Week 3 Report
Qian Jiang

1. Backpropagation of quantization neural network

   (Detailed notes attached below)

2. Deep Learning with Limited Numerical Precision [2015 ICML]

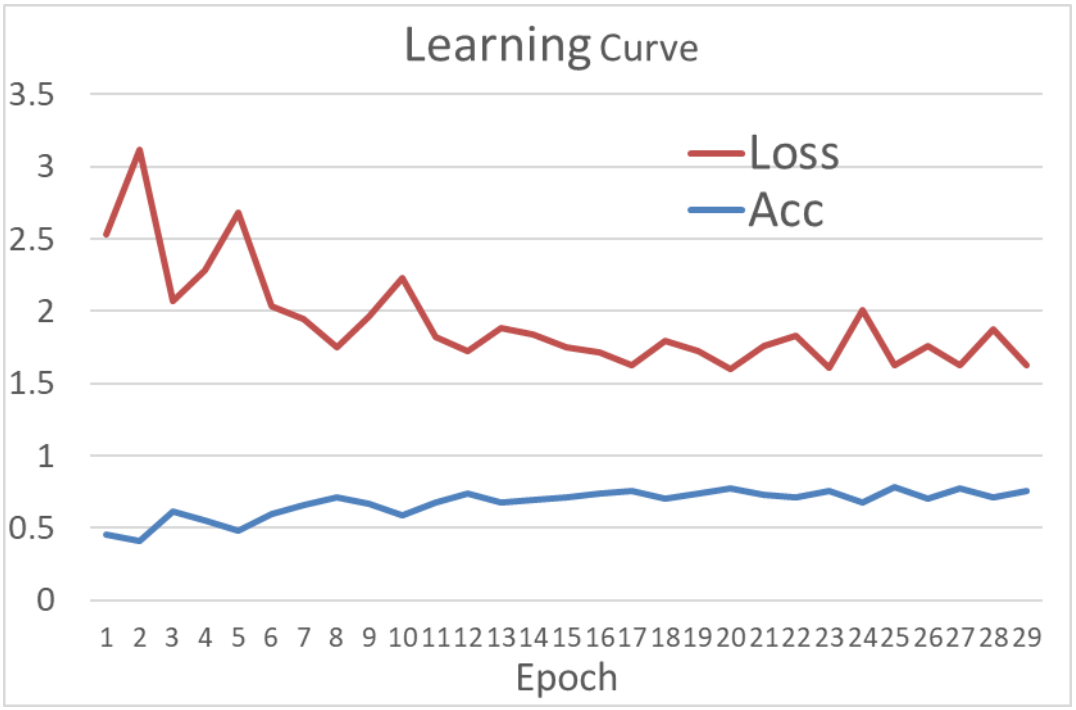
   Summary:

   1) Problem: Large model size& computational cost →Solu: HWGQ in feed forward
   2) Problem: Step-wise response leads to weak gradient when backward. →Solu: Using continuous approximation of operators(ReLU or tanh)
   3) Problem: mismatch in forward and backward → Solu: Clipped ReLU & Log-tailed ReLU when backward

   (Detailed notes attached below)

3. Learning Curve of Cifar_10

   (Currently unable to run on server, so for now only 30 epochs)

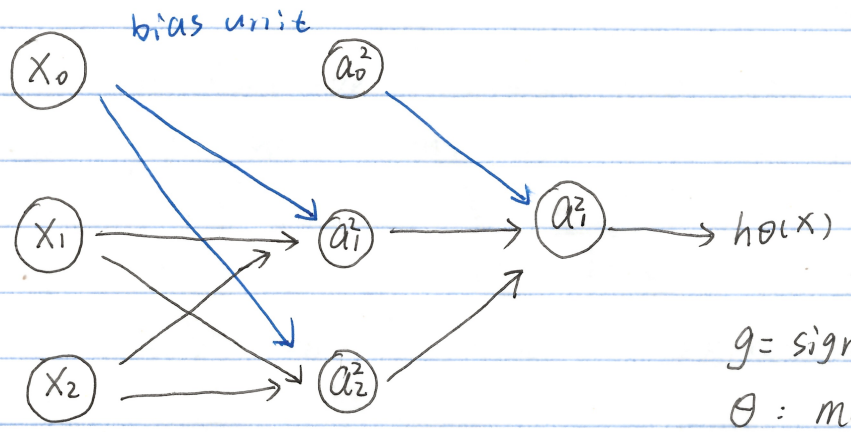


Backpropagation of XNOR

epoch.
One forward pass + one back pass

batch-size:
number of examples in one pass

iteration:
number of passes



g = sigmoid $= \frac{1}{1+e^{-z}}$

θ : matrix of weights

$$a_0^2 = g(\theta_{00}^{(1)} x_0 + \theta_{01}^{(1)} x_1 + \theta_{02}^{(1)} x_2) = g(\theta_0^T x) = g(z_0^2)$$
$$a_1^2 = g(\theta_{10}^{(1)} x_0 + \theta_{11}^{(1)} x_1 + \theta_{12}^{(1)} x_2) = g(\theta_1^T x) = g(z_1^2)$$
$$a_2^2 = g(\theta_{20}^{(1)} x_0 + \theta_{21}^{(1)} x_1 + \theta_{22}^{(1)} x_2) = g(\theta_2^T x) = g(z_2^2)$$
$$h_\theta(x) = a_1^3 = g(\theta_{10}^{(2)} a_0^2 + \theta_{11}^{(2)} a_1^{(2)} + \theta_{12}^{(2)} a_2^2)$$

$$g = \frac{1}{1+e^{-z}} \qquad g' = \frac{e^{-z}}{(1+e^{-z})^2} = g(1-g)$$

Forward Propagation:

$$x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} \qquad z^2 = \begin{pmatrix} z_0^2 \\ z_1^2 \\ z_2^2 \end{pmatrix}$$

$$z^2 = \theta^{(1)} \cdot x = \theta^{(1)} \cdot a^{(1)} \qquad a^2 = g(z^2)$$
$$z^3 = \theta^{(2)} a^2 \qquad h_\theta(x) = g(z^3)$$

