耳朵目标检测-可视化训练日志

训练模型时在 work_dirs 目录生成记录训练日志，解析其中损失函数、评估指标等信息，并可视化。

同济子豪兄：https://space.bilibili.com/1900783 (https://space.bilibili.com/1900783)

进入mmdetection主目录

In [1]:

```
import os
os.chdir('mmdetection')
```

导入工具包

In [2]:

```
import pandas as pd
from tqdm import tqdm

import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams['axes.unicode_minus']=False  # 用来正常显示负号
```

载入训练日志

In [3]:

```
# 日志文件路径
log_path = 'work_dirs/rtmdet_tiny_ear/20230604_143157/vis_data/scalars.json'
```

In [4]:

```
with open(log_path, "r") as f:
    json_list = f.readlines()
```

In [5]:

```
len(json_list)
```

Out[5]:

4257

In [6]:

```
eval(json_list[4])
```

Out[6]:

```
{'lr': 1.6055855855855853e-05,
 'data_time': 1.3774092674255372,
 'loss': 0.07204628437757492,
 'loss_cls': 0.06376696228981019,
 'loss_bbox': 0.008279320900328457,
 'time': 3.043340635299683,
 'epoch': 1,
 'memory': 2587,
 'step': 5}
```

In [7]:

```
df_train = pd.DataFrame()
df_test = pd.DataFrame()
for each in tqdm(json_list):
    if 'coco/bbox_mAP' in each:
        df_test = df_test.append(eval(each), ignore_index=True)
    else:
        df_train = df_train.append(eval(each), ignore_index=True)
```

```
  0%|
| 0/4257 [00:00<?, ?it/s]C:\Users\leaf8\AppData\Local\Temp\ipykernel_14172\22
27712475.py:7: FutureWarning: The frame.append method is deprecated and will
be removed from pandas in a future version. Use pandas.concat instead.
  df_train = df_train.append(eval(each), ignore_index=True)
C:\Users\leaf8\AppData\Local\Temp\ipykernel_14172\2227712475.py:7: FutureWarn
ing: The frame.append method is deprecated and will be removed from pandas in
a future version. Use pandas.concat instead.
  df_train = df_train.append(eval(each), ignore_index=True)
C:\Users\leaf8\AppData\Local\Temp\ipykernel_14172\2227712475.py:7: FutureWarn
ing: The frame.append method is deprecated and will be removed from pandas in
a future version. Use pandas.concat instead.
  df_train = df_train.append(eval(each), ignore_index=True)
C:\Users\leaf8\AppData\Local\Temp\ipykernel_14172\2227712475.py:7: FutureWarn
ing: The frame.append method is deprecated and will be removed from pandas in
a future version. Use pandas.concat instead.
  df_train = df_train.append(eval(each), ignore_index=True)
C:\Users\leaf8\AppData\Local\Temp\ipykernel_14172\2227712475.py:7: FutureWarn
ing: The frame.append method is deprecated and will be removed from pandas in
```

In [8]:

```
df_train
```

Out[8]:

	lr	data_time	loss	loss_cls	loss_bbox	time	epoch	memory	s
0	4.000000e-08	6.840638	0.065329	0.063868	0.001461	13.968867	1.0	2529.0	
1	4.043964e-06	3.420319	0.077665	0.063875	0.013790	7.121280	1.0	2587.0	
2	8.047928e-06	2.290473	0.073892	0.063843	0.010049	4.857160	1.0	2587.0	
3	1.205189e-05	1.717855	0.072786	0.063788	0.008998	3.721498	1.0	2587.0	
4	1.605586e-05	1.377409	0.072046	0.063767	0.008279	3.043341	1.0	2587.0	
...	...	...	...	...	...	...	...	...	
4195	3.058193e-03	0.064137	0.306200	0.087749	0.218451	0.350345	200.0	2587.0	419
4196	3.056557e-03	0.064337	0.304542	0.086985	0.217557	0.349759	200.0	2587.0	419
4197	3.054919e-03	0.064408	0.305163	0.087190	0.217974	0.349372	200.0	2587.0	419
4198	3.053280e-03	0.064637	0.304115	0.086657	0.217458	0.349467	200.0	2587.0	419
4199	3.051641e-03	0.064210	0.301987	0.084558	0.217429	0.348125	200.0	2587.0	420

