Spices for Successful ML Project

IndabaXTanzania 2019

Anthony Faustine

PhD machine learning researcher
IDLab research group-Ghent University
Belgium.

12^{th} April 2019

GHENT UNIVERSITY

IDLab
INTERNET & DATA LAB

@sambaiga


Mixed Nuts for Rise
Iraqi Rise
Chini Rise
Soup Spice
Hot Paprika
Chicken Spice
Ras El Hanout
Black Pepper
Indian Rise
Potteto Spice
Meat Spice
Kurkuma
Sumac
Roast Beef Spice
Toscana (Pizza)

Profile

PhD machine learning researcher, IDlab, imec

`A machine learning researcher passionate on using cutting-edge technology to create intelligence system that can reason and understand`

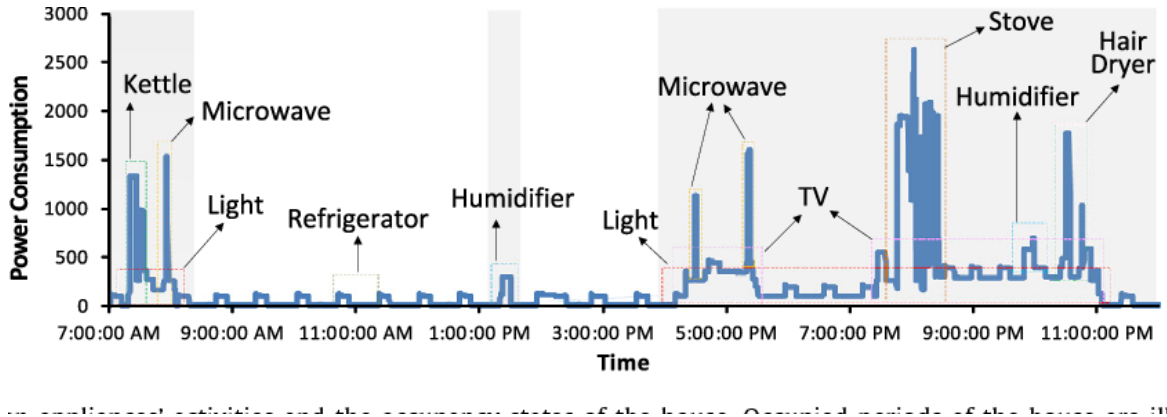


Figure 1: Research: NILM

co-founder: pythontz, indabatz, parrotai

`strive for excellence money will follow..`

@sambaiga

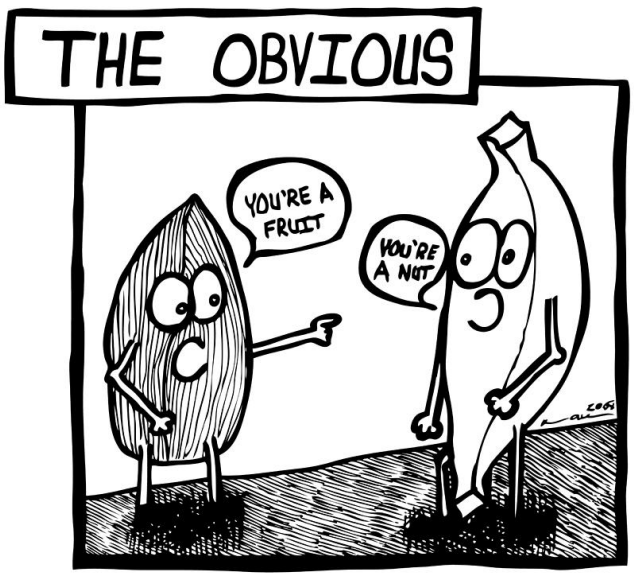
Introduction

To have successfully and publishable ML project:

- Identify open problem
- Design experiment
- Get dataset
- Define evaluation metric
- Write code
- Run an experiment
- Analyse results

Identify open problem

Dont do the obvious



1. Do literature review
   - Learn about common, methods, dataset and libraries.
   - Identify open questions that need answers
2. Establish hypothesis about the problem

Design experiment

1. Define performance metric
2. Establish baseline performance
   - Any publishable performance with simplest approach.
   - Define your baseline.
   - Use best published performance.
3. Establish upper bound.
4. Establish project management.
   - Folder structure.
   - Version control (gitlab, github etc).

