MINIMUM PRECISION REQUIREMENTS FOR THE SVM-SGD LEARNING ALGORITHM

Charbel Sakr, Ameya Patil, Sai Zhang, Yongjune Kim, and Naresh Shanbhag

Dept. of Electrical and Computer Engineering, University of Illinois at Urbana Champaign

ABSTRACT

It is well-known that the precision of data, weight vector, and internal representations employed in learning systems directly impacts their energy, throughput, and latency. The precision requirements for the training algorithm are also important for systems that learn on-the-fly. In this paper, we present analytical lower bounds on the precision requirements for the commonly employed stochastic gradient descent (SGD) on-line learning algorithm in the specific context of a support vector machine (SVM). These bounds are obtained subject to desired system performance. These bounds are validated using the UCI breast cancer dataset. Additionally, the impact of these precisions on the energy consumption of a fixed-point SVM with on-line training is studied. Simulation results in 45 nm CMOS process show that operating at the minimum precision as dictated by our bounds improves energy consumption by a factor of $5.3\times$ as compared to conventional precision assignments with no observable loss in accuracy.

Index Terms— machine learning, fixed point, precision, energy, accuracy

1. INTRODUCTION

Precision of data, weight vector, and internal signal representations in a machine learning implementation has a deep impact on its energy consumption and throughput. Recent works have empirically studied the effect of moderate [1] and heavy [2, 3, 4, 5] quantization on the performance of learning systems. A more analytical approach has recently emerged where signal-to-quantization noise ratio (SQNR) is used as a metric to attempt establishing precision to accuracy trade-offs for the feedforward path [6] and training [7] of deep neural networks.

Our work addresses the problem of a systematic way of assigning precision to a fixed-point learning system while providing accuracy guarantees. We study the specific case of a support vector machine (SVM) [8] being trained using the stochastic gradient descent (SGD) algorithm, i.e., the SVM-SGD algorithm. Our approach is analytical in contrast to the trial-and-error approach employed today. Energy consumption is also brought in as a metric for consideration in the design of learning systems.

This work was supported in part by Systems on Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.

1.1. Contributions

Our contributions in this paper are the following: we provide analytical lower bounds on the precision of the data, weight vector, and training parameters for the SVM-SGD algorithm subject to requirements on classification accuracy. We also study the impact of precision reduction on the energy consumption of a baseline SVM-SGD architecture by employing architectural level energy and delay models in a $45\,\mathrm{nm}$ CMOS process. We show that up to $5.3\times$ reduction in energy is achieved when precisions are assigned based on the lower bounds as compared to a $16\,\mathrm{b}$ baseline implementation.

The rest of this paper is organized as follows. Section 2 presents necessary background on which we build up our work. Section 3 proposes our analytical bounds on precision for fixed-point implementations. Experimental results are included in Section 4. We conclude our paper in Section 5.

2. BACKGROUND

2.1. Online Learning SVM (SVM-SGD)

SGD is an efficient training method for machine learning algorithms [9]. At each iteration, the weight vector of the algorithm is updated as follows:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \gamma \nabla_{\mathbf{w}} Q(z_n, \mathbf{w}_n) \tag{1}$$

where $Q(\cdot)$ is the loss function to be minimized, γ is the step-size, $\mathbf{w}_n \in \mathcal{R}^D$ is the weight vector to be learned, z_n is the streamed sample which usually consists of an input data vector $\mathbf{x}_n$ and a corresponding label y_n, and D is the dimensionality of the input data.

SVM [8] is a simple and popular supervised learning method for classification. SVM operates by determining a maximum margin separating hyperplane in the feature space. For regularization, some feature vectors may be allowed to lie inside the margin making it a *soft* margin. The SVM predicts the label $\hat{y}_n \in \{\pm 1\}$ given a feature vector (i.e., input data vector) $\mathbf{x}_n$ as follows:

$$\mathbf{w}^T \mathbf{x}_n + b \underset{\hat{y}_n=-1}{\overset{\hat{y}_n=1}{\gtrless}} 0 \tag{2}$$

where $\mathbf{w}$ is the weight vector and b is the bias term. The classification error for the SVM is defined as $p_e = P\{Y \neq$

