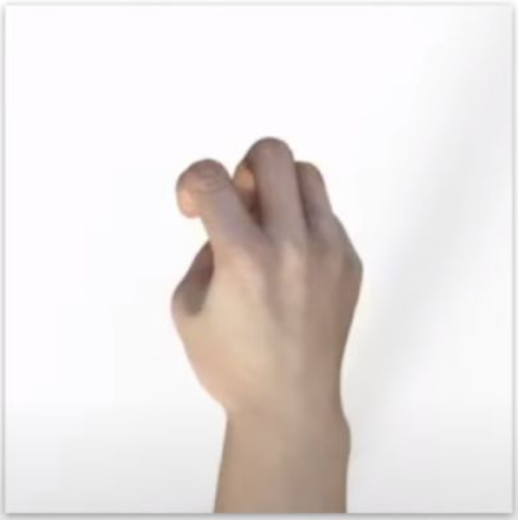
Gentle Introduction to Deep Learning

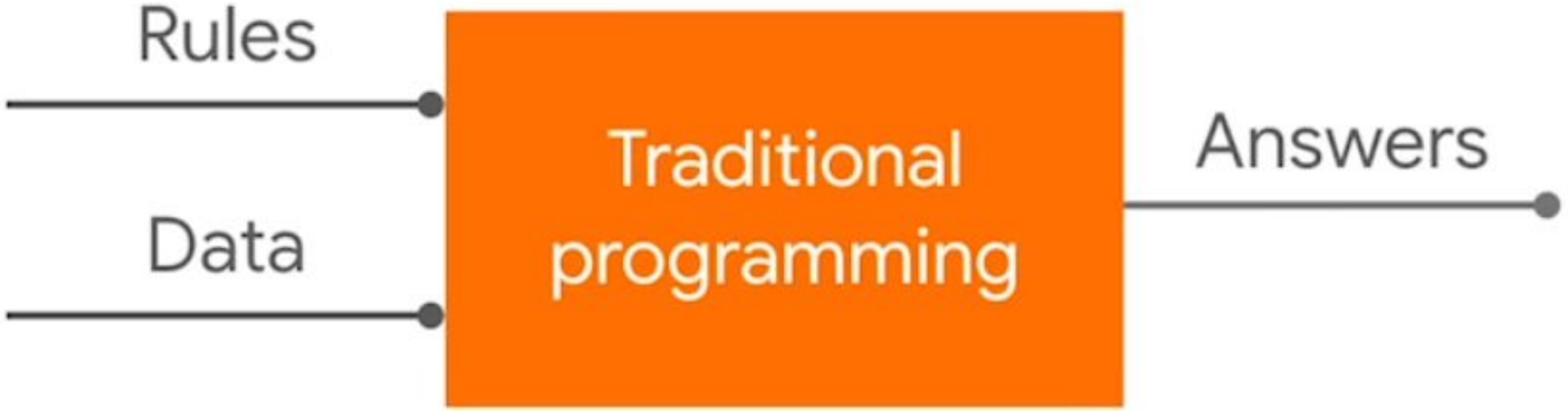
Nat Min Gyi@Thinkers Institute

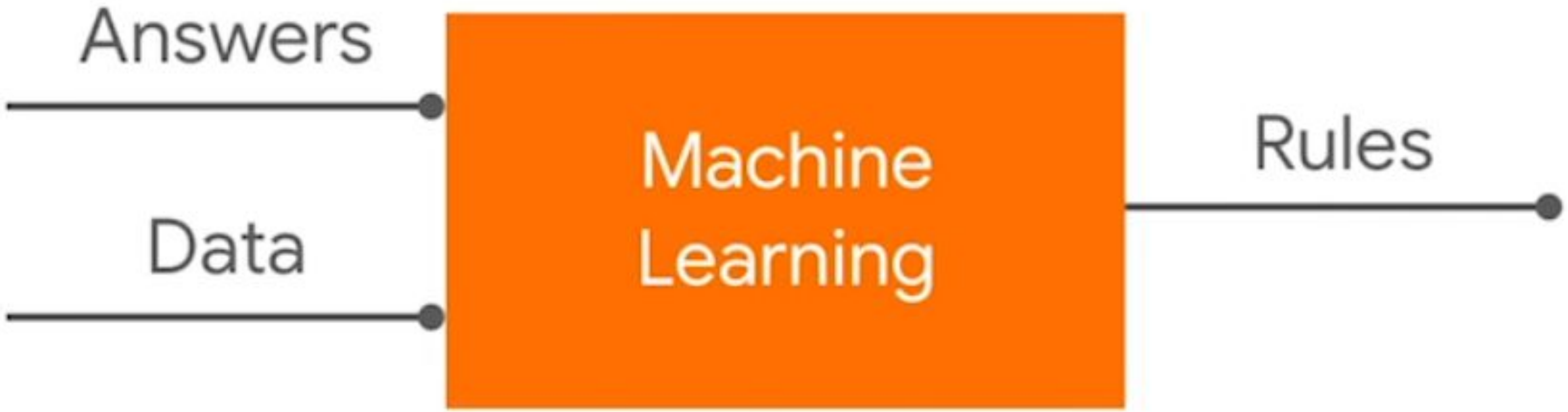
The following materials are from Machine Learning Zero to Hero(Google I/O’19) by Laurence Moroney
https://www.youtube.com/watch?v=VwVg9jCtqaU&t=361s


Rock

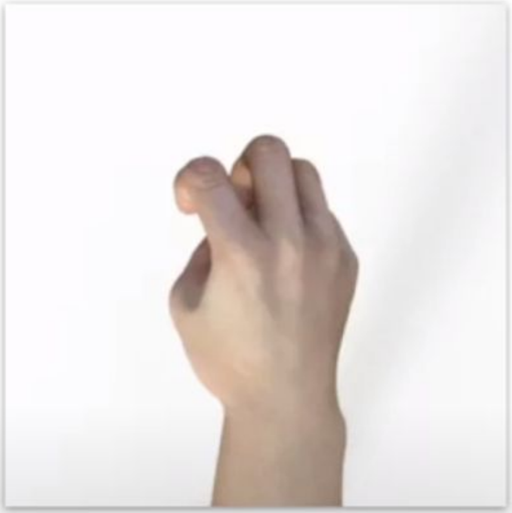
Scissors

Paper


Rules
Data
Traditional programming
Answers


Answers
Data
Machine Learning
Rules

Data and Labels



```
0101001010100101010 1
0010101010010111010 1
0010101001010100101 0
1001010100101010
```

Label = Rock

```
1001010011111010101 1
1010101110101011101 0
1010111101010101111 1
1110001111010101
```

