# Artificial Neural Networks & Deep Learning

## Practice #8
## (2 points)

# Practice #8 Practice – Important Notice

- You will tune hyperparameters of a CNN model for CIFAR10.
- Use the codes in the lecture slide, “CNN(4)”.

- I provided “chapter5_1_cifar10.py”.
- Follow the instruction in this file, “step-by-step” and collect the results.
- Each run would take 5-10 minutes (even with a GPU). Please start this practice & quiz early.

- To do the practice in this file, **I recommend to GPU use through a torque server.** See the lecture 8-3 and 3.
    - Copy and adapt conda_job.sh to your codes!

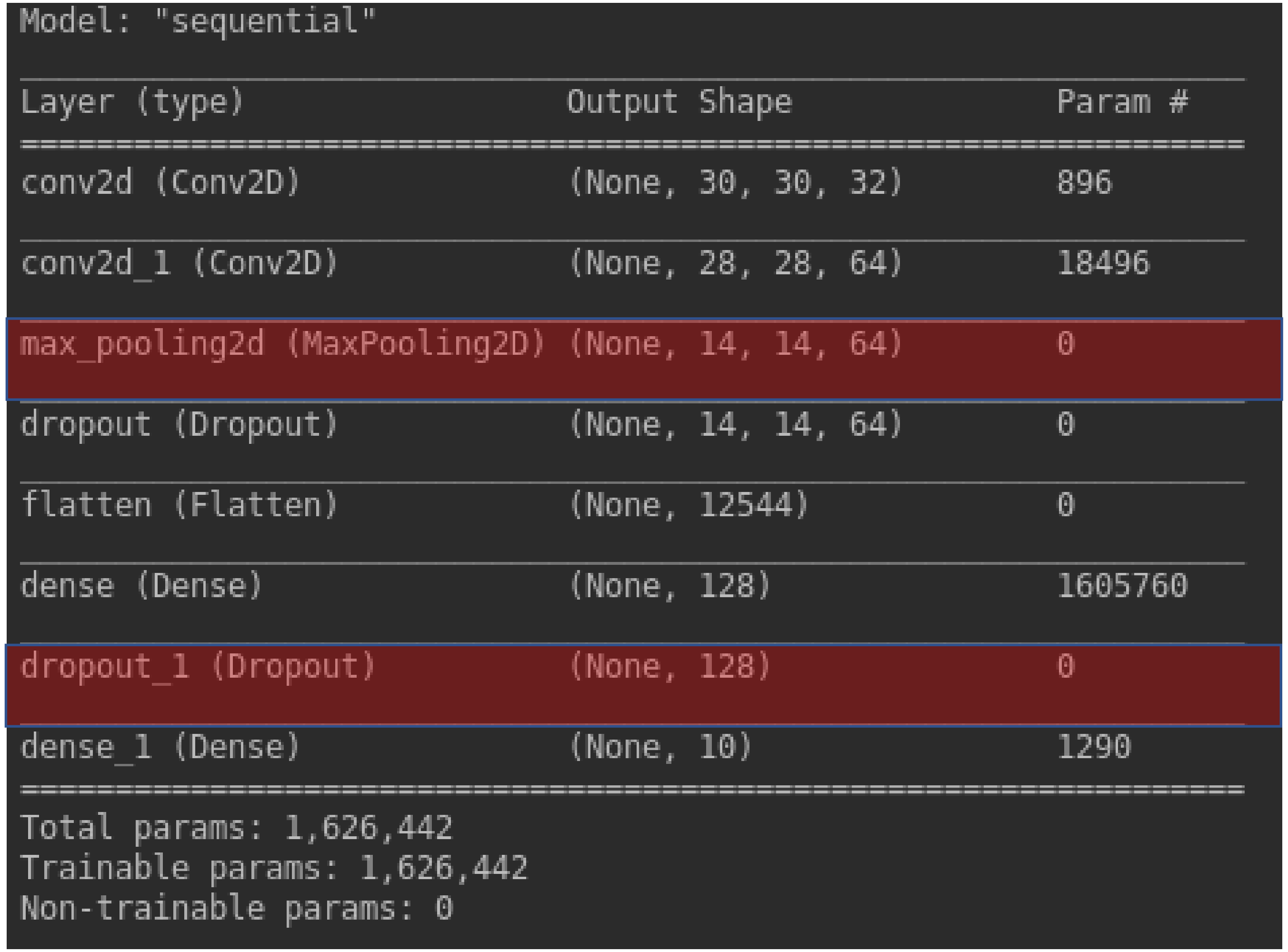
# Practice #8 – Q1 (0.25 points)

