<퍼셉트론>

-다수의 신호를 입력으로 받아 하나의 신호를 출력.

-신호 : 전류나 강물처럼 흐름이 있는 것

-퍼셉트론 신호 : 흐른다(1), 안 흐른다(0)의 값만을 가질 수 있음

-그래프의 각 원을 ‘노드’ 또는 ‘뉴런’ 이라고 함. x1, x2가 입력신호, y가 출력신호 일 때, 각 edge의 w1, w2는 가중치임. 입력신호가 뉴런에 보내질 때 가중치가 곱해짐. 즉, w1x1, w2x2 이런식으로.

뉴런에서 보내온 신호의 총합이 정해진 한계를 넘어설 때 1을 출력함 = 뉴런 활성화, 이 한계 값을 임계값이라 하며 세타’θ’로 표현함, 단층 네트워크에서 이처럼 임계값을 경계로 출력이 바뀌는 활성화 함수를 계단 함수라고함

* y = 0 (w1x1+w2x2 <= θ) or 1 ( w1x1+w2x2 > θ)
* θ = -b 로 치환시

=> y = 0 (b + w1x1 + w2x2 <= 0) or 1 ( b + w1x1 + w2x2 > 0)

이때 b를 bias(편향) 이라고 함, 편향은 뉴런이 얼마나 쉽게 활성화 되는가 조절하는 매개변수

이때 b를 가중치라 생각하고 입력신호를 1이라고 생각하면 입력신호의 총합은

* b\*1 + x1\*w1 + x2\*w2 가 되고 이것이 0을 초과할 때 y = 1 이됨
* 이때 y의 결과값을 구분해주는 함수를 h(x)라 함 이때 h(x) = 0(x<=0) or 1(x>0)
* 즉, y=h(b\*1 + x1\*w1 + x2\*w2).
* h(x)처럼 입력 신호의 총합을 출력 신호로 변환하는 함수를 활성화 함수라고 함(activation function)
* 같은 식인데 다르게 써보자면, a= b+w1x1+w2x2 일때 y=h(a).

가중치가 클수록 그 신호가 더 중요함을 의미함. (저항의 반대느낌)

-<학습> : 적절한 매개변수를 정하는 작업이며, 기계학습에서는 컴퓨터가 자동으로 이 작업을 함. 사람은 퍼셉트론의 구조(모델)을 고민하고 컴퓨터에 학습할 데이터를 주는 일을 함.

-XOR 연산처럼 층이 여러개인 퍼셉트론을 다층 퍼셉트론이라 함

<신경망> : 가중치 매개변수의 적절한 값을 데이터로부터 자동으로 학습하는 능력이 신경망의 중요 성질

- 신경망은 입력층, 은닉층, 출력층으로 이루어지고 이때 은닉층은 사람의 눈에 보이지 않음

-계단 함수 : 임계값을 경계로 출력이 바뀌는 활성화 함수(h(x)). 즉, 퍼셉트론에서는 활성화 함수로 계단함수를 사용함

-시그모이드 함수 : 신경망에서 자주 사용하는 활성화 함수

=> h(x) = 1/ 1 + exp(-x) , 이때 exp(-x) = e^-x (e=2.7182…)

즉 퍼셉트론에서는 0 혹은 1만 흐르지만 뉴런에서는 연속적인 실수가 흐름

두 함수 모두 출력 값은 0이상 1이하임

최근에는 시그모이드 함수 대신 ‘ReLU’ 함수를 사용함. ReLU는 입력이 0을 넘으면 그대로 출력하고 0 이하면 0을 출력함.

=> h(x) = x (x >0) or x( x<=0)

※ 파이썬에서 np.dot() 은 행렬을 내적해주는 메서드이다. 이때 행렬 내적은 행렬끼리 곱하는 것을 말한다. 내적을 스칼라곱(scalar product) 또는 점곱(dot product)라고 한다.

※ 출력층의 활성화 함수는 σ(½Ã±×¸¶)·Î ³ªÅ¸³»¾î Àº´ÐÃþÀÇ È°¼ºÈ­ ÇÔ¼ö¿Í ´Ù¸£´Ù´Â °ÍÀ» ¸í½ÃÇÑ´Ù.

