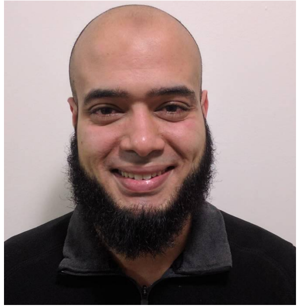
Machine Learning

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 in which two functions, (in this case, f and g), combine to generate a new function, (h), such that:
  $h(x) = g(f(x))$.
- This means that function g is applied to **the output of** function f for x
- Example: y = sin(sigmoid(**sqrt(x)**))
  - Given x:
  - Compute s = sqrt(x)
  - Then compute t = sigmoid(s)
  - Then compute y = sin(t)

Recall 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
  - $$\frac{d}{dx}[f(g(x))] = f'(g(x))g'(x)$$

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

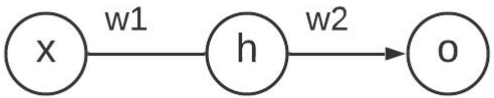
Example 1

- 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



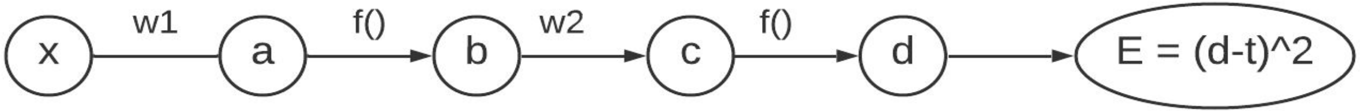
Assume h and o are followed by activation f(a) = a^3

E = (o-t)^2

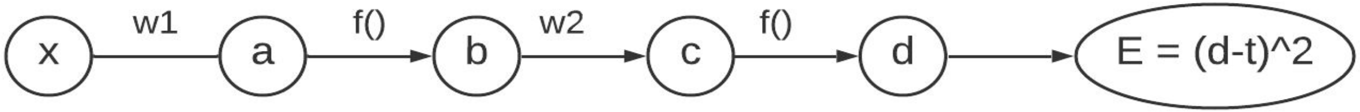
Compute **∂E / ∂w1**

In other words, in the extended form:

- a = w1 * x
  - So h represents and b
- b = f(a) = $(w1 * x)^3$
- c = w2 * b = w2 * $(w1 * x)^3$ and so on

