



人工智能 软件开发汇报

汇报人：黄家名

目录



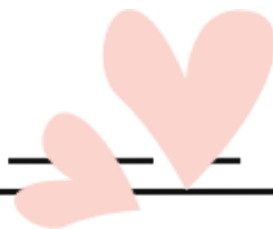
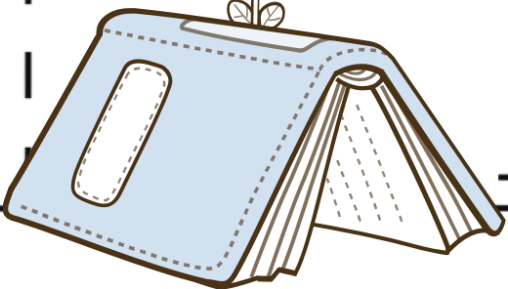
设计背景

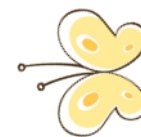
代码原理



演示过程

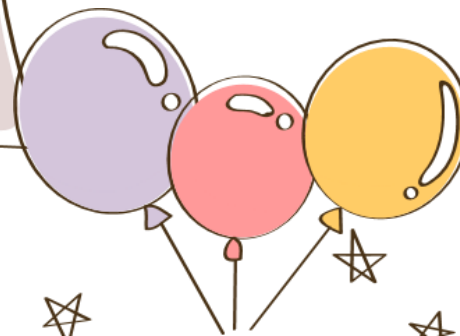
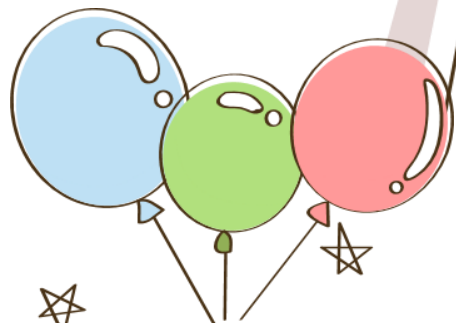
心得体会





01

设计背景



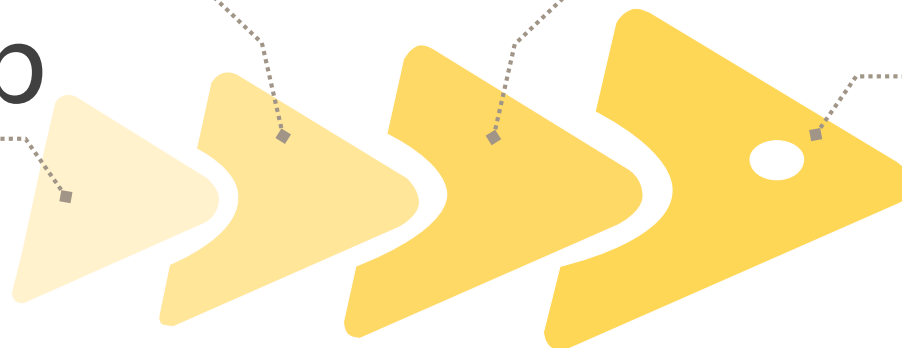
设计背景

JavaScript

CSS

HTML

Javaweb





设计背景



机器学习



线性回归 - Linear regression



逻辑回归 - Logistic regression



神经网络 - neural network



支持向量机 - SVM





设计背景

编译器:

sublime

jupyter notebook

编程语言:

HTML CSS JS

Python

浏览器:

Google Chrome

环境支持:

Python3.6 、 CUDA v10.0.130

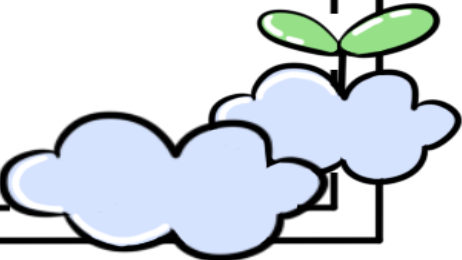

cuda v7.6.5.32 、 keras

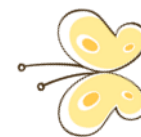
Tensorflow-gpu v1.13.1



设计背景

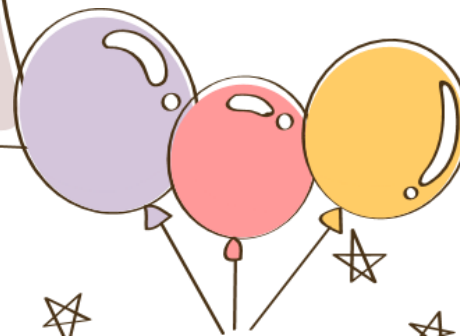
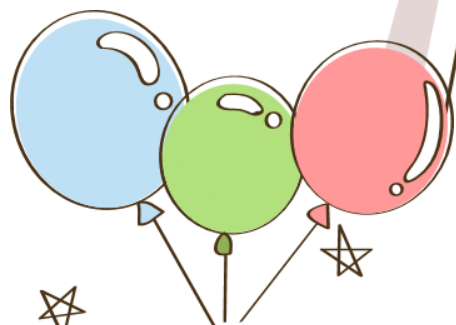
设计一个基于flask框架的pythonweb,
通过keras框架训练深度学习算法模型,
实现猫狗大作战的小软件系统。






