



HAL
open science

Neural Network Precision Tuning Using Stochastic Arithmetic

Quentin Ferro, Stef Graillat, Thibault Hilaire, Fabienne Jézéquel, Basile Lewandowski

► **To cite this version:**

Quentin Ferro, Stef Graillat, Thibault Hilaire, Fabienne Jézéquel, Basile Lewandowski. Neural Network Precision Tuning Using Stochastic Arithmetic. Sparse Days conference, Jun 2022, Saint-Girons, France. hal-03780505

HAL Id: hal-03780505

<https://hal.science/hal-03780505>

Submitted on 19 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Contributions

- Methodology for tuning the precision of an already trained neural network using stochastic arithmetic.
- Goal: obtain the lowest precision for each of its parameters, while keeping a certain accuracy on its results.

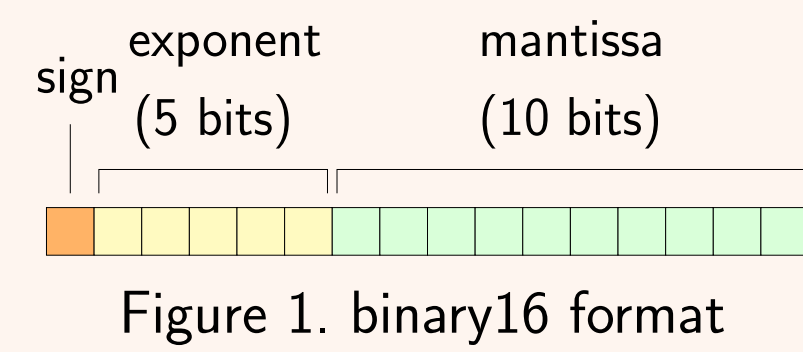
Floating-Point Arithmetic

IEEE754 Standard types

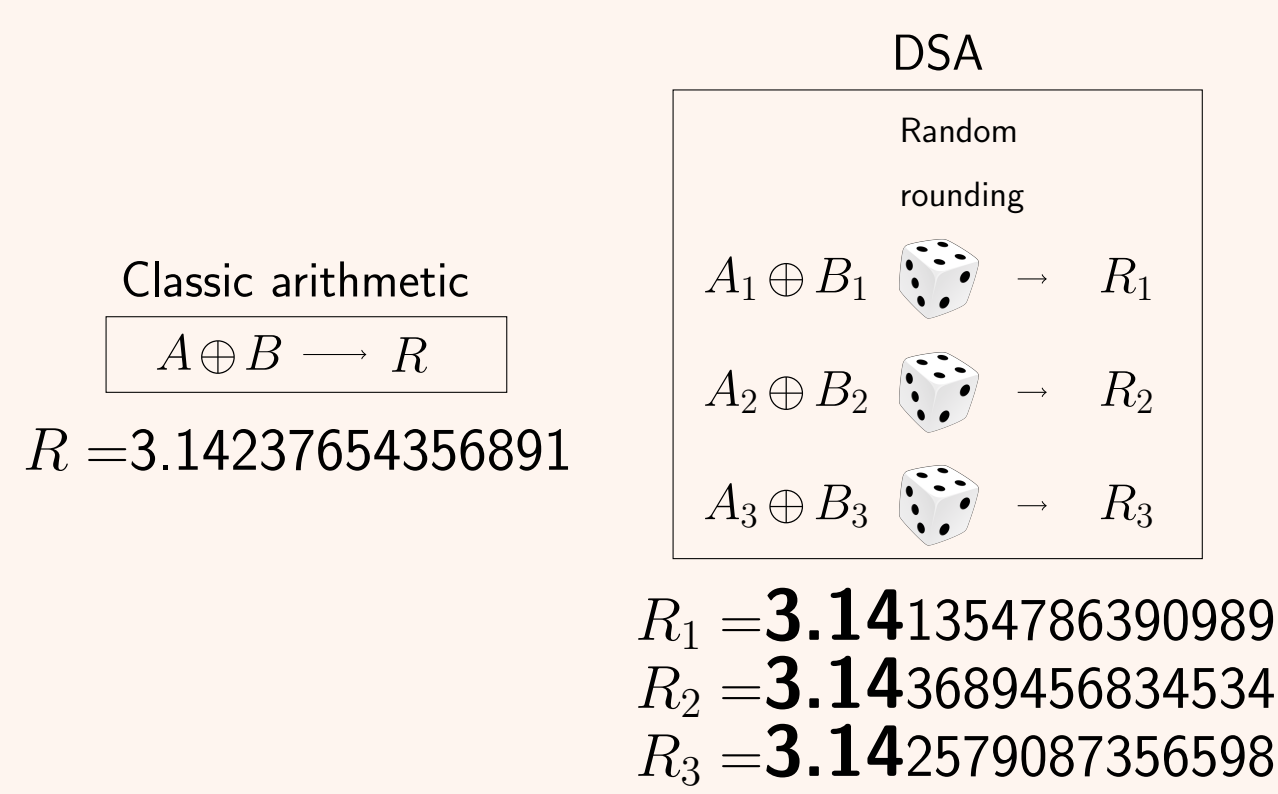
Format	Name	Length	Sign	Mantissa Length	Exponent Length
binary16	Half	16 bits	1 bit	11 bits	5 bits
binary32	Single	32 bits	1 bit	24 bits	8 bits
binary64	Double	64 bits	1 bit	53 bits	11 bits

