Neuromorphic Hardware As Database Co-Processors

Thomas Heinis

¶*Imperial College, London, United Kingdom*

1. MOTIVATION

Today's databases excel at processing data using fairly simple operators but are not efficient at executing operators which include pattern matching, speech recognition or other cognitive tasks. The only way to use such operators in data processing today is to simulate spiking neural networks.

Neuromorphic hardware is supposed to become ubiquitous in complementing traditional computational infrastructure. With it, spiking neural networks can be simulated time- and energy-efficient. Given the impact hardware acceleration had on query execution in the past, the question is how neuromorphic devices can be used to a similar effect.

2. HARDWARE FOR SPIKING NETWORKS

Spiking neural networks (SNNs) are the most recent type of neural networks thought to be the most bio-realistic. More important than bio-realism, however, is that they can simulate bigger networks with the same computational power, thus enabling them to perform human cognition tasks more efficiently. Additionally, SNNs also have the benefit of learning faster than traditional, backpropagation based methods.

While traditional neural networks can efficiently be simulated on conventional hardware (CPUs or GPUs), SNNs cannot. First, to simulate SNNs, a massive number of tiny messages (i.e., spikes) is exchanged between large numbers of neurons. Today's hardware lacks efficient support for small messages (header information alone is a multiple of the spike information) and for multicast protocols (to send spikes to several neurons). Second, while in the brain neurons are only active when receiving/processing spikes, traditional CPUs are always on rendering the simulation energy inefficient.

3. APPLICATION TO DATABASES

There are several application domains related to databases and data processing where SNNs are used successfully.

SNNs can in general be used for classification [4] but prove very useful for recognising audio, video or images where they outperform non-spiking methods [2]. The state of the art for these applications, for example, uses pixel counts leading to a gap between statistics and semantics of images. SNNs bridge this gap and content-based image, audio or video retrieval benefits from the use of neuromorphic hardware.

SNNs trained with maps of the environment can answer path planning queries [3] and thus also nearest neighbor queries (without using hierarchical structures which typically leads to performance problems as with the R-Tree).

A further challenging application is performance prediction based on learning. Learning has been successfully used to predict execution time of queries [1]. A SNN trained using query plans and their execution times can be run on neuromorphic hardware and be queried for an approximate, time- and energy-efficient answer for the best query plan.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

4. OPPORTUNITY & CHALLENGES

While small SNNs can easily be simulated on traditional hardware, bigger, more sophisticated networks so far cannot be executed time- and energy-efficient. Neuromorphic hardware, however, for the first time enables to do so and to make the execution of SSNs part of data processing.

Doing so, however, incurs multiple questions, partly because of the prototypic hardware but also more fundamental challenges. For example, transferring data to neuromorphic hardware is slow and it may be quicker to run small simulations on the CPU instead. A crucial challenge thus is to develop a cost model featuring the time to transfer data to and from the device, execution time as well as energy consumption. The mode is essential to plan on where to run the SNN.

Further, both, the data in the database as well as the query need to be encoded as temporal spikes. A difficult question is how to execute queries on multiple attributes, e.g., a query to find a pattern in a particular area. The challenge is how areas on an encoded image can be restricted.

Additional challenges include combining several operators in one SNN, indexing spikes (i.e., the encoded information), the uncertainty (or imprecision of results) and others.

5. REFERENCES

[1] A. Akdere et al. Learning-based Query Performance Modeling and Prediction. In *ICDE*, 2012.

[2] S. M. Bohte et al. Unsupervised Clustering with Spiking Neurons by Sparse Temporal Coding and Multilayer RBF Networks. *Transactions on Neural Networks.*

[3] J. J. Hopfield. Neurodynamics of Mental Exploration. *Proceedings of the National Academy of Sciences*, 2010.

[4] M. Schmuker et al. A Neuromorphic Network for Generic Multivariate Data Classification. *PNAS*, 2014.