02

代码原理





```
In [15]: fnames = ['cat. {}.jpg'.format(i) for i in range(10000)] #0-9999张猫图片作为训练集
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(train_cats_dir, fname)
    shutil.copyfile(src, dst)
```

```
In [20]: fnames = ['cat. {}.jpg'.format(i) for i in range(10000, 12500)] #10000-12499猫的图片为验证集
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(validation_cats_dir, fname)
    shutil.copyfile(src, dst)
```

```
In [18]: fnames = ['dog. {}.jpg'.format(i) for i in range(10000)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(train_dogs_dir, fname)
    shutil.copyfile(src, dst)
```

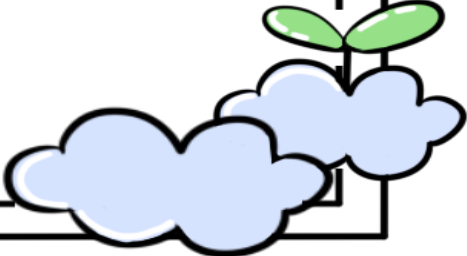

```
In [22]: fnames = ['dog. {}.jpg'.format(i) for i in range(10000, 12500)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(validation_dogs_dir, fname)
    shutil.copyfile(src, dst)
```





代码展示




```
In [5]: model = models.Sequential()
```

```
In [6]: model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 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(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))
model.summary()
```

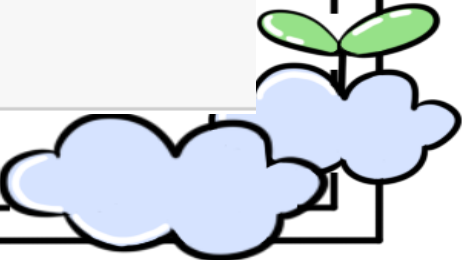




```
In [10]: train_datagen = ImageDataGenerator(rescale=1./255,
                                             shear_range=0.2,
                                             zoom_range=0.2,
                                             horizontal_flip=True)    #所有图像乘1/255, 图像增强旋转, 缩放
test_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
```



```
In [11]: train_generator = train_datagen.flow_from_directory(
        train_dir,
        target_size=(150,150),    #目标图片大小
        batch_size=32,             #每次放进模型里的图片的数量
        class_mode='binary'       #使用binary_crossentropy, 需用二进制标签
    )
validation_generator = test_datagen.flow_from_directory(
    validation_dir,
    target_size=(150,150),
    batch_size=32,
    class_mode = 'binary'
)
```





代码展示

```
In [12]: from keras.callbacks import TensorBoard, ModelCheckpoint, ReduceLROnPlateau, EarlyStopping, ReduceLROnPlateau
log_dir = "logs/001/" #日志和模型文件存放文件夹
checkpoint_path = log_dir + 'ep{epoch:03d}-loss{loss:.3f}-val_loss{val_loss:.3f}.h5' #模型文件存放地址
```

```
In [13]: tensorboard = TensorBoard(log_dir=log_dir)
checkpoint = ModelCheckpoint(checkpoint_path, monitor="val_loss", save_best_only=True, period=3)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=5, verbose=1)
```

```
In [15]: #利用批量生成器拟合模型
history = model.fit_generator(
    train_generator,
    steps_per_epoch=625, #整数, 当生成器返回steps_per_epoch次数据时计一个epoch结束
    epochs=50, #整数, 数据迭代的轮数
    validation_data=validation_generator,
    validation_steps=20, #当validation_data为生成器时, 本参数指定验证集的生成器返回次数
    callbacks = [tensorboard, checkpoint, reduce_lr]
)
```





jupyter Cats_vs_dogs Last Checkpoint: 上星期日22:20 (autosaved)



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3



Code



```
train_generator,
steps_per_epoch=625, #整数, 当生成器返回steps_per_epoch次数据时计一个epoch结束
epochs=50, #整数, 数据迭代的轮数
validation_data=validation_generator,
validation_steps=20, #当validation_data为生成器时, 本参数指定验证集的生成器返回次数
callbacks = [tensorboard, checkpoint, reduce_lr]
)
```