4200 rows × 9 columns

In [9]:

```
df_test
```

Out[9]:

	coco/bbox_mAP	coco/bbox_mAP_50	coco/bbox_mAP_75	coco/bbox_mAP_s	coco/bbox_
0	0.050	0.150	0.018	-1.0	
1	0.215	0.698	0.029	-1.0	
2	0.293	0.830	0.102	-1.0	
3	0.313	0.902	0.174	-1.0	
4	0.516	0.960	0.434	-1.0	
5	0.555	0.944	0.596	-1.0	
6	0.612	0.966	0.686	-1.0	
7	0.668	0.966	0.929	-1.0	
8	0.653	0.969	0.885	-1.0	
9	0.654	0.970	0.811	-1.0	
10	0.532	0.964	0.557	-1.0	
11	0.731	0.961	0.935	-1.0	
12	0.696	0.970	0.970	-1.0	
13	0.533	0.960	0.388	-1.0	
14	0.740	0.969	0.901	-1.0	
15	0.725	0.969	0.946	-1.0	
16	0.736	0.967	0.946	-1.0	
17	0.660	0.970	0.911	-1.0	
18	0.797	0.970	0.970	-1.0	
19	0.720	0.970	0.949	-1.0	
20	0.726	0.967	0.914	-1.0	
21	0.814	0.970	0.970	-1.0	
22	0.803	0.970	0.970	-1.0	
23	0.768	0.970	0.970	-1.0	
24	0.728	0.964	0.964	-1.0	
25	0.786	0.968	0.968	-1.0	
26	0.688	0.970	0.948	-1.0	
27	0.671	0.970	0.970	-1.0	
28	0.771	0.964	0.964	-1.0	
29	0.732	0.948	0.921	-1.0	
30	0.759	0.970	0.970	-1.0	
31	0.778	0.970	0.970	-1.0	
32	0.777	0.951	0.950	-1.0	
33	0.791	0.970	0.949	-1.0	
34	0.774	0.970	0.947	-1.0	
35	0.797	0.963	0.963	-1.0	
36	0.810	0.970	0.950	-1.0	

	coco/bbox_mAP	coco/bbox_mAP_50	coco/bbox_mAP_75	coco/bbox_mAP_s	coco/bbox_
37	0.757	0.970	0.949	-1.0	
38	0.755	0.970	0.970	-1.0	
39	0.724	0.959	0.959	-1.0	
40	0.754	0.955	0.930	-1.0	
41	0.670	0.957	0.804	-1.0	
42	0.771	0.967	0.945	-1.0	
43	0.790	0.969	0.969	-1.0	
44	0.748	0.970	0.970	-1.0	
45	0.795	0.968	0.948	-1.0	
46	0.799	0.967	0.967	-1.0	
47	0.789	0.967	0.967	-1.0	
48	0.760	0.968	0.968	-1.0	
49	0.794	0.970	0.970	-1.0	
50	0.760	0.968	0.968	-1.0	
51	0.763	0.951	0.950	-1.0	
52	0.766	0.952	0.918	-1.0	
53	0.789	0.964	0.946	-1.0	
54	0.794	0.965	0.965	-1.0	
55	0.803	0.966	0.966	-1.0	
56	0.763	0.967	0.967	-1.0	

导出训练日志表格

In [10]:

```
df_train.to_csv('训练日志-训练集.csv', index=False)
df_test.to_csv('训练日志-测试集.csv', index=False)
```

设置Matplotlib中文字体

In [11]:

```
# # windows操作系统
# plt.rcParams['font.sans-serif']=['SimHei']  # 用来正常显示中文标签
# plt.rcParams['axes.unicode_minus']=False  # 用来正常显示负号
```

In [12]:

```
# Mac操作系统，参考 https://www.ngui.cc/51cto/show-727683.html
# 下载 simhei.ttf 字体文件
# !wget https://zihao-openmmlab.obs.cn-east-3.myhuaweicloud.com/20220716-mmclassification/datase
```

