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

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

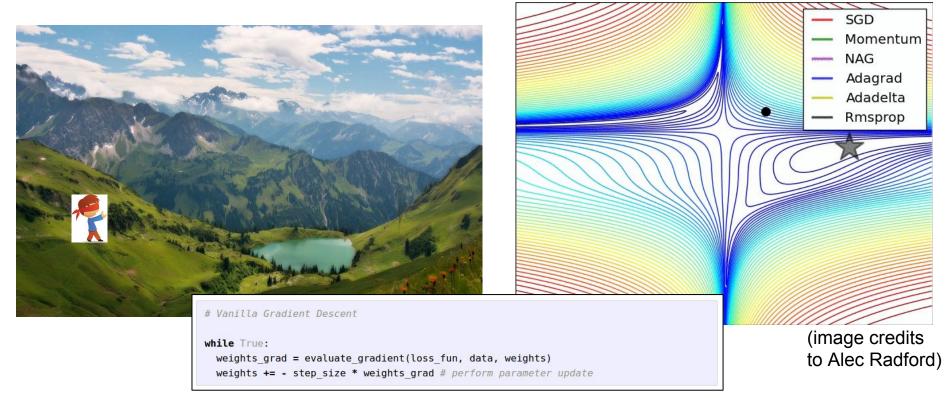
SVM loss

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

data loss + regularization

want $\nabla_W L$

Optimization



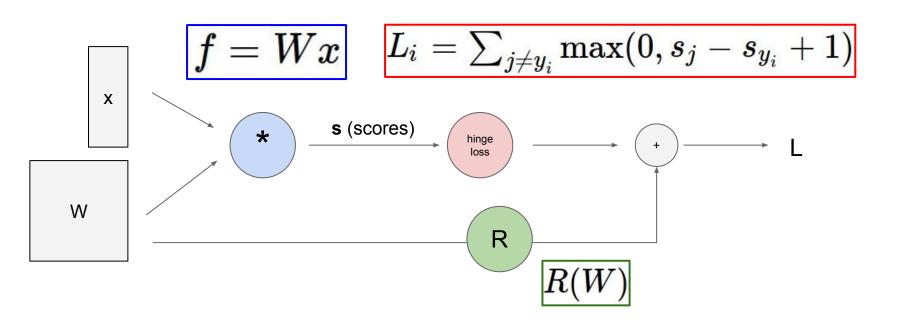
Gradient Descent

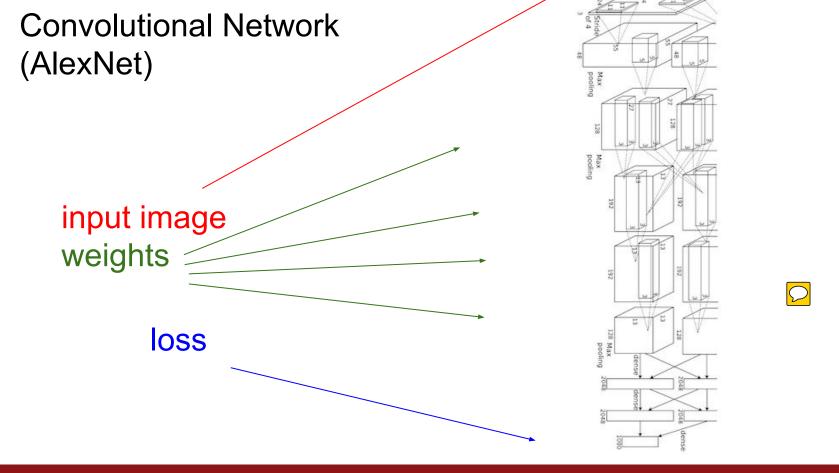
$$rac{df(x)}{dx} = \lim_{h o 0} rac{f(x+h) - f(x)}{h}$$

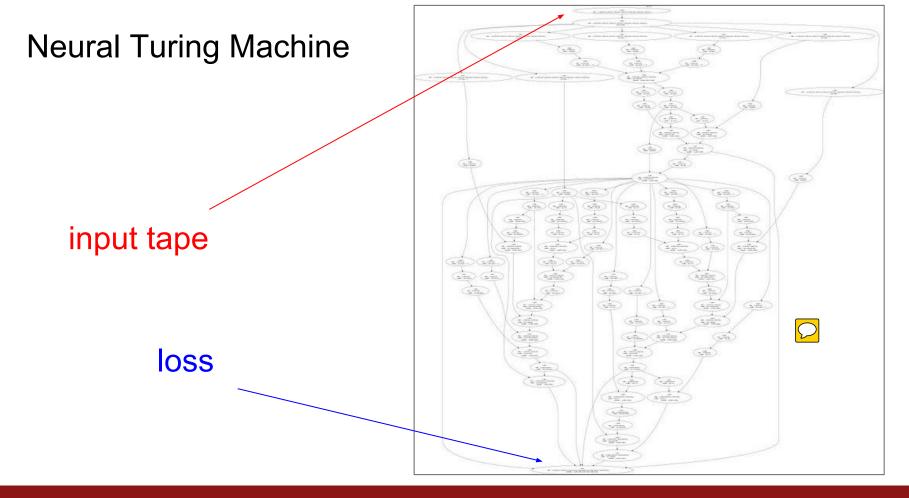
Numerical gradient: slow :(, approximate :(, easy to write :) Analytic gradient: fast:), exact:), error-prone:(

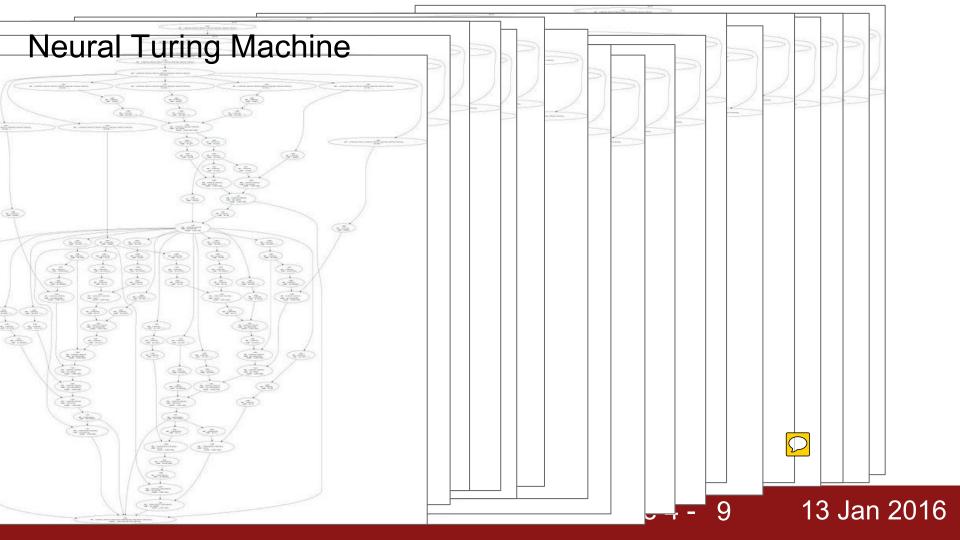
In practice: Derive analytic gradient, check your implementation with numerical gradient

