

# 实验一

---

实现 ResNet 网络子模块，并在 Kaggle 猫/狗数据集上进行训练和测试

## 1.加载 keras 模块

---

21

```
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Model
from keras.layers import Conv2D, AveragePooling2D, Input, BatchNormalization
from keras.layers import Activation, Dropout, Flatten, Dense, Concatenate
from keras.utils import to_categorical
from keras.preprocessing.image import img_to_array
from keras.applications.vgg16 import preprocess_input
from keras import backend as K
import numpy as np
No output
```

## 定义 Inception 网络结构

33

```
img_width, img_height = 50, 50

if K.image_data_format() == 'channels_first':
    input_shape = (3, img_width, img_height)
    bn_axis = 1
else:
    input_shape = (img_width, img_height, 3)
    bn_axis = 3

input_img = Input (shape = input_shape)

#ResNet Module
x = layers.Conv2D(64, (1, 1), padding='same',
kernel_initializer='he_normal')(input_img)
x = layers.BatchNormalization(axis=bn_axis)(x)
x = layers.Activation('relu')(x)

x = layers.Conv2D(64, (3,3), padding='same',
kernel_initializer='he_normal')(x)
```

```

x = layers.BatchNormalization(axis=bn_axis)(x)
x = layers.Activation('relu')(x)

x = layers.Conv2D(256, (1, 1), padding='same',
kernel_initializer='he_normal')(x)
x = layers.BatchNormalization(axis=bn_axis)(x)

x = layers.add([x, input_img])
x = layers.Activation('relu')(x)

#fully connected layer
out = Flatten()(out)
out = Dense(48, activation='relu')(out)

#output layer
out = Dense(2, activation='softmax')(out)

model = Model(x, out)

model.compile(loss='categorical_crossentropy',
              optimizer='rmsprop',
              metrics=['accuracy'])

```

*No output*

## 查看 model 架构

34

```

model.summary()
Model: "model_17"

```

Layer (type)	Output Shape	Param #	Connected to
=====			
input_18 (InputLayer)	(None, 50, 50, 3)	0	
=====			
conv2d_109 (Conv2D)	(None, 50, 50, 16)	48	input_18[0][0]
=====			
conv2d_111 (Conv2D)	(None, 50, 50, 16)	48	input_18[0][0]
=====			

batch_normalization_102 (BatchN	(None, 50, 50, 16)	64	
conv2d_109[0][0]			
<hr/>			
batch_normalization_104 (BatchN	(None, 50, 50, 16)	64	
conv2d_111[0][0]			
<hr/>			
activation_102 (Activation)	(None, 50, 50, 16)	0	
batch_normalization_102[0][0]			
<hr/>			
activation_104 (Activation)	(None, 50, 50, 16)	0	
batch_normalization_104[0][0]			
<hr/>			
average_pooling2d_25 (AveragePo	(None, 50, 50, 3)	0	input_18[0][0]
<hr/>			
conv2d_108 (Conv2D)	(None, 50, 50, 16)	48	input_18[0][0]
<hr/>			
conv2d_110 (Conv2D)	(None, 50, 50, 48)	6912	
activation_102[0][0]			
<hr/>			
conv2d_112 (Conv2D)	(None, 50, 50, 24)	9600	
activation_104[0][0]			
<hr/>			
conv2d_113 (Conv2D)	(None, 50, 50, 16)	48	
average_pooling2d_25[0][0]			
<hr/>			
batch_normalization_101 (BatchN	(None, 50, 50, 16)	64	
conv2d_108[0][0]			
<hr/>			
batch_normalization_103 (BatchN	(None, 50, 50, 48)	192	
conv2d_110[0][0]			
<hr/>			
batch_normalization_105 (BatchN	(None, 50, 50, 24)	96	
conv2d_112[0][0]			

---

---

batch_normalization_106	(BatchN	(None, 50, 50, 16)	64	
-------------------------	---------	--------------------	----	--

conv2d\_113[0][0]

