Mathematical Concepts (2/2)

Functions, Minimization, Gradient

Fundamentals of Artificial Intelligence

Instructor: Chenhui Chu

Email: chu@i.kyoto-u.ac.jp

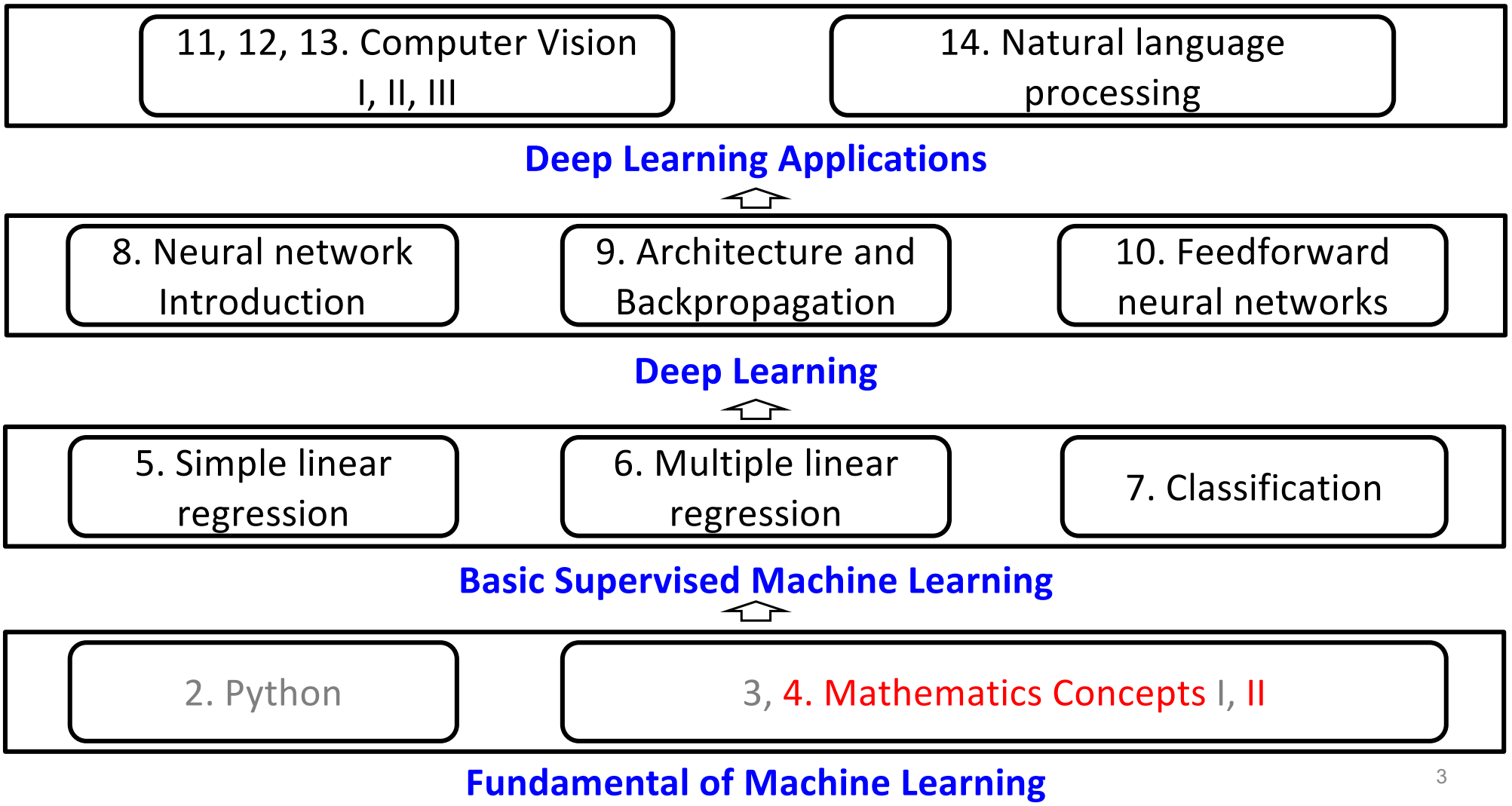
Teaching Assistant: Youyuan Lin

E-mail: youyuan@nlp.ist.i.kyoto-u.ac.jp

Schedule

- 1. Overview of AI and this Course (4/14)
- 2. Introduction to Python (4/21)
- 3, 4. Mathematics Concepts I, II (4/28, 5/12)
- 5, 6. Regression I, II (5/19, 5/26)
- 7. Classification (6/2)
- 8. Introduction to Neural Networks (6/9)
- 9. Neural Networks Architecture and Backpropagation (6/16)
- 10. Fully Connected Layers (6/23)
- 11, 12, 13. Computer Vision I, II, III (6/30, 7/7, 7/14)
- 14. Natural Language Processing (7/17)

Overview of This Course

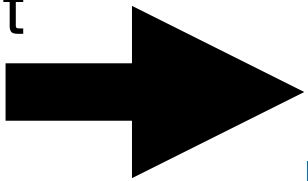


Gradient Descent Algorithm (1/4)

- This suggests some procedure for finding a minimum:
  - Start at any x (e.g., x = 0)
  - Compute f’(x)
    - If f’(x) > 0: Decrease x a bit
    - If f’(x) < 0: Increase x a bit
  - Repeat

Gradient Descent Algorithm (2/4)

- This suggests some procedure for finding a minimum:
  - Start at any x (eg. x = 0)
  - Compute f'(x)
    - If f'(x) > 0: Decrease x a bit
    - If f'(x) < 0: Increase x a bit
  - Repeat



In practice, we do this:

$$x := x - lr \cdot f'(x)$$

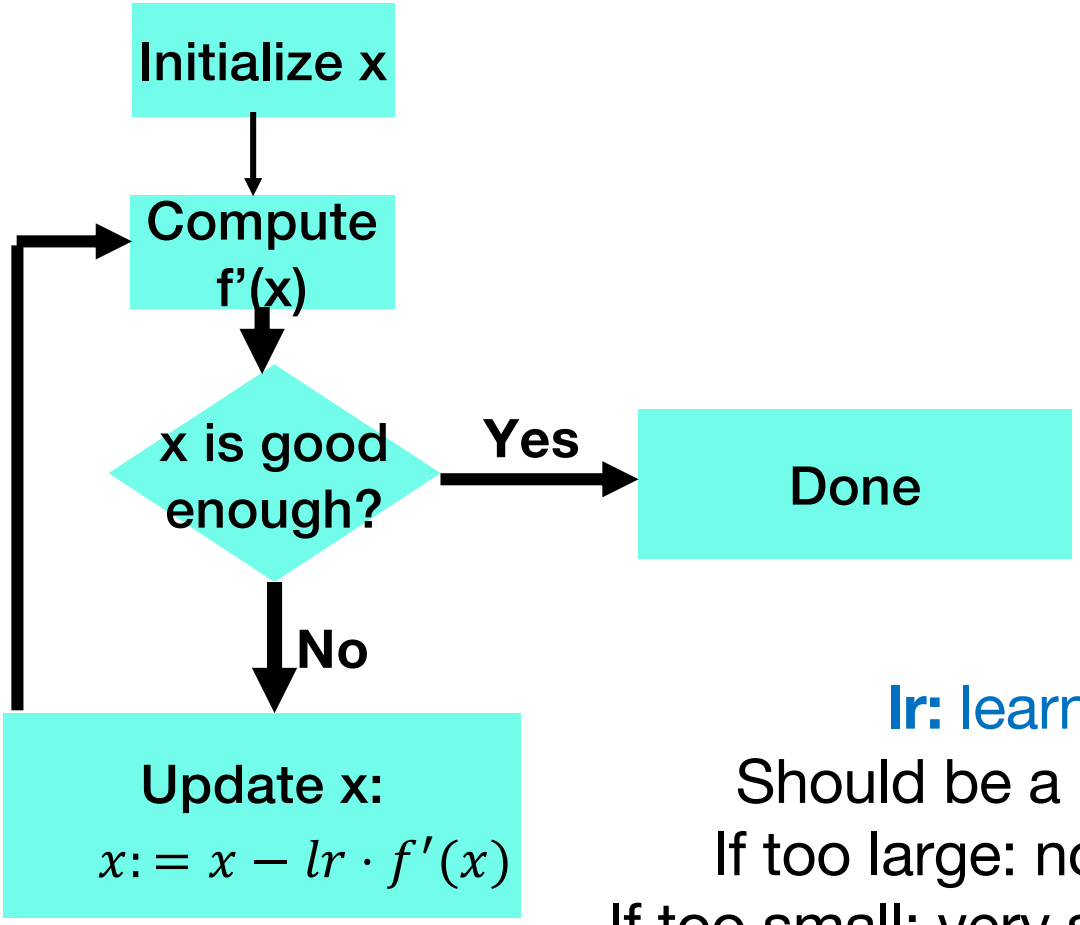
lr: learning rate

Should be a positive value

If too large: no convergence

If too small: very slow convergence

Gradient Descent Algorithm (3/4)



$$f(x) = x^2 - x$$
$$f'(x) = 2x - 1$$
$$\underset{x}{\operatorname{argmin}} f(x) = 0.5$$

lr = 0.2

x = 0
f'(x) = -1

x=0.2
f'(x) = -0.6

x=0.32
f'(x) = -0.36

x = 0.392
…
…
x=0.493
f'(x) = -0.014

STOP?

lr: learning rate
Should be a positive value
If too large: no convergence
If too small: very slow convergence

Gradient Descent Algorithm (4/4)

- Gradient descent works well **even** when we have functions of millions of variable
  - This is why it is so useful for Machine Learning and Neural Networks
  - Other methods will not be practical in such settings
- Convergence will depend on the choice of a good **learning rate**
  - In experiments, a good deal of time is often spent finding an optimal learning rate
  - **Too large** learning rate: **no** convergence (ie. the system learn nothing)
  - **Too small** learning rate: **slow** convergence (ie. the system takes a long time to learn)

Minimizing a Function of Several Variables

Functions of Several Variables

