

Introduction to Deep Learning for Computer Vision

Assignment 6: Debug a deep neural network

Due: March 24th, 2020

Abstract

This is the final assignment! In this assignment We will try to help Bob to debug his convolutional neural network.

Please read the instructions carefully.

1 Instructions

1.1 Submission package

Within the homework assignment package, there should be a `submission-package.zip`, which contains the directory structure and empty files for you to get started. Please edit the contents within the file for code, and then create a `zip` archive with the file name `submission-package.zip`, and submit it. **Do not use other archive formats such as rar or tar.gz.** Also, submit the report in `pdf`, **as a separate file alongside** the archive with the codes.

All assignments should be submitted electronically. Hand written reports are **not** accepted. You can, however, include scanned pages in your report. For example, if you are not comfortable with writing equations, you can include a scanned copy.

1.2 Assignment Report

For this assignment, all execution results should be summarised in an assignment report. Reports should be in `pdf` format. Though not mandatory, students are encouraged to submit their reports written in `LATEX`. In the assignment package, you should have been given an empty skeleton file for you to get started.

However, it is **not required** for you to explain your code in the reports. Reports are for discussing results. You **should** however, provide comments in your code, well enough to be understood by directly reading the code.

1.3 Code

All assignments should be in `Python 3`. Codes that fail to run on `Python 3` will receive 20% deduction on the final score. In other words, do **not** use `Python 2.7`.

For this assignment, you should **not** need to create additional files. Fill in the skeleton files in the submission package. Do **not** change the name of these scripts.

It is **strongly encouraged** to follow PEP8. It makes your code much more readable, and less room for mistakes. There are many open source tools available to automatically do this for you.

1.4 Delayed submission

In case you think you will not meet the deadline due to network speed or any other reasons, you can send an email with the `SHA-256` hash of your `.zip` archive first, and then submit your assignment through email later on. This will **not** be considered as a delay.

Delayed submissions are subject to 20% degradation per day. For example, an assignment submitted 1 minute after the deadline will receive 80% of the entire mark, even if it was perfect.

Submission will close 24 hours after the original deadline, as the solution for the assignment will be released to be used for the next assignment.

1.5 Use of open source code

Any library under any type of open source license is allowed for use, given full attribution. This attribution should include the name of the original author, the source from which the

code was obtained, and indicate terms of the license. Note that using copyrighted material without an appropriate license is not permitted. Short snippets of code on public websites such as StackOverflow may be used without an explicit license, but proper attribution should be given even in such case. This means that if you embed a snippet into your own code, you should properly cite it through the comments, and also embed the full citation in a LICENSES file. However, if you include a full, unmodified source, which already contains the license within the source file, this is unnecessary. Please note that without proper attribution, *it will be considered plagiarism*.

In addition, as the assignments are intended for you to learn, (1) if the external code implements the core objective of the task, no points will be given; (2) code from other CSC486B/CSC586B students will count as plagiarism. To be more clear on (1), with the assignment, we will release a `requirements.txt` file, that you can use with `pip` to setup your environment. On top, you are also allowed to use `OpenCV3.X`, which we will not include in `requirements.txt` as `OpenCV` installation depends on your own framework.

2 Help Bob to debug

This assignment is about debugging.

Bob is trying to train a deep neural net to classify CIFAR10 images. After some development, the code runs, there is no runtime error reported, and Bob is very happy.

2.1 Hyperparameters (30 points)

However, during training, he got a warning "Warning: NaN or Inf found in input tensor." Here is his command:

```
python solution.py --learning_rate=10 --mode=train --rep_intv=1 --conv2d=torch
```

Please find which hyperparameter is wrong?

Hint: This is a single hyperparameter bug. Make `--data_dir` point to CIFAR10 on your disk.

2.2 model.py/MyNetwork/forward (30 points)

After fixing his hyperparameters, Bob is able to get rid of the warning. However, the accuracy is still close to a random guess, around 10%.

