10-26-2018 Keras API Functions

1

```
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
```

2

```
model.add(layers.MaxPooling2D((2, 2)))
```

3

```
model.add(layers.Flatten())
```

4

```
model.add(layers.Dense(64, activation='relu'))
```

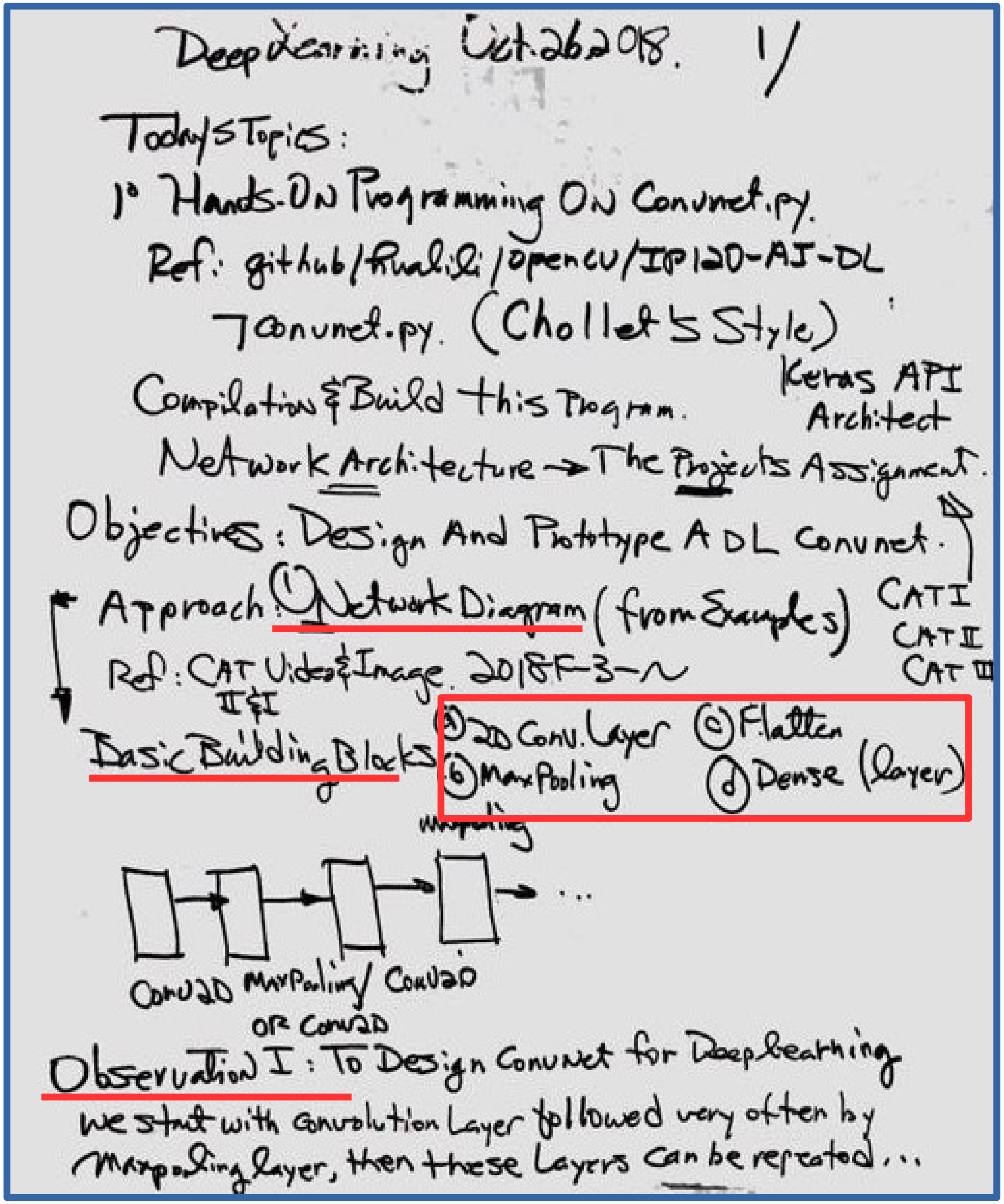
5

```
tf.keras.layers.Dropout(0.2)(fc_1)
```

6

```
model.summary()
```

10-26-2018 ConvNet Architecture Design



Deep Learning Oct.26 2018. 1/

Today's Topics:

1° Hands-On Programming On Convnet.py.

Ref: github/hualili/opencv/IP120-AI-DL

7convnet.py. (Chollet's Style)

Compilation & Build this Program.

Keras API Architect

Network Architecture → The Projects Assignment.

Objectives: Design And Prototype A DL Convnet.

Approach: ① Network Diagram (from Examples) CAT I CAT II CAT III

Ref: CAT Video & Image 2018F-3-~ II & I

Basic Building Blocks

ⓐ 2D Conv. Layer ⓒ Flatten

ⓑ MaxPooling ⓓ Dense (layer)

Observation I: To Design Convnet for Deep Learning we start with Convolution Layer followed very often by Maxpooling layer, then these Layers can be repeated...

