AI for Good Workshop

March 2019

Session 4

Artificial Intelligence for everyone

Build your 1st solution on Deep Learning

https://sites.google.com/view/AIforEveryone

+

Topics in this video

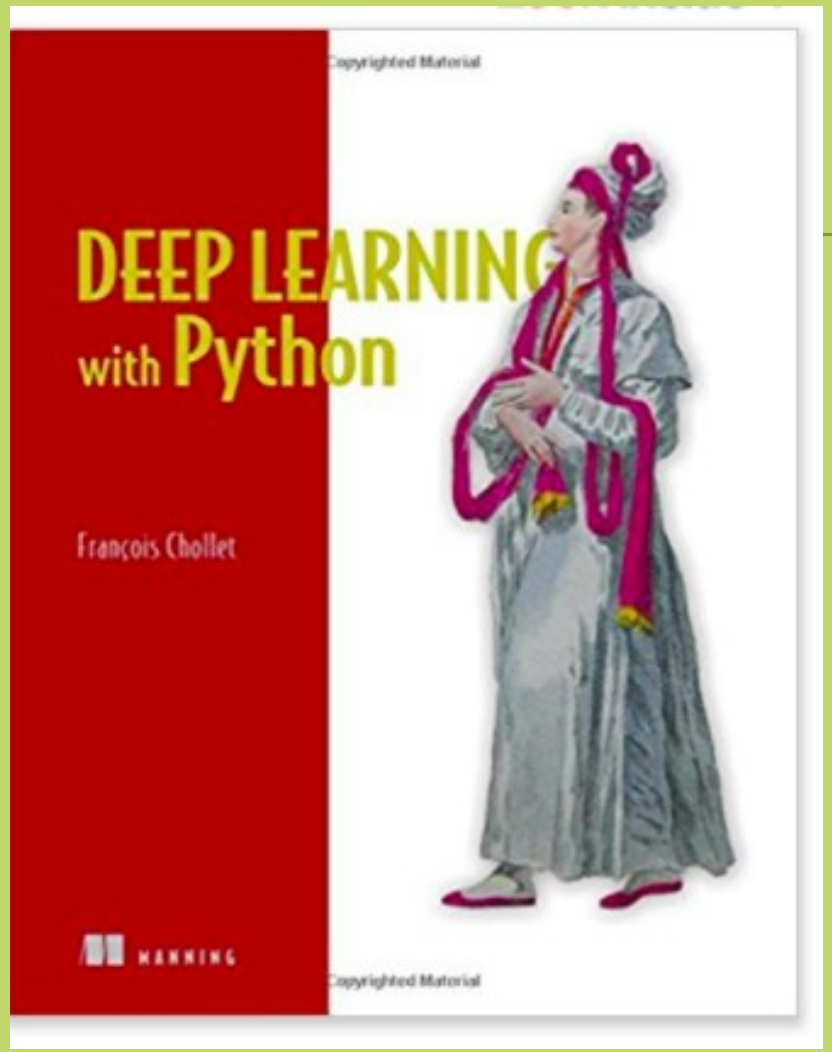
Jump start your AI journey !

1. Define a purpose

2. Apply the Strategy to jump start

3. Hands-on : Create your 1st Deep Learning solution

The inspiration



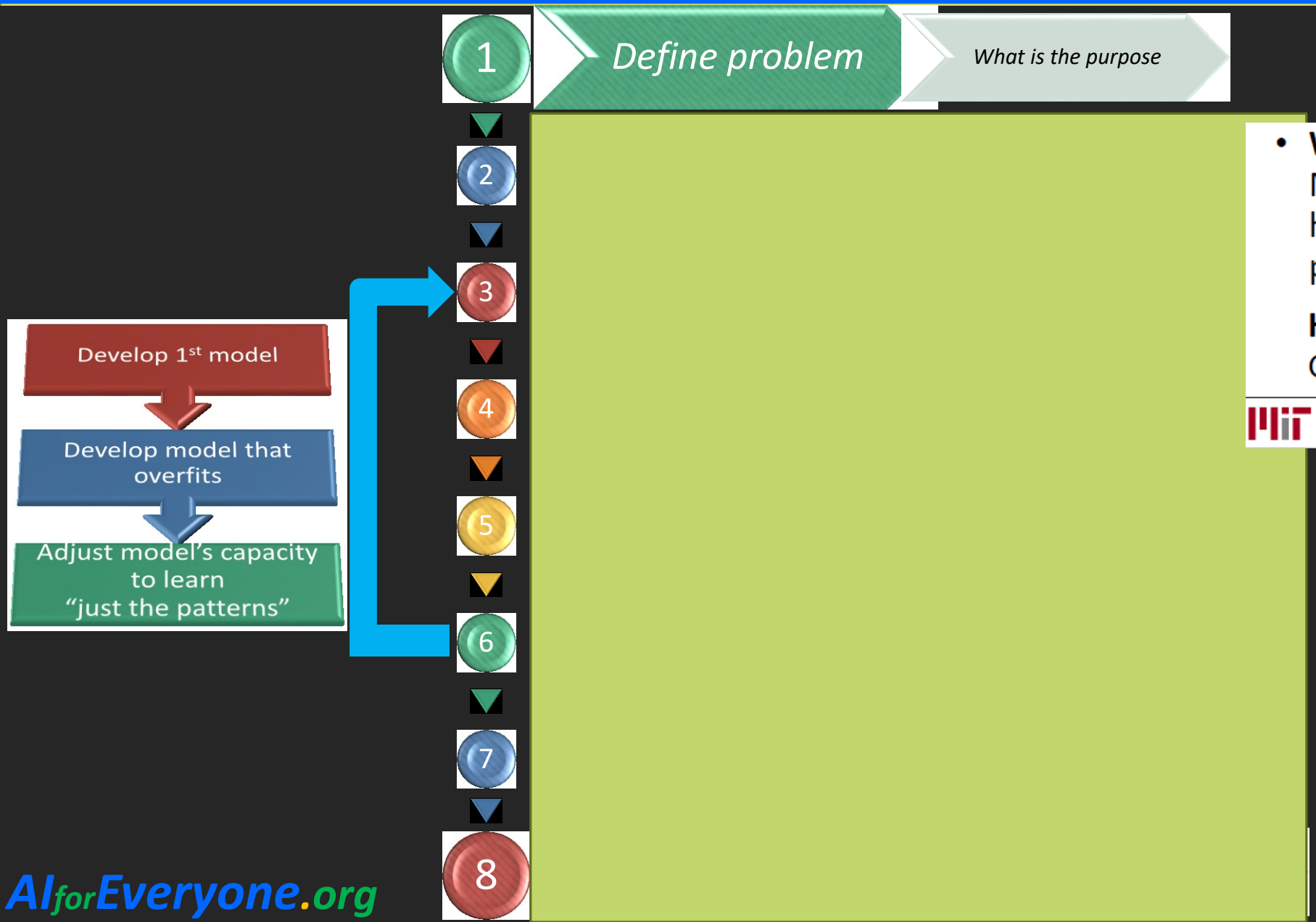
Francois Chollet

Artificial Intelligence Researcher,
Google

Deep Learning with Python
https://www.manning.com/books/deep-learning-with-python

ISBN-13: 978-1617294433

Design of Deep Learning experiment



- **Where do we stand?**
  Most big questions of intelligence have not been answered nor properly formulated

Hard Part:
Good Questions + Good Data

Massachusetts Institute of Technology

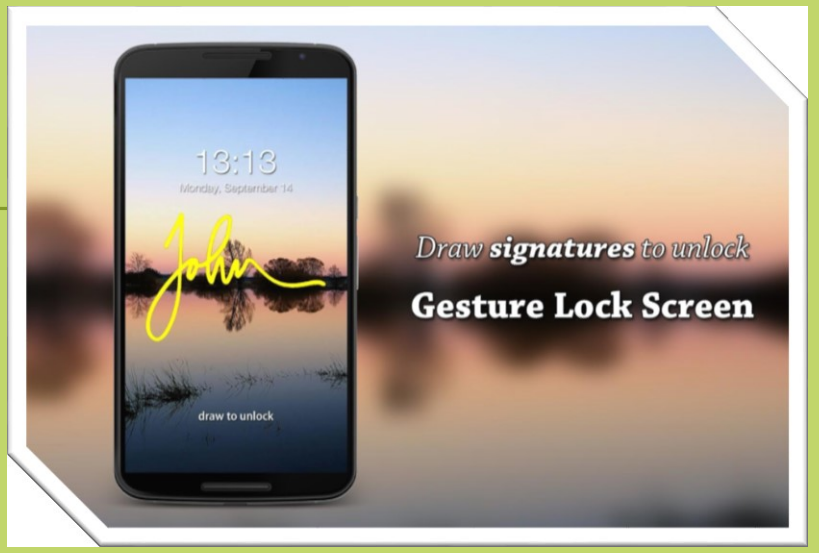
For the full list of references visit:
https://hcai.mit.edu/references

1 *Define your larger purpose*

Define the idea

- App idea: Recognize your hand gesture to unlock your smartphone (Secure unlock)

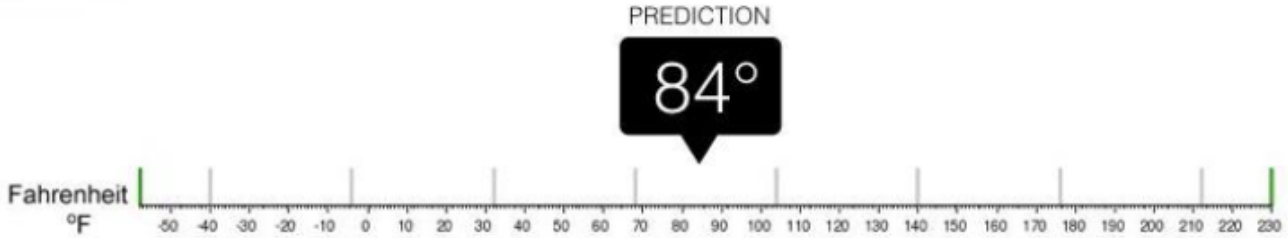
- Problem: Recognize hand draw gestures to unlock your phone



Regression vs Classification

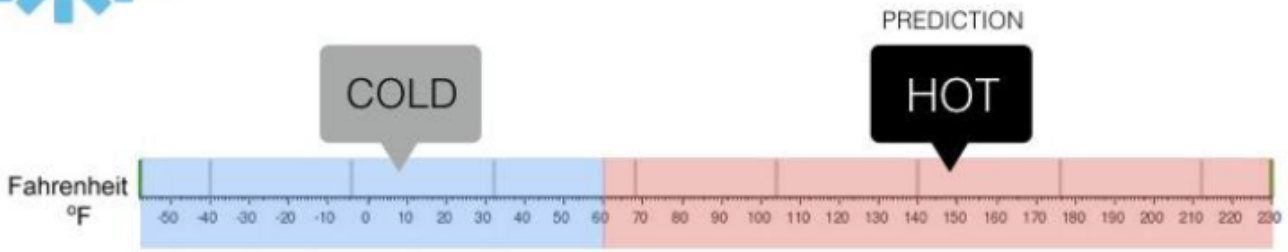
Regression

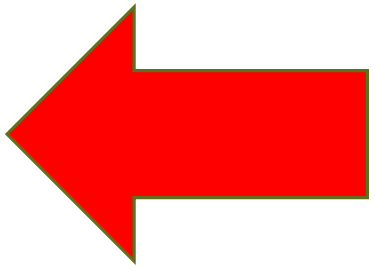
What is the temperature going to be tomorrow?



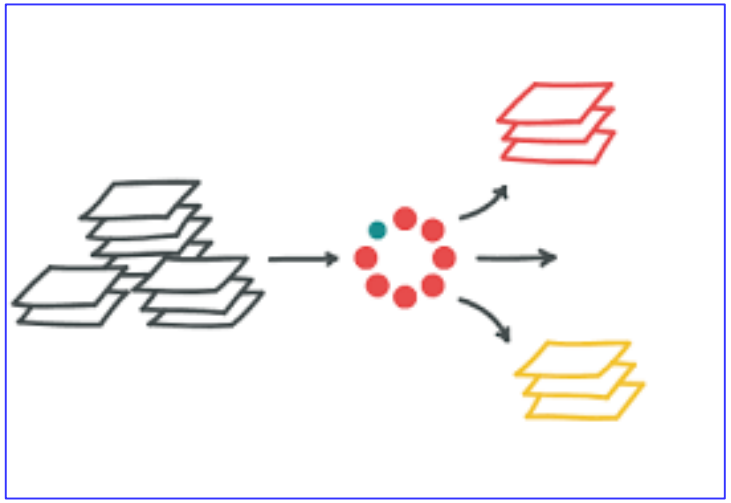
Classification

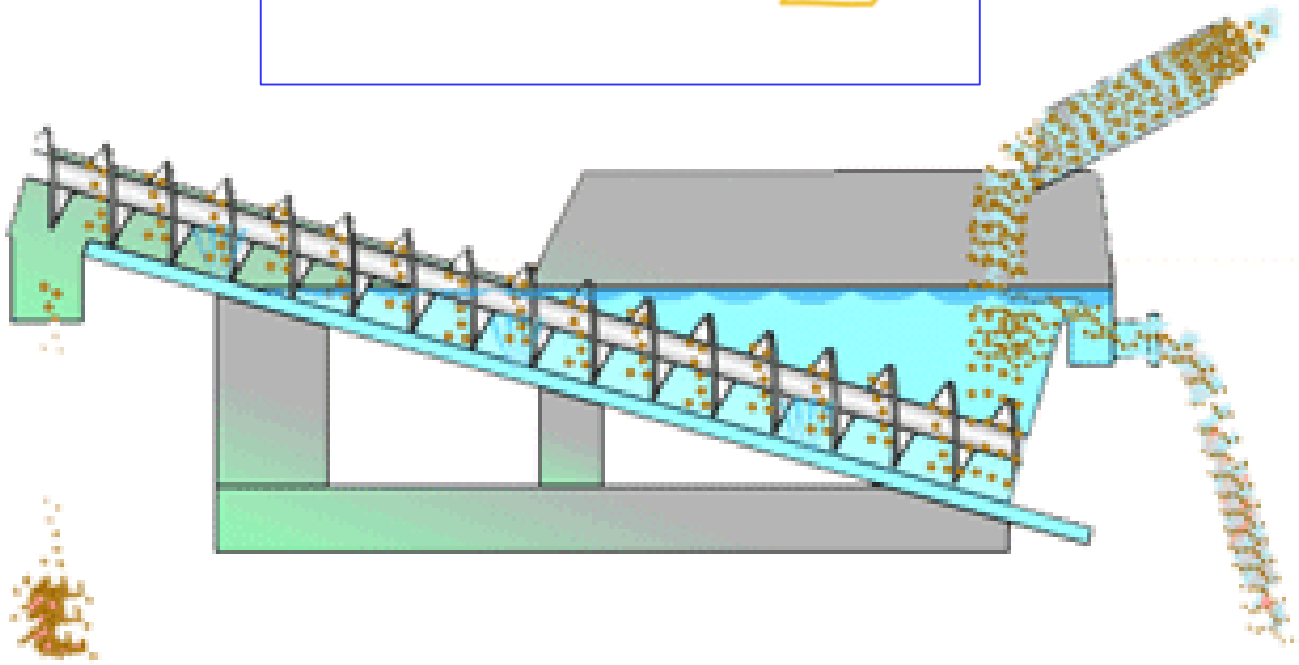
Will it be Cold or Hot tomorrow?

