# 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

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

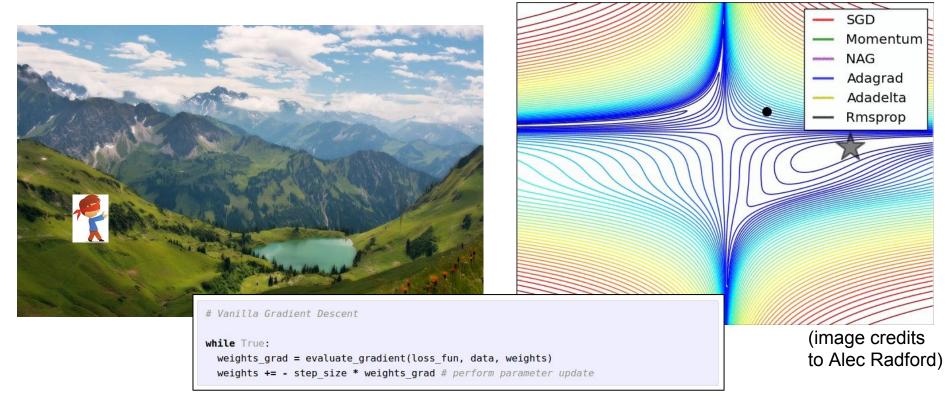
SVM loss

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

data loss + regularization

want  $\nabla_W L$ 

## Optimization



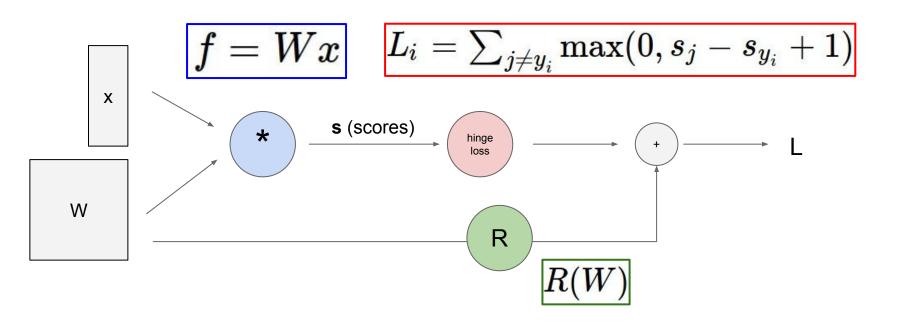
#### Gradient Descent

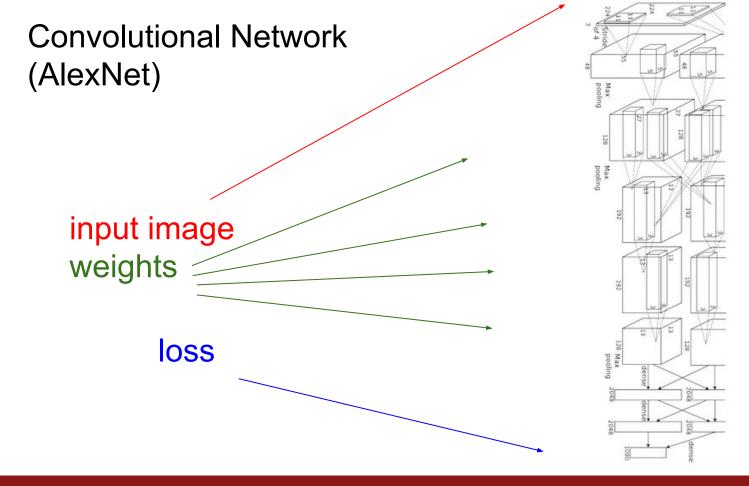
$$rac{df(x)}{dx} = \lim_{h o 0} rac{f(x+h) - f(x)}{h}$$

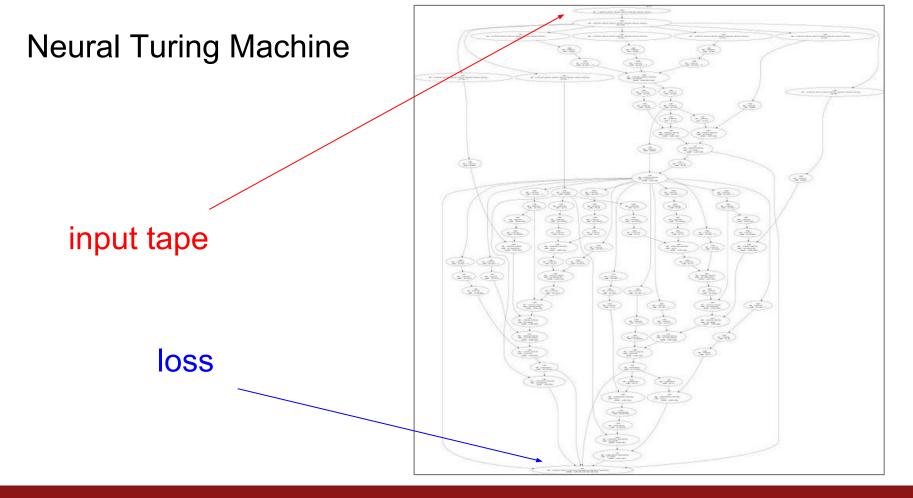
**Numerical gradient**: slow :(, approximate :(, easy to write :) Analytic gradient: fast:), exact:), error-prone:(

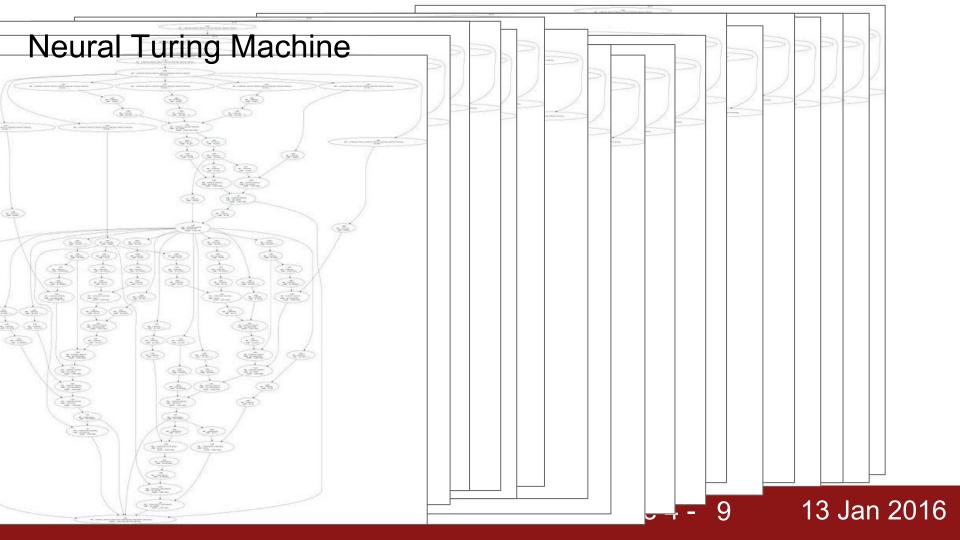
In practice: Derive analytic gradient, check your implementation with numerical gradient

## **Computational Graph**

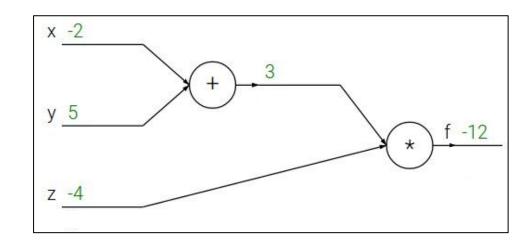








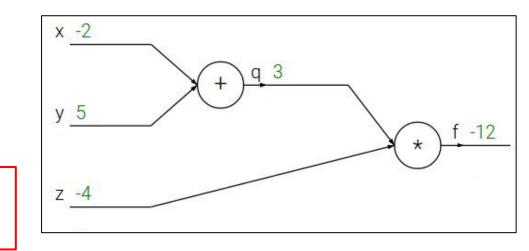
$$f(x, y, z) = (x + y)z$$
  
e.g. x = -2, y = 5, z = -4



$$f(x, y, z) = (x + y)z$$
  
e.g. x = -2, y = 5, z = -4

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

$$f=qz$$
  $rac{\partial f}{\partial q}=z, rac{\partial f}{\partial z}=q$ 

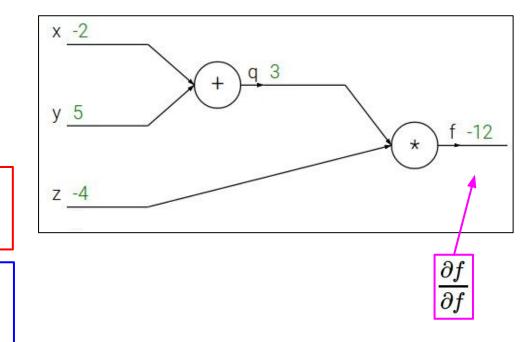


$$f(x, y, z) = (x + y)z$$
  
e.g. x = -2, y = 5, z = -4

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

$$f=qz$$
  $rac{\partial f}{\partial q}=z, rac{\partial f}{\partial z}=q$ 

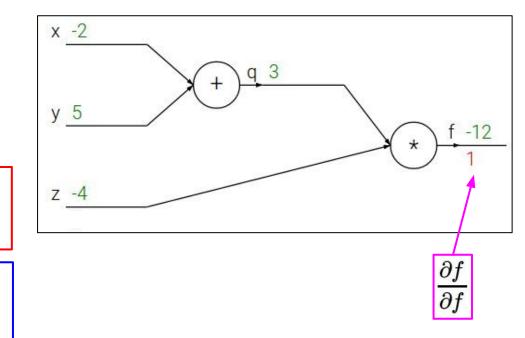
Want: 
$$\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$$



