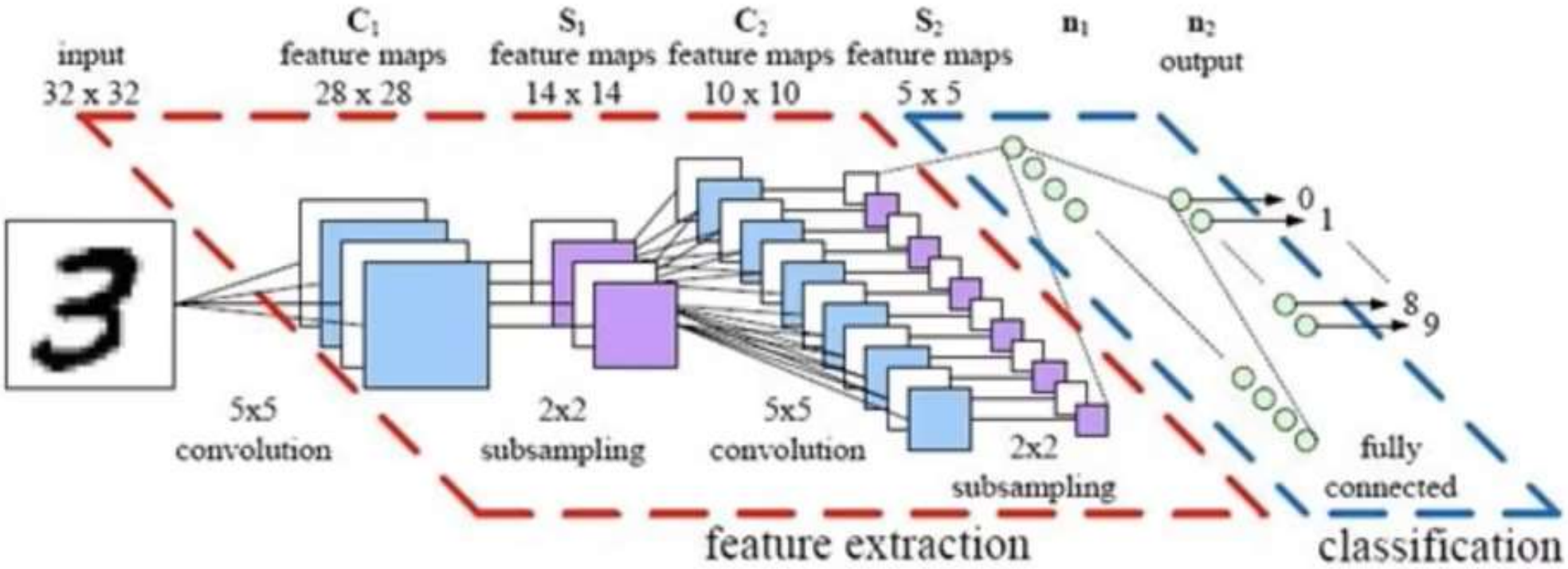
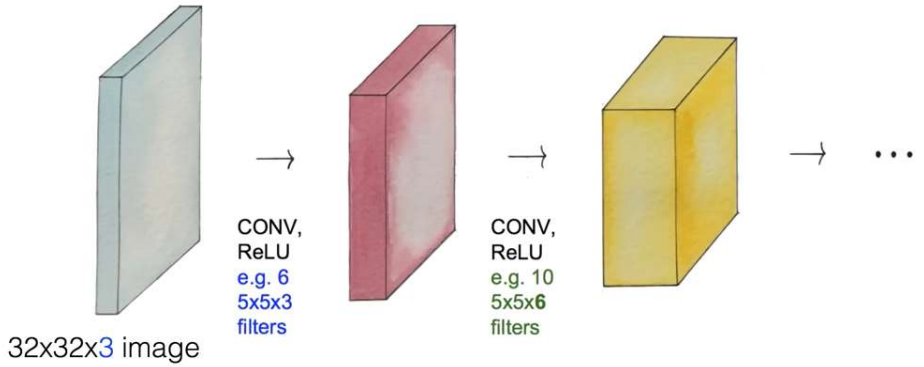
Deep Learning #4

20 Jan. 2021

자율주행시스템 개발팀
신 주 석

- **Convolution Neural Network**
  - **MNIST 99% using CNN**



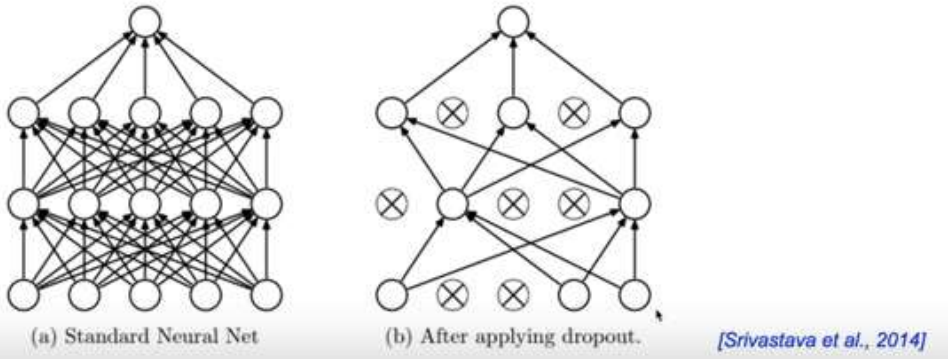


- **Deep Neural Network**
  - **Dropout**
    - » **Avoid Overfitting**

Dropout: A Simple Way to Prevent Neural Networks from Overfitting [Srivastava et al. 2014]

Regularization: **Dropout**

"randomly set some neurons to zero in the forward pass"

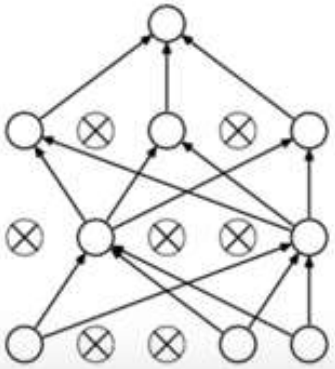


(a) Standard Neural Net (b) After applying dropout. [Srivastava et al., 2014]

```
dropout_rate = tf.placeholder("float")
_L1 = tf.nn.relu(tf.add(tf.matmul(X, W1), B1))
L1 = tf.nn.dropout(_L1, dropout_rate)
```