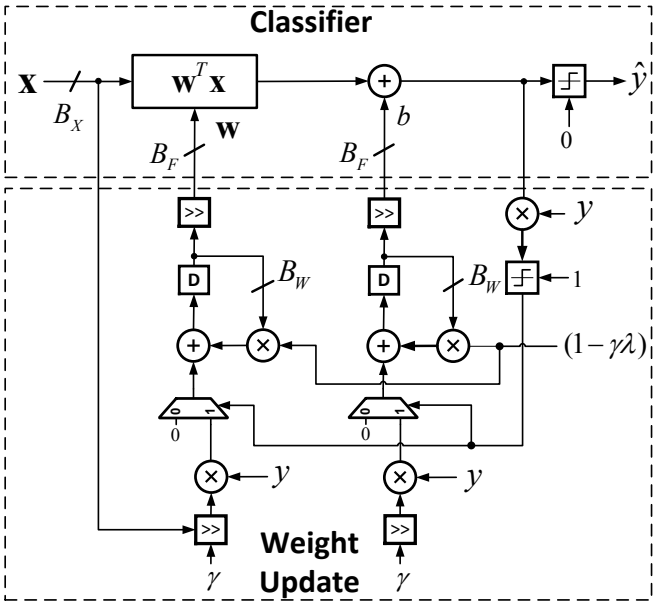


Fig. 1: SVM-SGD algorithm's architecture showing the precision per dimension.

$\hat{Y}\}$. In the rest of this paper, we employ capital letters to denote random variables.

The optimum weight vector $\mathbf{w}$ in (2) maximizes the margin or minimizes the average of following loss function:

$$Q(z_n, \mathbf{w}) = \lambda(\|\mathbf{w}\|^2 + b^2) + \max\{0, 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b)\} \quad (3)$$

It can be shown that, when applied to the SVM algorithm, specifically the loss function defined in (3), the SGD update equation is [9]:

$$\mathbf{w}_{n+1} = (1 - \gamma\,\lambda)\mathbf{w}_n + \begin{cases} 0 & \text{if } y_n(\mathbf{w}_n^T\mathbf{x}_n + b) > 1, \\ \gamma\, y_n\, \mathbf{x}_n & \text{otherwise.} \end{cases}$$
$$b_{n+1} = (1 - \gamma\lambda)b_n + \begin{cases} 0 & \text{if } y_n(\mathbf{w}_n^T\mathbf{x}_n + b) > 1, \\ \gamma\, y_n & \text{otherwise.} \end{cases} \quad (4)$$

2.2. Architectural Energy and Delay Models

Fig. 1 shows the architecture of the SVM-SGD algorithm and indicates the precision assignments (per dimension) for the input B_X, the classifier B_F, and the weight update B_W. The critical path of the architecture is any path between two latch nodes having the maximum delay, assuming inputs and outputs are also latched. The main building block of such computational system is the 1 b full-adder (FA). Hence the critical path is the one having maximum number of FAs. The maximum throughput at which the architecture can operate is the reciprocal of the critical path delay as follows [10]:

$$f_{max} = \frac{I_{ON}}{\beta L_{FA} C_{FA} V_{dd}} \quad (5)$$

where L_{FA} is the number of FAs in the critical path of the architecture, C_{FA} is the load capacitance of one FA, β is an empirical fitting parameter, I_{ON} is the ON current of a transistor, and V_{dd} is the supply voltage at which the architecture is being operated at. The energy consumption of the architecture *operating at* f_{max} is given by [10]:

$$E = \alpha N_{FA} C_{FA} V_{dd}^2 + \beta N_{FA} L_{FA} C_{FA} V_{dd}^2 10^{-\frac{V_{dd}}{S}} \quad (6)$$

where N_{FA} is the total number of FAs in the system, α is the activity factor, and S is the subthreshold slope.

It is found [11] that both N_{FA} and L_{FA} are linear in the precisions B_X, B_F, and B_W. Since E is proportional to the product of N_{FA} and L_{FA}, E is a quadratic function of these precisions. This clearly indicates the importance of minimizing B_X, B_F, and B_W. Note, this analysis assumes standard implementation using ripple carry adders and Baugh-Wooley multipliers.

3. ANALYTICAL BOUNDS ON PRECISION

The impact of precisions of data (B_X), weight vector (B_F), and weight update block (B_W) on the overall SQNR is well established for finite-impulse response (FIR) filters and least mean-squared (LMS) adaptive filters [12]. In this section, we analytically predict the accuracy of the SVM-SGD algorithm as a function of B_X, B_F, and B_W. All details concerning the derivation and proofs of the upcoming results can be found in [11].

3.1. Classification Block

We assume that the input data is quantized to B_X bits and is hence corrupted by quantization noise that is independently uniformly distributed on $[-\frac{\Delta_x}{2}, \frac{\Delta_x}{2}]$ per dimension [13]. Note that $\Delta_x = 2^{-(B_X-1)}$ is the input quantization step. Similarly, the weights are quantized to B_F bits and are hence corrupted by quantization noise that is independently uniformly distributed on $[-\frac{\Delta_f}{2}, \frac{\Delta_f}{2}]$ per dimension where $\Delta_f = 2^{-(B_F-1)}$.