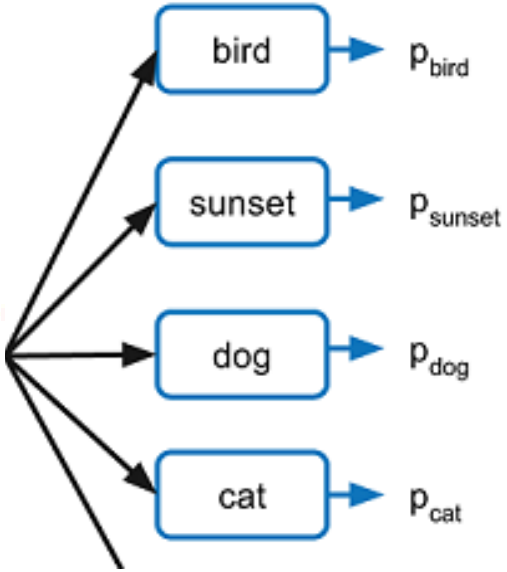




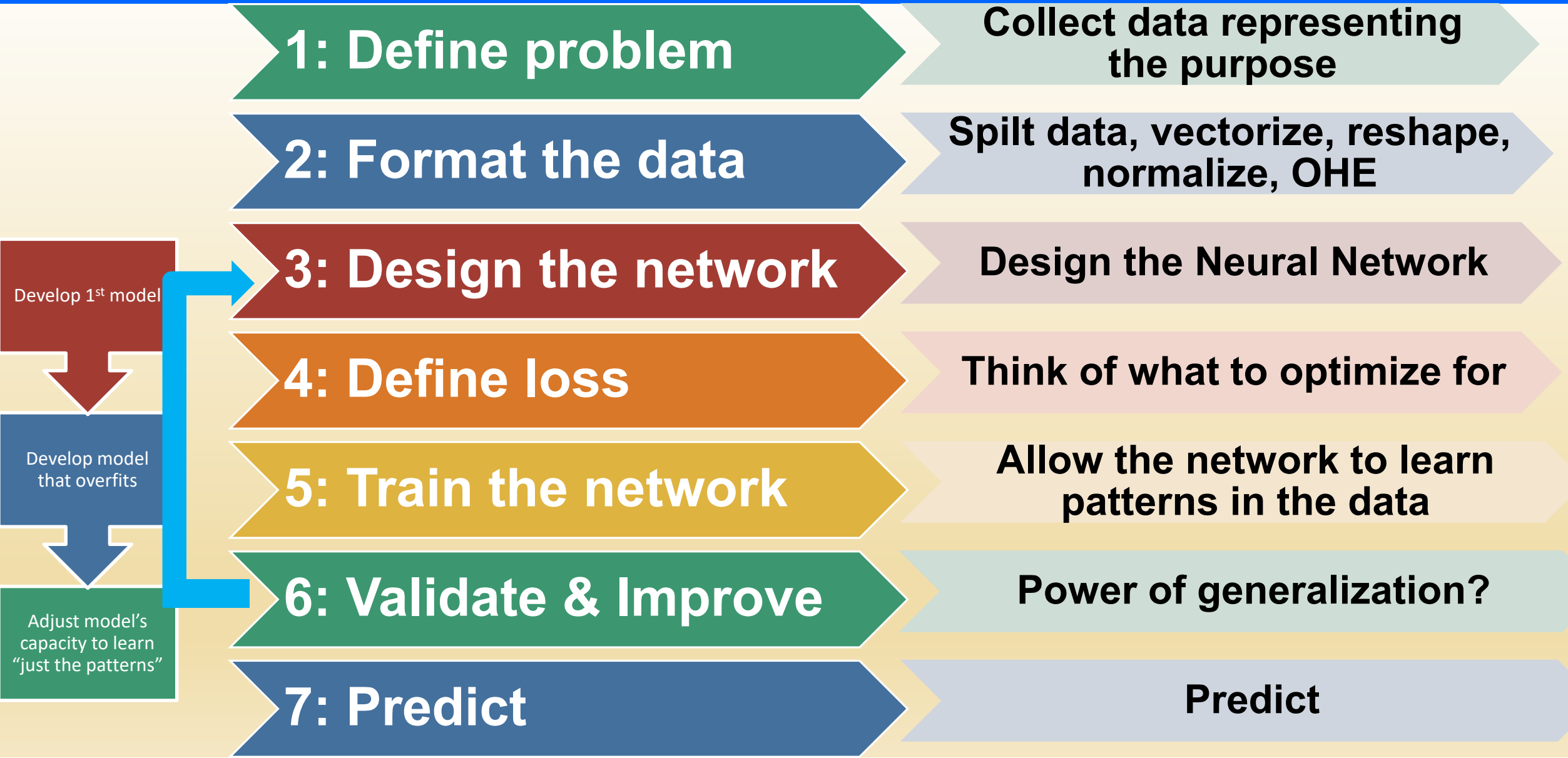
visual recognition – image classification







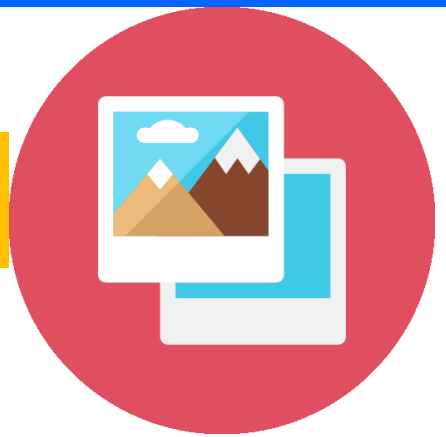
Method: 7 steps to develop your Deep Learning model



A → B

X → Y

X = image



Input Image:

TensorFlow Model:

Neural Network

Output:

5

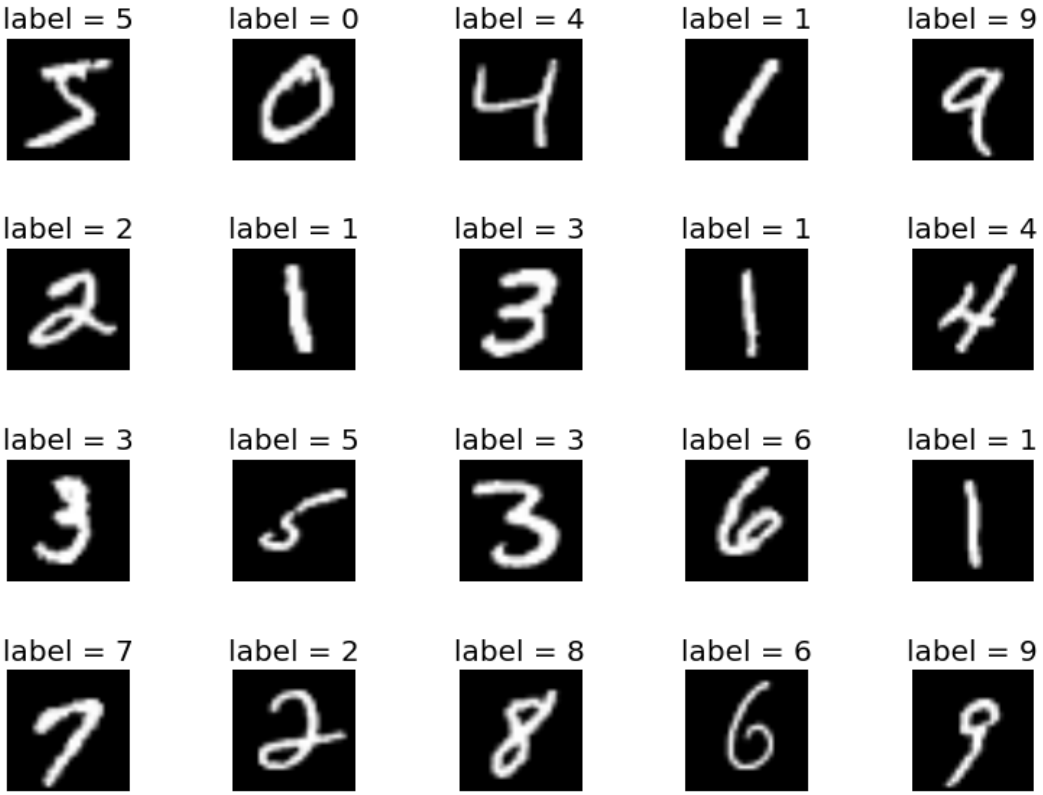
Y = Category of image

1 *Define your problem*

Simplify your goal

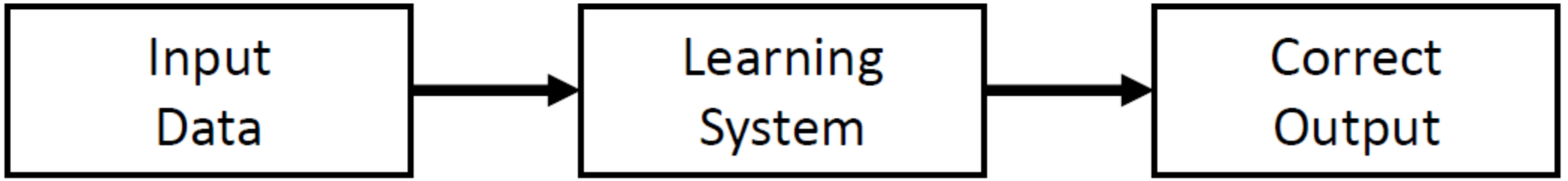
Problem: Users draws a hand drawn gestures, your app identifies it.

Simplified Problem: User can draw any number between 0 to 9



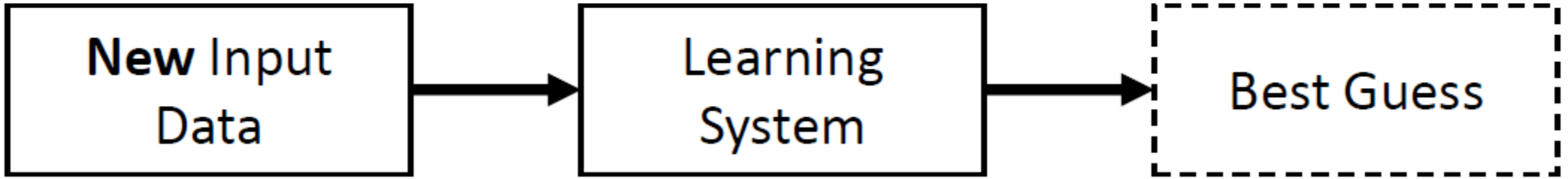
Deep Learning: Training and Testing

Training Stage:



(aka "Ground Truth")

Testing Stage:



The Challenge of Deep Learning:
Efficient Teaching + Efficient Learning

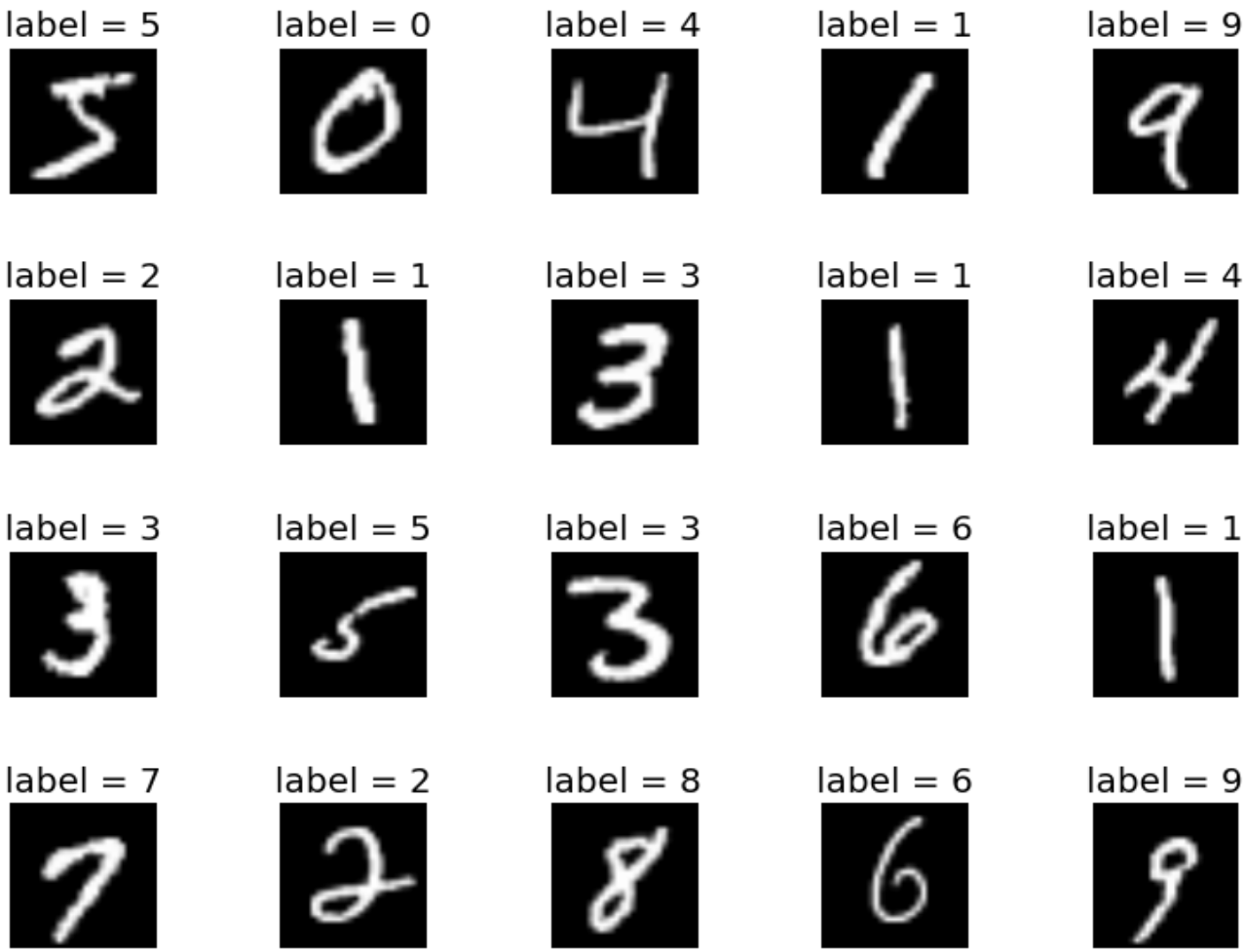
- Humans can learn from very few examples
- Machines (in most cases) need thousands/millions of examples

1. Define problem

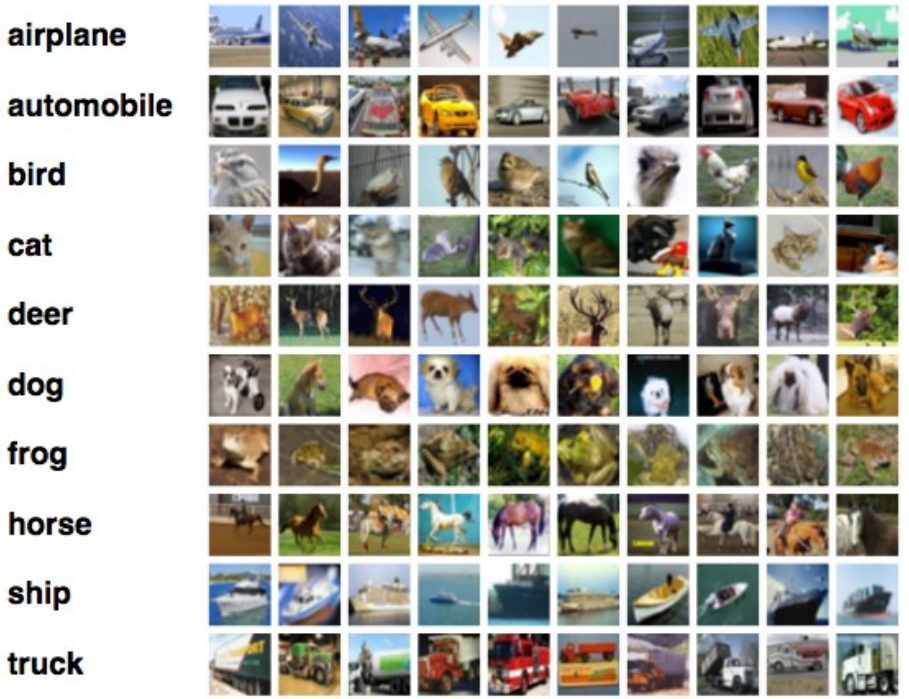
Collect the data representing the world you what to predict

Load the data

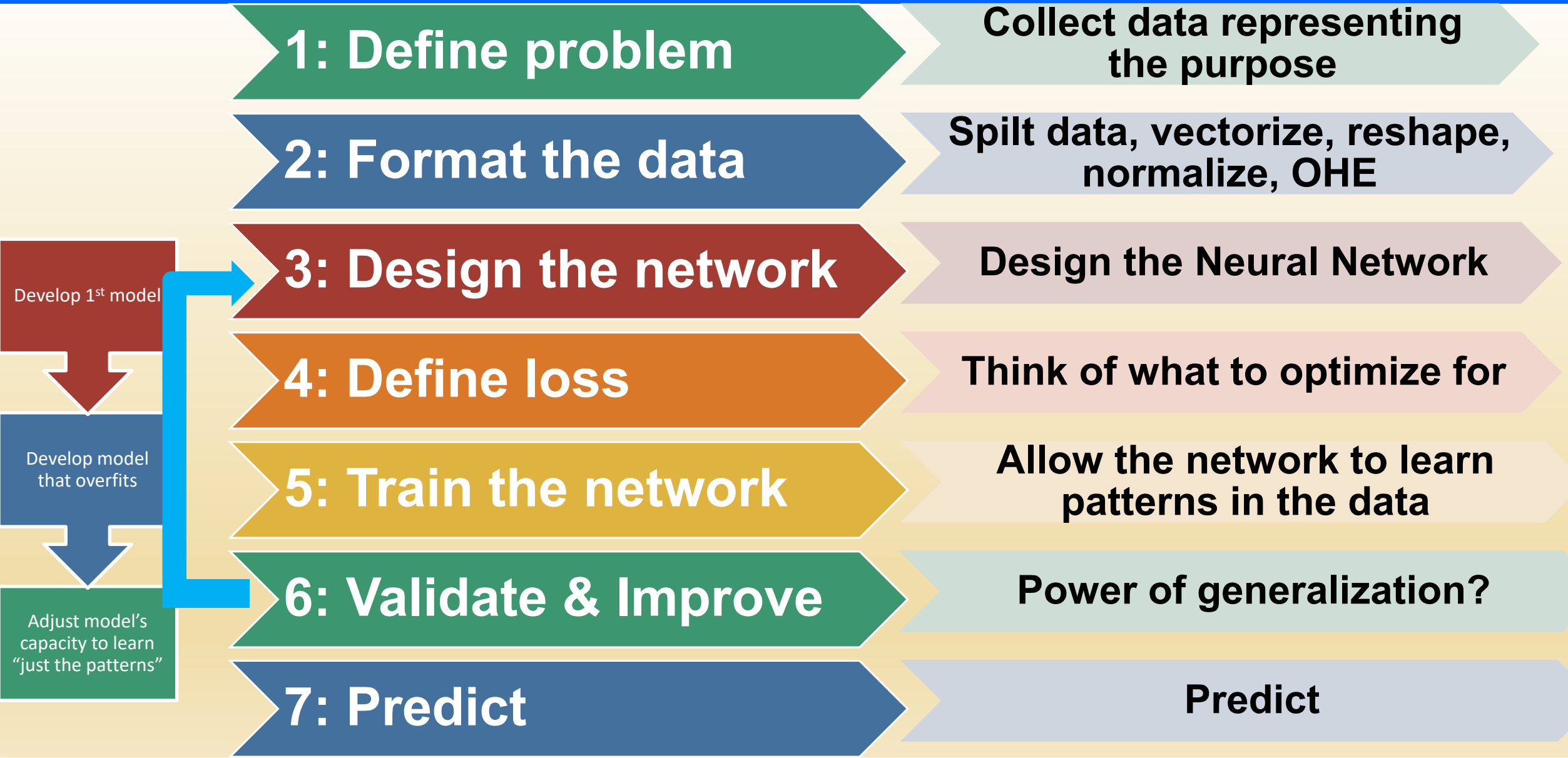
Visualize data



Labeled dataset



Method: 7 steps to develop your Deep Learning model



1. Define problem

Collect the data representing the world you what to predict

Load the data

Visualize data

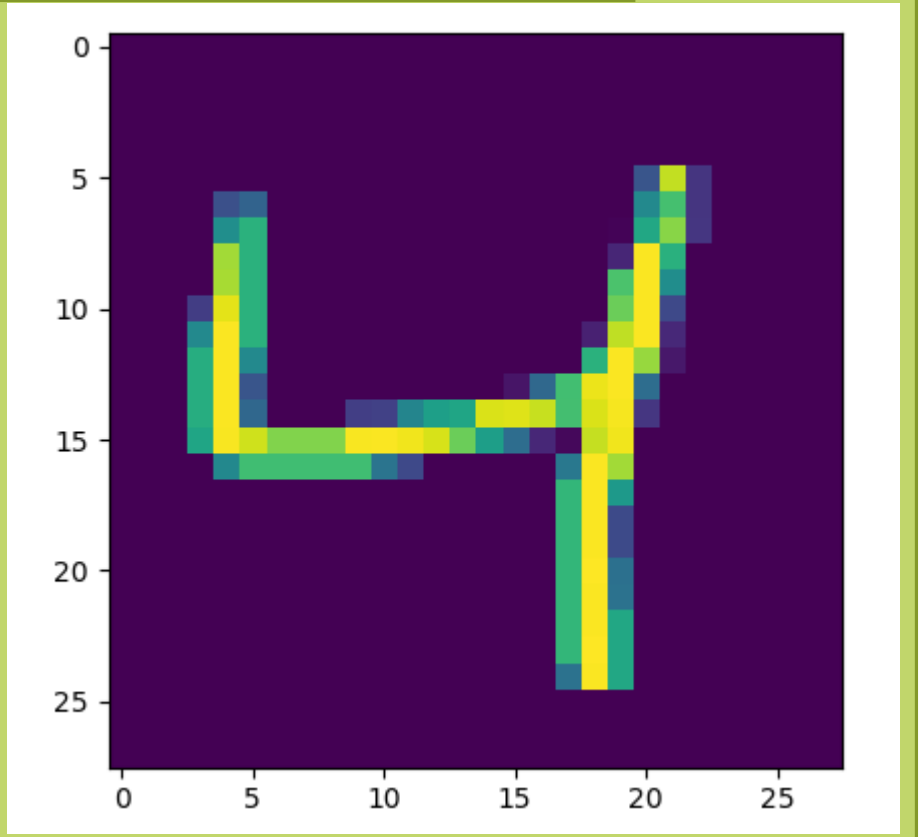
keras. datasets. mnist. load_data()

```
from keras.datasets import mnist

(trainX, trainY), (testX, testY) =  mnist. load_data()

len(trainX)
```