---

---

activation_101	(Activation)	(None, 50, 50, 16)	0	
----------------	--------------	--------------------	---	--

batch\_normalization\_101[0][0]

---

---

activation_103	(Activation)	(None, 50, 50, 48)	0	
----------------	--------------	--------------------	---	--

batch\_normalization\_103[0][0]

---

---

activation_105	(Activation)	(None, 50, 50, 24)	0	
----------------	--------------	--------------------	---	--

batch\_normalization\_105[0][0]

---

---

activation_106	(Activation)	(None, 50, 50, 16)	0	
----------------	--------------	--------------------	---	--

batch\_normalization\_106[0][0]

---

---

concatenate_18	(Concatenate)	(None, 50, 50, 104)	0	
----------------	---------------	---------------------	---	--

activation\_101[0][0]

---

activation\_103[0][0]  
activation\_105[0][0]  
activation\_106[0][0]

---

---

conv2d_114	(Conv2D)	(None, 50, 50, 16)	1664	
------------	----------	--------------------	------	--

concatenate\_18[0][0]

---

---

flatten_17	(Flatten)	(None, 40000)	0	conv2d_114[0][0]
------------	-----------	---------------	---	------------------

---

---

dense_33	(Dense)	(None, 48)	1920048	flatten_17[0][0]
----------	---------	------------	---------	------------------

---

---

dense_34	(Dense)	(None, 1)	49	dense_33[0][0]
----------	---------	-----------	----	----------------

---

=====  
Total params: 1,939,009

Trainable params: 1,938,737

Non-trainable params: 272

---

## 定义 ImageDataGenerator

27

```
train_data_dir = r'C:\Users\coffe\Desktop\dogs-vs-cats\train'
validation_data_dir = r'C:\Users\coffe\Desktop\dogs-vs-cats\validation'
nb_train_samples = 10835
nb_validation_samples = 4000
epochs = 1
batch_size = 5

# this is the augmentation configuration we will use for training
train_datagen = ImageDataGenerator(
    rescale=1. / 255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)

# this is the augmentation configuration we will use for testing:
# only rescaling
test_datagen = ImageDataGenerator(rescale=1. / 255)

train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical')

validation_generator = test_datagen.flow_from_directory(
    validation_data_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical')

Found 10835 images belonging to 2 classes.
Found 4000 images belonging to 2 classes.
```

## 训练模型

```

model.fit_generator(
    train_generator,
    steps_per_epoch=nb_train_samples // batch_size,
    epochs=epochs,
    validation_data=validation_generator,
    validation_steps=nb_validation_samples // batch_size)

```

**ResourceExhaustedError**

Traceback (most recent call last)

```

C:\ProgramData\Anaconda3\lib\site-
packages\tensorflow\python\client\session.py in _do_call(self, fn, *args)

    1355     try:

-> 1356         return fn(*args)

    1357     except errors.OpError as e:

C:\ProgramData\Anaconda3\lib\site-
packages\tensorflow\python\client\session.py in _run_fn(feed_dict,
fetch_list, target_list, options, run_metadata)

    1340     return self._call_tf_sessionrun(

-> 1341         options, feed_dict, fetch_list, target_list, run_metadata)

    1342

C:\ProgramData\Anaconda3\lib\site-
packages\tensorflow\python\client\session.py in _call_tf_sessionrun(self,
options, feed_dict, fetch_list, target_list, run_metadata)

    1428     self._session, options, feed_dict, fetch_list, target_list,

-> 1429     run_metadata)

    1430

```

**ResourceExhaustedError:** OOM when allocating tensor with shape[2340000,48] and type float on /job:localhost/replica:0/task:0/device:CPU:0 by allocator cpu

```
[[{{node dense_29/random_uniform/RandomUniform}}]]
```

Hint: If you want to see a list of allocated tensors when OOM happens, add `report_tensor_allocations_upon_oom` to `RunOptions` for current allocation info.

