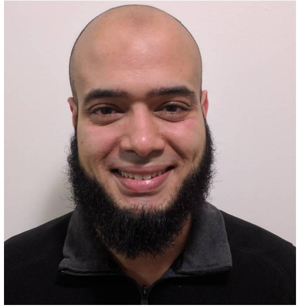
Machine Learning

Multivariate Chain Rule

Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / MSc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Function Composition

- An operation where two functions say f and g generate a new function say h in such a way that h(x) = g(f(x)).
- It means here function g is applied to **the output of** function of x
- Example: y = sin(sigmoid(**sqrt(x)**))
    - Given x
    - Compute s = sqrt(x)
    - Then Compute t = sigmoid(s)
    - Then Compute y = sin(t)

Chain Rule

- A rule that makes our life easy when we compute the derivative of a composition of functions
- Example:
  - Let y = sin(sigmoid(**sqrt(x)**))
  - Compute ∂y/∂x
- Rule
  - $$\frac{d}{dx}[f(g(x))] = f'(g(x))g'(x)$$

$$\frac{d}{dx}\left[f\left(g(h(x))\right)\right] = f'\left(g(h(x))\right)g'(h(x))h'(x)$$

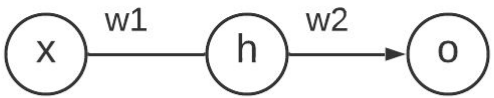
Example 1

- Compute $\partial y/\partial x$ where $y = (3x+5)^4$
- Use series of **symbols and compute partial derivatives relative to them then multiply their results**
  - **$y = a^4$**
  - **$a = 3x+5$**
- Compute $\partial y/\partial x = \partial y/\partial a * \partial a/\partial x$
- $\partial y/\partial a = 4a^3$
- $\partial a/\partial x = 3$
- $\partial y/\partial x = 4a^3 * 3 = 4(3x+5)^3 * 3 = 12 (3x+5)^3$

Example 2

- Compute $\partial y/\partial x$ where $y = 2x^3 + (3x+5)^4$
- The rule here just **add** the parts together
- $\partial/\partial x\ 2x^3 + \partial/\partial x\ (3x+5)^4$
- $6x^2 + 12\ (3x+5)^3$

Example 3



Assume h and o are followed by activation $f(a) = a^3$

$E = (o-t)^2$

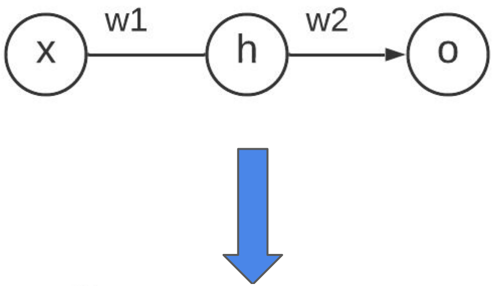
Compute **$\partial E / \partial w1$**

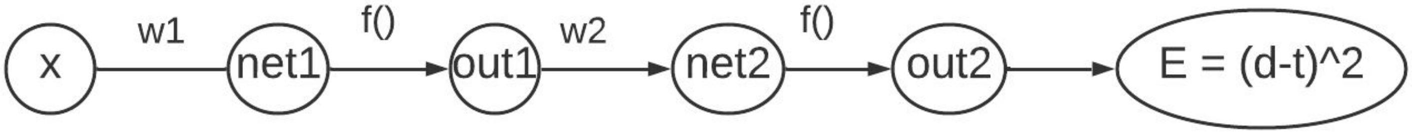
- Let's express E (a simple NN) fully mathematically
- $E = (f(\ f(x * w1) * w2\ ) - t)^2$
- Express as series of symbols [tip start from the inner $x * w1 = a$]
  - $E = (d-t)^2$ where $d = f(\ f(x * w1) * w2\ )$
  - $d = f(c)$
  - $c = b * w2$ where $c = f(x * w1) * w2$
  - $b= f(a)$ node h represents **2 operations**: a and b
  - $a = x * w1$
- $\partial E/\partial w1 = \partial E/\partial d * \partial d/\partial c * \partial c/\partial b * \partial b/\partial a * \partial a/\partial w1$
- $\partial E/\partial w1 = 2d \quad * 3c^2 \quad * w2 \quad * 3a^2 \quad * w1$

