# How Can We Be So Slow?
# Realizing The Performance Benefits of Sparse Networks

Numenta

**Lawrence Spracklen, Kevin Hunter & Subutai Ahmad**

[lspracklen,khunter,sahmad]@numenta.com

## Sparse Networks Should be Fast Networks

- Unfortunately, performance on CPUs and GPUs is lackluster
  - Speedups of less than 3X are typical
  - Weight and activation sparsity not simultaneously exploited

## Sparse Networks can be Fast Networks

- Hardware friendly fine-grain sparsity patterns exist
- Demonstrated **100X+** speedup from sparsity on FPGAs
  - Developing library for generalized build-your-own sparse networks
- Achieved **linear speedup with sparsity** on CPUs for MatMul primitives
  - High-performance transformer under development

## The Sparsity Problem?

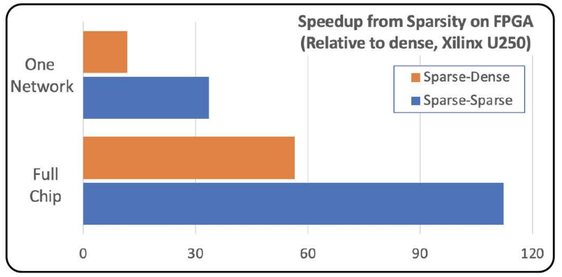
- Hardware thrives on dense regular structure and predictable data access patterns
  - Enables effective use of data prefetchers and wide SIMD engines
- Selectively handling 'randomly' positioned non-zero elements is inefficient
  - Often faster to just execute the zero-valued multiplications

## Are Hardware-Friendly Sparsity Patterns Feasible?

- Block sparsity is a commonly imposed constraint
  - Large blocks reduce accuracy
  - Still runtime uncertainty in memory access patterns, impacting performance
- Developing accommodating hardware is complex, expensive and slow
  - Need to develop sparsity patterns that are compatible with today's processors

## 112X Speedup on FPGAs

- Sparse-weights, dense-activations outperforms dense by over **50X**
- Also leveraging activation sparsity doubles performance to **112X**
- FPGA implementation derives performance from
  1) Faster throughput per network
  2) Reduced resources per network allows multiple network placement




- Decreased resource utilization allows placement on small FPGAs
  - Extreme power efficiency for AI at the Edge

Sparse network energy efficiency [Xilinx U250@225W]

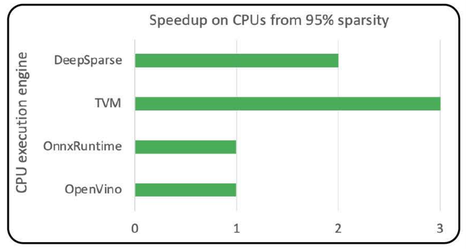
| Network type | # of networks | Inferences/sec/ watt | Relative efficiency |
|---|---|---|---|
| Dense | 4 | 54 | 100% |
| Sparse-Dense | 24 | 3065 | 5675% |
| Sparse-Sparse | 20 | 6088 | **11274%** |

## An Incredible Opportunity for Performance

- Weight sparsity can reduce non-zero weights by 5-20X
- Activation sparsity can reduce activations by 5-10X
- Simultaneously exploiting both has a multiplicative effect
  - Up to **200X** reduction in non-zero computations

## The Current Reality

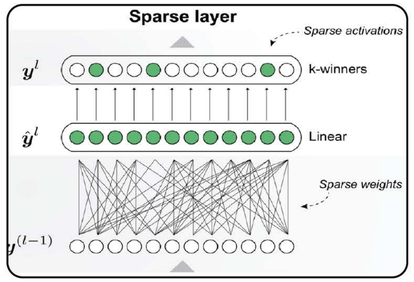
- Sparse networks on CPUs and GPUs often less than 3X faster
  - Weight and activation sparsity not exploited simultaneously




## Experimental setup

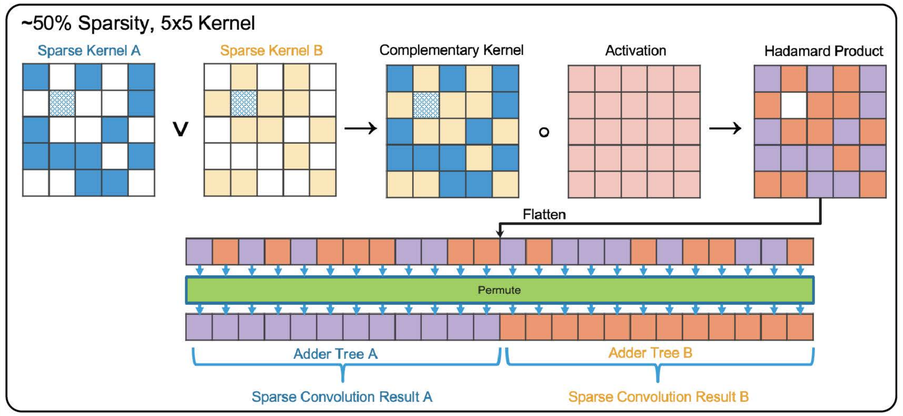
- CNN for speech recognition
  - Two CNN layers and two FC layers
  - 2.5M parameters

1) Sparse-dense version
   - 95% weight sparsity (varies by layer)
2) Sparse-sparse version
   - Also leverage 88% activation sparsity
   - K-winners selection
- Sparse network accuracy not compromised




## Hardware Friendly Fine-Grain Sparsity

- Leverage sparsity patterns that allow computation to be framed as a dense operation
  - Applicable to both linear and convolutional layers
  - Light-weight constraint doesn't impact achievable accuracy
- Enforce non-overlapping patterns across multiple sets of kernels or linear-layer weights
  - Allows multiple kernels/weights to be elegantly combined into a single dense entity
  - Speedup scales linearly with degree of sparsity
- Use sparse static mask training to provide control over weight placement
  - Accurate networks while exploiting extreme sparsity




- Flexibility of HW architecture dictates severity of sparsity pattern constraints
  - Depends on the cost and flexibility of supported permute operations
- The reconfigurability of FPGAs makes them an ideal architecture for sparse networks
  - Simultaneously use weight and activation sparsity with minimal constraints

## Speedup on CPUs [Early results]

- Compared 95% weight sparse MatMul with Intel's MKL
  - Outperforms Intel dense by **18X** and Intel 's best sparse by **3X**
  - Doesn't rely on large batch sizes to derive performance

## Next Steps

**Generalize FPGA support**
- Creating general-purpose library for high-performance sparse-sparse networks on FPGAs

**Deepen CPU & GPU support**
- Efficiently support sparse-sparse networks
- Demonstrate potential with fast sparse transformer

**Accelerate Sparse network training**

## Further Reading

1) Can CPUs leverage sparsity? https://tinyurl.com/sparsednncpu
2) 100X FPGA Whitepaper https://tinyurl.com/fastsparsefgpa
3) How can we be so dense? https://tinyurl.com/whysodense