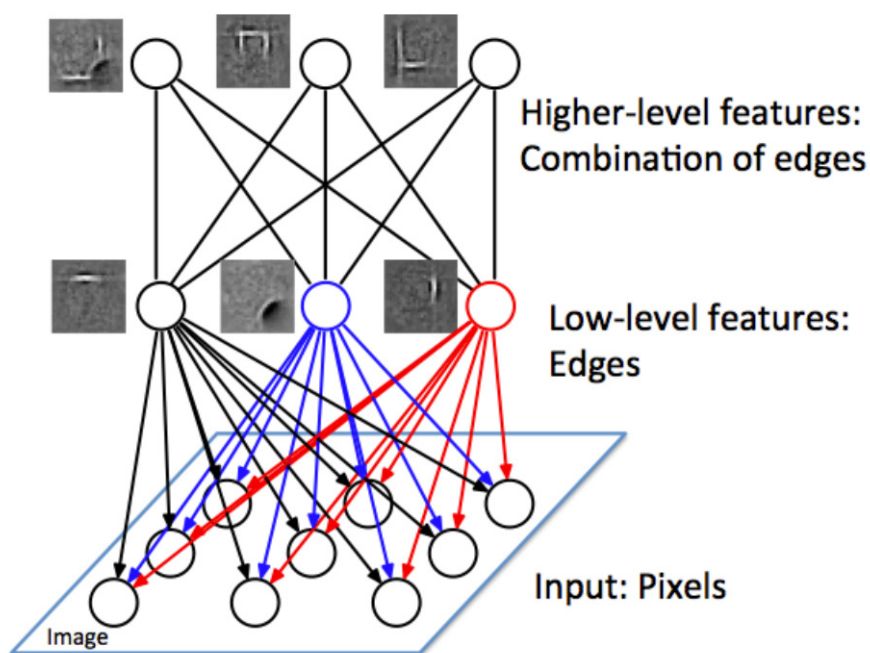


그림 7. Deep Belief Network (DBN)의 구조

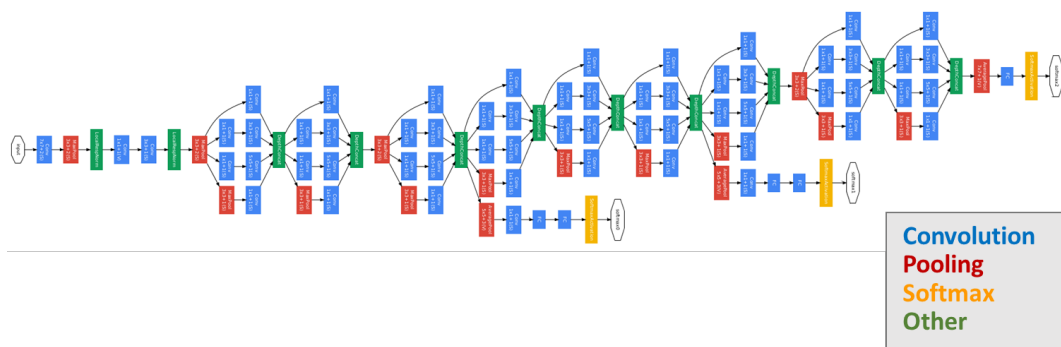
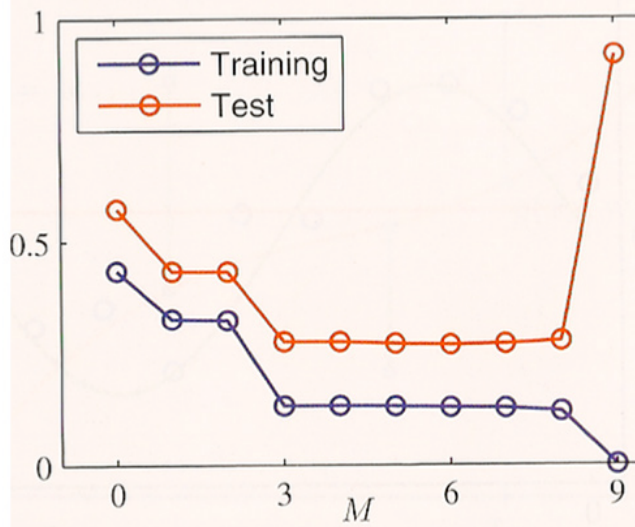