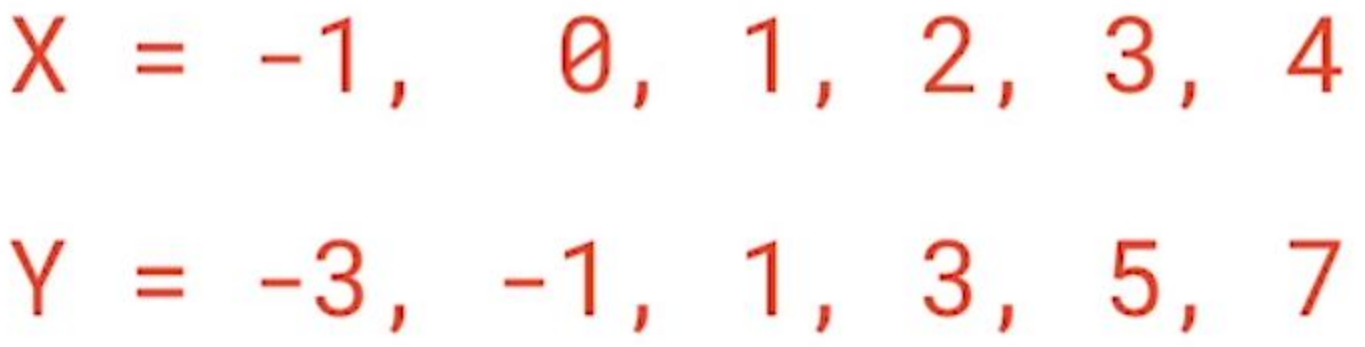
Label = Scissors

```
1010100101001010101 0
1010100100100100010 0
1001111101010111110 1
0100100111101011
```

Label = Paper

Lots of pictures of each

Simple Example



X = -1, 0, 1, 2, 3, 4

Y = -3, -1, 1, 3, 5, 7

What is the relationship between Xs and Ys???

$X = -1, \ 0, \ 1, \ 2, \ 3, \ 4$

$Y = -3, \ -1, \ 1, \ 3, \ 5, \ 7$

$$Y = 2X - 1$$

Code

```
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0,5.0, 7.0], dtype=float)

model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
```

(အရိုးရှင်းဆုံး neural network, layer ၁ခုတည်း, neuron တခုတည်း ရှိမယ်, value တခုထဲ ထည့်ပေးမယ်)

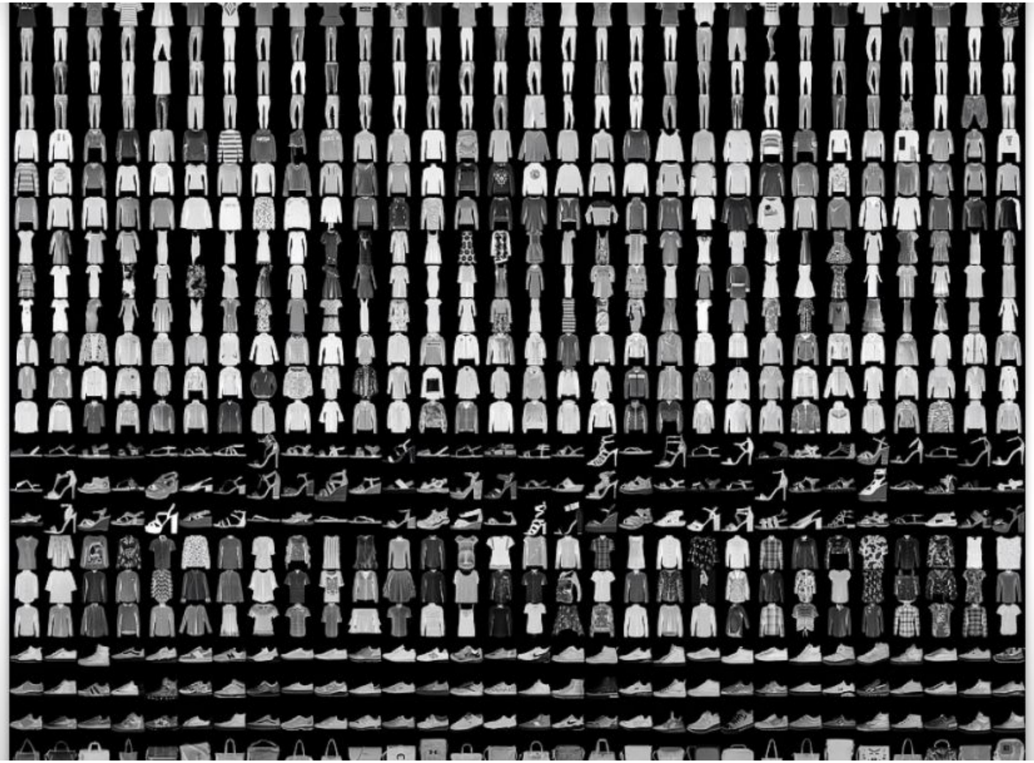
```
model.compile(optimizer='sgd', loss='mean_squared_error')

model.fit(xs, ys, epochs=500)

print(model.predict([10.0]))
```

How to see and recognize images (Basic Computer Vision with ML)

Learn from experience

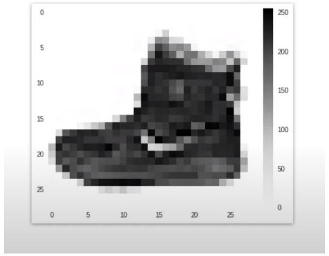
Fashion MNIST

- 70k Images
- 10 Categories
- Images are 28x28
- Can train a neural net!



```
import tensorflow as tf
from tensorflow import keras

mnist = tf.keras.datasets.fashion_mnist
(training_image, traning_labels), (test_images,test_labels) = mnist.load_data()
```