$$f(x, y, z) = (x + y)z$$
  
e.g. x = -2, y = 5, z = -4

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

$$f=qz$$
  $rac{\partial f}{\partial q}=z, rac{\partial f}{\partial z}=q$ 

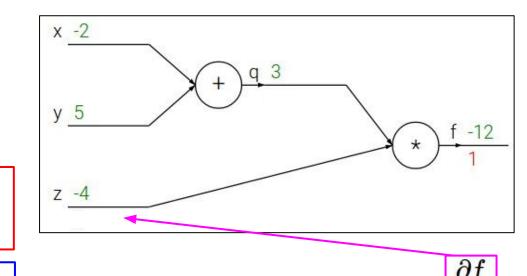


$$f(x, y, z) = (x + y)z$$
  
e.g. x = -2, y = 5, z = -4

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

$$f=qz$$
  $rac{\partial f}{\partial q}=z, rac{\partial f}{\partial z}=q$ 

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

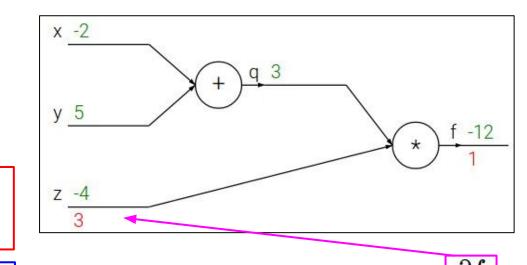


$$f(x, y, z) = (x + y)z$$
  
e.g. x = -2, y = 5, z = -4

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

$$f=qz$$
  $rac{\partial f}{\partial q}=z, rac{\partial f}{\partial z}=q$ 

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



Fei-Fei Li & Andrej Karpathy & Justin Johnson

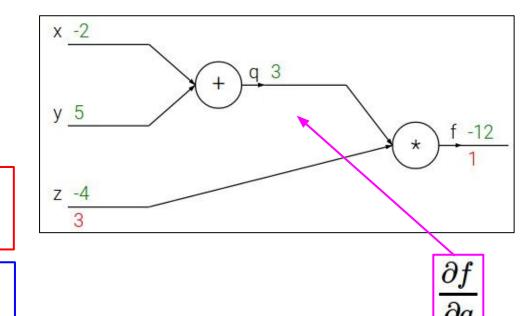
Lecture 4 - 15

13 Jan 2016

$$f(x, y, z) = (x + y)z$$
  
e.g. x = -2, y = 5, z = -4

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

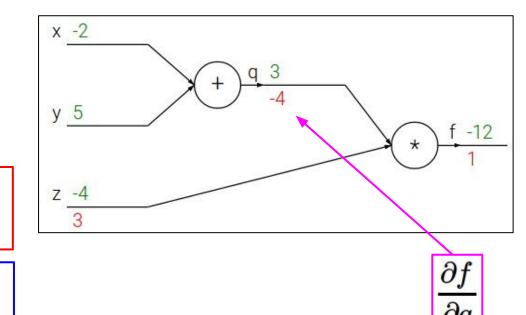
$$f=qz$$
  $rac{\partial f}{\partial q}=z, rac{\partial f}{\partial z}=q$ 



$$f(x, y, z) = (x + y)z$$
  
e.g. x = -2, y = 5, z = -4

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

$$f=qz$$
  $rac{\partial f}{\partial q}=z, rac{\partial f}{\partial z}=q$ 

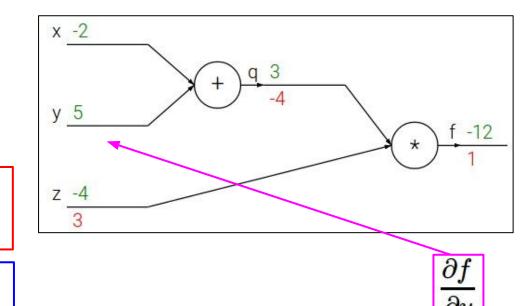


$$f(x, y, z) = (x + y)z$$
  
e.g. x = -2, y = 5, z = -4

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

$$f=qz$$
  $rac{\partial f}{\partial q}=z, rac{\partial f}{\partial z}=q$ 

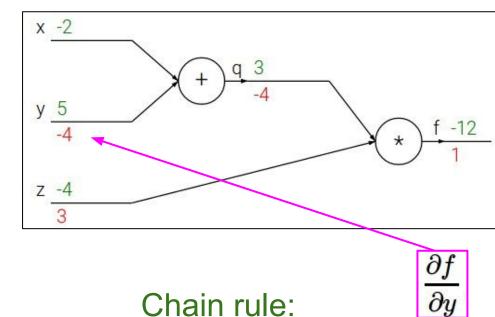
Want: 
$$\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$$

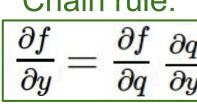


$$f(x, y, z) = (x + y)z$$
  
e.g. x = -2, y = 5, z = -4

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

$$f=qz$$
  $rac{\partial f}{\partial q}=z, rac{\partial f}{\partial z}=q$ 

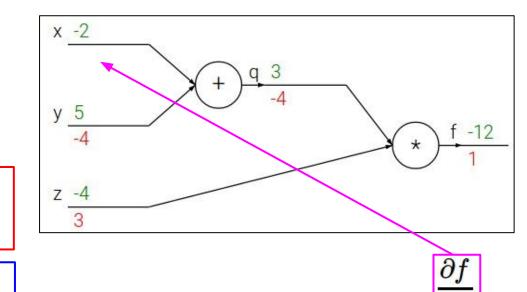




$$f(x, y, z) = (x + y)z$$
  
e.g. x = -2, y = 5, z = -4

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

$$f=qz$$
  $rac{\partial f}{\partial q}=z, rac{\partial f}{\partial z}=q$ 

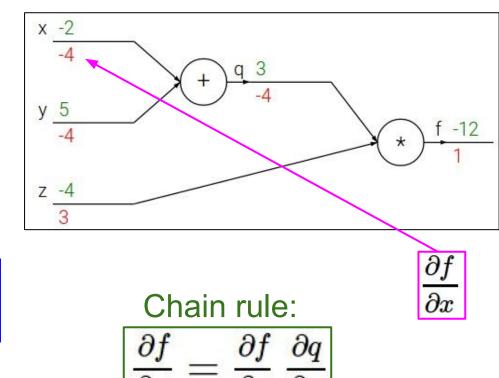


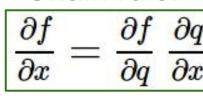
$$f(x, y, z) = (x + y)z$$
  
e.g. x = -2, y = 5, z = -4

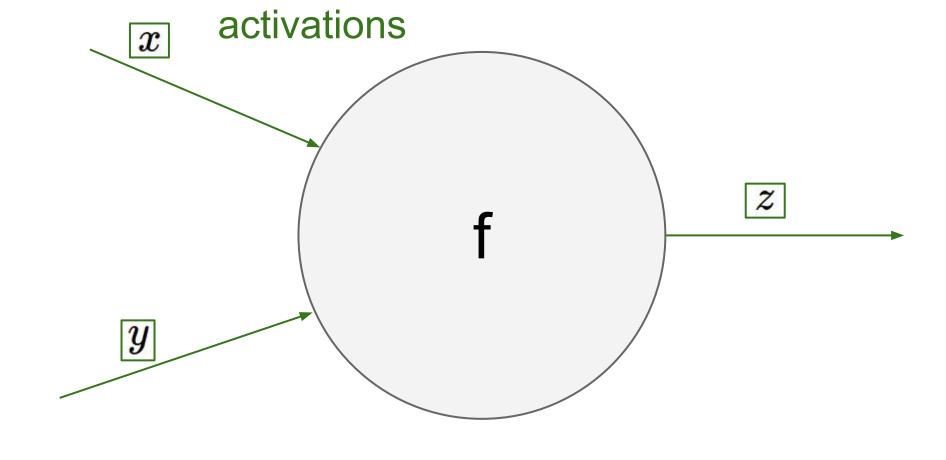
$$q=x+y \qquad rac{\partial q}{\partial x}=1, rac{\partial q}{\partial y}=1$$

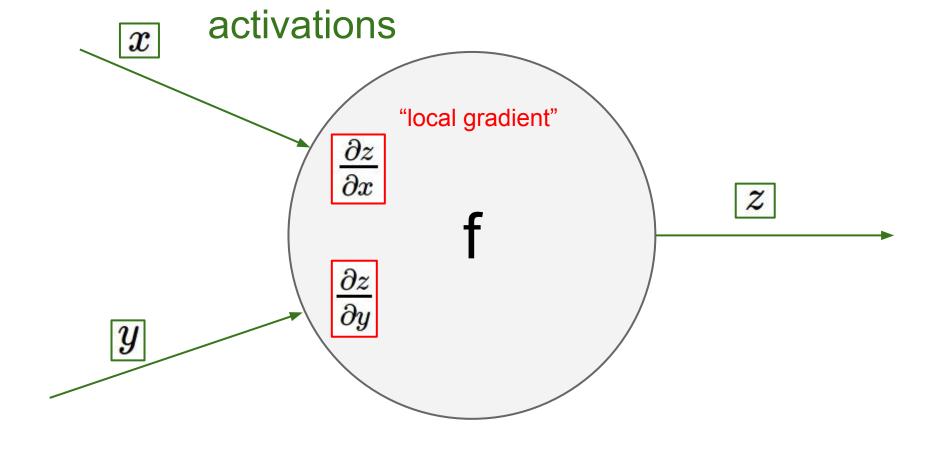
$$f=qz$$
  $rac{\partial f}{\partial q}=z, rac{\partial f}{\partial z}=q$ 

