The Gradient and Gradient Descent

Recall the situation, the deep/feed forward neural net (also called MLP = multi layered perceptron) has the form

$$F(x,\eta) = F_L \circ \cdots \circ F_1(x) \quad \text{with each}$$

$$F_i(x) = \sigma(A_L x + \vec{b}_L)$$

- The parameters are $\eta = A_1, \ldots, A_L, b_1, \ldots, b_L$
- The training data is $x_1, \ldots, x_N$ with correct output $y_1, y_2, \ldots, y_N$
- The loss or error or objective function is (simplest version)

$$\Phi(\eta) = \frac{1}{N} \sum_{i=1}^{N} \| F(x_i, \eta) - y_i \|^2$$

- Learning consists of adjusting the parameters η so that the error diminishes.

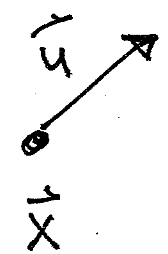
- To accomplish this we need to recall the tools from multi-variable Calculus, specifically, the <u>gradient</u>.

Review of the Gradient

- Now we let $x_1, .., x_n$ revert to the usual Calculus roles as componets of $\vec{x} = (x_1, ., x_n) \in \mathbb{R}^n$
- Let Φ be a real valued function of $\vec{x}$

  so $\Phi(\vec{x}) = \Phi(x_1, .., x_n)$ is a scalar
- We want to understand how Φ is changing in various directions in $\mathbb{R}^n$
- As a simple example let $\Phi(x_1, x_2) = 9x_1^2 + x_2^2$

- We think of Φ as the temperature at each point in the plane for concreteness
- Starting at the point $\vec{x} = (x_1, x_2)$ we walk in a direction given by the unit vector $\vec{u} = (u_1, u_2)$ [I am writing things as row vectors like is done in Calculus]

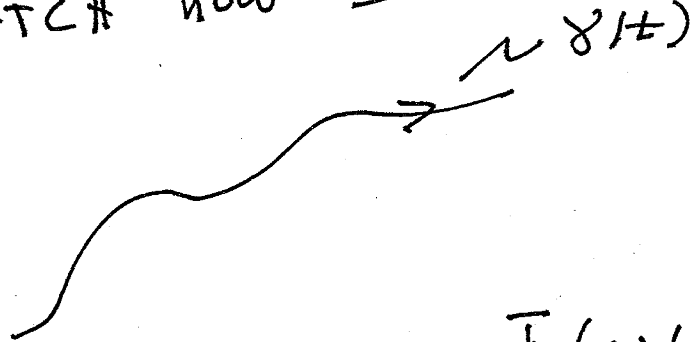


- How does Φ change? We can write a limit

$$\lim_{t \to 0} \frac{\Phi(\vec{x} + t\vec{u}) - \Phi(\vec{x})}{t} = D_{\vec{u}}\Phi(x)$$

- This is called the directional derivative of Φ in the direction $\vec{u}$

- How do we compute this?
- Let $\gamma(t)$ by a path with unit speed
  So $\left|\frac{d\gamma}{dt}\right| = 1$ i.e. the velocity
- We WATCH how Φ changes along the path



and call this $g(t) = \Phi(\gamma(t))$

- So if $\gamma(0) = \vec{x}$ we seek $g'(0) =$ $D_{\vec{u}}\Phi(x)$
  where $\vec{u} = \frac{d\gamma}{dt}(0)$
- We compute this from the chain rule

$g(t) = \Phi(\gamma(t))$

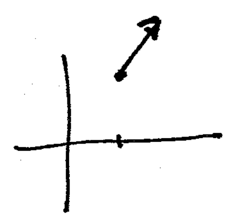
so $\frac{d\,g(t)}{dt} = \nabla\Phi(\gamma(t)) \cdot \frac{d\gamma}{dt}(t)$ with $\nabla\Phi = \left[\frac{\partial\Phi}{\partial x_1}, \ldots, \frac{\partial\Phi}{\partial x_n}\right]$

evaluating at $t=0$, $\gamma(0) = x$, $\frac{d\gamma(0)}{dt} = \vec{u}$

so $\frac{dg}{dt} = \nabla\Phi(x)\cdot\vec{u} = D_{\vec{u}}\Phi(x)$

the directional derivative

- Back to $\Phi(x_1,x_2) = 9x_1^2 + x_2^2$, Find $D_u\Phi(x)$ which $\vec{x} = (1,2)$ and $\vec{u} = \left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right)$. Note $\nabla\Phi = [18x_1, \cancel{\#}2x_2)$
- $D_u f(x) = \nabla\Phi(x)\cdot\vec{u} = (18,4)\cdot\left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right) = 9\sqrt{3} + 2.$