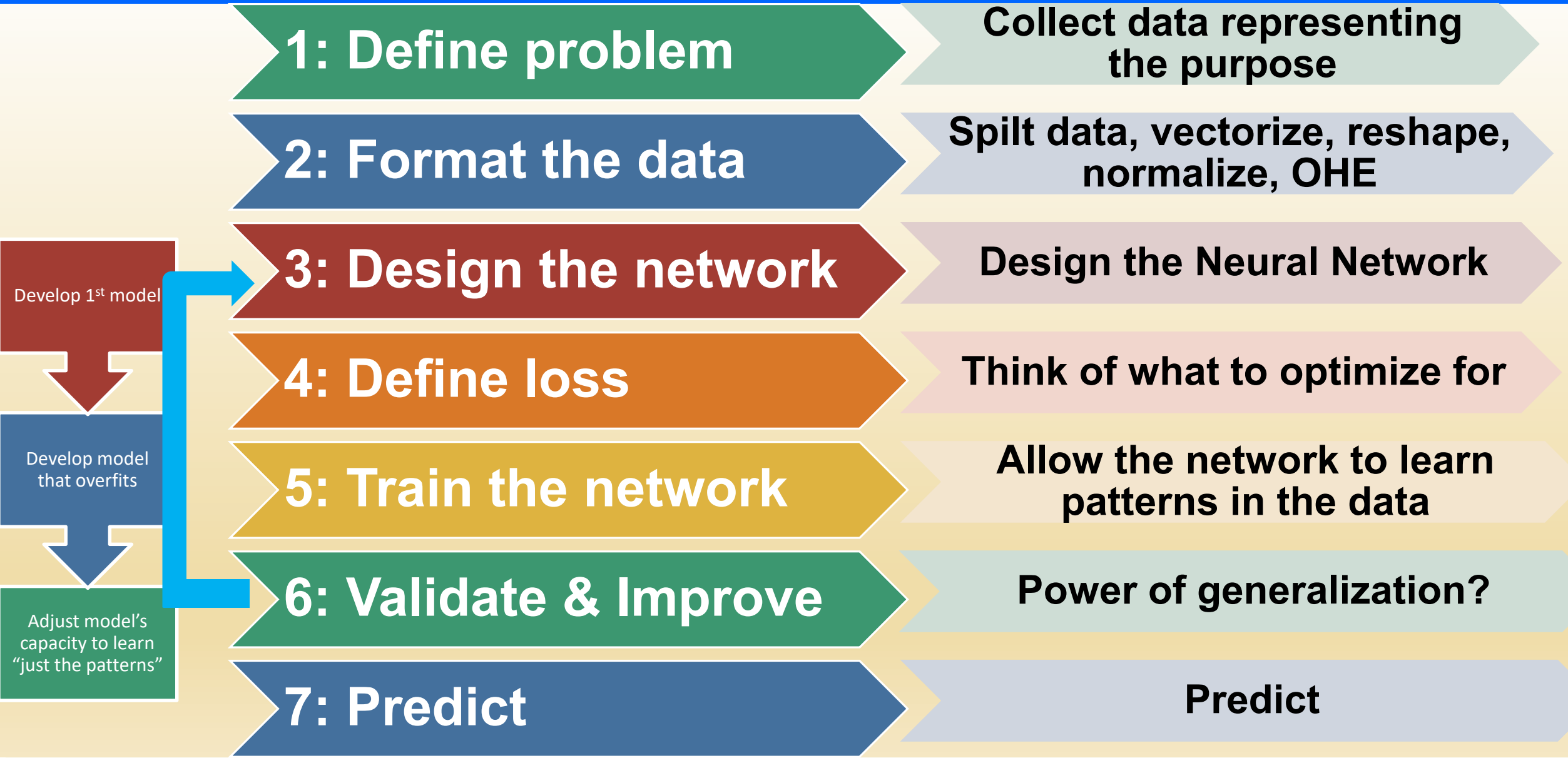
type (myObject)

matplotlib. pyplot. imshow (myArrayImage)

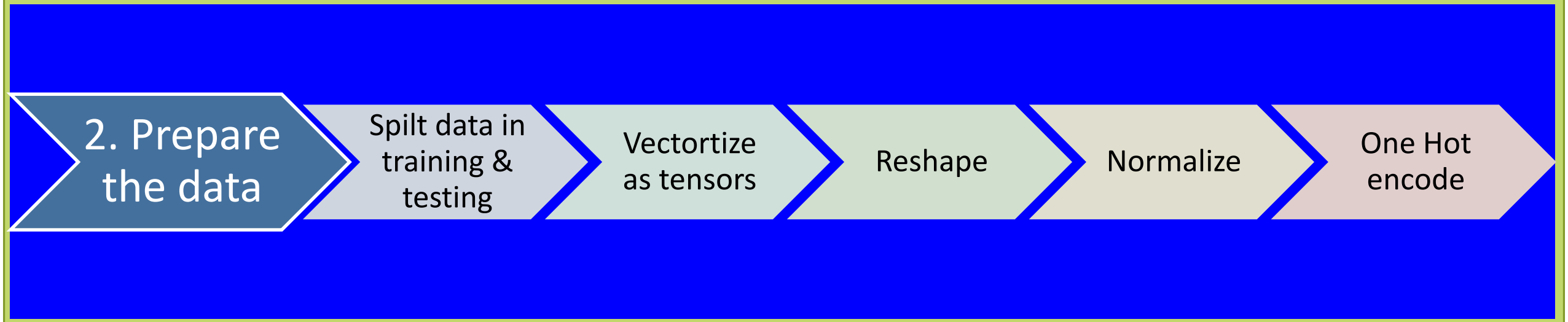
```
from matplotlib import pyplot

exampleImage  = trainY [2]

pyplot. imshow  (exampleImage )
pyplot. show()
```

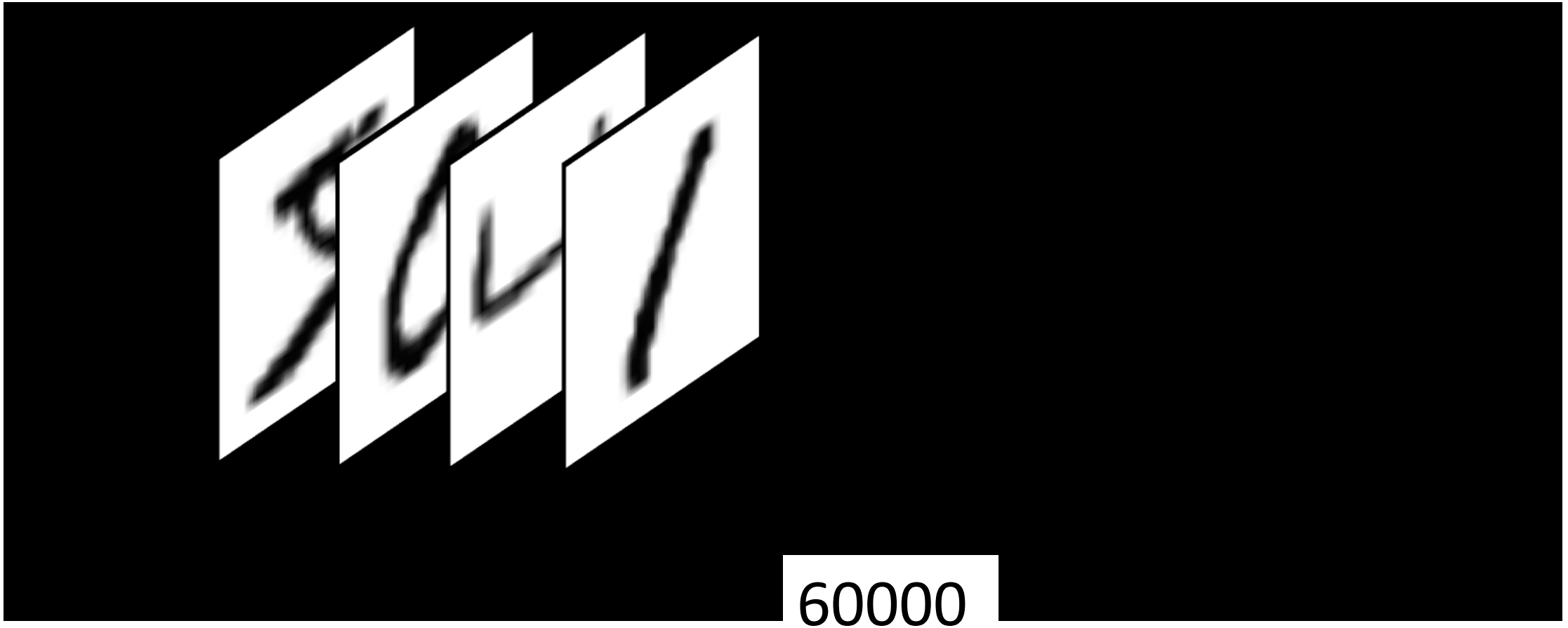


Method: 7 steps to develop your Deep Learning model

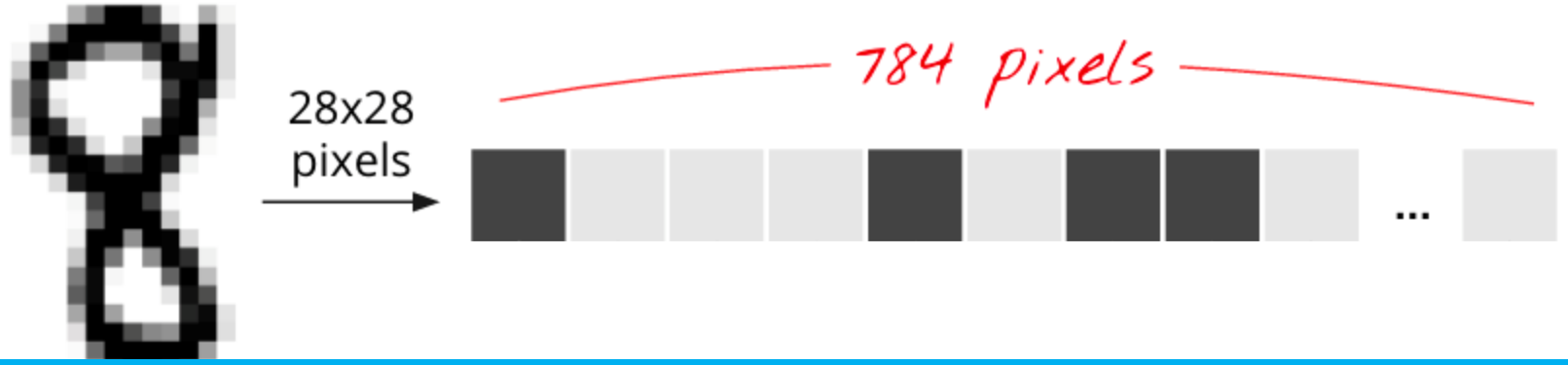


2. Prepare data





No of training records = 60000
No of test records = 10000
Shape of training image = (60000, 28, 28)



newArray = myArray . reshape (28 * 28)

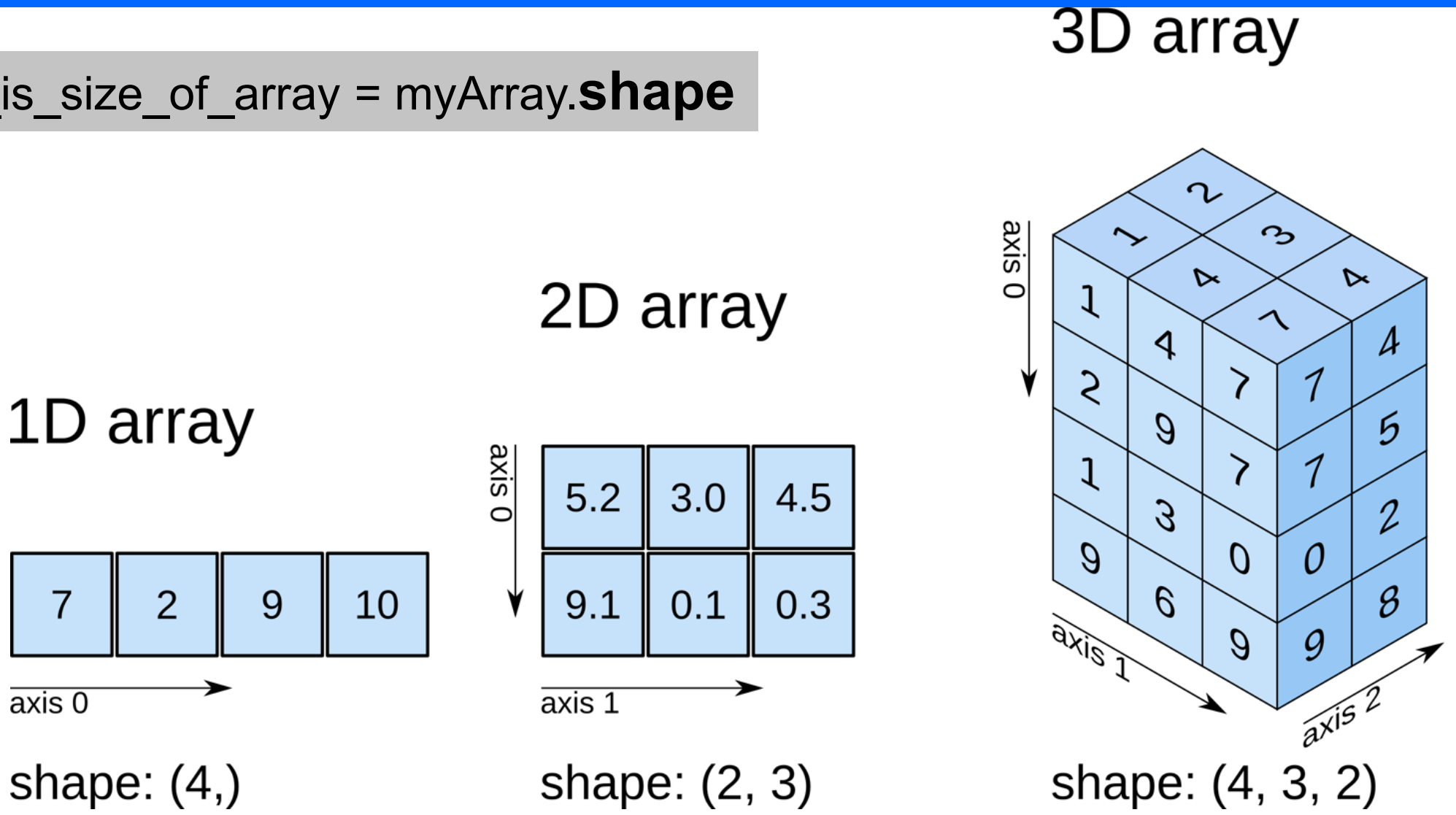
trainXready = trainX. **reshape** (800, 28*28)

trainX.**shape**

Shape of trainX: (800, 28, 28)
Shape of trainXready: (800, 784)

