Project: a Simple Run of Stochastic Gradient

Last updated: March 10, 2021
Goal

- A basic understanding of how stochastic gradient is used to train CNN
- Familiar with our simpleNN package
- So at this moment we don’t care the algorithm and implementation details yet. We just want to learn how to use it
Download (or clone) the code simpleNN at https://github.com/cjlin1/simpleNN
We will use the Python part for this project.
Use CPU!!
Later we will do timing comparisons with MATLAB/Octave code on CPU
To use the package you need to install Tensorflow
Follow, for example, instructions at https://www.tensorflow.org/install/pip?lang=python3
The development was done on Linux. We recommend you to do the same.

Read README in detail

Then make sure first you can run the training and prediction examples shown in README

You should be able to install and run the software without problem as we have been using it for a while

However, if you really have questions, you can just ask them through Github

Consider the following sets

- MNIST
CIFAR10
from the LIBSVM data set https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
Please use data in the MATLAB format. Our python code can read them
Training and test sets are available
Run these two sets by the simple stochastic gradient algorithm. That is, use SGD rather than Adam
For the architecture, let’s do
### Project Contents IV

<table>
<thead>
<tr>
<th>Our symbol</th>
<th>filter size $h \times h \times d^m$</th>
<th>#filters $d^{m+1}$</th>
<th>stride $s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv 1</td>
<td>$5 \times 5 \times 3$</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td>pool 1</td>
<td>$2 \times 2$</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>conv 2</td>
<td>$3 \times 3 \times 32$</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td>pool 2</td>
<td>$2 \times 2$</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>conv 3</td>
<td>$3 \times 3 \times 64$</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td>pool 3</td>
<td>$2 \times 2$</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>full 1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- This network has been implemented as the one called CNN_4layers
Thus you don’t need to handle the network at this moment.

To see details you can trace net/net.py. But you can do the project without knowing details now.

For other options, use the default values.

Don’t copy the parameter setting --lr 0.01 --C 0.01 --bsize 256 in the example shown on the software page.

No need to set them as we want to use the default values. You can check train.py to see our default values.
This is a project but not a homework. We don’t expect the same result.

A minor difference in your code or your settings may cause your results slightly different from those of others.

To run the default 500 epochs, it may take several hours.

Don’t complain on this.

You should start early.
Later in this course we will do more on analyzing code/results instead of doing such long training processes.
For the padding size, we avoid the shrinkage of the output image in each convolutional layer by

$$a^m_{\text{conv}} = a^m. \quad (1)$$

For the convolution operation, we enlarge $a^m$ to $a^m_{\text{pad}}$ so that

$$a^m = a^m_{\text{conv}} = \left\lfloor \frac{a^m_{\text{pad}} - h}{s} \right\rfloor + 1.$$
Thus

\[ a^m = \left\lfloor \frac{2p + a^m - h}{s} \right\rfloor + 1. \]

Because \( s = 1 \) in our setting, we can let the padding size be

\[ p = \frac{h - 1}{2} \]

so that (1) holds.
For activation function, we use

\[ \sigma(x) = \max(x, 0) \]

(2)

for convolution layers and a linear function for the last full layer

\[ \sigma(x) = x. \]

(3)
Your Report I

- Write a report with $\leq 2$ pages in pdf
- In your report, beside the training/prediction results, you may discuss the following issues
  - Your environment and any difficulties on installation
  - Difficulties in using the package? Which part you think is not very friendly?
  - Running time
  - and anything you think is interesting
- No need to write lots of things. What I will check are
Your Report II

- insight of your observations
- whether your argument is clear and logical
- organization and English writing of your report

Those writing a clear report often get better scores than those getting better accuracy or lots of results but having a badly written report

Another note is that you want to well organize your code directory as in this course you will do many projects
Students selected for presentation please do a 10-minute talk (9-minute the contents and 1-minute Q&A)

Please submit your presentation slides (in pdf) before the class

People not chosen for presentation do not need to prepare/submit slides

You can use your computer for the presentation, but please use the submitted version of slides
You may want to have your code available in your computer. To answer some of our questions, showing the code is easier.