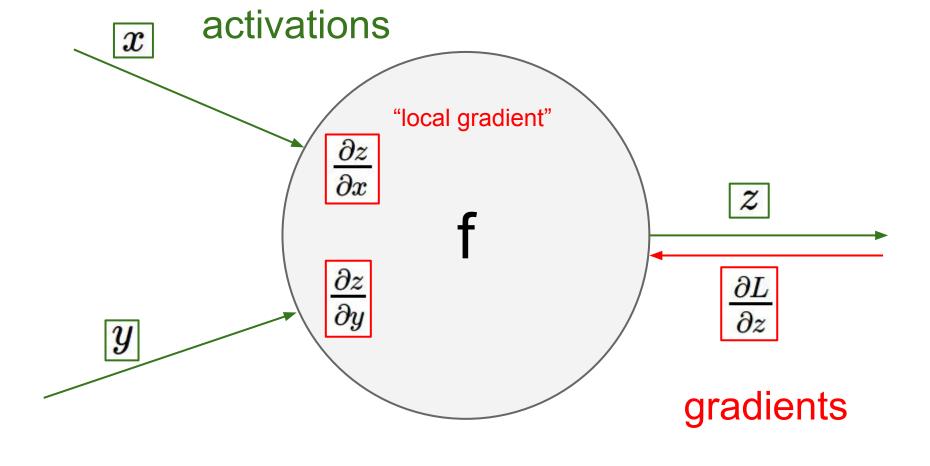
Want:

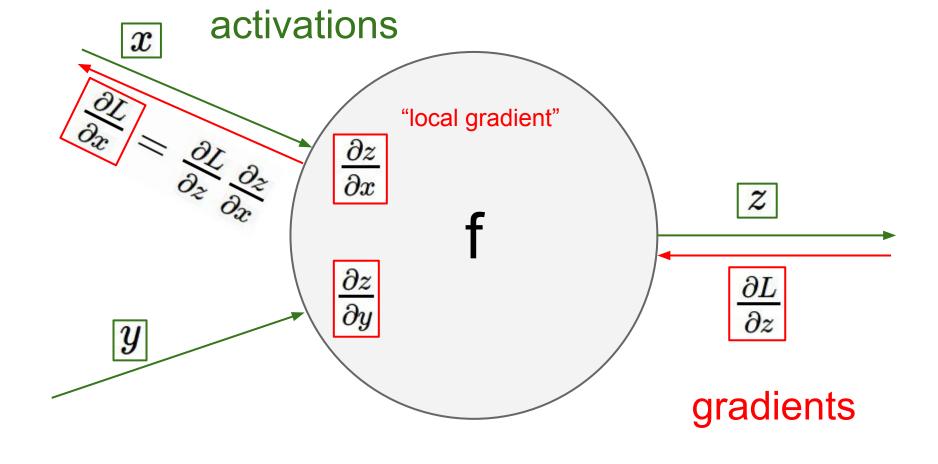


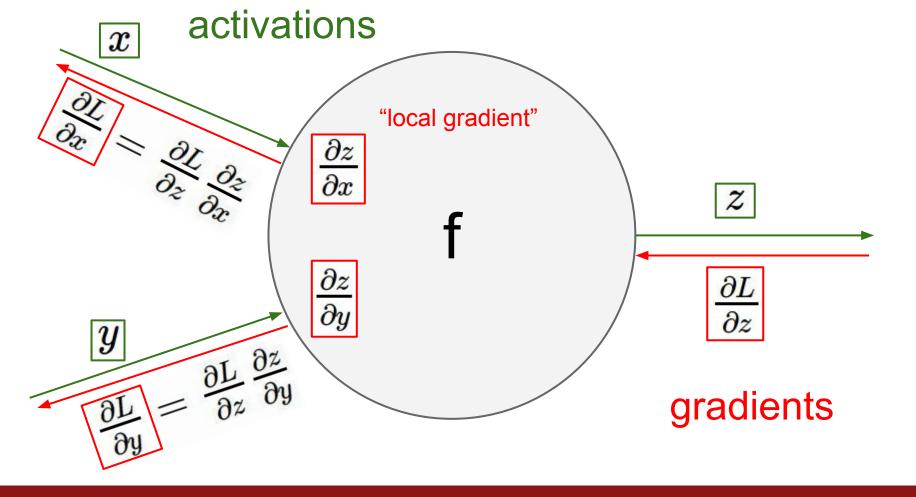


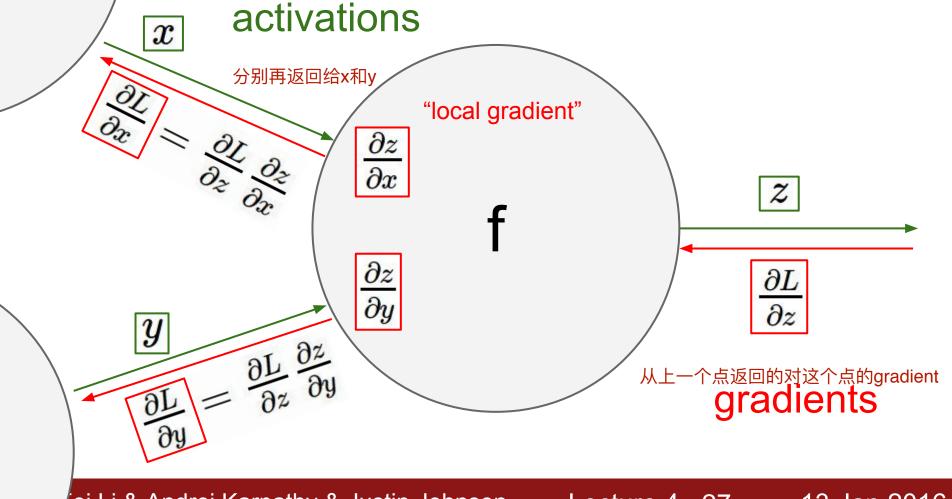




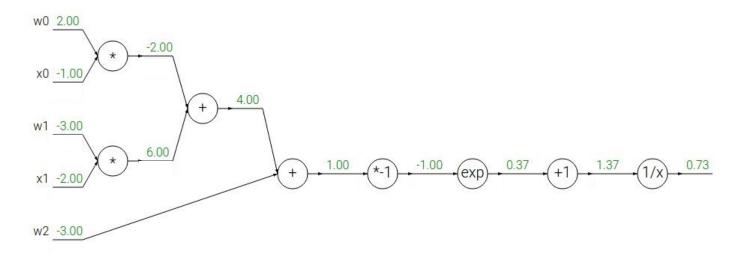




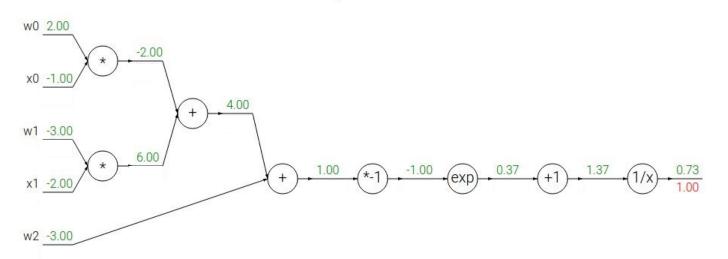




## Another example: $f(w,x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$

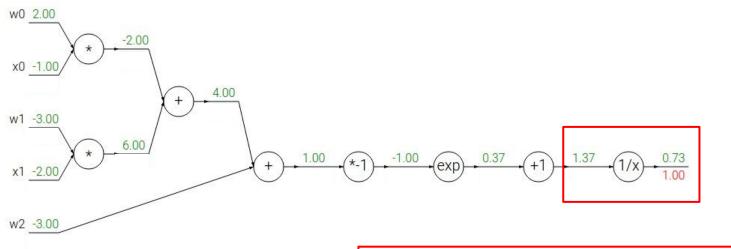


$$f(w,x)=rac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \hspace{1cm} o \hspace{1cm} rac{df}{dx} = e^x \hspace{1cm} f(x) = rac{1}{x} \hspace{1cm} o \hspace{1cm} rac{df}{dx} = -1/x^2 \ f_a(x) = ax \hspace{1cm} o \hspace{1cm} rac{df}{dx} = a \hspace{1cm} f_c(x) = c + x \hspace{1cm} o \hspace{1cm} rac{df}{dx} = 1$$

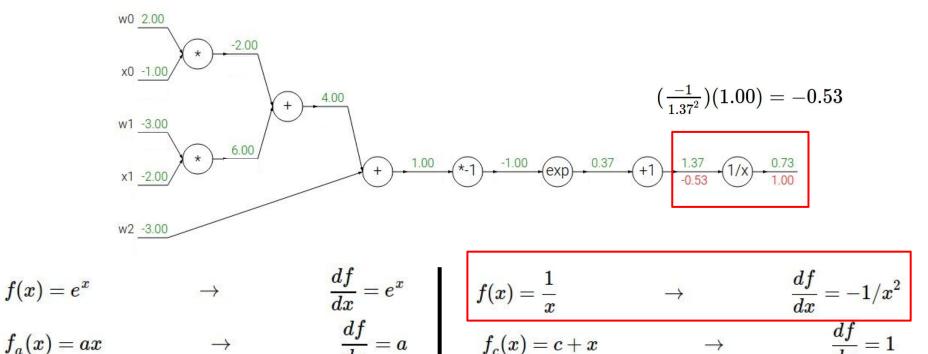
$$f(w,x)=rac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