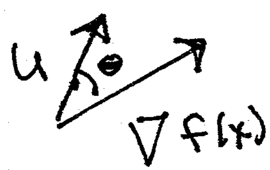


- Recall our goal is to find the direction in which Φ is decreasing most rapidly

- We know that the rate of change of Φ at $\vec{x}$ in the direction $\vec{u}$ is

$$D_u \Phi(x) = \nabla \Phi(x) \cdot \vec{u} = |\nabla \Phi(x)|\,|\vec{u}| \cos\theta$$

where θ is the angle between $\nabla \Phi(x)$ and $\vec{u}$



Now we now $\cos\theta$ is most positive when $\theta = 0$ $(\cos(0)=1)$ and most negative when $\theta = \pi$ $(\cos(\pi)=-1)$

A function is increasing when $g' > 0$ and decreasing when $g' < 0$ so

$g(t) = \Phi(\gamma(t))$ has its maximum increase when $\theta = 0$ or when $\frac{d\gamma}{dt} = \vec{u}$ is parallel to $\nabla\Phi(x)$ and has its maximum decrease when $\frac{d\gamma}{dt}$ points in the opposite direction

$|\nabla\Phi(x)|$

$\cos\theta$

Since $D_u \Phi(x) = |\nabla\Phi(x)|\,|\vec{u}|\cos\theta =$ the result we want is since $\vec{u}$ is a unit vector

Theorem The direction of maximum decrease of Φ at the point x is the direction of $-\nabla\Phi(x)$

Important minus sign.

So back to $\Phi(x_1, x_2) = 9x_1^2 + x_2^2$.

the direction of maximum decrease of Φ at $(1,2)$ is $-\nabla\Phi(1,2) = -(18, 2)$

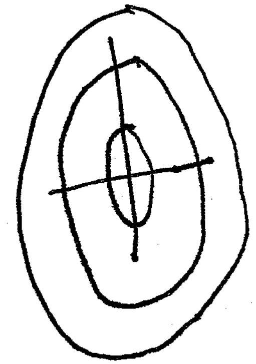
- Before we get back to the task of diminishing Φ to decrease the error (learning) we need to learn one more thing about $\nabla\Phi$

- A level set of Φ is a set of the form $\{\vec{x} : \Phi(\vec{x}) = c\}$ for some constant c

For $\Phi(x_1, x_2) = 9x_1^2 + x_2^2 = c$, dividing by $c > 0$

we get $\frac{x_1^2}{\frac{c}{9}} + \frac{x_2^2}{c} = 1$ which

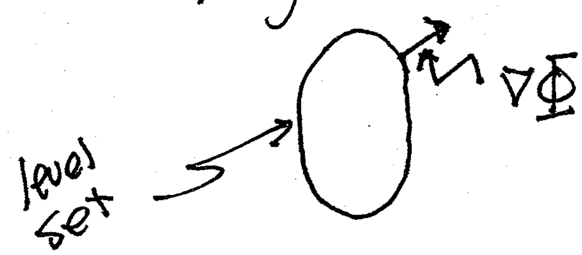
is an ellipse with x_1 width $\frac{\sqrt{c}}{3}$ and x_2 width $\sqrt{c}$



Now recall that $D_u \Phi(x) = \nabla \Phi(x) \cdot \vec{u} = |\nabla \Phi(x)| \cos \theta$

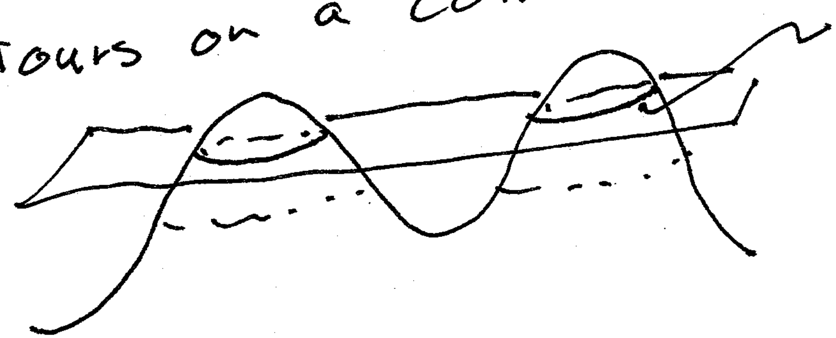
This is zero when $\theta = \frac{\pi}{2}, \frac{3\pi}{2}$.

