Optimizing the Consumption of Spiking Neural Networks with Activity Regularization

Simon Narduzzi$^{1,2}$  Siavash A. Bigdeli$^1$  Shih-Chii Liu$^2$  L. Andrea Dunbar$^1$

$^1$ CSEM, Neuchâtel, Switzerland  $^2$ Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland
The energy consumption problem

Compute power of common deep learning models

| Type       | Parameters |
|------------|------------|
| CNN        | 1B         |
| Transformer| 10B        |
| BERT Large | 100B       |

Adapted from [https://github.com/amirgholami/ai_and_memory_wall](https://github.com/amirgholami/ai_and_memory_wall)
Edge computing

Advantages

• Rapid decision making
• Efficient pre-processing
• Privacy-preserving applications

MAJOR CHALLENGE: Energy consumption

1.8B by 2026*

*ABI Research, Artificial intelligence and Machine Learning, 2 QTR 2021
## Techniques to reduce consumption

### Software

| Technique               | Description                                      |
|-------------------------|--------------------------------------------------|
| Pruning                 | Weights / neurons                                |
| Quantization            | 8bits, 4bits, …                                   |
| Distillation            | Teacher – student                                 |
| Efficient operators     | Separable convolutions, etc…                     |
| Event-based processing  | Spiking neural networks (SNNs)                   |

### Hardware

| Category                          | Description                                                                 |
|-----------------------------------|-----------------------------------------------------------------------------|
| Semi-conductor process tech       | FinFET, Fully Depleted Silicon-On-Insulator, etc…                            |
| Resource optimization             | Power management, flexible accelerators, etc…                                |
| Specialized units                 | Convolution accelerators, zero-skipping, etc…                                 |
| Event-based processing            | Neuromorphic hardware: Intel Loihi, IBM TrueNorth, SpiNNaker, etc…           |
Artificial vs Spiking Neurons

Artificial Neural Network

\[ z = \sigma \left( \sum_{j=1}^{N} W_{ij} x_j + b_i \right) \]

Information processing in artificial neural networks (ANN)

Spiking Neural Network

\[ z = \sigma_{thr} \left( \sum_{j=1}^{N} W_{ij} x_{t,j} + b_i \right) \]

Information processing in spiking neural networks (SNN)
Metrics

Computation cost for ANN : ‘Effective’ FLOPS

\[ EFLOPS = \sum_{l=1}^{L} \phi(W_l) \times \phi(A_{l-1}) + \phi(B_l) \]

\[ \phi(x) := x \neq 0 \]

Computation cost for SNN: SynOps

\[ SynOps = \sum_{t=1}^{T} \sum_{l=1}^{L} f_{out,l} \times s_l(t) \]

“A million spiking-neuron integrated circuit with a scalable communication network and interface”, Merolla et al, 2014
GOAL: Increase sparsity to reduce the computational cost

IDEA: Exploit the natural sparsity of SNNs

PROBLEM: SNNs training is difficult with common back-propagation
Experimental setup

"Conversion of continuous-valued deep networks to efficient event-driven networks for image classification", Rueckauer et al., 2017
Sparsity

- Sparsity reduces computational cost
- Pruning of weights or activation maps

Weight pruning

Neuron (activity) pruning
Related work

• Zhao et al., 2021; Pellegrini et al., 2021 – SNN trained from scratch
• Sorbaro et al., 2020 – Optimize SynOps
• Rückauer et al., 2017 – L₁ regularization on weights

• Ours:
  • Lᵖ-regularization and Hoyer,
  • Comparison between ANN and SNN,
  • EFLOPs
Constraint: Regularizers

- Enforce sparsity using regularizers on activity maps

Loss function:

\[ \mathcal{L} = CE + \lambda_{reg} \sum_{l} \psi(X_l) \]
Results on MNIST

Results of the MLP with respect to $\lambda_{\text{reg}}$
Activity regularization effect

**MNIST**

### Computation cost of MLP

- Baseline
- L2
- L1
- L0.5
- L0.01
- H
- HS

|            | Baseline | L2 | L1 | L0.5 | L0.01 | H | HS |
|------------|----------|----|----|------|-------|---|----|
| EFLOPS     | 6.0E+04  | 4.0E+04 | 3.0E+04 | 2.0E+04 | 1.0E+04 | 3.0E+04 | 2.0E+04 |
| SyrOps     | 7.0E+04  | 5.0E+04 | 4.0E+04 | 3.0E+04 | 2.0E+04 | 1.0E+04 | 3.0E+04 |

### Computational cost of LeNet-5

- Baseline
- L2
- L1
- L0.5
- L0.01
- H
- HS

|            | Baseline | L2 | L1 | L0.5 | L0.01 | H | HS |
|------------|----------|----|----|------|-------|---|----|
| EFLOPS     | 2.0E+05  | 1.5E+05 | 1.0E+05 | 5.0E+04 | 3.0E+04 | 2.0E+04 | 1.5E+08 |
| SyrOps     | 3.0E+05  | 2.5E+05 | 2.0E+05 | 1.5E+05 | 1.0E+05 | 5.0E+04 | 3.0E+05 |
Activity regularization effect

CIFAR-10

Computation cost of LeNet-5

Baseline, L2, L1, L0.5, L0.01

EFLOPS, SynOps

-21%, -82%
Conclusion

• Activity regularization of ANNs is a simple way to reduce the number of SynOps in converted SNNs

• Hoyer regularization has limited effect compared to $L_p$-regularization

• SynOps and EFLOPs are not correlated, as a reduction in EFLOPs does not necessary result in a similar reduction in SynOps

• Better approximations of $L_0$ can be found, as $L_{0.01}$ is too aggressive
THANK YOU