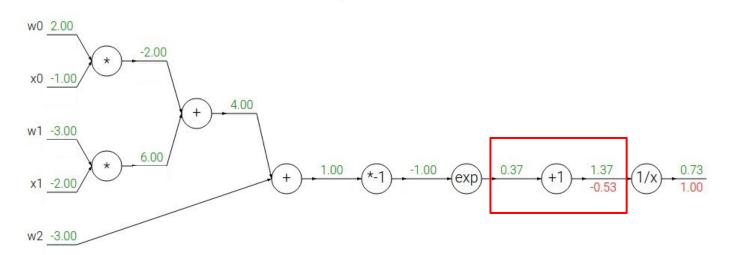
$$egin{aligned} f(x) = e^x & 
ightarrow & rac{df}{dx} = e^x \ & \ f_a(x) = ax & 
ightarrow & rac{df}{dx} = a \end{aligned}$$

$$f(x)=rac{1}{x} \qquad \qquad \qquad rac{df}{dx}=-1/x^2 \ f_c(x)=c+x \qquad \qquad \qquad \qquad rac{df}{dx}=1$$

$$f(w,x) = rac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2}}$$

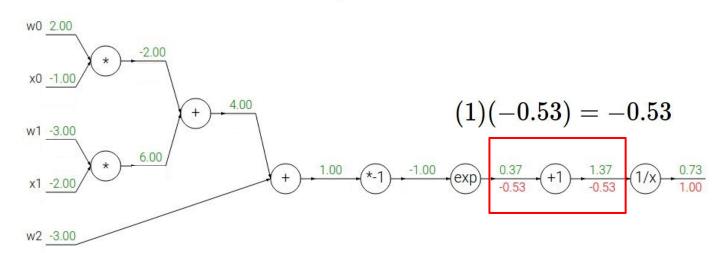


$$f(w,x) = rac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2}}$$

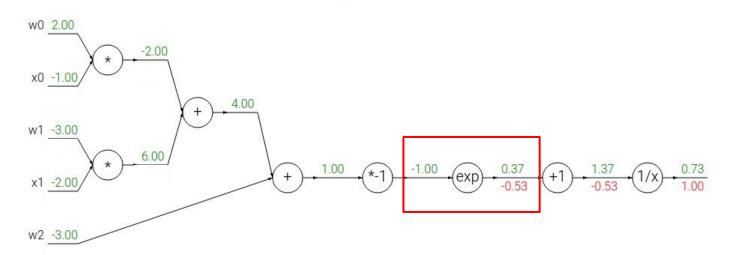


$$f(x) = e^x \hspace{1cm} o \hspace{1cm} rac{df}{dx} = e^x \hspace{1cm} f(x) = rac{1}{x} \hspace{1cm} o \hspace{1cm} rac{df}{dx} = -1/x^2 \ f_a(x) = ax \hspace{1cm} o \hspace{1cm} rac{df}{dx} = a \hspace{1cm} f_c(x) = c + x \hspace{1cm} o \hspace{1cm} rac{df}{dx} = 1$$

$$f(w,x) = rac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$$

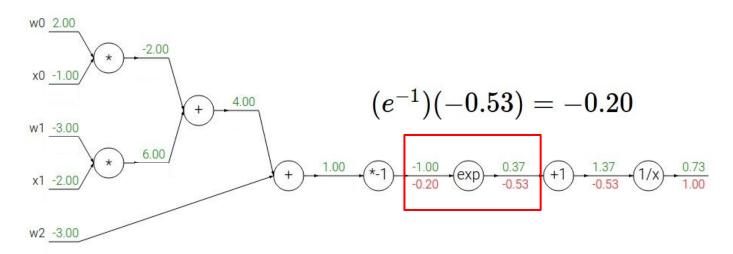


$$f(w,x)=rac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



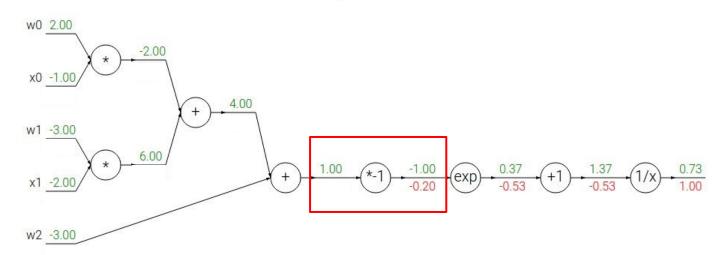
$$egin{aligned} f(x) = e^x & 
ightarrow & rac{df}{dx} = e^x \ f_a(x) = ax & 
ightarrow & rac{df}{dx} = a \ \end{array} \qquad egin{aligned} f(x) = rac{1}{x} & 
ightarrow & rac{df}{dx} = -1/x^2 \ f_c(x) = c + x & 
ightarrow & rac{df}{dx} = 1 \ \end{array}$$

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



$$egin{aligned} f(x) = e^x & 
ightarrow & rac{df}{dx} = e^x \ f_a(x) = ax & 
ightarrow & rac{df}{dx} = a \ \end{array} \qquad egin{aligned} f(x) = rac{1}{x} & 
ightarrow & rac{df}{dx} = -1/x^2 \ f_c(x) = c + x & 
ightarrow & rac{df}{dx} = 1 \ \end{array}$$

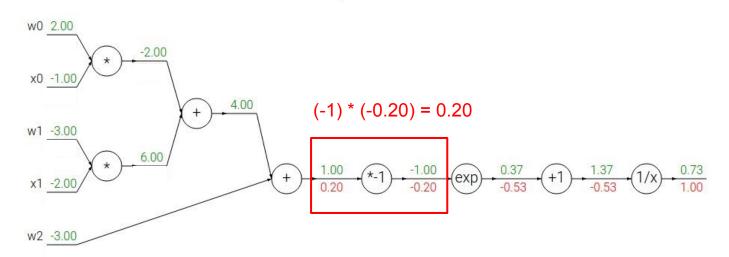
$$f(w,x)=rac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$f(x) = e^x \qquad o \qquad rac{df}{dx} = e^x \ f_a(x) = ax \qquad o \qquad rac{df}{dx} = a$$

$$f(x)=rac{1}{x} \qquad \qquad 
ightarrow \qquad rac{df}{dx}=-1/x \ f_c(x)=c+x \qquad \qquad 
ightarrow \qquad rac{df}{dx}=1$$

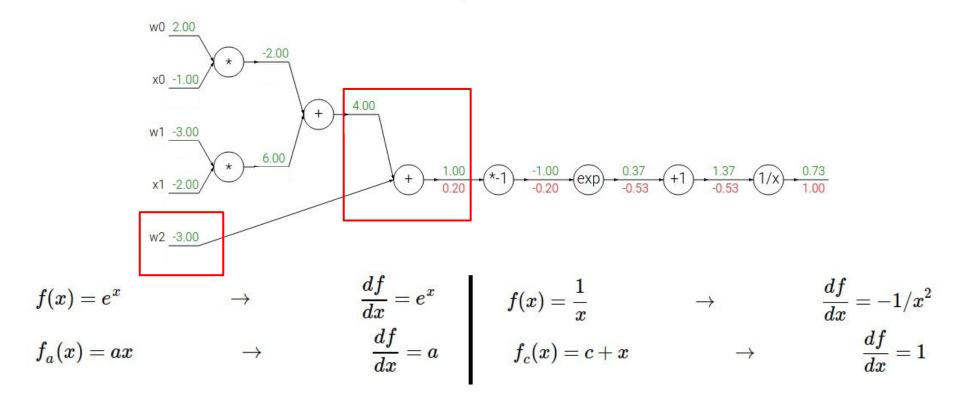
$$f(w,x) = rac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2}}$$



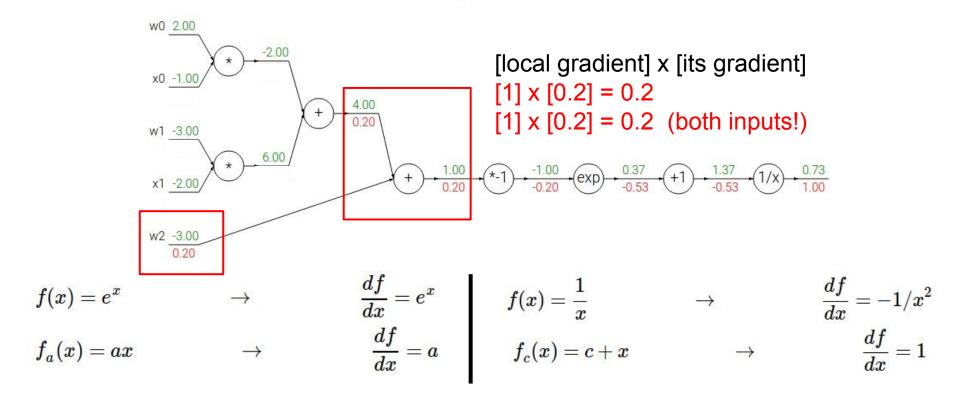
$$f(x) = e^x \qquad o \qquad rac{df}{dx} = e^x \ f_a(x) = ax \qquad o \qquad rac{df}{dx} = a$$

$$f(x)=rac{1}{x} \qquad \qquad 
ightarrow \qquad rac{df}{dx}=-1/x \ f_c(x)=c+x \qquad \qquad 
ightarrow \qquad rac{df}{dx}=1$$

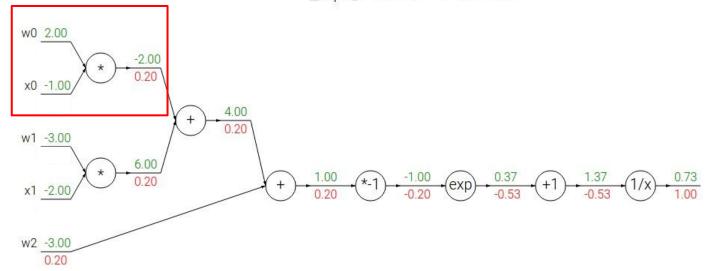
$$f(w,x)=rac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