During handling of the above exception, another exception occurred:

```
ResourceExhaustedError                                Traceback (most recent call last)

<ipython-input-35-2dedade68c7a> in <module>

      4     epochs=epochs,

      5     validation_data=validation_generator,

----> 6     validation_steps=nb_validation_samples // batch_size)

C:\ProgramData\Anaconda3\lib\site-packages\keras\legacy\interfaces.py in
wrapper(*args, **kwargs)

     89         warnings.warn('Update your ``' + object_name + ``' call to
the ' +

     90         'Keras 2 API: ' + signature, stacklevel=2)

---> 91         return func(*args, **kwargs)

     92     wrapper._original_function = func

     93     return wrapper

C:\ProgramData\Anaconda3\lib\site-packages\keras\engine\training.py in
fit_generator(self, generator, steps_per_epoch, epochs, verbose, callbacks,
validation_data, validation_steps, validation_freq, class_weight,
max_queue_size, workers, use_multiprocessing, shuffle, initial_epoch)

    1730         use_multiprocessing=use_multiprocessing,

    1731         shuffle=shuffle,

-> 1732         initial_epoch=initial_epoch)

    1733

    1734     @interfaces.legacy_generator_methods_support
```

```
C:\ProgramData\Anaconda3\lib\site-  
packages\keras\engine\training_generator.py in fit_generator(model,  
generator, steps_per_epoch, epochs, verbose, callbacks, validation_data,  
validation_steps, validation_freq, class_weight, max_queue_size, workers,  
use_multiprocessing, shuffle, initial_epoch)
```

```
40
```

```
41     do_validation = bool(validation_data)
```

```
--> 42     model._make_train_function()
```

```
43     if do_validation:
```

```
44         model._make_test_function()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\keras\engine\training.py in  
_make_train_function(self)
```

```
331             updates=updates + metrics_updates,
```

```
332             name='train_function',
```

```
--> 333             **self._function_kwargs)
```

```
334
```

```
335     def _make_test_function(self):
```

```
C:\ProgramData\Anaconda3\lib\site-  
packages\keras\backend\tensorflow_backend.py in function(inputs, outputs,  
updates, **kwargs)
```

```
3004 def function(inputs, outputs, updates=None, **kwargs):
```

```
3005     if _is_tf_1():
```

```
-> 3006         v1_variable_initialization()
```

```
3007     return tf_keras_backend.function(inputs, outputs,
```

```
3008         updates=updates,
```

```
C:\ProgramData\Anaconda3\lib\site-  
packages\keras\backend\tensorflow_backend.py in v1_variable_initialization()
```



```

418

419 def v1_variable_initialization():

--> 420     session = get_session()

421     with session.graph.as_default():

422         variables = tf.global_variables()

C:\ProgramData\Anaconda3\lib\site-
packages\keras\backend\tensorflow_backend.py in get_session()

383         '`get_session` is not available when '

384         'TensorFlow is executing eagerly.')

--> 385     return tf_keras_backend.get_session()

386

387

C:\ProgramData\Anaconda3\lib\site-
packages\tensorflow\python\keras\backend.py in get_session(op_input_list)

460     if not _MANUAL_VAR_INIT:

461         with session.graph.as_default():

--> 462             _initialize_variables(session)

463     return session

464

C:\ProgramData\Anaconda3\lib\site-
packages\tensorflow\python\keras\backend.py in
_initialize_variables(session)

884         v._keras_initialized = True

885         if uninitialized_vars:

--> 886

session.run(variables_module.variables_initializer(uninitialized_vars))

```

```

887

888

C:\ProgramData\Anaconda3\lib\site-
packages\tensorflow\python\client\session.py in run(self, fetches,
feed_dict, options, run_metadata)

948     try:

949         result = self._run(None, fetches, feed_dict, options_ptr,
--> 950                             run_metadata_ptr)

951     if run_metadata:

952         proto_data = tf_session.TF_GetBuffer(run_metadata_ptr)

C:\ProgramData\Anaconda3\lib\site-
packages\tensorflow\python\client\session.py in _run(self, handle, fetches,
feed_dict, options, run_metadata)

1171     if final_fetches or final_targets or (handle and
feed_dict_tensor):

1172         results = self._do_run(handle, final_targets, final_fetches,
-> 1173                             feed_dict_tensor, options, run_metadata)

1174     else:

1175         results = []

C:\ProgramData\Anaconda3\lib\site-
packages\tensorflow\python\client\session.py in _do_run(self, handle,
target_list, fetch_list, feed_dict, options, run_metadata)

1348     if handle is None:

1349         return self._do_call(_run_fn, feeds, fetches, targets, options,
-> 1350                             run_metadata)

1351     else:

1352         return self._do_call(_prun_fn, handle, feeds, fetches)

