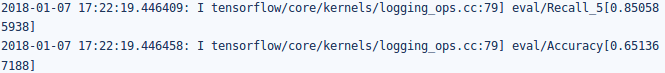
# 作业具体过程

## 作业1

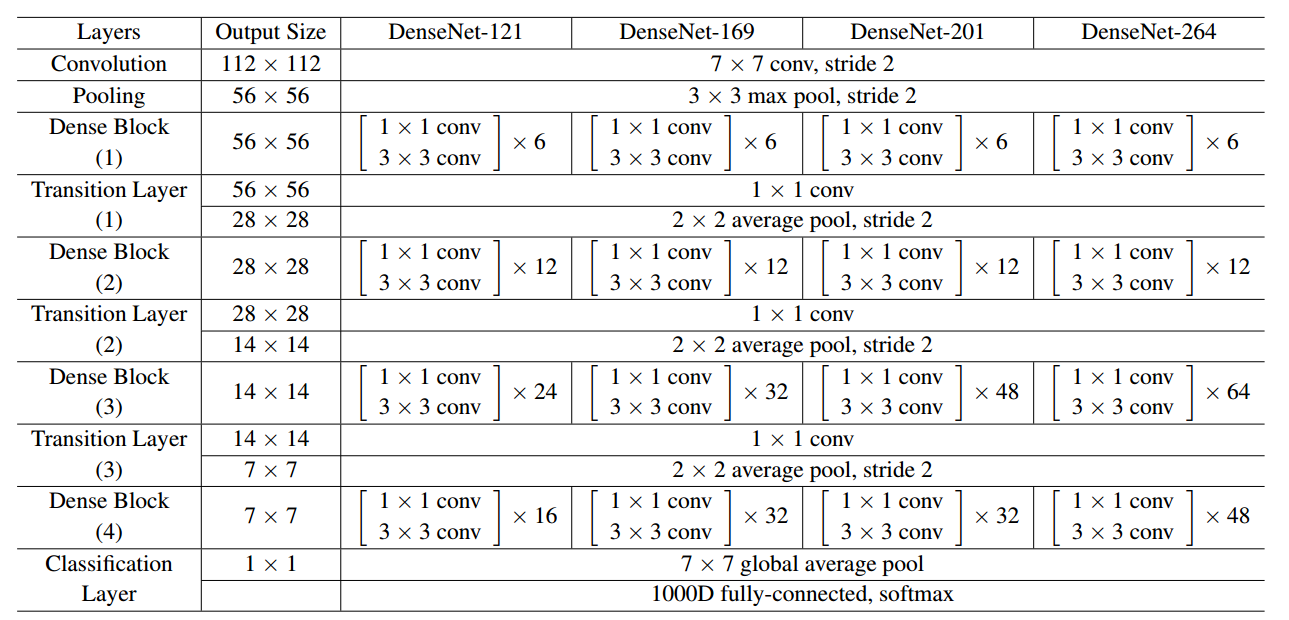
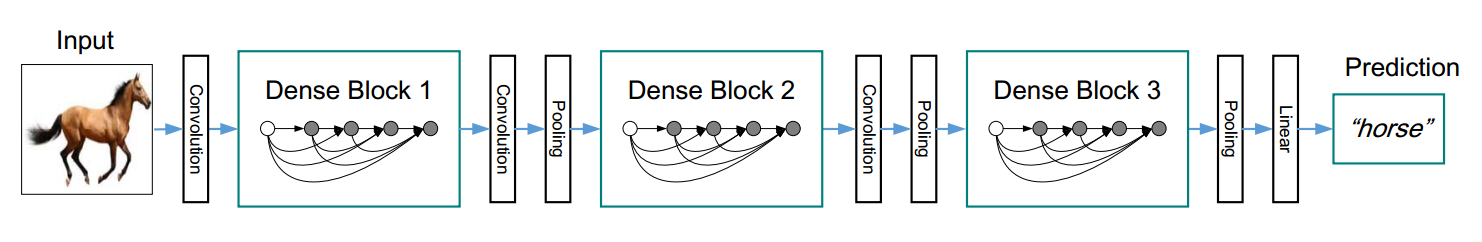
1. 将代码上传到github下面。
2. 建立模型---->>>将github代码，数据放到相应的位置-------->>>准备好入口脚本----->>>>设置好相应的参数------>>>>耐心等待即可。
3. 以下为相应参数的意义
4. 调参心得
5. 结果截图



1. 相应的model位置：

https://www.tinymind.com/executions/8obnj8it

## 作业2



# 论文解析

一、Convolution:

输入的图片的size为：224×224 input images

第一次的卷积为：initial convolution layer comprises 2k，相应的kernel的size 7×7 with stride 2

其他的卷积的设置为k

1. Pooling
2. Dense Block层

1×1 convolution减少input feature-maps。并且在进行1x1卷积时需要依次进行batch\_normal和relu和1x1卷积dropout

备注：x‘ = H‘([x0; x1; : : : ; x‘−1])

四、Transition Layer层

在Transition Layer层需要注意为了model compactness需要设置θ，然后减少input feature-maps

1. Classification Layer

相对简单参考inception\_v4即可。

备注：

(H,W)output=( (H,W)input - (H,W)kernel + padding \*2 )/stride + 1这是老师给出的公式

本次论文应该用的公式时

Padding=’VALID’

New\_height = new\_height=(W-F+1)/S（结果向上取整）

Padding=’SAME’

New\_height = new\_height=W/S(结果向上取整）

# 二、具体代码

# Put your code here.

#Convolution 2k convolutions of size 7×7 with stride 2

net=slim.conv2d(images, 2\*growth, [7,7],padding='SAME', scope='Conv2d\_0a\_7x7')

end\_points['Conv2d\_0a\_7x7'] = net

#Pooling the number of feature-maps in all other layers also follow from setting k

#Pooling out size 56x56,kernel size[3,3] stride 2,method pooling:max

#Pooling Layer

net=slim.max\_pool2d(net,[3,3],stride=2,padding='VALID',scope='MaxPool\_0a\_3x3')

end\_points['MaxPool\_0a\_3x3'] = net

#Dense Block 1

#layers:6

net=block(net,6,growth,'Dense\_Block\_1')

end\_points['Dense\_Block\_1'] = net

#Transition Layer 1

output\_num\_1=reduce\_dim(net)

net=slim.conv2d(net,output\_num\_1,[1,1],padding='SAME',scope='Conv2d\_1a\_1x1')

end\_points['Conv2d\_1a\_1x1'] = net

net=slim.avg\_pool2d(net,[2,2],stride=2,padding='VALID',scope='AvgPool\_1a\_2x2')

end\_points['AvgPool\_1a\_2x2'] = net

#Dense Block 2

#layers:12

net=block(net,12,growth,'Dense\_Block\_2')

end\_points['Dense\_Block\_2'] = net

#Transition Layer 2

output\_num\_2=reduce\_dim(net)

net=slim.conv2d(net,output\_num\_2,[1,1],padding='SAME',scope='Conv2d\_2a\_1x1')

end\_points['Conv2d\_2a\_1x1'] = net

net=slim.avg\_pool2d(net,[2,2],stride=2,padding='VALID',scope='AvgPool\_2a\_2x2')

end\_points['AvgPool\_2a\_2x2'] = net

#Dense Block 3

#layers:32

net=block(net,32,growth,'Dense\_Block\_3')

end\_points['Dense\_Block\_3'] = net

#Transition Layer 3

output\_num\_3=reduce\_dim(net)

net=slim.conv2d(net,output\_num\_3,[1,1],padding='SAME',scope='Conv2d\_3a\_1x1')

end\_points['Conv2d\_3a\_1x1'] = net

net=slim.avg\_pool2d(net,[2,2],stride=2,padding='VALID',scope='AvgPool\_3a\_2x2')

end\_points['AvgPool\_3a\_2x2'] = net

#Dense Block 4

#layers:32

net=block(net,32,growth,'Dense\_Block\_4')

end\_points['Dense\_Block\_4'] = net

with tf.variable\_scope('Logits'):

#7x7x128

net=slim.avg\_pool2d(net,[7,7],padding='VALID',scope='AvgPool\_3a\_7x7')

#1x1x128

end\_points['AvgPool\_4a\_7x7'] = net

net = slim.flatten(net, scope='PreLogitsFlatten')

end\_points['PreLogitsFlatten'] = net

#128

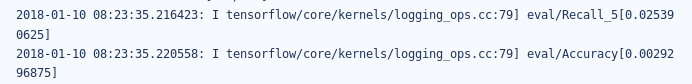
logits = slim.fully\_connected(net, num\_classes, activation\_fn=None,scope='Logits')

end\_points['Logits'] = logits

end\_points['Predictions'] = tf.nn.softmax(logits, name='Predictions')

##########################

# 三、结果展示



# model位置

https://www.tinymind.com/executions/uvjqdoyp