5장. 딥러닝 - I

5.1 딥러닝

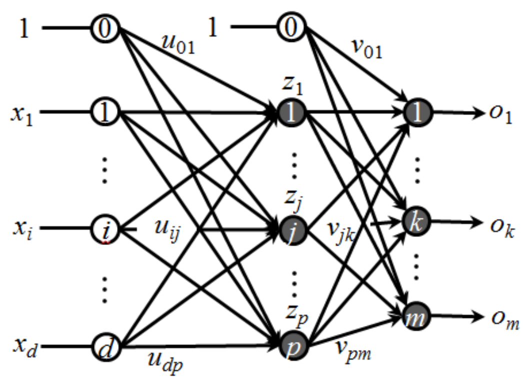
5.1.1 기울기 소멸 문제

5.1.2 기중치 초기화

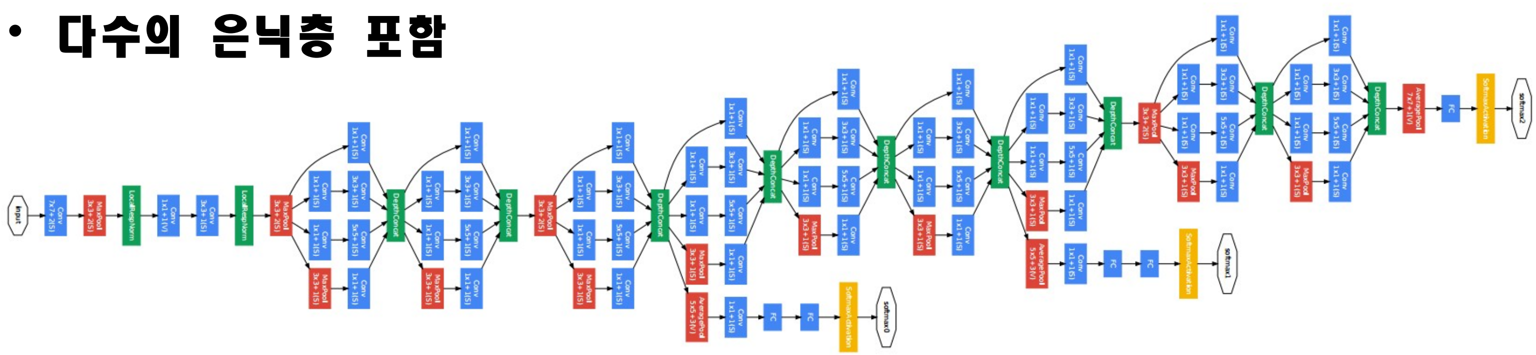
5.1.3 과적합 문제

5.1 딥러닝

- ❖ **딥러닝**(deep learning, **심층학습**, 깊은 학습, 심화학습)
  - ▪ 일반 신경망
    - • 소수의 은닉층 포함



  - ▪ 딥러닝 신경망 (심층 신경망, deep neural network)
    - • 다수의 은닉층 포함



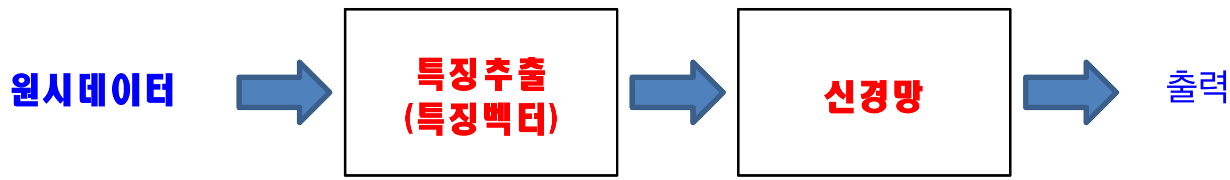
출전: googLeNet (google)

딥러닝

❖ 일반 신경망과 딥러닝 신경망

▪ 일반 신경망 모델

- 원시 데이터(original data)에서 직접 특징(**handcrafted feature**)을 추출해서 만든 **특징 벡터**(feature vector)를 입력으로 사용
- 특징 벡터들의 품질에 영향