- A function of several variables is just that: a function which has several variables

$$f: \mathbb{R}^3 \to \mathbb{R}$$

$$f(x, y, z) = (x - y)^2 + z^2 - z$$

$$f(0,0,0) = 0$$
$$f(1,2,3) = 7$$
$$f(-1,2,2) = 11$$
$$f(0,1,1) = ?$$
$$f(2,2,0) = ?$$

- Like before, we want to find its minimum:

$$\underset{x,y,z}{\operatorname{argmin}} f(x, y, z) = (0,0,0.5)$$

Parameterized Functions (1/5)

- By fixing one of the variable, we can obtain a function with one less variable

Function of 3 variables: $f(x, y, z) = (x - y)^2 + z^2 - z$

Fixing z: $z = 2$

$f(x, y, 2) = (x - y)^2 + 4 - 2$

Function of 2 variables: $f(x, y) = (x - y)^2 + 2$

Parameterized Functions (2/5)

- By fixing one of the variable, we can obtain a function with one less variable

Function of 3 variables: $f(x, y, z) = (x - y)^2 + z^2 - z$

Fixing y: $y = 2$

$f(x, 2, z) = (x - 2)^2 + z^2 - z$

Function of 2 variables: $f(x, z) = (x - 2)^2 + z^2 - z$

Parameterized Functions (3/5)

- By fixing one of the variable, we can obtain a function with one less variable

Function of 3 variables: $f(x, y, z) = (x - y)^2 + z^2 - z$

Fixing y and z: $y = 2 \quad z = 3$

$$f(x, 2, 3) = (x - 2)^2 + 9 - 3$$

Function of 1 variable: $f(x) = (x - 2)^2 + 6$

Parameterized Functions (4/5)

