Week 3

Kexin Fan

- Another version of XNOR-Net

This task would be finished after vocation on Linux system in the school

- Backpropagation of quantization neural network

According to paper: XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks.

Forward propagation: Binarization->Input ->Hidden layer -> Output

Backpropagation: partial derivatives -> update weights -> binarization

Backpropagation of quantization neural network

According to XNOR-net paper

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

$J(B,\alpha) = \|W - \alpha B\|^2$ $\quad \alpha^*, B^* = \underset{\alpha,B}{\arg\min}\, J(B,\alpha)$

Binarizing weights: $A_{lk} = \frac{1}{n}\|W_{lk}^t\|_{l1}$ $\quad C(Y,\hat{Y})$ => cost function

$B_{lk} = \mathrm{sign}(W_{lk}^t)$ $\quad (I,Y)$ => input and targets

$\tilde{W}_{lk} = A_{lk}B_{lk}$ $\quad W^t$ => weights

Forward propagation $\tilde{W}_{lk} = \mathrm{sign}(W_{lk}^t)\frac{1}{n}\|W_{lk}^t\|$ $\quad \hat{Y}_i = f(\alpha_i)$ $\quad \alpha_i = \sum \tilde{W}_{lk} I_i$

Backpropagation $\frac{dC}{d\tilde{W}} = \frac{dC}{d\hat{Y}} \cdot \frac{d\hat{Y}}{d\alpha_i} \cdot \frac{d\alpha_i}{d\tilde{W}_{lk}}$ => gradient descent

Update weights $W_{t+1} = W_{t(old)} - \frac{\eta}{N}\sum_{i=1}^{N} \frac{dC_i}{dW_i} = W_{t(old)} - \eta \sum_{i=1}^{N} \frac{dC_i}{dW_i}$

- Report learning curve of CIFAR-10 of XNOR-Net

The Tensorflow version of XNOR-Net:

Since the training speed is too slow, there are only ten times over one day.