Example 3: Observe

Assume h and o are
followed by activation
$f(a) = a^3$

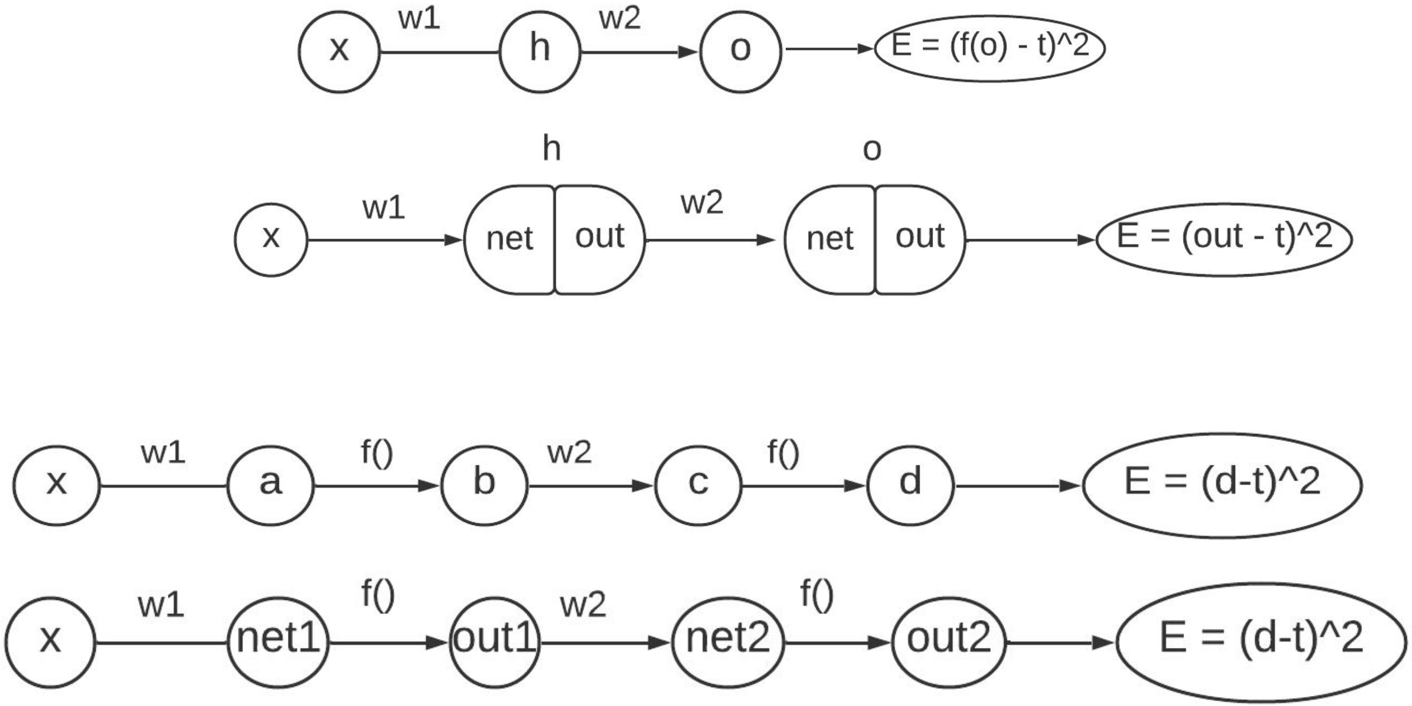
$E = (o-t)^2$
Compute $\partial E / \partial w1$





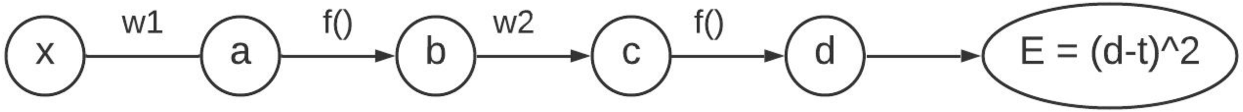
- Observe, every original node actually consists of **2 operations** (due to activation), then it is actually **2 nodes** (NN notation is to merge)
  - Step 1) Compute net: sum wi * inpi step2: activation(net)

Example 3: Correspondance



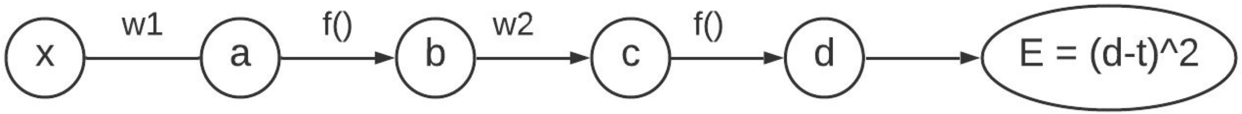
Multivariate Chain Rule

- The previous example is actually about multivariables (w1 and w2)
- It highlights this connection between complex functions and DAG
- We see that From E to W1 we need to pass with many steps (**nodes**)