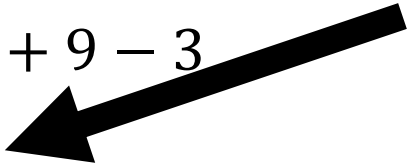
- Therefore, in this case, variables **y** and **z** can be used to describe a "family" of functions. We say they parameterize the

Function of 3 variables: $f(x, y, z) = (x - y)^2 + z^2 - z$

Fixing y and z: $y = 2 \qquad z = 3$

$$f(x, 2, 3) = (x - 2)^2 + 9 - 3$$

For each value of y and z, we have one function of one variable



Function of 1 variable: $f(x) = (x - 2)^2 + 6$

Parameterized Functions (5/5)

- In such a case, we will say that f is a function *parameterized* by **y** and **z.**
- And we note the parameters separately, as subscripts

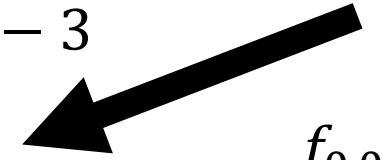
Function of 3 variables: $f_{y,z}(x) = (x-y)^2 + z^2 - z$

Fixing y and z: $y = 2 \qquad z = 3$

$f_{2,3}(x) = (x-2)^2 + 9 - 3$

Function of 1 variable: $f_{2,3}(x) = (x-2)^2 + 6$

For each value of y and z, we have one function of one variable



$f_{0,0}(x) = x^2$

$f_{0,2}(x) = x^2 + 2$

Partial Derivatives (1/4)

- What is the equivalent of our “high school” derivatives when we have several variables?
- One part of the answer is ***partial derivatives***
- Partial derivatives are computed by choosing one variable and fixing the others
- In other words, we see the function of several variables as a parameterized function of one variable

Partial Derivatives (2/4)

- What is the equivalent of our “high school” derivatives when we have several variables?
- One part of the answer is ***partial derivatives***
- Partial derivatives are computed by choosing one variable and fixing the others
- In other words, we see the function of several variables as a parameterized function of one variable $f(x, y, z) = (x - y)^2 + z^2 - z$
- Indeed, if we choose y, and fix x and z, we can see f(x, y, z) as a function of one variable and compute its derivative

$$\frac{\partial f}{\partial x} = 2(x - y) \qquad \frac{\partial f}{\partial y} = 2(y - x) \qquad \frac{\partial f}{\partial z} = 2z - 1$$

Partial Derivatives (3/4)

$$f(x, y, z) = (x - y)^2 + z^2 - z$$

Fix other variables

$$f_{y,z}(x) = (x - y)^2 + z^2 - z \quad f_{x,z}(y) = (x - y)^2 + z^2 - z \quad f_{x,y}(z) = (x - y)^2 + z^2 - z$$

Compute derivative

$$f'_{y,z}(x) = 2(x - y) \qquad f'_{x,z}(y) = 2(y - x) \qquad f'_{x,y}(z) = 2z - 1$$

Partial Derivatives (4/4)

$$f(x, y, z) = (x - y)^2 + z^2 - z$$

Fix other variables

$$f_{y,z}(x) = (x - y)^2 + z^2 - z \quad f_{x,z}(y) = (x - y)^2 + z^2 - z \quad f_{x,y}(z) = (x - y)^2 + z^2 - z$$

Compute derivative

$$f'_{y,z}(x) = 2(x - y) \qquad f'_{x,z}(y) = 2(y - x) \qquad f'_{x,y}(z) = 2z - 1$$

In practice, we use this notation for partial derivatives:

$$\frac{\partial f}{\partial x} = 2(x - y) \qquad \frac{\partial f}{\partial y} = 2(y - x) \qquad \frac{\partial f}{\partial z} = 2z - 1$$