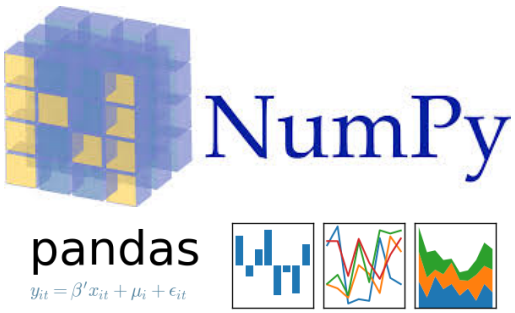
Dataset

You may need more than one data-set to benchmark your solution.

1. At least one dataset that appeared in related prior work.
2. Source of dataset
   - Build them.
   - Scrape them.
   - Find them (contact authors).
   - Generate them (Artificial data).
   - Folder structure.
   - Version control (gitlab, github etc).
3. Prepare them for your experiment.

Write code quickly: Use Framework

Make sure you can bypass the abstraction when needed

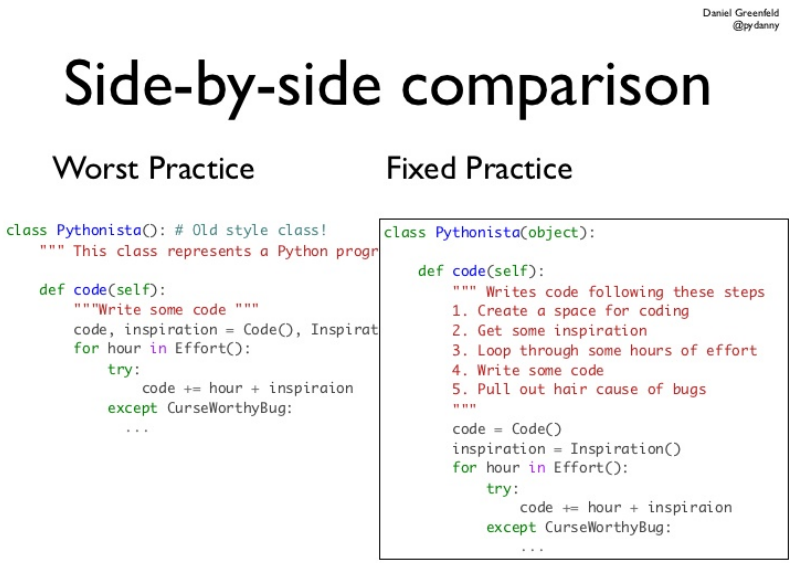


Write code quickly: Get a good starting point

First get a baseline running ⇒this is good research practise.

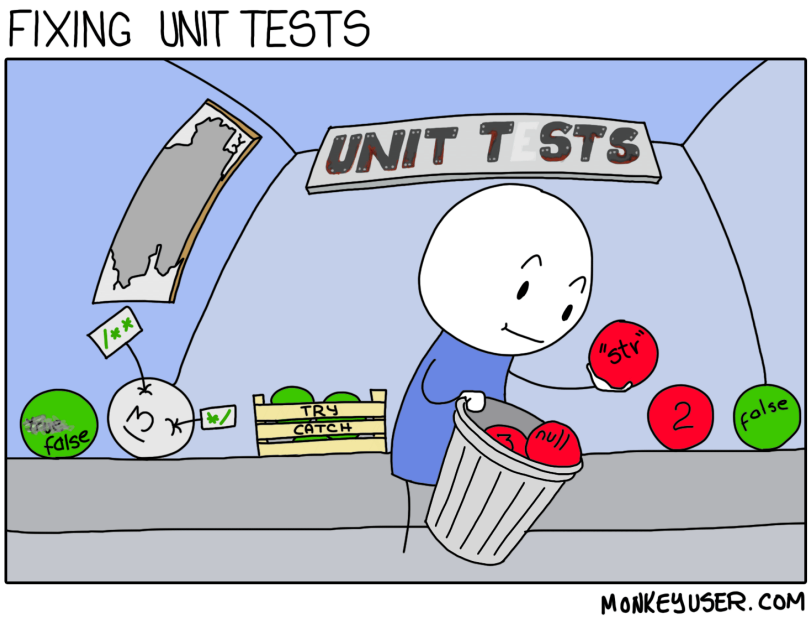
Write code quickly: Use good code style

Write code for people, not machines



- Add comments and include expression in your module.
- Use meaningful names.
- Add comments about tensors shape
- Add comments describing non-obvious logic