Multi dimensional Array == Tensor

What_is_size_of_array = myArray.**shape**



Reshape() an array

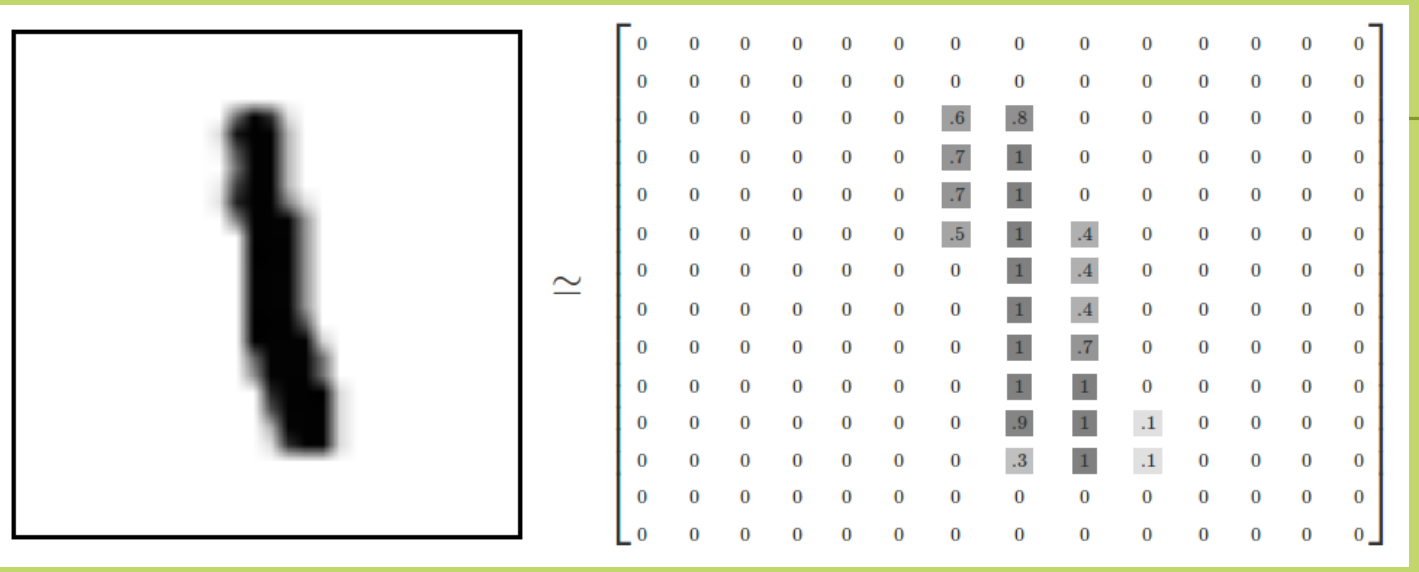
`reshape()` takes a NumPy array of one shape ...

1	2	3	4	5	6
7	8	9	10	11	12

And reconfigures it into an array with a new shape

1	2
3	4
5	6
7	8
9	10
11	12

2. Prepare the data | Spilt data in training & testing | Vectortize as tensors | Reshape | **Normalize** | One Hot encode



Data Types

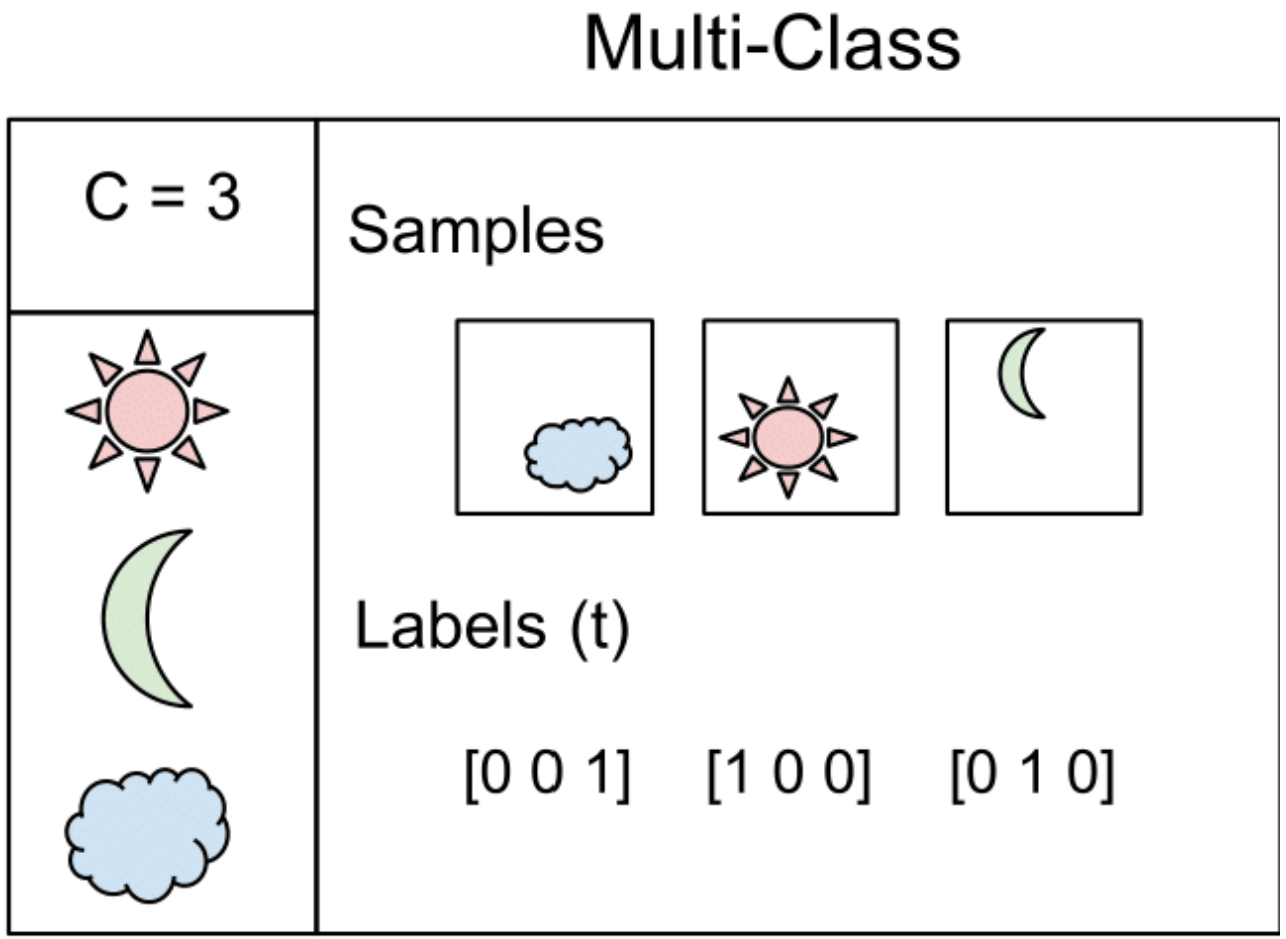
- Boolean
- Integer
- Unsigned Integer
- Float
- Complex
- String

afloat = aInteger. **astype('float32')**

newArray = myArray / 255

```
trainXready =  temptrainX.astype('float32') / 255
```

One Hot Encoding



yNew = to_categorical (y)

Data:
[2]

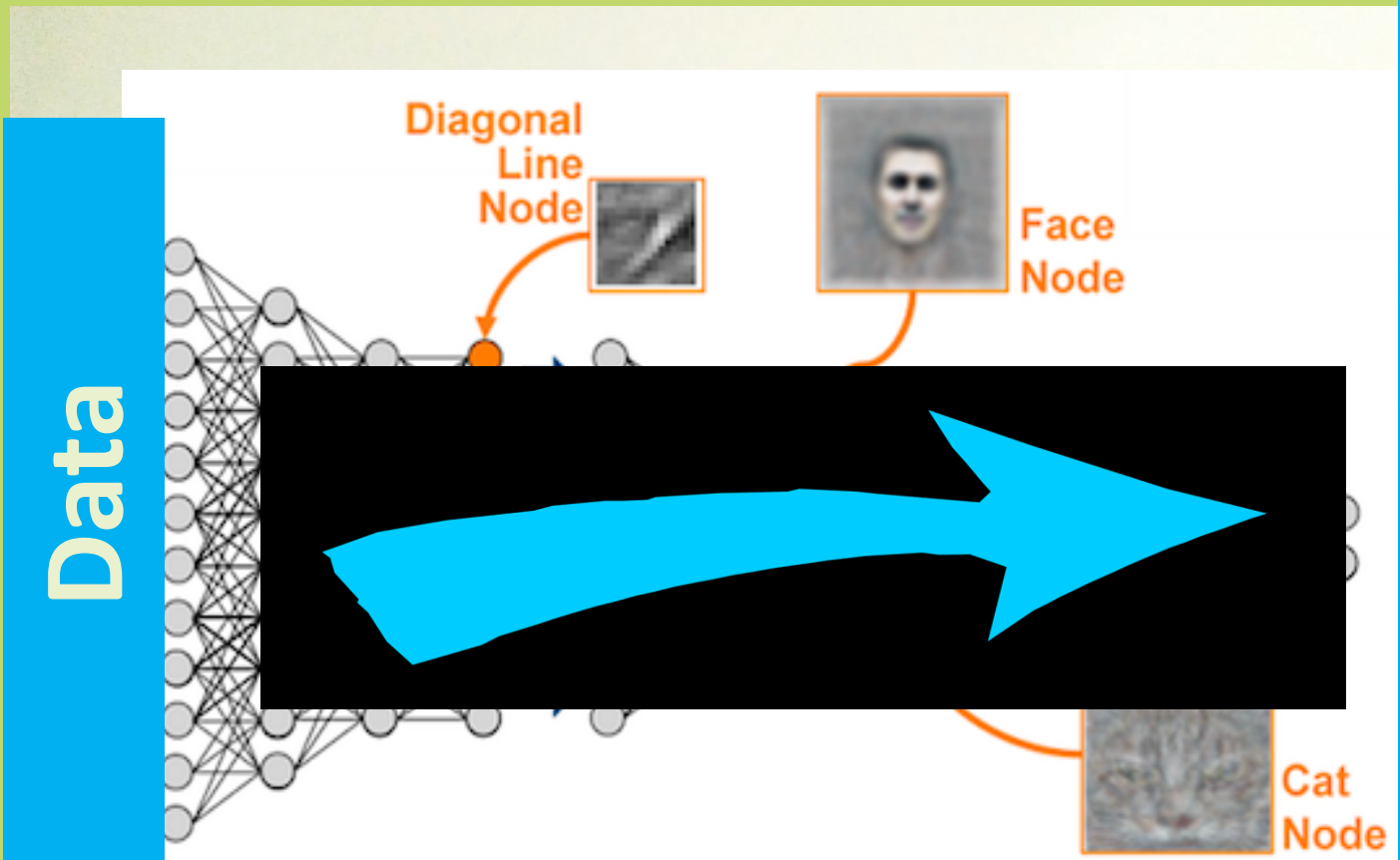
Data after **One Hot Encoding**:
[[0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]]

myArray = keras.utils. to_categorical (aInteger)

Data: [2]
Data after OHE : [[0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]]

