Gradient-Based Learning Applied to Document Recognition

YANN LECUN, MEMBER, IEEE, LÉON BOTTOU, YOSHUA BENGIO, AND PATRICK HAFFNER

배성은

목차

I. ABSTRACT & INTRODUCTION

- 역전파 알고리즘으로 훈련된 Multi Layer Neural Network 경우 좋은 성공 사례다.
- 특히 손글씨와 같이 고차원의 패턴을 가진 데이터를 분류함에 있어서 최소한의 전처리로 좋은 성능을 보여줌.
- 이 논문에서 손글씨를 인식함에 있어 기존의 인식방법과 비교한 결과를 리뷰함.

- 2D 형상을 다루기 위해 고안된 CNN은 다른 기법보다 좋은 성능을 보임.
- 실제 문서 인식 시스템은 field extraction, segmentation, recognition, language modeling을 포함하는 여러 개의 모듈들로 구성되었다.
- GTN이라고 불리우는 새로운 학습 패러다임은 이러한 multi-module 시스템이 경사기반 방법을 사용하여 전반적인 성능 지표를 최소화할 정도로 전역적으로 학습되도록 한다.
- 온라인 손글씨 인식과 은행 수표를 읽는 것에 대한 두가지 시스템, 특히 은행 수표인식의 경우 CNN을 사용하며 비즈니스/개인 수표를 읽는데 기록 정확도를 높여 상업적으로 인식하고 있다.

I. ABSTRACT & INTRODUCTION

- 머신러닝 기술이 특히 neural networks에 적용되는 것이 패턴인식에서 점점 더 중요해졌다.
- 학습기법의 유효성은 음성인식, 필기인식과 같은 패턴인식 성공에 중요한 요인이다.

- 이 논문 주요 메시지는 "패턴인식 시스템은 hand-designed heuristics을 줄이고 자동화학습에 주력해서 만드는 것이 더 좋은 성능을 가진다"
- 보통 패턴인식 시스템은 자동학습 테크닉과 hand-craft 알고리즘 조합으로 만들어짐.
- 개별패턴 방법에는 두 메인 모듈이 있음
  - 1) feature extraction module
    - 입력패턴을 낮은 차원의 특징 벡터로 변화하는 역할
      - A) 쉽게 비교하기 위해
      - B) 변형 및 왜곡에 대해 상대적으로 불변함.
      - 사전 지식 포함, 작업에 구체적임 hand-crafted
  - 2) trainable classifier module
    - General purpose and trainable
    - 이 방법은 디자이너의 능력에 따라 성능이 결정됨.
    - 새로운 문제에는 다시 수행해야함.
- Feature extraction 필요성은 classifier에 사용된 학습기법이 나누어지기 쉬운 클래스의 저차원 공간에서 제한되기 때문이었으나.
- 지난 10년간 3가지 조합으로 바뀌게 됨.
  1) 저렴해진 컴퓨터 가격 성능이 올라 brutal-force 풀이법 가능
  2) 거대 데이터베이스가 생겨서 실제에 가까운 데이터
  3) 고차원 입력 처리 및 의사결정 기능을 할 수 있는 강력한 기계학습 기술
- 이러한 변화로 기존의 인위적인 알고리즘을 통한 feature extraction이 아닌 픽셀 이미지를 직접 이용하는 알고리즘 설명

I. ABSTRACT & INTRODUCTION

A) learning from Data

- Neural network에서 가장 유명한 접근방식 중 하나는 gradient-based learning이 있다.

- Learning machine 다음 함수를 계산
  - Z^P : p번째 입력패턴, W : adjustable(trainable) 매개변수, Y^p : 패턴 Z^p에 대해 예측한 class의 label이거나 각각의 class에 관련된 확률

$$Y^P = F(Z^P, W)$$

- 위 loss function은 p번째 패턴의 실제 label을 의미하는 D^p와 machine 출력사이 불일치(Discrepancy)를 측정

$$E^P = \mathcal{D}(D^P, F(W, Z^P))$$

- Average loss function은 {(Z^1, D^1),..., (Z^p, D^p)}까지 평균오차 의미. 학습과정에서 min의 W찾아야함.

$$\text{value of } W \text{ that minimizes } E_{\text{train}}(W).$$

- 다른 연구에 따르면 예상되는 E_test과 E_train의 관계는 다음식에 근접함.
  - P : train data 크기, h : 복잡도, 알파 : 0.5~1.0, k는 상수

$$E_{\text{test}} - E_{\text{train}} = k(h/P)^{\alpha}$$

- 다음을 최소화하는 방향으로 학습
  - H(W) : regularization function, 베타 : 상수
  - H(W)를 최소화하는 것은 매개변수 공간의 접근 가능한 부분집합이 제한됨.

$$\text{minimizing } E_{\text{train}} + \beta H(W),$$

E_train을 최소화하는 것과 E_test와 E_train간의 차이를 최소화하는 것 사이의 trade-off 제한

I. ABSTRACT & INTRODUCTION

B) Gradient-Based Learning

- Loss function은 loss function에서 매개 변수 값의 작은 변화의 영향을 측정하는 것으로 쉽게 최소화가 가능하다.
- 이런 측정은 loss function의 현재 매개변수에서의 gradient를 이용함.

