Inspire…Educate…Transform.

Artificial neural networks

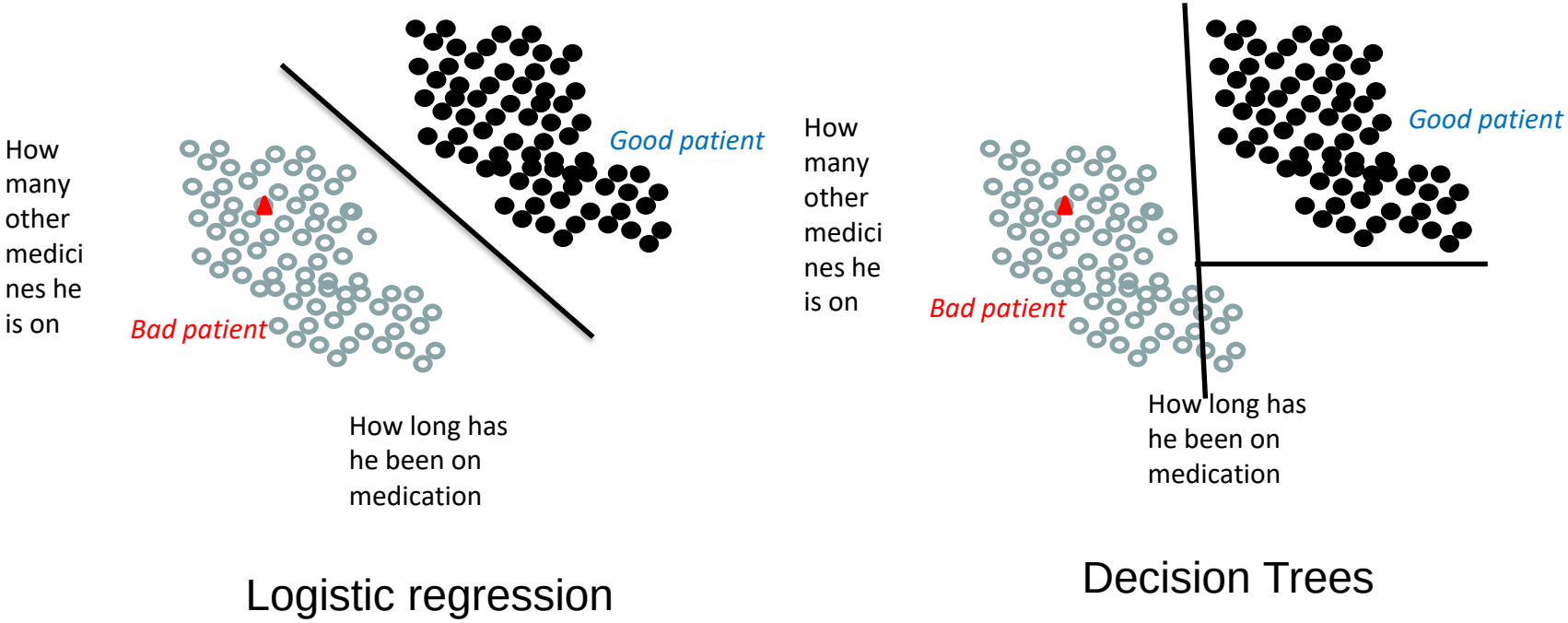
Dr. K. V. Dakshinamurthy

President, International School of Engineering

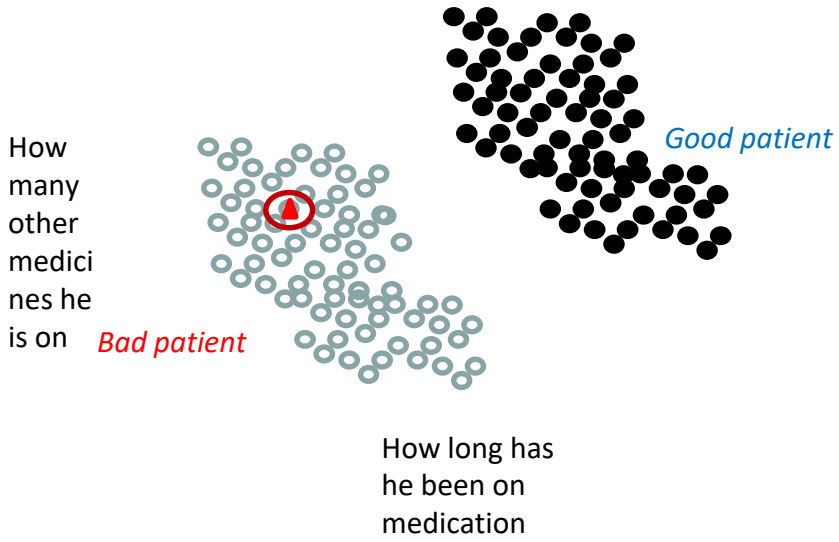
Graphical Review

ML ALGORITHMS

Will the patient take the medicines?

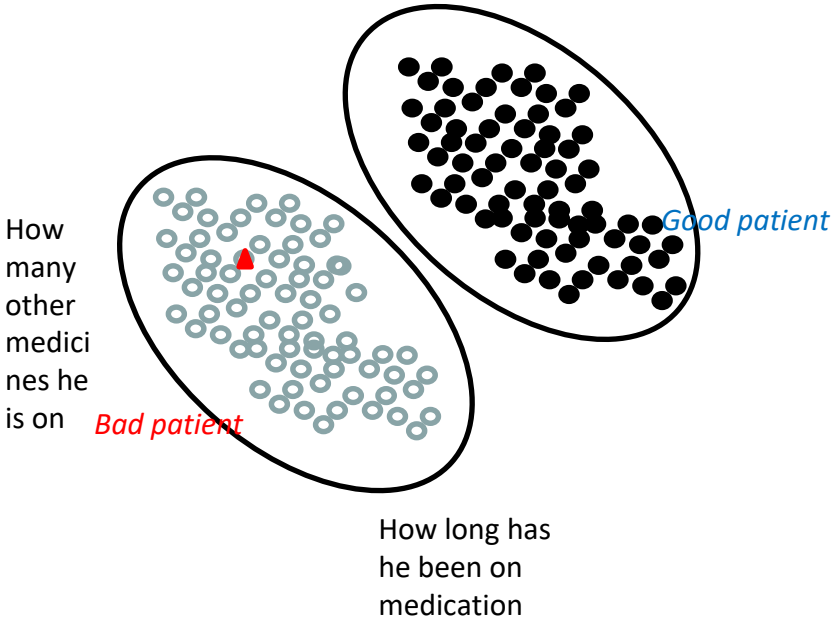


Classification: Graphical intuition
Will the patient take the medicines?



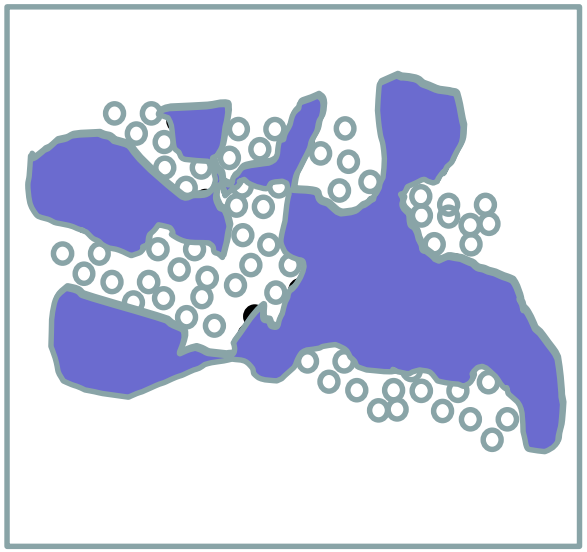
Nearest neighbors

Classification: Graphical intuition
Will the patient take the medicines?