3. Design the network

Data



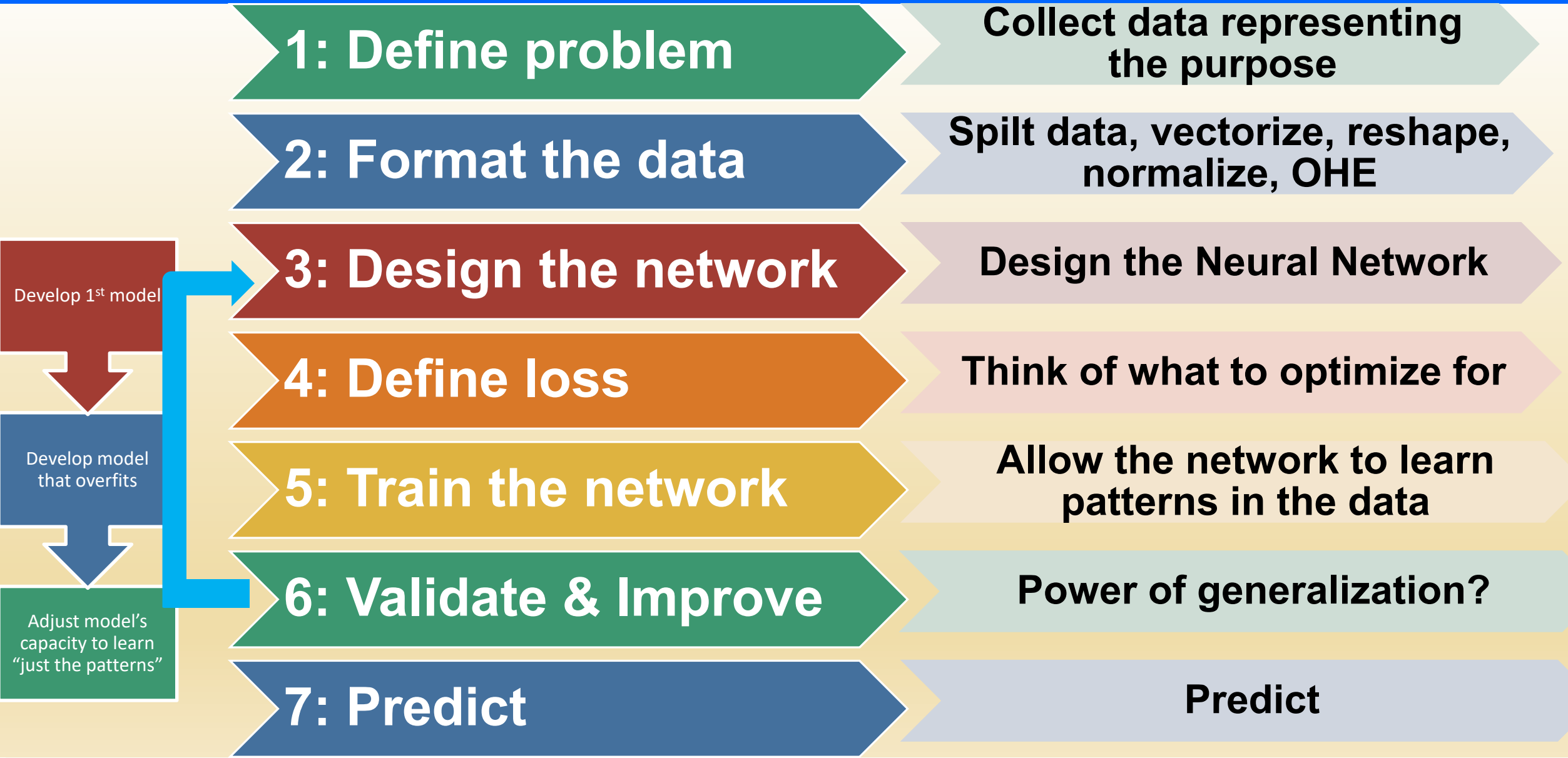
Representation of Data

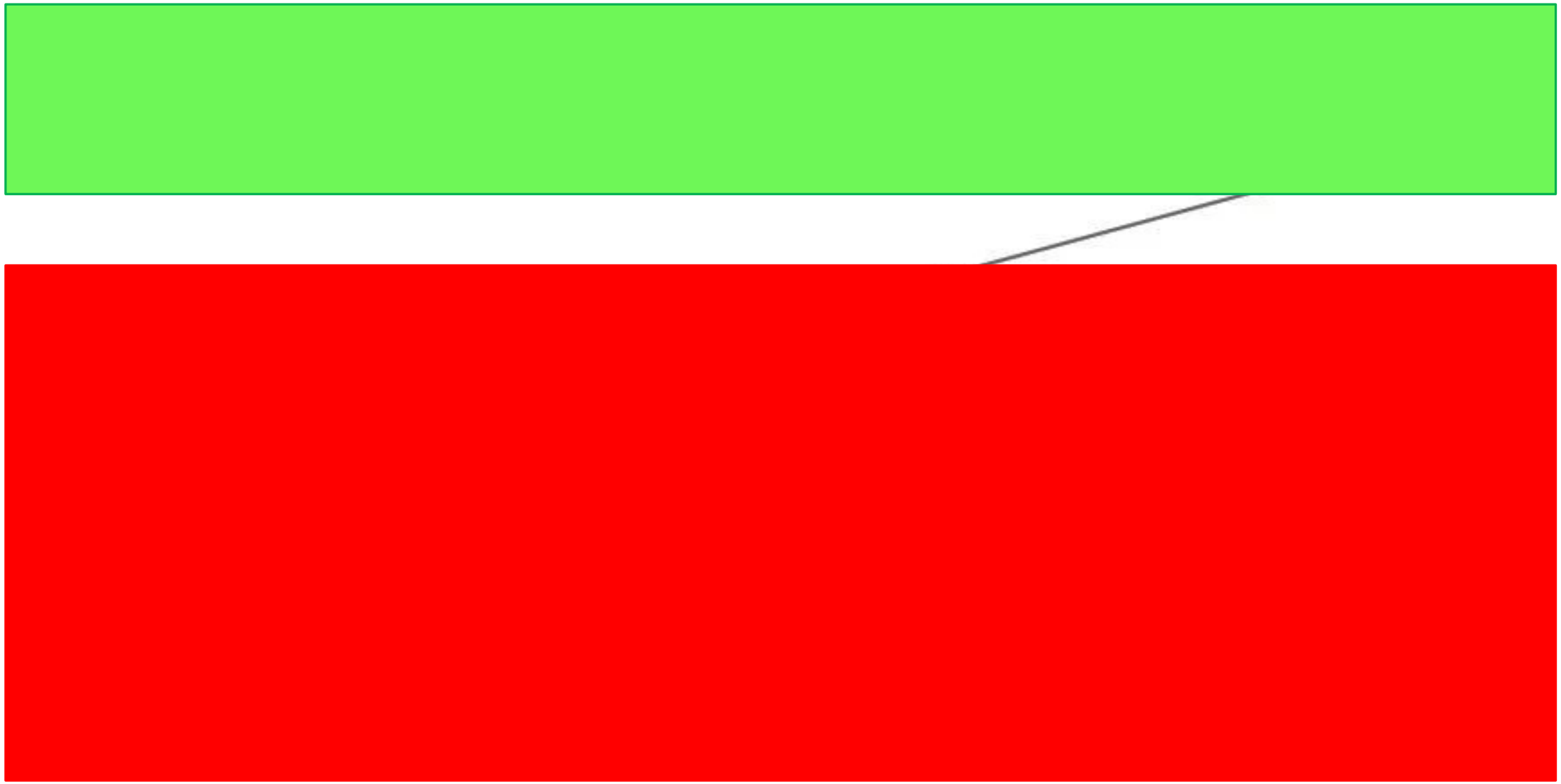
Develop 1st model

Develop model that overfits

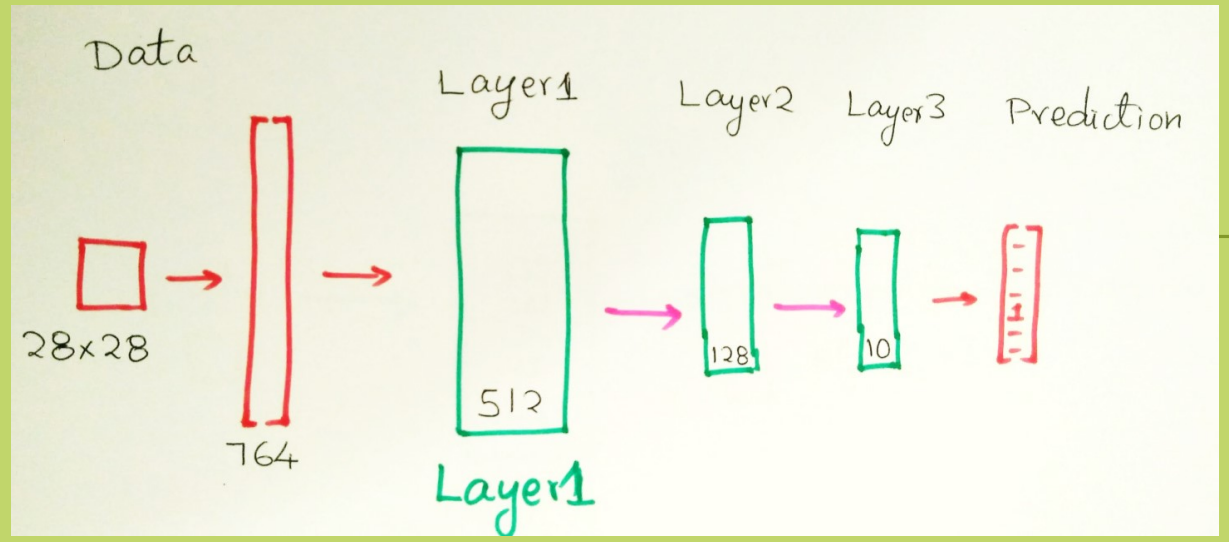
Adjust model's capacity to learn "just the patterns"

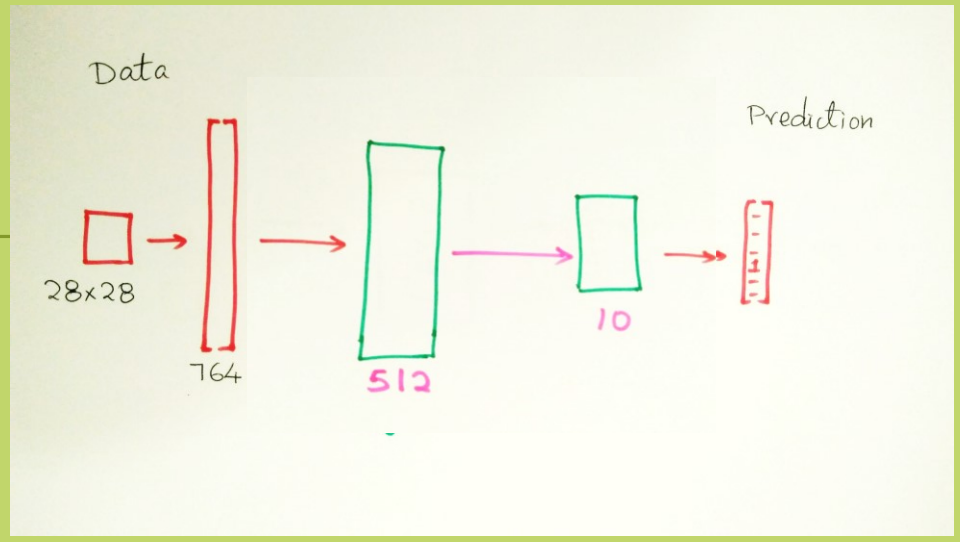
Method: 7 steps to develop your Deep Learning model