Step 1. Repeating conv2D( ) layer and MaxPooling( ) layer

Example

1

```
model.add(layers.Conv2D(32, (3, 3),
activation='relu', input_shape=(28, 28, 1)))
```

2

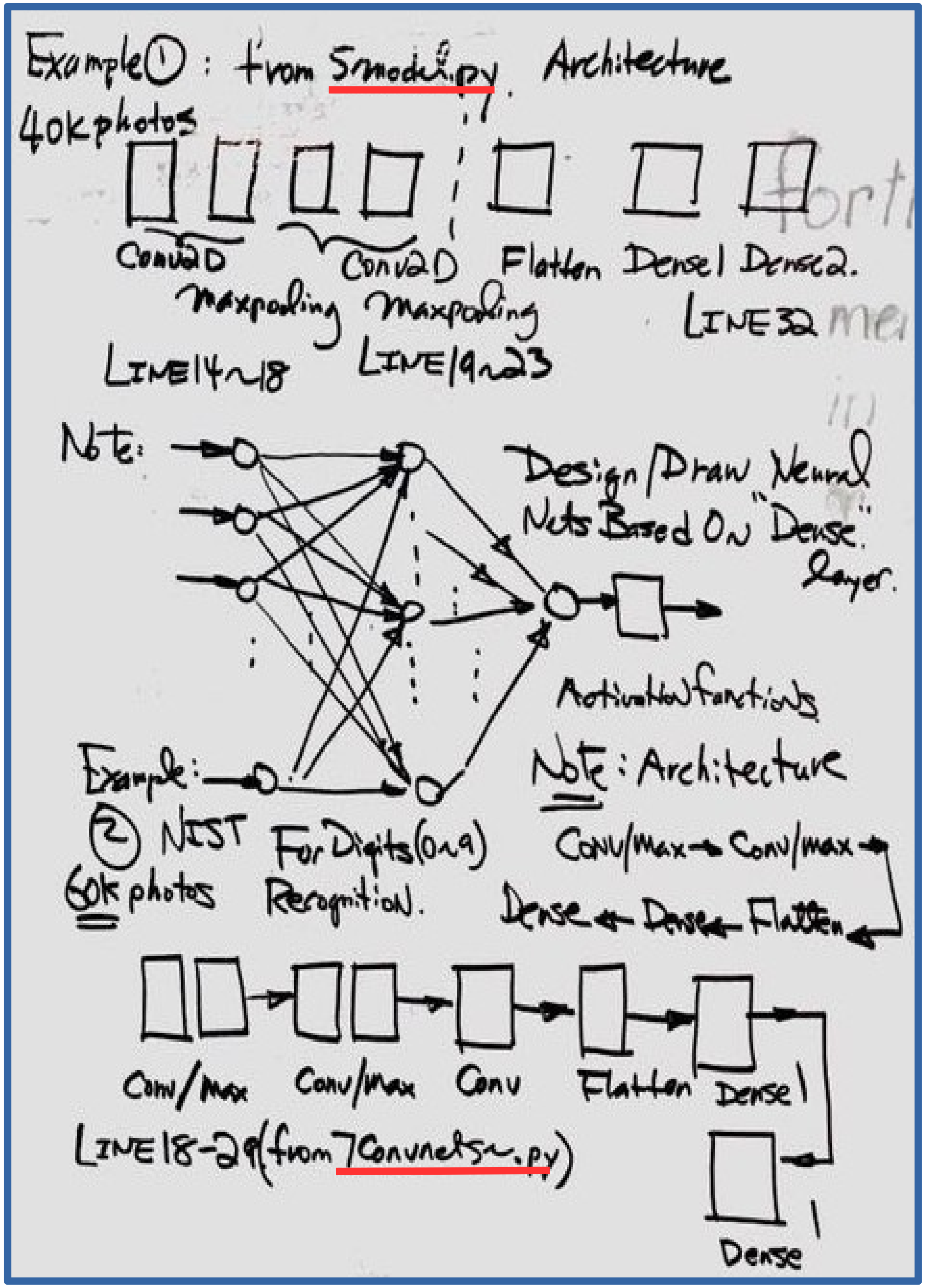
```
model.add(layers.MaxPooling2D((2, 2)))
```

6

```
model.summary()
```

To display the network architecture

10-26-2018 3 ConvNet Examples



Example① : from 5model.py Architecture

40k photos

ConvaD maxpooling ConvaD maxpooling Flatten Dense1 Dense2.

LINE14~18 LINE19~23 LINE32

Note: Design/Draw Neural Nets Based On "Dense" layer.

Activation functions

Example:

Note: Architecture

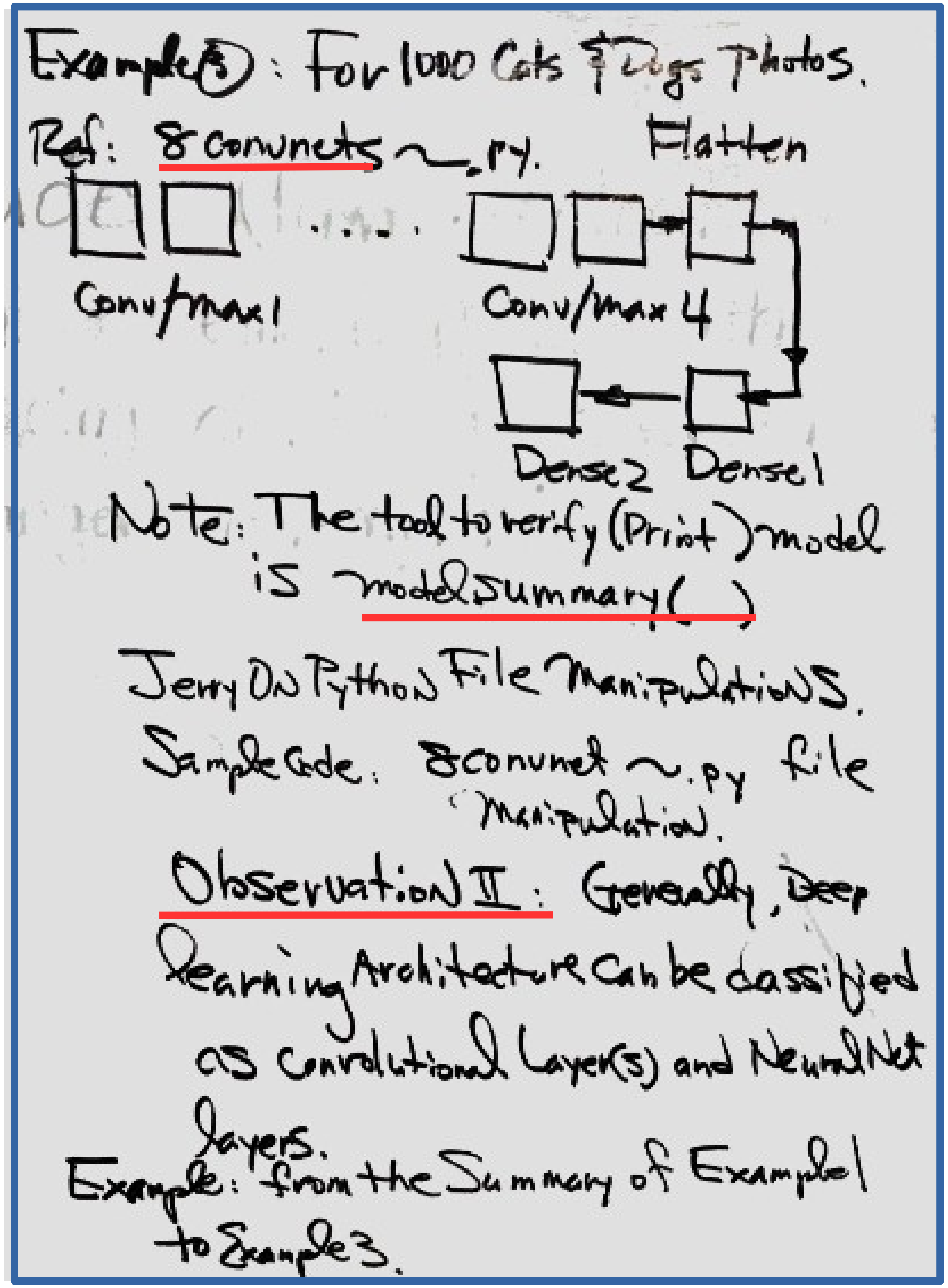
Conv/max → Conv/max → Flatten → Dense → Dense

② NIST For Digits (0~9) Recognition.

60k photos

Conv/max Conv/max Conv Flatten Dense1 Dense

LINE18-29 (from 7Convnets~.py)



Example③ : For 1000 Cats & Dogs Photos.

Ref: 8convnets~.py.

Conv/max1 Conv/max4 Flatten Dense1 Dense2

Note: The tool to verify (Print) model is model.summary( )

Jerry On Python File Manipulations.

Sample Code: 8convnet~.py file "manipulation.

Observation II: Generally, Deep learning Architecture can be classified as Convolutional Layer(s) and Neural Net layers.

Example: from the Summary of Example1 to Example3.

