Research Project

May 3rd, 2021

1 Research

1.1 Basic Knowledge

Need to know

- Linear Algebra
- Multivariate Calculus
- Python

1.2 Problem

We would like to solve the problem

$$Ax = b, \tag{1}$$

where $A \in \mathbb{R}^{n \times d}$, $x \in \mathbb{R}^d$, and $b \in \mathbb{R}^n$.

Question: Given A and b. Find x

L_2 Norm: $\|x\|_2$

Function:

$$g(x) = c^\top x,$$

where $c \in \mathbb{R}^d$ is a vector which has the same size as x. $x = [x_1, \ldots, x_d]^\top \in \mathbb{R}^d$.

Compute: $\nabla_x g(x)$.

$$g(x) = c_1 x_1 + c_2 x_2 ... + c_d x_d$$
$$\nabla_x g(x) = c = [c_1, \ldots, c_d]^\top \in \mathbb{R}^d.$$

Function:

$$f(x) = \frac{1}{2}\|Ax - b\|^2$$

Compute: $\nabla_x f(x)$ and apply gradient descent.

Gradient descent is an iterative method. (You can search for the formula in the internet). You can experiment with different starting points and step sizes (or learning rates). For each iteration, we compute and plot $\|\nabla_x f(x)\|^2$ and $f(x)$ to measure the performance.

We have

$$\|x|^2 = x^\top x$$

$A \in \mathbb{R}^{n\times d}$, $x \in \mathbb{R}^d$, and $b \in \mathbb{R}^n$

$$\begin{aligned}
\|Ax-b\|^2 &= (Ax-b)^\top(Ax-b) = (Ax)^\top(Ax) - (Ax)^\top(b) - b^\top(Ax) + b^\top(b) \\
&= x^\top A^\top Ax - x^\top A^\top b - b^\top Ax + b^\top b \\
&= x^\top A^\top Ax - (x^\top A^\top b)^\top - b^\top Ax + b^\top b \\
&= x^\top A^\top Ax - b^\top Ax - b^\top Ax + b^\top b \\
&= x^\top A^\top Ax - 2b^\top Ax + b^\top b
\end{aligned}$$

$$\begin{aligned}
&(a+b)(c+d) = ac + ad + bc + bd \\
&(a+b)^\top = a^\top + b^\top \\
&(Ax)^\top = x^\top A^\top \\
&e^\top = e,\ e \in \mathbb{R}
\end{aligned}$$

Next week:

- Calculate gradient and implement GD.
- Set the stopping criteria: square norm of gradient $<$ some number
- Learn how to write a matrix in numpy ndarray.

Solution:

Using gradient previously derived for $g(x) = c^T x$, we let $c = A^T b$ along with property that $\rho(x) = x^T Bx$ gives $\nabla_x \rho(x) = (B + B^T)x$, we have:

$$\begin{aligned}
\nabla_x(\|Ax-b\|^2) &= -2A^T b + (A^T A + (A^T A)^T)x \\
&= -2A^T b + 2A^T Ax \\
&= 2A^T(Ax-b)
\end{aligned}$$

Hence, the final answer to the original problem $\nabla_x f(x)$ for $f(x) = \frac{1}{2}\|Ax-b\|^2$ is $A^T(Ax-b)$.

2 Linear Regression

Data $\{x_i, y_i\}_{i=1}^n$, $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, $i = 1, \ldots, n$ Find $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ so that $\sum_{i=1}^n (w^T x_i + b - y_i)^2$ is minimized.

1. Formulate this linear regression problem to the least squares problem above.
2. Find two regression datasets (e.g. UCI) and implement linear regression.
3. Compare your result with LR algorithm from sklearn. (Compare the loss and the distance between two solutions)

2.1 Solution

1. The problem above can be rewritten as

$$min_w \sum_{i=1}^{n} (x_i^T w - y_i)^2$$

This is in fact the same as finding the minimum for $f(x)$ for

$$f(x) = \|Ax - b\|^2$$

which we know how to solve by using the $\nabla_x f(x)$.

Next week

- Find out why the output of our algorithm and sklearn are different. Try creating a perfectly fitted dataset to see if sklearn results in correct solution.
- Write the loss function for logistic regression here. Calculate the gradient of that loss function.
- Implement the GD algorithm. Please choose d large (not 1 or 2). If possible, test the performance vs sklearn.

