Week 2 Report

Weitian Li weitian.li@rutgers.edu

1. XNOR-Net/util.py Function Version: 0.3

完成情况：由于代码较长就不放进来报告，简单解释一下 util 包里有多少个函数，什么功能。此 util 包有 7 个函数，主要是为 XNOR-net 提供二值化、保存参数等功能。

(1) 定义部分。

记录卷积层数量以及二值化的范围，定义保存参数的 list。

(2) binarization

调用 self.meancenterConvParams()，self.clampConvParams()，self.save_params()，self.binarizeConvParams() 等函数进行二值化的定义和权值的保存。

(3) meancenterConvParams

通过获取网络层的参数个数，

(4) clampConvParams

对参数进行大小比较，返回-1 或 1 值。

(5) save_paramsz.　在。不能说话。。

从目标模块保存参数。

(6) binarizeConvParams

对卷积层参数进行二值化，但是目前这个式子 m = self.target_modules[index].data.norm(1, 3, keepdim=True)\.sum(2, keepdim=True).sum(1, keepdim=True).div(n) 我还没搞懂。

(7) restore

保存数据和参数

(8) updateBinaryGradWeight

更新二值化的梯度权重，里面的计算过程还在了解中。

问题：1．在理解二值化的数值计算操作上遇到一定的困难。

2. Simple BP in traditional network

PDF

完成情况：完成。

3.BP in XNOR-net

PDF

完成情况：BNET 完成，XNOR-net 还未完成。

问题：理解 XNOR-net 的二值化的求导有些不通。

4.Run XNOR-net on CIFAR (Tensorflow Version)

Reference: https://github.com/ljhandlwt/xnor-net-tf

完成情况：引用了 https://github.com/ljhandlwt/xnor-net-tf 的代码并运行，这是一个 tf 版的 XNOR-net，数据库是 CIFAR-10，epoch 预设是 320，但是我只跑了 269，因为后面明显出现了过拟合。

