Assignment-1

DeepEigen|| Course: AI-1.0X: Machine Learning | Year: 2020 | Instructor: Sanjeev Sharma

1 Linear Regression

- In the lectures, we have seen the loss function, L_2-Norm Squared (L_2^2), can be minimized using both the Normal Equations solution or using the gradient descent algorithm. In this assignment, we will discuss an important technique, known as data-normalization.

Data Normalization:
Assume that you have a data-set $\mathcal{D}\{x^i, y^i\}; i = 1, ..., m$. Lets assume that we construct a feature representation $\phi(x)$ as discussed in the lectures. We assume that the first term in the feature-vector $\phi(x) \in \mathbb{R}^N$ is a constant 1, which helps in learning the bias term. The other $N-1$ components in $\phi(x)$ will depend on x.

The first step is to compute the mean and standard-deviation of the features. We can construct a matrix $\Phi \in \mathbb{R}^{m\times N}$, where the i^{th} row matrix Φ is $\phi(x^i)^T$. Now for each of the colums of Φ, except for the 1^{st} column, which is a vector of 1, compute the mean and standard-deviation of the column-vector.

$$\begin{aligned}\mu_i &= \mathbf{mean}(\Phi^i)\\ \sigma_i &= \mathbf{std}(\Phi^i)\end{aligned}$$

where Φ^i is the i^{th} column of the matrix Φ. The next step is to subtract the mean and divide by the feature vector.

$$\hat{\Phi}^i = \Phi^i - (\mathbf{1}_m)\mu_i, \qquad \text{for } i = 2, ..., N \tag{1}$$

$$\hat{\Phi}^i \leftarrow \frac{\hat{\Phi}^i}{\mathbf{1}_m \sigma^i}, \qquad \text{for } i = 2, ..., N \tag{2}$$

where step-1 and step-2 must be followed in the sequence. Here $\mathbf{1}_m$ denotes a vector of dimension m with all components of the vector being 1. Furthermore, the division in step-2 is an element-wise division. Essentially we are dividing each of the components of Φ^i with the value σ^i.

Now, the weights should be learned using the feature matrix $\hat{\Phi}$ or the new feature representation $\hat{\phi}$. Now, once the weights are learned, $w^\star$, on the normalized data. Suppose now a new data-point $\hat{x}$ is given. To predict its corresponding target value $\hat{y}$, we should construct $\phi(x)$, then we should use the same mean μ and σ that were computed on the origianl dataset $\mathcal{D}$, to normalize the feature vector as:

$$\begin{aligned}\hat{\phi}(\hat{x})_i &= \phi(\hat{x})_i - \mu_i; \quad i = 2, ..., N\\ \hat{\phi}(\hat{x})_i &\leftarrow \frac{\phi(\hat{x})_i}{\sigma_i}; \quad i = 2, ..., N\end{aligned} \tag{3}$$

Now compute the target $\hat{y} = w^{\star\mathsf{T}}\hat{\phi}(\hat{x})$.

Problem 1 Download the dataset for the 1st problem in the assignment **"data_regress.txt"**. The first column in this file represents the x values, and the second column represents the target y values. This dataset is generated using:

$$\begin{aligned}y &= \frac{x^3}{2} + x^2 + \epsilon\\ \epsilon &\sim \mathcal{N}(0, \sigma^2)\\ \sigma &= 10\end{aligned}$$

Your tasks are:s

- Create an appropriate feature representation, $\phi(x)$ (hint: it should be a 3-degrees polynomial function, plus the bias term 1).
- Try to fit the polynomial on the downloaded dataset, with the gradient descent algorithm on the data provided, without any normalization.
- Try to fit the polynomial on the downloaded dataset, with the gradient descent algorithm on the data provided with normalization.

For the tasks mentioned above, you should run the gradient descent algorithm for a maximum of 1000 epochs only, and share the output with us.

You should share with us:
username is your username with which you registered for DeepEigen (username should not have spaces)

- Python file named "code_1_**username**.py" which runs the gradient descent algorithm without the normalization.
- A text file named "learningRate_1_**username**.txt" which has the learning rate with which the gradient descent algorithm worked without normalization. For most the learning rates you will try, the weights vector w will become either NaN or Inf ($\pm\infty$). Just save the value, do not write any text in this file. For example, if you selected the learning rate as 100, then the file should only have 100 written in it without any space either before or after the value. (hint: after selecting a sufficiently small learning rate, the weights will converge to optimal value).
- Python file named "code_2_**username**.py" which runs the gradient descent algorithm with the normalization.
- A text file named "learningRate_2_**username**.txt" which has the learning rate with which the gradient descent algorithm worked with normalization. Just save the value, do not write any text in this file. For example, if you selected the learning rate as 100, then the file should only have 100 written in it without any space either before or after the value.

2 How to Submit the Assignment

Make a folder named **Assgn_1_username**, and put all the above mentioned files in that folder. Compress that folder and create "Assgn_1_username.zip" and submit this file.