회귀에서는 σ를 항등함수, 2클래스 분류에는 σ를 sigmoid 함수, 다중 클래스 분류에서는 σ를 softMax 함수를 사용하는 것이 일반적이다.

- 신경망은 분류와 회귀 모두에 사용가능. 일반적으로 회귀에서는 항등함수를 분류에서는 softMax함수를 사용.

=> 분류 : 데이터가 어느 클래스에 속하느냐는 문제. Ex) 사진 속 인물의 성별을 분류하는 문제

=> 회귀 : 입력 데이터에서 (연속적인) 수치를 예측하는 문제. Ex) 사진 속 인물의 몸무게를 예측하는 문제. 회귀는 평균으로 회규하려는 경향에 기원하여 이름 붙혀짐(완두콩 이야기)

- softMax 함수 : yk = exp(ak)/ ∑(from i=1 to n) exp(ai). e는 자연상수를 말하고 n은 출력층의 뉴런 수, yk는 그중 k번째 출력임을 뜻함. 즉 소프트맥스 함수의 분자는 e^ak이고 분모는 모든 입력 신호의 지수함수의 합이다.

=> yk = exp(ak)/ ∑(from i=1 to n) exp(ai). = Cexp(ak)/ C∑(from i=1 to n) exp(ai).

= exp(ak + log C)/ ∑(from i=1 to n) exp(ai + log C).

= exp(ak+C’)/ ∑(from i=1 to n) exp(ai+C’).

=> 즉, softMax의 지수 함수를 계산할 때 어떤 정수를 더하거나 빼도 결과는 바뀌지 않는다.

=> C’에 어떤 값을 대입해도 되나, 오버플로를 막을 목적으로 입력 신호중 최댓값을 이용한다.

=> 이때 등장하는 E-05 같은 것은 10^-5 이라는 뜻이다.

=> softMax 함수의 출려은 0에서 1사이의 실수이고, 출력의 총 합은 1이다. 이 성질에 의해 softMax 함수의 출력을 ‘확률’로 해석할 수 있음

=> ex) 가령 [0.01821127 0.24519181 0.73659691] 의 출력 값이 나왔다면 각 원소는 각기 1.8%, 24.5%, 73.6%의 확률을 가짐. “마지막 원소가 확률이 가장 높으므로 답은 마지막 클래스다.” 라고 말할 수 있음. 또는 24.5%의 확률로 1번째 클래스다 라고 말할 수도 있음(첫 원소가 0번째 클래스)

=> 또한 softMax 함수를 적용해도 각 원소의 대소 관계는 변하지 않는다. 왜냐하면 지수 함수 y=exp(x)가 단조 증가 함수이기 때문이다. 단조 증가 함수란 정의역의 원소 a,b 가 a<=b일 때, f(a)<=f(b)가 성립하는 함수를 말한다. 신경망분류에서는 일반적으로 가장 큰 출력을 내는 뉴런에 해당하는 클래스로 인식하기 때문에, 입력신호들간의 대소관계가 변하지 않는 softMax 함수를 생략하여 사용하는 것이 일반적이다.

=> 학습 : 모델을 학습하는 것, softMax 함수를 사용함. 일반적으로 훈련 데이터로부터 가중치 매개변수의 최적값을 자동으로 획득하는 것을 뜻함

=> 추론 : 앞서 학습한 모델로 미지의 데이터에 대한 추론(분류)를 수행함. softMax 함수를 사용하지 않음.

- 출력층에 뉴런 수는 분류에서 분류하고 싶은 클래스 수로 설정하는 것이 일반적이다.

<손글씨 숫자 인식> : 추론 과정만 구현

- 순전파 : 추론 과정을 신경망의 순전파라고 한다.

- one-hot-encoding = 정답을 뜻하는 원소만 1(hot)이고 나머지는 0인 배열

-x.reshape(n,n) : 넘파이 배열의 형상을 n x n 으로 바꾸어줌.

