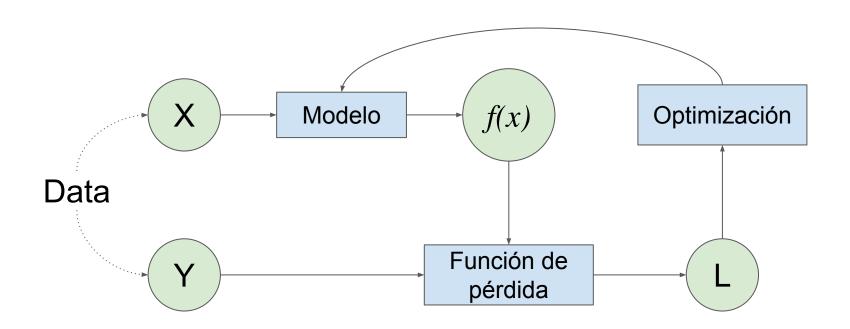
Técnicas Avanzadas de Data Mining y Sistemas Inteligentes

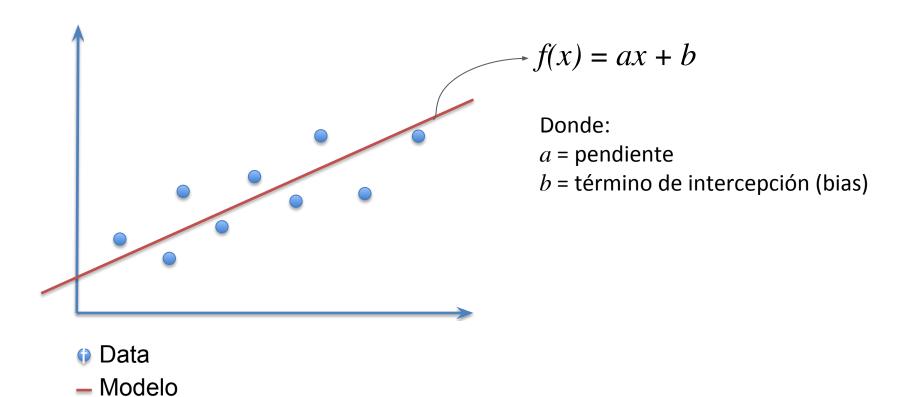
Maestría en Informática
Escuela de Posgrado
Pontificia Universidad Católica del Perú

2018-2





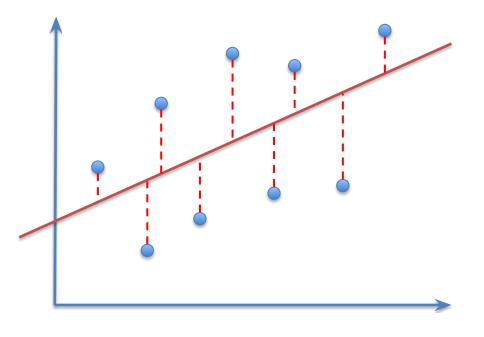
Regresión lineal simple



$$\begin{array}{ccc}
\mathbf{X} & \mathbf{W} \\
\begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 2 & 1 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 5 \\ 3 \end{bmatrix} \\
\mathbf{4x2} & \mathbf{2x1} & \mathbf{4x1}
\end{array}$$

http://matrixmultiplication.xyz

Función de pérdida: MSE (Mean squared error)



Mean squared error:

$$L=rac{1}{2m}\sum_{i=1}^m(f(x_i)-y_i)^2$$

- Data
- Modelo f(x)
- -- Error f(x) y

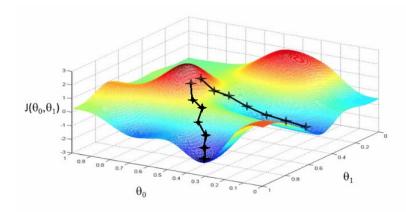
Optimización: Gradient descent

$$egin{aligned} f(x_i) &= ax_i + b \ L &= rac{1}{2m} \sum_{i=1}^m (f(x_i) - y_i)^2 \ rac{dL}{da} &= \sum_{i=1}^m (f(x_i) - y_i)(x_i) \ rac{dL}{db} &= \sum_{i=1}^m (f(x_i) - y_i) \end{aligned}$$

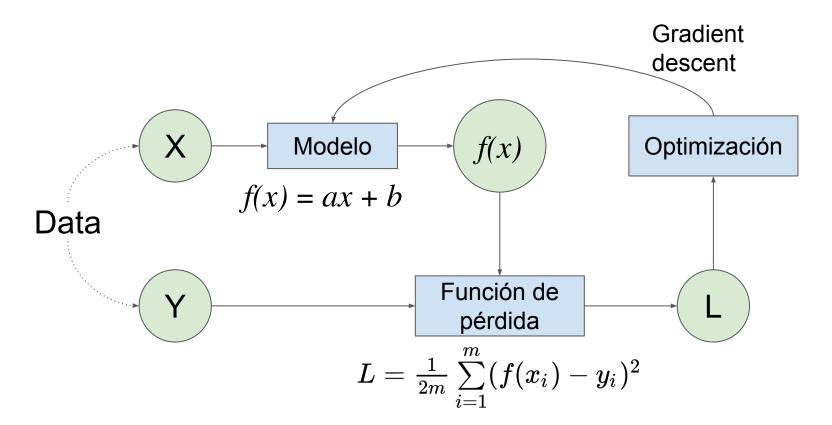
iterar:

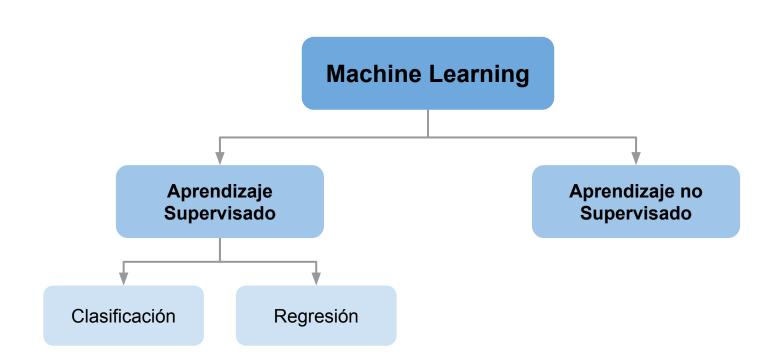
$$new \, a = a - lr * (rac{dL}{da})$$

$$new\,b = b - lr*(rac{dL}{db})$$

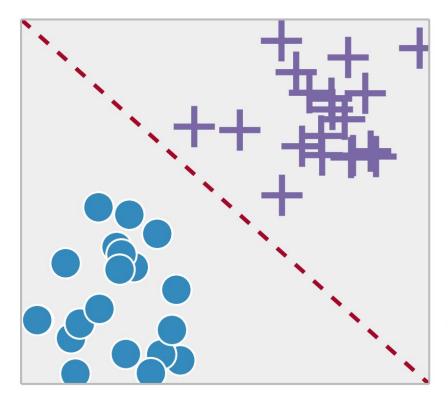


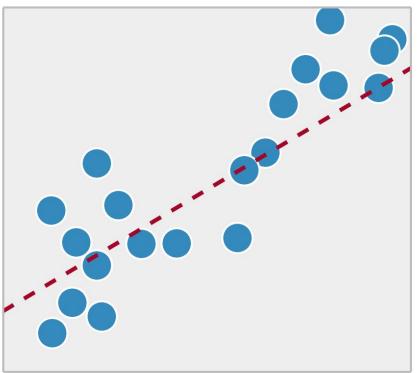
Regresión lineal simple





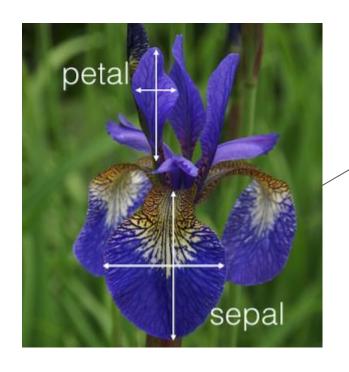
Clasificación vs Regresión





Regresión Logística

Iris dataset



sepa	l_length	sepal_width	petal_length	petal_width
	4.6	3.6	1.0	0.2
	5.7	4.4	1.5	0.4
	6.7	3.1	4.4	1.4
	4.8	3.4	1.6	0.2
	4.4	3.2	1.3	0.2

Target: class of iris plant

Iris dataset



Iris setosa



Iris versicolor



Iris virginica

Objetivo

sepal_length	sepal_width	petal_length	petal_width	???
4.6	3.6	1.0	0.2	

setosa! versicolor virginica

sepal_length	sepal_width	petal_length	petal_width	
4.6	3.6	1.0	0.2	

Variables independientes: [4x1]

??? setosa!

3 resultados, indicando la probabilidad de cada clase:

[1, 0, 0]Setosa Versicolor Virginica (one hot encoding)

Definición del modelo

data weights

 sepal_length
 sepal_width
 petal_length
 petal_width

 4.6
 3.6
 1.0
 0.2

f(x, W)

setosa!

Variables independientes: [4x1]

3 resultados, indicando la probabilidad de cada clase:

[1, 0, 0]
Setosa Versicolor Virginica
(one hot encoding)

[3x1]

data weights

sepal_lengthsepal_widthpetal_lengthpetal_width4.63.61.00.2

 $\frac{f(\mathbf{x}, \mathbf{W})}{f(\mathbf{x}, \mathbf{W}) - \mathbf{W}}$

f(x,W)=Wx 3 resultados,

Variables independientes:

[4x1]

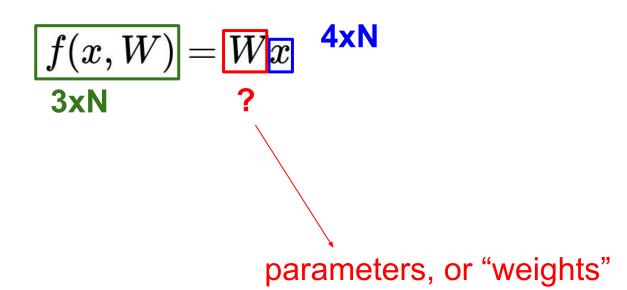
setosa!