```

```

C:\ProgramData\Anaconda3\lib\site-
packages\tensorflow\python\client\session.py in _do_call(self, fn, *args)

1368         pass

1369         message = error_interpolation.interpolate(message, self._graph)

-> 1370         raise type(e)(node_def, op, message)

1371

1372     def _extend_graph(self):

```

**ResourceExhaustedError:** OOM when allocating tensor with shape[2340000,48] and type float on /job:localhost/replica:0/task:0/device:CPU:0 by allocator cpu

```

[[node dense_29/random_uniform/RandomUniform (defined at
C:\ProgramData\Anaconda3\lib\site-
packages\keras\backend\tensorflow_backend.py:4357) ]]

```

Hint: If you want to see a list of allocated tensors when OOM happens, add report\_tensor\_allocations\_upon\_oom to RunOptions for current allocation info.

Original stack trace for 'dense\_29/random\_uniform/RandomUniform':

```

File "C:\ProgramData\Anaconda3\lib\runpy.py", line 193, in
_run_module_as_main

```

```

    "__main__", mod_spec)

```

```

File "C:\ProgramData\Anaconda3\lib\runpy.py", line 85, in _run_code

```

```

    exec(code, run_globals)

```

```

File "C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py",
line 16, in <module>

```

```

    app.launch_new_instance()

```

File "C:\ProgramData\Anaconda3\lib\site-packages\traitlets\config\application.py", line 658, in launch\_instance

```
app.start()
```

File "C:\ProgramData\Anaconda3\lib\site-packages\ipykernel\kernelapp.py", line 505, in start

```
self.io_loop.start()
```

File "C:\ProgramData\Anaconda3\lib\site-packages\tornado\platform\asyncio.py", line 148, in start

```
self.asyncio_loop.run_forever()
```

File "C:\ProgramData\Anaconda3\lib\asyncio\base\_events.py", line 438, in run\_forever

```
self._run_once()
```

File "C:\ProgramData\Anaconda3\lib\asyncio\base\_events.py", line 1451, in \_run\_once

```
handle._run()
```

File "C:\ProgramData\Anaconda3\lib\asyncio\events.py", line 145, in \_run

```
self._callback(*self._args)
```

File "C:\ProgramData\Anaconda3\lib\site-packages\tornado\ioloop.py", line 690, in <lambda>

```
lambda f: self._run_callback(functools.partial(callback, future))
```

File "C:\ProgramData\Anaconda3\lib\site-packages\tornado\ioloop.py", line 743, in \_run\_callback

```
ret = callback()
```

File "C:\ProgramData\Anaconda3\lib\site-packages\tornado\gen.py", line 781, in inner

```
self.run()
```

File "C:\ProgramData\Anaconda3\lib\site-packages\tornado\gen.py", line 742, in run

```
yielded = self.gen.send(value)
```

File "C:\ProgramData\Anaconda3\lib\site-packages\ipykernel\kernelbase.py",  
line 357, in process\_one

```
yield gen.maybe_future(dispatch(*args))
```

File "C:\ProgramData\Anaconda3\lib\site-packages\tornado\gen.py", line 209,  
in wrapper

```
yielded = next(result)
```

File "C:\ProgramData\Anaconda3\lib\site-packages\ipykernel\kernelbase.py",  
line 267, in dispatch\_shell

```
yield gen.maybe_future(handler(stream, idents, msg))
```

