TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 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 "backpopagate" 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)

$$y = f(x)$$
$$z = g(y, x)$$
$$u = h(\textcolor{red}{z})$$
$$\mathcal{L} = u$$

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

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

Backpropagation (Scalar Values)

$$y = f(x)$$
$$z = g(\textcolor{red}{y}, x)$$
$$u = h(z)$$
$$\mathcal{L} = u$$

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

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

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

Backpropagation (Scalar Values)

$$y = f(x)$$
$$z = g(y, x)$$
$$u = h(z)$$
$$\mathcal{L} = u$$

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

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

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

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

Backpropagation (Scalar Values)

$$y = f(x)$$
$$z = g(y, x)$$
$$u = h(z)$$
$$\mathcal{L} = u$$

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)

$$
\begin{aligned}
y &= f(x) \\
z &= g(y, x) \\
\textcolor{red}{u} &= \textcolor{red}{h(z)} \\
\textcolor{red}{\mathcal{L}} &= \textcolor{red}{u} \\
\textcolor{red}{z.\text{grad}} &= \textcolor{red}{y.\text{grad} = x.\text{grad} = 0} \\
\textcolor{red}{u.\text{grad}} &= \textcolor{red}{1} \\
\textcolor{red}{z.\text{grad}} \mathrel{\textcolor{red}{+\!\!=}} &\ \textcolor{red}{u.\text{grad} * (\partial u / \partial z)}
\end{aligned}
$$

Invariant: The gradients are correct for the red program.

Backpropagation (Scalar Values)

$$
\begin{aligned}
y &= f(x) \\
\color{red}{z} &\color{red}{= g(y, x)} \\
\color{red}{u} &\color{red}{= h(z)} \\
\color{red}{\mathcal{L}} &\color{red}{= u} \\
\color{red}{z.\text{grad}} &\color{red}{= y.\text{grad} = x.\text{grad} = 0} \\
\color{red}{u.\text{grad}} &\color{red}{= 1} \\
\color{red}{z.\text{grad}} &\color{red}{\mathrel{+}= u.\text{grad} * (\partial u / \partial z)} \\
\color{red}{y.\text{grad}} &\color{red}{\mathrel{+}= z.\text{grad} * (\partial z / \partial y)} \\
\color{red}{x.\text{grad}} &\color{red}{\mathrel{+}= z.\text{grad} * (\partial z / \partial x)}
\end{aligned}
$$

Backpropagation (Scalar Values)

$$y \;=\; f(x)$$

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

$$u \;=\; h(z)$$

$$\mathcal{L} \;=\; u$$

$$z.\mathrm{grad} \;=\; y.\mathrm{grad} = x.\mathrm{grad} = 0$$

$$u.\mathrm{grad} \;=\; 1$$

$$z.\mathrm{grad} \;\mathtt{+=}\; u.\mathrm{grad} * (\partial u / \partial z)$$

$$y.\mathrm{grad} \;\mathtt{+=}\; z.\mathrm{grad} * (\partial z / \partial y)$$

$$x.\mathrm{grad} \;\mathtt{+=}\; z.\mathrm{grad} * (\partial z / \partial x)$$

$$x.\mathrm{grad} \;\mathtt{+=}\; y.\mathrm{grad} * (\partial y / \partial x)$$

END