-정규화(normalization) : 데이터를 특정 범위로 변환하는 처리. softMax함수를 거치면 마지막 출력값이 정규화 되어 나온다. 또한 정규화 되지 않는 출력결과는 즉, softMax함수 앞에 affine계층의 출력 결과를 점수라고 한다.

-전처리(pre-procession) : 신경망의 입력 데이터에 특정 변환을 가하는 것. 전처리를 하면 정규화가 가 될 수도 있다

-백색화(whitening) : 전체 데이터를 균일하게 분포시키는 것.

-배치(batch) : 하나로 묶은 입력 데이터

- 파이썬의 np.argmax(리스트,axis= 0 or 1)는 해당 리스트의 최댓값의 인덱스를 가져온다. 이때 axis는 어느 차원으로 계산을 할지 정하는 인수로 axis=0 이면 열연산이고 axis=1이면 행연산이다. 즉 열연산은 열끼리 비교하는 것이고 행연산은 행끼리 비교하는 것이다.

<ch4. 신경망 학습>

-손실 함수 : 신경망이 학습할 수 있도록 해주는 지표, 손실 함수의 결과값을 가장 작게 만드는 가중치 매개변수를 찾는 것이 학습의 목표. 손실 함수의 값을 가급적 작게 만드는 기법으로 함수의 기울기를 활용하는 경사법이 있음. 손실함수는 나쁨을 나타내는 지표이고, 이 지표의 부호를 바꾸면 얼마나 좋은지에 대한 지표가 된다.

- 특징 feature: 입력 데이터에서 본질적인 데이터(중요 데이터)를 정확하게 추출할 수 있도록 설계된 변환기를 가리킨다. 이미지의 특징은 보통 벡터로 기술하고, 컴퓨터 비전 분야에서는 SIFT, SURF, HOG 등의 특징을 많이 사용한다. 이렇게 변환된 벡터를 가지고 지도 학습 방식의 대표 분류 기법인 SVM, KNN 등으로 학습할 수 있다.

※ 딥러닝은 종단간 기계학습이라고도 하는데, 이는 처음부터 끝까지라는 의미로, 데이터(입력)에서 목표한 결과(출력)을 얻는다는 뜻이다. 신경망은 사람의 개입이 다른 기계학습에 비해 현저히 적은데, 이로인해 모든 문제를 같은 맥락에서 풀 수 있다. 즉 end to end (종단간) 학습이 가능하다.

- 훈련 데이터(training data) : 학습을 할 때 사용하는 데이터로 최적의 매개변수를 찾기 위함이다.

- 시험 데이터(test data) : 앞서 훈련한 모델의 실력을 평가하기 위함이다.

=> 이렇게 데이터를 나누는 이유는 범용적으로 사용할 수 있는 모델을 설계하기 위해서이다.

=> 범용능력이란 아직 보지 못한 데이터(훈련 데이터에 포함되지 않은 데이터)로도 문제를 올바르게 풀어내는 능력이다. 이것이 기계학습의 최종 목표라고 할 수 있다.

- 오버피팅(overfitting) : 한 데이터셋에 지나치게 최적화된 상태를 말한다. 기계학습에서는 오버피팅을 피해야 한다.

신경망 학습에서는 현재의 상태를 하나의 지표로 표현하고 이 지표를 가장 좋게 만들어주는 죄적의 가중치 매개변수를 찾는 것이 목표이다. 이때 사용하는 지표가 손실 함수(loss function)이다.

=> 손실 함수는 일반적으로 평균 제곱 오차와 교차 엔트로피 오차를 사용한다.

-평균 제곱 오차(mean squared error,MSE) : 주로 사용되는 손실함수 중 하나이다.

=> E = 1/2∑(from k) (yk-tk)^2 이때 yk는 신경망의 출력(신경망이 추정한 값), tk는 정답 레이블, k 는 데이터의 차원 수를 나타낸다. 이때 정답레이블은 one-hot-encoding이다.

=> MSE의 결과값이 작을수록 정답에 근접하다.

