Neural networks. Backpropagation.

November 25, 2016

Agenda

- History
- Deep learning success
- Artificial neuron
- Neural networks
  - Architecture
  - Parameters
- Backpropagation

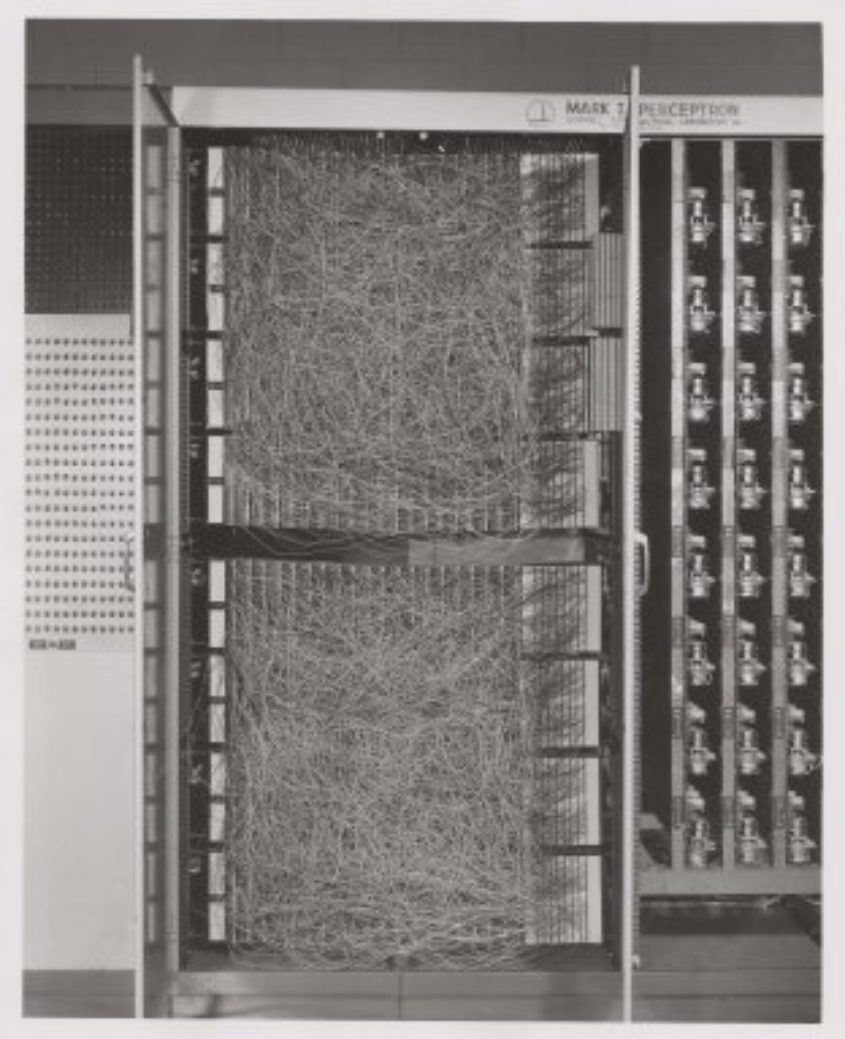
Credits

- cs231n - Fei-Fei Li, Andrej Karpathy, Justin Johnson (lectures 4, 5)
- A Practical Introduction to Deep Learning with Caffe and Python - Adil Moujahid
- `http://neuralnetworksanddeeplearning.com` - Michael Nielsen

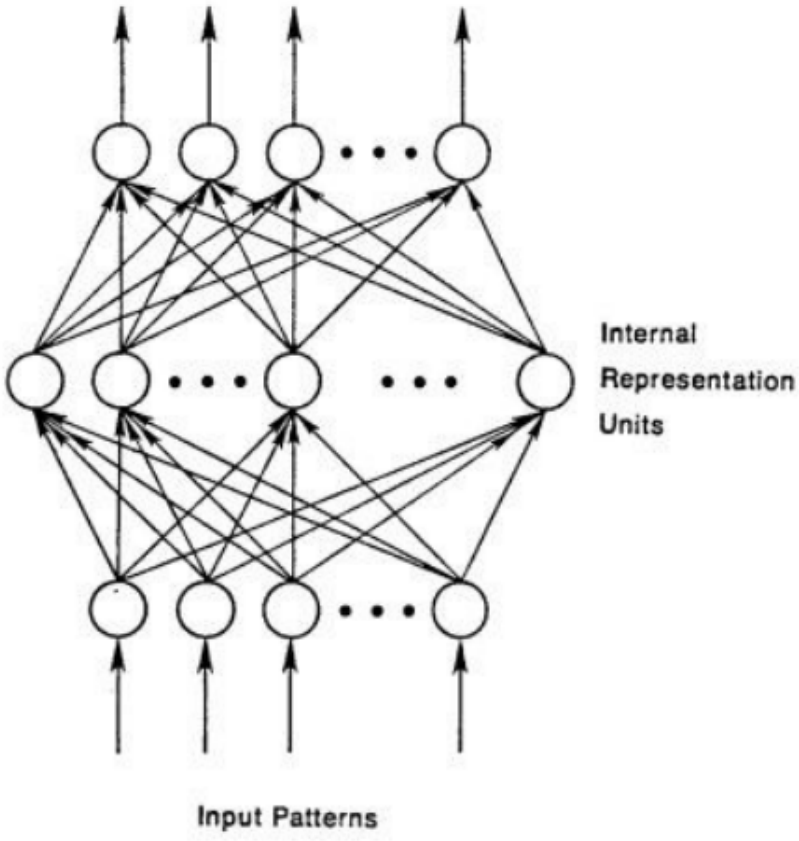
Quick history. 1957 - Rosenblatt - Perceptron

This machine was designed for image recognition.
The Mark I Perceptron machine.
The machine was connected to a camera that used 20x20 cadmium sulfide photocells to produce a 400-pixel image. The main visible feature is a patchboard that allowed experimentation with different combinations of input features. To the right of that are arrays of potentiometers that implemented the adaptive weights.

$$f(x) = \begin{cases} 1, & \text{if } w * x + b > 0 \\ 0, & \text{otherwise} \end{cases}$$



Quick history. 1986 - Rumelhart, Hinton, Williams - backpropagation.



To be more specific, then, let

$$E_p = \frac{1}{2}\sum_j (t_{pj} - o_{pj})^2 \qquad (2)$$

be our measure of the error on input/output pattern p and let $E = \sum E_p$ *be our overall measure of the error. We wish to show that the delta rule implements a gradient descent in E when the units are linear. We will proceed by simply showing that*

$$-\frac{\partial E_p}{\partial w_{ji}} = \delta_{pj} i_{pi},$$

which is proportional to $\Delta_p w_{ji}$ *as prescribed by the delta rule. When there are no hidden units it is straightforward to compute the relevant derivative. For this purpose we use the chain rule to write the derivative as the product of two parts: the derivative of the error with respect to the output of the unit times the derivative of the output with respect to the weight.*

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial w_{ji}}. \qquad (3)$$

The first part tells how the error changes with the output of the j*th unit and the second part tells how much changing* w_{ji} *changes that output. Now, the derivatives are easy to compute. First, from Equation 2*

$$\frac{\partial E_p}{\partial o_{pj}} = -(t_{pj} - o_{pj}) = -\delta_{pj}. \qquad (4)$$

Not surprisingly, the contribution of unit u_j *to the error is simply proportional to* δ_{pj}*. Moreover, since we have linear units,*