3 resultados, indicando la probabilidad de cada clase:

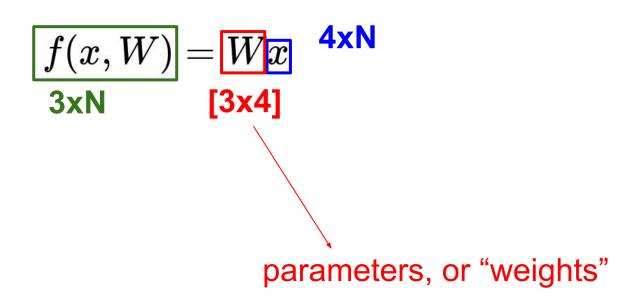
[1, 0, 0]
Setosa Versicolor Virginica
(one hot encoding)

[3x1]

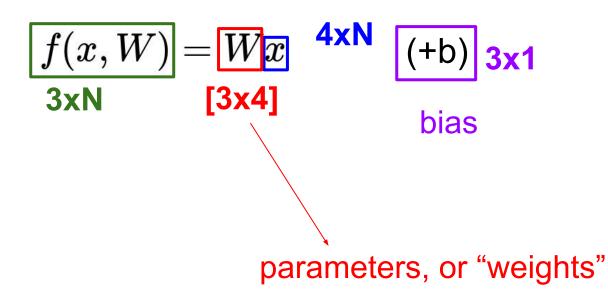
sepal_length	sepal_width	petal_length	petal_width	setosa
4.6	3.6	1.0	0.2	———



ength se	epal_width	petal_length	petal_width	setosa
4.6	3.6	1.0	0.2	

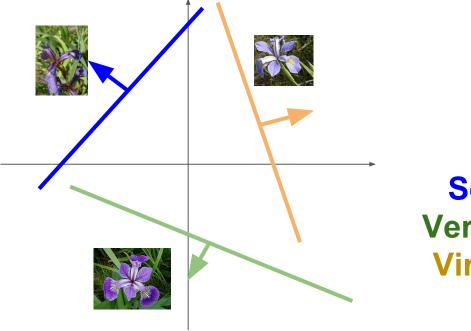


	set	seto	setos
	-	-	



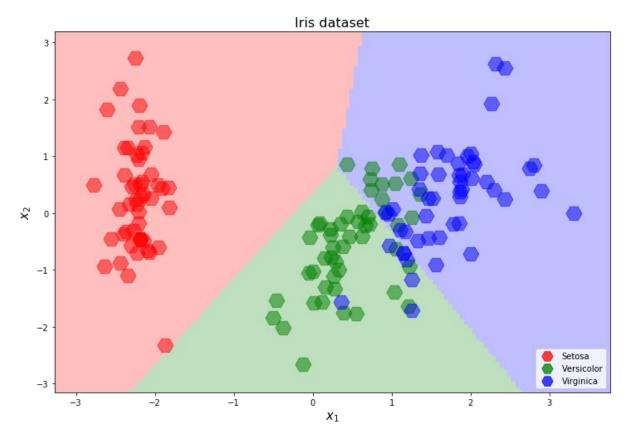
Interpretando el clasificador lineal

$$f(x_i) = Wx_i + b$$

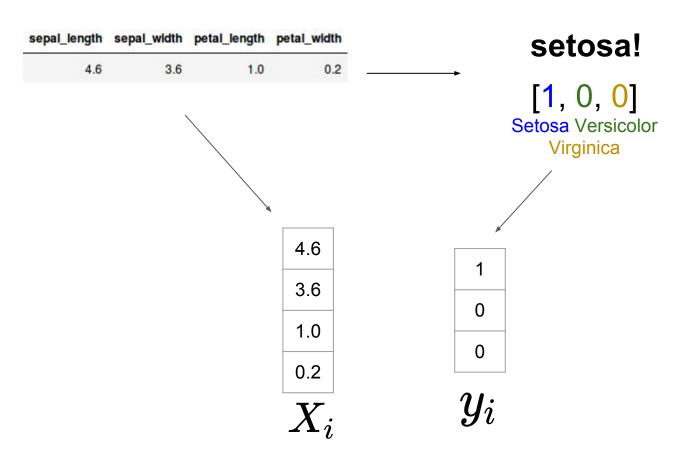


Setosa Versicolor Virginica

Interpretando el clasificador lineal



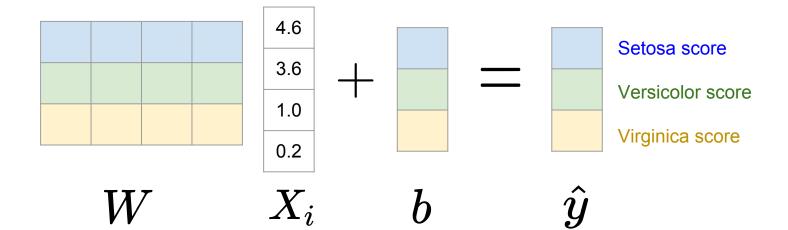
Ej:



Ej:

sepal_length	sepal_width	petal_length	petal_width	
4.6	3.6	1.0	0.2	

$$\hat{y} = f(x_i) = Wx_i + b$$



Ej:

sepal_length	sepal_width	petal_length	petal_width	
4.6	3.6	1.0	0.2	

Random numbers!

$$\hat{y} = f(x_i) = Wx_i + b$$

-0.4	-0.1	-2.1	1.6
-1.8	-0.8	0.5	-1.2
-1.1	-0.9	0.6	2.3

4.6			ı		
		1.8		-2.18	Setosa score
3.6	+	0.4		-10.5	Versicolor score
1.0		0.1		-7.14	Virginica score
0.2					3

W

 X_{i}

b

 \hat{y}

¿Cómo interpretamos estos scores?

-2.18

Setosa score

-10.5

Versicolor score

-7.14

Virginica score



-2.18	Se

Setosa score

-10.5 Versicolor score

-7.14

Virginica score

$$softmax(\hat{y_i}) = rac{e^{y_i}}{\sum\limits_{j} e^{\hat{y_j}}}$$

$$softmax(\hat{y_i}) = rac{e^{y_i}}{\sum\limits_{j} e^{\hat{y_j}}}$$

Setosa score

Versicolor score

Virginica score

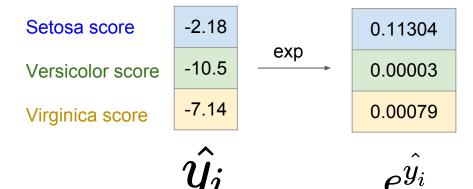
-2.18

-10.5

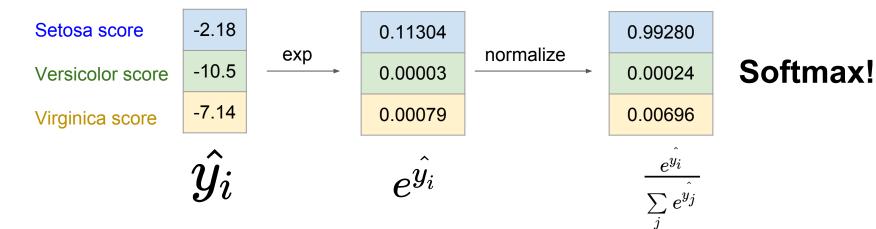
-7.14

 $\hat{y_i}$

$$softmax(\hat{y_i}) = rac{\hat{e^{y_i}}}{\sum\limits_{j} e^{\hat{y_j}}}$$

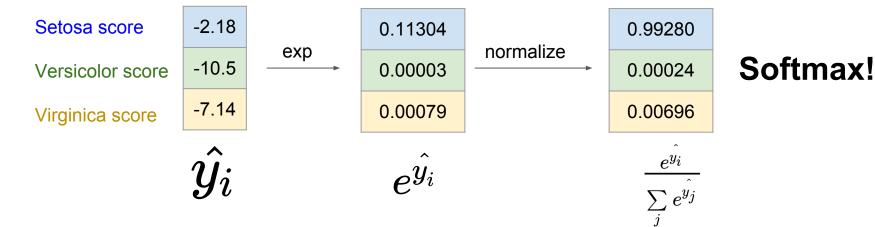


$$softmax(\hat{y_i}) = rac{e^{\hat{y_i}}}{\sum\limits_{i} e^{\hat{y_j}}}$$

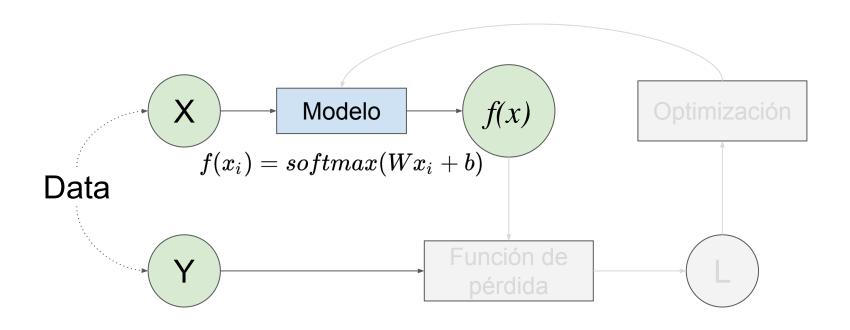


Definición final del modelo

$$f(x_i) = softmax(Wx_i + b)$$



Regresión logística



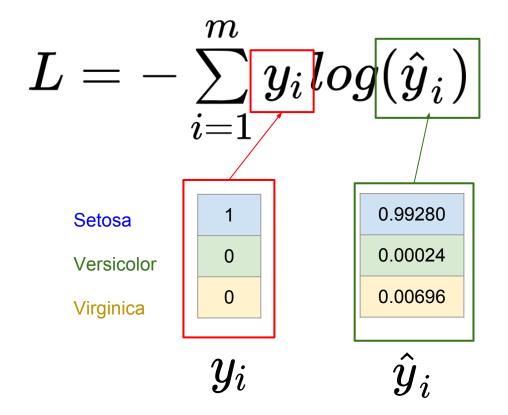
¿Cómo evaluamos el modelo?

