Towards Optimal Quantization of Neural Networks

Avhishek Chatterjee and Lav R. Varshney
University of Illinois at Urbana-Champaign

***Abstract*—Due to the unprecedented success of deep neural networks in inference tasks like speech and image recognition, there has been increasing interest in using them in mobile and in-sensor applications. As most current deep neural networks are very large in size, a major challenge lies in storing the network in devices with limited memory. Consequently there is growing interest in compressing deep networks by quantizing synaptic weights, but most prior work is heuristic and lacking theoretical foundations. Here we develop an approach to quantizing deep networks using functional high-rate quantization theory. Under certain technical conditions, this approach leads to an optimal quantizer that is computed using the celebrated backpropagation algorithm. In all other cases, a heuristic quantizer with certain regularization guarantees can be computed.**

***Index Terms*—deep neural network; quantization theory**

I. Introduction

Deep neural networks have revolutionized speech processing, image recognition, and many other inference tasks such as regression where the natural performance criterion is mean squared error (MSE). Current deep networks are enormous cloud-based structures that cannot fit on devices at network edge. This prevents real-time inference for internet of things (IoT) applications due to latency in going back and forth with the cloud, limiting their use in addressing information overload. Hence, hardware implementations of neural networks for mobile, in-sensor, and in-memory inference have drawn significant attention [1]–[4], e.g. in designing special-purpose chips [5]–[7]. As these computing units have limited computational and storage capacity, finding compact but accurate representation of deep networks is now a necessity.

Compact representation of deep networks has attracted recent interest [8]–[10]; an accurate representation of a deep network must encode both the connectivity pattern of synapses and the weights of those synapses. It is clear that the best compact representation should jointly compress the graph structure and the edge weights, but this is extremely challenging and further may yield representations that cannot directly be used for inference without decoding. Here we focus on fixed-rate scalar quantization of edge weights that can be used directly for inferential computation and discuss compression of network structure elsewhere [11].

Quantization of edge weights has been explored, either in training neural networks with quantized weights [8], [12], [13] or in quantizing the learned weights after training [9], [14]. Techniques include optimal uniform fixed-point quantization [14], hashing-based quantization [9], and quantization heuristics based on information-theoretic principles, whether through vector quantization [10] or Huffman coding [15]. A formal treatment of neural network quantization, which also recognizes representing weights for their own sake is not the same as for inference purposes, is lacking.

Here we develop a technique to optimally quantize learned weights for approximating real-valued functions by deep feedforward neural networks based on high-rate quantization theory [16]. Neural networks can be viewed as functions (possibly not known explicitly) of edge weights parametrized by input variables which are random. This motivates us to take an approach similar to functional quantization for parallel trees [17] or sequences [18]. Some key differences from that work include the facts: the functional form that the network approximates may not be known; the network output is a function of weights parametrized by *random* network inputs; the weights are placed in a directed acyclic graph rather than as inputs to a tree; and in realizing a particular function, the weights of different edges are independent.

Before considering approximating a particular function, we first consider approximating randomly chosen functions from a class of functions by developing optimal quantizers through connections to distributed functional quantization [17]. In the main setting of a particular function, unlike variates in distributed functional quantization, edge weights are not independent and the impact of quantization errors on the network output are modulated by the random inputs to the network. To address this new mathematical challenge, we start from first principles to derive an optimal quantizer. Under some technical conditions, optimality is not possible but the theoretical approach nevertheless yields a heuristic design that is provably close to optimal quantizers in relative entropy.

Some simulation evaluations compare our approach with naïve quantization, but detailed implementation on real-world networks is left for future work beyond this short paper.

II. Weight Quantization for a Class of Functions

Although two-layer neural networks with a continuous neuron activation function can approximate any continuous function by varying the number of neurons, the connectivity pattern between the two layers, and the edge weights [19], L-layer neural networks with a *given* connectivity pattern between consecutive layers can also realize a broad class of functions by varying just the edge weights.

Indeed in many modern applications, edge weights are trained within a given neural network structure to learn a

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 and in part by the IBM-Illinois Center for Cognitive Computing Systems Research (C3SR), a research collaboration as part of the IBM Cognitive Horizons Network.

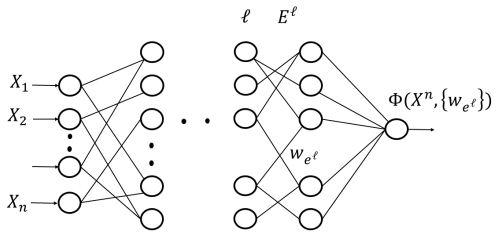

Fig. 1. A deep neural network.

function from a class of possible functions and these learned weights are stored for future computation. In such scenarios like regression, an important performance criterion for edge weight quantization is to have small MSE over the class of functions from the application domain.

