|  |  |
| --- | --- |
| 교육 제목 | Tensorflow Keras 패키지 활용하기 |
| 교육 일시 | 10월 25일(월) |
| 교육 장소 | 강의실 |
| **교육 내용** | |
|  | * Gradient Tape   with tf.GradientTape() as tape :  t3 = t1 \* t2  gradient = tape.gradient(t3, [t1, t2])  GradientTape은 자동적으로 미분을 해서 w와 b 를 찾는다   * Multi Regression   model = tf.keras.Sequential()  model.add(tf.keras.layers.Dense(1, input\_dim=3))   * 선언된 모델에 add를 통해 layer를 쌓아감. 입력변수 개수3, perceptron 1 * Regression의 loss 값 MSE * Regression의 optimizer 값 SGD * Regression의 metrics 값 MAE * Logistic Regression   model = tf.keras.Sequential()  model.add(tf.keras.layers.Dense(1, input\_dim=1, activation=’sigmoid’))   * 로지스틱회귀의 활성함수로 시그모이드 함수 꼭 선택(선형->비선형) * Logistic Regression의 loss값 MSE * Losistic Regression의 optimizer 값 SGD * Logistic Regression의 metrics 값 binary\_accuracy * XOR Problem   XOR Problem은 선형을 비선형으로 바꾸기 위해서 시그모이드 레이어를 층층이(더 많이) 쌓아준다. 마지막 아웃풋 레이어는 sigmoid perceptron 1개만 설정한다.   * XOR 의 loss값 binary\_crossentropy * XOR 의 optimizer 값 SGD * XOR 의 metrics 값 binary\_accuracy * House Price 예측 * Logistic Regression 의 optimizer = SGD * Logistic Regression 의 loss = MSE * Logistic Regression 의 metrics = RSE * 딥러닝의 기본 흐름  1. Data 생성 or DATA 읽기 2. DATA의 변수에 따른 차원 확인 3. 퍼셉트론의 입력 차원 맞추고 Layer 완성하기 4. 퍼셉트론 생성(이 때, Activation주의!) 5. 퍼셉트론 loss 함수 지정 6. 경사하강법 (loss 함수의 min을 찾기 위한 최적화 방법 선택 ) 7. Metric 설정하고 나서, compile(Task에 따라서 Metric 선택에 주의) 8. 학습완료에 따른 결과 보기 |
| 전체 내용 |