Cross Entropy (Función de pérdida)

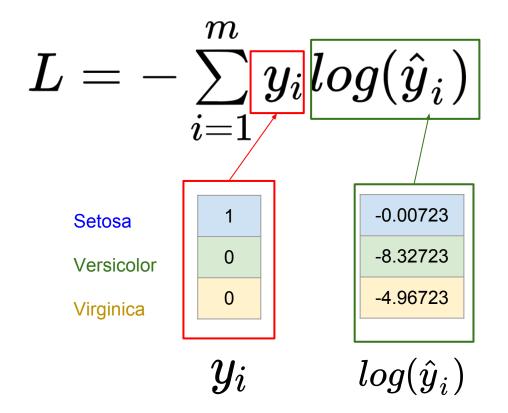
$$L = -\sum p(x) \log q(x)$$

$$L = -\sum\limits_{i=1}^m y_i log(\hat{y}_i)$$

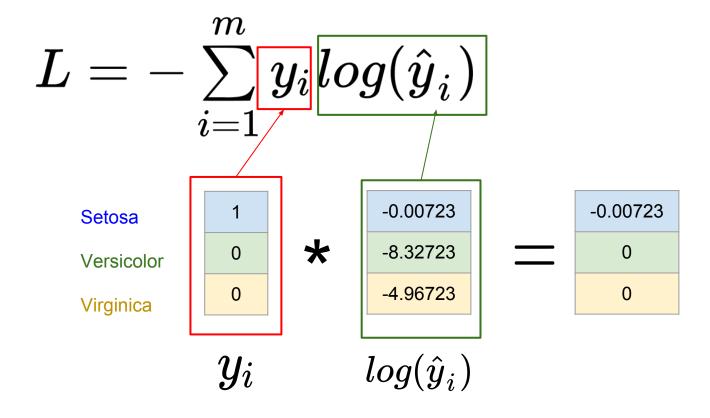
Cross Entropy (Función de pérdida)



Cross Entropy (Función de pérdida)



Cross Entropy (Función de pérdida)



Cross Entropy (Función de pérdida)

$$L = -\sum_{i=1}^m y_i log(\hat{y}_i)$$

$$L = -\sum_{0}^{\frac{-0.00723}{0}} = 0.00723$$

Review: Entropy, Cross-entropy

Ej: Queremos usar una antena para enviar el estado del clima.



Ej: Queremos usar una antena para enviar el estado del clima.



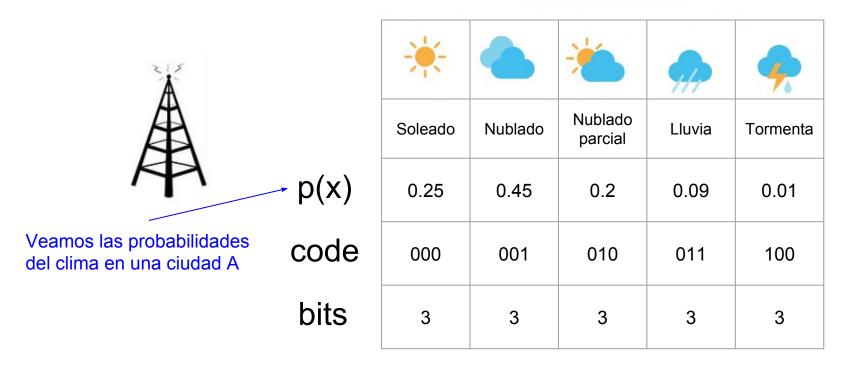
Soleado Nublado Nublado parcial Lluvia Tormenta
000 001 010 011 100
3 3 3 3 3 3

bits

code

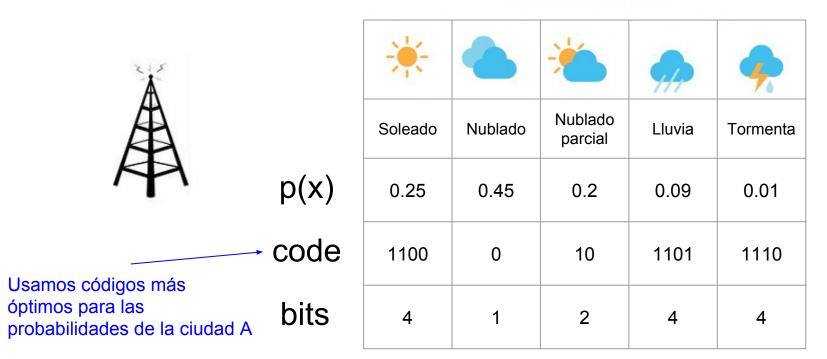
cantidad promedio de bits usados = 3

Ej: Queremos usar una antena para enviar el estado del clima.



bits promedio = 3(0.25) + 3(0.45) + 3(0.2) + 3(0.09) + 3(0.01) = 3

Ej: Queremos usar una antena para enviar el estado del clima.



bits promedio = 4(0.25) + 1(0.45) + 2(0.2) + 4(0.09) + 4(0.01) = 2.25

Ej: Queremos usar una antena para enviar el estado del clima.

*	4	*	1	•
Soleado	Nublado	Nublado parcial	Lluvia	Tormenta
0.25	0.45	0.2	0.09	0.01

p(x)

Ej: Queremos usar una antena para enviar el estado del clima.

	*	4	*	14	•
	Soleado	Nublado	Nublado parcial	Lluvia	Tormenta
p(x)	0.25	0.45	0.2	0.09	0.01
1/p(x)	4	2.22	5	11.11	100

Ej: Queremos usar una antena para enviar el estado del clima.

	*	4	*	1	•
	Soleado	Nublado	Nublado parcial	Lluvia	Tormenta
p(x)	0.25	0.45	0.2	0.09	0.01
1/p(x)	4	2.22	5	11.11	100
$-\log_2(p)$	2	1.15	2.32	3.47	6.64

$$\log_2(\frac{1}{p}) = -\log_2(p)$$

Ej: Queremos usar una antena para enviar el estado del clima.

	Soleado	Nublado	Nublado parcial	Lluvia	Tormenta	¿Cómo obtenemos mínimo número de bits que se puede usar?
p(x)	0.25	0.45	0.2	0.09	0.01	- - -
$-\log_2(p)$	2	1.15	2.32	3.47	6.64	$\log_2(rac{1}{p}) = -\log_2(p)$

bits promedio = 0.25(2) + 0.45(1.15) + 0.2(2.32) + 0.09(3.47) + 0.01(6.64) = 1.86

Ej: Queremos usar una antena para enviar el estado del clima.

	*	•	*	1	•
	Soleado	Nublado	Nublado parcial	Lluvia	Tormenta
p(x)	0.25	0.45	0.2	0.09	0.01
$-\log_2(p)$	2	1.15	2.32	3.47	6.64

Entropy
$$-\sum p(x)\log p(x)$$

bits promedio = 0.25(2) + 0.45(1.15) + 0.2(2.32) + 0.09(3.47) + 0.01(6.64) = 1.86

Ej: Queremos usar una antena para enviar el estado del clima.

	*	•	*	14	•
	Soleado	Nublado	Nublado parcial	Lluvia	Tormenta
p(x)	0.25	0.45	0.2	0.09	0.01
$-\log_2(p)$	2	1.15	2.32	3.47	6.64

Entropy
$$-\sum p(x)\log p(x)$$

La entropía en este caso es 1.86, esta cantidad de bits representa la información que uno obtendría en promedio, al conocer el clima en la ciudad A.

	*	4	*	1	•
	Soleado	Nublado	Nublado parcial	Lluvia	Tormenta
p(x) en A	0.25	0.45	0.2	0.09	0.01
$-\log_2(p_A)$	2	1.15	2.32	3.47	6.64
p(x) en B	0.3	0.2	0.2	0.25	0.05

Si la antena usa los bits óptimos para la ciudad A.

¿Cuál sería la cantidad de bits promedio que enviaría la antena en la ciudad B?

- / 1	*	4	*	1	•
	Soleado	Nublado	Nublado parcial	Lluvia	Tormenta
p(x) en A	0.25	0.45	0.2	0.09	0.01
$-\log_2(p_A)$	2	1.15	2.32	3.47	6.64
p(x) en B	0.3	0.2	0.2	0.25	0.05

Si la antena usa los bits óptimos para la ciudad A.

¿Cuál sería la cantidad de bits promedio que enviaría la antena en la ciudad B?

bits promedio = 0.3(2) + 0.2(1.15) + 0.2(2.32) + 0.25(3.47) + 0.05(6.64) = 2.5

*	4	*	A	•
Soleado	Nublado	Nublado parcial	Lluvia	Tormenta
0.25	0.45	0.2	0.09	0.01
2	1.15	2.32	3.47	6.64
0.3	0.2	0.2	0.25	0.05

p(x) en A

 $-\log_2(p_A)$

p(x) en B

Si la antena usa los bits óptimos para la ciudad A.

¿Cuál sería la cantidad de bits promedio que enviaría la antena en la ciudad B?