Our first result is a lower bound derived based on the geometric property of the SVM. This lower bound on B_X ensures that any datapoint lying outside the margin is classified correctly even after being perturbed by quantization noise.

Theorem 3.1 (Geometric Bound)**.**
Given D, B_F, and $||\mathbf{w}||$, a feature vector $\mathbf{x}$ lying outside the margin will be classified correctly if

$$B_X > \log_2\left(\frac{\sqrt{D}||\mathbf{w}||}{1 - (1 + \sqrt{D}||\mathbf{x}||)2^{-B_F}}\right). \quad (7)$$

Proof. The main idea is to make sure that the worst case quantization noise is less than the functional margin of the classifier. For details, please see [11]. □

The geometric bound reveals the following: larger SVM margin (i.e., smaller $||\mathbf{w}||$) allows greater reduction of B_X; there is a trade-off relation between B_X and B_F; higher dimension D and larger $||\mathbf{x}||$ require more input precision B_X.

Note that Theorem 3.1 is specific to a single feature vector $\mathbf{x}$. The following is a simple corollary that applies to all datapoints in the dataset lying outside the margin.

Corollary 3.1.1.
Given D, B_F, and $||\mathbf{w}||$, any feature vector in the dataset $\mathcal{X}$ lying outside the margin will be classified correctly if

$$B_X > \log_2\left(\frac{\sqrt{D}||\mathbf{w}||}{1-(1+\sqrt{D}\max\limits_{\mathbf{x}\in\mathcal{X}}||\mathbf{x}||)2^{-B_F}}\right). \quad (8)$$

It is worth noting that $\max\limits_{\mathbf{x}\in\mathcal{X}}||\mathbf{x}|| \leq \sqrt{D}$ because of normalization.

Let $\hat{Y}_{fx}$ be the output of the fixed-point classifier and $\hat{Y}_{fl}$ be the output of the floating-point classifier. Our next result is an upper bound on the mismatch probability $p_m = \Pr\{\hat{Y}_{fx} \neq \hat{Y}_{fl}\}$ as a function of precision.

Theorem 3.2 (Probabilistic Bound)**.**
Given B_X, B_F, $\mathbf{w}$, and b, the upper bound on the mismatch probability p_m is given by:

$$p_m \leq \frac{1}{24}\left(\Delta_x^2||\mathbf{w}||^2\mathbb{E}\left[\frac{1}{|\mathbf{w}^T\mathbf{X}+b|^2}\right] + \Delta_f^2\mathbb{E}\left[\frac{||\mathbf{X}||^2+1}{|\mathbf{w}^T\mathbf{X}+b|^2}\right]\right) \quad (9)$$

where $\mathbf{X}$ denotes the random variable of the dataset. Note that the statistics (i.e., the expected values in (9)*) are calculated empirically.*

Proof. The main idea is to consider the mismatch event for one datapoint and upper bound its probability using Chebyshev's inequality. The law of total probability is then used to obtain the upper bound over the whole dataset. For details, please see [11]. □

Theorem 3.2 can be reformulated to obtain the lower bound on precision to achieve a target worst-case mismatch rate. It can also be extended to obtain the upper bound on the actual classification accuracy of the fixed-point realization as follows:

Theorem 3.3.
The fixed-point probability of error is upper bounded as follows:

$$p_e \leq 1 + \min(p_{fl}, p_m) - \max(p_{fl}, p_m). \quad (10)$$

where $p_{fl} = Pr\{\hat{Y}_{fl} = Y\}$ denotes the probability of detection in floating-point.

Proof. Basic laws of probability theory were used. For details, please see [11]. □

3.2. Weight Update Block Analysis

In the preceding discussion, $\mathbf{w}$ is the converged weight vector of an infinite-precision algorithm. In practice, it is standard to first implement a floating-point algorithm with desirable convergence properties and then to quantize so as to keep the convergence behavior unaltered.

The upcoming setup assumes without loss of generality that the floating-point convergence of SGD in the context of SVM (4) is obtained for $\lambda = 1$ and some small value of γ (typically a negative power of 2)

Our next result ensures that the non-zero update term in (4) is represented correctly during training in spite of weight update quantization.

