Assignment 2 Theory Problem Set

DO NOT TAG

Name: Jing Yu
GT Email: jyu497@gatech.edu

Theory PS Q1.

$$\begin{bmatrix} W_{(0,0)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & W_{(0,2)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & W_{(2,0)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & W_{(2,2)} \end{bmatrix} \Rightarrow A$$

$$X = [X_{(0,0)}, X_{(0,1)}, X_{(0,2)}, X_{(1,0)}, X_{(1,1)}, X_{(1,2)}, X_{(2,0)}, X_{(2,1)}, X_{(2,2)}]^T$$

$$\Rightarrow Y = AX = [W_{(0,0)}X_{(0,0)}, W_{(0,2)}X_{(0,2)}, W_{(2,0)}X_{(2,0)}, W_{(2,2)}X_{(2,2)}]^T$$

Theory PS Q2.

$$x_0 = 2 \Rightarrow W = 6, b = 3$$
$$x_0 = -1 \Rightarrow W = 1.5, b = 3$$
$$x_0 = 1 \Rightarrow W = 6, b = 3$$

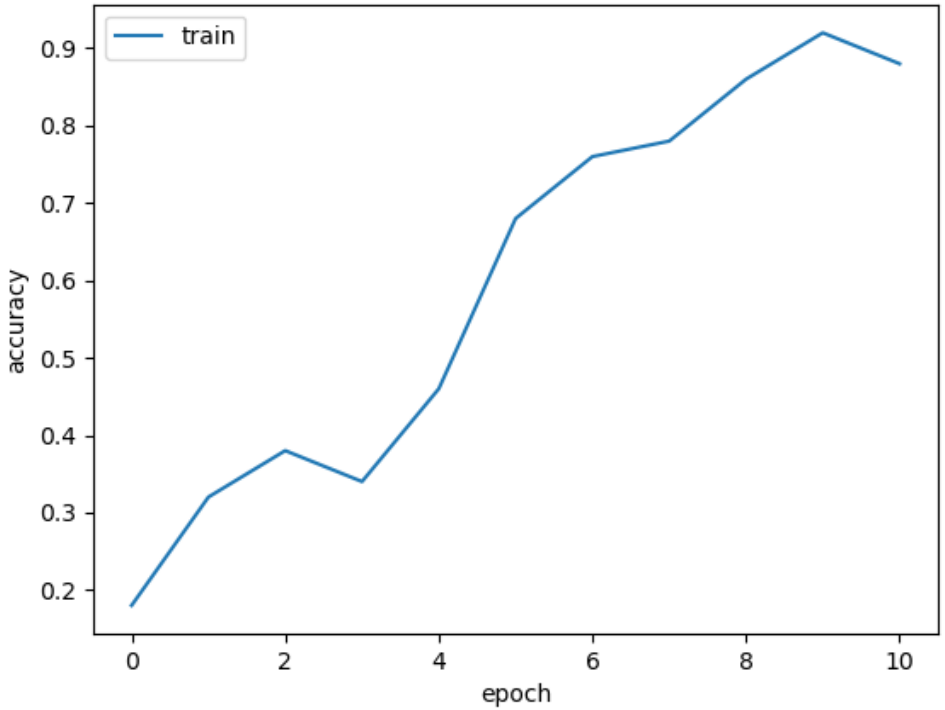
Assignment 2 Writeup

DO NOT TAG

Part-1 ConvNet

DO NOT TAG

Put your learning curve here:



My CNN Model

DO NOT TAG

Describe and justify your model design in plain text here:

Referred to LeNet and AlexNet architectures.

Cov1: kernel 64@3x3 => feature maps 64@30x30. Since the input images are 32x32, too large kernels are not appropriate

maxPool1: kernel 2x2, stride=2 => feature maps 64@15x15

Cov2: kernel 128@3x3 => feature maps 128@13x13

maxPool2 : kernel 3x3, stride=2 => feature maps 128@6x6

Cov3: kernel 256@3x3 => feature maps 256@4x4. 4x4 is already small, no more pooling layer to keep enough information.

Cov4: kernel 64@1x1 => feature maps 64@4x4. Decrease output channel number to avoid too large fully connected layer.

Fc1: input 1024, output 384

Fc2: input 384, output 192

Fc3: input 192, output 10

ReLU activation to accelerate the speed of the training process

Describe and justify your choice of hyper-parameters:
Batch_size: 128
Learning-rate: 0.05
Regularization: 0.0004
Epochs: 70
Steps: [42, 56]
Momentum: 0.9
Learning curve is helpful in tuning this parameters, especially for learning-rate, regularization term and epochs. As epoch increases, the loss decreases, the steps are used to decrease the learning-rate to avoid diverge.

What’s your final accuracy on validation set?
0.7857

Data Wrangling

DO NOT TAG

What's your result of training with regular CE loss on imbalanced CIFAR-10?

Fill in your per-class accuracy in the table

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
CE Loss	0.9110	0.3680	0.3150	0.0210	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

What's your result of training with CB-Focal loss on imbalanced CIFAR-10?

Tune the hyper-parameter beta and fill in your per-class accuracy in the table

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
beta=0.9999	0.2220	0.3020	0.0090	0.1690	0.3680	0.0080	0.0000	0.1240	0.0140	0.2730
beta=0.999	0.4530	0.0000	0.0000	0.0010	0.1690	0.0000	0.3570	0.0010	0.4180	0.0000
beta=0.99	0.8550	0.5980	0.3660	0.0130	0.0000	0.0000	0.0010	0.0000	0.0000	0.0000
beta=0.9	0.8090	0.6630	0.3070	0.0140	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Put your results of CE loss and CB-Focal Loss(best) together:

	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
CE Loss	0.9110	0.3680	0.3150	0.0210	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
CB-Focal	0.8550	0.5980	0.3660	0.0130	0.0000	0.0000	0.0010	0.0000	0.0000	0.0000

Describe and explain your observation on the result:

Focal loss reduce the relative loss for well-classified samples and focus on difficult samples.

The class-balanced loss uses a factor that is inversely proportional to the effective number of samples, influences the loss function by assigning relatively higher costs to examples from minor classes.

From the results, Resnet with CE loss performs poorly for weakly represented classes, the model is biased toward dominant classes. While the Resnet with CB Focal loss decreses this dominant effects. It also proves that beta = 0 corresponds to no re-weighting and beta $\rightarrow$ 1 corresponds to re-weighing by inverse class frequency. With larger beta, the major classes are more penalized.

I also observe that CB loss by itself performs better than Focal loss. It better addresses the class imbalance issue for this long-tailed data distribution.