Automated synthesis of mixed-signal ML inference hardware under accuracy constraints

Kishor Kunal, S Ramprashath, Jitesh Poojary, Ramesh Harjani, and Sachin S. Sapatnekar
University of Minnesota, Minneapolis, MN 55455

Abstract—Due to the inherent error-tolerance of machine learning (ML) algorithms, many parts of the inference computation can be performed with adequate accuracy and low power under relatively low precision. Early approaches have used digital approximate computing methods to explore this space. Recent approaches using analog-based operations achieve power-efficient computation at moderate precision. This work proposes a mixed-signal optimization (MISO) approach that optimally blends analog and digital computation for ML inference. Based on accuracy and power models, an integer linear programming formulation is used to optimize design metrics of analog/digital implementations. The efficacy of the method is demonstrated on multiple ML architectures.

I. INTRODUCTION

Machine learning (ML) hardware requires high energy-efficiency, but is primarily built using digital circuits today. For low-to-moderate precision tasks, at iso-precision, analog circuits are much more energy-efficient than their digital counterparts [1], [2]. The selective use of analog computing is thus an excellent fit for ML, where lower precision can be used for less sensitive operations without harming overall accuracy; for sensitive operations that require high precision, digital circuitry may be used. There has been no systematic EDA exploration of this tradeoff space to achieve optimal energy-efficiency.

DNN models are resilient to small computation errors, and this has been widely exploited to optimize digital ML hardware through low-precision fixed-point computations [3], approximate computing [4], and model compression [5]. There has been also work on specialized hardware for energy optimization of these quantized models [6]. Analog approaches often target small networks/datasets [7], [8]; those that address larger networks [9], [10] focus on single-layered analog operations, adding an analog-to-digital/digital-to-analog converter (ADC/DAC) after each simple analog operation. This results in massive ADC/DAC energy overheads. To amortize these costs, approaches such as interleaved bit-partitioning, or a combination of digital and charge-domain accumulation are proposed [9], [10], but data conversion costs remain a major system-level consideration.

We propose MiSO-ML (Mixed Signal Optimization for low-power ML), which builds optimal hardware architectures that bring the best of both worlds, analog and digital, for energy-efficient ML inference. We create energy and noise models for fundamental analog and digital operations, at different precision levels. Using these models, our system-level optimization creates energy-efficient hardware under an accuracy specification. Unlike prior single-layered approaches, we amortize the cost of ADCs/DACs across multiple layers of an ML architecture. This provides significant energy efficiency benefits, reducing data conversion overhead to just 13.2% (as opposed to 52.2% in [11]) while harnessing the efficiency gains of mixed-signal computation. The contributions of MiSO-ML are listed below:

1) We propose a framework for optimizing ML models for low power using mixed-signal computing.
2) We propose a novel hardware-aware mixed-precision quantization using an integer linear programming (ILP) formulation to find an optimal bit-precision setting, and to optimize the ADC overhead by performing analog-to-digital conversions after multiple layers, when possible within noise constraints.
3) We demonstrate substantial power improvement on common ML-architectures, as compared to a digital-only quantization.

Intuitive concept. When ML operations are performed using analog circuits, they inevitably accumulate an analog noise voltage, $V_n$, that acts as an offset to the equivalent digital value. Since the resolution of an ADC is 0.5 LSB (least-significant bit), as long as the analog computation is recovered to digital form while $V_n < 0.5$ LSB, the signal is “restored” to full digital precision, with no accuracy loss [1]. Firstly, our optimization scheme tracks the maximum noise over multiple layers of a DNN, and finds that one can often introduce ADCs after multiple layers of DNN computation. This allows the cost of data conversion to be amortized over multiple layers, unlike [7], [8]. Secondly, we optimize the precision for each layer: typically, 8-bit precision requires an ADC after each layer, but relaxed precision allows greater amortization, and with the right optimization, does not significantly affect accuracy. Our approach is guided by a metric of sensitivity of the output to noise in an operation, driving noise-sensitive computations to be performed in digital or higher precision analog modes, an less sensitive operations using analog circuits at lower precision. Through these ILP-based optimizations, we demonstrate large gains from mixed-signal computing.

