CS212 Topics in Computing 2
2018, Term 2 Second Half: Lab 3
Optimization, Gradients, Theano

Assessment & Submission

Same as the previous assignment. This assignment is worth 25% of your overall lab mark for this Half Module. The report for this assignment, including your screenshots and answers as instructed below, needs to be uploaded to myPlace before the end of the lab (1pm) on March 21st. You also need to show it to a demonstrator during the lab on March 21st or after the test on March 28th. Simply uploading your report will not give you any marks. Every late day will cost 20% deduction. You can do this lab in the groups of your choice no larger than 3 students. But make sure everyone retains the copy of the code and the screenshots and reports the accomplished lab individually.

Aims

Practice using Python
Practice using derivatives and gradients
Practice using an artificial neural network
Solve a realistic challenge: MNIST handwritten digits recognition

Instructions

Task 9. (10% marks)

(a) Compute dot product between the following two vectors: *[1, 2]* and *[3, 4]*.

(b) Verify the result using numpy. For example, as here for different vectors:

```
v1 = numpy.asarray([5, 6])
v2 = numpy.asarray([7, 8])
print numpy.dot(v1, v2)
```

(c) Compute dot product between the following matrix *W* and vector *x*:

```
     [1 , 2]
W =  [3 , 4]  , x = [1 , 1]
     [5 , 6]
```

(d) Verify the result using numpy. For example, as here for a different matrix and a vector:

```
W = numpy.asarray([[1, 2],[3, 4],[5, 6]])
v = numpy.asarray([1, 1])
print numpy.dot(W, v)
```

Take the screenshot(s) (or a photo) of your results, the screenshot of your output and your code. Add them to your report.

Task 10. (10% marks) Using our lecture 3 slides as guidance, train linear sentiment model using Theano and obtain the output similar to slide 20. Take the screenshot(s) to include the output and your code. Add it to your report.

Task 11. (10% marks)
(a) Using our lecture 3 slides as guidance, convert the linear sentiment model from Task 10 to a neural network model. Feed a single training example shown on slide 37 and obtain the output similar to slide 39. Feed the same training example two times. Are the predictions different on the second time? Which movie rating is now winning?

(b) Instead of the example from slide 37, similarly run two training steps with this review: "The movie is very very bad!" with the score of -2 ("very bad"). Are your outputs now different? Should they be? How?

Take the screenshot(s) of your code, your output and your answers.

Task 12 (20% marks)
By combining your model implemented in Task 11 and the posted template "L3-template.py", write a program that can train sentiment analysis model by taking the training data from the file "movie-reviews-4-3.txt". But this time, use 4 features instead of 2: so you will use the following word counts: "good", "super", "bad", and "terrible". Also, instead of 5 possible outcomes(ratings), it should use only three: "bad" (-1), "neutral" (0) and "good" (1).
Hint: you will need to change the size of the matrix *W*, the one-hot encodings for your 'score_vector' and the feature vector 'features' to include new words ("super" and "terrible"). You will also need to change the data file from "movie-reviews-2.txt" to "movie-reviews-4-3.txt" (both files posted).
Run your program. What is happening with the reported 'Accuracy' ? Does this happen with all inputs or only some? Why do you think that happens?

Take the screenshot(s) to include the output, your code and your answers. Add it to your report.

Task 13. (10% marks, challenge task). Explain why some of the gradients in Task 11 were different between the parts (a) and (b).

Task 14. (40% marks) Recognizing handwritten digits. Due to snow cancellations two weeks ago, this task will be posted and discussed later and be due separately at later time.