-교차 엔트로피 오차(cross entopy error, CEE) : 또 다른 자주사용되는 손실 함수이다

=> E = -∑(from k) tk \* log(yk) 이때 log의 밑은 자연상수 e이고. yk 는 신경망의 출력이며 tk는 정답 레이블이다. 이때 정답레이블은 one-hot-encoding이다. 그러므로 실질적으로 정답일 때의 추정의 자연로그를 계산하는 식이다. 왜냐하면 정답레이블에서 정답인 것 만 1이고, 나머지는 0이므로 0과 곱을 해봤자 0이기 때문이다. 즉 정답일 때의 출력이 전체 값을 정하게 된다.

=> CEE의 결과값이 작을수록 정답에 근접하다.

4.2.3 미니배치 학습

훈련 데이터가 N개일때의 교차 엔트로피 오차 CEE의 식은 다음과 같다.

=> E = -(1/N)∑(from n)∑(from k) tnk \* log(ynk) ÀÌ¶§ tnk´Â n¹øÂ° µ¥ÀÌÅÍÀÇ kÂ÷¿ø Â° °ªÀ» ÀÇ¹ÌÇÑ´Ù. NÀ¸·Î ³ª´®À¸·Î½á ‘Æò±Õ ¼Õ½Ç ÇÔ¼ö’¸¦ ±¸ÇÒ ¼ö ÀÖ´Ù. Æò±ÕÀ» ±¸ÇØ »ç¿ëÇÏ¸é ÈÆ·Ã µ¥ÀÌÅÍ °³¼ö¿Í °ü°è¾øÀÌ ¾ðÁ¦µç ÅëÀÏµÈ ÁöÇ¥¸¦ ¾òÀ» ¼ö ÀÖ´Ù.

- ¹Ì´Ï¹èÄ¡(mini-batch) : ºòµ¥ÀÌÅÍÀÇ ¾çÀº ³Ê¹« ¸¹¾Æ¼­ ¼Õ½Ç ÇÔ¼öÀÇ ÇÕÀ» ±¸ÇÏ·Á¸é ½Ã°£ÀÌ ¿À·¡°É¸°´Ù. ±×·¯¹Ç·Î µ¥ÀÌÅÍ¸¦ ÀÏºÎ Ãß·Á ÀüÃ¼ÀÇ ±Ù»çÄ¡·Î »ç¿ëÇÑ´Ù. Áï ÈÆ·Ã µ¥ÀÌÅÍ·ÎºÎÅÍ ÀÏºÎ¸¸ °ñ¶ó ÇÐ½ÀÀ» ¼öÇàÇÏ´Âµ¥ ÀÌ¸¦ ¹Ì´Ï¹èÄ¡, ¹Ì´Ï¹èÄ¡ ÇÐ½ÀÀÌ¶ó°í ÇÑ´Ù.

- np.random.choice(train\_size,batch\_size) : train\_size개의 원소 중에서 batch\_size 만큼 랜덤으로 골라냄, seed가 매번 달라짐, 중복이 허용됨, train\_size로 정수가 입력되면 0부터 그 정수까지 임의의 값을 골라냄

¼Õ½ÇÇÔ¼öÀÇ ¹ÌºÐ°ªÀ» ÀÌ¿ëÇÏ¿© °¡ÁßÄ¡ ¸Å°³º¯¼ö¸¦ ÃÖÀûÀÇ °ªÀ¸·Î °»½ÅÇÒ ¼ö ÀÖÀ½, ¹ÌºÐ °ªÀÌ Áï, ±â¿ï±â°¡ À½¼öÀÌ¸é ¾ç¼ö ¹æÇâÀ¸·Î, ¾ç¼ö¸é À½ÀÇ ¹æÇâÀ¸·Î º¯È­½ÃÄÑ ±â¿ï±â°¡ 0ÀÌ µÇµµ·Ï ÇÏ´Â °ÍÀÌ ¸ñÇ¥ÀÌ´Ù. Á¤È®µµ¸¦ ÁöÇ¥·Î ÇÏ¸é ¸Å°³º¯¼öÀÇ ¹ÌºÐÀÌ ´ëºÎºÐÀÇ Àå¼Ò¿¡¼­ 0ÀÌ µÇ±â ¶§¹®¿¡ °»½ÅÇÒ ¼ö ¾øÀ¸¹Ç·Î »ç¿ëÇÏ¸é ¾ÈµÈ´Ù.