Reduced precision:

- Shorter execution time ☺
- Less volume of results exchanged (less memory used) ☺
- Less energy consumption ☺
- Less accurate results - rounding errors ☺



Discrete Stochastic Arithmetic (DSA)



- Allows round-off error estimation
- Based on random rounding
- For each operation, computes 3 samples of a result R
- Number of correct digits estimated thanks to Student's test with confidence level 95%

Numerical Validation Tools

CADNA Software (cadna.lip6.fr)

- Implements DSA for C/C++ or Fortran codes
- Provides stochastic types: 3 values of a variable + 1 integer being the estimated accuracy
- Displays values with their exact number of correct digits



PROMISE (promise.lip6.fr)

- Auto-tunes a C/C++ code to provide a mixed-precision version satisfying a given accuracy
- Uses CADNA to validate a configuration
- Uses the Delta-Debug algorithm to test the different configurations with mean complexity $O(n \log(n))$ for n variables [Zeller, 2019]

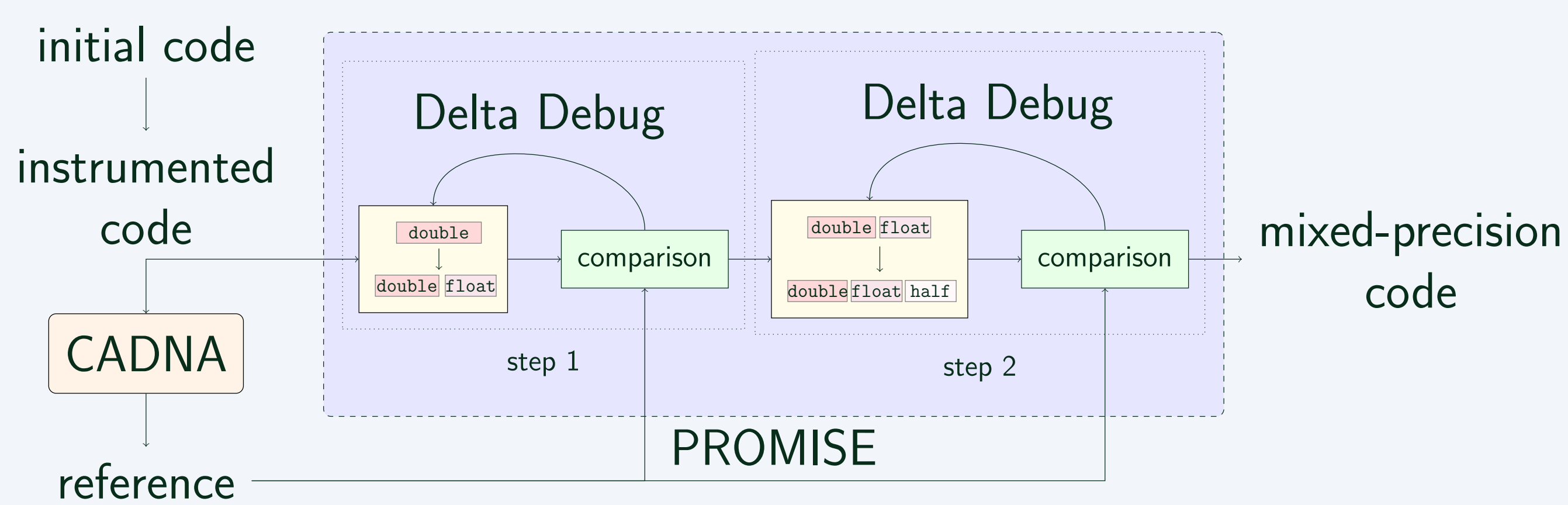
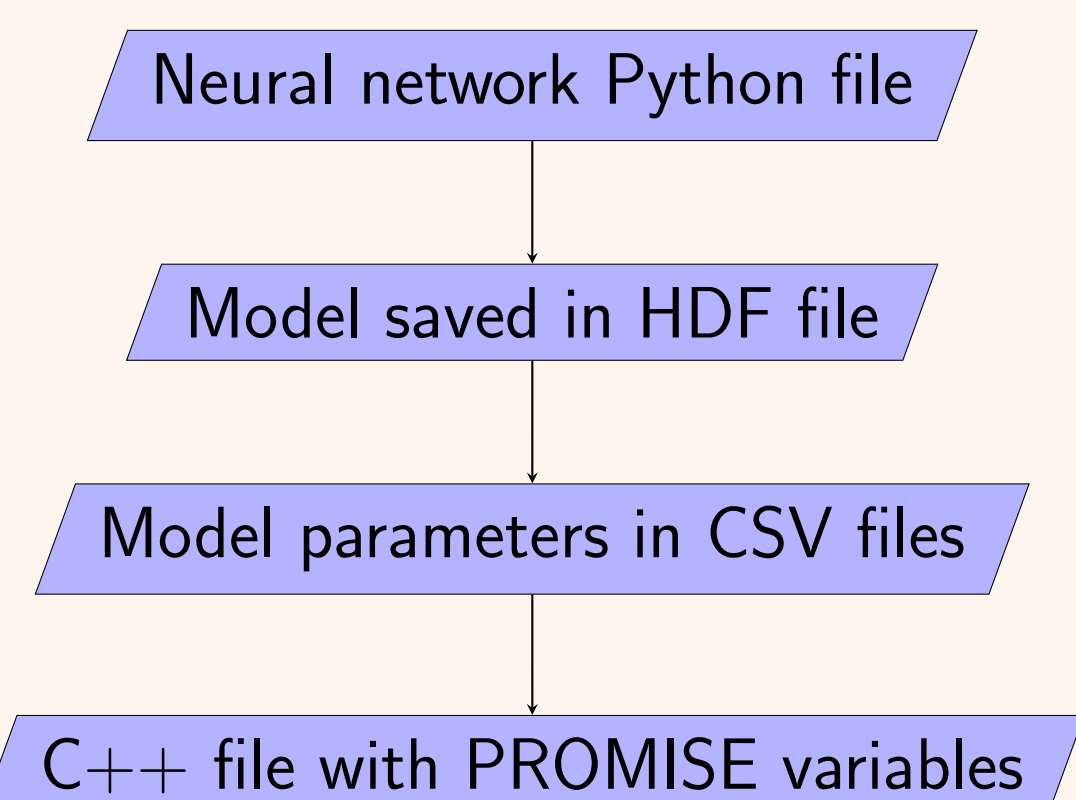