Harry Li, Ph.D. 2018

10-26-2018 Comparison of 3 ConvNets

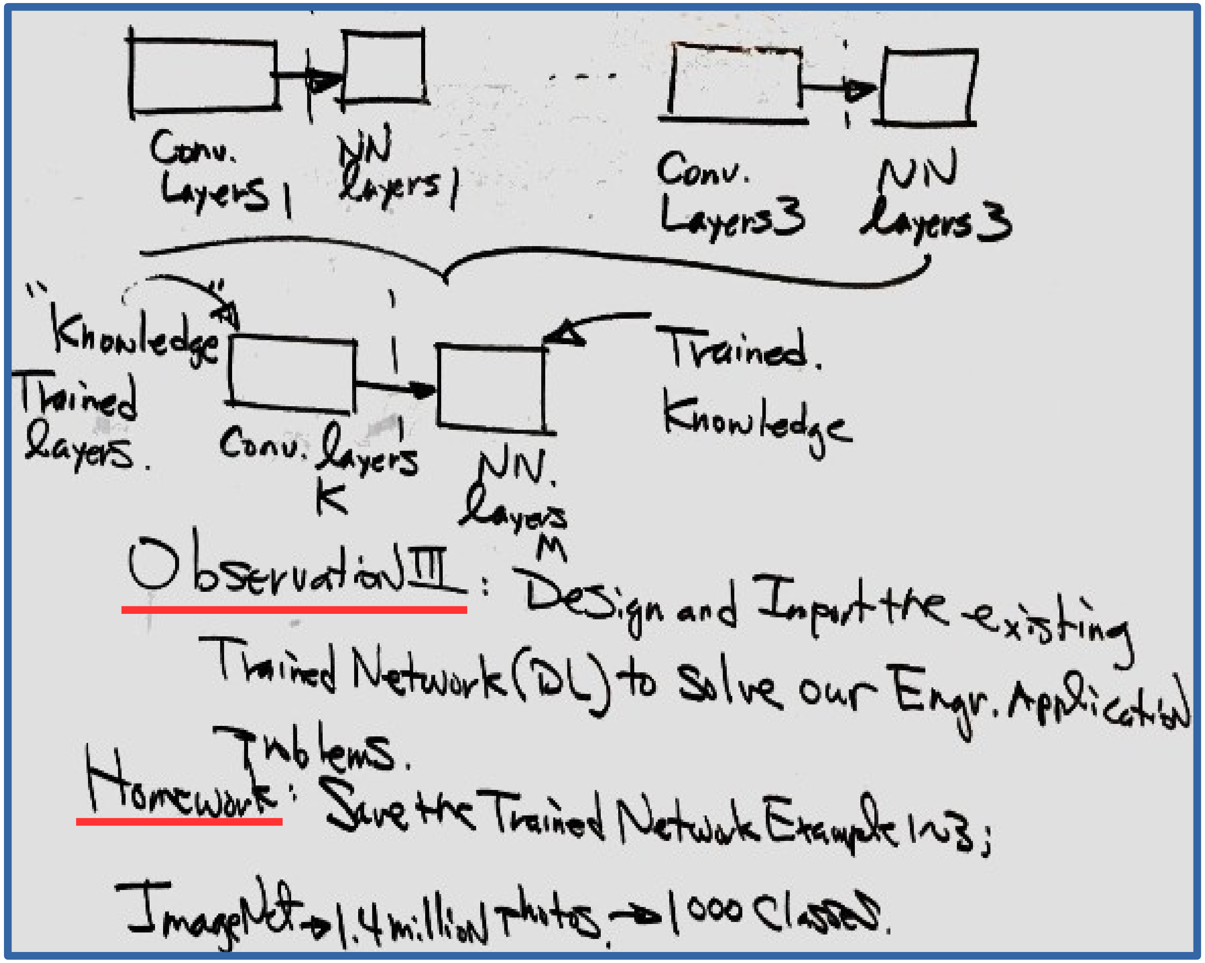
Table 1. Three well trained convnet examples

1 convnet1	Satellite imagery	40K images	ConvPool1+ConvPool2+Flatten+Den1+Den2
2 convnet2	NIST digits 0-9	60K images	ConvPool1+ConvPool2+Conv+Flatten+Den1+Den2
3 convnet3	Chollet cat-dog	1K images	ConvPool1+ConvPool2+ConvPool3 +ConvPool4 +Flatten+Den1+Den2

Table 2. Sample code for 3 well trained convnets

Network	Programs
convnet1	5model.py
convnet2	7convnets-NumeralDetection-ch05.py
convnet3	8convnets-SmallData-cats-dogs-ch05.py

10-26-2018 3 ConvNet Examples



Observation III: Design and Import the existing Trained Network (DL) to Solve our Engr. Application Problems.

Homework: Save the Trained Network Example 1~3;

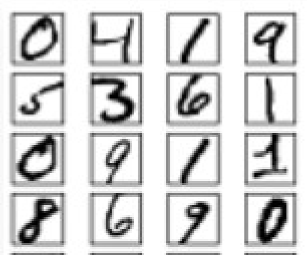
ImageNet → 1.4 million photos → 1000 classes.

10-26-2018 Dense Layer Output Design

Example: NIST 10 digits (0-9) convnet Dense layer

```
from keras import models
from keras import layers

network = models.Sequential()
network.add(layers.Dense(512, activation='relu', input_shape=(28 * 28,)))
network.add(layers.Dense(10, activation='softmax'))
```



Sample code :
https://github.com/fchollet/deep-learning-with-python-notebooks

The network consists of 2 Dense layers, densely-connected ("fully-connected") neural layers. The output layer is a 10-way "softmax" layer, it returns an array of 10 probability scores (summing to 1). Each score will be the probability that the current digit image belongs to one of our 10 digit classes.