Alice, Bob's friend, told Bob that she can get around 60% accuracy with the same hyperparameters and exactly the same architecture.

Here is Alice's and Bob's command(hyperparameters in white color):

```
python solution.py
```

Here is Alice's and Bob's tensorboard:
What did Bob do wrong in his code?

Hint: Please search for keyword BUG in the source code. This is a single line bug. Make `--data_dir` point to CIFAR10 on your disk.

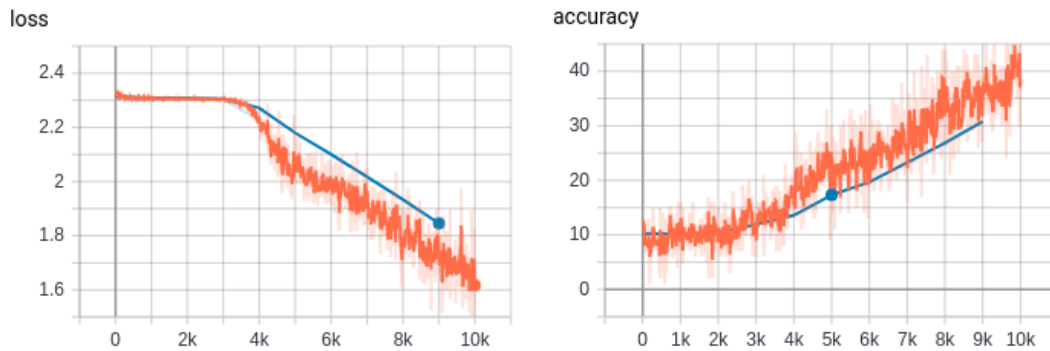


Figure 1: Alice's tensorboard

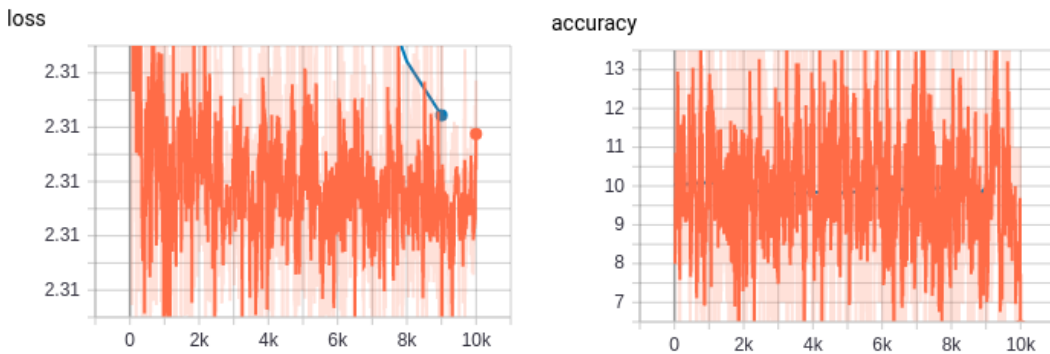


Figure 2: Bob's tensorboard

2.3 solution.py/test (40 points)

After fixing this problem, Bob wanted to try a fancier architecture, he changed `--conv2d` option to "custom". Meanwhile, he found a pretrained weights on the Internet, which claim to has 66% accuracy on CIFAR10 test set. Bod downloaded the pretrained weights, and loaded the weights, however, he only got 28% accuracy.

Here is his command:

```
python solution.py --mode=test --conv2d=custom --batch_size=1
```

However, if he change the batch size to 100, he can get 67% accuracy. What did Bod do wrong in his code? What is the reason causing the different accuracy(28% and 67%) using different batch size with the same weights?

Hint: Please search for keyword BUG in the source code. This is a single line bug. Make `--data_dir` point to CIFAR10 on your disk. Please load the provided pretrained weights. You don't need to train for this bug. The accuracy is not precise, you can think the 66 as 66 ± 5 , etc.