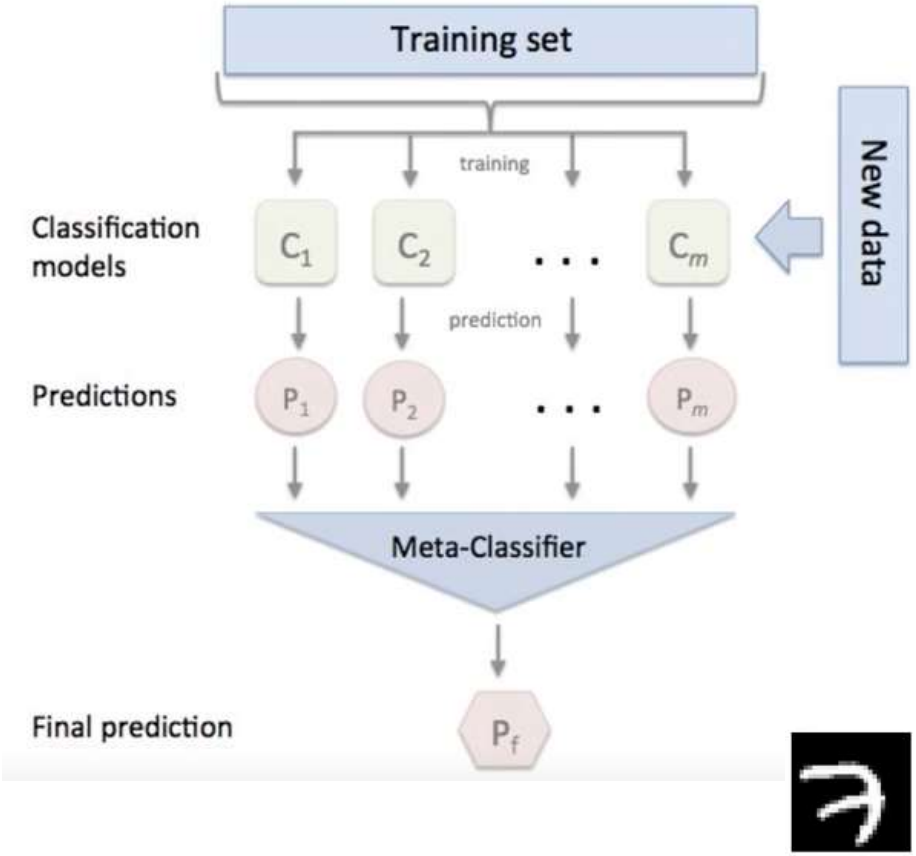
```
TRAIN:
  sess.run(optimizer, feed_dict={X: batch_xs, Y: batch_ys,
  dropout_rate: 0.7})
```

```
EVALUATION:
  print "Accuracy:", accuracy.eval({X: mnist.test.images, Y:
  mnist.test.labels, dropout_rate: 1})
```



Forces the network to have a redundant representation.

- **Convolution Neural Network**
  - **Ensemble**



Accuracy: 99.52%

	0	1	2	3	4	5	6	7	8	9
C_1 →	0.1	0.01	0.02	0.8	...	...	...	...	...	...
C_2 →	0.01	0.5	0.02	0.4	...	...	...	...	...	...
.										
.										
.										
C_m →	0.01	0.01	0.1	0.7	...	...	...	...	...	...
Sum	0.12	0.52	0.14	**1.9**	...	...	...	...	...	...

◆ Convolution Neural Network

– 실습: MNIST 99% using CNN

» Deep CNN (**Dropout + additional FC layer**)+**Callback**

```
class callback_Chk_ACC(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if(logs.get('accuracy')>0.99):
            print("\nAccuracy is 99%")
            self.model.stop_training = True

callbacks = callback_Chk_ACC()
```

```
tf.model.fit(x_train, y_train, batch_size=batch_size, epochs=training_epochs, callbacks=[callbacks])
```

◆ Convolution Neural Network

– Fashion MNIST

```
import tensorflow as tf
print(tf.__version__)
mnist = tf.keras.datasets.fashion_mnist
(training_images, training_labels), (test_images, test_labels) = mnist.load_data()
training_images=training_images.reshape(60000, 28, 28, 1)
training_images=training_images / 255.0
test_images = test_images.reshape(10000, 28, 28, 1)
test_images=test_images/255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(28, 28, 1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
model.summary()
model.fit(training_images, training_labels, epochs=1)
test_loss = model.evaluate(test_images, test_labels)
```

실습 #2: TSR(Traffic Sign Recognition) using CNN

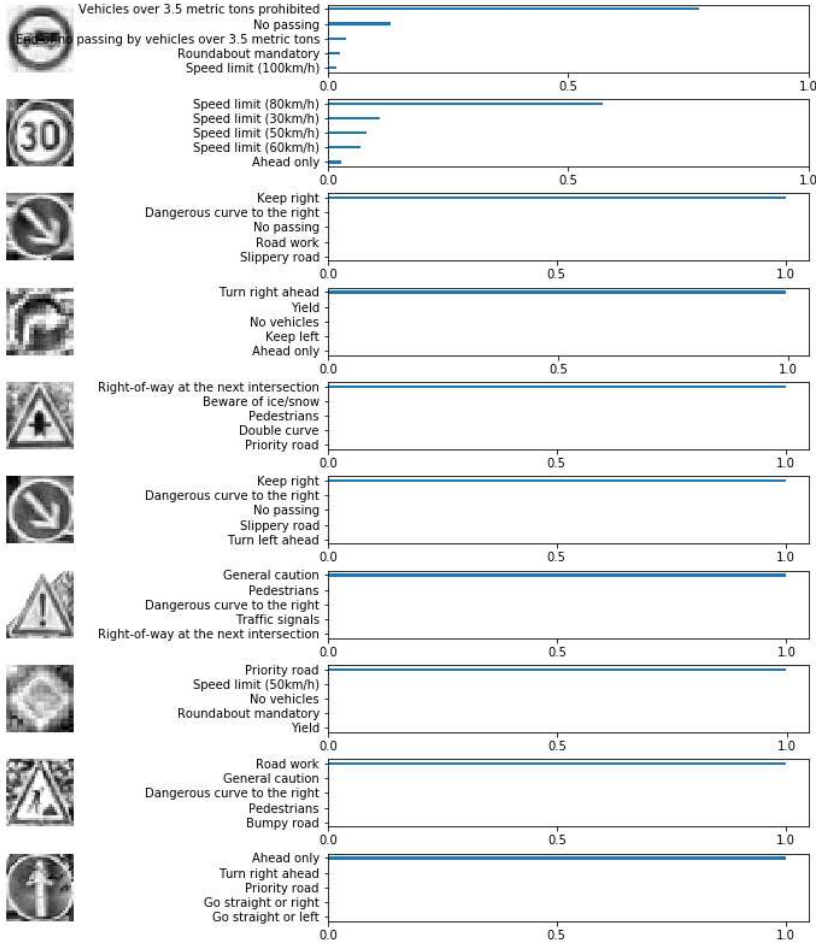
◆ TSR using CNN

– Build a Traffic Sign Recognition Project

- » Load the data set
- » Explore, Summarize and visualize the data set
- » Design, Train and Test a CNN Model architecture
- » Use the model to make predictions on new images
- » Analyze the softmax probabilities of the new images

