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

$$\begin{aligned} u &= x^2 \\ v &= y^2 \\ r &= u + v \\ \mathcal{L} &= \sqrt{r} \end{aligned}$$

Source Code

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

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

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

A More Abstract Example (Still Scalar Values)

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

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

We will "backpopagate" the assignments the reverse order.

Backpropagation (Scalar Values)

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

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

Backpropagation (Scalar Values)

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

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

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

Backpropagation (Scalar Values)

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

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

$\partial \mathcal{L} / \partial z = (\partial \mathcal{L} / \partial u)(\partial u / \partial z)$ (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)

Backpropagation (Scalar Values)

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

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

$\partial\mathcal{L}/\partial z = (\partial\mathcal{L}/\partial u)(\partial u/\partial z)$ (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)

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

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)

$$
\begin{aligned}
y &= f(x) \\
z &= g(y, x) \\
u &= h(z) \\
\mathcal{L} &= u \\
z.\mathrm{grad} &= y.\mathrm{grad} = x.\mathrm{grad} = 0 \\
u.\mathrm{grad} &= 1
\end{aligned}
$$

Invariant: The gradients are correct for the red program.

Backpropagation (Scalar Values)

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

Invariant: The gradients are correct for the red program.

Backpropagation (Scalar Values)

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

Backpropagation (Scalar Values)

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

$$w.\mathrm{grad} \text{ holds } \frac{\partial \mathcal{L}}{\partial w}$$

END