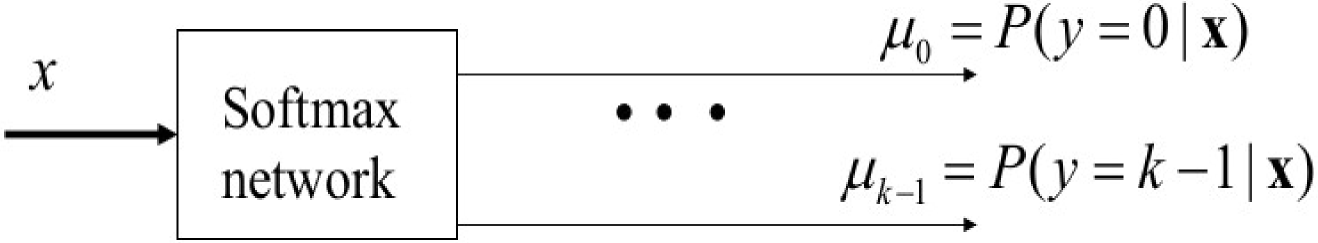
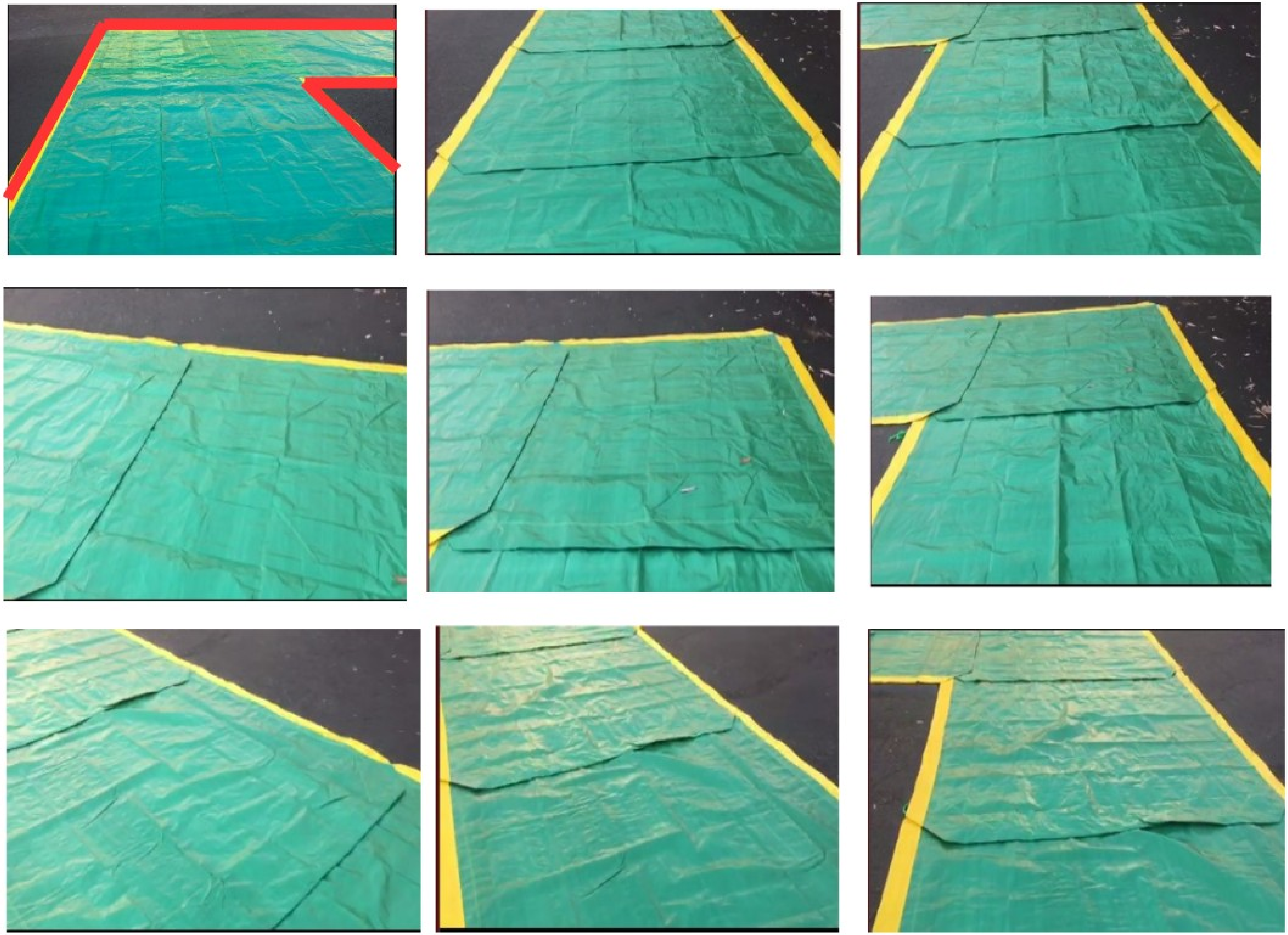
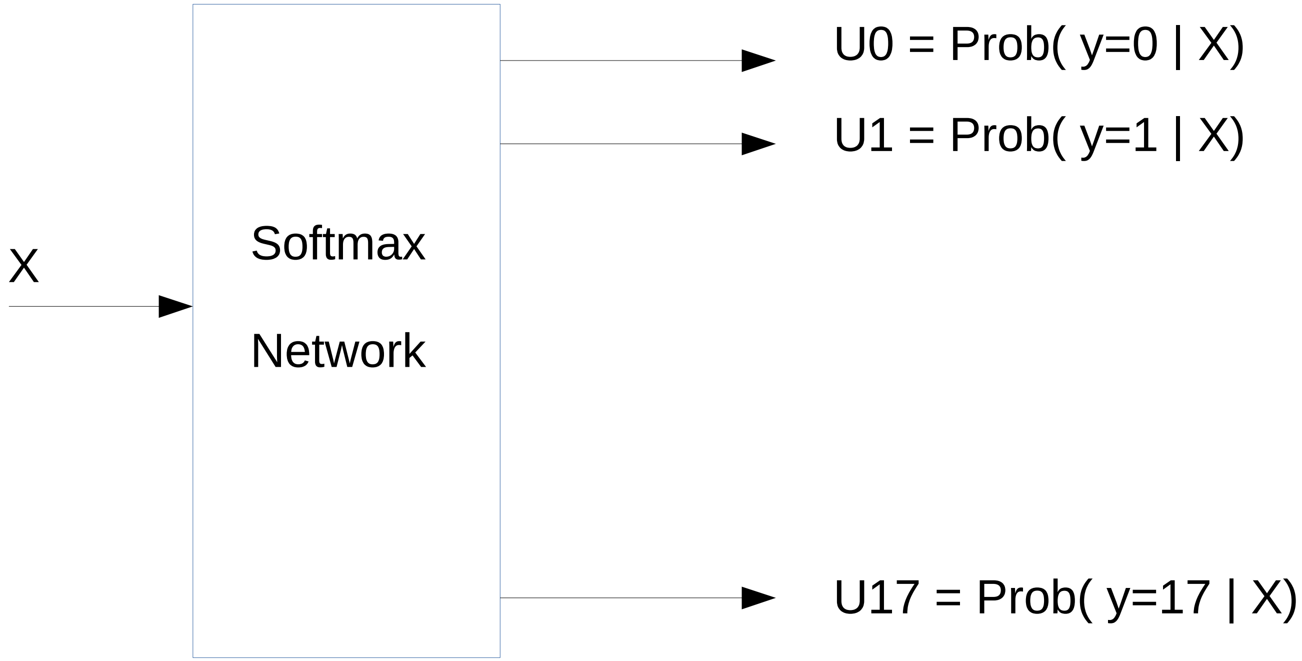