- 다음과 같은 방법으로 W가 조정된다.

$$W_k = W_{k-1} - \epsilon \frac{\partial E(W)}{\partial W}.$$

- 가장 유명한 절차는 SGD (stochastic gra-dient algorithm) (on-line update라고 불림)
  - Noise와 근사치를 사용하여 파라미터를 업데이트
  - 단일샘플 기반 업데이트
  - 일반적인 사례보다 학습속도 빠름.

$$W_k = W_{k-1} - \epsilon \frac{\partial E^{p_k}(W)}{\partial W}.$$

I. ABSTRACT & INTRODUCTION

C) Gradient Back-Propagation

- 1950년대 후반부터 Gradient기반 학습 절차가 사용되었지만, 선형으로 제한됨.
- 아래 일들이 일어나기 전에는 SGD기술의 널리 퍼지지 않았다.
  1) 손실함수의 local minima는 실제에서는 중요한 문제가 아니다.
  2) 여러 계층을 거친 비선형 시스템의 경사를 계산하는 오차 역전파 등장
  3) 오차 역전파법을 sigmoid 단위를 사용하는 다층 신경망에 사용하면 복잡한 학습 작업을 해결할 수 있음

- 역전파
  - 출력에서 입력으로의 전파를 통해 기울기를 효율적으로 계산
  - 초기의 오차 역전파법은 기울기를 사용하지 않고 중간 레이어의 units을 위한 가상 targets을 사용했음.

- 손실함수를 최소화하는 파라미터를 찾기 위해 경사하강법을 사용
- 기울기를 활용해 오차 역전파법으로 가중치 학습

- 지금까지 가장 널리 사용되는 신경망 학습 알고리즘

I. ABSTRACT & INTRODUCTION

D) Learning in Real Handwriting Recognition Systems

- 가장 성능이 좋은 신경망은 pixel images로부터 직접 관련 feature을 추출하는 방법으로 학습하도록 디자인된, convolutional networks이다.

- 개별 문자를 인식하는 것뿐이 아닌 segmentation이라고 알려진 단어나 문장 내에서 문자를 분리하는 것이 매우 어려움.
  - 이를 수행하기 위해 HOS(Heuristic Over-segmentation) 기술 사용
    - 글자 사이의 많은 잠재적인 다수의 cuts을 발생시키고 점수기반으로 최상의 cut 조합을 선택하는 것으로 구성
- 모델 정확도는 HOS에 의해 생성된 컷의 품질과 분할된 문자, 잘못 분할된 문자와 구별하는 인식기 능력에 따라 달라짐
- 근데 잘못 분할된 문자의 레이블 데이터를 만드는 것이 어려워 이렇게 인식기를 훈련하는 것은 어려움.
- 수동으로 레이블을 지정하면 좋지만 비용이 많이 들며, 라벨링이 모호함
  - Ex) 잘린 4의 오른쪽 절반은 1로할지, 문자가 아니라고 해야할지? 잘린 8은 어떻게 처리할지?

- 섹션 V에 설명된 해결 방법
  1) 문자수준이 아닌 전체 문자열 수준으로 학습한다.
     → gradient-based learning 활용
  2) Segmentation 전부를 제거
     → 입력 이미지의 가능한 모든 위치에 대해 인식기를 sweep하고 '문자 탐지' 능력에 의존한다.
     → 중앙에 있을 때만 받고 없으면 거부.
     → 비용이 많이 들지만, convolutional NN's를 사용하면 비용을 줄일 수 있음.

I. ABSTRACT & INTRODUCTION

E) Globally Trainable Systems

- 패턴인식 시스템은 multi modules로 구성된다.
  - 문석인식 시스템 : field locator(관심영역추출), field segmenter(입력 이미지를 후보캐릭터의 이미지로 자르는), recognizer(각 후보캐릭터를 분류하고 점수를 매기는 인식기),contextual post-processor(가장 정확한 답변을 선택)
- 모듈에서 모듈로 전달되는 정보는 그래프로 수치적인 정보를 가장 잘 표현한다.

- 전형적으로 각 모듈은 수동으로 최적화되거나 문맥 없이 학습된다.
- 더 좋은 대안은 문서 수준에서 문자 오분류 가능성과 같은 전역 오류 측정을 최소화하도록 전체 시스템을 어떻게든 훈련시키는 것.
  - 이상적으로는 모든 매개변수에 대해 전역 손실함수의 최소값을 찾고자함.
  - 성능을 측정하는 손실함수를 시스템의 조정가능한 매개변수와 관련해 미분가능하게 만들 수 있다면 최소값찾기 가능

- 그러나 복잡한 인식 시스템에서는 그래프가 가장 잘 표현된다.
- 이 경우 GT라고 하는 각 모듈은 하나이상의 그래프 입력을 사용하고 그래프를 출력으로 생성.
- 이러한 모듈의 네트워크를 GTN(Graph Transformer Networks)이라고 함.
- 뒤의 섹션에서 GTN의 개념, 전역 손실함수 최소화를 위한 훈련을 보여줌.