$$o_{pj} = \sum_i w_{ji} i_{pi}, \qquad (5)$$

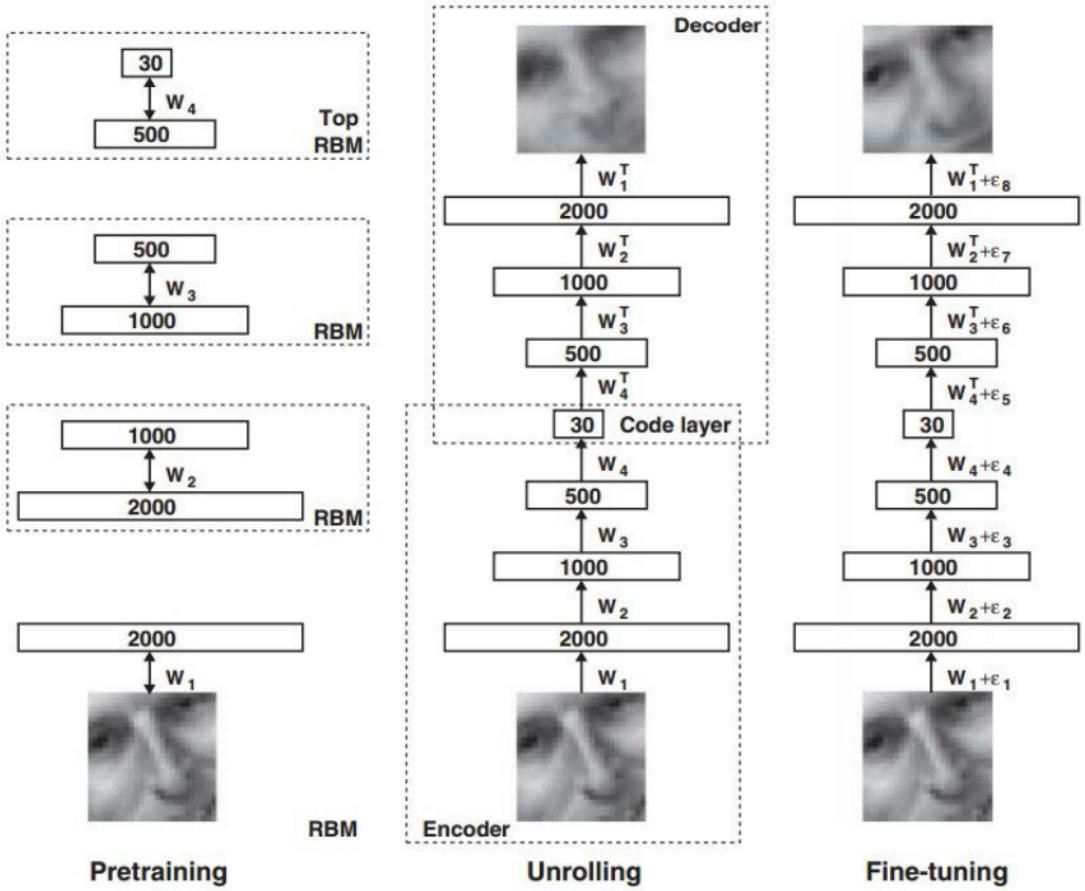
from which we conclude that

$$\frac{\partial o_{pj}}{\partial w_{ji}} = i_{pi}.$$

Thus, substituting back into Equation 3, we see that

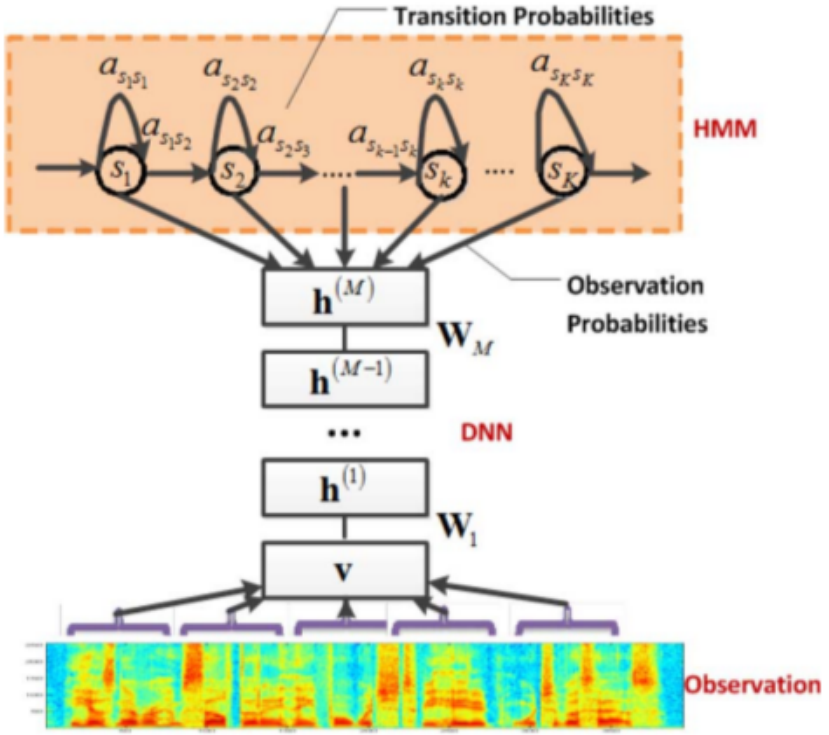
$$-\frac{\partial E_p}{\partial w_{ji}} = \delta_{pj} i_i \qquad (6)$$

Quick history. 2006 - Hinton, Salakhutdinov - Stacked RBM. Deep learning

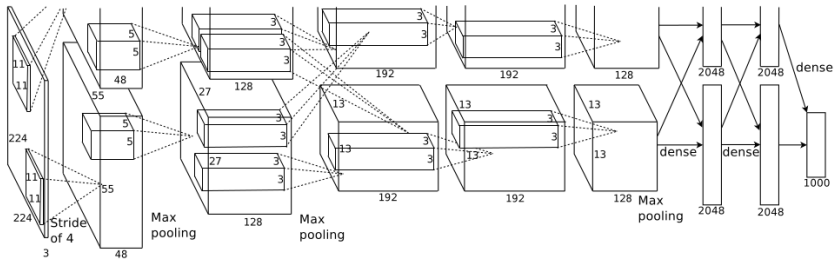


First strong results

Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition
George E. Dahl, Dong Yu, Li Deng, Fellow, Alex Acero, 2010



ImageNet Classification with Deep Convolutional Neural Networks
Krizhevsky, A., Sutskever, I. and Hinton, G. E., 2012



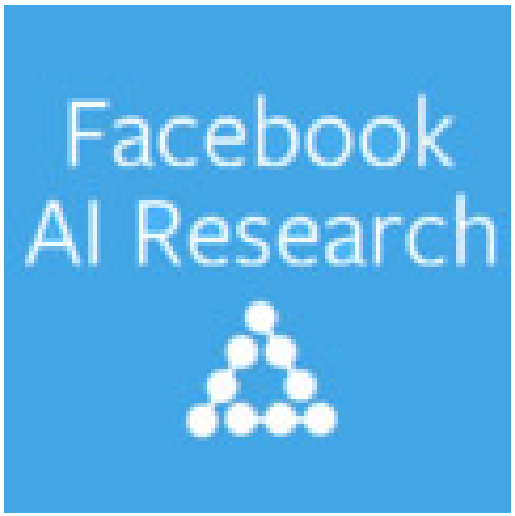
Neural networks state of the art performance

- Computer vision
  - Classification
  - Object localization
  - Object detection
- Speech recognition
- Machine translation

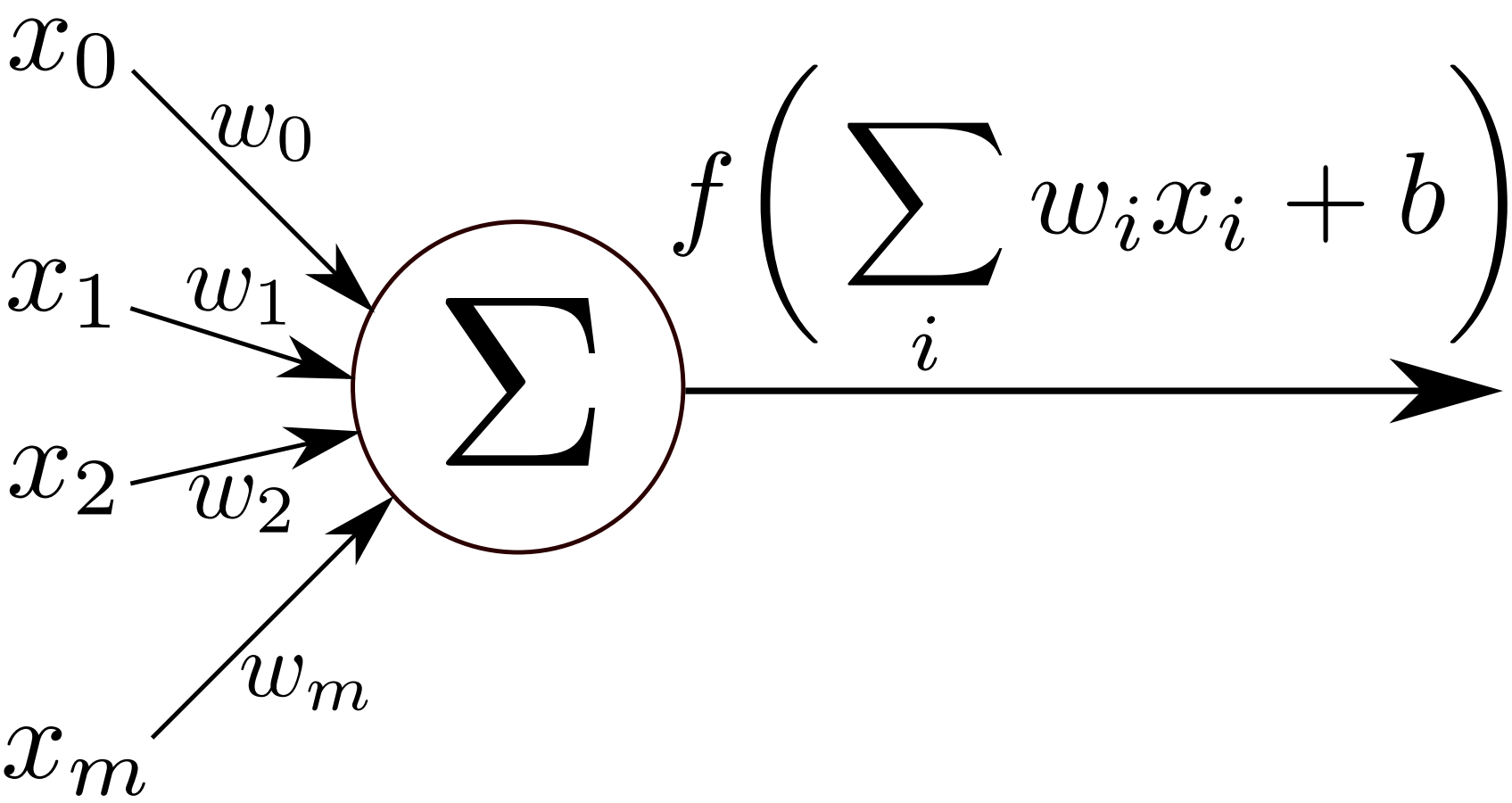
Deep learning success

- Data
- Computational power (GPU)
- Algorithms

Deep learning success

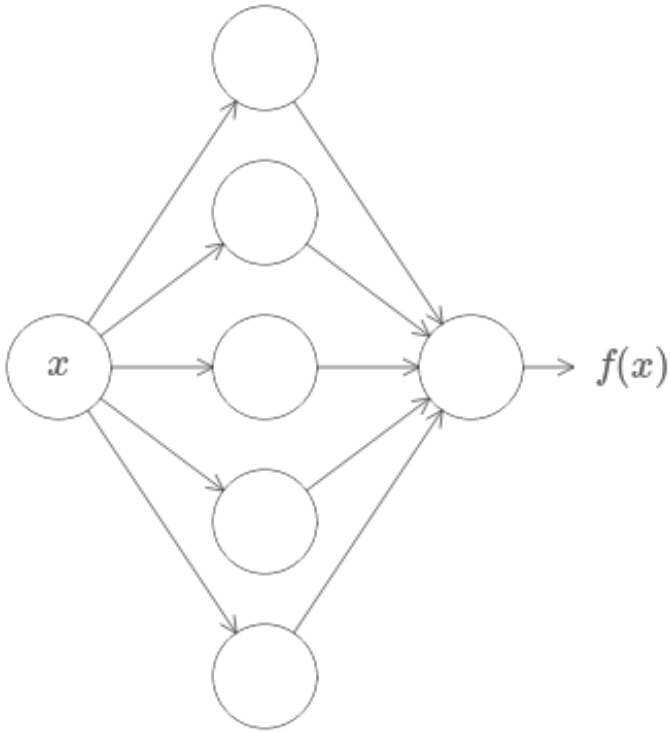


Artificial neuron



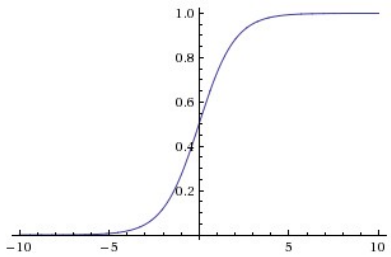
Universal function approximator

George Cybenko - 1989 for sigmoid activation functions.
Universality theorem tells that neural networks with a single hidden layer (with a sigmoid neuron) can be used to approximate any continuous function to any desired precision.
`http://neuralnetworksanddeeplearning.com/chap4.html`



