Dataflow Matrix Machines: a White Paper

Michael A. Bukatin

November 30, 2019

Dataflow matrix machines form a **novel class of neural machines** with remarkable properties. They combine **general-purpose programming powers of stream-oriented architectures** such as traditional dataflow programming and more novel functional reactive programming with **good machine learning properties of conventional neural networks.**

---

1 Future potential

The future potential and promise of dataflow matrix machines (DMMs) is high.

- **Program synthesis.** The task of synthesis of dataflow matrix machines should be more tractable than conventional program synthesis. When one works with DMMs, the task of *learning program sketches* is reformulated as *neural architecture search*, and converting a program sketch to a full program should be done by conventional methods of neural net training. The dimension of the network and the dimension of data are decoupled, so one has an option to synthesize compact neural machines under this approach, if so desired.

  Dataflow matrix machines combine

  – aspects of *program synthesis* setup
    (compact, human-readable programs);

  – aspects of *program inference* setup
    (continuous models defined by matrices).

- **Self-modification and learning to learn.** Using neural networks for metalearning is always non-trivial. In particular, dimension mismatch, namely the number of neuron outputs being much smaller than the number of network weights, means that a neural network can only modify itself in a highly constrained manner. Dataflow matrix machines address this problem and have **powerful and flexible self-modification facilities**.

  Therefore, a dataflow matrix machine can be equipped with a variety of primitives performing self-modifications, and it can fruitfully learn various linear combinations and compositions involving those primitives.

  Self-modification facilities of dataflow matrix machines are not limited to the weight changes for the existing connections in the network. The available primitives allow to modify the network topology as well. For example, primitives allowing the network to control its own fractal-like growth by the means of cloning its own subnetworks are available.

  So, this is a very promising architecture not only for methods of learning to learn better in a traditional sense, but also for methods of learning to perform neural architecture search better.

  A dataflow matrix machine can comfortably host an evolving population of other DMMs inside itself, so it is an excellent environment for neuroevolution experiments and, in particular, for the experiments aiming to learn to evolve better (or to evolve to evolve better).

- **Neuro-symbolic architectures and hierarchical learning.** The availability of **flexible tensors** based on *tree-shaped indices* makes this architecture quite promising for neuro-symbolic methods and for hierarchies.

- **Visual animation based on composition of unit generators.** DMMs generalize digital audio synthesis via composition of unit generators, but work with more general streams than just streams of numbers. The old-fashioned analog video synthesizers also work in the style of composition of unit generators. Modern computer graphics tends to be imperative, oriented towards specific data flows implemented inside GPUs, and to require the software practitioners to focus on manual optimizations. The promise of doing animations via functional reactive programming is to focus again on semantically meaningful data flows, and to leave the hardware-oriented optimization to the underlying system. Dataflow matrix machines aspire to make this promise a reality.

2 How they work

The essence of neural model of computations is that linear and non-linear computations are interleaved. Hence, the natural degree of generality for neuromorphic computations is to work not with streams of numbers, but with arbitrary streams supporting the notion of linear combination of several streams (**linear streams**).

Dataflow matrix machines (DMMs) form a novel class of neural machines, which work with wide variety of **linear streams** instead of streams of numbers. The neurons have arbitrary arity (arity of a neuron can be fixed or variable). Of particular note are self-referential facilities: ability to change weights, topology, and the size of the active part of the network dynamically, on the fly, and the reflection capability (the ability of the network to analyze its current configuration).

There are various kinds of linear streams. They include streams of numbers, sparse vectors and sparse tensors (both of finite and infinite dimension), streams of functions and distributions. We found streams of V-values (**flexible tensors** based on tree-shaped indices) to be of particular use.

A single dataflow matrix machine can process a large variety of different kinds of linear streams, or it can be based on a single kind of linear streams, sufficiently expressive for a given class of situations.

3 What have been done

During the last seven years, I have been leading the efforts which resulted in the discovery and study of dataflow matrix machines. The work was done by several research collaborations between members of American and British academic communities. We published three papers and a number of preprints and research notes, and released several related open source research-grade implementations. During this period, I gave about 20 presentations on related topics at various venues in the United States, Canada, and Europe.

We investigated a variety of programming patterns and primitives in dataflow matrix machines and performed experiments with various uses of self-referential facilities.

We are continuing our efforts towards the next generation of DMMs and towards future programming and machine learning technologies based on DMMs.

Dataflow Matrix Machines links:

GitHub Pages: `https://anhinga.github.io`

Open source implementation (Clojure): `https://github.com/jsa-aerial/DMM`

One-page overview: `https://www.cs.brandeis.edu/∼bukatin/dataflow-matrix-machines-2016.pdf`

Reference slide deck: `https://researcher.watson.ibm.com/researcher/files/us-lmandel/aisys18-bukatin.pdf`

Reference paper: `https://arxiv.org/abs/1712.07447`