Figure 2. PROMISE Dataflow

Methodology

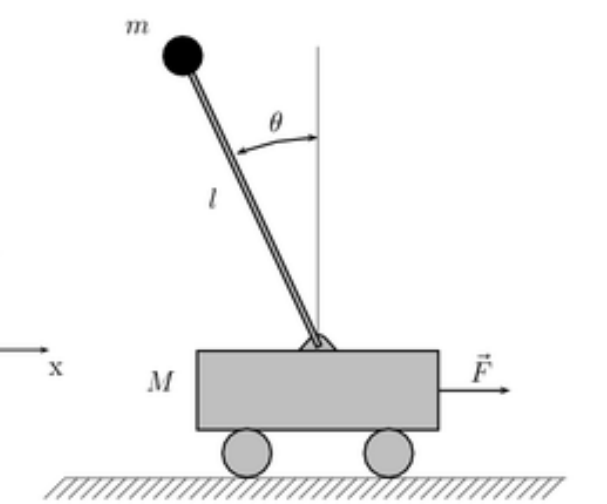
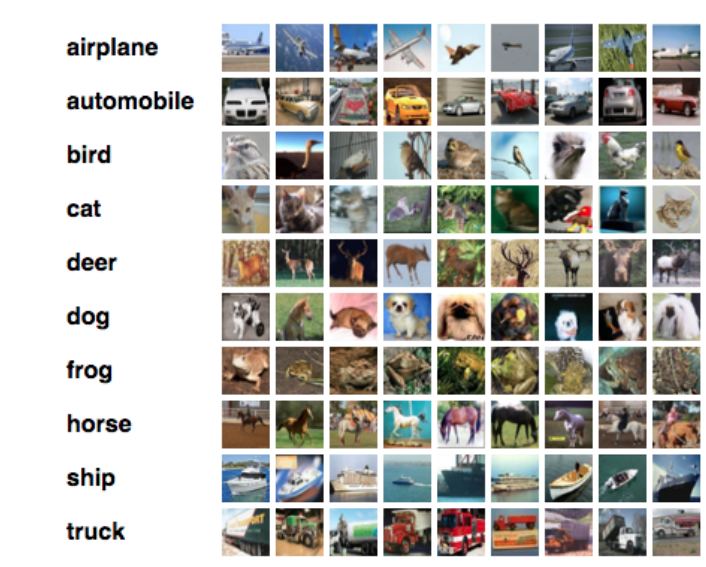
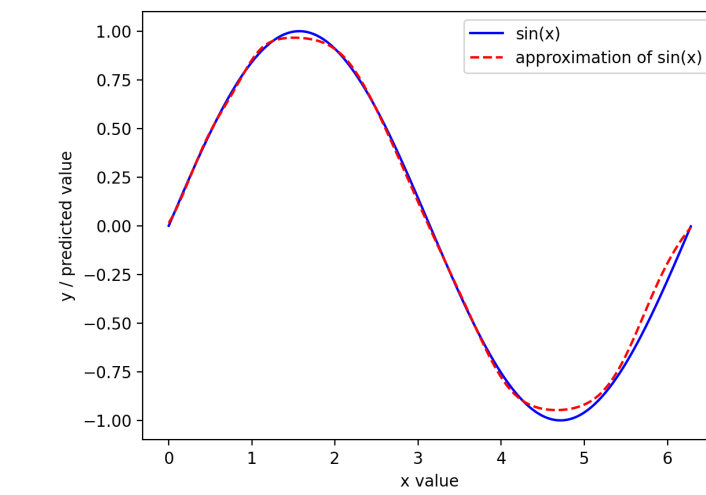


Application of PROMISE with two approaches:

- Considering one type per neuron (weight vector and bias of one neuron have the same precision).
- Considering one type per layer (weights and bias of one layer have the same precision).