Cross-entropy

$$-\sum p(x)\log q(x)$$

bits promedio = 0.3(2) + 0.2(1.15) + 0.2(2.32) + 0.25(3.47) + 0.05(6.64) = 2.5

	*	4	*		•	Si la antena usa los bits óptimos para la ciudad A.
	Soleado	Nublado	Nublado parcial	Lluvia	Tormenta	¿Cuál sería la cantidad de bits promedio que enviaría
p(x) en A	0.25	0.45	0.2	0.09	0.01	la antena en la ciudad B?
$-\log_2(p_A)$	2	1.15	2.32	3.47	6.64	Cross-entropy
p(x) en B	0.3	0.2	0.2	0.25	0.05	$-\sum p(x)\log q(x)$

La entropía cruzada en este caso es 2.5, esta cantidad de bits representa la información que uno obtendría en promedio al conocer el clima en la ciudad B usando la codificación óptima para la ciudad A.

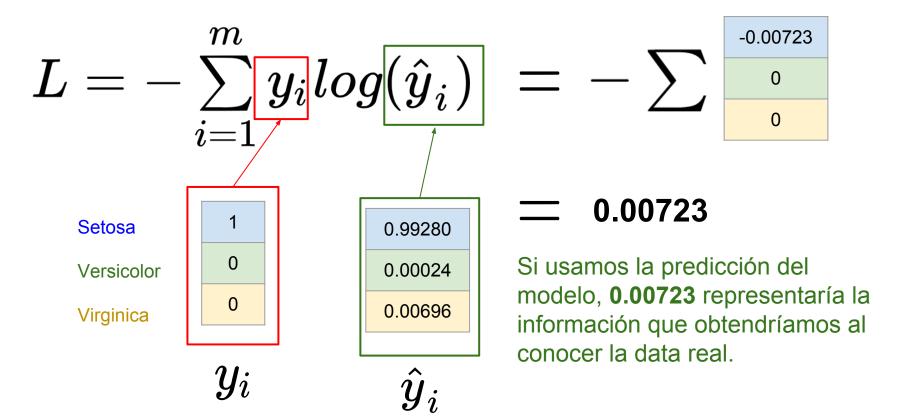
Entropy

$$-\sum p(x)\log p(x)$$

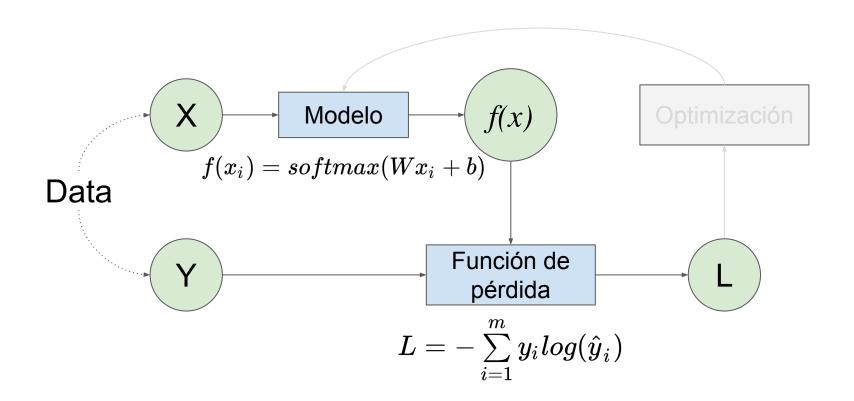
Cross-entropy

$$-\sum p(x)\log q(x)$$

Cross Entropy (Función de pérdida)



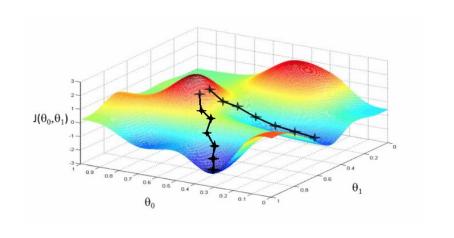
Regresión logística

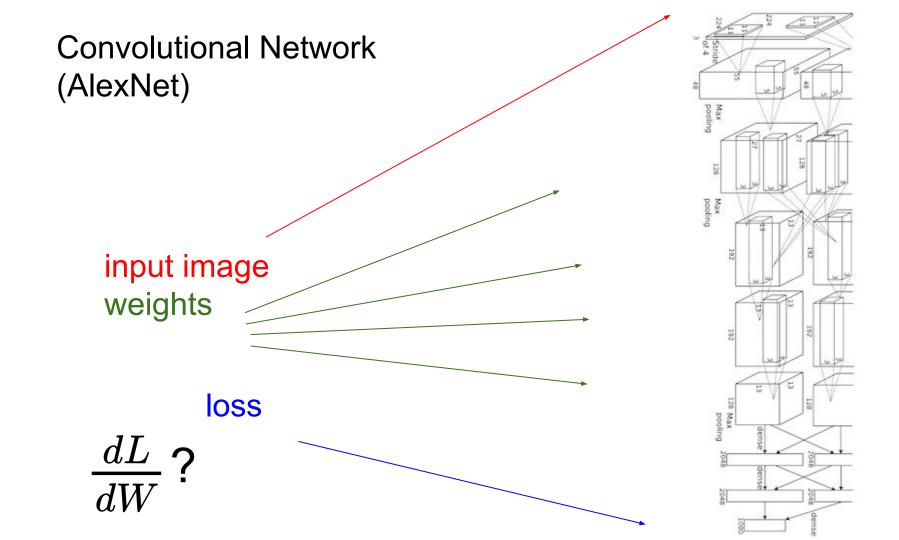


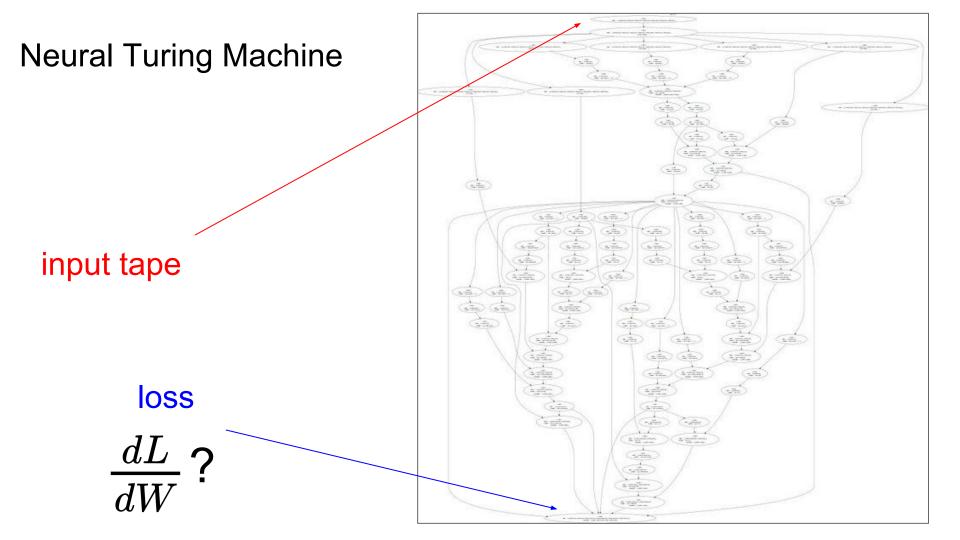
Optimización

Optimización

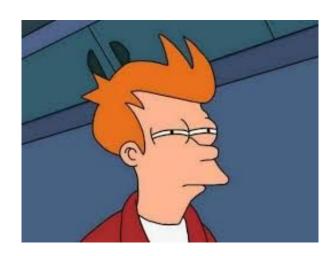
$$egin{aligned} f(x_i) &= softmax(Wx_i + b) \ L &= -\sum\limits_{i=1}^m y_i log(\hat{y}_i) \ &rac{dL}{dW}, rac{dL}{db} oldsymbol{?} \end{aligned}$$





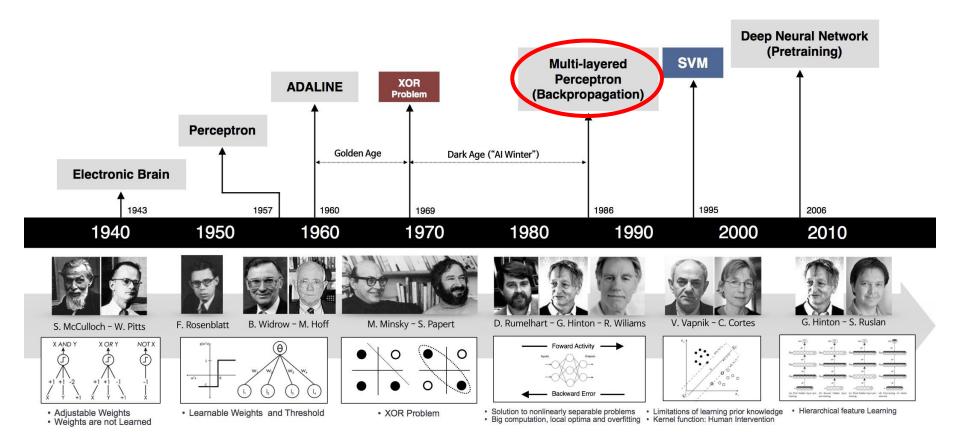


¿Cómo derivamos las gradientes?



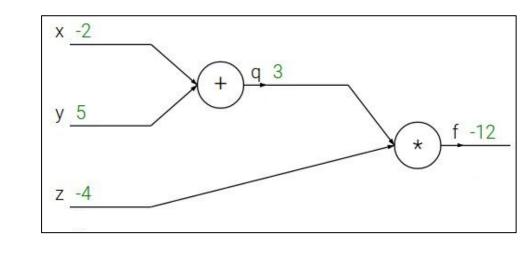
Backpropagation!