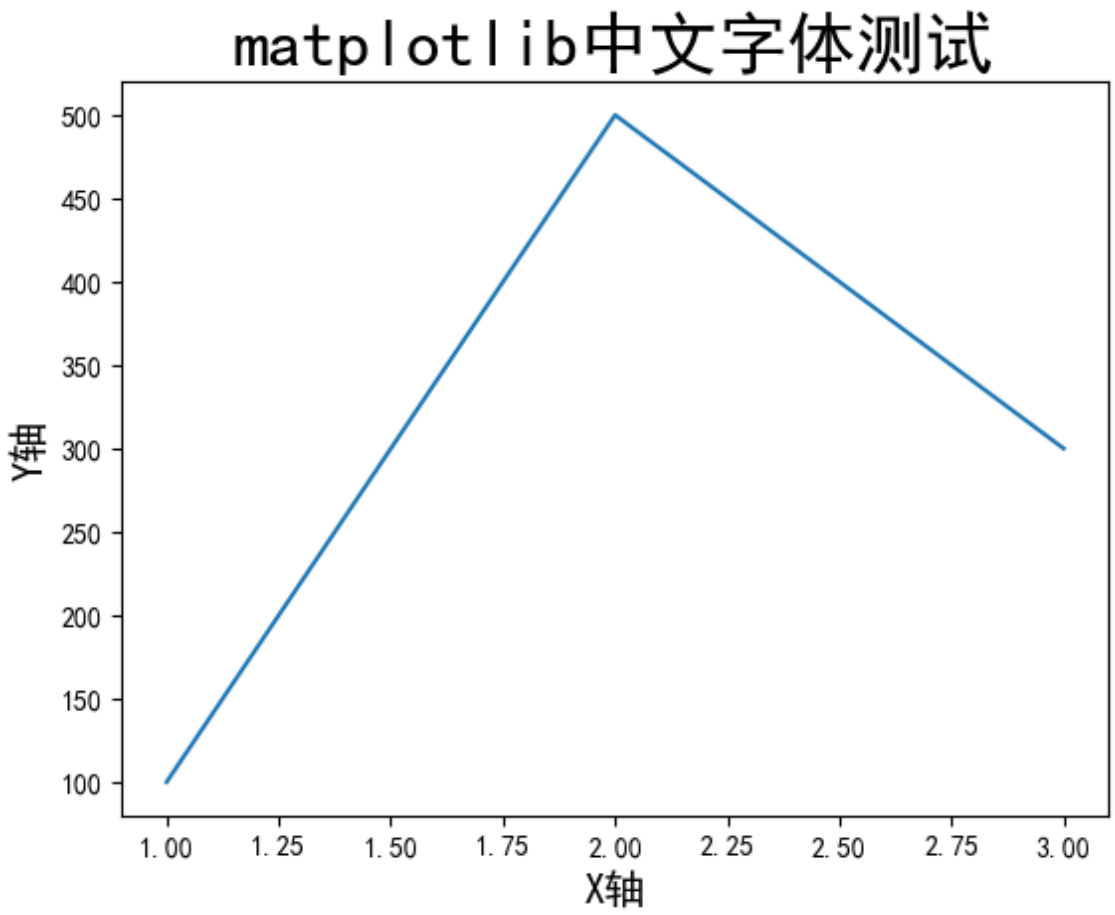
In [13]:

```
# Linux操作系统，例如 云GPU平台：https://featurize.cn/?s=d7ce99f842414bfcaea5662a97581bd1
# 如果遇到 SSL 相关报错，重新运行本代码块即可
!wget https://zihao-openmmlab.obs.cn-east-3.myhuaweicloud.com/20220716-mmclassification/dataset
!rm -rf /home/featurize/.cache/matplotlib

import matplotlib
import matplotlib.pyplot as plt
matplotlib.rc("font",family='SimHei') # 中文字体
```

```
/environment/miniconda3/lib/python3.7/site-packages/matplotlib/mpl-data/fonts/tt
f/SimHei.ttf: No such file or directory
'rm' 不是内部或外部命令，也不是可运行的程序
或批处理文件。
```

