

模型评估指标可视化，自动画Loss、Accuracy曲线图工具，无需人工参与！

mantch AINLP 4天前

注：本文转载自公众号AIArea，作者mantch



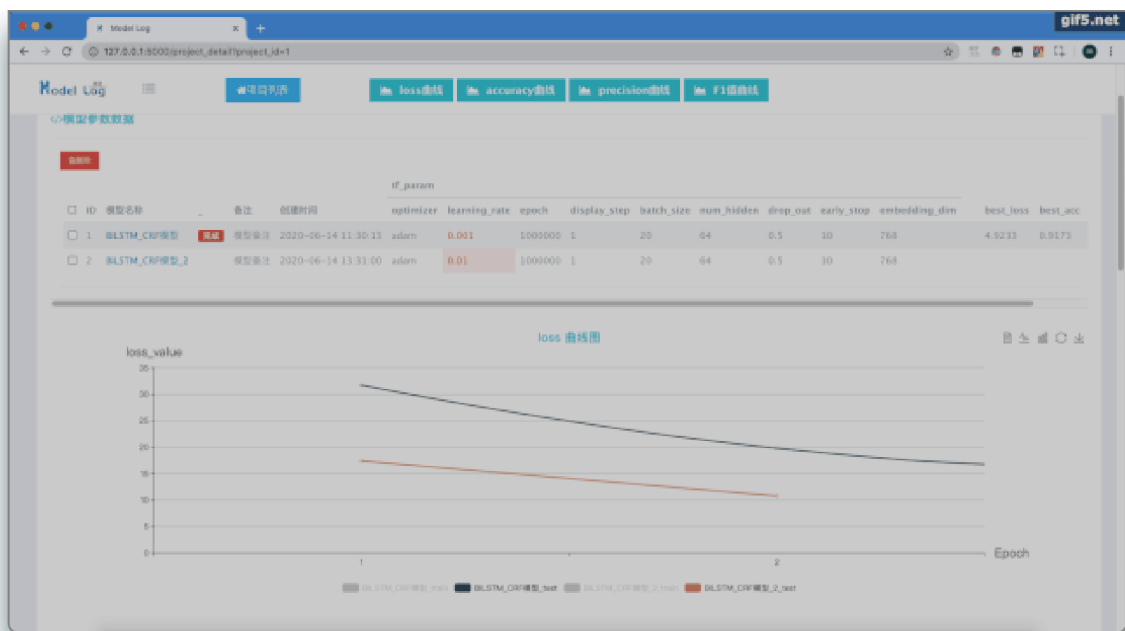
1. Model Log 介绍

Model Log 是一款基于 Python3 的轻量级机器学习(Machine Learning)、深度学习(Deep Learning)模型训练评估指标可视化工具，与 TensorFlow、Pytorch、PaddlePaddle 结合使用，可以记录模型训练过程当中的**超参数、Loss、Accuracy、Precision、F1值等，并以曲线图的形式进行展现对比**，轻松三步即可实现。GitHub项目地址：

https://github.com/NLP-LOVE/Model_Log

通过调节超参数的方式多次训练模型，并使用 Model Log 工具进行记录，可以很直观的进行模型对比，堪称调参神器。以下是使用工具后模型训练时 Loss 的变化曲线图。访问线上体验版：

http://mantchs.com/model_log.html



Loss变化曲线

通过上图可以很清晰的看出两个模型的训练效果，而且在表格当中高亮显示修改过的超参数，方便进行模型分析。

2. Model Log 特性

- 轻量级、无需任何配置、极简API、开箱即用。
- 只需要把模型的超参数和评估指标数据通过API添加即可，轻松三步即可实现。
- 高亮显示修改过的超参数，方便进行模型分析。
- 自动检测和获取正在训练的模型数据，并进行可视化，无需人工参与。
- 使用 SQLite 轻量级本地数据库存储，可供多个用户同时使用，保证每个用户看到的数据是独立的。
- 可视化组件采用 Echarts 框架，交互式曲线图设计，可清晰看到每个 epoch 周期的指标数据和变化趋势。