As dictated by the application, consider the class of functions of interest $\mathcal{C}$ with $n \in \mathbb{Z}_+$ inputs and a probability measure μ on the class. Let the input variables $X_1, X_2, \ldots, X_n$ be distributed independently in $[0,1]$ and the functions in $\mathcal{C}$ be realizable by bounded edge weights. With this property, one can emulate a large class of distributions over $\mathcal{C}$ by choosing appropriate edge weight distributions, so we proceed in terms of these edge weight distributions rather than μ. As in Fig. 1, let the edges between layer ℓ and $\ell+1$ be E^ℓ and the density function of weight w_{e^ℓ} for edge $e^\ell \in E^\ell$ be p_{e^ℓ} on $[0,1]$.

Consider quantizing weights of different edges with possibly different quantizers. Let a choice of weights $\{w_{e^\ell}\}$ realize function $f \in \mathcal{C}$. For inputs $x^n = (x_1, x_2, \ldots, x_n)$ the output of the neural network is $\Phi(x^n, \{w_{e^\ell}\})$, which equals $f(x^n)$. Under quantizers $\{Q_{e^\ell}\}$ for edges $\{e^\ell\}$, the output of the neural network is $\Phi(x^n, \{Q_{e^\ell}(w_{e^\ell})\})$. We aim to choose $\{Q_{e^\ell}\}$ to minimize MSE over the function class and all inputs:

$$\mathbb{E}_{\{w_{e^\ell}\}}\mathbb{E}_{X^n}\left(\Phi(x^n,\{w_{e^\ell}\}) - \Phi(x^n,\{Q_{e^\ell}(w_{e^\ell})\})\right)^2.$$

Given the number of quantization points K for edge weights, the goal is to find partitions and representation points in $[0,1]$; we use techniques from high-rate quantization theory for large K. At high rate, the problem of partitions and point placement is posed as choosing an optimal quantizer point density function $\lambda : [0,1] \to \mathbb{R}_+$, $\int \lambda(w)dw = 1$, where $\lambda(w)\delta w$ is the number of quantization points to be placed around $w \in [0,1]$ and thus yields a companding quantizer strategy, cf. [17].

The optimal point density function for each edge weight in a neural network governed by $\{p_{e^\ell}\}$ is as follows.

Proposition 1: The point density functions

$$\lambda_{e^\ell} = \frac{(g_{e^\ell}\ p_{e^\ell})^{1/3}}{\int (g_{e^\ell}\ p_{e^\ell})^{1/3}},$$

for $g_{e^\ell}(w_{e^\ell}) = \mathbb{E}_{X^n}\mathbb{E}_{\{w_{e^\ell}\}}\left[(\partial\Phi(x^n,\{Q_{e^\ell}(w_{e^\ell})\}/\partial w_{e^\ell})^2 | w_{e^\ell}\right]$, for edges $\{e^\ell\}$ minimize the MSE value $\mathbb{E}_{\{w_{e^\ell}\}}\mathbb{E}_{X^n}\left(\Phi(x^n,\{w_{e^\ell}\}) - \Phi(x^n,\{Q_{e^\ell}(w_{e^\ell})\})\right)^2$.

Proof: Follows by reusing techniques from [17, Thm. 13], as Φ is a function of $\{w_{e^\ell}\}$, parameterized by X^n. ∎

An important concern is how to obtain $g_{e^\ell}(w_{e^\ell})$ for a neural network. Unlike functional quantization [17], this may not have an explicit form, but we give efficient algorithms in Sec. III. Another issue is that we may have different quantizers for different edges. If, however, a neural network has regular and symmetric connections with the same p_{e^ℓ} for all edges, $\{g_{e^\ell}\}$ turns out to be the same for all edges. Hence, in the commonplace practical setting of symmetric networks, a single quantizer can optimally quantize all the edge weights. Is it, however, sufficient to use a naïve uniform quantizer for symmetric networks? The following analytical examples illustrate the importance of optimal non-uniform quantizers even when the neural network is symmetric.