In [14]:

```
plt.plot([1,2,3], [100,500,300])
plt.title('matplotlib中文字体测试', fontsize=25)
plt.xlabel('X轴', fontsize=15)
plt.ylabel('Y轴', fontsize=15)
plt.show()
```



可视化辅助函数

In [15]:

```
from matplotlib import colors as mcolors
import random
random.seed(124)
colors = ['b', 'g', 'r', 'c', 'm', 'y', 'k', 'tab:blue', 'tab:orange', 'tab:green', 'tab:red', '
markers = [".",",","o","v","^","<",">","1","2","3","4","8","s","p","P","*","h","H","+","x","X","
linestyle = ['--', '-.', '-']

def get_line_arg():
    '''
    随机产生一种绘图线型
    '''
    line_arg = {}
    line_arg['color'] = random.choice(colors)
    # line_arg['marker'] = random.choice(markers)
    line_arg['linestyle'] = random.choice(linestyle)
    line_arg['linewidth'] = random.randint(1, 4)
    # line_arg['markersize'] = random.randint(3, 5)
    return line_arg
```

训练集损失函数

In [16]:

```
df_train.columns
```

Out[16]:

```
Index(['lr', 'data_time', 'loss', 'loss_cls', 'loss_bbox', 'time', 'epoch',
       'memory', 'step'],
      dtype='object')
```

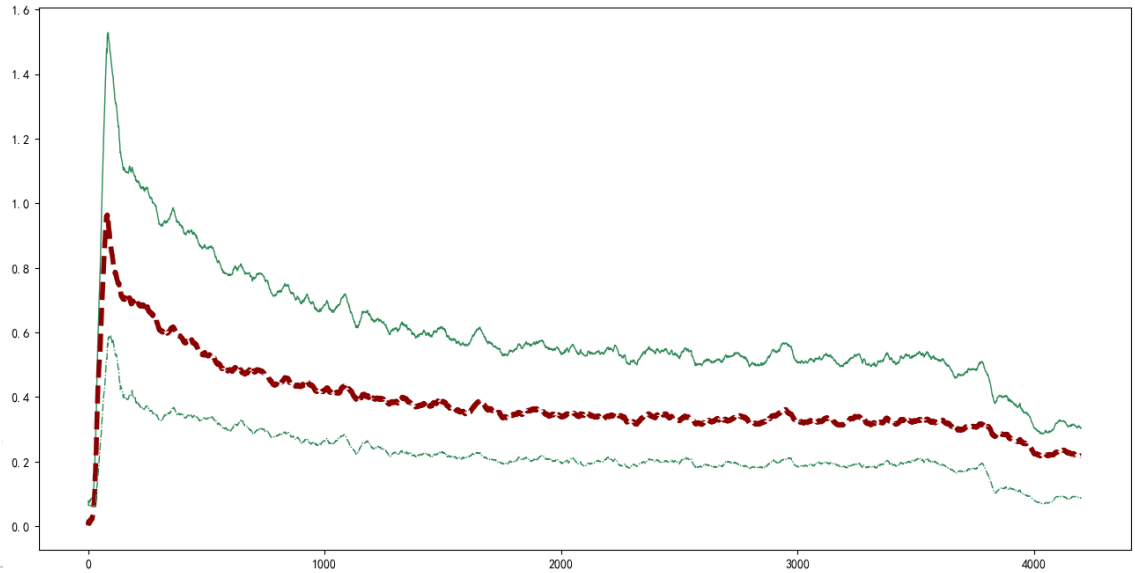
In [17]:

```
metrics = ['loss', 'loss_bbox', 'loss_cls', 'loss_rpn_cls', 'loss_rpn_bbox']
```

