SM in AI

Assignment2

Deadline : 3 october 2017, 11:55 pm

Problem 1. Clone `https://github.com/dracarys983/SVM` repository. All the instructions are there.

Problem 2. Come up with a CNN architecture containing a maximum of 8 FC + CONV + POOL layers to classify the dataset.

1. (a) Add the following layers in your architecture and report the accuracies and convergence time in each ease in *q2_a_report.pdf*.
   i. Batch normalization
   ii. Dropout
   (b) Use the following activation functions and report accuracy and convergence time in each case in *q2_a_report.pdf*
   i. ReLu
   ii. tanh
   iii. Sigmoid
   iv. Any other functions you find interesting.
2. Use the best architecture obtained to classify the dataset.
3. Use the CNN as a feature training network and use SVM to classify the same.

NOTE :

- Directory Structure:

```
q2/
├── q2_a_report.pdf
├── q2_b.py
└── q2_c.py
```

- The output of *q2_b.py* and *q2_c.py* should be N lines and each line should contain the class label. Here N denotes the number of images in the *test_batch* file.
- If additional files are found in the q2 directory zero marks will be awarded.

- The dataset is divided into multiple batches and each batch is a python pickled object. Refer *a.py* on how to unpack and use the data.
- The codes will be invoked as *python q2_x.py* $< data\_batches\_folder >< test\_batch\_file >$
- You can use scikit-learn to implement SVM in 'c' part.
- Use keras to implement CNNs
- dataset : `http://10.4.16.28:8082/smai/datasets/Q2/`
- Timelimit for b and c part is 30 min each

Problem 3. Logistic Regression and Regularization. The problem structure is the classic classification problem. Our data set $\mathcal{D}$ is composed of N samples. Each sample is a tuple containing a feature vector and a label. For any sample n the feature vector is a $d+1$ dimensional column vector denoted by $\mathbf{x}_n$ with d real-valued components known as features. Samples are represented in homogeneous form with the first component equal to 1: $x_0 = 1$. Vectors are bold-faced. The associated label is denoted y_n and can take on only two values: $+1$ or -1.

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\}$$
$$\mathbf{x}_n = \begin{bmatrix} 1 & x_1 & ... & x_d \end{bmatrix}^T$$

In logistic regression, the output signal(s) is processed through a non-linear probability(θ):

$$s = \sum_{i=0}^{n} w_i x_i = \mathbf{w}^T \mathbf{x}$$

$$\theta(s) = \frac{e^s}{1 + e^s}$$

The likelihood for a given sample is:

$$P(y_i|\mathbf{x}_i) = \theta(y_i \mathbf{w}^T \mathbf{x}_i)$$

The resultant likelihood for all samples becomes:

$$\prod_{n=1}^{N} \theta(y_n \mathbf{w}^T \mathbf{x}_n)$$

The hypothesis which best fits the sample data and maximizes the likelihood minimizes the model loss:

$$\mathcal{L}_{model} = \frac{1}{N} \sum_{n=1}^{N} \ln \left(\frac{1}{\theta(y_n \mathbf{w}^T \mathbf{x}_n)} \right)$$

To reduce overfitting, a regularization loss is added to the cost function:

$$\mathcal{L}_{reg} = \frac{\lambda}{2} \|w\|^2 = \frac{\lambda}{2} \sum_{j=1}^{m} w_j^2$$

Here λ is the regularization parameter. The L2 regularization loss is the most common form of regularizing the model parameters. Similarly an L1 loss function can be used which is a sum of absolute values of the model parameter weights.

Vary the regularization parameter(λ) and report its effect on the values of w and the performance of the resultant models. Juxtapose the effect of L1 loss(1a) and L2 loss(1b) using the same λ. Obtain activation maps(weights w) for the same as shown in figure 1. Write a report with your findings and conclusions. Use scikit-learn's `LogisticRegression` package for your experiments.

(a) L1 loss weights (b) L2 loss weights

Figure 1: Activation Maps.

The data contains 28×28 grayscale images of letters 'B' and 'E' and `reg.ipynb` jupyter notebook contains starter code for reading and visualizing the data.

Find the dataset and starter code at `https://web.iiit.ac.in/~hemanth.veeranki/smai/datasets/Q3/`

Directory Structure:

```
q3/
└── q3_report.pdf
```

Problem 4. Classify the given dataset with the least squares regression problem.

1. Use following regularizations
   (a) Lasso (L1)
   (b) Ridge (L2)
   (c) Elastic net (Lasso and Ridge combined)
   (d) No Regularization
2. For each regularizations, experiment with the hyperparameters(if any) and report the accuracies in the reports *q4_report.pdf*
3. For each regularization, use the best set of hyper parameters (in terms of accuracy) to classify given dataset

4. The output of each python file should be "N" lines and each line should contain the class label. Here N denotes the size of test_file

5. The codes will be executed as $python q4\_x.py < train\_file >< test\_file >$

6. You can use scikit-learn library to solve the regression problem.

7. Sample dataset link : `https://web.iiit.ac.in/~hemanth.veeranki/smai/datasets/Q4/`

Directory Structure:

```
q4/
├── q4_a.py
├── q4_b.py
├── q4_c.py
├── q4_d.py
└── q4_report.pdf
```