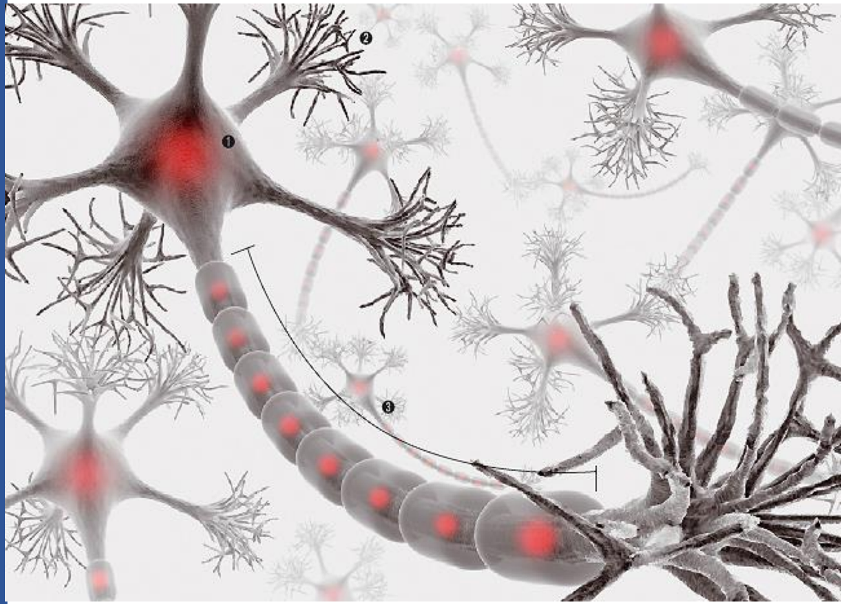
(CNN) 옷 이미지 분류

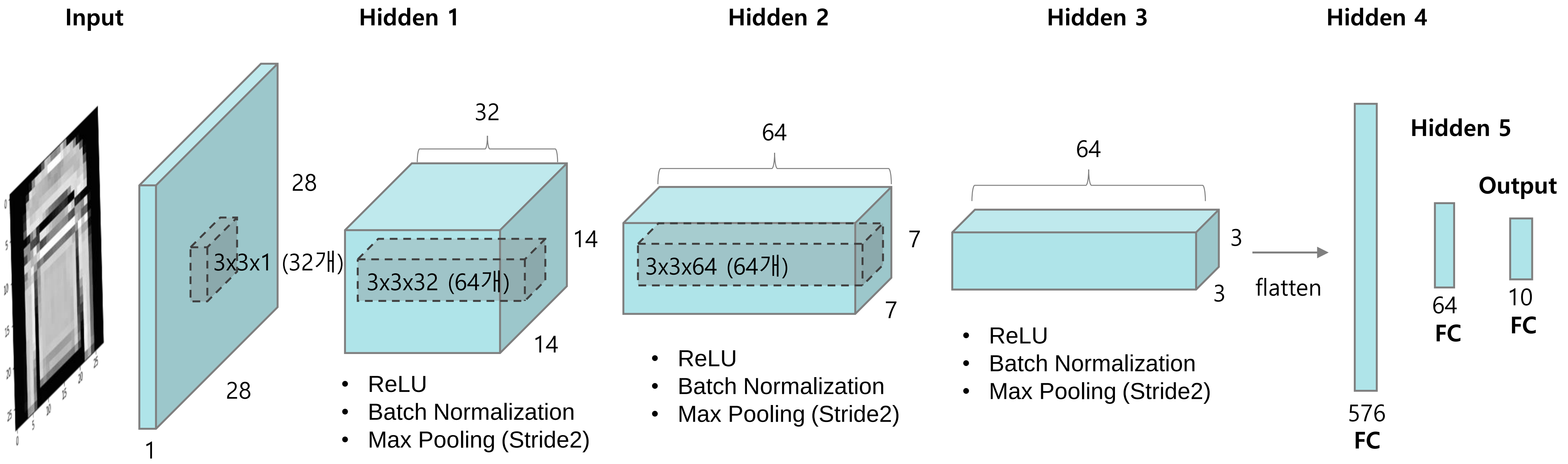
학습 목표

- 운동화나 셔츠 같은 옷 이미지를 분류하는 CNN 신경망 모델을 만들어 본다.



문제

DNN으로 개발된 Fashion MNIST 모델을 다음의 CNN 모델을 바꿔보자!



