

Machine Learning

Indah Agustien Siradjuddin

Convolutional Neural Network

Semester Gasal 2019-2020

Deep Learning memiliki arsitektur Jaringan Syaraf Tiruan yang memiliki lebih banyak layer dan neuron (*deeper* dan *wider*). Setiap layer memiliki fungsi tersendiri. Kelebihan dari Deep Learning :

- Arsitektur yang lebih banyak layer dan neuron, dapat menangkap data secara detail
- Tidak dibutuhkan *Feature engineering*, deep learning memiliki kemampuan *feature learning*

Convolutional Neural Network - CNN, merupakan salah satu arsitektur Deep learning.

Architecture of CNN :

Convolutional Layers :

- *Image Convolution : n kernels atau filters
- Setiap channel dari citra dikonvolusi dengan setiap kernel
- Jumlahkan semua hasil konvolusi : feature maps. Oleh karena itu, jika terdapat n kernel, maka jumlah *feature map* yang dihasilkan adalah n

Rectified Linear Units (RELU)

Fungsi aktivasi RELU activation digunakan agar semua nilai pada *feature maps* adalah positif :

$$f(x) = \max(0, x)$$

Subsampling/Pooling

digunakan untuk mengurangi ukuran *feature map* dengan cara menyatukan fitur-fitur yang dianggap memiliki nilai hampir sama (hanya ukuran, bukan jumlah *feature map*).

max pooling atau *average pooling* :



Fully Connected Layer

atau dikenal dengan *densed layer* . Output dari klasifikasi adalah output dari layer ini (FC)

Input dari FC layer adalah array 1D, oleh karena itu harus dilakukan *reshape* feature map.

Misalkan :

- Jumlah kelas adalah tiga
- output layer adalah *one hot encoding*, maka jumlah neuron pada output layer adalah tiga, yaitu [1 0 0], [0 1 0], and [0 0 1]

CNN dengan KERAS framework

In [1]:

```
from keras.datasets import mnist  
  
(X_trainOrg, y_trainOrg), (X_testOrg, y_testOrg) = mnist.load_data()
```

Using TensorFlow backend.

C:\Users\Indah Agustin\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:516: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint8 = np.dtype(["qint8", np.int8, 1])
```

C:\Users\Indah Agustin\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:517: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_quint8 = np.dtype(["quint8", np.uint8, 1])
```

C:\Users\Indah Agustin\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:518: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint16 = np.dtype(["qint16", np.int16, 1])
```

C:\Users\Indah Agustin\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:519: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_quint16 = np.dtype(["quint16", np.uint16, 1])
```

C:\Users\Indah Agustin\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:520: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint32 = np.dtype(["qint32", np.int32, 1])
```

C:\Users\Indah Agustin\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:525: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
np_resource = np.dtype(["resource", np.ubyte, 1])
```

C:\Users\Indah Agustin\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:541: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint8 = np.dtype(["qint8", np.int8, 1])
```

C:\Users\Indah Agustin\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:542: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_quint8 = np.dtype(["quint8", np.uint8, 1])
```

C:\Users\Indah Agustin\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:543: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint16 = np.dtype(["qint16", np.int16, 1])
```

C:\Users\Indah Agustin\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:544: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_quint16 = np.dtype(["quint16", np.uint16, 1])
```

C:\Users\Indah Agustin\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:545: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
_np_qint32 = np.dtype(["qint32", np.int32, 1])
```

C:\Users\Indah Agustin\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:550: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.

```
np_resource = np.dtype(["resource", np.ubyte, 1])
```


In [2]:

```
print(X_trainOrg.shape)  
print(X_trainOrg[0])
```

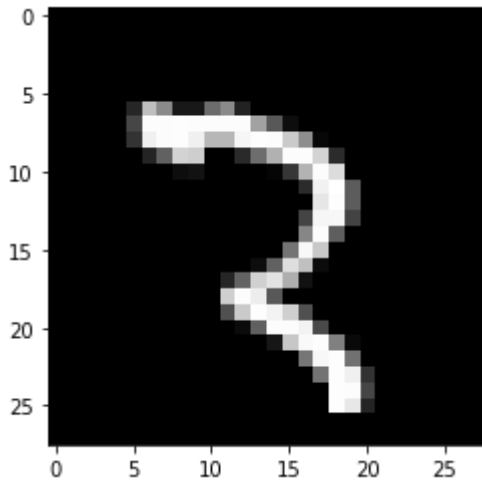
[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0]							
[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0]							
[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0]							
[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0]							
[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0]							
[0	0	0	0	0	0	0	0	0	0	0	0	3	18	18	18	126
	175	26	166	255	247	127	0	0	0	0]							136
[0	0	0	0	0	0	0	0	30	36	94	154	170	253	253	253	253
	225	172	253	242	195	64	0	0	0	0]							
[0	0	0	0	0	0	0	49	238	253	253	253	253	253	253	253	251
	93	82	82	56	39	0	0	0	0	0]							
[0	0	0	0	0	0	0	18	219	253	253	253	253	253	198	182	247
	0	0	0	0	0	0	0	0	0	0]							241
[0	0	0	0	0	0	0	0	80	156	107	253	253	205	11	0	43
	0	0	0	0	0	0	0	0	0	0]							154
[0	0	0	0	0	0	0	0	0	14	1	154	253	90	0	0	0
	0	0	0	0	0	0	0	0	0	0]							
[0	0	0	0	0	0	0	0	0	0	0	139	253	190	2	0	0
	0	0	0	0	0	0	0	0	0	0]							
[0	0	0	0	0	0	0	0	0	0	0	11	190	253	70	0	0
	0	0	0	0	0	0	0	0	0	0]							
[0	0	0	0	0	0	0	0	0	0	0	0	35	241	225	160	108
	0	0	0	0	0	0	0	0	0	0]							1
[0	0	0	0	0	0	0	0	0	0	0	0	0	0	81	240	253
	25	0	0	0	0	0	0	0	0	0]							119
[0	0	0	0	0	0	0	0	0	0	0	0	0	0	45	186	253
	150	27	0	0	0	0	0	0	0	0]							253
[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	93
	253	187	0	0	0	0	0	0	0	0]							252
[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	253	249	64	0	0	0	0	0	0	0]							249
[0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	130	183
	253	207	2	0	0	0	0	0	0	0]	</						

In [7]:

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.imshow(X_trainOrg[500],cmap='gray')
```