Next, in Section II, we propose our building block module, a foundational element for modeling a range of digital/analog operations; Section III delves into the modeling of energy and noise values for various state-of-the-art computation operations; Section IV then models the propagation of noise propagation from digital/analog operations across layers in an ML architecture. These are used to build energy lookup table and noise sensitivity models for ML architectures in Section V, and invoked in our ILP-based approach in Section VI. Finally, Section VII evaluates MiSO-ML across several ML architectures, and Section VIII concludes this article.

II. PROPOSED MIXED-SIGNAL ML HARDWARE SCHEME

To extend analog computations to deeper architectures with a diverse set of operations, hybrid computation is necessary, in which analog processing must be followed by restoration of the analog signal to discrete values to overcome noise accumulation problems. As pointed out above, the high energy overhead of these domain-switching operations (using ADCs/DACs) must be balanced with the amount of noise accumulation over multiple analog stages.

For this optimization, we propose a building block (Fig. 1) to map operations from ML architectures, leveraging multiple stacked analog operations before the conversion to the digital domain. This block takes two input operands (e.g., weight and activation), and generates its output after performing the operation. For each operation, we have three optimization parameters: (1) the operation domain (digital or analog), (2) the operation precision, and (3) the presence/absence of a data converter (ADC or DAC), required when the input signal
There are several sources of noise in analog ratio (SQNR), where quantization noise reflects the loss in accuracy to the signal strength, is quantified as signal-to-quantization noise impact of bitwidth-dependent quantization in digital blocks on error. [6], [14], [15], and are summarized in Section V. We focus here on the Energy models for digital computation blocks are well understood [4], A. Digital computations: quantization models

For example, in an analog sum operation, digital weights and activations from memory are first converted into the analog domain using DAs. Next, the addition is performed using charge sharing [12]. The analog output is sent out to the next stage, where it may be passed through an analog max-pooling operation and an analog sigmoid/tanh operator without any data domain conversion. Multiple such analog stages can be cascaded, with the total noise increasing as more stages are cascaded, until the noise reaches the threshold level of requisite precision. The MiSO-ML unit block includes data conversion (ADC/DAC), which enables the ability to handle individual operations as well as complex operations with specialized circuits, e.g., digital-to-analog MAC operations [9], [13].

Fig. 2 shows a CNN architecture being mapped to hardware by MiSO-ML. The convolution operation is mapped to analog MAC hardware with 4-bit precision, whose results are fed through an analog max-pooling circuit with 4-bit precision before being converted to the digital domain. Next, the fully-connected layer is mapped to digital MAC hardware with 6-bit precision and is then sent to ReLU hardware with 6-bit precision. In this hardware mapping, the input features size is larger thus larger number of operations (21600 MAC + 216 Max-pool) are performed in the analog domain and very few (432 MAC + 2 ReLU) operations are carried out in the digital domain, thus improving the energy cost of this model over a purely digital implementation, even after including the cost of the ADC operations.

III. ANALYTICAL ENERGY AND ERROR MODELS

A. Digital computations: quantization models

Energy models for digital computation blocks are well understood [4], [6], [14], [15], and are summarized in Section V. We focus here on the impact of bitwidth-dependent quantization in digital blocks on error. The error arising from quantizing analog signals in an ADC, relative to the signal strength, is quantified as signal-to-quantization noise ratio (SQNR), where quantization noise reflects the loss in accuracy due to quantization. For B bits of precision, the SQNR is [16]:

$$\text{SQNR} = 1.76 + 6.02B$$  \hspace{1cm} (1)

B. Analog computations: Noise and energy models

Mapping an ML algorithm to hardware requires hardware blocks for multiple types of operations such as MAC, sum, linear scaling, ReLU/sigmoid/tanh, and max-pooling. We analyze analog building blocks for these operations and present their energy and noise models. Sources of variation. There are several sources of noise in analog circuits. Process-induced drifts and parasitic effects can largely be canceled out [17] by using state-of-the-art precise sub-femtofarad capacitors with 1% standard deviation (in active switched-capacitor-based structures) and differential structures. This leaves intrinsic mechanisms, i.e., thermal noise and 1/f flicker noise: at >100 MHz, as in this work, thermal noise is the dominant contributor [16].