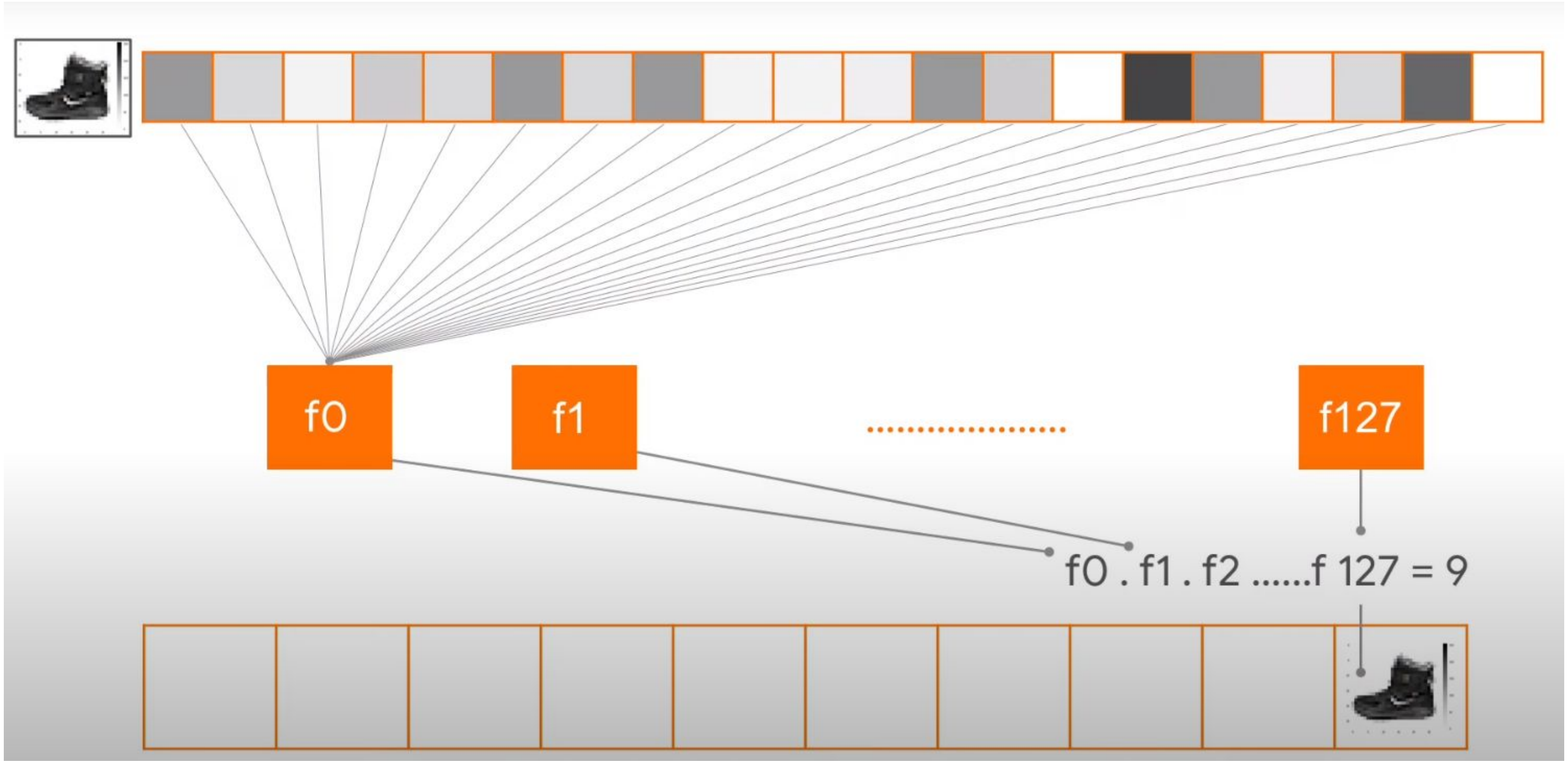
(၆၀၀၀) (၁၀၀၀)

