The Document

João Morais

October 8, 2019

Contents

3 Artificial Intelligence / Machine Learning

3.1 Supervised Learning

Extrema Conditions and Hessian Matrix

Videos: 87 - Warm up to the second partial derivative test to... 89 - Second partial derivative test intuition

This sub (...) sub section aims to explain why the conditions imposed on the Hessian matrix - the second derivative matrix - correspond to imposing in 2D what we know already.

In practice, in 2D, to find a extrema we need to guarantee that the first derivative is zero, which in N dimensions is correspondent to guaranteeing that the gradient is zero in that point, because the gradient is nothing more than all partial derivatives in that point. Therefore, setting $\nabla f = 0$ means that, in that point, the function is not increasing or decreasing in any of the N directions. So the first derivative makes sense.

However, when we go to the second derivative, the meanings get a bit more complicated.

In 2D, if the second derivative was 0, it was probably (not certainly) a saddle point, if it was > 0 or < 0 it was, respectively, a local minimum or maximum.

The conditions we should impose in 3D is to have a positive (for finding a minimum) or negative (for finding a maximum) definite Hessian Matrix. While second derivatives in order to just one variable is possible to attribute a sense to it, why do the cross derivatives play a role as well? And, why do they mean really?

Well, first the explanation on why it is needed: there are functions that across multiple dimensions still show that it is an extrema but then there's an inflexion along directions that are not along the axis. So, checking the axis is not enough. Why checking the cross partial derivatives makes it enough?

The Second Derivative Test

$$f_{xx}(x_o, y_o) f_{yy}(x_o, y_o) - f_{xy}(x_o, y_o)^2 \gtrless 0$$

If it is greater than 0, we have a maximum or a minimum and have to check the value of f_{xx} or f_{yy} to be sure. If it is less than 0, we have a saddle point. If it equals 0, then we don't know if it is a saddle point, but it is not a min or max therefore, at least for now, we certainly don't care.

Cross or Mixed partial derivatives can be switched? Yes if the function is C^2. (Boring to prove theorem called: Schwarz' Theorem)

Therefore, we just need to compute one of the cross derivatives.

Also this works because the second derivative test is nothing more than the determinant of the Hessian matrix. The determinant is the product of every eigenvalue of that matrix, therefore it can only be positive if they are both positive or both negative, in which cases there is, respectively, a minimum or a maximum.

But why do eigenvalues tell us this? Because they tell us how the eigenvectors are scaled! And the eigenvectors of such matrix will be the greatest and the least curvatures. Therefore, they either have the same signal / are scaled the same amount, or

Some other links that helped with this:

- Differential Geometry
- Criterior for critical points - Maximum, Minimum or Saddle?
- David Butler - Facts about Eigenvalues

3.1.1 Neural Networks - BackPropagation

Two of the most useful websites to check while trying to demonstrate the backpropagation algorithm:

- all backpropagation derivatives
- Error (deltas) derivation for backpropagation in neural networks
- 3Blue1Brown Neural Networks - Specially the last one, on backpropagation.

Why kernels are important? They facilitate a lot the mapping of features to higher dimensions!

Why kernels?

3.2 Unsupervised Learning

3.2.1 Lagrange Multipliers

Perfect Explanation why it works:

Quora - Intuition on Langrange Multipliers

And a video showing exactly how it's done:

Khan Academy - Lagrange Multipliers Example

3.3 Reinforcement Learning and Decision Making

An agent to make the wisest decision in its situation has to have knowledge on what state he is in, how the environment will evolve, how it will be like if he performs a certain action (taking into account stochastic environments) and a utility function to know how much that state contributes to its happiness.

3.3.1 Search Problems

There are some goal states and one initial state. The objective is to find the goal state that is closer (with the shortest path to the root/ with the least search cost). Is given as the solution the steps necessary to perform that path.

To keep the formulation as general as possible, abstractions are required, keeping in mind that they will need a correspondence when applied to a real problem.

6 Python

6.1 Important Concepts

Python is a language with many characteristics, ones more obvious and easy to understand, like in-fix notation, others are more complex, like Dynamic Typing and Automatic Memory Allocation.

Moreover, many programming principles like anonymous functions, map, zip are important and should be addressed in order to achieve flexible and overall good programming skills.

Beforehand, check the Python Notebook and the slides from Rodrigo Ventura in the "Additional Material" Folder. They are very complete and give a proper insight on how everything works.

Tuples are faster than lists.

6.1.1 Lambda and Anonymous functions

Lambda is the keyword used to make an anonymous function. The following 2 pieces of code do exactly the same.

```
    def my_key(x):
    return x[0]

    l.sort(key=my_key)


    OR

    l.sort(key = lambda x: x[1])
```

And a short example is:

```
    -----short example on how cool python is-----
    def make_multiplier(factor):
    return lambda x: factor*x

    f = make_multiplier(2)
    ---------------------------------------------
```

Therefore, lambda is nothing more than defining a function, without giving it a name. It helps keeping the code simple specially when the function is just going to be used once.

6.2

6.3 Some useful tools

To check if a variable points to a certain data type:

```
    isinstance(var, [list, tuple, int])
```

To define functions with optional arguments and call them in the incorrect order:

```

    def draw_point(x, y, color='red', thickness=2):
        print('x =', x, 'y =', y, 'color=', color, 'thickness=', thickness)

    x = 1
    y = 2
    draw_point(x,y,'blue', 5)

    draw_point(x,y,thickness=2, color='blue')
```

6.4 Pandas

Check this link :)

6.5 Jupyter Notebooks

Along with Anaconda comes a full installation of the most recent python version and the Jupyter Notebooks, which are awesome to write python.

Here are some very useful shortcuts to tame that beast:

There are 2 modes of shortcutting:

- The Command Mode (when border of the cell is blue). Press ESC to access this mode.
- The Edit Mode (when border of the cell is green). Press ENTER to access this mode.

The Edit Mode is clearly superior:

- Ctrl + Enter to run cell
- Shift + Enter to run cell and get directly to the next
- Ctrl + D to delete a whole line
- Ctrl + arrows jumps words (usual)
- Ctrl + Backspace deletes whole word (usual)

The Command Mode can be usefull sometimes, especially when adding cells is needed, but remember that the cell has to have a blue border:

- A - insert cell before
- B - insert cell after
- DD - delete current cell
- Z - undo cell deletion

6.6 From Python 2 to Python 3

In Linux, if 2to3 is installed, one may simply run:

```
2to3 -w -n file.py
```

This will write the file translated into python 3 in the same file(-w) and without creating a backup(-n).