周报-4

Incremental quantization

增量型量化的基本思路是，部分的对参数或者激活函数进行量化处理，使得整个量化过程更加平稳。这里的 部分 可以有多种层面，可以是数量上的部分，也可以是程度上的部分。

Previous work

Incremental Network Quantization

Incremental Network Quantization: Towards Lossless CNNs with Low-Precision Weights[1]
是从数量上，不断增加 quantized 参数的数量。整个算法的过程可以用一幅图表达清楚：

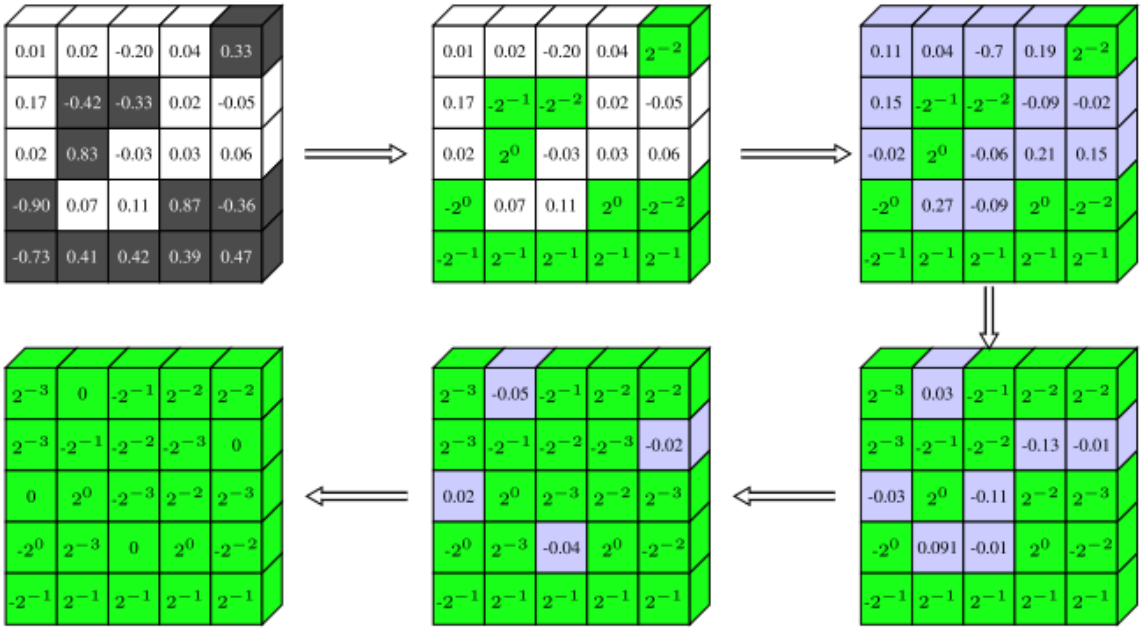

Figure 2: Result illustrations. First row: results from the 1^{st} iteration of the proposed three operations. The top left cube illustrates weight partition operation generating two disjoint groups, the middle image illustrates the quantization operation on the first weight group (green cells), and the top right cube illustrates the re-training operation on the second weight group (light blue cells). Second row: results from the 2^{nd}, 3^{rd} and 4^{th} iterations of the INQ. In the figure, the accumulated portion of the weights which have been quantized undergoes from 50%→75%→87.5%→100%.

优化过程是：

$$\begin{aligned} \min_{\mathbf{W}_l} \quad & E(\mathbf{W}_l) = L(\mathbf{W}_l) + \lambda R(\mathbf{W}_l) \\ \text{s.t.} \quad & \mathbf{W}_l(i,j) \in \mathbf{P}_l, \text{ if } \mathbf{T}_l(i,j) = 0, \ 1 \le l \le L, \end{aligned}$$

公式（1）

$$\mathbf{W}_l(i,j) \leftarrow \mathbf{W}_l(i,j) - \gamma \frac{\partial E}{\partial(\mathbf{W}_l(i,j))} \mathbf{T}_l(i,j),$$

