***Abstract*—This report contains an analyses of a calculus subject and a linear algebra subject. Both subjects were chosen from a list that was supplied by the Open Universiteit. This report elaborates on the subjects to enhance the knowledge gained in the course "Wiskunde voor AI". The calculus method is "Hessian Matrices" and the lineair algebra subject is "".**

I. Hessian Matrices

As explained in section GradientDescent1

A. Introduction

The basis of machine learning algorithm that it tries to predict the right output using a certain input. During the training process difference between the predicted value and the actual value must be minimized. A cost function is used to quantize the difference. Therefore the value must be minimized. To find the minimum value, the machine learning algorithm iterates until it has found the minimum value. The methods for this iteration are numerous, the most well known algorithm is gradient descent (sections I-B and I-C), The method that is discussed in this report is the use of Hessian Matrices (section I-H). Both make use of the Netwon method (sections I-D and I-F).

B. Gradient descent with one variable

Assume that we have a continuous function f defined on R (fig 1):

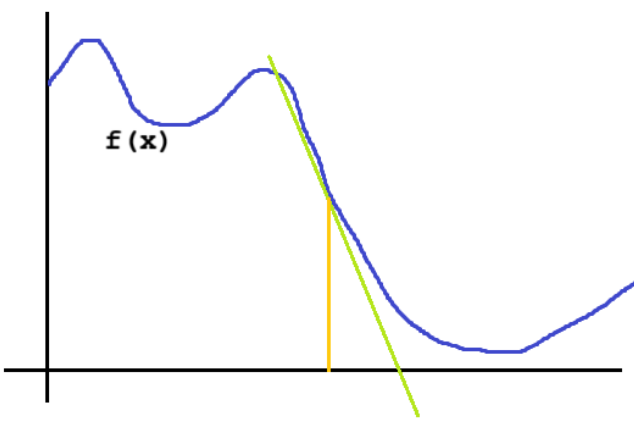


Fig. 1. A continuous function f defined on R

Assume that f is also differentiable with derivative $f'(x)$.
We also have a starting point x_0.
Then we get x_1 by subtracting $f'(x_0) \cdot \alpha$ from x, where α is called the learning rate, which we can choose before doing this procedure.
Usual values for α are 0.01 or 0.05.
We iterate this, so that we get an array which is recursively defined as:

$$x_{k+1} = x_k - f'(x_k) \cdot \alpha$$

This array will converge to the minimum of f. The pitchfalls here are, that the procedure may end in a local minimum, while f has a stronger minimum elsewhere.
Or with a less than optimal choice for the learning rate, the array could even diverge.

C. Gradient descent with two or more variables

With a function $f(x, y)$ of more variables, we can determine the gradient:

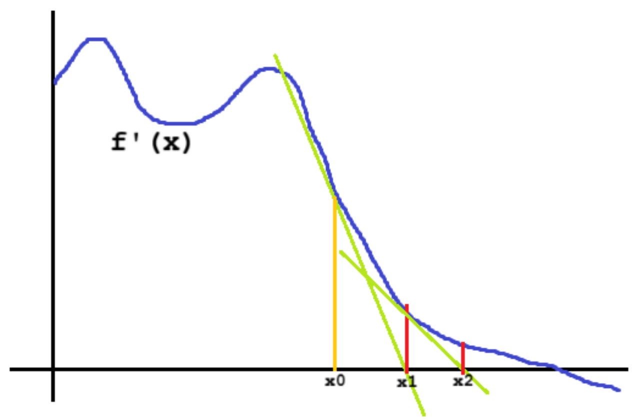
$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

The method is the same, but where we took the derivative for one variable, we will now take the gradient, and the recursive definition of our array (x_k, y_k) becomes:

$$(x_{k+1}, y_{k+1}) = (x_k, y_k) - \nabla f(x, y) \cdot \alpha$$

D. Newton's method with one variable

Newton's method finds the zeroes of a function f. Because we're interested in finding a minimum of f, Newton's method will help us find the zero of it's derivative f'.



Geometrically, when you have the graph of f, and you have a starting point x_0, we start by drawing the tangent line.
Then we see where this tangent line intersects with the x-axis. That will be x_1.
By iterating this procedure we get an array $(x_k)_{k=0,1,...}$.

From the geometrical aspect of the procedure, we can give a formula between x_{k+1} and x_k:

$$x_{k+1} = x_k - (f'(x_k)/f''(x_k))$$

(that is for finding the zero of f') The idea is that the array (x_k) converges to the value x where $f'(x) = 0$.