Computational Graph



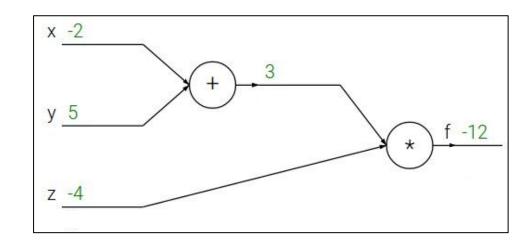






$$f(x, y, z) = (x + y)z$$

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





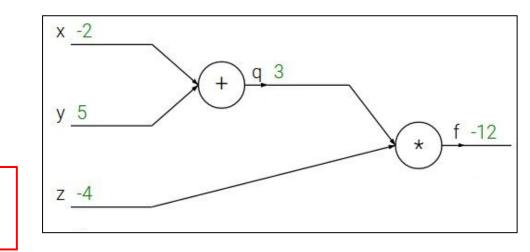
$$f(x, y, z) = (x + y)z$$

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

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

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

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

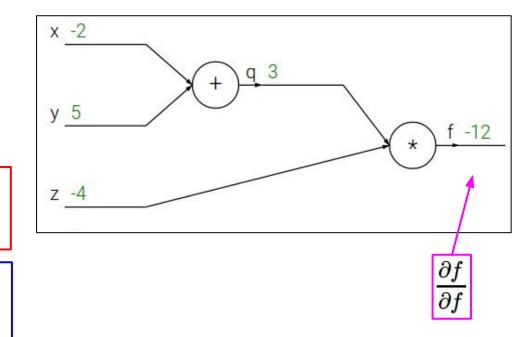


$$f(x, y, z) = (x + y)z$$

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

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

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

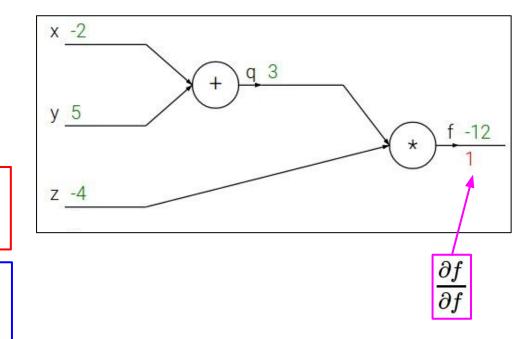


$$f(x, y, z) = (x + y)z$$

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

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

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

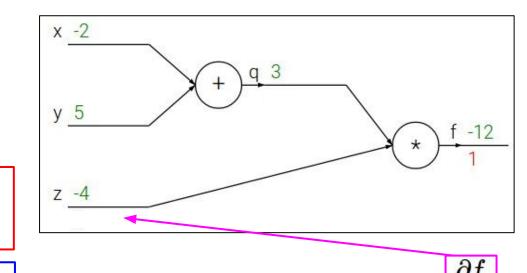


$$f(x, y, z) = (x + y)z$$

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

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

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



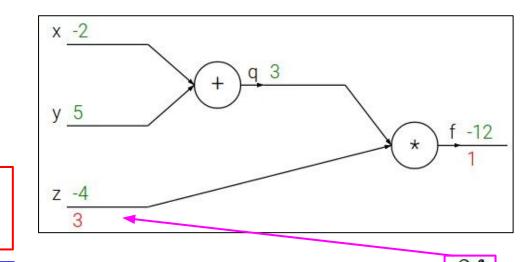
$$f(x, y, z) = (x + y)z$$

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

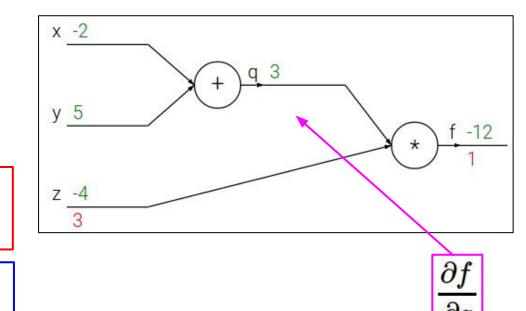


$$f(x, y, z) = (x + y)z$$

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

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

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



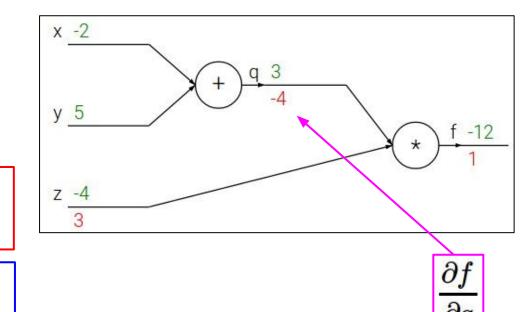
$$f(x, y, z) = (x + y)z$$

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

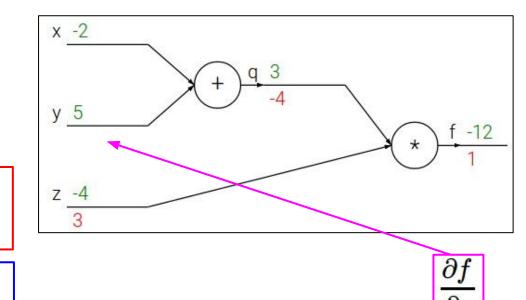


$$f(x, y, z) = (x + y)z$$

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

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

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

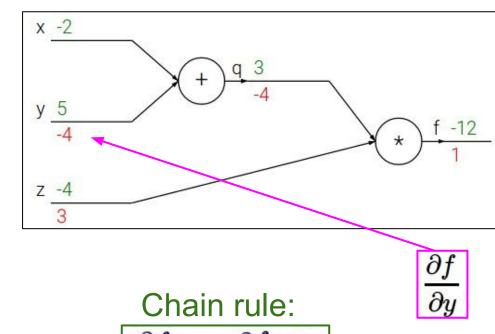


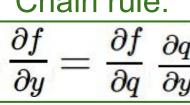
$$f(x, y, z) = (x + y)z$$

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

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

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



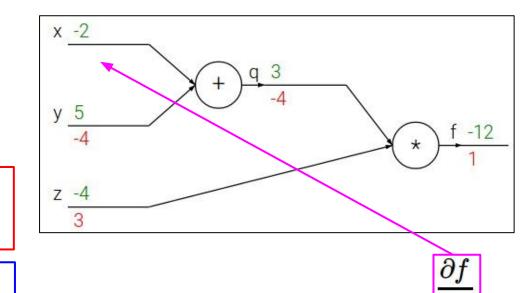


$$f(x, y, z) = (x + y)z$$

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

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

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

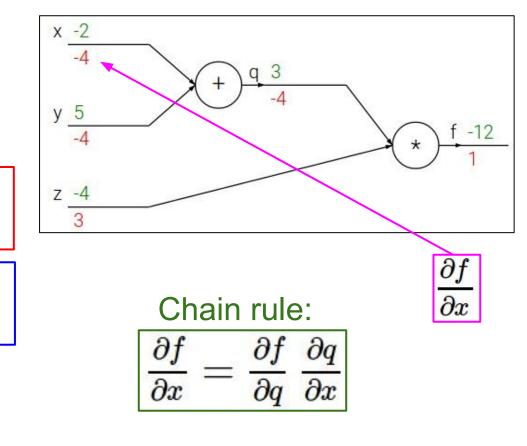


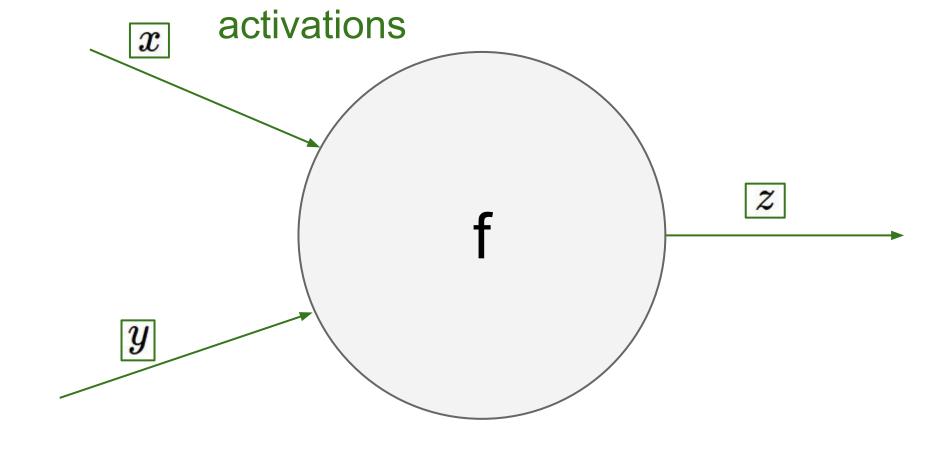
$$f(x, y, z) = (x + y)z$$

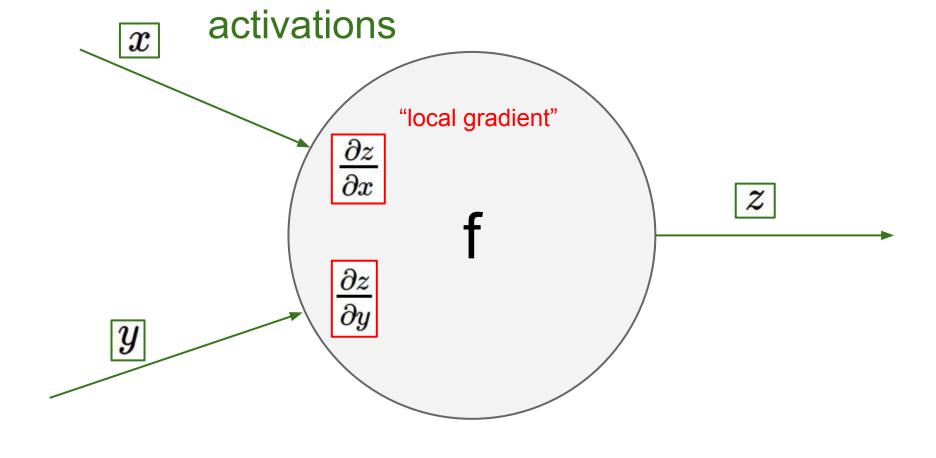
e.g. $x = -2$, $y = 5$, $z = -4$

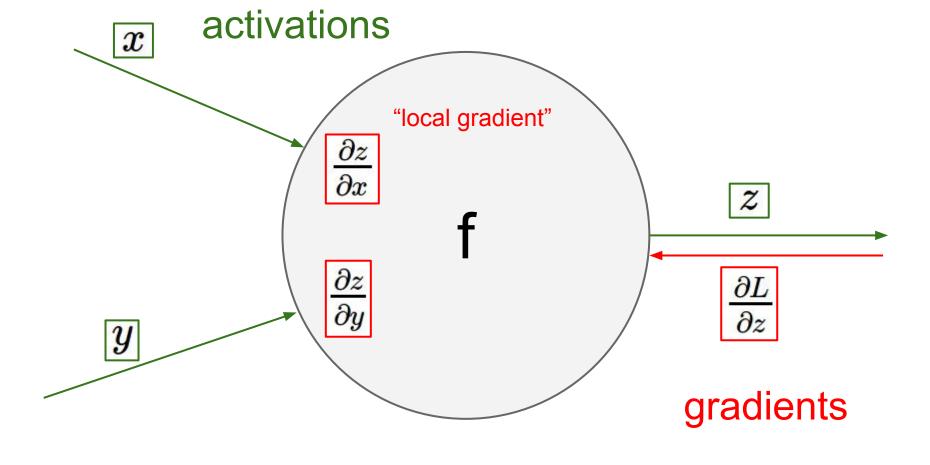
$$q=x+y \qquad rac{\partial q}{\partial x}=1, rac{\partial q}{\partial y}=1$$