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

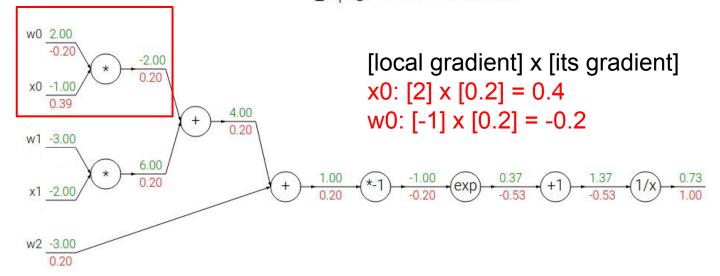


$$f(w,x)=rac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



### Another example:

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



$$f(x) = e^x \qquad \qquad o \qquad \qquad rac{df}{dx} = e$$
  $f_a(x) = ax \qquad \qquad o \qquad \qquad rac{df}{dx} = e$ 

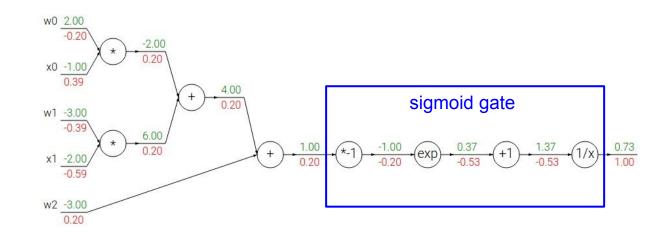
$$rac{df}{dx} = e^x \qquad \qquad f(x) = rac{1}{x} \qquad 
ightarrow \qquad rac{df}{dx} = -1/x \ rac{df}{dx} = a \qquad \qquad f_c(x) = c + x \qquad 
ightarrow \qquad rac{df}{dx} = 1$$

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

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

sigmoid function

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

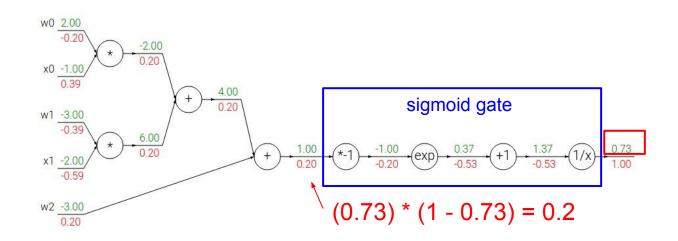


$$f(w,x)=rac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

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

sigmoid function

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

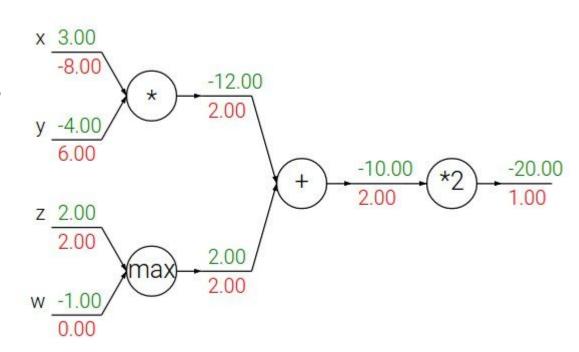


#### Patterns in backward flow

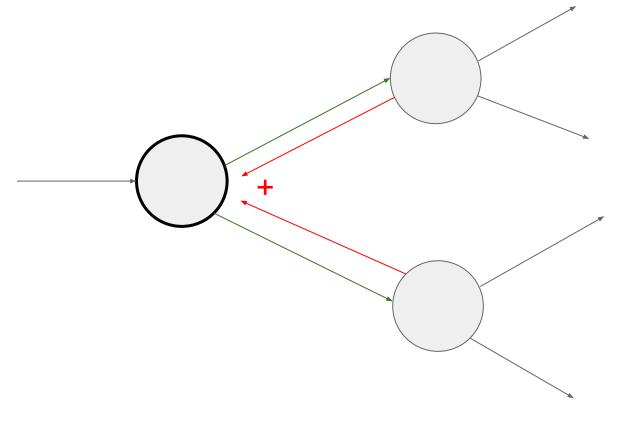
add gate: gradient distributor

max gate: gradient router

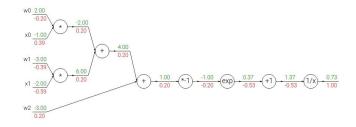
mul gate: gradient... "switcher"?



## Gradients add at branches



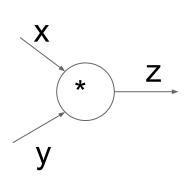
## Implementation: forward/backward API



#### 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
```

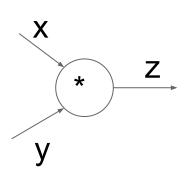
## Implementation: forward/backward API



(x,y,z are scalars)

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

## Implementation: forward/backward API



