|  |  |
| --- | --- |
| 교육 제목 | NN-MNIST |
| 교육 일시 | 10월 27일(수) |
| 교육 장소 | 재택 |
| **교육 내용** | |
|  | * 딥러닝의 대략적 흐름  1. 데이터 불러오기 2. 데이터 확인하기 3. X,Y 나누기(train, test set) 4. 전체 데이터에서 학습데이터와 테스터 데이터(0.2) 구분 5. 딥러닝 설계 🡪 kernel\_initializer = tf.keras.initializer.he\_normal(),   Tf.keras.initializers.glorot\_uniform()   1. Model.compile 2. Model.fit 3. 그래프 결과 확인 4. Test 정확도 확인  * NN\_MNIST   MNIST 문제에 있어서 은닉층이 하나라면 뉴럴 네트워크는 입력층, 은닉층, 출력층으로 구성이 가능하다. 입력층과 은닉층 사이에는 활성화 함수가 있어야 하며 출력층은 그 자제가 활성화 함수인 Softmax 가 사용되므로 은닉층과 출력층 사이에 별도의 활성화 함수를 넣을 필요는 없다. 입력층 다음에 넣게 될 활성화 함수로는 Relu 가 적당하다.  뉴럴 네트워크에서 하나의 은닉층을 도입할 경우 Depth 가 1 이 며 Wide 한 폭 즉 Width를 얼마로 잡느냐에 따라서 인식률이 민감하게 반응하게 된다. MNSIT 문제에서 출력층에서의 분류되는 클라스 수는 0∼9까지의 수에 해당하는 10으로 정해져 있다. Width 의 범위는 10 이상으로서 unlimited 라 볼 수 있으나 현실적으로 CPU 및 GPU를 사용하여 연산을 실행해야 하므로 입력 데이터 784를 기준으로 10∼900 의 범위를 추천한다. 784를 넘어가게 되면 컴퓨팅 부담이 늘어나지만 가장 높은 인식률을 얻을 수 있다는 점을 미리 지적해 둔다.  아울러 Wide 한계를 900으로 두고 deep=3 즉 3개의 은닉층을 구성하여 실행해 본 결과 은닉층 하나만 사용할 때와 별반 차이가 없이98% 수준의 인식률이 얻어진다.  Gradient Descent Optimizer 나 Adam Optimizer를 사용하느냐에 따라 learning rate 값은 0.1 아니면 0.001 이 된다. Adam Optimizer 가 학습 횟수가 늘어남에 따라 능동적으로 learning rate를 조절하는 기능이 있으나 MNIST 문제에서는 Optimizer 선택에 따른 결과 영향은 미미한 듯하다. |
| 전체 내용 |