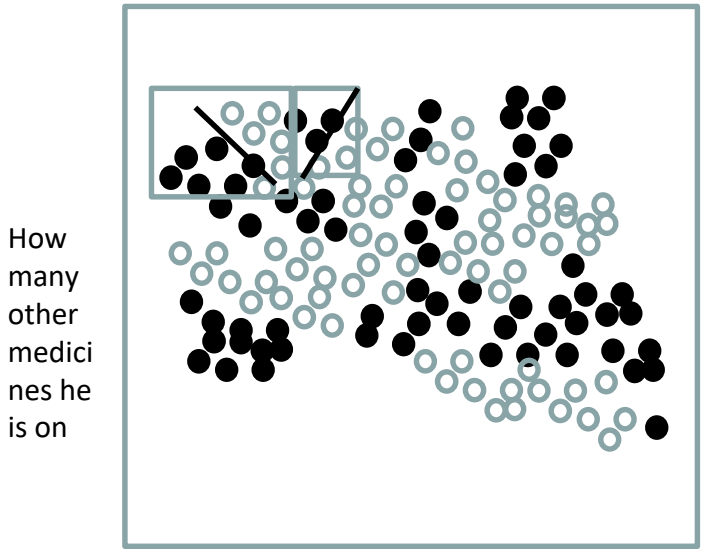
Bayesian models

Classification: On complex data



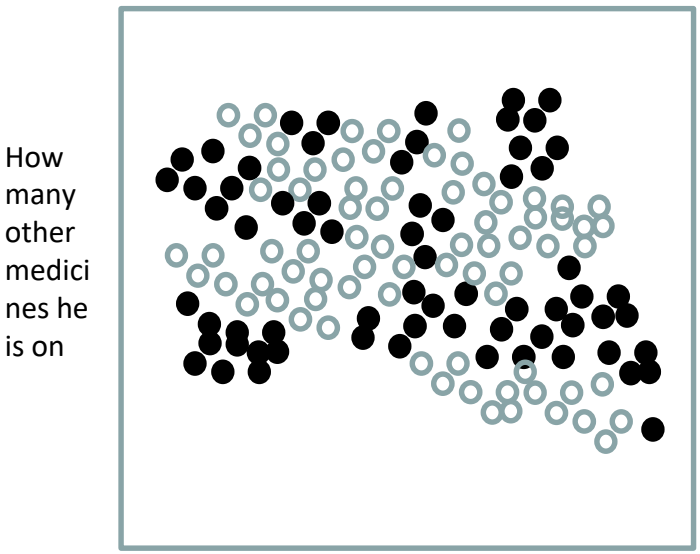
Ensembles

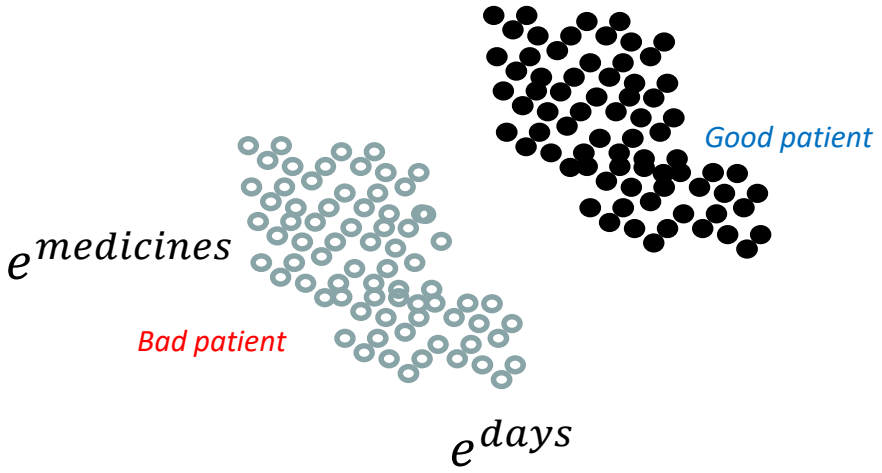
One mega model on entire data
Multiple models on small sets of data and combining the predictions
Later is mostly better



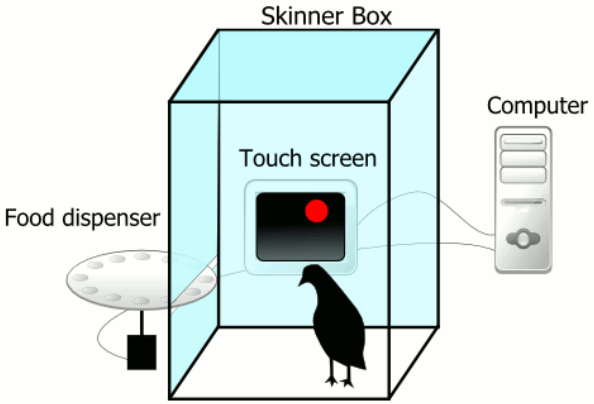
58

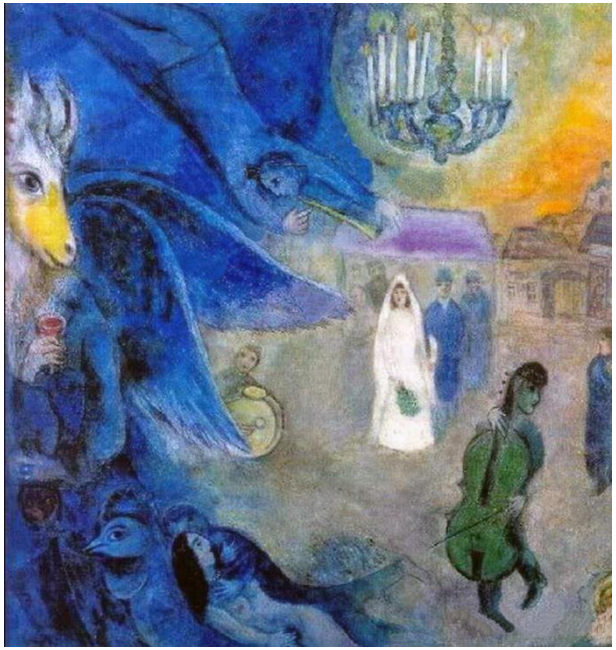
On complex data





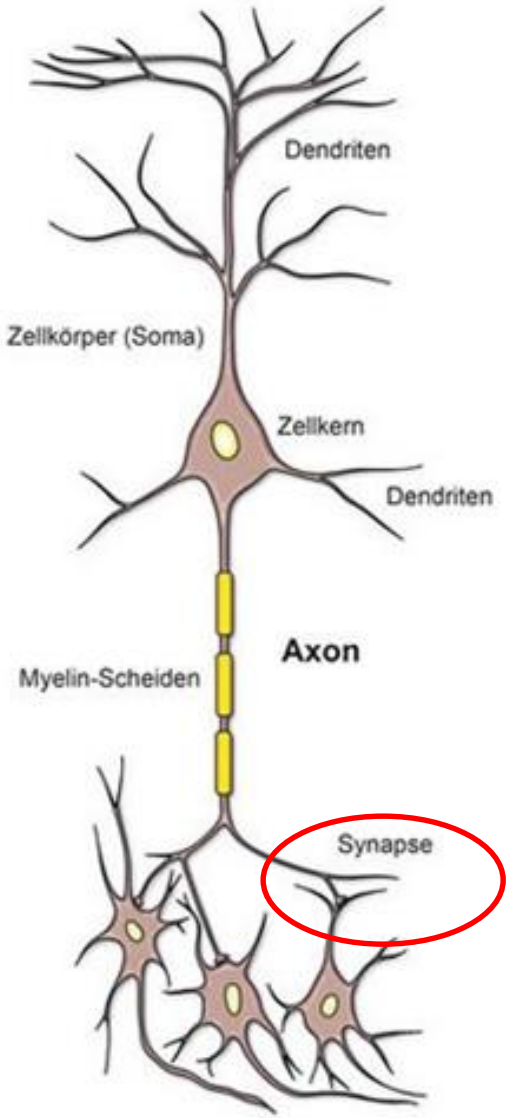
Spectral methods

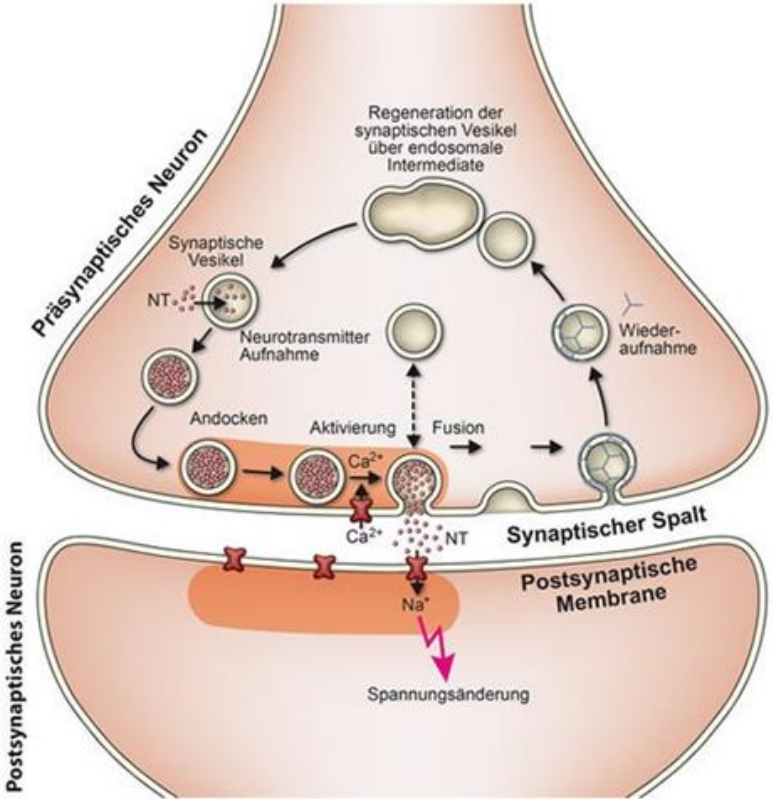




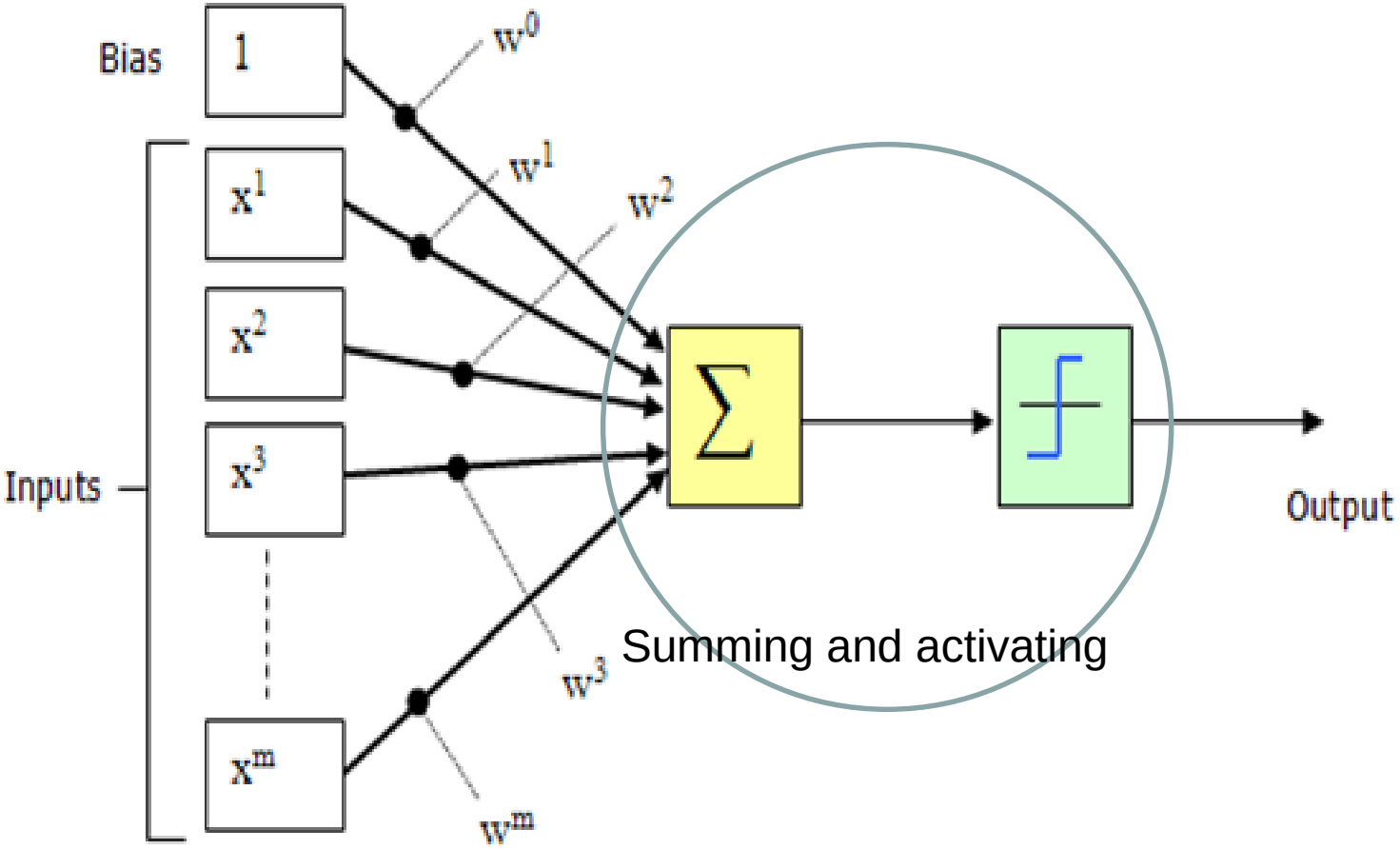
Is it worth imitating a brain?