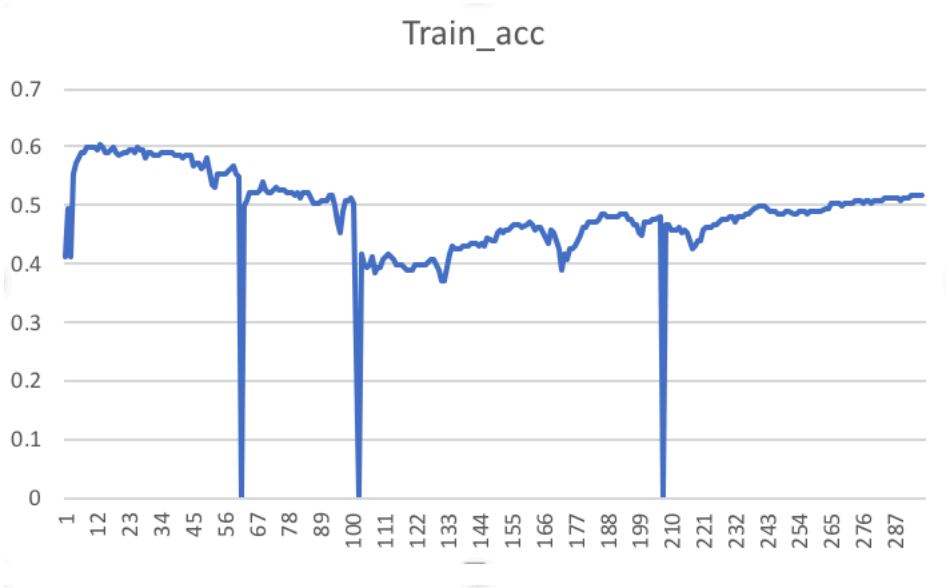
参数： --learn_rate', default=0.01

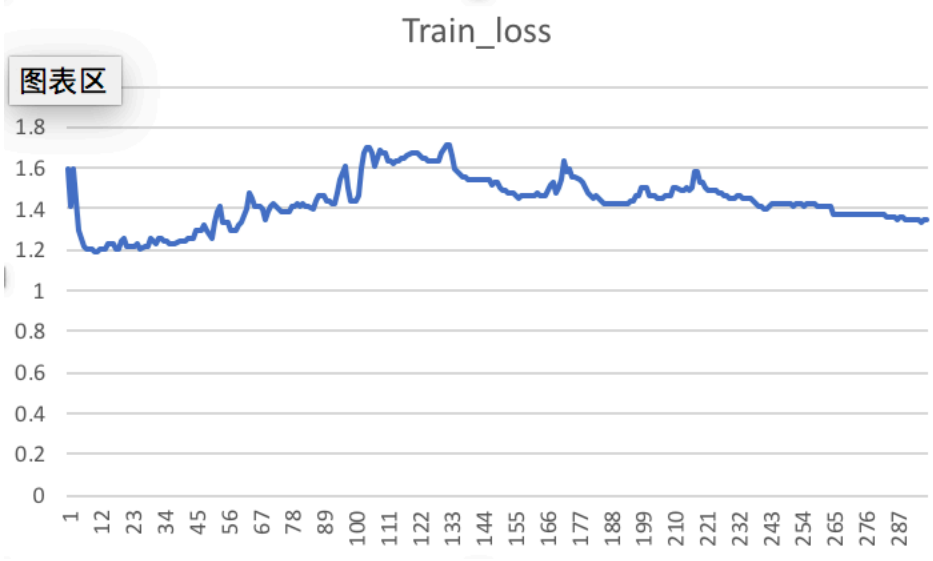
--epochs', default=320

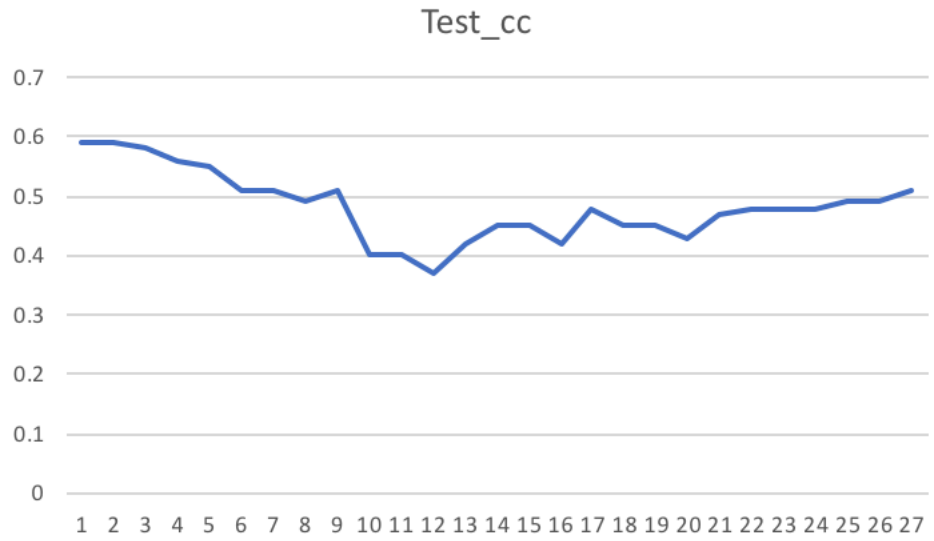
--eps', default=1e-8

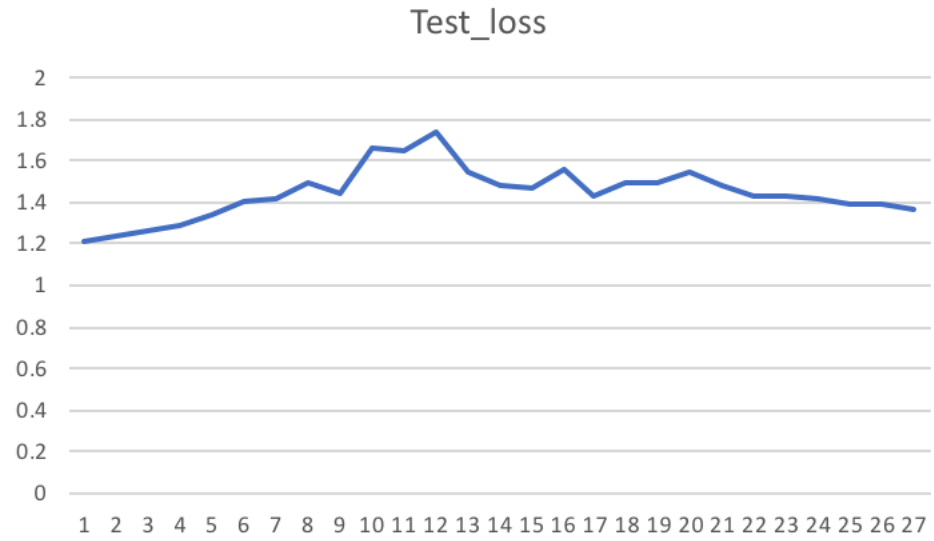
--weight_decay', default=1e-5

--batch_size', default=128









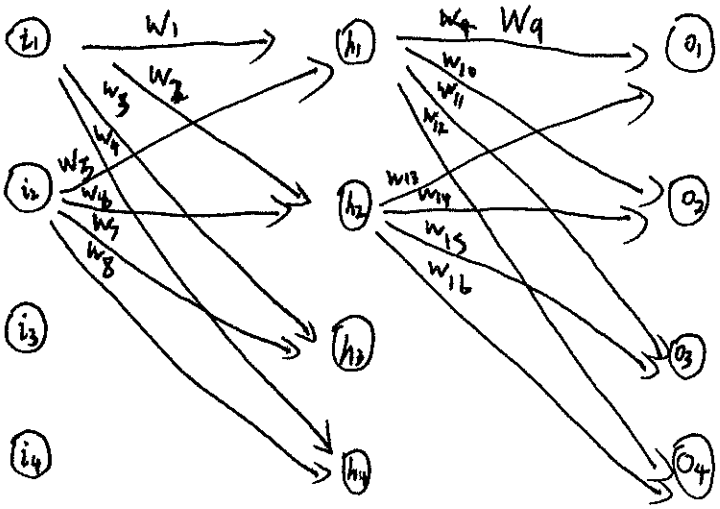
问题：1. Test_acc 一开始一直在 0.1，后来通过修改以下语句以及 epoch 勉强得到值。

```
test_acc, test_loss = run_epoch(epoch, model, sess, dtest, is_train=True)
```

2.训练到了一半出现很明显的过拟合，测试集的准确率过了一半明显下降。调整学习率到 1e-4 等效果变好一点，但是还是会很明显的出现过拟合现象。

3. 达不到文章的效果。

BP In Traditional Networks



INPUT

INPUT: $i_1 = 0.05$, $i_2 = 0.1$, i_3, i_4

LABEL OUTPUT: $o_1 = 0.01$, $o_2 = 0.99$, o_3, o_4

Initial weight:

$w_1 = 0.01$, $w_2 = 0.4$, $w_3 = 0.5$, $w_4 = 0.6$,

$w_5 = 0.08$, $w_6 = 0.04$, $w_7 = 0.5$, $w_8 = 0.8$, ...

Activation Function: sigmoid $= \frac{1}{1+e^{-x}}$

η: learning rate

