TP 7: Logistic and Softmax regression

Thursday 9th May, 2019
joao.candido@hesge.ch
Deadline: Tuesday 21st May, 2019, 23:59

1 Objective

The goals of this TP is to understand and implement the Logistic regression and the Softmax classifier with an $L2$ norm regularizer.

You are going to use the *CIFAR-10* data set. If you are using **windows** the get_dataset.sh script will not work (only mac and linux are available) !

2 Detailed Instructions

You are going to fill a few missing functions in the python script[1] and in jupyter notebook to implement the exercises that we ask. So first of all read and understand the given python script. To run your code you have to run the TP7_logistic_softmax_classifiers.ipynb notebook. You are going to use the Cifar 10 data set.

You have to send a **formal** report and your code.

For this TP the following steps will need to be done for the Logistic and Softmax classifier, all this formulas and derivation have to be in your report. If you don't include in your report your derivations **your code will not be taken into account**. Same if your code it is not based on your derivations.

2.1 Logistic regression

1. The forward step, i.e. the computation of the scores.
2. The cost function of the logistic regression.
    (a) Write down the cost function of the logistic regression on your report.

[1] Part of the given code is based on Stanford's repository

(b) Derive the gradient of the logistic regression cost with respect to the weights (learning parameters).

(c) Write down the cost function of the logistic regression when a L_2 regularizer is added.

(d) Derive the the gradient of the logistic regression cost when a L_2 regularizer is added.

3. Based on your derivations, implement the loss function in the logistic regression class and its derivative with an L_2 regularizer. **Reminder :** All the formulas and derivations (i.e. how do you get a given formula) must be in the report ! Otherwise the code it's not taken into account.

   (a) Fill the *scores* part in loss function inside the logistic_regression.py

   (b) Fill the *loss* part in loss function inside the logistic_regression.py

   (c) Fill the *grads* part in loss function inside the logistic_regression.py

4. Implement Stochastic Gradient Descent, SGD

   (a) Fill the missing part of the train() method inside the classifier.py script

5. Fill the missing part in TP7_logistic_softmax_classifiers.ipynb notebook and train your classifier for different learning rates and regularization strengths.

   (a) Comment **in details** how the different learning rates and regularization strengths influence the performance of the classifier. Which is the effect of very large/small regularizers? How the learning rate change the prediction? Which (and why) is the optimal why to update the weights? etc..

2.2 Softmax classifier

Follow exactly the same thing as before (logistic regression) but with softmax classifier. All the steps, formulas derivations have to be in report to make your code count !

3 Reminders

- The Softmax classifier is the generalization of the binary Logistic Regression classifier to multiple classes. The Softmax classifier gives as output normalized class probabilities

- When we minimize the cost function using Gradient Descent (GD) the weights are updated after seeing all the training instances

- When we minimize the cost function using Stachastic Gradient Descent (SGD) the weights are updated after seeing a mini batch the training instances.
- The L2 regularizer is also called Ridge Regression. It adds ”squared magnitude” of weights as penalty term to the cost function. If the regularizer parameter is zero then you can imagine we get back to the cost function.

General instructions

You have to put your work in *cyberlearn* saved in a zip file using as name this format: TP_7_LASTNAME_Firstname. You can clean the *datasets* folder by running *clean.sh* before upload your work.