Compute the Partial Derivatives

$$f(x, y, z) = (x - y)^2 + z^2 - z$$

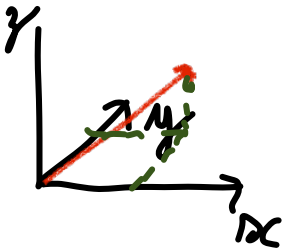
$$\frac{\partial f}{\partial x} =$$

$$\frac{\partial f}{\partial y} =$$

$$\frac{\partial f}{\partial z} =$$

Vectors (1/3)

- What are vectors?
- You probably have used vectors in physics classes to represent force and speed
  - 3-dimensional vectors: [2.3, 4.5, -1]



- In machine learning, we also use them a lot
- Except that they can have more than 3 dimensions
  - 5-dimensional vector: [-1, 3, 4.1, 5.2, 4]
  - We often note the set of all n-dimensional vectors $\mathbb{R}^n$

$$[1.2, 1.4, 1, -1, -1] \in \mathbb{R}^5$$

Vectors (2/3)

- For now, we only need to know the following about vectors:
  - A n-dimensional vector is a list of n numbers
  - We can add 2 vectors (if they have the same dimension)

$$[2.1, 3.4, 1.1, 3.2] + [-1, 2.1, 3.1, -2] = [1.1, 5.5, 4.2, 1.2]$$

$$[2.1,3.4] + [-1,2.1,3.1, -2] =$$ ✗

  - We can multiply a vector by a number

$$0.5 \times [2, 3, -1, -2] = [1, 1.5, -0.5, -1]$$

Vectors (3/3)

- We will usually denote a vector by a letter with an arrow on it: $\vec{x}$
- We denote the i^{th} component of $\vec{x}$ by x_i
- If $\vec{x} = [1, 2.2, -1, 4]$
  - Then we have $x_0=1$, $x_1=2.2$, $x_2=-1$, $x_3=4$

Vectors and Numpy

- In Python, Numpy arrays are a convenient way to represent vectors

$$\vec{x} = [1, 5, -2, 0.5]$$

  - x = np.array([1, 5, -2, 0.5])
  - x[0] == x_0 == 1
  - x[1] == x_1 == 5
  - Vector operations: x+0.5*y

```
In [981]: x = np.array([1, 5, -2, 0.5])
          y = np.array([2,2,10,10])
          z = np.array([3, -3, 0])
          print(x)
          print(y)
          print(z)

          [ 1.   5.  -2.   0.5]
          [ 2  2 10 10]
          [ 3 -3  0]

In [982]: print(len(x), len(y), len(z))

          4 4 3

In [983]: print(x[1], y[2], z[0], y[0])

          5.0 10 3 2

In [984]: print(x+y)

          [ 3.   7.   8.  10.5]

In [985]: print(x+0.5*y)

          [2.  6.  3.  5.5]

In [986]: print(y+z)

          ---------------------------------
          ValueError
```

Vectors and Multivariate Functions

- For now, we have represented the variables of a multivariate function with the letters x, y, z as in: $f(x, y, z) = (x - y)^2 + z^2 - z$
- In practice, we can have <u>any number of variables</u>. So it is more convenient to use: x_0 (instead of x) , x_1 (instead of y), x_2 (instead of z), x_3 .. x_n (if we need more than 3 variables) $f(x_0, x_1, x_2) = (x_0 - x_1)^2 + x_2^2 - x_2$
- We can also **<u>use a vectorial notation</u>** to represent all of the variables as one vector variable: $\vec{x} = [x_0, x_1, x_2]$ $f(\vec{x}) = (x_0 - x_1)^2 + x_2^2 - x_2$
- So, keep in mind that the <u>3 following expressions actually refer to the same function</u>:

$$f(x, y, z) = (x - y)^2 + z^2 - z$$
$$f(x_0, x_1, x_2) = (x_0 - x_1)^2 + x_2^2 - x_2$$
$$f(\vec{x}) = (x_0 - x_1)^2 + x_2^2 - x_2$$