Theorem 3.4.
The lower bound on the weight update precision B_W in order to ensure full convergence is given by:

$$B_W \geq B_X - \log_2(\gamma). \quad (11)$$

Proof. The idea is to prevent the updates from adding additional quantization. For details, please see [11]. □

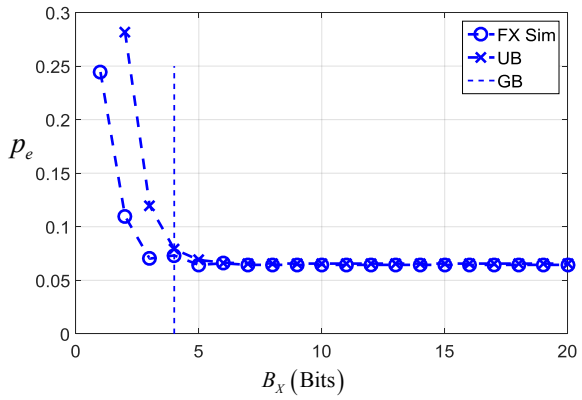
The setup of Theorem 3.4 is quite conservative. In fact, it is possible to sometimes break the bound in (11) and still obtain satisfactory convergence behavior [11]. Two particular cases are to be noted. If $B_W = 1 - \log_2(\gamma)$, then we obtain a sign-SGD behavior only for negative updates. This is because only the sign bit of the input data is retrieved after truncation at the end of the update. Similarly, if $B_W = -\log_2(\gamma)$, then we still obtain a sign-SGD behavior, but the effective step-size doubles due to truncation.

4. EXPERIMENTAL RESULTS

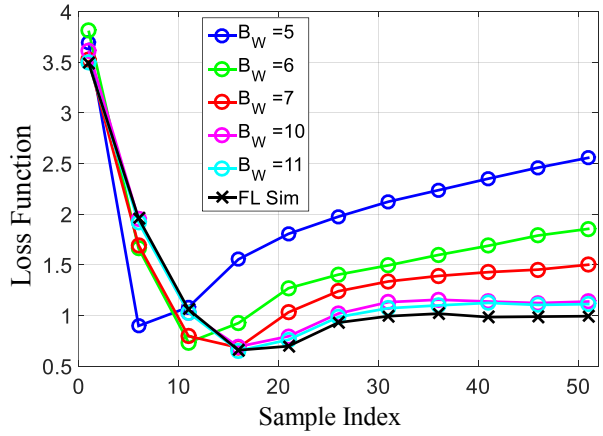
4.1. Classification Accuracy and Convergence

In this section, we validate our analytical results based on the UCI Breast Cancer dataset [14]. For fixed-point classification with $B_F = 6$, Fig. 2 (a) shows that $B_X \geq 5$ is sufficient to achieve a target classification probability of $p_e \leq 0.06$. This is consistent with the analytical values determined by the geometric bound (8) and the probabilistic bound (9): $B_X = 4$ and $B_X = 5$, respectively.

Fig. 2 (b) shows the convergence curve of the average loss function of SVM as described in (3) for various weight update block precisions B_W. The minimum precision given by (11) is $B_W = 11$ when $\gamma = 2^{-5}$, and $B_X = 6$ chosen to be consistent with the bounds shown in Fig. 2(a). We note that the fixed-point convergence curve tracks the floating-point curve very closely. We also show convergence curves for precisions $B_W = 10, 7, 6$, and 5. The curves are similar to the floating-point curve though with an observable loss in the accuracy for lower precisions. Further analysis explaining this loss in



(a)



(b)

Fig. 2: Experimental results for the Breast Cancer dataset: (a) fixed-point classifier simulation (FX sim), probabilistic upper bound (UB), and geometric bound (GB) for $B_F = 6$, and (b) convergence curves for fixed-point ($B_F = 6$ and $B_X = 6$) and floating-point (FL sim) SGD training ($\gamma = 2^{-5}$ and $\lambda = 1$).

accuracy can be found in [11]. Note: the initial faster convergence but eventual loss in accuracy when $B_W = 5$ is because it corresponds to $B_W = -\log_2(\gamma)$ meaning a sign-SGD behavior with double the step size as discussed in Section 3.

