

# TTIC 31230, Fundamentals of Deep Learning

David McAllester, Autumn 2020

## Backpropagation for Scalar Source Code

# Backpropagation (backprop)

Backpropagation is the method frameworks use to compute  $\nabla_{\Phi} \mathcal{L}_{\Phi}(z)$  for source code  $\mathcal{L}_{\Phi}(z)$ .

## Some Simple Source Code

The expression

$$\mathcal{L} = \sqrt{x^2 + y^2}$$

can be transformed to the assignment sequence

$$u = x^2$$

$$v = y^2$$

$$r = u + v$$

$$\mathcal{L} = \sqrt{r}$$

## Source Code

1.  $u = x^2$
2.  $w = y^2$
3.  $r = u + w$
4.  $\mathcal{L} = \sqrt{r}$

For each variable  $z$ , the derivative  $\partial\mathcal{L}/\partial z$  will get computed in reverse order.

- (4)  $\partial\mathcal{L}/\partial r = \frac{1}{2\sqrt{r}}$
- (3)  $\partial\mathcal{L}/\partial u = \partial\mathcal{L}/\partial r$
- (3)  $\partial\mathcal{L}/\partial w = \partial\mathcal{L}/\partial r$
- (2)  $\partial\mathcal{L}/\partial y = (2y) * (\partial\mathcal{L}/\partial w)$
- (1)  $\partial\mathcal{L}/\partial x = (2x) * (\partial\mathcal{L}/\partial u)$

## A More Abstract Example (Still Scalar Values)

$$y = f(x)$$

$$z = g(y, x)$$

$$u = h(z)$$

$$\mathcal{L} = u$$

For now assume all values are scalars (single numbers rather than arrays).

We will “backpropagate” the assignments the reverse order.

## Backpropagation (Scalar Values)

$$y = f(x)$$

$$z = g(y, x)$$

$$u = h(z)$$

$$\mathcal{L} = u$$

$$\partial \mathcal{L} / \partial u = 1$$

## Backpropagation (Scalar Values)

$$y = f(x)$$

$$z = g(y, x)$$

$$u = h(\textcolor{red}{z})$$

$$\mathcal{L} = u$$

$$\partial \mathcal{L} / \partial u = 1$$

$$\textcolor{red}{\partial \mathcal{L} / \partial z} = (\partial \mathcal{L} / \partial u)(\partial u / \partial z) \quad (\text{this uses the value of } z)$$

## Backpropagation (Scalar Values)

$$y = f(x)$$

$$z = g(\textcolor{red}{y}, x)$$

$$u = h(z)$$

$$\mathcal{L} = u$$

$$\partial \mathcal{L} / \partial u = 1$$

$$\textcolor{red}{\partial \mathcal{L} / \partial z} = (\partial \mathcal{L} / \partial u)(\partial u / \partial z) \text{ (this uses the value of } z\text{)}$$

$$\textcolor{red}{\partial \mathcal{L} / \partial y} = (\partial \mathcal{L} / \partial z)(\partial z / \partial \textcolor{red}{y}) \text{ (this uses the value of } y \text{ and } x\text{)}$$



## Backpropagation (Scalar Values)

$$y = f(\textcolor{red}{x})$$

$$z = g(y, \textcolor{red}{x})$$

$$u = h(z)$$

$$\mathcal{L} = u$$

$$\partial \mathcal{L} / \partial u = 1$$

$$\partial \mathcal{L} / \partial z = (\partial \mathcal{L} / \partial u)(\partial u / \partial z) \text{ (this uses the value of } z\text{)}$$

$$\partial \mathcal{L} / \partial y = (\partial \mathcal{L} / \partial z)(\partial z / \partial y) \text{ (this uses the value of } y \text{ and } x\text{)}$$

$$\partial \mathcal{L} / \partial x = ??? \text{ Oops, we need to add up multiple occurrences.}$$

## Backpropagation (Scalar Values)

$$y = f(\textcolor{red}{x})$$

$$z = g(y, \textcolor{red}{x})$$

$$u = h(z)$$

$$\mathcal{L} = u$$

Each framework program variable denotes an **object** (in the sense of C++ or Python).

**$x.value$**  and  **$x.grad$**  are attributes of the **object  $x$** .

Values are computed “forward” while gradients are computed “backward”.

## Backpropagation (Scalar Values)

$$y = f(x)$$

$$z = g(y, x)$$

$$u = h(z)$$

$$\mathcal{L} = u$$

$$z.\text{grad} = y.\text{grad} = x.\text{grad} = 0$$

$$u.\text{grad} = 1$$

**Invariant:** The gradients are correct for the red program.

## Backpropagation (Scalar Values)

$$y = f(x)$$

$$z = g(y, x)$$

$$u = h(z)$$

$$\mathcal{L} = u$$

$$z.\text{grad} = y.\text{grad} = x.\text{grad} = 0$$

$$u.\text{grad} = 1$$

$$z.\text{grad} += u.\text{grad} * (\partial u / \partial z)$$

**Invariant:** The gradients are correct for the red program.

## Backpropagation (Scalar Values)

$$y = f(x)$$

$$z = g(y, x)$$

$$u = h(z)$$

$$\mathcal{L} = u$$

$$z.\text{grad} = y.\text{grad} = x.\text{grad} = 0$$

$$u.\text{grad} = 1$$

$$z.\text{grad} += u.\text{grad} * (\partial u / \partial z)$$

$$y.\text{grad} += z.\text{grad} * (\partial z / \partial y)$$

$$x.\text{grad} += z.\text{grad} * (\partial z / \partial x)$$

## Backpropagation (Scalar Values)

$$y = f(x)$$

$$z = g(y, x)$$

$$u = h(z)$$

$$\mathcal{L} = u$$

$$z.\text{grad} = y.\text{grad} = x.\text{grad} = 0$$

$$u.\text{grad} = 1$$

$$z.\text{grad} += u.\text{grad} * (\partial u / \partial z)$$

$$y.\text{grad} += z.\text{grad} * (\partial z / \partial y)$$

$$x.\text{grad} += z.\text{grad} * (\partial z / \partial x)$$

$$x.\text{grad} += y.\text{grad} * (\partial y / \partial x)$$

**END**