STEP 1: INPUT → Hidden → OUTPUT (Forward Propaga...)

h_1: $w_1 \times i_1 + w_3 \times i_2 + w_{i_3 \to h_1} \times i_3 + w_{i_4 \to h_1} \times i_4 = net_{h_1}$

$out_{h_1} = \frac{1}{1+e^{-net_{h_1}}}$

h_2: $w_2 \times i_1 + w_6 \times i_2 + w_{i_3 \to h_2} \times i_3 + w_{i_4 \to h_2} \times i_4 = net_{h_2}$

~~out~~ $out_{h_2} = \frac{1}{1+e^{-net_{h_2}}}$

⋮

O_1:

$out_{h_1} = i_{h_1}$, $out_{h_2} = i_{h_2}$ ...

$i_{h_1} \times w_9 + i_{h_2} \times w_{13} + i_{h_3} \times w_{h_3 \to o_1} + i_{h_4} \times w_{h_4 \to o_1} = net_{o_1}$

$out_{o_1} = \frac{1}{1+e^{-net_{o_1}}}$

O_2:

$i_{h_1} \times w_{10} + i_{h_2} \times w_{14} + $ ~~$i_{h_3} \times w_{h_3 \to o_2}$~~ $+ i_{h_3} \times w_{h_3 \to o_2} + i_{h_4} \times w_{h_4 \to o_2} = net_{o_2}$

$out_{o_2} = \frac{1}{1+e^{-net_{o_2}}}$

⋮

STEP 1 Forward Propagation Done.

LABEL (O_1, O_2, O_3, O_4 ...)

OUTPUT (out_{o_1}, out_{o_2}, out_{o_3}, out_{o_4} ...)