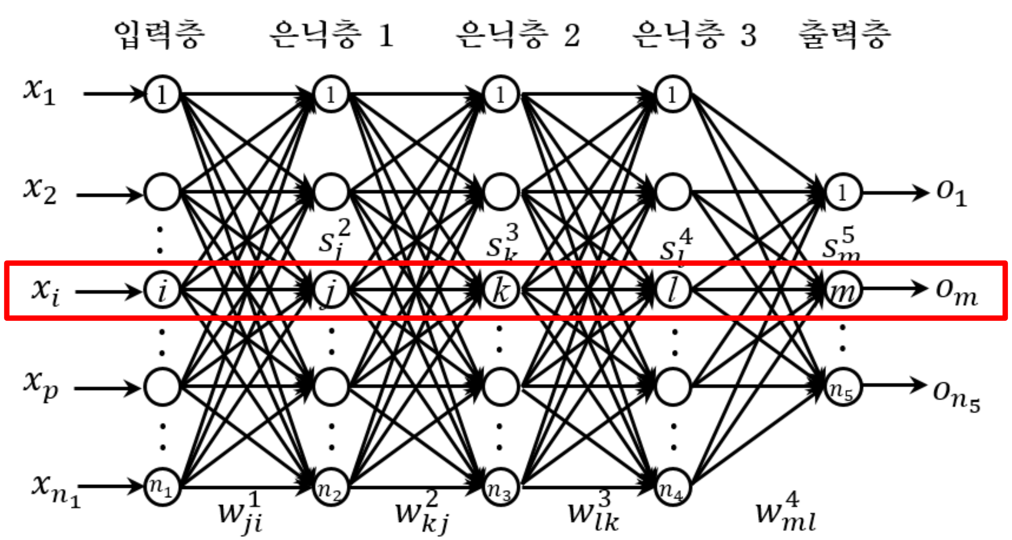
▪ 딥러닝 신경망

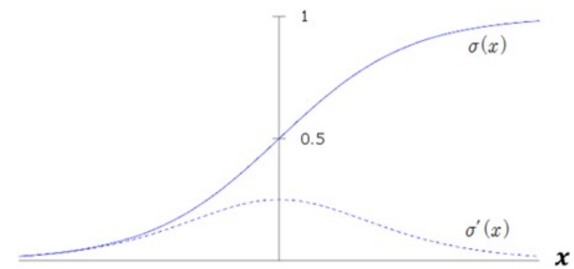
- **특징추출**과 **학습**을 함께 수행
- 데이터로부터 **효과적인 특징**을 학습을 통해 추출 ⊠ 우수한 성능

5.1.1 기울기 소멸 문제

❖ 기울기 소멸 문제(Vanishing gradient problem)

- 은닉층이 많은 다층 퍼셉트론에서, 출력층에서 아래 층으로 갈 수록 전달되는 오차가 크게 줄어들어, 학습이 되지 않는 현상

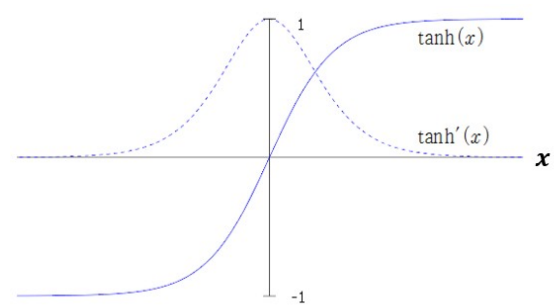




시그모이드(sigmoid)

$$E = \frac{1}{2}\sum_{m=1}^{n_5}(y_m - o_m)^2$$

$$\frac{\partial E}{\partial w_{ji}^1} = \sum_{m=1}^{n_5}\sum_{l=1}^{n_4}\sum_{k=1}^{n_3}\sum_{j=1}^{n_2}(y_m - o_m)f'(s_m^5)\,w_{ml}^4 f'(s_l^4)w_{lk}^3 f'(s_k^3)\,w_{kj}^2 f'(s_j^2)\,x_i$$

