

TCT-기술인증테스트

AI

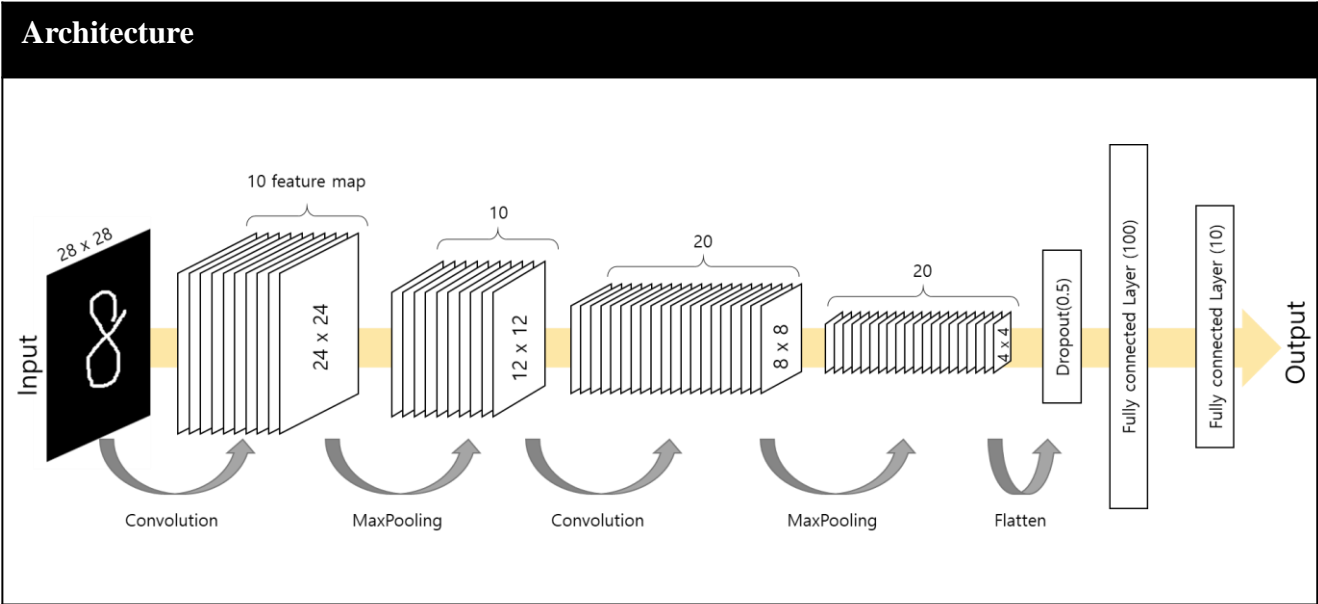
실기형 문제지

[2020년 #차]

사번	성명
유의 사항	<ul style="list-style-type: none">▪ 본 실기형 문제지에 제시된 시나리오와 문제를 읽고, 별도 배포되는 답안지에 응시자의 답안을 작성해 주시기 바랍니다.▪ 배포된 문제지는 시험 종료 후 답안지와 함께 감독관에게 제출해 주시기 바랍니다.▪ 공정한 평가를 위하여 답안 작성 시 동료를 도와 주는 행위, 보여주는 행위를 금지하고 있으며, 평가에 대한 부정행위 적발 시, 응시한 평가는 0점 처리됩니다.

[실기형 1번] TensorFlow를 이용한 MNIST 분류 모델 구현 (4점)

아래 그림의 아키텍처를 반영하는 TensorFlow 모델을 구현하시오.



Code

[실기형 2번] TensorFlow 모델 코드 수정 및 학습하기 (4점)

실습 환경에 주어진 코드는 학습이 정상적으로 진행되지 않는 코드이다. 학습이 정상적으로 이루어질 수 있도록 코드를 수정하고 학습 및 평가(evaluate) 결과를 출력하시오.

단, 정상적이라는 것은 테스트 데이터셋(test dataset, 코드에서 변수명 `test_norm` 및 `test_y`에 해당)에 대해 평가(evaluate) 했을 때 accuracy가 0.9 이상의 결과가 나오는 상태를 뜻한다.

[실기형 3번] Learning rate 조절하기 (6점)

철수는 손글씨 숫자 이미지 데이터인 MNIST 데이터셋을 입력 받아 분류하는 심층 신경망(Deep Neural Network)을 만들고자 한다. Fully Connected(FC) Layer만을 활용하여 간단하게 신경망을 구성하고, Stochastic Gradient Descent를 활용해 파라미터를 업데이트하려고 한다. 단, Learning rate를 아래의 표와 같이 조절하여 학습 초반에는 빠르게 파라미터가 업데이트 되도록 하고, 학습 후반부에는 미세한 조정을 통해 안정적으로 파라미터가 수렴할 수 있도록 유도하는 효과를 얻고자 한다.

