Project report

Neronet

Toolbox for managing the training neural networks

CSE-C2610
Software Project

Aalto University

April 13, 2016

Aalto University
School of Science
and Technology

Outline

Forecitation

"I have supervised several bachelors' theses and other student projects.

You and the product surpassed my expectations. I did not expect such professional results."

Pyry Takala, PO

Outline

Team

At the start of the course, we were all both

- terrified, and
- inspired

by our project topic!

Product owner

Pyry Takala

- Machine learning researcher at Aalto
- Past: Amazon, McKinsey & Company, Goldman Sachs, August...

Agile coach

Eero Laukkanen

- Continuous delivery researcher at Aalto
- Past: Prosys PMS Ltd, WiRCA-miehet, Aalto...

Outline

Pyry's problem

Challenges in training neural networks:

- Long duration of individual experiments
- Difficult to manage of experiment variations, queues and computing nodes
- Difficult to monitor and analyse experiment progress
- Difficult to log experiment histories

Ideas

Could a toolbox be developed that would

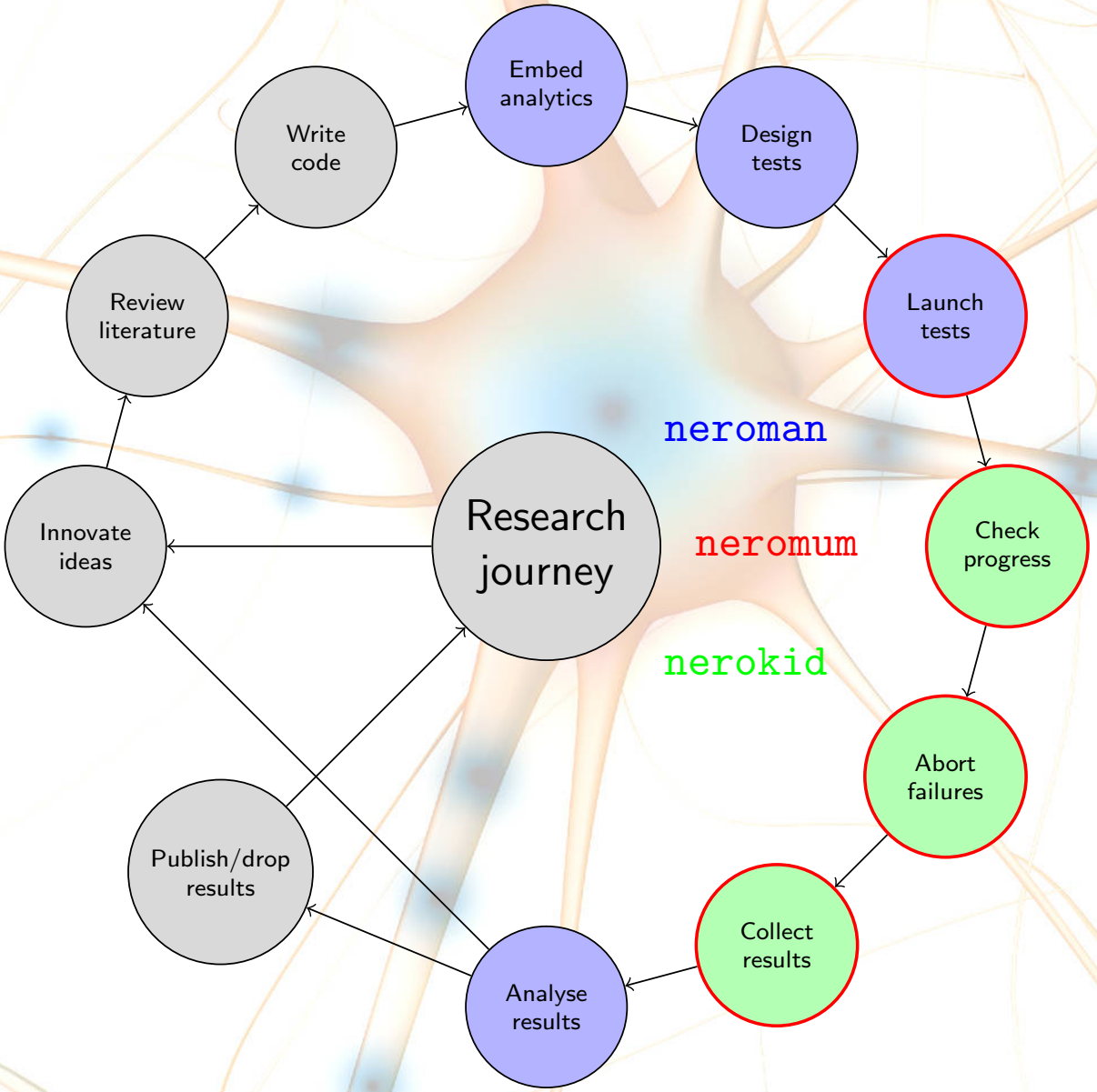
- monitor experiment logs and detect problems
- notify of and/or autoterminate poorly developing experiments
- facilitate experiment variation
- faciliate experiment and queue management
- enable easy access to experiment data
- potentially also visualize parameter evolution