```
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]
```

(x,y,z are scalars)



## **Example: Torch Layers**



E LogSigmoid las	Add THNN conversion of (ELU, LeakyReLU, LogSigmoid, LogSofMax, Looku	7 days ag
E LogSofMax lua	Add THNN conversion of (ELU, LeakyReLU, LogSigmoid, LogSoftMax, Looku	7 days ag
€ LookupTable lua	Harmonize LookupTable signature with curn impl	5 days ag
E MM.lus	Rename unpack to table unpack for Lua 5.2	8 months ag
MSECriterion lua	Add SizeAverage to criterions in the constructor	2 months ag
MarginCriterion.kus      ✓ MarginCriterion.kus	modernized MarginCriterion	a year ag
MarginRankingCriterion.lux	Fix batch mode in MarginRankingCriterion	4 days ag
Max.lua	Merge pull request #464 from vigire/master	2 months ag
Mean.lus     Mean.lus	Add support for negative dimension and both batch and non batch input	2 months ag
MinJua     Mi	Merge pull request #464 from vgire/master	2 months ag
MintureTable lua	cancel unused variable and useless expression	29 days ag
Module.lus	Revert "Don't re-flatten parameters if they are already flattered"	15 hours ag
Mullus .	removing the requirement for providing size in nn.Mul	a year ag
MulConstant.lus	Ignore updateGradinput if self-gradinput is nil	3 months ag
MultiCriterion lua	asserts in MultiCriterion and ParallelCriterion add	2 months ag
MuttLabeMarginCriterion.lua  MuttMarginCriterion.lua	initial revemp of torch? tree	4 years ag
Narrowlus     Narrowl	multimargin supports p=2 typeAs in Narrow not done in place.	11 months ag 6 months ag
NarrowTable lus	Namos/Table	6 months ag
Fi Nomalizatua	Remove brem and beddbrem from Normalize, because they allocate memory,	20 days ap
R PReLUtus	Buffers for PReLU cuto implementation.	8 months ag
Padding lus	fixed broken nn.Padding: input was returned in backprop	5 months ag
PairwiseDistanceJua	Merge pull request #532 from xwgeng/master	29 days ag
Paralel ka	fix a bug in conditional expression	a month ag
ParallelCriterion.lpa	asserts in MultiCriterion and ParallelCriterion add	2 months ag
Paralle/Table.lua	Parallel optimization. ParallelTable inherits Container, unit tests	a year ag
Powertus	Use UNIX line endings	7 months ag
README.nd	doc readthedocs	5 months ag
RReLUlus	Add randomized leaky rectified linear unit (RReLU)	3 months ag
P ReLUtes	adds in-place ReLU and fixes a potential divide-by-zero in nn.Sgrt	9 months ag
Replicate lua	Replicate batchMode	8 months ag
Reshape.lus	Added more informative pretty-printing.	a year ag
Selectiva	initial reviewp of terch? tree	4 years ag
SelectTable.lua	nn.Module preserve type sharing semantics (#187); add nn.Module apply	4 months ag
Sequential.lua	fixing Sequential remove corner case	6 months ag
	Initial revenue of terch7 tree	4 years ag
SmoothL1Criterion.kus	Add SizeAverage to criterions in the constructor	2 months ag
SoftMax.lus	Fix various unused variables in nn	a year ag
⊕ SotMin.lus	Fix various unused variables in nn	a year ag
SoftPlus kas	fixed a numerical issue in the SoffPlus module (it breaks for input g	2 years ag
SofShink lua SofSign lua	initial reverse of torch? tree	4 years ag
Sottlign.lus  SparseJacobian.lus	initial revering of torch? tree Fix various unused variables in nn	4 years ag a year ag
SparseJacober ka	Fix various unused variaties in in  Using source implementation of zeroGradParameters for SourceLinear	a year ag a month ag
SpatialAdaptiveMasPooling	Added Spatial AdaptiveMacPooling	a nom ag a year ag
Spatial temporal cong	SpatialAveragePooling supports padding, cell mode and exclude_pad div	29 days ag
SpetialBatchNormalization lua	Add C implementation of SpatialBatchNormalization	7 days ag
	Make type() truly recursive.	9 months ap
SpatialConvolution.lus	Fix hoel) in SpatialConvolution	3 months ag
SpetialConvolutionMM.lux	Fix type() in SpatialConvolution	3 months ag
	Remove unused and expensive initialization logic from nn Spatial Conve	8 months ag
□ SpatialCrossMapLRN kus	cuda consistency	18 days ag
Spatial Divisive Normalizatio	Spatial[Constructive,Divisive,Subtractive]Normalization work with but	8 months ag
	small fix on error message	6 months ag
SpatialFractionalMadPoolin	Adding Fractional Max Pooling	3 months ag
SpatialFullConvolution.tus	Add adjustment term to SpatiaFullConvolution to control the size of	5 days ag
	New NN classes	3 years ag
SpatialLPPcolingJua	SpatialAveragePooling divides by kWNH	10 months ag
SpatiaMaxPooling lua	SpatiaMaxPooling supports padding and cell mode	6 months ag
SpatiaMaxUrpooling lua	Add SpatialMaxUnpooling	26 days ag
Speta/SoftMaxiua	Update SoftMax to work in spetial mode	4 months ag
SpetialSubSampling lua	Merge branch 'nn_fast_reset'	3 years ag
SpatialSubtractiveNormaliz	Spatial(Constructive, Divisive, Subtractive)Normalization work with but	8 months ag
■ SpatialUpSamplingNearest.L.	Use UNIX line endings	7 months ag
SpatialZeroPadding lux	Added more informative pretty-printing.	a year ag

## **Example: Torch Layers**



El LogSigmoid lug	Add THNN conversion of (ELU, LeskyReLU, LogSigmoid, LogSofMax, Looku	7 days ago
E LogSofMax lua	Add THNN conversion of (ELU, LeakyReLU, LogSigmoid, LogSofMax, Looku	7 days ago
	Harmonize LookupTable signature with curn impl	5 days ago
MM lus     MM	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 kus	modernized MarginCriterion	a year ago
MarginRankingCriterion.lus	Fix batch mode in MarginRankingCriterion	4 days ago
Maxiua	Merge pull request #464 from vgire/master	2 months ago
Meanlus     Meanlus	Add support for negative dimension and both batch and non-batch input	2 months ago
MinJua     Mi	Merge pull request #464 from vgire/master	2 months ago
MintureTable lua	cancel unused variable and useless expression	29 days ago
Module.lus	Revert "Don't re-flatten parameters if they are already flattened"	15 hours ago
® Mullus	removing the requirement for providing size in nn.Mul	a year ago
MulConstant.lus	Ignore updateGradinput if self-gradinput is nil	3 months ago
MultiCriterion.loa	asserts in MultiCriterion and ParallelCriterion add	2 months ago
MutiLabeMarginCriterion.lua	initial reversp of torch? tree	4 years ago
MultiMarginCriterion.lua	multimargin supports p+2	11 months ago
Narrow/ua	typeAs in Narrow not done in place.	6 months ago
NarrowTable.lus     NarrowTable.lus	NarrowTable	6 months ago
Normalize lua	Remove brem and baddbern from Normaliza, because they allocate memory,	20 days ago
PROLUMB	Buffers for PReLU cude implementation.	8 months ago
Padding kis	fixed broken nn.Padding: input was returned in backgrop	5 months ago
PairwiseDistance.lue	Merge pull request #532 from xwgeng/master	29 days ago
⊕ Paralel ka	fix a bug in conditional expression	a month ago
ParallelCriterion.lpa	asserts in MultiCriterion and ParaflelCriterion add	2 months ago
Paralle/Table.lua	Parallel optimization. ParallelTable inherits Container, unit tests	a year ago
Power lus	Use UNIX line endings	7 months ago
IR READWEINS	doc readfledoca	5 months ago
RReLUlus	Add randomized leaky rectified linear unit (RReLU)	3 months ago
P ReLUtes	adds in-place ReLU and foxes a potential divide-by-zero in nn.Sgrt	9 months ago
Replicate lua	Replicate batchMode	8 months ago
F) Reshape lus	Added more informative pretty-printing.	a year ago
R Selectiva	initial revenue of torch? tree	4 years ago
□ SelectTable lua	nn Module preserve type sharing semantics (#187); add nn Module apply	4 months ago
E Sequential lua	fixing Sequential remove corner case	6 months ago
R Sigmoid lua	initial reviews of torch? tree	4 years ago
F) Smooth 1Criterion kas	Add SizeAverage to criterions in the constructor	2 months ago
R SoftMax kas	Fix various unused variables in m	a year ago
E Sottlin lus	Fix various unused variables in nn	a year ago
Pi SofPlus kar	fixed a numerical issue in the SoffPlus module (it breaks for input g	2 years ago
R SofShirk tus	initial revenue of torch? tree	4 years ago
F) SoftSion lua	initial reverse of torch? tree	4 years ago
IR SparseJacobian ka	Fix various unused variables in m	a year ago
Sparred (near lux	Using sparse implementation of zeroGradParameters for SparseLinear	a month ago