4.2. Energy Consumption

We employed the energy-delay models in Section 2.2 to estimate the energy consumption of the SVM-SGD architecture (Fig. 1) when processing the Breast Cancer dataset. Our methodology is the same as the one in [10] but it uses a 45 nm semiconductor process parameters. This methodology is useful for estimating the benefits of precision reduction without a time-consuming and laborious ASIC design process. The model parameters obtained through circuit simulations of a full adder and their values can be found in [11]. Based on these parameters, we evaluate the energy consumption in (6). Fig. 3 shows the energy consumption of the architecture as a function of supply voltage for different precisions in order to achieve $p_m < 0.01$. An architecture that employs 16 b for all variables is chosen as reference as recent works on fixed-point learning have employed commercial 16 b digital signal processors [1]. As discussed in [10], the trade-off between leakage and dynamic energy (the two terms in (6)) leads to the well known minimum energy operating point (MEOP) for each curve. As shown in Fig. 3, the precision assignment using (9)-(11) results in 5.3× energy savings at the MEOP over a 16 b fixed-point implementation.

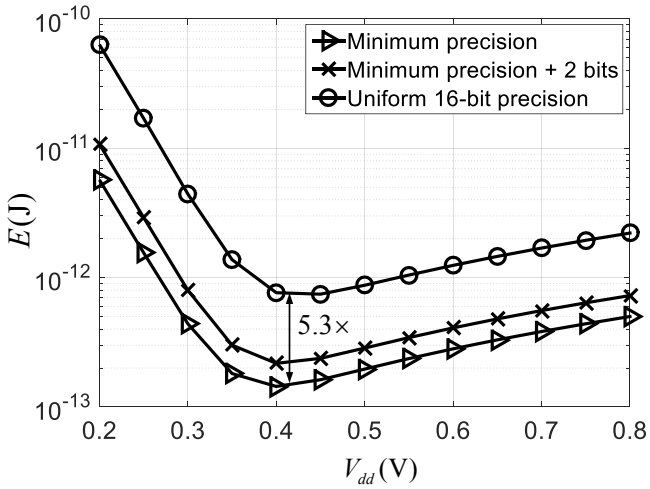


Fig. 3: Energy savings using minimum precision ($B_F = 6$, $B_X = 5$, $B_W = 10$) for $p_m < 0.01$, compared to 2 b higher precision ($B_F = 8$, $B_X = 7$, $B_W = 12$) and uniform precision assignment of 16 b ($B_F = 16$, $B_X = 16$, $B_W = 16$).

5. CONCLUSION

In this paper, we presented analytical bounds on the precisions of data, weight vector, and weight update block for the SVM-SGD algorithm. These bounds are closely related to performance metrics such as the classification accuracy and mismatch probability. Moreover, we quantified the impact of precision on the energy cost of inference in a commercial semiconductor process technology using circuit analysis and models. We observed significant energy savings when assigning precision at the lower bounds as compared to prior work. Using our results, designers of fixed-point realization can efficiently determine precision requirements analytically. Future work includes developing a similar fixed-point analysis of the popular Deep Neural Networks (DNN). Such networks tend to be highly complex. A systematic study of precision requirements will be extremely valuable.

References

[1] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *Proceedings of The 32nd International Conference on Machine Learning*, 2015, pp. 1737–1746.

[2] M. Kim and P. Smaragdis, "Bitwise Neural Networks," *arXiv preprint arXiv:1601.06071*, 2016.

[3] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *Advances in Neural Information Processing Systems*, 2015, pp. 3123–3131.

[4] Matthieu Courbariaux and Yoshua Bengio, "Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1," *arXiv preprint arXiv:1602.02830*, 2016.

[5] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi, "XNOR-Net: ImageNet classification csing binary convolutional neural networks," *arXiv preprint arXiv:1603.05279*, 2016.

[6] Darryl D Lin, Sachin S Talathi, and V Sreekanth Annapureddy, "Fixed point quantization of deep convolutional networks," *arXiv preprint arXiv:1511.06393*, 2015.

[7] Darryl D Lin and Sachin S Talathi, "Overcoming challenges in fixed point training of deep convolutional networks," *arXiv preprint arXiv:1607.02241*, 2016.

[8] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[9] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

[10] R. Abdallah and N. Shanbhag, "Reducing energy at the minimum energy operating point via statistical error compensation," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 22, no. 6, pp. 1328–1337, 2014.

[11] Charbel Sakr, Ameya Patil, Sai Zhang, and Naresh Shanbhag, "Understanding the energy and precision requirements for online learning," *arXiv preprint arXiv:1607.00669*, 2016.

[12] M. Goel and N. Shanbhag, "Finite-precision analysis of the pipelined strength-reduced adaptive filter," *Signal Processing, IEEE Transactions on*, vol. 46, no. 6, pp. 1763–1769, 1998.

[13] K. Parhi, *VLSI digital signal processing systems: design and implementation*, John Wiley & Sons, 2007.

[14] A. Asuncion and D. Newman, "UCI machine learning repository," `http://archive.ics.uci.edu/ml/`, 2007, [online].