In resistors, the thermal noise voltage is proportional to $\sqrt{KT}$, where $k$ is Boltzmann’s constant, $T$ is the temperature, and $R$ is the resistance. In switched capacitor circuits (which we will use extensively), the RMS thermal noise is proportional to $\sqrt{KT/C}$ (and independent of $R$ [16]). To improve the precision of switched capacitor networks by one bit, $\sqrt{KT/C}$ must be halved: thus, every bit of precision requires 4× larger capacitors, with 4× higher energy. This makes analog circuits unsuitable for high precision, but for < 8-bit precision, analog implementations remain attractive [1].

1) Addition/subtraction: Addition can be performed in the analog domain using charge sharing [12], as illustrated in Fig. 3(a). Initially, the switch is open and the operands are loaded as analog voltages on the capacitors. When the switch is closed, the operands are averaged, thus implementing a scaled addition operation. The energy consumed in the circuit is the switching energy for the transistor, used to charge/discharge the gate capacitor of the transistor, and the energy for charging the two capacitors. The energy and noise are given by:

$$E_{ADD} = E_{switch} + C(V_a^2 + V_b^2); \quad N_{ADD} = 2KT/C$$

2) Analog multiplication: Matrix multiplication ($y = Ax$, $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$), is a fundamental operation in ML that has been traditionally realized by MAC units. We explore two analog multipliers [17], with analog or digital inputs, and an analog output. Analog input voltage MAC (AMAC): In ML hardware, the weights are typically stored in digital form. For a MAC operation with analog inputs, we use a highly energy-efficient switched-capacitor matrix multiplier as shown in Fig. 3(b) [17]. Here $C_1$ is a capacitive DAC that is controlled based on the weights stored in the memory. In the first phase ($\Phi_1$), the input voltage $V_a$ is multiplied by $C_1$, and in the second phase ($\Phi_2$), the multiplied charge $V_aC_1$ is redistributed on capacitor $C_2$. For an $N$-dimensional inner product, this operation is performed $N$ times such that $V_{2C_1} = (1/C_2) \sum_{i=1}^{N} V_a[i]C_1[i]$. For low-resolution multiplication, $C_2 \gg C_1$, i.e., for 3-bit precision, $C_2$ is 39× larger than the maximum $C_1$, and for every bit of precision,
we increase the size of $C_2$ by $4\times$ to overcome thermal noise. The total energy is the sum of the energy dissipated in the capacitors and in the switches. The energy and noise equations for this circuit are:

$$E_{AMAC} = 4E_{switch} + (C_1 + C_2)V_{DD}^2; \quad N_{AMAC} \approx kT/C_2$$

where $E_{switch}$ is the average energy dissipated in charging/discharging the gate capacitor of each of the four switches and $C_2 \gg C_1$.

Digital input voltage MAC (DMAC): When both the weights and the signal are digital, one option is to convert the data input to an analog value using a DAC, and then use the analog MAC as explained in the previous paragraph. In some cases, the digital properties of the signal can be utilized to perform a bit-wise product to reduce the energy loss in the DAC capacitance. In [9], XNOR gates are used for bitwise multiplication, reducing the energy consumed in the DACs. The noise in this operation comes from digital quantization noise and from thermal noise ($kTC$) from the switching of the unit capacitors connected to the output of XNOR gates. The energy consumption and noise for this circuit for a B-bit operation is given by:

$$E_{DMAC} = B\alpha(E_{switch} + C_uV_{DD}^2); \quad N_{DMAC} = kT/(2^B C_u)$$

3) Max-pooling operation: Pooling layers downsample features to reduce the computational burden on subsequent layers. Two common pooling methods are average- and max-pooling. Average-pooling can be implemented using the sum operation described earlier, and max-pooling can be implemented using a voltage-mode-max circuit [18].

Using transistor noise equations for a single pooling operation, energy and noise are a function of the signal path current ($I_D$):

$$E_{pool} = 2I_DV_{DD}; \quad N_{pool} \approx K'/I_D$$

where $K'$ is derived using the circuit topology and device sizes.

