Zhuo Wang, Jintao Zhang,
and Naveen Verma

Reducing Quantization Errors for Inner-Product Operations in Embedded Digital Signal Processing Systems

Inner-product operations are used extensively in embedded digital signal processing (DSP) systems. Their applications range from signal processing (filtering/convolution) to inference (classification). In embedded systems, resources (energy, area, etc.) are typically highly constrained, making tradeoffs with computational precision a fundamental concern. Indeed, with increasing requirements on algorithmic performance, many systems are trending toward higher computational precision to ensure accuracy of results.

This article describes an approach to significantly enhance accuracy for inner-product operations at very low bit precisions, significantly improving the energy/area-versus-accuracy tradeoffs traditionally incurred [1]. Low-energy embedded systems often employ linear, fixed-precision representation for operands due to the simplicity and relative efficiency at lower dynamic range. We focus on the use of a floating-point representation. For specific distributions of operands very commonly observed, such a representation substantially enhances the accuracy of most operands. However, particularly at the low bit precisions we focus on, it raises the issue that many operands can incur a large quantization error, much greater than that with standard linear, fixed-precision representation. To address this, a simple optimization is applied within the quantization process, which is shown to significantly improve accuracy at a given bit precision. By targeting the way in which operands are represented, the approach incurs no added hardware or computational overheads and in fact is shown to reduce energy and area thanks to simplified computation associated with the representation.

A distribution of operands commonly encountered

An inner-product operation can be represented as follows:

$$y = \sum_i h_i x_i. \tag{1}$$

Here, x_i is input data, while h_i is typically predefined coefficients. For example, in the case of a finite impulse response (FIR) filter, x_i are signal samples to be filtered, while h_i are the filter coefficients; in the case of a perceptron (i.e., linear classifier), x_i are input feature elements, while h_i are the weights and bias. Quite often, the distribution of h_i values can result in severe quantization errors when implementing the

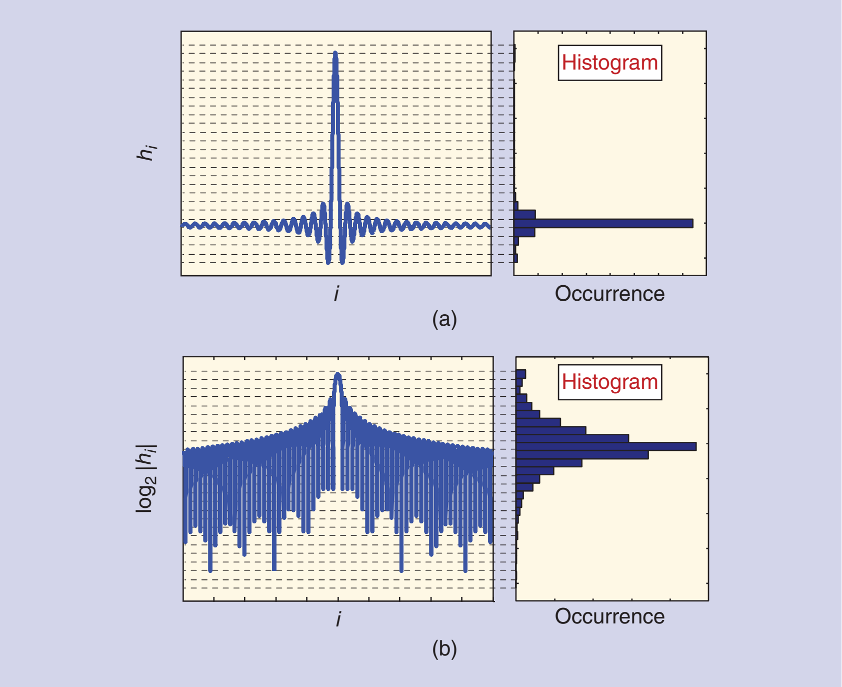

FIGURE 1. The distribution of filter coefficients for a low-pass FIR filter, with (a) standard linear fixed-point quantization and (b) floating-point representation.

Digital Object Identifier 10.1109/MSP.2016.2585278
Date of publication: 4 November 2016

inner-product operation in finite precision, as in an embedded DSP system. As an example, Figure 1 shows the distribution of h_i values for a low-pass FIR filter (plotted is $h_i = \sin c(n_i\omega)$ for $n_i\omega \in [-10\pi, 10\pi]$).). The situation is similar for band-pass filters, for instance, achieved by modulating the low-pass impulse response with a sinusoid. With standard linear, fixed-point quantization, as shown in Figure 1(a), the majority of the coefficients fall in low-value bins. This implies inefficient use of the available dynamic range. Even worse, depending on the level of quantization, many of the h_i values fall in the zero-valued bin, resulting in severe output errors, as the number of such elements can generally be quite high. Such a result is commonly observed, e.g., high-order FIR filters (low pass, band pass), linear regression under near colinearity conditions, etc. [2]. To address this, we consider employing a floating-point representation. First, we examine a standard floating-point representation in terms of how it impacts the quantization error. Then, we examine the new opportunity this raises for optimizing the quantization error. Note that we assume that x_i remains in a fixed-point representation, as this may be determined by the preceding signal processing stage and therefore required for system simplicity (i.e., to avoid the overhead of explicitly altering the representation). Thus, the approach actually employs a mixed fixed-/floating-point representation.