 # TODO: Reference provided code and some test data

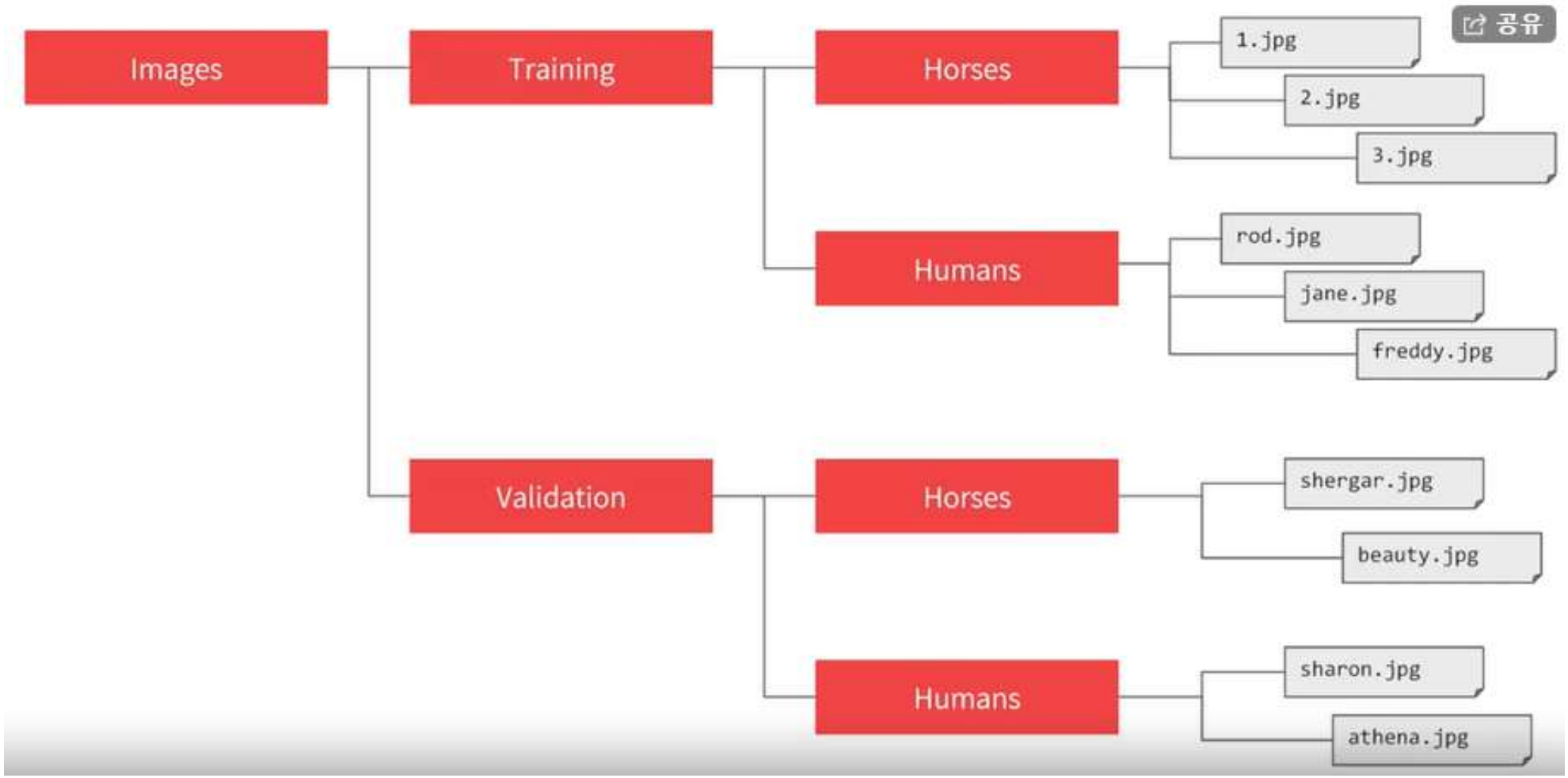
```
def plot_test_images(images,n):
    fig, axes = plt.subplots(1, n, figsize=(13,5))
    fig.subplots_adjust(hspace=0.1, wspace=0.1)

    for i, ax in enumerate(axes.flat):
        ax.imshow(images[i])
        ax.set_title(i+1)
        ax.set_xticks([])
        ax.set_yticks([])
#     fig.savefig('in5.png')
### Load the images
from skimage import io
imgs = [ io.imread('test0/test{}.png'.format(i + 11)) for i in range(10) ]

plot_test_images(imgs,10)
```



- **Method for avoid Overfitting**
  - **Image Augmentation using “ImageDataGenerator in Tensroflow”**
  - **Image Augmentation with Dropout**
  - **Transfer Learning (w/Dropout)**

- **Understanding ImageDataGenerator in tensorflow**

```
from tensorflow.keras.preprocessing.image
import ImageDataGenerator
```



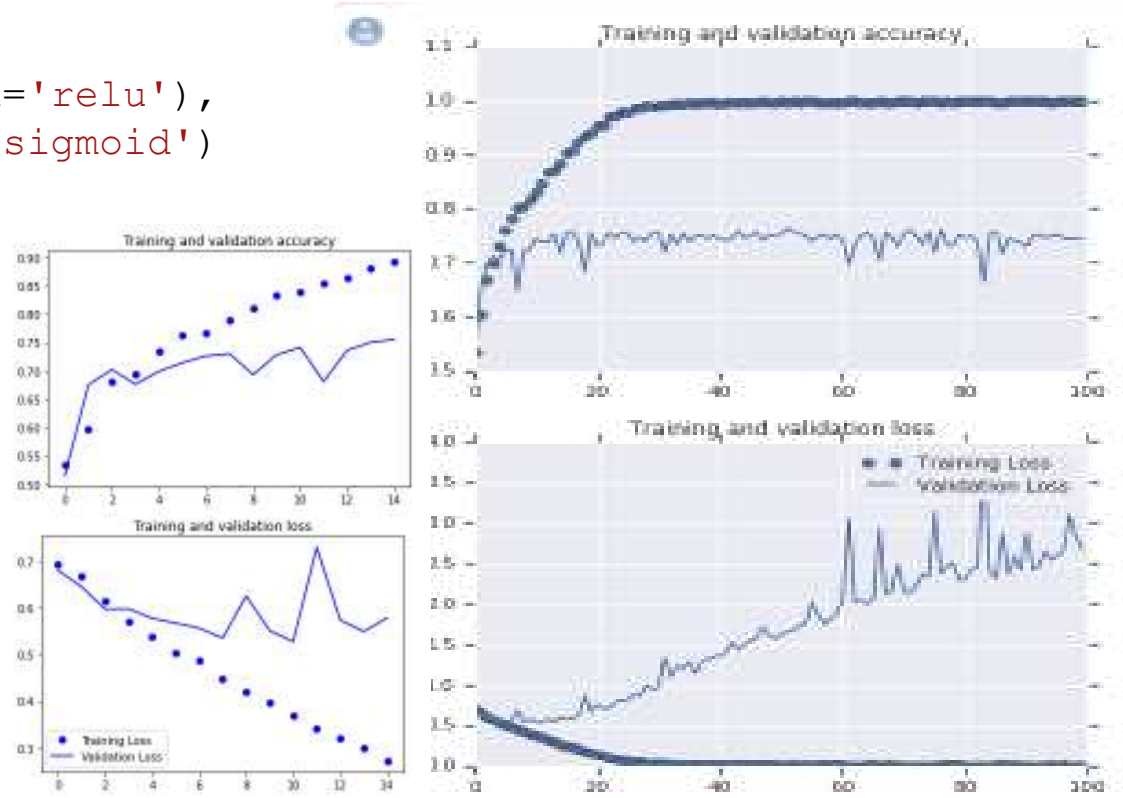
- **Understanding ImageDataGenerator in tensorflow**

```
train_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
        train_dir,
        target_size=(300, 300),
        batch_size=128,
        class_mode='binary')
```

```
test_datagen = ImageDataGenerator(rescale=1./255)

validation_generator = test_datagen.flow_from_directory(
        validation_dir,
        target_size=(300, 300),
        batch_size=32,
        class_mode='binary')
```

