Training Quantized Neural Networks to Global Optimality via Semidefinite Programming

ICML 2021

Burak Bartan Mert Pilanci

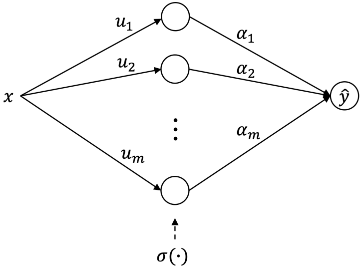
{bbartan, pilanci}@stanford.edu
Department of Electrical Engineering, Stanford University

Introduction

- Consider training *quantized* neural networks for efficient machine learning models

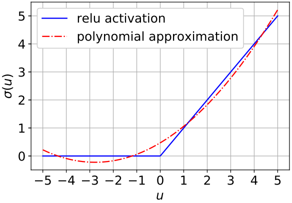
$$f(x) = \sum_{j=1}^{m} \sigma(x^T u_j)\alpha_j \tag{1}$$

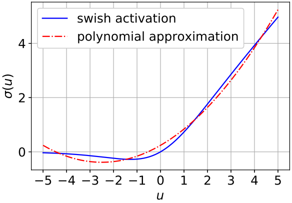
where $u_j \in \{-1, 1\}^d$ and $\alpha_j \in \mathbb{R}$. This is a two-layer fully connected architecture with scalar output, $f(x) : \mathbb{R}^d \to \mathbb{R}$ (see the paper for extension to different architectures).



Activation Functions

- The theory holds for quadratic activation $\sigma(u) = u^2$, degree-2 polynomial activation $\sigma(u) = au^2 + bu + c$, and bilinear activation $\mathcal{X} \to u^T \mathcal{X} v$ where $\mathcal{X} := xx^T$.
- We show that bilinear activation NN can be represented as a polynomial activation NN.
- It is demonstrated in (Allen-Zhu, Li, 2020)[1] that the degree-2 polynomial activation performs comparably to ReLU activation in deep networks.



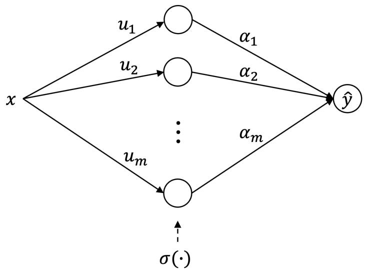


[1] Zeyuan Allen-Zhu and Yuanzhi Li. Backward feature correction: How deep learning performs deep learning.

Problem Setup

- Let $X \in \mathbb{R}^{n\times d}$ denote the data matrix and $y \in \mathbb{R}^n$ denote the output vector.
- Combinatorial NP-hard problem:

$$p^* = \min_{\text{s.t. } u_j \in \{-1,1\}^d, \alpha_j \in \mathbb{R}\, j \in [m]} \ell\left(f(X), y\right) + \beta d \sum_{j=1}^{m} |\alpha_j| \,. \tag{2}$$



Lower Bounding SDP

- For bilinear activation, we obtain the lower-bounding problem via duality as

$$
\begin{aligned}
p_b^* \geq d_{\mathrm{bSDP}} := \min_{Q,\rho} \quad & \ell\left(\hat{y},\, y\right) + \beta d \rho \\
\text{s.t.} \quad & \hat{y}_i = 2 x_i^T Z x_i,\ i = 1, \ldots, n \\
& Q_{jj} = \rho,\ j = 1, \ldots, 2d \\
& Q = \begin{bmatrix} V & Z \\ Z^T & W \end{bmatrix} \succeq 0\,.
\end{aligned} \tag{3}
$$

- This is a convex SDP, which can be solved efficiently in polynomial time.

Algorithm

Algorithm 1: Sampling algorithm for quantized neural networks

1. Solve the SDP in (3). Define the scaled matrix $Z_s^* \leftarrow Z^*/\rho^*$.
2. Solve the problem

$$Q^* := \arg \min_{Q \succeq 0, Q_{jj}=1 \forall j} \|Q_{(12)} - \sin(\gamma Z_s^*)\|_F^2 . \quad (4)$$

3. Sample the first layer weights $u_1, \ldots, u_m, v_1, \ldots, v_m$ from multivariate normal distribution as $\begin{bmatrix} u \\ v \end{bmatrix} \sim \mathrm{sign}(\mathcal{N}(0, Q^*))$ and set the second layer weights as $\alpha_j = \rho^* \frac{\pi}{\gamma m}, \forall j$.

Main Result

Theorem

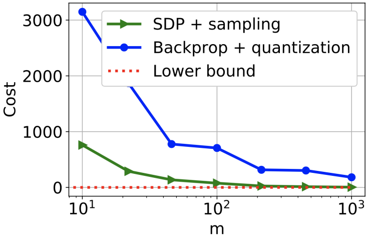
Let θ represent the neural network weights $u_j, v_j \in \{-1, +1\}^d, \alpha_j \in \mathbb{R}$, $j = 1, \ldots, m$. Algorithm 1 returns a neural network with weights $\hat{\theta}$ that achieve near optimal loss, i.e.,

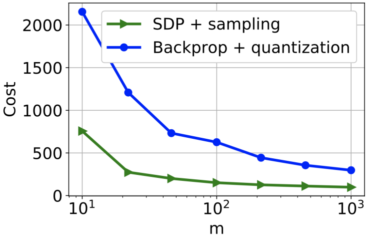
$$\left| \ell\big(f_{\hat{\theta}}(X),\, y\big) - \ell\big(f_{\theta^*}(X),\, y\big) \right| \leq \epsilon \tag{5}$$

with high probability. The weights θ^* are the optimal network weights for the non-convex combinatorial problem.

Numerical Results

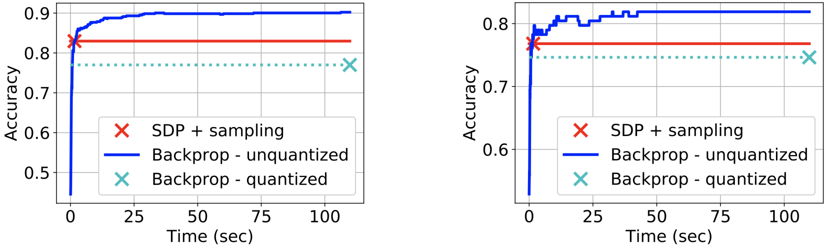
- Cost against the number of neurons m on the training (left) and the test (right) sets. Dataset X has been synthetically generated and has dimensions $n = 100$, $d = 20$.





Numerical Results

- Classification accuracy on the training (left) and test (right) sets against wall-clock time for the credit approval dataset with $n = 552, d = 15$.



Discussion

- We have shown that bilinear activation architectures with binary quantization are sufficient to train optimal multi-level quantized networks with polynomial activations.
- We have developed a sampling algorithm to generate quantized neural networks using the lower-bounding SDP by leveraging Grothendieck's identity and the connection to approximating the cut norm.
- Future direction: Application of the proposed algorithm in layerwise training.