Nonlinearities

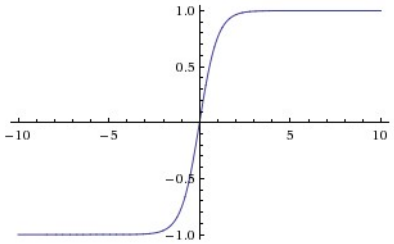
sigmoid



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

---

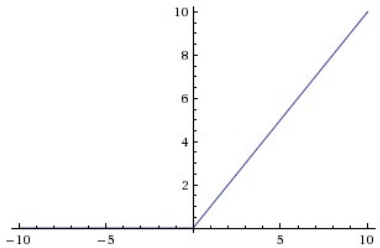
tanh



$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

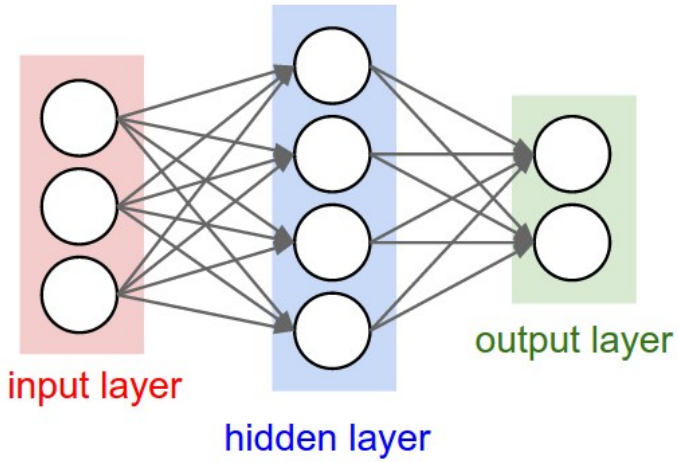
---

relu

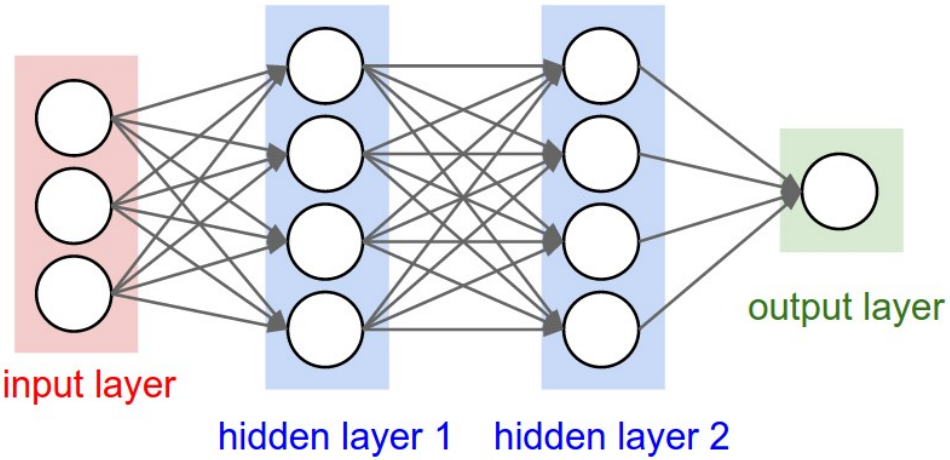


$$f(x) = \max(0, x)$$

Architecture



A 2-layer Neural Network (one hidden layer of 4 neurons (or units) and one output layer with 2 neurons), and three inputs.

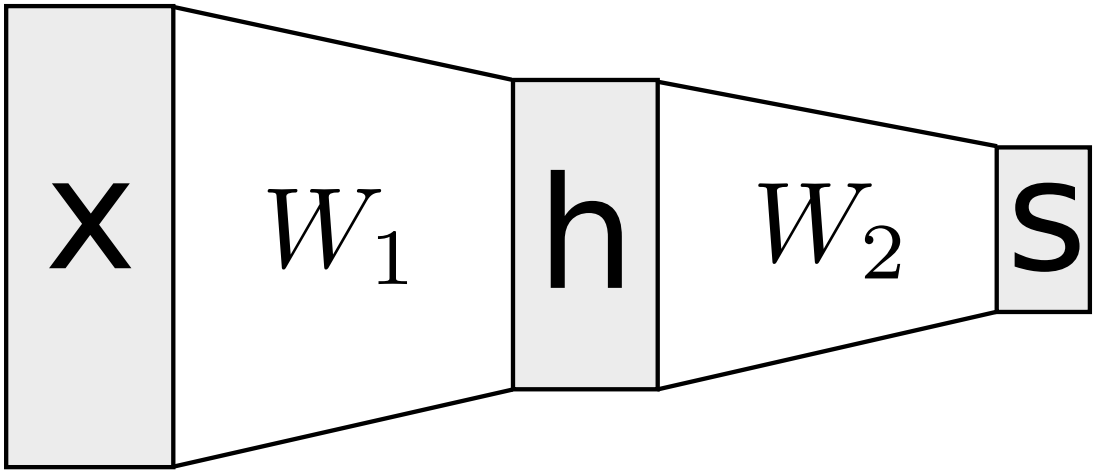


A 3-layer Neural Network

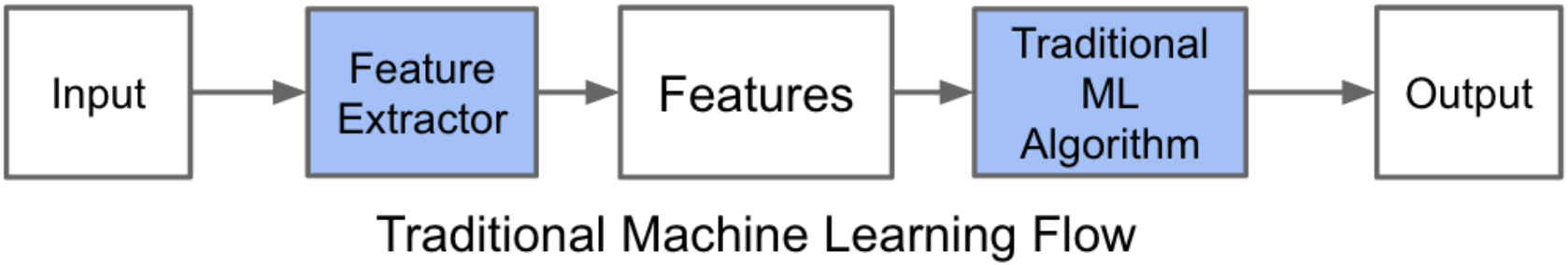
Neural networks

(**Before**) Linear regression $f = Wx$

(**Now**) 2-layer neural network $f = W_2 max(0, W_1 x)$



Neural networks (deep learning) vs "classic" approach



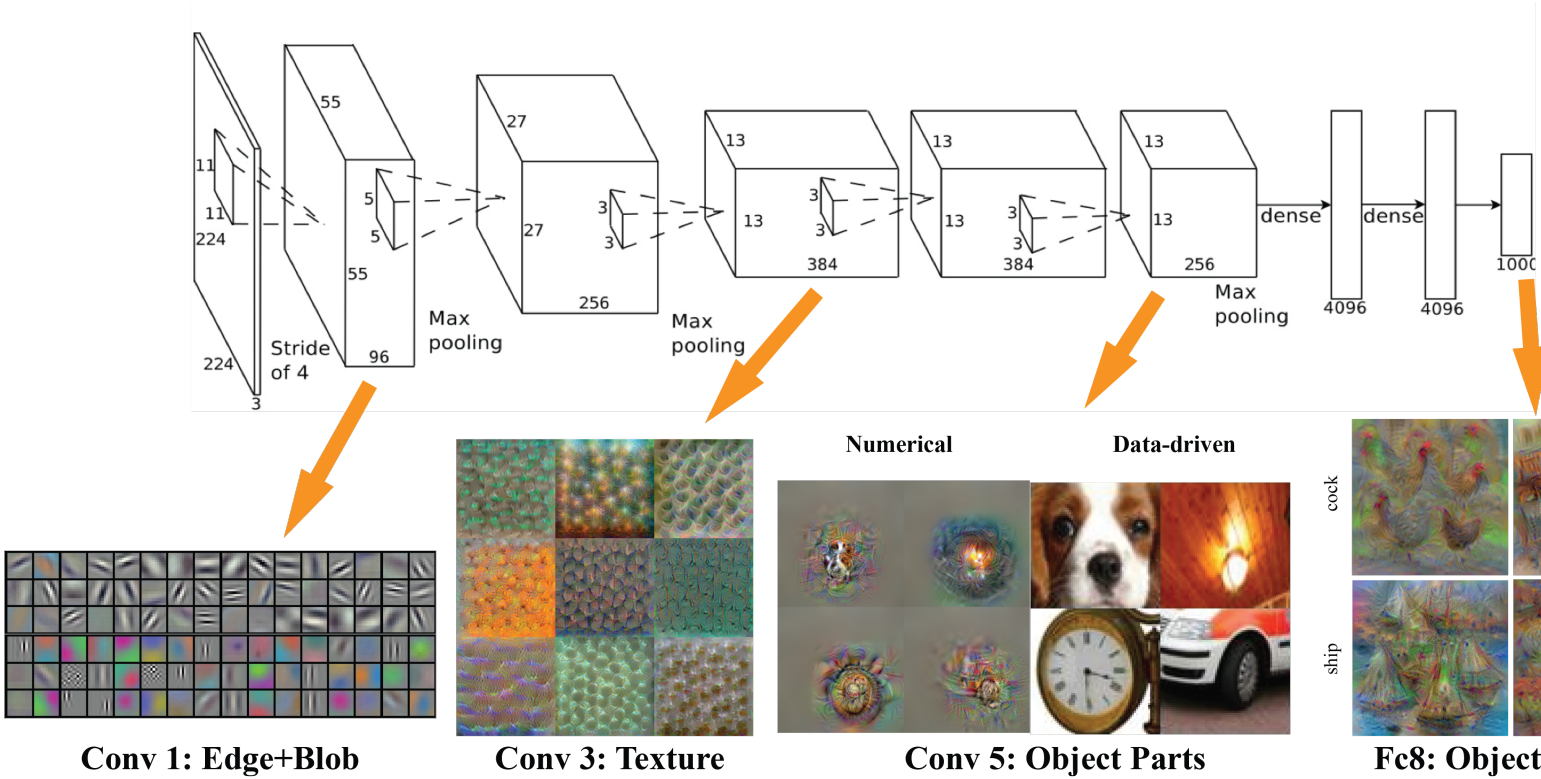
Traditional Machine Learning Flow

Deep Learning Flow

Image by Adil Moujahid (A Practical Introduction to Deep Learning with Caffe and Python)