STEP2 Back Propagation

① Square Error. $E_{total} = \sum (E_{o_1} + E_{o_2} \cdots) = \sum \frac{1}{2}(target - output)^2$

$E_{o_1} = \frac{1}{2}(O_1 - out_{o_1})^2 = \frac{1}{2}(0.01 - out_{o_1})^2 = e_1$

$E_{o_2} = \frac{1}{2}(O_2 - out_{o_2})^2 = \frac{1}{2}(0.99 - out_{o_2})^2 = e_2$

$E_{total} = e_1 + e_2 + \cdots = e_{total}$

② UPDATE Hid → OUTPUT weight

w_9: $\frac{\partial E_{total}}{\partial w_9} = \frac{\partial E_{total}}{\partial out_{o_1}} \times \frac{\partial out_{o_1}}{\partial net_{o_1}} \times \frac{\partial net_{o_1}}{\partial w_9}$

$= \left(-2 \times \frac{1}{2}(target_{o_1} - out_{o_1})^{\#}\right) \times \left(out_{o_1}(1 - out_{o_1})\right)$

$\times \left(out_{h_1} \times w_9^{\circ}\right) = E'_{w_9}$

$E'_{w_9} = -(target_{o_1} - out_{o_1}) \times out_{o_1}(1 - out_{o_1}) = \delta_{o_1} \times out_{h_1}$

UPDATE: $w_{9new} = w_9 - \eta * \frac{\partial E_{total}}{\partial w_9} = w_9 - \eta \delta_{o_1} \times out_{h_1}$.

⋮

③ UPDATE Hid → Input weight

$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h_1}} \times \frac{\partial out_{h_1}}{\partial net_{h_1}} \times \frac{\partial net_{h_1}}{\partial w_1}$

$= \left(\frac{\partial E_{o_1}}{\partial out_{h_1}} + \frac{\partial E_{o_2}}{\partial out_{h_1}}\right) \times \frac{\partial out_{h_1}}{\partial net_{h_1}} \times \frac{\partial net_{h_1}}{\partial w_1}$

$= (E'_{h_1} + E'_{h_2}) \times out_{h_1}(1 - out_{h_1}) \times i_1 = E'_1$

$\frac{\partial E_{o_1}}{\partial out_{h_1}} = \frac{\partial E_{o_1}}{\partial net_{o_1}} \times \frac{\partial net_{o_1}}{\partial out_{h_1}} = E_{o_1} E'_{h_1}$

UPDATE: $w_{1new} = w_1 - \eta \frac{\partial E_{total}}{\partial w_1} = w_1 - \eta E'_1$

⋮

Back Propagatim Done

KEEP Iteration

BP In XNOR-NET

Binary Neural Networks

Formula: $I \times W \approx (I \oplus B)\alpha$, B is binary,

I: Input, W: weight, alpha: argmin, $\oplus$: no mupl, Conv, $B = Sign(W)$

$W \approx \alpha B$

① FORWARD

1): Binary weight $\tilde{W}$

For each $\tilde{w} = Sign(W)$

Hidden

h_1: ~~$\tilde{W}_1 x i_1 + \tilde{W}_2 x i_2 \cdots$~~ $= net_{h_1}$

$h_1 = f(net_{h_1})$, $f(x)$ is activation

$h_1 = Batch\ Norm(h_1)$

$\tilde{h}_1 = Sign(h_1)$

Output

O_1: ~~$\tilde{h}_1 x \tilde{w}_{h_1} + \tilde{h}_2 x \tilde{w}_{h_2} = out_{o_1}$~~ net_{o_1}

For each, $\tilde{w}_{h_1} = sign(W_{h_1})$

$Out_{o_1} = f(net_{o_1})$

② BP

h_1: $\frac{\partial E_{total}}{\partial \tilde{W}_{h_1}} = \frac{\partial E_{total}}{\partial out_{h_1}} \times \frac{\partial out_{h_1}}{\partial net_{h_1}} \times \frac{\partial net_{h_1}}{\partial w_{h_1}}$

$= \frac{\partial E_{total}}{\partial out_{h_1}} \times \frac{\partial out_{h_1}}{\partial net_{h_1}} \times H\tanh(x)$

$H\tan(x) = Clip(x, -1, 1) = \max(-1, \min(1, x))$

BUT UPDATE is to UPDATE before binary

$W_{h_1 new} = W_{h_1} - \eta \frac{\partial E_{total}}{\partial \tilde{W}_{h_1}}$, $W_{h_1 new}$ will binary in Forward Propagation