

Ambient AI Bootcamp

Practice 2



SNU Graduate School of Data Science

Contents

- Useful TensorFlow APIs
- Implementation of a CNN for 2D object classification
- HW2: Training a CNN with TensorFlow

2-1. Useful TensorFlow APIs

0. Load Dataset and Model Build

Dataset: MNIST dataset

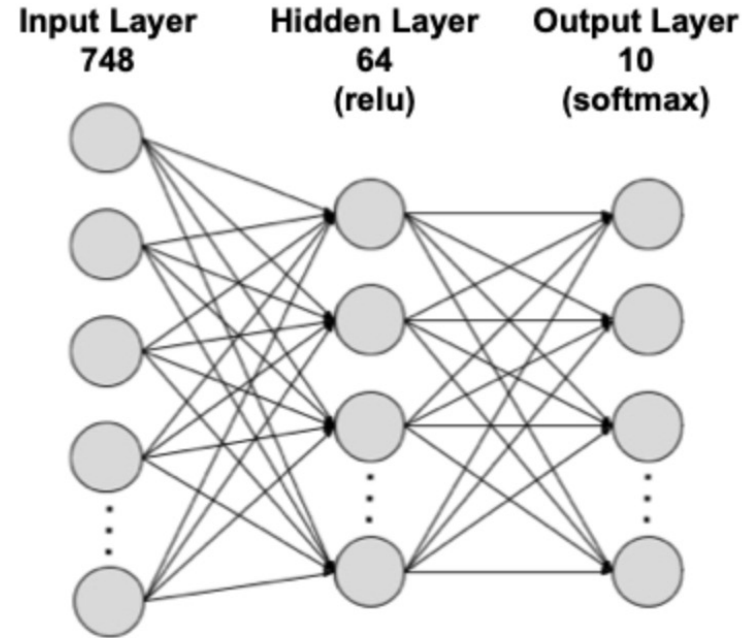
- 간단한 컴퓨터 비전 데이터세트로, 0-9까지 손으로 쓰여진 숫자 이미지들로 구성되어 있음
(<http://yann.lecun.com/exdb/mnist/>)
- 각 이미지는 28x28로 크기 표준화되고 중심에 배치
- 784 features 1D numpy array
- 60,000개의 training set과 10,000개의 test set으로 이루어져 있음



o. Load Dataset and Model Build

Model / Training

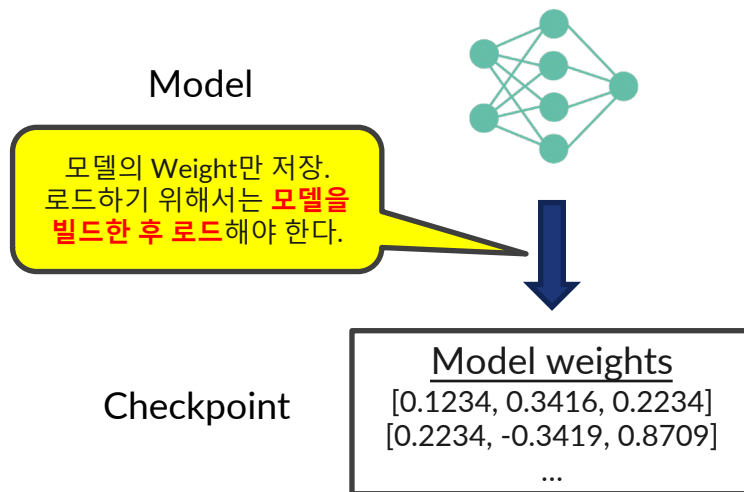
- Loss: Sparse Categorical Cross Entropy
- Optimizer = Adam
- Learning rate = 0.001



