TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2020

Backpropagation with Arrays and Tensors

Handling Arrays

$$
\begin{aligned}
h &= \sigma\left(W^0 x - B^0\right) \\
s &= \sigma\left(W^1 h - B^1\right) \\
P_\Phi[\hat{y}] &= \operatorname*{softmax}_{\hat{y}}\ s[\hat{y}] \\
\mathcal{L} &= -\ln P[y]
\end{aligned}
$$

Each array W is an object with attributes W.value and W.grad.

W.grad is an array storing $\nabla_W \mathcal{L}$.

W.grad has same indeces (same "shape") as W.value.

Source Code Loops

$$s = \sigma\left(Wh - B\right)$$

Can be written as

$$\begin{array}{lrcl} \text{for } j & \tilde{h}[j] & = & 0 \\ \text{for } j, i & \tilde{h}[j] & \mathrel{+}= & W[j,i]x[i] \\ \text{for } j & s[j] & = & \sigma(\tilde{h}[j] - B[j]) \end{array}$$

Backpropagation on Loops

the backpropagation for

$$\text{for } j \;\; h[j] = \sigma(\tilde{h}[j] - B[j])$$

is

$$\text{for } j \;\; \tilde{h}.\text{grad}[j] \mathrel{\texttt{+=}} h.\text{grad}[j]\sigma'(h[j] - B[j])$$

$$\text{for } j \;\; B.\text{grad}[j] \mathrel{\texttt{-=}} h.\text{grad}[j]\sigma'(h[j] - B[j])$$

Backpropagation on Loops

the backpropagation for

$$\text{for } j, i \;\; \tilde{h}[j] \;\; \texttt{+=} \;\; W[j,i]x[i]$$

is

$$\text{for } j, i \;\; W.\mathrm{grad}[j,i] \;\; \texttt{+=} \;\; \tilde{h}.\mathrm{grad}[j]x[i]$$

$$x.\mathrm{grad}[i] \;\; \texttt{+=} \;\; \tilde{h}.\mathrm{grad}[j]W[j,i]$$

General Tensor Operations

In practice all deep learning source code can be written as a series of scalar assignments and loops where the body of each loop operates on scalars.

Scalar backpropagation can then be applied to the loops.

$$\text{for} \cdots \quad Y[\cdots] \mathrel{+}= e(A[\cdots], B[\cdots])$$

has backpropagation loops

$$\begin{array}{l} \text{for } \cdots \; A.\text{grad}[\cdots] \mathrel{+}= Y.\text{grad}[\cdots](\partial e/\partial A[\cdots]) \\ \text{for } \cdots \; B.\text{grad}[\cdots] \mathrel{+}= Y.\text{grad}[\cdots](\partial e/\partial B[\cdots]) \end{array}$$

END