◆ Dogs and Cats Classification (Kaggle Dataset)

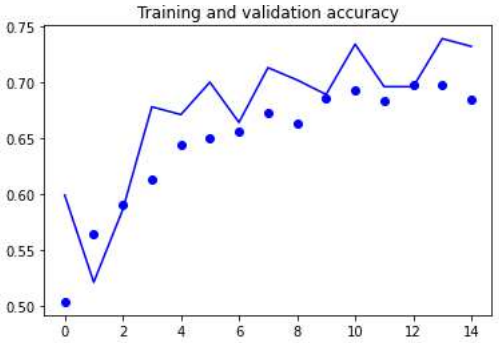
```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model.compile(loss='binary_crossentropy',
              optimizer=RMSprop(lr=1e-4),
              metrics=['accuracy'])
```

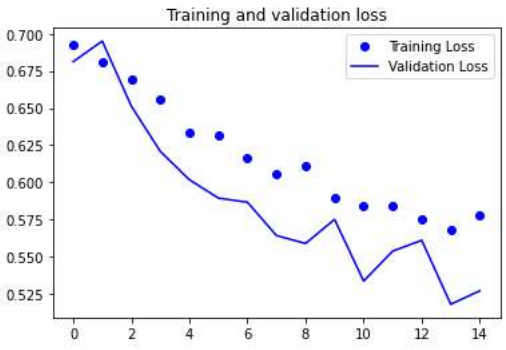


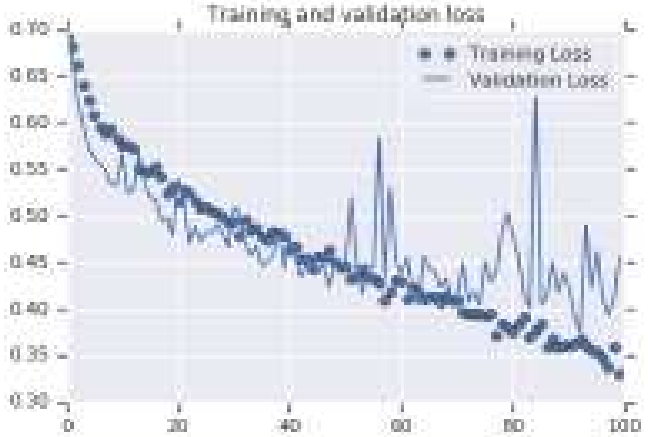
◆ Dogs and Cats Classification (Kaggle Dataset)

– **Data Augmentation (applied to only training Dataset) using ImageDataGenerator**

```
train_datagen = ImageDataGenerator(
        rescale=1./255,
        rotation_range=40,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True,
        fill_mode='nearest')

test_datagen =
         ImageDataGenerator(
         rescale=1./255)
```







◆ Dogs and Cats Classification (Kaggle Dataset)

- Data Augmentation (applied to only training Dataset) using ImageDataGenerator
- Image data preprocessing (keras.io)

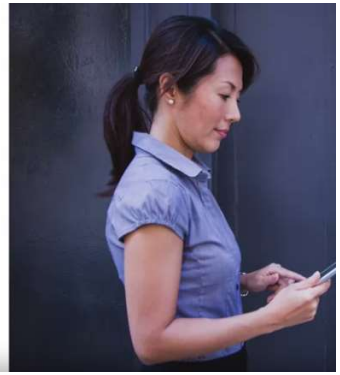
- **rotation_range**: Int. Degree range for random rotations.
- **width_shift_range**: Float, 1-D array-like or int - float: fraction of total width, if < 1, or pixels if >= 1. - 1-D array-like: random elements from the array. - int: integer number of pixels from interval `(-width_shift_range, +width_shift_range)` - With `width_shift_range=2` possible values are integers `[-1, 0, +1]`, same as with `width_shift_range=[-1, 0, +1]`, while with `width_shift_range=1.0` possible values are floats in the interval [-1.0, +1.0).
- **height_shift_range**: Float, 1-D array-like or int - float: fraction of total height, if < 1, or pixels if >= 1. - 1-D array-like: random elements from the array. - int: integer number of pixels from interval `(-height_shift_range, +height_shift_range)` - With `height_shift_range=2` possible values are integers `[-1, 0, +1]`, same as with `height_shift_range=[-1, 0, +1]`, while with `height_shift_range=1.0` possible values are floats in the interval [-1.0, +1.0).
- **brightness_range**: Tuple or list of two floats. Range for picking a brightness shift value from.
- **shear_range**: Float. Shear Intensity (Shear angle in counter-clockwise direction in degrees)
- **zoom_range**: Float or [lower, upper]. Range for random zoom. If a float, `[lower, upper] = [1-zoom_range, 1+zoom_range]`.
- **channel_shift_range**: Float. Range for random channel shifts.
- **fill_mode**: One of {"constant", "nearest", "reflect" or "wrap"}. Default is 'nearest'. Points outside the boundaries of the input are filled according to the given mode: - 'constant': kkkkkkkk|abcd|kkkkkkkk (cval=k) - 'nearest': aaaaaaaa|abcd|dddddddd - 'reflect': abcddcba|abcd|dcbaabcd - 'wrap': abcdabcd|abcd|abcdabcd
- **cval**: Float or Int. Value used for points outside the boundaries when `fill_mode = "constant"`.
- **horizontal_flip**: Boolean. Randomly flip inputs horizontally.
- **vertical_flip**: Boolean. Randomly flip inputs vertically.
- **rescale**: rescaling factor. Defaults to None. If None or 0, no rescaling is applied, otherwise we multiply the data by the value provided (after applying all other transformations).

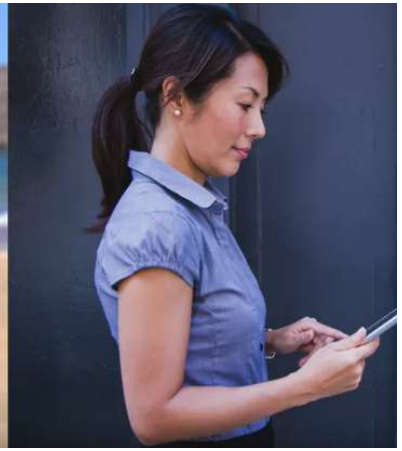
◆ Dogs and Cats Classification (Kaggle Dataset)

- Data Augmentation (applied to only training Dataset) using ImageDataGenerator
- **Rotation range**

- **Shear range**

- **Zoom range**

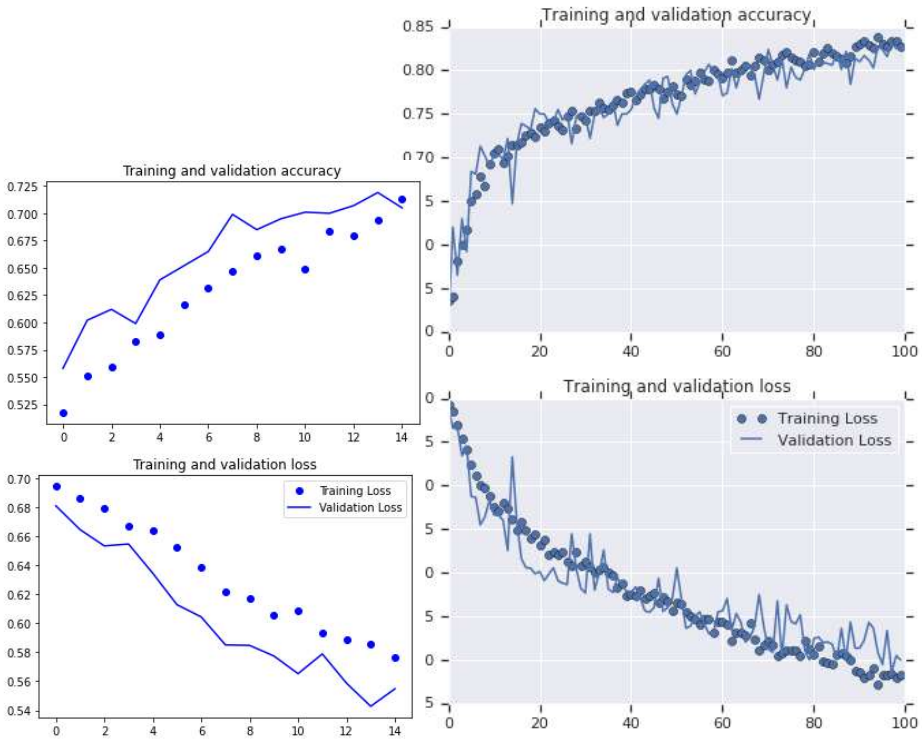




◆ Dogs and Cats Classification (Kaggle Dataset)

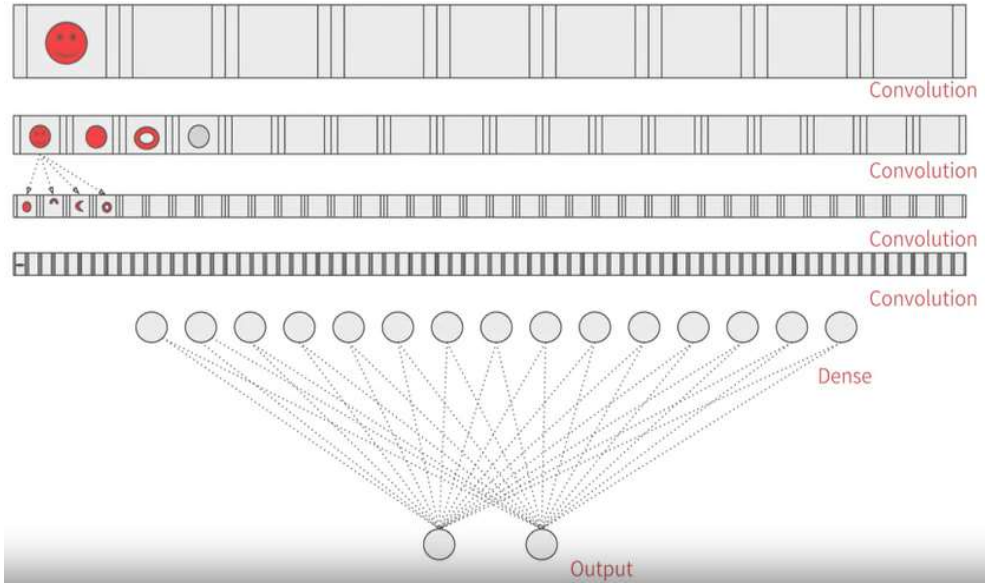
– Dropout

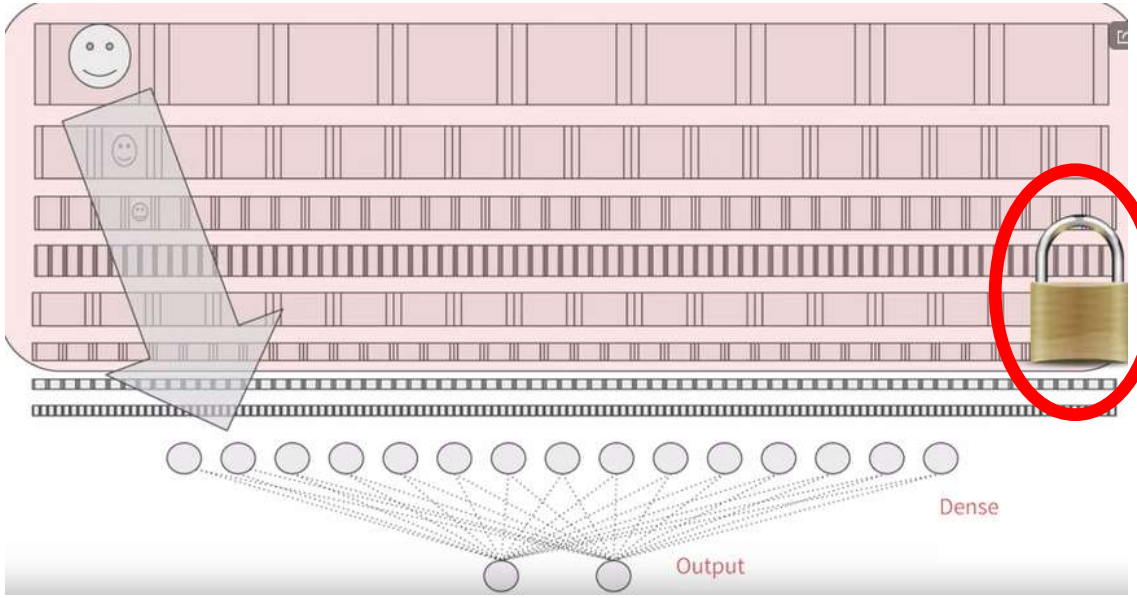
```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```



◆ Transfer Learning (전이학습)

- 학습 데이터 수가 적을 때 (Augmentation을 하더라도 Overfitting이 발생하거나 인식률이 좋지 않을 경우)
- 이미 성능이 검증된 학습 파라미터 이용
  - » 1,000개의 클래스를 잘 Classification할 수 있는(error 3~5%미만)
  - » 이미 학습되어 있는 좋은 convolution 필터를 이용
  - » 성능 향상 및 overfitting 방지
- Concept





◆ Transfer Learning (전이학습)

– Inception v3 Network (Transfer learning)

```
import os

from tensorflow.keras import layers
from tensorflow.keras import Model
```

```
https://storage.googleapis.com/mledu-datasets/
inception_v3_weights_tf_dim_ordering_tf_kernels
```