ML inspired by brain



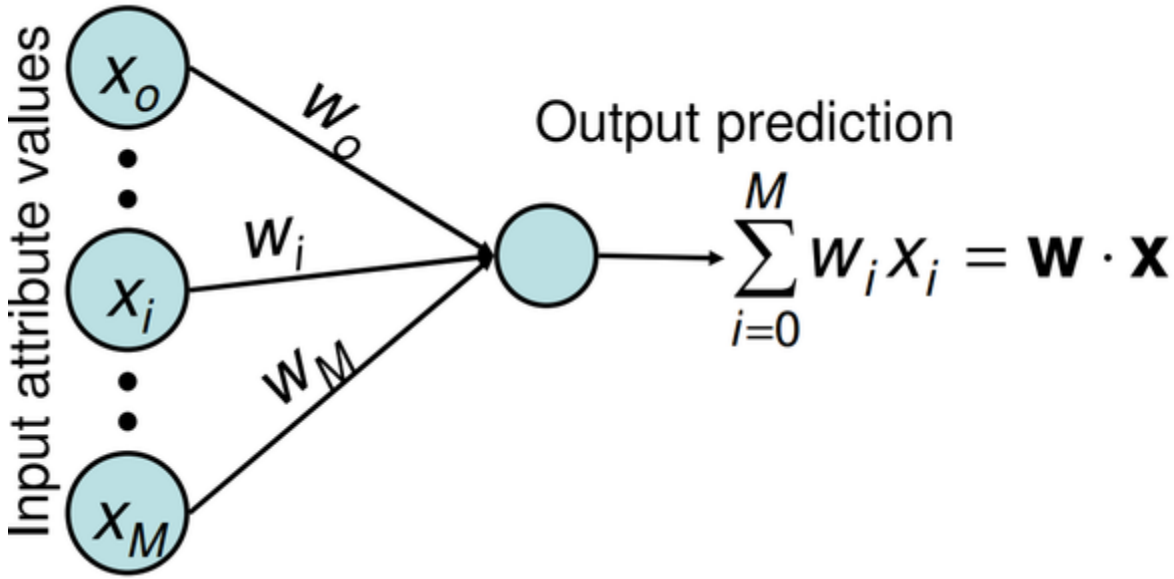


Perceptron


Bias
1
w^0
x^1
w^1
x^2
w^2
x^3
Inputs
x^m
w^3
w^m
Σ
Summing and activating
Output

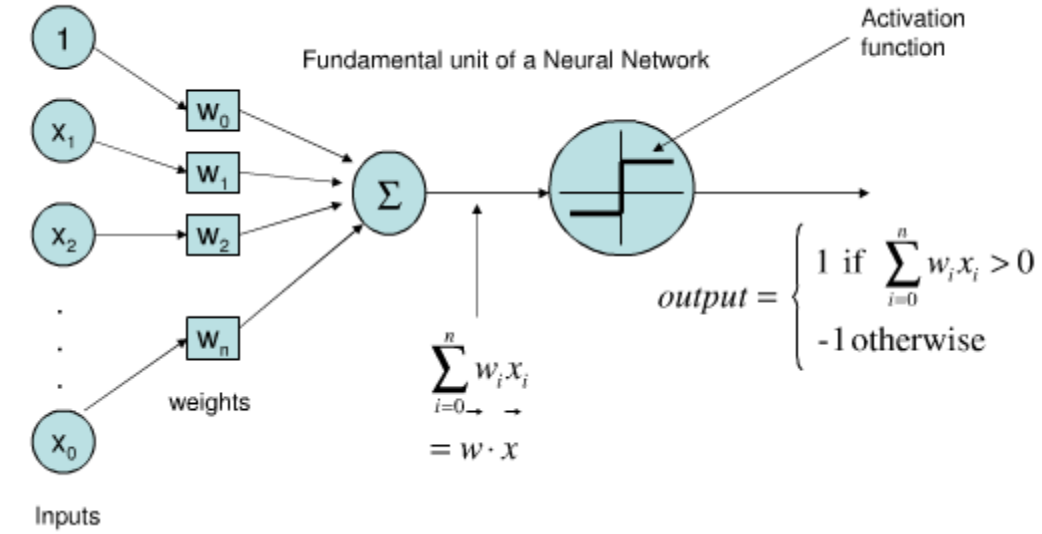
Popular architectures and activations

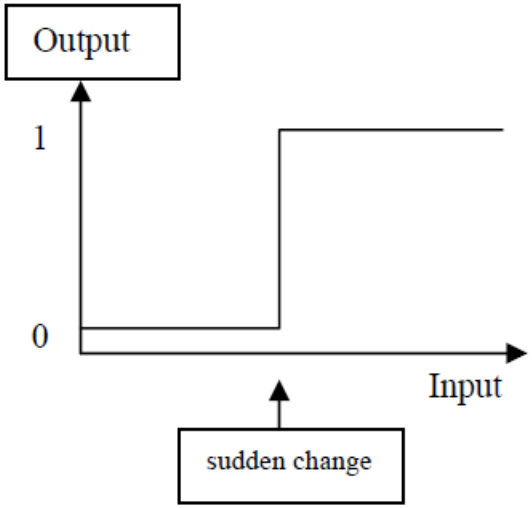
Regression activation: Identity

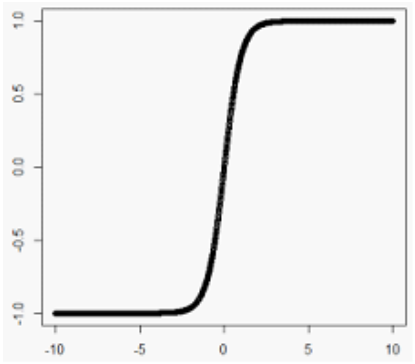


Popular architectures and activations

Binary classification

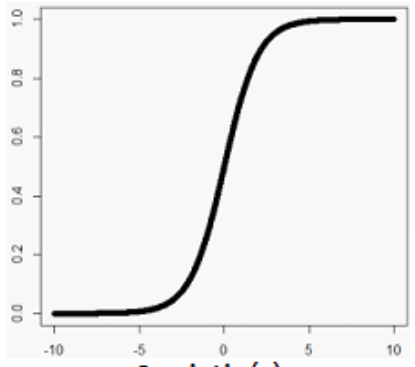






Tanh(x)

