CSCE 489: Machine Learning (Spring 2019)

Homework #6

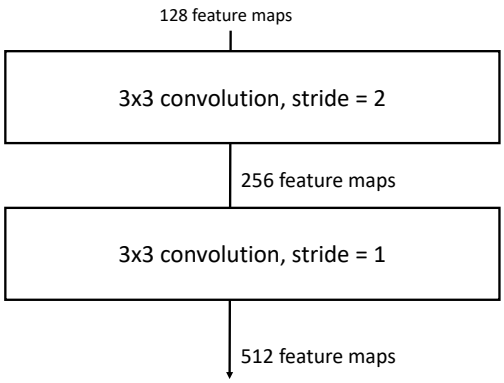
Due 5/3/2019 before final exam

---

1. You need to submit a report in hard-copy before lecture and your code to eCampus. Your hard-copy report should include (1) answers to the non-programming part, and (2) results of the programming part. Your submission to eCampus should be your code files ONLY. Please put all your code files into a compressed file named "HW#_FirstName_LastName.zip"
2. Hard-copy is due in class before the final exam, and code files are due before final exam to eCampus on the due date.
3. Unlimited number of submissions are allowed on eCampus and the latest one will be graded. If you make a resubmission after the due date, it will be considered late.
4. Please read and follow submission instructions. No exception will be made to accommodate incorrectly submitted files/reports.
5. All students are highly encouraged to typeset their reports using Word or LaTeX. In case you decide to hand-write, please make sure your answers are clearly readable.
6. The problems in red are for students in the honors section ONLY and should be submitted together with other parts of HW6. The final score for students in the honors section will be 0.85 * (score of regular problems) + 0.15 * (score of honor problems).

---

1. (15 points) Given the convolutional neural network block as below



   (a) What is the total number of parameters in the block (you can skip bias terms)?

2. (35 points) Using batch normalization in neural networks requires computing the mean and variance of a tensor. Suppose a batch normalization layer takes vectors $z_1, z_2, \cdots, z_m$ as input, where m is the mini-batch size. It computes $\hat{z}_1, \hat{z}_2, \cdots, \hat{z}_m$ according to

$$\hat{z}_i = \frac{z_i - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

where

$$\mu = \frac{1}{m}\sum_{i=1}^{m} z_i, \quad \sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(z_i - \mu)^2.$$

It then applies a second transformation to obtain $\tilde{z}_1, \tilde{z}_2, \cdots, \tilde{z}_m$ using learned parameters γ and β as

$$\tilde{z}_i = \gamma \hat{z}_i + \beta.$$

In this question, you can assume that $\epsilon = 0$.

(a) (10 points) You forward-propagate a mini-batch of $m = 4$ examples in your network. Suppose you are at a batch normalization layer, where the immediately previous layer is a fully connected layer with 3 units. Therefore, the input to this batch normalization layer can be represented as the below matrix:

$$\begin{bmatrix} 12 & 14 & 14 & 12 \\ 0 & 10 & 10 & 0 \\ -5 & 5 & 5 & -5 \end{bmatrix}$$

What are $\hat{z}_i$? Please express your answer in a 3×4 matrix.

(b) (10 points) Continue with the above setting. Suppose $\gamma = (1, 1, 1)$, and $\beta = (0, -10, 10)$. What are $\tilde{z}_i$? Please express your answer in a 3×4 matrix.

(c) (10 points) Describe the differences of computations required for batch normalization during training and testing.

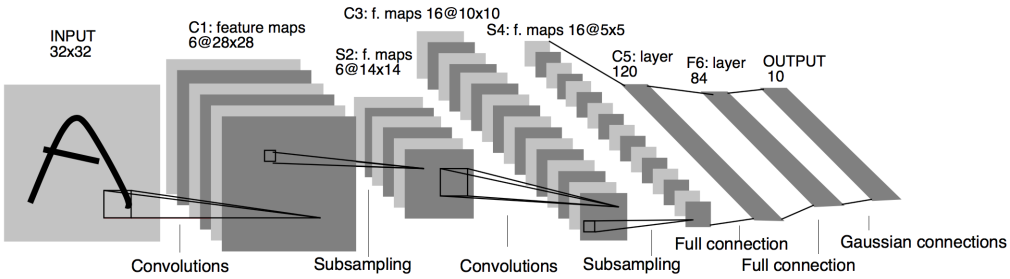
(d) (5 points) Describe how the batch size during testing affect testing results.

3. (50 points) **LeNet for Image Recognition:** In this coding assignment, you will need to complete the implementation of LeNet (LeCun Network) using Tensorflow and apply the LeNet to the image recognition task on Cifar-10 (10-classes classification). The access to the Cifar-10 Dataset are here (https://www.cs.toronto.edu/kriz/cifar.html). In addition, you will need to install the python packages "tqdm" and "tensorflow". The installation guides of Tensorflow are in "readme.txt". Please read carefully and follow the instructions. You are expected to implement your solution based on the given codes. The only file you need to modify is the "solution.py" file. You can test your solution by running the "main.py" file. For the honor section, we recommand you finish the basic parts first and then working on the honor section. You can duplicate the "solution.py" file and then finish them seperately.

(a) (10 points) Download and extract the Cifar10 Dataset from the link above. Put the data folder "cifar-10-batches-py" in the same directory of "code". Read carefully the instructions and then complete the function ***load_data()***.

(b) (10 points) Complete the function ***preprocessing()***, which rescales the image pixels from range 0-255 to range 0-1.
Honor section (30 points): you will need to replace the rescaling with normalization. In particular, to centralize and rescale each image using its mean and variance.

(c) (20 points) Complete the class ***LeNet_Cifar10()***. In particular, only the function ***_network()*** in the class is needed to be filled. The paper for LeNet can be found here (http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf) The network architecture is shown in the figure below.



The subsampleing is implemented by using the max pooling. And the kernel size for all the convolutional layers are 5×5. The sequential layers are:

Inputs $\rightarrow$
Convolution (6 out channels) $\rightarrow$ **Max Pooling** $\rightarrow$
Convolution (16 out channels) $\rightarrow$ **Max Pooling** $\rightarrow$
Reshape to vector $\rightarrow$ **Fully-connected (120 out units)** $\rightarrow$
Fully-connected (84 out units) $\rightarrow$ **Outputs (n_classes out units)**

For this part, you are only allowed to use the APIs in ***tf.layers*** and ***tf.nn***. Please refer to the Tensorflow API documents below for the usage of those APIs before you use them:
https://www.tensorflow.org/api_docs/python/tf/layers and
https://www.tensorflow.org/api_docs/python/tf/nn
Honor section (40 points): Please also include the batch normalization (after each convolutional layers and fully-connected layer except for the output layer) and the dropout (before the output layer) in the network. Be careful with the difference between the training phase and the other phrases. You will need to modify the ***feed_dict***s for training, validation and testing to specify the phrase.

(d) (10 points) Try to read and understand the other parts of the class ***LeNet_Cifar10()***. Run the main.py to train and test the model. Include your training log and test accuracy, as well as a short analysis of the results in your report. The training and testing may take 10 - 30 minutes to finish on a CPU (20 epochs, 30-90 seconds per epoch).

Honor section (30 points): Run the basic model (i.e., rescaling only and the basic LeNet) and the model with additional setting (i.e., image normalization only and the LeNet with Dropout and Batchnorm). Compare the results of the two models and analyze why there is an improvement on (or why it harms) the performance.