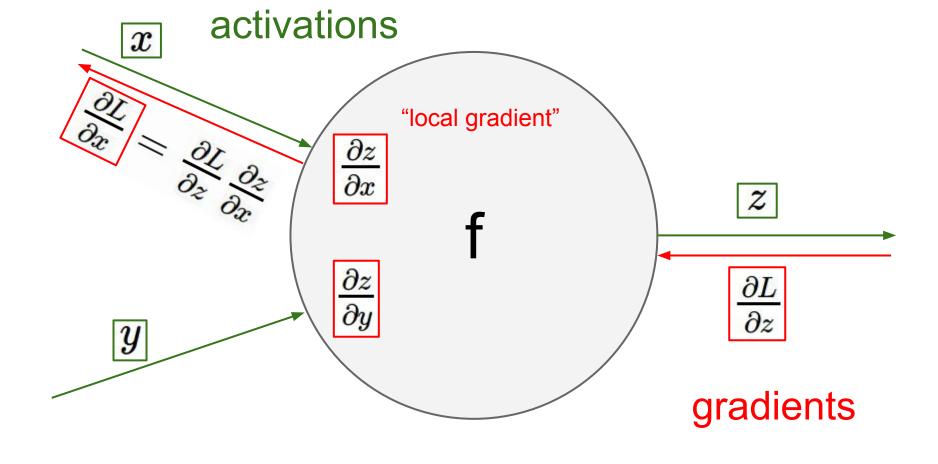
$$f=qz$$
 $rac{\partial f}{\partial q}=z, rac{\partial f}{\partial z}=q$

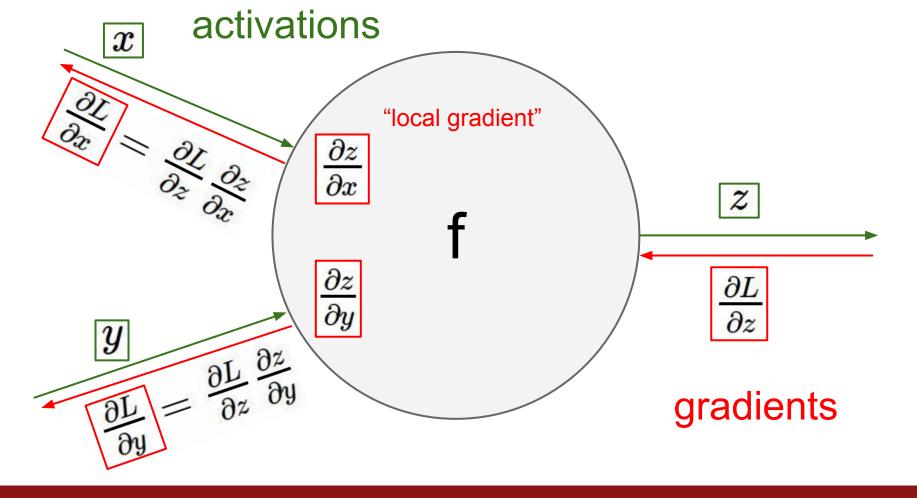


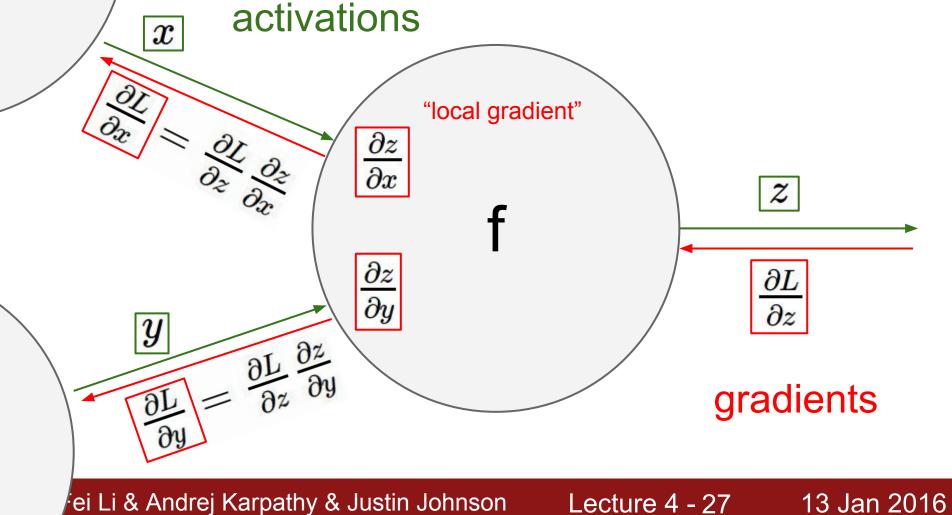




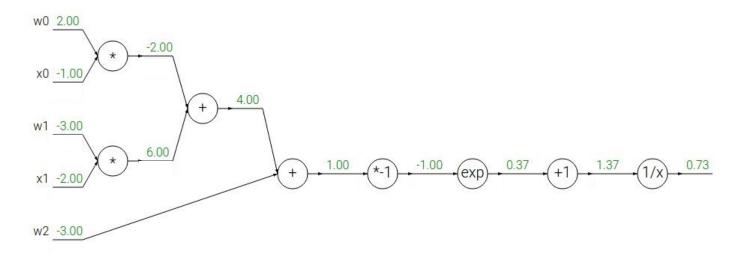




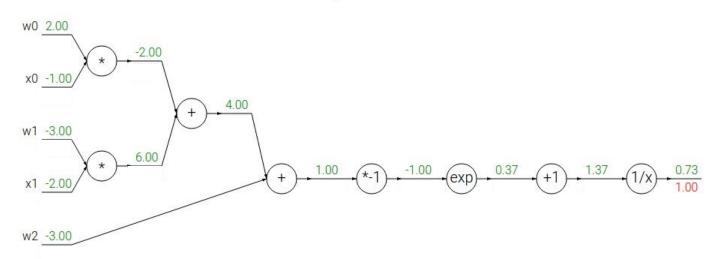




Another example: $f(w,x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$

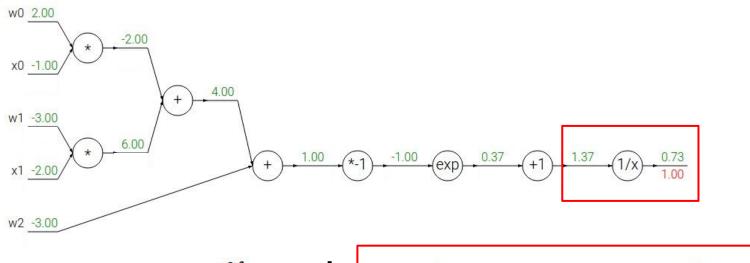


$$f(w,x)=rac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



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

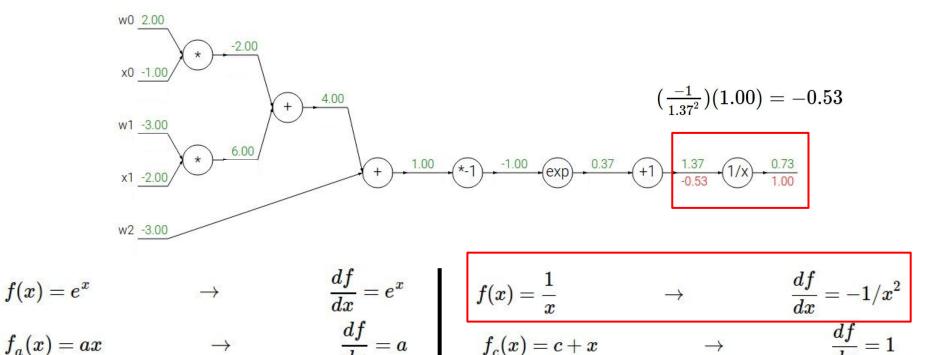
$$f(w,x)=rac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



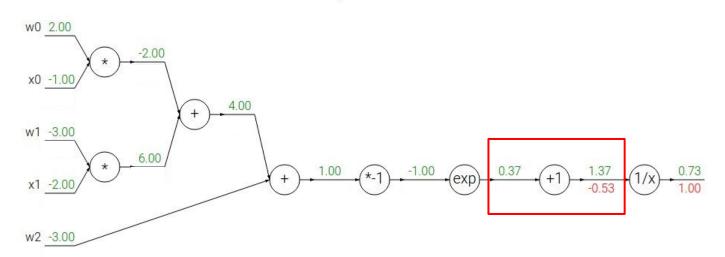
$$egin{aligned} f(x) = e^x &
ightarrow & rac{df}{dx} = e^x \ & &
ightarrow & rac{df}{dx} = a \end{aligned}$$