In order to know if $f'(x)$ points to a minimum of f, we need to look at the second derivative $f''(x)$:
$f''(x) > 0 \Rightarrow$ f has a minimum at x
$f''(x) < 0 \Rightarrow$ f has a maximum at x
$f''(x) = 0 \Rightarrow$ inconclusive, perhaps an inflection point

E. Hessian matrix

F. Newton's method with two or more variables

Say we have function $f(x, y)$ of 2 variables.

Then here we have it's Hessian matrix:

$$H_f(x, y) = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix}.$$

Now at a given point (x,y) we'll calculate it's eigenvalues $\lambda_1, \lambda_2, \ldots$
(A 2 by 2 matrix would have at most two eigenvalues)

If the gradient has value (0,0) at point (x,y) then:

- If all the eigenvalues of H_f at (x,y) are positive, it's a minimum
- If all the eigenvalues of H_f at (x,y) are negative, it's a maximum
- In other cases, it's inconclusive

In Newton's method generalized to more than one variables, the formula for the next point is:

$$(x_{k+1}, y_{k+1}) = (x_k, y_k) - (H_f^{-1}(x_k, y_k) \cdot \nabla f(x_k, y_k))$$

G. Example of a function of two variables

We will look at this function (fig 2):

$$f(x, y) = 85 - \frac{1}{90} x^2 (x - 6) y^2 (y - 6)$$

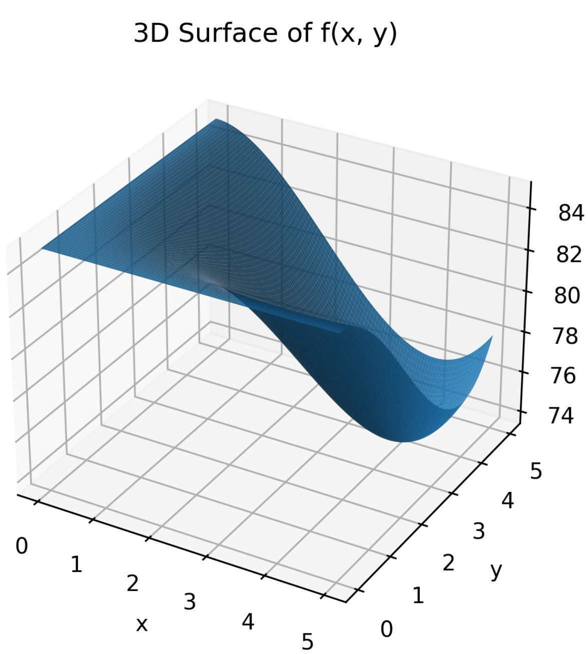


Fig. 2. 3D surface of $f(x, y)$

Visually we see a possible minimum near point $(x, y) = (4, 4)$. We'll take $(x_0, y_0) = (1, 1)$ and $\alpha = 0.01$ and from there, carry out gradient descent as actual example.

For gradient descent, we know that we need the formula for the gradient:

$$\nabla f(x, y) = \left(-\frac{1}{90}(3x^2 - 12x) y^2 (y - 6), -\frac{1}{90} x^2 (x - 6)(3y^2 - 12y) \right)$$

When we fill in x=1 and y=1 we calculate a gradient of (-0.5, -0.5).
We subtract this, multiplied by the learning rate, from (0, 0) and go to the next iteration.
Here we continued with a Python script that produced the following output:

```
0. x 1 y 1 --> (-0.500000000000000,-0.500000000000000)
1. x 1.00500000000000 y 1.00500000000000 --> (-0.50618497489593
2. x 1.01006184974896 y 1.01006184974896 --> (-0.51248365251982
3. x 1.01518668627416 y 1.01518668627416 --> (-0.51889866970464
...
246. x 3.98976321048153 y 3.98976321048153 --> (-0.043564336099
247. x 3.99019885384252 y 3.99019885384252 --> (-0.041715006830
248. x 3.99061600391082 y 3.99061600391082 --> (-0.039943794808
249. x 3.99101544185891 y 3.99101544185891 --> (-0.038247432931
```

We see (x,y) approach (4,4) and we see the gradient approach (0,0). It takes 249 iterations. We also tried learning rate 0.05. Then we saw the same behaviour, but only 56 iterations. A tolerance of 1e-4 was used for comparing floats.