F) Spatial Adaptive Mast Pooling	Added Soutial Adaptive May Pooling	a year ago
Spatial/versosPooling.lus	SpatialAveragePooling supports padding, cell mode and exclude_pad div	29 days ago
	Add C implementation of SostialilatchNormalization	7 days ago
R Spatial Contractive Normaliz	Make troof) buty recursive.	9 months ago
SpetalConvolution.kas	Fix hoe() in SpatialConvolution	3 months ago
Fil SpatialConvolutionMM.lua	Fix host in SoutistConvolution	3 months ago
SpatialConvolutionMap.lua	Remove unused and expensive initialization logic from nn Spatial Convo	8 months ago
F) SpatialCrossMapl RN km	cude consistency	18 days ago
Spatial DivisiveNormalizatio	Soatief Constructive Divisive Subtractive/Normalization work with bat	8 months ago
SpetaDropout los	small fix on error message	6 months ago
SpatialFractionalMadPoolin	Adding Fractional Max Pooling	3 months ago
Spatial Full Convolution Aug	Add adjustment form to Spatia Full Convolution to control the size of	5 days ago
SpetalFullConvolutionMap.lus		3 years ago
IR Spatial Provincias	SostialAveneePooling divides by KWNH	10 months ago
SpetaMaxPooling.lus	SpatialNasPooling supports padding and cell mode	6 months ago
SpatialMarUrpooling.lus	Add Soutieffest Incooling	9 morens ago 26 days ago
Spatial SoftMax lua	Update SoftMax to work in spatial mode	4 months ago
SpataSotMaxua     SpataSotMaxua	Update SoftMax to work in special mode  Mercie branch 'nn fast meet'	4 months ago 3 years ago
□ SpatalSubSampling lua     □ SpatalSubtractiveNormaliz	Merge branch 'nn_fast_reset'  Spatial[Constrastive,Divisive,Subtractive]Normalization work with bat	3 years ago 8 months ago
Spatial SubtractiveNormaliz     Spatial UpSamplingNearest I		8 months ago 7 months ago
	Use University and Control of the Co	r months ago
Pi SostalZeroPaddino.lus	Added more informative pretty-printing.	a year ago







#### 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') 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) return self.output 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) return self.gradInput

## **Example: Torch MulConstant**

$$f(X) = aX$$

initialization

forward()

backward()

# **Example: Caffe Layers**



memory_data_layer.cpp	dismantle layer headers	a month ago
multinomial_logistic_loss_la	dismantie layer headers	a month ago
mvn_layer.cpp	dismantie layer headers	a month ago
mvn_layer.cu	dismarible layer headers	a month ago
neuron_layer.cpp	dismaritie layer headers.	a month ago
pooling_layer.cpp	dismantie layer headers	a month ago
pooling_layer.cu	dismantie layer headers	a month ago
power_Jayer.cpp	dismantle layer headers	a month ago
D power_Jayer.cu	dismantle layer headers	a month ago
prelu_layer.cpp	dismantle layer headers	a month ago
prelu_layer.cu	dismantie layer headers	a month ago
reduction_layer.cop	dismantle layer headers	a month ago
reduction_layer.cu	dismaritie layer headers	a month ago
relu_layer.cop	dismantle layer headers	a month ago
nelu_layer.cu	dismantle layer headers	a mooth ago
reshape_layer.cpp	dismantle layer headers	a month ago
sigmoid_cross_entropy_los	dismantie layer headers	a month ago
sigmoid_cross_entropy_los	dismantle layer headers	a month ago
sigmoid_layer.cpp	dismantie layer headers	a month ago
sigmoid_tayer.cu	dismantie layer headers	a month ago
stience_layer.cpp	dismantie layer headers	a month ago
stience_layer.cu	dismantie layer headers.	a month ago
slice_layer.cpp	dismantie layer headers	a month ago
slice_layer.cu	dismantie layer headers	a month ago
softmax_layer.cpp	dismantle layer headers	a month ago
softmax_layer.cu	dismantie layer headers	a month ago
softmax_loss_layer.cpp	dismantie layer headers	a month ago
softmax_loss_layer.cu	dismantie layer headers	a month ago
spit_layer.cpp	dismarifie layer headers	a month ago
split_layer.cu	dismantie layer headers.	a month ago
spp_layer.cpp	dismantle layer headers	a month ago
atanh_layer.cpp	dismantie layer headers	a month ago
arh_layer.cu	dismantile layer headers	a month ago
threshold_layer.cpp	dismantie layer headers	a month ago
threshold_layer.ou	dismantie layer headers	a month ago
Tile_layer.cpp	dismantie layer headers	a month ago
Ette_tayer.cu	dismarifie layer headers	a month ago
window_data_layer.cpp	dismantle layer headers	a month ago

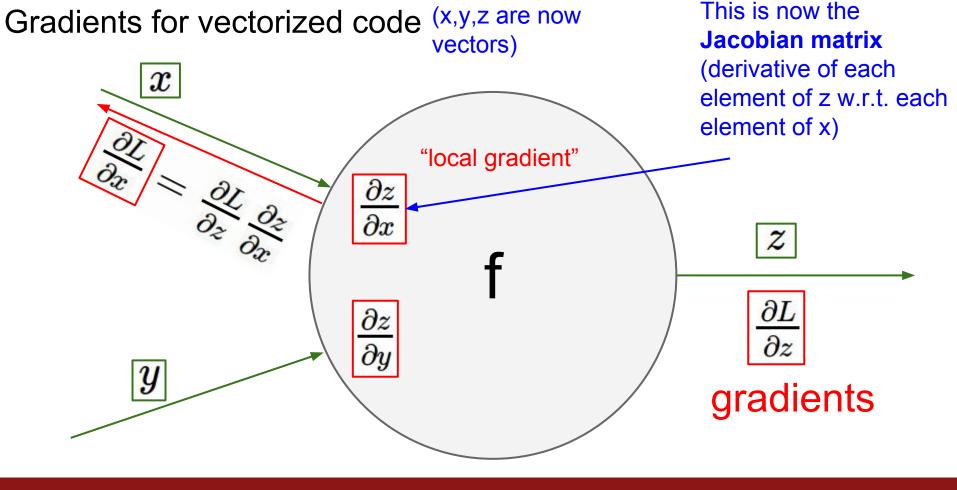
```
#include <cmath>
     #include <vector>
    #include "caffe/layers/sigmoid_layer.hpp"
    namespace caffe {
    template <typename Dtype>
    inline Dtype sigmoid(Dtype x) {
      return 1. / (1. + exp(-x));
     template <typename Dtype>
     void SigmoidLayer<Dtype>::Forward_cpu(const vector<Blob<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);
47 } // namespace caffe
```

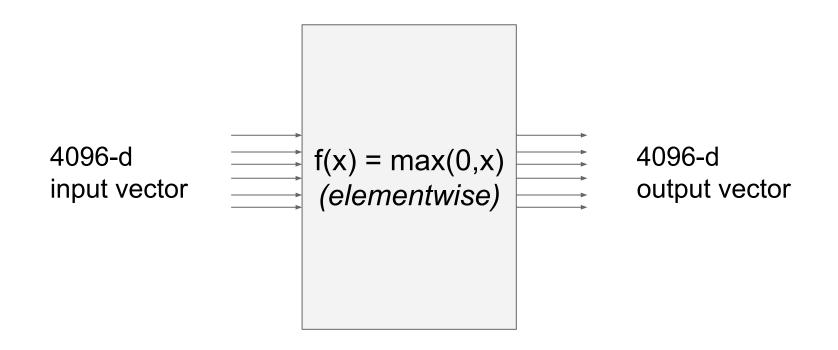
# Caffe Sigmoid Layer

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

$$(1-\sigma(x))\,\sigma(x)$$

 $(1-\sigma(x))\sigma(x)$  \*top\_diff (chain rule)





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

Jacobian matrix

4096-d output vector

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

Jacobian matrix

4096-d 
$$f(x) = max(0,x)$$
 (elementwise)

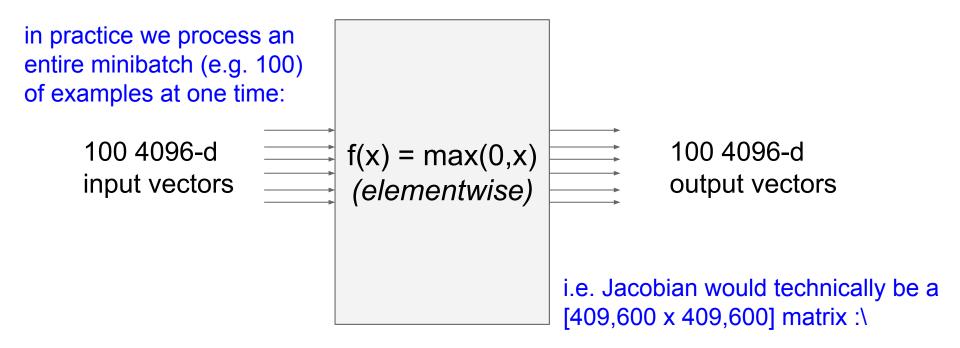
Q: what is the size of the Jacobian matrix?

[4096 x 4096!]

output vector

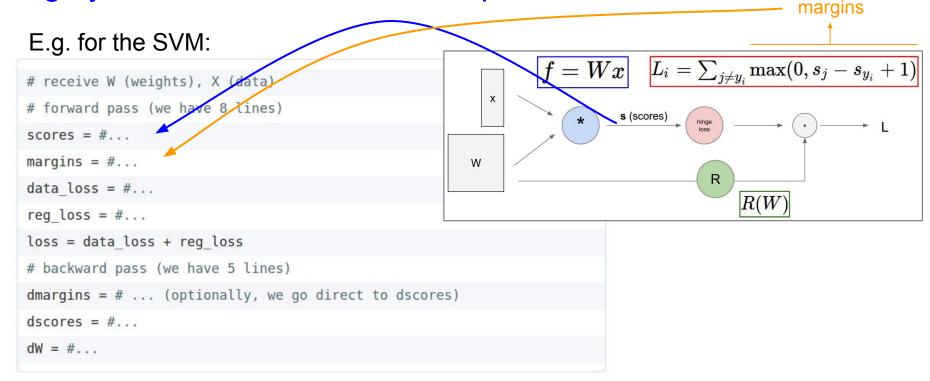
Q2: what does it look like?

4096-d



## Assignment: Writing SVM/Softmax

Stage your forward/backward computation!



# 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.



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

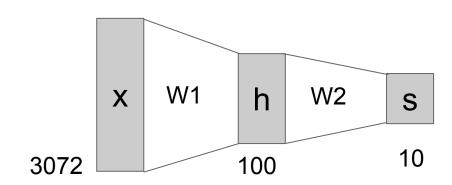
(**Before**) Linear score function: f = Wx

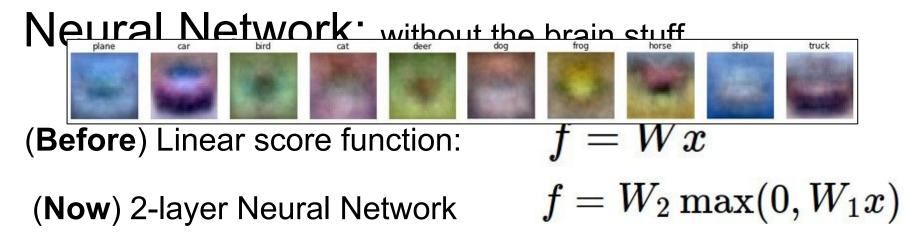
(Now) 2-layer Neural Network J

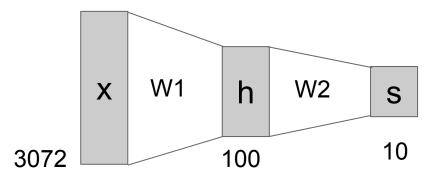
(**Before**) Linear score function:

(Now) 2-layer Neural Network

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







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

$$f = Wx$$

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

$$f=W_2\max(0,W_1x)$$

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

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

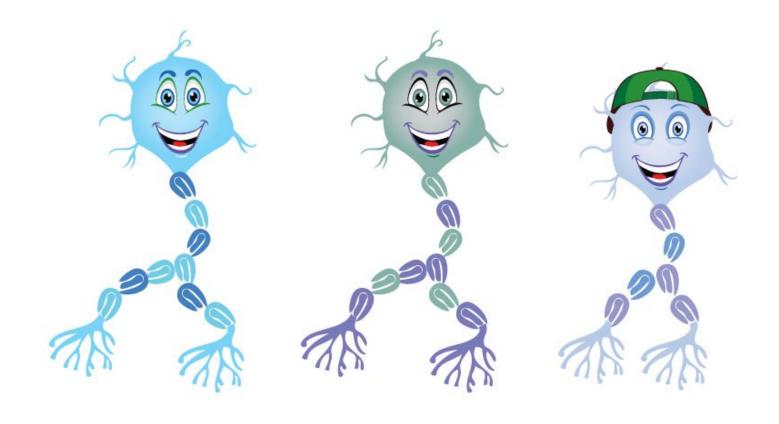
```
01.
      X = \text{np.array}([[0,0,1],[0,1,1],[1,0,1],[1,1,1]])
02.
      y = np.array([[0,1,1,0]]).T
03.
      syn0 = 2*np.random.random((3,4)) - 1
04.
      syn1 = 2*np.random.random((4,1)) - 1
05.
      for j in xrange (60000):
06.
           11 = 1/(1+np.exp(-(np.dot(X,syn0))))
07.
           12 = 1/(1+np.exp(-(np.dot(11,syn1))))
08.
           12 \text{ delta} = (y - 12) * (12 * (1-12))
09.
           l1 delta = 12 delta.dot(syn1.T) * (l1 * (1-l1))
10.
           syn1 += 11.T.dot(12 delta)
11.
           syn0 += X.T.dot(11 delta)
```

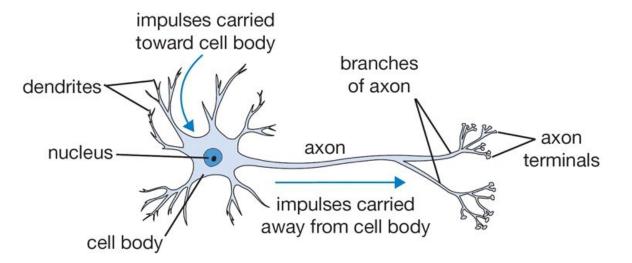
from @iamtrask, http://iamtrask.github.io/2015/07/12/basic-python-network/

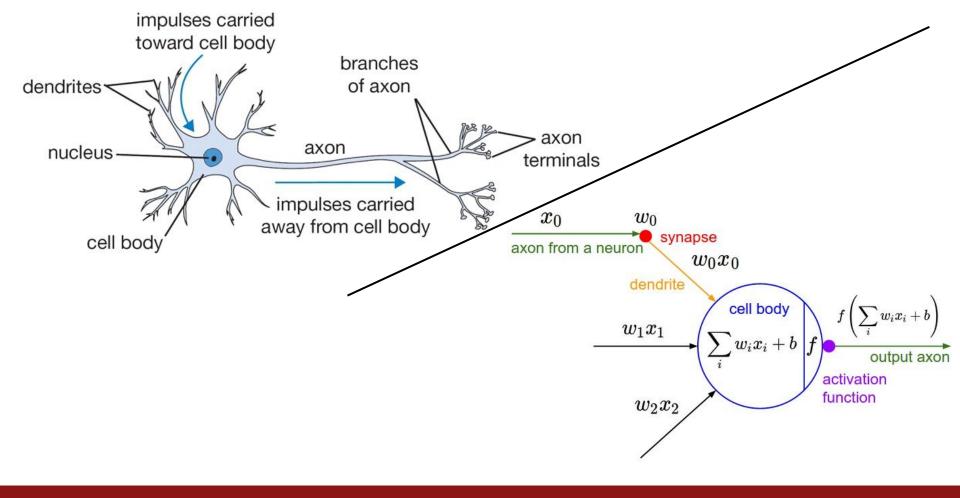
# Assignment: Writing 2layer Net Stage your forward/backward computation!

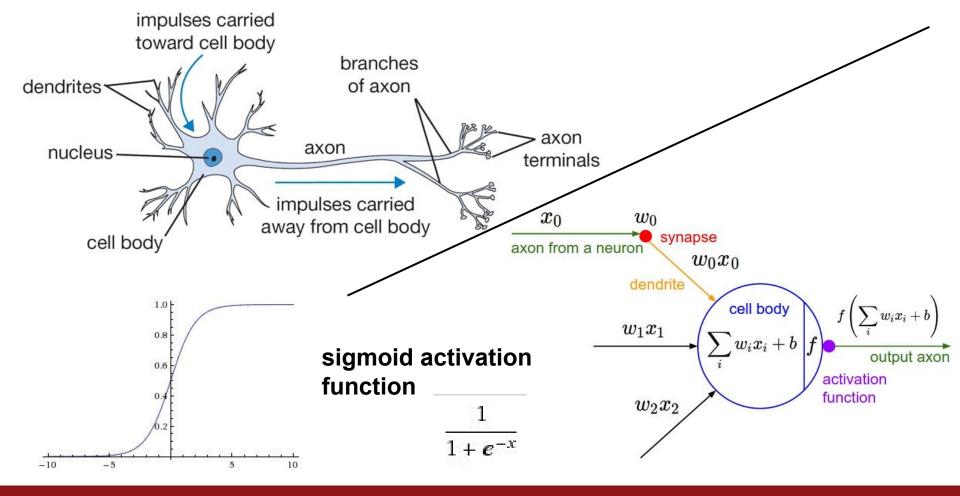
```
# receive W1,W2,b1,b2 (weights/biases), X (data)
# forward pass:
h1 = \#... function of X,W1,b1
scores = #... function of h1, W2, b2
loss = #... (several lines of code to evaluate Softmax loss)
# backward pass:
dscores = #...
dh1, dW2, db2 = #...
dW1, db1 = #...
```

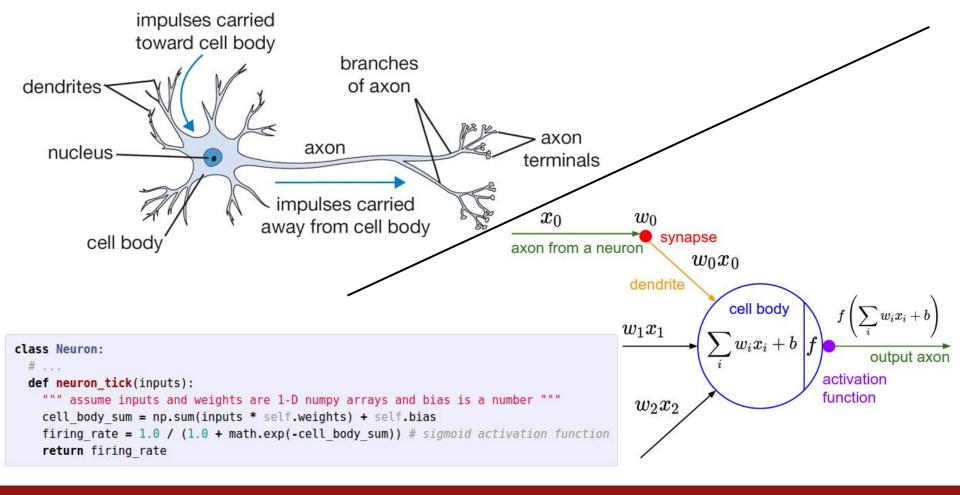








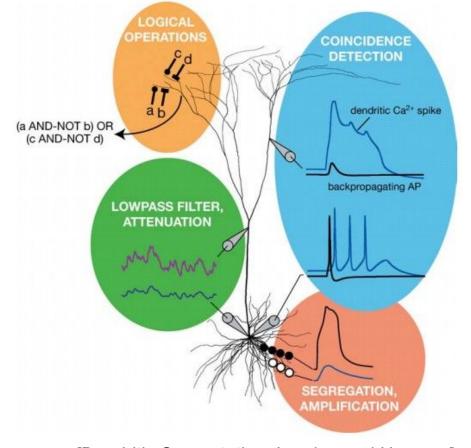




#### Be very careful with your Brain analogies:

#### **Biological Neurons:**

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



[Dendritic Computation. London and Hausser]

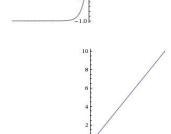
# **Activation Functions**

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

Leaky ReLU max(0.1x, x)

# **Sigmoid**

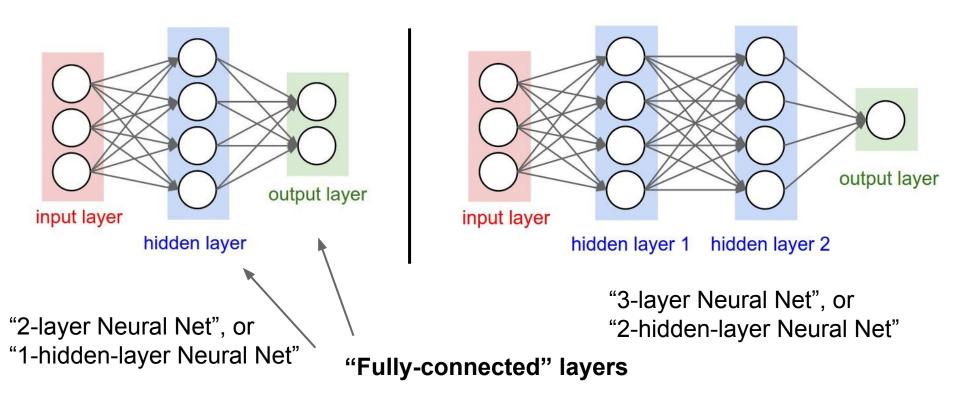
#### tanh(x) tanh



## **Maxout** $\max(w_1^T x + b_1, w_2^T x + b_2)$ $f(x) = \begin{cases} x & \text{if } x > 0\\ \alpha (\exp(x) - 1) & \text{if } x \le 0 \end{cases}$ **ELU**

ReLU max(0,x)

## **Neural Networks: Architectures**

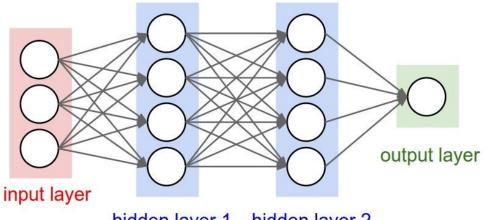


### Example Feed-forward computation of a Neural Network