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

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

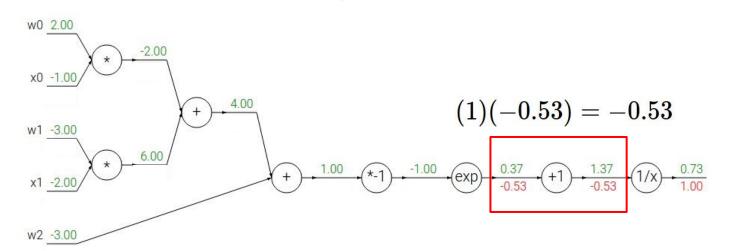


$$f(w,x)=rac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

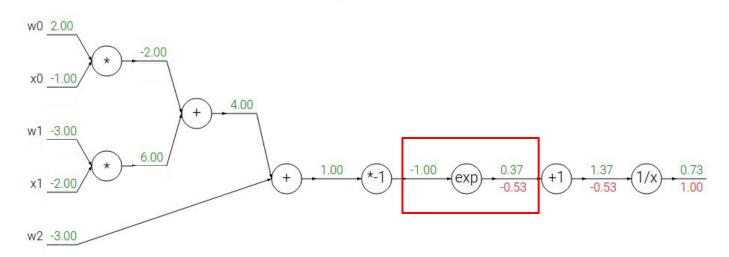


$$f(x)=e^x \hspace{1cm} o \hspace{1cm} rac{df}{dx}=e^x \hspace{1cm} f(x)=rac{1}{x} \hspace{1cm} o \hspace{1cm} rac{df}{dx}=-1/x^2 \ f_a(x)=ax \hspace{1cm} o \hspace{1cm} rac{df}{dx}=a \hspace{1cm} f(x)=c+x \hspace{1cm} o \hspace{1cm} rac{df}{dx}=1$$

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

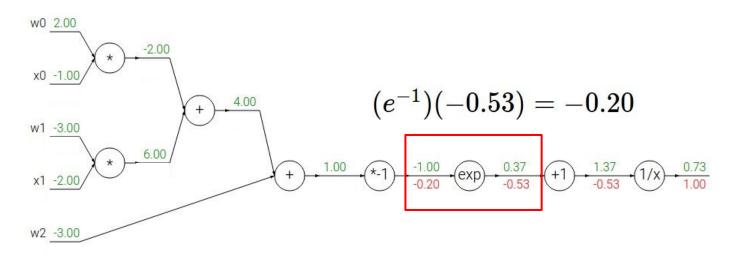


$$f(w,x)=rac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

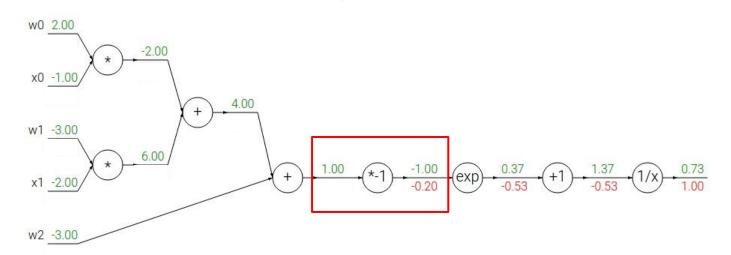


$$f(x)=e^x \qquad
ightarrow \qquad rac{df}{dx}=e^x \ f(x)=ax \qquad
ightarrow \qquad rac{df}{dx}=a \ f(x)=ax \qquad
ightarrow \qquad f(x)=ax \qquad
ightarrow \qquad f(x)=ax \qquad \qquad f(x)=ax$$

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



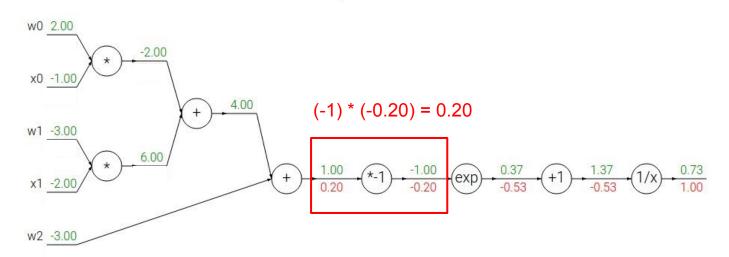
$$f(w,x) = rac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2}}$$



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

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

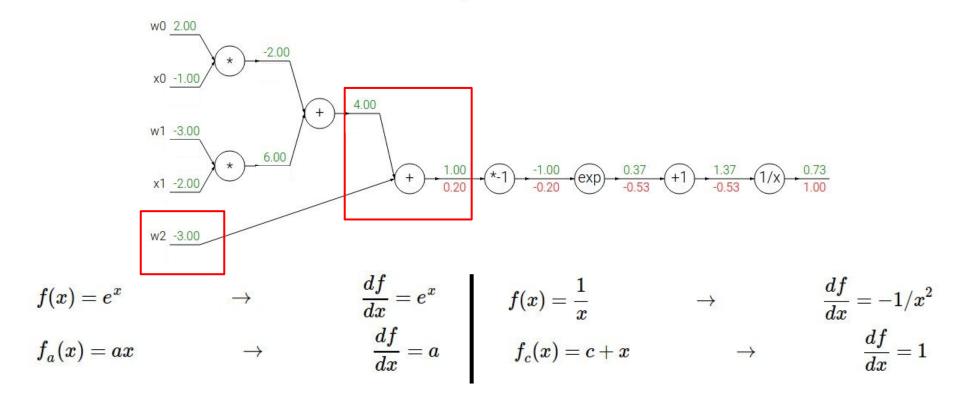
$$f(w,x)=rac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



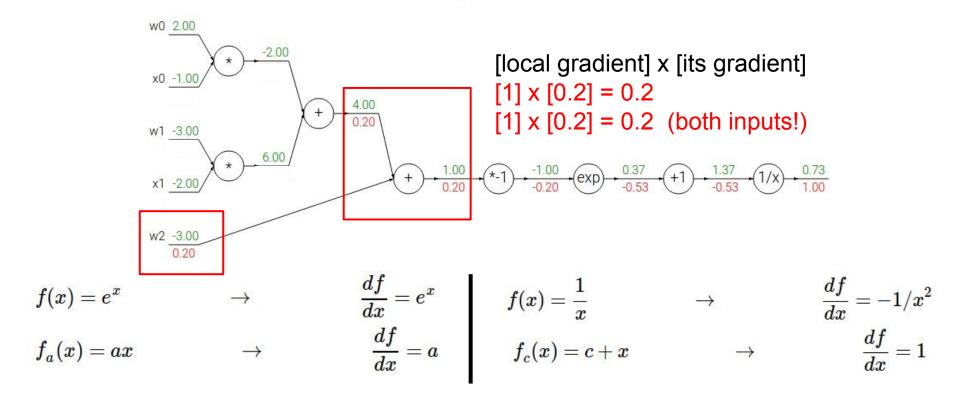
$$f(x) = e^x \qquad o \qquad rac{df}{dx} = e^x \ f_a(x) = ax \qquad o \qquad rac{df}{dx} = a$$

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

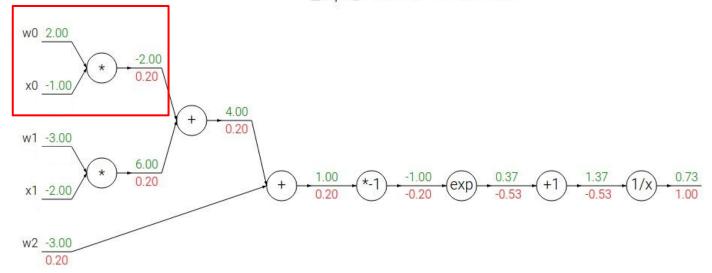
$$f(w,x)=rac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



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



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

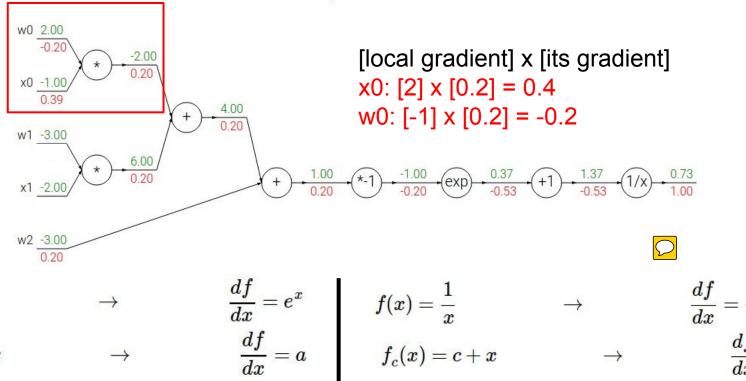


$$egin{array}{lll} f(x)=e^x &
ightarrow & rac{df}{dx}=e^x & f(x)=rac{1}{x} &
ightarrow & \ f_a(x)=ax &
ightarrow & rac{df}{dx}=a & f_c(x)=c+x &
ightarrow &
ightarrow$$

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

Another example:

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



$$f(x)=e^x \hspace{1cm} o \ f_a(x)=ax \hspace{1cm} o$$

$$ightarrow rac{df}{dx}=$$

$$f(x) = \frac{1}{x}$$

$$f_c(x)=c+x$$

$$rac{df}{dx} = -1/x^2$$

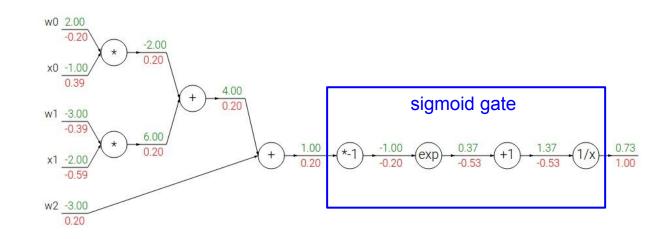
$$rac{df}{dx}=1$$

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

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

sigmoid function

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



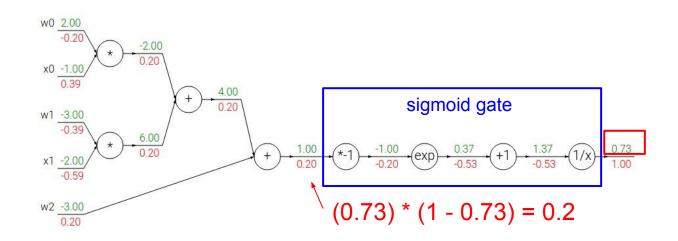


$$f(w,x)=rac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

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

sigmoid function

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

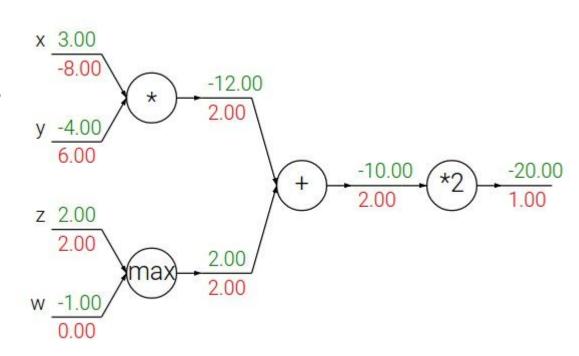


Patterns in backward flow

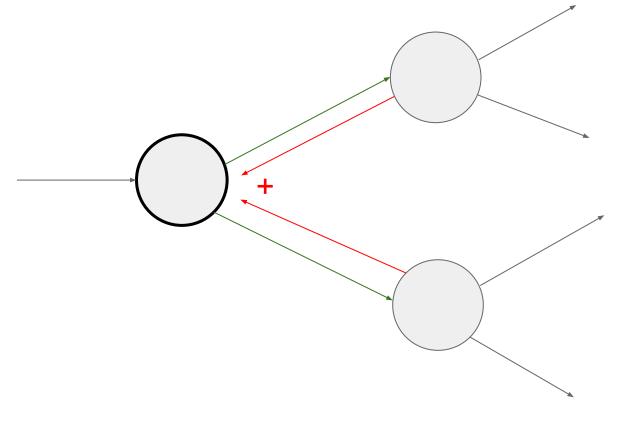
add gate: gradient distributor

max gate: gradient router

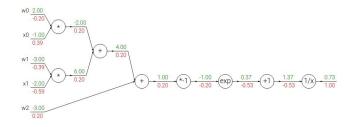
mul gate: gradient... "switcher"?