Mixed fixed-/floating-point computation

The potential benefit of using floating-point representation for h_i is that increased resolution is allocated where the density of h_i values is greatest, i.e., low-valued bins. Further, no h_i values fall in a zero-valued bin. As illustrated in Figure 1(b), by spanning more quantization bins, such a representation leads to more efficient use of the available dynamic range for the example of the low-pass FIR filter.

To be more precise, the floating-point representation used for h_i is shown in (2)–(5), where l_i represents the sign of h_i, s_i represents the exponent, and $1+m_i$ represents the significand in the range of $[1,2)$. Among these parameters, only m_i can take on continuous values. Thus, values of m_i in the range of $[0,1)$ must be quantized ($\hat{m}_i$) for system implementation.

$$h_i = l_i \times 2^{s_i} \times (1+m_i) \tag{2}$$

$$l_i = sign(h_i) \tag{3}$$

$$s_i = \lfloor \log_2 |h_i| \rfloor \tag{4}$$

$$m_i = \frac{|h_i|}{2^{\lfloor \log_2 |h_i| \rfloor}} - 1. \tag{5}$$

There is a practical consideration in choosing the number of bits assigned to m_i and s_i. For example, (6) and (7) show two approaches to represent the number -100 with 6 bits when using floating-point representation. The first approach assigns more bits to s_i, thus can represent a larger range of h_i (i.e., $(\mathbf{max}|\boldsymbol{h_i}|)/(\mathbf{min}|\boldsymbol{h_i}|) \approx 2^{16}$ versus 2^8). On the other hand, the second approach achieves a higher resolution within the range by assigning more bits to m_i (i.e., $\Delta m_i = 0.5$ versus 0.25). Thus, how to best allocate the bits is determined by the distribution of h_i; as a rule of thumb, we start by allocating just enough bits for s_i to cover the full range of h_i (i.e., $(\mathbf{max}|\boldsymbol{h_i}|)/(\mathbf{min}|\boldsymbol{h_i}|)$), and then allocate bits to m_i to achieve the highest bin resolution

$$l_i = 1'b0;\; s_i = 4'b0110;\; \hat{m}_i = 1'b1 \tag{6}$$

$$l_i = 1'b0;\; s_i = 3'b110;\; \hat{m}_i = 2'b10. \tag{7}$$

Now, let's explore the benefits afforded by such a mixed fixed-/-floating-point representation. The first benefit, first seen in Figure 1, is shown more explicitly through quantization-error histograms in Figure 2. Here, we consider two types of FIR filters that are used within a seizure-detection system [3], which is employed later in the experimental demonstration for detailed analysis of the approach. The system extracts spectral-energy features from electroencephalogram (EEG) data sampled at 256 Hz. Since seizure-detection features correspond to low-frequency ranges, the system employs front-end decimation, implemented using a 256-order low-pass FIR filter. Following this, eight 64-order band-pass FIR filters (centered from 0 to 21 Hz) are used to isolate the signal in 3-Hz bands, for subsequent energy accumulation over a 2-second epoch. Then, the spectral-energy features from three consecutive epochs are combined to form the overall feature vector. Note that only one band-pass feature-extraction filter is considered for illustration, though all have similar coefficient distributions

As seen in Figure 2, floating-point representation has a quantization error with greatest density at values well below 1 least significant bit (LSB), while fixed-point representation has a quantization error more evenly distributed up to 1 LSB. However, we also see that, with fixed-point representation, the quantization error is limited to 1 LSB; on the other hand, with floating-point representation, the quantization error can exceed 1 LSB. Indeed, for the examples shown, there are many cases where the errors far exceed 1 LSB. This can lead to a significant error. To address this, the error can be further minimized thanks to a simple optimization enabled by the floating-point representation. This is described next, using the concrete example of the seizure-detection system to illustrate the approach and its rationale.

However, before proceeding, we highlight an additional benefit of floating-point representation, which is that it leads to an implementation for multiplier hardware that consumes less energy and area. The reason is that input signals now only need to multiply with $\hat{m}_i$, which is represented by fewer bits compared to h_i. The remaining operations are simply barrel shifting and sign application, which can be trivially implemented. Given the prominence of multiplication operations in DSP systems, this can present a significant advantage.