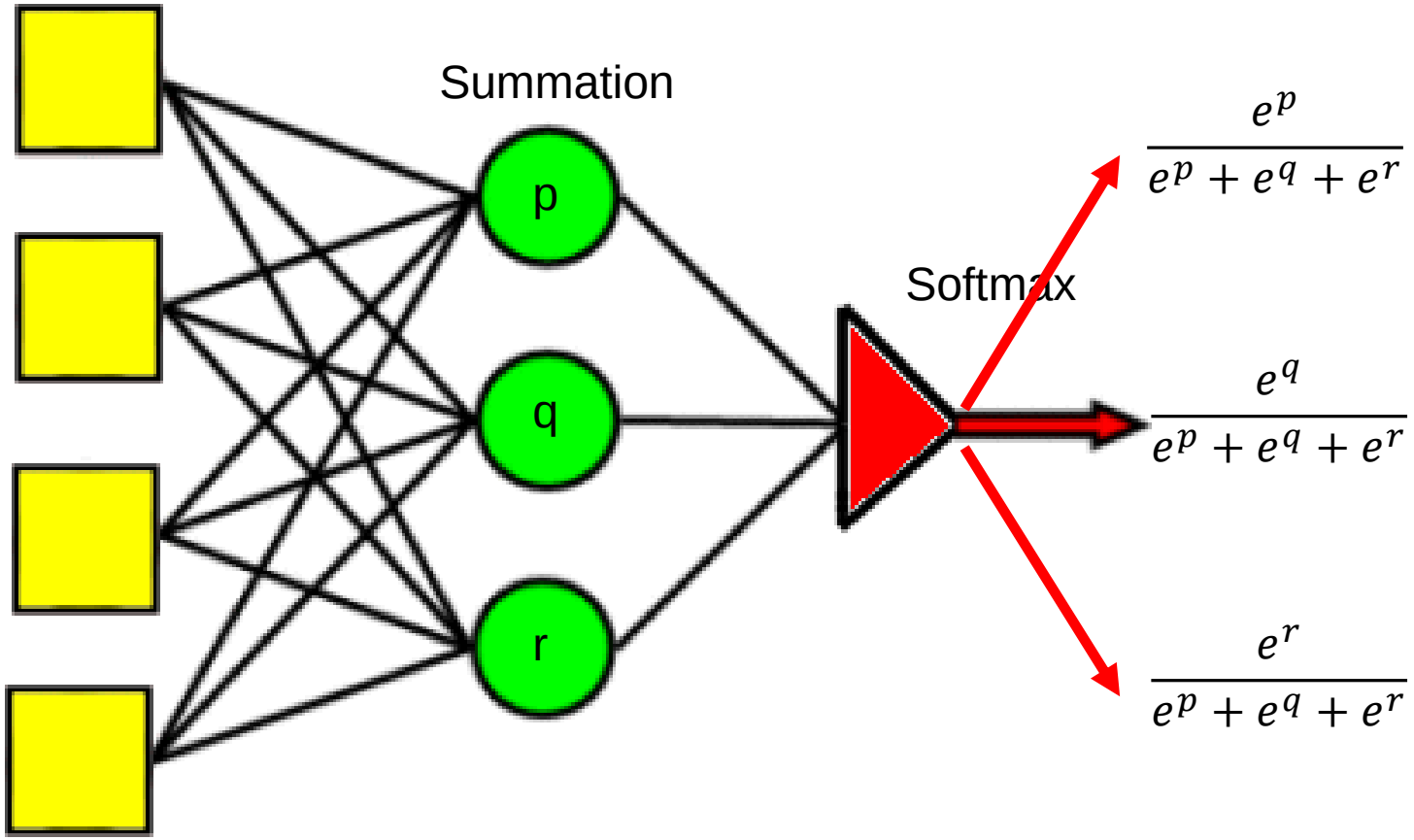
$$\frac{e^x - e^{-x}}{e^x + e^{-x}}$$



Logistic(x)

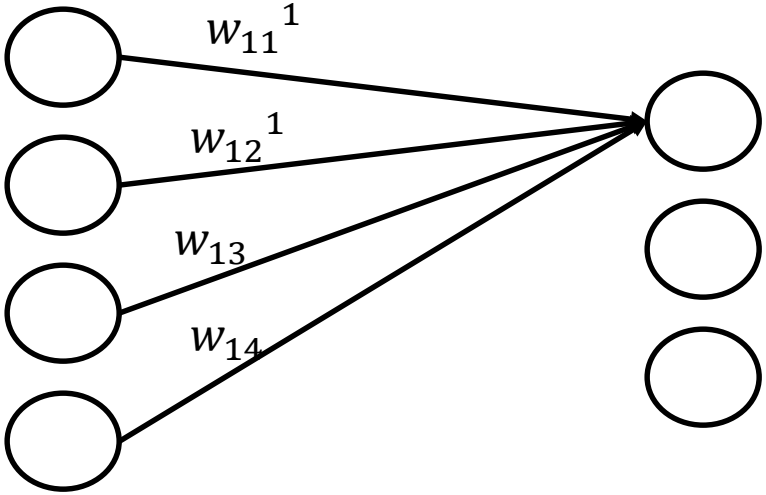
$$\frac{1}{1 + e^{-x}}$$

Multi class classification



4X3 Perceptron: Neural Representation

- Let us say, there are 4 features; Let there be three classes;



First subscript in weights refers to output and second subscript to input

4X3 Perceptron: Matrix analysis

- Each data point will be a 4X1 vector (column)
- In Weights, rows represent output and columns inputs (3X4 matrix)
- We also add three bias terms (3X1)
- Output will be 3 class scores (3x1)

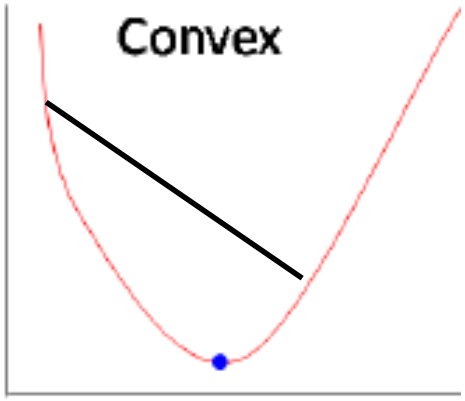
$$y_{3X1} = W_{3X4} * x_{4X1} + b_{3X1}$$

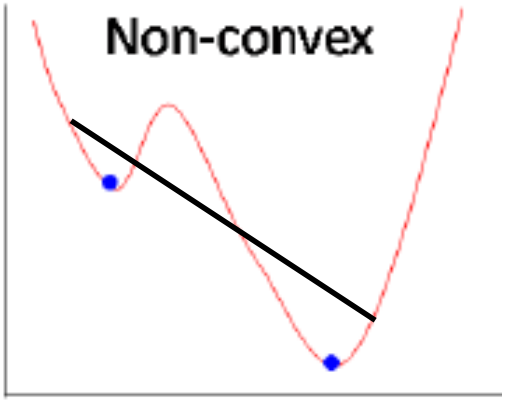
$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{44} \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Excel view

How are the best coefficients found?

- Define a goodness metric (called loss function) to measure how good the fit (for regression) or separation (for classification) is;
  - The coefficients that give the best fitness or separation are the best
- Ideal properties of an loss function
  - Robust: Does not explode with outliers
  - Non-ambiguous: Multiple coefficient values should not give same error
  - Sparse: Should use as little data as possible
  - And



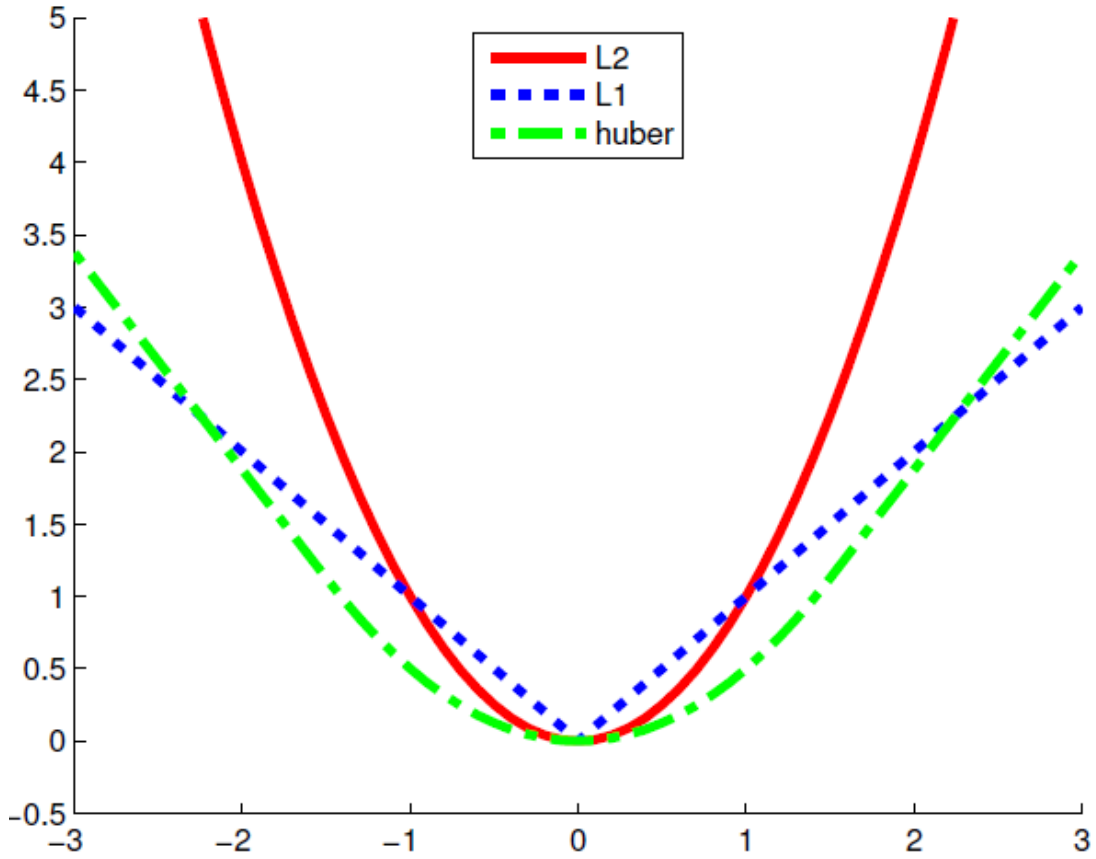


It must be convex

- We need to find the set of parameters that minimize error. This involves finding derivative of the error function.
- Hence, convexity is an important characteristic of an error function
- If we find a minima of a convex function, it will be a global minima

Common loss functions in regression

- Square (l_2) loss $(y - w_i x_i)^2$
  - Properties
    - Convex, Not robust (but OK), Non-Sparse
- Absolute (l_1) loss $|y - w_i x_i|$
  - Properties
    - Convex (not-smooth), Not robust (but OK), Non-Sparse
- Huber loss until some small error, it is l_2 and then l_1
  - More robust and differentiable



Binary classification: Margin

- $y.f(x) is\ called\ margin$
- Classes are assumed to be -1 and 1
- If prediction and reality have same sign, margin is +ve, no loss. Else, loss of 1

Binary classification: Exponential loss

$$L(w, b, X, y) = \sum_{i=1}^{n} e^{-y_i(w_1 x_1 + b)}$$

If prediction and real are same sign, loss is low. Else, it is high.