Gradients add at branches



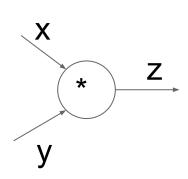
Implementation: forward/backward API



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
```

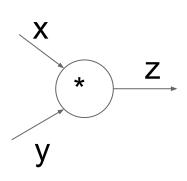
Implementation: forward/backward API



(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



■ LogSigmoid lue	Add THNN conversion of (ELU, LeskyReLU, LogSigmoid, LogSofMax, Looku	7 days ag
□ LogSofMax.lua	Add THNN conversion of (ELU, LeekyReLU, LogSigmoid, LogSofMax, Looku	7 days ag
	Harmonize LookupTable signature with ours impl	5 days ag
	Rename unpack to table unpack for Lus 5.2	8 months ag
MSECriterion.lua	Add SizeAverage to criterions in the constructor.	2 months ag
MarginCriterion.kus	modernized MarginCriterion	a year ag
MarginRankingCriterion kus	Fix batch mode in MarginRankingCriterion	4 days ag
Max.lua	Merge pull request #464 from vgire/master	2 months ag
Mean.lus	Add support for negative dimension and both batch and non batch input	2 months ag
MinJue	Merge pull request #464 from vgire/master	2 months ag
MatureTable.lua	cancel unused variable and useless expression	29 days ag
Module.lus	Revert "Don't re-Batten parameters if they are already flattered"	15 hours ag
€ Mellus	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 ParadelCriterion add	2 months ag
MultiLabeMarginCriterion.lua	initial reviamp of torch? tree	4 years ag
MultiMarginCriterion.lua	multimargin supports p=2	11 months ag
NarrowJua	typeAs in Narrow not done in place.	6 months ag
NarrowTable.lua	NarrowTable	6 months ag
Normaliza lua	Remove brim and baddbrim from Normaliza, because they allocate memory,	20 days ag
	Buffers for PReLU cuda 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	for a bug in conditional expression	a month ag
ParallelCriterion.lux	asserts in MultiCriterion and PanallelCriterion add	2 months ag
Paralle/Table.lua	Parallel optimization. ParallelTable inherits Container, unit tests	a year ag
Power.las	Use UNIX line endings	7 months ag
README.md	doc readthedocs	5 months ag
☐ RReLUtus	Add randomized leaky rectified linear unit (RReLU)	3 months ag
□ ReLUlus	adds in-place ReLU and foxes a potential divide-by-zero in nn.Sgrt	9 months ag
Replicate Jua	Reglicate batchWode	8 months ag
☐ Reshape.lus	Added more informative pretty-printing.	a year ag
Selectiva	initial revenip of torch? 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 revamp of torch? tree	4 years ag
SmoothL1Criterion.lus	Add SizeAverage to criterions in the constructor	2 months ag
SoftMax.lua	Fix various unused variables in nn	a year ag
☐ Sottlife Jus	Fix various unused variables in nn	a year ag
SoftPlus λas	fixed a numerical issue in the SofPlus module (it breaks for input g	2 years ag
SoftShrink.lua	initial revamp of torth? tree	4 years ag
⊕ SotSign.lus	initial reversity of torch? thee	4 years ag
SperseJacobian ka	Fix various unused variables in nn	a year ag
☐ SparseLinear lus	Using sparse implementation of zeroGradParameters for SparseLinear	a month ag
	Added Spatial Adaptive MacPooling	a year ag
Spatial/waragePooling.lua	Spatial/veragePooling supports pedding, cell mode and exclude_pad div	29 days ag
	Add C implementation of SpatialBatchNormalization	7 days ag
SpatialContrastiveNormalz	Make type() truly recursive.	9 months ag
SpatialConvolution lua D	Fix type() in SpatialConvolution	3 months ag
SpatialConvolutionAffV.lux	Fix type() in SpatialConvolution	3 months ag
SpatialConvolutionMap.lua	Remove unused and expensive initialization logic from nn Spatial Convo usda consistency	8 months ag
SpetialCrossMepLRN.lus SpetialDivisiveNormalizatio		18 days ag 8 months ao
	Spatial(Constructive, Divisive, Subtractive)Normalization work with but small fix on error message	8 months ag 6 months ag
■ SpetalDropout.lua		
El Santiefontinedhad		
SpatialFractionalMaxPoolin Control of Control of the Landson of the L	Adding Fractional Max Pooling	3 months ag
SpatialFullConvolution.kua	Adding Practional Max Pooling Add adjustment term to Spatial-Full Convolution to control the size of	3 months ag 5 days ag
SpetalFullConvolution.kus	Adding Fractional Max Pooling Add adjustment form to SpatialFullConvolution to control the size of New NN classes	3 months ag 5 days ag 3 years ag
SpatialFullConvolution.ka SpatialFullConvolutionMap.ka SpatialLPPooling.ka	Adding Fractional Max Pooling Add adjustment form to Spatial-EulConvolution to control the size of New NN classes SpatialAveragePooling divides by kNYHH	3 months ag 5 days ag 3 years ag 10 months ag
SpatialFullConvolution Aus SpatialFullConvolutionMap Aus SpatialLPPooling Jua SpatialLPPooling Jua SpatialMaxPooling Jua	Adding Fractional Max Proling Add signatures form to SpatiaFullConvolution to control the size of New NN classes Spatial-AverageProling divides by NN*HH Spatial-AverageProling supports pedfing and cell mode	3 months ag 5 days ag 3 years ag 10 months ag 6 months ag
SpatialFulConvolution.lua SpatialFulConvolutionMap.lua SpatialFulConvolutionMap.lua SpatialFulConvolutionMap.lua SpatialfulCooling.lua SpatialfulCooling.lua	Adding Fractional Max Probing Add adjustment term to SpokeF-UTC-weakdon to control the size of New TeV Classes Spoked NewsgeProbing divides by NTPH Spoked NewsgeProbing apports getting and cell mode Add SpokeMAX-	3 months ag 5 days ag 3 years ag 10 months ag 6 months ag 20 days ag
SpatiaFullConvolution.lua SpatiaFullConvolutionMep.lua SpatiatFullConvolutionMep.lua SpatiatMaxPooling.lua SpatiatMaxPooling.lua SpatiatMaxPooling.lua SpatiatMaxLua	Adding Precious Man Proling Add adjusted in time 16 global Full Convolution to control the size of New Not Comers Spatial Average Proving divides by INP14 Signatural Engineering Spatial New Open pacting and only mode Add Spatial Engineering Spatial	3 months ap 5 days ap 3 years ap 90 months ap 2 disays ap 2 disays ap 4 months ap
SpatiaFul/Convolution Jua SpatiaFul/ConvolutionMap Jua Spatial PPooling Jua Spatial MaxPooling Jua Spatial MaxPooling Jua Spatial MaxInpooling Jua Spatial MaxInpooling Jua Spatial Satt Max	Adding Freedomal Mass Pholing Adding Spreedomal Mass Pholing Adding Spreedom International Transport Control State Spreedom Research State State Spreedom American Spreedom Adding Spreedom Spreedom Adding Spreedom Adding Spreedom Adding Spreedom	3 months ag 5 days ag 3 years ag 10 months ag 6 months ag 20 days ag 4 months ag 3 years ag 3 years ag
SpatiaFull Convolution Jus SpatiaFull Convolution Map Jus Spatial E-Pooling Jus Spatial E-Pooling Jus Spatial March pooling Jus Spatial March pooling Jus Spatial SubSouther Jus	Adoley Transcribe Life Printing Adolegation of two to System Food Distriction of Lorentz the value of New 101 classes: Special Anneal Printing Adoles by 1977-14 Special Anneal Printing Adoles by 1977-14 And Special Anneal Printing Adoles Adoles Special Anneal An	3 months ag 5 days ag 3 yeans ag 6 months ag 26 days ag 4 months ag 3 years ag 8 months ag 8 months ag
SpatiaFul/Convolution Jua SpatiaFul/ConvolutionMap Jua Spatial PPooling Jua Spatial MaxPooling Jua Spatial MaxPooling Jua Spatial MaxInpooling Jua Spatial MaxInpooling Jua Spatial Satt Max	Adding Freedomal Mass Pholing Adding Spreedomal Mass Pholing Adding Spreedom International Transport Control State Spreedom Research State State Spreedom American Spreedom Adding Spreedom Spreedom Adding Spreedom Adding Spreedom Adding Spreedom	3 months ag 5 days ag 3 years ag 10 months ag 6 months ag 20 days ag 4 months ag 3 years ag 3 years ag

Example: Torch Layers



€ LogSigmoid lug	Add THNN conversion of (ELU, LeakyReLU, LogSigmold, LogSoffMax, Losku	7 days ago
E LogSofMax kas	Add THNN conversion of (ELU, LeakyReLU, LogSigmoid, LogSofMax, Looku	7 days ago
	Harmonize LookupTable signature with curn impl	5 days ago
⊞ MMJus	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
Maxius	Merge pull request #464 from vgire/master	2 months ago
Mean.lus	Add support for negative dimension and both batch and non batch input	2 months ago
Mirulus	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-flatien parameters if they are already flatiened"	15 hours ago
Mullia	nemoving the requirement for providing size in nn.Mul	a year ago
MulConstant.lus	Ignore updateGradinput if self gradinput is nil	3 months ago
MultiCriterion.lua	asserts in MultiCriterion and ParallelCriterion add	2 months ago
MultiLabeMarginCriterion.lua	initial reverse of terch? tree	4 years ago
MultiMarginCriterion.lua	multimargin supports p=2	11 months ago
NarrowJua	typeAs in Narrow not done in place.	6 months ago
NarrowTable.lua	NerrowTable	6 months ago
Nomalizatua	Remove brinn and beddbrinn from Normalize, because they allocate memory,	20 days ago
PReLUJua	Buffers for PReLU cude implementation.	8 months ago
Padding.lus PairwiseDistance.lus	fixed broken nn.Padding: input was returned in backgrop Merce out request #532 from xwpenginester.	5 months ago 29 days ago
Parallel ka	fix a bug in conditional expression	a month ago
ParallelCriterion.lus ParallelTable.lus	asserts in Multi-Criterion and Parallel Criterion add	2 months ago
E) Paraser aceus	Parallel optimization. ParallelTable inherits Container, unit tests Use UNIX line endings	a year ago 7 months ago
E) POWERUS E) READWE not	doc readhedos	
REPORT NO.	Add randomized leaky recified linear unit (RRei,U)	5 months ago 3 months ago
E) Rel Ulina	adds in-place ReLU and fixes a potential divide-by-zero in nn. Sgrt.	9 months ago
R Replicate lus	Replicate batchMode	8 months ago
P. Restancius	Added more informative pretty-printing.	a year ago
Selection	initial revenue of torch? tree	4 years ago
☐ SelectTable lus ☐ SelectTable lus ☐ Description ☐ De	nn Module preserve type sharing semantics (#187); add nn Module apply	4 months ago
Sequential lua	fixing Sequential remove corner case	6 months ago
E Signoid has	initial revenue of terch? tree	4 years ago
F) Smooth 1Criterion kas	Add SizeAverage to criterions in the constructor	2 months ago
(ii) SoftMax.lua	Fix various unused variables in nn	a year ago
⊕ Sottlin lus	Fix various unused variables in nn	a year ago
© SoftPlus tue	fixed a numerical issue in the SoftPlus module (it breaks for input g	2 years ago
€ SoftShrink.tup	initial reverse of torch? tree	4 years ago
⊕ SoftSign.lus	initial reversp of torch? tree	4 years ago
SparseJacobian kus	Fix various unused variables in nn	a year ago
E SparseUnear lua	Using sparse implementation of zeroGradParameters for SparseLinear	a month ago
SpatialAdaptiveMasPooling	Added SpatialAdaptiveMaxPooling	a year ago
SpatialAveragePooling.lua	SpatialAveragePooling supports padding, cell mode and exclude_pad div	29 days ago
SpetialBatchNormalization.lue	Add C implementation of SpatialBatchNormalization	7 days ago
SpatialContrastiveNormaliz	Make type() truly recursive.	9 months ago
	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
SpetisiCrossMepLRN λus	cude consistency	18 days ago
Spatial Divisive Normalizatio	Spatial(Constructive, Divisive, Subtractive)Normalization work with bat	8 months ago
■ SpetaiDropout.lue	small fix on error message	6 months ago
SpatialFractionalMaxPoolin	Adding Fractional Max Pooling	3 months ago
SpatialFullConvolution.lua	Add adjustment ferm to Spatia/FullConvolution to control the size of	5 days ago
□ SpatialFullConvolutionMap.lua		3 years ago
SpatialLPPcoling.lua	SpatialAveragePooling divides by kWNH	10 months ago
■ SpetaMaxPooling lus	SpatialNasPooling supports padding and cell mode	6 months ago
SpatialMaxUrpooling lua	Add SpatistNexUnpooling	26 days ago
SpatialSoftMaxius	Update SoftMax to work in spetial mode	4 months ago
SpetialSubSampling.lua	Merge branch 'nn_fast_reset'	3 years ago
Spatial SubtractiveNormaliz	Spatial(Constructive, Divisive, Subtractive)Normalization work with bet	8 months ago
■ SpatialUpSamplingNearest1		7 months ago
SpatialZeroPadding.lua	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	dismantle layer headers.	a month ago
mvn_tayer.cpp	dismantie layer headers	a month ago
myn_layer.cu	dismanife layer headers	a month age
neuron_layer.cpp	dismarifie 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	dismantie 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	dismanifie layer headers	a month ago
relu_layer.cpp	dismanile layer headers.	a month ago
nelu_layer.cu	dismantle layer headers	a moeth ago
reshape_layer.cpp	dismantie layer headers	a month ago
sigmoid_cross_entropy_los	dismantle layer headers	a month ago
sigmoid_cross_entropy_los	dismantie layer headers	a month ago
sigmoid_layer.cpp	dismantie layer headers	a month ago
sigmoid_tayer.cu	dismanile layer headers	a month ago
stience_layer.cpp	dismanifie layer headers.	a month ago
stience_layer.cu	dismantie layer headers.	a month ago
slice_layer.cpp	dismanile layer headers	a mooth ago
slice_layer.cu	dismantle layer headers	a month ago
softmax_layer.cpp	dismantie layer headers	a month ago
softmax_layer.cu	dismantle layer headers	a month ago
softmax_loss_layer.cpp	dismantie layer headers	a month ago
softmax_loss_layer.cu	dismanile layer headers	a month ago
split_layer.cpp	dismaritie layer headers	a month ago
spit_layer.cu	dismantle layer headers.	a month ago
spp_layer.cpp	dismantle layer headers	a moeth ago
tanh_layer.cpp	dismantie layer headers	a month ago
anh_layer.cu	dismantle layer headers	a month ago
threshold_layer.cpp	dismantie layer headers	a month ago
threshold_layer.cu	dismantie layer headers	a month ago
Tie_layer.cpp	dismanile layer headers	a month ago
☐ tie_byer.cu	dismantie layer headers.	a month ago
Date to be a		

window data layer.cop dismantle layer headers

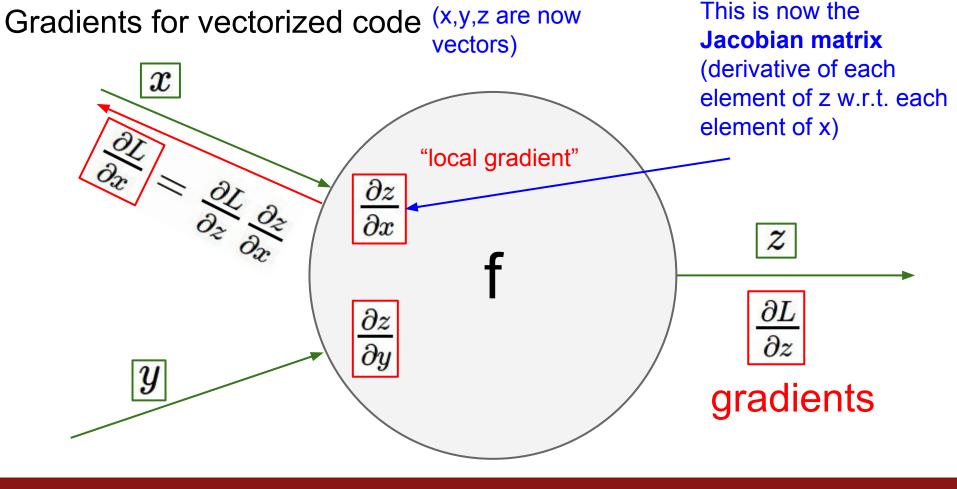
```
#include <cmath>
     #include <vector>
    #include "caffe/layers/sigmoid_layer.hpp"
    namespace caffe {
    template <typename Dtype>
    inline Dtype sigmoid(Dtype x) {
      return 1. / (1. + exp(-x));
     template <typename Dtype>
     void SigmoidLayer<Dtype>::Forward_cpu(const vector<Blob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

