From Differential Calculus to Automatic Differentiation

Gabriel Peyré

November 10, 2019

Part 1 – Gradients

1) For $g : \mathbb{R} \to \mathbb{R}$, compute the derivative of $x \in \mathbb{R} \mapsto g(ax + b) \in \mathbb{R}$ using the definition of the derivative and using the chain rule.

2) For $g : \mathbb{R}^p \to \mathbb{R}$ and $A \in \mathbb{R}^{p,n}, b \in \mathbb{R}^p$, compute the derivative of $x \in \mathbb{R}^n \mapsto g(Ax + b) \in \mathbb{R}$ using the definition of the derivative and using the chain rule.

3) What is the gradient of $x \in \mathbb{R}^n \mapsto \|x\|_p$ where $\|x\|_p^p \overset{\text{def.}}{=} \sum_i |x_i|^p$ (and give the domain on which it is differentiable)?

4) Compute the gradient of $X \in \mathbb{R}^{n\times n} \mapsto \mathrm{tr}(X) = \sum_i X_{i,i}$, and $X \in \mathbb{R}^{n\times n} \mapsto \det(A)$.

Part 2 – Jacobians

1) Using the definition of the Jacobian, compute the Jacobian of $X \in \mathbb{R}^{n\times p} \mapsto X^\top$, $X \in \mathbb{R}^{n\times p} \mapsto XX^\top$, $X \in \mathbb{R}^{n\times n} \mapsto X^2$, $X \mapsto X^{-1}$.

2) What is the Jacobian of $X \in \mathbb{R}^{n\times p} \mapsto AXB$ where A and B are also matrices (you will indicate their size and the size of the output matrix).

3) If $X \in \mathcal{S}_n^+$ is symmetric positive semi-definitive (i.e. its eigenvalues are positives), show that there exists a unique matrix $\sqrt{X} \in \mathcal{S}_n^+$ such that $X = \sqrt{X}\sqrt{X}$. Compute the Jacobian of $X \mapsto \sqrt{X}$.

Part 3 – Gradients for matrix functions

1) Compute the gradient of $X \in \mathbb{R}^{n\times n} \mapsto \mathrm{tr}(X^2)$, $X \in \mathbb{R}^{n\times n} \mapsto \det(X^2)$ using the definition of a differential and the chain rule.

2) Same question for $X \in \mathbb{R}^{n\times p} \mapsto \mathrm{tr}(XX^\top)$, $X \in \mathbb{R}^{n\times p} \mapsto \det(XX^\top)$.

3) Same question for $X \in \mathbb{R}^{n\times p} \mapsto \mathrm{tr}(\sqrt{XX^\top})$. When $X \in \mathbb{R}^{n\times 1}$ or $X \in \mathbb{R}^{1\times p}$, what formula do you recognize ?

Part 4 – Smoothed total variation

1) What are the derivative and second derivative of $f : x \in \mathbb{R} \mapsto \sqrt{x^2+\varepsilon^2}$. Prove that f has a Lipschitz derivative and give an upper bound on the Lipschitz constant.

2) Same question with $x \in \mathbb{R}^n \mapsto \|x\|_\varepsilon \overset{\text{def.}}{=} \sum_{i=1}^n \sqrt{x_i^2+\varepsilon^2}$.

3) For $x \in \mathbb{R}^n$, we consider the vector of finite differences ("discretized gradient") $Gx \overset{\text{def.}}{=} (x_2-x_1, x_3-x_2, \dots, x_n-x_{n-1}) \in \mathbb{R}^{n-1}$. Show that G is linear and compute its adjoint $u \in \mathbb{R}^{n-1} \mapsto G^\top u \in \mathbb{R}^n$.

4) Compute the gradient of the 1-D smoothed total variation $x \in \mathbb{R}^n \mapsto \|Gx\|_\varepsilon \in \mathbb{R}$.

5) What is the limit as $\varepsilon \to 0$ of $\|\cdot\|_\varepsilon$ and of its gradient? When is this limit differentiable?

Part 5 – Calculus graph and differentiation modes

1) Produce an efficient computational graph (DAG) for the function

$$f(x) = \frac{\log(x+\sqrt{x^2+1})}{\sqrt{x^2+1}} - \log^3(x+\sqrt{x^2+1}).$$

2) Write the pseudo-code associated to the forward differentiation method applied to this graph (i.e. using the classical chain rule).

3) Write the pseudo-cove associated to the backward differentiation method applied to this graph (i.e. using the adjoint chain rule).

4) Which one is the fastest? Why?

Part 6 – Differential calculus for neural layers

We consider a function computed using a neural network with two-layers $f(x, A, b) = b\rho(Ax)$, where $x \in \mathbb{R}^p$, $A \in \mathbb{R}^{q\times p}$ (q is the number of neurons) and $b \in \mathbb{R}^{1\times q}$. Here $\rho : \mathbb{R} \to \mathbb{R}$ is a smooth non-linearity and with a slight abuse of notation, for $u \in \mathbb{R}^q$, we denote $\rho(u) = (\rho(u_k))_{k=1}^q \in \mathbb{R}^q$.

1) What is the Jacobian of $\rho : \mathbb{R}^q \mapsto \mathbb{R}^q$ defined this way? What are the Jacobians of $x \mapsto Ax$ and $A \mapsto Ax$?

2) Using the chain rule, compute the derivative of f with respect to x and with respect to the network weights (A, b). What is its complexity in function of (p, q) ?

3) Implement the same derivative but this time using backward differentiation. What is the resulting complexity ? How does this compare to directly computing the gradient of f by computing its Taylor expansion ?

4) Using $\nabla_{A,b} f$ compute the gradient of the training error $\sum_{i=1}^n (f(x_i, A, b) - y_i)^2$.

5) We now consider a "residual network" $F(x, A) = x + A^\top \rho(Ax) \in \mathbb{R}^n$ (this type of architecture shows up for instance when doing descent methods or discretizing ODEs and PDEs, and one might want to optimize the kernel A). Given some loss function $L : \mathbb{R}^n \to \mathbb{R}$, what is the gradient of $L(F(x, A))$ with respect to x and A? Write the pseudo code to apply the adjoints Jacobian $\left(\frac{\partial F}{\partial x}\right)^\top u \in \mathbb{R}^n$ and $\left(\frac{\partial F}{\partial A}\right)^\top u \in \mathbb{R}^{n\times p}$ to some vector $u \in \mathbb{R}^n$ (typically $u = \nabla L(F(x, A))$), using the backward chain rule.