File "C:\ProgramData\Anaconda3\lib\site-packages\tornado\gen.py", line 209,  
in wrapper

```
yielded = next(result)
```

File "C:\ProgramData\Anaconda3\lib\site-packages\ipykernel\kernelbase.py",  
line 534, in execute\_request

```
user_expressions, allow_stdin,
```

File "C:\ProgramData\Anaconda3\lib\site-packages\tornado\gen.py", line 209,  
in wrapper

```
yielded = next(result)
```

File "C:\ProgramData\Anaconda3\lib\site-packages\ipykernel\ipkernel.py",  
line 294, in do\_execute

```
res = shell.run_cell(code, store_history=store_history, silent=silent)
```

File "C:\ProgramData\Anaconda3\lib\site-packages\ipykernel\zmqshell.py",  
line 536, in run\_cell

```
return super(ZMQInteractiveShell, self).run_cell(*args, **kwargs)
```

File "C:\ProgramData\Anaconda3\lib\site-  
packages\IPython\core\interactiveshell.py", line 2848, in run\_cell

```
raw_cell, store_history, silent, shell_futures)
```

File "C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py", line 2874, in \_run\_cell

```
return runner(coro)
```

File "C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\async\_helpers.py", line 67, in \_pseudo\_sync\_runner

```
coro.send(None)
```

File "C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py", line 3049, in run\_cell\_async

```
interactivity=interactivity, compiler=compiler, result=result)
```

File "C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py", line 3214, in run\_ast\_nodes

```
if (yield from self.run_code(code, result)):
```

File "C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py", line 3296, in run\_code

```
exec(code_obj, self.user_global_ns, self.user_ns)
```

File "<ipython-input-29-59f85435779c>", line 66, in <module>

```
out = Dense(48, activation='relu')(out)
```

File "C:\ProgramData\Anaconda3\lib\site-packages\keras\engine\base\_layer.py", line 463, in \_\_call\_\_

```
self.build(unpack_singleton(input_shapes))
```

File "C:\ProgramData\Anaconda3\lib\site-packages\keras\layers\core.py", line 895, in build

```
constraint=self.kernel_constraint)
```

File "C:\ProgramData\Anaconda3\lib\site-packages\keras\engine\base\_layer.py", line 279, in add\_weight

```
weight = K.variable(initializer(shape, dtype=dtype),
```

File "C:\ProgramData\Anaconda3\lib\site-packages\keras\initializers.py", line 227, in \_\_call\_\_

```
dtype=dtype, seed=self.seed)
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py", line 4357, in random_uniform
```

```
shape, minval=minval, maxval=maxval, dtype=dtype, seed=seed)
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\keras\backend.py", line 5253, in random_uniform
```

```
shape, minval=minval, maxval=maxval, dtype=dtype, seed=seed)
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\ops\random_ops.py", line 247, in random_uniform
```

```
rnd = gen_random_ops.random_uniform(shape, dtype, seed=seed1, seed2=seed2)
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\ops\gen_random_ops.py", line 859, in random_uniform
```

```
name=name)
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\framework\op_def_library.py", line 788, in _apply_op_helper
```

```
op_def=op_def)
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\util\deprecation.py", line 507, in new_func
```

```
return func(*args, **kwargs)
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\framework\ops.py", line 3616, in create_op
```

```
op_def=op_def)
```

```
File "C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\python\framework\ops.py", line 2005, in __init__
```

```
self._traceback = tf_stack.extract_stack()
```

## 使用训练后模型预测图像

48

```
import cv2
img = cv2.resize(cv2.imread(r'C:\Users\coffe\Desktop\dogs-vs-
cats\test\7.jpg'), (img_width, img_height)).astype(np.float32)
# img[:, :, 0] -= 103.939
# img[:, :, 1] -= 116.779
# img[:, :, 2] -= 123.68
#img = img.transpose((2,0,1))
x = img_to_array(img)

x = np.expand_dims(x, axis=0)

#x = preprocess_input(x)

score = model.predict(x)

print(score)
[[0.99895906]]
```