Optimizing the quantization error

This section discusses the optimization enabled by floating-point representation to further reduce the quantization error. The opportunity arises from the fact that, in DSP systems, outputs can often be scaled arbitrarily. For instance, in the case of fixed-point representation, this is

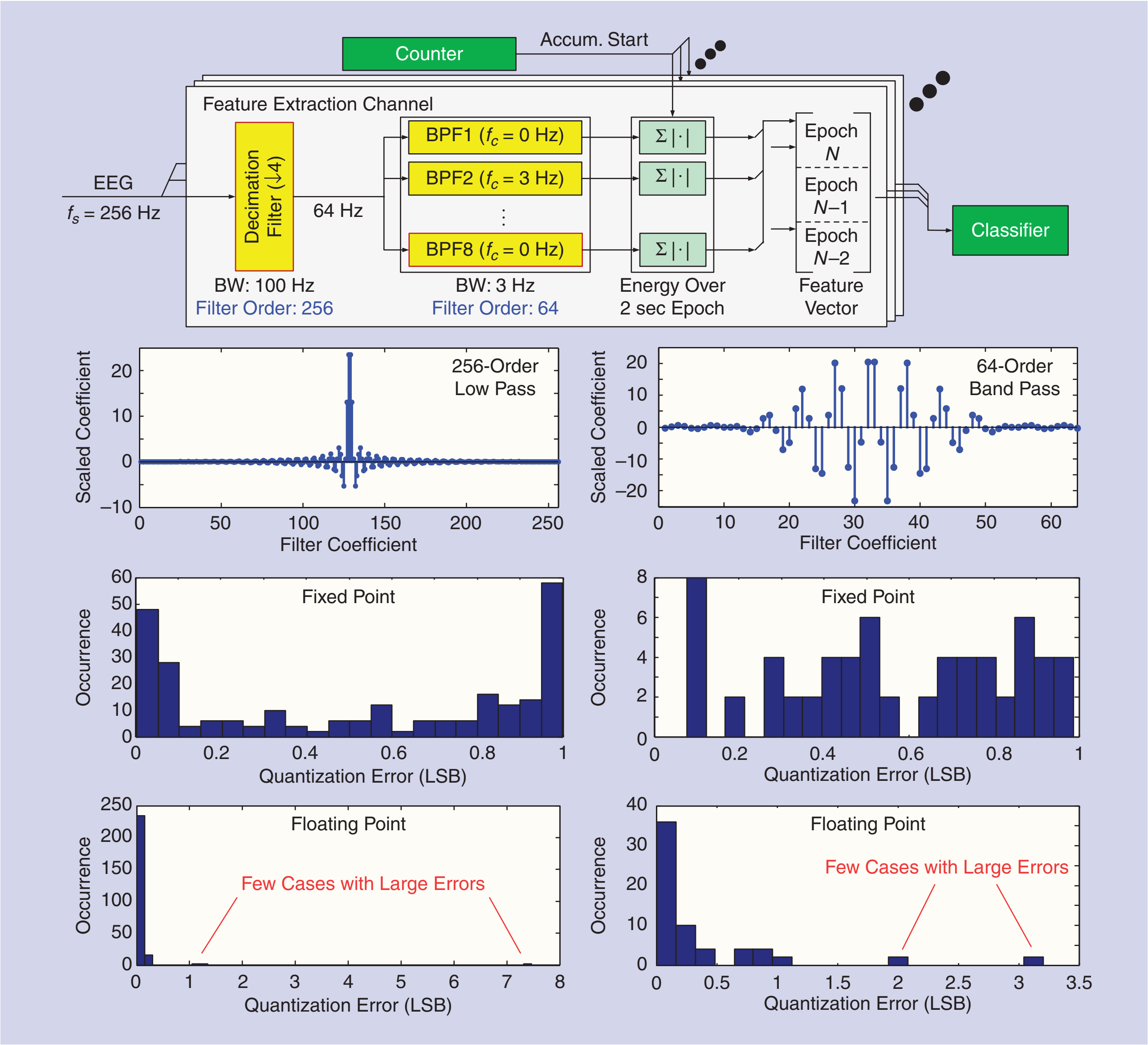

FIGURE 2. A comparison of quantization error for fixed-point and floating-point quantization of FIR filter coefficients. FIR filters are from an EEG-based seizure-detection application consisting of low-pass decimation and band-pass feature extraction.

often done to best utilize the available dynamic range of the system. On the other hand, in the case of a floating-point representation, which affords representation of very large-valued numbers, let's consider how multiplication of all h_i values by a single positive factor α improves utilization of dynamic range, but in a different way. The approach is shown in Figure 3. With floating-point representation of the quantization levels, multiplication by α can be used to move the h_is closer to the quantization levels. This is made possible because, with the large range that can be covered using floating-point representation, we can ensure that all values remain within the representation bounds even after the multiplication. With this ensured, the question now is simply how to find an optimal α that leads to minimum output error $\varepsilon_{\hat{y}}$ for the inner-product operation.