```
from tensorflow.keras.applications.inception_v3 import InceptionV3

local_weights_file = '/tmp/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5'

pre_trained_model = InceptionV3(input_shape = (150, 150, 3),
                                include_top = False,
                                weights = None)

pre_trained_model.load_weights(local_weights_file)
```

```
for layer in pre_trained_model.layers:
  layer.trainable = False
```

I can iterate through its layers and **lock them**, saying that they're **not going to be trainable with this code**.

"The inception V3 has a **fully-connected layer at the top**.
So by setting **include_top to false**, you're specifying that you want to ignore this and get straight to the convolutions."

◆ Transfer Learning (전이학습)

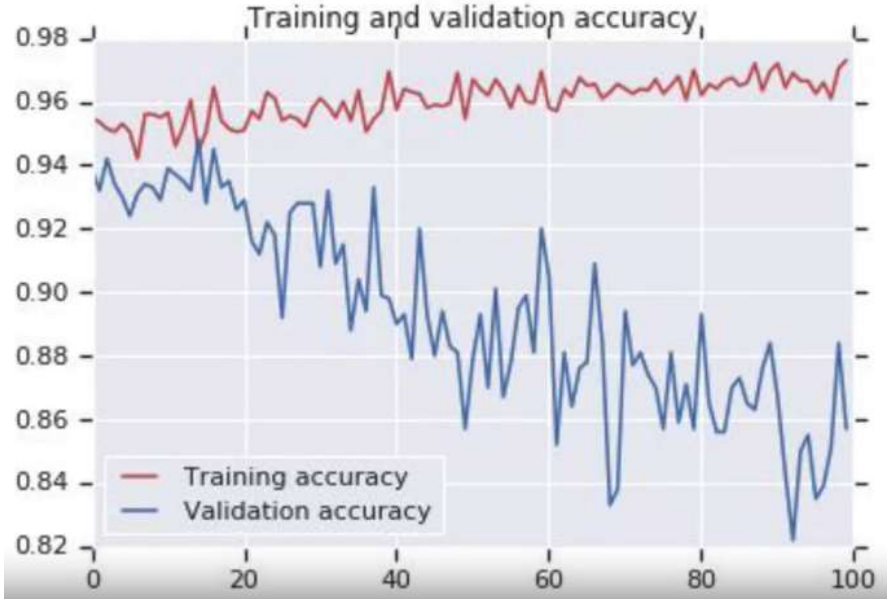
– Inception v3 Network (Transfer learning)

conv2d_161 (Conv2D)	(None, 7, 7, 192)	258048	activation_160[0][0]
batch_normalization_156 (BatchN	(None, 7, 7, 192)	576	conv2d_156[0][0]
batch_normalization_161 (BatchN	(None, 7, 7, 192)	576	conv2d_161[0][0]
activation_156 (Activation)	(None, 7, 7, 192)	0	batch_normalization_156[0][0]
activation_161 (Activation)	(None, 7, 7, 192)	0	batch_normalization_161[0][0]
average_pooling2d_15 (AveragePo	(None, 7, 7, 768)	0	mixed6[0][0]
conv2d_154 (Conv2D)	(N		
conv2d_157 (Conv2D)	(N		
conv2d_162 (Conv2D)	(N		
conv2d_163 (Conv2D)	(N		
batch_normalization_154 (BatchN	(None, 7, 7, 192)	576	conv2d_154[0][0]
batch_normalization_157 (BatchN	(None, 7, 7, 192)	576	conv2d_157[0][0]
batch_normalization_162 (BatchN	(None, 7, 7, 192)	576	conv2d_162[0][0]
batch_normalization_163 (BatchN	(None, 7, 7, 192)	576	conv2d_163[0][0]
activation_154 (Activation)	(None, 7, 7, 192)	0	batch_normalization_154[0][0]
activation_157 (Activation)	(None, 7, 7, 192)	0	batch_normalization_157[0][0]