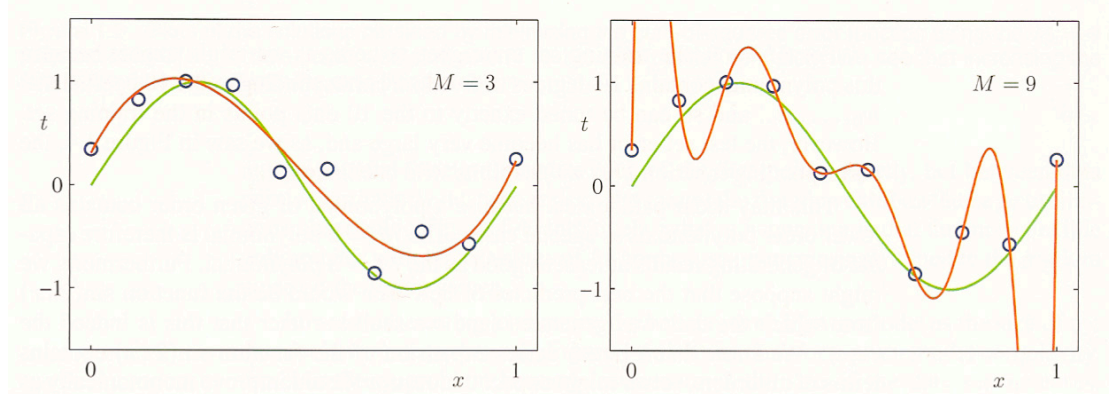
그림 8. GoLeNet 구조

3.4 모델복잡도 이론과 정규화

3.4.1 과다학습



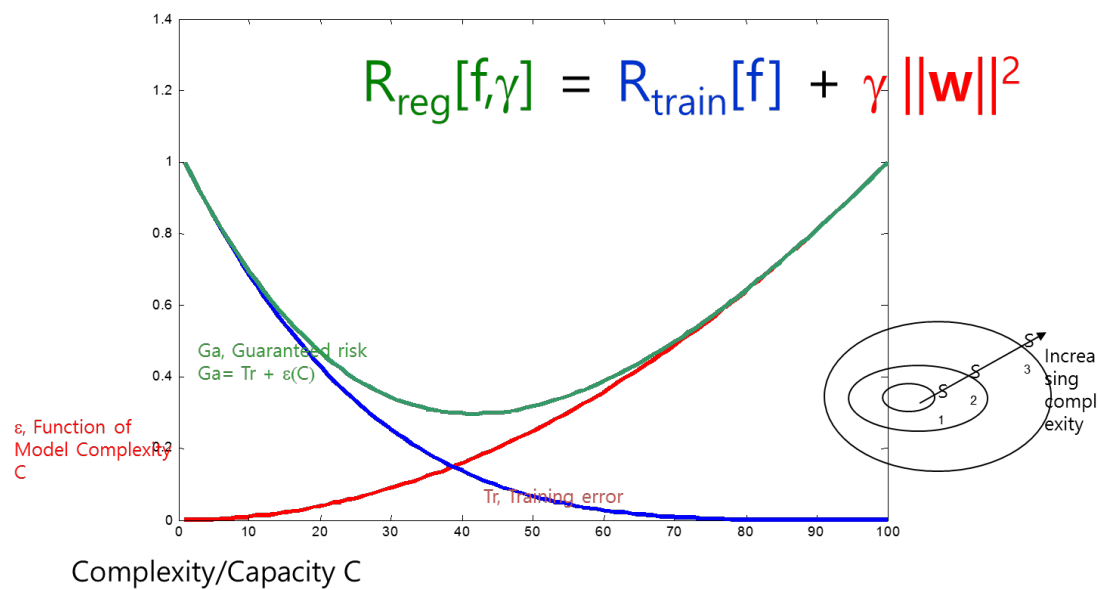
3.4.2 모델복잡도



3.4.3 Occam's Razor

- Principle proposed by William of Ockham in the fourteenth century: "Pluralitas non est ponenda sine neccesitate".
- Of two theories providing similarly good predictions, prefer the simplest one.
- Shave off unnecessary parameters of your models.