Write code quickly: Include minimum testing



- Test some parts of your code.
- Make sure data processing works consistently.
- Test if tensor operations runs as expected
- Test weather gradients are non-zero.

Write code quickly: Reduce hard-coding

Reduce hard coding as much as you can.

```
class MyModel(Model):
    def __init__(self):
        self.input_embedding = Embedding(100)
        self.encoder = LSTM(100, 200)
        self.my_novel_bits = ...
```

```
class MyModel(Model):
    def __init__(self,
                 input_embedding: TextFieldEmbedder,
                 encoder: Seq2SeqEncoder):
        self.input_embedding = input_embedding
        self.encoder = encoder
        self.my_novel_bits = ...
```

- Use configurations files (JSON, YAML, or text files) and or argparse module.
- Allow you to start simple and later expand without rewriting your code.
- Make controlled experiments easier.

Write code: important take-away

- Build and test code to load and process your data.
- Build and test code for simple baseline.
- Build and test code to evaluate results.
- Write re-usable codes.

Run experiment

Keep track of what you ran

- keep track of what happen, when and with what code.

Experimenter	git SHA	Background Search Method	Model	Dataset	Train Acc	Validation Acc	Notes
Pradeep	fc8d6ca3	Lucene	QAMNS (50d)	Intermediate	0.3114	0.3045	patience=20
Pradeep	fc8d6ca3	Lucene	QAMNS (300d)	Intermediate	0.8317	0.3864	patience=20
Pradeep	fc8d6ca3	BOW-LSH question+answers Glove 50d	QAMNS (50d)	Intermediate	0.3008	0.35	patience=20
Pradeep	fc8d6ca3	BOW-LSH question+answers Glove 50d	QAMNS (300d)	Intermediate	0.7466	0.4227	patience=20

- Save model checkpoint files for all reasonably effective/interesting experiments
- Not recommended: modifying code to run different variants → hard to keep track of what you ran.
- Analyse model behaviour during training →Use Tensor board, Logging etc.
- Take notes of what each experiment was meant to test.

Quantitative evaluation

- Follow prior work precisely in how to choose and implement main evaluation metric.
- Show metric as many variants of your model as you can
- Test for statistical significance (for highly variable models or small difference performance).
- If your results are not significant. say so and explain what you found.

Qualitative-evaluation

This is the analysis section

- convince reader for your hypothesis.
- Look to prior work to get started
- Show examples of system output.
- Present error analysis.
- Visualize your hidden states.
- Plot how your model performance varies with the amount of data.
- Include an on-line demo.
- If your results are not significant. say so and explain what you found.

Formative vs summative evaluation

When the cook tastes the soup that is formative; when the customer testes that is summative.

Formative evaluation

- They guide further investigations
- Compare design option A to B, tune hyper-parameters etc

Summative evaluation

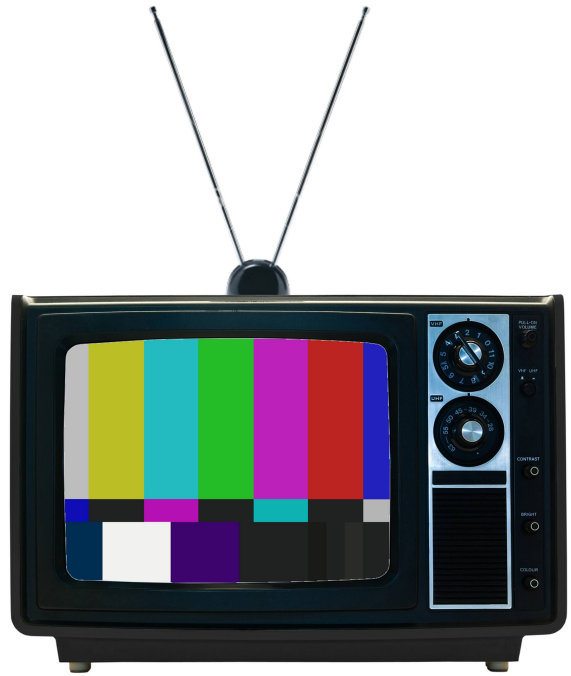
- compare your approach to previous approaches,
- compare different major variants of your approach.
- only use test set.