4) Activation functions: ML models use nonlinear activation functions such as ReLU, tanh, and sigmoid. The ReLU function selects the maximum of the input signal and the reference (usually 0), and thus max-pooling circuits can implement ReLU. A tanh or sigmoid function can be realized by a common-source differential amplifier by leveraging intrinsic transistor nonlinearity [19]. The total energy and output-referred noise of the 5T-OTA circuit is [19]:

$$E_{sigmoid} = 2I_DV_{DD}; \quad N_{sigmoid} = 32kTn_f(V_{gs} - V_{ih})/(3\lambda^2 I_D)$$

where $\lambda = 1$ (channel length modulation); $n_f$ depends on the ratio of transistors sizes; $I_D$ is the current through each signal path.

C. Signal conversion

In multi-stage analog operations, the accumulation of noise over multiple stages can cause significant degradation in SNR, to the point where the system may not be able to retain enough precision. Therefore, as described at the end of Section I, after a set of analog operations, we must restore the signals to the digital domain using an ADC [1]. After restoration, the computations can be performed in the digital domain or converted back to analog using a DAC.

1) ADC: The ADC energy is dependent on the effective number of bits (ENOB) [9], where ENOB is derived from the full-scale signal range ($\text{signal}_{LSB}$) and noise at the ADC input. For a B-bit ADC,

$$E_{ADC} = K_1 \cdot \text{ENOB} + K_2 \cdot 4^{\text{ENOB}}$$

$$\text{ENOB} = \log_2 \left( \frac{\text{RMS(signal)}_{LSB}}{\text{RMS(noise}_{ADC}) \times \sqrt{12}} \right)$$

As the energy is proportional to ADC precision, we choose a minimum-precision ADC under the noise requirements. In convolution operations, since the ADC energy is shared across all filter elements, using a larger filter size can reduce the overall ADC energy.

2) DAC: We use a charge-distribution DAC [16], which also behaves as a sample-and-hold circuit (SHA). This saves chip area and power by removing the need for an external SHA. If $C_u$ is the minimum realizable capacitance, the energy and the noise for a B-bit DAC are:

$$E_{DAC} = 2^B \alpha(C_uV_{DD}^2 + E_{switch}); \quad N_{DAC} = kT/(2^B C_u)$$

IV. Noise Propagation in ML Architectures

Based on the operations described in the previous section, we can model the energy and noise within each layer. Next, we consider noise propagation across layers. The weight distribution across layers in a deep neural network spans diverse numerical ranges [3]; therefore, a layer-wise quantization scheme improves the overall accuracy. We use the quantization noise arising from a uniform quantization scheme to explain how the noise is propagated through typical analog or digital operations in ML inference [20].

Digital noise. We use linear integer quantization noise for digital signals based on chosen bit precision. Quantization noise in each digital operator is assumed independent. After propagation over multiple operations, by the Central Limit Theorem, noise can be approximated as a normal distribution with zero mean/constant variance, $\sigma^2$.

Analog noise. This is calculated for each analog operation using noise equations (Section III-B), for a chosen equivalent bit precision. ADC/DAC noise is captured using methods from Section III-C.

Analog/Digital noise propagation. We show how the noise is propagated through some typical ML inference operations. Assuming noise at each layer weight to be independent with zero mean, if the input noise variances are $\sigma_1^2$ and $\sigma_2^2$ for a two-input operation (or just $\sigma_1^2$ if unary), the output variance, $\sigma^2$, is:

- Add/Subtract: $\sigma^2 = \sigma_1^2 + \sigma_2^2$
- Multiply: $\sigma_1^2 \sigma_2^2$
- sigmoid [21]: $\sigma^2 = \frac{1}{\cosh(\alpha(x_1 + x_2)) - 1} \cdot \frac{1}{\cosh(\alpha x_1) - 1} \cdot \frac{1}{\cosh(\alpha x_2) - 1}$
- ReLU: $\sigma^2 / 2$

