Deep learning

2.4. Proper evaluation protocols

François Fleuret

https://fleuret.org/dlc/

Dec 20, 2020

Learning algorithms, in particular deep-learning ones, require the tuning of many meta-parameters.

Learning algorithms, in particular deep-learning ones, require the tuning of many meta-parameters.

These parameters have a strong impact on the performance, resulting in a "meta" over-fitting through experiments.

Learning algorithms, in particular deep-learning ones, require the tuning of many meta-parameters.

These parameters have a strong impact on the performance, resulting in a "meta" over-fitting through experiments.

We must be extra careful with performance estimation.

Learning algorithms, in particular deep-learning ones, require the tuning of many meta-parameters.

These parameters have a strong impact on the performance, resulting in a "meta" over-fitting through experiments.

We must be extra careful with performance estimation.

Running 100 times the MNIST experiment, with randomized weights, we get:

Worst	Median	Best
1.3%	1.0%	0.82%

The ideal development cycle is

Write code $\longrightarrow$ Train

The ideal development cycle is

Write code $\longrightarrow$ Train $\longrightarrow$ Test

The ideal development cycle is

Write code $\longrightarrow$ Train $\longrightarrow$ Test $\longrightarrow$ Paper

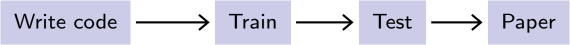
The ideal development cycle is

$$\begin{tikzcd} \text{Write code} \arrow[r] & \text{Train} \arrow[r] & \text{Test} \arrow[r] & \text{Paper} \end{tikzcd}$$

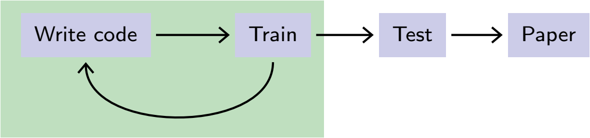
or in practice something like

$$\begin{tikzcd} \text{Write code} \arrow[r] & \text{Train} \arrow[r] \arrow[l, bend left] & \text{Test} \arrow[r] & \text{Paper} \end{tikzcd}$$

The ideal development cycle is

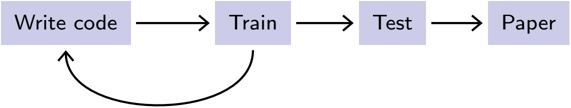


or in practice something like

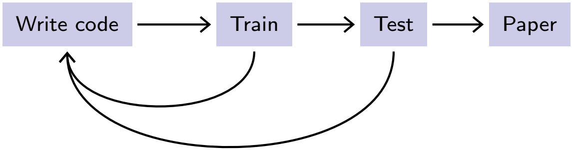


There may be over-fitting, but it does not bias the final performance evaluation.

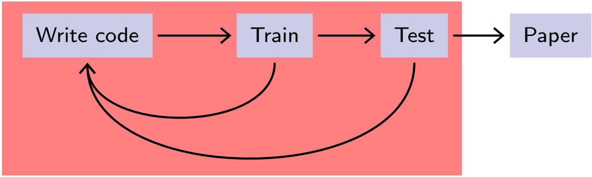
Unfortunately, it often looks like



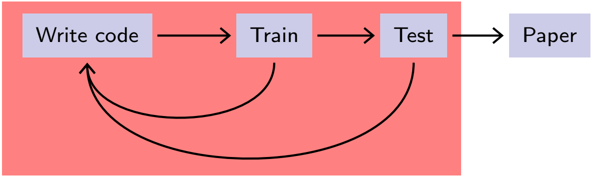
Unfortunately, it often looks like



Unfortunately, it often looks like

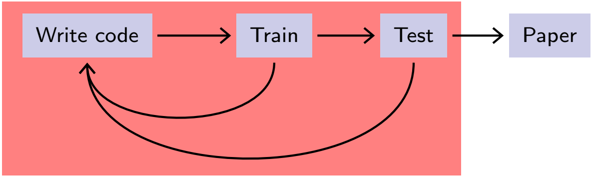


Unfortunately, it often looks like

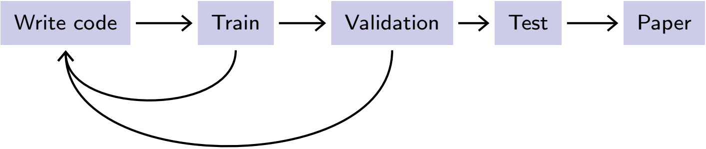


⚠ This should be avoided at all costs. The standard strategy is to have a separate validation set for the tuning.

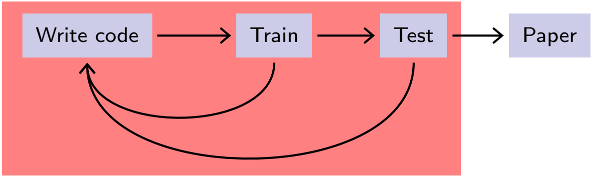
Unfortunately, it often looks like



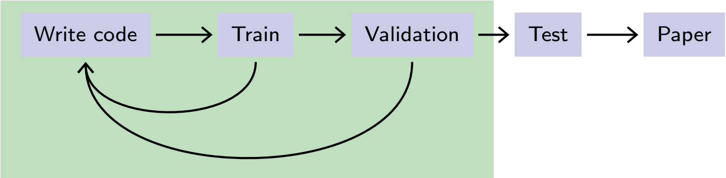
This should be avoided at all costs. The standard strategy is to have a separate validation set for the tuning.



Unfortunately, it often looks like



This should be avoided at all costs. The standard strategy is to have a separate validation set for the tuning.



When data is scarce, one can use cross-validation: average through multiple random splits of the data in a train and a validation sets.

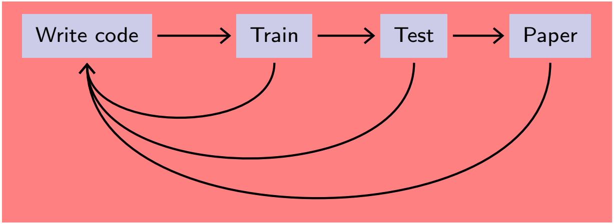
When data is scarce, one can use cross-validation: average through multiple random splits of the data in a train and a validation sets.

There is no unbiased estimator of the variance of cross-validation valid under all distributions (Bengio and Grandvalet, 2004).

Some data-sets (MNIST!) have been used by thousands of researchers, over millions of experiments, in hundreds of papers.

Some data-sets (MNIST!) have been used by thousands of researchers, over millions of experiments, in hundreds of papers.

The global overall process looks more like



"Cheating" in machine learning, from bad to "are you kidding?":

- "Early evaluation stopping",
- meta-parameter (over-)tuning,
- data-set selection,
- algorithm data-set specific clauses,
- seed selection.

"Cheating" in machine learning, from bad to "are you kidding?":

- "Early evaluation stopping",
- meta-parameter (over-)tuning,
- data-set selection,
- algorithm data-set specific clauses,
- seed selection.

Top-tier conferences are demanding regarding experiments, and are biased against "complicated" pipelines.

The community pushes toward accessible implementations, reference data-sets, leader boards, and constant upgrades of benchmarks.

The end

References

Y. Bengio and Y. Grandvalet. **No unbiased estimator of the variance of k-fold cross-validation**. Journal of Machine Learning Research (JMLR), 5:1089–1105, 2004.