Example 1: Consider a simple three-layer neural network with two input neurons, ten neurons in the hidden layer, and one output neuron. Let the connections between the first two layers and the last two layers both be complete bipartite graphs. Let us fix all weights between the last two layers at unity and consider the function class that can be realized by varying weights between the first two layers. For the rectified linear unit (ReLU) activation function at the neurons and any arbitrary distribution of the input variables, we compare optimal quantization to ordinary uniform quantization when edge weights are distributed as $p(w)$. The optimal quantizer turns out to be $(p(w))^{\frac{1}{3}} / \int (p(w))^{\frac{1}{3}} dw$ and the MSE for the optimal quantizer is $\frac{|E|\mathbb{E}[X^2]}{24}\left(\int (p(w))^{\frac{1}{3}} dw\right)^3$, where E is the set of edges.

Consider $p(w)$ to be the Kumaraswamy distribution [20] with both parameters 1, i.e., $\mathfrak{K}(1,1)$; the optimal quantizer is uniform. For $\mathfrak{K}(4,1)$, uniform is not optimal and the ratio of MSE between optimal quantizer and uniform quantizer is $\frac{1}{2}$, for $\mathfrak{K}(7,1)$ the ratio is $\frac{7}{27}$, and for $\mathfrak{K}(10,1)$ the ratio is $\frac{5}{32}$. As i increases $\mathfrak{K}(3i+1,1)$ has increasing mass at higher values, and in general, for $\mathfrak{K}(3i+1,1)$ the ratio of MSE between optimal and uniform quantizers is $\frac{3i+1}{(i+1)^3}$ which quickly goes to zero.

Example 2: Now consider another three-layer example with two input neurons, two hidden neurons, and one output neuron, with quadratic activation function $f(x) = x^2$. Inputs and weights (w) are distributed independently as $p(\cdot)$ on $[0,1]$. We could analytically compute the optimal quantizer and the optimal MSE to be $(c_0 + c_1 w + c_2 w^2 + c_3 w^3)^{\frac{1}{3}}(p(w))^{\frac{1}{3}} / \int (c_0 + c_1 w + c_2 w^2 + c_3 w^3)^{\frac{1}{3}}(p(w))^{\frac{1}{3}} dw$ and $\frac{1}{24}\left(\int (c_0 + c_1 w + c_2 w^2 + c_3 w^3)^{\frac{1}{3}}(p(w))^{\frac{1}{3}} dw\right)^3$, respectively for $c_0 = m_3^2 m_1 + m_2^2 m_1^2 + m_1 m_2 m_3$, $c_1 = 12m_2^3 + 4m_2^2 m_4 + 8m_1^3 m_3 + 4m_2^3 + 8m_1^2 m_2^2$, $c_2 = 12m_1^2 m_3$, and $c_3 = m_4$, where $m_k = \int w^k p(w) dw$ for $k \in \mathbb{Z}_+$. For $\mathfrak{K}(4,1)$ this ratio is 0.424.

These simple analytical examples for symmetric networks imply that the gain of optimal quantizer over ordinary quantizer increases rapidly as the weight distributions become more non-uniform and skewed. Here Kumaraswamy distribution was chosen for its analytical tractability and its ability to capture a wide variety of shapes of density functions through the variations of its parameters.

III. Weight Quantization for a Specific Function

Optimal quantizers in Sec. II guarantee minimum error over a class of functions. If a neural network is used for a *specific* inference problem, we want the stronger guarantee that there is small quantization error for a specific function corresponding to the inference problem. After a neural network learns to realize a specific function, weights on individual edges are fixed and hence the approach in Sec. II that model edge weights as coming from a distribution is not applicable. Also, in practice one would like to have the same quantization strategy for all edges to allow circuit implementations of inferential arithmetic. To achieve these goals, we need a high-rate quantization theory specific to neural networks and differing significantly in its mathematical development from the theory of distributed functional quantization [17].

When the number of quantization points K is high, for any edge e^ℓ, $|w_{e^\ell} - Q(w_{e^\ell})|$ is small. Hence, using a Taylor series approximation, for any smooth differentiable Φ at large K:

$$\begin{aligned}&\Phi(x^n, \{w_{e^\ell}\}) - \Phi(x^n, \{Q(w_{e^\ell})\}) \\ &= \sum_{e^\ell} \frac{\partial \Phi(x^n, \{Q(w_{e^\ell})\})}{\partial w_{e^\ell}} \left(w_{e^\ell} - Q(w_{e^\ell})\right) + o\left(\frac{1}{K}\right).\end{aligned}$$

Our goal is to find the best Q that minimizes an appropriate error metric. In this paper we use MSE

$$\mathbb{E}_{X^n}\left(\sum_{e^\ell} \frac{\partial \Phi(X^n, \{Q(w_{e^\ell})\})}{\partial w_{e^\ell}} \left(w_{e^\ell} - Q(w_{e^\ell})\right)\right)^2. \quad (1)$$