TensorFlow의 'LearningRateScheduler()'를 활용하여 아래 조건을 구현하고, callback method로 적용하시오.

[조건]

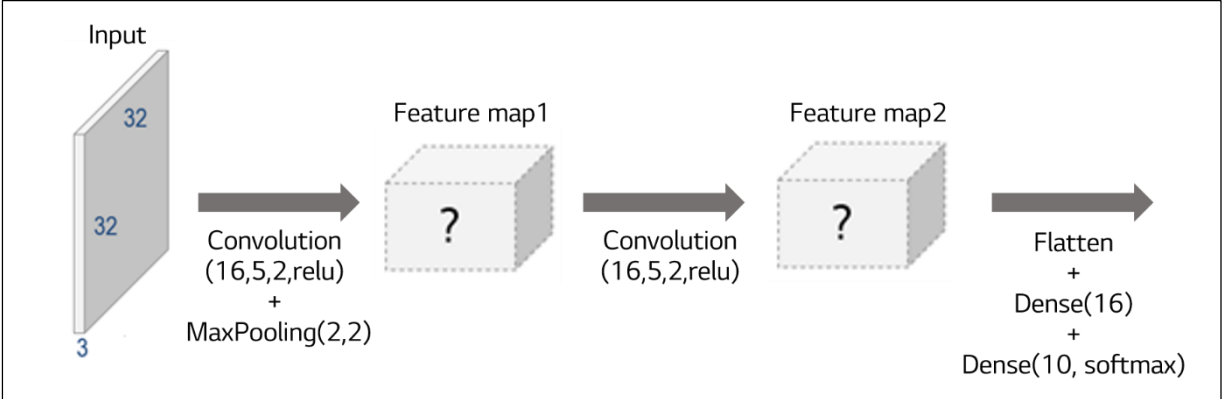
- MNIST 이미지를 입력으로 받아 50개의 뉴런을 가지며 ReLU를 이용하는 FC layer 1층, 10개의 숫자로 분류하도록 softmax를 이용하는 FC layer 1층을 구성한다.
- Cross entropy loss function을 사용한다.
- 기본 learning rate 값은 0.1이다.
- 첫 10 epoch동안 learning rate은 기본 learning rate 값에 epoch*0.1만큼을 추가로 더하여 적용한다.
- 이후 15 epoch동안 learning rate은 1.0에서 epoch*0.04만큼을 줄여나가며 적용한다.

[0~24 epoch까지 적용되는 learning rate]

epoch	learning rate	epoch	learning rate	epoch	learning rate
0	0.10	10	0.60	20	0.20
1	0.20	11	0.56	21	0.16
2	0.30	12	0.52	22	0.12
3	0.40	13	0.48	23	0.08
4	0.50	14	0.44	24	0.04
5	0.60	15	0.40		
6	0.70	16	0.36		
7	0.80	17	0.32		
8	0.90	18	0.28		
9	1.00	19	0.24		

[실기형 4번] CNN(Convolutional Neural Network) 모델 변경하기 (6점)

철수는 아래 [그림]과 같은 CNN(Convolutional Neural Network) 모델을 구성하였다.



[그림] 철수가 구성한 CNN 모델

[그림]에서 Convolution 연산 및 MaxPooling 연산, Flatten layer, Dense layer와 관련된 하이퍼파라미터는 아래와 같고, 위 모델의 전체 파라미터 수는 73,354개이다.

Convolution(a, b, c, d) : a = 필터(filter) 수, b = 필터의 W(width)와 H(height) (W, H는 동일).
c = padding size, d = activation function
MaxPooling(e, f) : e = pooling size, f = stride
Flatten : input을 flatten하는 layer. (tensorflow.keras.layers.Flatten 에 해당)
Dense(g, h) : g = Densely-connected layer의 units 수, h = (존재한다면) activation function