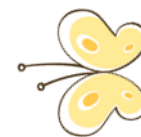
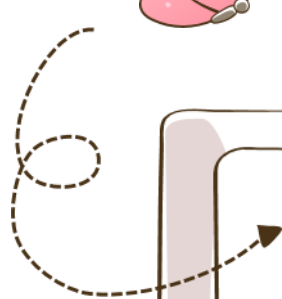
```
625/625 [=====] - 704s 1s/step - loss: 0.1121 - acc: 0.9575 - val_loss: 0.3231 - val_acc: 0.8859
Epoch 37/50
625/625 [=====] - 705s 1s/step - loss: 0.1119 - acc: 0.9571 - val_loss: 0.1926 - val_acc: 0.9203
Epoch 38/50
625/625 [=====] - 704s 1s/step - loss: 0.1118 - acc: 0.9580 - val_loss: 0.2299 - val_acc: 0.9156
Epoch 39/50
625/625 [=====] - 707s 1s/step - loss: 0.1100 - acc: 0.9586 - val_loss: 0.1837 - val_acc: 0.9297
Epoch 40/50
625/625 [=====] - 728s 1s/step - loss: 0.1104 - acc: 0.9583 - val_loss: 0.2393 - val_acc: 0.9107
Epoch 41/50
625/625 [=====] - 713s 1s/step - loss: 0.1082 - acc: 0.9582 - val_loss: 0.2744 - val_acc: 0.9016
Epoch 42/50
625/625 [=====] - 718s 1s/step - loss: 0.1121 - acc: 0.9582 - val_loss: 0.2211 - val_acc: 0.9328
Epoch 43/50
625/625 [=====] - 728s 1s/step - loss: 0.1113 - acc: 0.9576 - val_loss: 0.2144 - val_acc: 0.9094
Epoch 44/50
625/625 [=====] - 722s 1s/step - loss: 0.1109 - acc: 0.9584 - val_loss: 0.1784 - val_acc: 0.9203
Epoch 45/50
625/625 [=====] - 725s 1s/step - loss: 0.1084 - acc: 0.9590 - val_loss: 0.2540 - val_acc: 0.9141
Epoch 46/50
```



```
23 with sess_class.as_default():
24     with graph_class.as_default():
25         model = load_model(modelDir+"/"+modelName)
26
27 def predict_class(imgPath):
28     global graph_class,model
29     with sess_class.as_default():
30         with graph_class.as_default():
31             img = image.load_img(imgPath, target_size=(150, 150))
32             x = image.img_to_array(img)
33             x = np.expand_dims(x, axis=0)
34             y = model.predict_classes(x)
35             if(int(np.squeeze(y))==0):
36                 return "该图片是小猫！"
37             else:
38                 return "该图片是小狗!"
39
```

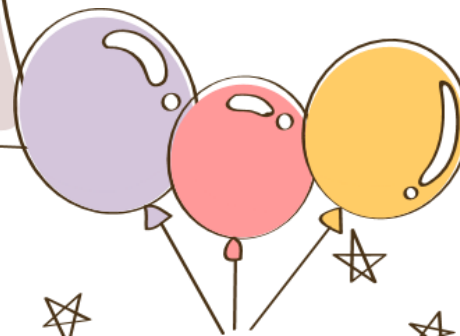
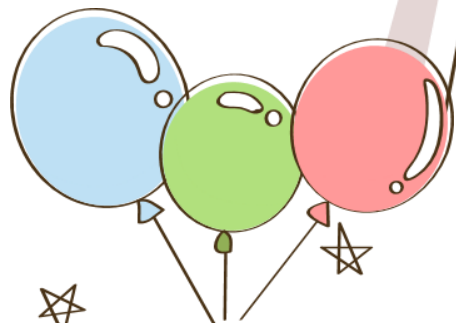


```
41
42 app = Flask(__name__)
43
44 @app.route('/')
45 def index():
46     return render_template("index.html")
47
48 @app.route('/posting', methods = ['POST', 'GET'])
49 def getimg():
50     if request.method == 'POST':
51         img = request.form['imgMsg']
52         data = json.loads(img)
53         for img_data in data:
54             img_base64 = str(img_data['base64']);
55             img_base64 = img_base64.replace("data:image/jpeg;base64,", "");
56             fh = open("pic.jpg", "wb")
57             fh.write(base64.b64decode(img_base64))
58             fh.close();
59             result = predict_class(imgPath)
60             return '%s' % result
61     else:
62         img = request.args.get('imgMsg')
63         return 'success! %s' % img
64
65
66 if __name__ == '__main__':
67     app.run(debug = True)
```

