ABC Net →

Idea is to approximate real valued weights W as a linear combination of binary weights.

$M \rightarrow$ # filters

$$W \approx \sum_{i=1}^{M} \alpha_i B_i$$

$$\min_{\alpha, B} J(\alpha, B) = \|w - B\alpha\|^2$$

$$S.T \quad B_{ij} \in \{-1, +1\}$$

$$B = [vec(B_1), \ldots vec(B_M)]$$

$$w = vec(W) \qquad \alpha = [\alpha_1, \ldots \alpha_M]^T$$

$$B_i = F_{ui}(W) := sign(\overline{W} + u_i \, std(W)) \qquad i = 1:M$$

$$\overline{W} = W - mean(W)$$

$$u_i = -1 + (i-1)\frac{2}{M-1} \qquad i = 1:M$$

weights are observed to be symmetric & have a non-sparse distribution close to a Gaussian.

as B_i's are fixed

$$\min_{\alpha} J(\alpha) = \|w - B\alpha\|^2$$

B_i's are fixed and are the basis/dictionary matrix.

C → cost function

Forward → $B_i = Fu_i(W)$

$$\alpha = \min_{\alpha} J(\alpha) = ||w - B\alpha||^2$$

$$0 = \sum_{m=1}^{M} \alpha_m \, Conv(B_m, A)$$

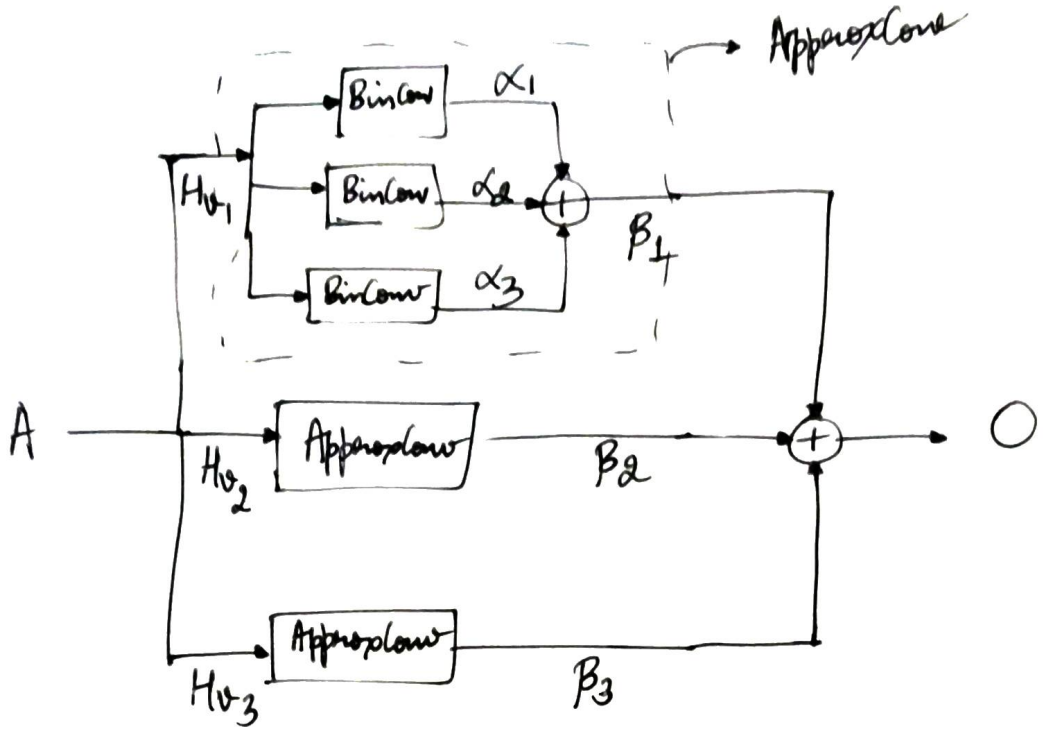
Backward → $$\frac{\partial C}{\partial W} = \frac{\partial C}{\partial O}\left(\sum_{m=1}^{M} \alpha_m \frac{\partial O}{\partial B_m}\frac{\partial B_m}{\partial W}\right)$$

Bengio's straight through estimator ← STE

$$\frac{\partial C}{\partial O}\left(\sum_{m=1}^{M} \alpha_m \frac{\partial O}{\partial B_m}\right)$$

$$= \sum_{m=1}^{M} \alpha_m \frac{\partial C}{\partial B_m}$$

STE is used to obtain gradient for a threshold operation. The threshold operation can't be differentiated. This can on average be aproximated as i/p gradient = o/p gradient.



i/p A is $\{-1, +1\}$ everything can be implemented as XNOR and BitCount.

H_{v_i} are used to binarize A.

↓

Follows DoReFa-Net paper and uses bounded rectifier for H_{v_i}.

→ Indicator func.

$$H_v(R) := 2\, \mathbb{I}_{h_v(R) \geq 0.5} - 1$$

$$h_v(x) = \text{clip}(x+v, 0, 1)$$

↓ shift.

Forward → $A = H_v(R)$

Backward → $\frac{\partial C}{\partial R} \overset{\text{STE}}{=} \frac{\partial C}{\partial A} \circ \mathbb{I}_{0 \leq R-v \leq 1}$

$\circ$ → Hadamard product.
Same as $\odot$

Binarization of losses is super noisy and crude. To make this work Batch Normalization (BN) is used before every activation to have zero-mean and unit variance.

Real valued activation R is approximated as a linear combination of N binary activations

$$R \approx \sum_{i=1}^{N} \beta_i A_i .$$

$$A_i = H_{v_i}(R)$$

β_i's and v_i's are both trainable.

$$Conv(W, R) \approx Conv\left(\sum_{m=1}^{M} \alpha_m B_m, \sum_{n=1}^{N} \beta_n A_n\right)$$

$$= \sum_{m=1}^{M} \sum_{n=1}^{N} \alpha_m \beta_n Conv(B_m, A_n)$$

Conv → Pool (Map.) → BN → Act.

$$H_v(BN(R)) = \begin{cases} +1, & aR + b \geq 0.5 - v \\ -1, & aR + b < 0.5 - v \end{cases}$$

$$= \begin{cases} +1, & R \geq (0.5 - v - b)/a \\ -1, & R < (0.5 - v, b)/a \end{cases}$$