Lecture 4:

Backpropagation and Neural Networks part 1

Administrative

A1 is due Jan 20 (Wednesday). ~150 hours left
Warning: Jan 18 (Monday) is Holiday (no class/office hours)

Also note:
Lectures are non-exhaustive.
Read course notes for completeness.

I'll hold make up office hours on Wed Jan20, 5pm @ Gates 259

Where we are...

$$s = f(x; W) = Wx$$

scores function

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

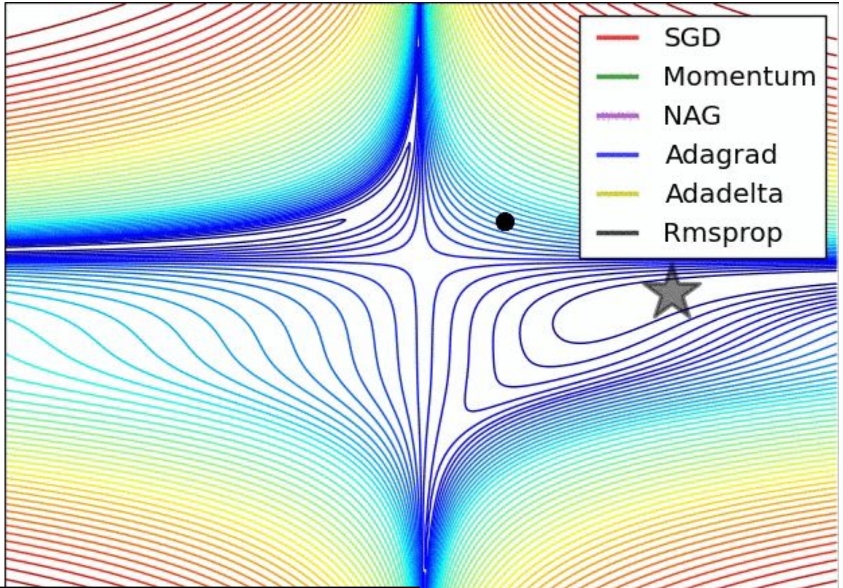
SVM loss

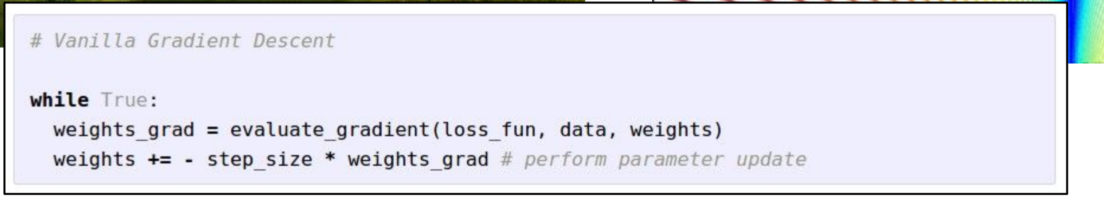
$$L = \frac{1}{N} \sum_{i=1}^{N} L_i + \sum_k W_k^2$$

data loss + regularization

want $\boxed{\nabla_W L}$

Optimization





```
# Vanilla Gradient Descent

while True:
  weights_grad = evaluate_gradient(loss_fun, data, weights)
  weights += - step_size * weights_grad # perform parameter update
```

(image credits
to Alec Radford)

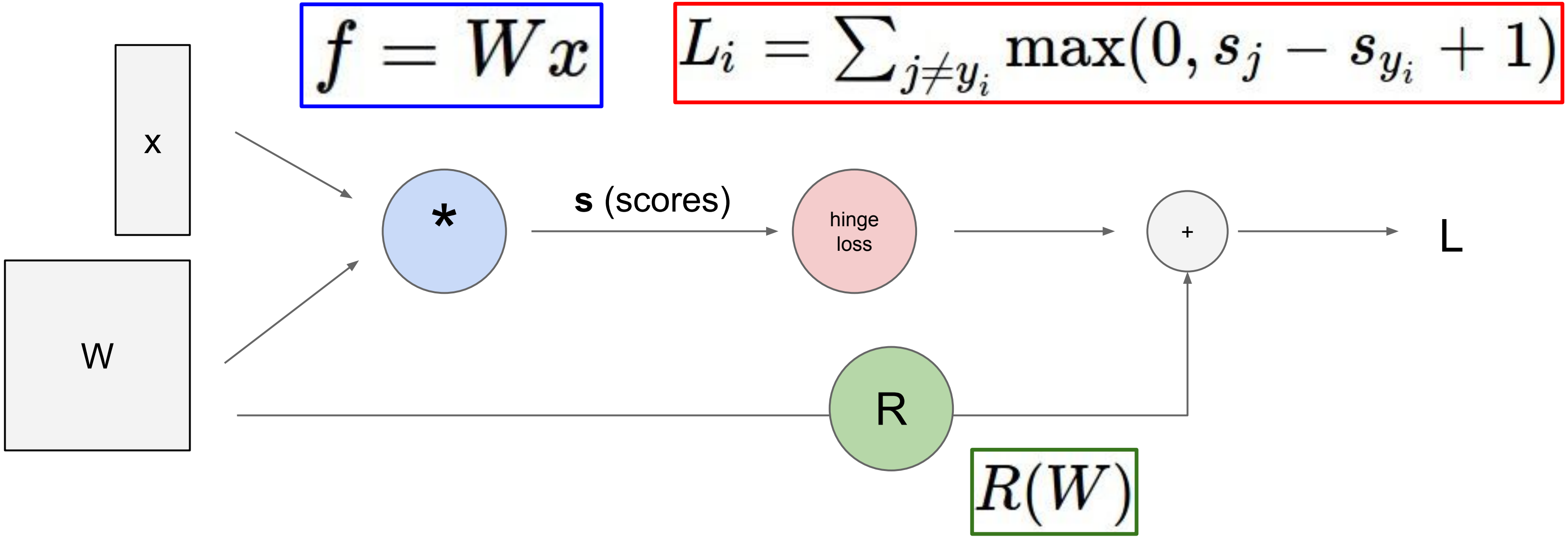
Gradient Descent

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

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

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

Computational Graph

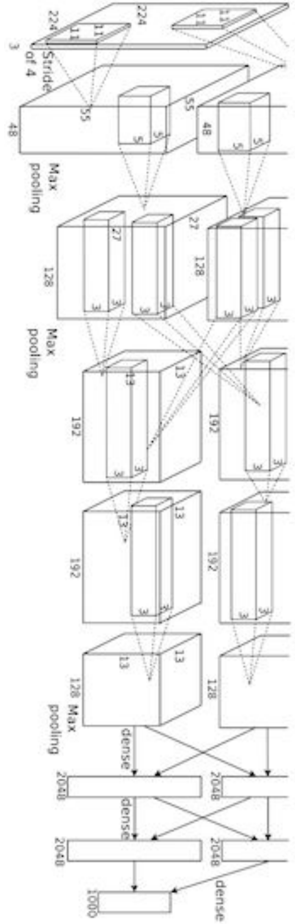


$$f = Wx$$

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$R(W)$$

Convolutional Network (AlexNet)

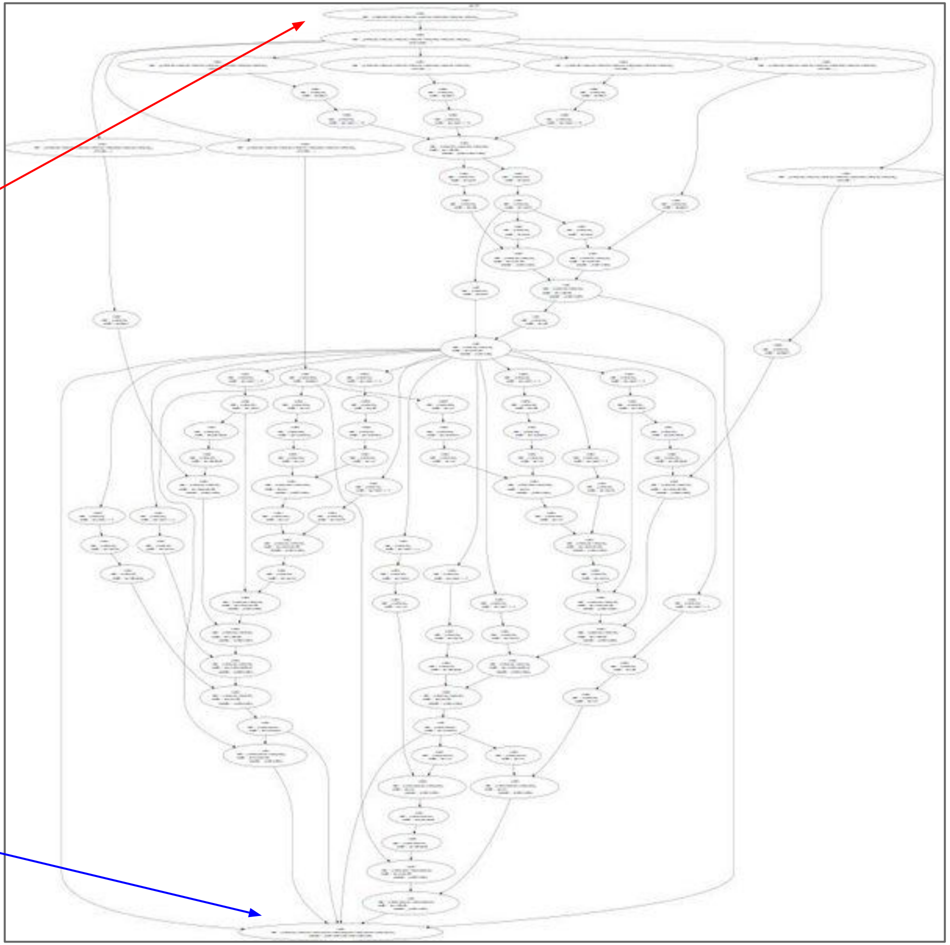


input image

weights

loss

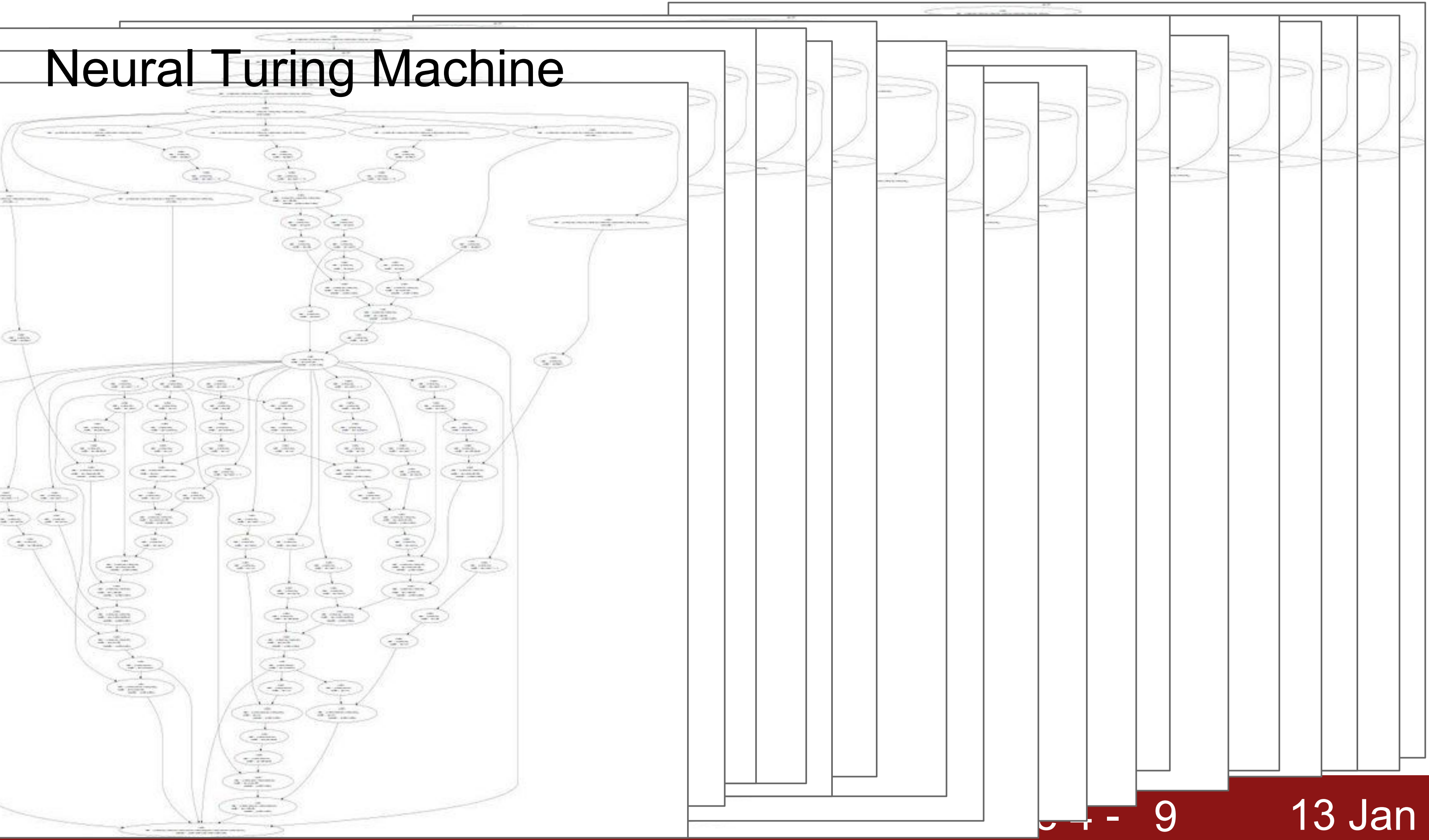
Neural Turing Machine



input tape

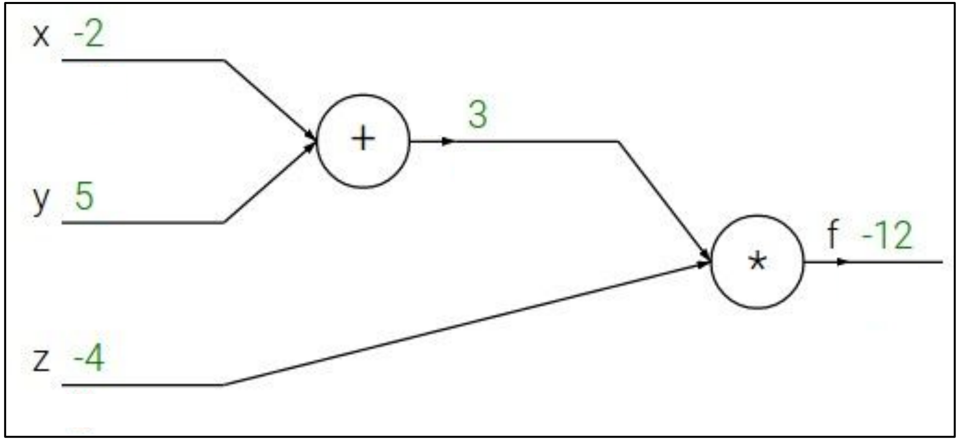
loss

Neural Turing Machine



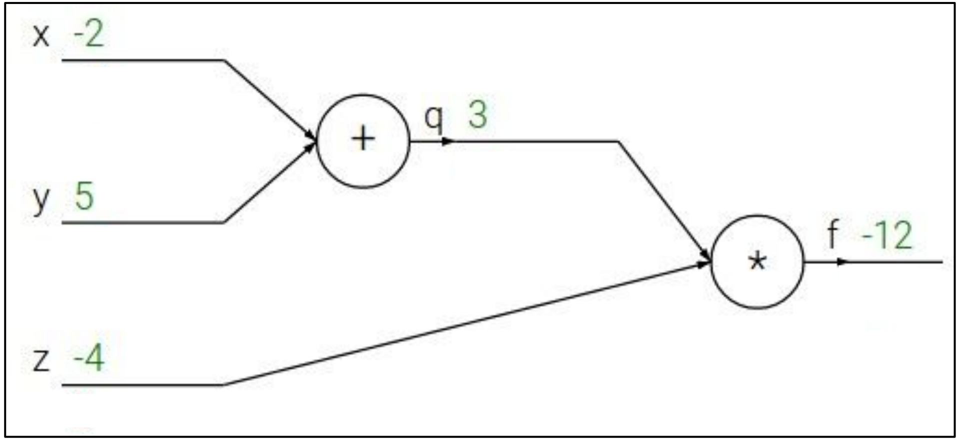
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4



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

e.g. x = -2, y = 5, z = -4



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

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

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

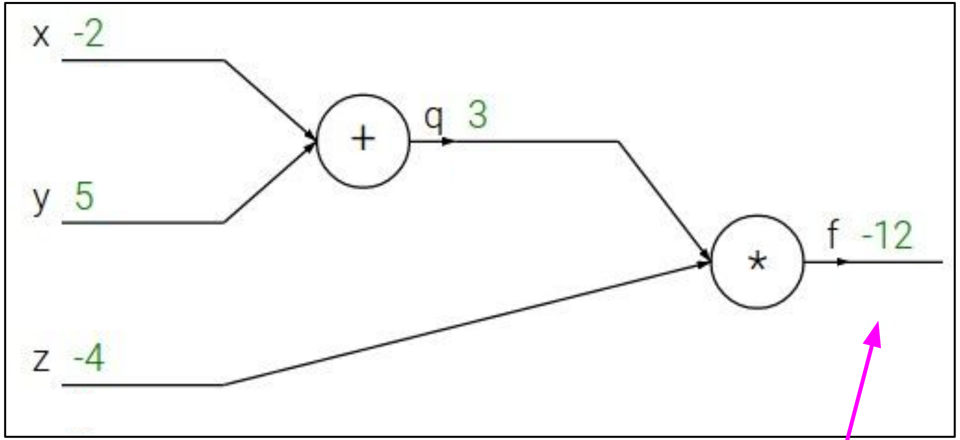
$$f(x,y,z) = (x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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



$\frac{\partial f}{\partial f}$

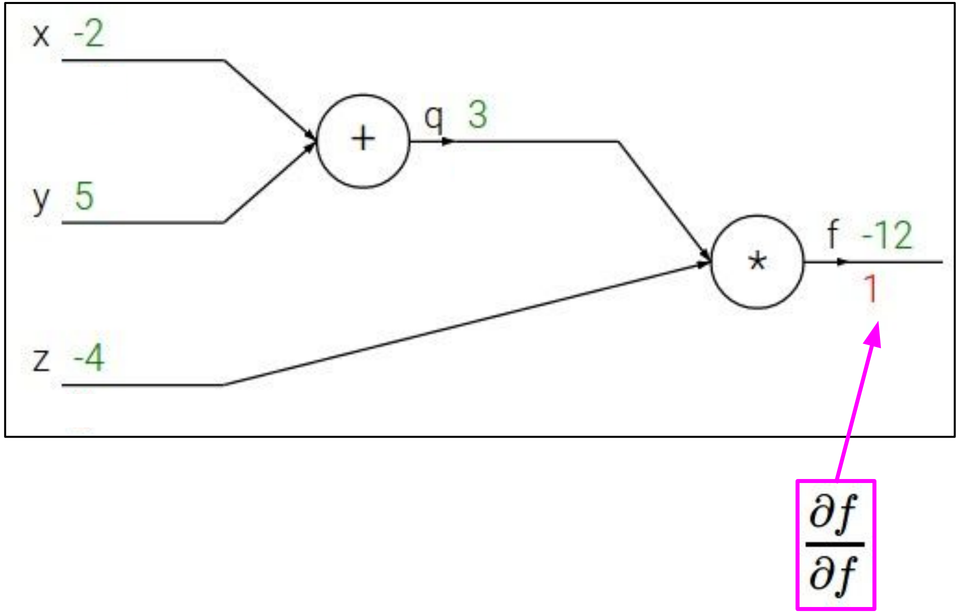
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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



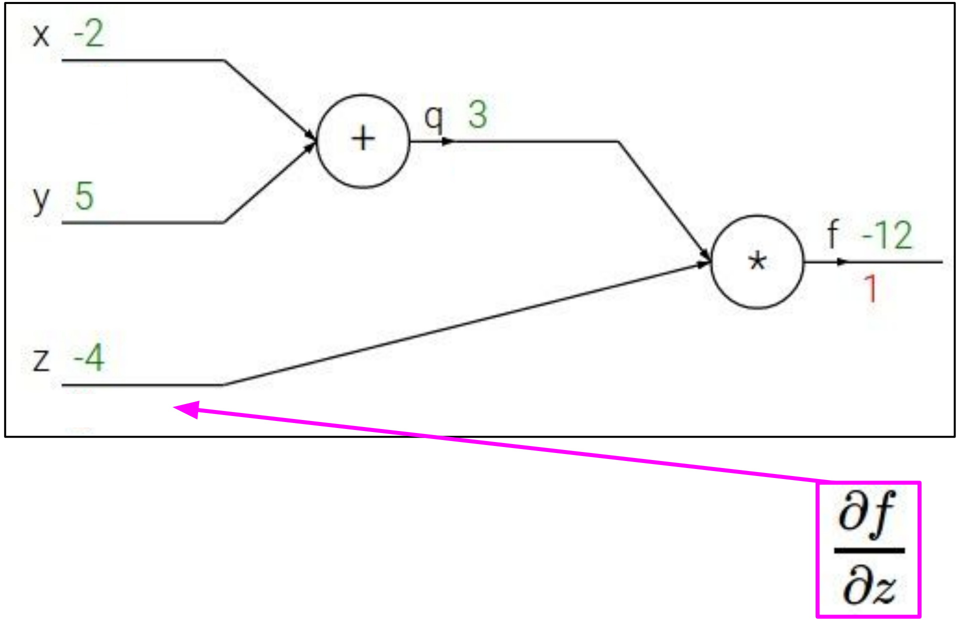
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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



$$\frac{\partial f}{\partial z}$$

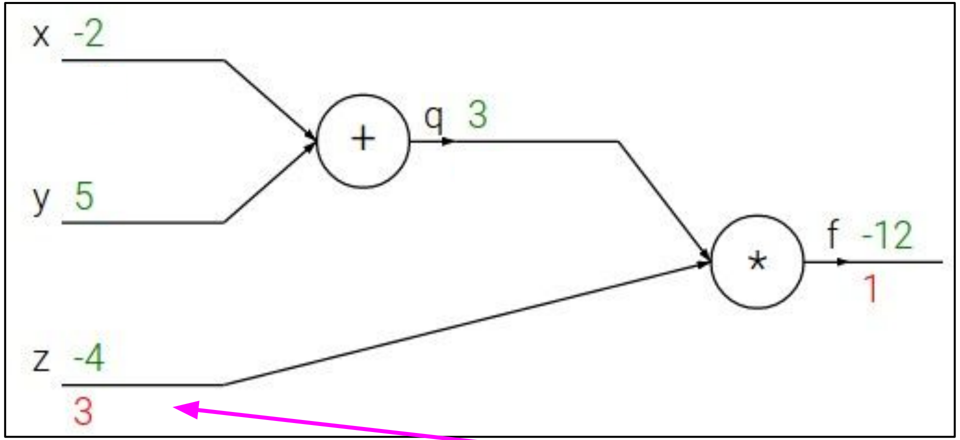
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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



$\frac{\partial f}{\partial z}$

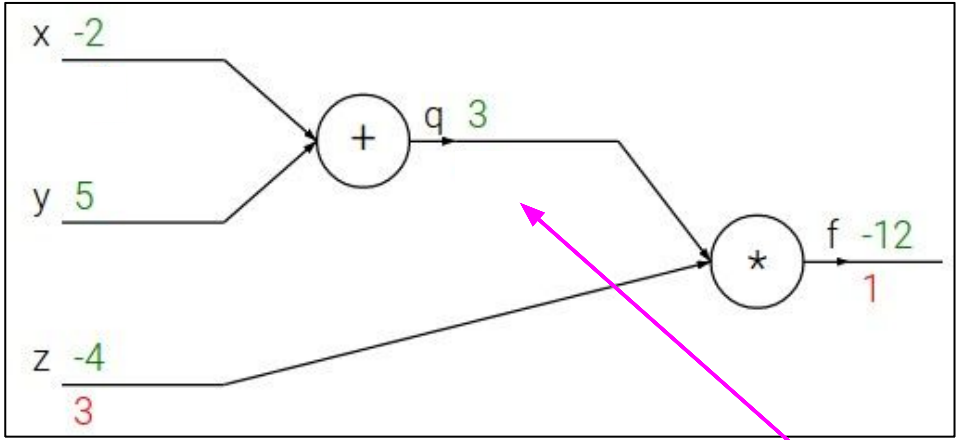
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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



$$\frac{\partial f}{\partial q}$$

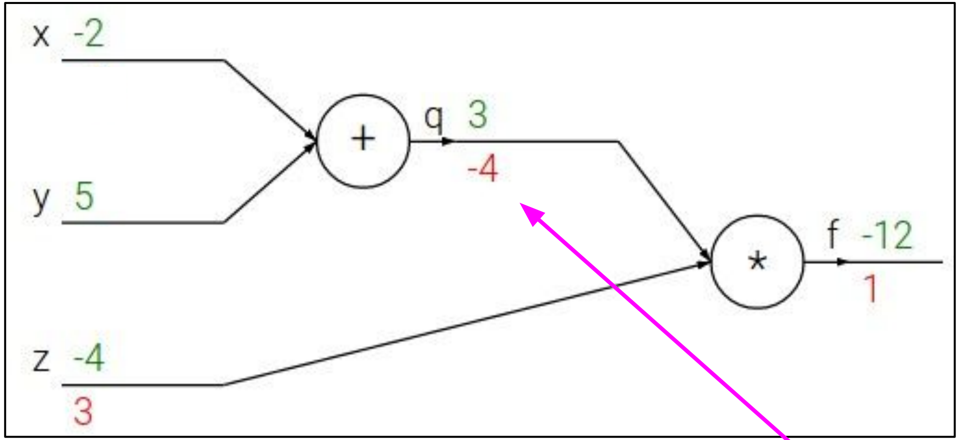
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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



$$\boxed{\frac{\partial f}{\partial q}}$$

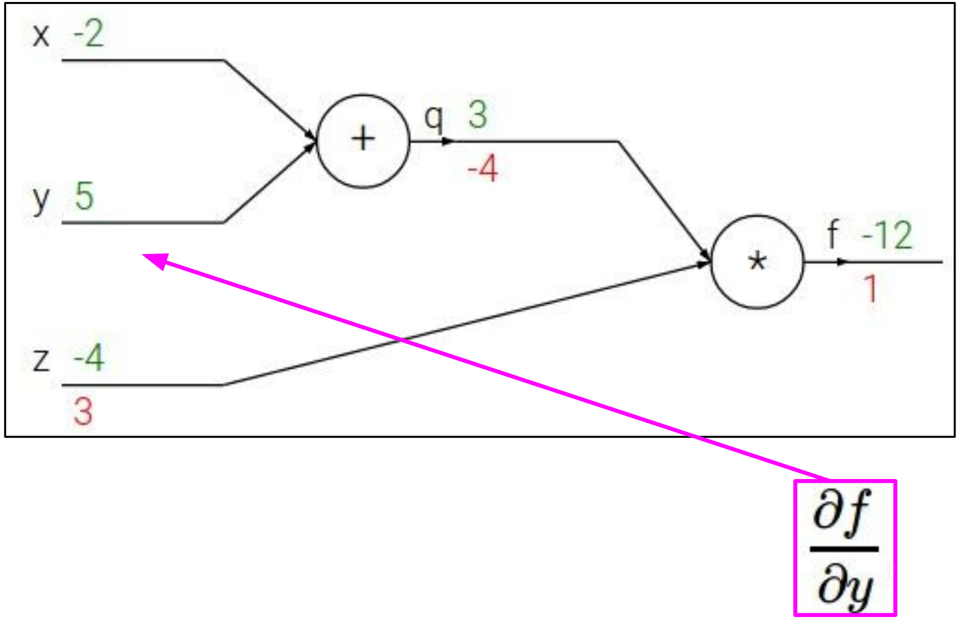
$$f(x,y,z) = (x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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



