The chain rule and the idea of ‘backpropagation’.

Group 2 - Ferschl Martin, Laktaoui Mohammed, Reiter Roman, Zenkic Mirza

December 8, 2025

1 Task specification

Let $f(x)$ be a function that depends on two parameters a and b. (More precisely, $f = f(x; a, b)$, meaning that f is actually a function in 3 variables.) Furthermore, let $F(x) = f(f(x + a) + b)$.

1. Compute the partial derivatives F_a and F_b of F with respect to a and b in terms of the corresponding derivatives of f. (This requires use of the **Chain Rule**).
2. Note that the expression for the partial derivative F_b also appears in the derivative F_a. Plan to first compute F_b and then substitute the result into F_a.
3. Write everything down precisely and explain the procedure.

2 Procedure and Definitions

To apply the chain rule precisely we must define the intermediate variables representing the “layers” of this composition. This approach mirrors the “backpropagation” method used in neural networks, where derivatives are computed layer by layer.

Let us define the partial derivatives of the function $f(u; a, b)$ with respect to its three arguments:

- $f_u(u; a, b) = \frac{\partial f}{\partial u}$ (Derivative with respect to the first argument/input).
- $f_a(u; a, b) = \frac{\partial f}{\partial a}$ (Derivative with respect to parameter a).
- $f_b(u; a, b) = \frac{\partial f}{\partial b}$ (Derivative with respect to parameter b).

Step-by-Step Decomposition:

1. **Inner Layer Input:** Let $z = x + a$.
2. **Inner Layer Output:** Let $y_{\text{inner}} = f(z; a, b)$.
3. **Outer Layer Input:** Let $y_{\text{outer}} = y_{\text{inner}} + b$.
4. **Final Output:** $F = f(y_{\text{outer}}; a, b)$.

We will compute the derivatives by moving from the outer layer inward.

3 Computation of F_b

We seek $\frac{\partial F}{\partial b}$. The parameter b influences F through three distinct paths:

1. Directly as a parameter in the outer function f.
2. Explicitly via the term $+b$ in the argument y_{outer}.

3. Implicitly via the parameter b inside the inner function y_{inner}.

Using the chain rule:

$$F_b = \frac{\partial F}{\partial b} = \underbrace{\frac{\partial f}{\partial u}(y_{\text{outer}}; a, b)}_{\text{Outer derivative}} \cdot \frac{\partial y_{\text{outer}}}{\partial b} + \underbrace{\frac{\partial f}{\partial b}(y_{\text{outer}}; a, b)}_{\text{Direct parameter derivative}}$$

Now we compute $\frac{\partial y_{\text{outer}}}{\partial b}$. Since $y_{\text{outer}} = f(z; a, b) + b$:

$$\frac{\partial y_{\text{outer}}}{\partial b} = \frac{\partial}{\partial b}(f(z; a, b) + b) = f_b(z; a, b) + 1$$

Substituting this back:

$$F_b = f_u(y_{\text{outer}}; a, b) \cdot [f_b(z; a, b) + 1] + f_b(y_{\text{outer}}; a, b)$$

This establishes the derivative with respect to b. Note that the term $f_u(y_{\text{outer}}; a, b)$ represents the sensitivity of the outer layer to its input, often denoted as the error signal or δ in backpropagation.

4 Computation of F_a

We seek $\frac{\partial F}{\partial a}$. The parameter a influences F through three paths:

1. Directly as a parameter in the outer function f.
2. Implicitly via the parameter a inside the inner function y_{inner}.
3. Implicitly via the term $+a$ in the innermost input z.

Using the chain rule:

$$F_a = \frac{\partial F}{\partial a} = \underbrace{\frac{\partial f}{\partial u}(y_{\text{outer}}; a, b)}_{\text{Outer derivative}} \cdot \frac{\partial y_{\text{outer}}}{\partial a} + \underbrace{\frac{\partial f}{\partial a}(y_{\text{outer}}; a, b)}_{\text{Direct parameter derivative}}$$

Notice that the first term, $f_u(y_{\text{outer}}; a, b)$, is the exact same factor we computed for F_b. This confirms the note in the exercise: we can reuse this computed value.

Now we compute $\frac{\partial y_{\text{outer}}}{\partial a}$. Since $y_{\text{outer}} = f(z; a, b) + b$ (and b is constant w.r.t a):

$$\frac{\partial y_{\text{outer}}}{\partial a} = \frac{\partial}{\partial a} f(z; a, b)$$

To differentiate the inner function $f(z; a, b)$ with respect to a, we must apply the chain rule again because $z = x + a$ depends on a:

$$\frac{\partial}{\partial a} f(z; a, b) = f_u(z; a, b) \cdot \frac{\partial z}{\partial a} + f_a(z; a, b)$$

Since $z = x + a$, $\frac{\partial z}{\partial a} = 1$. Thus:

$$\frac{\partial y_{\text{outer}}}{\partial a} = f_u(z; a, b) \cdot (1) + f_a(z; a, b) = f_u(z; a, b) + f_a(z; a, b)$$

Substituting this back:

$$F_a = f_u(y_{\text{outer}}; a, b) \cdot [f_u(z; a, b) + f_a(z; a, b)] + f_a(y_{\text{outer}}; a, b)$$

5 Summary of Results

Using the notation $z = x + a$ and $y = f(x + a; a, b) + b$ (where y is shorthand for y_{outer}):

1. **Partial Derivative F_b:**
$$F_b = f_u(y) \cdot [f_b(z) + 1] + f_b(y)$$

2. **Partial Derivative F_a:**
$$F_a = f_u(y) \cdot [f_u(z) + f_a(z)] + f_a(y)$$

Explanation of the Procedure: The procedure demonstrates the principle of **backpropagation**. We calculated the derivative of the outermost layer first. The term $f_u(y)$ appears in the calculation for both F_a and F_b. In a computational graph, this value would be computed once and propagated backwards to determine how the changes in parameters a and b affect the inner components and, ultimately, the final output. By substituting the previously calculated sensitivity of the outer layer, we efficiently derive the gradients for all parameters.