3. Model Log 演示地址

访问线上体验版：

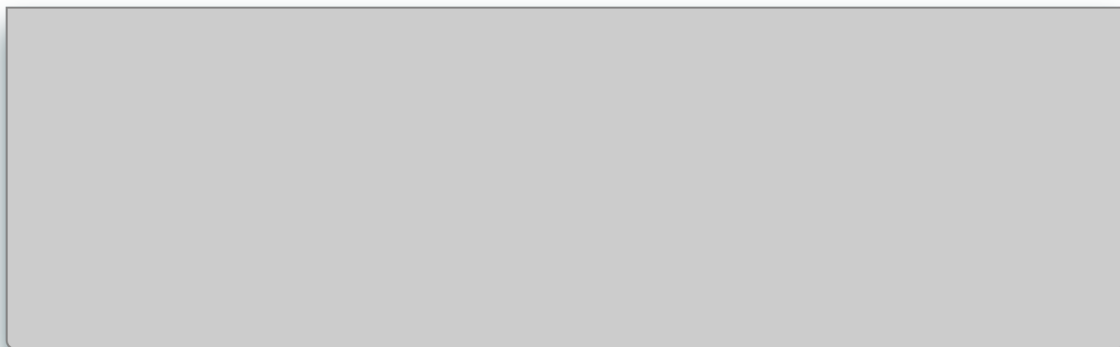
http://mantchs.com/model_log.html

4. Model Log 安装

Python3 版本以上，通过 pip 进行安装即可。

```
pip install model-log
```

注意：若安装的过程中出现以下情况，说明 **model-log** 命令已经安装到Python下的bin目录中，如果直接输入 **model-log** 可能会出现 **command not found**，可以直接到bin目录下执行。



img

5. Model Log 使用

5.1 启动 web 端

Model Log 安装成功后，Linux、Mac用户直接终端输入以下命令，Windows用户在cmd窗口输入：

```
model-log
```

默认启动 **5432端口**，可以在启动命令上使用参数 **-p=5000** 指定端口号。若提示命令不存在，可以直接到Python/3.7/bin目录下执行。

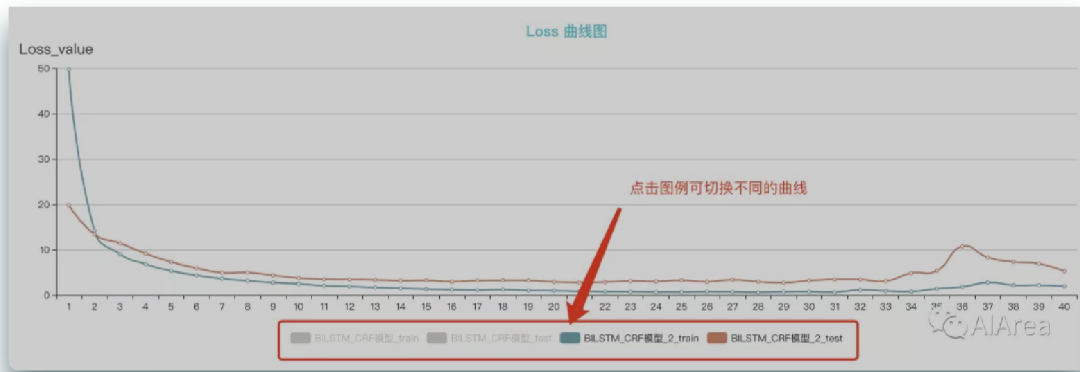
启动后可在浏览器输入网址进入：<http://127.0.0.1:5432>

也可访问线上体验版：

http://mantchs.com/model_log.html

- web首页是项目列表，一个项目可以有多个模型，这些模型可以在曲线图中直观比较。
- web 端会自动检测是否有新模型开始训练，如果有，直接会跳转到相应的 loss 等评价指标页，同时会自动获取指标数据进行呈现。
- 可供多个用户使用，添加昵称即可，SQLite 轻量级本地数据库存储，保证每个用户看到的数据是独立的。