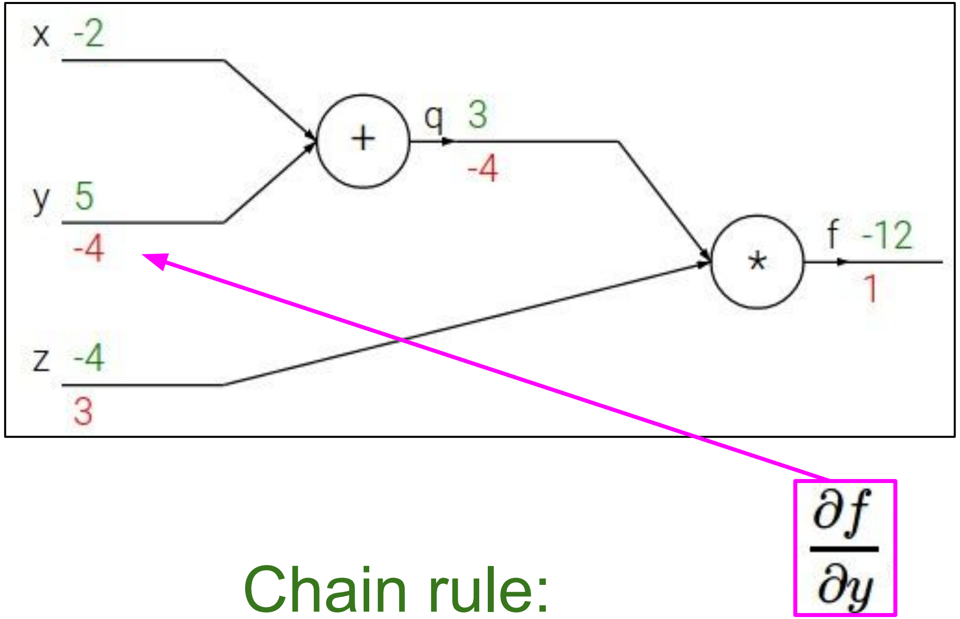
$$f(x,y,z) = (x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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



Chain rule:

$$\boxed{\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial y}}$$

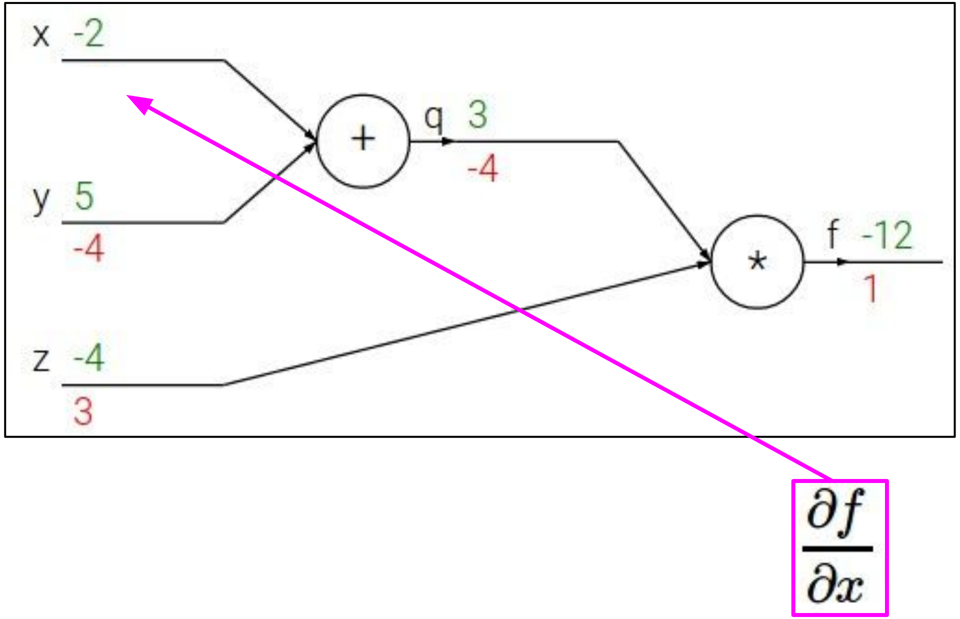
$$f(x,y,z) = (x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

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



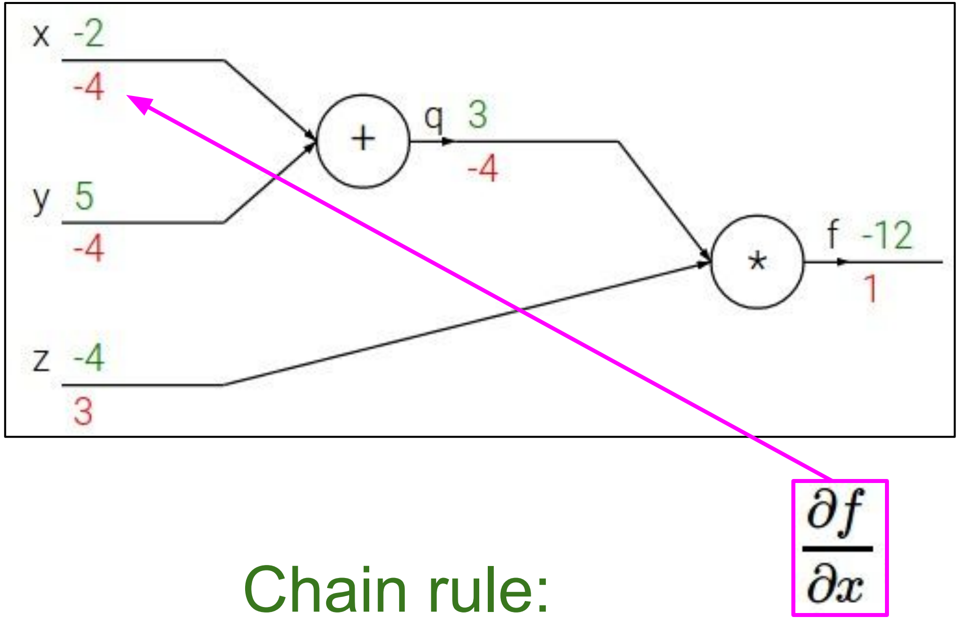
$$f(x,y,z) = (x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

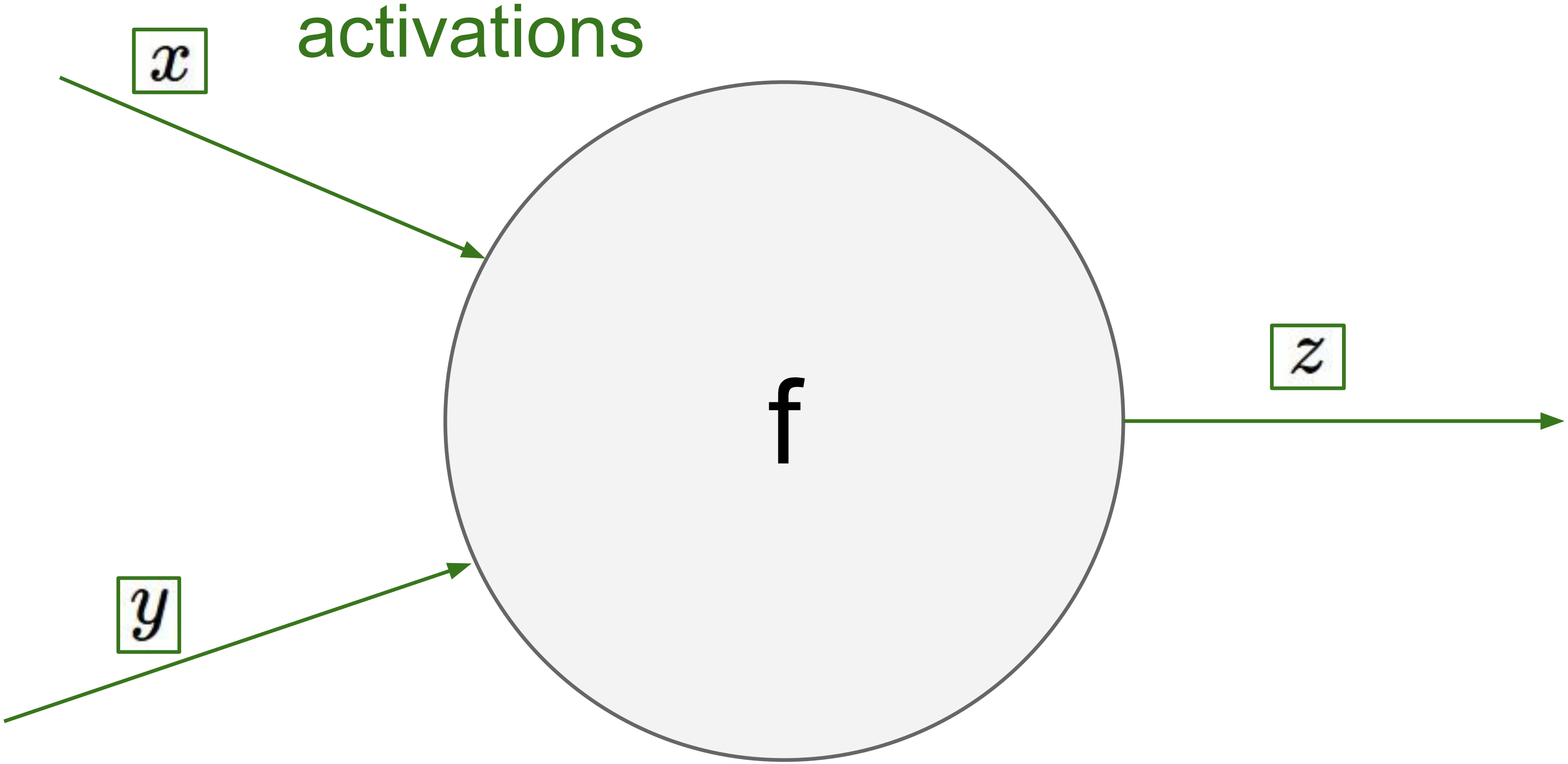
Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

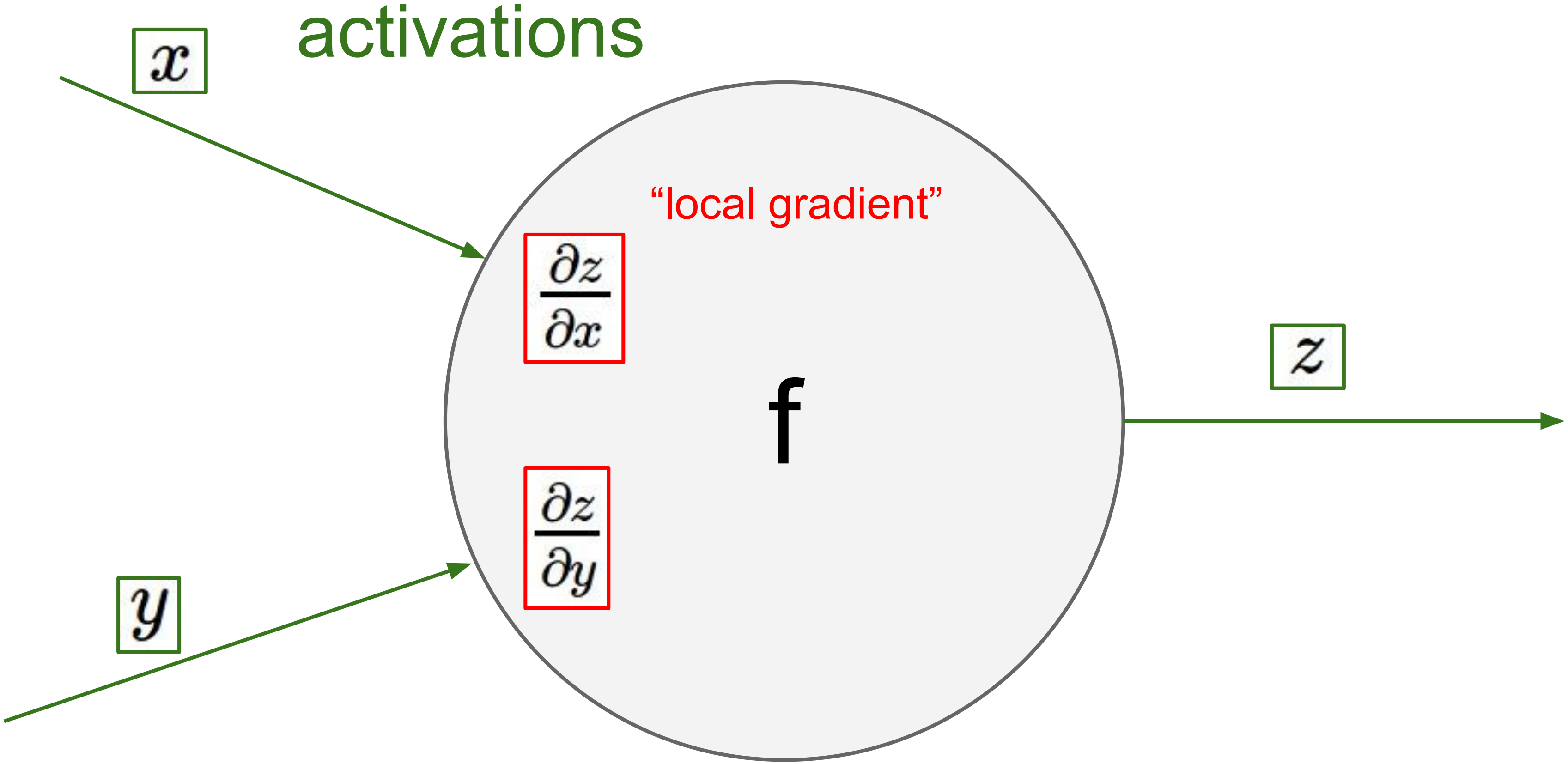


$$\boxed{\frac{\partial f}{\partial x}}$$

Chain rule:

$$\boxed{\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}}$$


activations
x
y
f
z



activations

x

y

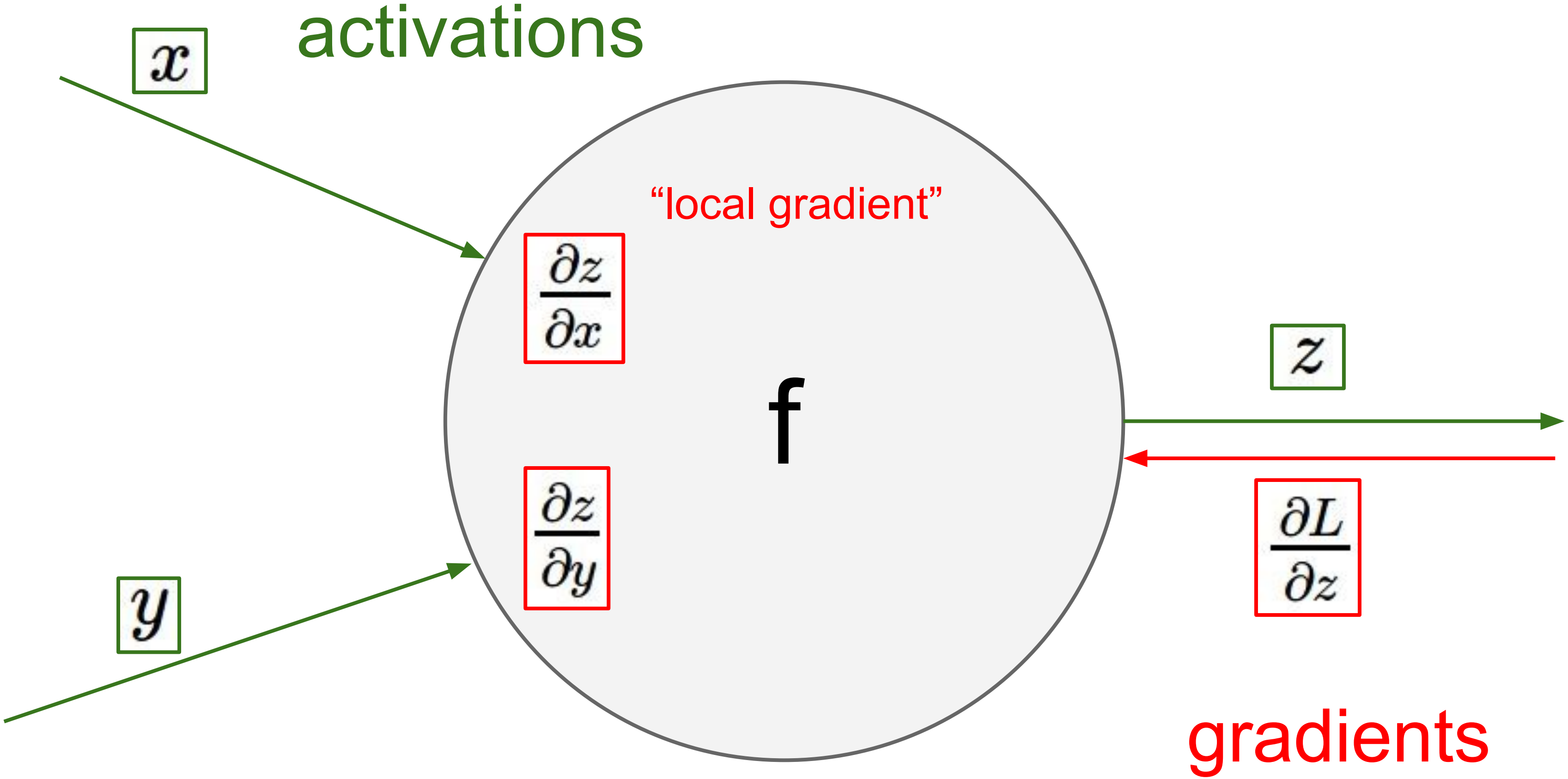
z

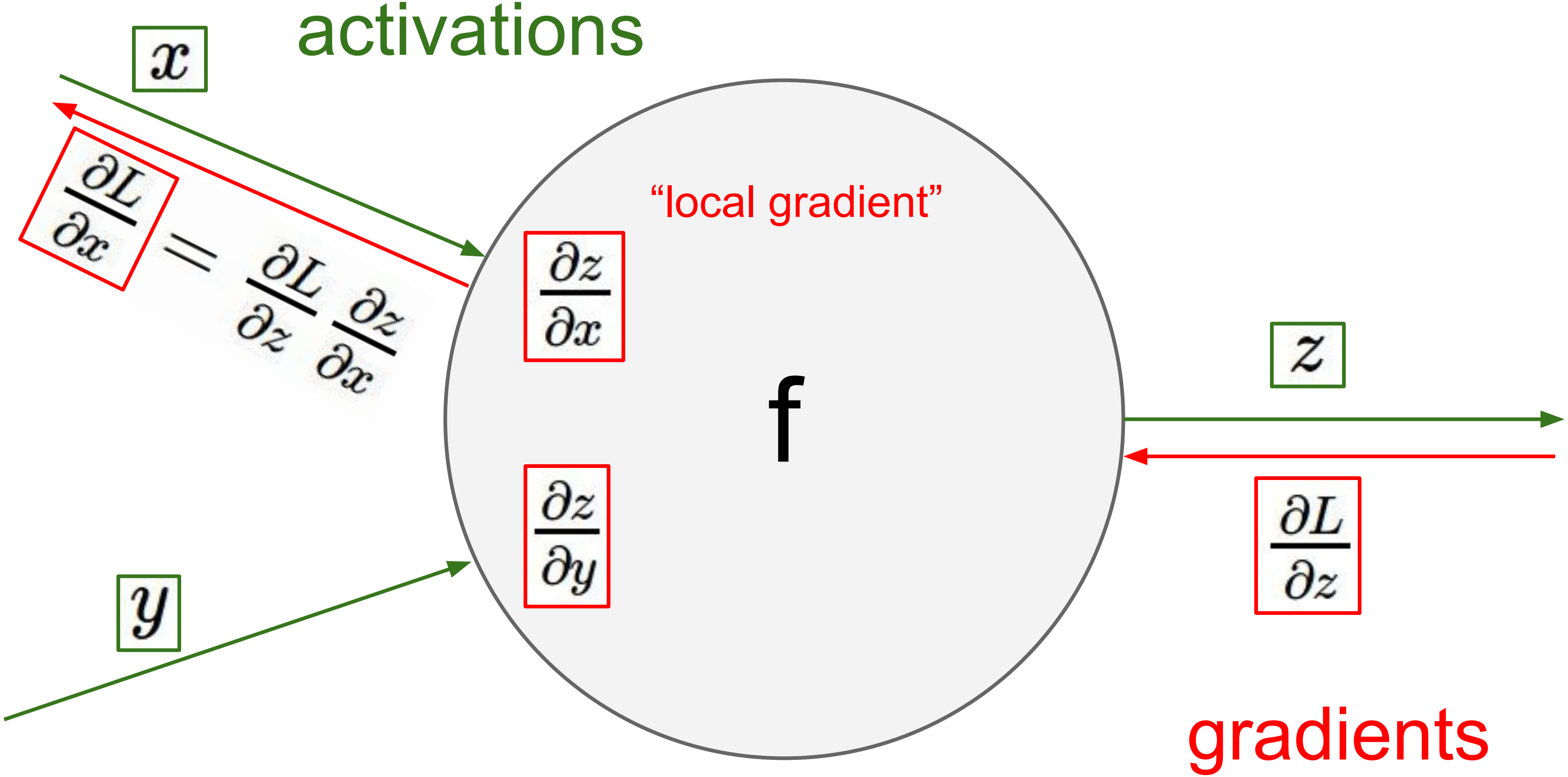
f

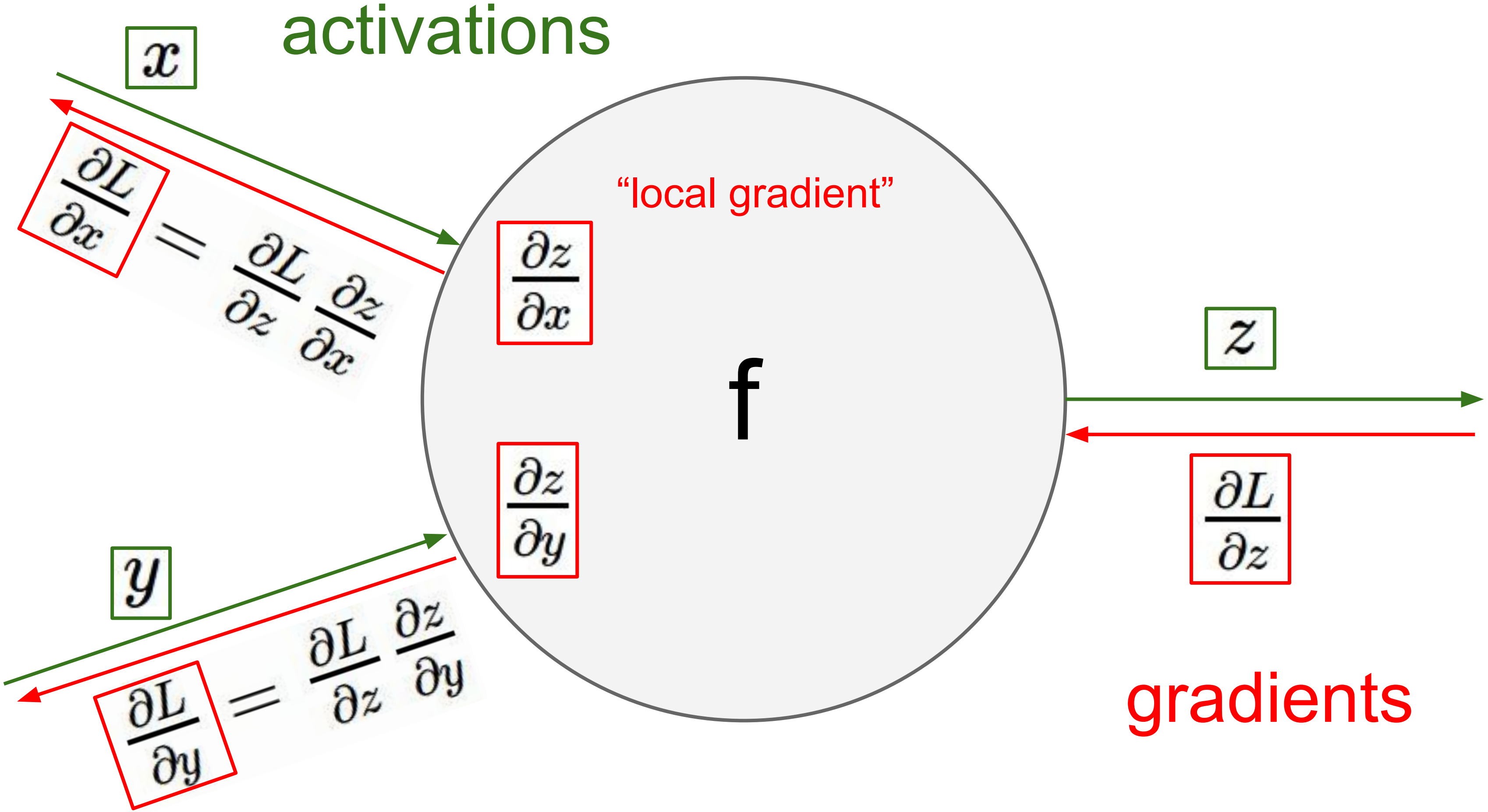
“local gradient”

$\frac{\partial z}{\partial x}$

$\frac{\partial z}{\partial y}$


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


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



activations

x

$$\boxed{\frac{\partial L}{\partial x}} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$$