II. CONVOLUTIONAL NEURAL NETWORKS FOR ISOLATED CHARACTER RECOGNITION

- 고차원 비선형 매핑 학습을 위한 경사 하강법으로 학습된 다중 네트워크는 이미지 인식에 좋은 성능을 보임.
- 전통적인 방법으로는 손으로 디자인한 feature extractor가 input과 관련된 정보를 추가하고 관련되지 않는 변수는 제거하는 방식으로 feature를 추출했다.
- 학습 가능한 분류기는 feature vector을 class로 분류함.

- Fully connected multilayer를 feature extractor로 주로 사용되었다.
  - Feature extraction을 스스로 수행
  - 대부분 raw 형태로 input을 넣는다.
- 하지만 문제점 몇개 존재.
  1) 일반적인 이미지는 수백개의 변수(픽셀)이 필요한데, 첫번째 layer에 이만큼의 은닉노드가 필요하고 수만개의 가중치도 필요
     1) 파라미터 수가 많으면 모델 복잡도가 높아지고 더 큰 training set이 필요해짐
     2) 메모리 자원 부족
     3) Translation 관점에서 불변하거나 input 지역 왜곡이 일어남 (고정된 크기의 input에 보내지기 전에 정규화, 중앙화하지만 한계)
     4) 원칙적으로 충분한 크기의 FC 신경망은 변화로부터 불변하는 결과를 생성 (변화가능한 공간 포함하기 어려움)
     5) CNN에서는 가중치 구성을 강제 복제하여 shift invariance가 자동으로 얻어진다.
  2) Input의 topology가 완전히 무시된다.
     1) Imput의 변수는 고정된 순서로 학습 결과에 영향을 미치지 않고 등장한다.
     2) 하지만 이미지는 강력한 2차원 구조로 공간적, 시간적으로 근처에 있는 변수끼리 높은 상관관계
     3) 지역연관성은 지역 feature를 잘 추출하고 조합하는 이유이다. 이웃변수의 구성을 분류하기 가능
     4) CNN은 hidden units의 receptive field를 통해 지역 feature가 추출되도록 한다.

→ FC Layer는 매우 많은 변수(픽셀)을 가진 이미지를 학습하기에 가중치가 너무 많이 필요하고 지역정보를 반영하지 못하지만 CNN은 가능

II. CONVOLUTIONAL NEURAL NETWORKS FOR ISOLATED CHARACTER RECOGNITION

A) Convolutional Networks

- CNN은 shift, scale, distortion invariance 3가지 아키텍처 결합
  1) Local receptive fields
  2) Shared weights
  3) Spatial or temporal subsampling
- Input은 size가 normalized되고 centered된 이미지만 받는다. 각 계층의 노드는 이전 계층의 small neighborhood가 위치한 노드 set을 input으로 받는다.
- Input을 지역 receptive field로 잇는 아이디어는 60년대의 퍼셉트론으로 거슬러 올라감.
  - 지역 receptive field로 뉴런들은 원소들의 가장자리, 끝점, 모서리와 같은 시각 feature를 추출할 수 있음.
  - 고차원의 기능을 감지하기 위해 후속 레이어에서 결합된다.
  - 입력의 왜곡이나 이동으로 인해 두드러진 특징 위치가 달라질 수 있음.
  - 서로 다른 위치에 있는 단위가 동일한 가중치 벡터를 갖도록 강제 적용가능
  - 레이어의 단위는 모든 단위가 동일한 가중치 집합을 공유하는 평면으로 구성된다.
  - 이러한 평면에 있는 단위의 출력 집합을 특징맵이라고 한다. 특징맵의 unit은 이미지의 다른 부분에서 같은 작용을 함.
  - 완전한 cnn은 각각의 위치에서 여러개의 feature를 추출하기 위해 weight vector가 다른 여러개의 feature map으로 구성되어있음.
  - 이러한 이유가 CNN이 왜곡과 이동에 강건하다는 기초적 특성