- 通过点击曲线图下方的图例，可切换不同模型的评估曲线。



切换曲线

5.2 Model Log API使用

轻松三步即可使用

- 第一步**：先创建 ModelLog 类，并添加必要的属性

```
from model_log.modellog import ModelLog

"""

:param nick_name:      str, 昵称，多人使用下可起到数据隔离。
:param project_name:   str, 项目名称。
:param project_remark: str, 项目备注，默认为空。

项目名称如不存在会新建

"""

model_log = ModelLog(nick_name='mantch', project_name='demo实体识别', project_remark='')

"""

:param model_name: str, 模型名称

"""

model_log.add_model_name(model_name='BILSTM_CRF模型')

"""

:param remark: str, 模型备注

"""

model_log.add_model_remark(remark='模型备注')

"""
```

```

:param param_dict: dict, 训练参数字典

:param param_type: str, 参数类型, 例如: TF参数、Word2Vec参数等。
"""

model_log.add_param(param_dict={'lr':0.01}, param_type='tf_param')

```

2. **第二步**: 模型训练的每次 epoch (周期)可以添加评估指标数据, 评估指标可以进行以下选择。

第一次调用该 API 时, 会把以上设置的数据(模型名称、备注等)持久化到 SQLite 数据库, 并且 web 端会自动获取评估指标数据进行图形化展示。

```

"""

:param metric_name: str, 评估指标名称,
    可选择['train_loss', 'test_loss', 'test_acc', 'test_recall', 'test_precision', 'tes

:param metric_value: float, 评估指标数值。

:param epoch: int, 训练周期

metric_name 参数只可以选择以上六种

第一次调用该 API 时, 会把以上设置的数据(模型名称、备注等)持久化到 SQLite 数据库, 并且 web 端会自动获
可以在每个 epoch 周期的最后使用该 API 添加训练集和测试集的评估指标, web 端会自动获取该数据。
"""

model_log.add_metric(metric_name='train_loss', metric_value=4.5646, epoch=1)

```

3. **第三步**: 模型训练完成后, 可以添加最好的一次评估数据。

```

"""

:param best_name: str, 最佳评估指标名称,

:param best_value: float, 最佳评估指标数值。

:param best_epoch: int, 训练周期

添加当前模型训练中最佳的评估数据, 一般放到模型训练的最后进行添加。
"""

model_log.add_best_result(best_name='best_loss', best_value=1.2122, best_epoch=30)

"""

关闭 SQLite 数据库连接

```

```
"""  
  
model_log.close()
```

5.3 Model Log 使用示例

MIST手写数字识别：https://github.com/NLP-LOVE/Model_Log/blob/master/demo_TF_MIST.py

推荐阅读

[这个NLP工具，玩得根本停不下来](#)

[征稿启示| 200元稿费+5000DBC（价值20个小时GPU算力）](#)

[文本自动摘要任务的“不完全”心得总结番外篇——submodular函数优化](#)

[Node2Vec 论文+代码笔记](#)

[模型压缩实践收尾篇——模型蒸馏以及其他一些技巧实践小结](#)

[中文命名实体识别工具（NER）哪家强？](#)

[学自然语言处理，其实更应该学好英语](#)

[斯坦福大学NLP组Python深度学习自然语言处理工具Stanza试用](#)

关于AINLP

AINLP 是一个有趣有AI的自然语言处理社区，专注于 AI、NLP、机器学习、深度学习、推荐算法等相关技术的分享，主题包括文本摘要、智能问答、聊天机器人、机器翻译、自动生成、知识图谱、预训练模型、推荐系统、计算广告、招聘信息、求职经验分享等，欢迎关注！加技术交流群请添加 AINLPer(id: ainlper)，备注工作/研究方向+加群目的。

AINLP

我爱自然语言处理

一个有趣有AI的自然语言处理社区



长按扫码关注我们

阅读至此了，分享、点赞、在看三选一吧 

[阅读原文](#)