With the scaling parameter α applied to each h_i, the resulting outputs y can now be expressed as in (8). Quantization of αh_i ($\widehat{\alpha h_i}$) then affects the output value as shown in (9). Thus, the output error caused by quantization of h_i can be expressed as in (10). Notice here h_i is assumed to be predefined values, while x_i is assumed to be drawn from a statistical distribution corresponding to the input X_i. As a result, output error $\varepsilon_{\hat{y}}$ is also a statistical distribution, which is parameterized by α.

$$y(\alpha) = \sum\nolimits_i (\alpha h_i) \cdot x_i \tag{8}$$

$$\hat{y}(\alpha) = \sum\nolimits_i \widehat{(\alpha h_i)} \cdot x_i \tag{9}$$

$$\varepsilon_{\hat{y}} = \frac{\left|\sum_i (\alpha h_i) x_i - \sum_i \widehat{(\alpha h_i)} x_i\right|}{\alpha}. \tag{10}$$

To find the optimal α that minimize output error $\varepsilon_{\hat{y}}$, we proceed by first finding the distribution of output error $E_{\hat{Y}}$. This requires us to know the distribution of inputs X_i. Note here that capital letters are used for representing the distribution of signal samples, rather than a signal. For simplifying our discussion without loss of generality, we present the optimization approach through an example. We focus on FIR filtering required within the EEG-based seizure-detection system [3], which allows us to specify X_i for the EEG data. For other applications, the same approach may be applied but with a corresponding assumption on the statistical distribution X_i.

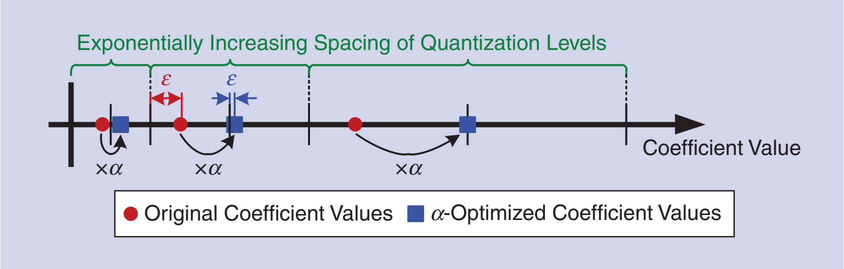
Exponentially Increasing Spacing of Quantization Levels
ε
ε
×α
×α
×α
Coefficient Value
● Original Coefficient Values
■ α-Optimized Coefficient Values

FIGURE 3. Introducing the scaling parameter α to enable coefficient optimization.

First, for FIR filtering, we assume the distribution of X_i over i is identical as all samples are drawn from the same signal source. We also make the assumption here that all X_i is independently distributed. Notice that independence may not hold in many cases. Nonetheless, the assumption is made for convenience of the derivation, and its validity is later verified through simulation of a practical application. The distribution of EEG samples (derived from actual patient signals) is shown in Figure 4(a). We model samples of the EEG data by a normal distribution $X_i \sim N(0, \sigma^2)$. The variable of interest, $\boldsymbol{E}_{\hat{Y}}$, can be derived as shown in (11). Note that we deliberately leave out the scaling parameter α here for simplicity of expression. With the previously stated assumptions on the input X_i, the distribution of $\boldsymbol{E}_{\hat{Y}}$ can be derived, as shown in (12). As seen, the output error is also normally distributed, with zero mean and variance dependent on the squared sum of the error on all h_i. Notice, it can be concluded from the previous derivations that the output $\hat{Y}$ itself also follows a zero-mean normal distribution. This implies that the same analysis can be applied in a scenario where inner-product operations are cascaded

$$\begin{aligned} E_{\hat{Y}} &= Y - \hat{Y} = \sum\nolimits_i h_i \cdot X_i - \sum\nolimits_i \hat{h}_i \cdot X_i \\ &= \sum\nolimits_i (h_i - \hat{h}_i) \cdot X_i \ = \ \sum\nolimits_i \varepsilon_{\hat{h}_i} \cdot X_i \end{aligned} \tag{11}$$

$$E_{\hat{Y}} \sim N(0, \ \sum\nolimits_i \varepsilon_{\hat{h}_i}{}^2 \sigma^2). \tag{12}$$

Using (12), a reasonable objective function $\boldsymbol{C_{\hat{h}}(\alpha)}$ would be one as shown in (13) that minimizes the variance of output error over possible values of the scaling factor α. Within this function, $\boldsymbol{\varepsilon_{\hat{h}_i}}$ can be derived as shown in (14), where l_i, s_i, and m_i can be directly specified from h_i as previously shown in (3)–(5), and $\hat{m}_i$ can be obtained from quantizing m_i, as in (15), with k being the quantization level. Consequently, for a given set of h_i, $\varepsilon_{\hat{h}_i}$ is only a function of α. Note, however, this function is not convex. It is not even continuous due to the quantization operation applied to $\hat{m}_i$. Fortunately, there is a structure to this function, making the optimization problem trivial to solve.