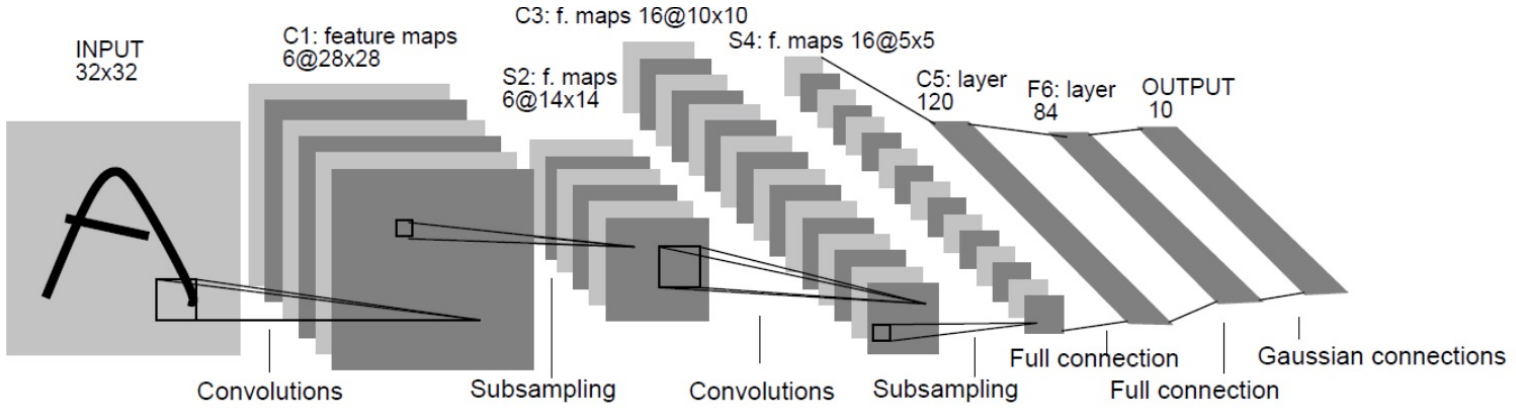


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

II. CONVOLUTIONAL NEURAL NETWORKS FOR ISOLATED CHARACTER RECOGNITION

A) Convolutional Networks

Layer C1 : 6개의 feature map

- 입력으로부터 5x5 filter 6개를 통해 feature map 28x28 사이즈로 6개 생성

  5x5 filter = 25(weight) x 6개 filter + 1(bias) x 6개 filter = 150 + 6 = 156개

- Convolutional networks과 동일하며 입력 이미지가 이동하면 특징 맵 출력도 같은 양만큼 이동하지만 그렇지 않으면 변경되지 않은 채 유지되어 입력의 이동과 왜곡에 대한 견고성이다.
- 그래서 정확한 위치가 덜 중요해서 대략적인 위치만 관련되게 된다.
- 다양한 가중치 및 bias를 이용해서 다양한 유형의 local feature를 추출함.
- Feature가 발견되면 정확한 위치는 덜 중요하고 다른 feature와의 상대적인 위치만이 관련있다.

- 특징맵에서 고유한 특징의 위치가 인코딩되는 정밀도를 줄이는 방법은 특징맵의 공간적 해상도를 줄이는 방법이 있다. → subsampling

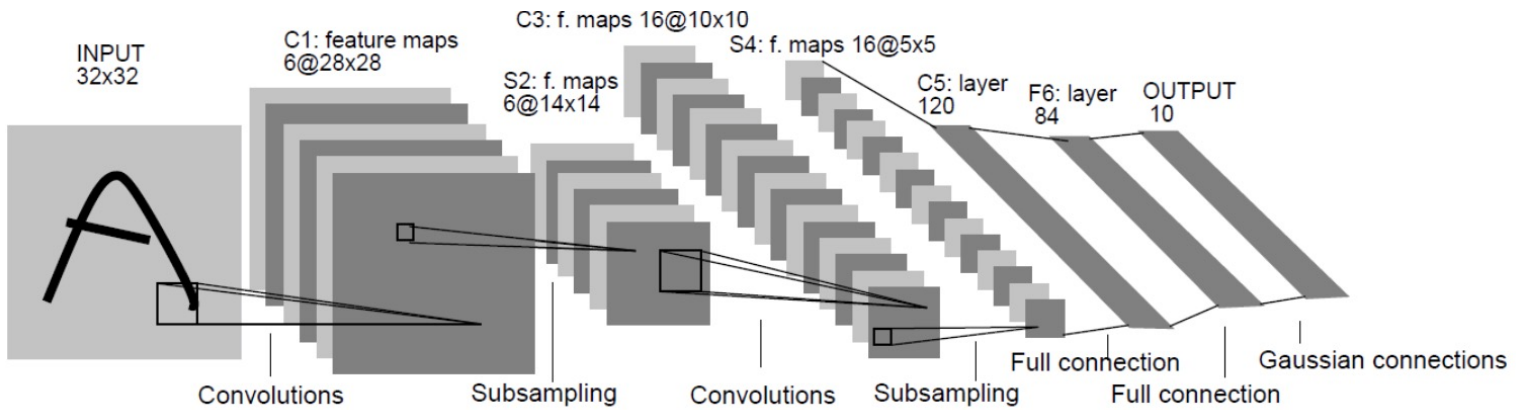
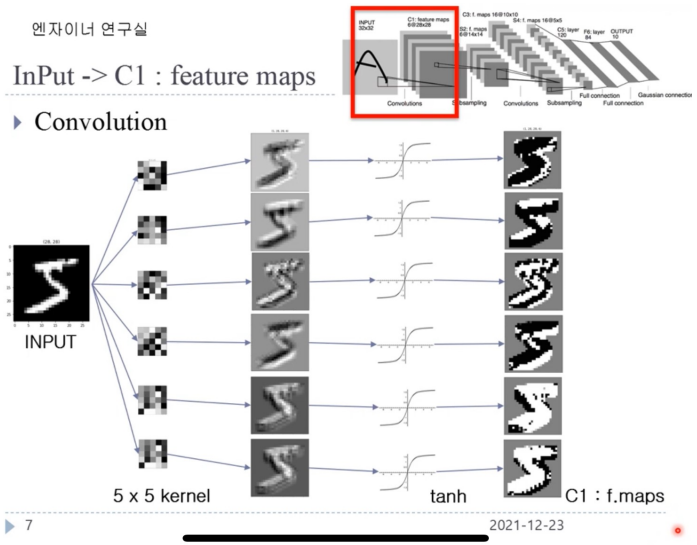