Now we will try to find the minimum using Newton's method. For this we need the second order derivatives:

$$\frac{\partial^2 f}{\partial x^2} = -\frac{1}{90}(6x - 12) y^2 (y - 6)$$
$$\frac{\partial^2 f}{\partial x \partial y} = -\frac{1}{90}(3x^2 - 12x)(3y^2 - 12y)$$
$$\frac{\partial^2 f}{\partial y \partial x} = -\frac{1}{90}(3x^2 - 12x)(3y^2 - 12y)$$
$$\frac{\partial^2 f}{\partial y^2} = -\frac{1}{90} x^2 (x - 6)(6y - 12)$$

We implemented the steps above in a python script and saw the following output:

```
(1.0000,1.0000) - (0.4054,0.4054) = (0.5946,0.5946)
(0.5946,0.5946) - (0.2190,0.2190) = (0.3756,0.3756)
(0.3756,0.3756) - (0.1327,0.1327) = (0.2429,0.2429)
(0.2429,0.2429) - (0.0839,0.0839) = (0.1589,0.1589)
(0.1589,0.1589) - (0.0542,0.0542) = (0.1047,0.1047)
(0.1047,0.1047) - (0.0354,0.0354) = (0.0693,0.0693)
(0.0693,0.0693) - (0.0233,0.0233) = (0.0460,0.0460)
(0.0460,0.0460) - (0.0154,0.0154) = (0.0305,0.0305)
(0.0305,0.0305) - (0.0102,0.0102) = (0.0203,0.0203)
(0.0203,0.0203) - (0.0068,0.0068) = (0.0135,0.0135)
(0.0135,0.0135) - (0.0045,0.0045) = (0.0090,0.0090)
(0.0090,0.0090) - (0.0030,0.0030) = (0.0060,0.0060)
(0.0060,0.0060) - (0.0020,0.0020) = (0.0040,0.0040)
(0.0040,0.0040) - (0.0013,0.0013) = (0.0027,0.0027)
(0.0027,0.0027) - (0.0009,0.0009) = (0.0018,0.0018)
(0.0018,0.0018) - (0.0006,0.0006) = (0.0012,0.0012)
(0.0012,0.0012) - (0.0004,0.0004) = (0.0008,0.0008)
(0.0008,0.0008) - (0.0003,0.0003) = (0.0005,0.0005)
(0.0005,0.0005) - (0.0002,0.0002) = (0.0004,0.0004)
(0.0004,0.0004) - (0.0001,0.0001) = (0.0002,0.0002)
(0.0002,0.0002) - (0.0001,0.0001) = (0.0002,0.0002)
```

```
Hessian from last iteration: [[-4.37584933e-08 -8.75203986e-08]
 [-8.75203986e-08 -4.37584933e-08]]
Eigenvalues of Hessian: [-1.31278892e-07  4.37619053e-08]
```

This converges to point (0,0). The eigenvalues of the Hessian are almost zero. So Newton's method brings us from (1,1) to a stationary point (0,0) which is not a minimum. This happened from many other points. When we choose a starting point real close to (4,4) only then it will converge to our expected minimum:

```
(3.2000,3.2000) - (-1.4933,-1.4933) = (4.6933,4.6933)
(4.6933,4.6933) - (0.7598,0.7598) = (3.9336,3.9336)
(3.9336,3.9336) - (-0.0677,-0.0677) = (4.0013,4.0013)
(4.0013,4.0013) - (0.0013,0.0013) = (4.0000,4.0000)
(4.0000,4.0000) - (0.0000,0.0000) = (4.0000,4.0000)

Hessian from last iteration: [[ 4.26666750e+00 -2.45347276e-13]
 [-2.45347276e-13  4.26666750e+00]]
Eigenvalues of Hessian: [4.2666675 4.2666675]
```

The point reached here is indeed (4,4). The eigenvalues of the Hessian are positive, and indeed this is in line with that we already know, that (4,4) is a minimum.
From (3.2,3.2) it took 5 iterations. When we try Gradient Descent from this point (3.2,3.2) we will see the number of iterations for that:

```
Gradient Descent from (3.2,3.2) with learning rate 0.01 takes 109 iterations to reach (4,4).
Gradient Descent from (3.2,3.2) with learning rate 0.05 takes 27 iterations to reach (4,4).
```

H. Hessian Matrices