3.4.4 Regularization & Structural Risk Minimization (SRM)



3.4.5 MAP

- **Maximum A Posteriori (MAP):**

$$\mathbf{f} = \operatorname{argmax} P(\mathbf{f}|\mathbf{D})$$

$$= \operatorname{argmax} P(\mathbf{D}|\mathbf{f}) P(\mathbf{f})$$

$$= \operatorname{argmin} -\log P(\mathbf{D}|\mathbf{f})$$

$$-\log P(\mathbf{f})$$

Negative log
likelihood =
Empirical risk $R_{\text{emp}}[\mathbf{f}]$

Negative log
prior =
Regularizer $\Omega[\mathbf{f}]$

- **Structural Risk Minimization (SRM):**

$$\mathbf{f} = \operatorname{argmin} R_{\text{emp}}[\mathbf{f}] + \Omega[\mathbf{f}]$$

3.4.6 MDL

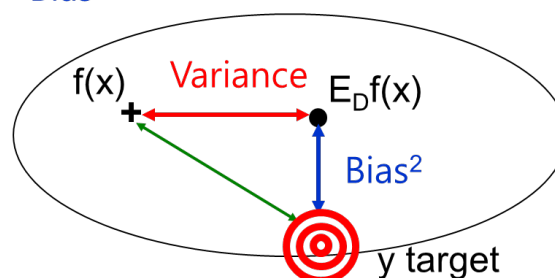
- MDL: minimize the total length of the “message”.
- Two part code: transmit the model and the residual.
- $f = \operatorname{argmin} -\log_2 P(D|f) - \log_2 P(f)$

| | |
|--|--|
| $\underbrace{\hspace{10em}}$ <p>Residual: length of the shortest code to encode the data given the model</p> | $\underbrace{\hspace{10em}}$ <p>Length of the shortest code to encode the model (model complexity)</p> |
|--|--|

3.4.7 Bias-Variance Tradeoff

- f trained on a training set D of size m (m fixed)
- For the square loss:

$$\underbrace{E_D[f(x)-y]^2}_{\text{Expected value of the loss over datasets } D \text{ of the same size}} = \underbrace{[E_D f(x)-y]^2}_{\text{Bias}^2} + \underbrace{E_D[f(x)-E_D f(x)]^2}_{\text{Variance}}$$



3.5 딥러닝 모델의 비교

No Free Lunch Theorem 에 의하면 학습 모델마다 잘 푸는 문제가 있으면 못 푸는 문제가 있기 마련이다. 딥러닝 모델도 마찬가지이다. 여기서는 딥하이퍼넷 모델의 특성을 DBN, CNN 과 비교함으로써 딥하이퍼넷이 어떤 문제에 더 적합하고 어떤 문제에 덜 적합한지를 분석한다. 표 1 은 다음의 7 가지의 비교 차원에 대해 요약 정리한 것이다.

| | Deep Belief Net (DBN) | Convolutional Neural Net (CNN) | Deep Hypernet (DHN) |
|-----------|-----------------------|--------------------------------|---------------------|
| 감독/무감독 | 감독/무감독 | 감독 | 감독/무감독 |
| 변별/생성 모델 | 생성 | 변별 | 생성 |
| 예측/모듈 이해 | 예측++/모듈- | 예측+++ /모듈+ | 예측+/모듈+++ |
| 추론 가능성 | 추론++ | 추론+ | 추론++++ |
| 연결성 | Full/Compact | Partial/Convolved | Partial/Sparse |
| 깊이 | 깊이+++ | 깊이++++ | 깊이++ |
| 배치/온라인 학습 | 배치 | 배치 | 온라인 |