Note: Don't save all your qualitative evaluation for the summative evaluation.

Strategies to improve ML performance

The challenges → so many things to try or change (hyper-parameters etc).

- Be specific on what to tune in order to try achieving one effect.
- For Supervised ML system focus to achieve:
  1. Best performance in training set.
  2. Best performance in validation/dev set.
  3. Best performance in test set.
  4. Perform well in real world.

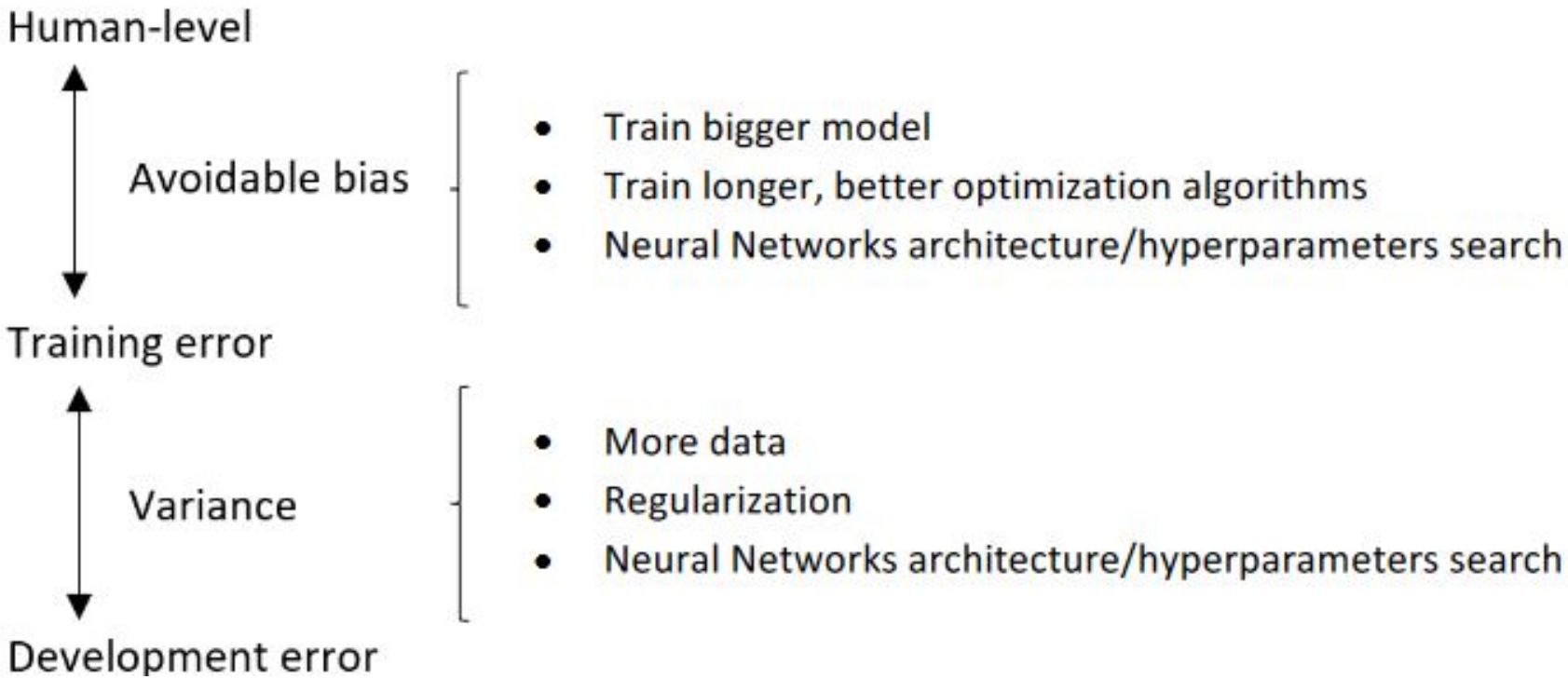


Use different knobs (parameters) to improve performance of each part.