Back Propagation:

$$\delta_j^3 = a_j^3 - y_j \quad \Rightarrow \quad \delta^3 = a^3 - y$$
$$\delta^2 = (\theta^2)^T \cdot \delta^3 \, g'(z^2)$$
$$= (\theta^2)^T \cdot \delta^3 \cdot a^2 \cdot (1 - a^2)$$

$$\frac{\partial J(\theta)}{\partial \theta_{ij}^L} = a_j^L \delta_i^{L+1}$$

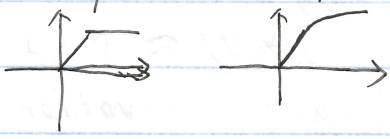
HWGQ.
caffe

Deep Learning with Low Precision by Half-wave Gaussian Quantization

Key: ① half-wave Gaussian quantizer in feedforward

② piece-wise continuous function in backpropagation

piecewise function = hybrid function



32× memory saving 32× computation saving

accuracy更高

hyperbolic-tangent "tanh"

1. Intro

Problem 1. large model size

2. large computational cost

⇓ solu:

compressed models: [quantization, low-rank matrix factorization, pruning, architecture design.] tips

binarization of activacations:

problem: step-wise response → weak gradient

⇓

solu: using continous approximation of operators

↳ problem: dismatch

⇓

Solu: view quantization operator to be 2 functions.

Relu ~~ta~~ than tanh: stronger gradient magnitude.

2. linearization, gradient clipping, gradient suppression

Related work:

1. redundancy of weights ···· (see above [ ])
2. weight binarization/quantization
3. activation quantization } speed up ↑ memory saving

3. Binary Networks.

① $z = g(w^T x)$ (1) complexity: 1. memory to store w (floating-point)
2. dot-product $w^T x$

② weight binarization

$$I * W \approx \alpha (I \oplus B) \quad (2)$$

③ binary activation quantization.

$$z = sign(x) = \begin{cases} +1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (3) \Rightarrow$$ simplify computation, increase difficulty of learning

$$\frac{\partial C}{\partial w} = \frac{\partial C}{\partial z} g'(w^T x) x \quad (4)$$

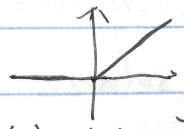
↓ if $g = sign$, $g'(\cdots) = 0$ everywhere

↓ solu: $g =$ hard tanh $\quad g'(\,) = \begin{cases} 1 & |x| \leq 1 \\ 0 & \text{otherwise} \end{cases}$ (5)

⇓ problem: gradient mismatch

4. HWGQ

① ReLU: $g(x) = \max(0, x)$ (6)



② Quantizer: $Q(x) = q_i$, if $x \in (t_i, t_{i+1}]$ (7)

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

$$Q^*(x) = \arg\min_Q E_x [(Q(x) - x)^2]$$

$$= \arg\min_Q \int p(x) (Q(x) - x)^2 dx \quad (9)$$

$p(x)$ → probability distribution

Lloyd's algorithm

dot-products tend to be Gaussian distribution

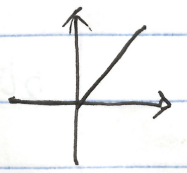
1. ReLU is half wave.

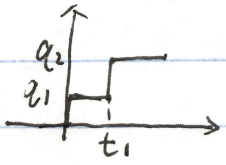
⇒ $Q(x) = \begin{cases} q_i & \text{if } x \in (t_i, t_{i+1}] \\ 0, & x \leq 0 \end{cases}$ → optimal parameters for Gaussian distribution (10)

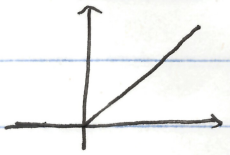
batch normalization: force layer to have zero mean, unit variance

③ Backward.

Vanilla ReLu: $\tilde{Q}'(x) = \begin{cases} 1, & \text{if } x>0 \\ 0, & \text{otherwise} \end{cases}$ (11)



forward: 

backward 

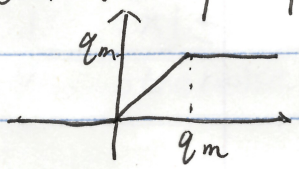
⇒ dismatch (large on the tail)

↓

solu: ① Clipped ReLU

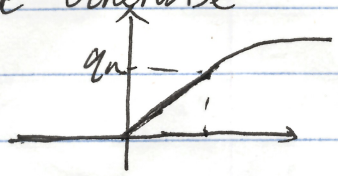
$Q' = \begin{cases} 0 \\ 1 \\ 0 \end{cases}$ $\tilde{Q}_c(x) = \begin{cases} q_m, & x>q_m \\ x, & x\in(0, q_m] \\ 0, & \text{otherwise} \end{cases}$

(con: loss of info)



② Log-tailed ReLU

$Q' = \begin{cases} \frac{1}{x-\tau} \\ 1 \\ 0 \end{cases}$ $\tilde{Q}_L(x) = \begin{cases} q_m + \log(x-\tau), & x>q_m \\ x, & x\in(0, q_m] \\ 0, & \text{otherwise} \end{cases}$



5. Experimental Results

① Full-precision Activation Comparison
② Low-bit Activation Quantization Results
③ Bakward Approximation Comparison
④ Bit-width Impact
⑤ Comparison with the state-of-the-art