3. Design the network

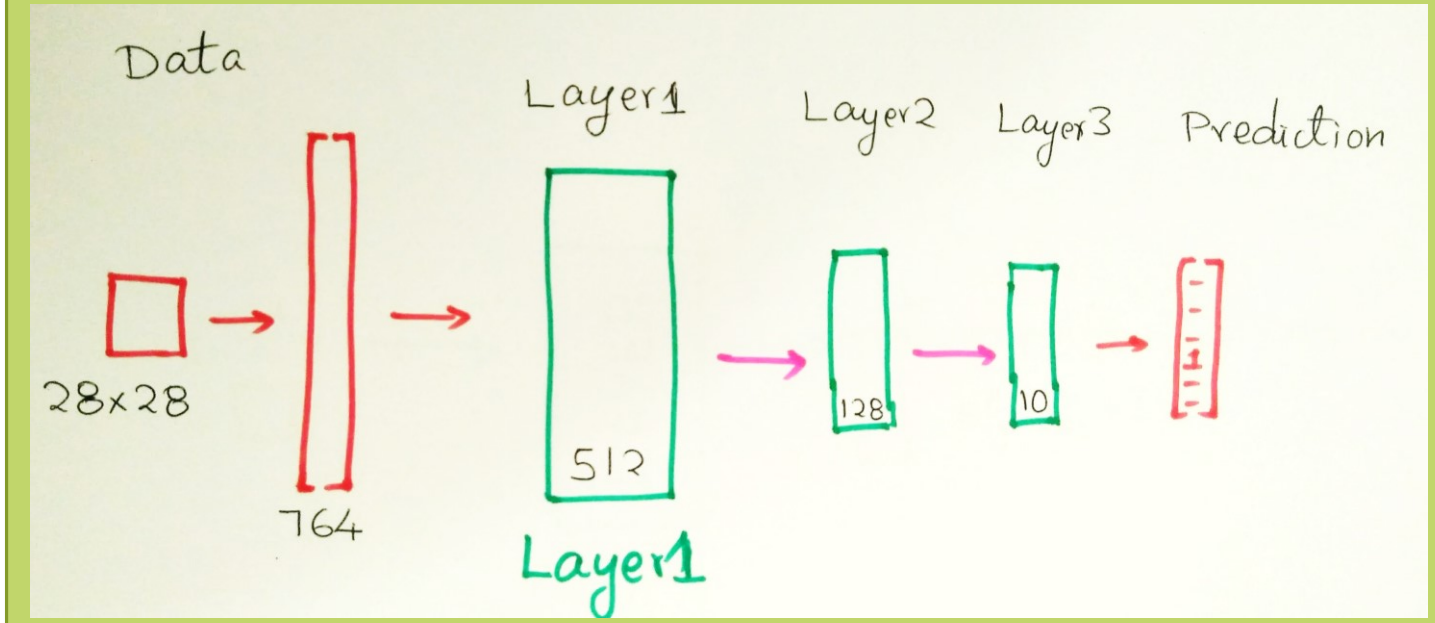


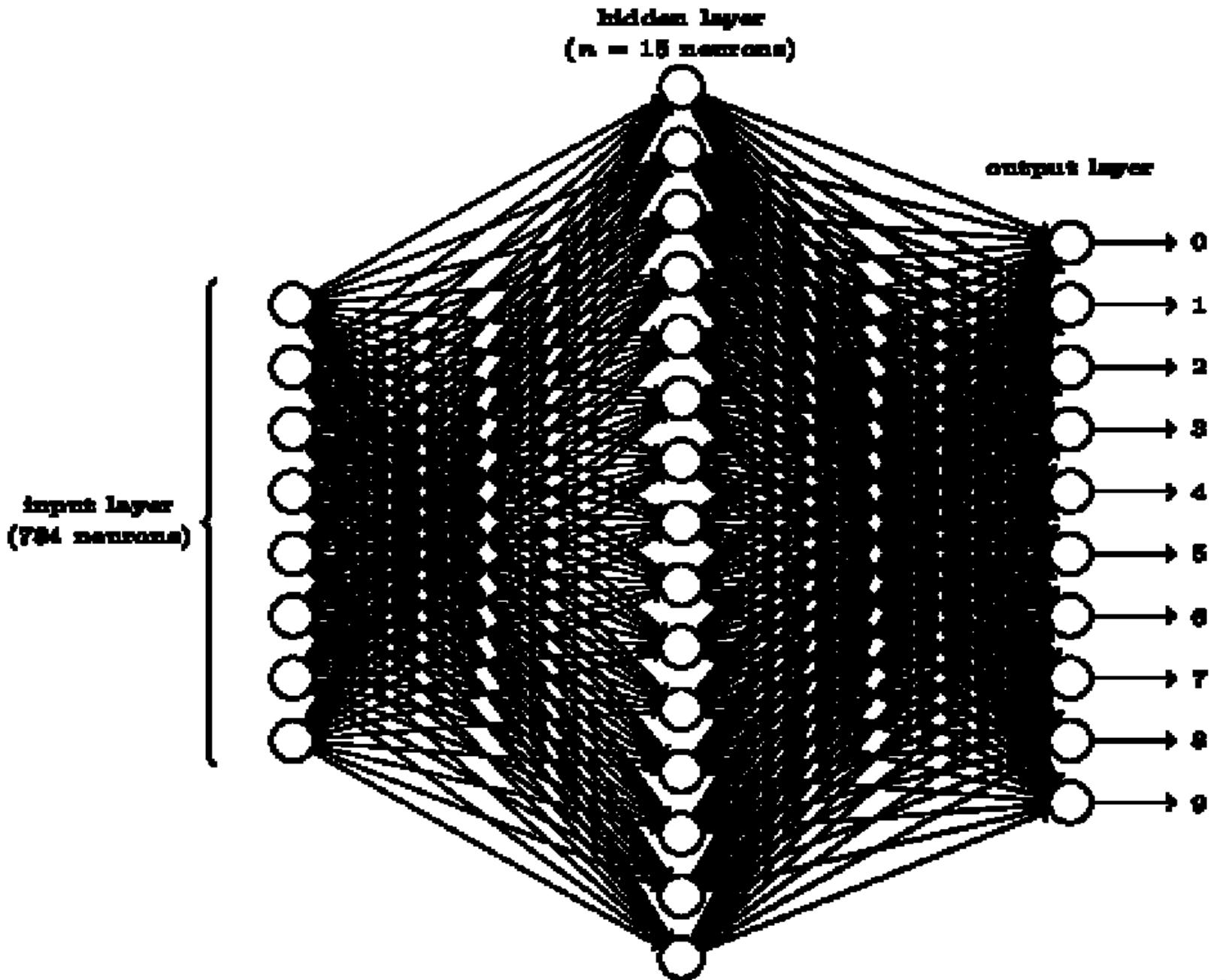



```
#Step 3: Define the network architecture
network = models.Sequential()
layer1 = layers.Dense(512, input_shape=(28*28,), activation='relu')
network.add(layer1)
layer2 = layers.Dense(128, activation='relu')
network.add(layer2)
layer3 = layers.Dense(10, activation='softmax')
network.add(layer3)
```

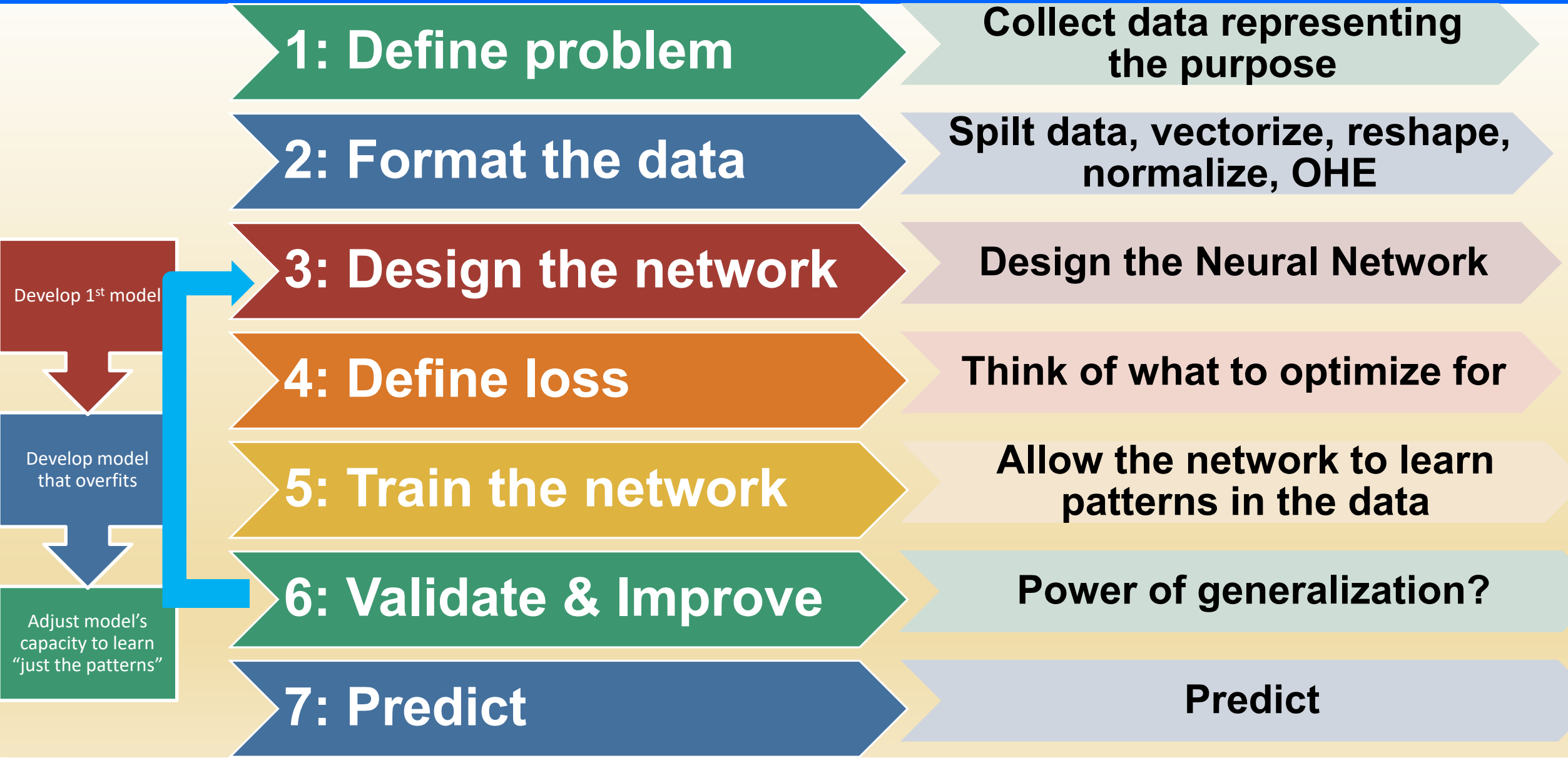
3. Design the network

```
#Step 3: Define the network architecture
network = models.Sequential()
layer1 = layers. Dense (512, input_shape=(28*28,))
network.add(layer1)
layer2 = layers. Dense (128)
network.add(layer2)
layer3 = layers. Dense (10)
network.add(layer3)
```

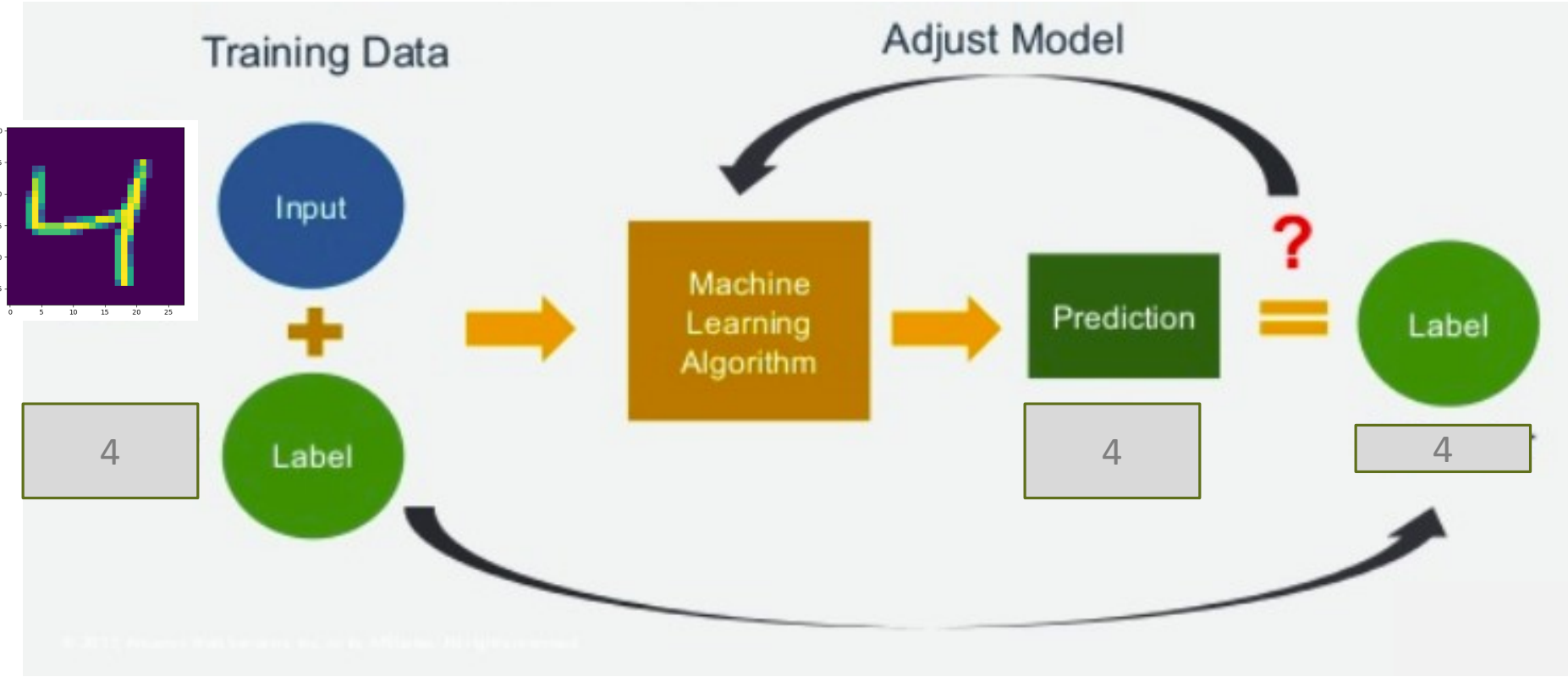



hidden layer
(n = 15 neurons)
output layer
input layer
(784 neurons)
0
1
2
3
4
5
6
7
8
9

Method: 7 steps to develop your Deep Learning model



5. Train


Training Data
Adjust Model
Input
Label
4
Machine Learning Algorithm
Prediction
4
?
Label
4

myModel.**fit**(X,Y)

```
import keras
from keras import layers

model = keras.Sequential()
model.add(layers.Dense(20, activation='relu', input_shape=(10,)))
model.add(layers.Dense(20, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))

model.fit(x, y, epochs=10, batch_size=32)
```

