TOWARDS A DEEPER UNDERSTANDING OF TRAINING QUANTIZED NETWORKS

Hao Li*, Soham De*, Zheng Xu, Christoph Studer, Hanan Samet, Tom Goldstein
DEEP NETS ARE BIG

Low power devices

SOLUTION: QUANTIZED/LOW-PRECISION NETWORKS

Advantages

• FAST & hardware friendly: no multiplications
• Low storage costs
• Low power consumption
HOW TO USE QUANTIZED NETS?

Train using HPC

Inference on low power devices

Can we train quantized models on resource-constrained devices?
HOW TO TRAIN QUANTIZED NETS?

Non-quantized: Stochastic Gradient Descent

\[ w^{k+1} = w^k - \alpha \nabla f(w^k) \]

Fully-quantized: Stochastic rounding [Gupta ICML'15]

\[ w^{k+1} = w^k - Q[\alpha \nabla f(w^k)] \]

Advantage: no floating-point weights
HOW TO TRAIN QUANTIZED NETS?

Non-quantized: Stochastic gradient descent

\[ w^{k+1} = w^k - \alpha \nabla f(w^k) \]

Fully-quantized: Stochastic rounding [Gupta ICML’15]

\[ w^{k+1} = w^k - Q[\alpha \nabla f(w^k)] \]

Semi-quantized: BinaryConnect [Courbariaux NIPS’15]

\[ w^{k+1} = w^k - \alpha \nabla f(Q[w^k]) \]

very popular

QNN[Hubara, arXiv’16] XNOR-Net [Rastegar, ECCV’16]
DoReFA-Net [Zhou, arXiv’16] and etc…

Disadvantage: still requires floating-point weights
Train CNNs (VGG-Net, ResNets, Wide-ResNet) with binary weight on CIFAR-10/100

The SR method cannot beat BC, why?
THIS TALK

Goal: develop *principled* framework for training quantized nets

Why are we able to train quantized nets at all?
Can we prove that SGD solves this difficult combinatorial problem?

Why does training require floating point weights?
Why can’t we train on embedded systems using SR?
CONVERGENCE UNDER CONVEXITY ASSUMPTIONS
**Theorem 2** Assume that $F$ is $\mu$-strongly convex and the learning rates are given by $\alpha_t = \frac{1}{\mu(t+1)}$. Let $G$ bound the gradient magnitude. Then

$$\mathbb{E}[F(\bar{w}^T) - F(w^*)] \leq \frac{(1 + \log(T + 1))G^2(1 + \Delta^{-1})}{2\mu T} \quad + \quad \frac{\sqrt{d}\Delta G}{2}$$

SR converges until it reaches an “**accuracy floor**”, which is determined by the quantization error $\Delta$. 
CONVERGENCE THEORY FOR BINARYCONNECT

\( L_2 \) is a Lipschitz constants for the Hessian

**Theorem 1** Assume that \( F \) is \( \mu \) -strongly convex and the learning rates are given by \( \alpha_t = \frac{1}{\mu(t+1)} \). Let \( G \) bound the gradient magnitude. Then

\[
\mathbb{E}[F(\bar{w}^T) - F(w^*)] \leq \frac{(1 + \log(T + 1))G^2}{2\mu T} + \frac{DL_2\sqrt{d\Delta}}{2}
\]

BC converges until it reaches an “**accuracy floor**”, which is determined by the quantization error \( \Delta \) and \( L_2 \) (0 if \( F \) is quadratic).

**Corollary:** BC finds **exact** solutions to quadratic problems
But this can’t be whole story.

What can we say about the bad behavior of SR on non-convex problems?

The answer has to do with exploration vs exploitation.
FLOATING POINT

Learning rate = 1

Quantized scalar weight $\Delta = 0.5$
FLOATING POINT

Learning rate = 1

Quantized scalar weight \( \Delta = 0.5 \)
FLOATING POINT

Learning rate = 1

Quantized scalar weight $\Delta = 0.5$
FLOATING POINT

Learning rate = 1

Quantized scalar weight \( \Delta = 0.5 \)
FLOATING POINT

Learning rate = 1

Histogram
FLOATING POINT

Learning rate = 0.1

![Histogram](image)
FLOATING POINT

Learning rate = 0.01
FLOATING POINT

Learning rate = 0.001

Histogram
QUANTIZED

Learning rate = 1

Histogram
QUANTIZED

Learning rate = 0.1

Histogram
Learning rate = 0.01

Histogram
QUANTIZED

Learning rate = 0.0001

Histogram
WHAT'S WRONG?

Floating Point / Binary Connect

Exploration  Exploitation

Shrink Learning rate

Exploration  Exploitation

Stochastic Rounding

Exploration  Exploitation

Shrink Learning rate

Exploration  Exploitation
MARKOV CHAIN INTERPRETATION

“Weight space”

\[ w_{k+1} = w_k - Q[\alpha \nabla f(w_k)] \]
Long term dynamics governed by the equilibrium distribution \( \pi_\alpha \)
MARKOV CHAIN INTERPRETATION

$\pi_\alpha$

equilibrium distribution
Theorem (Kushner and Clark, 1978)

Classical (floating-point) SGD converges to a stationary points almost surely

$$\lim_{k \to \infty} \| \nabla f (w^k) \| = 0$$

This result also applies to Binary Connect!

In other words…

The stationary distribution concentrates on stationary points.

These algorithms have an exploitation phase!
WHAT ABOUT STOCHASTIC ROUNding?

Fully discrete stochastic rounding **does not concentrate** on stationary points.

**Theorem**  Let $p_{x,k}$ denote the distribution function of the kth entry in the stochastic gradient $\tilde{\nabla}f(w)$. If $\int_\nu^\infty p_{x,k}(z) \, dz < C/\nu^2$, and $p_{x,k}$ has non-zero mass on both the positive and negative reals, then there exists a distribution $\tilde{\pi}$, with

$$\lim_{\alpha \to 0} \pi_\alpha = \tilde{\pi}.$$  

Furthermore, $\tilde{\pi}$ is not concentrated on stationary points.

**Assumptions are weak enough for neural nets!**
WHAT ABOUT STOCHASTIC ROUNDED?

Fully discrete stochastic rounding stops exploring as the learning rate gets small.

**Theorem** The mixing time $M_\alpha$ of the Markov chain induced by stochastic rounding SGD satisfies

$$\lim_{\alpha \to 0} M_\alpha = \infty.$$

Exploration slows down, but exploitation never happens!
Convergence theory for quantized nets

Convex problems: methods converge to until an “accuracy floor” is reached that depends on the discretization width.

Non-convex problems: fully quantized methods lack the important annealing properties enjoyed by floating-point methods.
Thank you!
Questions/Comments?

Towards a deeper Understanding of Training Quantized Networks
Hao Li*, Soham De*, Zheng Xu, Christoph Studer, Hanan Samet, Tom Goldstein