“local gradient”

$$\boxed{\frac{\partial z}{\partial x}}$$

$$\boxed{\frac{\partial z}{\partial y}}$$

f

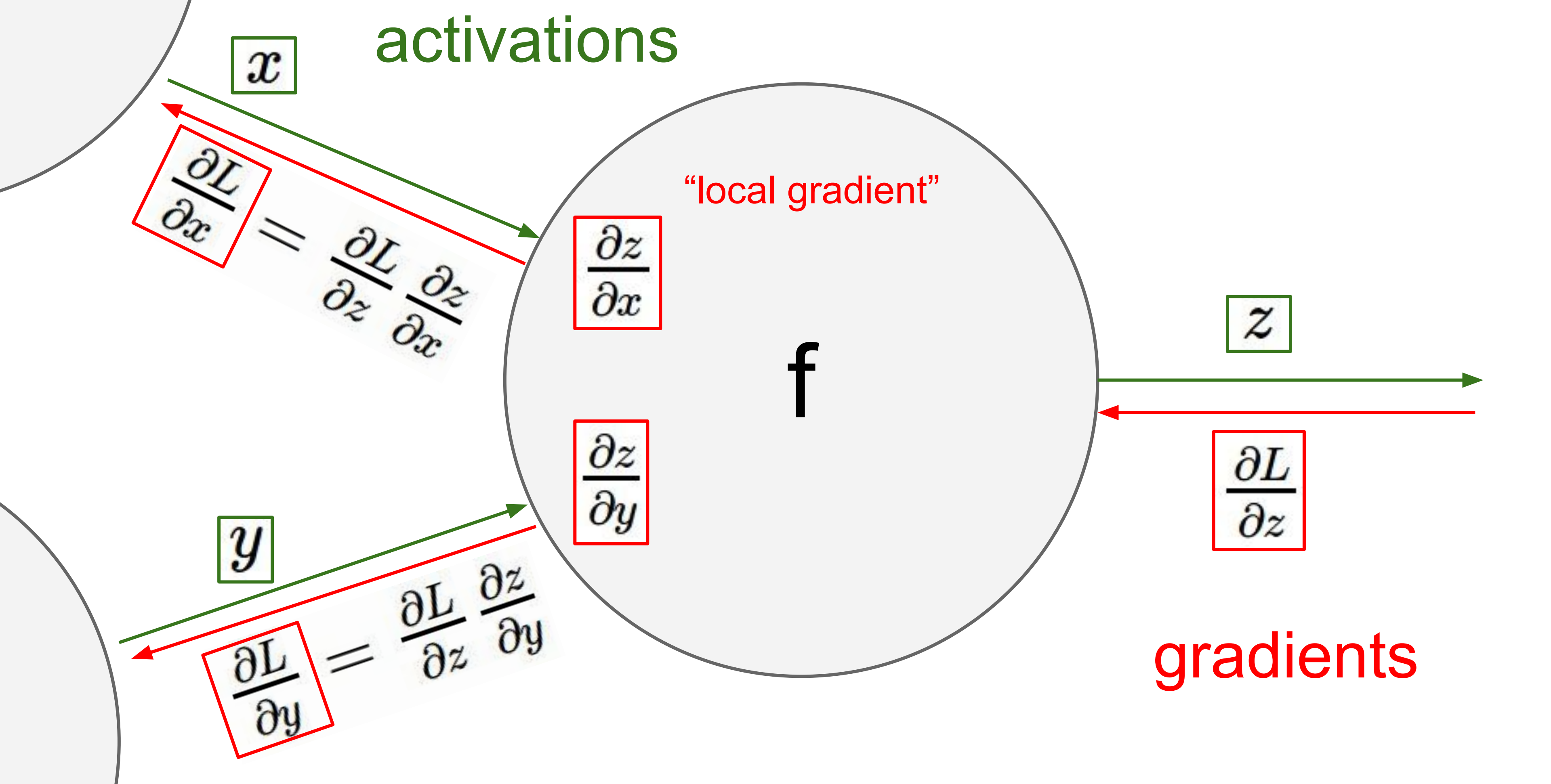
z

$$\boxed{\frac{\partial L}{\partial z}}$$

y

$$\boxed{\frac{\partial L}{\partial y}} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$$

gradients



activations

x

$$\boxed{\frac{\partial L}{\partial x}} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$$

“local gradient”

$\boxed{\frac{\partial z}{\partial x}}$

f

$\boxed{\frac{\partial z}{\partial y}}$

z

$\boxed{\frac{\partial L}{\partial z}}$

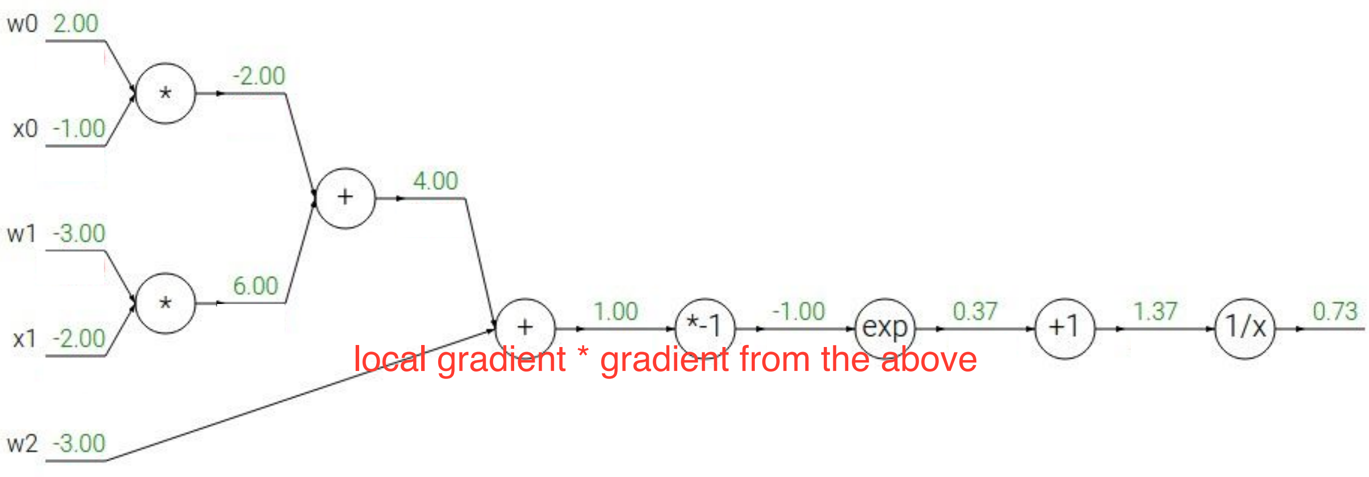
y

$$\boxed{\frac{\partial L}{\partial y}} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$$

gradients

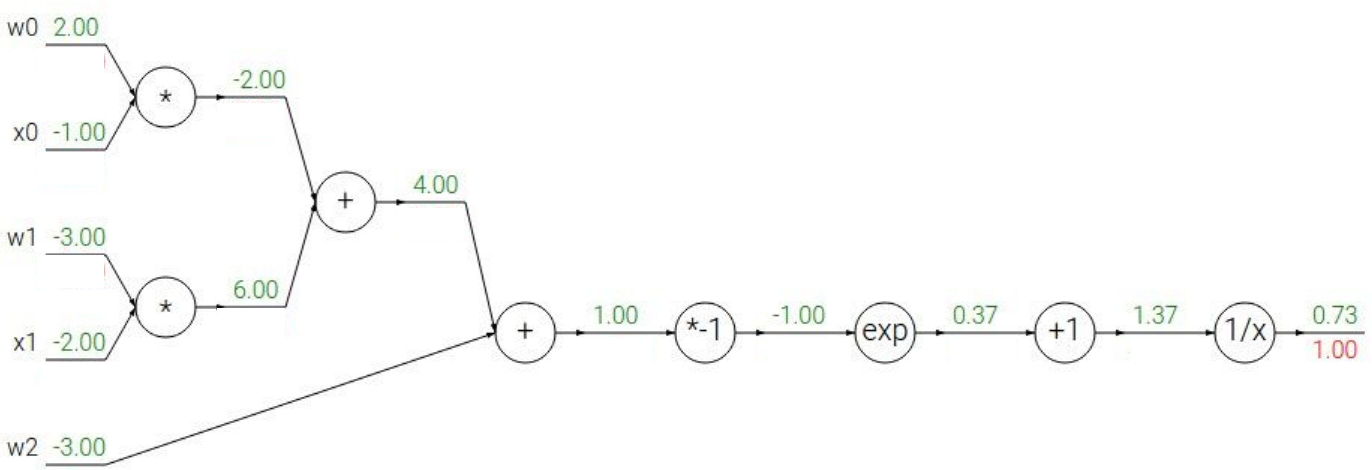
Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



local gradient * gradient from the above

Another example: $$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

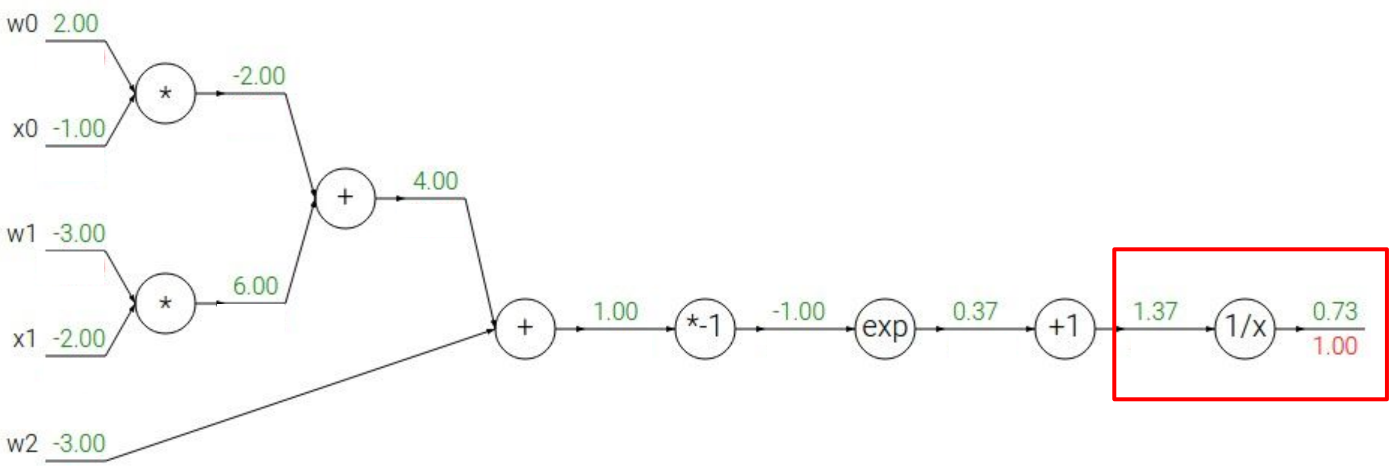
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

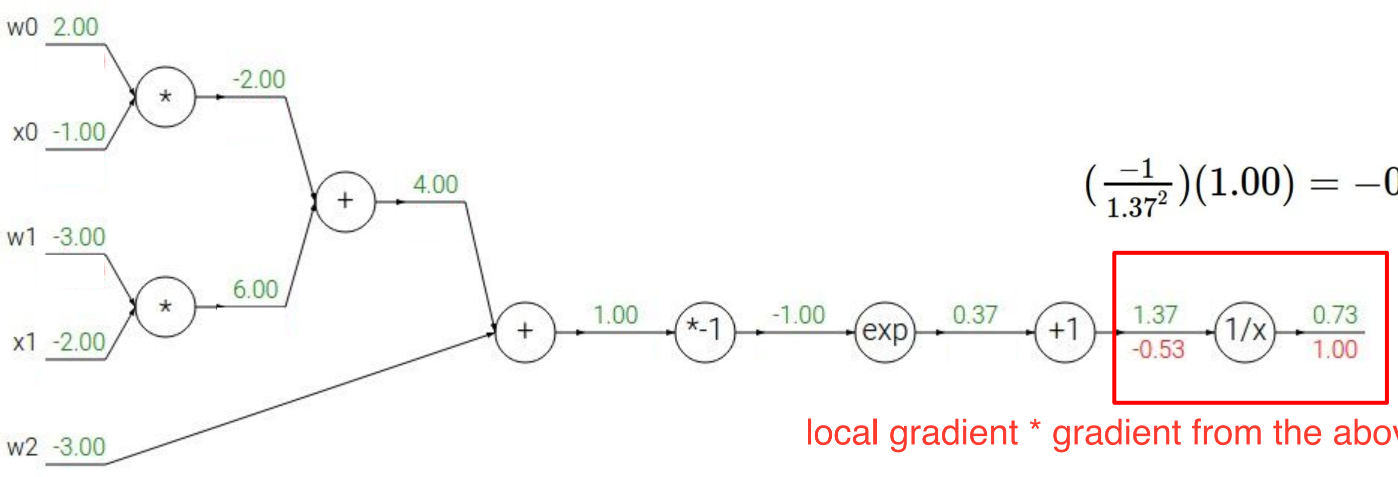
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$\left(\frac{-1}{1.37^2}\right)(1.00) = -0.53$$

local gradient * gradient from the above

$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

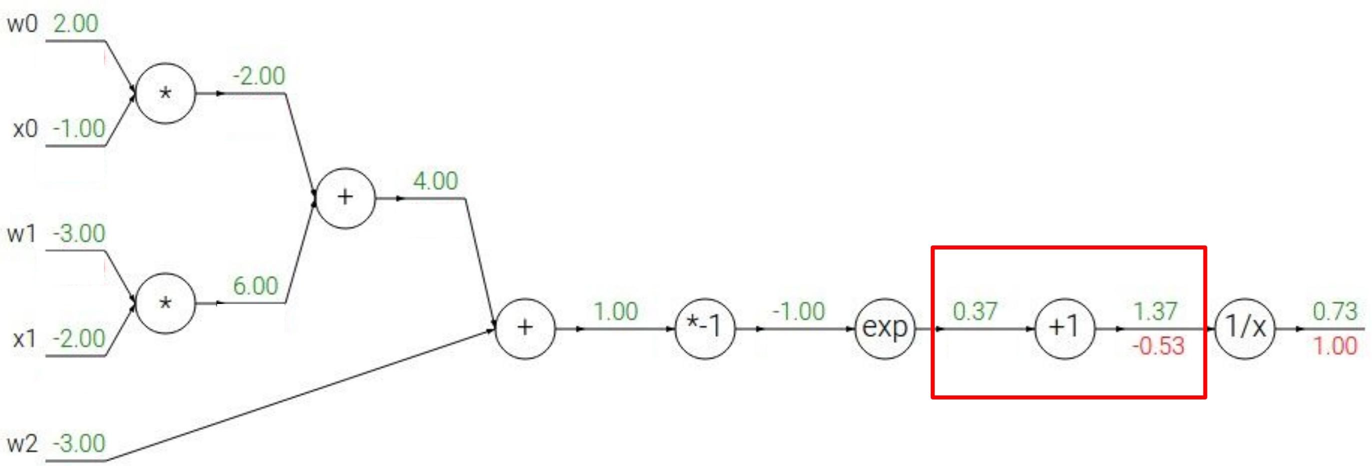
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

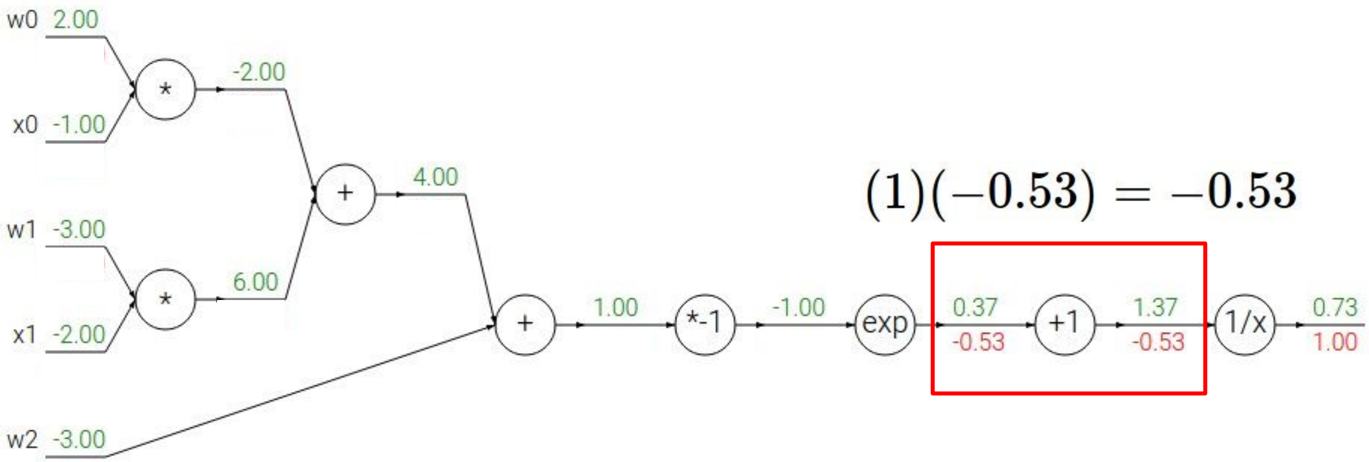
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$\boxed{f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1}$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$(1)(-0.53) = -0.53$$

$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

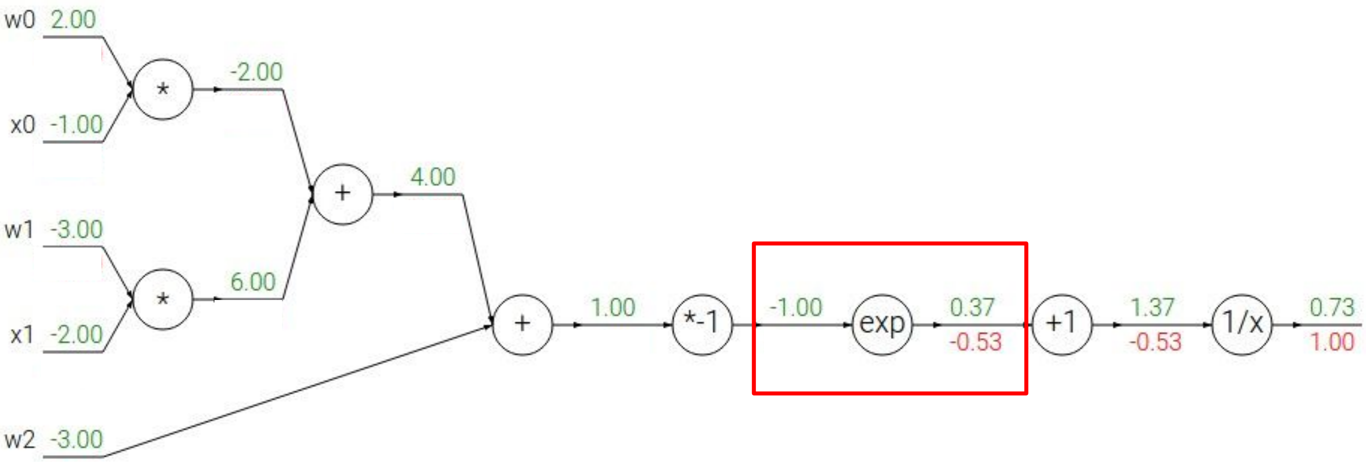
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

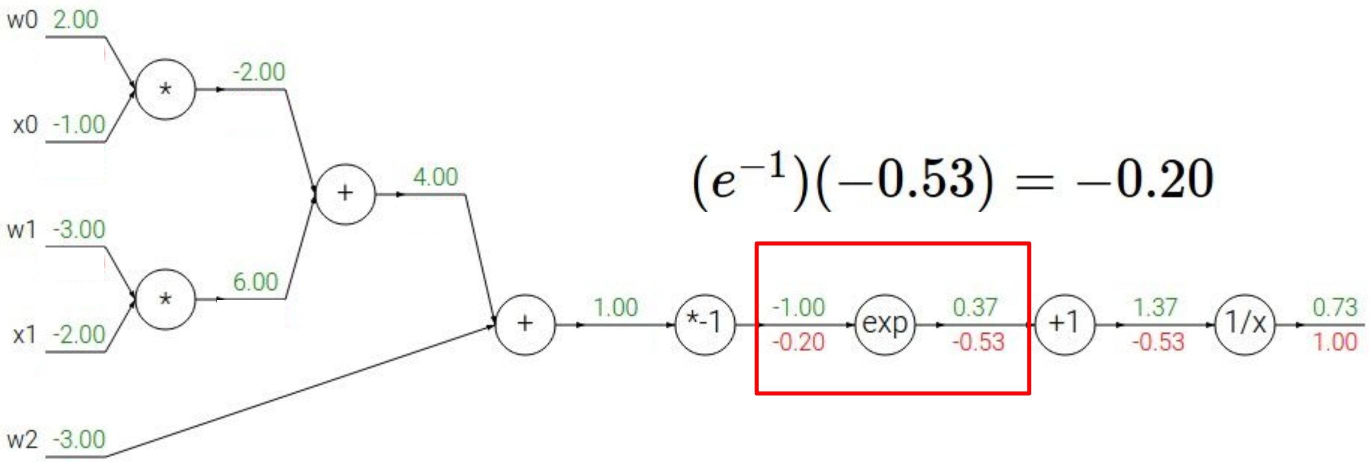
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$(e^{-1})(-0.53) = -0.20$$

$$\boxed{f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x}$$

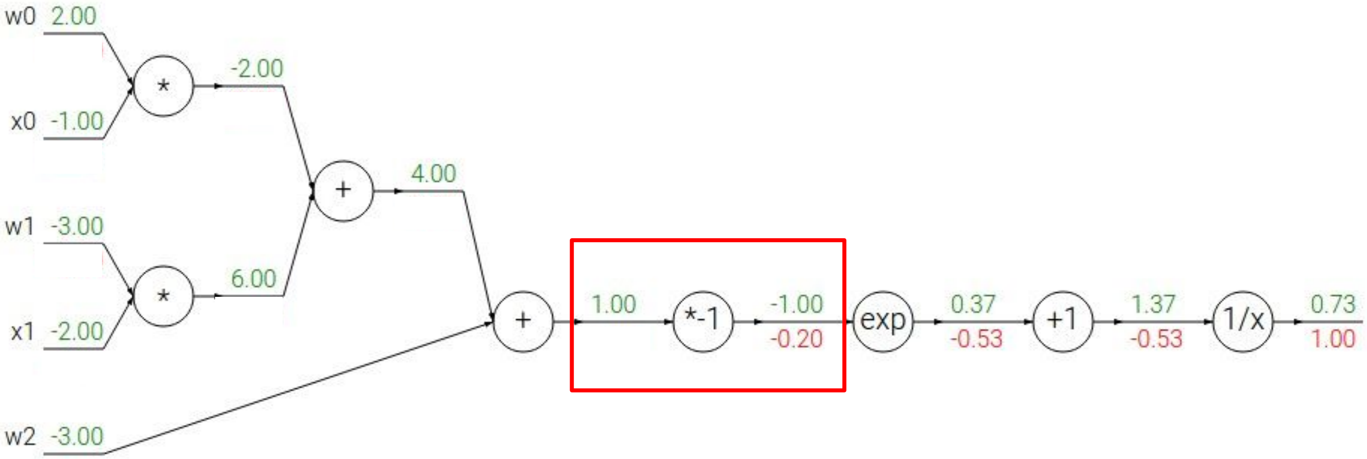
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

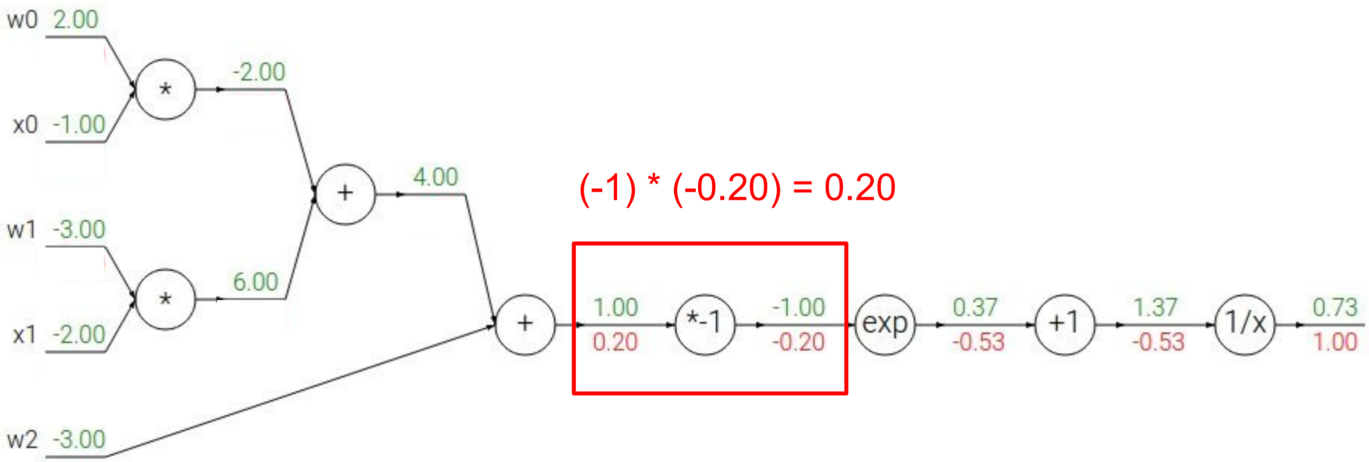
$$\boxed{f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a}$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



(-1) * (-0.20) = 0.20

$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

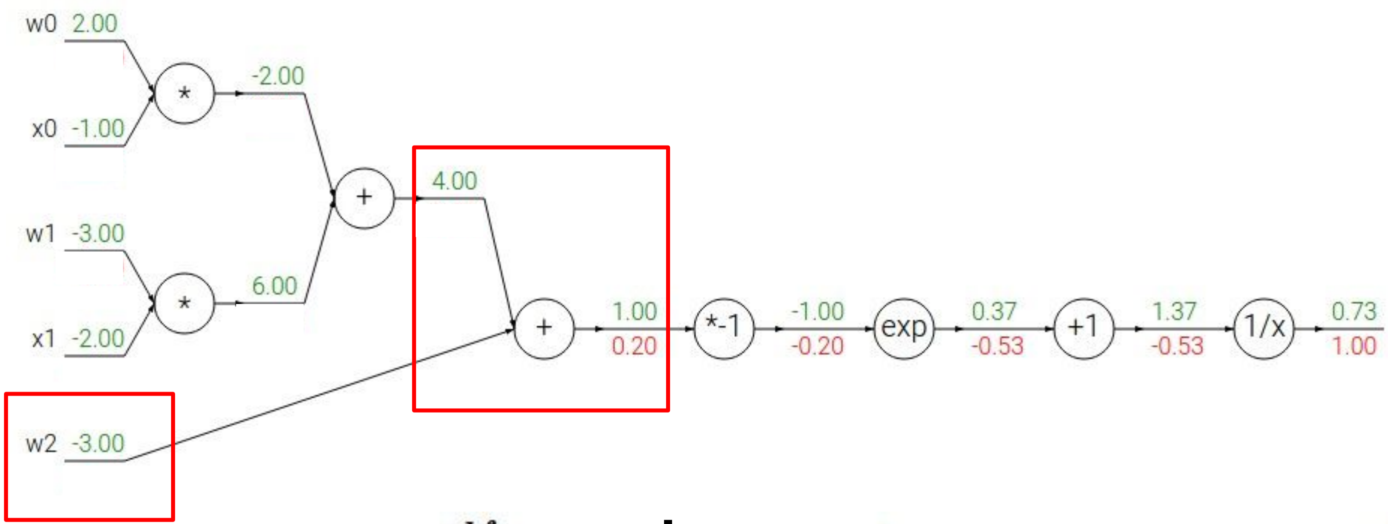
$$\boxed{f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a}$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

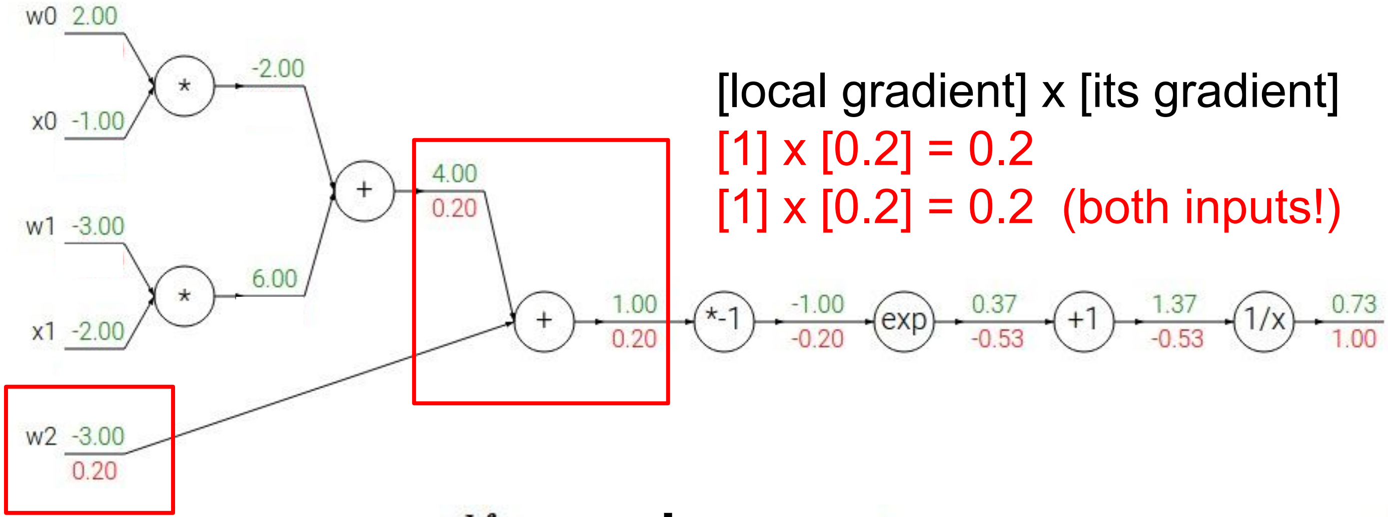
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



[local gradient] x [its gradient]

[1] x [0.2] = 0.2

[1] x [0.2] = 0.2 (both inputs!)

