Towards Mapping Lift To Deep Neural Networks

1. Context
   - GEMM is ubiquitous in Deep Neural Networks (DNNs).
   - It is the basis of both stencil and im2col convolution methods.
   - Hardware accelerators use N-dimensional computational units.
   - These units are exposed in ISAs via coarse-grained operators:
     - VVAdd32, VVAdd64, MVAdd64, MVAdd128
     - VVMul64, VVMul128, MVMul164, MVMul128

2. The problem
   - How can we combine device-specific operators optimally?
   - How can we make the optimisations performance portable?
   - How can we automate and abstract the process from the user?

3. The Lift approach
   3.1 Concept
      1. Separate algorithm (WHAT) from implementation (HOW)
      2. Detect and rewrite patterns
   3.2 Functional data-parallel IR Language
      - Data types
        - Int, Arrays
        - Float8 / Float16 / Float32
      - Algorithmic patterns
        - Map, Slide, Reduce, Zip, Join, Split
   3.3 Rewrite rules
      - Generic and customisable
      - 3 levels: DSL, algorithmic, hardware
      - Extensible

4. Example rewriting
   A fully connected layer

5. Preliminary results
   - Functional correctness on the BrainWave accelerator
   - Performance measurements on Mali GPU
     - Average runtime of conv layers in VGG
       - OpenBLAS: 3.99 (msec)
       - clBLAS: 1.96 (msec)
       - Lift: 1.00 (msec)
       - Lift: 0.71 (msec)