Consider the operation $x_3 = x_1 + x_2$. Let $\sigma_{x_1}^2, [\sigma_{x_2}^2]$ be the variance of noise at input $x_1, x_2$. Let $\sigma_{x_3}$ be variance of the noise introduced by the addition operator. For addition in the digital [analog] domain, $\sigma_{x_3}^2$ represents the noise variance from the quantization of the adder [analog circuit nonidealities]. Since the total noise variance of the operation is the sum of all three variances, if $\sigma_{x_3}^2$ is the signal power at the output, then the SNR at the output is:

$$\text{SNR}_0 = \frac{\sigma_{x_1}^2}{\sigma_{x_3}^2 + \sigma_{x_2}^2 + \sigma_{x_3}^2}$$

### Table I: LUT for analog/digital operations, at various precisions.

<table>
<thead>
<tr>
<th>Operations</th>
<th>Analog Energy ($\Omega$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>1-bit</td>
</tr>
<tr>
<td>ADC</td>
<td>100</td>
</tr>
<tr>
<td>Addition</td>
<td>2.5</td>
</tr>
<tr>
<td>DMAC</td>
<td>0.2</td>
</tr>
<tr>
<td>Max-pool</td>
<td>1</td>
</tr>
<tr>
<td>sigmoid</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operations</th>
<th>Digital Energy ($\Omega$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>1-bit</td>
</tr>
<tr>
<td>DAC</td>
<td>0.5</td>
</tr>
<tr>
<td>Addition</td>
<td>9.5</td>
</tr>
<tr>
<td>DMAC</td>
<td>9.7</td>
</tr>
<tr>
<td>Max-pool</td>
<td>3.7e3</td>
</tr>
<tr>
<td>sigmoid</td>
<td>2.3e3</td>
</tr>
</tbody>
</table>
**V. MiS0-ML SETUP**

Energy lookup table. The operations may be performed at a precision (or equivalent precision for analog computation) of 1–8 bits. Based on the models described previously, we generate a lookup table for energy and noise for each operation, as listed in Table I.

The energy for digital operations in Table I is based on a 12 nm node at $V_{DD} = 1V$ and includes the energy for memory read/write. We assume an activity factor $\alpha = 0.1$. The energy values for all bit precisions are derived by scaling the energy for 8-bit operations based on the number of bit operations, e.g., addition scales linearly and multiplication scales with the square of the number of bits.

For analog topologies, for any clock/data switching, we consider a switching energy of $2fJ$. For the energy model corresponding to 1-bit precision, the minimum realizable capacitor ($C_{int}$) of 0.3fF. We set current ($I_D$) of 50 nA, temperature $T = 300K$, $k = 1.38 \times 10^{-23}K/J$, and channel length modulation $\lambda = 1$. To increase the precision by one bit under same the signal strength, noise must be reduced by half; this is achieved by quadrupling the capacitor sizes and current ($I_D$). In Eq. (4) for the max-pooling operation, $K' = 2.96 \times 10^{-20}$ as derived based on the circuit in [18] to achieve minimum energy, and in Eq. (5) for sigmoid, $n_f = 1.5$. $\Delta V = 0.1V$ [19].

For the ADC, we select the power numbers that correspond to the best ADC design for a given number of bits using [22], which gives $K_1 = 100fJ$, $K_2 = 1aJ$. For lower precision, we use successive-approximation-register (SAR) ADCs which use $B + 1$ cycles for computation. Based on the choice of capacitor sizes the ADCs can easily work up to 1 GHz, thus keeping the throughput of the model at 100 MHz ($\approx 111$ MHz for an 8-bit ADC operation [16]).

**Modeling error sensitivity**. We estimate digital noise using the quantization model from Section IV. Analog noise is modeled as white Gaussian noise with zero mean and variance $= 1/2^B$ [1].

A pre-trained model using 32-bit floating point (FP32) weights is used and layer-wise quantization is implemented. Fig. 4(a) shows the layer-by-layer analysis of the resilience of the ResNet18 model towards quantization. Quantization levels above 6-bit integers perform well, and these curves are not shown in the graph for enhanced readability. Below 6-bit quantization, we can see that the ResNet18 model has significant degradation in accuracy, though some accuracy can be regained after quantization-aware fine tuning of the model. A similar characteristic was observed while adding Gaussian noise to the weights of the model (Fig. 4(b)), but in this case, the network shows minimal degradation up to noise equivalent to 5-bits ($\sigma_{noise} = 1/32$). This shows that the ResNet18 model is more resilient to analog noise than quantization noise.

We evaluate the sensitivity of the model $S_j^t$, indexed by the $j^{th}$ operation of type $t$, as the ratio of the error of ML model to the added noise (analog Gaussian or digital quantization) for the operation. We will utilize these relative sensitivities of the added quantization and Gaussian noise in the ILP formulation described next.

**VI. MiS0-ML PRECISION OPTIMIZATION**

Consider an ML architecture composed of M layers, each of which performs one operation from the available set of operations {Addition, MAC, Max-pool, ReLU, sigmoid}. Each operation can have a precision of B bits, 1 < B < 8, and can either be implemented using an analog or digital operation ($O^t$); if the output is very sensitive to an operation, it should preferably be performed using digital circuitry, but a less sensitive operation can be implemented in analog, using a noise threshold based on the desired precision to determine ADC insertion points (see **Intuitive concept**). The set of allowable operations $O$ is all combinations of (operation set, bit precision, implementation mode (analog/digital)).

Our goal is to minimize the overall energy and noise in the system, trading off noise/energy using digital and analog components, and inserting ADCs/DACs as needed. Using the notation in Table II, we formulate an ILP to find the bit-precision assignment, the choice of analog/digital for each operation, and the ADC/DAC locations. ILP Objective: We seek the best compromise in energy and noise as we solve the assignment problem. The objective function minimizes the sum of the overall energy consumed and noise generated (weighted by sensitivity $S_j^t$) by all operators and ADCs/DACs, as represented in the optimization formulation below:

$$
\text{min } \sum_{j=1}^{M} \left( E(O_j^t) + S_j^t N(O_j^t) \right) + E_{\text{total}} + E_{\text{total}}
$$

s.t. $\sum_{B=1}^{8} \left( I_j^{AB} + I_j^{BB} \right) = 1$

$$
E(O_j^t) = \sum_{B=1}^{8} I_j^{AB} \cdot E(A_j^B) + I_j^{BB} \cdot E(D_j^B)
$$

$$
I_j^{ADC} = I_j^{AB} \land \left( \bigwedge_{k \in P} \left( \bigvee_{l \in O(k)} N(A_j^{B_k}) \right) \cdot k^{D_j^B} \right) \leq N_T
$$

$$
E_{\text{total}} = \sum_{j=1}^{M} \sum_{B=1}^{8} I_j^{AB} \cdot E(A_j^B)
$$

The logical AND ($\land$), OR ($\lor$), and NOT ($\neg$) operations in the ILP formulation can be easily modeled using ILP constraints [23].

Unique variant constraint: A special order set (SOS) constraint is formulated in (9) to choose one of the available variants (analog or digital, and a specific number of bits) for an ML layer.

Energy computation: With the above SOS representation, the energy consumed by an operator $O_j^t$ (defined in Table II) can be evaluated as shown in (10). A similar expression represents the noise for $O_j^t$.

ADC/DAC constraint: A B-bit ADC is required at the output of an analog operator of B-bit equivalent precision, whose output drives at least one operator that is digital, regardless of the digital precision. This is specified as the logical constraint (11). As described in Table II, operator $O_j^t$ is an analog operator with B-bit equivalent...
The SNR may degrade significantly when multiple analog operators are cascaded. To prevent this, a chain of analog operators must be followed by a digital operator to restore signals to discrete values, as described in Section III-C. To specify this constraint, consider a path \( P \) in the network comprising successively adjacent operators, whose last operator drives a digital operator \( l \). Any path composed of all-analog operators must have an accumulated noise of less than \( N_T \). All the operators in a path are analog if and only if the analog indicator variables of all operators in the path are 1, which is captured by an AND constraint. A \( B_3 \)-bit precision for operator \( k \in P \) is specified as shown in (12).

The ILP solution assigns to each operator a bit-precision and analog/digital variant based on the indicator variable. An ADC/DAC is added to the output of an operator if its indicator variable is 1.