- We found that its chain rule is the multiplication of all these *∂/∂*
  - ∂E/∂w1 = ∂E/∂d * ∂d/∂c * ∂c/∂b * ∂b/∂a * ∂a/∂w1
- In fact, we can generalize that to a **tree diagram / computational graph**

Chain Components



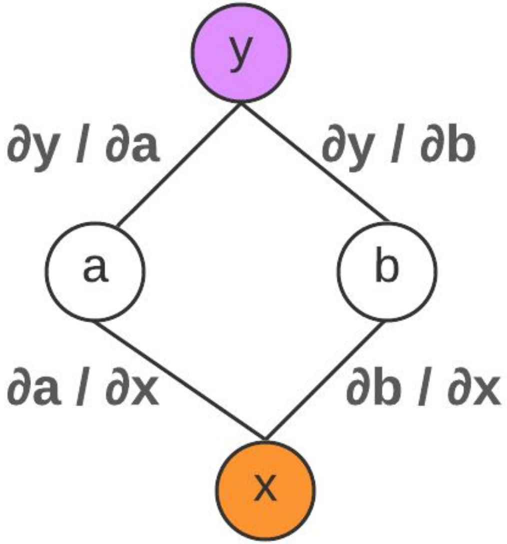
- $\partial E/\partial w1 = \partial E/\underline{\partial d * \partial d}/\partial c * \partial c/\partial b * \partial b/\partial a * \partial a/\partial w1$
- $\partial E/\partial w1 = \partial E/\underline{\partial c * \partial c}/\partial b * \partial b/\partial a * \partial a/\partial w1$
- $\partial E/\partial w1 = \partial E/\underline{\partial b * \partial b}/\partial a * \partial a/\partial w1$
- $\partial E/\partial w1 = \partial E/\underline{\partial a * \partial a}/\partial w1$
- $\partial E/\partial w1 = \partial E/\partial d * \partial d/\partial c * \partial c/\partial b * \partial b/\partial w1$ [canceled $\partial b/\underline{\partial a * \partial a}/\partial w1$]
- $\partial E/\partial w1 = \partial E/\partial d * \partial d/\partial a * \partial a/\partial w1$
- Keep this observation in mind: we can create several sub-path of derivatives from a single chain

From equation to a tree/dag

- Assume we are given a multivariate equation
  - E.g. $z = f(x,y)$, where x and y themselves depend on more variable
- We will draw a tree where its lowest leaves represents our given variables
- We will keep grouping basic operations and create new variables
- We will keep doing the same until reaching a single variable
- This is the root of the tree represents our expression