$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

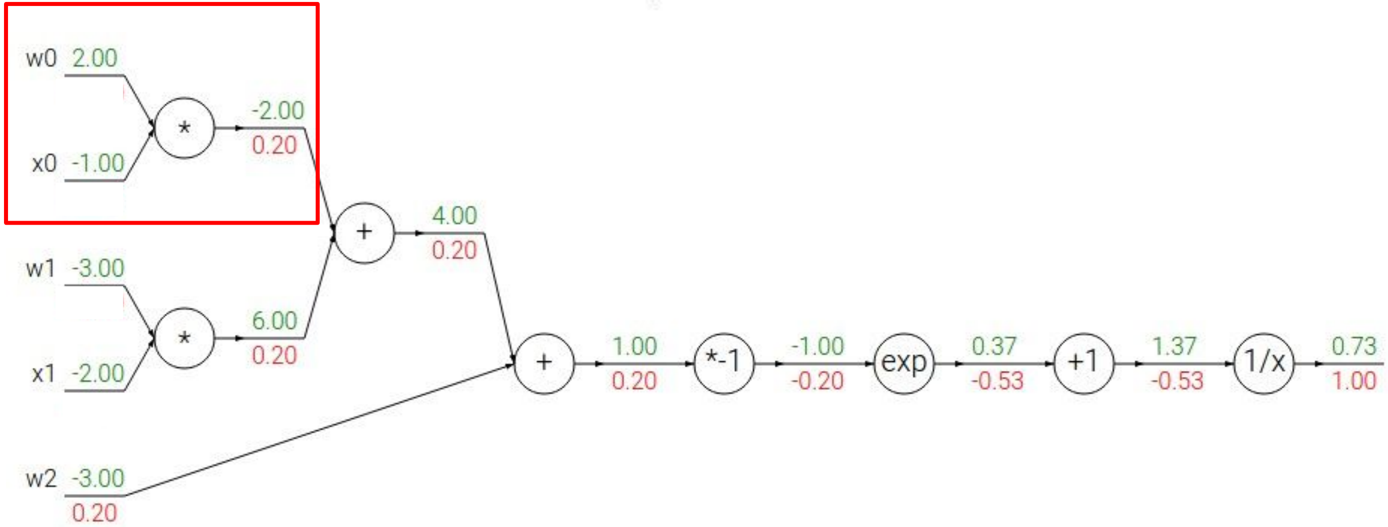
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

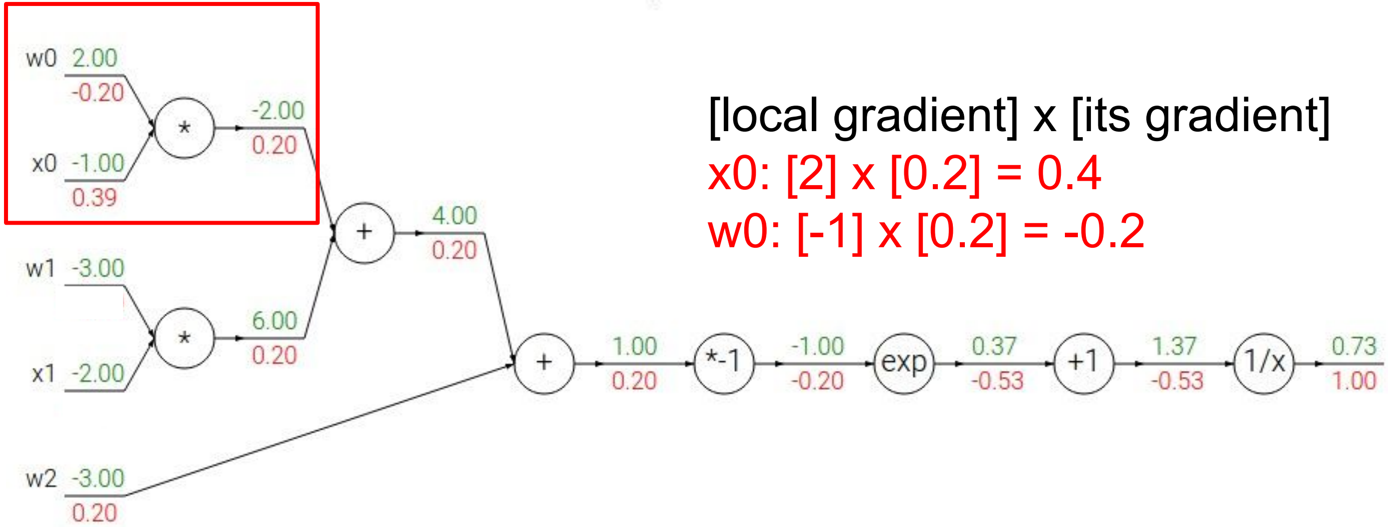
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

Another example:

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



[local gradient] x [its gradient]

x0: [2] x [0.2] = 0.4

w0: [-1] x [0.2] = -0.2

$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

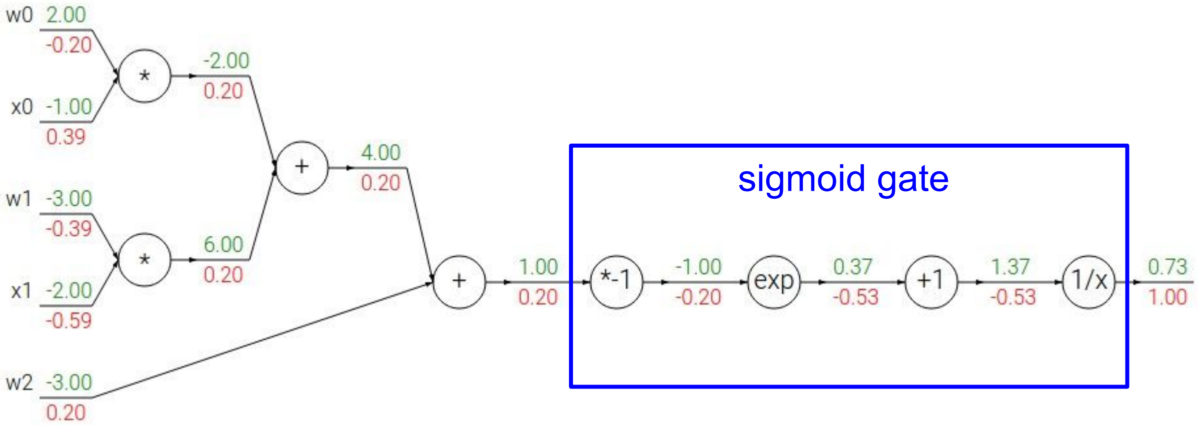
$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

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

sigmoid function

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\sigma(x)$$

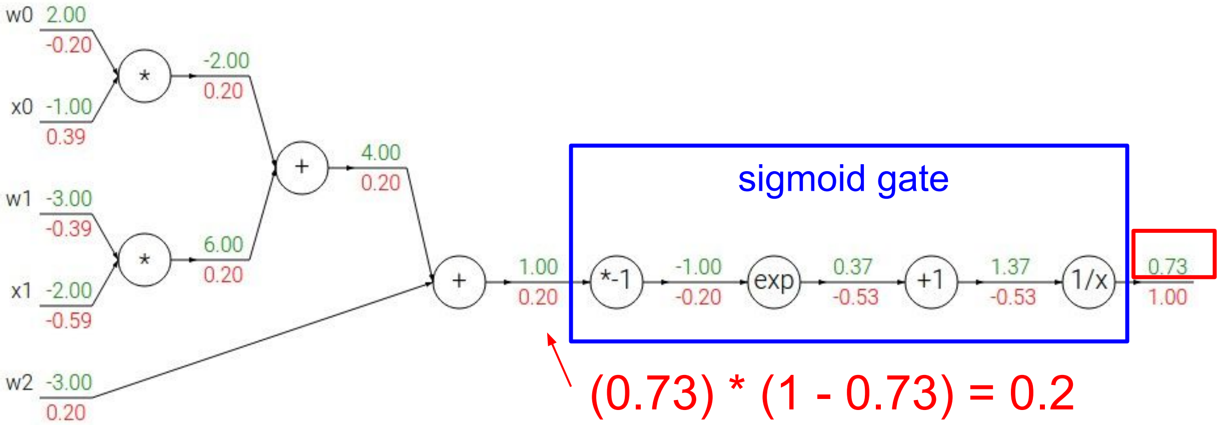


$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

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

sigmoid function

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$



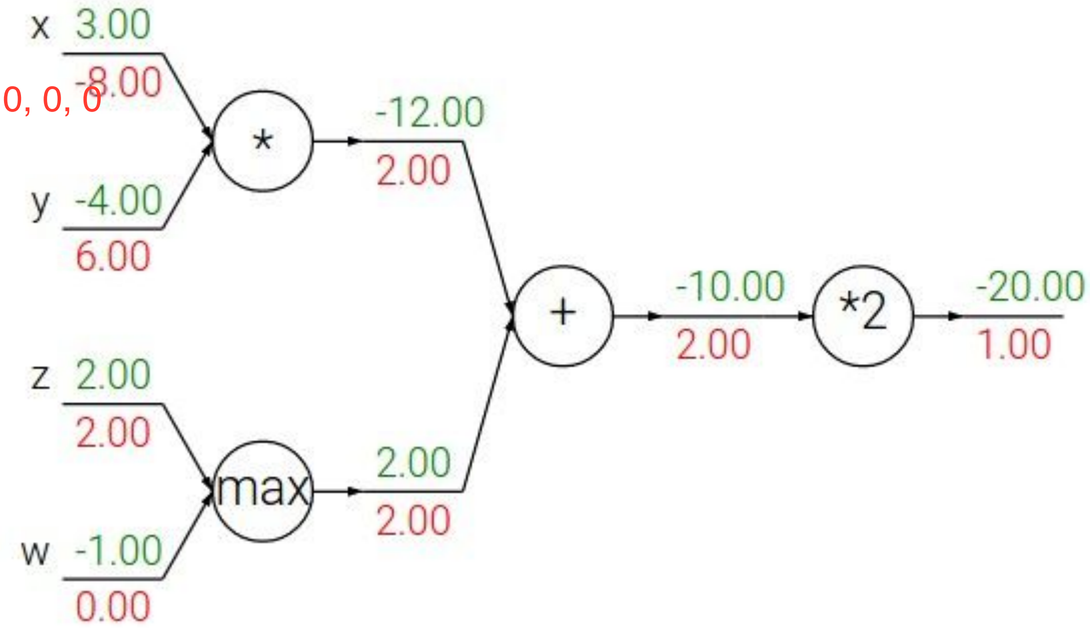
(0.73) * (1 - 0.73) = 0.2

Patterns in backward flow

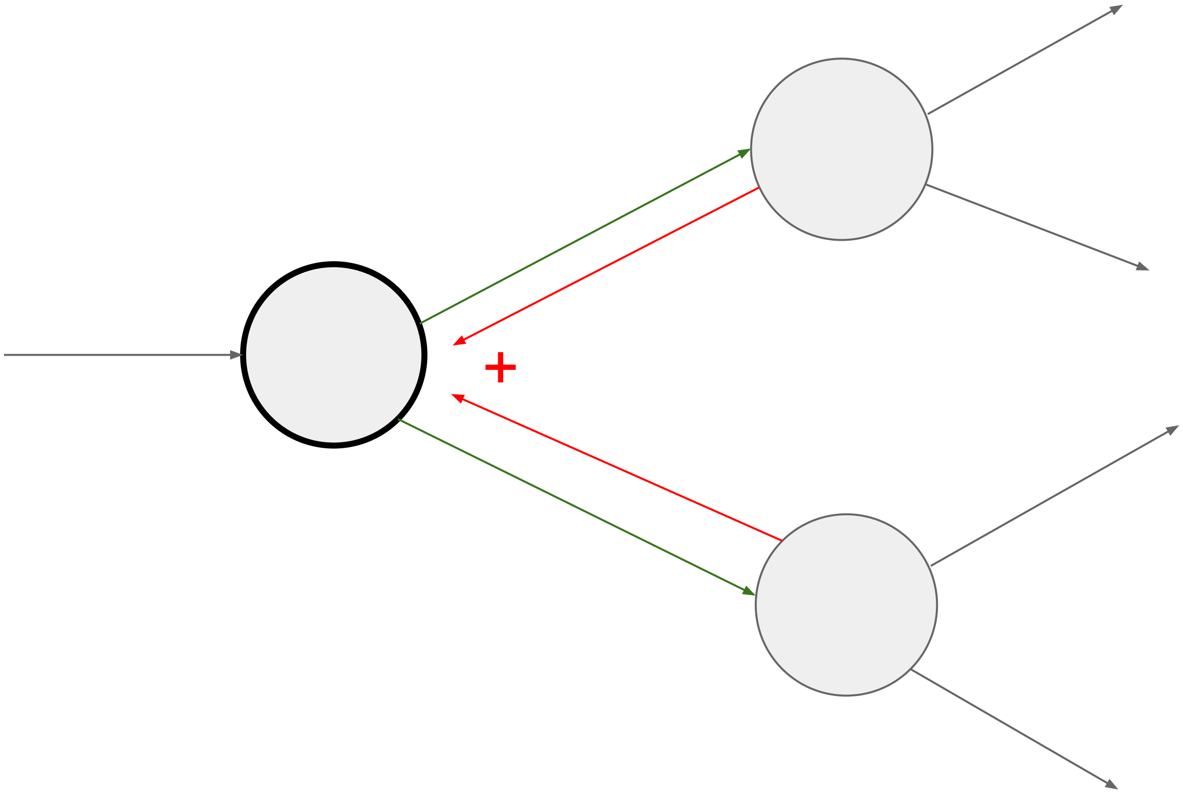
add gate: gradient distributor

max gate: gradient router 1(largest), 0, 0, 0

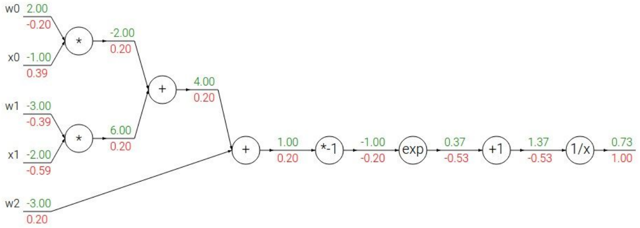
mul gate: gradient… “switcher”?



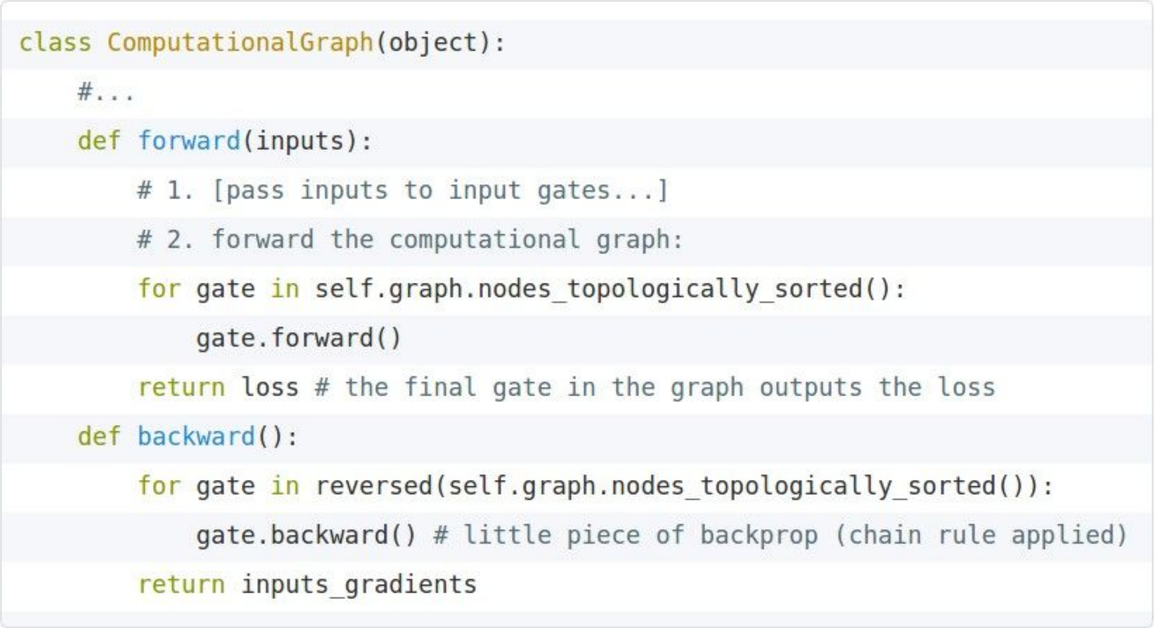
Gradients add at branches



Implementation: forward/backward API

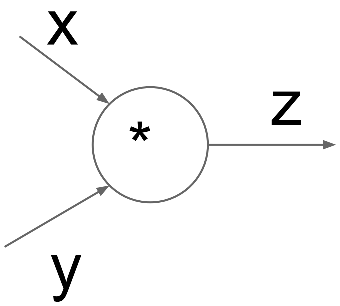


Graph (or Net) object. *(Rough psuedo code)*

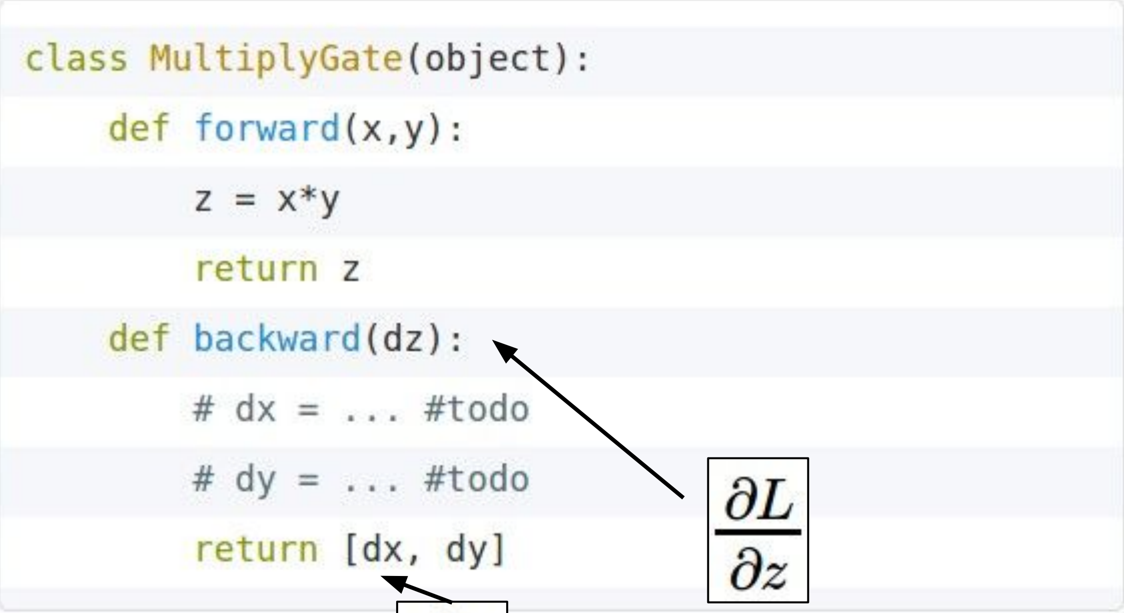


```
class ComputationalGraph(object):
    #...
    def forward(inputs):
        # 1. [pass inputs to input gates...]
        # 2. forward the computational graph:
        for gate in self.graph.nodes_topologically_sorted():
            gate.forward()
        return loss # the final gate in the graph outputs the loss
    def backward():
        for gate in reversed(self.graph.nodes_topologically_sorted()):
            gate.backward() # little piece of backprop (chain rule applied)
        return inputs_gradients
```

Implementation: forward/backward API



(x,y,z are scalars)

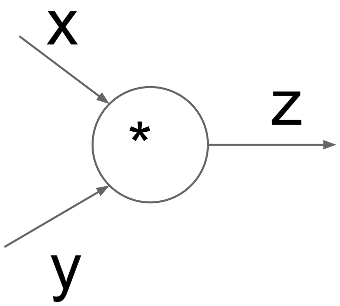


```
class MultiplyGate(object):
    def forward(x,y):
        z = x*y
        return z
    def backward(dz):
        # dx = ... #todo
        # dy = ... #todo
        return [dx, dy]
```

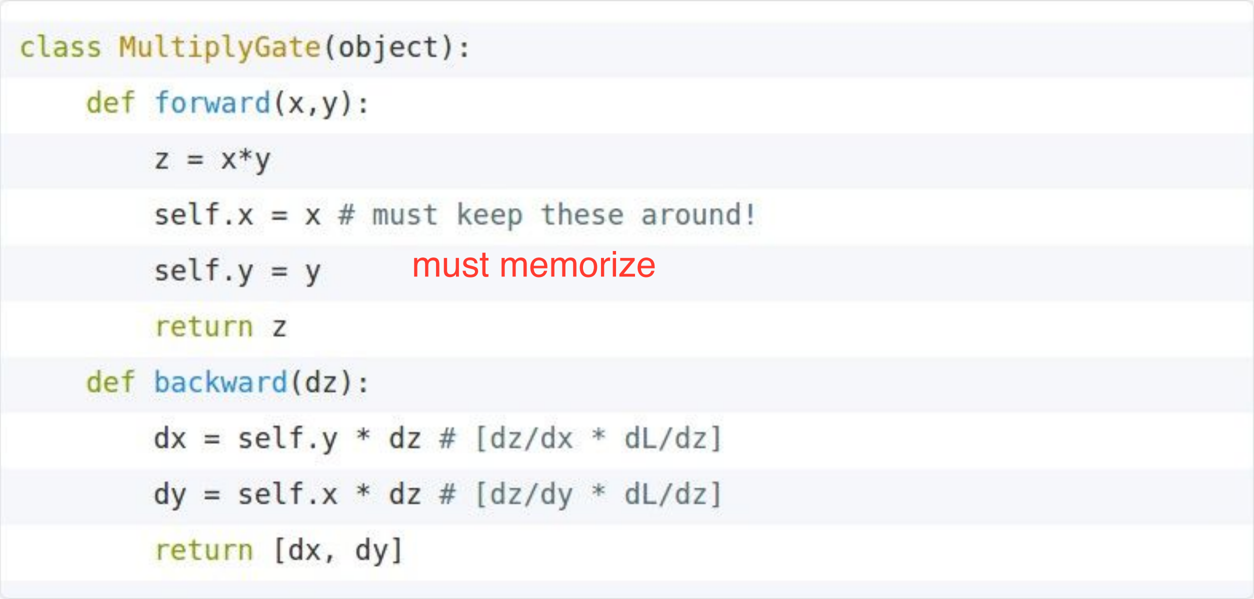
$$\boxed{\frac{\partial L}{\partial z}}$$

$$\boxed{\frac{\partial L}{\partial x}}$$

Implementation: forward/backward API



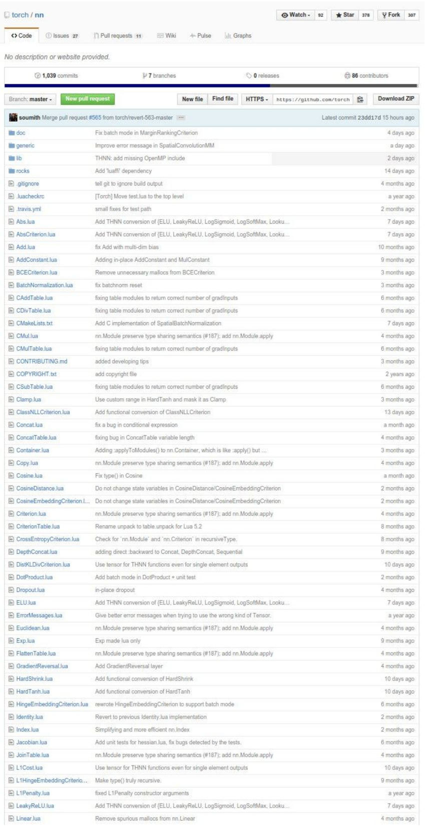
(x,y,z are scalars)



```
class MultiplyGate(object):
    def forward(x,y):
        z = x*y
        self.x = x # must keep these around!
        self.y = y
        return z
    def backward(dz):
        dx = self.y * dz # [dz/dx * dL/dz]
        dy = self.x * dz # [dz/dy * dL/dz]
        return [dx, dy]
```

must memorize

Example: Torch Layers

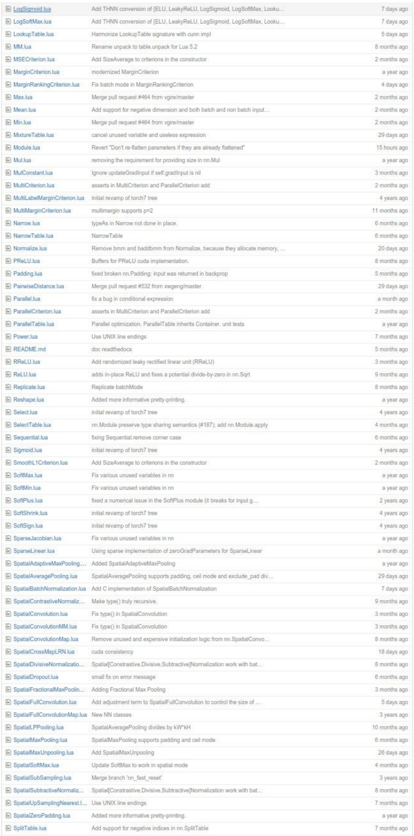


No description or website provided.