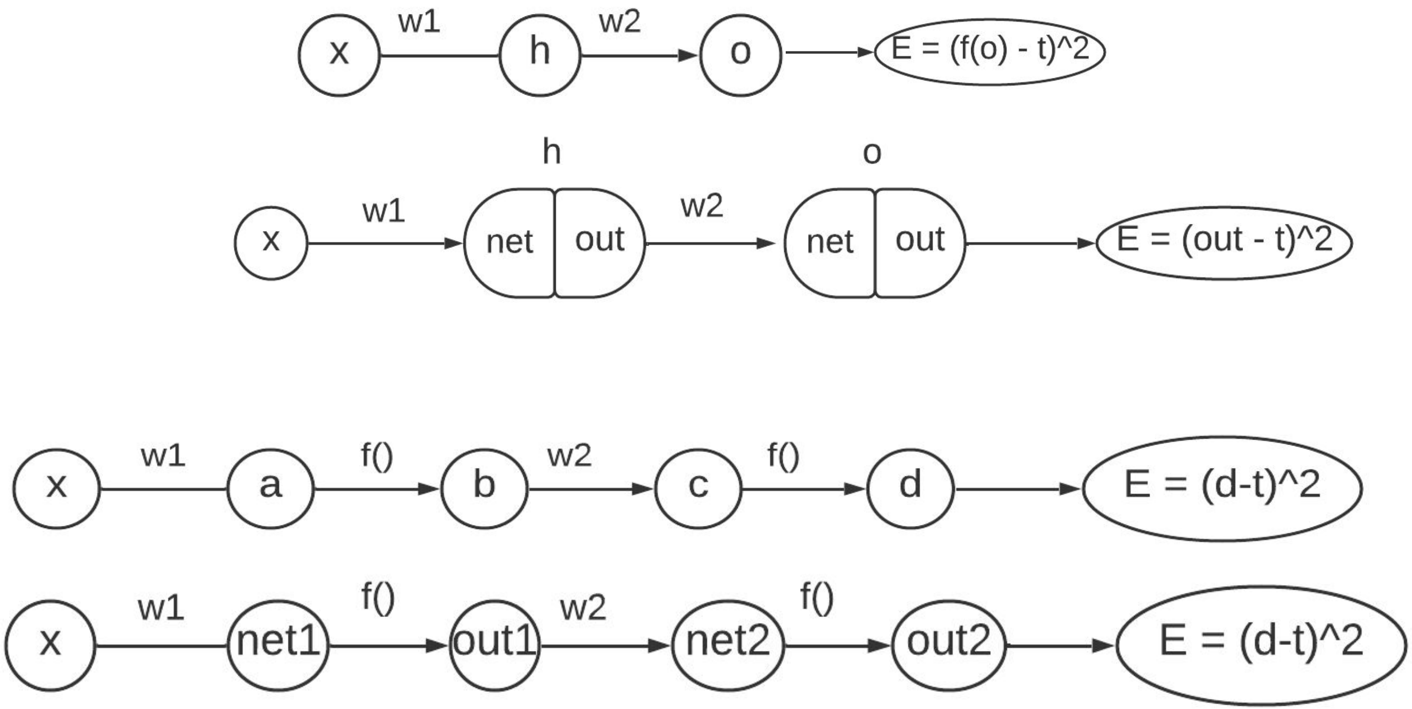


Example 3



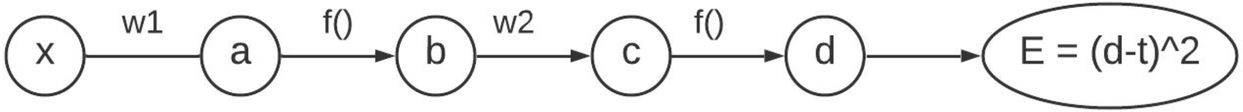
- Consider E (a simple NN) fully mathematically
- $E = (f(\ \ f(x * w1) * w2\ \ ) - t)^2$
- Express as series of symbols [tip start from the inner x * w1 = a]
  - $E = (d-t)^2$ where $d = f(\ \ f(x * w1) * w2\ \ )$
  - $d = f(c)$
  - $c = b * w2$ where $c = f(x * w1) * w2$
  - $b= f(a)$ node h represents **2 operations**: a and b
  - $a = x * w1$
- $\partial E/\partial w1 = \partial E/\partial d * \partial d/\partial c * \partial c/\partial b * \partial b/\partial a * \partial a/\partial w1$
- $\partial E/\partial w1 = 2(d-t) \quad * 3c^2 \quad * w2 \quad * 3a^2 \quad * x$

Example 3: Correspondence



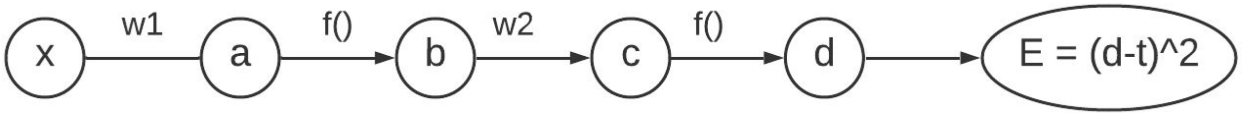
Multivariate Chain Rule

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



- The chain rule for multivariables involves the multiplication of partial derivatives, as shown below: *∂/∂*
  - ∂E/∂w1 = ∂E/∂d * ∂d/∂c * ∂c/∂b * ∂b/∂a * ∂a/∂w1
- In fact, this generalizes to a tree diagram or computational graph