- We will add “Dropout” layers after the max-pooling layer and the hidden layer of MLP.
- Add the following lines to the right locations.

```
model.add(layers.Dropout(0.25)) # after max-pooling
model.add(layers.Dropout(0.5))  # after the hidden layer of MLP
```

- **Show the modified “build_model” function.**

# Practice #8 – Q1 (0.25 points)



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 30, 30, 32)        896
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 28, 28, 64)        18496
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 14, 14, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 14, 14, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 12544)             0
_________________________________________________________________
dense (Dense)                (None, 128)               1605760
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1290
=================================================================
Total params: 1,626,442
Trainable params: 1,626,442
Non-trainable params: 0
```

**Added**

**Added**

# Practice #8 – Q2 (0.75 points)

- When we use the dropout, we need more time to learn the model. So, please increase the number of maximum epochs to 50.

- Run the code.
- **What is the test accuracy? Attach the loss graph.**

# Practice #8 – Q3 (0.5 points)

- The above learning curve is oscillating.
- Maybe it's better to decrease the learning rate. Change the learning rate of RMSprop to 0.0001.
- Also, the smaller size of minibatch also helps to stabilize the learning procedure. Set minibatch to 32. (it will increase the learning time about three times)

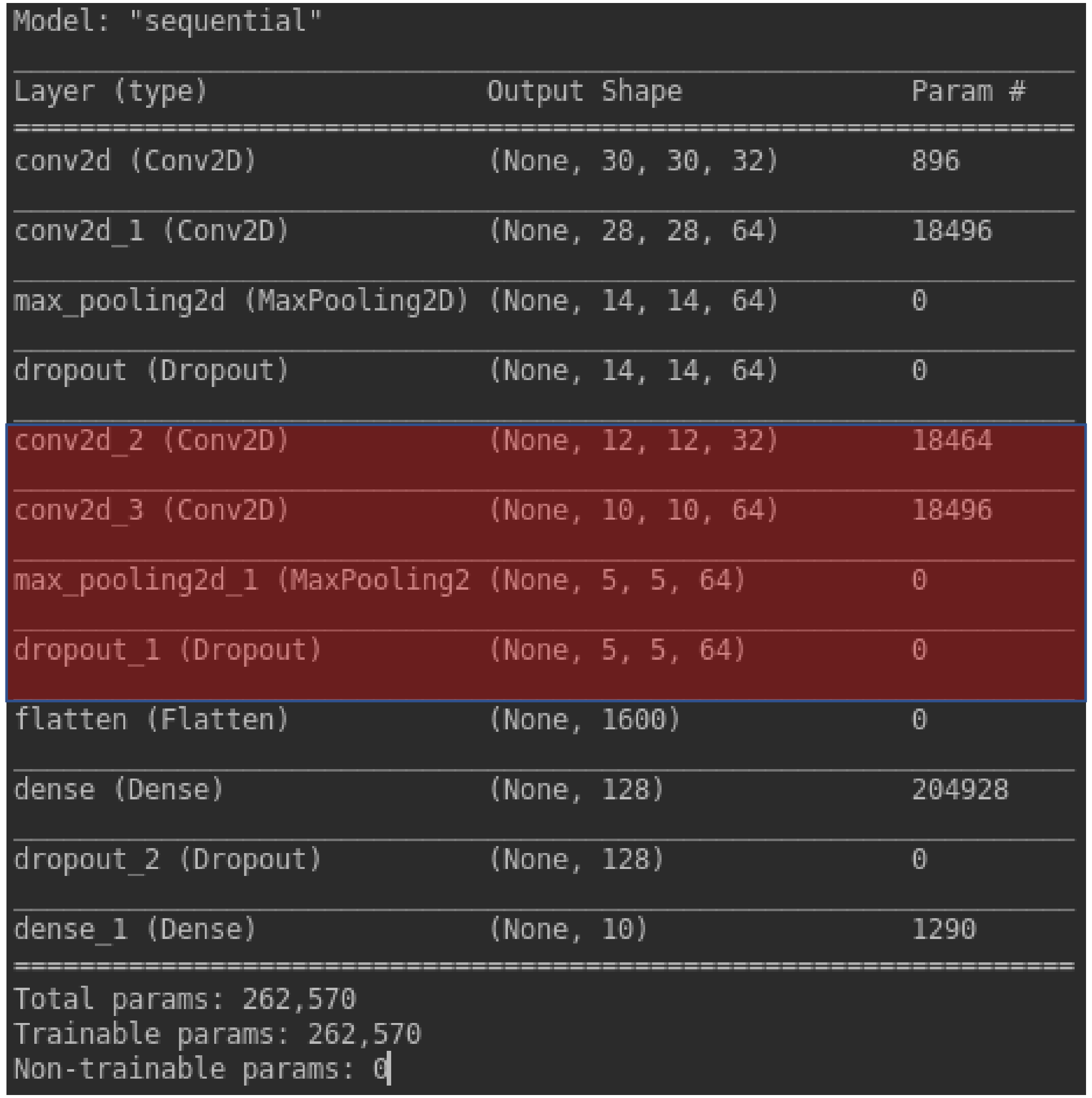
- Run the code.
- **What is the test accuracy? Attach the loss graph.**

# Practice #8 – Q4 (0.5 points)

- As I told before, the other approach of machine learning is **"high-capacity model with strong regularization".**
- Indeed, it is not recommended to add dropout layers to the small-sized NN model.
- Do you think our model is large enough? Is there any chance that our model is underfitting currently?
- To test it, let's increase model capacity. (See the structure in the next page)
- Since the model becomes more complex, we need more time to learn. Set the number of maximum epochs to 100.

- **What is the test accuracy? Attach the loss graph.**
- **Do you think the final model is better than what we used in the lecture (the original model)?**

# Practice #8 – Q4 (0.5 points)



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 30, 30, 32)        896
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 28, 28, 64)        18496
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 14, 14, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 12, 12, 32)        18464
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 10, 10, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 5, 5, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 1600)              0
_________________________________________________________________
dense (Dense)                (None, 128)               204928
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1290
=================================================================
Total params: 262,570
Trainable params: 262,570
Non-trainable params: 0
```

