Data evaluation - Baselines and MAML

Sebastian Pineda-Arango

To do (Last week)

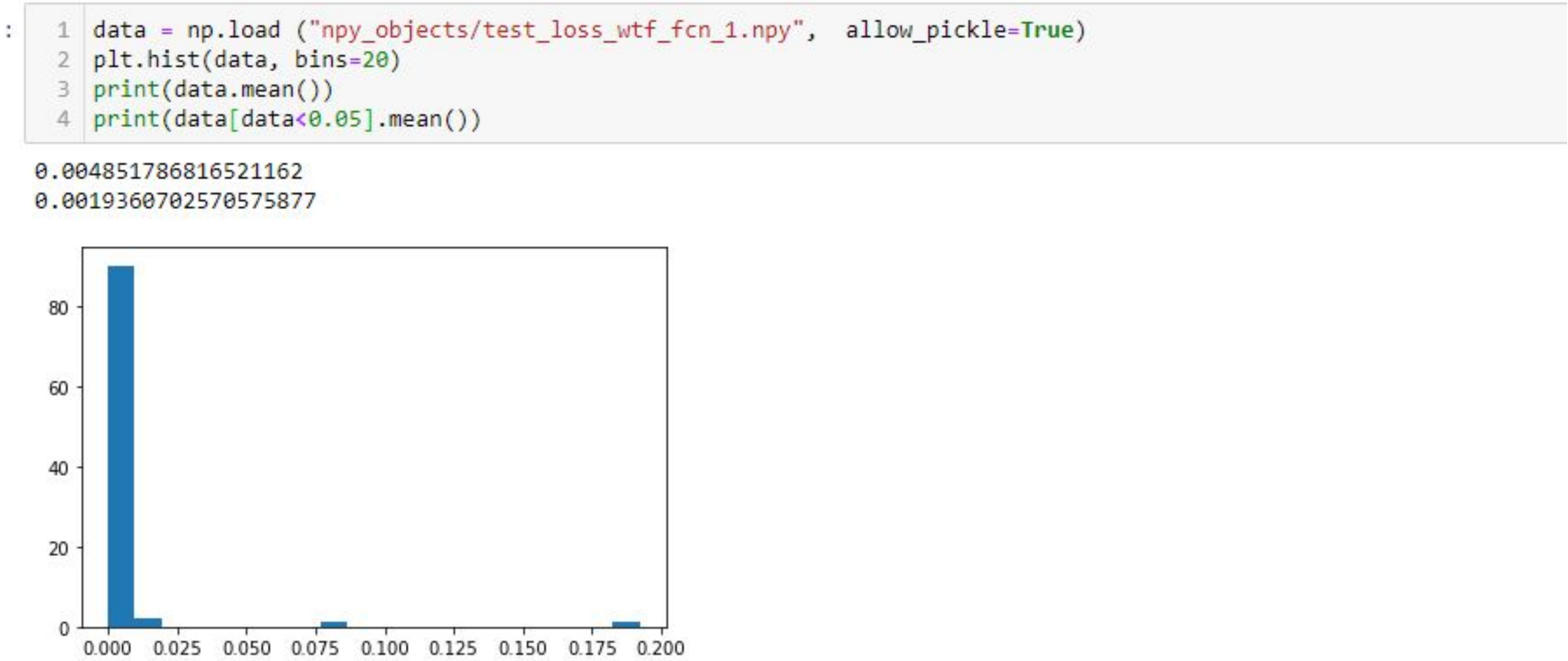
- FCN fine-tuning.
- MAML hyperparameter tuning.
- Compare the *learn2learn* with *higher.*

Comparison *Higher* and *Learn2learn*

- They do not need the definition of new modules
- Free
- Similar performance
- Also compared own implementation based on the original code of MMAML. It needs further validation.