$$\min_{\alpha} C_{\hat{h}}(\alpha), \ \alpha \in R^+, \qquad \text{where } C_{\hat{h}}(\alpha) = \sum\nolimits_i \varepsilon_{\hat{h}_i}{}^2(\alpha) \tag{13}$$

$$\begin{aligned} \varepsilon_{\hat{h}_i} &= h_i - \hat{h}_i = l_i \cdot 2^{s_i} \cdot (1 + m_i) \\ &\quad - l_i \cdot 2^{s_i} \cdot (1 + \hat{m}_i) \\ &= l_i \cdot 2^{s_i} \cdot (m_i - \hat{m}_i) \end{aligned} \tag{14}$$

$$\hat{m}_i = \lfloor m_i \cdot 2^k \rfloor / 2^k. \tag{15}$$

To see the structure, (16) modifies (14) by applying the α scaling factor. After plugging in (3)–(5) and (15), it can be seen that α only appears in the cost function in the form of $f(\alpha)$, shown in (17). From this, it can be easily seen that $f(\alpha) = f(2\alpha)$ for $\forall \alpha \in R^+$. This indicates that the cost function in

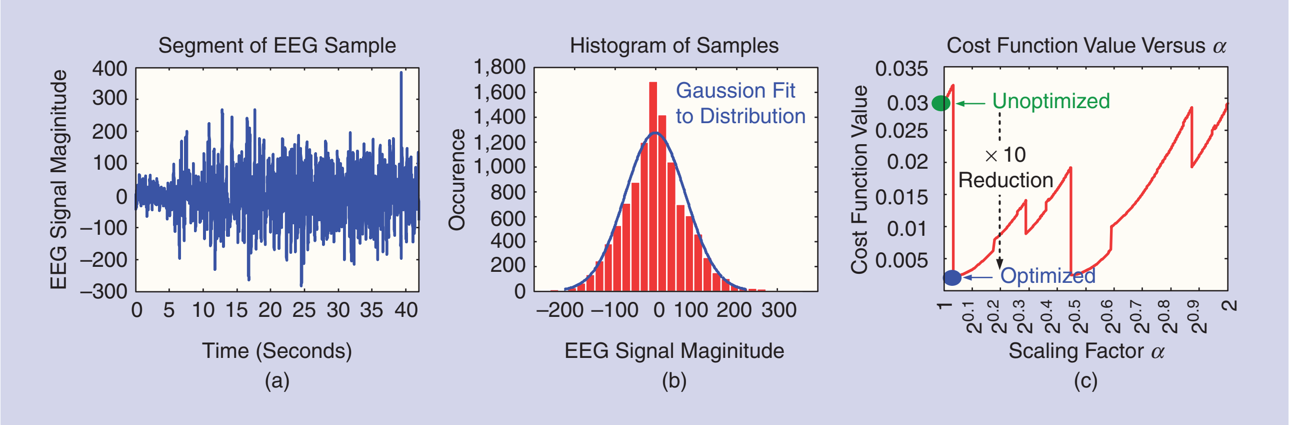

FIGURE 4. The distribution of EEG signals can be modeled as drawn from a Gaussian distribution, as shown by (a) representative time samples from a segment of an EEG channel and (b) the histogram of sample values for the segment, leading to (c) the cost function shown for the α-optimization.

(13) is a repeating function of α. As a result, a global minimum of $C_{\hat{h}}(\alpha)$ exists in the range $[1,2)$. Thus, the optimization is easily solved numerically by sweeping α within this range. For the filter considered, Figure 4 shows the resulting cost function and how its value is improved by the optimal choice of α

$$\varepsilon_{\hat{h}_i}(\alpha) = \frac{\alpha h_i - \widehat{\alpha h_i}}{\alpha} = l_i \cdot \frac{2^{s_i(\alpha)}}{\alpha} \cdot [m_i(\alpha) - \hat{m}_i(\alpha)] \quad (16)$$

$$f(\alpha) = 2^{\lfloor \log_2(\alpha|h_i|) \rfloor} / \alpha. \quad (17)$$

Summary of approach

Below, the specific procedure for applying the optimization generally to minimize quantization error is provided.

1) Select a precision requirement for h_i (based on preferred tradeoff of computational accuracy and hardware complexity), e.g., 6 bits in our application.
2) Perform floating-point quantization of h_i [assigning bits to significand and exponent based on the corresponding dynamic range $(\max|h_i|)/(\min|h_i|)$].
3) Determine the scaling parameter α.
   a) Model the statistical distribution of inputs X_i.
   b) Compute the distribution of output error $\varepsilon_{\hat{y}}$ due to quantization of h_i [see (10)].
   c) Define a cost function based on the distribution of output error $\varepsilon_{\hat{y}}$.
   d) Minimize the cost function to find optimal α.
4) Redetermine the quantization for h_i with the scaling parameter α.
5) Only if the absolute value of output y is critical, apply multiplication by $1/\alpha$ to the outputs.