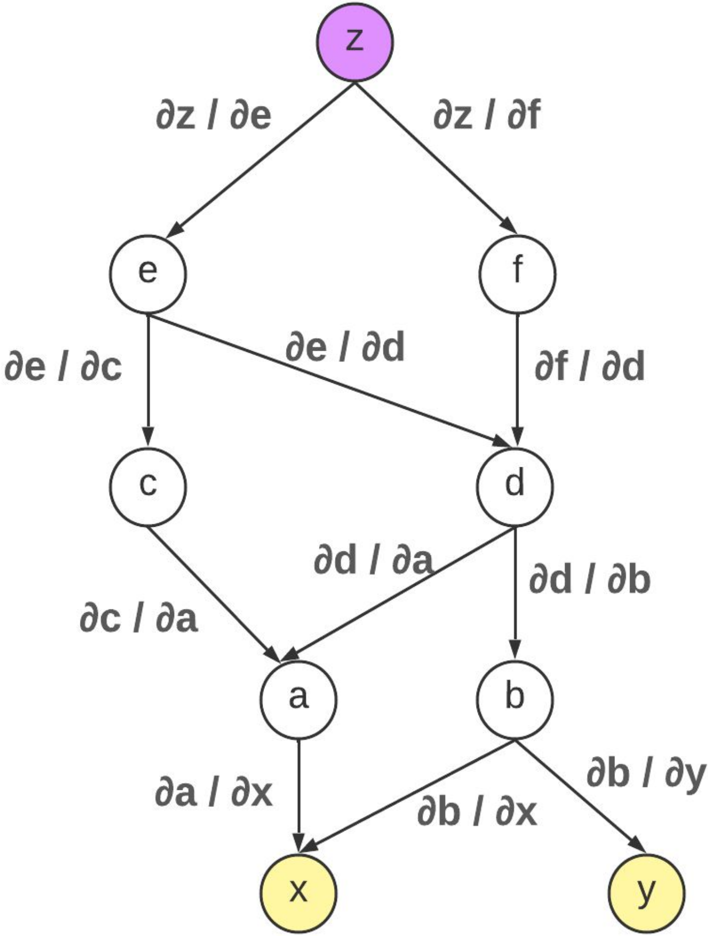
Example 4

- Let $y = 2x^5 - 4x^2$
  - This is actually a **univariate** variable (x)
  - So the bottom leave is just x
  - We can create 2 new variables (nodes)
  - $a = 2x^5$ and $b = 4x^2$
  - Finally we create a higher level y
- Observe every edge is a derivative
  - Edge ($g \Rightarrow h$) represents $\partial g / \partial h$
  - There are 2 paths
  - $y \Rightarrow a \Rightarrow x$: a **chain rule** with value $\partial y / \partial x$
  - $y \Rightarrow b \Rightarrow x$: a **chain rule** with value $\partial y / \partial x$
  - Then to compute $\partial y / \partial x$: **sum** the results from the 2 paths



Example 5

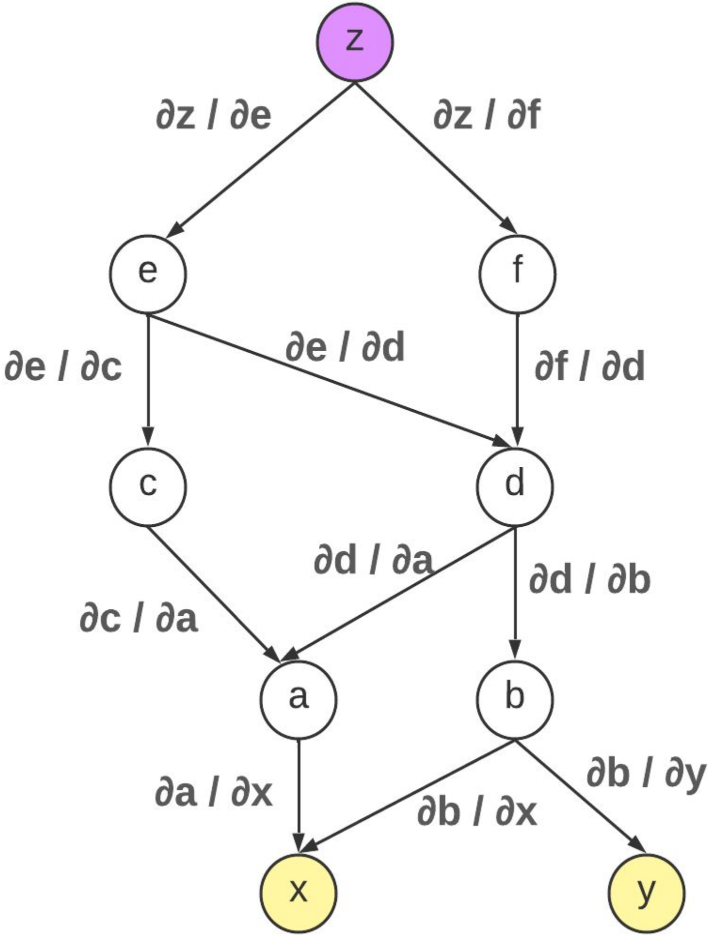
- Assume we have z = f(x, y)
  - Put x and y in the leaves
  - Build the tree up to z
- To compute any **partial derivative** from node(m) to node(n)
  - Find all paths from m to n
    - Each path is a simple chain
    - Multiply path value ⇒ chain rule value
  - Sum all the paths



