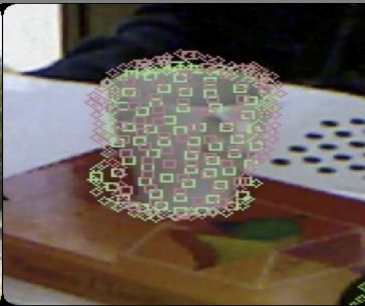
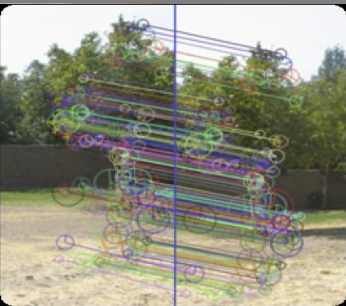


Deep Learning Basics

(#xx: Keras-based Neural Network Practice-Part2)



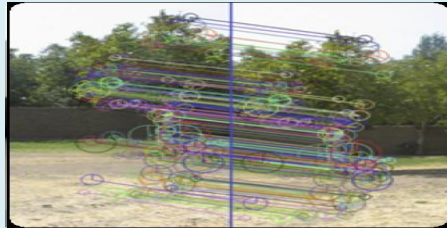
2023 Summer

Prof. Byung-Gyu Kim
Intelligent Vision Processing Lab. (IVPL)
<http://ivpl.sookmyung.ac.kr>

Dept. of IT Engineering, Sookmyung Women's University
E-mail: bg.kim@ivpl.sookmyung.ac.kr

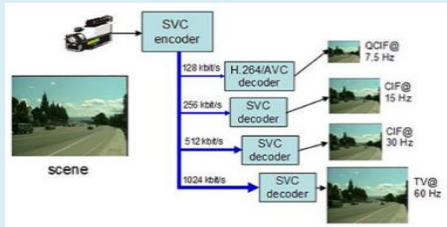
Gaol of this lecture

- ❖ Understand how to develop my Neural Network
 - Examples-based exercises



H.265 HEVC

High Efficiency Video Coding

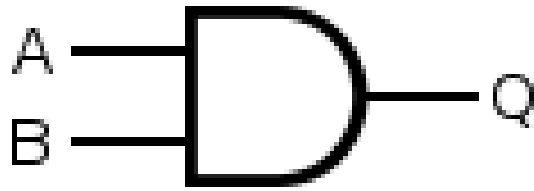


Contents

- AND logic implementation

Keras Neural Network Tutorial: AND logic (Digital logic)(1)

❖ AND logic

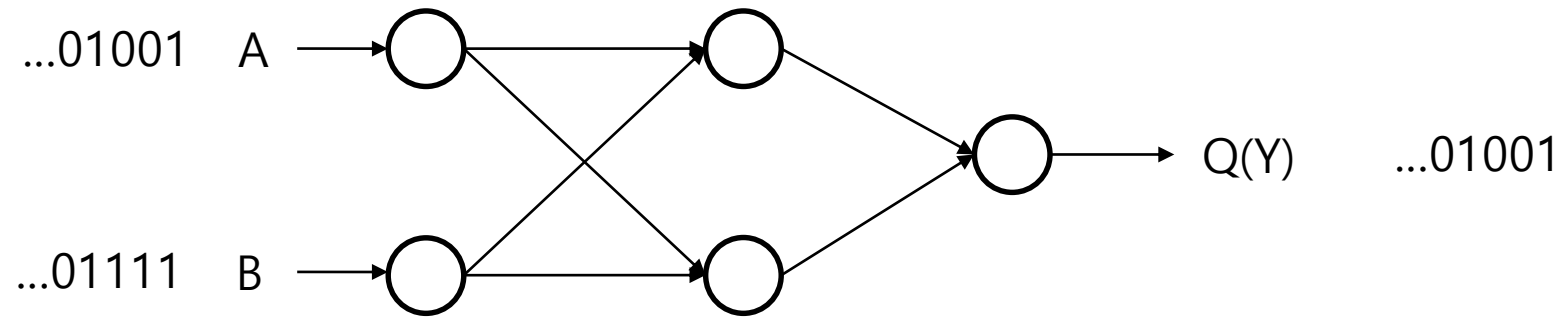


AND gate circuit

Truth Table

INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

❖ Network Structure



Keras Neural Network Tutorial: AND logic (Digital logic)(3)

❖ Training preparation

- Training data
 - Excel based CVS file with AND logic data (AND-gate.data.csv)

	A	B	C	D	E	F	G	H	I	J	K	L
198	1	1	1									
199	0	1	0									
200	1	1	1									
201	0	1	0									
202	0	0	0									
203	0	1	0									
204	1	1	1									
205	0	0	0									
206	1	0	0									
207	1	1	1									
208	0	0	0									
209	0	1	0									
210	1	1	1									
211	0	0	0									
212	1	0	0									
213	1	1	1									
214	0	0	0									
215	1	1	1									
216	1	1	1									
217	0	1	0									
218	0	1	0									
219	0	1	0									
220	1	1	1									
221	0	0	0									
222	0	1	0									
223	1	1	1									
224	1	0	0									
225	1	1	1									
226	0	0	0									
227	0	1	0									
228	0	0	0									
229	1	0	0									
230	0	1	0									
231	1	1	1									