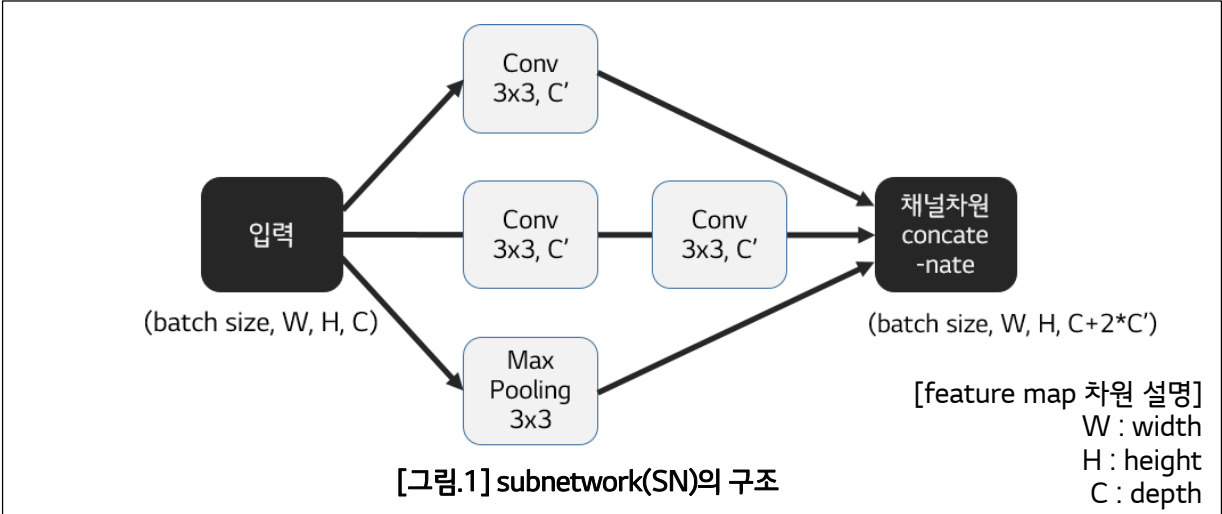
[문제4-1] TensorFlow를 이용하여 [그림]에 주어진 모델을 구현하시오. (3점)

[문제4-2] [그림]의 모델과 연산 및 레이어(layer) 구성은 동일하지만 아래의 조건을 만족하는 모델을 TensorFlow를 이용하여 구현하시오. 학습 데이터는 5천 개를 사용하고 5 epochs을 학습한다. (코드 참고) (3점)

- 모델의 전체 파라미터 수 $\leq 35,000$ 개
- 5 epochs 학습 후 학습 데이터에 대한 accuracy ≥ 0.5
- Convolution 및 MaxPooling 연산, Dense layer의 하이퍼파라미터는 변경할 수 있음.

[실기형 5번] TensorFlow 모델 구현하기 (7점)

철수는 CNN(Convolutional Neural Network)의 두 주요 연산인 convolution 연산과 pooling 연산을 병렬적으로 수행하는 subnetwork(이하 SN)를 만들었고, 구조는 [그림.1]과 같다.



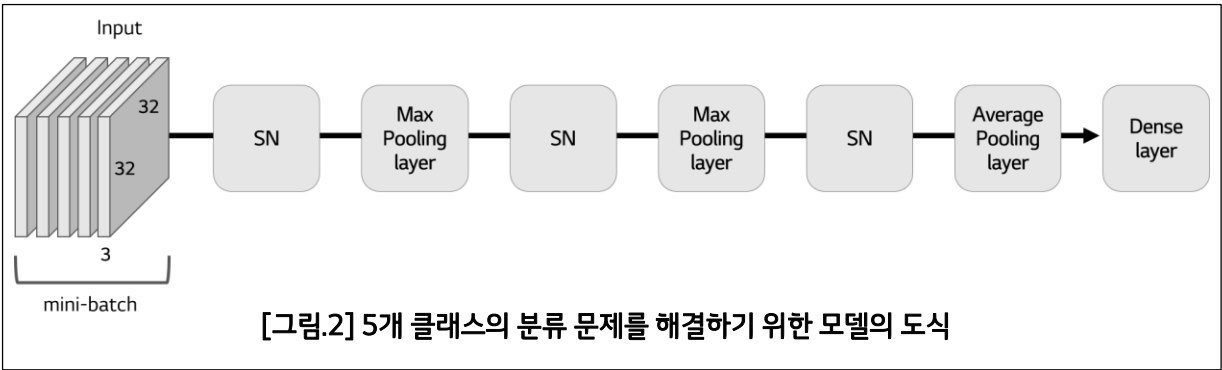
[그림.1]의 연산에 대한 설명

- Conv 3x3, C' : filter의 W(width)와 H(height)가 각각 3이고 필터 수가 C'인 convolution 연산
- MaxPooling 3x3 : pooling size가 3x3인 MaxPooling 연산

철수는 [그림.1]의 SN을 이용하여 아래 [그림.2]와 같은 모델을 만들었다. 이 모델을 이용하여 32x32x3(Width x Height x Channel)의 컬러 이미지의 mini-batch를 5개 클래스(class)로 분류(classification)하는 문제를 해결하려고 한다. 아래 주어진 [제약 조건 및 요구사항]을 반드시 충족하는 [그림.2]의 모델을 TensorFlow로 구현하시오.