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

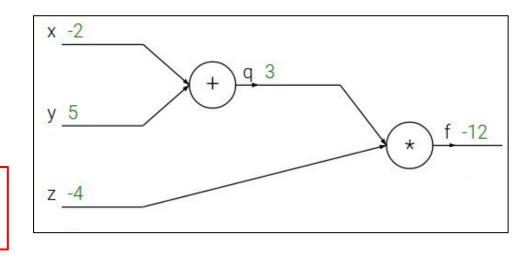
e.g. x = -2, y = 5, z = -4



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

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

$$\partial f \quad \partial f \quad \partial f$$

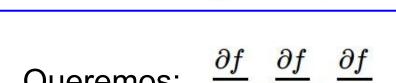


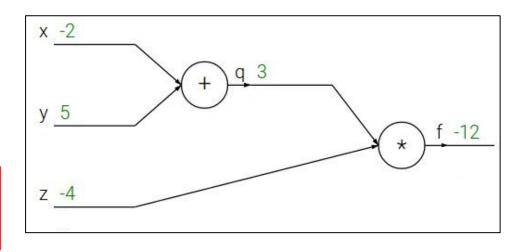
Queremos:
$$\frac{\partial f}{\partial x}$$
, $\frac{\partial f}{\partial y}$, $\frac{\partial f}{\partial z}$

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

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

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



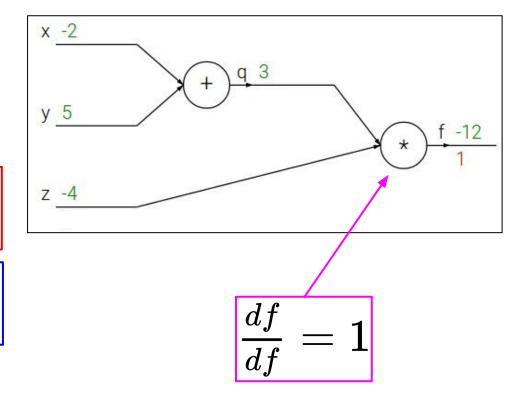


$$f(x, y, z) = (x + y)z$$

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

q = x + y

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

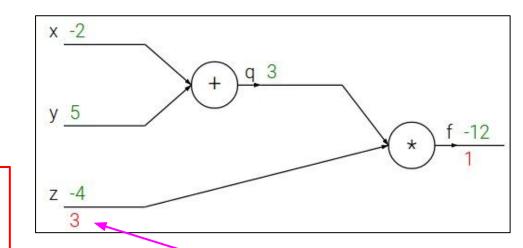


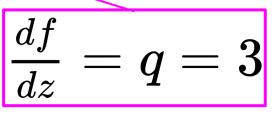
$$f(x, y, z) = (x + y)z$$

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

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

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



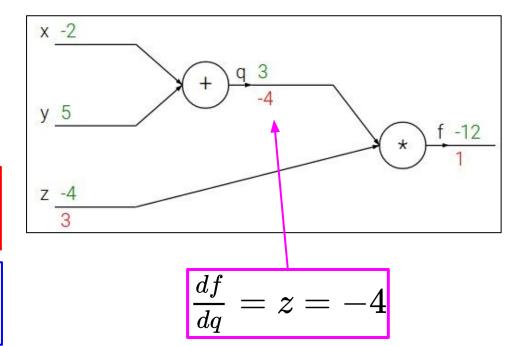


$$f(x, y, z) = (x + y)z$$

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

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

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



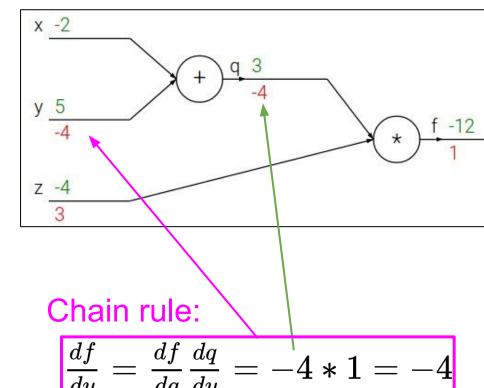
$$f(x, y, z) = (x + y)z$$

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

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

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

Queremos:



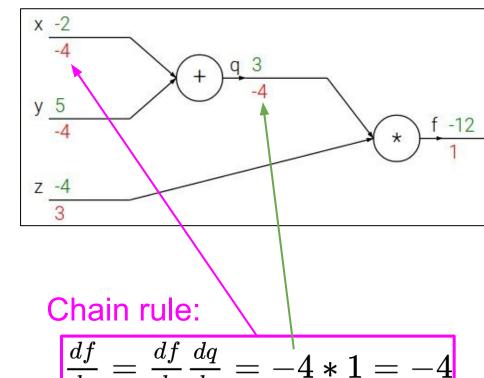
$$f(x, y, z) = (x + y)z$$

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

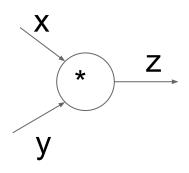
$$q=x+y$$
 $\frac{\partial q}{\partial x}=1, \frac{\partial q}{\partial y}=1$

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

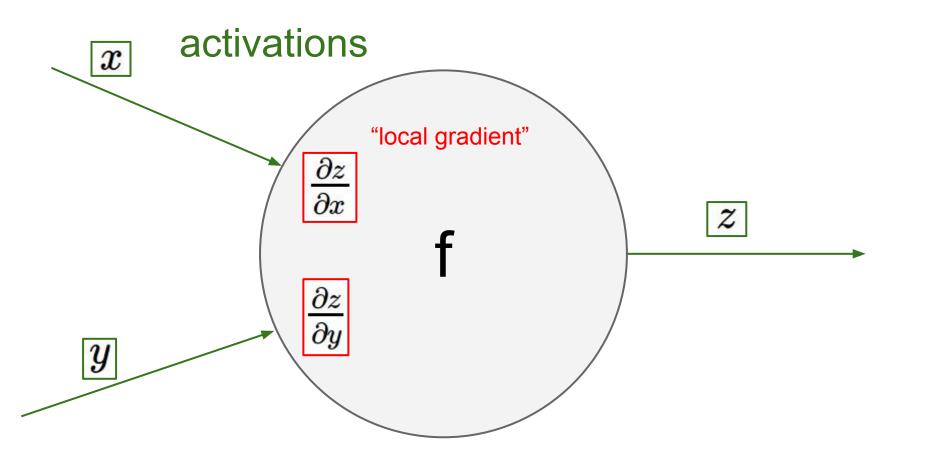
Queremos:



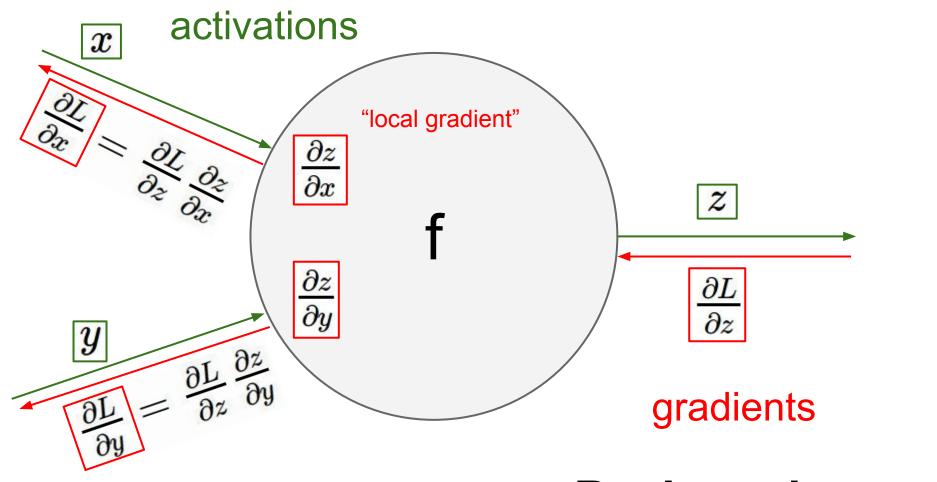
Implementation: forward/backward API



```
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]
```