Hierarchical representation of data

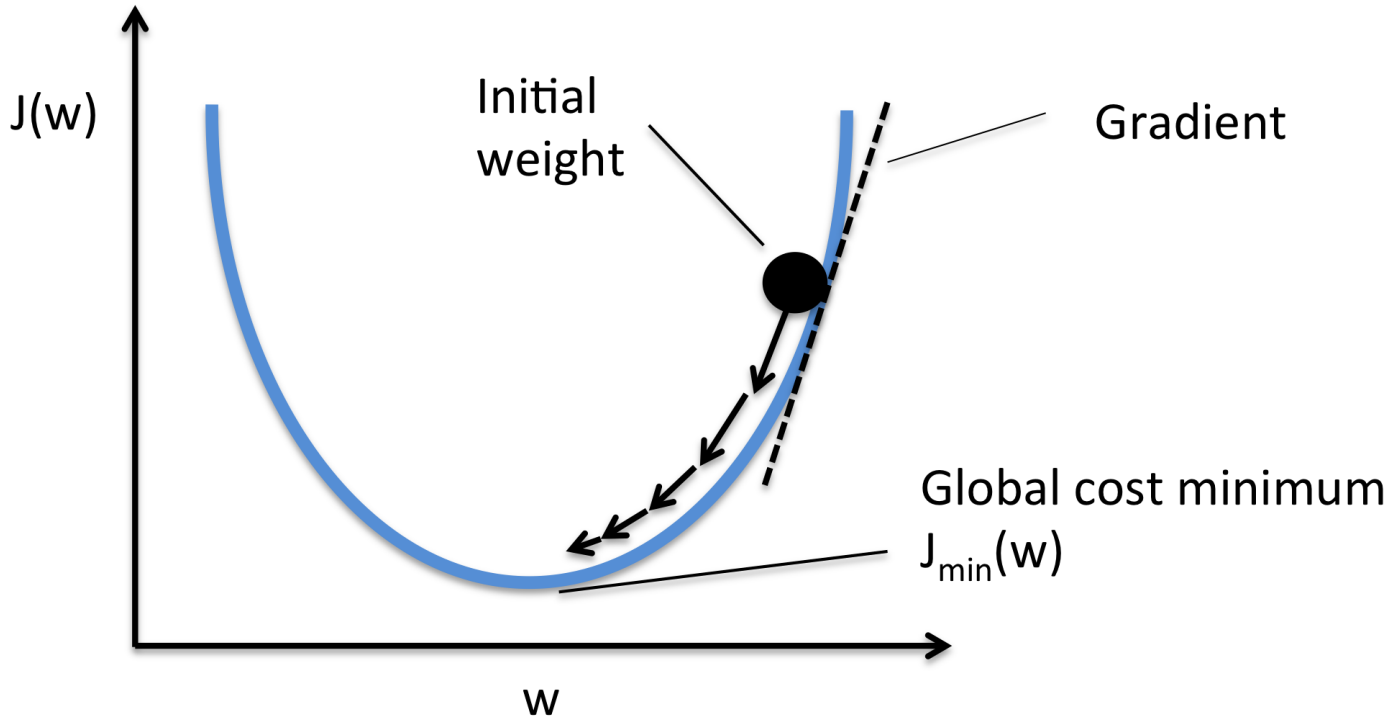
Each layer adds additional transformation.



http://vision03.csail.mit.edu/cnn_art/index.html

Gradient descent

$J(w)$ - cost (loss, error) function.



Gradient descent

Gradient descent algorithm

- Choose an initial vector of parameters w and learning rate η.
- Repeat until convergence
  $w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(w)$

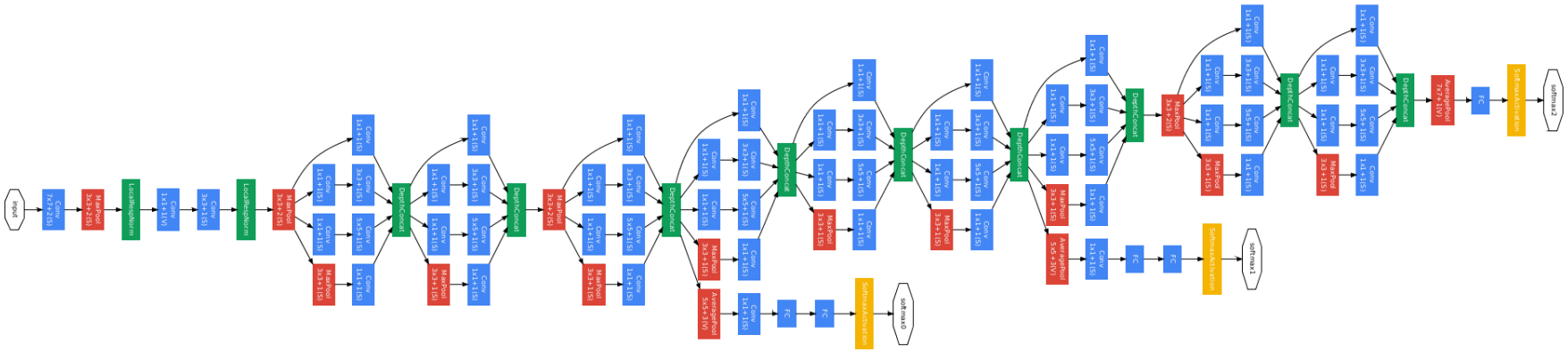
Gradient descent

$$\frac{\partial f}{\partial x} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Numerical gradient: slow :(, approximate :(, easy to write :)
Analytic gradient: fast :), exact :), error-prone :(

In practice: Derive analytic gradient, check implementation with numerical gradient.

GoogleNet



Backprop

$f(x, y, z) = (x + y) * z$
Want: $\nabla f(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z})$

$$q = x + y \implies \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \implies \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Chain rule: $\frac{\partial f(q,z)}{\partial x} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial x}$

Chain rule: $\frac{\partial f(q,z)}{\partial y} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial y}$

Let $x = -2$, $y = 5$, $z = -4$
Then $\frac{\partial f}{\partial x} = z * 1 = -4$,
$\frac{\partial f}{\partial y} = z * 1 = -4$,
$\frac{\partial f}{\partial z} = q = x + y = -2 + 5 = 3$

Backprop

$f(x, y, z) = (x + y) * z$

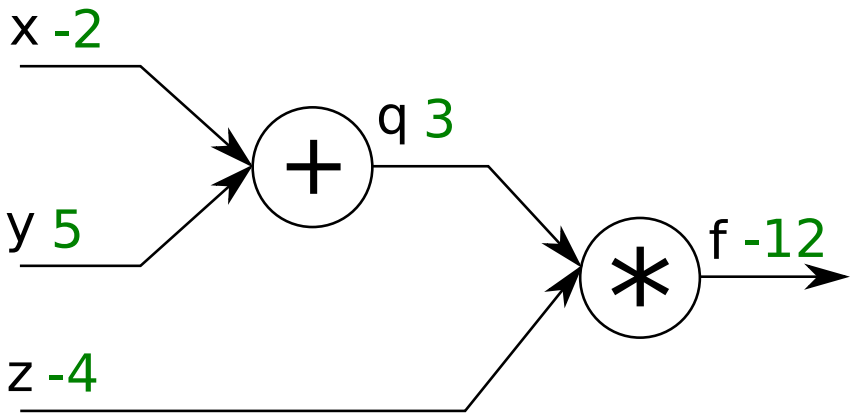
Want: $\nabla f(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z})$

$$q = x + y \implies \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \implies \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Chain rule: $\frac{\partial f(q,z)}{\partial x} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial x}$

Chain rule: $\frac{\partial f(q,z)}{\partial y} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial y}$



Backprop

$f(x, y, z) = (x + y) * z$

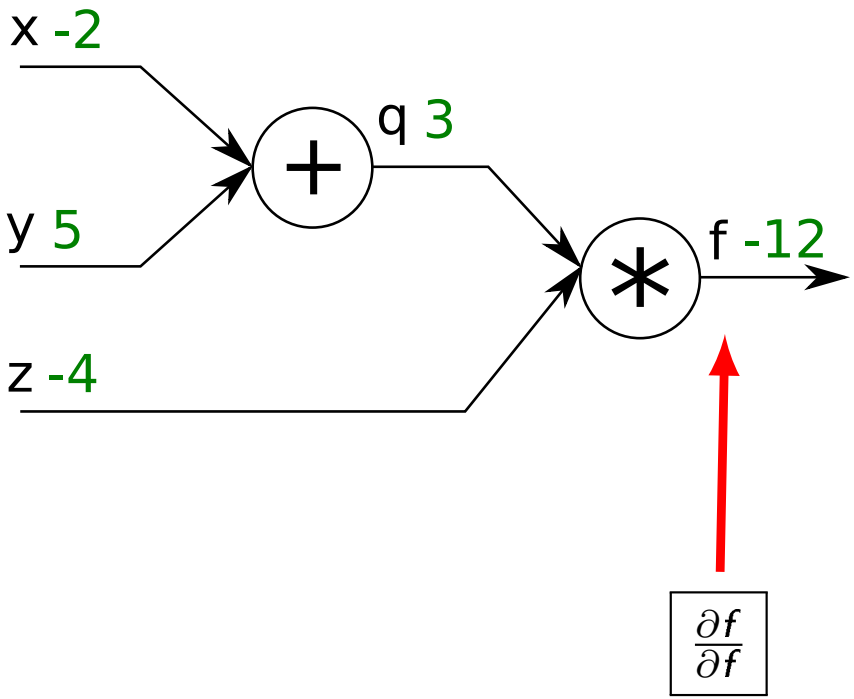
Want: $\nabla f(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z})$

$$q = x + y \implies \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \implies \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Chain rule: $\frac{\partial f(q,z)}{\partial x} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial x}$

Chain rule: $\frac{\partial f(q,z)}{\partial y} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial y}$



Backprop

$f(x, y, z) = (x + y) * z$

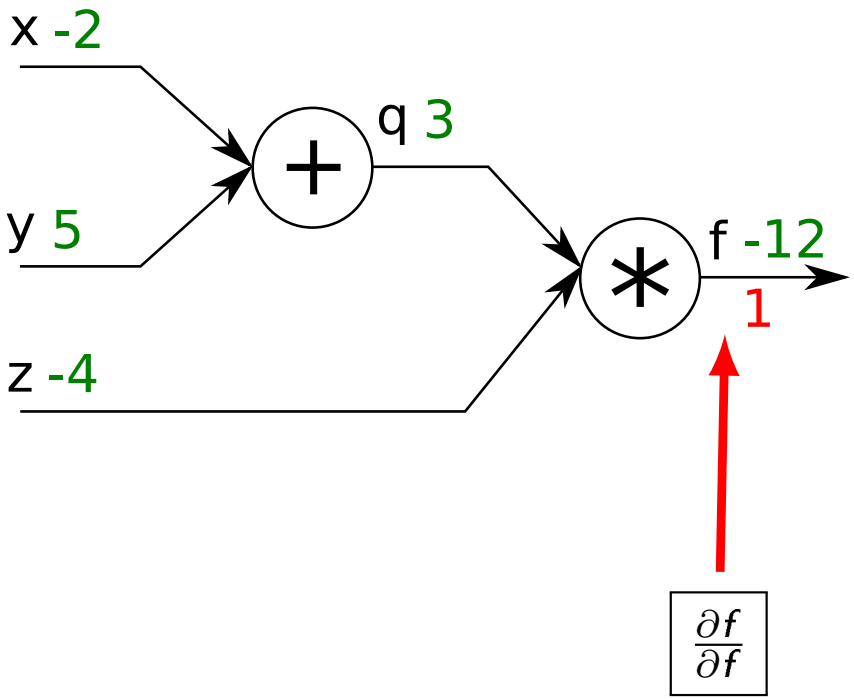
Want: $\nabla f(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z})$

$$q = x + y \implies \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \implies \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Chain rule: $\frac{\partial f(q,z)}{\partial x} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial x}$