Note that, when required, the energy of Step 5 is greatly amortized since it involves one multiplication, compared to many in the preceding convolution operation.

Experimental demonstration

For demonstration, the presented approach is applied to the EEG-based seizure-detection system of Figure 2. For analysis and demonstration of the approach, three hardware implementations of filters are compared, each employing 6–10 bits for representing h_i. The first implementation is a conventional fixed-point FIR filter. The second implementation is the mixed fixed-/floating-point FIR filter without any α optimizaton. For this, s_i is assigned 4 b for the decimation filter and 3 b for the band-pass filter to address the required range of h_i $((\mathbf{max}|\boldsymbol{h_i}|)/(\mathbf{min}|\boldsymbol{h_i}|) \approx 37\,k$ for the decimation filter, and 500 for the band-pass filter). m_i is assigned correspondingly for 6–10 bit overall precision of h_i. The third implementation is the presented mixed fixed-/floating-point FIR filter with α optimization. Simulation results are presented using 210 seconds of EEG data (sampled at 256 Hz). All implementations are developed in both MATLAB and Verilog RTL. For analysis of quantization error, the MATLAB implementations are used. For hardware energy and area analysis, the Verilog implementations are synthesized to standard cells and laid out in a 32-nm complementary metal–oxide–semiconductor technology, with transistor-level postlayout simulations performed using NanoSim.

The results in Figure 5 focus on quantization errors. The histograms show the quantization error distributions for the decimation filter and band-pass filter after applying the optimization. Comparing Figure 5 to Figure 2, we can see that with α optimization applied to the mixed fixed-/floating-point approach, the quantization errors are more densely clustered close to zero. The output error resulting from the three implementations for 6-bit coefficient representation are also compared. The samples are ordered based on the magnitude of

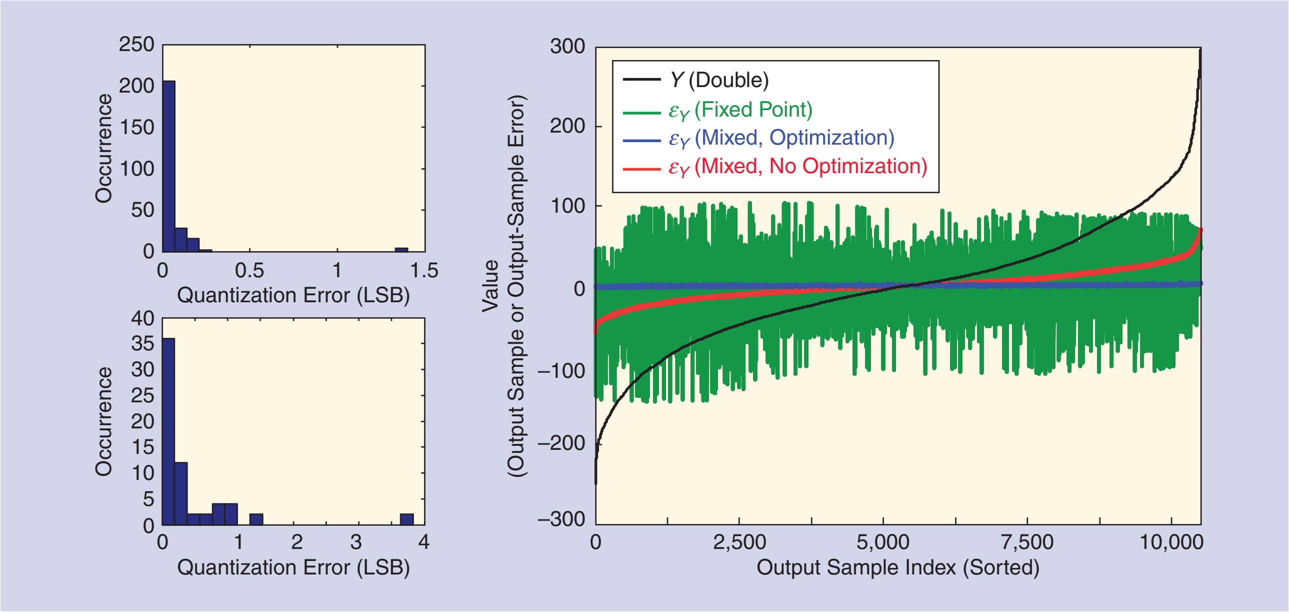

FIGURE 5. The reduction of quantization error by applying the presented optimization.