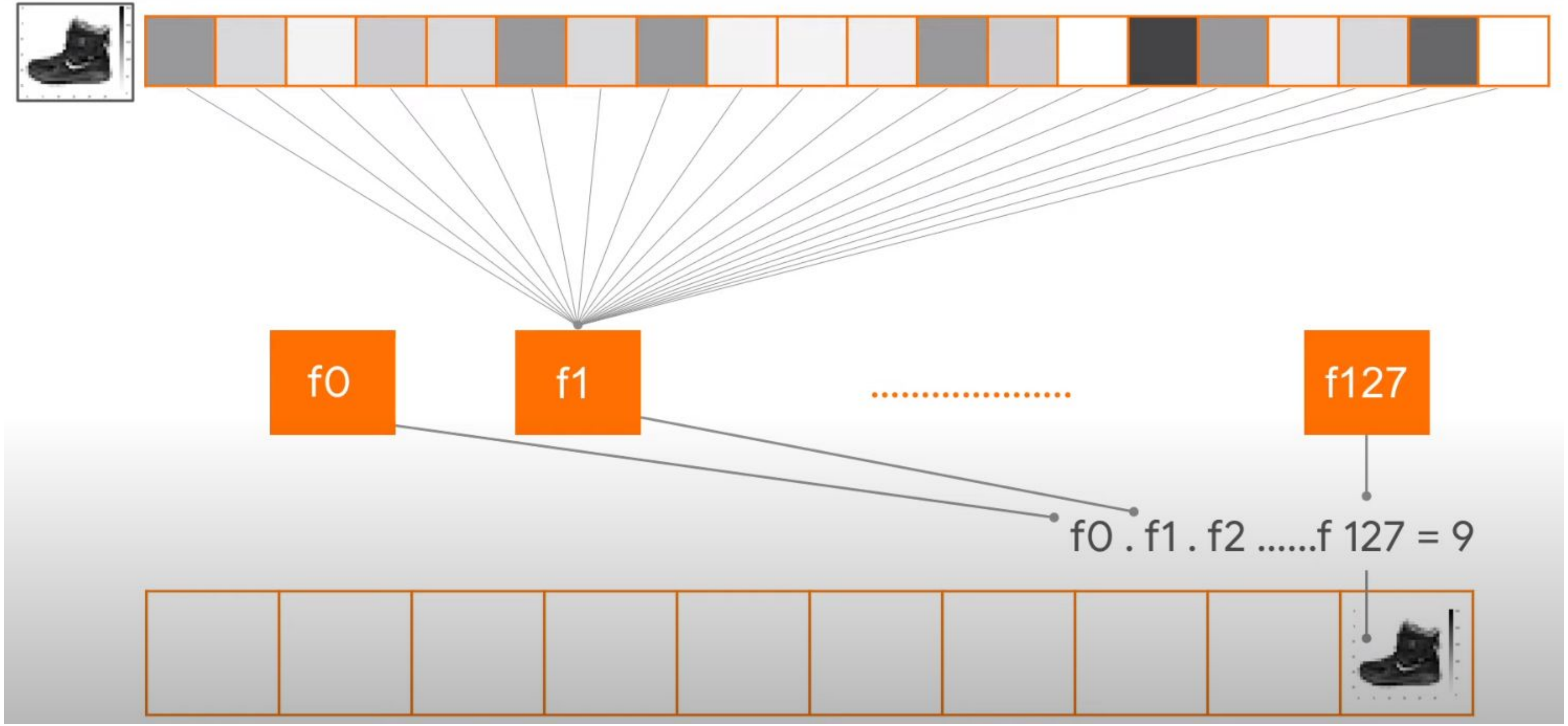


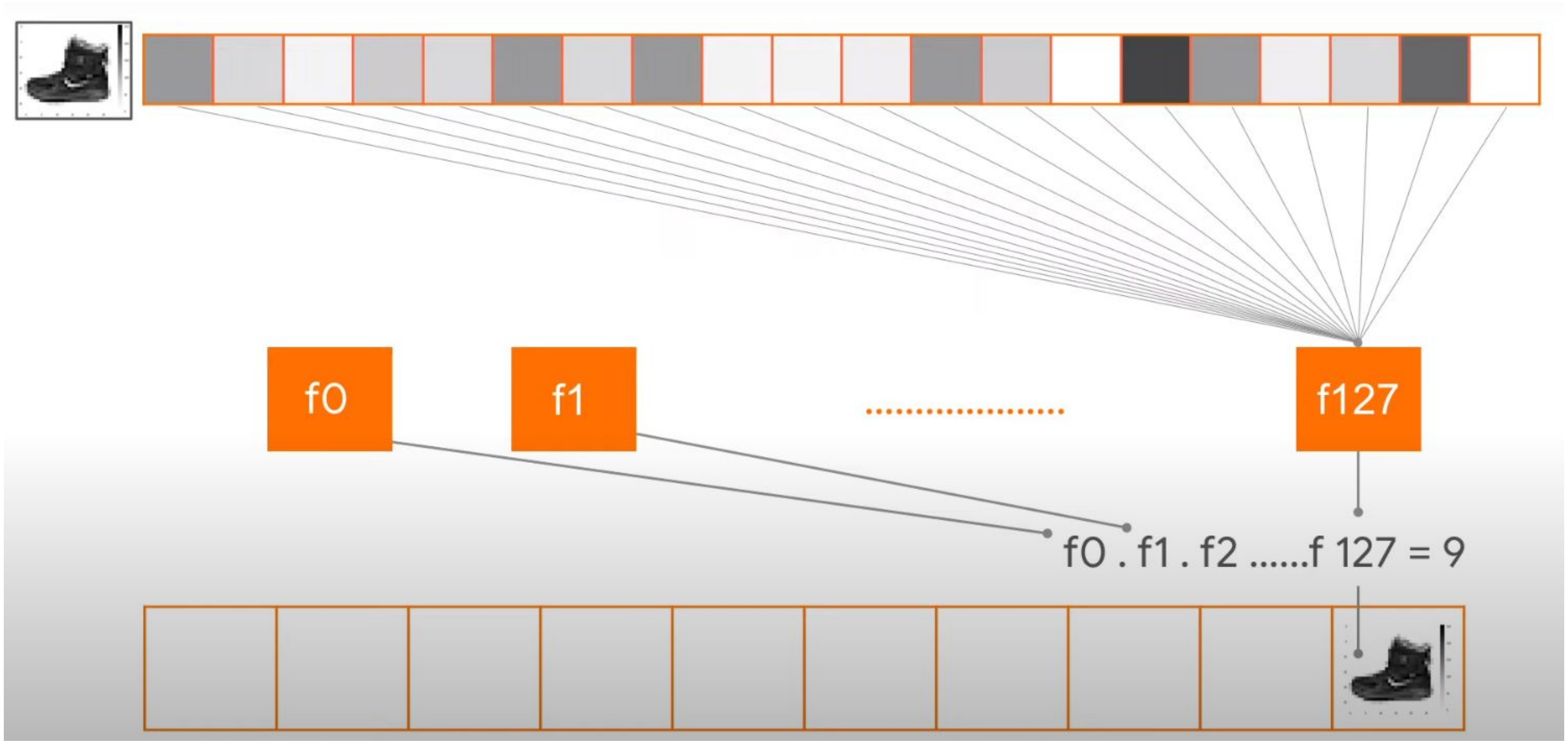
9

Code

```
model = tf.keras.models.Sequential([
                              tf.keras.layers.Flatten(input_shape=(28,28)),
                              tf.keras.layers.Dense(128, activation=tf.nn.relu),    (၁၂၈ က ဘာလဲ)
                              tf.keras.layers.Dense(10, activation=tf.nn.softmax)])

model.compile(optimizer = tf.keras.optimizers.Adam(),
          loss = 'sparse_categorical_crossentropy',
          metrics=['accuracy'])
```


f0
f1
....
f127
f0 . f1 . f2 ......f 127 = 9


f0
f1
f127
f0 . f1 . f2 ......f 127 = 9


f0
f1
f127
f0 . f1 . f2 ......f 127 = 9

Code

```
model = tf.keras.models.Sequential([
                                   tf.keras.layers.Flatten(input_shape=(28,28)),
                                   tf.keras.layers.Dense(128, activation=tf.nn.relu),
                                   tf.keras.layers.Dense(10, activation=tf.nn.softmax)])


model.compile(optimizer = tf.keras.optimizers.Adam(),
                          loss = 'sparse_categorical_crossentropy',)