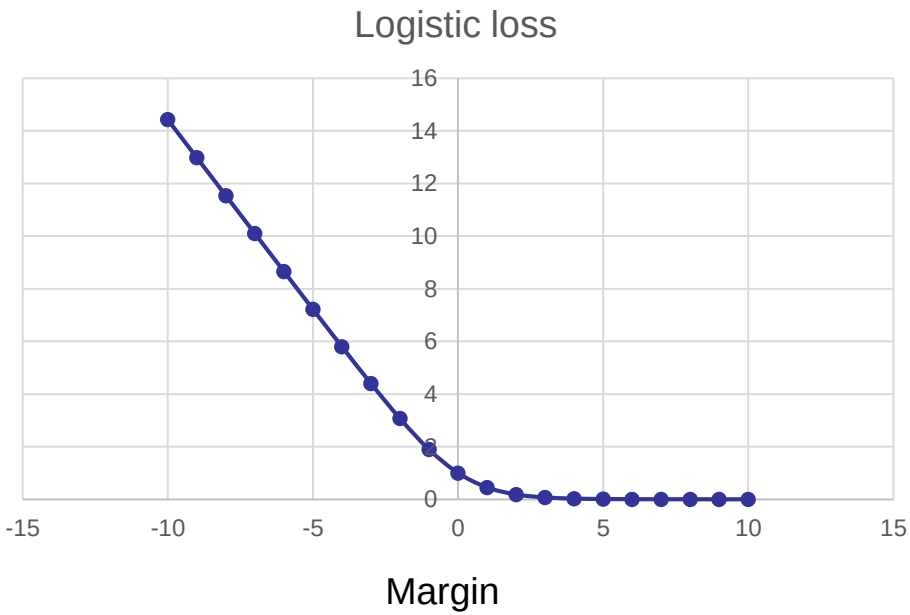
- Properties
  - Smooth and Convex
  - Not robust (bad)
  - Less sparse in dual but OK

Binary classification: Logistic loss function

$$L(w, b, X, y) = \sum_{i=1}^{n} \log(1 + e^{-y_i(w_1 x_1 + b)}) =$$

$$\sum_{i=1}^{m} y^{(i)} \log P(y = 1) + (1 - y^{(i)}) \log P(y = 0)$$

If prediction and real are same sign, loss is low. Else, it is high.



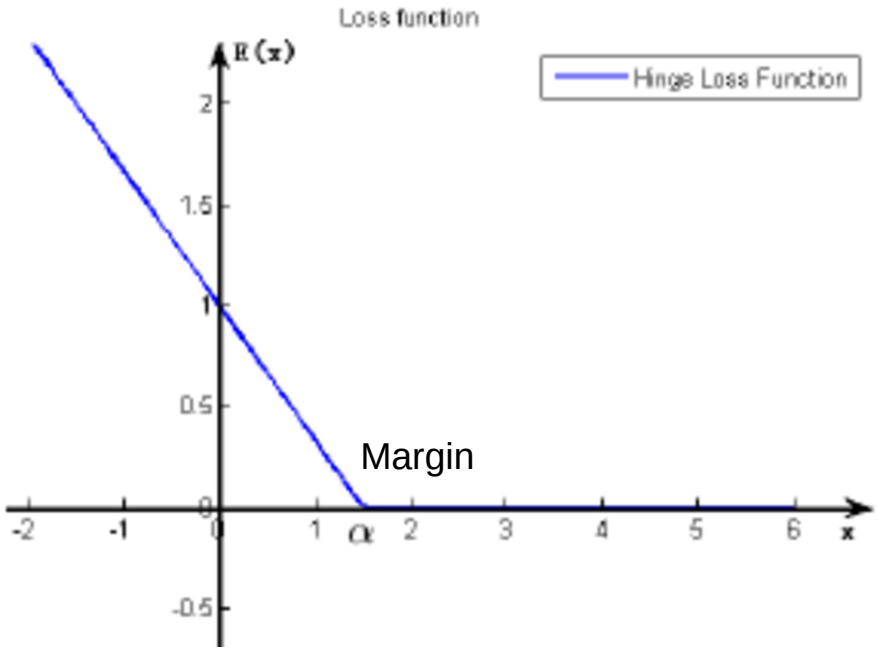
- Properties
  - Smooth and Convex
  - Not robust but better than exponential
  - Less sparse in dual but OK

Binary classification: Hinge loss

The margin should be more than k

- $\max(0, k - yf(x))$
  - Margin k, typically is 1
  - When margin is 0, it is called central hinge loss

- Properties
  - Convex
  - Not robust (but OK)
  - Sparse in dual

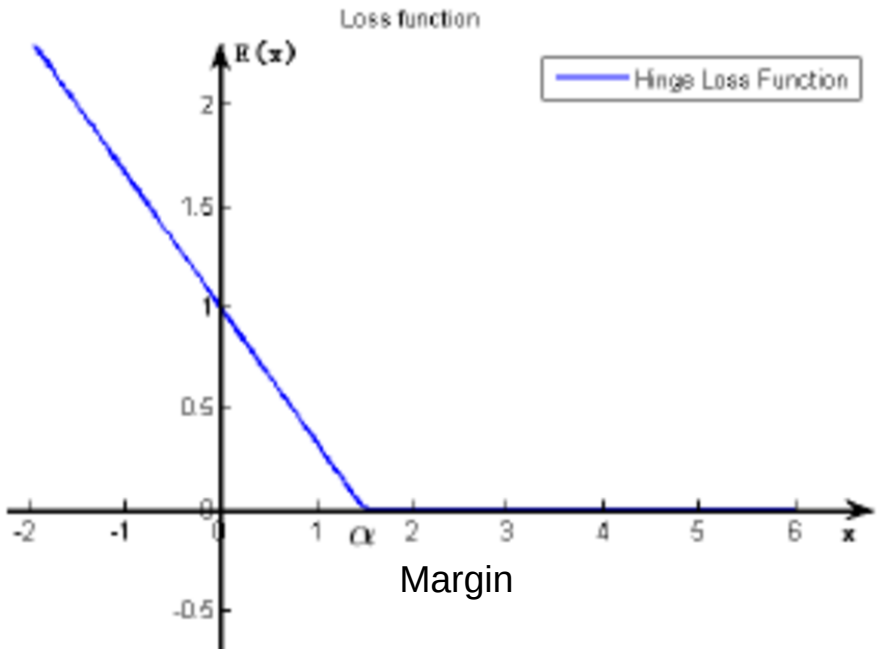


Multi class Hinge loss

Right answer must be k more than the wrong answer

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

- Example
  - Let us say, three classes receive 8, -6, 10) and 8 is the true class
  - =max(0, -6-8+1) + max(0, 10-8+1)=0+3=3
  - It wants the non-true classes to be having a value less than the margin from the true class. If score of the wrong class is within the margin or more, then there is a penalty



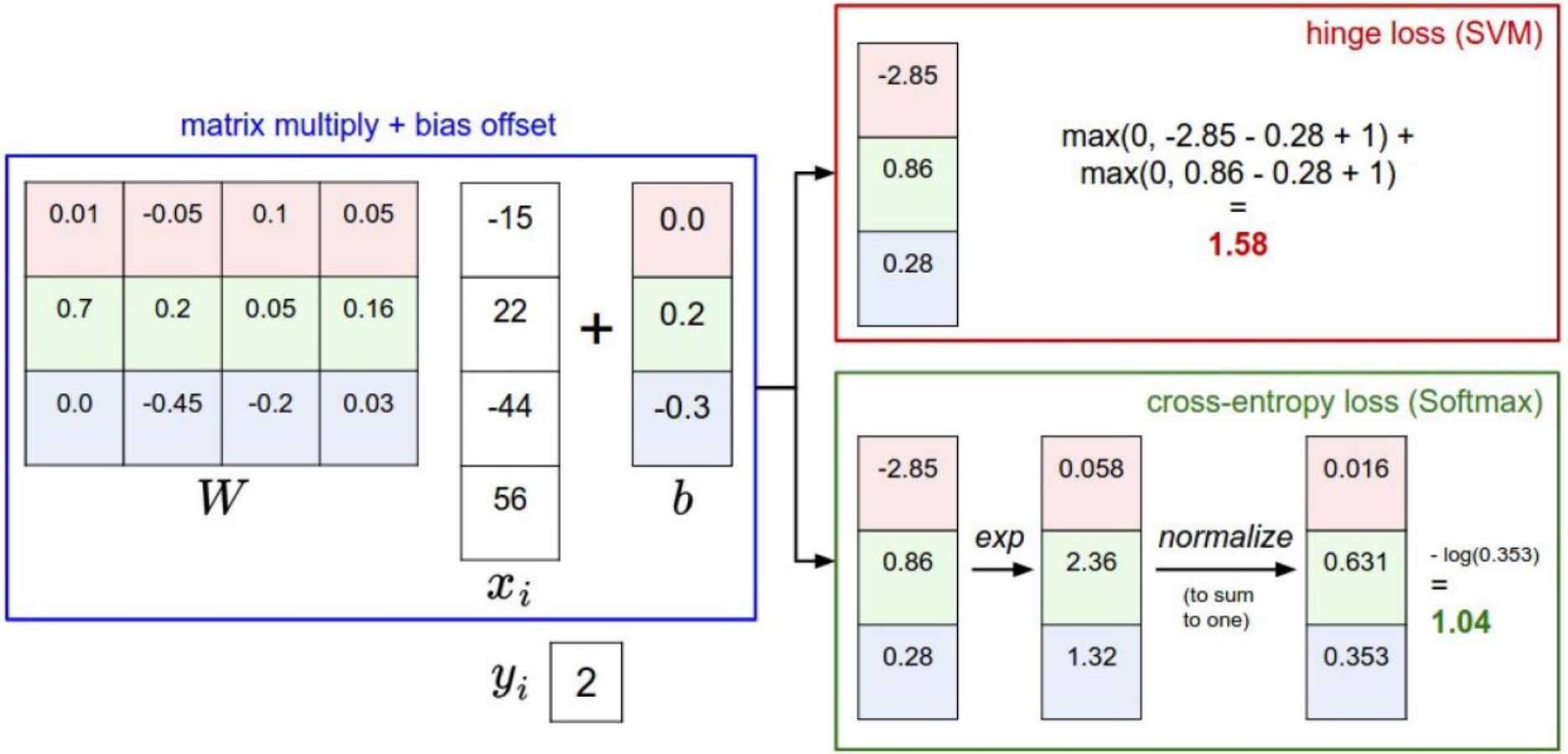
Multi class: Cross entropy loss of a softmax classifier