03

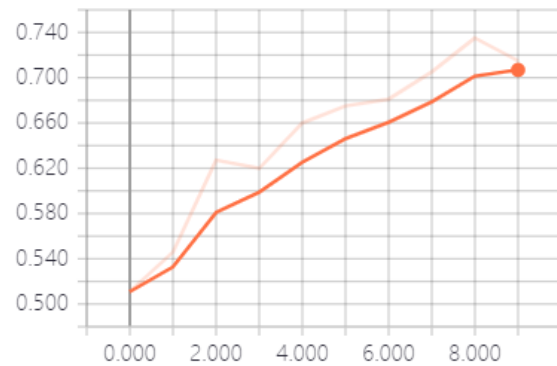
演示过程



epoch_acc

1

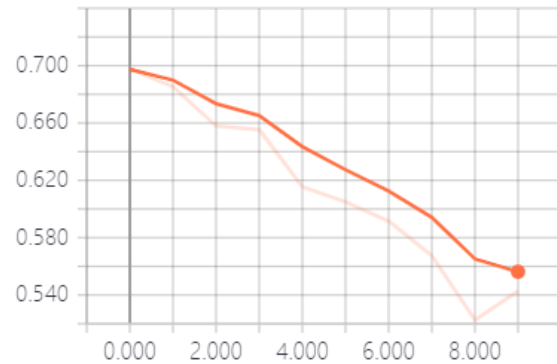
epoch_acc



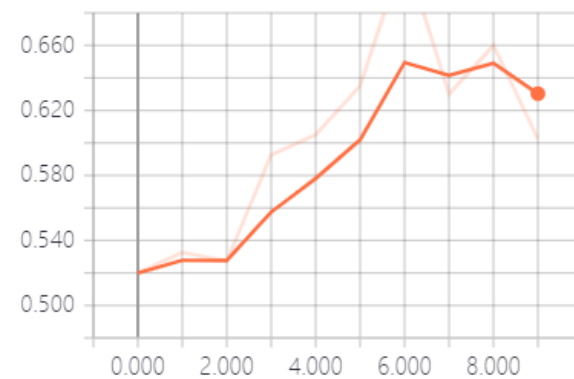
epoch_loss

1

epoch_loss



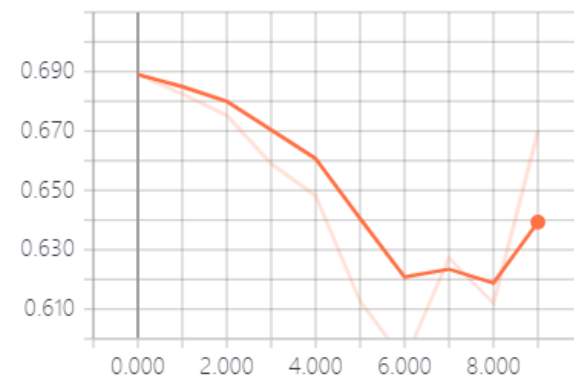