Let the representation points in $[0,1]$ for a given quantizer Q be $a_1, a_2, \ldots, a_K$. For Q, the error in quantizing a weight w depends only on w, which we denote by $\Delta(w)$. Thus,

$$\begin{aligned}&\Phi(x^n, \{w_{e^\ell}\}) - \Phi(x^n, \{Q(w_{e^\ell})\}) \quad (2)\\ &= \sum_{e^\ell} \frac{\partial \Phi(x^n, \{Q(w_{e^\ell})\})}{\partial w_{e^\ell}} \left(w_{e^\ell} - Q(w_{e^\ell})\right) \\ &= \sum_{v=0}^{V} \Delta(w) \sum_{e^\ell : w_{e^\ell} \in ((k-1)\frac{1}{V}, k\frac{1}{V}]} \frac{\partial \Phi(x^n, \{Q(w_{e^\ell})\})}{\partial w_{e^\ell}},\end{aligned}$$

for any $V \in \mathbb{Z}_+$. We make the following assumptions to derive an optimal high rate quantizer.

Assumption 1: $V \sum_{e^\ell : w_{e^\ell} \in ((k-1)\frac{1}{V}, k\frac{1}{V}]} \frac{\partial \Phi(\mathbf{X}, \{Q(w_{e^\ell})\})}{\partial w_{e^\ell}}$ converges almost surely (over the probability space of X^n) to a differentiable function $G_{\mathbf{X}}(w)$ as $V \to \infty$.

Assumption 2: We assume large network and high rate of quantization, i.e. $K, n \to \infty$, but $K \ll n$.

Assumption 1 holds for neural networks with many edges if the empirical distribution of edge weights approximates a smooth probability density function (pdf). Unlike Sec. II, the theory developed here is valid only for quantizing neural networks with many synapses. So, (2) can be written as $\int \Delta(w) G_{\mathbf{X}}(w) dw$. Defining $a_0 = 0$ and $a_{K+1} = 1$, another way to write (2) is

$$\sum_{k=1}^{K+1} \int_{w \in (a_{k-1}, a_k]} \Delta(w) G_{\mathbf{X}}(w) dw.$$

Consider $\int_{w \in (a_{k-1}, a_k]} \Delta(w) G_{\mathbf{X}}(w) dw$ for $2 \le k \le K$. Define $\bar{a}_k := \frac{a_{k-1} + a_k}{2}$ and $\delta_k := \frac{a_k - a_{k-1}}{2}$. Over $(a_{k-1}, \bar{a}_k)$ error $\Delta(w)$ is positive and changes from 0 to δ_k, whereas over $(\bar{a}_k, a_k)$ error $\Delta(w)$ is negative and changes from $-\delta_k$ to 0. This observation yields the following steps.

$$\begin{aligned}&\int_{w \in (a_{k-1}, a_k]} \Delta(w) G_{\mathbf{X}}(w) dw \\ &= \int_{a_{k-1}}^{\bar{a}_k} (w - a_{k-1}) G_{\mathbf{X}}(w) dw + \int_{\bar{a}_k}^{a_k} (w - a_k) G_{\mathbf{X}}(w) dw \\ &= \int_0^{\delta_k} \left[G_{\mathbf{X}}(\bar{a}_k - u) - G_{\mathbf{X}}(\bar{a}_k + u)\right](\delta_k - u)\, du.\end{aligned}$$

Here the last step follows by rearranging the integrals and change of variables where $w \in (a_{k-1}, a_k]$ is replaced by $u + \bar{a}_k$ for $u \in [-\delta_k, +\delta_k]$.

For large K, $|a_k - a_{k-1}|$ is small. In this regime

$$G_{\mathbf{X}}(\bar{a}_k - u) - G_{\mathbf{X}}(\bar{a}_k + u) = -2\left[\frac{\partial G_{\mathbf{X}}(w)}{\partial w}\right]_{\bar{a}_k} u + o(u^2).$$

Using this, $\int_{w \in (a_{k-1}, a_k]} \Delta(w) G_{\mathbf{X}}(w) dw$ becomes

$$\frac{1}{24}\left[\frac{\partial G_{\mathbf{X}}(w)}{\partial w}\right]_{\bar{a}_k} |a_k - a_{k-1}|^3 + o(|a_k - a_{k-1}|^3).$$

This expression holds for $k = 2$ to K, at large K. As terms corresponding to $k = 1$ and $k = K+1$ are $o(\frac{1}{K})$, we obtain (2) by summing this term from $k = 2$ to $k = K$:

$$\sum_{k=2}^{K} \frac{1}{24}\left(\left[\frac{\partial G_{\mathbf{X}}(w)}{\partial w}\right]_{\bar{a}_k} |a_k - a_{k-1}|^3 + o(|a_k - a_{k-1}|^3\right).$$