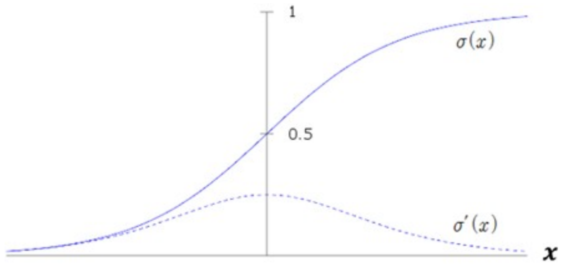


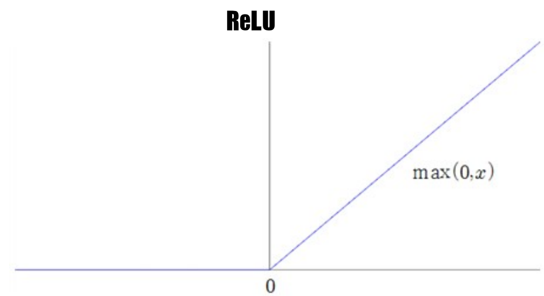
쌍곡 탄젠트(hypertangent)

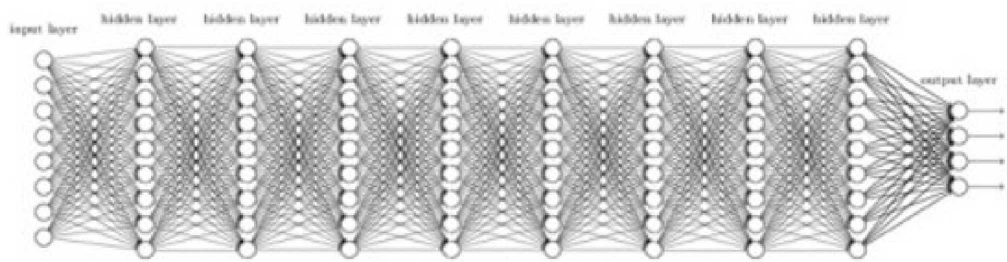
기울기 소멸 문제

❖ 기울기 소멸 문제 완화

- 시그모이드나 쌍곡 탄젠트 대신 **ReLU(Rectified Linear Unit)** 함수 사용

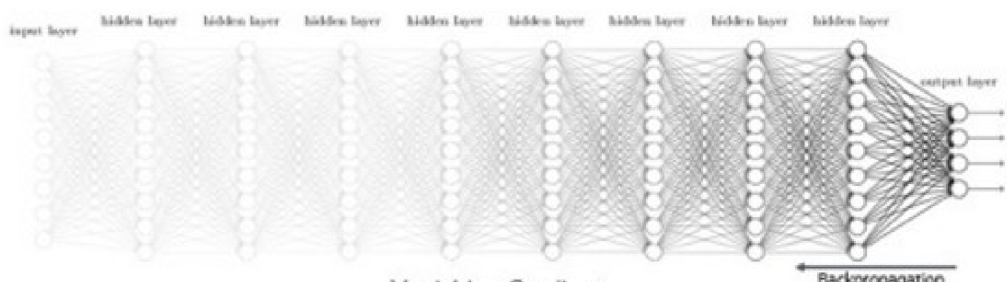






Deep Neural Network

ReLU 사용



Vanishing Gradient

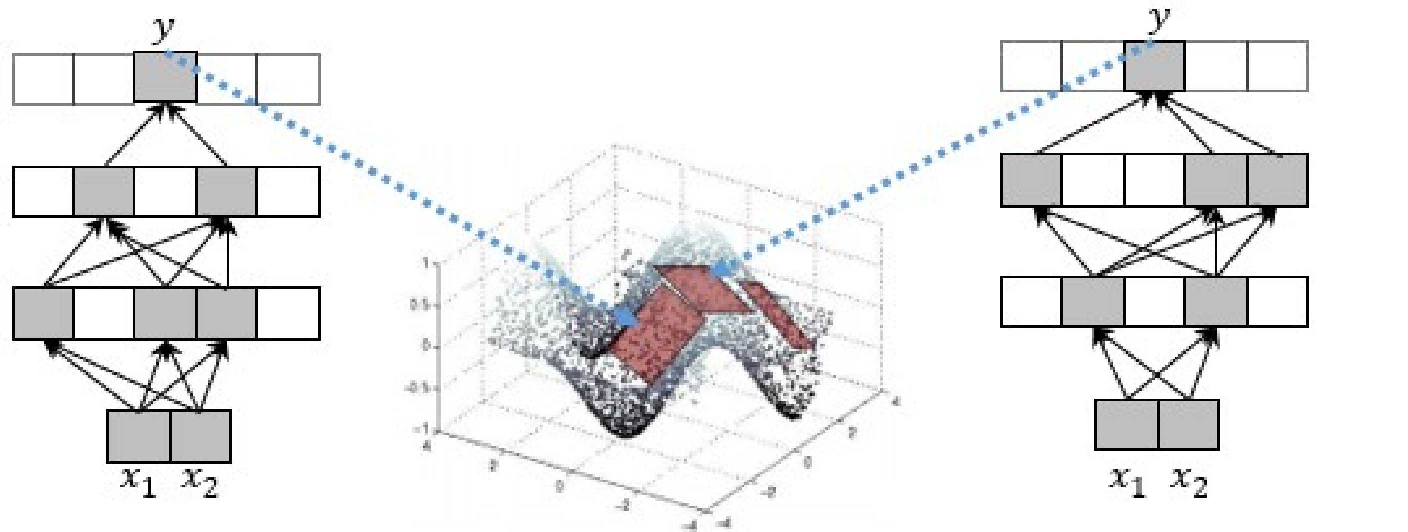
시그모이드 사용

기울기 소멸 문제

❖ ReLU 함수 사용과 함수 근사

- 함수를 부분적인 평면 타일들로 근사(approximate)하는 형태
- 출력이 0이상인 것들에 의해 계산되는 결과
  - 입력의 선형변환(입력과 가중치 행렬의 곱으로 표현)의 결과

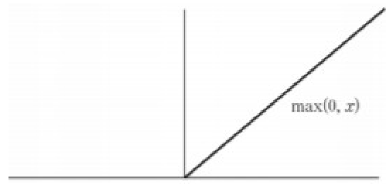