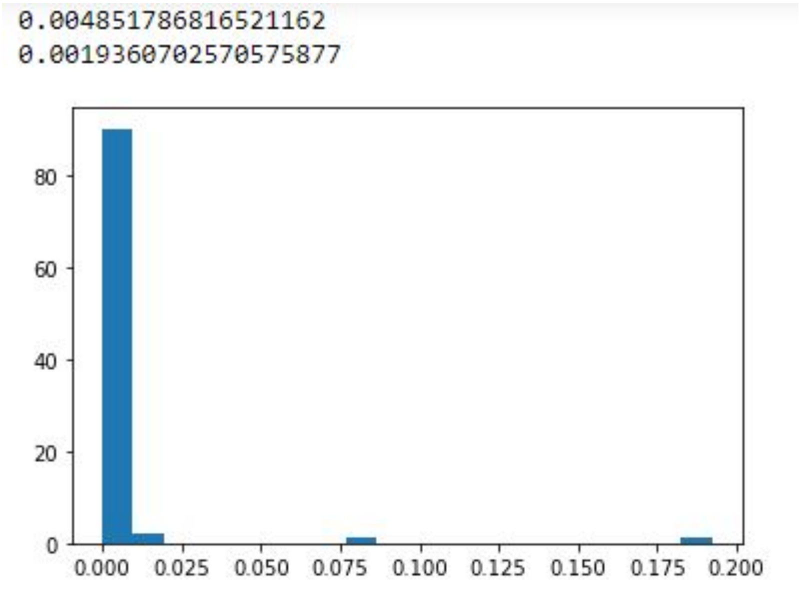
Performance of FCN fine-tuning

- Some tasks exhibit very bad performance affecting the overall mean.
- Even only fine-tuning the last layer.
- Hypothesis: strong distribution shift in some tasks (train -> test).

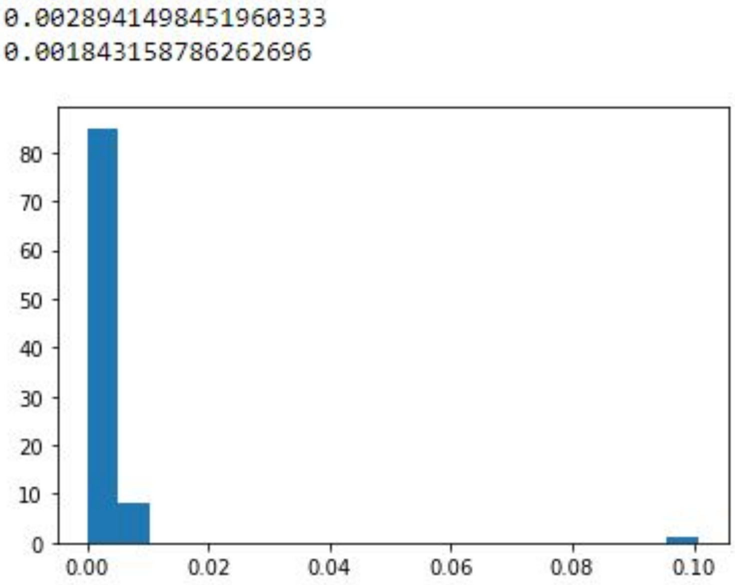


```
data = np.load ("npy_objects/test_loss_wtf_fcn_1.npy",  allow_pickle=True)
plt.hist(data, bins=20)
print(data.mean())
print(data[data<0.05].mean())
```

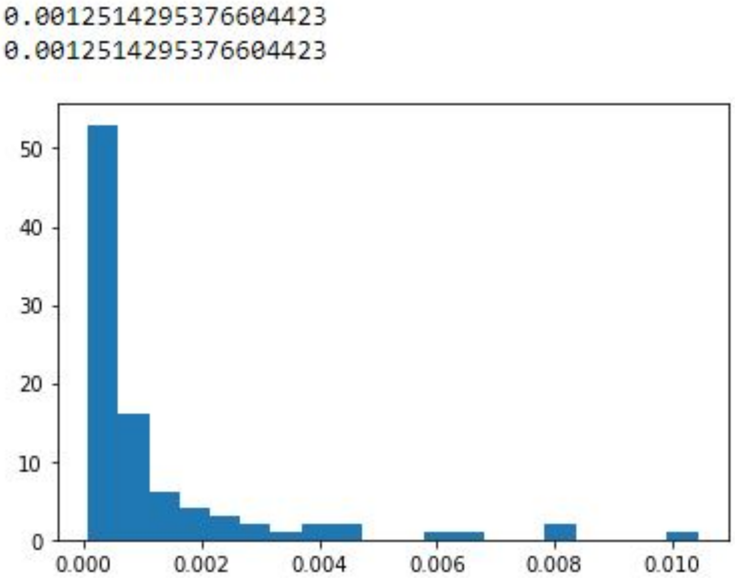
```
0.004851786816521162
0.0019360702570575877
```



```
0.004851786816521162
0.0019360702570575877
```



```
0.0028941498451960333
0.001843158786262696
```



```
0.0012514295376604423
0.0012514295376604423
```

Hyperparameter tuning

Hyper-parameters:

- Fast learning rate
- Slow learning rate (meta-learning rate): always 0.5*(fast learning rate)
- Adaptation steps

Meta-learning libraries

- Learn2learn: http://learn2learn.net/ (947 stars)
- Higher: https://github.com/facebookresearch/higher (890)
- Torchmeta: https://github.com/tristandeleu/pytorch-meta (990)

To do (Next week)

- FCN fine-tuning: Batch normalization, different number of tasks.
- MAML evaluation.

Timeline

Time	Tasks
August	Literature review, data exploration
September	Baselines implementation
October	Proposed model implementation
November	Experiments on models (Hyper. Tun., etc.)
December	Results evaluation and adjustments
January	Results report and thesis finalization