- ReLU
- Batch Normalization
- Max Pooling (Stride2)

- ReLU
- Batch Normalization
- Max Pooling (Stride2)

- ReLU
- Batch Normalization
- Max Pooling (Stride2)

x는 28x28x1 Tensor

$$y = (y_1, y_2, y_3, \ldots, y_{10})$$

0에서 9까지의 클래스에 속할 확률

Hint : 데이터셋 로드

```
fashion_mnist = keras.datasets.fashion_mnist

(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```

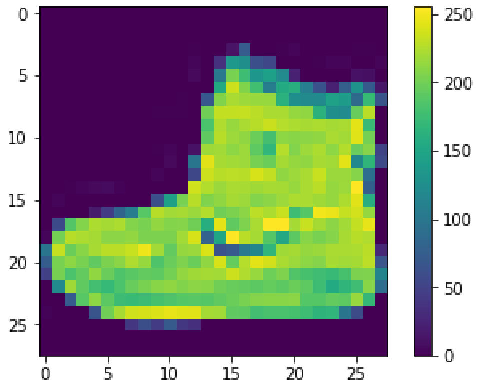
이미지를 (28,28,1) Tensor 타입으로 변환

```
train_images = train_images.reshape(train_images.shape[0], 28, 28, 1).astype('float32')
test_images = test_images.reshape(test_images.shape[0], 28, 28, 1).astype('float32')
```

Hint : 데이터 전처리

이미지 출력하는 곳 모두 수정

```
plt.figure()
plt.imshow(np.reshape(train_images[0],(28,28)))
plt.grid(False)
plt.show()
```

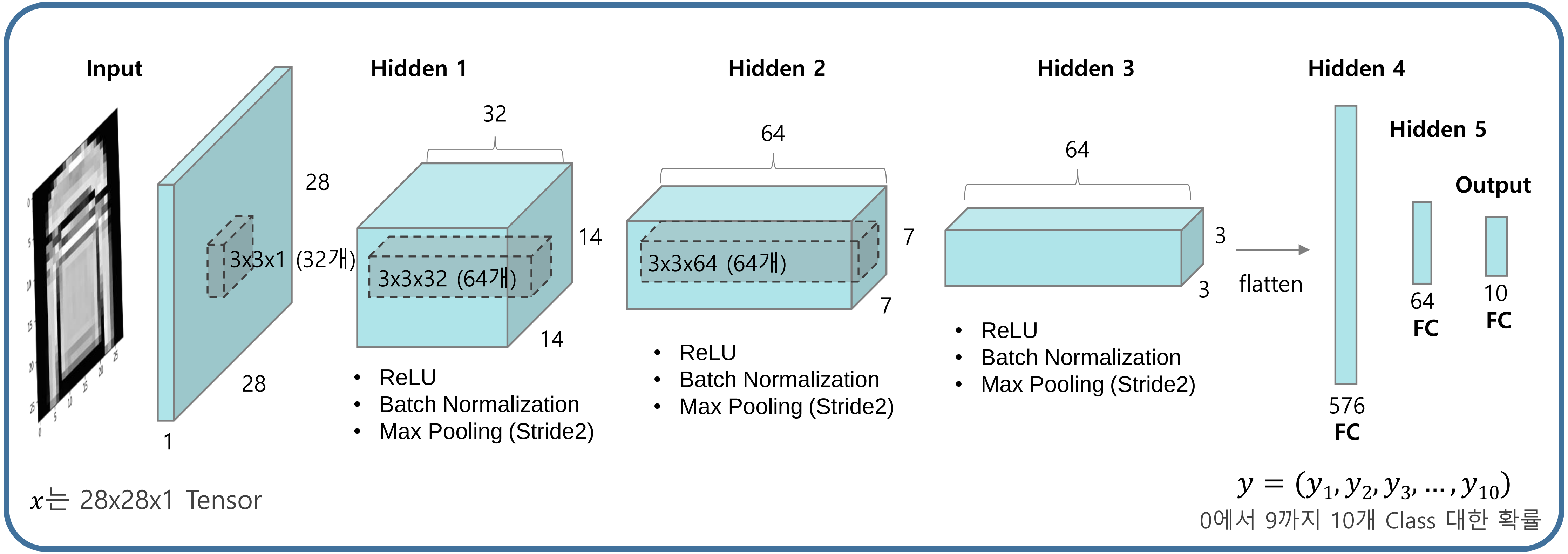


픽셀 값의 범위가 0~255 사이

모델 정의 (문제)

keras.layers.Conv2D, MaxPooling2D, Flatten, Dense를 사용해서 모델을 구성하시오.
단, 각 계층의 activation은 relu로 구성