3 Logistic Regression

Loss function:

$$l(\mathbf{w}) = \sum_{i=1}^{n} \log(1 + \exp(-y_i x_i^T \mathbf{w})) \qquad (y_i \in -1, 1)$$

the gradient of the loss function would then be

$$\nabla_w l(w) = \sum_{i=1}^{n} \frac{-y_i x_i \exp(-y_i x_i^T \mathbf{w})}{1 + \exp(-y_i x_i^T \mathbf{w})}$$

Next week: Write a notebook that includes:

- Linear Regression: choose a regression data (not too small, should have ≥ 100 samples, dimension ≥ 5), write the gradient function and run Linear Regression.
- Logistic Regression: similarly write the gradient function and run SGD for phishing data.
- Results: Plot the loss function for each iteration (maybe in log scale). In another figure, plot the squared norm gradient for each iteration. If you have time, visualize your final result in $\mathbb{R}^2$.

Note:

1. Please make sure that there is no bug and we can run your notebook(s) without any interruption.
2. Remember to include the data and every library needed - one way to test is to restart the kernel and run all.

3. If this is too long for a notebook, you can split it into 2. Please send them to our emails one or two day before the meeting so that we can run them.

(Optional): Fix bugs on your regression notebook:

1. For logistic regression, sklearn is not tested correctly. It produced a warning.
2. GD has some problems. Check the shape for linear reg.
3. Sklearn for linear reg also is not tested correctly.

If GD still have some bug and you want to find out more, let us know.

4 Project

Here are the temporary timeline for your project. Please message us your progress for each item. (Feel free to edit it)

- May 27: Decide what topic to do:
  1. Image Classification
  - Goal: Given an image dataset and label (cat/dog/other objects or digits from 0 to 9). Train a neural network to learn that label. Test your model on some other images. (Predict the class of test image).
  - Potential data: MNIST, FashionMnist, Cifar10 or Cifar100 (increasing level of difficulty).

  2. Facial recognition/ Facial detection
  - Goal: Given an image dataset and a vector of features (the coordinate of some points describing location of eyes, nose, mouth,...). Train a neural network to learn these features. Test your model on some other images. (Show the location of predicted eyes, nose, mouth,...)
  - Potential data:
  `https://ibug.doc.ic.ac.uk/resources/fiducial-sfacial-spoint-sdetector-s20052007/`
  Note: to simplify, you should not choose a data with too many feature points. The website above has many datasets and you need to do research to find a suitable one.

- May 30: Show us your 'basic' research:
  1. What dataset(s) will you use
  2. What method will you try to solve the problem (neural network architecture, optimization algorithm, other interesting insights from the tutorials you found - we can learn from you also!!)
  3. What platform will you use (Pytorch, Tensorflow, etc) and what tutorials will you follow.
  Also note questions if you don't understand or there is sth to ask about the tutorial(s).
  After receiving this info, we will give you some feedback to modify your plan if needed.
- From May 31 to June 6: Actually do the project, write notebook and test models etc.
- June 6 or 7: Gather all the results and prepare the slides.
- June 8 or 9: Final presentation (include citations)

5 Progress

5.1 Preliminary Plan

1. I will first start out with the MNIST and Fashion MNIST datasets to familiarize myself with the general syntax and functions of TensorFlow and visualization tools. After that I will move onto the CIFAR-10 (and possibly the CIFAR-100 and other larger image classification datasets if time allows).

2. For the MNIST and Fashion MNIST datasets I will just be using the simple Dense layers (and Dropout layers to prevent overfitting) both because I want to focus on the general structure of a Tensorflow project and also because these datatsets contain images that are centered so the benefits to using computer vision techniques such as convolutional layers will only result in marginal gains. However, when I train my CIFAR-10 model, I will be certain to use CNN layers along with Pooling layers as though these images are still relatively low resolution, the features in them are not nicely centered as for the case of the MNIST datasets. I will also be testing out some techniques such as Data Augmentation to increase the size of my training set. Pretrained Models could also be a good way to improve the results as I won't be so limited by my personal computer's processing power.

3. As mentioned before, I opted for TensorFlow 2.0 as my platform of choice both because of its ease to create industrial models and an abundance of tutorials for specifically the task at hand (its recent optimization for Apple's M1 chipset is also a bonus as I am looking to get a MacBook for college).
   For all the sources and tutorials I have used, please check out this link: **Sources**. I will keep on updating the list as I continue researching.
   Note: I also have some sample notebooks of what I have been doing so far by following the tutorials if you are interested. I have email these to you if you want to take a look (note: So far, the creating the model has been quite intuitive for me, possibly because I have done this sort of stuff for online Machine Learning courses a while back. The only part that I do not fully understand is the result visualization code which I have been mostly been copy and pasting. However, I think I will soon pick up on the syntax of pyplot by just using it and reading some documentations.