```

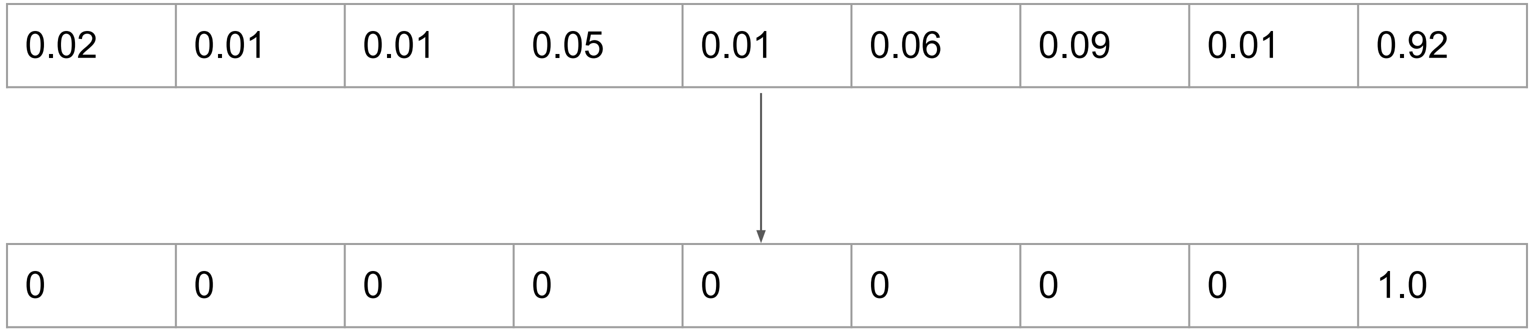
Code

```
model = tf.keras.models.Sequential([
                    tf.keras.layers.Flatten(input_shape=(28,28)),
                    tf.keras.layers.Dense(128, activation=tf.nn.relu),
                    tf.keras.layers.Dense(10, activation=tf.nn.softmax)])
```

```
if (x>0) {
return x;
}
else {
return 0;
}
```

Code

```
model = tf.keras.models.Sequential([
                    tf.keras.layers.Flatten(input_shape=(28,28)),
                         tf.keras.layers.Dense(128, activation=tf.nn.relu),
                         tf.keras.layers.Dense(10, activation=tf.nn.softmax)])
```



0.02	0.01	0.01	0.05	0.01	0.06	0.09	0.01	0.92

0	0	0	0	0	0	0	0	1.0

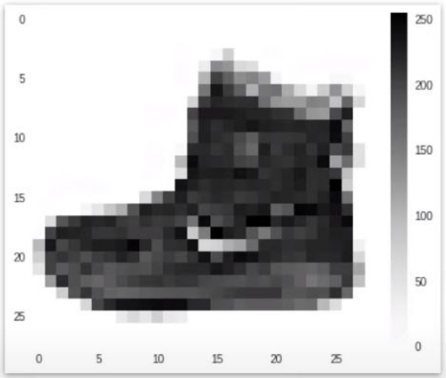
Code

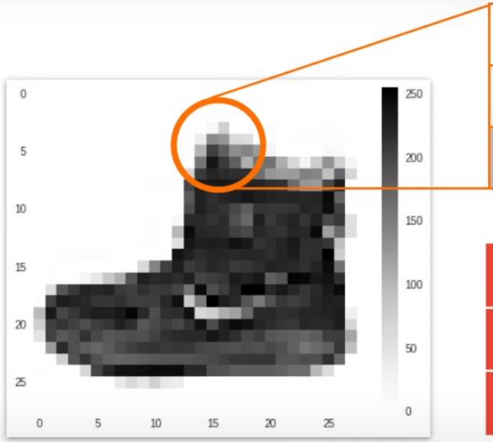
```
model.fit(training_images, training_labels, epochs=5)

test_ loss, test_acc = model.evaluate(test_image, test_labels)
(train မလုပ်ရသေးတဲ့ ပုံ ၁၀၀၀ ကို စမ်းမှာ)
predictions = model.predict(my_images)
```

Limitations

GrayScale Image and Image must be centered!