Chain rule: $\frac{\partial f(q,z)}{\partial y} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial y}$



Backprop

$f(x, y, z) = (x + y) * z$

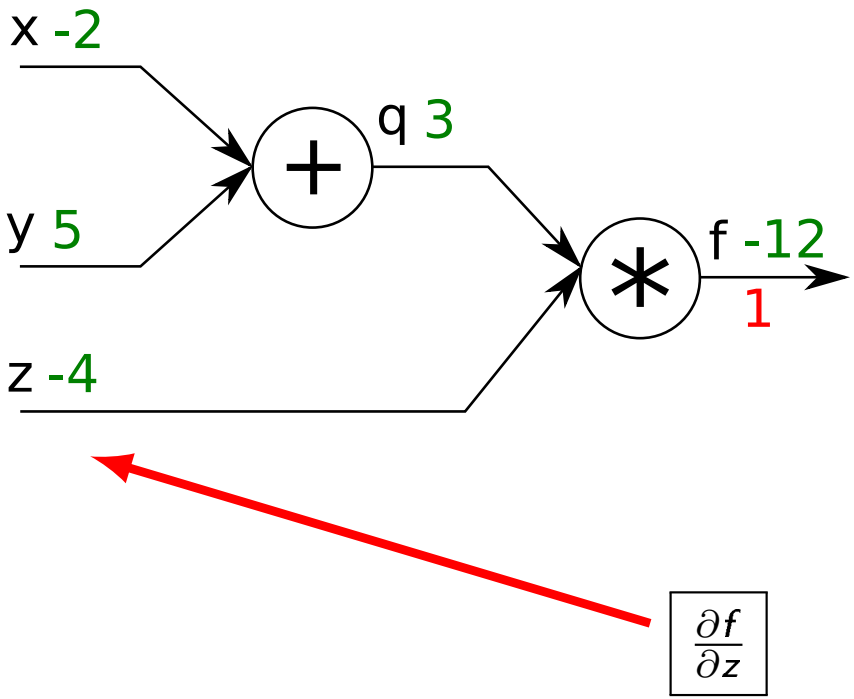
Want: $\nabla f(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z})$

$$q = x + y \implies \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \implies \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Chain rule: $\frac{\partial f(q,z)}{\partial x} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial x}$

Chain rule: $\frac{\partial f(q,z)}{\partial y} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial y}$



Backprop

$f(x, y, z) = (x + y) * z$

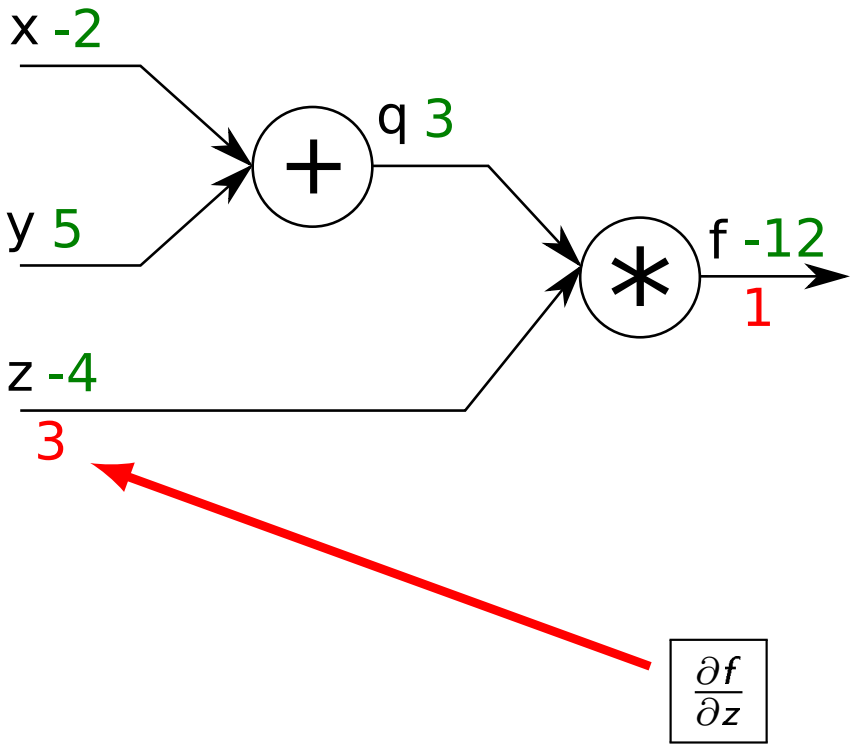
Want: $\nabla f(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z})$

$$q = x + y \implies \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \implies \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Chain rule: $\frac{\partial f(q,z)}{\partial x} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial x}$

Chain rule: $\frac{\partial f(q,z)}{\partial y} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial y}$



Backprop

$f(x, y, z) = (x + y) * z$

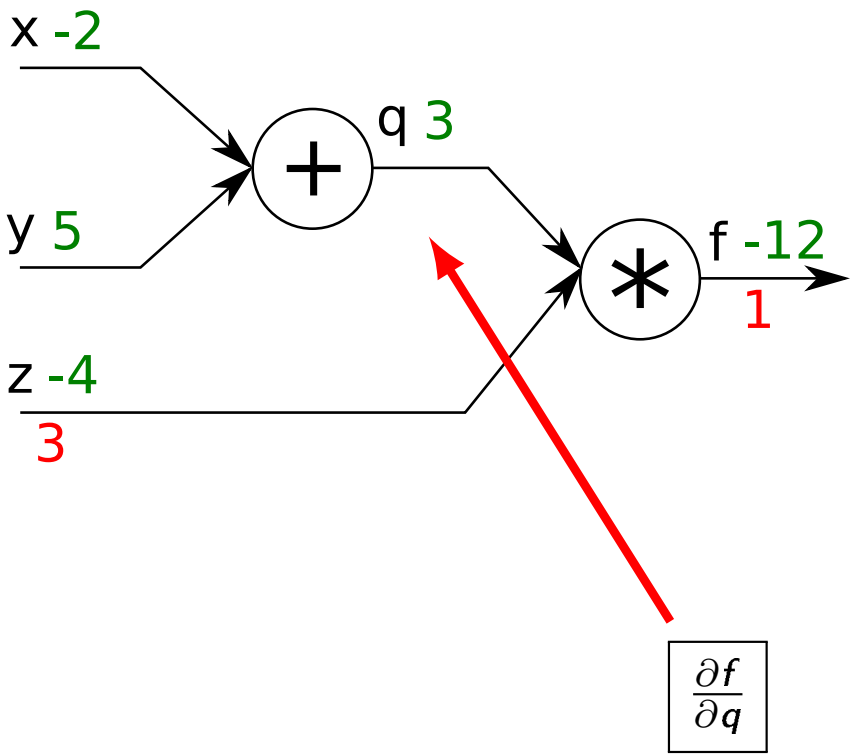
Want: $\nabla f(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z})$

$$q = x + y \implies \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \implies \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Chain rule: $\frac{\partial f(q,z)}{\partial x} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial x}$

Chain rule: $\frac{\partial f(q,z)}{\partial y} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial y}$



Backprop

$f(x, y, z) = (x + y) * z$

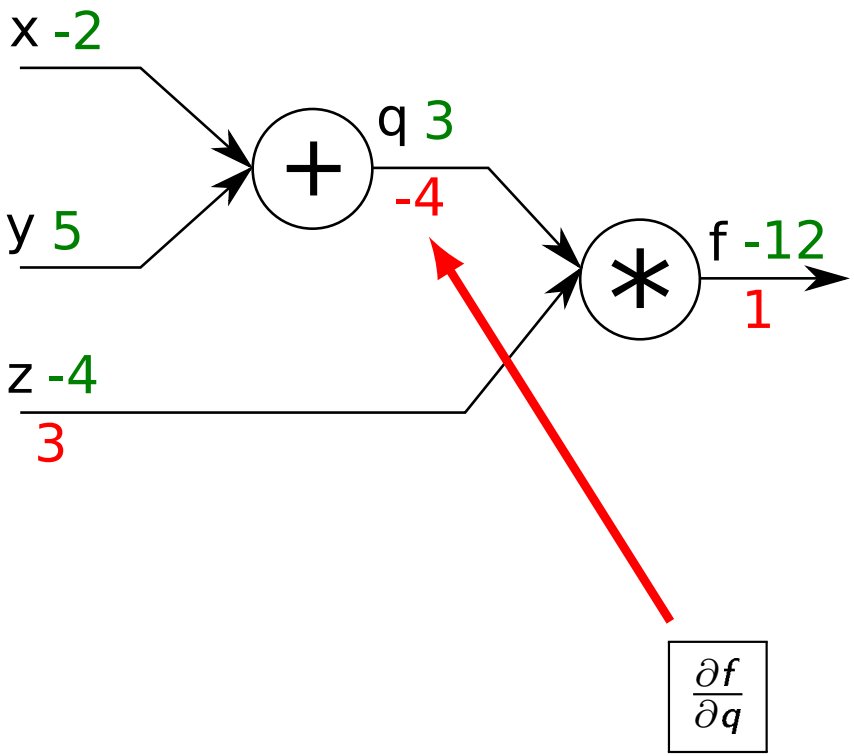
Want: $\nabla f(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z})$

$$q = x + y \implies \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \implies \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Chain rule: $\frac{\partial f(q,z)}{\partial x} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial x}$

Chain rule: $\frac{\partial f(q,z)}{\partial y} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial y}$