1,039 commits | 7 branches | 0 releases | 86 contributors

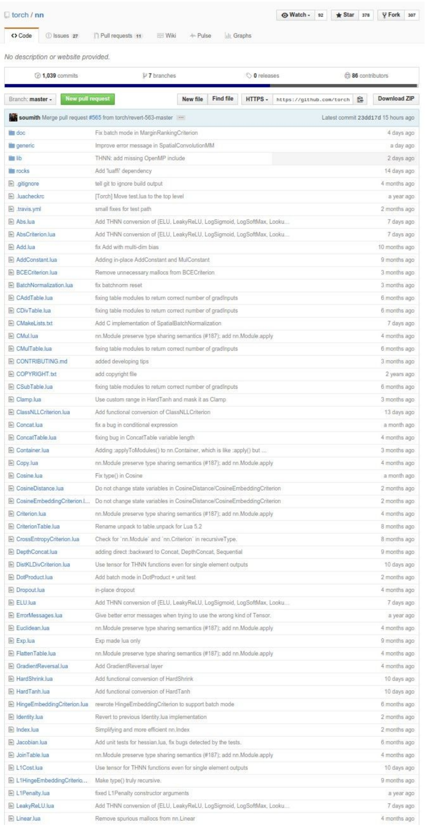
soumith Merge pull request #565 from torch/revert-563-master … Latest commit 23dd17d 15 hours ago

doc	Fix batch mode in MarginRankingCriterion	4 days ago
generic	Improve error message in SpatialConvolutionMM	a day ago
lib	THNN: add missing OpenMP include	2 days ago
rocks	Add 'luaffi' dependency	14 days ago
.gitignore	tell git to ignore build output	4 months ago
.luacheckrc	[Torch] Move test.lua to the top level	a year ago
.travis.yml	small fixes for test path	2 months ago
Abs.lua	Add THNN conversion of {ELU, LeakyReLU, LogSigmoid, LogSoftMax, Looku...	7 days ago
AbsCriterion.lua	Add THNN conversion of {ELU, LeakyReLU, LogSigmoid, LogSoftMax, Looku...	7 days ago
Add.lua	fix Add with multi-dim bias	10 months ago
AddConstant.lua	Adding in-place AddConstant and MulConstant	9 months ago
BCECriterion.lua	Remove unnecessary mallocs from BCECriterion	3 months ago
BatchNormalization.lua	fix batchnorm reset	3 months ago
CAddTable.lua	fixing table modules to return correct number of gradInputs	6 months ago
CDivTable.lua	fixing table modules to return correct number of gradInputs	6 months ago
CMakeLists.txt	Add C implementation of SpatialBatchNormalization	7 days ago
CMul.lua	nn.Module preserve type sharing semantics (#187); add nn.Module.apply	4 months ago
CMulTable.lua	fixing table modules to return correct number of gradInputs	6 months ago
CONTRIBUTING.md	added developing tips	3 months ago
COPYRIGHT.txt	add copyright file	2 years ago
CSubTable.lua	fixing table modules to return correct number of gradInputs	6 months ago
Clamp.lua	Use custom range in HardTanh and mask it as Clamp	3 months ago
ClassNLLCriterion.lua	Add functional conversion of ClassNLLCriterion	13 days ago
Concat.lua	fix a bug in conditional expression	a month ago
ConcatTable.lua	fixing bug in ConcatTable variable length	4 months ago
Container.lua	Adding :applyToModules() to nn.Container, which is like :apply() but ...	3 months ago
Copy.lua	nn.Module preserve type sharing semantics (#187); add nn.Module.apply	4 months ago
Cosine.lua	Fix type() in Cosine	a month ago
CosineDistance.lua	Do not change state variables in CosineDistance/CosineEmbeddingCriterion	2 months ago
CosineEmbeddingCriterion.l...	Do not change state variables in CosineDistance/CosineEmbeddingCriterion	2 months ago
Criterion.lua	nn.Module preserve type sharing semantics (#187); add nn.Module.apply	4 months ago
CriterionTable.lua	Rename unpack to table.unpack for Lua 5.2	8 months ago
CrossEntropyCriterion.lua	Check for \`nn.Module\` and \`nn.Criterion\` in recursiveType.	8 months ago
DepthConcat.lua	adding direct :backward to Concat, DepthConcat, Sequential	9 months ago
DistKLDivCriterion.lua	Use tensor for THNN functions even for single element outputs	10 days ago
DotProduct.lua	Add batch mode in DotProduct + unit test	2 months ago
Dropout.lua	in-place dropout	4 months ago
ELU.lua	Add THNN conversion of {ELU, LeakyReLU, LogSigmoid, LogSoftMax, Looku...	7 days ago
ErrorMessages.lua	Give better error messages when trying to use the wrong kind of Tensor.	a year ago
Euclidean.lua	nn.Module preserve type sharing semantics (#187); add nn.Module.apply	4 months ago
Exp.lua	Exp made lua only	9 months ago
FlattenTable.lua	nn.Module preserve type sharing semantics (#187); add nn.Module.apply	4 months ago
GradientReversal.lua	Add GradientReversal layer	4 months ago
HardShrink.lua	Add functional conversion of HardShrink	10 days ago
HardTanh.lua	Add functional conversion of HardTanh	10 days ago
HingeEmbeddingCriterion.lua	rewrote HingeEmbeddingCriterion to support batch mode	6 months ago
Identity.lua	Revert to previous Identity.lua implementation	2 months ago
Index.lua	Simplifying and more efficient nn.Index	2 months ago
Jacobian.lua	Add unit tests for hessian.lua, fix bugs detected by the tests.	6 months ago
JoinTable.lua	nn.Module preserve type sharing semantics (#187); add nn.Module.apply	4 months ago
L1Cost.lua	Use tensor for THNN functions even for single element outputs	10 days ago
L1HingeEmbeddingCriterio...	Make type() truly recursive.	9 months ago
L1Penalty.lua	fixed L1Penalty constructor arguments	a year ago
LeakyReLU.lua	Add THNN conversion of {ELU, LeakyReLU, LogSigmoid, LogSoftMax, Looku...	7 days ago
Linear.lua	Remove spurious mallocs from nn.Linear	4 months ago



LogSigmoid.lua	Add THNN conversion of {ELU, LeakyReLU, LogSigmoid, LogSoftMax, Looku...	7 days ago
LogSoftMax.lua	Add THNN conversion of {ELU, LeakyReLU, LogSigmoid, LogSoftMax, Looku...	7 days ago
LookupTable.lua	Harmonize LookupTable signature with cunn impl	5 days ago
MM.lua	Rename unpack to table.unpack for Lua 5.2	8 months ago
MSECriterion.lua	Add SizeAverage to criterions in the constructor	2 months ago
MarginCriterion.lua	modernized MarginCriterion	a year ago
MarginRankingCriterion.lua	Fix batch mode in MarginRankingCriterion	4 days ago
Max.lua	Merge pull request #464 from vgire/master	2 months ago
Mean.lua	Add support for negative dimension and both batch and non batch input...	2 months ago
Min.lua	Merge pull request #464 from vgire/master	2 months ago
MixtureTable.lua	cancel unused variable and useless expression	29 days ago
Module.lua	Revert "Don't re-flatten parameters if they are already flattened"	15 hours ago
Mul.lua	removing the requirement for providing size in nn.Mul	a year ago
MulConstant.lua	Ignore updateGradInput if self.gradInput is nil	3 months ago
MultiCriterion.lua	asserts in MultiCriterion and ParallelCriterion add	2 months ago
MultiLabelMarginCriterion.lua	initial revamp of torch7 tree	4 years ago
MultiMarginCriterion.lua	multimargin supports p=2	11 months ago
Narrow.lua	typeAs in Narrow not done in place.	6 months ago
NarrowTable.lua	NarrowTable	6 months ago
Normalize.lua	Remove bmm and baddbmm from Normalize, because they allocate memory, ...	20 days ago
PReLU.lua	Buffers for PReLU cuda implementation.	8 months ago
Padding.lua	fixed broken nn.Padding: input was returned in backprop	5 months ago
PairwiseDistance.lua	Merge pull request #532 from xwgeng/master	29 days ago
Parallel.lua	fix a bug in conditional expression	a month ago
ParallelCriterion.lua	asserts in MultiCriterion and ParallelCriterion add	2 months ago
ParallelTable.lua	Parallel optimization. ParallelTable inherits Container. unit tests	a year ago
Power.lua	Use UNIX line endings	7 months ago
README.md	doc readthedocs	5 months ago
RReLU.lua	Add randomized leaky rectified linear unit (RReLU)	3 months ago
ReLU.lua	adds in-place ReLU and fixes a potential divide-by-zero in nn.Sqrt	9 months ago
Replicate.lua	Replicate batchMode	8 months ago
Reshape.lua	Added more informative pretty-printing.	a year ago
Select.lua	initial revamp of torch7 tree	4 years ago
SelectTable.lua	nn.Module preserve type sharing semantics (#187); add nn.Module.apply	4 months ago
Sequential.lua	fixing Sequential.remove corner case	6 months ago
Sigmoid.lua	initial revamp of torch7 tree	4 years ago
SmoothL1Criterion.lua	Add SizeAverage to criterions in the constructor	2 months ago
SoftMax.lua	Fix various unused variables in nn	a year ago
SoftMin.lua	Fix various unused variables in nn	a year ago
SoftPlus.lua	fixed a numerical issue in the SoftPlus module (it breaks for input g...	2 years ago
SoftShrink.lua	initial revamp of torch7 tree	4 years ago
SoftSign.lua	initial revamp of torch7 tree	4 years ago
SparseJacobian.lua	Fix various unused variables in nn	a year ago
SparseLinear.lua	Using sparse implementation of zeroGradParameters for SparseLinear	a month ago
SpatialAdaptiveMaxPooling....	Added SpatialAdaptiveMaxPooling	a year ago
SpatialAveragePooling.lua	SpatialAveragePooling supports padding, ceil mode and exclude_pad div...	29 days ago
SpatialBatchNormalization.lua	Add C implementation of SpatialBatchNormalization	7 days ago
SpatialContrastiveNormaliz...	Make type() truly recursive.	9 months ago
SpatialConvolution.lua	Fix type() in SpatialConvolution	3 months ago
SpatialConvolutionMM.lua	Fix type() in SpatialConvolution	3 months ago
SpatialConvolutionMap.lua	Remove unused and expensive initialization logic from nn.SpatialConvo...	8 months ago
SpatialCrossMapLRN.lua	cuda consistency	18 days ago
SpatialDivisiveNormalizatio...	Spatial[Constrastive,Divisive,Subtractive]Normalization work with bat...	8 months ago
SpatialDropout.lua	small fix on error message	6 months ago
SpatialFractionalMaxPoolin...	Adding Fractional Max Pooling	3 months ago
SpatialFullConvolution.lua	Add adjustment term to SpatialFullConvolution to control the size of ...	5 days ago
SpatialFullConvolutionMap.lua	New NN classes	3 years ago
SpatialLPPooling.lua	SpatialAveragePooling divides by kW*kH	10 months ago
SpatialMaxPooling.lua	SpatialMaxPooling supports padding and ceil mode	6 months ago
SpatialMaxUnpooling.lua	Add SpatialMaxUnpooling	26 days ago
SpatialSoftMax.lua	Update SoftMax to work in spatial mode	4 months ago
SpatialSubSampling.lua	Merge branch 'nn_fast_reset'	3 years ago
SpatialSubtractiveNormaliz...	Spatial[Constrastive,Divisive,Subtractive]Normalization work with bat...	8 months ago
SpatialUpSamplingNearest.l...	Use UNIX line endings	7 months ago
SpatialZeroPadding.lua	Added more informative pretty-printing.	a year ago
SplitTable.lua	Add support for negative indices in nn.SplitTable	7 months ago

Example: Torch Layers



torch / nn

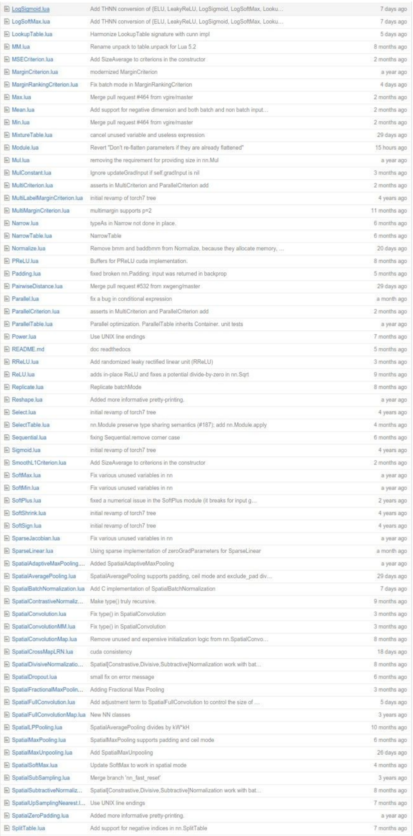
Watch 92 | Star 378 | Fork 307

No description or website provided.

1,039 commits | 7 branches | 0 releases | 86 contributors

soumith Merge pull request #565 from torch/revert-563-master ... Latest commit 23dd17d 15 hours ago

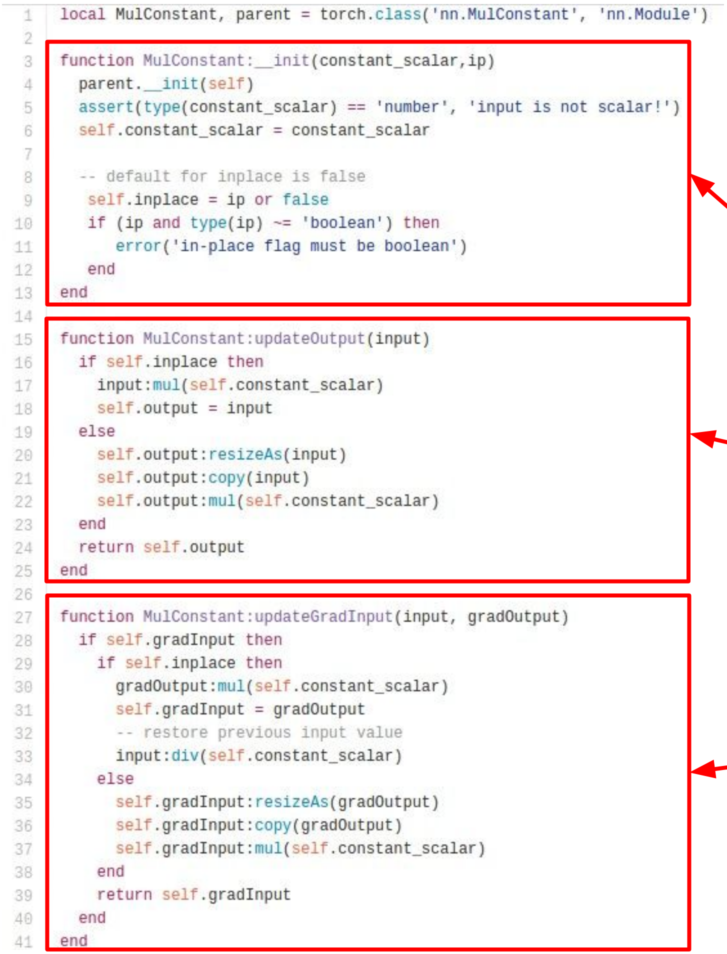
doc	Fix batch mode in MarginRankingCriterion	4 days ago
generic	Improve error message in SpatialConvolutionMM	a day ago
lib	THNN: add missing OpenMP include	2 days ago
rocks	Add 'luaffi' dependency	14 days ago
.gitignore	tell git to ignore build output	4 months ago
.luacheckrc	[Torch] Move test.lua to the top level	a year ago
.travis.yml	small fixes for test path	2 months ago
Abs.lua	Add THNN conversion of {ELU, LeakyReLU, LogSigmoid, LogSoftMax, Looku...	7 days ago
AbsCriterion.lua	Add THNN conversion of {ELU, LeakyReLU, LogSigmoid, LogSoftMax, Looku...	7 days ago
Add.lua	fix Add with multi-dim bias	10 months ago
AddConstant.lua	Adding in-place AddConstant and MulConstant	9 months ago
BCECriterion.lua	Remove unnecessary mallocs from BCECriterion	3 months ago
BatchNormalization.lua	fix batchnorm reset	3 months ago
CAddTable.lua	fixing table modules to return correct number of gradInputs	6 months ago
CDivTable.lua	fixing table modules to return correct number of gradInputs	6 months ago
CMakeLists.txt	Add C implementation of SpatialBatchNormalization	7 days ago
CMul.lua	nn.Module preserve type sharing semantics (#187); add nn.Module.apply	4 months ago
CMulTable.lua	fixing table modules to return correct number of gradInputs	6 months ago
CONTRIBUTING.md	added developing tips	3 months ago
COPYRIGHT.txt	add copyright file	2 years ago
CSubTable.lua	fixing table modules to return correct number of gradInputs	6 months ago
Clamp.lua	Use custom range in HardTanh and mask it as Clamp	3 months ago
ClassNLLCriterion.lua	Add functional conversion of ClassNLLCriterion	13 days ago
Concat.lua	fix a bug in conditional expression	a month ago
ConcatTable.lua	fixing bug in ConcatTable variable length	4 months ago
Container.lua	Adding :applyToModules() to nn.Container, which is like :apply() but ...	3 months ago
Copy.lua	nn.Module preserve type sharing semantics (#187); add nn.Module.apply	4 months ago
Cosine.lua	Fix type() in Cosine	a month ago
CosineDistance.lua	Do not change state variables in CosineDistance/CosineEmbeddingCriterion	2 months ago
CosineEmbeddingCriterion.l...	Do not change state variables in CosineDistance/CosineEmbeddingCriterion	2 months ago
Criterion.lua	nn.Module preserve type sharing semantics (#187); add nn.Module.apply	4 months ago
CriterionTable.lua	Rename unpack to table.unpack for Lua 5.2	8 months ago
CrossEntropyCriterion.lua	Check for `nn.Module` and `nn.Criterion` in recursiveType.	8 months ago
DepthConcat.lua	adding direct :backward to Concat, DepthConcat, Sequential	9 months ago
DistKLDivCriterion.lua	Use tensor for THNN functions even for single element outputs	10 days ago
DotProduct.lua	Add batch mode in DotProduct + unit test	2 months ago
Dropout.lua	in-place dropout	4 months ago
ELU.lua	Add THNN conversion of {ELU, LeakyReLU, LogSigmoid, LogSoftMax, Looku...	7 days ago
ErrorMessages.lua	Give better error messages when trying to use the wrong kind of Tensor.	a year ago
Euclidean.lua	nn.Module preserve type sharing semantics (#187); add nn.Module.apply	4 months ago
Exp.lua	Exp made lua only	9 months ago
FlattenTable.lua	nn.Module preserve type sharing semantics (#187); add nn.Module.apply	4 months ago
GradientReversal.lua	Add GradientReversal layer	4 months ago
HardShrink.lua	Add functional conversion of HardShrink	10 days ago
HardTanh.lua	Add functional conversion of HardTanh	10 days ago
HingeEmbeddingCriterion.lua	rewrote HingeEmbeddingCriterion to support batch mode	6 months ago
Identity.lua	Revert to previous Identity.lua implementation	2 months ago
Index.lua	Simplifying and more efficient nn.Index	2 months ago
Jacobian.lua	Add unit tests for hessian.lua, fix bugs detected by the tests.	6 months ago
JoinTable.lua	nn.Module preserve type sharing semantics (#187); add nn.Module.apply	4 months ago
L1Cost.lua	Use tensor for THNN functions even for single element outputs	10 days ago
L1HingeEmbeddingCriterio...	Make type() truly recursive.	9 months ago
L1Penalty.lua	fixed L1Penalty constructor arguments	a year ago
LeakyReLU.lua	Add THNN conversion of {ELU, LeakyReLU, LogSigmoid, LogSoftMax, Looku...	7 days ago
Linear.lua	Remove spurious mallocs from nn.Linear	4 months ago



LogSigmoid.lua	Add THNN conversion of {ELU, LeakyReLU, LogSigmoid, LogSoftMax, Looku...	7 days ago
LogSoftMax.lua	Add THNN conversion of {ELU, LeakyReLU, LogSigmoid, LogSoftMax, Looku...	7 days ago
LookupTable.lua	Harmonize LookupTable signature with cunn impl	5 days ago
MM.lua	Rename unpack to table.unpack for Lua 5.2	8 months ago
MSECriterion.lua	Add SizeAverage to criterions in the constructor	2 months ago
MarginCriterion.lua	modernized MarginCriterion	a year ago
MarginRankingCriterion.lua	Fix batch mode in MarginRankingCriterion	4 days ago
Max.lua	Merge pull request #464 from vgire/master	2 months ago
Mean.lua	Add support for negative dimension and both batch and non batch input...	2 months ago
Min.lua	Merge pull request #464 from vgire/master	2 months ago
MixtureTable.lua	cancel unused variable and useless expression	29 days ago
Module.lua	Revert "Don't re-flatten parameters if they are already flattened"	15 hours ago
Mul.lua	removing the requirement for providing size in nn.Mul	a year ago
MulConstant.lua	Ignore updateGradInput if self.gradInput is nil	3 months ago
MultiCriterion.lua	asserts in MultiCriterion and ParallelCriterion add	2 months ago
MultiLabelMarginCriterion.lua	initial revamp of torch7 tree	4 years ago
MultiMarginCriterion.lua	multimargin supports p=2	11 months ago
Narrow.lua	typeAs in Narrow not done in place.	6 months ago
NarrowTable.lua	NarrowTable	6 months ago
Normalize.lua	Remove bmm and baddbmm from Normalize, because they allocate memory, ...	20 days ago
PReLU.lua	Buffers for PReLU cuda implementation.	8 months ago
Padding.lua	fixed broken nn.Padding: input was returned in backprop	5 months ago
PairwiseDistance.lua	Merge pull request #532 from xwgeng/master	29 days ago
Parallel.lua	fix a bug in conditional expression	a month ago
ParallelCriterion.lua	asserts in MultiCriterion and ParallelCriterion add	2 months ago
ParallelTable.lua	Parallel optimization. ParallelTable inherits Container. unit tests	a year ago
Power.lua	Use UNIX line endings	7 months ago
README.md	doc readthedocs	5 months ago
RReLU.lua	Add randomized leaky rectified linear unit (RReLU)	3 months ago
ReLU.lua	adds in-place ReLU and fixes a potential divide-by-zero in nn.Sqrt	9 months ago
Replicate.lua	Replicate batchMode	8 months ago
Reshape.lua	Added more informative pretty-printing.	a year ago
Select.lua	initial revamp of torch7 tree	4 years ago
SelectTable.lua	nn.Module preserve type sharing semantics (#187); add nn.Module.apply	4 months ago
Sequential.lua	fixing Sequential.remove corner case	6 months ago
Sigmoid.lua	initial revamp of torch7 tree	4 years ago
SmoothL1Criterion.lua	Add SizeAverage to criterions in the constructor	2 months ago
SoftMax.lua	Fix various unused variables in nn	a year ago
SoftMin.lua	Fix various unused variables in nn	a year ago
SoftPlus.lua	fixed a numerical issue in the SoftPlus module (it breaks for input g...	2 years ago
SoftShrink.lua	initial revamp of torch7 tree	4 years ago
SoftSign.lua	initial revamp of torch7 tree	4 years ago
SparseJacobian.lua	Fix various unused variables in nn	a year ago
SparseLinear.lua	Using sparse implementation of zeroGradParameters for SparseLinear	a month ago
SpatialAdaptiveMaxPooling....	Added SpatialAdaptiveMaxPooling	a year ago
SpatialAveragePooling.lua	SpatialAveragePooling supports padding, ceil mode and exclude_pad div...	29 days ago
SpatialBatchNormalization.lua	Add C implementation of SpatialBatchNormalization	7 days ago
SpatialContrastiveNormaliz...	Make type() truly recursive.	9 months ago
SpatialConvolution.lua	Fix type() in SpatialConvolution	3 months ago
SpatialConvolutionMM.lua	Fix type() in SpatialConvolution	3 months ago
SpatialConvolutionMap.lua	Remove unused and expensive initialization logic from nn.SpatialConvo...	8 months ago
SpatialCrossMapLRN.lua	cuda consistency	18 days ago
SpatialDivisiveNormalizatio...	Spatial[Constrastive,Divisive,Subtractive]Normalization work with bat...	8 months ago
SpatialDropout.lua	small fix on error message	6 months ago
SpatialFractionalMaxPoolin...	Adding Fractional Max Pooling	3 months ago
SpatialFullConvolution.lua	Add adjustment term to SpatialFullConvolution to control the size of ...	5 days ago
SpatialFullConvolutionMap.lua	New NN classes	3 years ago
SpatialLPPooling.lua	SpatialAveragePooling divides by kW*kH	10 months ago
SpatialMaxPooling.lua	SpatialMaxPooling supports padding and ceil mode	6 months ago
SpatialMaxUnpooling.lua	Add SpatialMaxUnpooling	26 days ago
SpatialSoftMax.lua	Update SoftMax to work in spatial mode	4 months ago
SpatialSubSampling.lua	Merge branch 'nn_fast_reset'	3 years ago
SpatialSubtractiveNormaliz...	Spatial[Constrastive,Divisive,Subtractive]Normalization work with bat...	8 months ago
SpatialUpSamplingNearest.l...	Use UNIX line endings	7 months ago
SpatialZeroPadding.lua	Added more informative pretty-printing.	a year ago
SplitTable.lua	Add support for negative indices in nn.SplitTable	7 months ago

=

Example: Torch MulConstant



```lua
local MulConstant, parent = torch.class('nn.MulConstant', 'nn.Module')

function MulConstant:__init(constant_scalar,ip)
  parent.__init(self)
  assert(type(constant_scalar) == 'number', 'input is not scalar!')
  self.constant_scalar = constant_scalar

  -- default for inplace is false
   self.inplace = ip or false
   if (ip and type(ip) ~= 'boolean') then
      error('in-place flag must be boolean')
   end
end

function MulConstant:updateOutput(input)
  if self.inplace then
    input:mul(self.constant_scalar)
    self.output = input
  else
    self.output:resizeAs(input)
    self.output:copy(input)
    self.output:mul(self.constant_scalar)
  end
  return self.output
end

function MulConstant:updateGradInput(input, gradOutput)
  if self.gradInput then
    if self.inplace then
      gradOutput:mul(self.constant_scalar)
      self.gradInput = gradOutput
      -- restore previous input value
      input:div(self.constant_scalar)
    else
      self.gradInput:resizeAs(gradOutput)
      self.gradInput:copy(gradOutput)
      self.gradInput:mul(self.constant_scalar)
    end
    return self.gradInput
  end
end
```

$$f(X) = aX$$

scalar

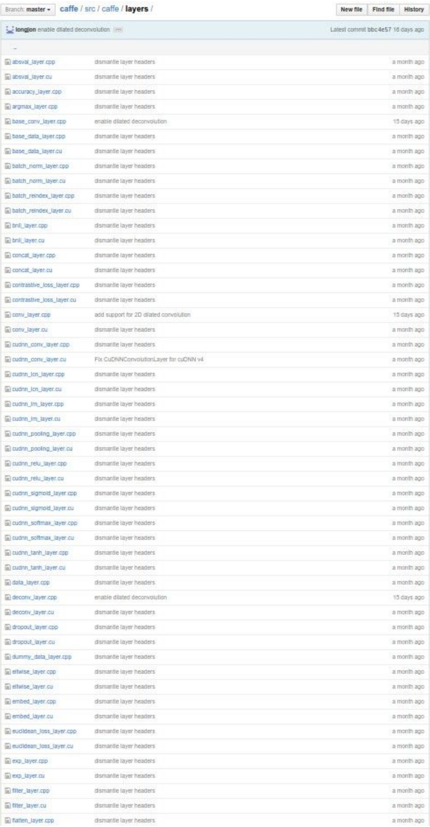
tensor

initialization

forward()

backward()