Gradient (1/2)

- The partial derivatives become the component of a vector we call the ***gradient***

$$grad \cdot f(x,y,z) = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}]$$

- For example:

$$f(x,y,z) = (x-y)^2 + z^2 - z$$

$$grad \cdot f(x,y,z) = [2(x-y), 2(y-x), 2z-1]$$

Gradient (2/2)

$$grad \cdot f(x, y, z) = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}]$$

- In this case, the function has 3 variables. Therefore the gradient is a vector of size 3
- If the gradient has n variables, it is a vector of size n
- More precisely, the gradient of f is itself a function that return a vector

$$f: \mathbb{R}^n \to \mathbb{R}$$

$$f(x_1, x_2, \ldots, x_n)$$

$$grad \cdot f: \mathbb{R}^n \to \mathbb{R}^n$$

$$grad \cdot f(x_1, x_2, \ldots, x_n) = [g_1, \ldots, g_n]$$

Interpreting the Gradient

- At a given point, the gradient is the direction for which the value of the function increase fastest

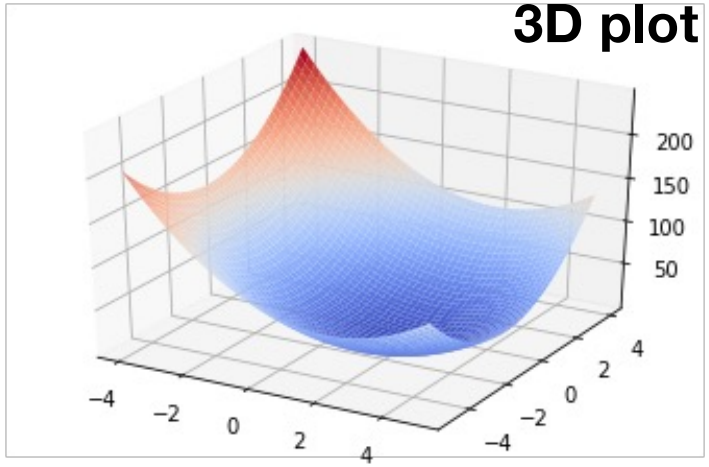
- Therefore, in general, it points in the direction **opposite to the minimum**

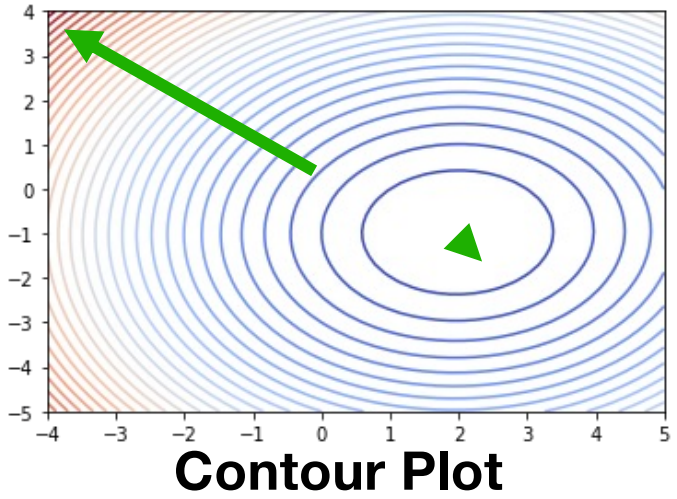
$$f(x,y) = 4(x-2)^2 + 4(y+1)^2 - 0.1xy$$

$$grad \cdot f(x,y) = [8(x-2) - 0.1y, 8(y+1) - 0.1x]$$

$$grad \cdot f(0,0) = [-16, 16]$$

$$grad \cdot f(2,-1) = [0.1, -0.2]$$





Gradient Descent

- Because we know that the gradient point in a direction opposite to the minimum, we can use the same idea as in the case of one variable

