

# AI Prototype Class 11 code deep learning

🕒 Created	@February 19, 2024 2:43 PM
📁 Class	AI

Keras: API ระดับสูงสำหรับ TensorFlow | TensorFlow Core

ข้อมูลเบื้องต้นเกี่ยวกับ Keras ซึ่งเป็น API ระดับสูงสำหรับ TensorFlow

🔗 <https://www.tensorflow.org/guide/keras?hl=th>



Google Colaboratory

🔗 [https://colab.research.google.com/drive/1sQ7SugSz\\_5l0j0JrhZxmqufq3NwAQWv7](https://colab.research.google.com/drive/1sQ7SugSz_5l0j0JrhZxmqufq3NwAQWv7)



## Network Deeplearning

1. Sequential (ง่ายๆ สร้างได้แบบไม่ซับซ้อน (วิ่งเป็นเส้นตรง)
2. Functional ( ยากขึ้น ปรับเส้นทางให้มีความซับซ้อนได้)

Deep Learning inception



```

1 #Define Sequential model with 3 Layers
2 model = keras.Sequential(
3     [
4         keras.Input(shape=(4,)), #4 Node
5         layers.Dense(2, activation="relu", name="layer1"), #Dense = fully connected
6         layers.Dense(3, activation="relu", name="layer2"),
7         layers.Dense(4, name="output") #4 class
8     ]
9 )

```

Model: "sequential"

Layer (type)	Output Shape	Param #
layer1 (Dense)	(None, 2)	10
layer2 (Dense)	(None, 3)	9
output (Dense)	(None, 4)	16
Total params: 35 (140.00 Byte)		
Trainable params: 35 (140.00 Byte)		
Non-trainable params: 0 (0.00 Byte)		

```

model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))

```

```
model.summary()
```

---

Model: "sequential\_3"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d_2 (MaxPooling 2D)	(None, 15, 15, 32)	0
conv2d_4 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_3 (MaxPooling 2D)	(None, 6, 6, 64)	0
conv2d_5 (Conv2D)	(None, 4, 4, 64)	36928
flatten_2 (Flatten)	(None, 1024)	0
dense_8 (Dense)	(None, 64)	65600
dense_9 (Dense)	(None, 10)	650

---

Total params: 122,570  
Trainable params: 122,570  
Non-trainable params: 0

- นำภาพผ่าน Conv2D size 32 layer มี 3 channel 3\*3 ผ่าน filter 3\*3
- max pooling ทุกๆ 2 pixel ทั้งแนวนอน แนวตั้ง สรุปลเหลือ 1
- นำภาพผ่าน Conv2D size 64 layer

## ต่อด้วย flatten และ dense เพื่อทำนาย

```
[ ]: model.add(layers.Flatten())
      model.add(layers.Dense(64, activation='relu'))
      model.add(layers.Dense(10))
```

- Flatten เอาพารามิเตอร์มาจัดเป็นเส้นตรง
- ผ่าน Dense = fully connected
- output = 10 class

# Functional network

---

## Data pipeline

- data ในแต่ละรอบให้โหลดมาทีละส่วนๆ
  - Folder in google drive

### 1. Image ornamentation

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')
```