Automatic differentiation

Daniel Hsu

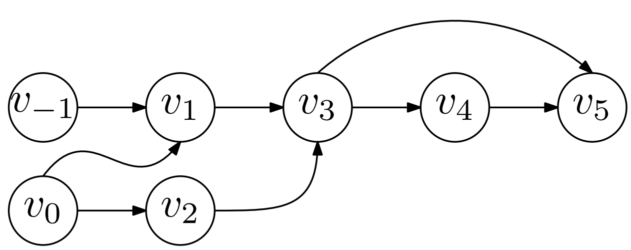
November 13, 2025

1 Straight-line programs and computation graphs

A *straight-line program* (SLP) computes a function of input variables using a sequence of lines, where each line sets the value of a new variable as a function of previously defined variables or input variables; the function must come from a "standard library". The final variable is taken to be the value of the function computed by the SLP. The following is an example of an SLP for a function of input variables (v_{-1}, v_0):

$$\begin{aligned} v_1 &:= \mathrm{prod}(v_{-1}, v_0) \\ v_2 &:= \sin(v_0) \\ v_3 &:= \mathrm{prod}(v_1, v_2) \\ v_4 &:= \mathrm{square}(v_3) \\ v_5 &:= \mathrm{sum}(v_3, v_4) \end{aligned}$$

Let V be the set of all variables in the SLP (including input variables), and let E be the set of ordered pairs of variables (v_i, v_j) such that the value of v_j is determined by a standard library function of v_i (and possibly other variables). Then $G = (V, E)$ defines a directed acyclic graph called the *computation graph* for the SLP. The order of the lines in the SLP gives a topological ordering of the vertices (with all input variables at the front of the ordering). The computation graph for the example SLP above is as follows:



2 Forward pass

Each line (or variable defined in the line) in the SLP corresponds to a function of the input variables. Upon setting the values of the input variables, the *forward pass* computes the "values" of all such function—i.e., the values of the corresponding variables—by evaluating the lines of the SLP in the given order. Each line requires:

- looking up the values of a subset of previously-defined variables or input variables, and then
- applying a standard library function to those values (and storing the result in a new variable).

3 Backward pass

Let F denote the function of the input variables computed by the SLP. For each variable v_i, we can view F as a function of v_i, with all other variables held fixed at their values from the forward pass (which we take to include the setting of the input variables). The *backward pass* computes the value of the partial derivative functions $\frac{\partial F}{\partial v_i}$ for each variable v_i, evaluated at the values of the variables from the forward pass. In particular, this gives the values of the partial derivative functions of F with respect to input variables.

The backward pass is so named because these partial derivatives $\frac{\partial F}{\partial v_i}$ are considered in reverse order: from the end of the SLP to the start of the SLP. (Here, imagine that the input variables are listed at the start of the SLP.) The base case $\frac{\partial F}{\partial v_N}$, where v_N is the last variable defined in the SLP, is the constant 1 function.

To compute $\frac{\partial F}{\partial v_i}$ where v_i is not the last variable, we view F as a composition $F = f \circ g$, where

- f is a function of all variables v_j such that $(v_i, v_j) \in E$, and
- g is a multi-output function of v_i, with one output per v_j such that $(v_i, v_j) \in E$.

Observe that f and F must be identical when viewed as functions of only v_j, with all other variables held fixed at their values from the forward pass. Hence, they have the same partial derivative functions $\frac{\partial f}{\partial v_j} = \frac{\partial F}{\partial v_j}$ with respect to each v_j. Furthermore, the output of g corresponding to v_j is given by the line in the SLP that defines v_j:

$$v_j := g_j(\dots, v_i, \dots)$$

The standard library provides the function g_j, and the standard library is required to also provide the partial derivative function $\frac{\partial g_j}{\partial v_i}$.

The partial derivative of F with respect to v_i (evaluated at x) is obtained via the chain rule of differentiation for function composition:

$$\frac{\partial F}{\partial v_i}(x) = \frac{\partial (f \circ g)}{\partial v_i}(x) = \sum_{(v_i,v_j)\in E} \frac{\partial f}{\partial v_j}(g(x)) \cdot \frac{\partial g_j}{\partial v_i}(x) = \sum_{(v_i,v_j)\in E} \frac{\partial F}{\partial v_j}(g(x)) \cdot \frac{\partial g_j}{\partial v_i}(x).$$

Since $(v_i, v_j) \in E$, it must be that v_j is defined after v_i is defined in the SLP. Because the backward pass considers variables in reverse order, the values of $\frac{\partial F}{\partial v_j}$ will have already been computed by the time v_i is considered. The values of the $\frac{\partial g_j}{\partial v_i}$ are determined by calling the provided standard library functions.