VII. EVALUATION OF MiSO-ML

We demonstrate our approach on multiple ML architectures. ResNet18 with ImageNet. We first apply our approach on ResNet18 with a workload from the ImageNet 1k dataset. Our ILP solution provides the bit-precision, choice of analog/digital implementation, and the locations and bitwidths of the ADCs/DACs corresponding to the optimal noise and minimum energy. We compare this with a reduced bitwidth digital optimization, i.e., digital quantization.

Table III shows energy and quantization numbers corresponding to uniform digital quantization across all layers of ResNet18 on the ImageNet dataset. Models with uniform 8-bit quantization achieve similar accuracy compared to the baseline FP32 model. The use of 6-bit and 4-bit uniform quantization sees some accuracy drop, which is recovered after retraining, accounting for quantization and analog noise (“fine tuning”). However, for 2-bit uniform quantization, the accuracy degrades greatly and cannot be recovered by fine tuning.

Table III: Energy vs top-1 accuracy trade-off for ResNet-18 architecture on the ImageNet dataset.

<table>
<thead>
<tr>
<th>Precision (D) = Digital, (A) = Analog</th>
<th>Accuracy</th>
<th>Energy (Improvement)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>Fine tuning</td>
<td></td>
</tr>
<tr>
<td>FP32 (D)</td>
<td>72.2%</td>
<td>69.8%</td>
</tr>
<tr>
<td>8-bit Activation, 8-bit Weight (D)</td>
<td>69.49%</td>
<td>69.99%</td>
</tr>
<tr>
<td>6-bit Activation, 6-bit Weight (D)</td>
<td>68.43%</td>
<td>69.21%</td>
</tr>
<tr>
<td>4-bit Activation, 4-bit Weight (D)</td>
<td>66.82%</td>
<td>68.66%</td>
</tr>
<tr>
<td>2-bit Activation, 2-bit Weight (D)</td>
<td>62.14%</td>
<td>65.38%</td>
</tr>
<tr>
<td>MiSO-ML (4/3B Activation, 4/3B Weight (mix))</td>
<td>67.14%</td>
<td>69.16%</td>
</tr>
</tbody>
</table>

The MiSO-ML mixed precision method, with mixed analog/digital implementations with cascaded analog MAC structures, uses 6× less energy than 6-bit uniform quantization, which provides the same accuracy. Relative to [9], [13], which use ADCs between successive layers, the benefit of MiSO-ML comes due to explicit optimization of the number of ADCs between multiple analog stages: since ResNet uses convolution filters of size \( 3 \times 3 \) for all of the convolution stages (except the first convolution layer with \( 7 \times 7 \) filter size), the ADC energy cost becomes a significant portion of MAC energy in [9], [13].

The results of optimizing ResNet18 on ImageNet using MiSO-ML are shown in Fig. 5. For each layer (1–18 on the x-axis), the weights are always digital; depending on the ILP optimization, the activations and operation may be digital (“D”) or analog (“A”). The bars show the equivalent number of bits (if analog) or truncated bits (if digital) for the weights, activations, and operation, according to the left y-axis. The ILP places data converters at layers 6, 9, 13, and 17.

The green line shows the energy reduction per layer (right y-axis) as compared to a 6B digital implementation, which has similar accuracy (Table III). Layers with analog operations provide large energy reductions; those with digital operations (1, 17, 18) less so, but maintain accuracy. For a threshold noise of \( \sigma_{max}(N_T) = 1/20 \), 2 to 3 layers can be stacked in the analog domain before data conversion.

Accuracy vs. energy tradeoffs. For the same experimental setup (ResNet18 on ImageNet), Fig. 6 shows the energy vs. accuracy tradeoff for different noise thresholds for MiSO-ML, and with equivalent bit truncation for a digital implementation. We see a large energy improvement by the MiSO-ML model for lower bit precisions, and more modest improvement for digital truncation. The accuracy cost is minimal up to 4-bit precision but is more noticeable for a lower number of bits. Using this energy vs. accuracy tradeoff curve, we select 4-bit precision, and this limits the AND constraints in the ILP.