Out[7]:

<matplotlib.image.AxesImage at 0x1dbc984c588>



In [8]:

```
print(y_trainOrg[500])
```

3

In [9]:

```
from keras.utils import to_categorical
#one-hot encode target column
y_train = to_categorical(y_trainOrg)
y_test = to_categorical(y_testOrg)
```

In [12]:

```
print(y_train[500])
```

[0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]

In [48]:

```
#reshape data to fit model
X_train = X_trainOrg.reshape(60000,28,28,1)
X_test = X_testOrg.reshape(10000,28,28,1)
```


In [29]:

```
from keras.models import Sequential
from keras.layers import Dense, Conv2D, Flatten
#create model
model = Sequential()
#add model layers
model.add(Conv2D(64, kernel_size=3, activation='relu', input_shape=(28,28,1)))
model.add(Conv2D(32, kernel_size=3, activation='relu'))
model.add(Flatten())

model.add(Dense(10, activation='softmax')) #output
```

WARNING: Logging before flag parsing goes to stderr.

W1020 14:01:59.105266 6948 deprecation_wrapper.py:119] From C:\Users\Indah Agustin\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:74: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.

W1020 14:01:59.190231 6948 deprecation_wrapper.py:119] From C:\Users\Indah Agustin\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:517: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.

W1020 14:01:59.206512 6948 deprecation_wrapper.py:119] From C:\Users\Indah Agustin\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:4138: The name tf.random_uniform is deprecated. Please use tf.random.uniform instead.

In [30]:

```
#compile model using accuracy to measure model performance
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

W1020 14:02:11.719328 6948 deprecation_wrapper.py:119] From C:\Users\Indah Agustin\Anaconda3\lib\site-packages\keras\optimizers.py:790: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

W1020 14:02:11.752698 6948 deprecation_wrapper.py:119] From C:\Users\Indah Agustin\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3295: The name tf.log is deprecated. Please use tf.math.log instead.

In [31]:

```
#train the model
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=3)
```

W1020 14:02:37.197518 6948 deprecation.py:323] From C:\Users\Indah Agustina\Anaconda3\lib\site-packages\tensorflow\python\ops\math_grad.py:1250: add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.

Instructions for updating:

Use tf.where in 2.0, which has the same broadcast rule as np.where

W1020 14:02:37.257520 6948 deprecation_wrapper.py:119] From C:\Users\Indah Agustina\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:986: The name tf.assign_add is deprecated. Please use tf.compat.v1.assign_add instead.

Train on 60000 samples, validate on 10000 samples

Epoch 1/3

60000/60000 [=====] - 111s 2ms/step - loss: 14.46

92 - acc: 0.1022 - val_loss: 14.4902 - val_acc: 0.1010

Epoch 2/3

60000/60000 [=====] - 115s 2ms/step - loss: 14.47

11 - acc: 0.1022 - val_loss: 14.4902 - val_acc: 0.1010

Epoch 3/3

60000/60000 [=====] - 117s 2ms/step - loss: 14.47

11 - acc: 0.1022 - val_loss: 14.4902 - val_acc: 0.1010

Out[31]:

<keras.callbacks.History at 0x19f5f30dfd0>

In [38]:

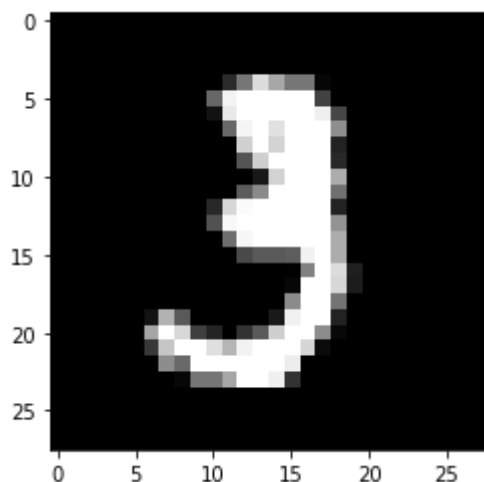
```
result=model.predict(X_test)
```

In [52]:

```
plt.imshow(X_trainOrg[10],cmap='gray')
```

Out[52]:

<matplotlib.image.AxesImage at 0x19f61fd1550>



In [53]:

```
print(result[10])
```

```
[0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
```