Example: Caffe Layers



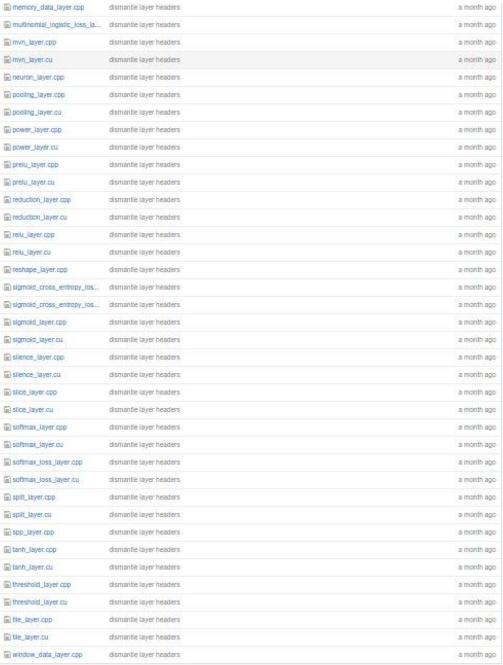
Branch: master ▾ caffe / src / caffe / layers /

New file | Find file | History

longjon enable dilated deconvolution … Latest commit bbc4e57 16 days ago

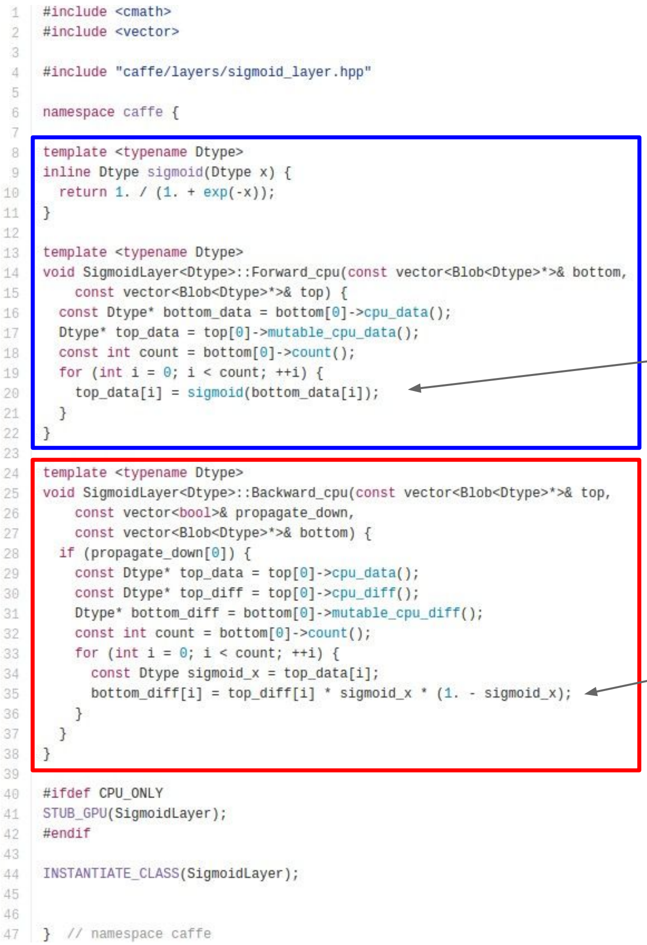
..

absval_layer.cpp	dismantle layer headers	a month ago
absval_layer.cu	dismantle layer headers	a month ago
accuracy_layer.cpp	dismantle layer headers	a month ago
argmax_layer.cpp	dismantle layer headers	a month ago
base_conv_layer.cpp	enable dilated deconvolution	15 days ago
base_data_layer.cpp	dismantle layer headers	a month ago
base_data_layer.cu	dismantle layer headers	a month ago
batch_norm_layer.cpp	dismantle layer headers	a month ago
batch_norm_layer.cu	dismantle layer headers	a month ago
batch_reindex_layer.cpp	dismantle layer headers	a month ago
batch_reindex_layer.cu	dismantle layer headers	a month ago
bnll_layer.cpp	dismantle layer headers	a month ago
bnll_layer.cu	dismantle layer headers	a month ago
concat_layer.cpp	dismantle layer headers	a month ago
concat_layer.cu	dismantle layer headers	a month ago
contrastive_loss_layer.cpp	dismantle layer headers	a month ago
contrastive_loss_layer.cu	dismantle layer headers	a month ago
conv_layer.cpp	add support for 2D dilated convolution	15 days ago
conv_layer.cu	dismantle layer headers	a month ago
cudnn_conv_layer.cpp	dismantle layer headers	a month ago
cudnn_conv_layer.cu	Fix CuDNNConvolutionLayer for cuDNN v4	a month ago
cudnn_lcn_layer.cpp	dismantle layer headers	a month ago
cudnn_lcn_layer.cu	dismantle layer headers	a month ago
cudnn_lrn_layer.cpp	dismantle layer headers	a month ago
cudnn_lrn_layer.cu	dismantle layer headers	a month ago
cudnn_pooling_layer.cpp	dismantle layer headers	a month ago
cudnn_pooling_layer.cu	dismantle layer headers	a month ago
cudnn_relu_layer.cpp	dismantle layer headers	a month ago
cudnn_relu_layer.cu	dismantle layer headers	a month ago
cudnn_sigmoid_layer.cpp	dismantle layer headers	a month ago
cudnn_sigmoid_layer.cu	dismantle layer headers	a month ago
cudnn_softmax_layer.cpp	dismantle layer headers	a month ago
cudnn_softmax_layer.cu	dismantle layer headers	a month ago
cudnn_tanh_layer.cpp	dismantle layer headers	a month ago
cudnn_tanh_layer.cu	dismantle layer headers	a month ago
data_layer.cpp	dismantle layer headers	a month ago
deconv_layer.cpp	enable dilated deconvolution	15 days ago
deconv_layer.cu	dismantle layer headers	a month ago
dropout_layer.cpp	dismantle layer headers	a month ago
dropout_layer.cu	dismantle layer headers	a month ago
dummy_data_layer.cpp	dismantle layer headers	a month ago
eltwise_layer.cpp	dismantle layer headers	a month ago
eltwise_layer.cu	dismantle layer headers	a month ago
embed_layer.cpp	dismantle layer headers	a month ago
embed_layer.cu	dismantle layer headers	a month ago
euclidean_loss_layer.cpp	dismantle layer headers	a month ago
euclidean_loss_layer.cu	dismantle layer headers	a month ago
exp_layer.cpp	dismantle layer headers	a month ago
exp_layer.cu	dismantle layer headers	a month ago
filter_layer.cpp	dismantle layer headers	a month ago
filter_layer.cu	dismantle layer headers	a month ago
flatten_layer.cpp	dismantle layer headers	a month ago



memory_data_layer.cpp	dismantle layer headers	a month ago
multinomial_logistic_loss_la…	dismantle layer headers	a month ago
mvn_layer.cpp	dismantle layer headers	a month ago
mvn_layer.cu	dismantle layer headers	a month ago
neuron_layer.cpp	dismantle layer headers	a month ago
pooling_layer.cpp	dismantle layer headers	a month ago
pooling_layer.cu	dismantle layer headers	a month ago
power_layer.cpp	dismantle layer headers	a month ago
power_layer.cu	dismantle layer headers	a month ago
prelu_layer.cpp	dismantle layer headers	a month ago
prelu_layer.cu	dismantle layer headers	a month ago
reduction_layer.cpp	dismantle layer headers	a month ago
reduction_layer.cu	dismantle layer headers	a month ago
relu_layer.cpp	dismantle layer headers	a month ago
relu_layer.cu	dismantle layer headers	a month ago
reshape_layer.cpp	dismantle layer headers	a month ago
sigmoid_cross_entropy_los…	dismantle layer headers	a month ago
sigmoid_cross_entropy_los…	dismantle layer headers	a month ago
sigmoid_layer.cpp	dismantle layer headers	a month ago
sigmoid_layer.cu	dismantle layer headers	a month ago
silence_layer.cpp	dismantle layer headers	a month ago
silence_layer.cu	dismantle layer headers	a month ago
slice_layer.cpp	dismantle layer headers	a month ago
slice_layer.cu	dismantle layer headers	a month ago
softmax_layer.cpp	dismantle layer headers	a month ago
softmax_layer.cu	dismantle layer headers	a month ago
softmax_loss_layer.cpp	dismantle layer headers	a month ago
softmax_loss_layer.cu	dismantle layer headers	a month ago
split_layer.cpp	dismantle layer headers	a month ago
split_layer.cu	dismantle layer headers	a month ago
spp_layer.cpp	dismantle layer headers	a month ago
tanh_layer.cpp	dismantle layer headers	a month ago
tanh_layer.cu	dismantle layer headers	a month ago
threshold_layer.cpp	dismantle layer headers	a month ago
threshold_layer.cu	dismantle layer headers	a month ago
tile_layer.cpp	dismantle layer headers	a month ago
tile_layer.cu	dismantle layer headers	a month ago
window_data_layer.cpp	dismantle layer headers	a month ago

Caffe Sigmoid Layer



```
#include <cmath>
#include <vector>

#include "caffe/layers/sigmoid_layer.hpp"

namespace caffe {

template <typename Dtype>
inline Dtype sigmoid(Dtype x) {
  return 1. / (1. + exp(-x));
}

template <typename Dtype>
void SigmoidLayer<Dtype>::Forward_cpu(const vector<Blob<Dtype>*>& bottom,
    const vector<Blob<Dtype>*>& top) {
  const Dtype* bottom_data = bottom[0]->cpu_data();
  Dtype* top_data = top[0]->mutable_cpu_data();
  const int count = bottom[0]->count();
  for (int i = 0; i < count; ++i) {
    top_data[i] = sigmoid(bottom_data[i]);
  }
}

template <typename Dtype>
void SigmoidLayer<Dtype>::Backward_cpu(const vector<Blob<Dtype>*>& top,
    const vector<bool>& propagate_down,
    const vector<Blob<Dtype>*>& bottom) {
  if (propagate_down[0]) {
    const Dtype* top_data = top[0]->cpu_data();
    const Dtype* top_diff = top[0]->cpu_diff();
    Dtype* bottom_diff = bottom[0]->mutable_cpu_diff();
    const int count = bottom[0]->count();
    for (int i = 0; i < count; ++i) {
      const Dtype sigmoid_x = top_data[i];
      bottom_diff[i] = top_diff[i] * sigmoid_x * (1. - sigmoid_x);
    }
  }
}

#ifdef CPU_ONLY
STUB_GPU(SigmoidLayer);
#endif

INSTANTIATE_CLASS(SigmoidLayer);


}  // namespace caffe
```

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

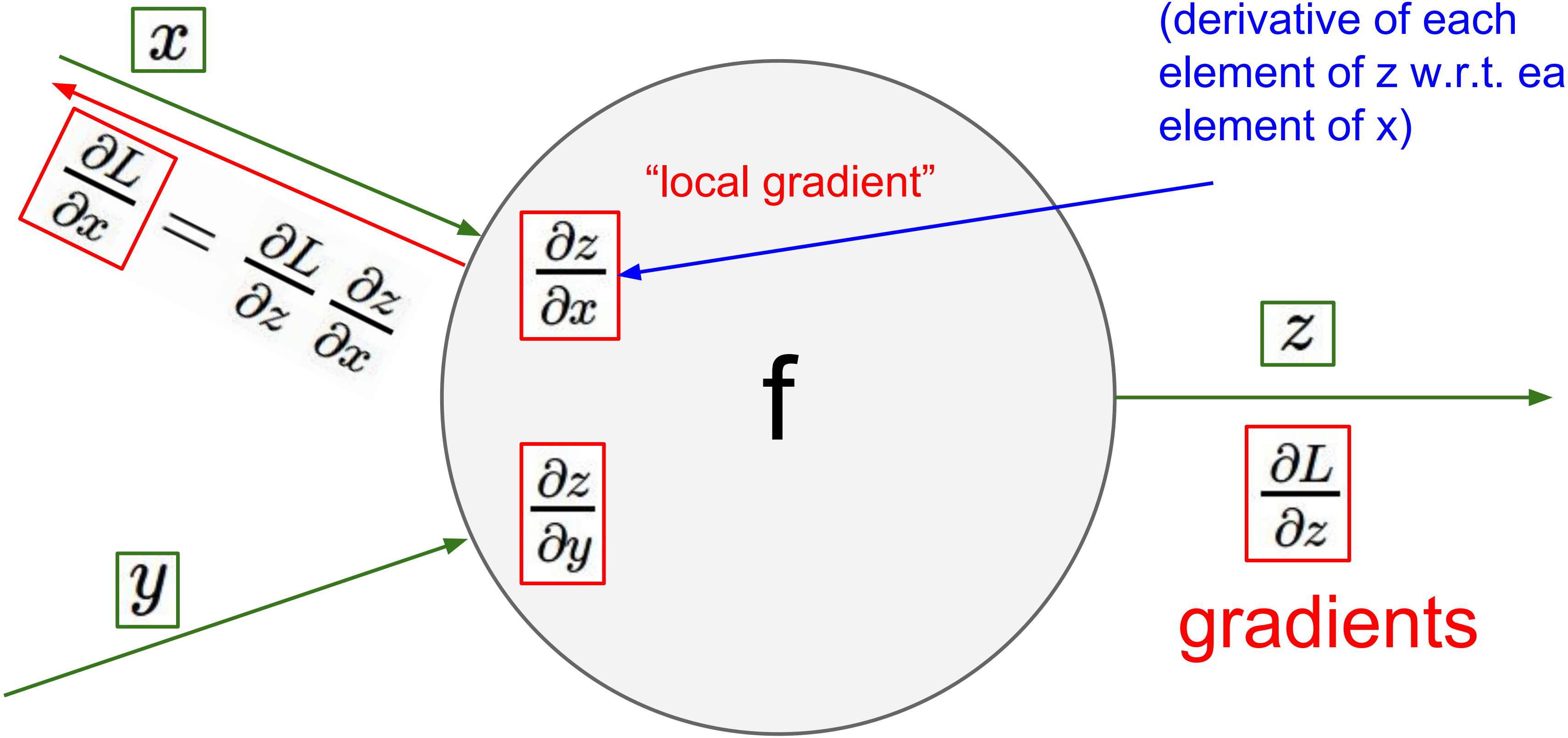
forward - backward - upgrade for W

$$\boxed{(1-\sigma(x))\sigma(x)} \quad \text{*top\_diff} \quad \text{(chain rule)}$$

Gradients for vectorized code

(x,y,z are now vectors)

This is now the **Jacobian matrix** (derivative of each element of z w.r.t. each element of x)



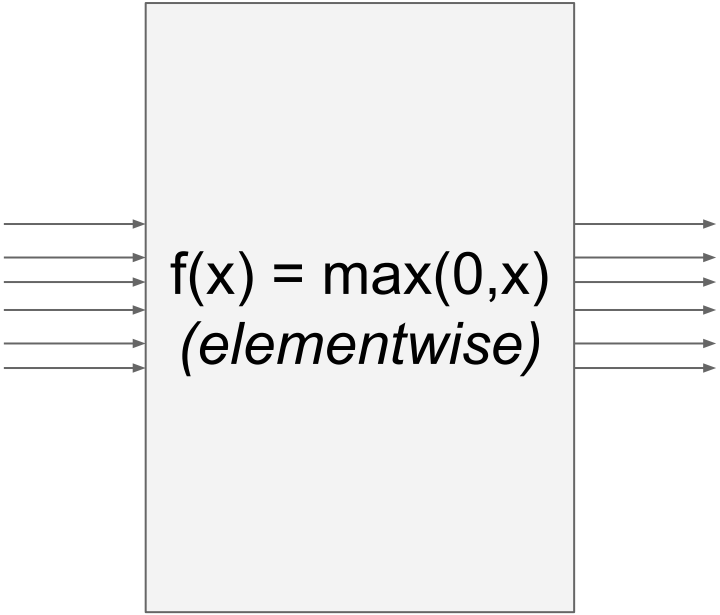
x
y
z
f
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
$\frac{\partial L}{\partial z}$

“local gradient”

gradients

Vectorized operations

4096-d
input vector



f(x) = max(0,x)
(elementwise)

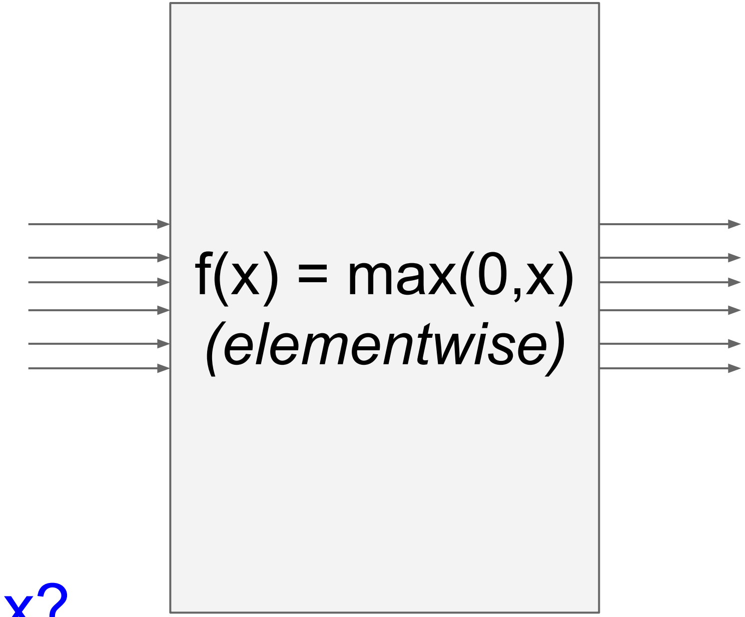
4096-d
output vector

Vectorized operations

$$\frac{\partial L}{\partial x} = \boxed{\frac{\partial f}{\partial x}} \frac{\partial L}{\partial f}$$

Jacobian matrix

4096-d
input vector



4096-d
output vector

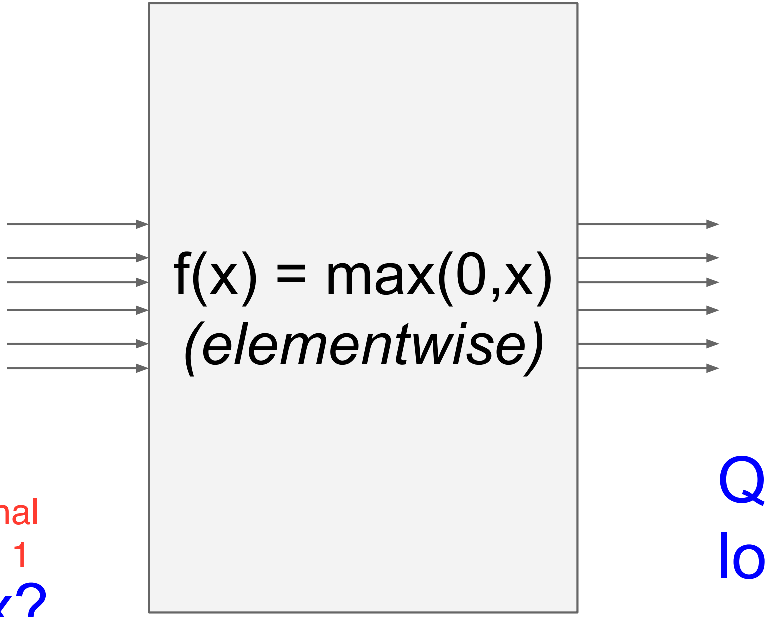
Q: what is the size of the Jacobian matrix?

Vectorized operations

$$\frac{\partial L}{\partial x} = \boxed{\frac{\partial f}{\partial x}} \frac{\partial L}{\partial f}$$

Jacobian matrix

4096-d
input vector



4096-d
output vector

Q: what is the size of the Jacobian matrix?
[4096 x 4096!]

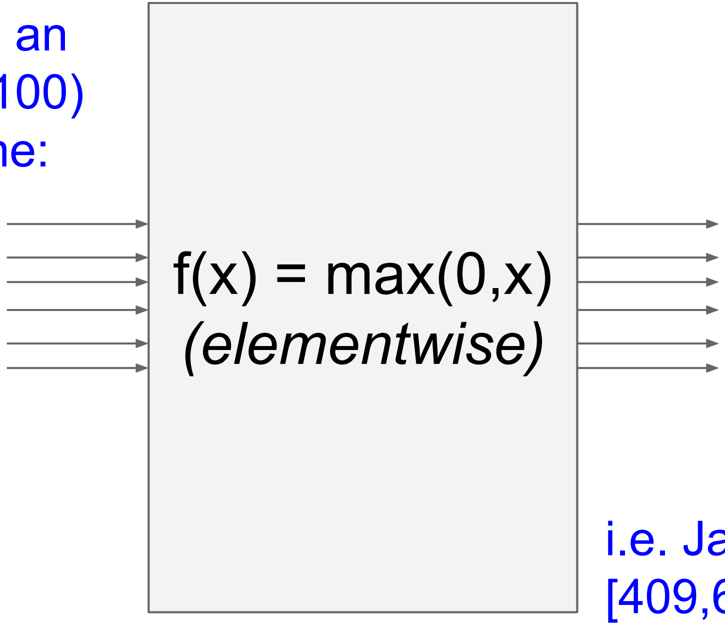
diagonal
0 and 1

Q2: what does it look like?

Vectorized operations

in practice we process an entire minibatch (e.g. 100) of examples at one time:

100 4096-d
input vectors



f(x) = max(0,x)
(elementwise)

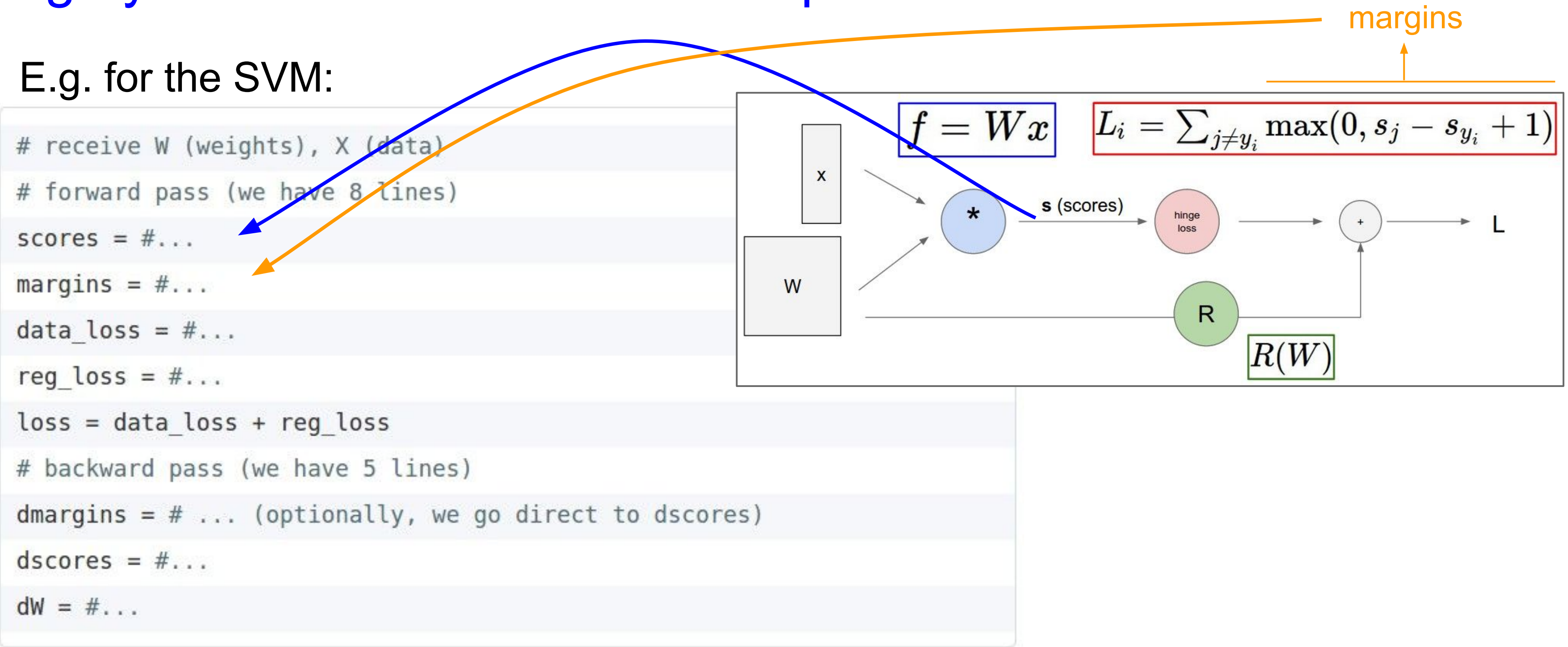
100 4096-d
output vectors

i.e. Jacobian would technically be a [409,600 x 409,600] matrix :\

Assignment: Writing SVM/Softmax

Stage your forward/backward computation!