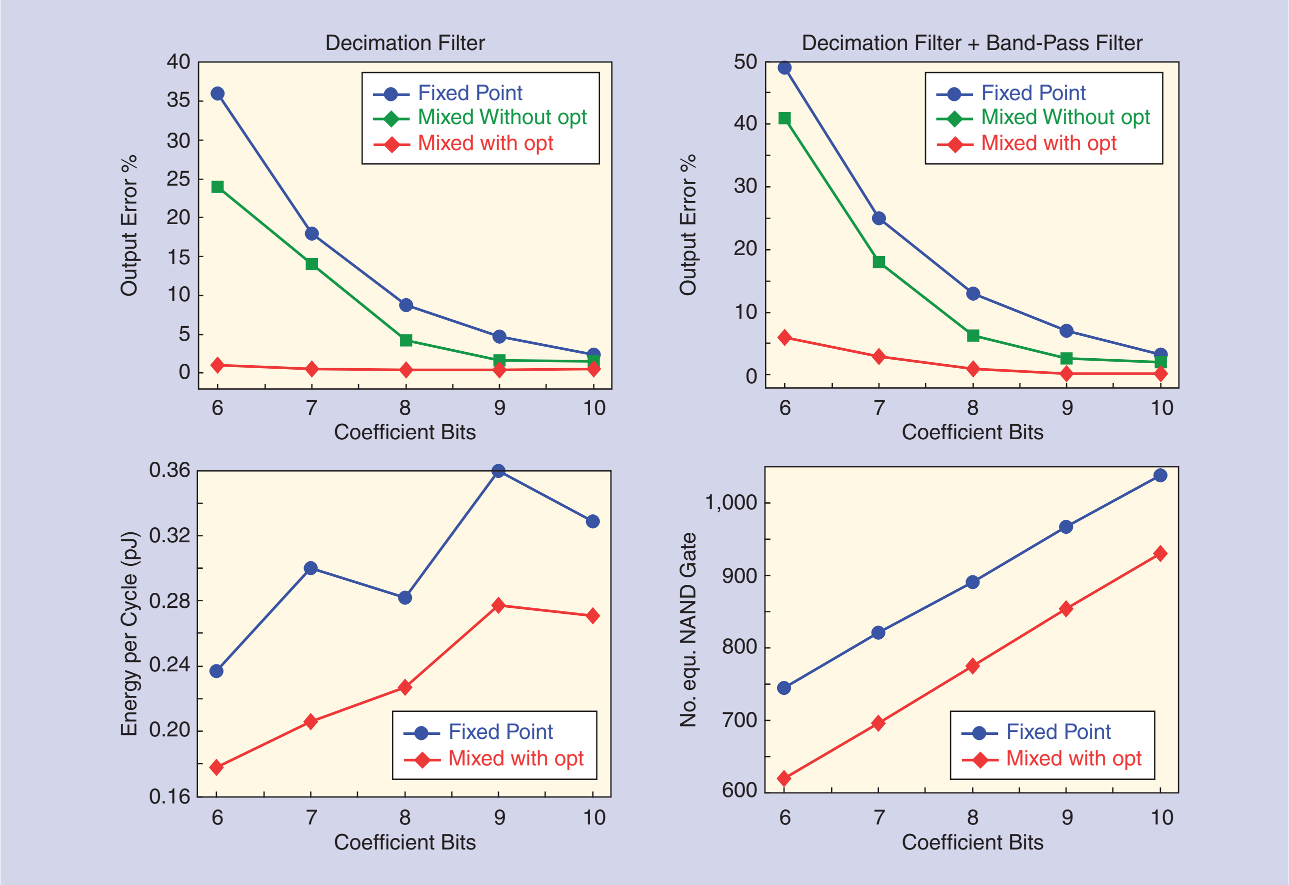

FIGURE 6. A comparison of approaches in terms of quantization error, energy, and hardware complexity.

the output values to aid visualization. As seen, fixed-point coefficient representation results in substantial error. While floating-point coefficient representation reduces this, the error is substantially reduced even further thanks to α optimization.

Figure 6 provides an overall summary, showing the output error (due to quantization), computational energy, and hardware area, versus bit level for representing h_i, for the three implementations. The output error corresponds to the root-mean-square error of the output computed with 6-bit quantized filter coefficients h_i versus double-precision h_i, normalized to the root-mean-square value of the output computed with double-precision h_i. The computational energy is the average energy per inner-product operation determined from postlayout transistor-level (NanoSim) simulation of the accelerator, and the hardware area is represented as the equivalent number of NAND gates for the implementation (note that the energy and area of the mixed fixed-/floating-point representation implementation is the same with and without α optimization). The presented implementation with optimization leads to a substantially lower error than an implementation based on fixed-point coefficient representation, especially at low bit levels. For example, for the decimation filter alone, the presented approach leads to a 37× error reduction compared to a conventional fixed-point implementation at the 6-bit level. When considering cascading of the decimation filter and band-pass filter, the corresponding error reduction is 28×.