Forward pass



Backward pass

```
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
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()

TensorFlow Layers

⊕ 1,039 commits	₽7trancher			O relea	1565	⊕ 86 cor	tributors
Branch: master - New pull r	nequest	New Sie	Find file	HTTPS.	herma://atth	ab.com/torch 💲	Download Z
	IS65 from torch/revert-563-master					Latest commit 23d	
						Laws commi 220	
doc	Fix batch mode in MarginRanking						4 days as
a generic	Improve error message in Spatia		100				a day ay
in 16	THNN: add missing OpenMP inc	ude					2 days a
in socks	Add 'luaffi' dependency						14 days a
gitgrore	tell git to ignore build output						4 months a
B Juacheckro	[Torch] Move test lue to the top is	rvel					a year a
El Atovia york	small fixes for test path						2 months as
El Abs.lus	Add THWN conversion of (ELU, L						7 days a
AbsCritarion.lua	Add THNN conversion of (ELU, I	eakyReLU.	LogSigmoid, L	ogSoftMax, I	.ooku		7 days a
Add.lua	fix Add with multi-dim bias						10 months of
AddConstant.lus	Adding in-place AddConstant an						9 months as
BCECriterion lua	Remove unnecessary mallocs for	en BOECrite	rion.				3 months a
BatchNormalization.lua	fix batchnorm reset						3 months a
☐ CAddTableJus	fixing table modules to return cor						6 months of
E) CDivTable.loa	fixing table modules to return con	ect number	of gradinputs				6 months a
CMakeLists.txt.	Add C implementation of SpatialBatchNormalization					7 days a	
⊕ CWuttus	on Module preserve type sharing semantics (#187); add nn Module apply					4 months as	
CVL/Table.lus	fixing table modules to return cor	ect number	of gradinputs				6 months as
CONTRIBUTING and	added developing tips						3 months as
COPYRIGHT.bt	add copyright file						2 years o
CSubTableJua	fixing table modules to return cor	ect number	of gradinputs				6 months as
R Clamp.kus	Use custom range in HardTanh and mask it as Clamp					3 months ay	
ClassNLLCriterion.lua	Add functional conversion of ClassNLLCriterion				13 days as		
Concetius	fix a bug in conditional expressio						a month as
Fi ConcetTable lus	foring bug in ConcetTable variable	e length					4 months as
P. Container kus	Adding:applyToModules() to no.	Container w	bich is like us	tud ("vior			3 months of
Pi Copy kan	nn.Module preserve type sharing				i i		4 months as
Fi Cosine Jua	Fix type() in Cosine						a month as
R) CosineDistance lua	Do not change state variables in	Cosine Dista	non/CosineFo	hedding Crite	rion		2 months as
CosineEmbeddingCriterion I	Do not change state variables in						2 months or
F) Criterion kas	on Module preserve type sharing						4 months as
CriterionTableJua	Rename unpack to table unpack						8 months of
CrossEntropyCriterion Jus	Check for 'nn Module' and 'nn C		and the Town				8 months as
Pi DephConcat lus	adding direct backward to Cong						9 months as
Pi DistRLDivCriterion lua	Use tensor for THNN functions e						10 days a
DisProduction	Add batch mode in DotProduct +		e element out	UG			2 months as
El DosProductiva Pi Dropoutiva	Add betch mode in DotProduct + in-classe dropout	und 1860					2 months as
El Dispositiva	in-place dropout Add THNN conversion of (ELU, L)	and Callin					
E) Ecotos E) Errorklessages kus	Add THNN conversion of (ELU, L Give heter error messages when						7 days as
El ErrorMessages.lua El Euclidean.lua		-1-0					a year a 4 months a
El Euclidean Jus El Femilios	nn Module preserve type sharing Fen marks has note	semantics (mory; add no	Assoure appl	E		
							9 months a
FlattenTable.lup	nn Module preserve type sharing	semantics (#197); add nn	Module appl	ř		4 months a
GradientReversal.lus	Add GradientReversal layer						4 months a
E HardStvink.lua	Add functional convention of Har						10 days a
HardTanhJua	Add functional conversion of Har						10 days a
HingeEmbeddingCriterion Jus	rewrote HingeEmbeddingCriterio						6 months as
El Identity.kus	Revert to previous Identity.lus im						2 months a
€ Index.lue	Simplifying and more efficient no						2 months a
☐ Jacobian lua	Add unit tests for hession (us, fix						6 months a
JoinTable lua	nn Module preserve type sharing				y .		4 months as
E) L1Costkia	Use tensor for THNN functions e	ven for single	element out;	uts			10 days a
L1HingeEmbeddingCriterio	Make type() truly recursive.						9 months as
a consideration and a second							
E) L1Pensity.lus	fixed L1Penalty constructor argu-	rients					a year a

LogSigmoid lug	Add THNN conversion of (ELU, LeakyReLU, LogSigmoid, LogSoftMax, Looku	7 days a
E LogSofMax.lus	Add THNN convenion of (ELU, LeekyReLU, LogSigmoid, LogSoftMax, Looku	7 days a
€ LookupTable Jua	Harmonize LookupTable signature with ourn impl	5 days a
MM.lus	Rename unpack to table unpack for Lus 5.2	8 months a
IR MSECriterion las	Add SizeAverage to criterions in the constructor	2 months a
MarginCriterion kus	modernized MarginCriterion	a year a
Fi MarginRankingCriterion kus	Fix batch mode in MarsinRankinoCriterion	4 days a
E Marko	Merge pull request #464 from vgiro/master	2 months a
E Man in	Add support for negative dimension and both batch and non-batch input	2 months a
R Mindus	Merge pull request #464 from vgre/master	2 months a
MinureTable lus	cancel unused variable and useless expression	29 days a
P Module lus	Revert "Don't re-flatten parameters if they are already flattened"	15 hours
E Mollon	renoving the requirement for providing size in nn.Mul	0.0000
E MulConstant.lus	ignore updateGradinput if self-gradinput is nil	a year o
E Multi-Colorisor Los		
	asserts in MultiCriterion and ParadelCriterion add	2 months
MuttLabeMarginCriterion.lus	Initial reviewp of torch? tree	4 years
MultiMarginCriterion.lua	multimargin supports p=2	11 months
Narrow/ua	typeAs in Narrow not done in place.	6 months
NarrowTable.lua	NarrowTable	6 months
Normalize Jua	Remove brom and beddbrom from Normaliza, because they allocate memory,	20 days
PReLUtus	Buffers for PReLU cuda implementation:	8 months
Padding.lus	fixed broken nn.Padding: input was returned in backgrop.	5 months
