User-centric Composable Services: A New Generation of Personal Data Analytics

Jianxin Zhao * Richard Mortier Jon Crowcroft Liang Wang

University of Cambridge, UK

first.last@cl.cam.ac.uk

Nowadays Machine Learning (ML) techniques such as Deep Neural Network (DNN) are used in numerous services. However, there is a big gap between the current ML systems and users' requirements. On one hand, most existing machine learning frameworks, such as TensorFlow and Caffe, focus mainly on the "training" phase. They aim at accelerating the training speed, enhancing performance on GPU, or improving prediction accuracy. On the other hand however, the users, either individuals who want to use the ML-based services or researchers who do not fully commit to the ML field, care less about those benchmarks, but rather about issues such as expressiveness of the tool for constructing a neural network, fast development of new algorithms or neurons on existing systems, access to ML models on local devices, service response time, etc.

Most current end-side services, such as personal intelligent assistants and smart home service, either only support simple ML models or require users to upload raw data (speech, image, etc.) to complex data analytics services host on the cloud. The latter practice is known to associate with issues such as communication cost, latency, and personal data privacy. Some systems begin to focus on mobile platforms, such as Facebook's `Caffe2go`, but they still place emphasis on shifting what existing computing platform can do from data centre to mobile devices, and have not provided systematic solutions to address the aforementioned issues.

In this poster, we present overall design of Owl system, its advantages over other learning platforms, and propose a "Zoo" module built on Owl to mitigate this gap. Owl [1] is an open-source numerical computing system in OCaml language. Owl provides a full stack support for numerical methods, scientific computing, and advanced data analytics on OCaml. Built on the core data structure of matrix and n-dimensional array, Owl supports a comprehensive set of classic analytics such as math functions, statistics, linear algebra, as well as advanced analytics techniques, namely optimisation, algorithmic differentiation, and regression. On top of them, Owl provides Neural Network (NN) and Natural Language Processing (NLP) modules.

Owl system can be extended towards two directions. First, it can use the parallel and distributed engine at lower level to support distributed numerical computing and data analytics. It supports different protocols and multiple barrier control techniques[2]. Second, based on the ML modules, Owl can connect to Zoo, a module that support *Composable Services*. Its basic idea is users should not have to construct new ML services every time new application requirements arise. In fact, many services can be composed from basic ML services: image recognition, speech-to-text, recommendation, etc. The Zoo module aims at providing user-centric, ML-based services, enabling service pulling, sharing, compatibility checking, and composing on local devices.

The most exciting feature of Owl is its expressiveness. We have constructed InceptionV3, one of the most complex network architecture in existing image recognition models, with only 150 LoC, while constructing the same model requires 400 LoC using TensorFlow code. Besides enabling shorter and more compact code, another of its advantages compared with existing popular learning platforms is its flexibility to add new features. As an example, we insert instrumentation code into Owl to collect the computing latency of each node in a neural network when doing inference. Adding this feature only takes 50 LoC. Our initial experiment shows an acceptable performance tradeoff, which is only about 2 times slower than state-of-the-art TensorFlow and Caffe2.

In summary, based on Owl system, we are building the Zoo system, seeking to mitigate the current gap between current ML computing systems and users' requirements. We believe this area of research is only just beginning to gain momentum.

Acknowledgments This work is funded in part by the EPSRC Databox project (EP/N028260/1), NaaS (EP/K031724/2) and Contrive (EP/N028422/1).

References

[1] L. Wang. Owl: A general-purpose numerical library in ocaml. *CoRR*, abs/1707.09616, 2017.

[2] L. Wang, B. Catterall, and R. Mortier. Probabilistic Synchronous Parallel. *ArXiv e-prints*, 2017.