Ideas

Such a toolbox should also be

- lightweight
- easily extensible
- open source
- framework agnostic
- convenient in practice

Journey map


Embed analytics
Write code
Design tests
Review literature
Launch tests
neroman
Research journey
Innovate ideas
neromum
Check progress
nerokid
Abort failures
Publish/drop results
Collect results
Analyse results

Technical challenges & ASRs

- deeply technical and an advanced domain
- remote computing nodes and cluster management
- good usability? according to whom?
- framework agnostic?

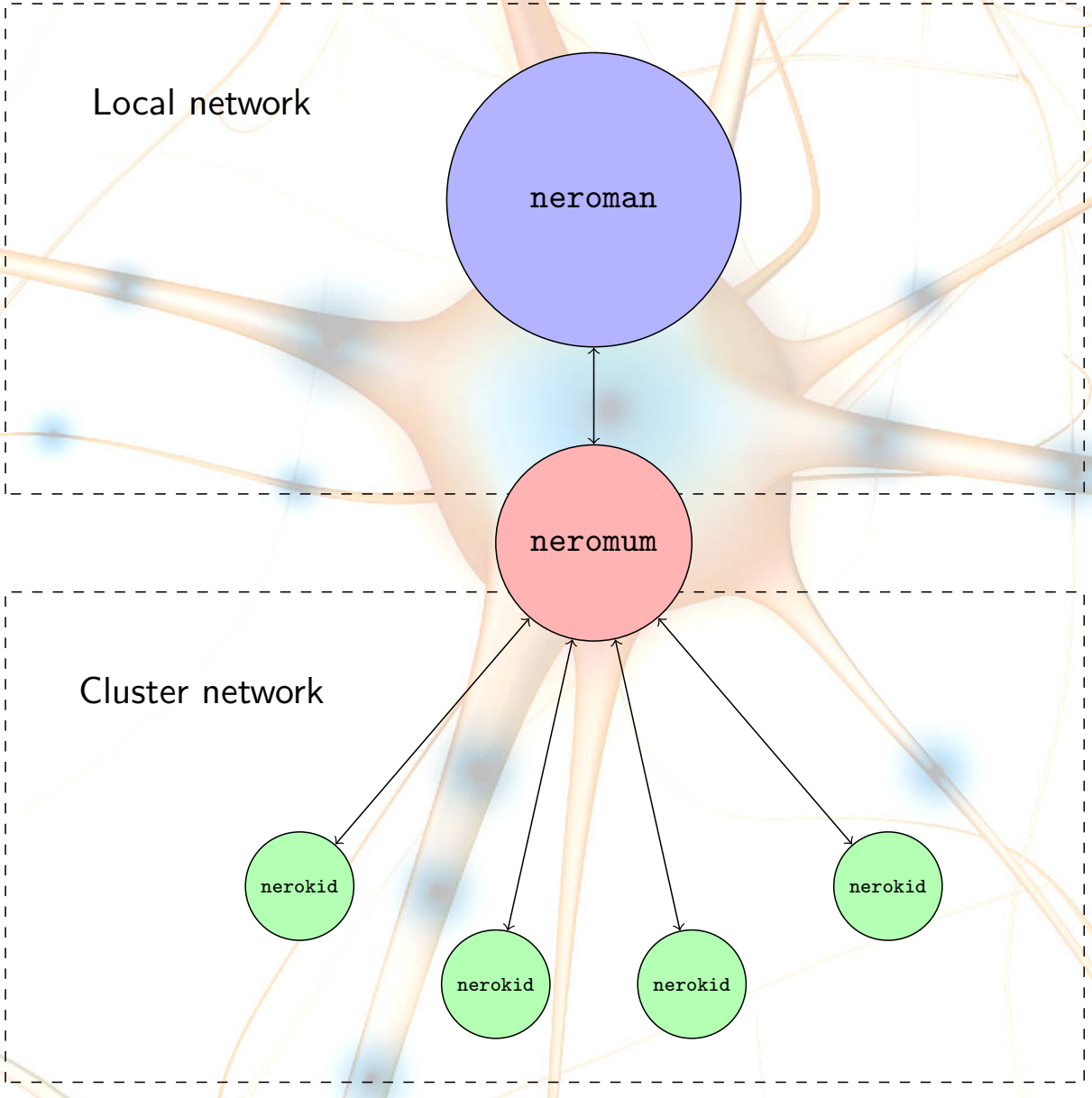
Outline

Result

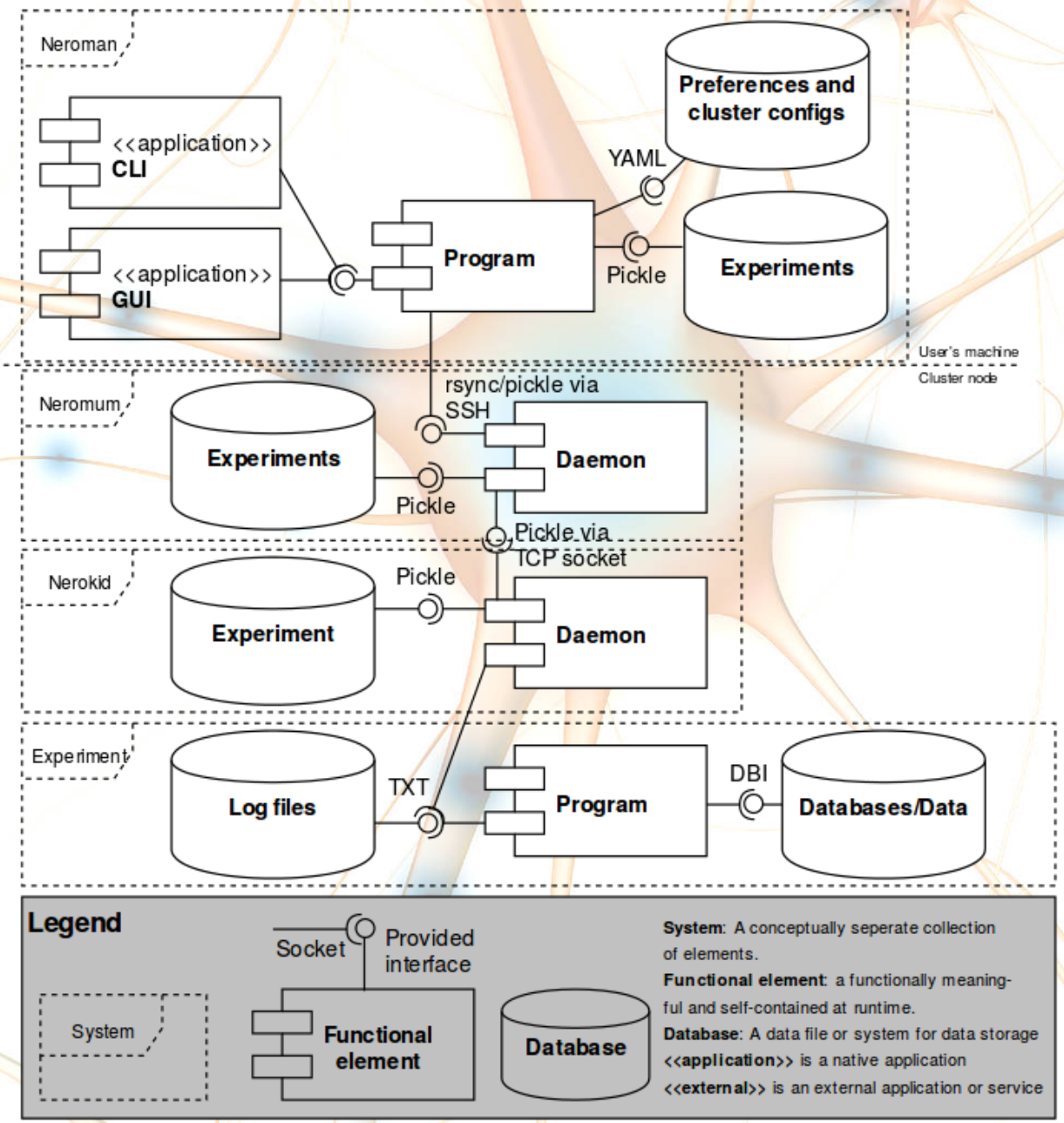
A three component tool involving Python, SSH and sockets that ... meets the vision!