1. Model Save and Load

(1) Checkpoint Save / Load

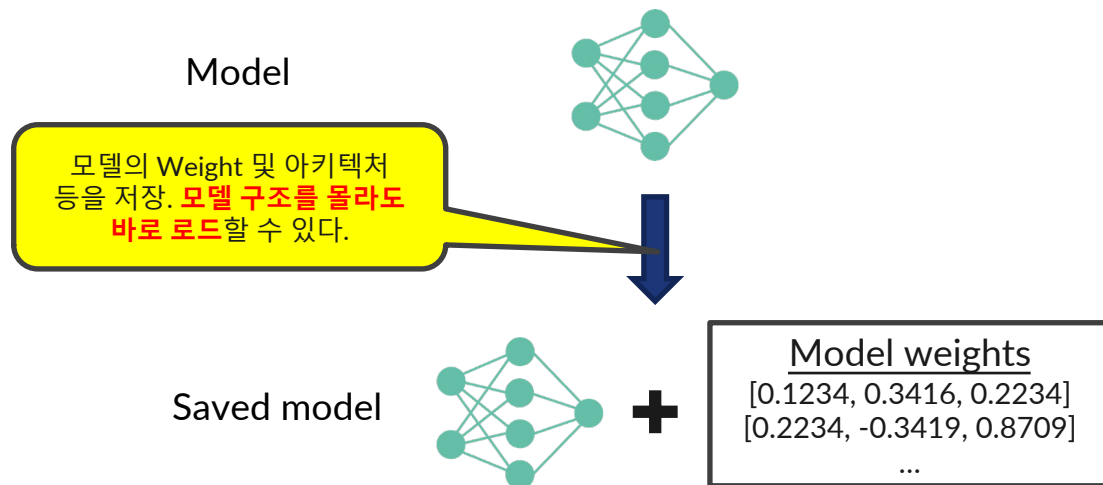
- 모델은 Build된 상태라고 가정하고 학습된 모델의 Weight만 저장한다.



1. Model Save and Load

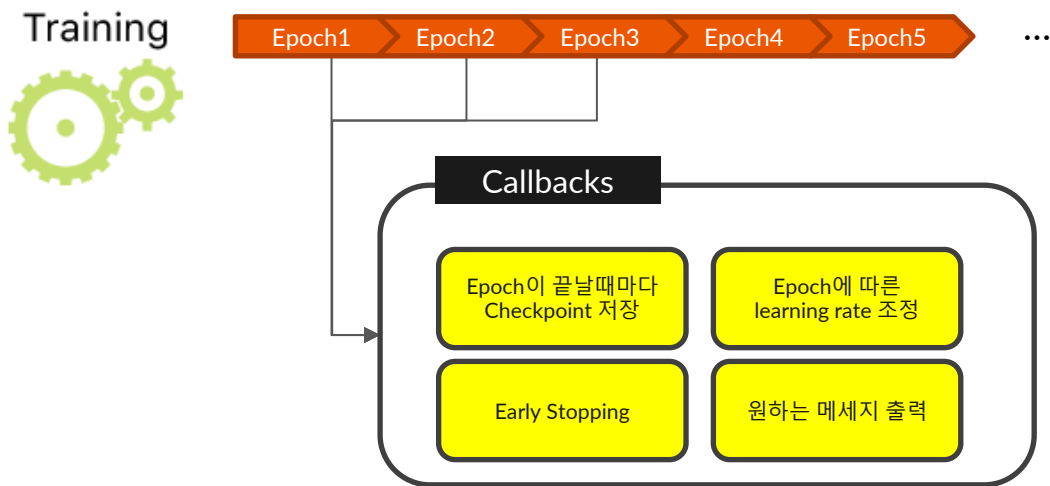
(2) Model Save / Load

- `tf.keras.models.save_model` / `load_model` 을 활용하면 Model의 아키텍처 및 구성, 모델의 Weight, Compile 정보, Optimizer와 그 상태 등이 같이 저장/로드된다.



2. Callbacks

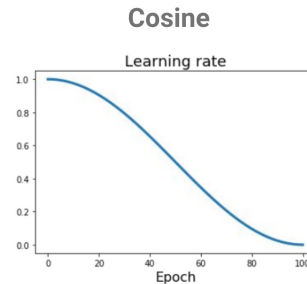
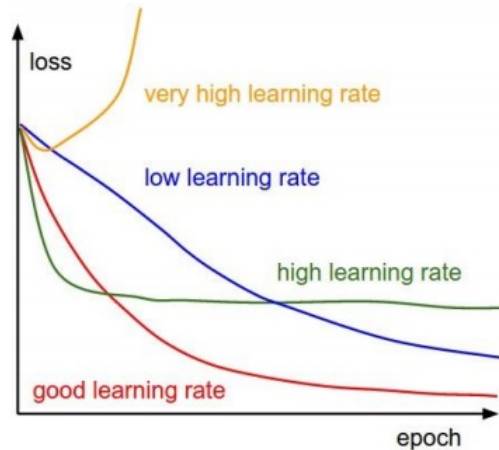
- 모델을 트레이닝하며 중간에 원하는 작업을 넣고 싶을 때 **Callback** 함수를 활용한다.
- `tf.keras.callbacks.Callback` 라이브러리에 있는 built-in callback 함수들을 잘 활용한다. (https://www.tensorflow.org/api_docs/python/tf/keras/callbacks)