epoch_val_acc

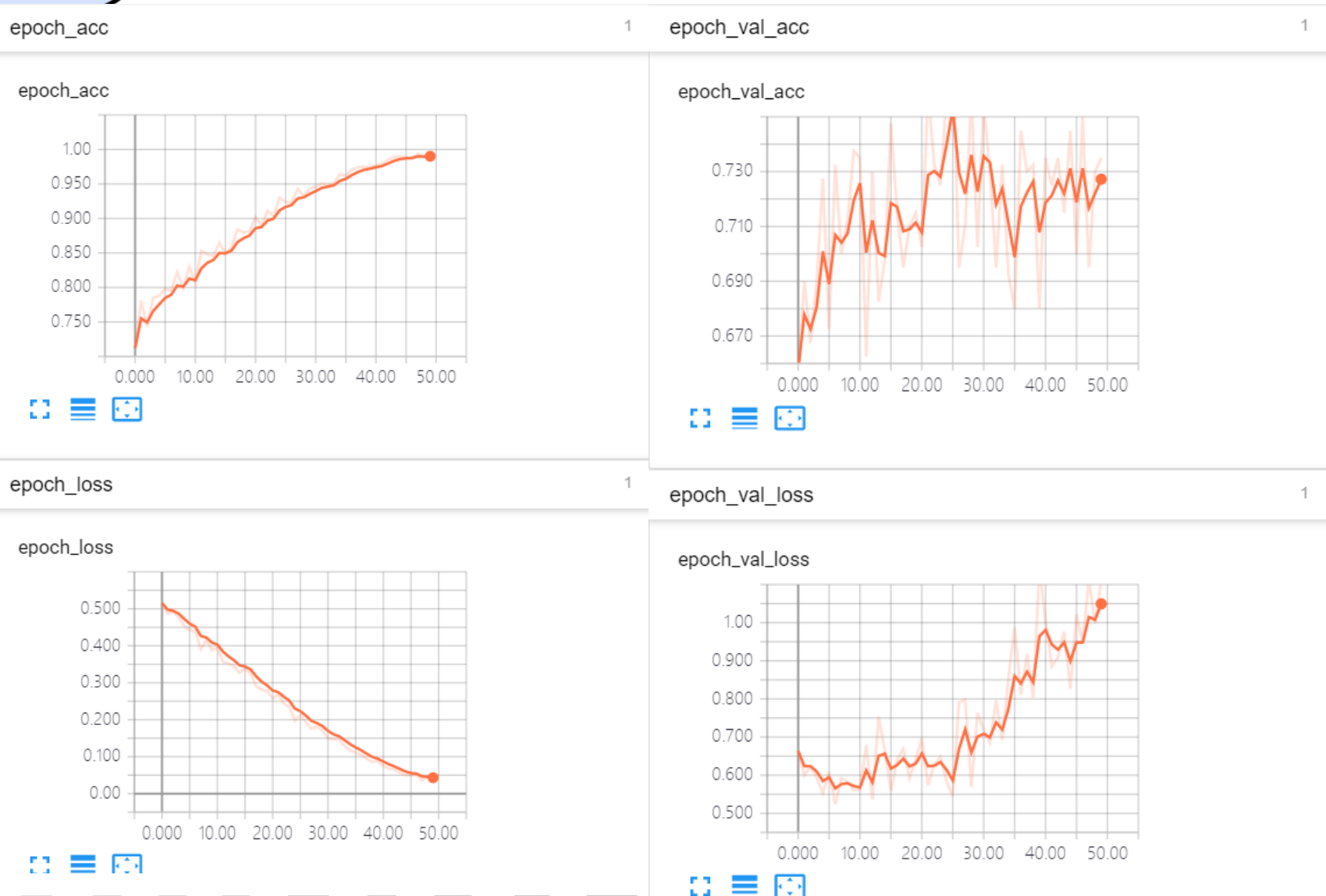


epoch_val_loss

1

epoch_val_loss

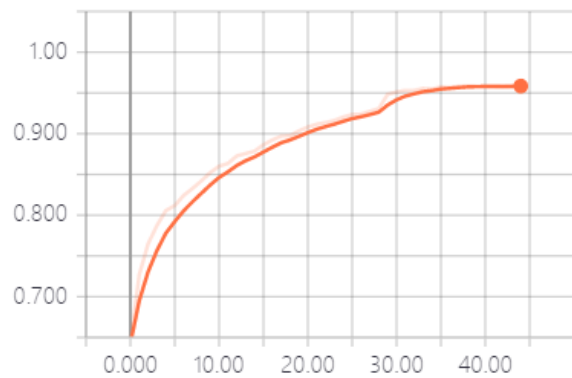




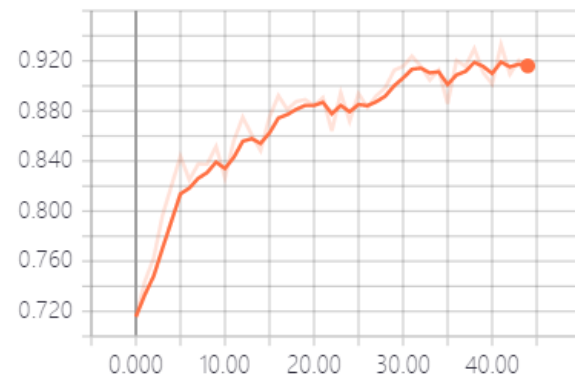
acc

1

acc



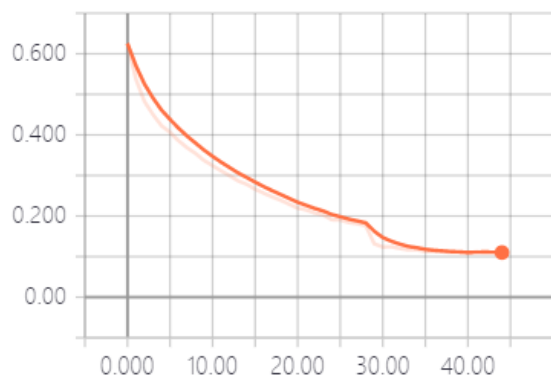