PairwiseDistance.lue	Merge pull request #532 from xwgeng/master	29 days
Parallel lua	fix a bug in conditional expression	a month
ParatletCriterion.toa	asserts in MultiCriterion and PassfulCriterion add	2 months
Paralle/Table.lua	Parallel cotimization. ParallelTable inherits Container, unit tests	a year
P Power kan	Use UNIX (ine eridings	7 months
R README red	dos readfledosa	5 months
RReLUke	Add randomized leaky rectified linear unit (RReLU)	3 months
ReLUlus	adds in-clace ReLU and foxes a potential divide-by-pero in nn. Sort	9 months
	Registro batchillode	
Replicate Jua		8 months
Reshape.lus	Added more informative pretty-printing.	a year
Selectiva	initial revamp of torsh? tree	4 years
SelectTable.lua	nn.Module preserve type sharing semantics (#187); add nn.Module apply	4 months
Sequential.lua	fixing Sequential remove corner case	6 months
Sigmoid.lua	initial reverse of torch7 tree	4 years
Smoothl,1Criterion.kas	Add SizeAverage to criterions in the constructor	2 months
SoftMax lus	Fix various unused variables in nn	a year
Sottlin kus	Fix various unused variables in no	a year
Soffice.tue	fixed a numerical issue in the SoffPlus module (it breaks for input g	2 years
SoftShrink lua	initial reverse of torch? tree	4 years
SotSign.lus	initial reversip of torch? tree	4 years
SparseJacobian kus	Fix various unused variables in m	8 9997
P. Sparsel inear tus	Using sparse implementation of zeroGradParameters for Sparsel invar	a month
SpatialAdaptiveMasPooling	Added SostalAdaptiveMaxPooling	a year
Spetial Average Pooling. lug	Spatial Average Pooling supports padding, cell mode and exclude and div	29 days
SpatialBatchNomelization lue	Add C implementation of SpatialBatchNormalization	7 days
Spatialistic (Northead and America) Spatial Contraction Normalize	Add C implementation of agranuationvormatization Make troo() truly recursive.	9 months
SpetalConvolution.lus	Fix type() in SpatialConvolution	3 months
SpatialConvolutionMM.lua	Fix type() in SpatialConvolution	3 months
SpatialConvolutionMap.lua	Remove unused and expensive initialization logic from nn Spatial Convo	8 months
SpatialCrossMapLRN Aus	cude consistency	18 days
Spatia/DivisiveNormalizatio	Spatial[Constructive,Divisive,Subtractive]Normalization work with bat	8 months
	small fix on error message	6 months
	Adding Fractional Max Pooling	
SpatialFractionalMaxPoolin		3 months
SpatialFractionalMaxPoolis SpatialFullConvolution.lus	Adding Fractional Max Pooling	3 months 5 days
SpatialFractionalMaxPools SpatialFullConvolution.lua SpatialFullConvolutionMap.lua	Adding Practional Max Pooling Add adjustment term to Spalia/FullConvolution to control the size of	3 months 5 days 3 years
Spatial Fractional MaxPools Spatial Full Convolution Jus Spatial Full Convolution Map Jus Spatial EPPooling Jus Spatial EPPooling Jus	Adding Practional Max Pooling Add adjustment form to Spatia Full Convolution to control the size of New NN classes	3 months 5 days 3 years 10 months
Spatial Fractional MaxPoolin Spatial Full Convolution Jua Spatial Full Convolution Map Jua Spatial II. Proofing Jua Spatial MaxPooling Jua	Adding Fractional Max Pooling Add algorithment form to SpatialFullConvolution to control the size of New BNY disease SpatialAnnagePooling divides by XXYYH	3 months 5 days 3 years 10 months 6 months
Spatial Fractional MacPoolin. Spatial Full Convolution Jua Spatial Full Convolution Map Jua Spatial Full Convolution Map Jua Spatial I PPooling Jua Spatial MacPooling Jua Spatial MacPooling Jua Spatial MacPopoling Jua	Adding Fractional Max Proding Add adjustment from to SpalieFullConvolution to control the size of New NV classes SpatialAverageFulling divides by WYHH SpatialAverageFulling divides by WYHH SpatialAverageFulling control	3 months 5 days 3 years 10 months 6 months 26 days
Spatial Fractional MacPoolin. Spatial Full Convolution Jua Spatial Full Convolution Map Jua Spatial II Phooling Jua Spatial MacPooling Jua Spatial MacPooling Jua Spatial MacPooling Jua Spatial MacConvolution Jua Spatial MacConvolution Jua Spatial MacLua Spatial MacLua	Adding Manacoust Man Pushing Add aplacement was founder/full-convolution to control the size of New NAY closures Special Annexing Pushing and Annexing Special Annexing An	3 months 5 days. 3 years 10 months 6 months 26 days. 4 months
Spatial Fractional Matheolin Spatial Full Convolution Jua Spatial Full Convolution Map Jua Spatial L Phooling Jua Spatial Max Pooling Jua Spatial Match Unpooling Jua Spatial Match Unpooling Jua Spatial Match Unpooling Jua Spatial Spat	Addag framework Mar Pulling Addag dispersion the Marginal ACConnection to control the sight of Nex 1911 classes Special Acceptance of the Section of the sight of Special Acceptance pulling sight of the Section of the Sectio	3 months 5 days 3 years 10 months 6 months 20 days 4 months 3 years
Spatial Fractional Mast Poolin Spatial Full Convolution Jua Spatial Full Convolution Jua Spatial Full Convolution Mappha Spatial Full Convolution Mappha Spatial Mast Pooling Jua Spatial Mast Pooling Jua Spatial Mast Jua Spatial Mast Jua Spatial Sub Serripling Jua Spatial Sub Serripling Jua Spatial Sub Servipling Jua Spatial Sub Servipling Jua Spatial Sub Servipling Jua	Adulty fractional file in finite of the control that in finite of the control that in the significant control of t	3 months 5 days. 3 years 50 months. 6 months. 26 days. 4 months. 3 years. 8 months.
Spatial Fractional MacProcis. Spatial Full Connectation Jul Spatial Full Connectation Library Spatial Full Connectation Library Spatial Full Connectation Library Spatial MacProcing Jul Spatial MacProcing Jul Spatial Conference Spatial Confe	Adalign Securities Mar Parling Adal agramment with in Studies Address of the size of New 300 classes Securities of the size of the size of the size of New 300 classes Securities of the size of the	3 months 5 days 3 years 50 months 6 months 26 days 4 months 8 months 7 months