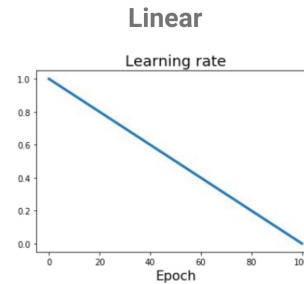
2. Callbacks

Learning Rate

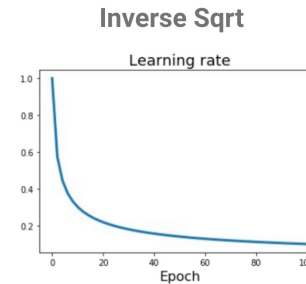
- Optimizers require to specify the learning rate as a hyperparameter
- In most cases, we start with a large initial learning rate, then decay over time
- **Learning rate decay**



$$\alpha_t = \frac{1}{2} \alpha_0 (1 + \cos(t\pi/T))$$



$$\alpha_t = \alpha_0 (1 - t/T)$$



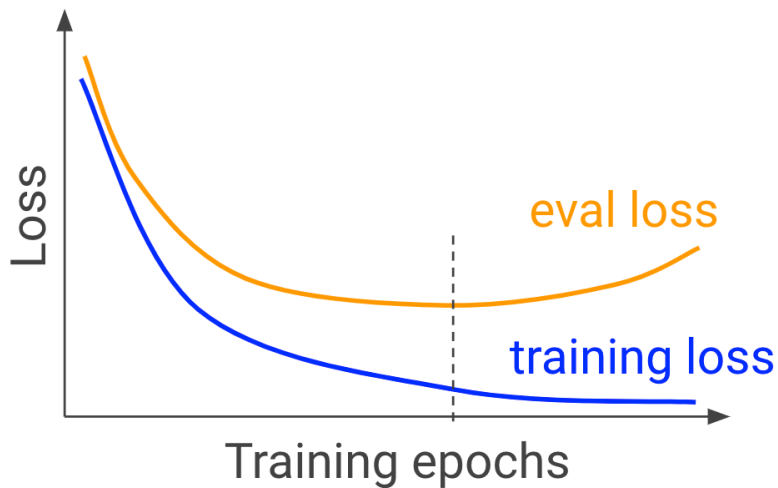
$$\alpha_t = \alpha_0 / \sqrt{t}$$

α_0 : Initial learning rate
 α_t : Learning rate at epoch t
 T : Total number of epochs