[제약 조건 및 요구사항]

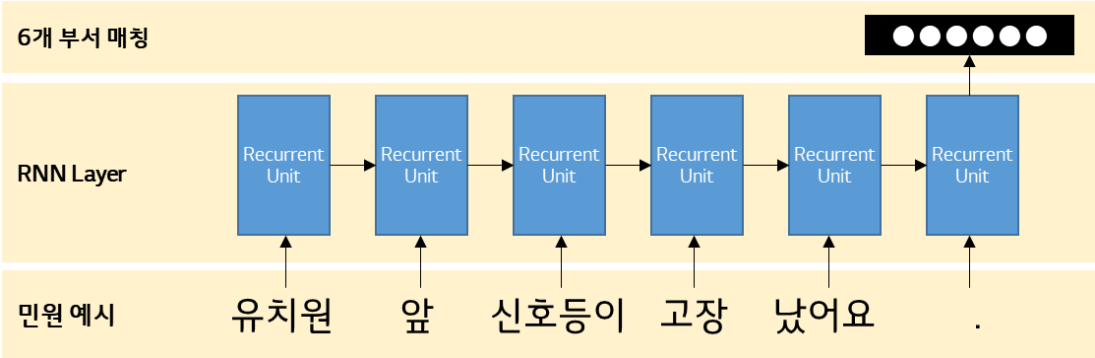
- SN에 대해 $2 < C' < 10$ 를 만족한다.
- [그림.2]의 MaxPooling layer는 feature map의 width와 height를 $\frac{1}{2}$ 로 줄인다.
- [그림.2]의 AveragePooling layer 직후의 feature map의 width와 height는 모두 1이다.
- Dense layer의 activation function은 softmax 이다.



[실기형 6번] 민원 자동 분류 (7점)

철수는 민원이 들어오면 민원 내용에 따라 관련 부서로 민원을 넘겨주는 업무를 자동화하려고 한다. 민원 자동 분류 시스템 구축을 위해 작년 한 해 누적된 민원과 담당 부서를 매핑한 10만 건의 데이터를 활용할 수 있다. 철수는 텍스트 데이터인 민원 내용을 처리하기 위해 RNN 계열의 알고리즘을 활용하여 모델을 만들고자 한다. 민원 내용은 길이가 제각각이며, 민원을 담당하는 부서는 6개이다.

아래 조건을 고려하여 TensorFlow를 활용한 코드를 작성하시오.



- [조건]
- Input으로 들어올 수 있는 단어는 한국어 단어 100만 개를 대상으로 하며, 여기에 추가로 padding을 위한 '[PAD]'와 vocabulary 외의 단어를 처리하는 '[OOV]'를 포함하여 vocabulary를 구축한다.
 - Vocabulary의 각 단어는 길이 32의 벡터로 임베딩하여 사용한다.
 - Output layer는 Fully-connected layer를 활용하여 6개 부서 분류를 할 수 있도록 softmax를 사용한다.

[문제6-1] 단순 Recurrent Unit을 활용하였더니 긴 문장의 민원을 잘 학습하지 못하여 성능이 좋지 못했다. 이러한 문제점을 보완할 수 있는 LSTM Unit을 활용하여 민원을 분류하는 모델을 구현하시오. 단, LSTM의 unit size는 50으로 한다. (3점)

[문제6-2] [문제6-1]를 통해 모델의 성능이 개선되었으나, 과소적합(Underfitting)의 경향이 보여 동일한 unit size를 가진 LSTM 레이어(layer)를 한 층 더 쌓기로 한다. 또한 너무 긴 민원을 처리하는 데에 시간이 오래 걸리기 때문에 최대 민원 문장의 길이를 25 단어로 제한하려 하는데, 민원 데이터 특성상 문장의 앞부분에 불만을 표출하는 경우가 대부분이어서 25단어 이상의 민원이 들어오는 경우 문장의 뒷부분을 잘라내려고 한다. 이를 반영하여 모델을 구현하시오. (4점)