Backprop

$f(x, y, z) = (x + y) * z$

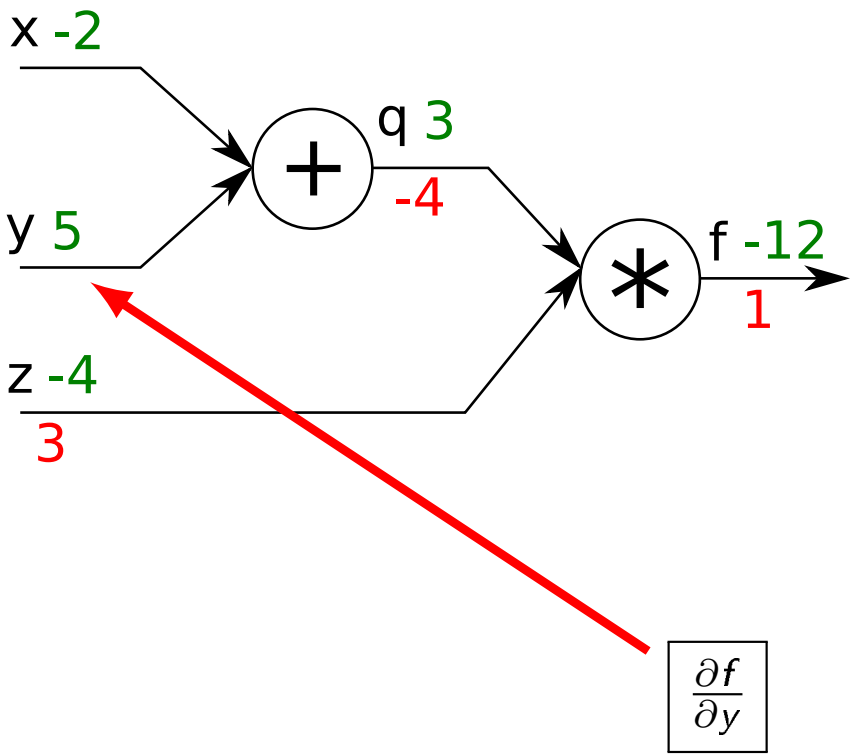
Want: $\nabla f(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z})$

$$q = x + y \implies \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \implies \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Chain rule: $\frac{\partial f(q,z)}{\partial x} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial x}$

Chain rule: $\frac{\partial f(q,z)}{\partial y} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial y}$



Backprop

$f(x, y, z) = (x + y) * z$

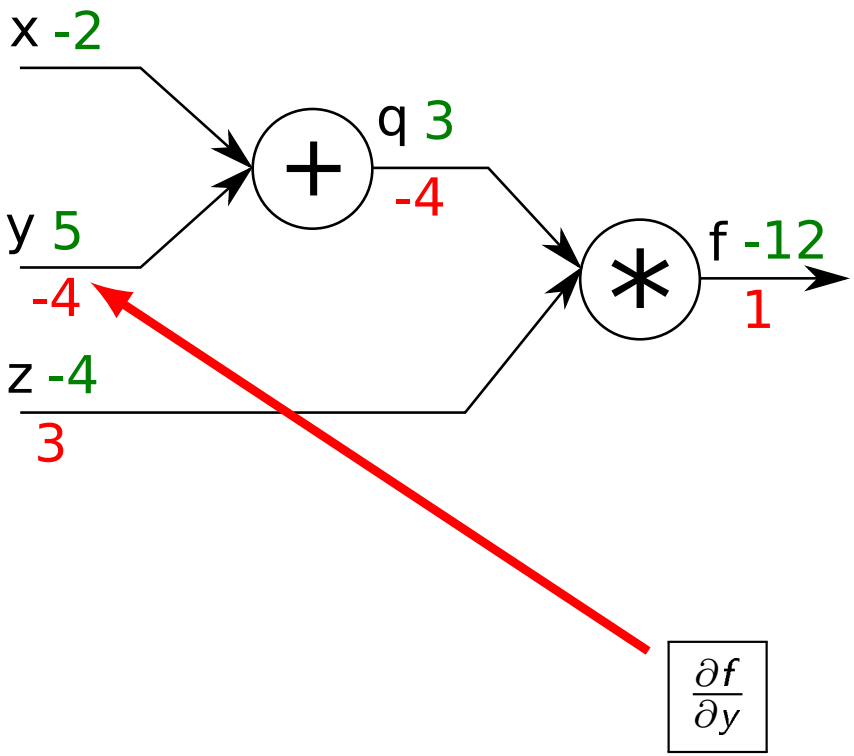
Want: $\nabla f(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z})$

$$q = x + y \implies \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \implies \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Chain rule: $\frac{\partial f(q,z)}{\partial x} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial x}$

Chain rule: $\frac{\partial f(q,z)}{\partial y} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial y}$



Backprop

$f(x, y, z) = (x + y) * z$

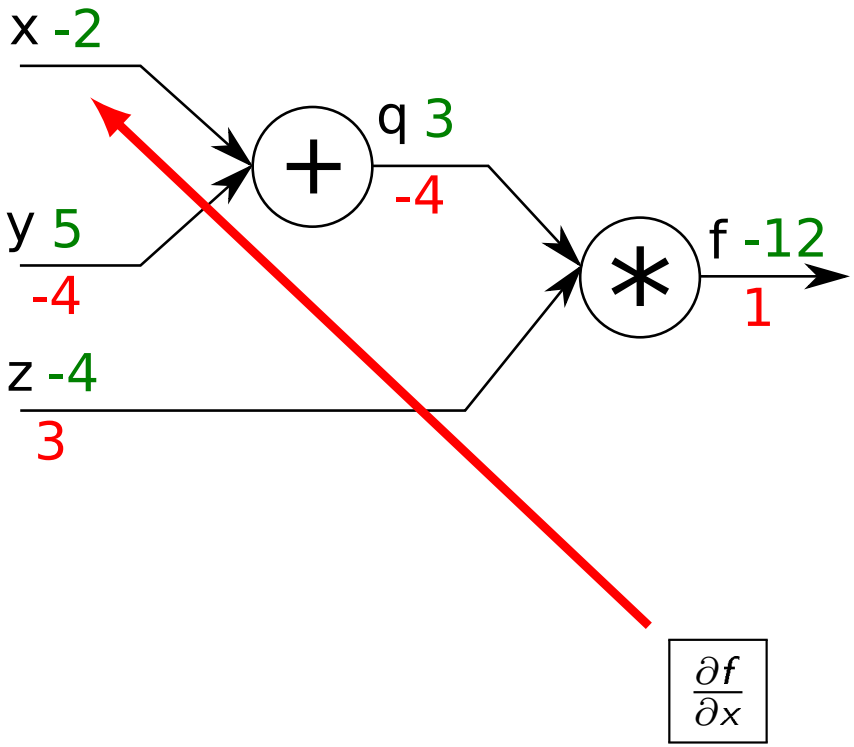
Want: $\nabla f(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z})$

$$q = x + y \implies \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \implies \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Chain rule: $\frac{\partial f(q,z)}{\partial x} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial x}$

Chain rule: $\frac{\partial f(q,z)}{\partial y} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial y}$



Backprop

$f(x, y, z) = (x + y) * z$

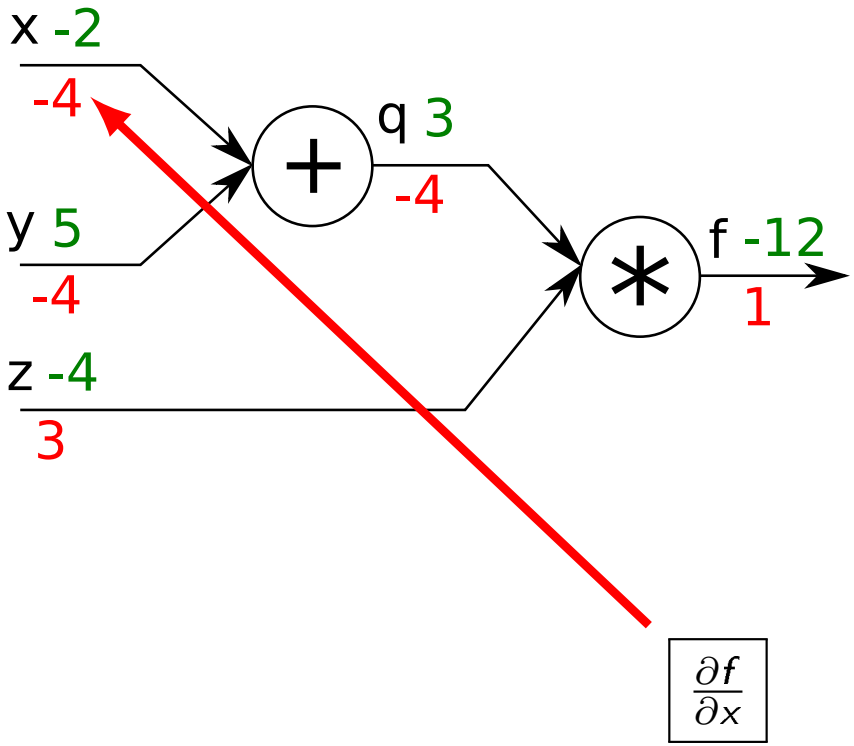
Want: $\nabla f(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z})$

$$q = x + y \implies \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

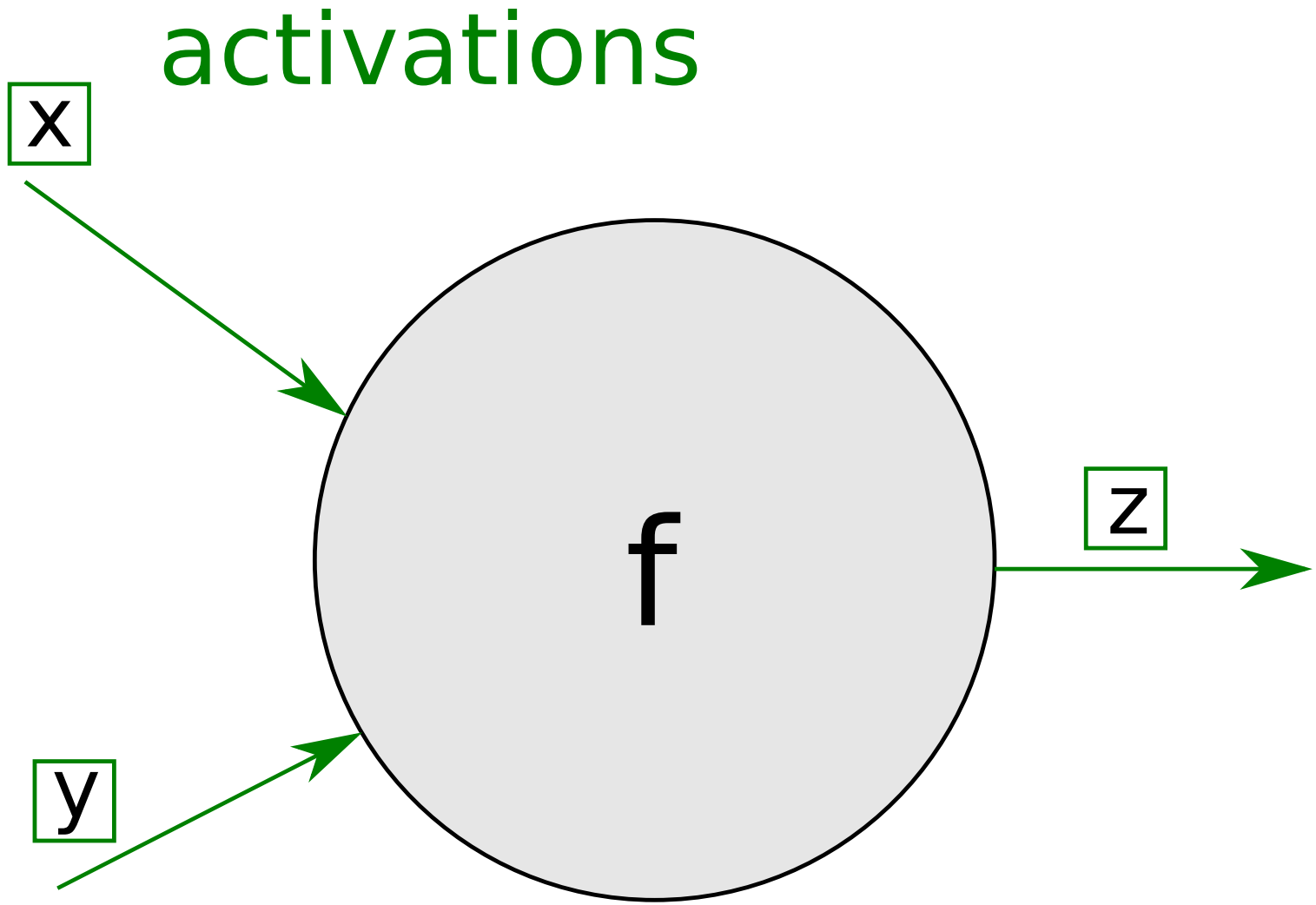
$$f = qz \implies \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Chain rule: $\frac{\partial f(q,z)}{\partial x} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial x}$