val_acc



loss

1

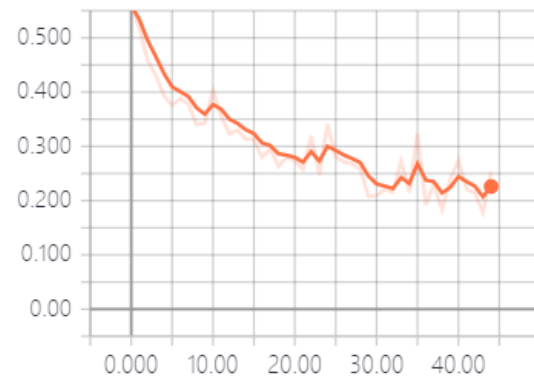
loss

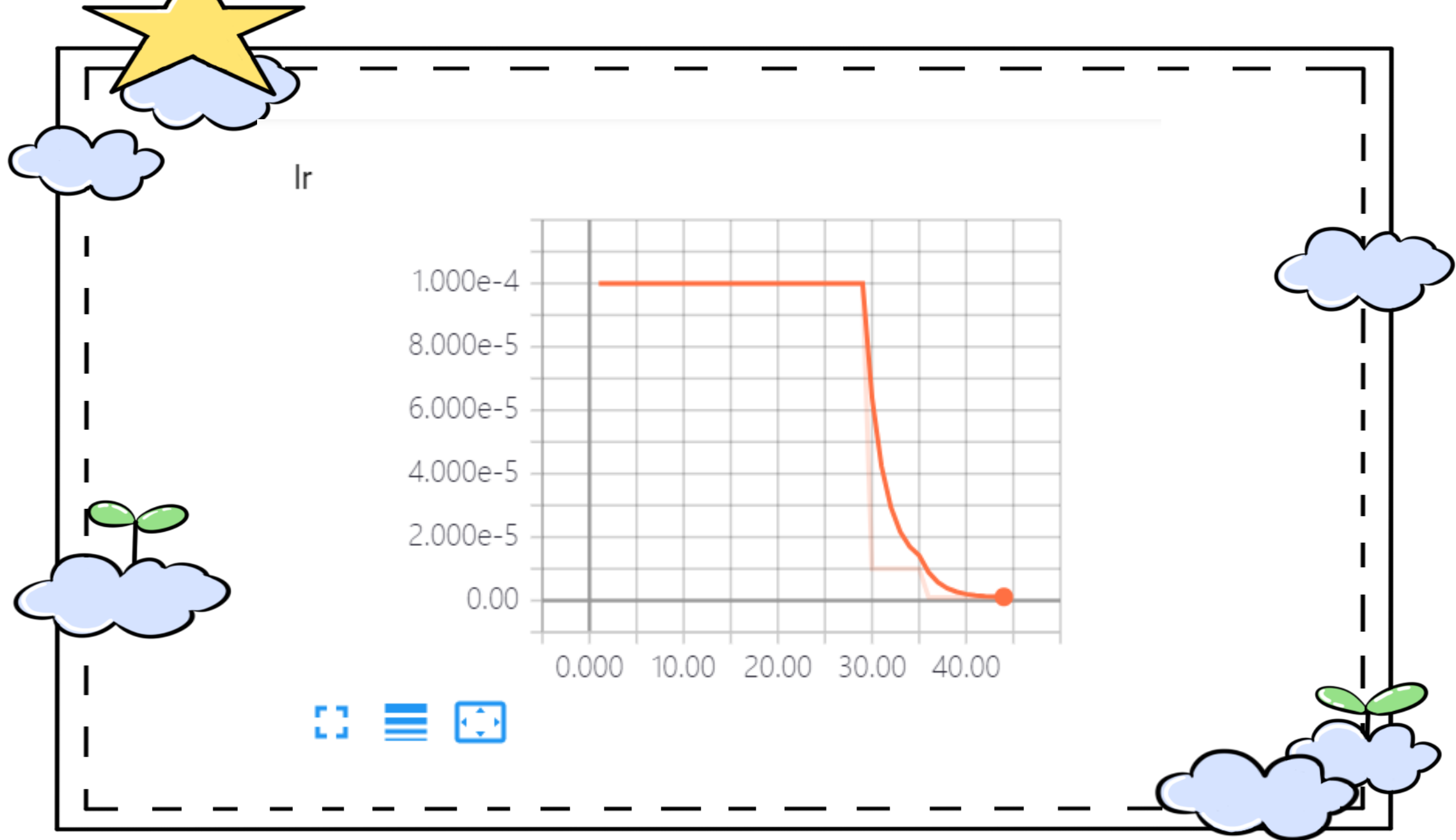


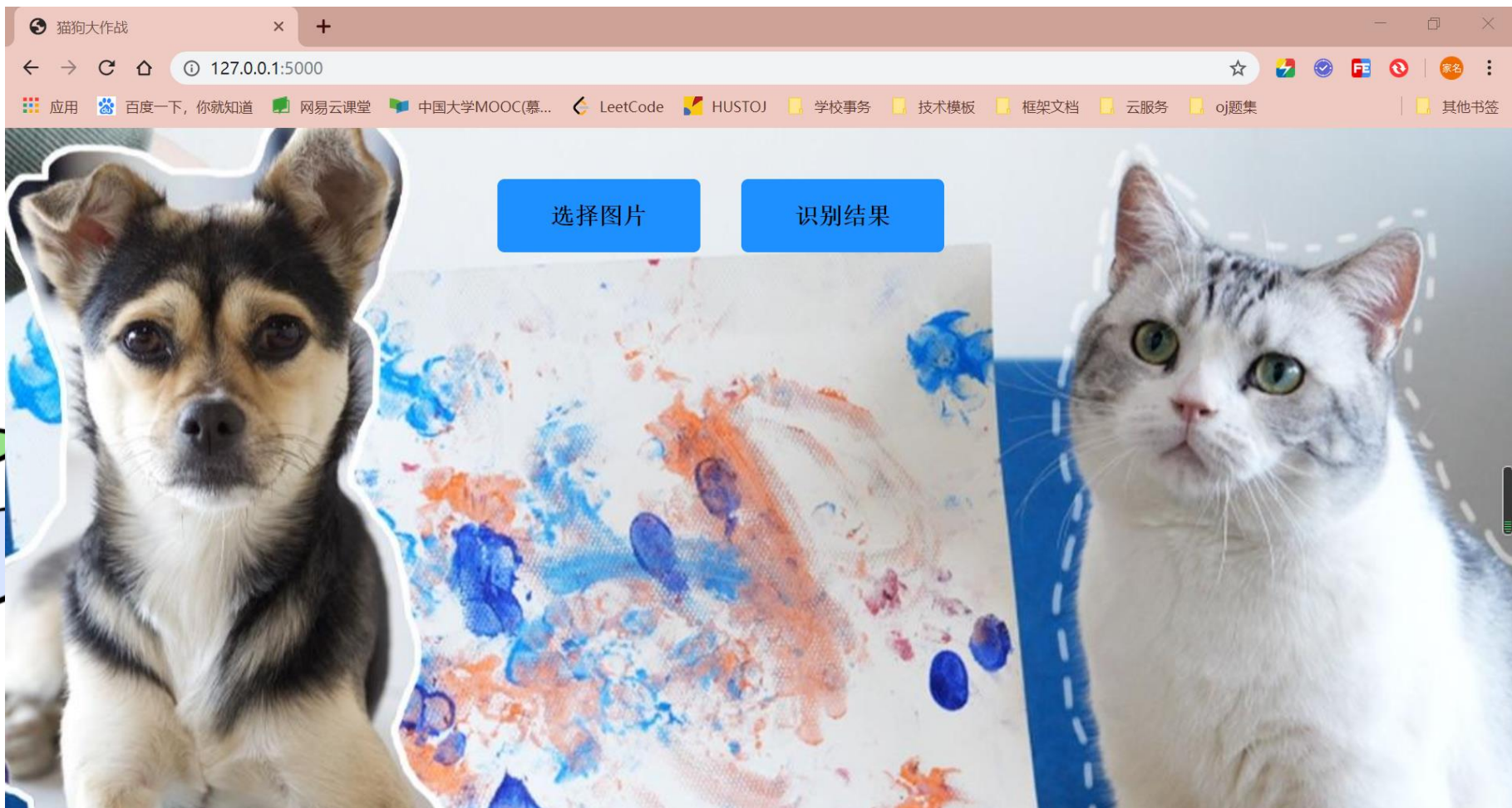
val_loss

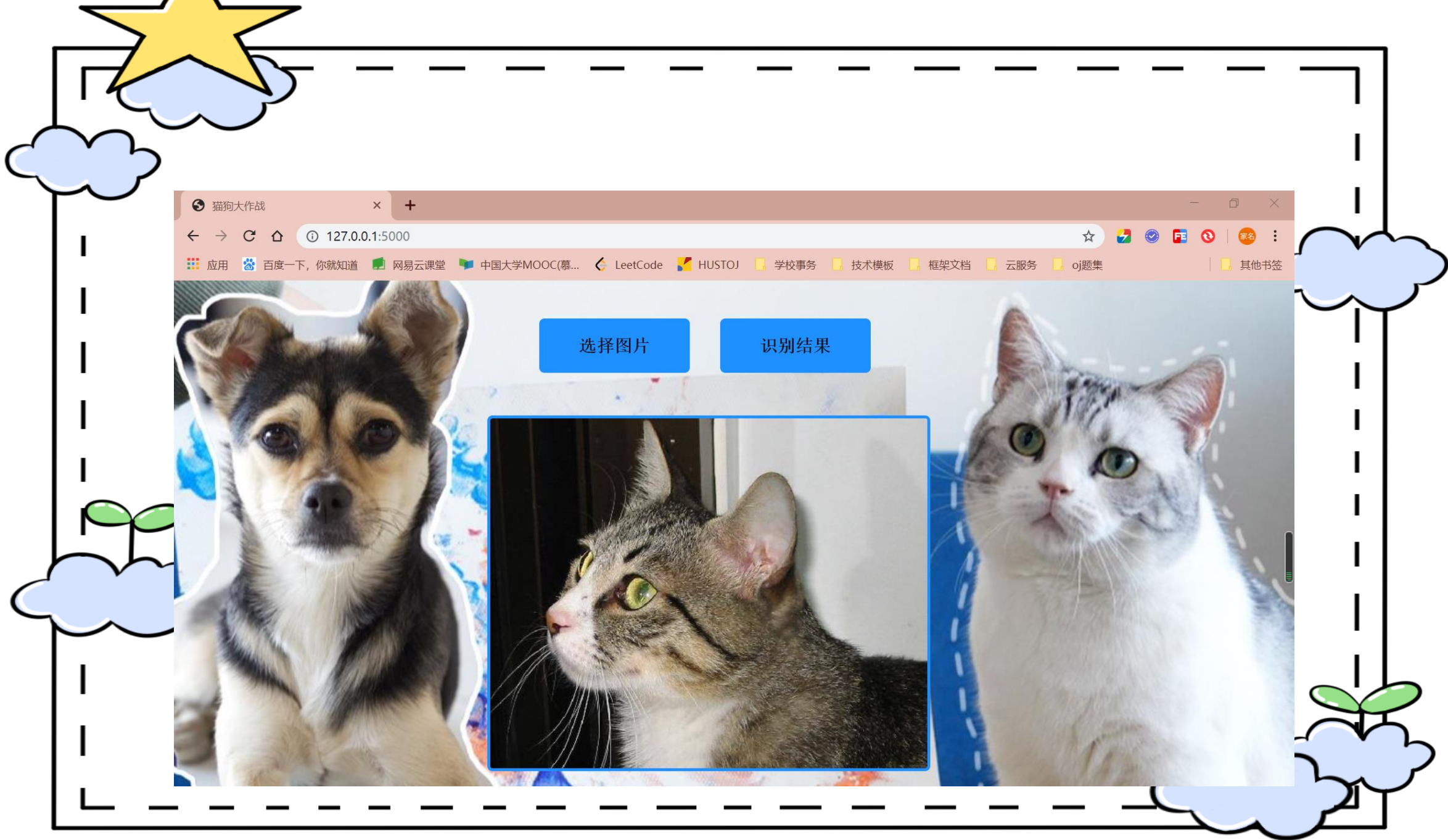
1