Cross entropy between true and predicted distributions is defined as

$$H(p, q) = -\sum_{x} p(x) \log q(x)$$

In cross entropies, predicted scores are converted to probabilities by normalizing, true probability is 1 always for the right class. So, cross entropy loss is written as

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$


matrix multiply + bias offset
0.01
-0.05
0.1
0.05
0.7
0.2
0.05
0.16
0.0
-0.45
-0.2
0.03
W
-15
22
-44
56
x_i
+
0.0
0.2
-0.3
b
y_i
2
hinge loss (SVM)
-2.85
0.86
0.28
max(0, -2.85 - 0.28 + 1) +
max(0, 0.86 - 0.28 + 1)
=
1.58
cross-entropy loss (Softmax)
-2.85
0.86
0.28
exp
0.058
2.36
1.32
normalize
(to sum
to one)
0.016
0.631
0.353
- log(0.353)
=
1.04

Regularization functions

Ridge

Simplest is whose sum of squares is the least

$$\min_{w}\{\frac{1}{n}(\hat{X}w - \hat{Y})^2 + \lambda\|w\|_2^2\}$$

Lasso

Simplest is whose absolute sum is the least

$$\min_{w\in\mathbb{R}^p}\{\frac{1}{n}\|\hat{X}w - \hat{Y}\|^2 + \lambda\|w\|_1\}$$

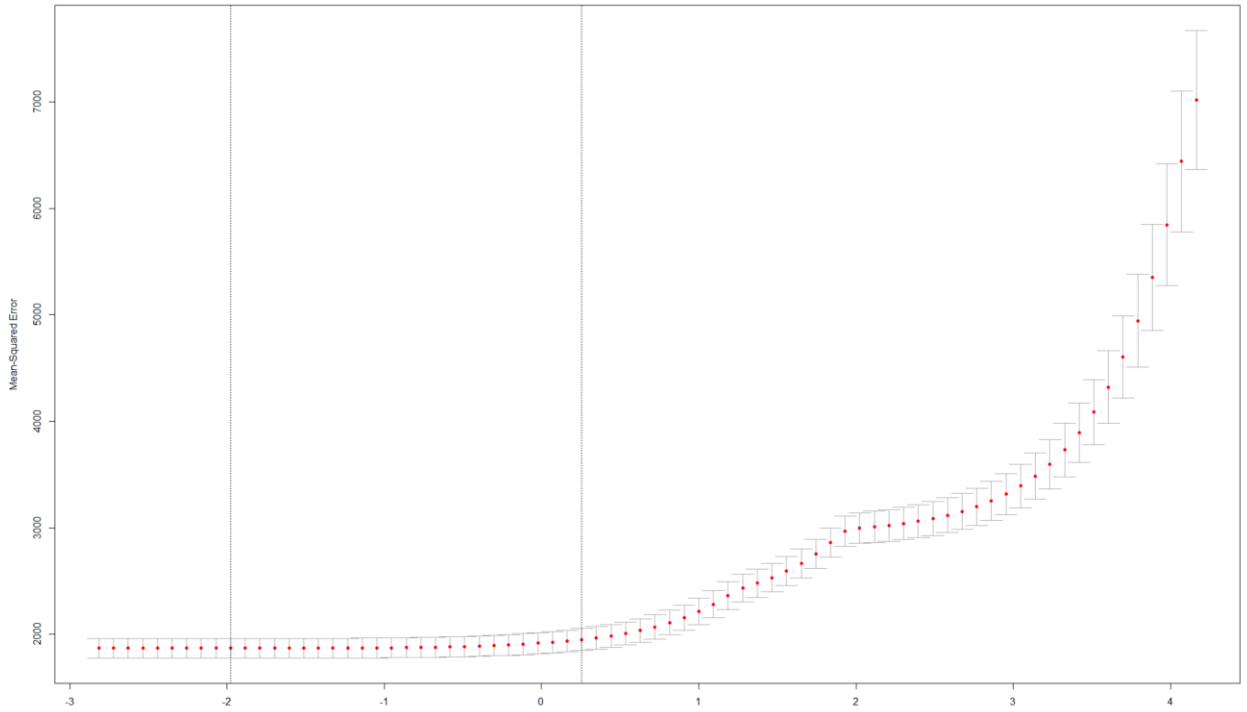
ElasticNet

$$\min_{w\in\mathbb{R}^p}\{\frac{1}{n}\|\hat{X}w - \hat{Y}\|^2 + \lambda(\alpha\|w\|_1 + (1-\alpha)\|w\|_2^2), \alpha \in [0,1]\}$$

High lambda gives simplest models; At infinity, we have no model

Pick the right Weight Decay

- A logarithmic monotonous search for the best weight decay



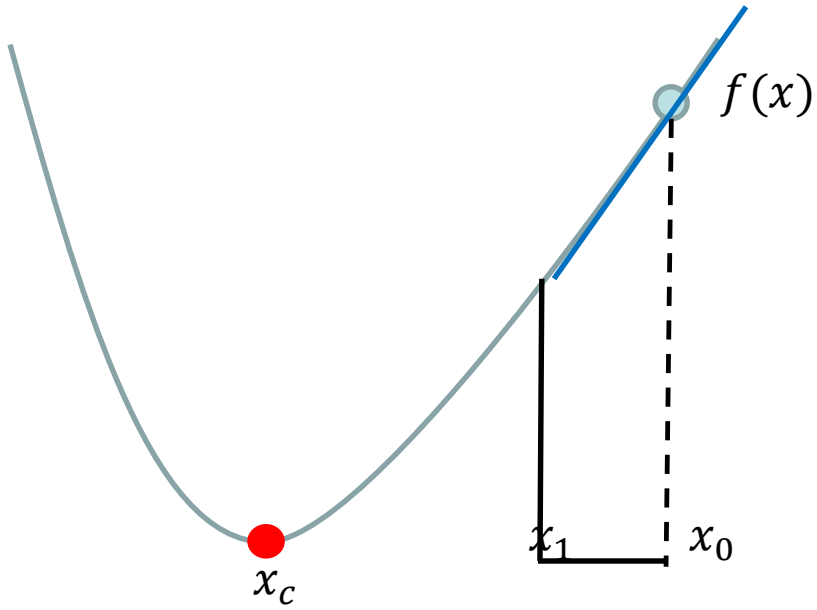
Loss functions and problems

	Activation in the final layer	Loss function	Regularization
Regression	Unity	Squared loss, Absolute loss, Huber loss	Ridge regularization, Lasso regularization, Elastic net regularization
Binary classification	Sigmoid, Tanh, RELU, Threshold	Exponential loss, Logistic loss, hinge loss	
Multi-Class	Soft max if probability should add up to 1. Sigmoids if they are independent	Cross entropy loss, multi-class hinge loss	

Finding the minima

GD intuition

- In practice, we scale down the gradient by a parameter, α, called learning rate. Learning rate is between 0 and 1



Gradient descent

- Any function can be approximated to be a quadratic in a small vicinity. Let us start at x0

$$f(x) = a(x - x_0)^2 + b(x - x_0) + c$$

- The first derivative is

$$f'(x) = 2a(x - x_0) + b$$

$$f'(x_0) = b;$$

Gradient descent

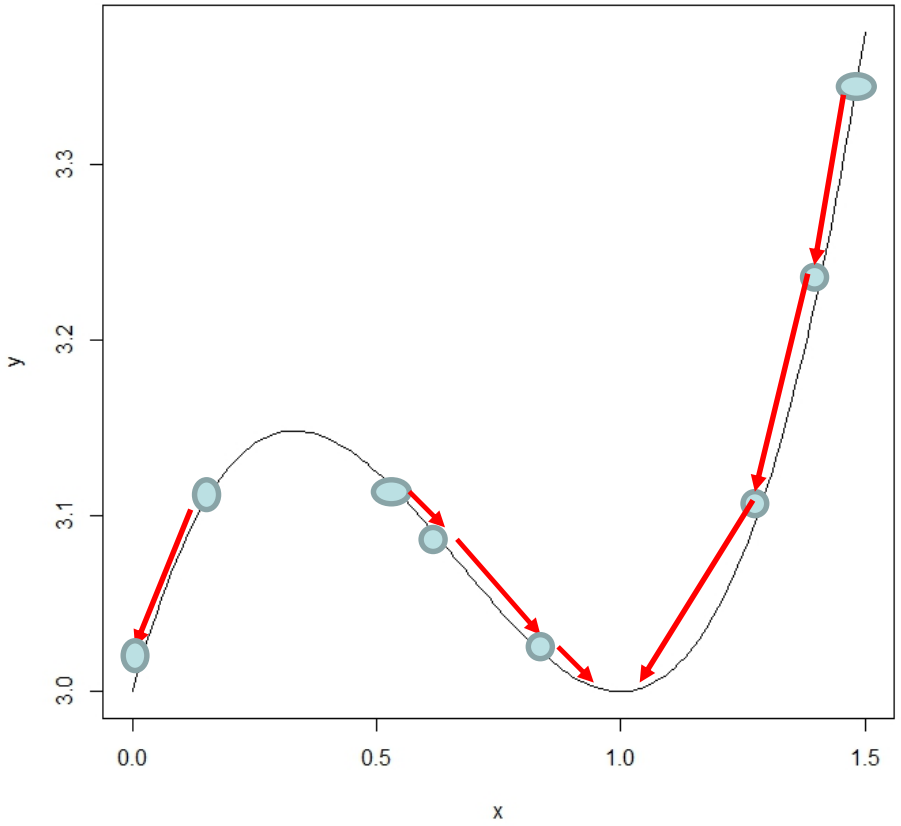
- We set the gradient to zero for critical points