E.g. for the SVM:



```
# receive W (weights), X (data)
# forward pass (we have 8 lines)
scores = #...
margins = #...
data_loss = #...
reg_loss = #...
loss = data_loss + reg_loss
# backward pass (we have 5 lines)
dmargins = # ... (optionally, we go direct to dscores)
dscores = #...
dW = #...
```

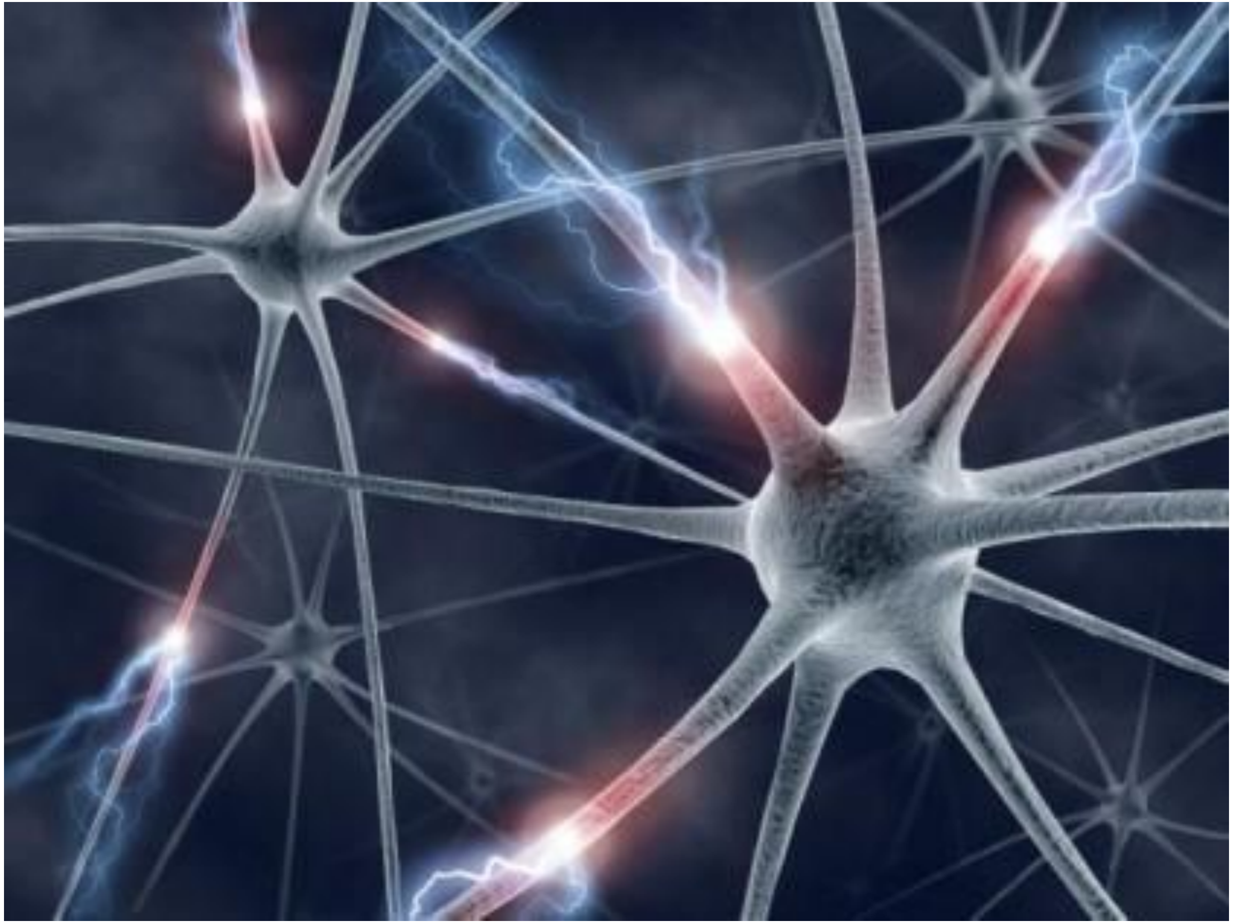
$f = Wx$

$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$

$R(W)$

Summary so far

- neural nets will be very large: no hope of writing down gradient formula by hand for all parameters
- **backpropagation** = recursive application of the chain rule along a computational graph to compute the gradients of all inputs/parameters/intermediates
- implementations maintain a graph structure, where the nodes implement the **forward**() / **backward**() API.
- **forward**: compute result of an operation and save any intermediates needed for gradient computation in memory
- **backward**: apply the chain rule to compute the gradient of the loss function with respect to the inputs.



Neural Network: without the brain stuff

(**Before**) Linear score function: $f = Wx$

Neural Network: without the brain stuff

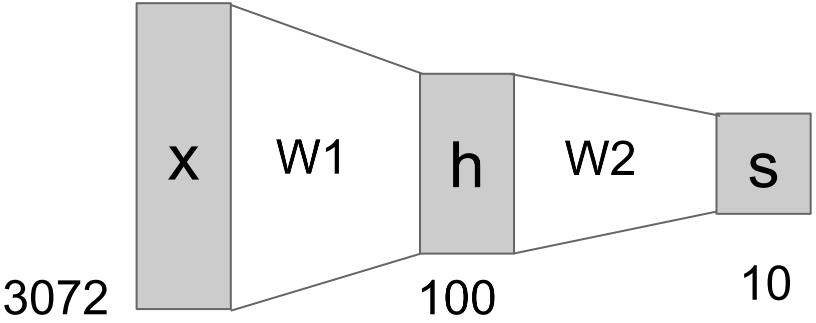
(**Before**) Linear score function: $f = Wx$

(**Now**) 2-layer Neural Network $f = W_2 \max(0, W_1 x)$

Neural Network: without the brain stuff

(**Before**) Linear score function: $f = Wx$

(**Now**) 2-layer Neural Network $f = W_2 \max(0, W_1 x)$

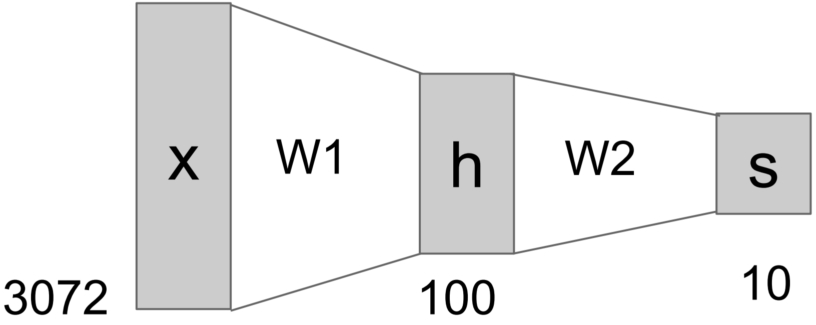


Neural Network: without the brain stuff

plane
car
bird
cat
deer
dog
frog
horse
ship
truck

(**Before**) Linear score function: $f = Wx$

(**Now**) 2-layer Neural Network $f = W_2 \max(0, W_1 x)$



Neural Network: without the brain stuff

(**Before**) Linear score function:

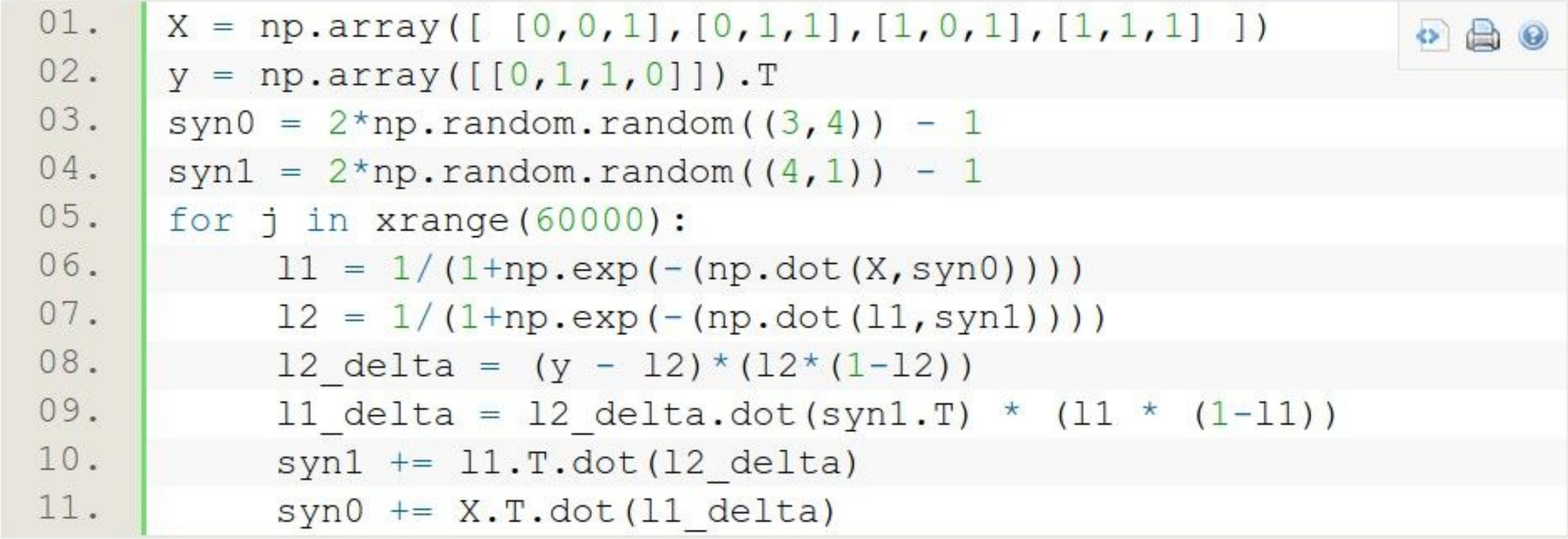
$$f = Wx$$

(**Now**) 2-layer Neural Network
or 3-layer Neural Network

$$f = W_2 \max(0, W_1 x)$$

$$f = W_3 \max(0, W_2 \max(0, W_1 x))$$

Full implementation of training a 2-layer Neural Network needs ~11 lines:

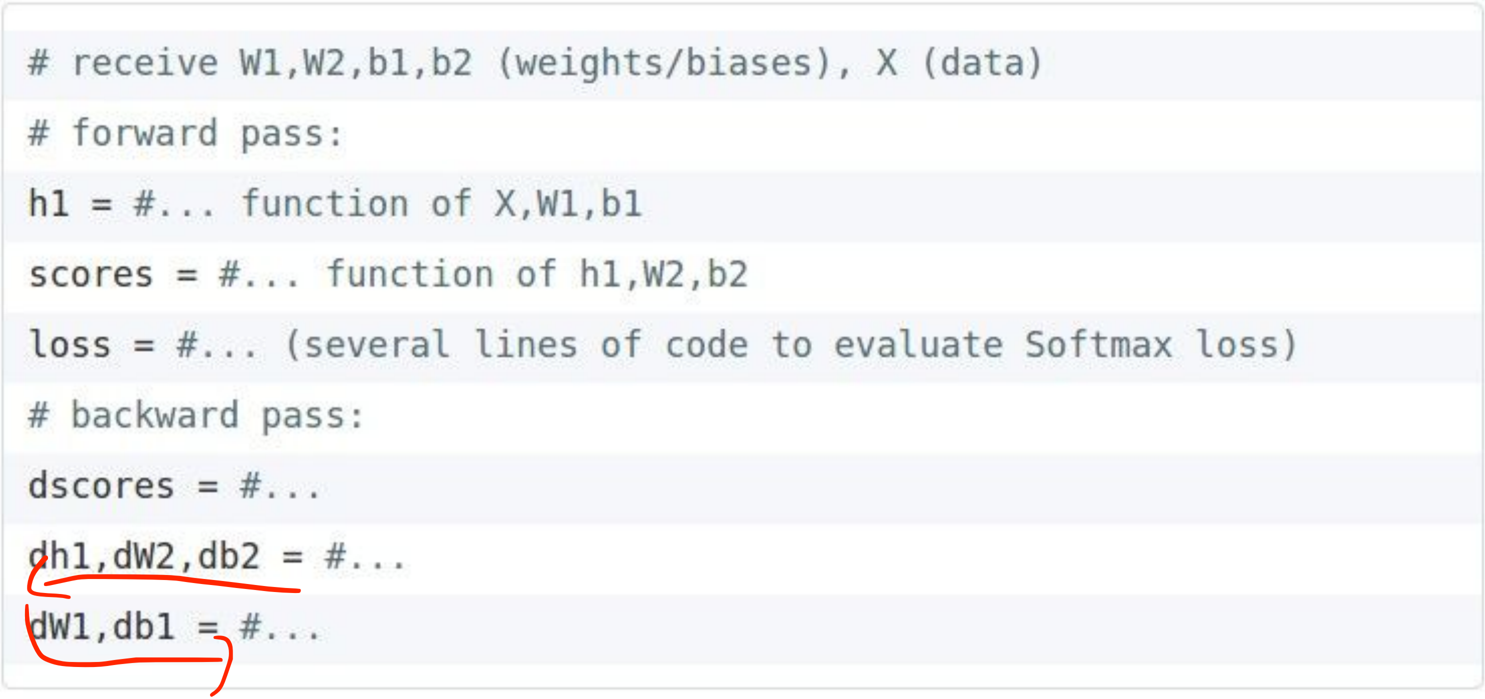


```
X = np.array([ [0,0,1],[0,1,1],[1,0,1],[1,1,1] ])
y = np.array([[0,1,1,0]]).T
syn0 = 2*np.random.random((3,4)) - 1
syn1 = 2*np.random.random((4,1)) - 1
for j in xrange(60000):
    l1 = 1/(1+np.exp(-(np.dot(X,syn0))))
    l2 = 1/(1+np.exp(-(np.dot(l1,syn1))))
    l2_delta = (y - l2)*(l2*(1-l2))
    l1_delta = l2_delta.dot(syn1.T) * (l1 * (1-l1))
    syn1 += l1.T.dot(l2_delta)
    syn0 += X.T.dot(l1_delta)
```

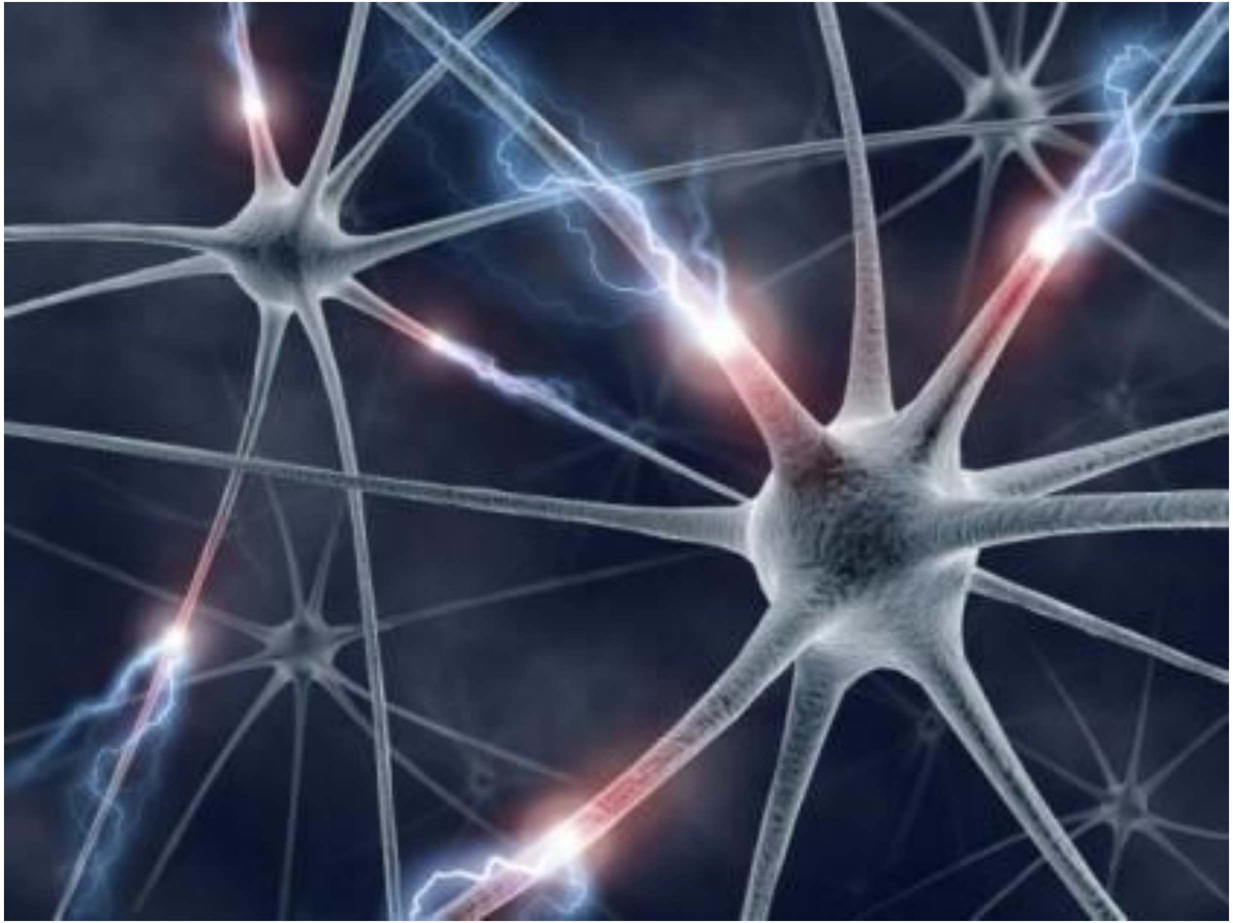
from @iamtrask, http://iamtrask.github.io/2015/07/12/basic-python-network/

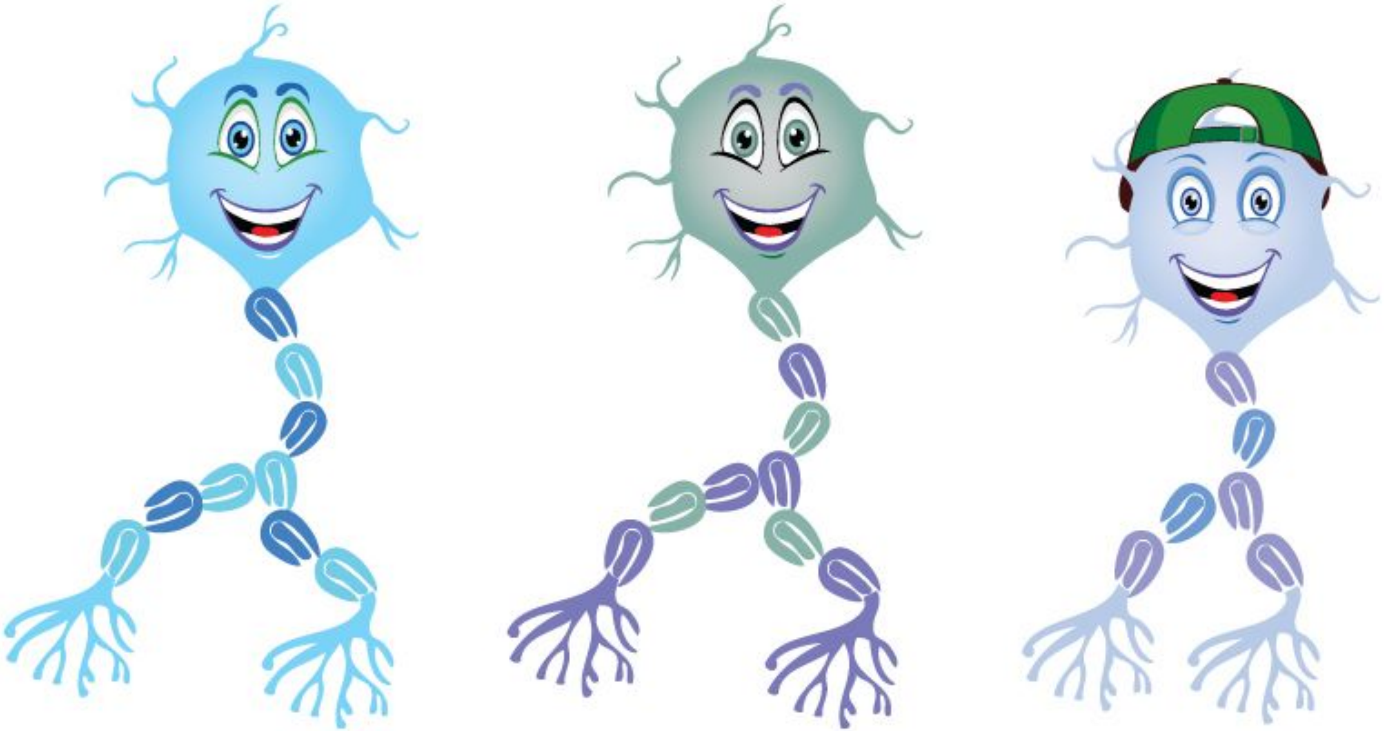
Assignment: Writing 2layer Net

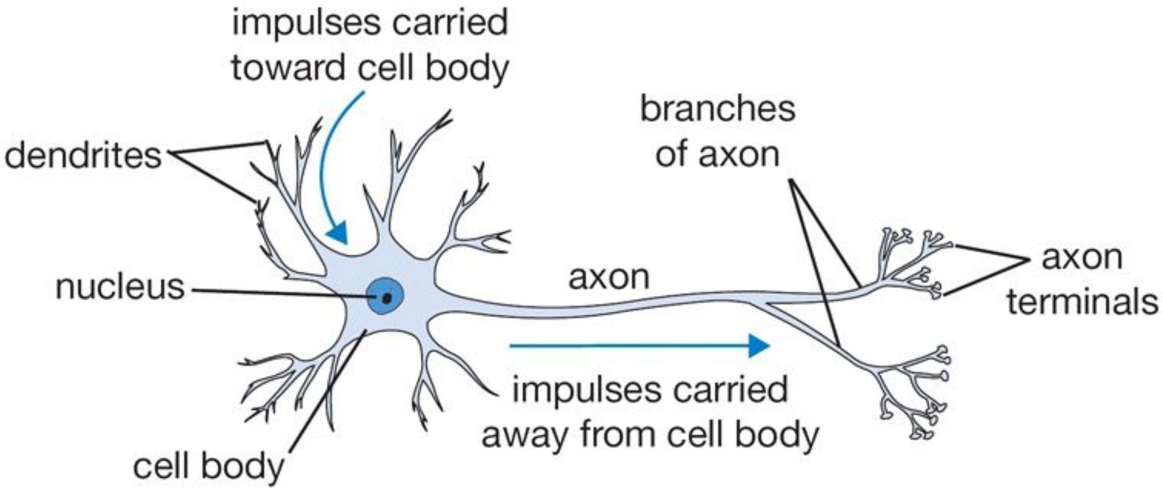
Stage your forward/backward computation!

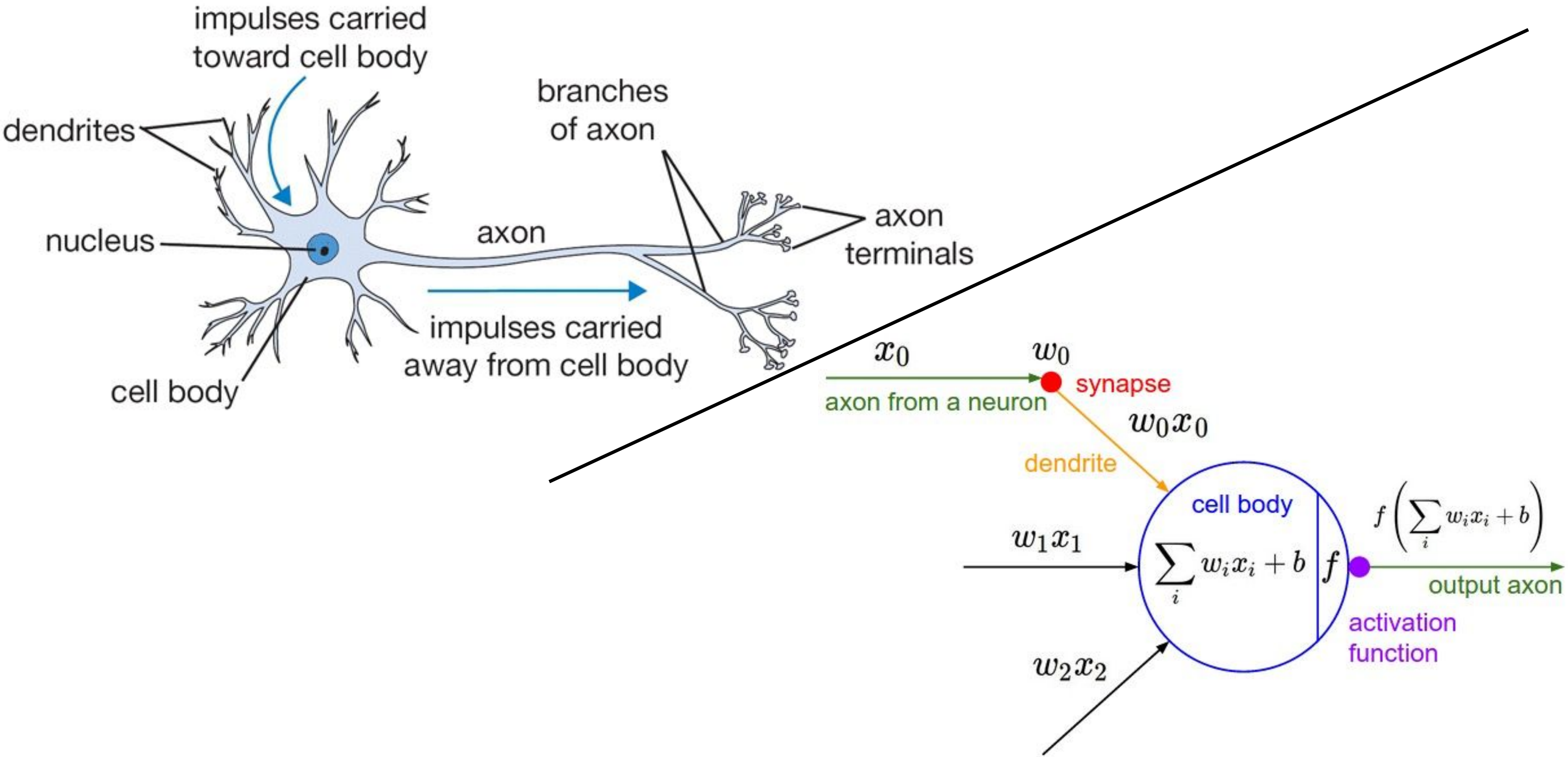


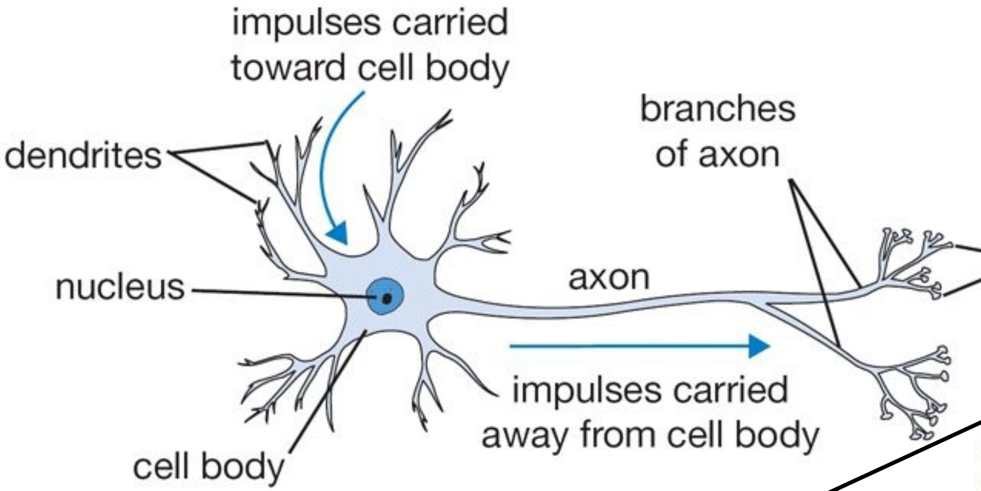
```
# receive W1,W2,b1,b2 (weights/biases), X (data)
# forward pass:
h1 = #... function of X,W1,b1
scores = #... function of h1,W2,b2
loss = #... (several lines of code to evaluate Softmax loss)
# backward pass:
dscores = #...
dh1,dW2,db2 = #...
dW1,db1 = #...
```