activation_162 (Activation)	(None, 7, 7, 192)	0	batch_normalization_162[0][0]
activation_163 (Activation)	(None, 7, 7, 192)	0	batch_normalization_163[0][0]
mixed7 (Concatenate)	(None, 7, 7, 768)	0	activation_154[0][0] activation_157[0][0] activation_162[0][0] activation_163[0][0]

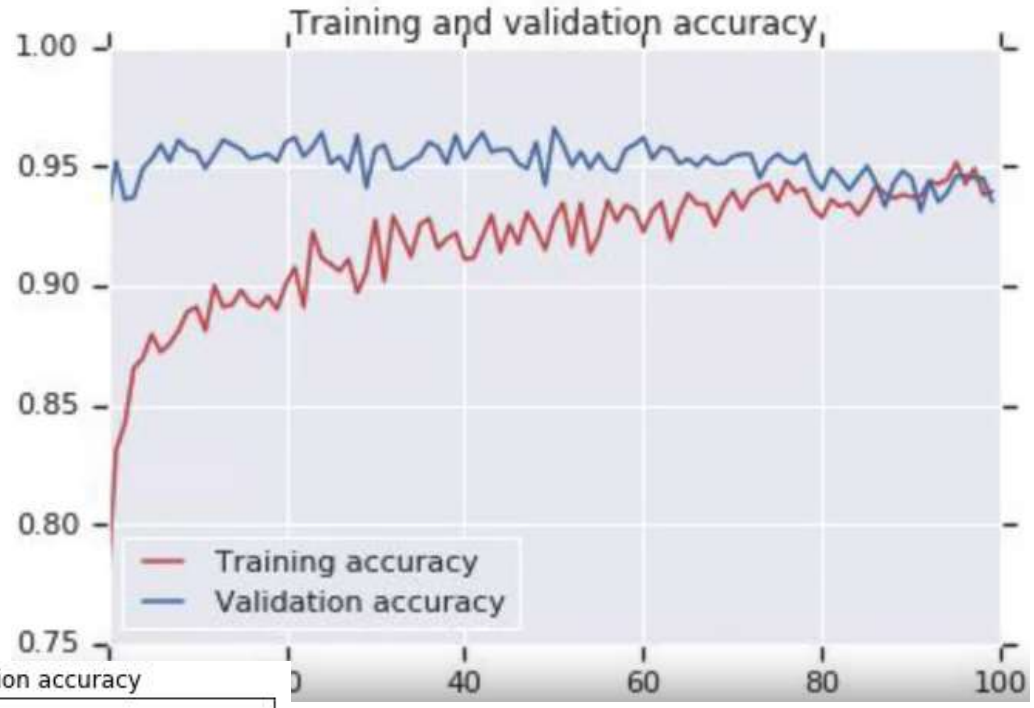
```
last_layer = pre_trained_model.get_layer('mixed7')
print('last layer output shape: ', last_layer.output_shape)
last_output = last_layer.output
```

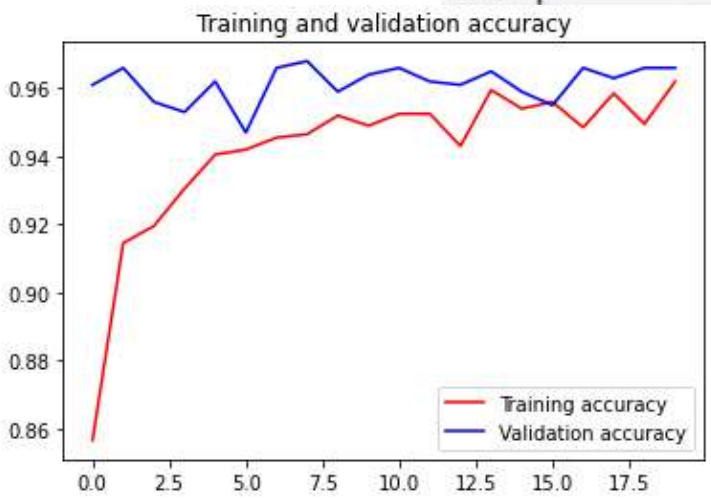
◆ Transfer Learning (전이학습)

– **Inception v3 Network (Transfer learning)**



Dropout 미적용시





Dropout 적용시

◆ 모델 성능 평가 지표

- **Accuracy**

$$\frac{TruePositives + TrueNegatives}{TruePositives + TrueNegatives + FalsePositives + FalseNegatives}$$

- **Precision**

$$\frac{TruePositives}{TruePositives + FalsePositives}$$

- **Recall**

$$\frac{TruePositives}{TruePositives + FalseNegatives}$$

		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

- **F1 Score** 조화 평균

$$2 * \frac{Precision * Recall}{Precision + Recall}$$

Recall: 1
Precision:0.01

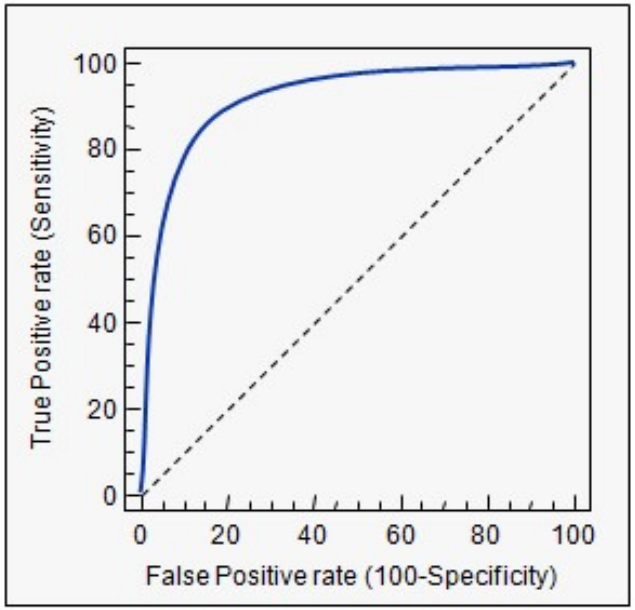
$$\frac{1 + 0.01}{2} = 0.505$$

$$2 * \frac{1 * 0.01}{1 + 0.01} = 0.019$$

- **ROC (Receive Operating Characteristic) Curve**

◆ 모델 성능 평가 지표

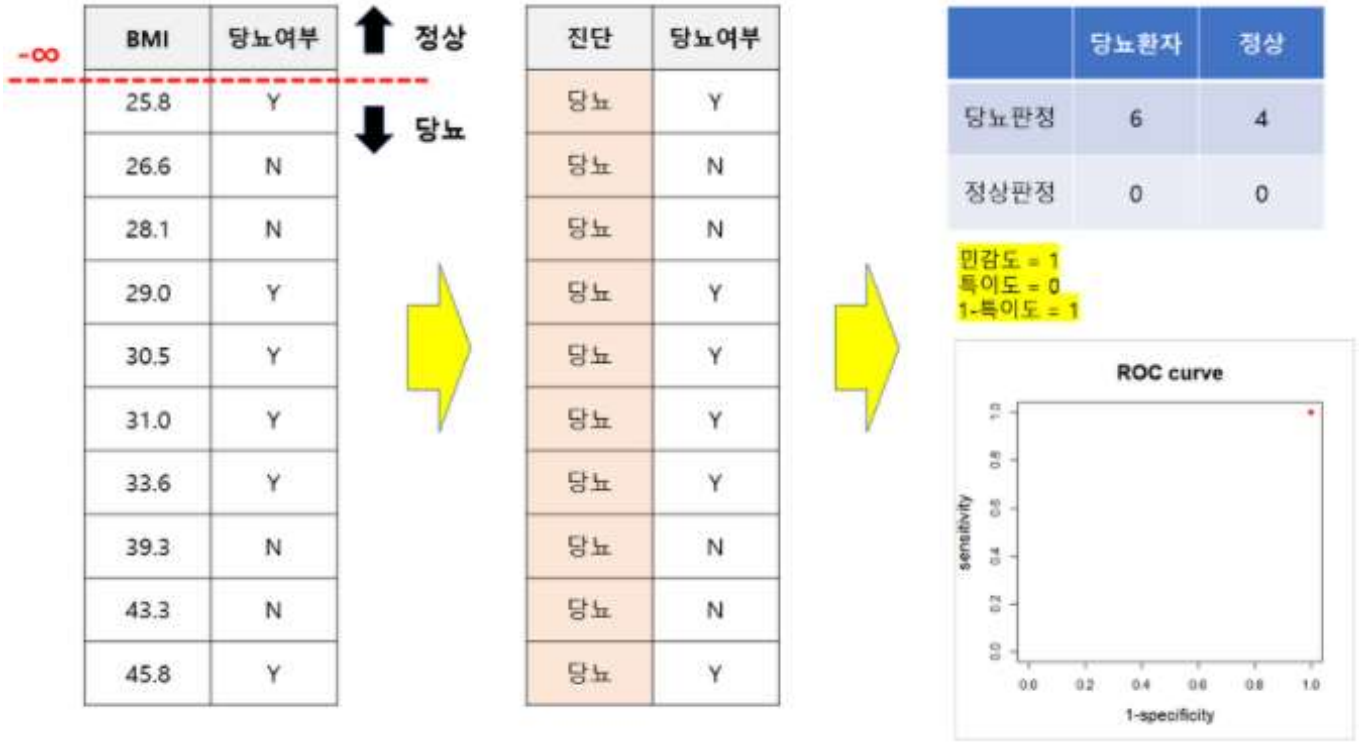
- **ROC (Receive Operating Characteristic) Curve**
  - » 여러 임계값들을 기준으로 Recall-Fallout의 변화를 시각화한 것
  - » Fallout은 실제 False인 data 중에서 모델이 True로 분류
  - » Recall은 실제 True인 data 중에서 모델이 True로 분류한 비율을 나타낸 지표
  - » 이 두 지표를 각각 x, y의 축으로 놓고 그려지는 그래프를 해석
  - » curve가 왼쪽 위 모서리에 가까울수록 모델의 성능이 좋다고 평가
    - Recall이 크고 Fall-out이 작은 모형이 좋은 모형



◆ 모델 성능 평가 지표

– **ROC (Receive Operating Characteristic) Curve 생성**

» 여러 임계값들을 기준으로 Recall-Fallout의 변화를 시각화한 것



◆ 모델 성능 평가 지표

– **ROC (Receive Operating Characteristic) Curve 생성**