½Ã±×¸ðÀÌµå ÇÔ¼ö´Â ¾î´À °÷¿¡¼­µµ ±â¿ï±â°¡ 0ÀÌ µÇÁö ¾Ê±â ¶§¹®¿¡ ÃÖÀûÀÇ °ªÀ» ÇâÇØ ¸Å°³º¯¼ö °ªÀ» Áö¼ÓÀûÀ¸·Î °»½ÅÇÒ ¼ö ÀÖ´Ù.

ÄÄÇ»ÅÍ¿¡¼­´Â ¹ÌºÐÀÌ h=10^-4 Á¤µµ¸¦ »ç¿ëÇÑ´Ù. ÀÌ Á¤µµÀÇ °ªÀ» »ç¿ëÇÏ¸é ÁÁÀº °á°ú¸¦ ¾ò´Â´Ù°í ¾Ë·ÁÁ® ÀÖ´Ù.

- Â÷ºÐ : µÎ Á¡¿¡¼­ ÇÔ¼ö °ªµéÀÇ Â÷ÀÌ¸¦ ¸»ÇÑ´Ù.

=> Àü¹æ Â÷ºÐ : (x+h)¿Í xÀÇ Â÷ºÐÀ» ¸»ÇÑ´Ù.

=> Áß¾Ó Â÷ºÐ(Áß½É Â÷ºÐ) : x¸¦ Áß½ÉÀ¸·Î ÀüÈÄÀÇ Â÷ºÐÀ» °è»êÇÑ´Ù. Áï, (x+h)¿Í (x-h)ÀÏ¶§ÀÇ fÀÇ Â÷ºÐÀ» °è»êÇÑ´Ù. Àü¹æ Â÷ºÐÀÇ ÄÄÇ»ÅÍ»óÀÇ ¿ÀÂ÷¸¦ ÁÙÀÌ±â À§ÇØ »ç¿ëÇÑ´Ù.

-¼öÄ¡ ¹ÌºÐ : ¾ÆÁÖ ÀÛÀº Â÷ºÐÀ¸·Î ¹ÌºÐÀ» ±¸ÇÏ´Â °ÍÀ» ¸»ÇÑ´Ù.

=> ÇØ¼®Àû(analytic) : ¼ö½ÄÀ» Àü°³ÇØ ¹ÌºÐÀ» ±¸ÇÏ´Â °ÍÀ» ¸»ÇÑ´Ù. ÇØ¼®Àû ÇØ ¶Ç´Â ÇØ¼®ÀûÀ¸·Î ¹ÌºÐÀ» ±¸ÇÏ´Ù µîÀ¸·Î Ç¥ÇöÇÑ´Ù. ÇØ¼®Àû ¹ÌºÐÀº ¿ÀÂ÷¸¦ Æ÷ÇÔÇÏÁö ¾Ê´Â ‘ÁøÁ¤ÇÑ ¹ÌºÐ’ °ªÀ» ±¸ÇØÁØ´Ù.

-Æí¹ÌºÐ : º¯¼ö°¡ ¿©·¯ °³ÀÎ ÇÔ¼ö¿¡ ´ëÇÑ ¹ÌºÐÀ» ¸»ÇÑ´Ù. Æí¹ÌºÐ½Ã ¹ÌºÐÇÒ º¯¼ö´Â ¹ÌºÐÇÏ°í ³ª¸ÓÁö º¯수´Â »ó¼ö Ãë±ÞÇÑ´Ù. ÆÄÀÌ½ã¿¡¼­ Ç®¶§´Â º¯¼ö°¡ ÇÏ³ªÀÎ ÇÔ¼ö¸¦ Á¤ÀÇÇÏ°í ±× ÇÔ¼ö¿¡ ´ëÇÑ ¹ÌºÐÀ» ±¸ÇÏ´Â ÇüÅÂ·Î ±¸ÇöÇÒ ¼ö ÀÖ´Ù.