Benefit of stacking analog operations. To understand the benefit of stacking multiple layers before ADC insertion, we plot the percentage contribution of the ADCs to the overall energy in Fig. 7. We assume that the input and output layers of the ML architecture are in the digital domain. The yellow bars correspond to the MiSO-ML architecture with ADCs inserted based on the ILP, and the gray bars correspond to the operations at a bit precision optimized in MiSO-ML, but with an ADC/DAC after each ML layer, similar to [9], [13]. The energy gain is considerable for 4–6 bit operations as compared to 7–8 bit operations. At stricter 7- and 8-bit precision, smaller benefits from analog operations are seen, due to high ADC/DAC energy costs.

The improved benefit of analog operation at 6-bits and below is attributed to the amortization of the cost of ADC and DAC over a stack of multiple layers; this is less so at 7–8 bits. We observe a stacking of up to two layers for a 6-bit equivalent precision and a stacking of up to four layers for a 4-bit equivalent precision. At 3-bit and lower precision, we do not observe any significant reduction of ADC energy component due to reduced energy scaling for the ADC operation, as seen in Table I. For a 4-bit noise threshold, the ADC energy overhead is just 13.2% of the total energy of ResNet18, much smaller as compared to single-layer architectures (52.2% in [11]).

General ML Architectures. We have tested various networks and observed consistent improvement across all. In Table IV, we describe the energy efficiency achieved by our model on multiple ML architectures. We chose four datasets to test our model: MNIST (using image-pixel \( 28 \times 28 \)), ImageNet (using image-pixel \( 224 \times 224 \)), Imagenette (224x224), Imagenette (224x224), ImageNet workload. MiSO-ML energy improvement (gray columns) is shown in Fig. 6 (ResNet18 on ImageNet), Fig. 6 shows the energy vs. accuracy tradeoff for different noise thresholds for MiSO-ML, and with equivalent bit truncation for a digital implementation. We see a large energy improvement by the MiSO-ML model for lower bit precisions, and more modest improvement for digital truncation. The accuracy cost is minimal up to 4-bit precision but is more noticeable for a lower number of bits. Using this energy vs. accuracy tradeoff curve, we select 4-bit precision, and this limits the AND constraints in the ILP.
Table IV: Energy reduction and accuracy for different methods.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ML Architecture</th>
<th>Accuracy</th>
<th>Energy Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>SVM (Accuracy = 94.23%)</td>
<td>MiSO-ML: 92.31%</td>
<td>5.99%</td>
</tr>
<tr>
<td></td>
<td>ResNet18 (Accuracy = 96.57%)</td>
<td>MiSO-ML: 91.46%</td>
<td>6.08%</td>
</tr>
<tr>
<td>ImageNet</td>
<td>VGG16 (Accuracy = 92.32%)</td>
<td>MiSO-ML: 90.51%</td>
<td>5.48%</td>
</tr>
<tr>
<td></td>
<td>GoogleNet (Accuracy = 93.23%)</td>
<td>MiSO-ML: 91.57%</td>
<td>5.86%</td>
</tr>
<tr>
<td></td>
<td>ResNet10 (Accuracy = 90.55%)</td>
<td>MiSO-ML: 93.76%</td>
<td>3.16%</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>ResNet10 (Accuracy = 91.12%)</td>
<td>MiSO-ML: 89.50%</td>
<td>3.86%</td>
</tr>
<tr>
<td></td>
<td>ResNet20 (Accuracy = 91.55%)</td>
<td>MiSO-ML: 92.55%</td>
<td>2.89%</td>
</tr>
</tbody>
</table>

Figure 7: ADC energy (% of total energy) vs. noise threshold constraints for ResNet18. Gray bars: ADC inserted after each layer [9].

We propose MiSO-ML, a mixed-signal optimization framework for low-power ML inference. We demonstrate that it can enable image recognition for multiple ML architectures. We observe a gain of 5-8× lower energy than 8-bit quantized digital implementations, with minimal accuracy loss. The main energy benefit comes from the low ADC energy (13.2%), amortized across multiple layers and selection of energy-efficient analog/digital hardware for different precisions. The ILP formulation ensures that accuracy loss is minimal.

REFERENCES