|  |  |
| --- | --- |
| 교육 제목 | **Gradient Tape & Regression과 Classification에서의 Gradient** |
| 교육 일시 | 21.10.25 |
| 교육 장소 | 오프라인 (영우) |
| **교육 내용** | |
| 오전 | **▣ Gradient Tape**  -------------------------------------------------------------------------------------------  **∇ 미분해주는 함수**  \* t3를 각각 t1, t2에 대해서 미분해 줌.  with tf.GradientTape() as tape:    t3 = t1 \*t2  gradient = tape.gradient(t3, [t1, t2])  -------------------------------------------------------------------------------------------  **▣ Multi Regression에서의 Gradient**  **∇ 모델 설정**  model = tf.keras.Sequential()  model.add(tf.keras.layers.Dense(1, input\_dim = 3))  model.summary()  **∇ 모델 학습방법 결정하기**  optimizer = tf.keras.optimizers.SGD(lr=0.01)  loss = tf.keras.losses.**mse**  metrics = tf.keras.metrics.**mae**  **∇ 모델 컴파일하기**  model.compile  (loss = loss, optimizer = optimizer, metrics = [metrics]  **∇ 모델 피팅하기**  model.compile  (loss = loss, optimizer = optimizer, metrics = [metrics]  -------------------------------------------------------------------------------------------**▣ Classification에서의 Gradient**  **∇ 모델 설정**  model = tf.keras.Sequential()  model.add(tf.keras.layers.Dense(1, input\_dim = 3))  model.summary()  **∇ 모델 학습방법 결정하기**  optimizer = tf.keras.optimizers.SGD(lr=0.01)  loss = tf.keras.losses.**crossentropy**  metrics = tf.keras.metrics.**accuracy**  **∇ 모델 컴파일하기**  model.compile  (loss = loss, optimizer = optimizer, metrics = [metrics]  **∇ 모델 피팅하기**  model.compile  (loss = loss, optimizer = optimizer, metrics = [metrics]  ------------------------------------------------------------------------------------------- |
| 오후 | **▣ Logistic Regression에서의 Gradient**  **∇ 모델 설정**  model = tf.keras.Sequential()  model.add(tf.keras.layers.Dense(1, input\_dim = 3, activation = ‘sigmoid’)) **activation = ‘sigmoid’ 사용!!!!!**  model.summary()  **∇ 모델 학습방법 결정하기**  optimizer = tf.keras.optimizers.SGD(lr=0.01)  loss = tf.keras.losses.mse  metrics = tf.keras.metrics.**binary\_accuracy**  **∇ 모델 컴파일하기**  model.compile  (loss = loss, optimizer = optimizer, metrics = [metrics]  **∇ 모델 피팅하기**  model.compile  (loss = loss, optimizer = optimizer, metrics = [metrics]  ------------------------------------------------------------------------------------------- |