-기울기 : 변수가 2개 이상일 때 모든 변수의 편미분을 벡터로 정리한 것을 기울기라고 한다.

-경사법(경사 하강법) : 손실 함수가 최소값이 되는 최적의 매개변수를 학습시 찾아야 되는데 이를 알아내기가 쉽지 않다. 그때 사용하는 방법이 기울기를 활용한 경사법이다.

=> 학습률 : 매개변수의 값을 갱신하는 양이다.

=> x0 = x0 – lr(df/dx0) , x1 = x1 – lr(df/dx1) 이때 lr은 학습률(learning rate)를 말한다.

-하이퍼 파라미터(초매개변수) : 학습률 같은 매개변수를 말한다. 이는 가중치와 편향 같은 신경망의 매개변수와는 성질이 다르다. 신경망 매개변수는 자동으로 획득되지만, 초매개변수는 사람이 직접 설정해야 한다.

※ np.random.randn(2,3) = 2 x 3 행렬을 랜덤 정규분포로 초기화

<신경망 학습의 절차 요약>

전제 - 신경망에는 적응 가능한 가중치와 편향이 있고, 이 가중치와 편향을 훈련 데이터에 적응하도록 조정하는 과정을 ‘학습’이라고 한다.

1. 미니배치 : 훈련 데이터중 일부를 무작위로 가져오는 것을 말하며, 이 미니배치의 손실 함수 값을 줄이는 것을 목표로 한다.

2. 기울기 산출 : 미니배치의 손실 함수 값을 줄이기 위해 각 가중치 매개변수의 기울기를 구한다. 기울기는 손실 함수의 값을 가장 작게 하는 방향을 제시한다.

3. 매개변수 갱신 : 가중치 매개변수를 기울기 방향으로 아주 조금 갱신한다.

4. 반복 : 1~3단계를 반복한다.

=> 미니배치로 데이터를 무작위 선정하는 방식의 경사하강법을 확률적 경사하강법(stochastic gradient descent, SGD) 이라고한다.

- 에폭 : 하나의 단위로서, 1에폭은 학습에서 훈련 데이터를 모두 소진했을 때의 횟수에 해당한다. 훈련데이터 10000개를 100개의 미니배치로 학슬할 경우 확률적 경사 하강법을 100회 반복시 모든 훈련 데이터를 소진하게 된다. 이때 100회가 1에폭이다.

-순전파 : 계산 그래프에서 계산을 왼쪽에서 오른쪽으로 진행하는 단계를 말한다.

-역전파 : 계산 그래프에서 계산을 순전파 반대항향으로 하는 것을 말한다. 미분 계산시 중요한 역할을 한다.

- 연쇄 법칙 : 합성 함수의 미분에 대한 성질이며 다음과 같이 정의된다. 합성 함수의 미분은 합성 함수를 구성하는 각 함수의 미분의 곱으로 나타낼 수 있다.

=> z= (x+y)^2 에서 t=x+y로 치환시 z=t^2가 된다. 이때 z를 x에 대해 미분시 dz/dt \* dt/dx 가 된다. 이것은 2t\*1 =2(x+y)가 된다.

- 덧셈 노드의 역전파 : 입력 값을 그대로 흘려보낸다. 즉 입력값과 출력값이 같다.

- 곱셈 노드의 역전파 : 상류의 값에(오른쪽에서 흘러들어온 값에) 순전파 때의 입력 신호들을 서로 바꾼 값을 곱해서 하류(왼쪽으로)로 보낸다.

※ mask = self.mask 처럼 사용하는 mask는 True/False로 구성된 넘파이 배열이다. ReLU계층 구현에서 forward 함수에서는 순전파 입력 x의 원소값이 0이하인 인덱스는 True, 그 외는 False로 유지한다

- 뉴런의 가중치 합은 Y= np.dot( X,W) + B로 계산한다. 이때 X=입력, W=가중치, B=편향이다.