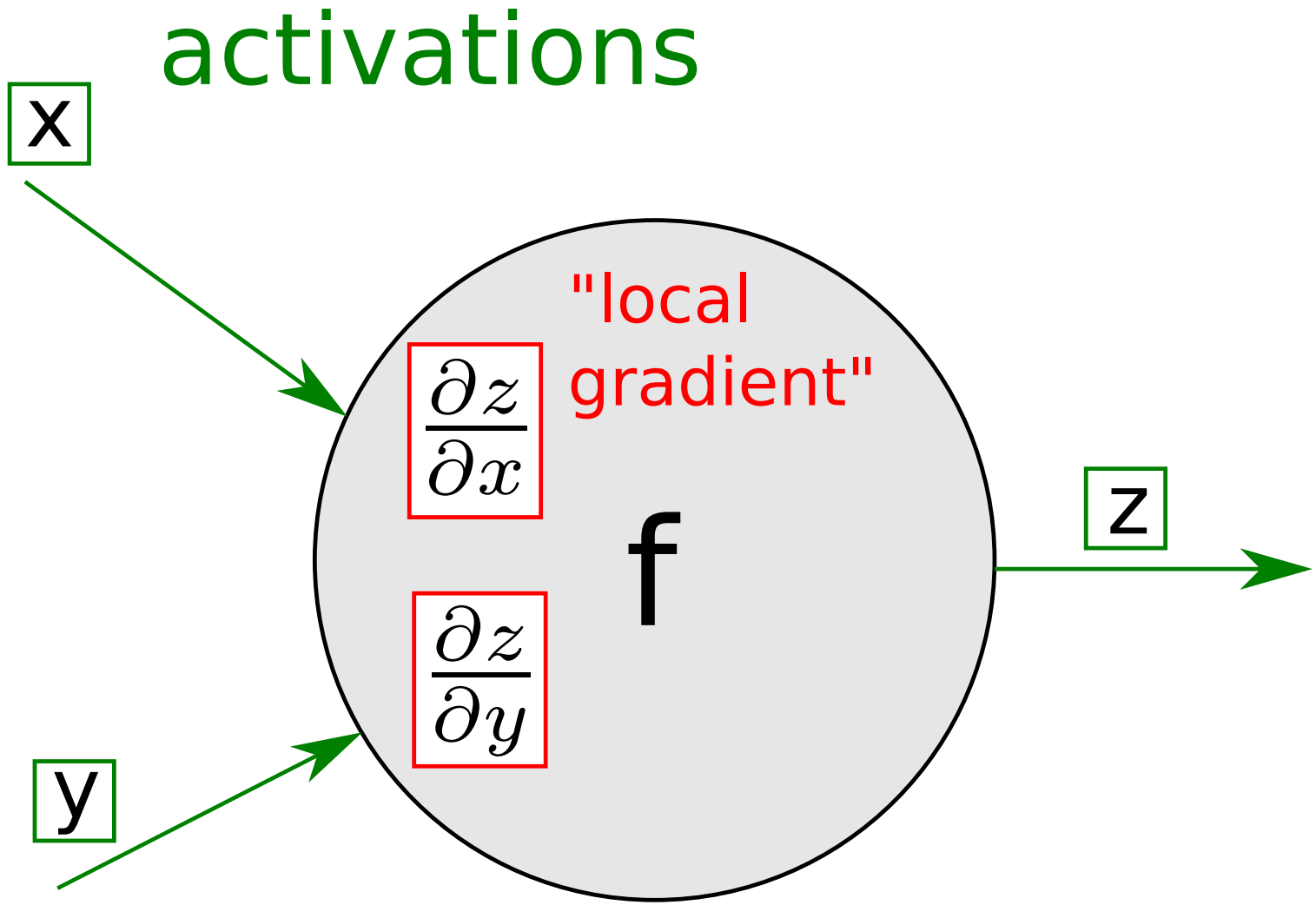
Chain rule: $\frac{\partial f(q,z)}{\partial y} = \frac{\partial f(q,z)}{\partial q} \frac{\partial q(x,y)}{\partial y}$