Strategies to improve ML performance

1. To improve performance in training set
   - use bigger neural network or switch to a better optimization algorithms (adam etc)
2. To improve performance in validation/dev set
   - Apply regularization or use bigger training set.
3. To improve performance in test set
   - Increase size of dev set.
4. Poor performance in real world.
   - Change development set, modify your objective function/hypothesis.

Bias Variance Analysis: Avoidable bias



- If avoidable bias $>$ variance focus on reducing bias.
- If avoidable bias $<$ variance focus on reducing variance.

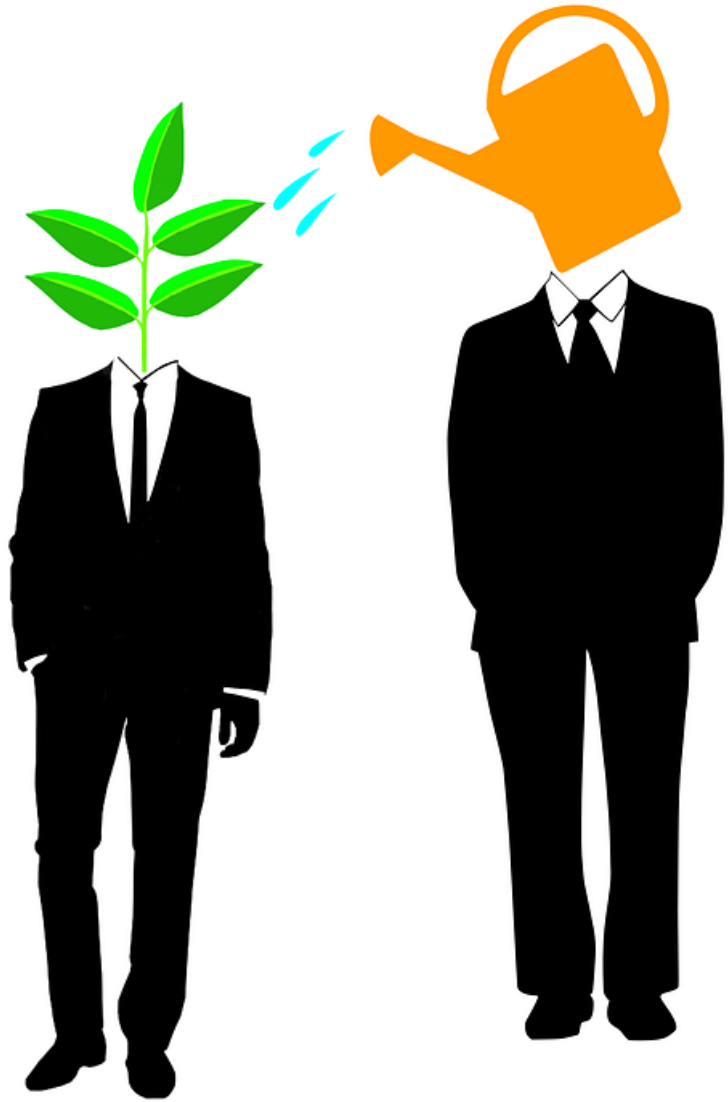
Error-Analysis

If the performance of your ML algorithm is still poor compared to human level performance →perform error analysis

- Manually examine mistakes that your ML algorithm is making → gain insight of what to do next.

Ethics and integrity

Professional ML/DS mentor



Conclusion

plug & play

The Scientific Method

References

1. My personal experiences.
2. Writing Code for NLP Research, Joel Grus.
3. Foundations: How to design experiments in NLU, Sam Bowman.
4. Machine Learning Yearning, Andrew Ng.