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.



II. CONVOLUTIONAL NEURAL NETWORKS FOR ISOLATED CHARACTER RECOGNITION

A) Convolutional Networks

Layer S2: 14x14 6개의 feature map

- Subsampling을 통해 특징맵의 해상도를 줄이고 이동 및 왜곡에 대한 출력의 민감도를 줄인다.

- 2x2 filter 6개를 통해 14x14 feature map 6개를 생성, 여기서 filter가 적용될때 2x2 receptive field가 overlapping되지 않도록 적용.
- Average pooling을 수행함으로 weight 1개, bias 1개의 파라미터를 가지고 최종적으로 sigmoid 함수가 적용.
- 학습가능한 파라미터 개수는 총 12개

- 2x2 filter = 1(weight, average pooling) x 6개 filter + 1(bias) x 6개 filter = 6 + 6 = 12개

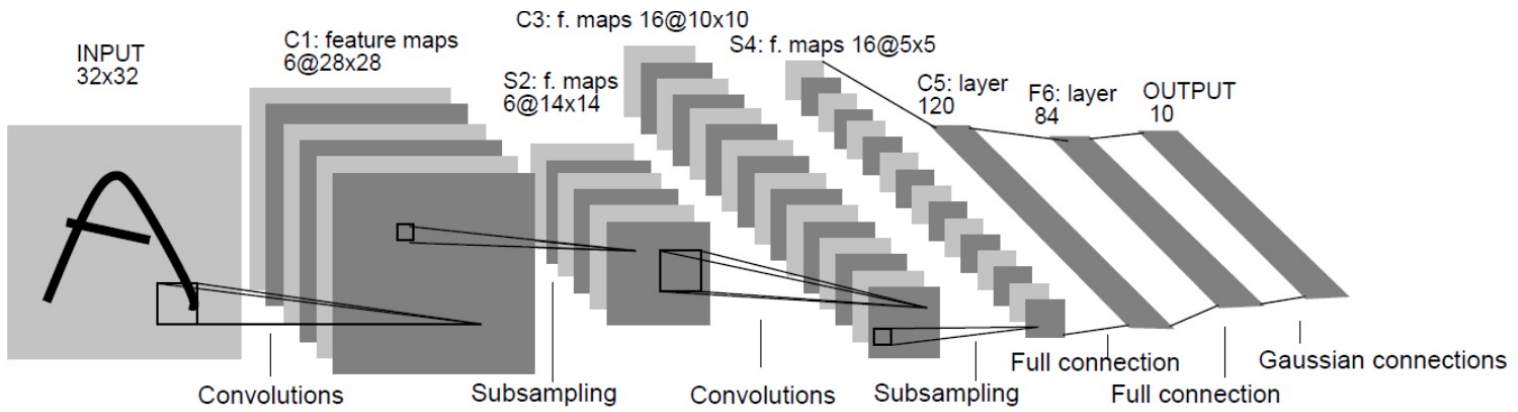
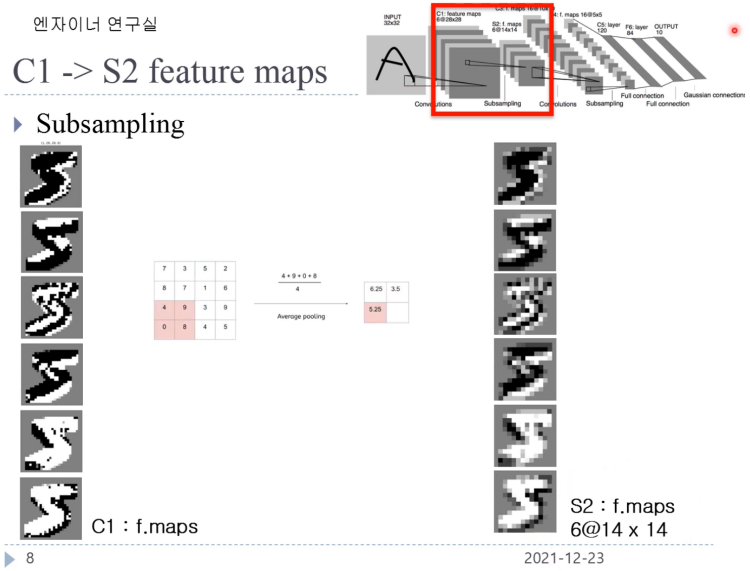


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.



