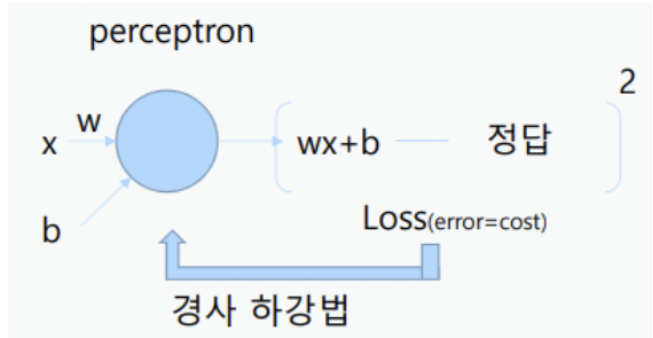
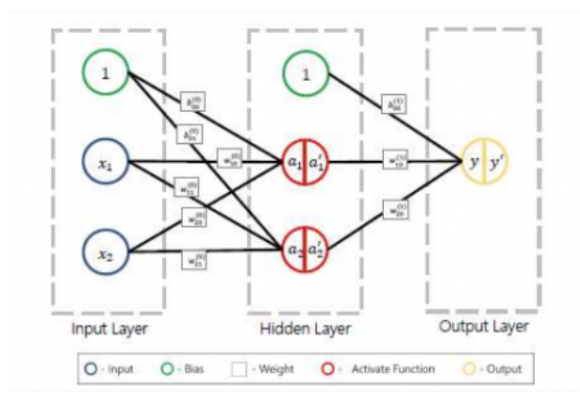


교육 제목	데이터 기반 인공지능 시스템 엔지니어 양성 과정
교육 일시	2021년 10월 28일
교육 장소	YGL C-6 학과장 & 자택(디스코드 이용한 온라인)
교육 내용	
오전	<p>Review</p> <p>1. 딥러닝 최소 기본 구조 : Perceptron</p>  <p>Regression : 출력함수 = None, loss = mse</p> <p>Binary classification : 출력함수 = sigmoid, loss = binary crossentropy</p> <p>Categorical classification : 출력함수 = softmax loss = categorical crossentropy</p> <p>2. Multi Layer Perceptron (= Neural Network)</p>  <p>Multi Layer Perceptron(=Neural Network) 구조</p> <ul style="list-style-type: none"> • Input layer : 입데이터 전달 받는 역할 • Hidden Layer : 데이터를 representation • Output Layer : representation 을 전달받아 regression/classification

Hidden layer의 activation function

- 여러 layer를 쌓는 효과를 보기 위해 비선형적이어야 함.
- Layer들의 w,b가 잘 업데이트 되기 위해 오차 역전파에서 오차의 미분값이 잘 전달 되도록 해야 함
- 미분 가능해야 함

Output layer의 activation function은 Task에 따라 선정.

- Regression의 activation function은 None
(regression은 출력값이 숫자값으로 입력*w + b값이 예측값)
- Classification의 activation function은 sigmoid or softmax
(classification은 0 or 1의 제한된 출력)

3. 딥러닝 모델별 특징 비교

Model	Example	[Output] Activation Fun.	Loss Fun.	Metric
Simple Linear Regression	이자율을 이용해서 집값을 예상하기	Linear Activation	Mean Squared Error	Mean Absolute Error
Multivariate Linear Regression	이자율 외 여러 변수를 이용해서 집값을 예상하기	Linear Activation	Mean Squared Error	Mean Absolute Error
Binary Classification	해당 이미지는 고양이일까요? 아닐까요?	Sigmoid / Logistic	Binary Cross Entropy	Binary Accuracy
Multi-Class Classification	해당 이미지는, 강아지일까요? 고양이일까요? 자동차일까요??	Softmax	Categorical Cross Entropy	Categorical Accuracy

MLP를 이용하는 이미지 분류 주요 코드

```
# MNIST 데이터 불러오기
(X_train, Y_train), (X_test, Y_test) = tf.keras.datasets.mnist.load_data()

X_train = X_train.reshape(X_train.shape[0], 784).astype('float32')
X_test = X_test.reshape(X_test.shape[0], 784).astype('float32')

Y_train = tf.keras.utils.to_categorical(Y_train, 10)
Y_test = tf.keras.utils.to_categorical(Y_test, 10)
```

```
# 모델 설계
# method 1
```

```
input_Layer = tf.keras.layers.Input(shape=(784,))
x = tf.keras.layers.Dense(512, activation='relu')(input_Layer)
Out_Layer = tf.keras.layers.Dense(10, activation='softmax')(x)

model = tf.keras.Model(inputs=[input_Layer], outputs=[Out_Layer])
model.summary()
```

```
# 모델 컴파일
```

```
loss=tf.keras.losses.categorical_crossentropy
optimizer = tf.keras.optimizers.Adam(lr=0.001)
model.compile(loss=loss,
              optimizer=optimizer,
              metrics=['accuracy'])
```

<p>오후</p>	<div data-bbox="399 253 1129 286">CNN(Convolution Neural Network)을 이용하는 이미지 분류</div> <div data-bbox="399 302 574 336">conv 와 pool</div> <div data-bbox="399 347 1412 728"> <p>Reference: https://www.cnblogs.com/yxwxf/p/3831310.html http://blog.cs.cmu.edu/enrwei2</p> </div> <div data-bbox="399 761 1412 1209"> <p>Convolution Layer는 filter(w,b)을 통해 지역적 특징을 추출.</p> <ul style="list-style-type: none"> 여기서, 학습은 filter가 가진 w와 b가 학습되어 짐. 즉, 학습되어진 filter의 w와 b가 입력 이미지에 곱해져서 activation map 생성 <p>Max pooling은 학습 되는 파라미터가 없음. 단순히 입력받은 이미지의 특징맵에서 더 도드라지는 특징 추출</p> <p>Fully Connected를 통해서 앞서 학습한 regression/classification을 함.</p> <p>사람이 설계한 특징을 추출하는 것이 아닌 학습을 통해 FC가 Regression/classification을 잘하도록 Feature라고 하는 Activation map(=feature map)을 잘 추출하도록 Filter의 w,b가 학습을 통해 업데이트 됨</p> </div>