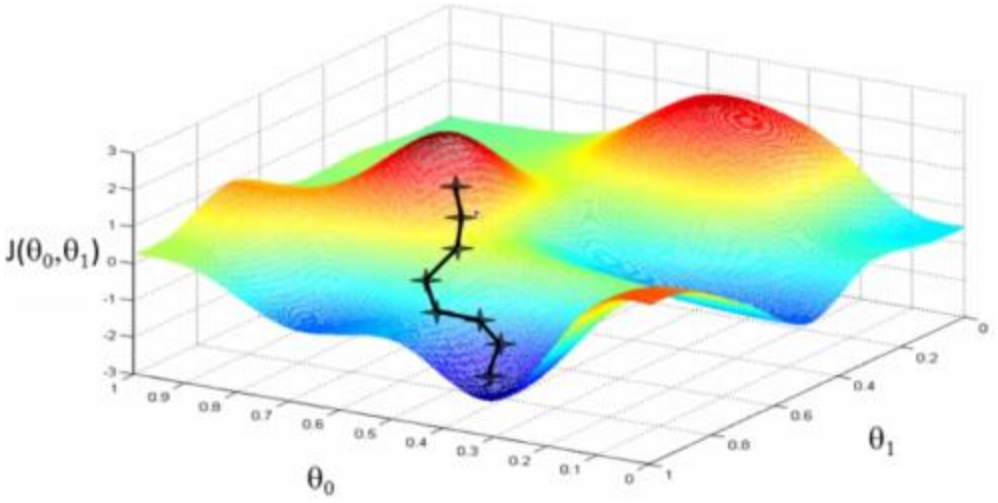
$$0 = 2a(x_{crit} - x_0) + b$$

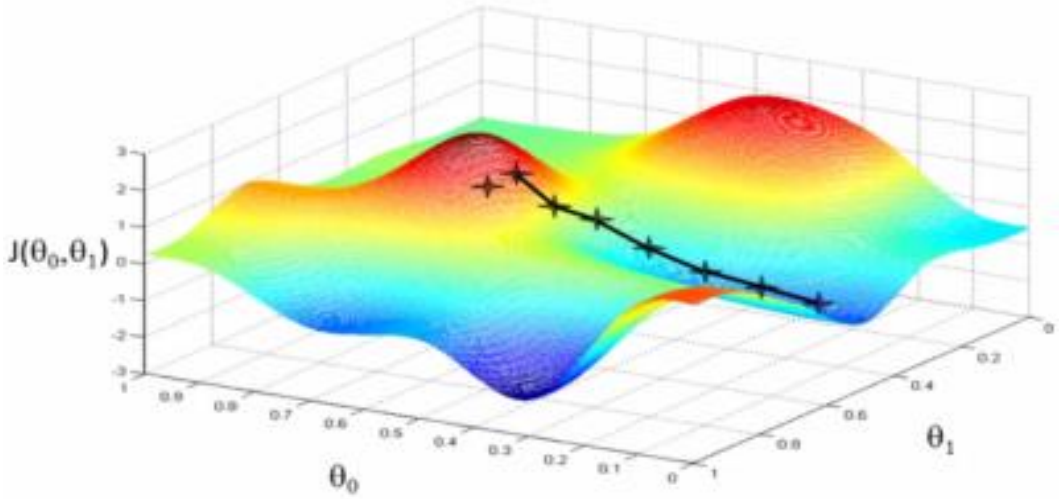
$$x_{crit} = x_0 - \frac{b}{2a} = x_0 - \frac{f'(x_0)}{2a} = x_0 - \alpha f'(x_0)$$

We chose nice loss functions that can be differentiated w.r.t weights! Start at some point and iterate

GD and local minima







Delta rule

- Gradient through chain rule

$$E \text{ is normally expressed as } (y_{real} - y_{pred})$$

$$y_{pred} = act\left(\sum w_i x_i\right) = f(wx)$$

$$Using\ chain\ rule, \frac{\partial E}{\partial w} = \frac{\partial E}{\partial y}.\frac{\partial y}{\partial w} = \frac{\partial E}{\partial y}.f'.x$$

Derivative of error function w.r.t y * derivative of activation * input

Perceptron learning

- Learning is changing weights
- In the very simple cases
  - **Start random**
  - **If** the output is correct **then** do nothing.
  - **If** the output is too high, decrease the weights attached to high inputs
  - **If** the output is too low, increase the weights attached to high inputs

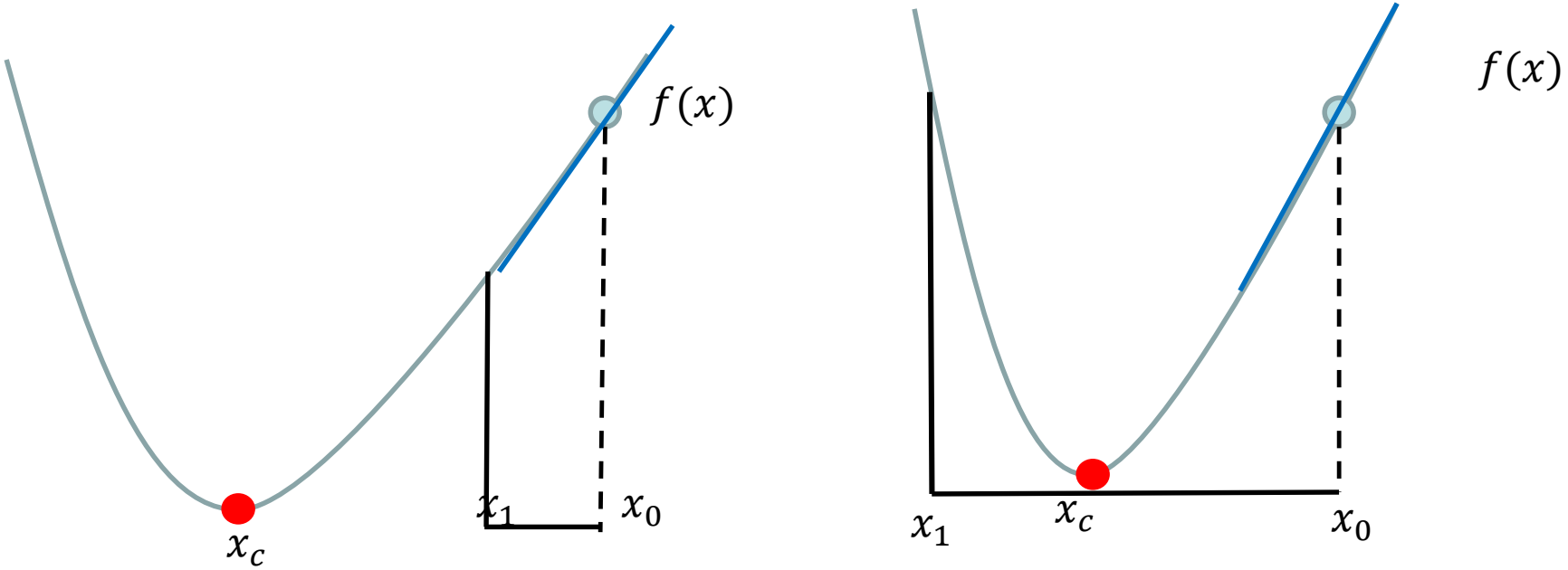
$$w_{t+1} = w_t - \frac{\partial E}{\partial W} = w_t - \frac{\partial E}{\partial y}.f'.x$$

For a square error and linear activation, this translates to

$$w_i(t+1) = w_i(t) + (d_j - y_j(t))x_{j,i}$$

Learning rate

- In practice, we scale down the gradient by a parameter, α, called learning rate. Learning rate is between 0 and 1



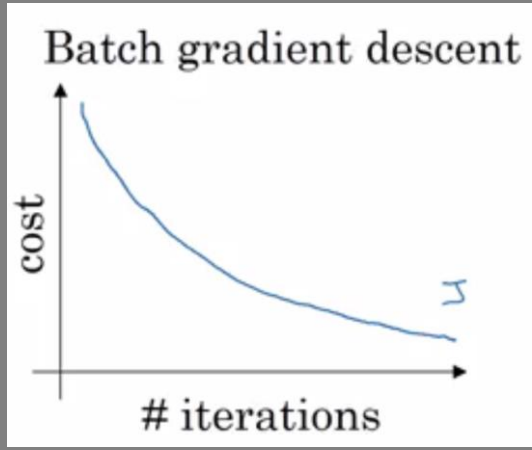
Excel

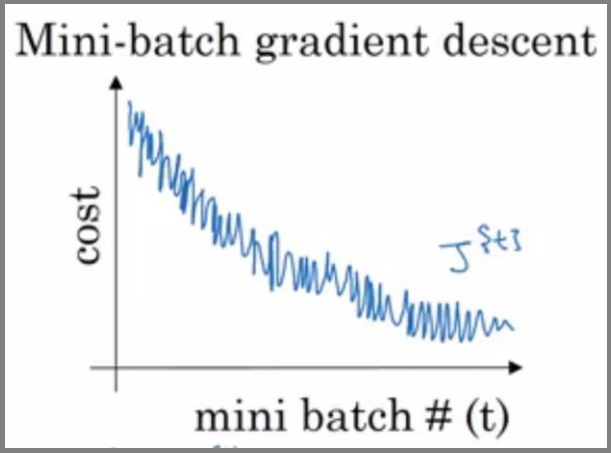
Multiple methods of updating weights

Online	Batch	Mini-Batch
Show an input…adjust weights, show another and adjust weights…,once the input is all over, start with row 1 if needed	Show an input, compute the adjustment needed to the weights and store it. Show the second input (with original weights) and continue the process. Update once all inputs are shown	Pick a small random sample of inputs. Perform batch update. Then pick another set of sample inputs randomly…

More on min-batch gradient descent

- Let us say we have 500,000 samples and let us say, each mini-batch contains 512 samples. We make ~1000 mini batches
- We construct a matrix of 512 samples and forward prop. Make 500 predictions. Then find average cost over 512 samples. Then update the weights once.
- An epoch is doing all 1000 mini batches once. We do multiple epochs before convergence
- If mini batch is 1, it is stochastic gradient descent. If it is M, you have gradient descent
- For less than 2000 samples, go for batch. For large data sets, the mini-batch is 64-512 (a power of 2) is good.