Keras Neural Network Tutorial: AND logic (Digital logic)(4)

- Test data
 - Excel based CVS file with AND logic data (AND-gate-test.data.csv)

	A	B	C	D	E	F	G	H	I	J	K
1	0	0									
2	1	1									
3	1	1									
4	0	0									
5	0	0									
6	0	1									
7	1	1									
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											
31											
32											
33											
34											

A/B 열: input → output from Neural Network

Keras Neural Network Tutorial: AND logic (Digital logic)(5)

❖ Neural Network Implementation (editor: Visual Studio)

```
##-- AND logic with Neural Network (Multilayer
perceptron)
##-- Created by Prof. Kim Byung-Gyu on 23 July,
2019
from keras.models import Sequential
from keras.layers import Dense
import matplotlib.pyplot as plt
import numpy
##-- fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)
##-- load pima indians dataset
dataset = numpy.loadtxt("AND-gate.data.csv",
delimiter=",")
##-- split into input (X) and output (Y) variables
X = dataset[:,0:2]
Y = dataset[:,2]

(continue)
```

```
##-- create model
#sequential network by adding
model = Sequential()
# input layer
model.add(Dense(2, input_dim=2,
kernel_initializer='uniform', activation='sigmoid'))
# hidden layer
model.add(Dense(2, kernel_initializer='uniform',
activation='sigmoid'))
# output layer
model.add(Dense(1, kernel_initializer='uniform',
activation='sigmoid'))
##-- Compile model
model.compile(loss='mean_squared_error', optimizer='adam',
metrics=['accuracy'])
##-- Fit the model
history = model.fit(X, Y, validation_split=0.2, epochs=200,
batch_size=1, verbose=0)

(continue)
```


Keras Neural Network Tutorial: AND logic (Digital logic)(6)

❖ Neural Network Implementation (editor: Visual Studio)

```
##-- summarize history for accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
##-- summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
        (continue)
```

```
##-- Actual test for the trained model --##
dataset_test = numpy.loadtxt("AND-gate-test.data.csv",
delimiter=",")
x_test = dataset_test[:,0:2]
yhat = model.predict(x_test)

print('#-- X_tested --#')
print(x_test)
print('#-- Y_predicted --#')
print(yhat)
```

Keras Neural Network Tutorial: AND logic (Digital logic)(7)