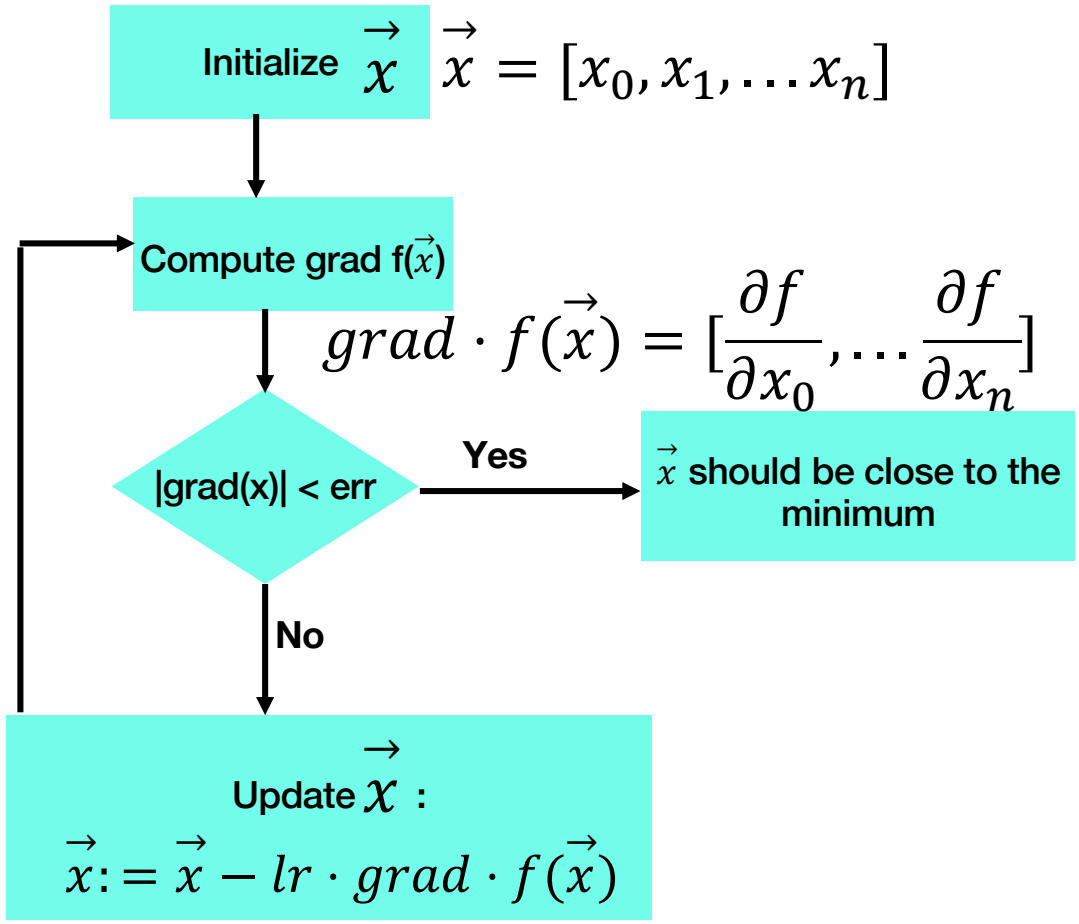
One variable:

$$x := x - lr \cdot f'(x)$$

Multiple variables:

$$\vec{x} := \vec{x} - lr \cdot grad \cdot f(\vec{x})$$

Gradient Descent Algorithm



$$f(\vec{x}) = (x_0 - x_1)^2 + x_2^2 - x_2$$

$$grad \cdot f(\vec{x}) = [2(x_0 - x_1), 2(x_1 - x_0), 2x_2 - 1]$$

$$lr = 0.2$$

$$\vec{x} = [0,1,0]$$
$$grad \cdot f(\vec{x}) = [-2, 2, -1]$$

$$\vec{x} = [0.4,0.6,0.2]$$
$$grad \cdot f(\vec{x}) = [-0.4, 0.4, -0.6]$$

$$\vec{x} = [0.41,0.43,0.51]$$
$$grad \cdot f(\vec{x}) = [-0.04, 0.04, 0.01]$$

What is the Equivalent of Second Derivative for Multivariate Functions?