- All core functionality Done.
- Most features accessible both via CLI and GUI.
- PO satisfied.
- We are proud!

Solution


Local network
neroman
neromum
Cluster network
nerokid
nerokid
nerokid
nerokid

Structural architecture



Features

Core features:

- extensible experiment specification system
- easy experiment submission to remote nodes
- experiment state and progress information easily accessible
- access to computing node resources information
- customizable notification and autotermination system
- additional conveniences: drag drop, experiment template, customizable parameter value plotting

Outline

Demo

Live demonstration of the final product including:

1. Cluster & experiment configuration
2. Experiment submission
3. Fetching updates/results
4. Viewing plots

Focus on the GUI this time.

Recap

In essence the product is a Python-based tool that enables computational researchers to conduct their research more effectively.

- It utilizes SSH and TCP sockets to distribute computational experiments into remote computer nodes.
- It is framework agnostic in that it permits the use of a very wide variety of tools to actually conduct the computing needed (Theano, Torch, Scipy, Matlab, R, C++, whatever).

User feedback

We utilized lots of end user testing and received great feedback from Simo Tuomisto, a Triton admin:

- Thought that our design and architecture was well devised.
- Offered performance improvement ideas.
- In general he liked our usability.

User feedback

From our peer team we received feedback as well.
Thus we did several

- bug fixes
- usability improvements
- improvements to project documentation and the user guide

User feedback

Pyry also gave us great feedback during development, regarding requirements and usability etc.
In the end he claimed that

- the end product was surprisingly professional
- the usability was excellent
- and that our team had expertise

Outline

Quality

Quality assurance was based on

- the DoD, and
- our QA practices

We updated our DoD and practices several times.

Definition of done

We defined Done at the BI, sprint and project level.

- BI level: unit tests, code confromity, commenting, documentation, peer review
- Sprint level: BIs Done, unit and system tests ok, sprint goal achieved.
- Project level: all sprints Done and PO satisfied.

QA practices

Used QA practices and tools:

- Unit test framework
- Commenting & documentation
- System testing
- Peer review

Quality attributes

Added emphasis on these attributes:

- Usability
- Reliability
- Extendability
- Performance

Confessions

Achieving great quality was/is very challenging!

- Several issues are still unsolved
- More stress testing would be necessary
- There is room for refactoring

Outline

Process overview

- Each developer as a sprint team leader in turn
- Teamwork sessions almost every Wed & Fri at Maari
- Scrum as required
- Tried to employ professional SD practices
- Tools: Github, Agilefant, Flowdock, WhatsApp, Google Calendar, ShareLaTeX, Skype, Floobits

Schedule

Time	Event	Participants
30.10. 16-18	Project kickoff	team, PO
13.11. 15-17	S0 demo	team, Coach
16.11. 11-13	S1 planning	team, PO
04.12. 16-17	S1 & progress review	team, PO, Coach
04.12. 17-18	S2 planning	team, PO
13.01. 19-20	S2 review & S3 planning	team, PO
01.02. 14-16	S3 review & S4 planning	team, PO
29.02. 13-14	S4 & progress review	team, PO, Coach
29.02. 14-15	S5 planning	team, PO
30.03. 16-18	S5 review & S6 planning	team, PO
13.04. 16-17	S6 & project review	team, PO, Coach
19.04. 16-20	Quality award & party	team, Coach

Sprints

- S1: Develop a prototype that offers the most basic functionality via a CLI Done
- S2: Develop a stable version for end user testing Done
- S3: Finish asynchronous system functionality and create a GUI mockup Done
- S4: Publish Neronet as an open source project Done
- S5: Finish Neronet 1.0 Done
- S6: Wrap things up Done

There were few stories that were deferred or completely cancelled, but no unpleasant surprises.