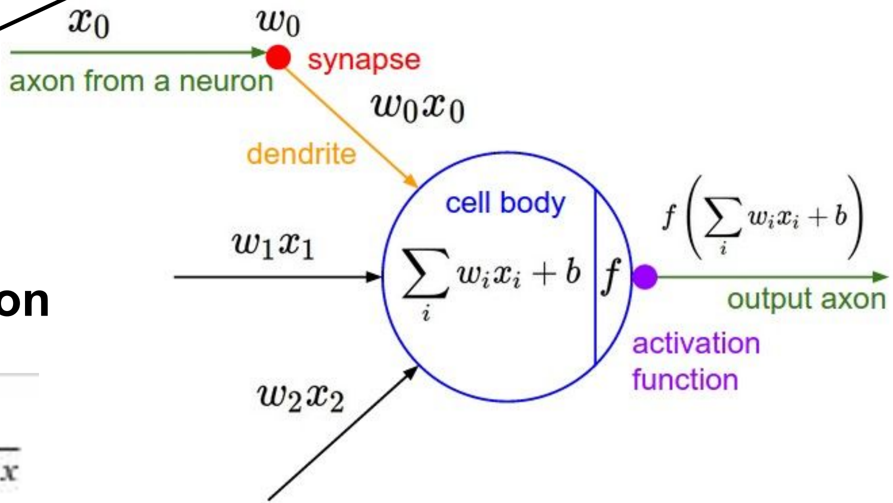


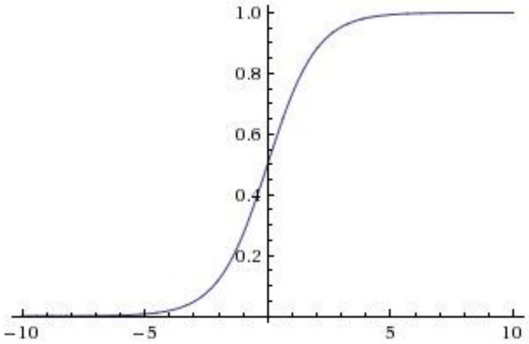



impulses carried toward cell body
dendrites
branches of axon
nucleus
axon
axon terminals
impulses carried away from cell body
cell body


impulses carried toward cell body
dendrites
branches of axon
nucleus
axon
axon terminals
impulses carried away from cell body
cell body
x_0
w_0
synapse
axon from a neuron
$w_0 x_0$
dendrite
cell body
$f\left(\sum_i w_i x_i + b\right)$
$w_1 x_1$
$\sum_i w_i x_i + b$
f
output axon
activation function
$w_2 x_2$

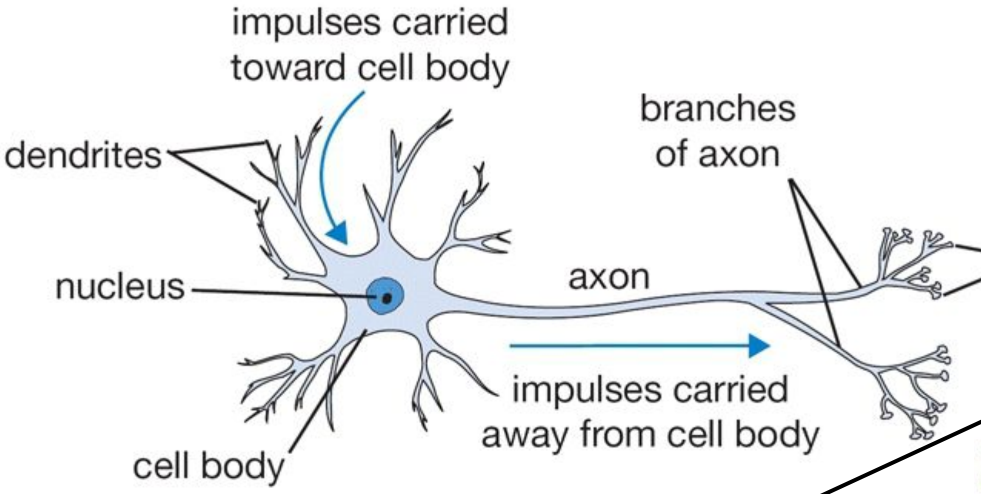

impulses carried toward cell body
dendrites
branches of axon
axon
axon terminals
nucleus
impulses carried away from cell body
cell body

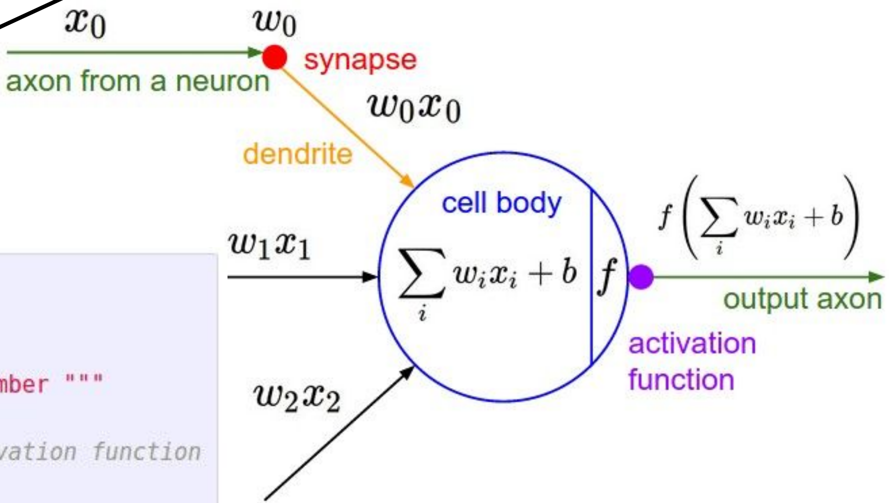




sigmoid activation function

$$\frac{1}{1+e^{-x}}$$


impulses carried toward cell body
dendrites
branches of axon
nucleus
axon
axon terminals
impulses carried away from cell body
cell body

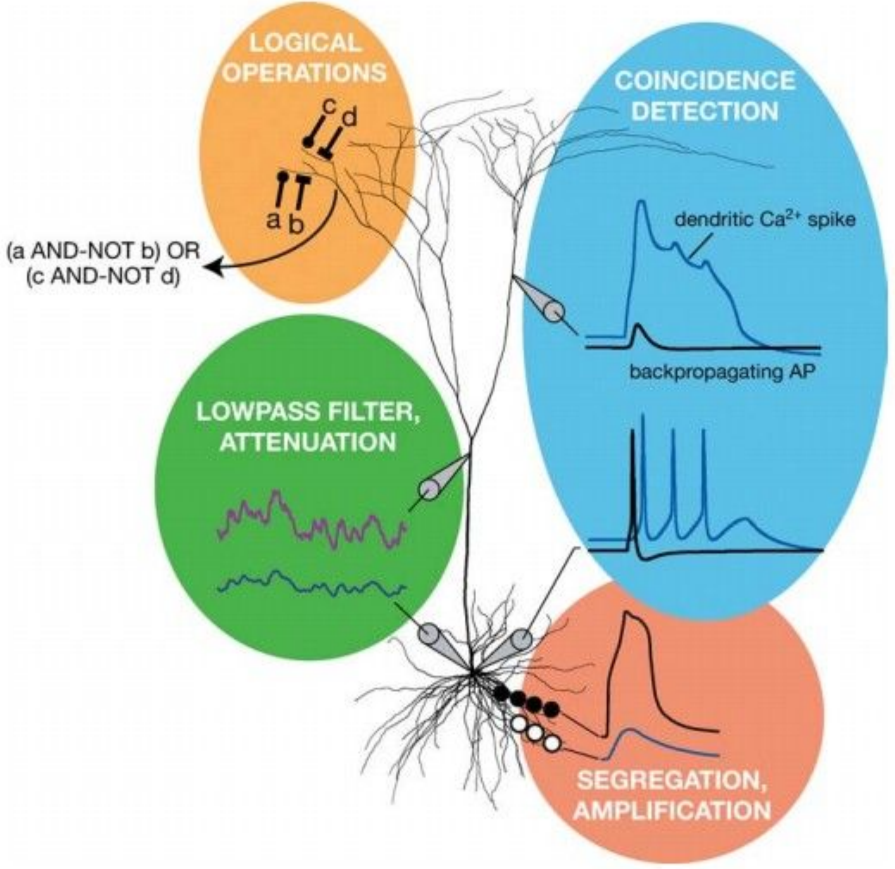


```
class Neuron:
  # ...
  def neuron_tick(inputs):
    """ assume inputs and weights are 1-D numpy arrays and bias is a number """
    cell_body_sum = np.sum(inputs * self.weights) + self.bias
    firing_rate = 1.0 / (1.0 + math.exp(-cell_body_sum)) # sigmoid activation function
    return firing_rate
```

Be very careful with your Brain analogies:

Biological Neurons:

- Many different types
- Dendrites can perform complex non-linear computations
- Synapses are not a single weight but a complex non-linear dynamical system
- Rate code may not be adequate

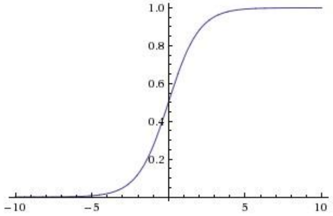

LOGICAL OPERATIONS
c d
a b
(a AND-NOT b) OR (c AND-NOT d)
COINCIDENCE DETECTION
dendritic Ca^{2+} spike
backpropagating AP
LOWPASS FILTER, ATTENUATION
SEGREGATION, AMPLIFICATION

[Dendritic Computation. London and Hausser]

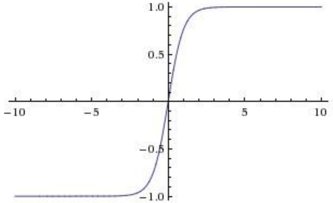
Activation Functions

Sigmoid

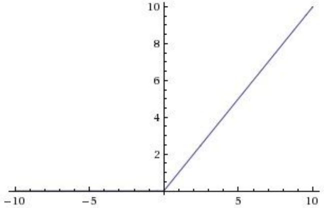
$$\sigma(x) = 1/(1+e^{-x})$$



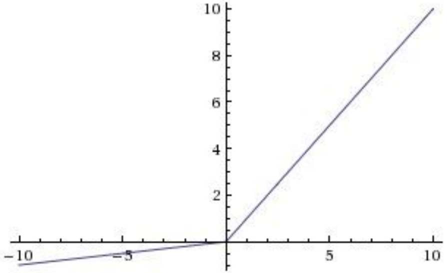
tanh tanh(x)



ReLU max(0,x)



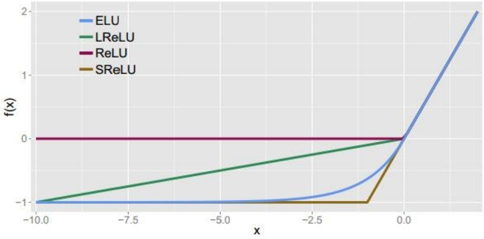
Leaky ReLU
max(0.1x, x)



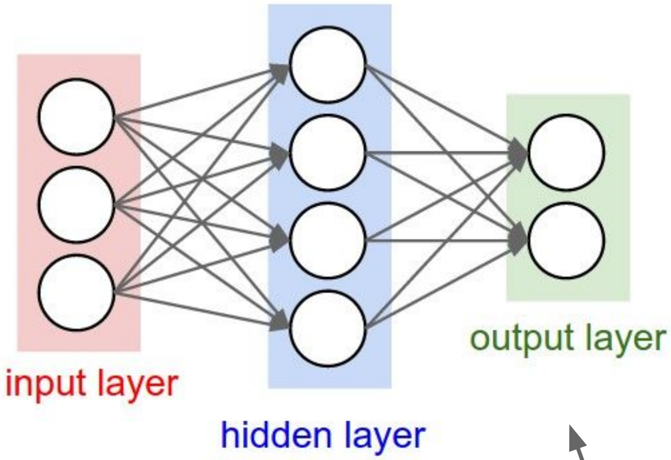
Maxout $\max(w_1^T x + b_1, w_2^T x + b_2)$

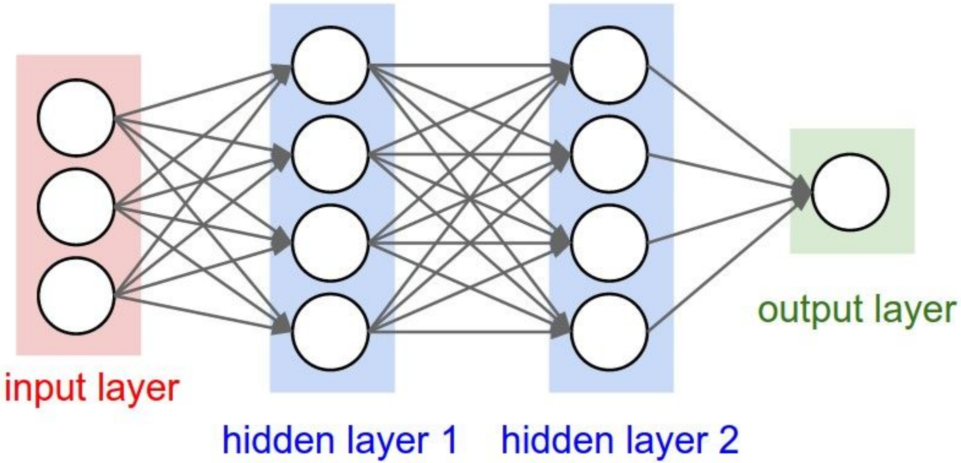
ELU

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha\,(\exp(x) - 1) & \text{if } x \leq 0 \end{cases}$$



Neural Networks: Architectures



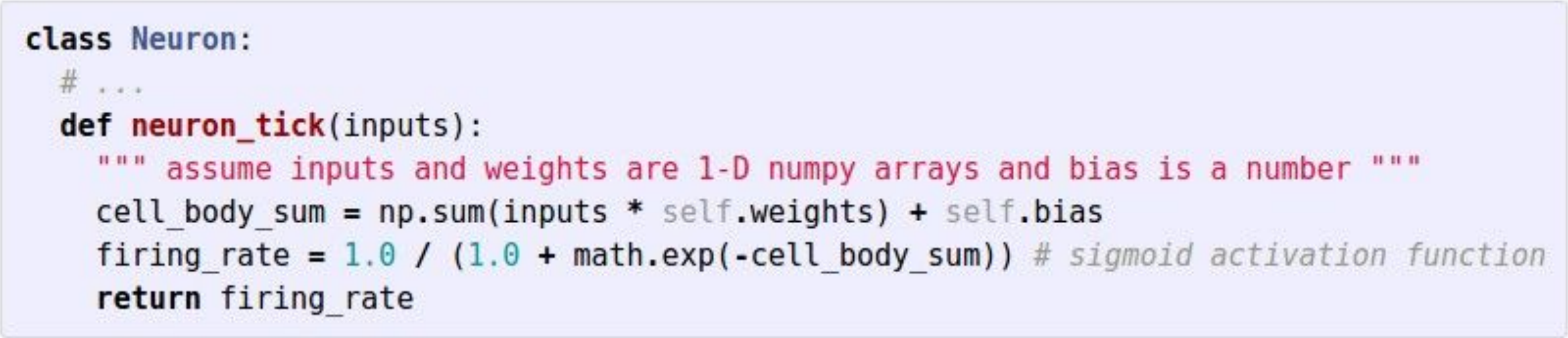


"2-layer Neural Net", or
"1-hidden-layer Neural Net"

"3-layer Neural Net", or
"2-hidden-layer Neural Net"

"Fully-connected" layers

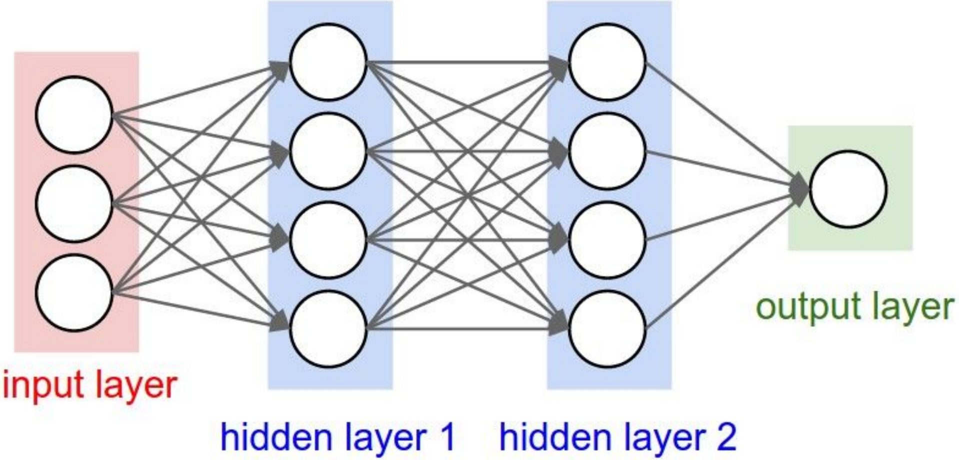
Example Feed-forward computation of a Neural Network

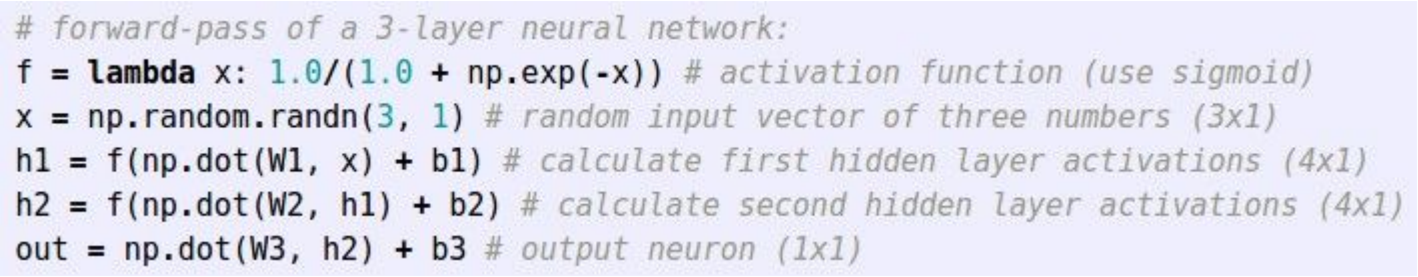


```
class Neuron:
  # ...
  def neuron_tick(inputs):
    """ assume inputs and weights are 1-D numpy arrays and bias is a number """
    cell_body_sum = np.sum(inputs * self.weights) + self.bias
    firing_rate = 1.0 / (1.0 + math.exp(-cell_body_sum)) # sigmoid activation function
    return firing_rate
```

We can efficiently evaluate an entire layer of neurons.

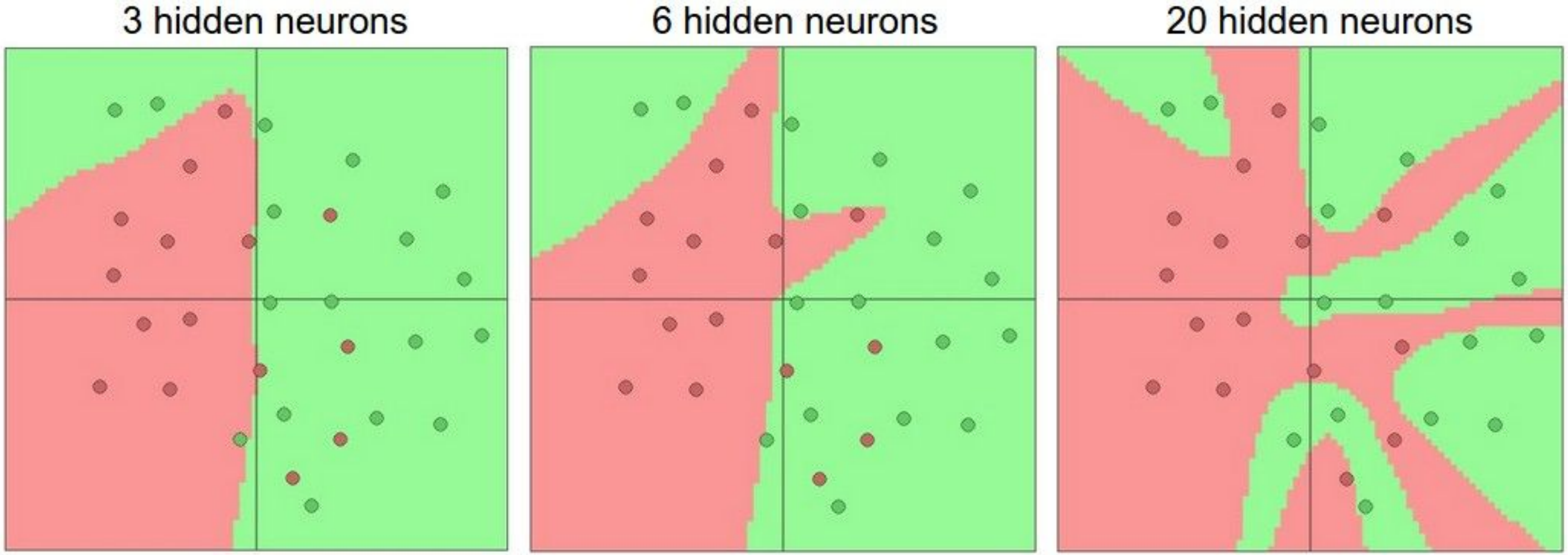
Example Feed-forward computation of a Neural Network





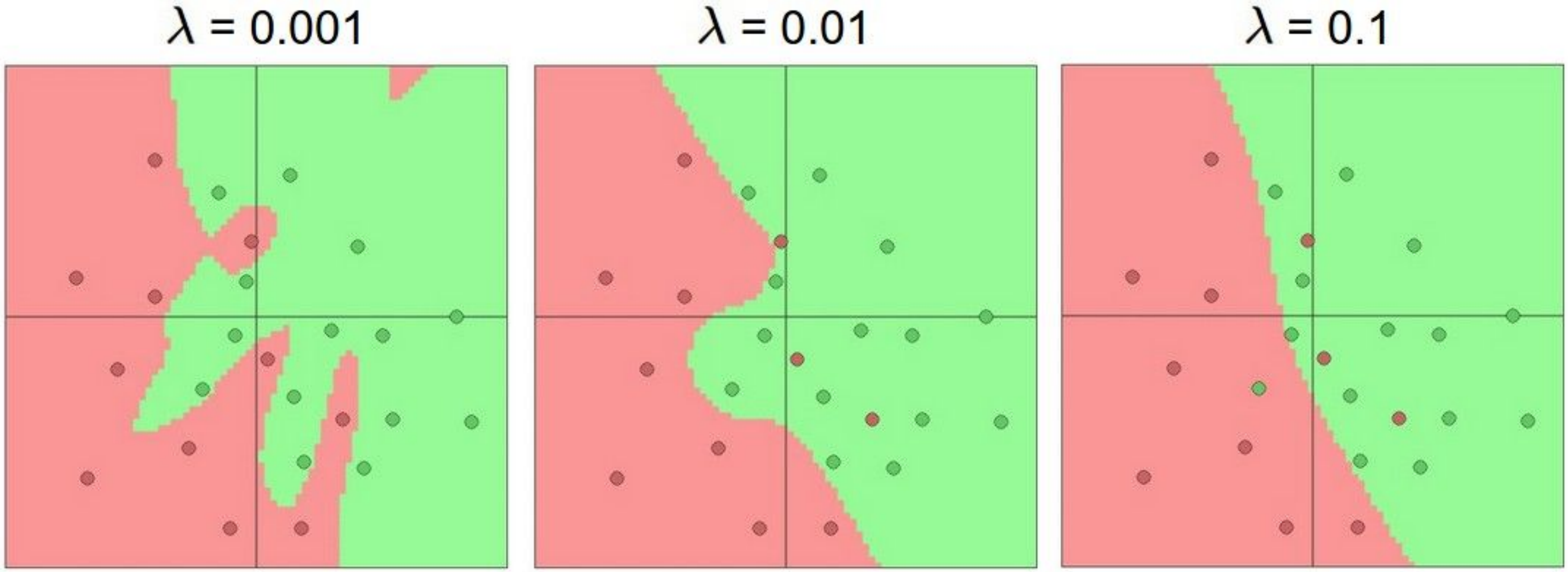
```
# forward-pass of a 3-layer neural network:
f = lambda x: 1.0/(1.0 + np.exp(-x)) # activation function (use sigmoid)
x = np.random.randn(3, 1) # random input vector of three numbers (3x1)
h1 = f(np.dot(W1, x) + b1) # calculate first hidden layer activations (4x1)
h2 = f(np.dot(W2, h1) + b2) # calculate second hidden layer activations (4x1)
out = np.dot(W3, h2) + b3 # output neuron (1x1)
```

Setting the number of layers and their sizes



more neurons = more capacity

Do not use size of neural network as a regularizer. Use stronger regularization instead:



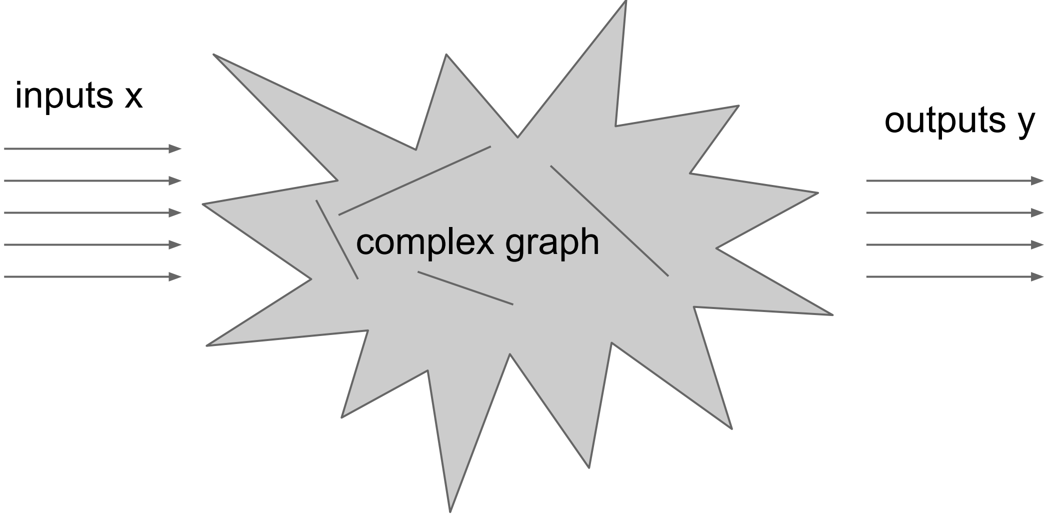
(you can play with this demo over at ConvNetJS: http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html)

Summary

- we arrange neurons into fully-connected layers
- the abstraction of a **layer** has the nice property that it allows us to use efficient vectorized code (e.g. matrix multiplies)
- neural networks are not really *neural*
- neural networks: bigger = better (but might have to regularize more strongly)

Next Lecture:

More than you ever wanted to know about Neural Networks and how to train them.



reverse-mode differentiation (if you want effect of many things on one thing)

$\frac{\partial y}{\partial x}$ for many different x

forward-mode differentiation (if you want effect of one thing on many things)

$\frac{\partial y}{\partial x}$ for many different y