- It is the Hessian Matrix:

$$\begin{vmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial x \partial z} & \frac{\partial^2 f}{\partial y \partial z} & \frac{\partial^2 f}{\partial z^2} \end{vmatrix}$$

- But thankfully, we will not need to use it
- But for your information, this would be the equivalent of the "High School" minimization when we have several variables:

To minimize f(x, y, z):
1. **Compute gradient of f(x, y , z)**
2. **Compute hessian of f(x)**
3. **Find x, y, z such that grad f(x,y,z) = 0**
4. **If hessian of f(x,y,z) is definite positive then (x,y,z) is a local minimum of f**

Gradient Descent Algorithm

- You can see that, in the case of the gradient descent, the algorithm is the same for univariate functions and multivariate functions
- It is a simple algorithm, but it scales very well
- There exists many variations of it:
  - Gradient Descent with Momentum
  - Stochastic Gradient Descent
  - Adagrad, Adadelta, Adam, …

Gradient Descent with Momentum

- Compute a “gradient with momentum” at each iteration:

$$gm_t = 0.6grad \cdot f(\vec{x}) + 0.4gm_{t-1}$$

Update $\vec{x}$:

$$\vec{x} := \vec{x} - lr \cdot gm_t$$

Stochastic Gradient Descent (1/2)

- What happens if the gradient is noisy?
- That is, we can only compute a value that is equal to the true gradient “on average”?
  - A bit like if you are drunk and trying to get home

Stochastic Gradient Descent (2/2)

- What happens if the gradient is noisy?
- That is, we can only compute a value that is equal to the true gradient “on average”?
  - A bit like if you are drunk and trying to get home
- It turns out it works.
  - But you have to decrease your learning rate over time to stabilize
  - Convergence will be slower

$$lr = \frac{lr_0}{\sqrt{(t+1)}}$$

- Very interesting because a noisy gradient can be million times faster to compute than a “true” gradient

Optimization Libraries

- You can also minimize a function by using a specialized library
- It gives you access to more sophisticated minimization algorithms
- However, these more sophisticated algorithms do not scale as well as Gradient Descent
  - Which is one reason for Gradient Descent and its variants are still the main tool for large scale Machine Learning (In particular, Deep Learning)

```
In [103]: scipy.optimize.minimize(f_numpy, np.array([0,0]), jac=grad_f_numpy, method="L-BFGS-B")
```

```
Out[103]:       fun: 0.1969057665260197
          hess_inv: <2x2 LbfgsInvHessProduct with dtype=float64>
               jac: array([-3.88578059e-16,  2.77555756e-17])
           message: b'CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL'
              nfev: 5
               nit: 4
            status: 0
           success: True
                 x: array([ 1.9878106 , -0.97515237])
```

Google Colab Notebook

- Let us check gradient descent in practice with Google Colab notebook

https://shorturl.at/dTCs9

Report

- Submit Exercise 1 and 2 **in pdf** via PandA
  - Submission due: **next lecture**
  - Name the pdf file as **student id_name**.

Exercise 1

- Compute the partial derivatives of:

$$f(x, y, z) = xyz - z^2 - y^2$$

$$f(x, y, z) = e^{x+y} - log(z)$$

Exercise 2

$$\vec{x} = [1.5, -2.0, 5]$$
$$\vec{y} = [2, 2, 10, 10]$$
$$\vec{z} = [3, -3, 0]$$

- Dimensions of $\vec{x}, \vec{y}, \vec{z}$?
- Values of x_1, y_2, z_0, y_0?
- Compute: $\vec{x} + \vec{y}$ $\vec{x} + 0.5 \times \vec{y}$ $\vec{y} + \vec{z}$