x
Softmax network
$\mu_0 = P(y = 0 \mid \mathbf{x})$
• • •
$\mu_{k-1} = P(y = k-1 \mid \mathbf{x})$

Illustration of the softmax block diagram from Milos Hauskrecht, milos@cs.pitt.edu,5329 Sennott Square

CAT-II Path Primitives



10-26-2018 Softmax Output Design for CATIII



X

Softmax

Network

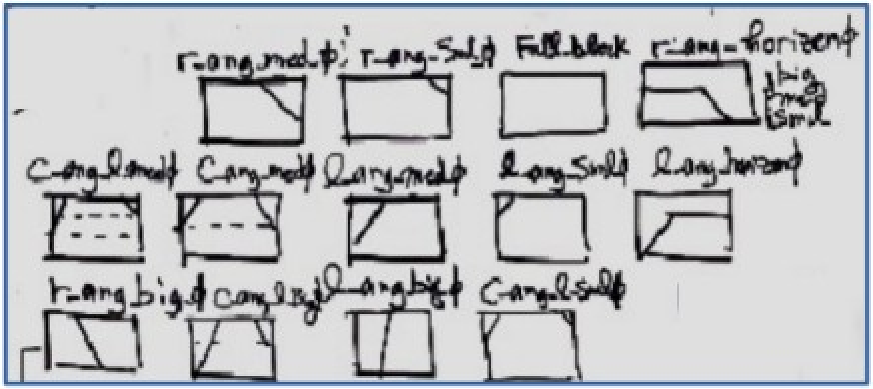
U0 = Prob(y=0 | X)

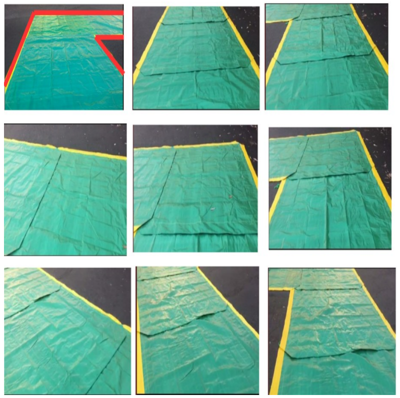
U1 = Prob(y=1 | X)

U17 = Prob(y=17 | X)

Where Prob(y=0 | X) + Prob(y=1 | X) + … + Prob(y=17 | X) = 1

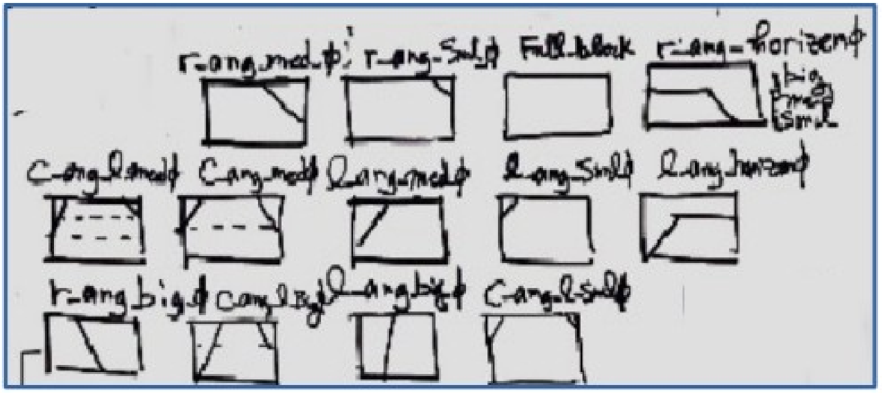
1. r_ang_big	right angle big
2. r_ang_med	right angle medium
3. r_ang_sml	right angle small
4. r_ang_hor	right angle horizon
5. l_ang_big	left angle big
6. l_ang_med	left angle medium
7. l_ang_sml	left angle small
8. l_ang_hor	left angle horizon
9. cr_ang_big	centeral-right big
10. cr_ang_med	central angle medium
11. cr_ang_sml	central angle small
12. cl_ang_big	centeral-right angle
13. cl_ang_med	central angle medium
14. cl_ang_sml	central angle small
15. c_ang_big	centeral-angle big
16. c_ang_med	central-ang medium
17. c_ang_sml	central-ang small
18. full	full block