From this we obtain an expression for (2) in terms of point density function λ. At w there are $K\lambda(w) + o(K)$ points and hence, around w the partition width $|a_k - a_{k-1}| = \frac{1}{K\lambda(w)} + o\left(\frac{1}{K}\right)$. Hence,

$$\begin{aligned}&\sum_{k=2}^{K} \frac{1}{24}\left[\frac{\partial G_{\mathbf{X}}(w)}{\partial w}\right]_{\bar{a}_k} |a_k - a_{k-1}|^3 \\ &= \sum_{(w, w+dw]} \sum_{k : a_k \in (w, w+dw]} \frac{1}{24} \frac{\partial G_{\mathbf{X}}(w)}{\partial w} \frac{1}{\left(K\lambda(w)\right)^3} + o\left(\frac{1}{K^3}\right).\end{aligned}$$

Now, as the number of quantization points in $(w, w+dw]$ is $K\lambda(w)dw$, and as $\frac{\partial G_{\mathbf{X}}(w)}{\partial w}$ can be treated as constant over $(w, w+dw]$ (changes by $o(dw)$), we can further derive the following expression for (2).

$$\begin{aligned}&\sum_{w} (K\lambda(w)dw)\left[\frac{\partial G_{\mathbf{X}}(w)}{\partial w} \frac{1}{24\left(K\lambda(w)\right)^3} + o\left(\frac{1}{\left(K\lambda(w)\right)^3}\right)\right] \\ &= \frac{1}{24K^2} \int \frac{\partial G_{\mathbf{X}}(w)}{\partial w} \frac{1}{\lambda^2(w)} dw + o\left(\frac{1}{K^2}\right).\end{aligned}$$

Before delving into error analysis and choice of point density function for a general setting, consider a few special cases.

For uniform quantization, it follows that MSE has a simple expression: $\frac{1}{24K^2}\mathbb{E}_{\mathbf{X}}\left[(G_{\mathbf{X}}(1) - G_{\mathbf{X}}(0))^2\right]$. Thus, if $G_{\mathbf{X}}(1) =$

$G_{\mathbf{X}}(0)$ for all $\mathbf{X}$, then at high rate, the uniform quantizer is optimal and achieves 0 MSE.

Another special case is the scenario where the input distribution is such that for some function $\tilde{G}(w)$, $\mathbb{P}_{\mathbf{X}}\{\tilde{G}(w) = \frac{\partial G_{\mathbf{X}}(w)}{\partial w}$ for all $w\} = 1$, i.e., $\frac{\partial G_{\mathbf{X}}(w)}{\partial w}$ is almost surely independent of $\mathbf{X}$. Then the error is $\frac{1}{24K^2}\int \frac{\tilde{G}(w)}{\lambda^2(w)}dw$. From Hölder's inequality it follows that the error is minimized at a point density $\lambda^*(w) = \frac{(\tilde{G}(w))^{1/3}}{\int(\tilde{G}(w))^{1/3}}$, and the minimum is $\frac{1}{24K^2}||\tilde{G}(w))||_{1/3}$.

Now, we move back to the general case. For a point density function λ and a large K, the error in (1) becomes

$$\frac{1}{24K^2}\mathbb{E}_{\mathbf{X}}\left[\left(\int \frac{\partial G_{\mathbf{X}}(w)}{\partial w}\frac{1}{\lambda^2(w)}dw\right)^2\right] \tag{3}$$
$$= \frac{1}{24K^2}\iint \mathbb{E}_{\mathbf{X}}\left[\frac{\partial G_{\mathbf{X}}(w)}{\partial w}\frac{\partial G_{\mathbf{X}}(w')}{\partial w'}\right]\frac{1}{\lambda^2(w)\lambda^2(w')}dwdw',$$

where the exchange of the integrals and the expectation is possible by Fubini's theorem, as $|\frac{\partial G_{\mathbf{X}}(w)}{\partial w}|$ is bounded for any bounded $\mathbf{X}$ and w for all smooth neuron activation functions; for smooth activation functions with bounded inputs, derivatives are bounded. The final goal is to find the optimal point density function λ^* to minimize (3).