Chain Components



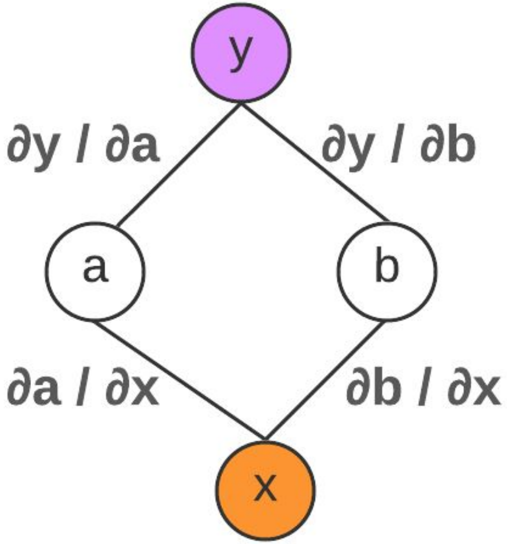
- $\partial E/\partial w1 = \partial E/\underline{\partial d * \partial d}/\partial c * \partial c/\partial b * \partial b/\partial a * \partial a/\partial w1$
- $\partial E/\partial w1 = \partial E/\underline{\partial c * \partial c}/\partial b * \partial b/\partial a * \partial a/\partial w1$
- $\partial E/\partial w1 = \partial E/\underline{\partial b * \partial b}/\partial a * \partial a/\partial w1$
- $\partial E/\partial w1 = \partial E/\underline{\partial a * \partial a}/\partial w1$
- $\partial E/\partial w1 = \partial E/\partial d * \partial d/\partial c * \partial c/\partial b * \partial b/\partial w1$ [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-paths of derivatives from a single chain

From an Equation to a Tree/DAG

- Suppose we have a multivariate equation, for example, $z = f(x, y)$
  - where x and y depend on other variables
- We can represent this equation as a tree, with the lowest nodes (or leaves) representing our given variables, x and y
- We group basic operations and create new variables until we reach a single variable
- This becomes the root of the tree representing our expression
  - This process of building a tree helps us visualize and understand the composition of complex functions

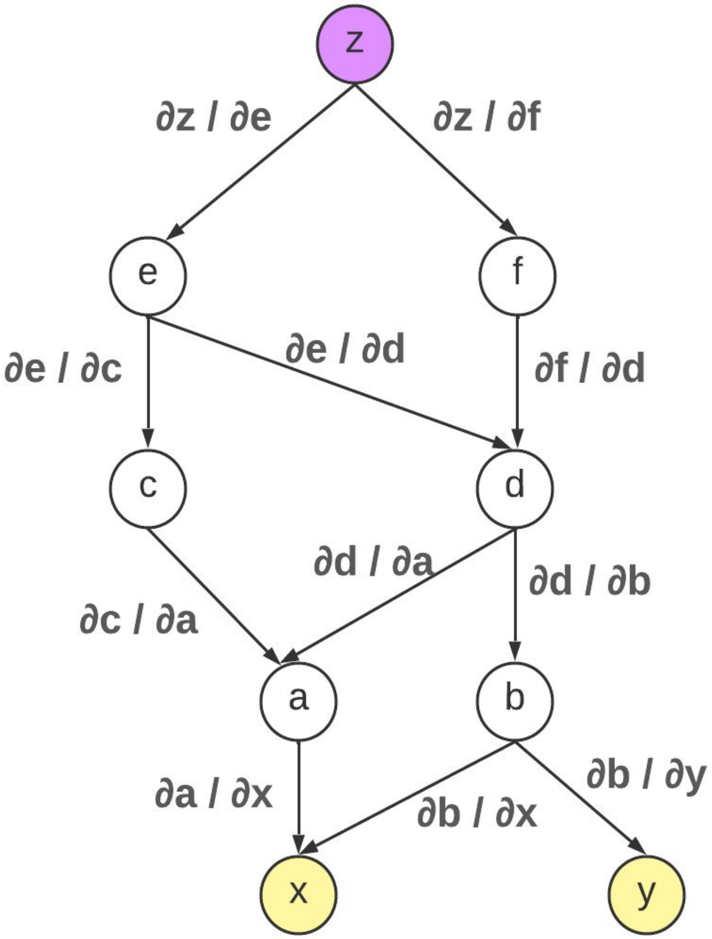
Example 4

- Let $y = 2x^5 + 4x^2$
  - This is actually a **univariate** variable (x)
  - The bottom leaf node of the tree represents x
  - We can create 2 new variables (nodes)
  - $a = 2x^5$ and $b = 4x^2$
  - Finally, we create a higher-level node y that combines a and b with addition
- Observe that every edge in the tree represents a derivative
  - Rule for any Edge ($g \Rightarrow h$), it represents $\partial g / \partial h$
  - There are two paths from y to x: one through a and one through b
  - $y \Rightarrow a \Rightarrow x$: 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 $\partial y / \partial x$: we **sum** the results from these 2 paths