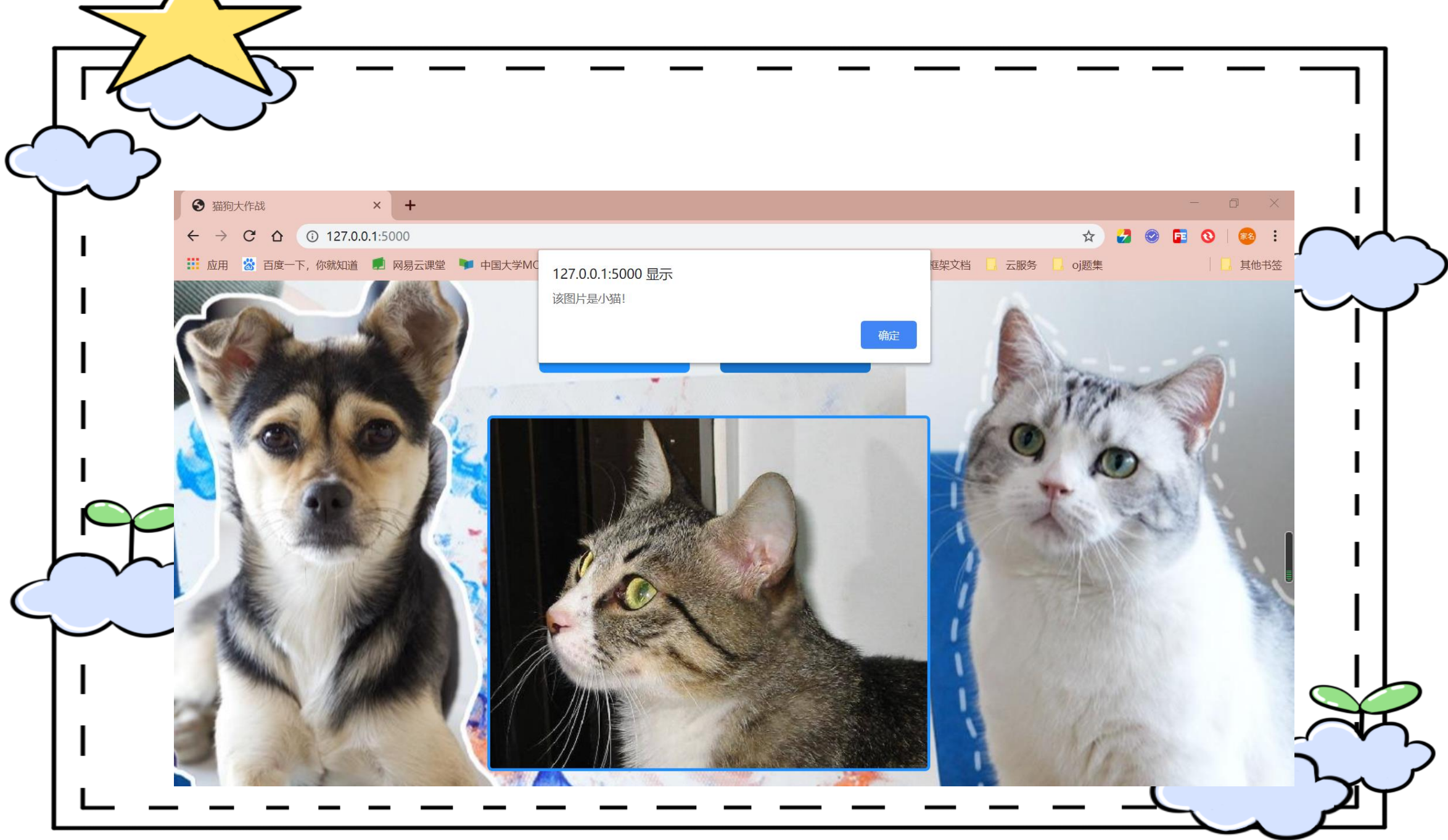
val_loss

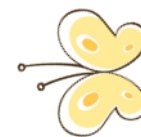






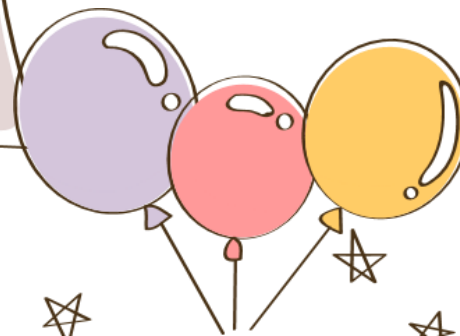
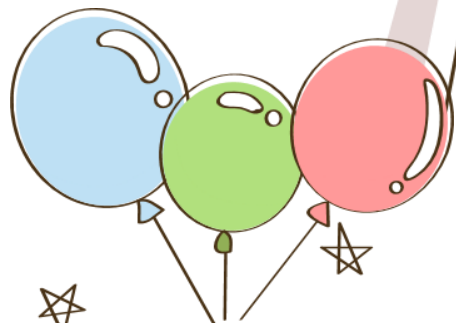







04

心得体会








心得体会



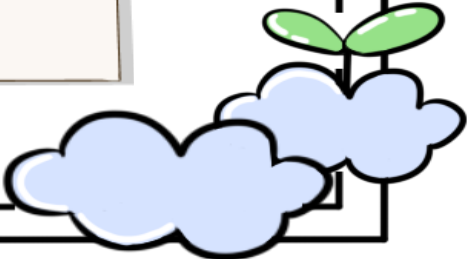
环境的配置很关键。
因为安装的是
tensorflow-gpu, 所以
花了很长时间安装
cuda和cudnn, 然后
tensorflow和cudnn
的版本还要对应。



合理设计神经网络
结构和训练轮
次与每次训练的
样本数。不然会
导致模型过拟合,
准确率不高。



学会对数据进行预
处理。可以使数据
增强, 数据集增多。
实现所有像素值都
归一化, 方便训练。



谢谢观看