Convolutional Neural Network!!!





0	64	128
48	192	144
142	226	168

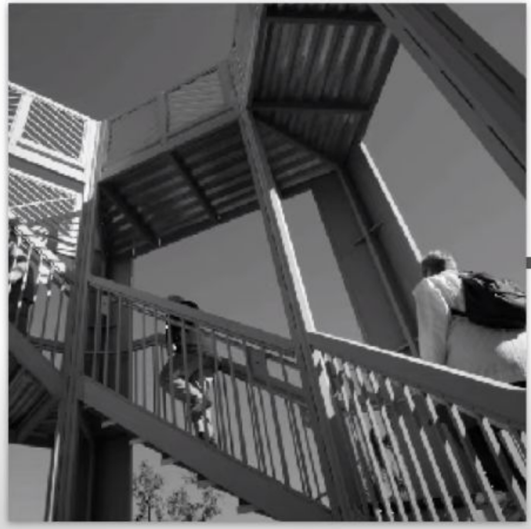
Current pixel value is 192

Consider neighbor values

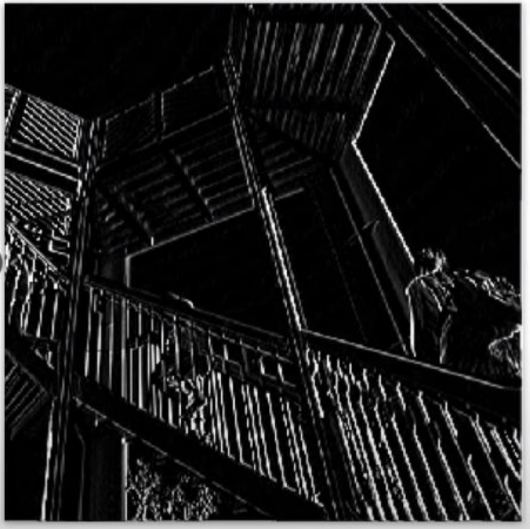
-1	0	-2
.5	4.5	-1.5
1.5	2	-3

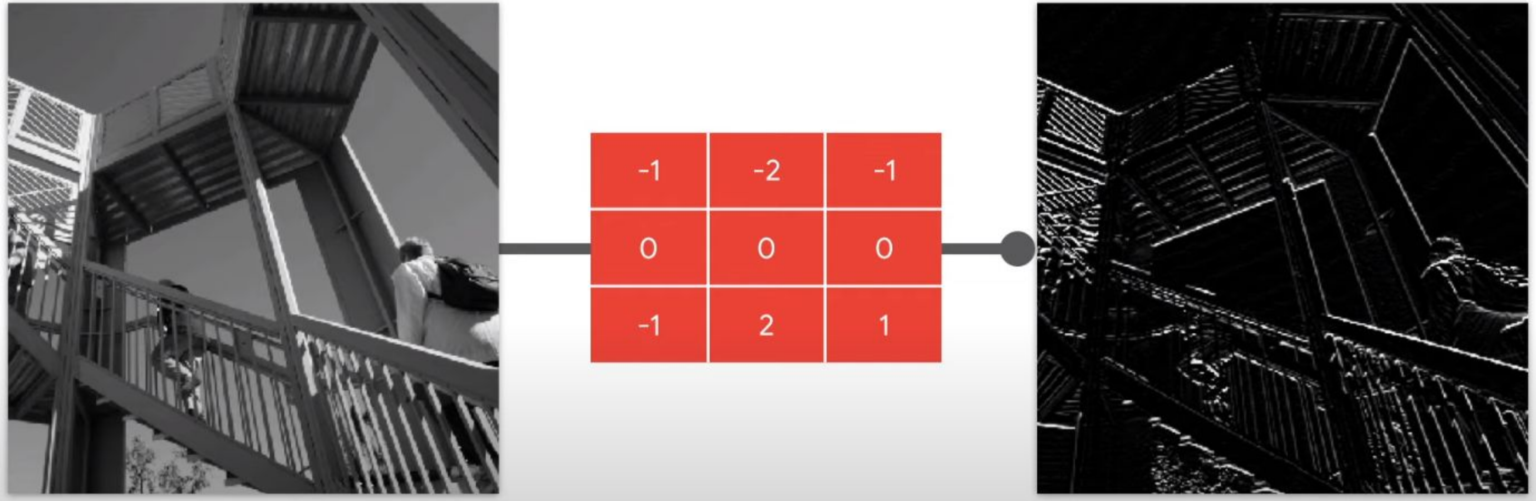
Filter definition

```
CURRENT_PIXEL_VALUE = 192
NEW_PIXEL_VALUE = (-1 * 0) + (0 * 64) + (-2 * 128) +
                  (.5 * 48) + (4.5 * 192) + (-1.5 * 144) +
                  (1.5 * 42) + (2 * 226) + (-3 * 168)
```