* PhD Student and presenter of this poster

SOSP'17 October 29–31, 2017, Shanghai, China

ISBN [to be supplied]. . . $15.00

User-centric Composable Services: A New Generation of Personal Data Analytics

Jianxin Zhao, Richard Mortier, Jon Crowcroft, Liang Wang

first.last@cl.cam.ac.uk

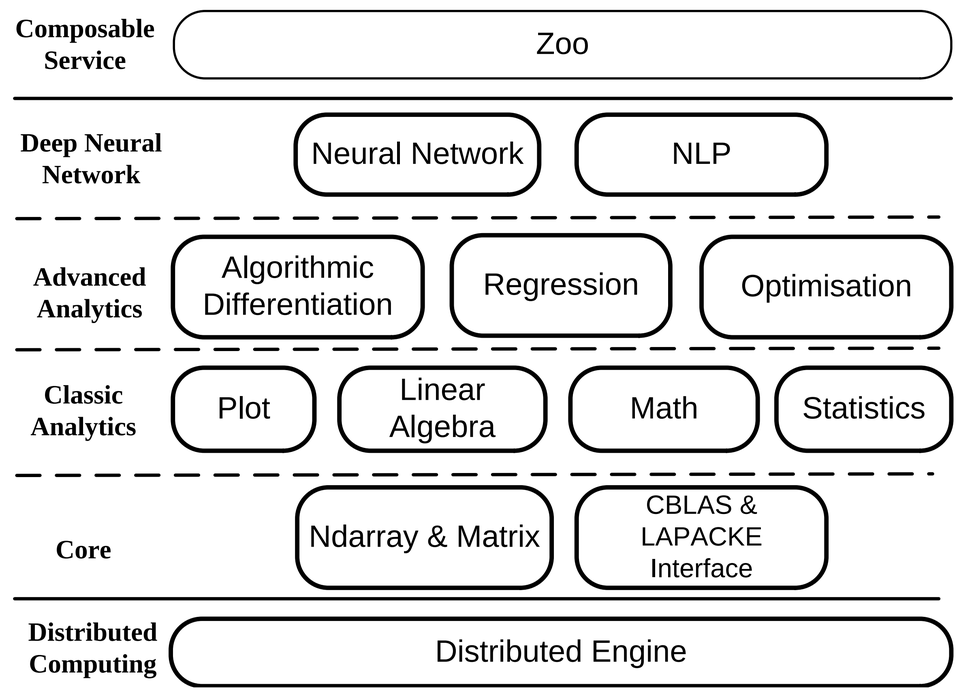
Problem

- Machine Learning (ML) techniques, such as Neural Network (NN), are widely used in today's applications. However, there is still a big gap between the current ML systems and user's requirements:
  - The focuses of TensorFlow *etc.*: How to improve training speed? How to push prediction accuracy one percentage forward? How to enhance performance on GPU?...
  - By contrast, the focuses of users: How to use an ML service to finish my task quickly? How to get faster response? How to construct new services even if I'm not an expert in ML?...
- As a consequence of this gap, most existing ML services require users to upload raw data (speech, images, videos, *etc.*) to complex analytics services hosted in the cloud. This practice is known to associate with issues such as communication cost, latency, and personal data privacy.

Proposed Solution: Zoo

- **Basic Idea**: users should not have to construct new ML services every time new application requirements arise; many services can actually be composed from basic ML services.
- Enables pulling, pushing, running, and composing different ML services on user devices.
- Provides management functions to check model consistency of version, naming, and branching.
- Local on-device inference and model transmission through Virtual Private Network ensures personal data privacy.
- Built on Owl system.

System Architecture

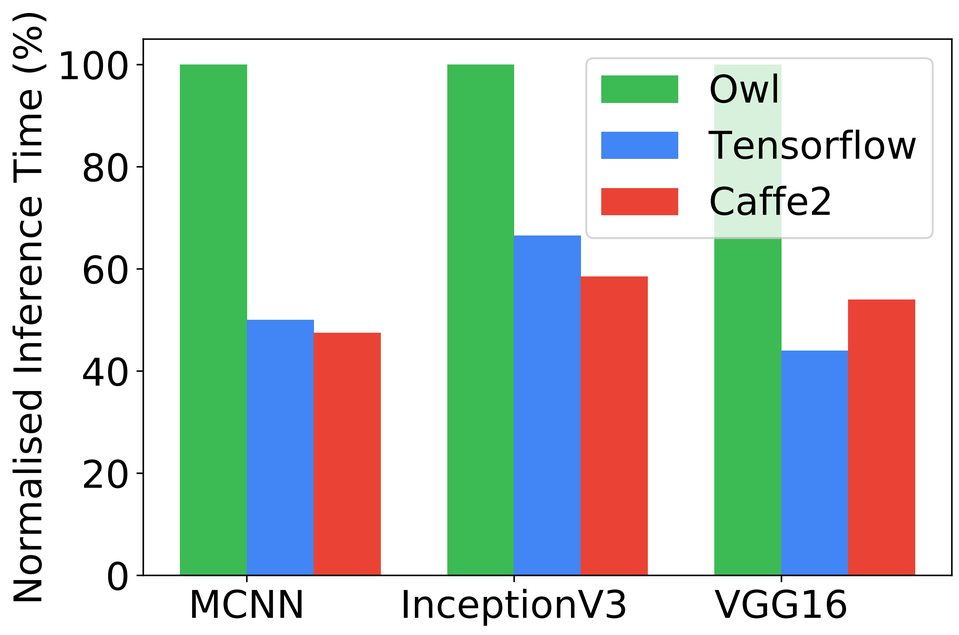


Why NN on Owl?

Owl is an open-source numerical computing system in OCaml, which provides a full stack support for numerical methods, scientific computing, and data analytics. Initial experiment proves its outstanding expressiveness and acceptable performance tradeoff.

- We construct the InceptionV3 model, one of the most complex network architectures in existing image recognition models, with only **150 LoC**, while constructing this same model requires 400 LoC using the up-to-date TensorFlow code, after omitting the comments in code.
- Adding instrumentation function to collect the computing latency of each node in a neural network only takes **50 LoC**.
- Performance tradeoff: only about 2 times slower than state-of-the-art TensorFlow and Caffe2.

Performance



Conclusion

Owl is a well-designed numerical computing system that supports high level data analytics techniques and fast prototyping. We propose to build the Zoo module on Owl to mitigate the gap between current ML systems and users requirements.

Acknowledgements

This work is funded in part by the EPSRC NaaS (EP/K031724/2), Databox project (EP/N028260/1), and Contrive (EP/N028422/1).

References

[1] Liang Wang. Owl: A general-purpose numerical library in ocaml. *CoRR*, abs/1707.09616, 2017.

[2] L. Wang, B. Catterall, and R. Mortier. Probabilistic Synchronous Parallel. *ArXiv e-prints*, 2017.

UNIVERSITY OF CAMBRIDGE

DATABOX

01000100 01100001 01110100 01100001 01100010 01111000