□ 1. 딥러닝 모델의 특성 비교

- 1) 감독/무감독 학습: CNN 은 명확히 감독학습만을 위한 모델이다. 이에 반해서 DBN 과 DHN 은 기본적으로 무감독학습을 염두에 둔 모델이며 마지막 층에 감독학습층을 추가함으로써 감독학습으로 활용할 수도 있다. 감독학습으로 사용될 경우 무감독학습에 의해 먼저 고차 특징을 추출함으로써 라벨이 없는 데이터를 활용하여 자동으로 표현을 학습할 수 있는 특성이 있다.
- 2) 변별/생성 모델: 변별모델은 입력패턴들의 차이를 구별하는데 집중하며 생성모델은 입력패턴들의 유사성을 파악하는데 집중한다. CNN 은 변별 모델이며 패턴분류 문제에 적합하다. 반면에 DBN 과 DHN 은 생성모델로서 데이터를 압축하거나 샘플을 생성하는데 유용하다.
- 3) 예측/모듈 이해: 값을 예측하는 정확도가 중요한 문제가 있고 데이터의 숨은 구조를 찾아내는 것이 더 중요한 응용이 있을 수 있다. 이 점에서 CNN 은 예측에 가장 적합한 모델이며 모듈 이해는 어렵다. 한편 DHN 은 모듈 이해에 중점을 두는 딥러닝 구조로써 데이터를 재생성하는 빌딩블록을 찾으려는 시도를 한다. DBN 은 CNN 과 DHN 의 중간 정도의 모델로 볼 수 있으며 차원을 축소하며 압축을 반복하면 저차원상에서 복잡한 데이터의 구조를 찾을 가능성이 있다. 다만 이 구조는 빌딩 블록이라기 보다는 전역적인 특성에 해당한다.
- 4) 추론 가능성: 생성모델은 학습된 모델을 이용하여 추론이 가능하다. 즉 관측 변수값을 모델에 넣고 다른 미관측 변수값들을 예측할 수 있다. 이 점에서 DBN 과 DHN 은 추론이 가능한 모델이며 CNN 은 그렇지 못하다. DHN 은 하이퍼에지의 순차적 연결에 의한 구조적인 추론을 통해서 보다 복잡한 표현을 재구성하는 추론을 수행할 수 있다. 위에서 살펴본 바와 같이 DHN 은 만화영화 비디오로부터 영상과 문장을 생성할 수 있다. DBN 은 벡터형태의 정형화된 표현에 대한 추론이 가능하나 DHN 과 같은 구조를 생성하지는 못한다.
- 5) 연결성: 가장 자유로운 연결 구조를 가지는 모델은 DHN 이다. 이에 비해서 DBN 은 인접층간에 항상 완전 연결 구조를 가진다. CNN 은 국부적인 수용영역을 갖는 컨볼루션에 의해 정해진 정규적인 연결 구조를 반복한다. DHN 학습 알고리즘은 자유로운 연결 구조의 공간에서 가장 희소한 구조를 탐색한다. 만화영화 학습 예에서 개념망은 복잡하면서도 희소성을 갖는 이러한 구조에 속한다.

- 6) 깊이: 컴퓨팅 파워의 증가로 딥구조의 층의 수가 증가하고 있다. 영상과 같이 고차원의 데이터에서 고차 패턴을 추출하는데는 CNN 과 같은 선형 지식이 들어간 컨볼루션 커널을 사용하는 다층 구조가 적합하다. 그러나 고차의 문법 구조를 찾는 개념망을 학습하는 문제와 같은 구조적 학습은 음성이나 영상 데이터와는 특성이 달라 너무 층수가 많을 경우 해석이 불가능할 수 있다. 따라서 적정 층의 수는 문제의 특성이나 해의 특성에 따라 다를 수 있다. 현재까지의 응용으로 볼 때 일반적으로 DHN, DBN, CNN 의 순으로 더욱 많은 뉴런층을 사용하는 경향이 있다.
- 7) 배치/온라인 학습: CNN 과 DBN 은 배치 학습을 기본으로 한다. 반면에 DHN 은 온라인 점진적 학습을 목적으로 설계되었다. 이미 많은 데이터가 수집되어 있어 순차적인 학습을 고려할 필요가 없을 때는 CNN, DBN, DHN 방식이 모두 사용될 수 있으나 데이터가 순차적으로 관측되는 상황에서는 DHN 이 더욱 적합하다.