> **The energy-precision tradeoff associated with quantization in a DSP system has always been a critical concern during design.**

Comparing the energy consumption and hardware complexity, for the fixed-point and mixed fixed-/floating-point implementations, the presented approaches also lead to hardware-resource savings in addition to reducing errors (note, the computational energy and hardware area with and without α optimization are equivalent). As an example, at the 6-bit level, the presented approach leads to 1.4× energy savings and 1.2 × area savings compared to conventional fixed-point implementation. Indeed, that approach is shown to yield such advantages in a broad range of system implementations, involving convolution operations for various tasks (e.g., [4] presents its use for optimizing the implementation of a linear classifier).

Summary

The energy-precision tradeoff associated with quantization in a DSP system has always been a critical concern during design. Considering the

importance of inner-product operations, this article focuses on an approach by which this tradeoff can be substantially improved. The first aspect of the approach is the use of a mixed fixed-/floating-point representation, by which the dynamic range is more efficiently used. The second aspect is a simple optimization enabled by the representation whereby quantized coefficients can all be scaled to minimize the overall error incurred. Filters within an EEG-based seizure-detection system are used to demonstrate the approach. Substantial reduction in the error, computational energy, and hardware area is observed as a result of the approach.

Authors

Zhuo Wang (zhuow@princeton.edu) received his B.S. degree in microelectronics and his M.S. degree from Peking University, Beijing, China, in 2011 and 2013, respectively. Currently, he is working toward his Ph.D. degree in the Department of Electrical Engineering, Princeton University, New Jersey. His research focuses on leveraging statistical approaches such as machine learning for achieving hardware relaxation in an algorithmic and architectural level in resource-constrained platforms such as embedded sensing systems.

Jintao Zhang (jintao@princeton.edu) received his B.S. degree in electrical engineering from Purdue University, West Lafayette, Indiana, in 2012. Currently, he is working toward his Ph.D. degree at Princeton University, New Jersey. His research is focused on energy-efficient signal analysis (mainly classification and on chip). His primary research interests are very-large-scale integration system design, exploring mixed signal circuit design by using digital controls to enable various analog functionalities of the complementary metal–oxide–semiconductor and, thus, enhancing the system performance with both the benefit from analog circuits and digital circuits.

Naveen Verma (nverma@princeton.edu) received his B.A.S. degree in electrical and computer engineering from the University of British Columbia, Vancouver, Canada, in 2003, and his M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 2005 and 2009, respectively. Currently, he is an associate professor of electrical engineering at Princeton University, New Jersey, where he has been since 2009. His research focuses on advanced sensing systems, including low-voltage digital logic and static random access memories, low-noise analog instrumentation and data conversion, large-area sensing systems based on flexible electronics, and low-energy algorithms for embedded inference, especially for medical applications.

References

[1] Z. Wang, J. Zhang, and N. Verma, "Reducing quantization error in low-energy FIR filter accelerators," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Brisbane, Australia, 2015.

[2] D. Montgomery, E. Peck, and G. Vining, *Introduction to Linear Regression Analysis.* Hoboken, NJ: Wiley, 2015.

[3] A. Shoeb, "Application of machine learning to epileptic seizure onset detection and treatment," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, 2009.

[4] Z. Wang, J. Zhang, and N. Verma, "Realizing low-energy classification systems by implementing matrix multiplication directly within an ADC," *IEEE Trans. Biomed. Circuit. Syst.*, vol. 9, no. 6, pp. 825–837, 2015.

SP

LECTURE NOTES

(continued from page 140)

Mathematical Optimization Society. His current research is centered around deriving efficient optimization methods for large-scale data analysis, with an emphasis on techniques for sparse and low-rank models.

References

[1] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering.* Philadelphia: SIAM, 2006.

[2] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, Mar. 2009.

[3] *MATLAB and Wavelet Toolbox Release R2015b.* Natick, MA: The MathWorks, Inc., 2015.

[4] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way.* London, U.K.: Elsevier, 2009.

[5] G. Strang and T. Nguyen, *Wavelets and Filter Banks.* Cambridge, MA: Wellesley-Cambridge, 1996.

[6] A. Weber, "The USC-SIPI image database: Version 5," USC-SIPI Tech. Report 315, Univ. of Southern California, Los Angeles, 1993.

[7] S. Rout, "Orthogonal vs. biorthogonal wavelets for image compression," Ph.D. dissertation, Dept. Electrical and Computer Engineering Virginia Tech, Blacksburg, VA, 2003.

[8] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[9] A. Ahmed, B. Recht, and J. Romberg, "Blind deconvolution using convex programming," *IEEE Trans. Inform. Theory,* vol. 60, no. 3, pp. 1711–1732, 2014.

[10] J. F. Claerbout, *Earth Soundings Analysis: Processing Versus Inversion*, vol. 6, Cambridge, MA: Blackwell, 1992.

[11] B. Rideout and E. M. Nosal. Personal communication.

[12] M. Schmidt, "Graphical model structure learning with l1-regularization," Ph.D. dissertation, Dept. Computer Science, Univ. British Columbia, Vancouver, 2010.

SP