$$y = [\ \ ]_{1\times 2}\cdot[\ \ ]_{2\times 3}\cdot[\ \ ]_{3\times 2}\,\boldsymbol{x} \qquad y = [\ \ ]_{1\times 3}\cdot[\ \ ]_{3\times 2}\cdot[\ \ ]_{2\times 2}\,\boldsymbol{x}$$



기울기 소멸 문제

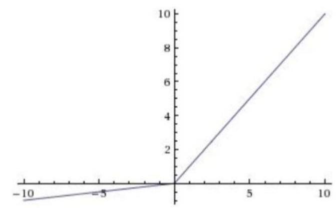
❖ ReLU와 변형된 형태

- **ReLU**



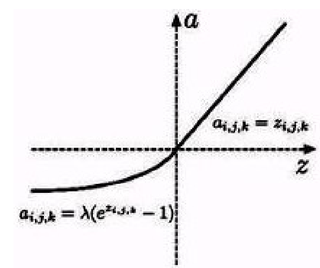
- **누수 ReLU(Leaky ReLU)**

$$f(x) = \max(\alpha x, x)$$



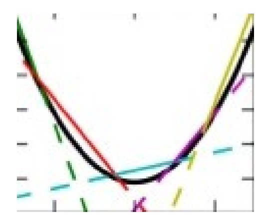
- **ELU (exponential Linear Unit)**

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(\exp(x) - 1) & \text{otherwise} \end{cases}$$



- **Maxout**

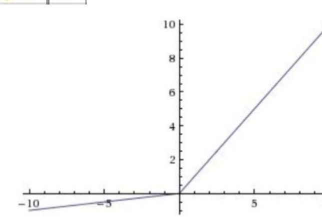
$$f(\boldsymbol{x}) = \max_{i \in \{1,\ldots,k\}} \{\boldsymbol{w}_i^\top \boldsymbol{x} + b_i\}$$