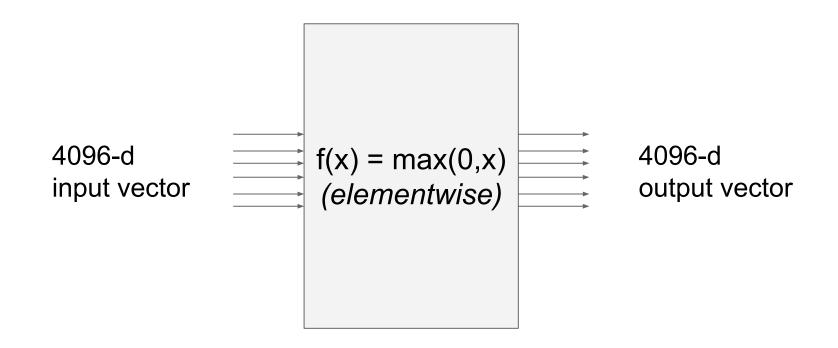
Caffe Sigmoid Layer

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

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

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

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

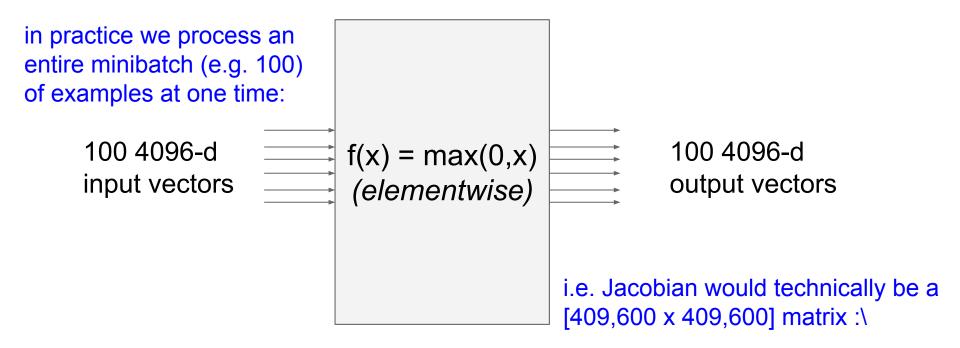
Jacobian matrix

Q2: what does it

output vector

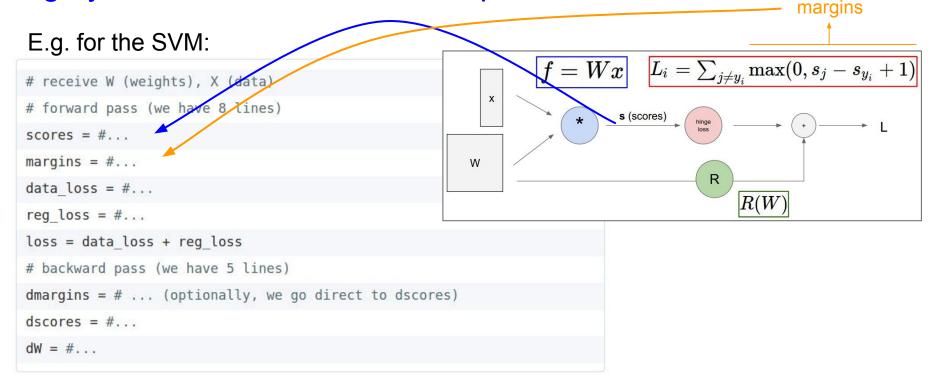
look like?

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.



Fei-Fei Li & Andrej Karpathy & Justin Johnson

Lecture 4 - 61

13 Jan 2016

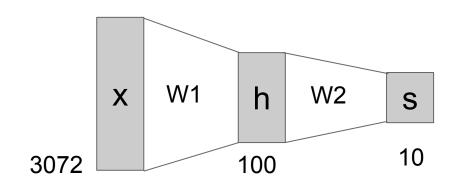
(**Before**) Linear score function: f=Wx

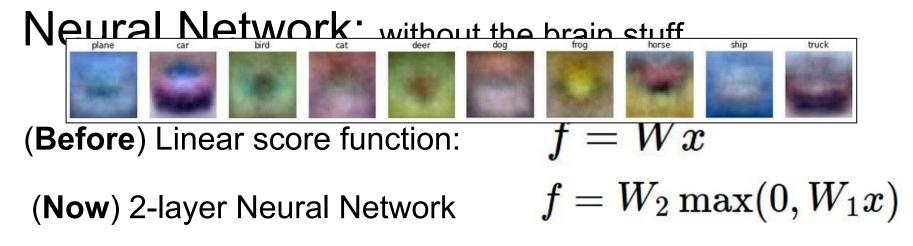
(**Before**) Linear score function: f = Wx

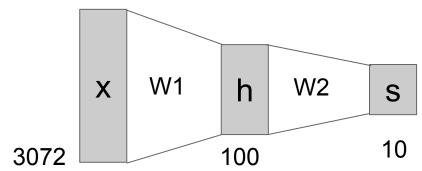
(Now) 2-layer Neural Network $f = W_2 \max(0, W_1 x)$

(**Before**) Linear score function:

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







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

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

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

$$f=W_3\max(0,W_2\max(0,W_1x))$$

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!

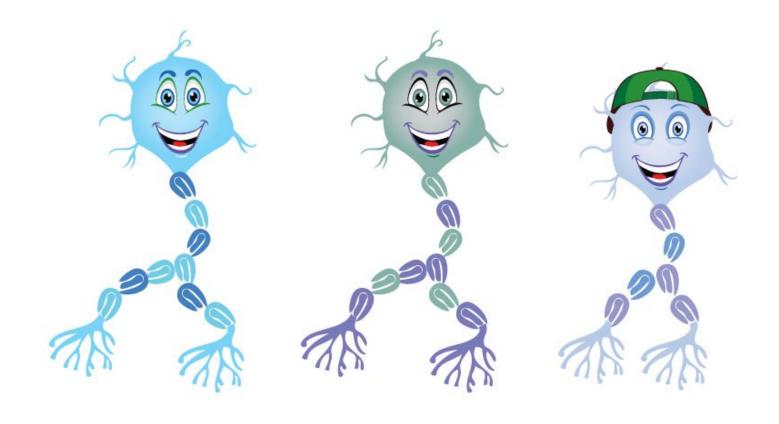
```
# receive W1,W2,b1,b2 (weights/biases), X (data)
# forward pass:
h1 = \#... function of X,W1,b1
scores = #... function of h1, W2, b2
loss = #... (several lines of code to evaluate Softmax loss)
# backward pass:
dscores = #...
dh1, dW2, db2 = #...
dW1, db1 = #...
```

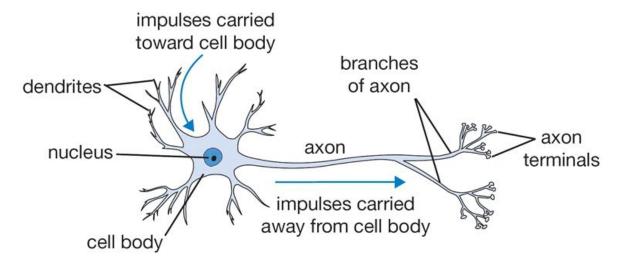


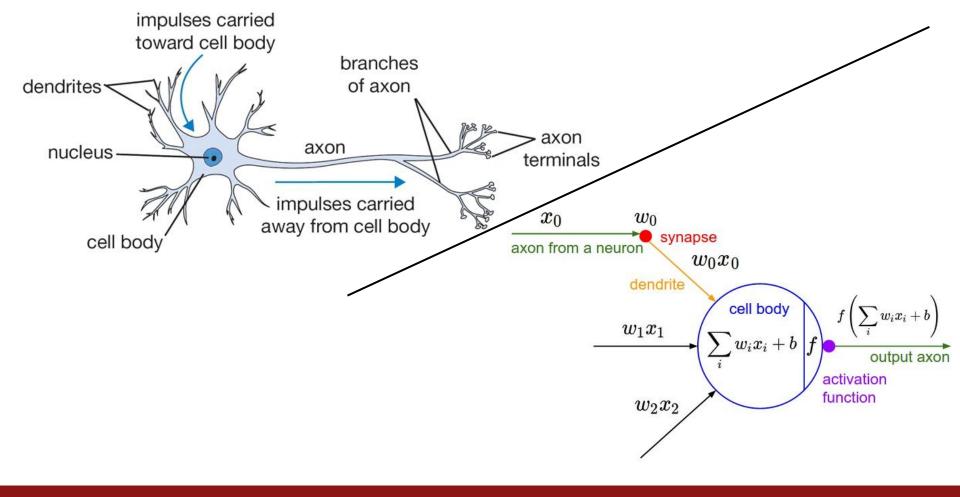
Fei-Fei Li & Andrej Karpathy & Justin Johnson