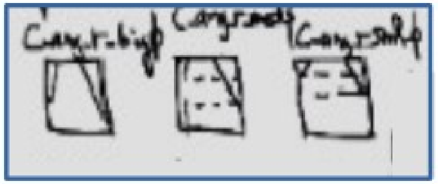




Harry Li, Ph.D. 2018

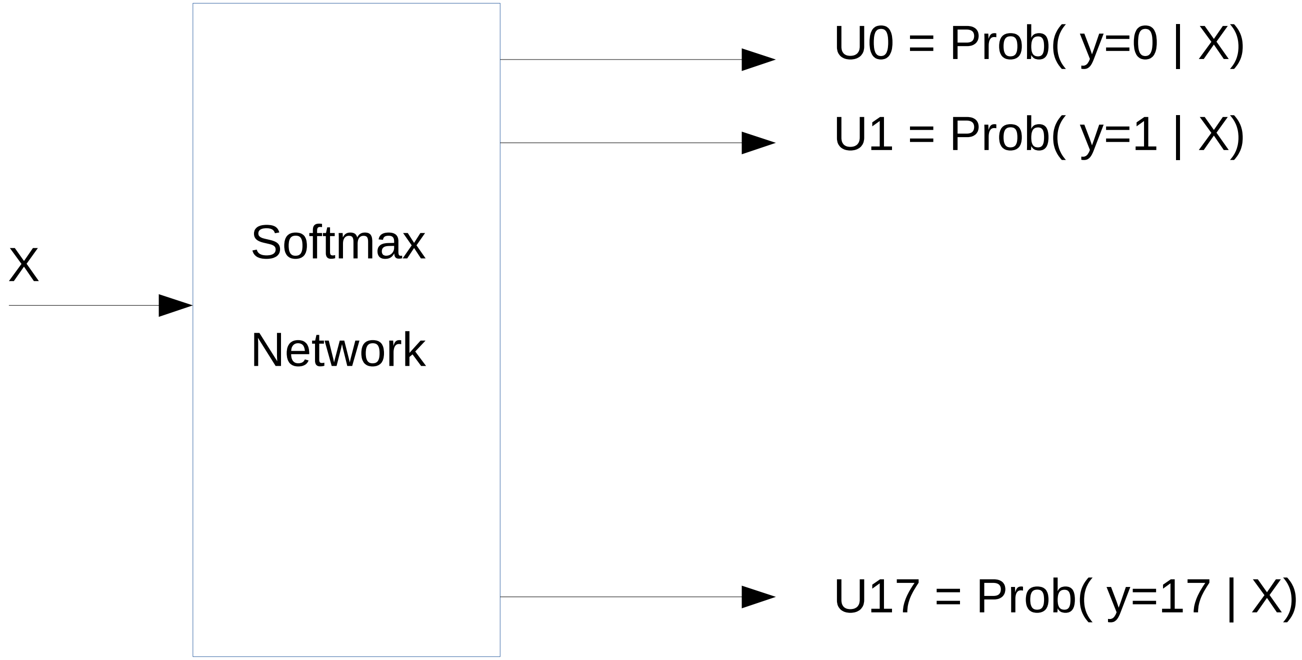
9-25-2018 Primitive Features for CAT-II Path Classification





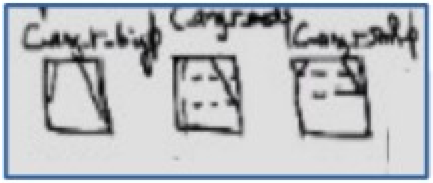
1. r_ang_big	right angle big
2. r_ang_med	right angle medium
3. r_ang_sml	right angle small
4. r_ang_hor	right angle horizon
5. l_ang_big	left angle big
6. l_ang_med	left angle medium
7. l_ang_sml	left angle small
8. l_ang_hor	left angle horizon
9. cr_ang_big	centeral-right big
10. cr_ang_med	central angle medium
11. cr_ang_sml	central angle small
12. cl_ang_big	centeral-right angle
13. cl_ang_med	central angle medium
14. cl_ang_sml	central angle small
15. c_ang_big	centeral-angle big
16. c_ang_med	central-ang medium
17. c_ang_sml	central-ang small
18. full	full block