公式（1-2）

这里 T_l 是一个 `mask matrix`，每个 iteration 在 retrain 的时候，只更新还没有被 quantized 的参数。

> 整个过程比较 tricky，但是作者对结果非常满意。其他 Researcher 反应此方法训练起来速度比较慢，而且超参数调起来比较麻烦。这个方法可能借鉴的点是，他其实限制了一些参数的变化。二值网络在训练的过程中，不稳定，难收敛的原因在于，参数的变化，都太过剧烈了，而在 real-value 网络训练过程中，他们的变化其实是

> 比较温和的。

Bottom-to-Top

Overcoming Challenges in Fixed Point Training of Deep Convolutional Networks[2] 使用了分层 来增量型量化。具体来说就是，优先量化最底层的参数，然后再优化上层的函数，最后所有的参数都能得到优化。

Table 1. Example showing the phases of iterative fine-tuning

	Phase 1		Phase 2		Phase 3	
	Acts	Wgts	Acts	Wgts	Acts	Wgts
Layer4	Float	-	Float	-	Float	update
Layer3	Float	-	Float	update	FixPt	-
Layer2	Float	update	FixPt	-	FixPt	-
Layer1	FixPt	-	FixPt	-	FixPt	-

> 这篇论文的简单，实验效果上是非常有效的。这无疑相互印证了，上面的猜想，就是所有的参数，直接训练可能比较难找到一个收敛点。另外这篇论文使用这种分层的 增量训练 的出发点之一也是：梯度误差积累效应。

Two-Step Quantization

Two-Step Quantization for Low-bit Neural Networks[3]

Object	Name	Method
low-bit weights	transformations	non-linear least square regression problem
low-bit activations	encodings	sparse suantization

先是对 activation 做离散化处理，采用的方式和 Half-Wave Gaussian Quantizer (HWGQ) 无他，只是又强行加入了 sparse 的处理，让整个激活值更加的稀疏，从结果上来看，一定程

度的稀疏，反而能让效果更好。

Table 2. Two-bit activation quantization comparison. Our sparse quantization method, denoted by SQ, is conducted under different sparsity.

Model	Sparsity (%)	Top-1 (%)	Top-5 (%)
AlexNet	50.00	58.5	81.5
HWGQ [2]	50.00	55.8	78.7
SQ-1	56.25	58.2	80.7
SQ-2	62.50	59.0	81.3
SQ-3	68.75	58.9	80.8
SQ-4	75.00	57.9	79.8

对 activation 操作之后，紧接着就是 weights 了。binarized 的规则跟以前一样，但是他是一层一层做的，其实跟之前我做的那篇 paper 方法一样。

$$\underset{\{\alpha_i\},\{\hat{w}_i^T\}}{\text{minimize}} \quad \sum_i \| y_i^T - Q_\epsilon(\alpha_i \hat{w}_i^T X) \|_2^2$$

公式（2）

通过转换，最后是用非梯度下降方法，半闭式的解出了 离散化 weights。

> 按照这篇论文的提法，他是 decouple 了 activation 和 weights，但是这种说法其实只是形式上的问题。
> 这篇文章对activation的处理并没有太大新意，对weights的处理，其实和之前的方法无太大差别，从某种程度上可以理解成一种distillation。

BinaryRelax

BinaryRelax[4]的增量型策略是，不断的提高 binary 的程度

$$x^{k+1} = \frac{\lambda \, \text{proj}_Q(y^{k+1}) + y^{k+1}}{\lambda + 1}.$$

公式（3）

这里的$lambda$控制了 binary 的程度。

---

1. Zhou, A., Yao, A., Guo, Y., Xu, L., & Chen, Y. (2017). Incremental Network Quantization: Towards Lossless CNNs with Low-Precision Weights. Retrieved from http://arxiv.org/abs/1702.03044 ↩
2. Lin, D. D., & Talathi, S. S. (2016). Overcoming Challenges in Fixed Point Training of Deep Convolutional Networks. Retrieved from http://arxiv.org/abs/1607.02241 ↩
3. Wang, P., Hu, Q., Zhang, Y., Zhang, C., Liu, Y., & Cheng, J. (2018). Two-Step Quantization for Low-bit Neural Networks. CVPR. https://doi.org/10.1109/CVPR.2018.00460 ↩
4. Yin, P., Zhang, S., Lyu, J., Osher, S., Qi, Y., & Xin, J. (2018). BinaryRelax: A Relaxation Approach For Training Deep Neural Networks With Quantized Weights, 1–17. Retrieved from http://arxiv.org/abs/1801.06313 ↩