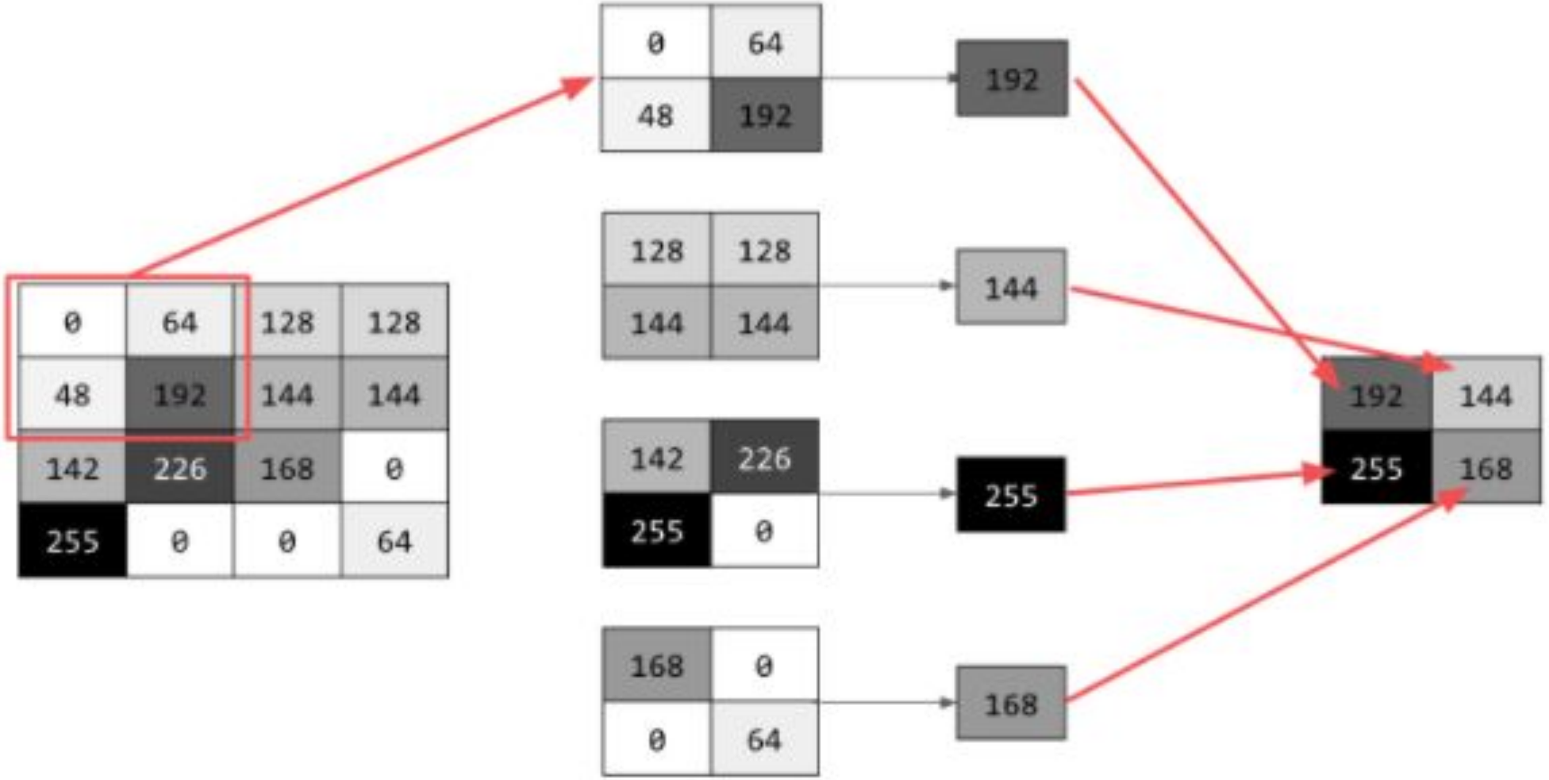


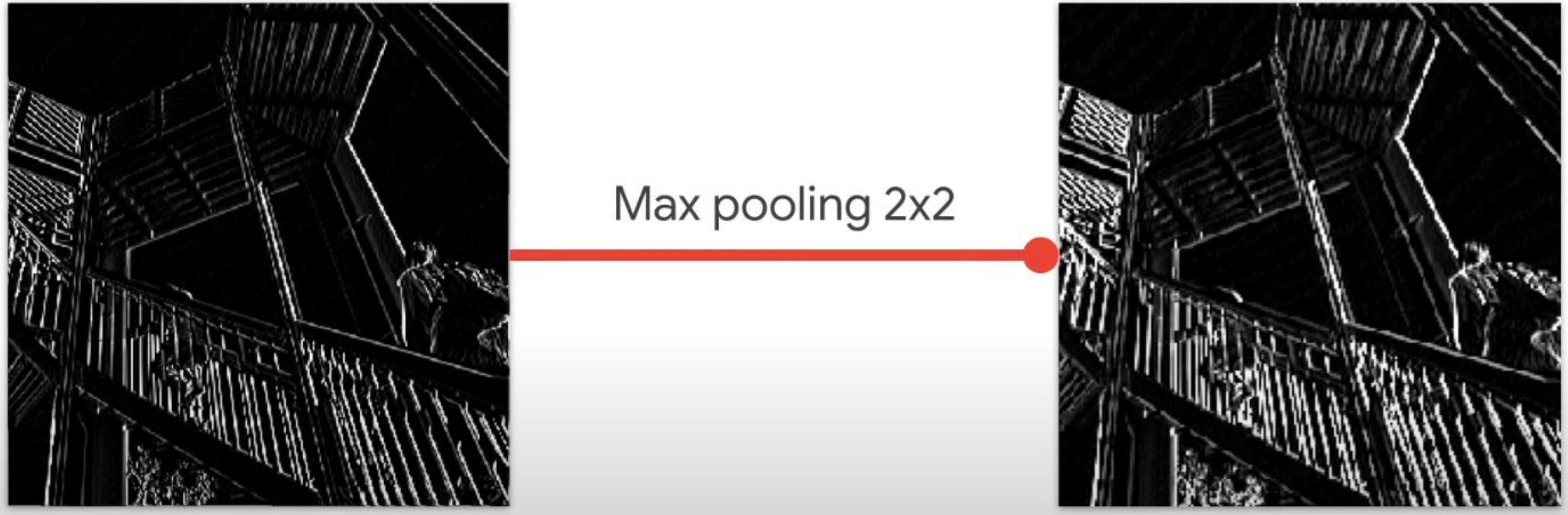
-1	0	1
-2	0	2
-1	0	1

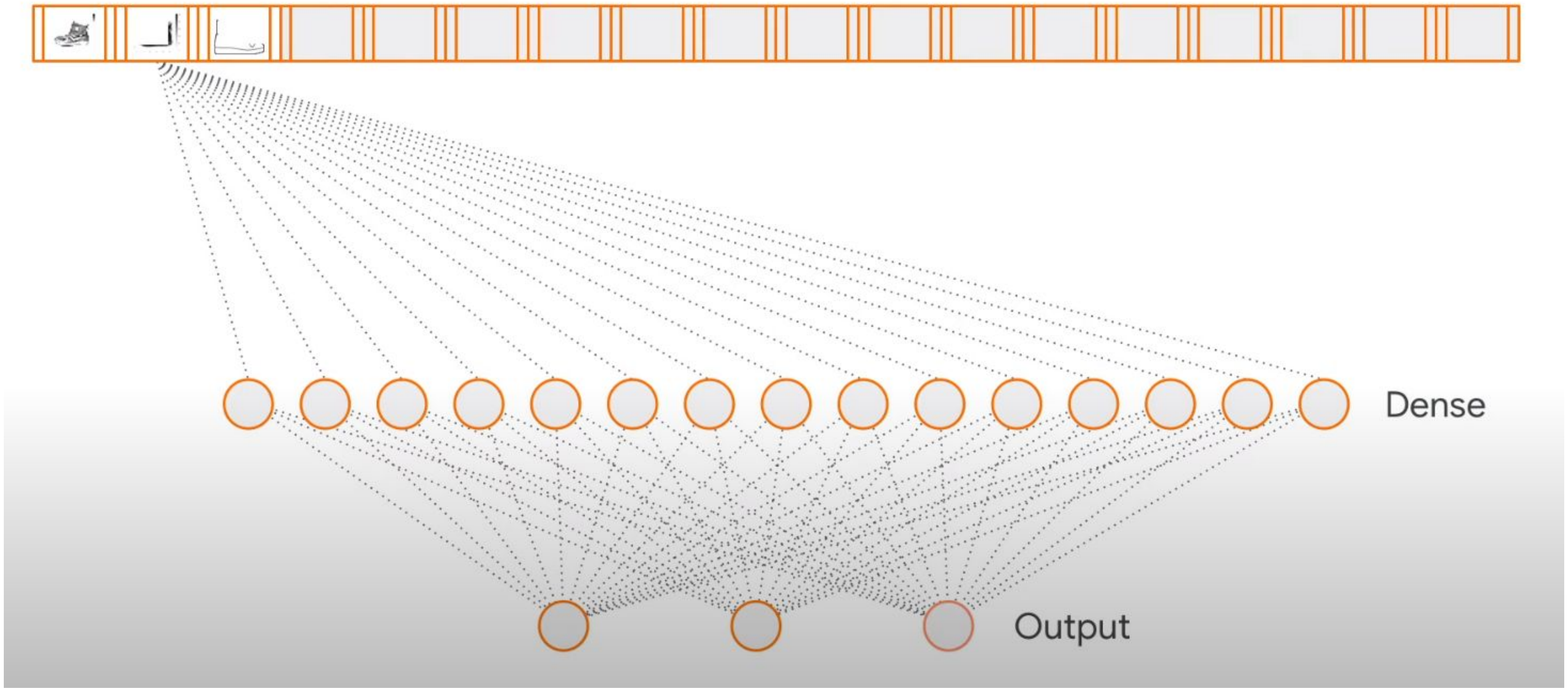




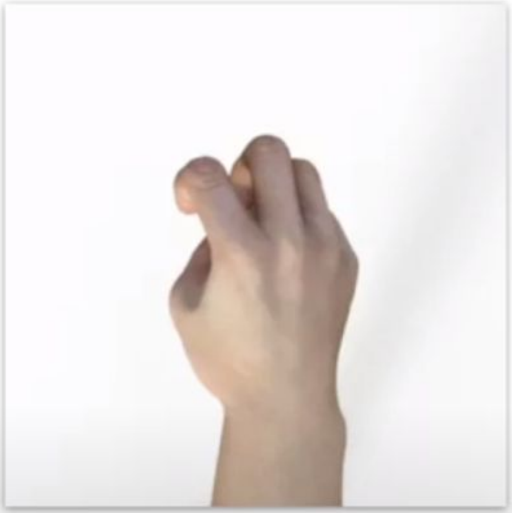
-1	-2	-1
0	0	0
-1	2	1


0
64
128
128
48
192
144
144
142
226
168
0
255
0
0
64
0
64
48
192
192
128
128
144
144
144
142
226
255
0
255
168
0
0
64
168
192
144
255
168