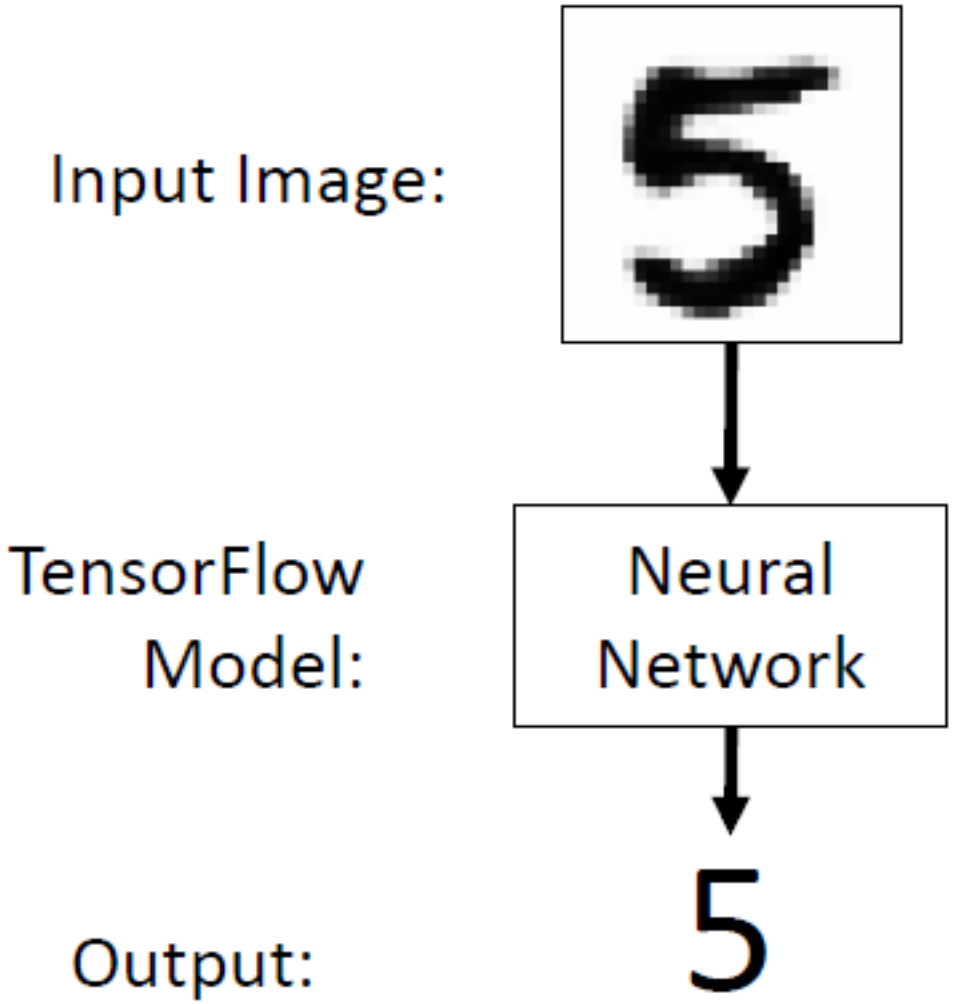
X → Y

X = image

Sample image

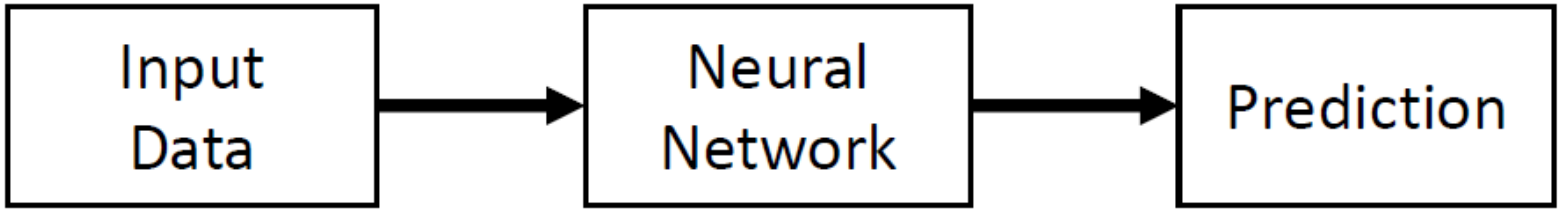
Y = 5

Label of the image

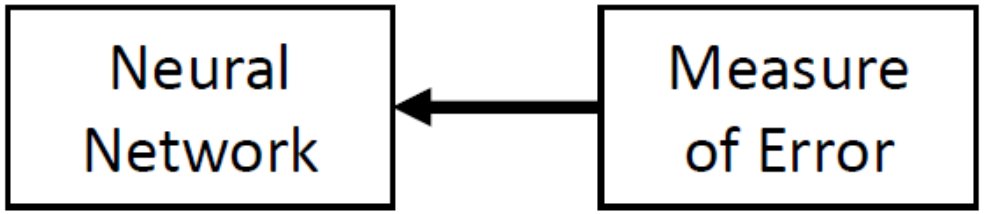


How Neural Networks Learn: Backpropagation

Forward Pass:

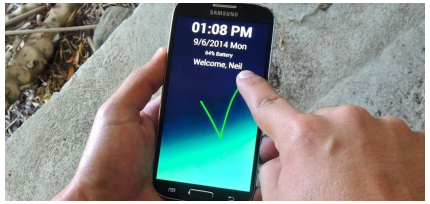


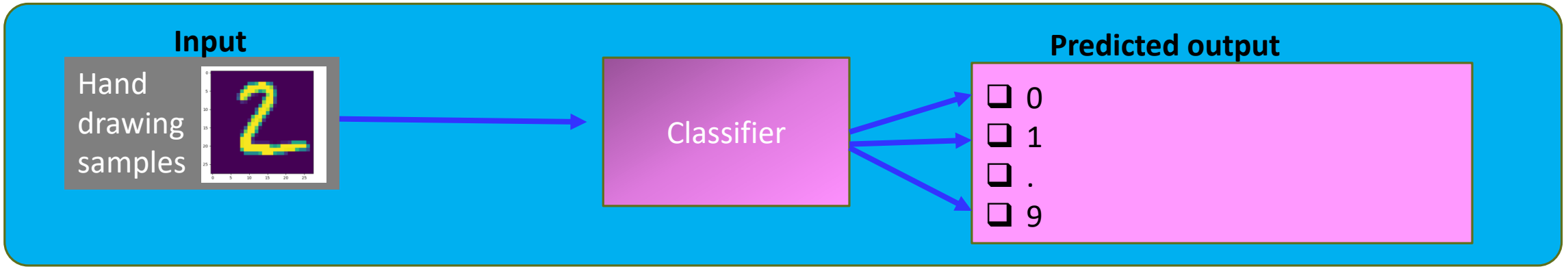
Backward Pass (aka Backpropagation):



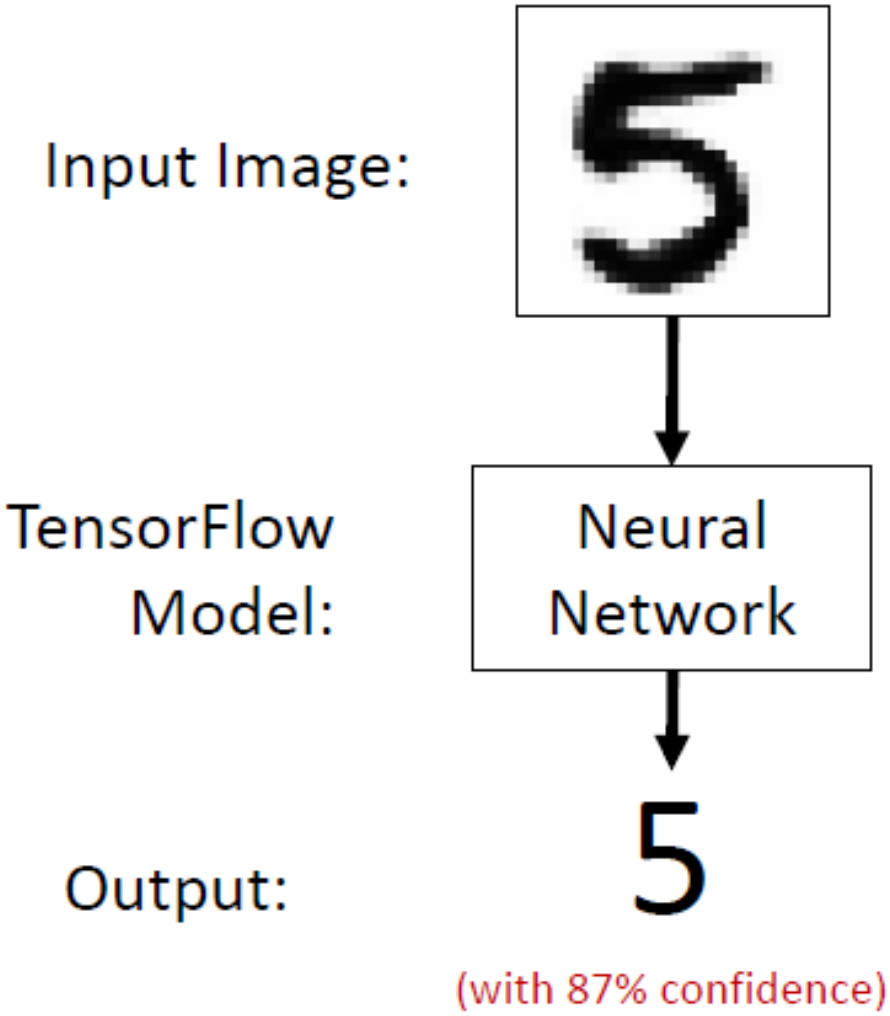
Adjust to Reduce Error

7. Predict

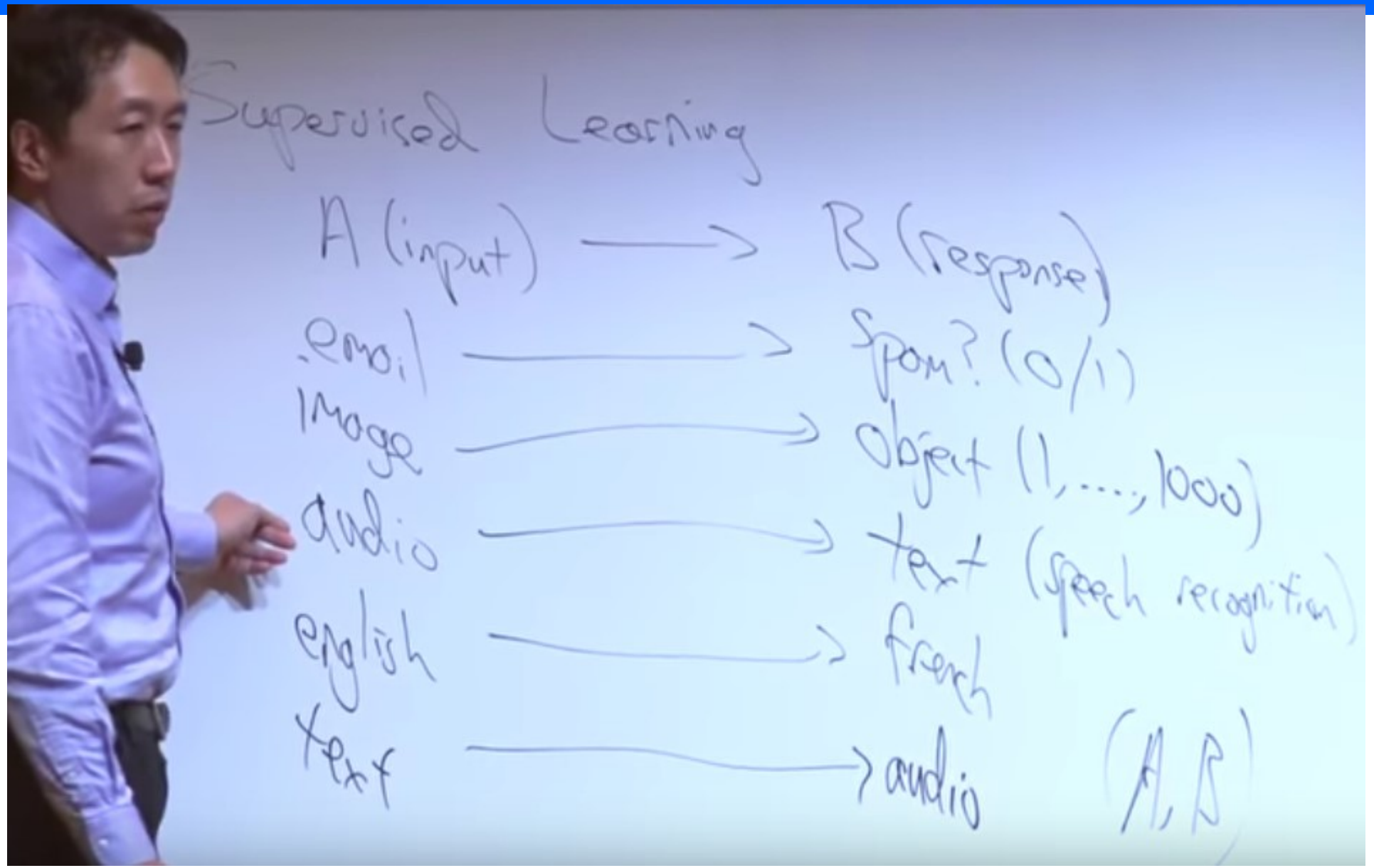

01:08 PM
9/6/2014 Mon
64% Battery
Welcome, Neil


Input
Hand drawing samples
Classifier
Predicted output
0
1
.
9

y = myModel . Predict(X)



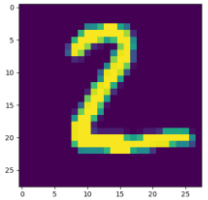
X → Y

- 
Supervised Learning
A (input)
B (response)
email
spam? (0/1)
image
object (1, ..., 1000)
audio
text (speech recognition)
english
french
text
audio
(A, B)

7. Predict

Problem: Users draws a hand drawn gestures , your app predicts it.

predictedResults = network.predict(testXready[1:3])



Predicted Results:

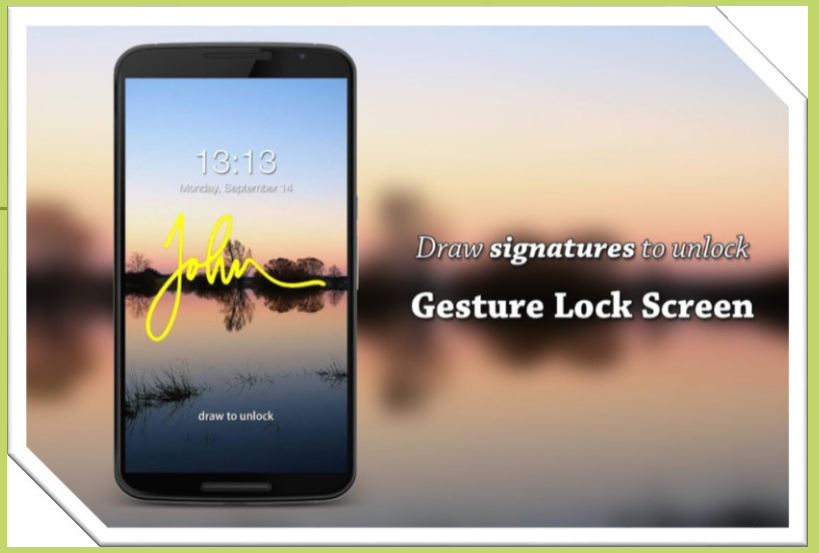
[1.35398892e-08 3.91871922e-08 9.99876499e-01 1.22501457e-04
1.77109036e-10 1.30937892e-07 2.99046292e-07 2.58880729e-11
4.58700157e-07 2.09212196e-11]

Ground Truth : [2]

Ground Truth OHE : [[0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]]

Let’s it together

- App idea: Recognize your hand gesture to unlock your smartphone (Secure unlock)
- Biz outcome: Recognize hand draw gestures to unlock your phone




```
#Step 6: Evaluate the network
metrics_test_loss, metrics_test_accuracy = network.evaluate(testXready, testYready)
```

Test accuracy is: 0.9157

Type this URL

github.com/rajagopalmotivate1/AILab/

Design of Deep Learning experiment