In [18]:

```
plt.figure(figsize=(16, 8))

x = df_train['step']
for y in metrics:
    plt.plot(x, df_train[y], label=y, **get_line_arg())

plt.tick_params(labelsize=20)
plt.xlabel('step', fontsize=20)
plt.ylabel('loss', fontsize=20)
plt.title('训练集损失函数', fontsize=25)
plt.savefig('训练集损失函数.pdf', dpi=120, bbox_inches='tight')

plt.legend(fontsize=20)

plt.show()
```

```
---------------------------------------------------------------------------
-----
KeyError                                  Traceback (most recent call last)
File d:\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3802, in
Index.get_loc(self, key, method, tolerance)
   3801 try:
-> 3802     return self._engine.get_loc(casted_key)
   3803 except KeyError as err:

File d:\anaconda3\lib\site-packages\pandas\_libs\index.pyx:138, in panda
s._libs.index.IndexEngine.get_loc()

File d:\anaconda3\lib\site-packages\pandas\_libs\index.pyx:165, in panda
s._libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:5745, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:5753, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

KeyError: 'loss_rpn_cls'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[18], line 5
      3 x = df_train['step']
      4 for y in metrics:
----> 5     plt.plot(x, df_train[y], label=y, **get_line_arg())
      7 plt.tick_params(labelsize=20)
      8 plt.xlabel('step', fontsize=20)

File d:\anaconda3\lib\site-packages\pandas\core\frame.py:3807, in DataFra
me.__getitem__(self, key)
   3805 if self.columns.nlevels > 1:
   3806     return self._getitem_multilevel(key)
-> 3807 indexer = self.columns.get_loc(key)
   3808 if is_integer(indexer):
   3809     indexer = [indexer]

File d:\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3804, in
Index.get_loc(self, key, method, tolerance)
   3802     return self._engine.get_loc(casted_key)
   3803 except KeyError as err:
-> 3804     raise KeyError(key) from err
   3805 except TypeError:
   3806     # If we have a listlike key, _check_indexing_error will raise
   3807     #  InvalidIndexError. Otherwise we fall through and re-raise
   3808     #  the TypeError.
   3809     self._check_indexing_error(key)

KeyError: 'loss_rpn_cls'
```




```
metrics = ['acc']
```

In [ ]:

```
plt.figure(figsize=(16, 8))

x = df_train['step']
for y in metrics:
    plt.plot(x, df_train[y], label=y, **get_line_arg())

plt.tick_params(labelsize=20)
plt.xlabel('step', fontsize=20)
plt.ylabel('loss', fontsize=20)
plt.title('训练集准确率', fontsize=25)
plt.savefig('训练集准确率.pdf', dpi=120, bbox_inches='tight')

plt.legend(fontsize=20)

plt.show()
```

测试集评估指标-MS COCO Metric

In [ ]:

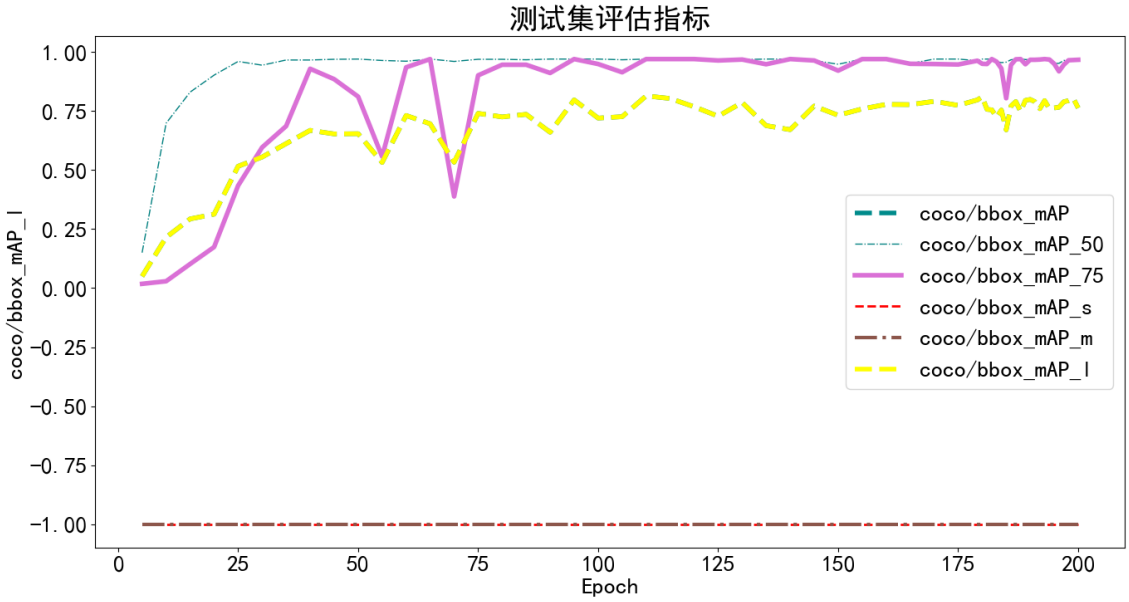
```
df_test.columns
```

In [19]:

```
metrics = ['coco/bbox_mAP', 'coco/bbox_mAP_50', 'coco/bbox_mAP_75', 'coco/bbox_mAP_s', 'coco/bbo
```

In [20]:

```
plt.figure(figsize=(16, 8))

x = df_test['step']
for y in metrics:
    plt.plot(x, df_test[y], label=y, **get_line_arg())

plt.tick_params(labelsize=20)
# plt.ylim([0, 100])
plt.xlabel('Epoch', fontsize=20)
plt.ylabel(y, fontsize=20)
plt.title('测试集评估指标', fontsize=25)
plt.savefig('测试集分类评估指标.pdf', dpi=120, bbox_inches='tight')

plt.legend(fontsize=20)

plt.show()
```



测试集评估指标-PASCAL VOC Metric

In [21]:

```
metrics = ['pascal_voc/mAP', 'pascal_voc/AP50']
```

In [22]:

```
plt.figure(figsize=(16, 8))

x = df_test['step']
for y in metrics:
    plt.plot(x, df_test[y], label=y, **get_line_arg())

plt.tick_params(labelsize=20)
# plt.ylim([0, 100])
plt.xlabel('Epoch', fontsize=20)
plt.ylabel(y, fontsize=20)
plt.title('测试集评估指标', fontsize=25)
plt.savefig('测试集分类评估指标.pdf', dpi=120, bbox_inches='tight')

plt.legend(fontsize=20)

plt.show()
```

```
---------------------------------------------------------------------------
-----
KeyError                                  Traceback (most recent call last)
File d:\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3802, in
Index.get_loc(self, key, method, tolerance)
   3801 try:
-> 3802     return self._engine.get_loc(casted_key)
   3803 except KeyError as err:

File d:\anaconda3\lib\site-packages\pandas\_libs\index.pyx:138, in panda
s._libs.index.IndexEngine.get_loc()

File d:\anaconda3\lib\site-packages\pandas\_libs\index.pyx:165, in panda
s._libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:5745, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:5753, in pandas._libs.hashtab
le.PyObjectHashTable.get_item()

KeyError: 'pascal_voc/mAP'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[22], line 5
      3 x = df_test['step']
      4 for y in metrics:
----> 5     plt.plot(x, df_test[y], label=y, **get_line_arg())
      7 plt.tick_params(labelsize=20)
      8 # plt.ylim([0, 100])

File d:\anaconda3\lib\site-packages\pandas\core\frame.py:3807, in DataFra
me.__getitem__(self, key)
   3805 if self.columns.nlevels > 1:
   3806     return self._getitem_multilevel(key)
-> 3807 indexer = self.columns.get_loc(key)
   3808 if is_integer(indexer):
   3809     indexer = [indexer]

File d:\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3804, in
Index.get_loc(self, key, method, tolerance)
   3802     return self._engine.get_loc(casted_key)
   3803 except KeyError as err:
-> 3804     raise KeyError(key) from err
   3805 except TypeError:
   3806     # If we have a listlike key, _check_indexing_error will raise
   3807     #  InvalidIndexError. Otherwise we fall through and re-raise
   3808     #  the TypeError.
   3809     self._check_indexing_error(key)

KeyError: 'pascal_voc/mAP'


<Figure size 1600x800 with 0 Axes>
```

In [ ]:

```

```

In [ ]:

```

```