Spatial Propositius Spatial Propositius Spatial Processitius Peole Spatial Spatial Processitius Spatial Spati	Adulty fractional file in finite of the control that in finite of the control that in the significant control of t	3 months 5 days 3 years 10 months 6 months 20 days 4 months 3 years



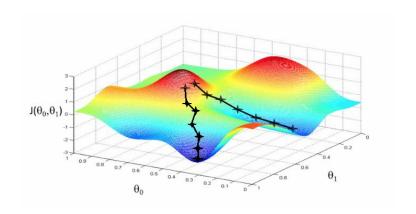




Optimización: Gradient descent

Iterar:

- 1. **Forward**: obtener la pérdida (Loss).
- 2. Backprop: calcular las gradientes.
- 3. Actualizar los parámetros del modelo.

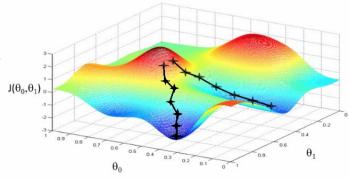


Optimización: Gradient descent

Iterar:

- 1. Forward: obtener la pérdida (Loss).
- 2. Backprop: calcular las gradientes.
- 3. Actualizar los parámetros del modelo.

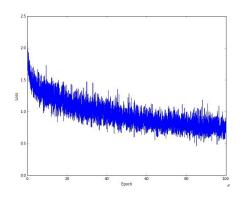
¿Y si tenemos mucha data?

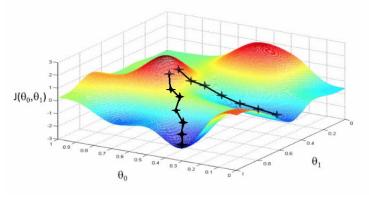


Optimización: Mini-batch Gradient descent

Iterar:

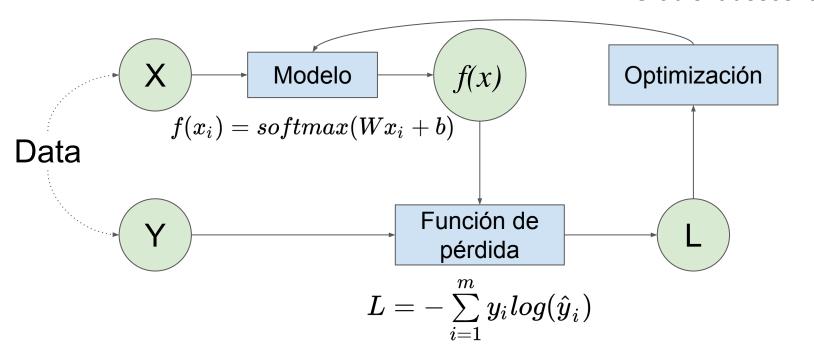
- 1. Sample: obtener una muestra de la data.
- 2. Forward: obtener la pérdida (Loss).
- 3. Backprop: calcular las gradientes.
- 4. Actualizar los parámetros del modelo.





Regresión logística

Backprop +
Mini-batch
Gradient descent



Hasta ahora teníamos funciones lineales antes de aplicar softmax:

$$f(x_i) = softmax(Wx_i + b)$$

¿Cómo incrementamos la complejidad del modelo?

Hasta ahora teníamos funciones lineales antes de aplicar softmax:

$$f(x_i) = softmax(Wx_i + b)$$

¿Cómo incrementamos la complejidad del modelo?

Podemos añadir operaciones intermedias

$$h(x_i) = W_1 x_i + b_1$$
10xN 10x4 4xN 10x1

$$h(x_i) = W_1 x_i + b_1$$
10xN 10x4 4xN 10x1
$$f(x_i) = softmax(W_2 h(x_i) + b_2)$$
3xN 3x10 10xN 3x1

$$egin{aligned} h_1(x_i) &= W_1 x_i + b_1 \ h_2(x_i) &= W_2 h_1(x_i) + b_2 \ &\vdots \ &\vdots \end{aligned}$$

$$f(x_i) = softmax(W_m h_n(x_i) + b_m)$$

$$h_1(x_i)=W_1x_i+b_1$$

$$h_2(x_i) = W_2 h_1(x_i) + b_2$$

Pero una serie de operaciones lineales se puede reducir a solo 1.

$$y_1 = 2x + 10$$

 $y_2 = 3y_1 - 5$

$$y_2 = 6x + 25$$

$$f(x_i) = softmax(W_m h_n(x_i) + b_m)$$

$$h_1(x_i) = \sigma(W_1x_i + b_1)$$

 $h_2(x_i) = \sigma(W_2h_1(x_i) + b_2)$

Pero una serie de operaciones lineales se puede reducir a solo 1.

Añadimos funciones de activación

 $f(x_i) = softmax(W_m h_n(x_i) + b_m)$

$$h_1(x_i) = \sigma(W_1x_i + b_1)$$

$$h_2(x_i) = \sigma(W_2 h_1(x_i) + b_2)$$

Pero una serie de operaciones lineales se puede reducir a solo 1.

Añadimos funciones de activación

$$f(x_i) = \underbrace{softmax}(W_m h_n(x_i) + b_m)$$

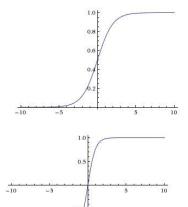
softmax también es una función de activación

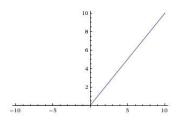
Sigmoid

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

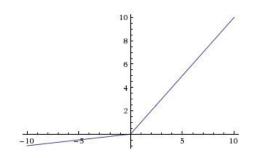
tanh tanh(x)

ReLU max(0,x)

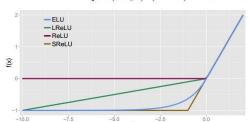




Leaky ReLU max(0.1x, x)



ELU
$$f(x) = \begin{cases} x \\ \alpha \text{ (e.)} \end{cases}$$

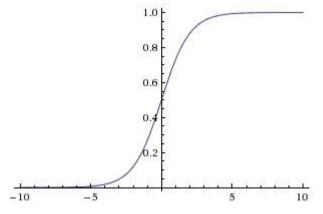


$$\sigma(x) = 1/(1 + e^{-x})$$

Sigmoid

- Reduce el rango a [0,1]
- Históricamente popular, por la interpretación de "activar" una neurona.

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

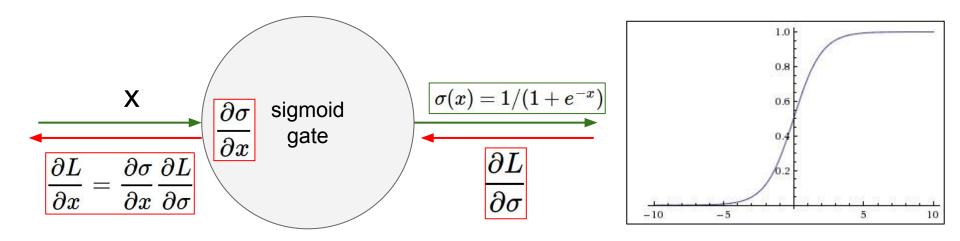


Sigmoid

- Reduce el rango a [0,1]
- Históricamente popular, por la interpretación de "activar" una neurona.

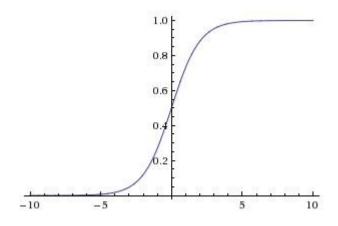
3 problemas:

 Las neuronas saturadas "matan" las gradientes.



¿Qué pasa cuando x = -10? ¿Qué pasa cuando x = 10?

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

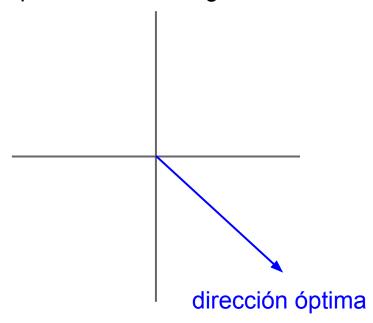


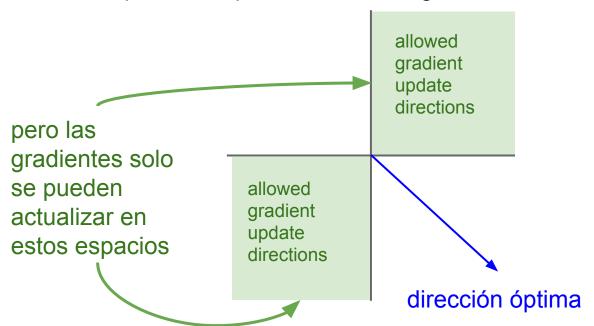
Sigmoid

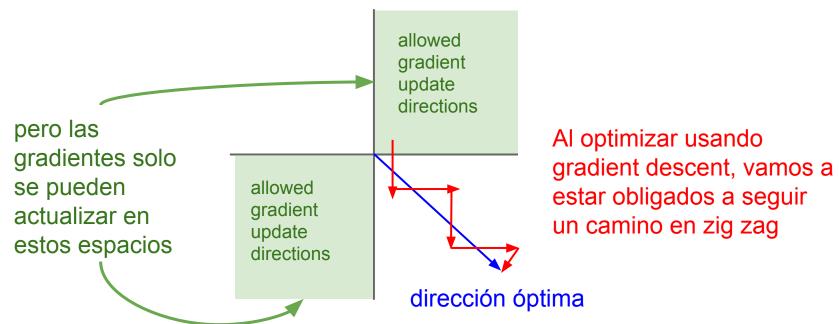
- Reduce el rango a [0,1]
- Históricamente popular, por la interpretación de "activar" una neurona.

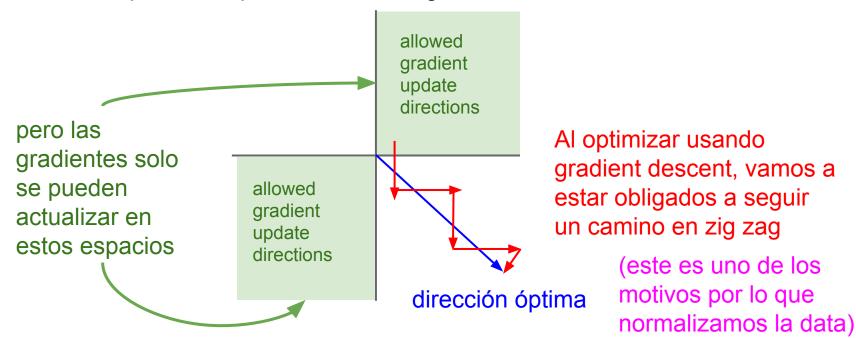
3 problemas:

- Las neuronas saturadas "matan" las gradientes.
- El outputs no está centrado en cero.

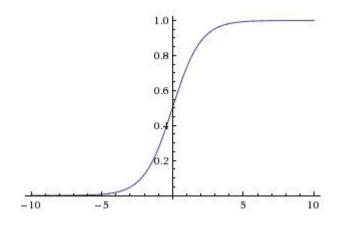








$$\sigma(x)=1/(1+e^{-x})$$

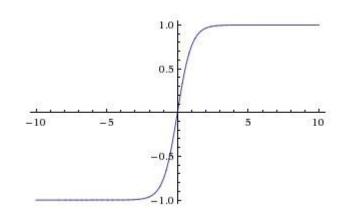


Sigmoid

- Reduce el rango a [0,1]
- Históricamente popular, por la interpretación de "activar" una neurona.

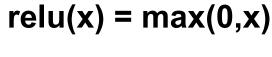
3 problemas:

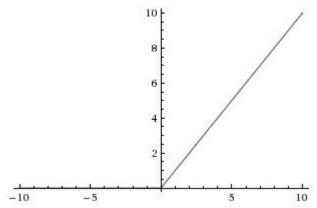
- Las neuronas saturadas "matan" las gradientes.
- 2. El outputs no está centrado en cero.
- La función exponencial es pesada de calcular.



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

- Reduce el rango a [-1,1]
- Centrada en cero :D
- Neuronas saturadas :(

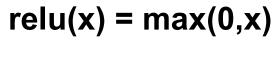


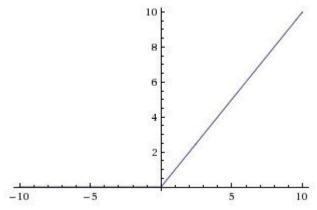


- No se satura (en los positivos).
- Computacionalmente eficiente.

- No está centrada en cero.
- Las gradientes cuando x<0 es?

ReLU (Rectified Linear Unit)

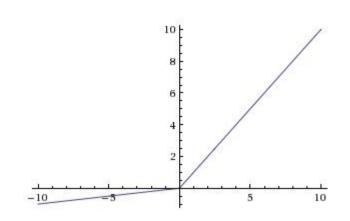




- No se satura (en los positivos).
- Computacionalmente eficiente.

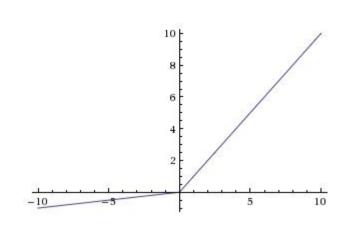
- No está centrada en cero.
- Las gradientes cuando x<0 es 0.

ReLU (Rectified Linear Unit)



Leaky ReLU

$$f(x) = \max(0.01x, x)$$



Leaky ReLU

$$f(x) = \max(0.01x, x)$$

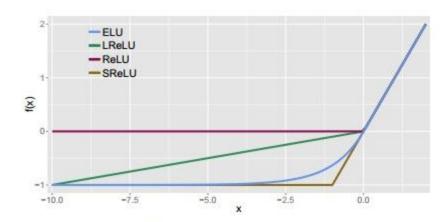
Parametric Rectifier (PReLU)

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

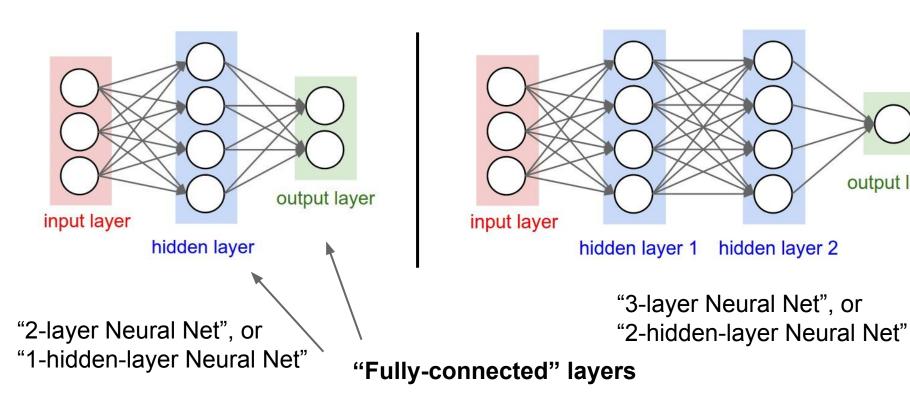
alpha es un parámetro a optimizar

[Mass et al., 2013] [He et al., 2015]

Exponential Linear Units (ELU)

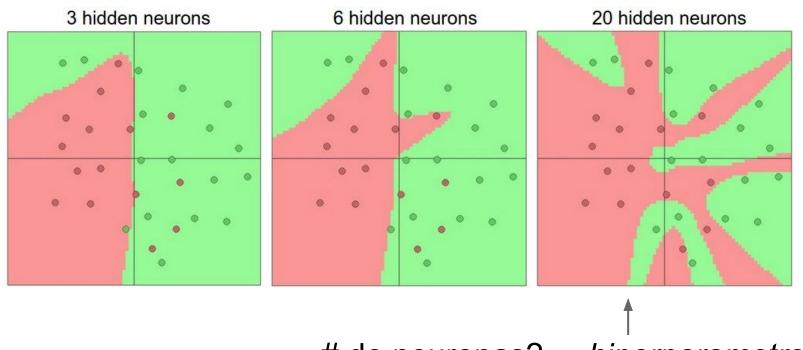


$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha (\exp(x) - 1) & \text{if } x \le 0 \end{cases}$$

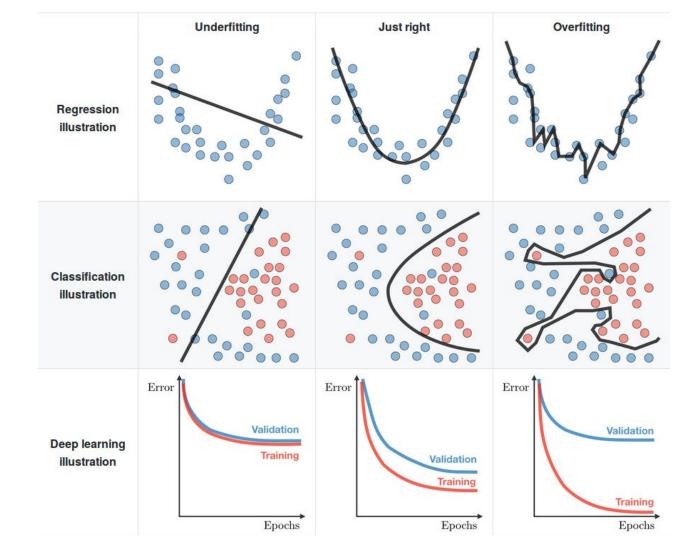


output layer

¿Número de capas, de neuronas?



de neuronas?.... hiperparametros



	Underfitting	Just right	Overfitting
Symptoms	High training errorTraining error close to test errorHigh bias	Training error slightly lower than test error	 Very low training error Training error much lower than test error High variance
Possible	Complexify model Add more features		Perform regularization

Get more data

remedies

Train longer

Backprop +
Mini-batch
Gradient descent