Example 5

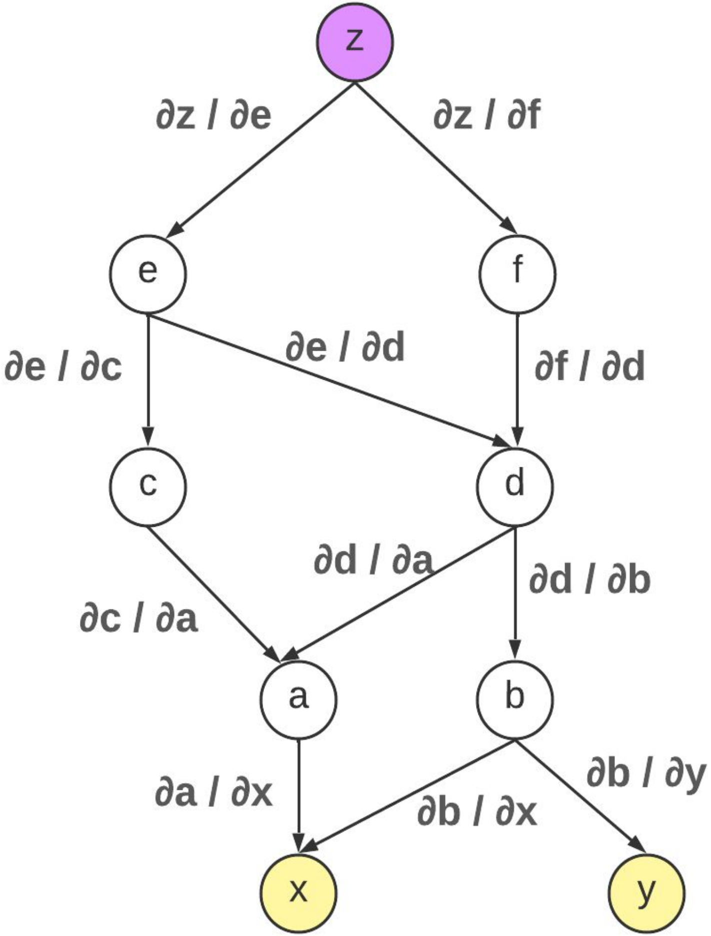
- Assume we have z = f(x, y)
  - Put x and y in the leaves
  - Build the tree up to z
- To compute any **partial derivative** from node(m) to node(n)
  - Find all paths from m to n
    - Each path is a simple chain
    - Multiply path value ⇒ chain rule value
  - Sum all the paths