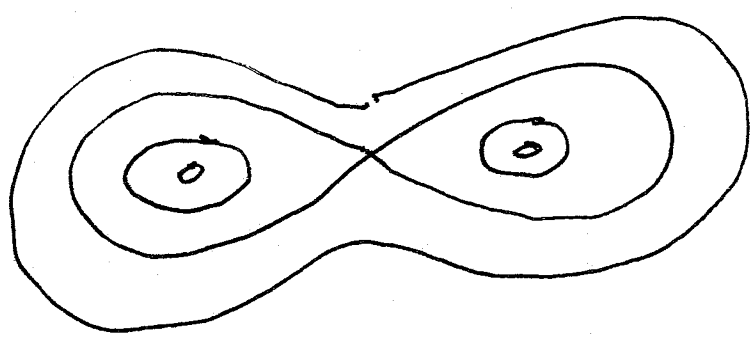
So ~~In~~ the direction of the level set where Φ is not changing is perpendicular to $\nabla \Phi$



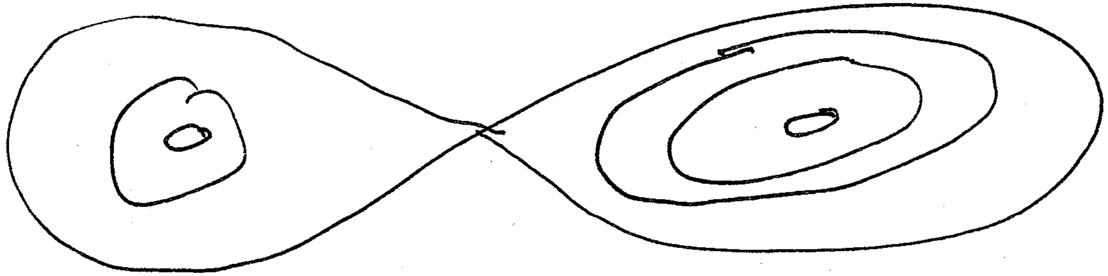
<u>Summary</u>: Given $\Phi: \mathbb{R}^n \to \mathbb{R}$ at a point $\vec{x} \in \mathbb{R}^n$ the direction of maximal increase of Φ is given by $\nabla\Phi(x)$ and the direction of maximal decrease is given by $-\nabla\Phi(x)$. Further, for a level set $L_c = \{\vec{x} \in \mathbb{R}^n : \Phi(\vec{x}) = c\}$ at a point $\vec{x}$ in L_c, $\nabla\Phi(\vec{x})$ is perpendicular to the level set [provided level set is nice at x, no corners, etc]

- How do we use this information to decrease the error Φ
- Let's stick with the role of $\vec{x}$ as the independent variable
- As an example, let $\Phi(x_1, x_2)$ be the height of a terrain. So the level sets are called contours on a contour map

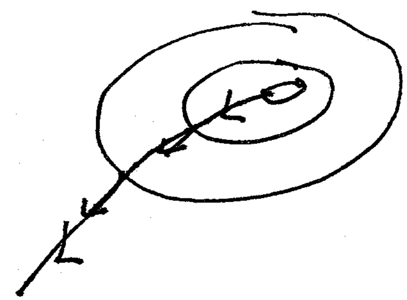




you want to get off the mountain as quickly as possible

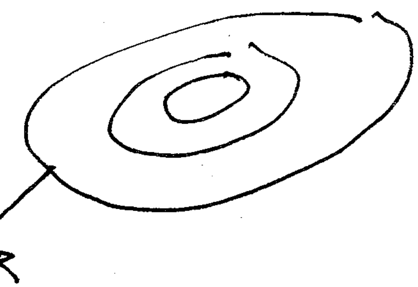


So at each step you go in the direction of steepest descent (biggest decrease in Φ) which is $-\nabla\overline{\Phi}$ and this will be perpendicular to the contour lines

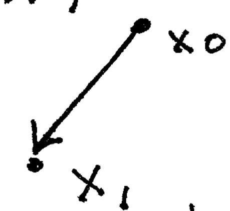


If your path is $\gamma(t)$ its satisfies the differential equation $\frac{d\gamma}{dt}(t) = -\nabla\Phi(\gamma(t))$

rather than solve this, we discretize and take jumps since we have to take discret steps.



So if your intial position is $\vec{x}_0$ and your steps have length h after one step you are at $\vec{X}_0 + h(-\nabla\Phi(x_0)) = \vec{X}_1$, your new position



[We are assuming that your step h is magnified by $|\nabla\Phi(x_0)|$. On your next step

$$\vec{X}_2 = \vec{X}_1 - h\nabla\Phi(x_1), \text{ etc.}$$

So gradient descent is given by

- Requires initial point $\vec{x}_0$ and subroutine to compute $\nabla \Phi$ and step size h
- for $i = 1$ to n

  $x_i = x_{i-1} - h \nabla \Phi(x_{i-1})$

  end.

This is the basic outline, but it raises many questions

(1) How do you choose h = stepsize or learning rate?

(2) How do you choose n, or other halting condition?