Example 5A

- Compute ∂d / ∂x

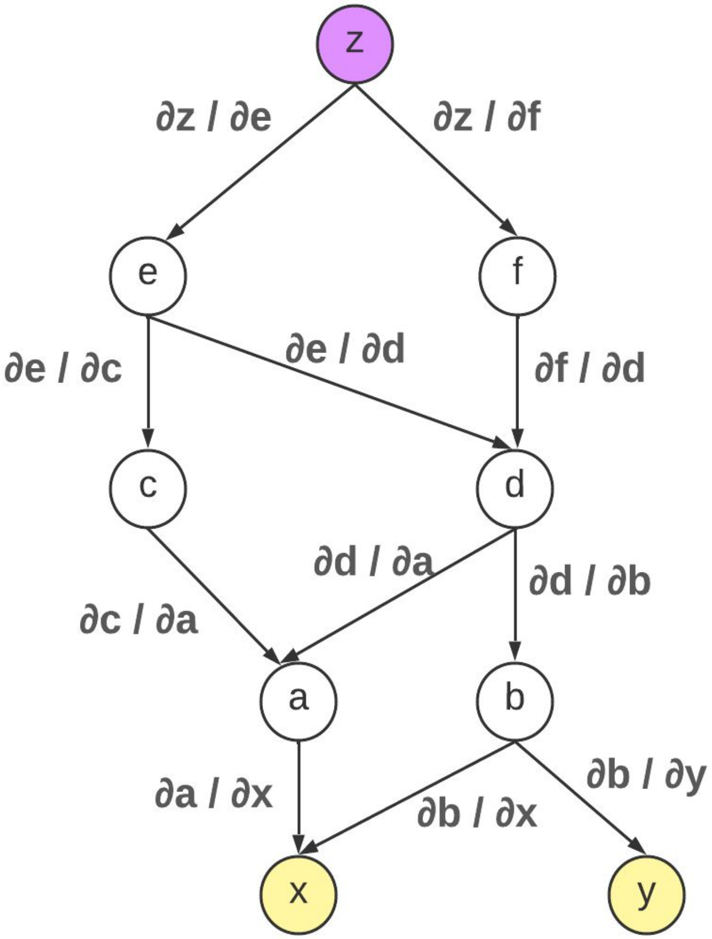
- We have 2 paths
- d ⇒ a ⇒ x
  - Represents: ∂d / ∂a * ∂a / ∂x
- d ⇒ b ⇒ x
  - Represents: ∂d / ∂b * ∂b / ∂x
- Let's pretend that our calculations are
- ∂d / ∂x = 4



Example 5B

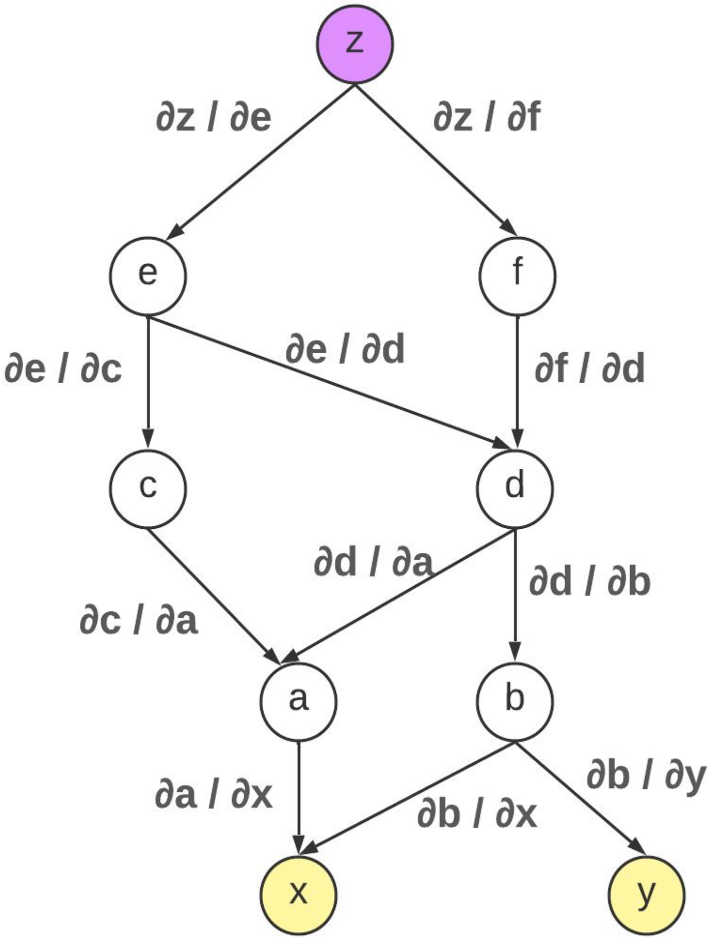
- Compute ∂f / ∂x
    - Assume ∂f / ∂d = 3

- We have 2 paths
- f ⇒ d ⇒ a ⇒ x
- f ⇒ d ⇒ b ⇒ x
- Multiply and sum
- But this is waste of time!
- Can you find it **faster**!



Example 5C

- Compute ∂f / ∂x
  - Assume ∂f / ∂d = 3
- ∂f / ∂x = **∂f / ∂d *** ∂d / ∂x
  - We already computed ∂d / ∂x = 4
  - Then ∂f / ∂x = 3 * 4 = 12
  - This caching trick is the core of backpropagation algorithm
  - It is simply based on bottom-up processing starting **from x and y up to z**



Relevant Materials

- Try to solve some of the examples in this page
  - Solve #: 3, 11, 12, 13
  - This online calculator might be helpful
- Link
- Link
- Link

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”