Example 5A

- Compute $\partial d / \partial x$

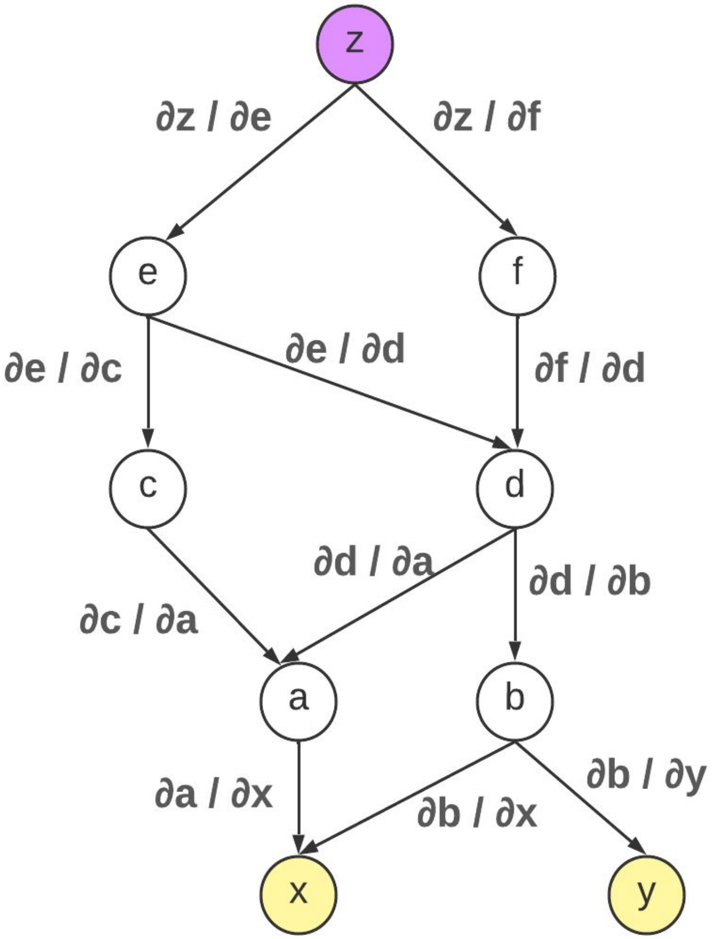
- We have 2 paths
- $d \Rightarrow a \Rightarrow x$
  - This represents: $\partial d / \partial a * \partial a / \partial x$
- $d \Rightarrow b \Rightarrow x$
  - This represents: $\partial d / \partial b * \partial b / \partial x$
- Let’s pretend that our calculations are
- $\partial d / \partial x = 4$



Example 5B

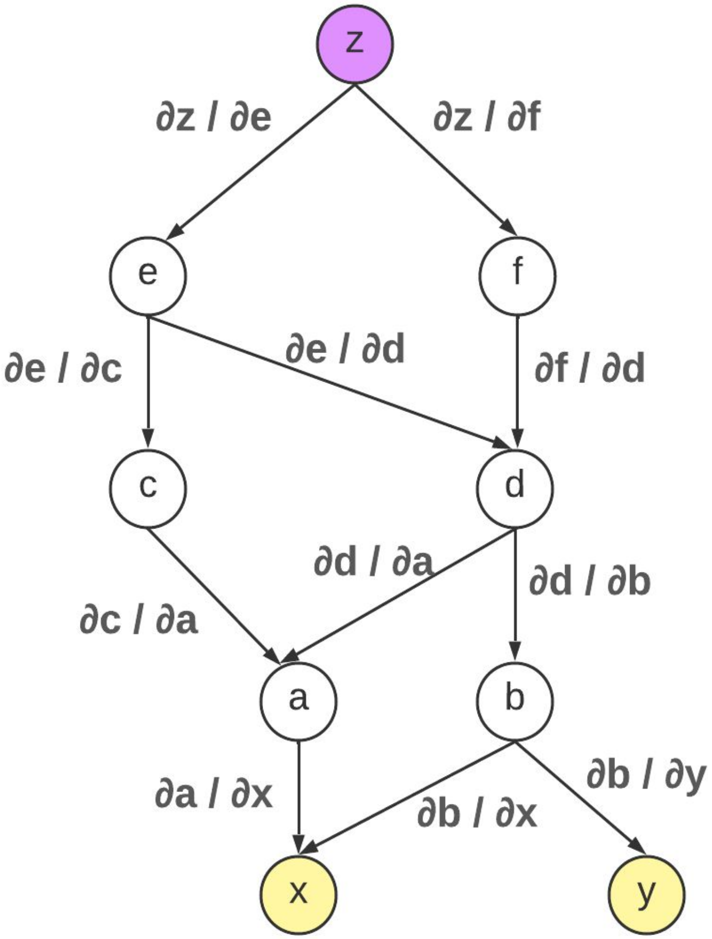
- Compute $\partial f / \partial x$
  - Assume $\partial f / \partial d = 3$

- We have 2 paths
- $f \Rightarrow d \Rightarrow a \Rightarrow x$
- $f \Rightarrow d \Rightarrow b \Rightarrow x$
- We could multiply and sum the chain rule values...
- But this is a waste of time!
- Can you find it **more quickly?**



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 the backpropagation algorithm
    - It is simply based on bottom-up processing starting **from x and y up to z**



Relevant Materials

- Link
- Link
- Link

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."