- **PReLU** (parameteric ReLU)

$$f(x) = \max(\alpha x, x)$$

α : 학습되는 파라미터



5.1.2 가중치 초기화

❖ 가중치 최기화

- 신경망의 성능에 큰 영향을 주는 요소
- 보통 가중치의 초기값으로 0에 가까운 무작위 값 사용

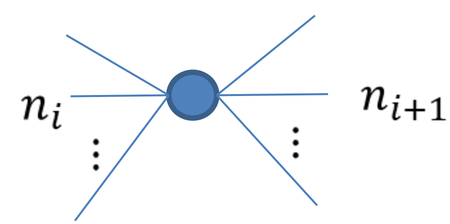
❖ 개선된 가중치 초기화 방법

- 각 노드의 입력 노드 개수 n_i와 출력 노드의 개수 n_{i+1}를 사용하는 방법
  - 균등 분포 $U\left[-\sqrt{\frac{6}{n_i+n_{i+1}}}, \sqrt{\frac{6}{n_i+n_{i+1}}}\right]$ 에서 무작위로 선택
  - 제이비어(Xavier) 초기화

    $\frac{N(0,1)}{\sqrt{n_i/2}}$에서 무작위로 선택

    $N(0,1)$: 표준 정규분포
  - 허(He) 초기화

    $\frac{N(0,1)}{\sqrt{n_i}}$에서 무작위로 선택

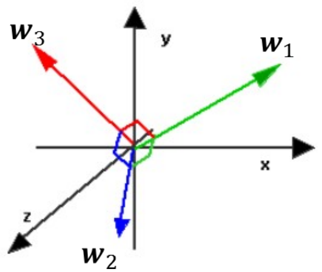


가중치 초기화

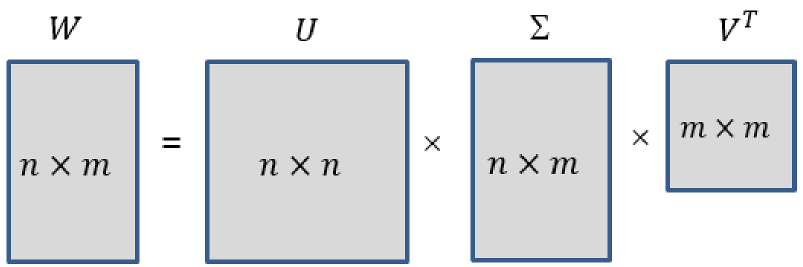
❖ 개선된 가중치 초기화 방법 – cont.

- 입력 데이터를 제한적 볼츠만 머신(Restricted Boltzmann Machine, RBM)을 학습시킨 결과의 가중치 사용

- 인접 층간의 가중치를 **직교 행렬**로 초기화
  - 가중치 행렬 W을 표준 정규분포 $N(0,1)$에서 무작위 추출하여 채움
  - 가중치 행렬 W을 특이값 분해(SVD)하여, 직교하는 벡터를 사용하여 가중치 초기화
    - **특이값 분해**
      - » 행렬 W를 $W = U\Sigma V^T$로 분해하는 행렬곱으로 표현 방법
      - » 여기에서 U, V는 각 열의 서로 직교하는 직교행렬



직교

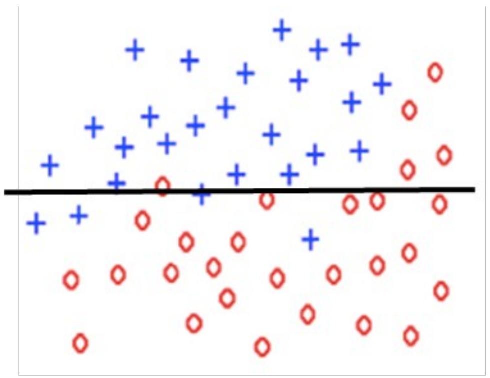


특이값 분해

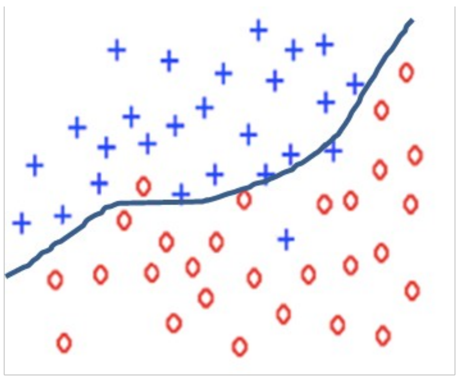
5.1.3 과적합 문제

❖ 과적합(Overfitting)