Considered Neural Networks

- Sine NN: dense neural network, approximation of sine function
- MNIST NN: dense neural network, classification of handwritten digits
- CIFAR NN: convolutional neural network, classification of pictures among 10 classes
- Inverted Pendulum: dense neural network, computation of a Lyapunov function [Chang et al. 2020]



Precision Auto-tuning Applied to CIFAR NN

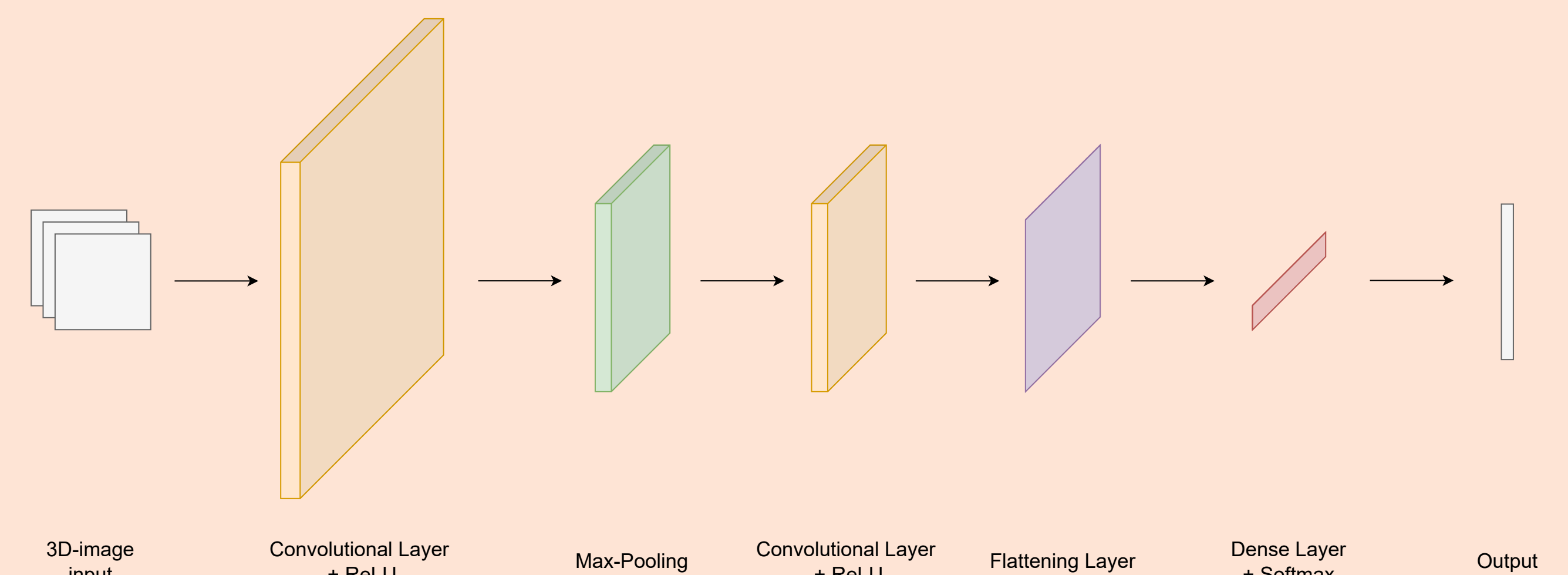


Figure 3. CIFAR NN architecture

Type distribution and PROMISE execution time:

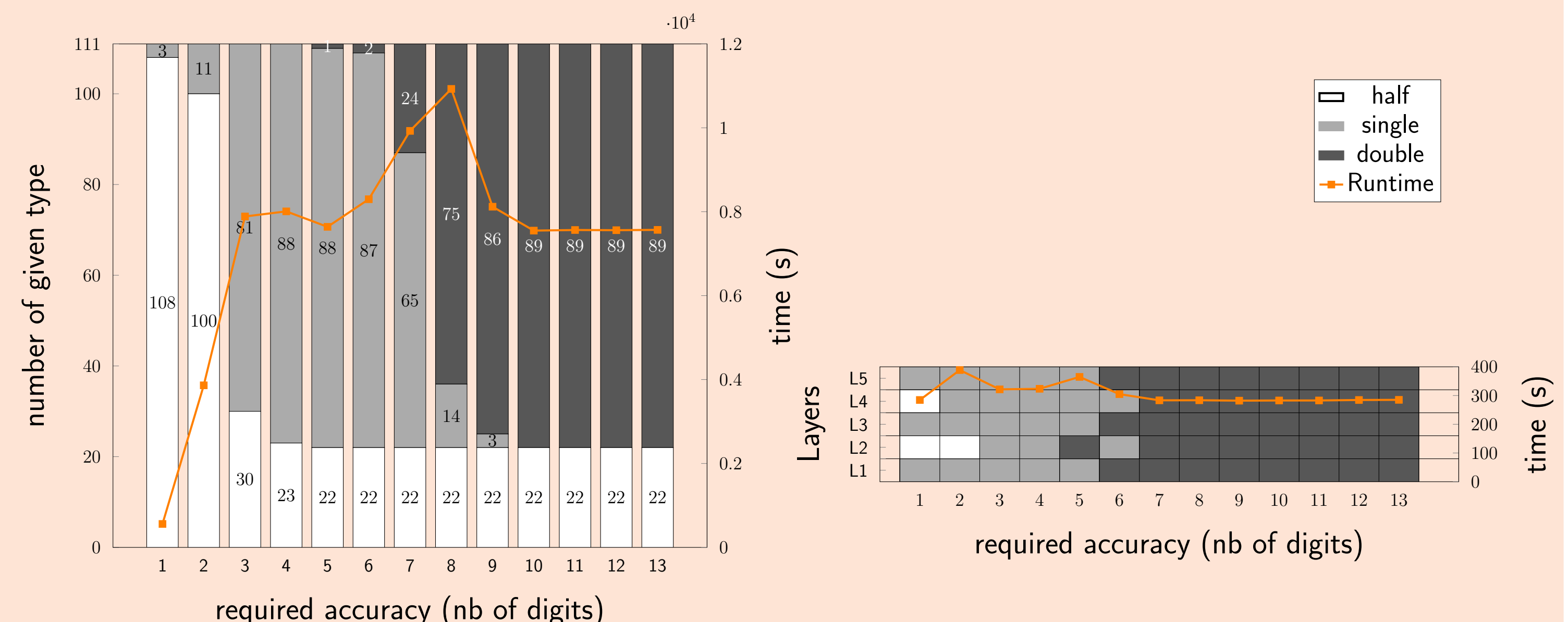


Figure 4. Type distribution per neuron (left) and per layer (right) with image test data [386]

- Mixed precision programs taking into account the required accuracy.
- One type per layer: PROMISE execution time is reduced, but often leads to uniform precision programs.
- Input values have actually a low impact on the type configurations obtained.

Future Works

- Analyse actual gain in time and memory
- Consider the parallelization of the Delta-Debug Algorithm and PROMISE
- Extend PROMISE to GPUs and to arbitrary precision on FPGAs
- Extend PROMISE to other types such as bfloat16

Related works

Precision tuning tools with static approach ([Chiang et al. 2017], etc.) or dynamic approach ([Lam et al. 2013], etc.).

Auto-tuning of neural networks in [Ioualalen and Martel 2019], with a different approach, not using stochastic arithmetic.

References

- Chang, Y.-C. et al. (Dec. 2020). "Neural Lyapunov Control". In: *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.
- Ioualalen, A. and M. Martel (2019). "Neural Network Precision Tuning". In: *Quantitative Evaluation of Systems*. Ed. by D. Parker and V. Wolf. Cham: Springer International Publishing.
- Chiang, W.-F. et al. (2017). "Rigorous Floating-Point Mixed-Precision Tuning". In: *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*. POPL 2017. Paris, France: ACM.
- Lam, M. O. et al. (2013). "Automatically Adapting Programs for Mixed-Precision Floating-Point Computation". In: *Proceedings of the 27th International ACM Conference on International Conference on Supercomputing*. ICS '13. Eugene, Oregon, USA: ACM.

Further information:

