Week4 Report

Yuchen.cai.uestc@gmail.com

1. Summarized optimization problems and methods for quantization neural networks.

近几年神经网络的压缩算法成了一个研究热点，主要的网络压缩途径有 4 种，量化、剪枝、低秩分解、轻量化网络设计，量化就是将以往用 32bi 表达的浮点数用 1bit、2bit 占用较少内存空间的形式进行存储。剪枝的目的是为了去掉一些不重要的神经元、连接、通道等，低秩分解主要是通过各种分解方法用精简的张量来表达复杂张量，轻量化网络设计主要是类似 MobileNet 这种设计的非常精简但性能又好的网络。

- BinaryConnect: 2015 年发表，第一篇归纳出完整量化流程的文章，提出 DNN 的前向和反向训练中*使用 1bit 的二值化权重代替浮点权重*，将硬件计算中的乘法操作简化成累加操作，能够大量减少储存空间。
- BinarizedNeuralNetwork: 与 BC 属同一作者，是 BC 算法的扩展。其主要贡献在于同时对权重 weight 和激活值 activations 进行量化到 1bit，还分析了针对低 bit 如何进行更有效的计算。
- XNOR-NET: 全文分为 BWN 和 XNOR-NET 两个部分。BWN 只是将权重 weights 量化为二值，而 XNOR-NET 则将权重和输入/激活值都量化为了二值。在 BWN 中，复杂的卷积点乘被加减法代替了，而在 XNOR-NET 当中，复杂的卷积点乘则可以用异或运算 XNOR 来实现，进一步加快了硬件计算的速度。在第一个部分 BWN 中，作者的贡献在于引入了 scaling factor α，量化问题变成了一个优化问题。 在第二个部分 XNOR-NET 当中，

对激活值的量化，作者引入了尺度系数β.

- Dorefa-Net: 不同于前作几篇论文，Dorefa 提出了对梯度也进行量化，并且支持量化到任意 bit，优点是不光在 interference 时能够加速，且训练时由于梯度也被量化了，训练时候也可以加速，这使得直接在硬件平台上进行训练成为可能。
- HWGQ：主要从理论上结合实际的激活值高斯分布, BatchNormalizaiton 和 RELU 函数，介绍了如何去选择并设计一个 quantizer 作为激活函数，从而保证量化后的低 bit 网络性能接近浮点网络. 首先，针对激活值的量化需要处理不可微的操作，主要切入点在于 ReLU 函数，神经网络每个单元都计算了一个激活函数，即权重与输入相乘后经过一个非线性变换，这个操作的多少决定了整个网络的复杂度。其次，在对激活值进行量化时，如果直接按照符号函数来定义量化 levels，那么对量化激活值进行求导时，导数处处为 0，所以有人提出，对符号函数求导如果输入绝对值小于 1 则梯度为 1，其他位置取 0。结合 relu 函数，本文对激活值量化的目标是拟合 relu 函数的输出。quantizer 是一个分段常数函数，量化 levels 就是量化到的值，量化 step 就是两个量化 level 之间的差，针对每个浮点数值，它只需要保存一个索引值 i，对应到第 i 个量化 level，非均匀量化情况下表达浮点权重需要多余 $log2m$log2m 的 bit 数，如果是均匀量化则 $log2m$log2m 个 bit 就够了。激活值的统计结构倾向于对称非稀疏分布类似高斯分布，再结合 relu，就是变成了一个半波高斯量化子 half-wave Gaussian quantizer。这里的量化就是变成了求针对高斯分布的量化 levels 和 step，但是在不同层的神经元所得到的类高斯分布，它们的均值方差不一定是相同的，为了得到相同的数据分布，通过 batchnorm 来得到 0 均值 1 方差的分布，然后就能在神经元之间得到

均匀的量化参数。以上是在解释构造这样的 HWGQ，它是阶梯常数函数，梯度处处为 0，目标就变成了选哪个函数作为 HWGQ 在量化后才能最好的拟合 relu 函数的效果。最后，作者考虑了三种不同的 relu 函数，发现使用长尾 RELU 最不容易丢失信息。

- Problem:
  - 很多方法只能在小数据集，如 CIFAR-10,MNIST 上取得效果，而在 ImageNet 这种大数据集上表现乏力
  - 关于如何设计量化，绝大部分的研究针对的是人工设计量化器，而没有考虑到如何设计一个能使网络“自学习”量化器的机制

2. Read Code of HWGQ

Café 使用尚不熟练，公式仍在理解中.

No.

Date

Deep Learning with Low Precision by Half-wave Gaussian Quantization

- 解决的问题 { Large Model Size
  Large Computation Cost

- Core network operation : dot product between a weight and an activatio

入手点 { weight → 二值化
Activations → 二值化
量化

- 面临难题：二值化操作或者量化操作会产生
  Step-wise response that produces weak gradient signals during

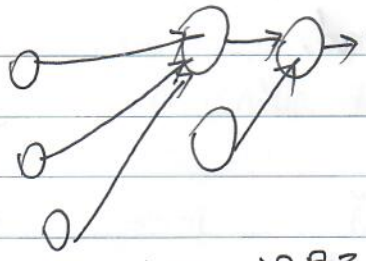
- 前人工作：
  - 低阶矩阵分解；Pruning；模型结构设计（去掉FC，用1×1 conv代
  - 参数的二值化和量化
  - Activations 的量化
    - 均无法在大型数据集ImageNet上面取得理想效果

No.

Date

• Binary Network

◦ Each unit computes an Activation function:



$z = g(W^T X)$ (1)

$W \in R^{c \cdot w \cdot h}$, weight vector

$X \in R^{c \cdot w \cdot h}$, input vector

CNN → 问题 { W: a tensor, 大的储存空间

dot product $W^T X$: 对计算能力有要求

↳ 出现了"Low Precision Network" { Binarized NN
XNOR-Net
Dorefa-Net

◦ 参数二值化 (由XNOR-Net提出)

$W \approx \alpha B$ (2) α: scaling factor

B: Binary matrix

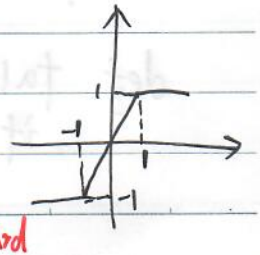
◦ Activation 二值化 (XNOR-Net; BNN)

$$z = sign(X) = \begin{cases} +1 & 若 X > 0 \\ -1 & otherwise \end{cases} \quad (3)$$

Forward Approximation

对(1)有 $\frac{\partial C}{\partial W} = \frac{\partial C}{\partial z} g'(W^T X) X$ (4)

✳ 为了解决梯度→0问题, BNN提出使用 Hard TAnh ($\widetilde{sign}$)



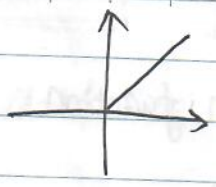
$$\widetilde{sign}'(x) = \begin{cases} 1 & 若 |x| \le 1 \\ 0 & otherwise \end{cases} \quad (5)$$

Backward

No.

Date

○ 本文方法：HWGQ

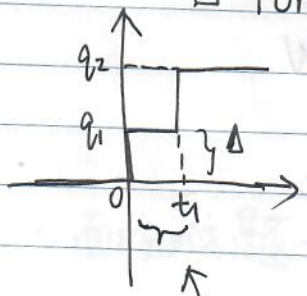
□ Relu  $g(x) = \max(0, x)$ (6)

Unlike squashing non-linearity like tanh or sigmod, ReLu improves the efficiency of BP.

□ 在 FeedForward 中：提出 Quantizer $Q(x)$ 去近似 (6)

在 BP 中：提出 piecewise linear 近似 $Q(x)$

□ Forward Approximation:



$$Q(x) = q_i, \quad \text{if } x \in (t_i, t_{i+1}] \quad (7)$$

A quantizer is denoted uniform, if

$$q_{i+1} - q_i = \Delta, \ \forall i \quad (8)$$

(上台阶)

Lloyd's Algorithm $Q^*(x) = \arg\min_Q E_x[(Q(x) - x)^2]$ (9)

$$= \arg\min_Q \int p(x)(Q(x) - x)^2 dx$$

↑ statistic feature

clip ( ) * delta, mi

lambda 回传

ReLu + $Q(x)$ = HWGQ

↑ not zero mean

def fa(x):

计 bitA == 32

Restoring to Batch Normalization

No.

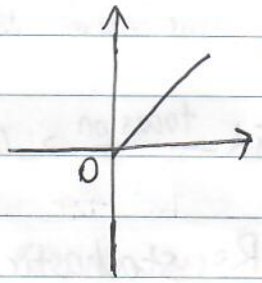
Date

□ Backward Approximation

We seek a piece-wise function that provides a good approximation to the Relu and to the HWGQ

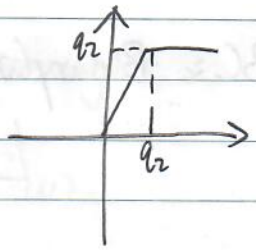
△ vanilla ReLu (原始的)

$$\tilde{Q}(x) = \begin{cases} 1, & \text{if } x>0 \\ 0, & \text{otherwise} \end{cases}$$



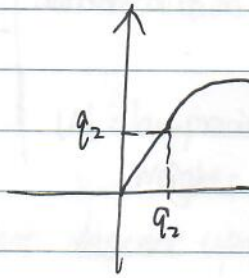
△ Clipped ReLu

$$\hat{Q}_c(x) = \begin{cases} q_m, & x>q_m \\ x, & x\in(0,q_m] \\ 0, & \text{otherwise} \end{cases}$$



△ Log-tailed ReLU

$$\hat{Q}_l(x) = \begin{cases} q_m + \log(x-\tau), & x>q_m \\ x, & x\in(0,q_m] \\ 0, & x\le 0 \end{cases}$$



No.

Date

Training Quantized Nets: A Deeper Understanding

- Abstract:

  Training algorithm that exploits high-precision representations have an important greedy search phase that purely quantized training methods lac

- Intro:

  有人 —focus on→ coarsely quantized weights

  SR: stochastic rounding (2015 ICML)

  BC: BinaryConnect (Courv, 2015 NIPS)

  ↳ Annealing property

- Related work:

  Most extreme scenario of quantization → Binarization (only 1-bit)

  For training quantized NN from scratch, many:
  maintaining a high-precision floating point copy of weights
  while feeding quantized weights into backprop

No.

Date

3. Training Quantized Neural Nets

$$\min_{w \in W} F(w) := \frac{1}{m} \sum_{i=1}^{m} f_i(w)$$

:= Item on the left is being defined to be what is on the right hand side

SGD: $w^{t+1} = w^t - \alpha_t \nabla \hat{f}(w^t)$ α_t: learning rate

Convolutions 可以被 { "+", "−" 替代; Bitwise Operations (Dorefa-Net)

☆ To train networks using a low-precision 表示 of the weights:

Quantization Function $Q(\cdot)$ $\hat{w} = Q(w)$

Different quantized 优化路径 can be defined by selecting
- different quantizers
- when quantization happens during optimization

常见方法如下:

□ Deterministic Rounding (R)

$$Q_d(w) = \text{sign}(w) \cdot \Delta \cdot \left\lfloor \frac{|w|}{\Delta} + \frac{1}{2} \right\rfloor$$

Δ: resolution

$\lfloor x \rfloor$: floor of x, 即小于等于x的最大整数

$\lfloor 2.9 \rfloor = 2$

$\lfloor -2.6 \rfloor = -3$

R SGD: $w_b^{t+1} = Q_d(w_b^t - \alpha_t \nabla \hat{f}(w_b^t))$ (w_b: low-precision weights)

(Quantized using Q_d immediately after applying the gradient descent upda

□ Stochastic Rounding (SR)

$$Q_s(w) = \Delta \cdot \begin{cases} \lfloor \frac{w}{\Delta} \rfloor + 1, & \text{for } p \leq \frac{w}{\Delta} - \lfloor \frac{w}{\Delta} \rfloor \quad (p \in (0,1)) \\ \lfloor \frac{w}{\Delta} \rfloor, & \text{otherwise} \end{cases}$$

No.

Date

□ Binary Connect:

$$w_r^{t+1} = w_r^t - \alpha_t \nabla \tilde{f}(Q(w_r^t))$$

w_r:
A full-precision buffe

low-precision weights : $\{-1, 1\} \xrightarrow{\text{XT}} \{-\Delta, \Delta\}$

BWN in XNOR-Net, allows different filters to have different scales for quantization

4. Convergence Analysis