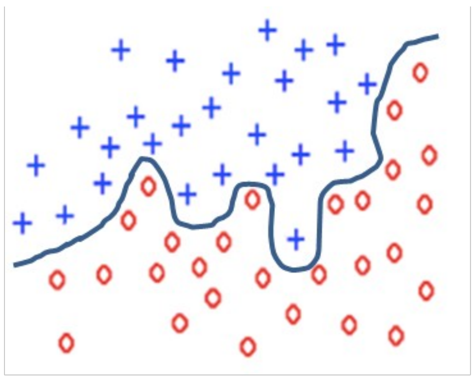
- 모델이 학습 데이터에 지나치게 맞추어진 상태
- 데이터에는 잡음이나 오류가 포함
  - 과적합된 모델은 학습되지 않는 데이터에 대해 성능 저하



부적합(underfitting)



정적합(good fitting)



과적합(overfitting)

- **과적합 문제 완화 기법**
  - 규제화
  - 드롭아웃
  - 배치 정규화

과적합 문제

❖ 규제화(Regularization) 기법

- 오차 함수를 오차(error) 항과 모델 복잡도(model complexity) 항으로 정의
  - 모델이 복잡해 지면 과적합이 될 수 있으므로, 모델 복잡도를 벌점(penalty) 항으로 추가

오차 함수 = (오차 항) + α (모델 복잡도 항)

α : 상대적인 반영비율을 조정

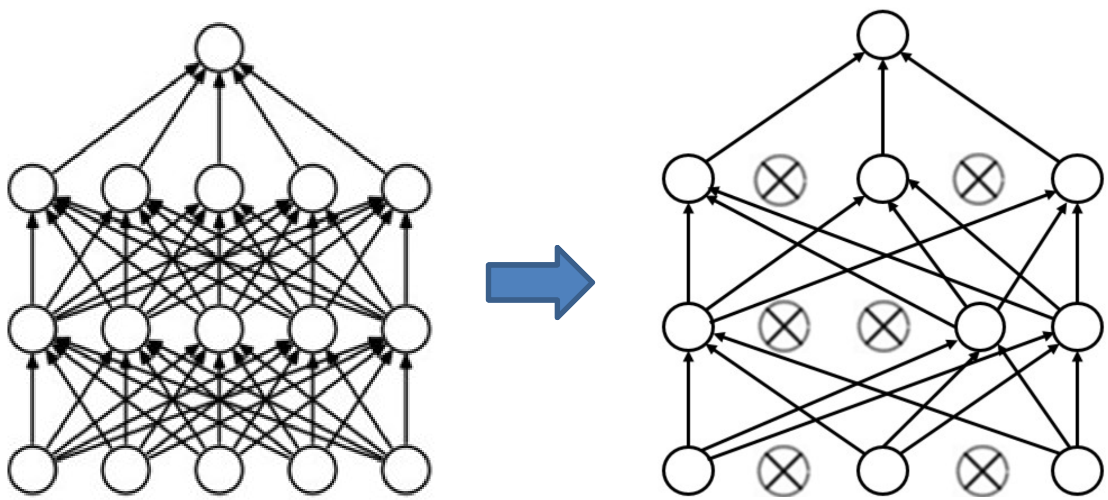
- **신경망 학습의 모델 복잡도 정의**
  - 절대값이 큰 가중치에 벌점 부여
  - 모델 복잡도 정의 $\boldsymbol{w} = (w_1, w_2, \cdots, w_n)$