Max pooling 2x2


Dense
Output

Data and Labels



01010010101001010101
00101010100101110101
00101010010101001010
1001010100101010

Label = Rock

10010100111110101011
10101011101010111010
10101111010101011111
1110001111010101

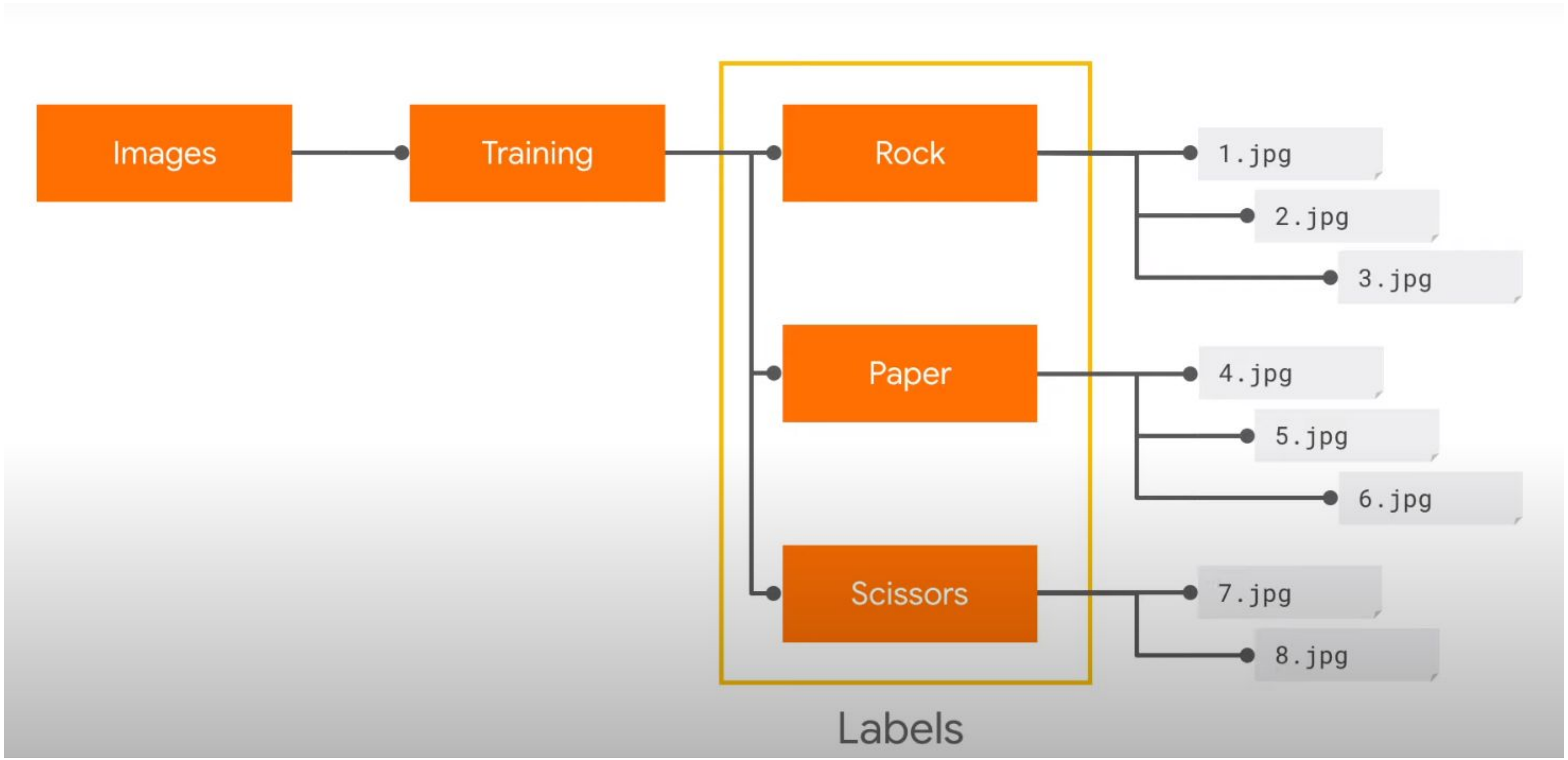
Label = Scissors

10101001010010101010
10101001001001000100
10011111010101111101
0100100111101011

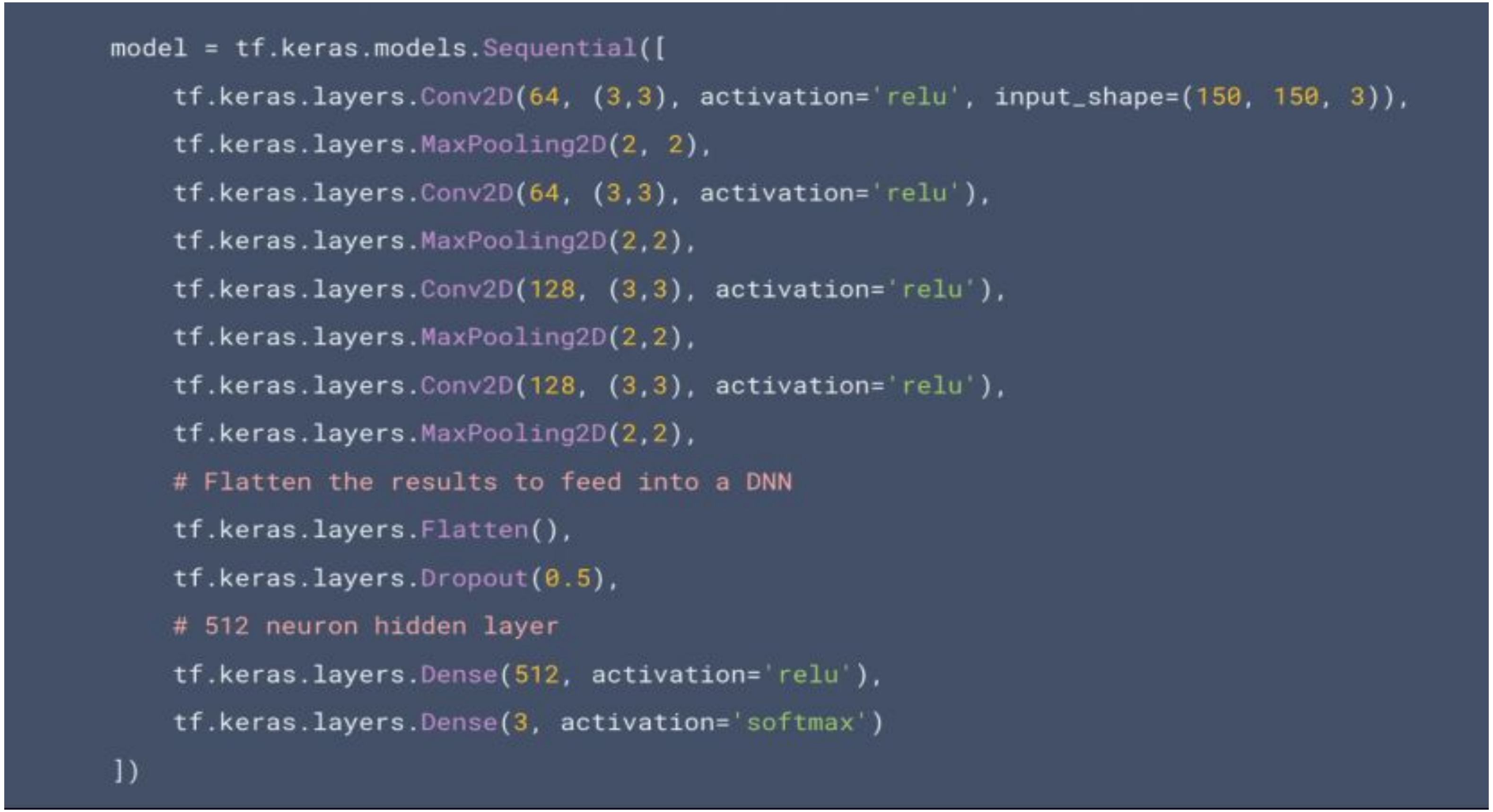
Label = Paper

Lots of pictures of each

```
!wget --no-check-certificate \
   https://storage.googleapis.com/laurencemoroney-blog.appspot.com/rps.zip \
   -O /tmp/rps.zip
 !wget --no-check-certificate \
   https://storage.googleapis.com/laurencemoroney-blog.appspot.com/rps-test-set.zip \
   -O /tmp/rps-test-set.zip

import os
import zipfile

local_zip = '/tmp/rps.zip'
zip_ref = zipfile.ZipFile(local_zip, 'r')
zip_ref.extractall('/tmp/')
zip_ref.close()

local_zip = '/tmp/rps-test-set.zip'
zip_ref = zipfile.ZipFile(local_zip, 'r')
zip_ref.extractall('/tmp/')
zip_ref.close()
```

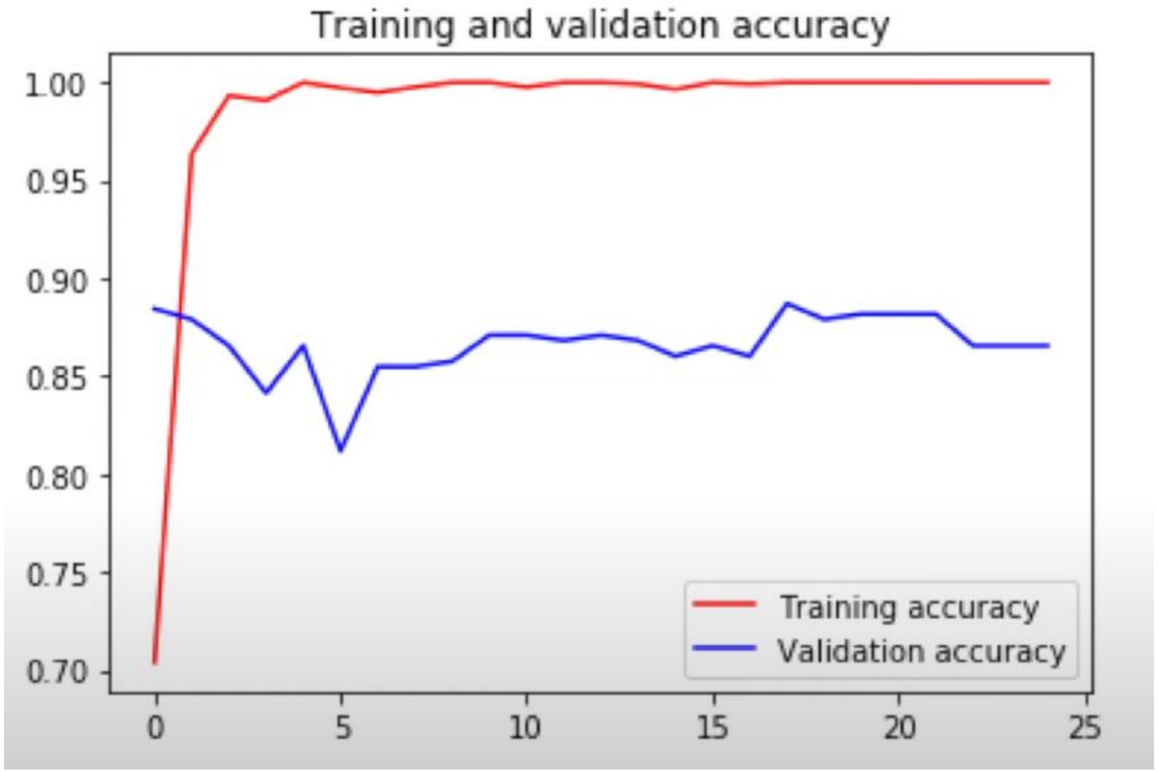

Images
Training
Rock
Paper
Scissors
1.jpg
2.jpg
3.jpg
4.jpg
5.jpg
6.jpg
7.jpg
8.jpg
Labels

Code



```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(64, (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),
    # Flatten the results to feed into a DNN
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.5),
    # 512 neuron hidden layer
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(3, activation='softmax')
])
```

Overfitting



Overfitting

Vs

https://github.com/NMG-thinkers/deep_learning/

https://www.youtube.com/watch?v=r5ZlPlpkB2A&list=PLI8VZl7GXFE94K2_y31Zv9sgyj3ttDM0m

Thank you