2. Callbacks

Early Stopping

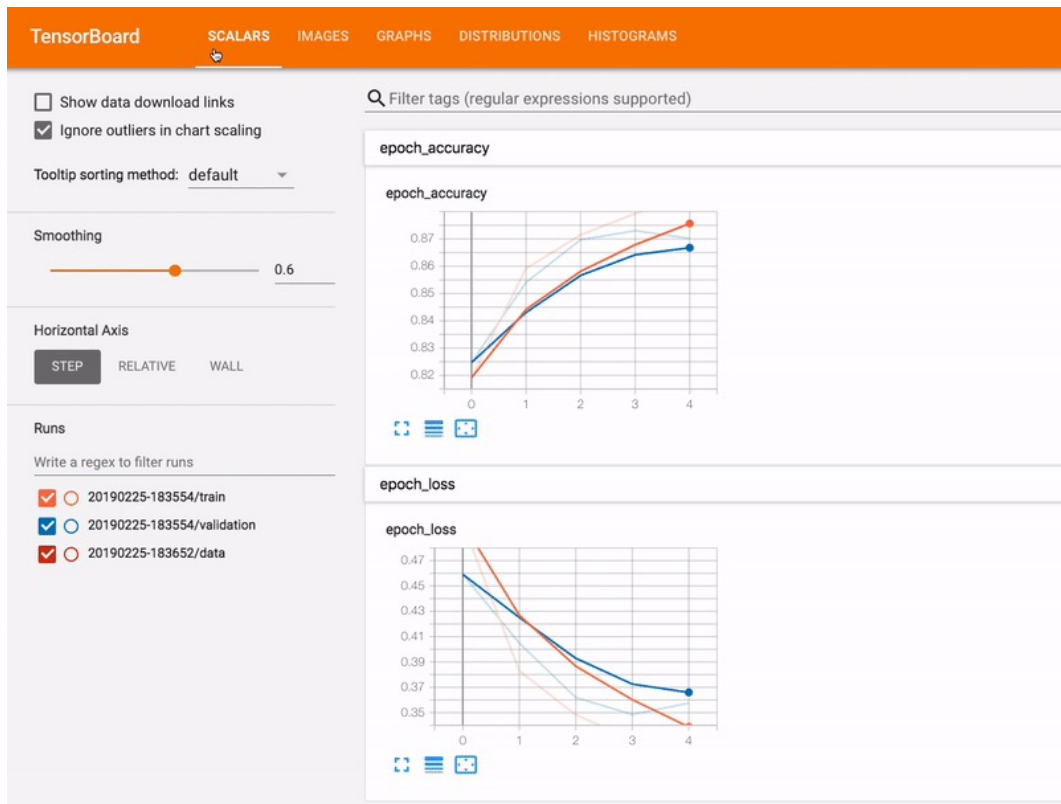
- Train set의 일부를 Validation set으로 이용한다.
- Validation set이 overfitting을 시작하면 학습을 멈춘다.



3. Tensorboard

- Tensorboard는 TensorFlow 시각화 도구로, 실험에 필요한 시각화 및 도구를 제공한다.











(<https://www.tensorflow.org/tensorboard>)



2-2. CNN for 2D Classification

0. Dataset

- MLP 실습과 동일하게 Fashion MNIST dataset을 이용하되, CNN 구조를 활용하여 이미지를 구분하는 모델을 만든다.
- Fashion MNIST data shape
 - (28, 28)

Label	Description	Examples
0	T-Shirt/Top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandals	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boots	

0. Dataset

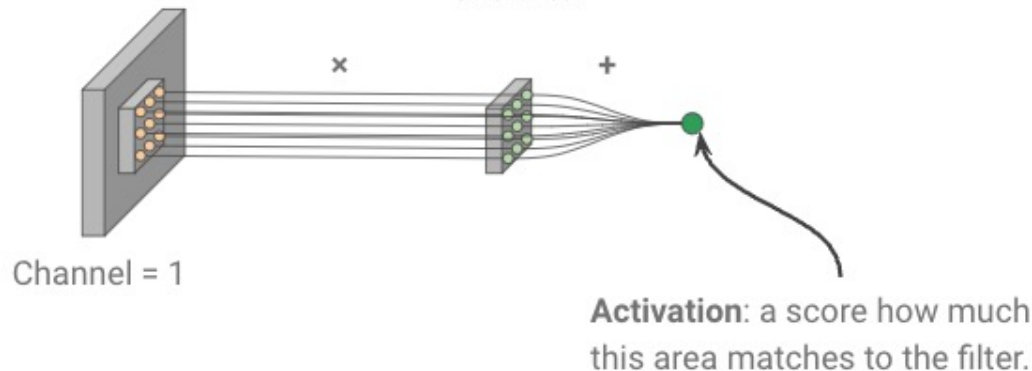
Data Shape

- Convolution layer에서는 (H, W, C) 형태의 Data shape을 요구한다. → Channel dimension 추가

Monotone image:

32×32 image

3×3 filter

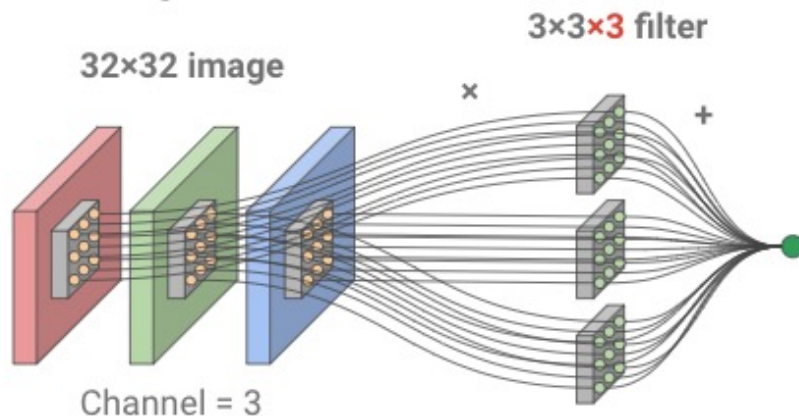


0. Dataset

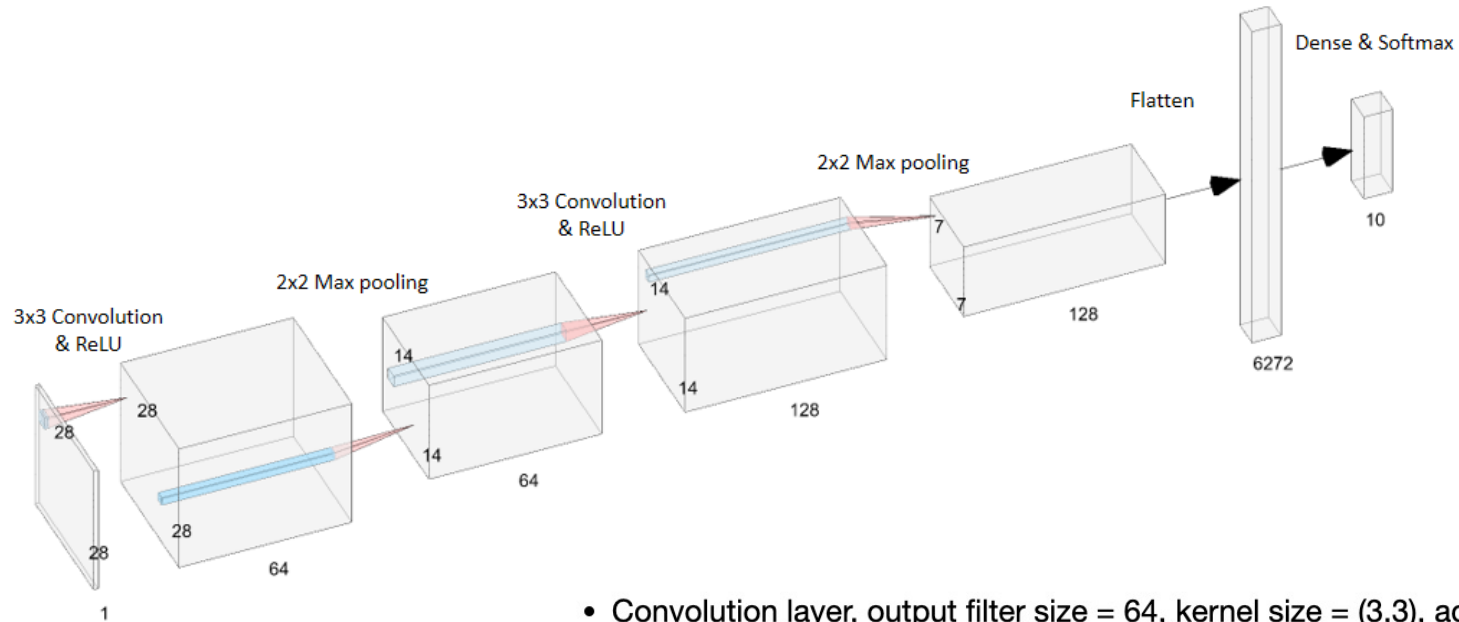
Data Shape

- Convolution layer에서는 (H, W, C) 형태의 Data shape을 요구한다. → Channel dimension 추가

RGB color image:



1. Model



- Convolution layer, output filter size = 64, kernel size = (3,3), activation = relu
- Max pooling layer, 2x2 pooling, 2x2 strides
- Convolution layer, output filter size = 128, kernel size = (3,3), activation = relu
- Max pooling layer, 2x2 pooling, 2x2 strides
- Flatten and Dense layer, softmax activation

2. Training

- 10 epoch 이상 training
- Callbacks 이용
 - Learning rate scheduler
 - Checkpoint save
- MLP 결과와 비교

MLP Model Training



Thanks!