$\sum_{i=1}^{n} w_i^2$ 또는 $\sum_{i=1}^{n} |w_i|$

과적합 문제

❖ 드롭아웃(Dropout) 기법

- 일정 확률로 노드들을 무작위로 선택하여, 선택된 노드의 앞뒤로 연결된 가중치 연결선은 없는 것으로 간주하고 학습
- 미니배치(mini-batch)나 학습주기(epoch) 마다 드롭아웃할 즉, 없는 것으로 간주할 노드들을 새롭게 선택하여 학습
- 추론을 할 때는 드롭아웃을 하지 않고 전체 학습된 신경망을 사용하여 출력 계산



과적합 문제

❖ 미니 배치(Minibatch)

- 전체 학습 데이터를 일정 크기로 나누어 놓은 것
- 학습 데이터가 큰 경우에는 미니배치 단위로 학습
- 경사 하강법(gradient-descent method) 적용시 **미니배치의 그레디언트**
  - 미니 배치에 속하는 각 데이터의 그레디언트의 평균 사용
    - 예. 10개 데이터로 구성된 미니배치의 그레디언드

$$\nabla g = \frac{1}{10}\sum_{i=1}^{10}\nabla g_i$$

- 미니 배치를 사용하여 데이터에 포함된 오류에 대해 둔감한 학습 가능
  - 과적합 문제 완화에 도움

과적합 문제

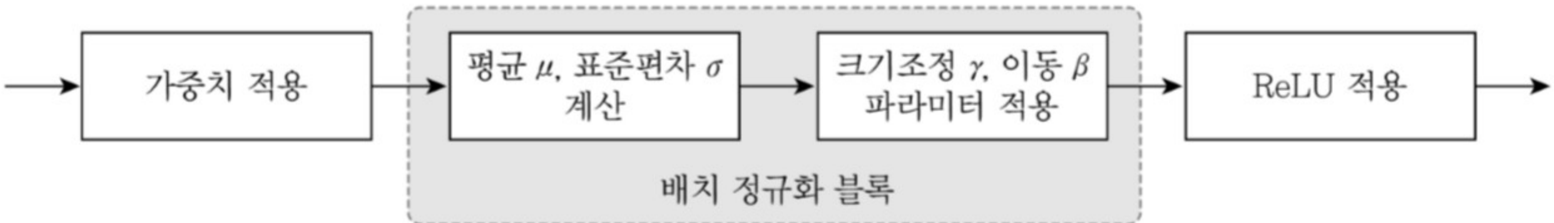
❖ 배치 정규화(Batch normalization) 기법

- **내부 공변량 이동(internal covariate shift) 문제**
  - 오차 역전파 알고리즘을 통한 학습
  - 이전 층들의 학습에 의해 이들 층의 가중치가 바뀌게 되면, 현재 층에 전달되는 데이터의 분포가 현재 층이 학습했던 시점의 분포와 차이 발생 ➔ 학습 속도 저하

- **배치 정규화**
  - 신경망의 각 층에서 미니배치 B의 각 데이터에 가중치 연산을 적용한 결과인 $\boldsymbol{x}_i$의 분포를 정규화하는 것
    1. $\boldsymbol{x}_i$의 평균 $\boldsymbol{\mu}_B$가 **0**이 되고 표준편차 $\boldsymbol{\sigma}_B$는 **I**가 되도록 변환
    2. 크기조정(scaling) 파라미터 γ와 이동(shift) 파라미터 β 적용
    3. 변환된 데이터 $\boldsymbol{y}_i$ 생성

과적합 문제

배치 정규화(Batch normalization) 기법



- 가중치 연산 결과의 미니 배치 : $B = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_m\}$
- 배치 정규화 적용 결과 : $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_m\}$

미니배치의 평균 : $\boldsymbol{\mu}_B = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i$

미니배치의 분산 : $\boldsymbol{\sigma}_B^2 = \frac{1}{m}\sum_{i-1}^{m}(\boldsymbol{x}_i - \boldsymbol{\mu}_B)^2$

정규화 : $\hat{\boldsymbol{x}}_i = \frac{\boldsymbol{x}_i - \boldsymbol{\mu}_B}{\sqrt{\boldsymbol{\sigma}_B^2 + \epsilon}}$

크기조정 및 이동변환 : $\boldsymbol{y}_i = \gamma\hat{\boldsymbol{x}}_i + \beta$