-affine transformation(어파인 변환) : 신경망의 순전파 때 수행하는 행렬의 내적을 기하학에서 어파인 변환이라고 한다. 어파인 변환을 수행하는 처리를 affine 계층이라고 한다.

-신경망을 추론할때는 softMax함수를 사용하여 정규화 하지 않는다. 추론 과정에서는 마지막 affine계층의 출력값 중 가장 점수가 높은 것을 찾으면 되기 때문이다. 허나 학습에서는 softMax함수가 사용된다.

-softMax 함수는 결과를 0과 1사이로 정규화하여 확률로 나타내기 위함이다. 분류에서 주로 사용된다.

-분류할 때 softMax하무의 손실함수로 교차 엔트로피 오차를 이용하면 역전파가 (y1-t1, y2-t2, y3-t3) 로 말끌히 떨어진다. 또한 회귀에서 항등함수의 손실하수로 평균 제곱 오차를 이용해도 (y1-t1, y2-t2, y3-t3)로 떨어진다.

<신경망 학습의 전체적인 순서>

- 전제 : 신경망에는 적응 가능한 가중치와 편향이 있고, 이 가중치와 편향을 훈련 데이터에 적응하도록 조정하는 과정을 ‘학습’이라 한다. 신경망 학습은 다음과 같은 4단계로 수행한다.

- 1단계 미니배치 : 훈련 데이터 중 일부를 무작위로 가져온다. 이렇게 선별한 데이터를 미니배치라 하며, 그 미니배치의 손실 함수 값을 줄이는 것을 목표로한다.

- 2단계 기울기 산출 : 미니배치의 손실 함수 값을 줄이기 위해 각 가중치 매개변수의 기울기를 구한다. 기울기는 손실 함수의 값을 가장 작게 하는 방향을 제시한다.

- 3단계 매개변수 갱신 : 가중치 매개변수를 기울기 방향으로 아주 조금 갱신한다.

- 4단계 반복 : 1~3단계를 반복한다.

[CNN] – Convolutional neural network, 합성곡 신경망

-완전연결은 신경망이 인접계층의 모든 뉴런과 결합되어 있는 것, 이를 Affine계층 이라고 함.

=> 완전연결 계층의 네트워크는

[Affine,Relu(sigmoid)] - > [Affine,Relu(sigmoid)] -> … -> [Affine,Relu(sigmoid)]

-> [Affine,SoftMax] 처럼 구성됨.

=> CNN구조는

[Conv,Relu,(Pooling)] -> [Conv,Relu,(Pooling)] -> … -> [Conv,Relu,(Pooling)]

-> [Affine,Relu(sigmoid)] -> [Affine,SoftMax] 으로 구성됨.

Pooling 계층은 생략되기도 함.

=> <완전연결 계층의 문제점>

- 데이터의 형상이 무시됨. 왜냐하면 3차원데이터를 1차원데이터로 표현해야 하기 때문이다.

반면 합성곱 계층은 형상을 유지한다. 그래서 이미지처럼 형상을 가진 데이터를 제대로 이해할 (가능성이) 있음.

- <특징 맵> : CNN에서의 합성곱계층의 입출력데이터를 특징맵 이라함. 입력 특징맵, 출력 특징맵등으로 말함.

=> <합성곱 연산> - 이미지처리에서의 필터연산에 해당함.

입력데이터 4x4 와 필터 3x3이 있으면 필터의 윈도우를 입력데이터에 덧댄후 각 쉘의 값을 곱하여 더한다음 윈도우를 다음 칸으로 옮겨서 만듬. 위의 경우에는 2x2의 출력데이터가 생성됨. 이를 단일 곱셈-누산 (fused multiply-add, FMA)라고 함. 이때 필터의 매개변수가 가중치에 해당함.

- 편향은 1x1 즉 하나만 존재하고, 필터를 적용한 후에 각 쉘에 더해짐.

=> <패딩>