Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or -1

2018-11-07

Overview

- BNN & BinaryNet
- 顾名思义，同时对weight和activation二值化
- update（accumulation）时使用real-valued variables

Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or −1
&&
BinaryNet: Training Deep Neural Networks with Weights and Activations Constrained to +1 or −1

- 两篇文章采用方法基本一致

- Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, Yoshua Bengio

- Matthieu Courbariaux, Yoshua Bengio

BNN & BinaryNet

- 使用Sign函数进行二值化 $x^b = \text{Sign}(x) = \begin{cases} +1 & \text{if } x \geq 0, \\ -1 & \text{otherwise.} \end{cases}$ (1)
- BinaryNet can be seen as a variant of Dropout.
- Activation：HardTanh
- Use **deterministic** rather than stochastic sampling of the bit（随机方法需要硬件生成随机数、耗时长）
- Propagation & update方式右图所示

{1. Computing the parameters' gradient:}
{1.1. Forward propagation:}
for $k = 1$ to L **do**
  $W_k^b \leftarrow \text{Sign}(W_k)$
  $s_k \leftarrow a_{k-1}^b W_k^b$
  $a_k \leftarrow \text{BatchNorm}(s_k, \theta_k)$
  if $k < L$ **then**
    $a_k^b \leftarrow \text{Sign}(a_k)$
  end if
end for
{1.2. Backward propagation:}
{Please note that the gradients are not binary.}
Compute $g_{a_L} = \frac{\partial C}{\partial a_L}$ knowing a_L and a^*
for $k = L$ to 1 **do**
  if $k < L$ **then**
    $g_{a_k} \leftarrow g_{a_k^b} \circ 1_{|a_k| \leq 1}$
  end if
  $(g_{s_k}, g_{\theta_k}) \leftarrow \text{BackBatchNorm}(g_{a_k}, s_k, \theta_k)$
  $g_{a_{k-1}^b} \leftarrow g_{s_k} W_k^b$
  $g_{W_k^b} \leftarrow g_{s_k}^\top a_{k-1}^b$
end for
{2. Accumulating the parameters' gradient:}
for $k = 1$ to L **do**
  $\theta_k^{t+1} \leftarrow \text{Adam}(\theta_k, \eta, g_{\theta_k})$
  $W_k^{t+1} \leftarrow \text{Clip}(\text{Adam}(W_k, \gamma_k \eta, g_{W_k^b}), -1, 1)$
  $\eta^{t+1} \leftarrow \lambda \eta$
end for

Details

- Sign函数导数几乎处处为0，使用HardTanh方便近似求导

$$\mathrm{Htanh}(x) = \mathrm{Clip}(x, -1, 1) = \max(-1, \min(1, x)) \quad (3)$$

- 在得到 $q = \mathrm{Sign}(r)$ 中q的梯度之后可以计算r的梯度

$$g_r = g_q 1_{|r|\leq 1}. \quad (2)$$

- 直接计算则r为0，所以近似计算，1|r|<1即为HardTanh

- 图片输入时将像素值变成8bit二进制编码，使用XNOR代替乘法，加速运算

Experiments

BNN

Table 1. Classification test error rates of DNNs trained on MNIST (MLP architecture without unsupervised pretraining), CIFAR-10 (without data augmentation) and SVHN.

Data set	MNIST	SVHN	CIFAR-10
Binarized activations+weights, during training and test			
BNN (Torch7)	1.40%	2.53%	10.15%
BNN (Theano)	0.96%	2.80%	11.40%
Committee Machines' Array (Baldassi et al., 2015)	1.35%	-	-
Binarized weights, during training and test			
BinaryConnect (Courbariaux et al., 2015)	1.29± 0.08%	2.30%	9.90%
Binarized activations+weights, during test			
EBP (Cheng et al., 2015)	2.2± 0.1%	-	-
Bitwise DNNs (Kim & Smaragdis, 2016)	1.33%	-	-
Ternary weights, binary activations, during test			
(Hwang & Sung, 2014)	1.45%	-	-
No binarization (standard results)			
Maxout Networks (Goodfellow et al.)	0.94%	2.47%	11.68%
Network in Network (Lin et al.)	-	2.35%	10.41%
Gated pooling (Lee et al., 2015)	-	1.69%	7.62%

BinaryNet

Method	Test error rate
Binary expectation backpropagation (Cheng et al., 2015)	2.12%
Bitwise Neural Networks (Kim & Smaragdis, 2016)	1.33%
BinaryConnect (Courbariaux et al., 2015)	1.18± 0.04%
BinaryNet (this work)	0.96%
Deep L2-SVM (Tang, 2013)	0.87%

Table 1. Test error rates of MLPs trained on the permutation invariant MNIST (without knowledge that inputs are images and without unsupervised learning) depending on the method. BinaryNet achieves near state-of-the-art results with only a single bit per weights and activations. This result suggests that augmenting the number of hidden units can compensate for the discretization noise.

Method	Test error rate
BinaryNet (this work)	11.40%
BinaryConnect (Courbariaux et al., 2015)	8.27%
Gated pooling (Lee et al., 2015)	7.62%

Table 2. Test error rates of ConvNets trained on the CIFAR-10 (without data-augmentation) depending on the method.