» 여러 임계값들을 기준으로 Recall-Fallout의 변화를 시각화한 것

BMI	당뇨여부
25.8	Y
26.6	N
28.1	N
29.0	Y
30.5	Y
31.0	Y
33.6	Y
39.3	N
43.3	N
45.8	Y

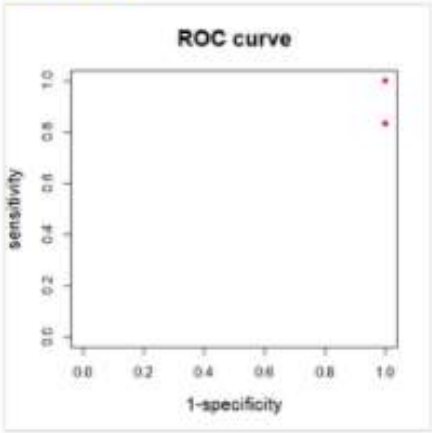
정상

당뇨

진단	당뇨여부
정상	Y
당뇨	N
당뇨	N
당뇨	Y
당뇨	Y
당뇨	Y
당뇨	Y
당뇨	N
당뇨	N
당뇨	Y

	당뇨환자	정상
당뇨판정	5	4
정상판정	1	0

민감도 = 5/6
특이도 = 0
1-특이도 = 1



◆ 모델 성능 평가 지표

– **ROC (Receive Operating Characteristic) Curve 생성**

» 여러 임계값들을 기준으로 Recall-Fallout의 변화를 시각화한 것

BMI	당뇨여부
25.8	Y
26.6	N
28.1	N
29.0	Y
30.5	Y
31.0	Y
33.6	Y
39.3	N
43.3	N
45.8	Y

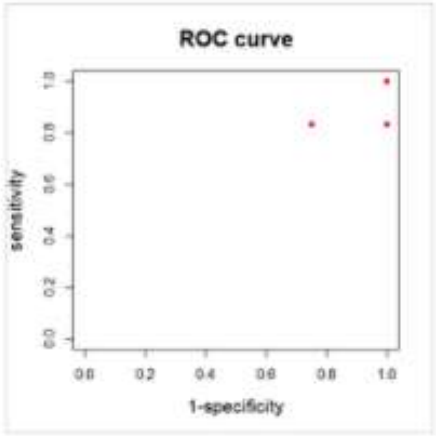
정상

당뇨

진단	당뇨여부
정상	Y
정상	N
당뇨	N
당뇨	Y
당뇨	Y
당뇨	Y
당뇨	Y
당뇨	N
당뇨	N
당뇨	Y

	당뇨환자	정상
당뇨판정	5	3
정상판정	1	1

민감도 = 5/6
특이도 = 1/4
1-특이도 = 3/4



◆ 모델 성능 평가 지표

– **ROC (Receive Operating Characteristic) Curve 생성**

» 여러 임계값들을 기준으로 Recall-Fallout의 변화를 시각화한 것

BMI	당뇨여부
25.8	Y
26.6	N
28.1	N
29.0	Y
30.5	Y
31.0	Y
33.6	Y
39.3	N
43.3	N
45.8	Y

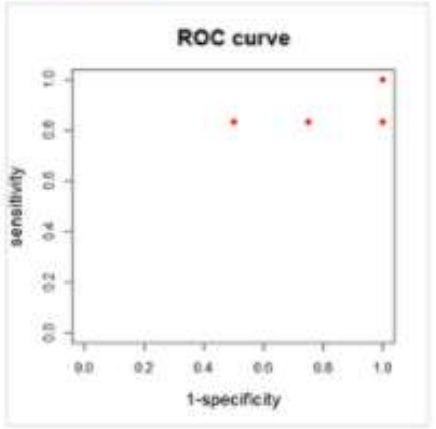
정상

당뇨

진단	당뇨여부
정상	Y
정상	N
정상	N
당뇨	Y
당뇨	Y
당뇨	Y
당뇨	Y
당뇨	N
당뇨	N
당뇨	Y

	당뇨환자	정상
당뇨판정	5	2
정상판정	1	2

민감도 = 5/6
특이도 = 2/4
1-특이도 = 2/4



◆ 모델 성능 평가 지표

– **ROC (Receive Operating Characteristic) Curve 생성**

» 여러 임계값들을 기준으로 Recall-Fallout의 변화를 시각화한 것

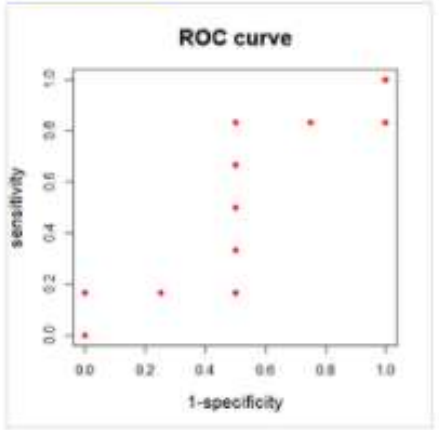
BMI	당뇨여부
25.8	Y
26.6	N
28.1	N
29.0	Y
30.5	Y
31.0	Y
33.6	Y
39.3	N
43.3	N
45.8	Y

정상

진단	당뇨여부
정상	Y
정상	N
정상	N
정상	Y
정상	Y
정상	Y
정상	Y
정상	N
정상	N
정상	Y

	당뇨환자	정상
당뇨판정	0	0
정상판정	6	4

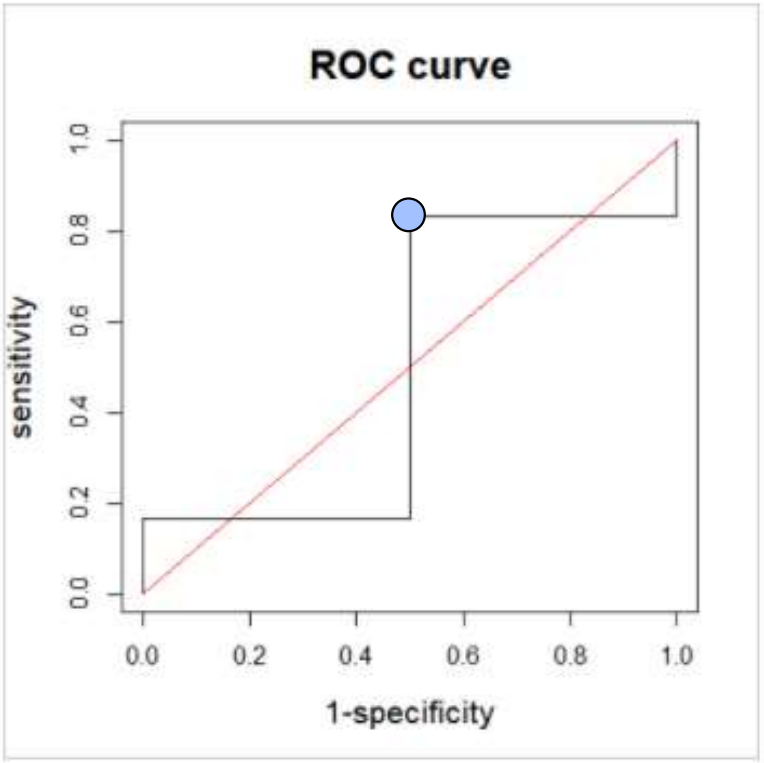
민감도 = 0/6
특이도 = 4/4
1-특이도 = 0/4



◆ 모델 성능 평가 지표

– **ROC (Receive Operating Characteristic) Curve 생성**

» 여러 임계값들을 기준으로 Recall-Fallout의 변화를 시각화한 것



BMI	당뇨여부
25.8	Y
26.6	N
28.1	N
29.0	Y
30.5	Y
31.0	Y
33.6	Y
39.3	N
43.3	N
45.8	Y

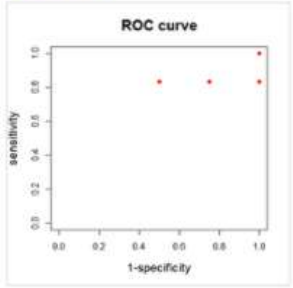
정상

당뇨

진단	당뇨여부
정상	Y
정상	N
정상	N
당뇨	Y
당뇨	Y
당뇨	Y
당뇨	Y
당뇨	N
당뇨	N
당뇨	Y

	당뇨환자	정상
당뇨판정	5	2
정상판정	1	2

민감도 = 5/6
특이도 = 2/4
1-특이도 = 2/4



28.1, 29.0 사이
평균: 28.55

Thank you & Good luck !