Define $\mathcal{G}(w,w') = \mathbb{E}_{\mathbf{X}}\left[\frac{\partial G_{\mathbf{X}}(w)}{\partial w}\frac{\partial G_{\mathbf{X}}(w')}{\partial w'}\right]$. Note that $\mathcal{G}(w,w')$ is symmetric in the input variables, i.e., $\mathcal{G}(w,w') = \mathcal{G}(w',w)$ for all w,w'. Also, define the following problem for a function $\mathcal{U}:[0,1]^2\to\mathbb{R}$:

$$\min_{\Lambda\geq 0:\int\Lambda dwdw'=1}\left|\int_{S_+}\frac{\mathcal{U}(w,w')}{(\Lambda(w,w'))^2}dwdw'\right|. \tag{4}$$

Note that the minimum of (4) for $\mathcal{U}=\mathcal{G}$ is a lower bound on the minimum of (3). This is because (3) is non-negative for all λ and if the optimal of (3) is $\lambda^*(w)$ then $\lambda^*(w)\lambda^*(w')$ is a feasible solution of (4). Intuitively, a point density λ with $\lambda(w)\lambda(w')$ closely resembling the optimum Λ of (4) would make (3) close to its minimum value. Indeed, if for a λ, $\lambda(w)\lambda(w')$ exactly matches the optimum of (4), then that λ is the optimal quantization point density.

Consider the sets $S_+ = \{(w,w') : \mathcal{G}(w,w') > 0\}$, $S_- = \{(w,w') : \mathcal{G}(w,w') < 0\}$, and the functions $\mathcal{G}_+ = \max(\mathcal{G},0), \mathcal{G}_- = \max(-\mathcal{G},0)$. Then (3) can be written as

$$\int_{S_+}\frac{\mathcal{G}_+(w,w')}{(\lambda(w)\lambda(w'))^2}dwdw' - \int_{S_-}\frac{\mathcal{G}_-(w,w')}{(\lambda(w)\lambda(w'))^2}dwdw'.$$

Let $\Lambda^*_+(w,w')$ and $\Lambda^*_-(w,w')$ be the solutions of (4) for $\mathcal{U}=\mathcal{G}_+$ and $\mathcal{U}=\mathcal{G}_-$ respectively. It follows that Λ^*_+ and Λ^*_- are 0 on S_- and S_+ respectively. Otherwise, the integral can be strictly decreased by equally distributing the mass that Λ^*_+ or Λ^*_- puts in S_- or S_+ among elements in S_+ or S_-, respectively.

Also, as $\mathcal{G}_+(w,w')$ or $\mathcal{G}_-(w,w')$ is symmetric in the arguments, so is Λ^*_+ or Λ^*_-. This can be argued by showing that if Λ is asymmetric, then distributing mass symmetrically between (w,w') and (w,w') reduces the integral. This is due to the fact that $1/x^2+1/y^2$ is minimized at $x=y$ when $x+y$ is constrained to be 1 and $x,y\geq 0$. For non-trivial S_+ and S_-, the errors terms (integrals) corresponding to S_+ and S_- are strictly positive, so there exists an $\alpha>0$ such that

$$\alpha\int_{S_+}\frac{\mathcal{G}_+(w,w')}{(\Lambda^*_+(w,w'))^2}dwdw' = \int_{S_-}\frac{\mathcal{G}_-(w,w')}{(\Lambda^*_-(w,w'))^2}dwdw'.$$

For non-trivial S_+ and S_-, define $\Lambda^*(w,w') = \frac{1}{\sqrt{\alpha}+1}\left(\sqrt{\alpha}\Lambda^*_+(w,w') + \Lambda^*_-(w,w')\right)$. It is not hard to see that Λ^* is the solution of (4) for $\mathcal{U}=\mathcal{G}$. This is because on S_+ we have $\Lambda^* = \sqrt{\alpha}\Lambda^*_+$ and on S_- $\Lambda^* = \Lambda_-$, and hence, the integral on S_+ and that on S_- cancel one another.

Using the fact that $\int_{S_+}\Lambda^*_+ = 1$ and Hölder's inequality, following the derivation in [17, Thm. 13], we obtain:

$$\Lambda^*_+(w,w') = \frac{(\mathcal{G}_+(w,w'))^{\frac{1}{3}}}{\int_{S_+}(\mathcal{G}_+(w,w'))^{\frac{1}{3}}}.$$

A closed-form expression for $\Lambda^*(w,w')$ then follows directly.