For the example SLP given above, the transcript of the backward pass is as follows (where $\bar{v}_i$ denotes the value of variable v_i from the forward pass):

$$\begin{aligned}
\frac{\partial F}{\partial v_5}(\bar{v}_5) &= 1 \\
\frac{\partial F}{\partial v_4}(\bar{v}_4) &= \frac{\partial F}{\partial v_5}(\bar{v}_5) \cdot \frac{\partial\, \mathrm{sum}(\bar{v}_3, v_4)}{\partial v_4}(\bar{v}_4) \\
&= \frac{\partial F}{\partial v_5}(\bar{v}_5) \cdot 1 \\
\frac{\partial F}{\partial v_3}(\bar{v}_3) &= \frac{\partial F}{\partial v_4}(\bar{v}_4) \cdot \frac{\partial\, \mathrm{square}(v_3)}{\partial v_3}(\bar{v}_3) + \frac{\partial F}{\partial v_5}(\bar{v}_5) \cdot \frac{\partial\, \mathrm{sum}(v_3, \bar{v}_4)}{\partial v_3}(\bar{v}_3) \\
&= \frac{\partial F}{\partial v_4}(\bar{v}_4) \cdot \mathrm{double}(\bar{v}_3) + \frac{\partial F}{\partial v_5}(\bar{v}_5) \cdot 1 \\
\frac{\partial F}{\partial v_2}(\bar{v}_2) &= \frac{\partial F}{\partial v_3}(\bar{v}_3) \cdot \frac{\partial\, \mathrm{prod}(\bar{v}_1, v_2)}{\partial v_2}(\bar{v}_2) \\
&= \frac{\partial F}{\partial v_3}(\bar{v}_3) \cdot \bar{v}_1 \\
\frac{\partial F}{\partial v_1}(\bar{v}_1) &= \frac{\partial F}{\partial v_3}(\bar{v}_3) \cdot \frac{\partial\, \mathrm{prod}(v_1, \bar{v}_2)}{\partial v_1}(\bar{v}_1) \\
&= \frac{\partial F}{\partial v_3}(\bar{v}_3) \cdot \bar{v}_2
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial F}{\partial v_0}(\bar{v}_0) &= \frac{\partial F}{\partial v_1}(\bar{v}_1) \cdot \frac{\partial \operatorname{prod}(\bar{v}_{-1}, v_0)}{\partial v_0}(\bar{v}_0) + \frac{\partial F}{\partial v_2}(\bar{v}_2) \cdot \frac{\partial \sin(v_0)}{\partial v_0}(\bar{v}_0) \\
&= \frac{\partial F}{\partial v_1}(\bar{v}_1) \cdot \bar{v}_{-1} + \frac{\partial F}{\partial v_2}(\bar{v}_2) \cdot \cos(\bar{v}_0) \\
\frac{\partial F}{\partial v_{-1}}(\bar{v}_{-1}) &= \frac{\partial F}{\partial v_1}(\bar{v}_1) \cdot \frac{\partial \operatorname{prod}(v_{-1}, \bar{v}_0)}{\partial v_{-1}}(\bar{v}_{-1}) \\
&= \frac{\partial F}{\partial v_1}(\bar{v}_1) \cdot \bar{v}_0
\end{aligned}
$$

4 Complexity of automatic differentiation

Automatic differentiation of a function F computed by a SLP is the forward pass followed by the backward pass based on the SLP. The forward pass is used to determine the values of all of the variables. The values of all variables defined in the SLP are stored, so the working space needed is $O(|V|)$. If each value look-up requires a unit of time, and each function application also requires a unit of time, then the overall time needed in the forward pass is $O(|V| + |E|)$.

The backward pass is used to compute the partial derivatives of F with respect to all of the variables in the SLP, evaluated at the values from the forward pass. The values of these partial derivatives are stored, so the working space needed is $O(|V|)$. If each value look-up requires a unit of time, each (partial derivative) function application also requires a unit of time, and each linear combination computation requires time proportional to the number of terms being combined, then the overall time needed in the backward pass is $O(|V| + |E|)$.