II. CONVOLUTIONAL NEURAL NETWORKS FOR ISOLATED CHARACTER RECOGNITION

A) Convolutional Networks

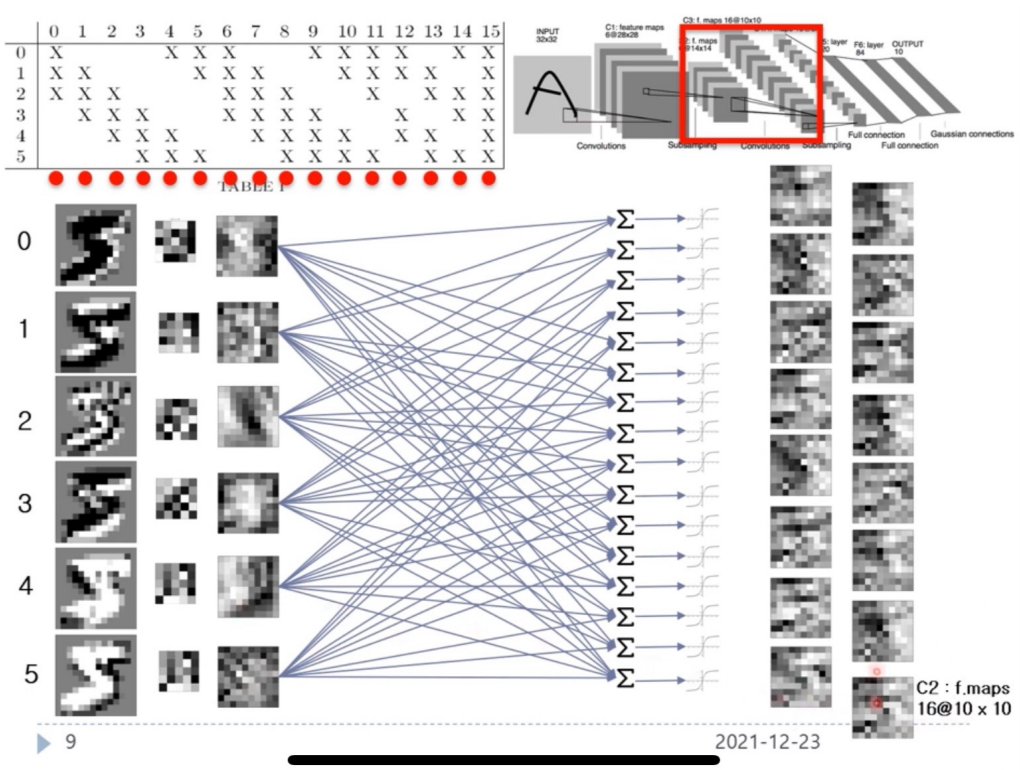
Layer C3: 16개의 feature map

- 입력데이터로부터 5x5 filter 16개를 통해 10x10 feature map 16개를 생성
- 6개의 데이터로부터 16개를 만드는데, 아래 테이블과 같이 선택적으로 연결시키며 network의 symmetry한 성질을 없애기 위함.
- Global feature를 얻기 위해.
- Feature map의 수가 증가하면 공간적 해상도(spatial resolution) 감소되고 입력 변환에 대해 invariance 달성가능.
- 5x5 filter = 25(weight) x 60(S2 feature map과 C3 feature map간 connection수, 아래 테이블 X 표시 개수) + 1(bias) x 16개 filter = 1,516개

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

TABLE I

EACH COLUMN INDICATES WHICH FEATURE MAP IN S2 ARE COMBINED BY THE UNITS IN A PARTICULAR FEATURE MAP OF C3.



II. CONVOLUTIONAL NEURAL NETWORKS FOR ISOLATED CHARACTER RECOGNITION

B) LeNet-5

Layer S4: 5x516개의 feature map

- 입력데이터로부터 2x2 filter 16개를 통해 5x5 feature map 16개를 생성

- 2x2 filter = 1(weight, average pooling) x 16개 filter + 1(bias) x 16개 filter = 16 + 16 = 32개