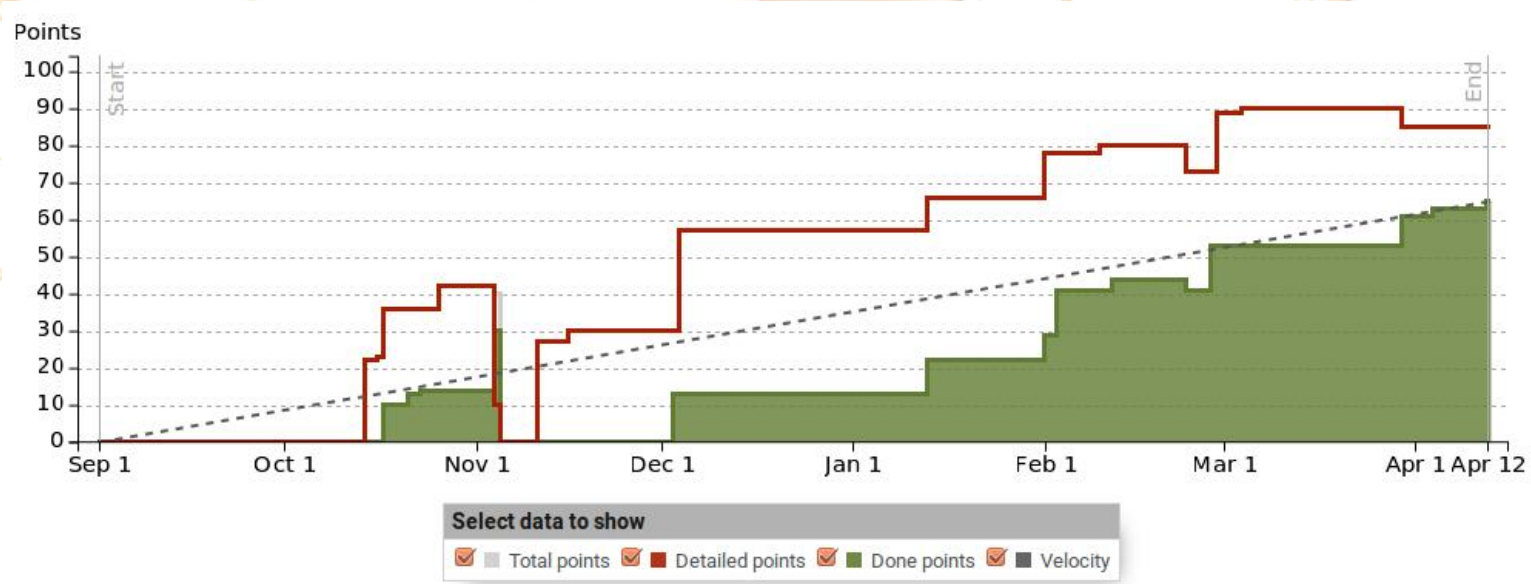
Outline

Effort

Spent and budgeted effort in hours by team member and sprint:

S	Samuel	Teemu	Tuomo	Joona	Iiro	Matias
0	140/50	36/35	45/35	40/35	36/35	43/35
1	52/30	37/33	42/33	46/33	32/33	37/33
2	42/30	27/33	41/33	25/33	27/33	30/33
3	24/15	28/33	14/33	23/33	20/33	27/33
4	37/15	35/33	28/33	32/33	25/33	32/33
5	18/20	31/40	49/40	16/40	39/40	29/40
6	17/15	9/18	10/18	16/18	16/18	10/18
	330/175	203/225	229/225	198/225	195/225	208/225

Burnup


Points
100
90
80
70
60
50
40
30
20
10
0
Start
End
Sep 1
Oct 1
Nov 1
Dec 1
Jan 1
Feb 1
Mar 1
Apr 1
Apr 12
Select data to show
Total points
Detailed points
Done points
Velocity

Outline

Future

The goal was to develop a tool with a bright future.

- Extensive documentation, manual and start guide
- Published as an open source project
- Located in Github and installable via `pip`

Complicating factors

Complicating factors

- PO mostly in London
- Little additional CSE expertise available (only the Coach)
- Very difficult domain
- Inexperienced team
- Only five developers
- The PO nor the scrum master had participated in the course before

Simplifying factors

Simplifying factors

- Our PO was straightforward, very motivated and gave great feedback
- Our coach also gave us valuable support and feedback
- We had high motivation

Additional points

Additional points

- Sprint team leader system
- User guide/manuals enhanced communication
- Early end user testing
- Open source future focus

Summary

Summary:

- The process was smooth
- All sprints were successful
- More than fulfilled the initial vision
- Surpassed PO's expectations

Thank you!

We learned and had fun together :)