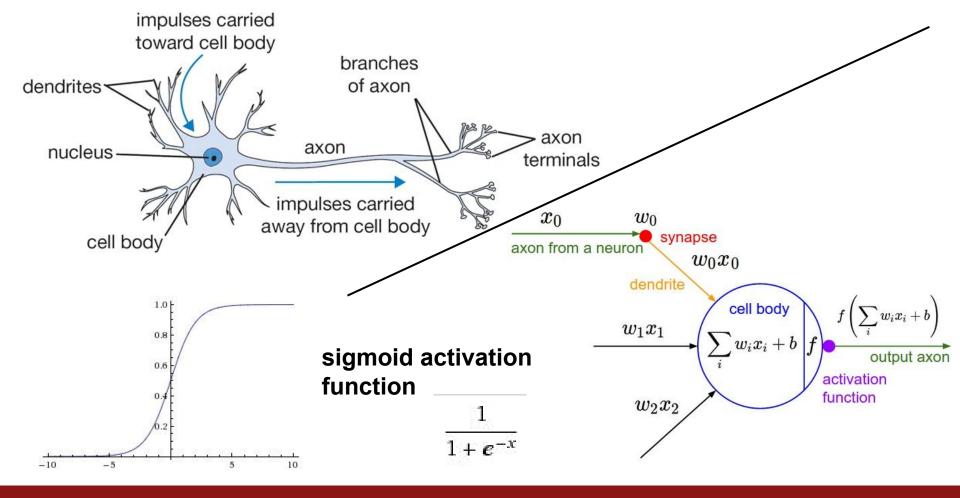
Lecture 4 - 69

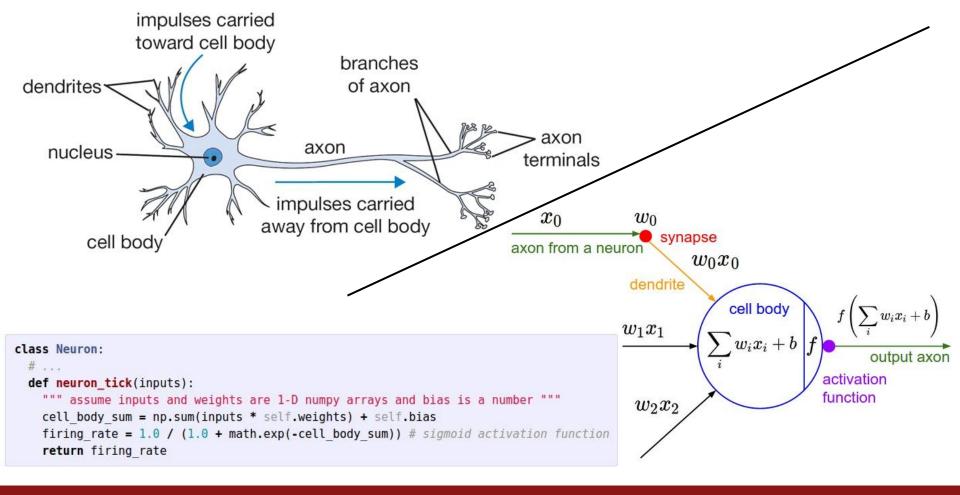
13 Jan 2016







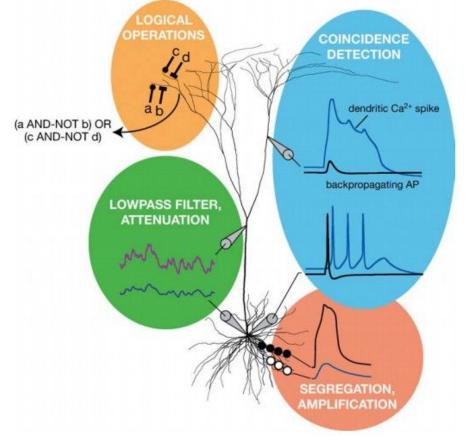




Be very careful with your Brain analogies:

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]

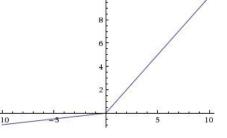
Activation Functions

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

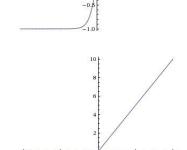
Leaky ReLU max(0.1x, x)

Sigmoid

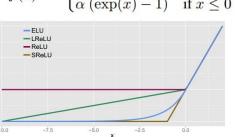
ReLU



tanh tanh(x)

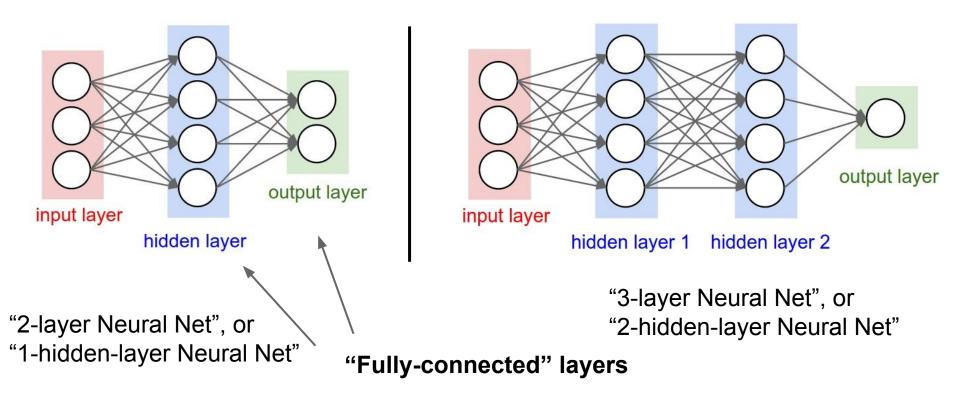


Maxout $\max(w_1^Tx+b_1,w_2^Tx+b_2)$ ELU $f(x)=\left\{egin{array}{ccc} x& ext{if }x>0 \ lpha& (\exp(x)-1)& ext{if }x\leq 0 \end{array} ight.$



max(0,x)

Neural Networks: Architectures



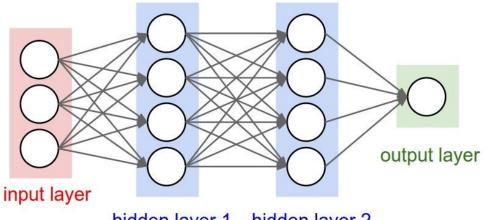
Example Feed-forward computation of a Neural Network

```
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.

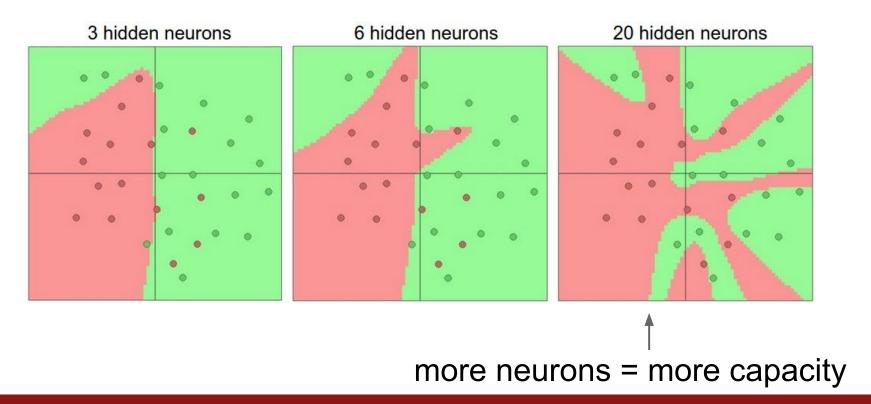
Example Feed-forward computation of a Neural Network



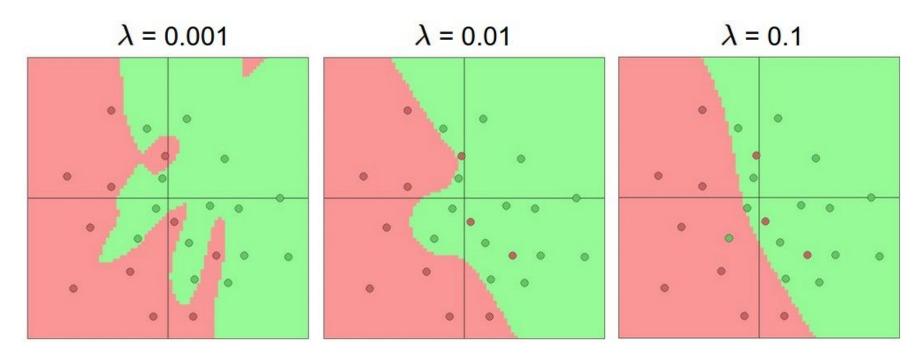
hidden layer 1 hidden layer 2

```
# 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:



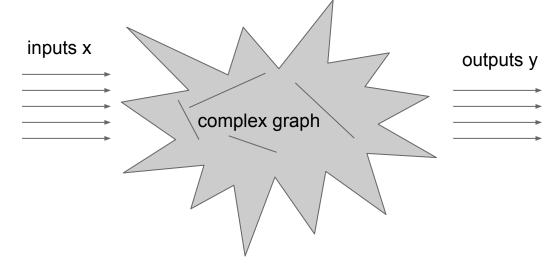
(you can play with this demo over at ConvNetJS: http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html)

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