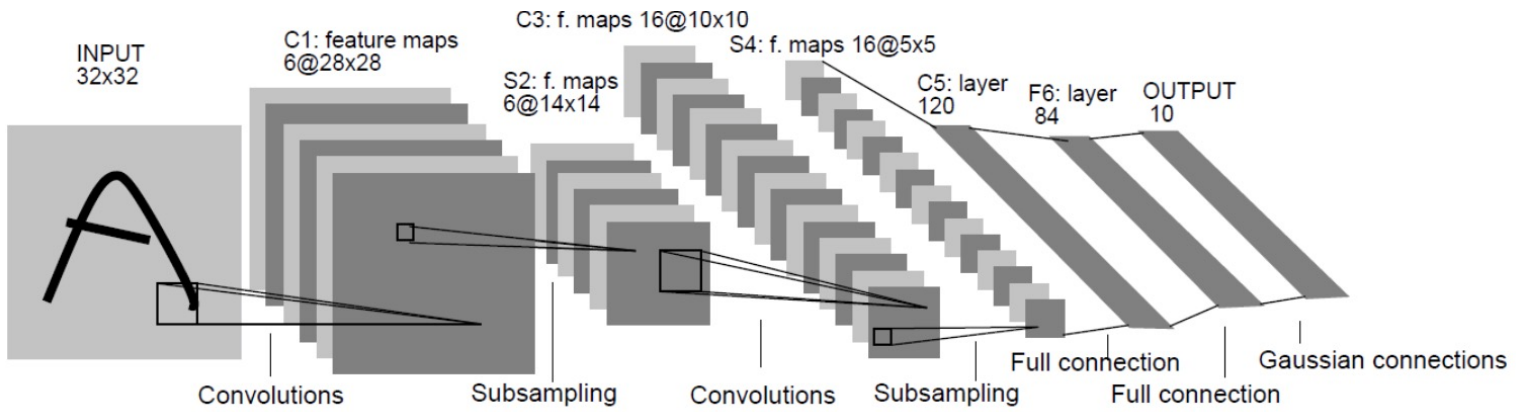
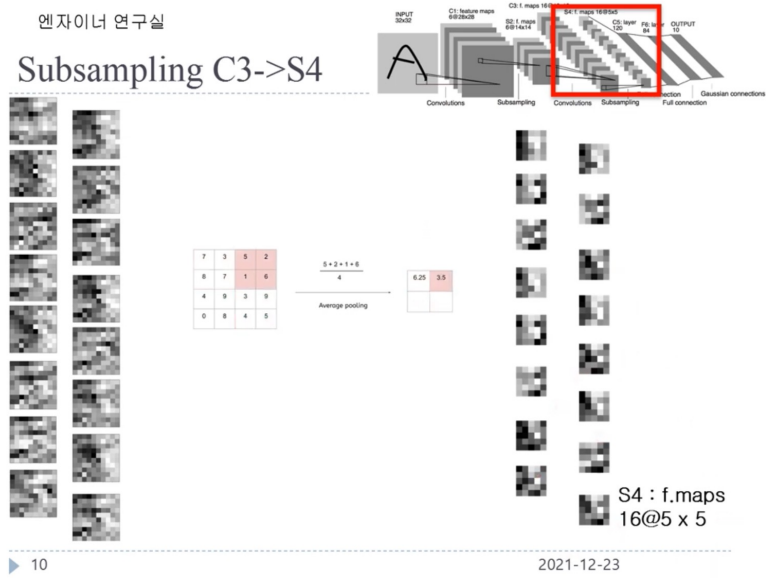


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.



II. CONVOLUTIONAL NEURAL NETWORKS FOR ISOLATED CHARACTER RECOGNITION

B) LeNet-5

Layer C5: 120개의 feature map

- 입력데이터로부터 5x5 filter 120개를 통해 1x1 feature map 120개를 생성(fully-connected)
- 이전단계에서 얻은 16개의 feature
- 5x5 filter = 25(weight) x 1,920(S4 feature map과 C5 feature map간 connection수) + 1(bias) x 120개 filter = 48,120개

Layer F6: 84units, C5 FC

120(weight) x 84개 filter + 1(bias) x 84개 = 10,164개

Output : 10개 클래스로 구분. 각 클래스마다 하나씩, 각각 84개의 입력을 가짐.