Added

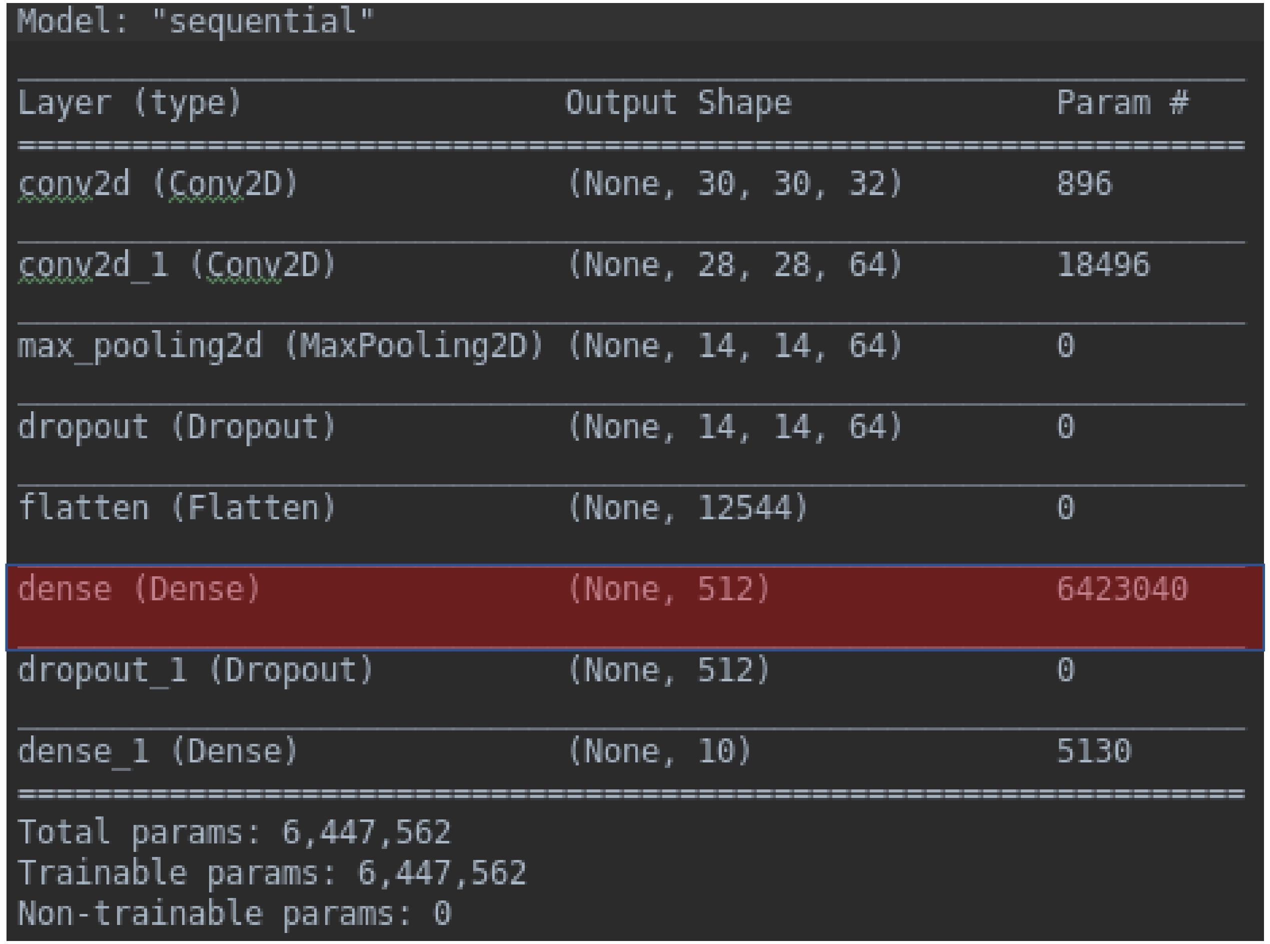
Copy-and-paste two cond2D layers, maxpooling layers and dropout layers before the MLP layer.

# Practice #8 – Extra (1 points)

- There is another possible way to have “high-capacity model”.
- Let’s try to increase the number of hidden neurons in MLP four times (512) without adding convolution layers. (see structure in the next page)

- Run the code.
- **What is the test accuracy? Attach the loss graph.**
- **How much is it improved?**
- **Is it worthy? (Consider the number of parameters, total learning time etc)**

# Practice #8 – Extra (1 points)



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 30, 30, 32)        896
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 28, 28, 64)        18496
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 14, 14, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 14, 14, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 12544)             0
_________________________________________________________________
dense (Dense)                (None, 512)               6423040
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                5130
=================================================================
Total params: 6,447,562
Trainable params: 6,447,562
Non-trainable params: 0
```

**Modified**