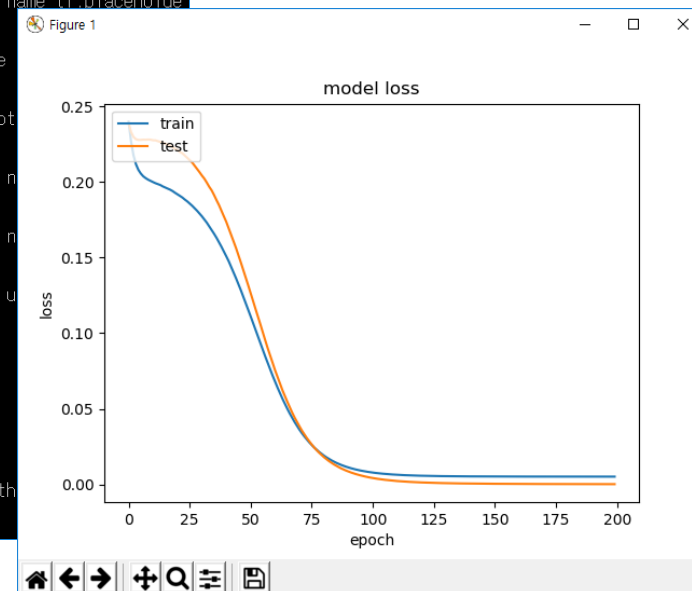
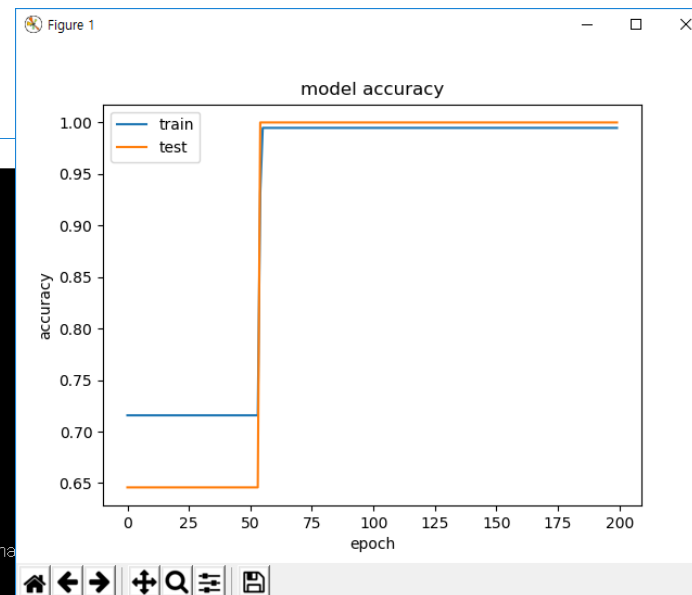
❖ Run the training and test the trained model

```
(BGKim) C:\Users\Wvicl\practices>python cv_keras_ANDGate.py
```

```
명령 프롬프트 - python cv_keras_ANDGate.py
[[0. 0.]
 [1. 1.]
 [1. 1.]
 [0. 0.]
 [0. 0.]
 [0. 1.]
 [1. 1.]]
#-- Y_predicted --#
[[9.5543265e-04]
 [9.7633725e-01]
 [9.7633725e-01]
 [9.5543265e-04]
 [9.5541775e-04]
 [1.8948067e-03]
 [9.7633731e-01]]

(BGKim) C:\Users\Wvicl\practices>python cv_keras_ANDGate.py
Using TensorFlow backend.
WARNING: Logging before flag parsing goes to stderr.
W0723 17:27:07.437990 11840 deprecation_wrapper.py:119] From C:\ProgramData\Anaconda3\envs\BGKim\lib\site-packages\keras\backend\tensorflow_backend.py:74: The name tf.Graph is deprecated. Please use tf.compat.v1.get_default_graph instead.
W0723 17:27:07.452948 11840 deprecation_wrapper.py:119] From C:\ProgramData\Anaconda3\envs\BGKim\lib\site-packages\keras\backend\tensorflow_backend.py:517: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.
W0723 17:27:07.453917 11840 deprecation_wrapper.py:119] From C:\ProgramData\Anaconda3\envs\BGKim\lib\site-packages\keras\backend\tensorflow_backend.py:4138: The name tf.random_uniform is deprecated. Please use tf.random.uniform instead.
W0723 17:27:07.477852 11840 deprecation_wrapper.py:119] From C:\ProgramData\Anaconda3\envs\BGKim\lib\site-packages\keras\optimizers.py:790: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.
W0723 17:27:07.553680 11840 deprecation_wrapper.py:119] From C:\ProgramData\Anaconda3\envs\BGKim\lib\site-packages\keras\backend\tensorflow_backend.py:986: The name tf.assign is deprecated. Please use tf.compat.v1.assign_add instead.
W0723 17:27:07.603546 11840 deprecation_wrapper.py:119] From C:\ProgramData\Anaconda3\envs\BGKim\lib\site-packages\keras\backend\tensorflow_backend.py:973: The name tf.assign is deprecated. Please use tf.compat.v1.assign instead.
2019-07-23 17:27:07.650232: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use
2019-07-23 17:27:07.657425: I tensorflow/stream_executor/platform/default/dso_loader.cc:42] Successfully opened dynamic library nvcuda.dll
2019-07-23 17:27:07.784575: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1640] Found device 0 with properties:
name: GeForce GTX 1070 Ti major: 6 minor: 1 memoryClockRate(GHz): 1.683
pciBusID: 0000:01:00:0
2019-07-23 17:27:07.792034: I tensorflow/stream_executor/platform/default/dlopen_checker_stub.cc:25] GPU libraries are statically linked, skip dlopen check.
2019-07-23 17:27:07.801368: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1763] Adding visible gpu devices: 0
2019-07-23 17:27:08.367409: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1181] Device interconnect StreamExecutor with strength 1 edge matrix:
2019-07-23 17:27:08.372648: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1187]
2019-07-23 17:27:08.375352: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1200] 0: N
2019-07-23 17:27:08.384388: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1326] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with
-> physical GPU (device: 0, name: GeForce GTX 1070 Ti, pci bus id: 0000:01:00:0, compute capability: 6.1)
```

Training results



Keras Neural Network Tutorial: AND logic (Digital logic)(7)

Test result

```
2019-07-23 17:27:07.792034: I tensorflow/stream_executor/platform/default/dlopen_checker_stub.cc:25] GPU library
2019-07-23 17:27:07.801368: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1763] Adding visible gpu devices
2019-07-23 17:27:08.367409: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1181] Device interconnect Stream
2019-07-23 17:27:08.372648: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1187] 0
2019-07-23 17:27:08.375352: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1200] 0: N
2019-07-23 17:27:08.384388: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1326] Created TensorFlow device
-> physical GPU (device: 0, name: GeForce GTX 1070 Ti, pci bus id: 0000:01:00.0, compute capability: 6.1)
#-- X_tested --#
[[0. 0.]
 [1. 1.]
 [1. 1.]
 [0. 0.]
 [0. 0.]
 [0. 1.]
 [1. 1.]]
#-- y_predicted --#
[[9.5543265e-04]
 [9.7633725e-01]
 [9.7633725e-01]
 [9.5543265e-04]
 [9.5541775e-04]
 [1.8948087e-03]
 [9.7633731e-01]]
(BGKim) C:\Users\vicl\practices>
```

Test Inputs

Network outputs

❖ OR logic

This is your time to exercise by yourself...!!!!

❖ EXOR logic

This is your time to exercise by yourself...!!!!

Thank you for your attention!!!
QnA

<http://ivpl.sookmyung.ac.kr>