Consider the case where S_+ is $[0,1]^2$, and $\mathcal{G}_+(w,w')$ has a separable product form, i.e., $\mathcal{G}_+(w,w') = \hat{\mathcal{G}}_+(w)\hat{\mathcal{G}}_+(w')$ for some $\hat{\mathcal{G}}$. Then, Λ^* is Λ^*_+. As $\mathcal{G}$ is separable, $\Lambda^*(w,w') = (\hat{\mathcal{G}}_+(w)\hat{\mathcal{G}}_+(w'))^{\frac{1}{3}}/\int_{S_+}(\mathcal{G}_+(w,w'))^{\frac{1}{3}}$. This implies that $\Lambda^*(w,w') = \lambda^*(w)\lambda^*(w')$ for $\lambda^*(w)$ is given by

$$\frac{(\hat{\mathcal{G}}_+(w))^{\frac{1}{3}}}{\int_{S_+}(\hat{\mathcal{G}}_+(w))^{\frac{1}{3}}} = \int\Lambda^*(w,w')dw'. \tag{5}$$

Though $\mathcal{G}$ may not have a closed-form expression in all neural network applications, from the empirical values of $\mathcal{G}$, S_+, and S_- can be easily computed and separability of $\mathcal{G}_+$ can be also be checked by verifying whether $\mathcal{G}_+(w,w') = (\int\mathcal{G}_+(w,w')dw)(\int\mathcal{G}_+(w,w')dw')$.

For general $\mathcal{G}$, there is no generic method to obtain λ^* from Λ^*. We will soon give some approaches to find good surrogates for λ^* based on regularization, but before that we address an important concern. Though neural networks can be thought of as $\Phi(\mathbf{w},\mathbf{X})$, obtaining Φ and $\frac{\partial\Phi(\mathbf{w},\mathbf{X})}{\partial w}$ as closed-form expressions may not be feasible. Obtaining these is crucial for computing $\mathcal{G}$ and therefore for designing the optimal quantizer. Notwithstanding, though Φ may not be known, at any given $\mathbf{X}$, $\frac{\partial\Phi(\mathbf{w},\mathbf{X})}{\partial w}$ can be evaluated using the chain rule of differentiation and the celebrated *backpropagation algorithm*, which is also used to train neural networks. As long as the neural activation functions can be differentiated analytically, $\frac{\partial\Phi(\mathbf{w},\mathbf{X})}{\partial w}$ can be obtained computationally. To obtain $\mathcal{G}$, one can collect many samples of $\frac{\partial\Phi(\mathbf{w},\mathbf{X})}{\partial w}$ for different $\mathbf{X}$, and obtain a good empirical estimate of the expected quantities.

For a given $\mathcal{G}$, S_+, and S_-, if we can find λ^* such that $\Lambda^*(w,w') = \lambda^*(w)\lambda^*(w')$, then λ^* is the optimal solution of (3). Λ^* can be seen as a two-dimensional pdf, whereas λ^* is a one-dimensional pdf. Thus finding λ^* such that $\Lambda^*(w,w') = \lambda^*(w)\lambda^*(w')$ can be seen as approximating the joint pdf Λ^* by a product form with the same marginals. Towards this, we minimize relative entropy between the joint pdf and the product pdf:

$$\lambda^\dagger = \arg\min_{\lambda\geq 0:\int\lambda=1}\int\Lambda^*(w,w')\log\frac{\Lambda^*(w,w')}{\lambda(w)\lambda(w')}dwdw'.$$

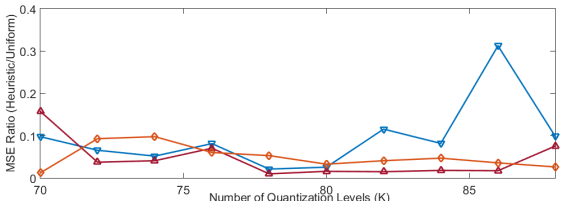

Fig. 2. MSE ratio between heuristic and uniform quantizers, as function of number of quantization levels, for several specific neural networks whose weights are from beta distributions $(2, 2)$ and have square-root activation functions.

If the objective is 0 at $\lambda^\dagger$ then $\lambda^* = \lambda^\dagger$, otherwise it is the best approximation of λ^* in relative entropy. Thus, $\lambda^\dagger$ that comes from the above regularization can be used as a surrogate for λ^* with a relative entropy guarantee.

It is not hard to see the optimization problem reduces to

$$\arg \min_{\lambda \geq 0: \int \lambda = 1} \int \left(\int \Lambda^*(w, w') dw' \right) (-\log \lambda(w)) dw.$$

Using variational methods it can be shown that the optimal solution $\lambda^\dagger = \int \Lambda^*(w, w') dw'$. If w, w' are from a finite discrete space and the integration is a finite sum, the result follows by considering the Lagrangian of the problem and the KKT conditions.