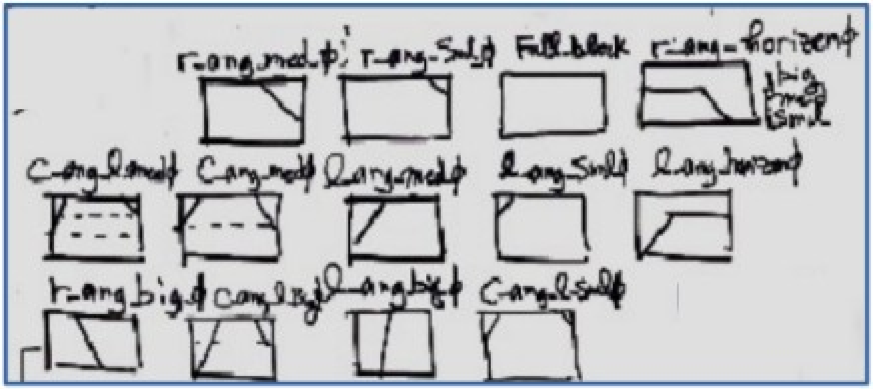
10-26-2018 Softmax Output Design for CATIII

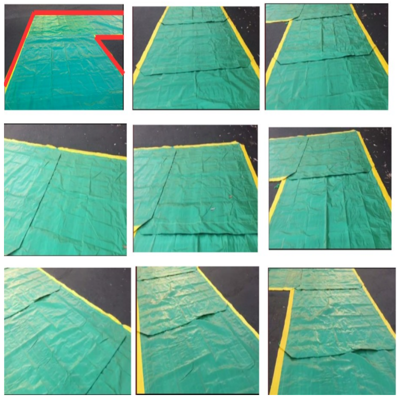


Where Prob(y=0 | X) + Prob(y=1 | X) + … + Prob(y=17 | X) = 1

1. r_ang_big	right angle big
2. r_ang_med	right angle medium
3. r_ang_sml	right angle small
4. r_ang_hor	right angle horizon
5. l_ang_big	left angle big
6. l_ang_med	left angle medium
7. l_ang_sml	left angle small
8. l_ang_hor	left angle horizon
9. cr_ang_big	centeral-right big
10. cr_ang_med	central angle medium
11. cr_ang_sml	central angle small
12. cl_ang_big	centeral-right angle
13. cl_ang_med	central angle medium
14. cl_ang_sml	central angle small
15. c_ang_big	centeral-angle big
16. c_ang_med	central-ang medium
17. c_ang_sml	central-ang small
18. full	full block

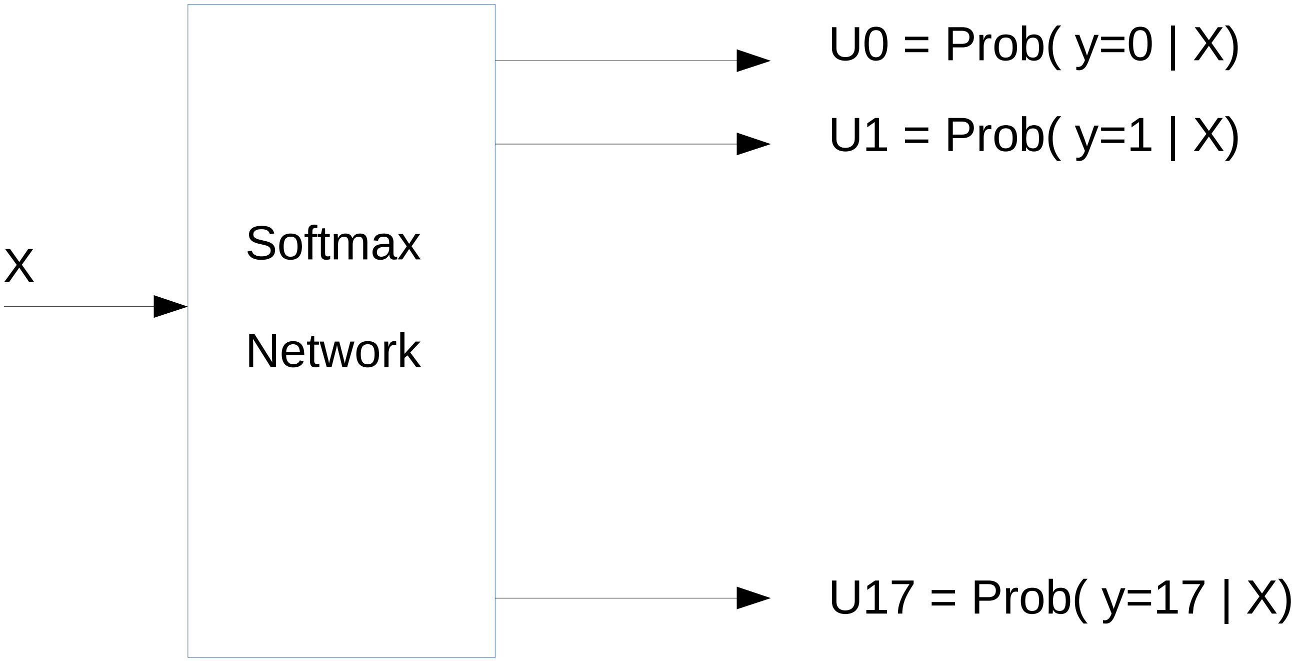






Harry Li, Ph.D. 2018

10-26-2018 Softmax Output Design for CATIII



X

Softmax

Network

U0 = Prob(y=0 | X)

U1 = Prob(y=1 | X)

U17 = Prob(y=17 | X)

Where Prob(y=0 | X) + Prob(y=1 | X) + … + Prob(y=17 | X) = 1