```
model = keras.Sequential(#your code)
```



- ReLU
- Batch Normalization
- Max Pooling (Stride2)

- ReLU
- Batch Normalization
- Max Pooling (Stride2)

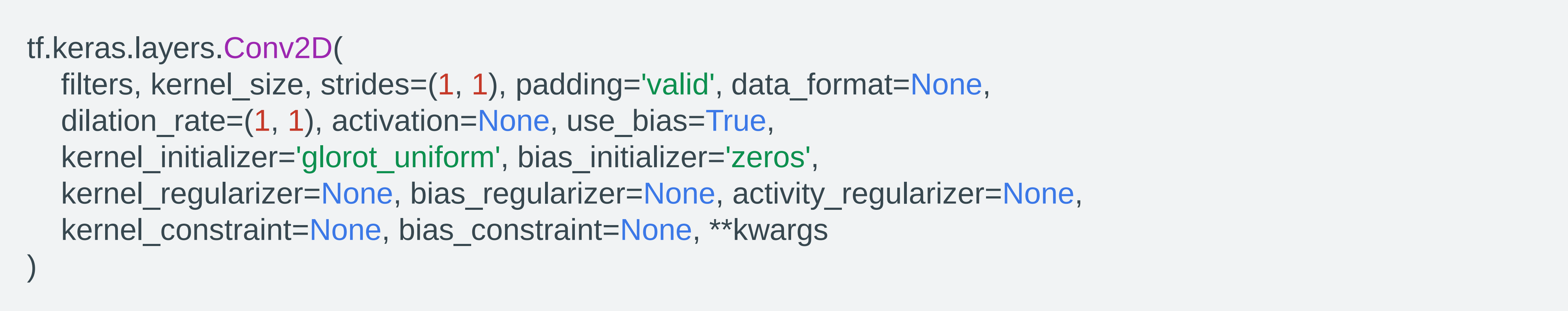
- ReLU
- Batch Normalization
- Max Pooling (Stride2)

x는 28x28x1 Tensor

$y = (y_1, y_2, y_3, \ldots, y_{10})$

0에서 9까지 10개 Class 대한 확률

참고 tf.keras.layers.Conv2D



```
tf.keras.layers.Conv2D(
    filters, kernel_size, strides=(1, 1), padding='valid', data_format=None,
    dilation_rate=(1, 1), activation=None, use_bias=True,
    kernel_initializer='glorot_uniform', bias_initializer='zeros',
    kernel_regularizer=None, bias_regularizer=None, activity_regularizer=None,
    kernel_constraint=None, bias_constraint=None, **kwargs
)
```

- **filters**: Convolution Filter 개수 the dimensionality of the output space (i.e. the number of output filters in the convolution).
- **kernel_size**: Convolution Filter 크기, An integer or tuple/list of 2 integers, specifying the height and width of the 2D convolution window. Can be a single integer to specify the same value for all spatial dimensions.
- **strides**: An integer or tuple/list of 2 integers, specifying the strides of the convolution along the height and width. Can be a single integer to specify the same value for all spatial dimensions. Specifying any stride value != 1 is incompatible with specifying any dilation_rate value != 1.
- **padding**: one of "valid" or "same" (case-insensitive).
- **activation**: Activation function to use. If you don't specify anything, no activation is applied (ie. "linear" activation: a(x) = x).
- **kernel_initializer**: Initializer for the kernel weights matrix.

https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv2D

참고 tf.keras.layers.BatchNormalization

```
tf.keras.layers.BatchNormalization(
    axis=-1, momentum=0.99, epsilon=0.001, center=True, scale=True,
    beta_initializer='zeros', gamma_initializer='ones',
    moving_mean_initializer='zeros', moving_variance_initializer='ones',
    beta_regularizer=None, gamma_regularizer=None, beta_constraint=None,
    gamma_constraint=None, renorm=False, renorm_clipping=None, renorm_momentum=0.99,
    fused=None, trainable=True, virtual_batch_size=None, adjustment=None, name=None,
    **kwargs
)
```

https://www.tensorflow.org/api_docs/python/tf/keras/layers/BatchNormalization

참고 tf.keras.layers.MaxPool2D

```
tf.keras.layers.MaxPool2D(
    pool_size=(2, 2), strides=None, padding='valid', data_format=None, **kwargs
)
```

- **pool_size**: integer or tuple of 2 integers, factors by which to downscale (vertical, horizontal). (2, 2) will halve the input in both spatial dimension. If only one integer is specified, the same window length will be used for both dimensions.
- **strides**: Integer, tuple of 2 integers, or None. Strides values. If None, it will default to pool_size.
- **padding**: One of "valid" or "same" (case-insensitive).
- **data_format**: A string, one of channels_last (default) or channels_first. The ordering of the dimensions in the inputs. channels_last corresponds to inputs with shape (batch, height, width, channels) while channels_first corresponds to inputs with shape (batch, channels, height, width). It defaults to the image_data_format value found in your Keras config file at ~/.keras/keras.json. If you never set it, then it will be "channels_last".

https://www.tensorflow.org/api_docs/python/tf/keras/layers/MaxPool2D

참고 : tf.keras.layers.MaxPool2D는 tf.keras.layers.MaxPooling2D로 호출해도 됨

Thank you!