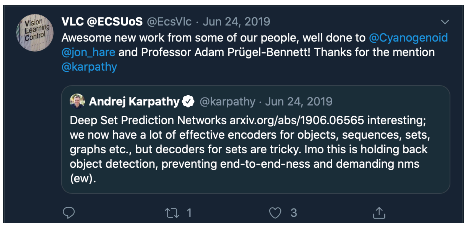
Make a forward pass before the backward pass

Backpropagation: Understanding the implications of the chain rule

Jonathon Hare

Vision, Learning and Control
University of Southampton

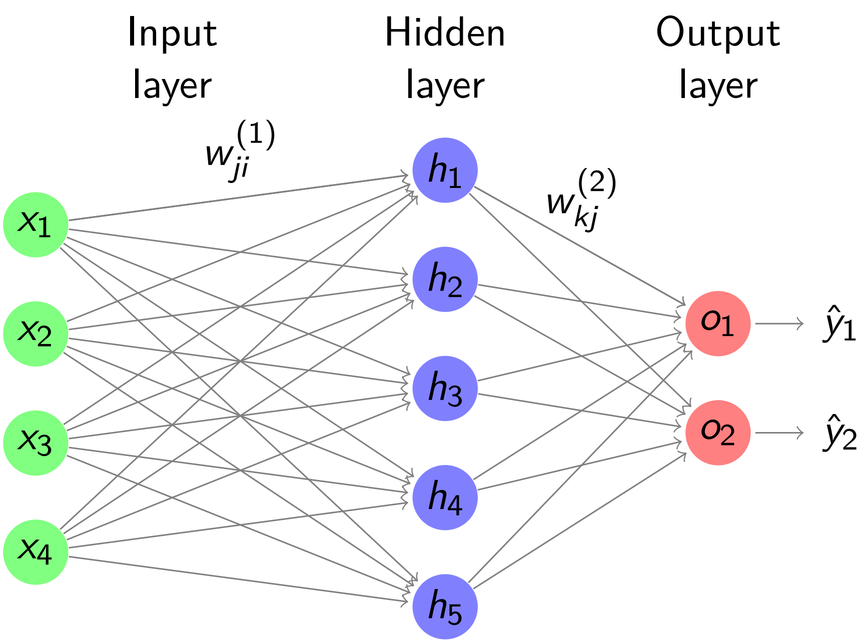
A lot of the ideas in this lecture come from Andrej Karpathy's blog post on backprop (`https://medium.com/@karpathy/yes-you-should-understand-backprop-e2f06eab496b`) and his CS231n Lecture Notes (`http://cs231n.github.io/optimization-2/`)



Topics

- A quick look at an MLP again
- The chain rule (again)
- A closer look at basic stochastic gradient descent algorithms

The unbiased Multilayer Perceptron (again)...



Without loss of generality, we can write the above as:

$$\hat{\boldsymbol{y}} = g(f(\boldsymbol{x}; \boldsymbol{W}^{(1)}); \boldsymbol{W}^{(2)}) = g(\boldsymbol{W}^{(2)} f(\boldsymbol{W}^{(1)} \boldsymbol{x}))$$

where f and g are activation functions.

Gradients of our simple unbiased MLP

- Let's assume MSE Loss

$$\ell_{MSE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \|\hat{\boldsymbol{y}} - \boldsymbol{y}\|_2^2$$

- What are the gradients?

$$\nabla_{\boldsymbol{W}^*} \ell_{MSE}(g(\boldsymbol{W}^{(2)} f(\boldsymbol{W}^{(1)} \boldsymbol{x})), \boldsymbol{y})$$

- Clearly we need to apply the chain rule (vector form) multiple times
- We could do this by hand
- (But we're not that crazy!)

Let's go back to a simpler expression

$$\begin{aligned} f(x, y, z) &= (x + y)z \\ &\equiv qz \text{ where } q = (x + y) \end{aligned}$$

Clearly the partial derivatives of the subexpressions are trivial:

$$\begin{aligned} \partial f/\partial z &= q \qquad & \partial f/\partial q &= z \\ \partial q/\partial x &= 1 \qquad & \partial q/\partial y &= 1 \end{aligned}$$

and the chain rule tells us how to combine these:

$$\begin{aligned} \partial f/\partial x &= \partial f/\partial q \cdot \partial q/\partial x = z \\ \partial f/\partial y &= \partial f/\partial q \cdot \partial q/\partial y = z \end{aligned}$$

so $\nabla_{[x,y,z]} f = [z, z, q]$

A computational graph