We study effectiveness of this design heuristic through simulations. Our simulation setting consists of three-layer, fully-connected neural networks with 50 input nodes, 100 nodes each in the next two layers, and a single output node. Weights between the input layer and the next layer are drawn randomly from a beta $(2, 2)$ distribution. All edges from the second hidden layer to the output node as well as those between the two hidden layers have unit weights. The activation function is a square-root. For a given draw of edge weights the network realizes a randomly chosen function. For numerical computation of $\mathcal{G}(w, w')$ we use the approximation: $\mathcal{G}(w, w') \approx Cp(w)p(w')$, where C depends on the moments of $\mathbf{X}$ and $p(w)$. This approximation is reasonable for computing $\mathcal{G}$ in a dense and symmetric network. Fig. 2 compares the MSE of the proposed heuristic design against that of the uniform quantizer for a few different draws of the network for different number of quantization levels K and shows significant gains. MSEs have been empirically computed by averaging the square of the differences between network outputs with exact and quantized edge weights over 1000 input values drawn from the uniform distribution.

IV. Conclusion

Efficient quantization of large neural networks is important for the success of real-time mobile and in-sensor inference. Motivated by a lack of formal approaches to this problem, we develop a framework based on high-rate quantization theory. Though intuitions from distributed functional quantization [17] are useful in this regard, quantization of neural networks differs significantly due to dependencies between edge weights and the impacts of random network inputs. We develop a new approach that addresses these issues and results in an optimal quantizer design. Under certain conditions, we get a quantizer with certain regularization guarantees with respect to the optimal quantizer and with a good empirical performance.

References

[1] M. Kang, M.-S. Keel, N. R. Shanbhag, S. Eilert, and K. Curewitz, "An energy-efficient VLSI architecture for pattern recognition via deep embedding of computation in SRAM," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP 2014)*, May 2014, pp. 8326–8330.

[2] Z. Wang, K. H. Lee, and N. Verma, "Overcoming computational errors in sensing platforms through embedded machine-learning kernels," *IEEE Trans. VLSI Syst.*, vol. 23, no. 8, pp. 1459–1470, Aug. 2015.

[3] S. Zhang and N. R. Shanbhag, "Reduced overhead error compensation for energy efficient machine learning kernels," in *Proc. 2015 IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2015, pp. 15–21.

[4] J. Ahn, S. Hong, S. Yoo, O. Mutlu, and K. Choi, "A scalable processing-in-memory accelerator for parallel graph processing," in *Proc. 42nd Annu. Int. Symp. Comput. Archit. (ISCA '15)*, Jun. 2015, pp. 105–117.

[5] R. Andri, L. Cavigelli, D. Rossi, and L. Benini, "YodaNN: An ultra-low power convolutional neural network accelerator based on binary weights," in *Proc. 2016 IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Jul. 2016, pp. 236–241.

[6] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[7] N. R. Shanbhag, "Energy-efficient machine learning in silicon: A communications-inspired approach," arXiv:1611.03109 [cs.LG]., Oct. 2016.

[8] M. Courbariaux, Y. Bengio, and J.-P. David, "Low precision arithmetic for deep learning," arXiv:1412.7024 [cs.LG]., Dec. 2014.

[9] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML 2015)*.

[10] Y. Gong, L. Liu, M. Yang, and L. Bourdev, "Compressing deep convolutional networks using vector quantization," arXiv:1412.6115, Dec. 2014.

[11] S. Basu and L. R. Varshney, "Universal source coding of deep neural networks," in *Proc. IEEE Data Compression Conf. (DCC 2017)*, Apr. 2017.

[12] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML 2015)*, Jul. 2015.

[13] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," arXiv:1603.05279 [cs.CV]., Mar. 2016.

[14] D. D. Lin, S. S. Talathi, and V. S. Annapureddy, "Fixed point quantization of deep convolutional networks," arXiv:1511.06393 [cs.LG]., Jun. 2015.

[15] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," arXiv:1510.00149 [cs.CV]., Oct. 2015.

[16] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2325–2383, Oct. 1998.

[17] V. Misra, V. K. Goyal, and L. R. Varshney, "Distributed scalar quantization for computing: High-resolution analysis and extensions," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5298–5325, Aug. 2011.

[18] V. Misra and K. Viswanathan, "Sequential functional quantization," in *Proc. 2013 IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 2359–2363.

[19] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control Signals Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989.

[20] P. Kumaraswamy, "A generalized probability density function for double-bounded random processes," *J. Hydrol. (Amst.)*, vol. 46, no. 1/2, pp. 79–88, Mar. 1980.