```
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.

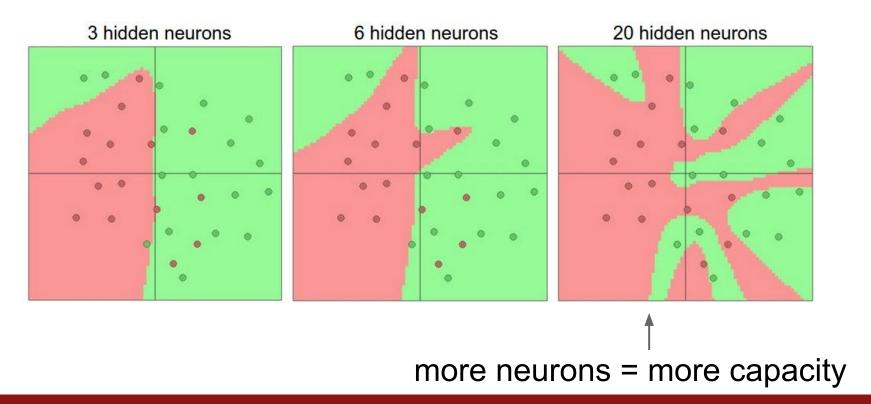
### Example Feed-forward computation of a Neural Network



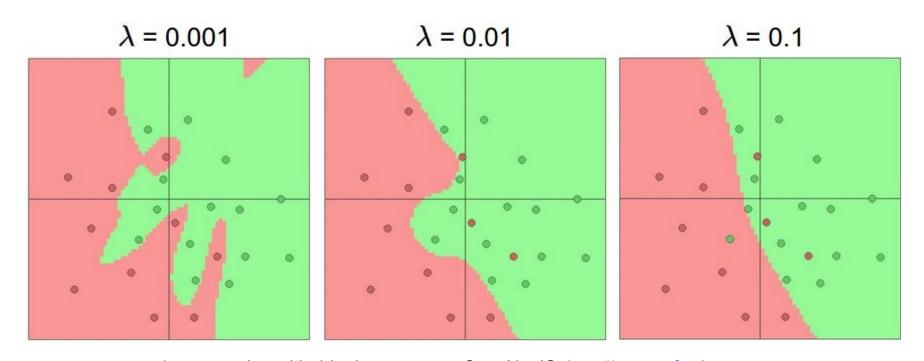
hidden layer 1 hidden layer 2

```
# 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



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



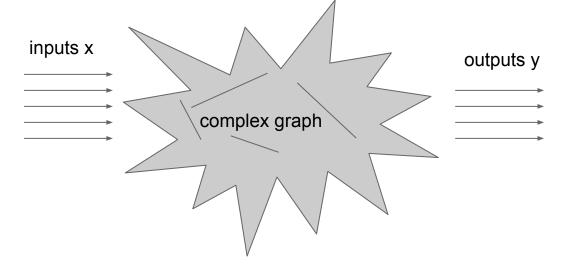
(you can play with this demo over at ConvNetJS: <a href="http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html">http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html</a>)

## **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