84(weight) x 10개 + 1(bias) x 10개 = 850개

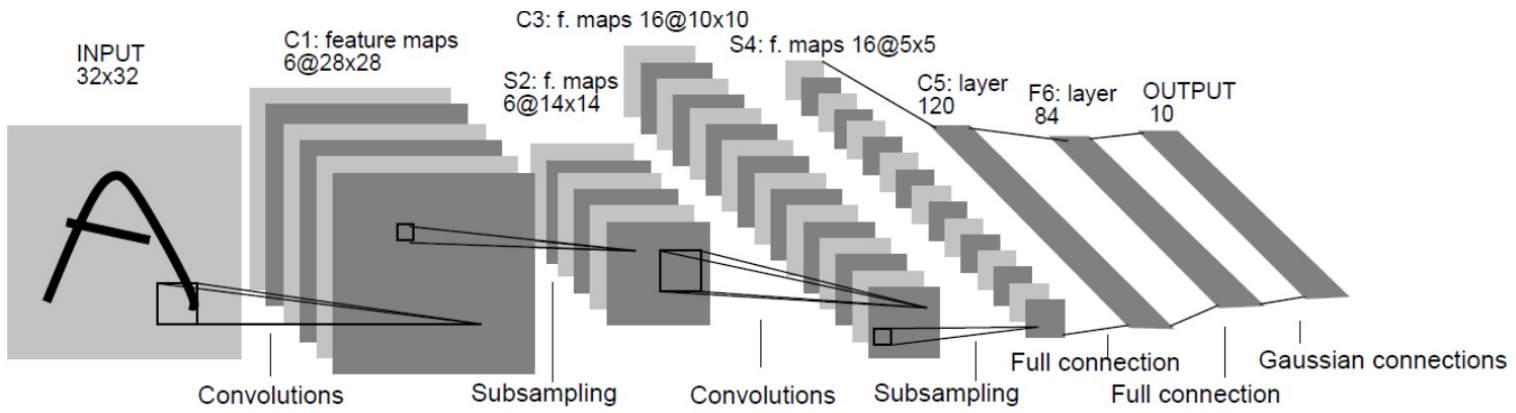
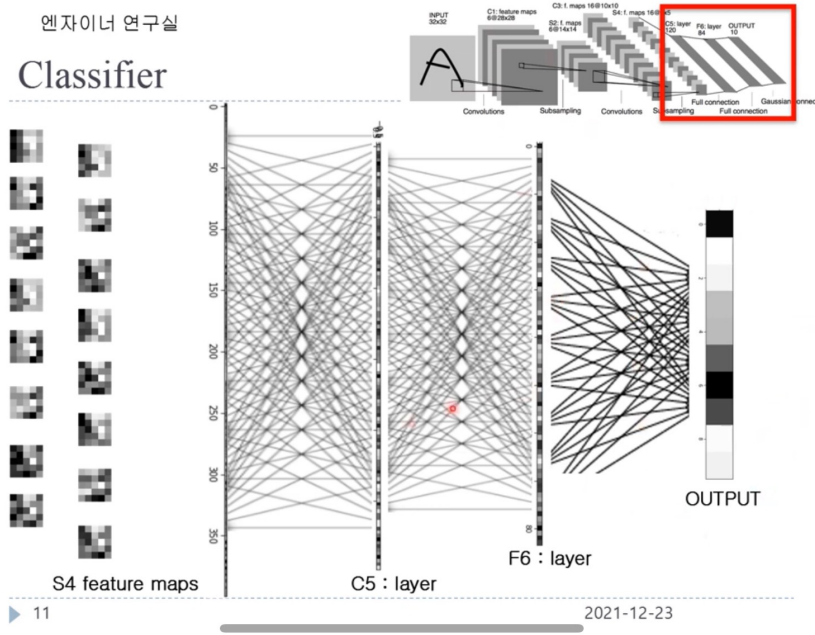


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.



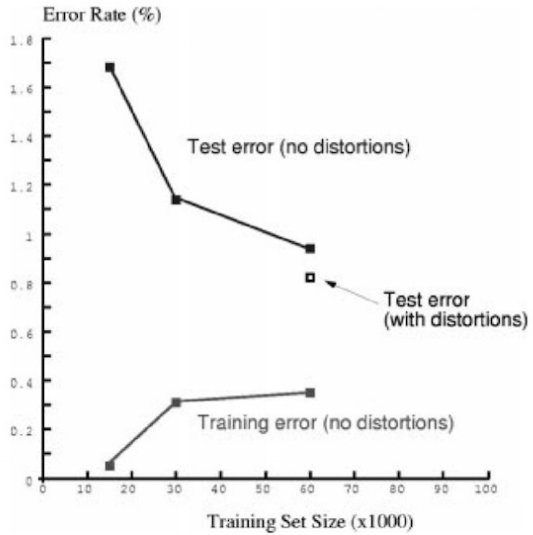
III. RESULTS AND COMPARISON WITH OTHER METHODS

- MNIST data set 사용
  - 20*20 이미지는 28*28 픽셀 중앙을 계산하여 위치하였음 → regular database
  - 글자를 기울이고 20*20 자름 → deslanted database

- LeNet-5 regular database로 20번 반복학습
  - Global 학습률은 처음 두 학습은 0.0005, 다음 세번은 0.0002, 다음 세번은 0.0001, 다음 4번은 0.00005 그 후는 0.00001
  - 각 반복후 500개의 샘플로 diagonal hessian approximation을 재 평가했고 학습시 고정
- 학습 데이터가 많을수록 성능 증가
  - 인공적으로 무작위 왜곡으로 원조 학습 이미지를 통해 이미지를 더 생성
  - Test error를 0.95 -> 0.8%로 감소시킴.

- raw data에서는 Boosted LeNet-4이 에러율 0.7%로 가장 낮았고 그 다음을 LeNet-5가 이었다.
  - boosting은 메모리와 컴퓨팅 비용에 penalty를 주며 성능을 높였다.
  - 왜곡(distortion) 모델은 더 많은 데이터 없이 더 효율적인 데이터 크기를 증가시킨다.

- LeNet-5 강건한 모델이다.



- 참고 링크

https://jjuon.tistory.com/21

https://chaelin0722.github.io/paperreview/Lenet-5/

https://velog.io/@rnjsdb72/%EB%85%BC%EB%AC%B8-%EB%A6%AC%EB%B7%B0-Gradient-Based-Learning-Applied-to-Document-RecognitionLeNet

https://chaelin0722.github.io/code/LENET-5_code/