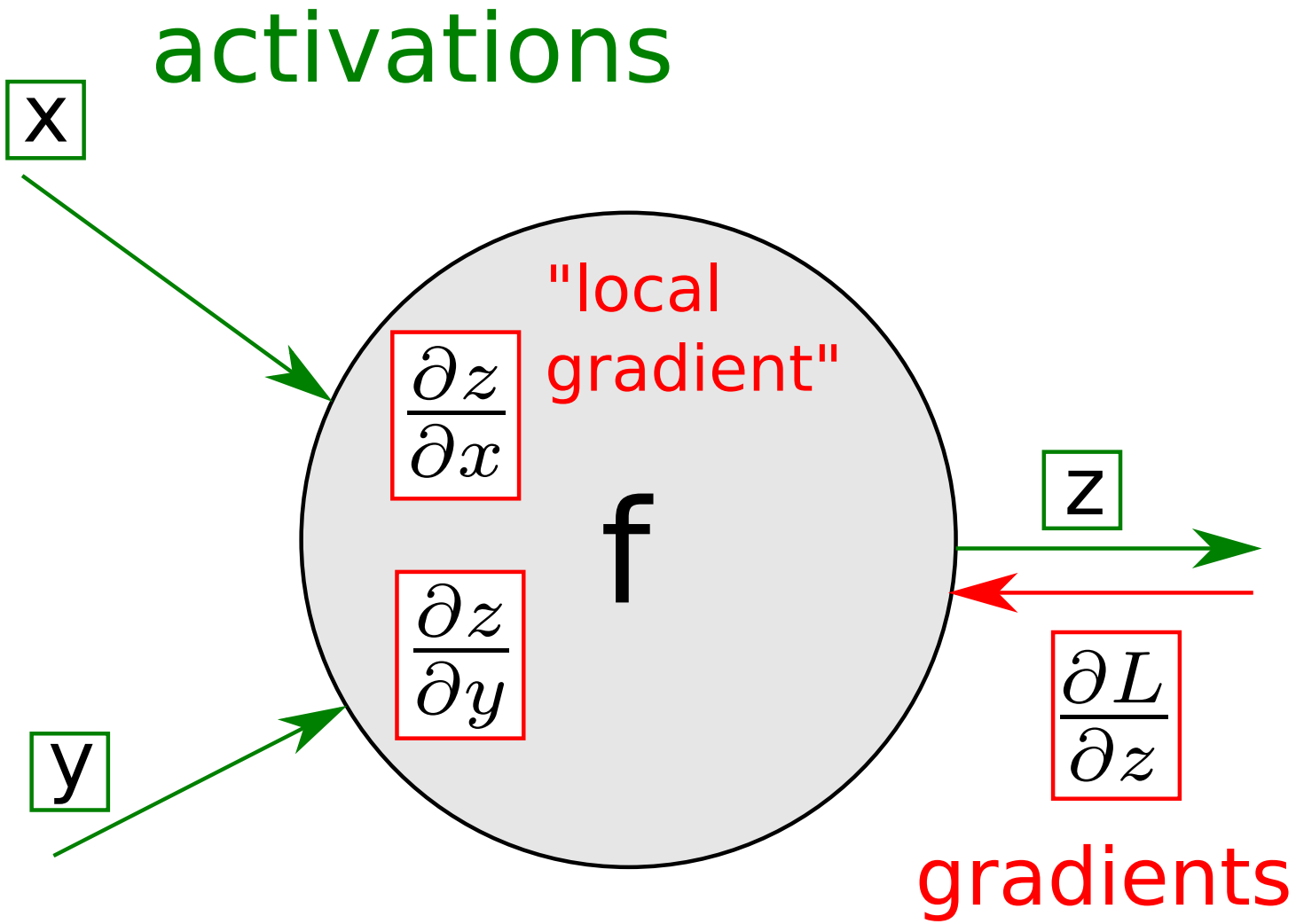
Backprop



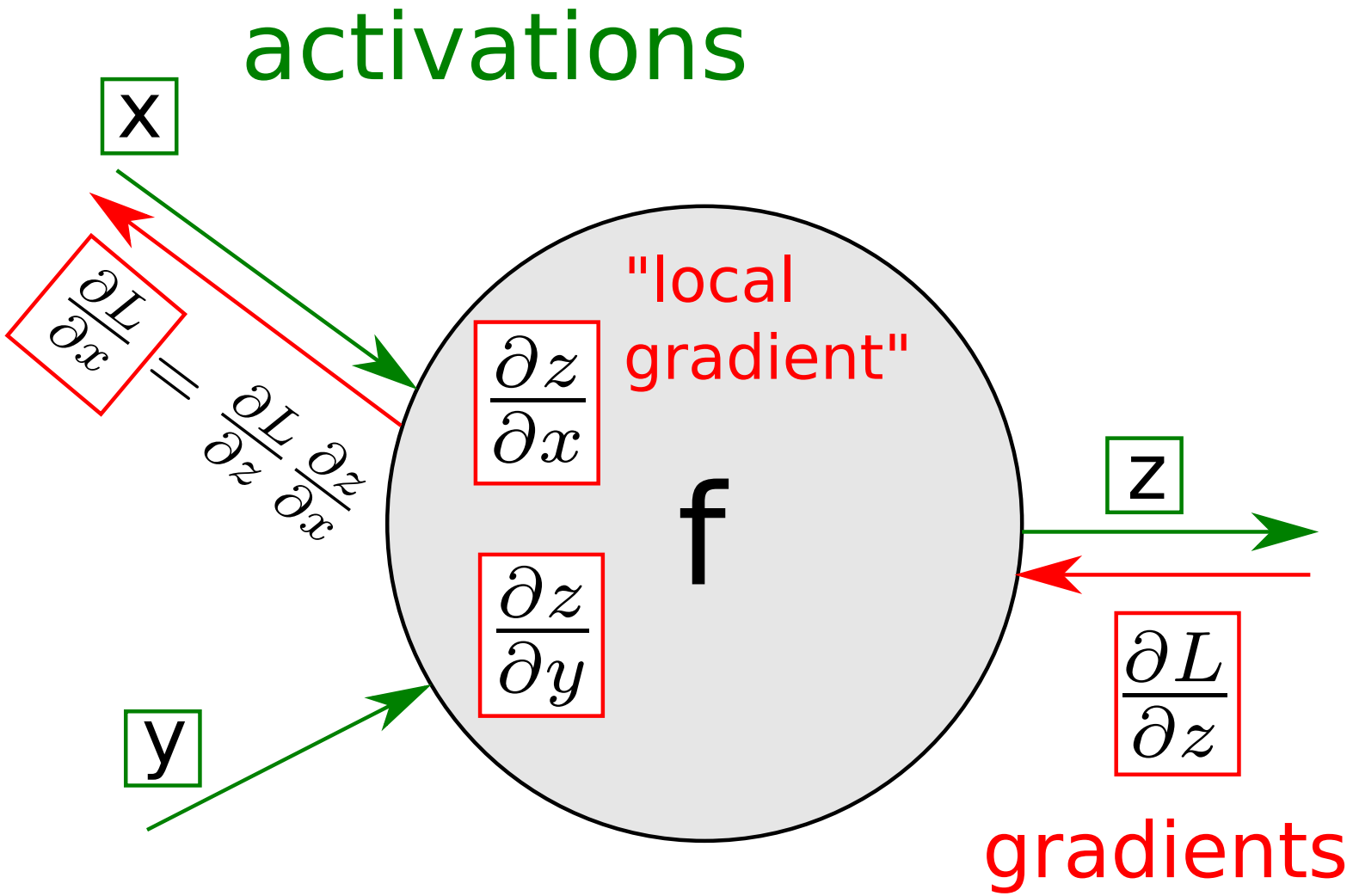
Backprop


activations
x
y
"local gradient"
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
f
z

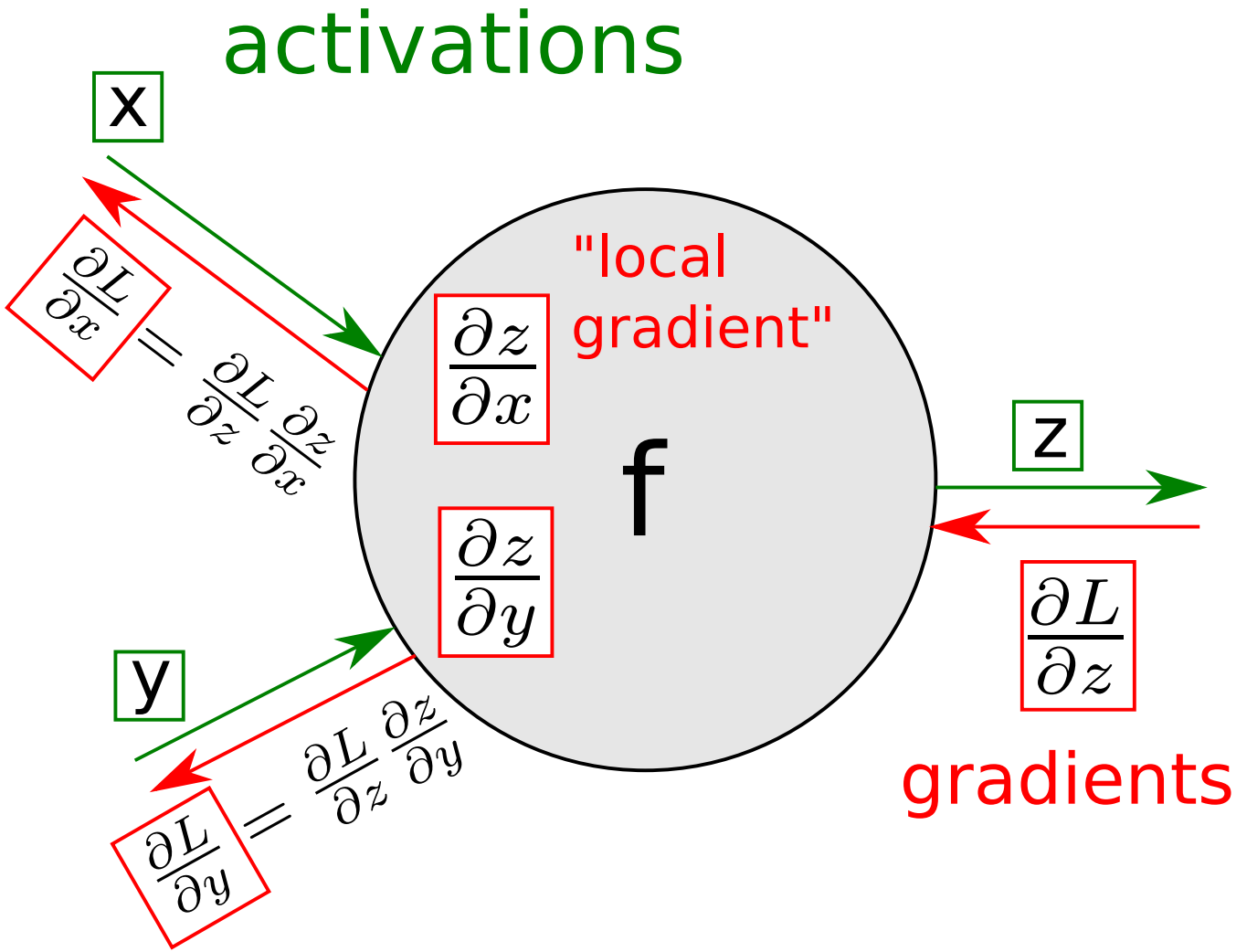
Backprop


activations
x
y
"local gradient"
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
f
z
$\frac{\partial L}{\partial z}$
gradients

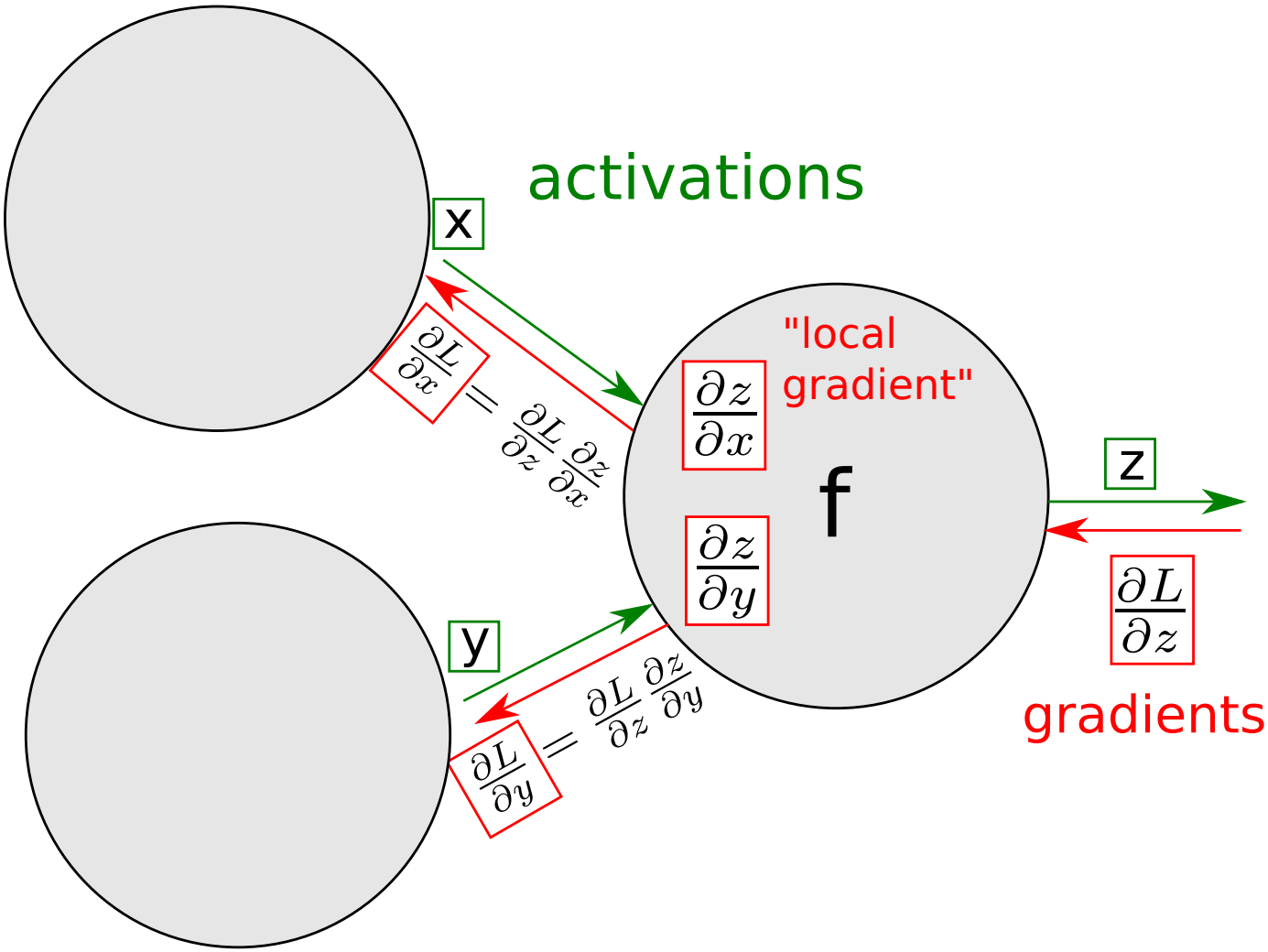
Backprop


activations
x
y
z
f
"local gradient"
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
$\frac{\partial L}{\partial z}$
gradients

Backprop


activations
x
y
f
z
"local gradient"
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
$\frac{\partial L}{\partial z}$
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
$\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$
gradients

Backprop


activations
x
y
z
"local gradient"
f
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
$\frac{\partial L}{\partial z}$
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
$\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$
gradients

Gradients add at branches