Process

- Stochastic gradient descent

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} & b_1 \\ w_{21} & w_{22} & w_{23} & b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$$

- Mini-Batch

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} & b_1 \\ w_{21} & w_{22} & w_{23} & b_2 \end{bmatrix} \begin{bmatrix} {x_1}^1 & {x_1}^2 & {x_1}^3 & .. \\ {x_2}^1 & {x_2}^2 & {x_2}^3 & .. \\ {x_3}^1 & {x_3}^2 & {x_3}^3 & .. \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Superscript represents record number.
Subscript represents feature

https://www.jeremyjordan.me/nn-learning-rate/

SEARCHING FOR THE RIGHT LEARNING RATE

Learning rate and Weight decay

Andrej Karpathy
@karpathy

3e-4 is the best learning rate for Adam, hands down.

408 8:31 AM - Nov 24, 2016

124 people are talking about this

Andrej Karpathy @karpathy · Nov 24, 2016
3e-4 is the best learning rate for Adam, hands down.

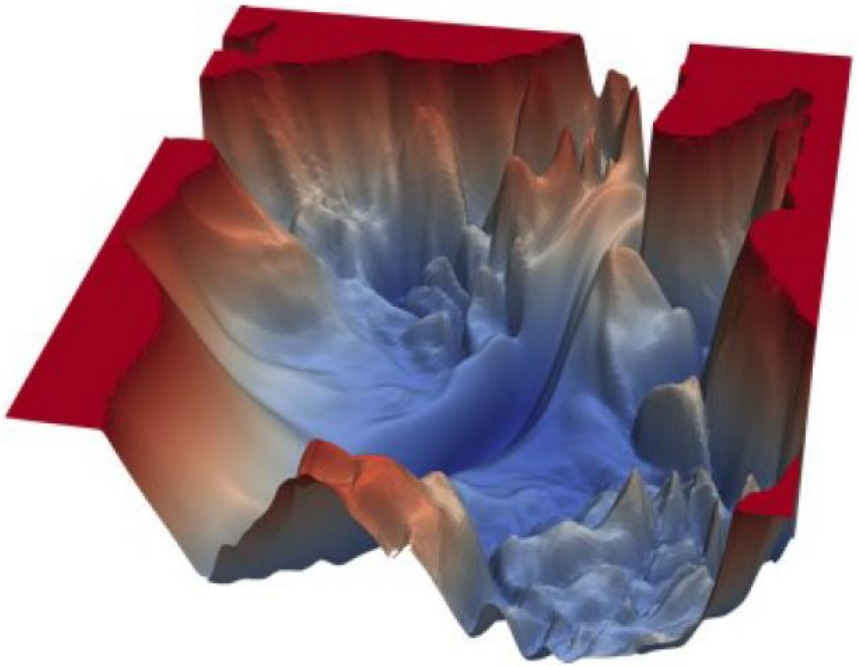
Andrej Karpathy
@karpathy

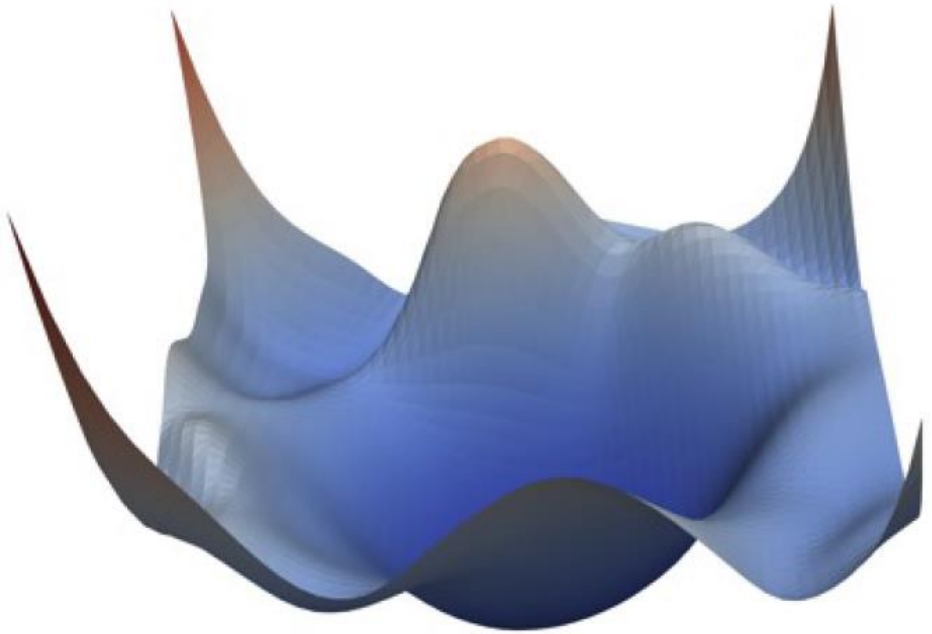
(i just wanted to make sure that people understand that this is a joke...)

103 1:21 PM - Nov 24, 2016

See Andrej Karpathy's other Tweets

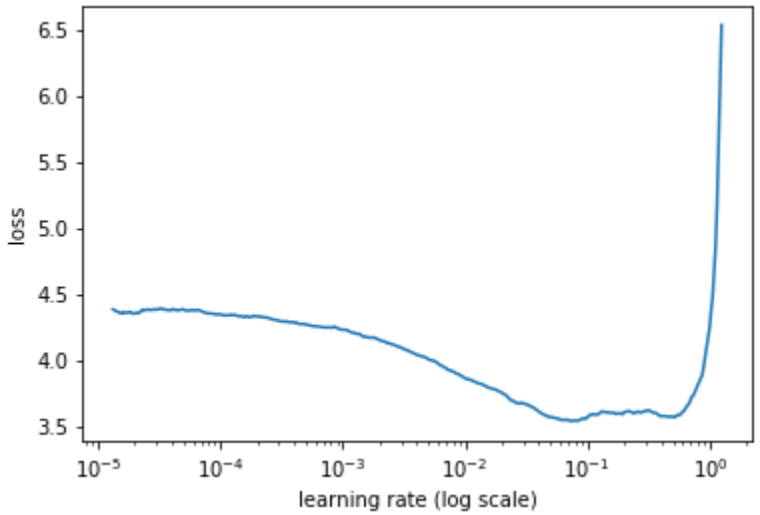
Why is it so difficult to find LR





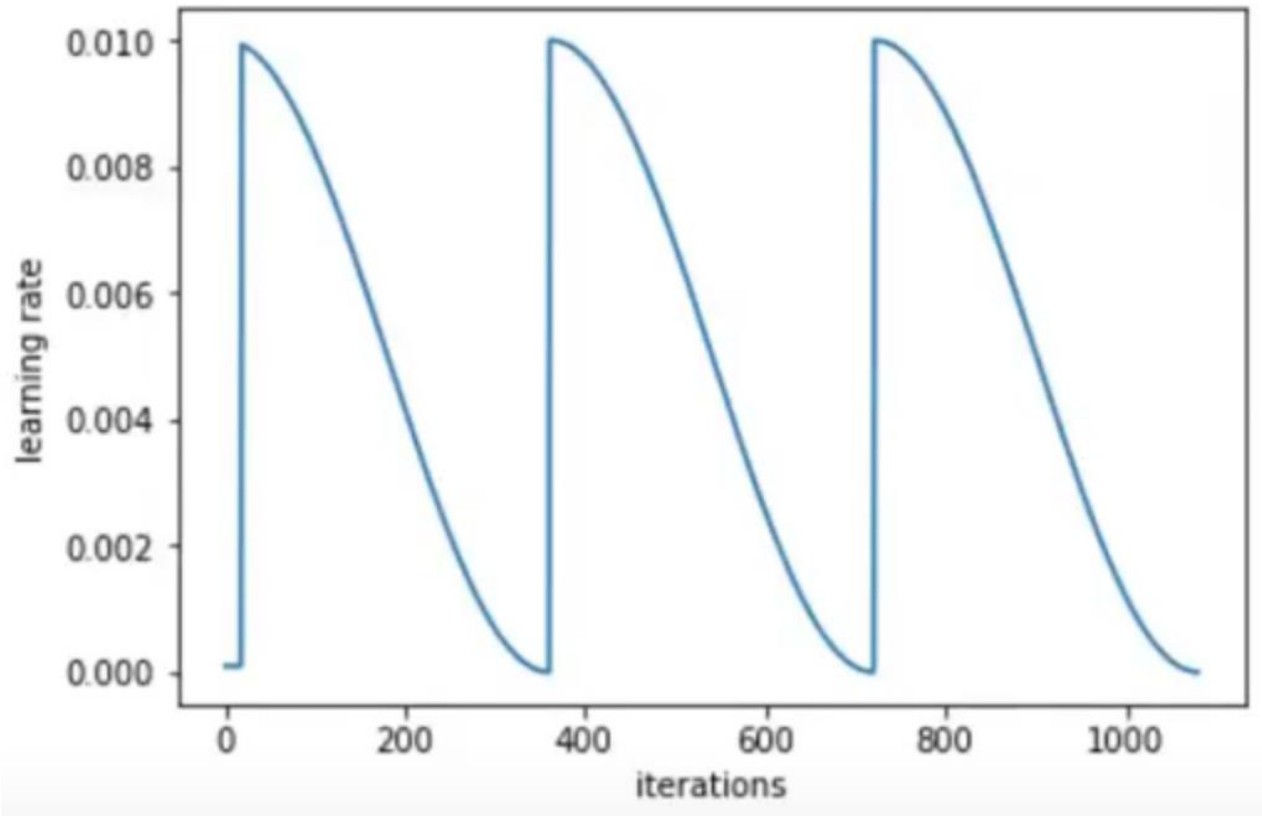
Hyper parameters in linear models

- Learning rate and weight decay are powerful hyper parameters
  - Use a default weight decay (10^{-2})
  - Search for LR using the following strategies



https://towardsdatascience.com/estimating-optimal-learning-rate-for-a-deep-neural-network-ce32f2556ce0

More on learning rate

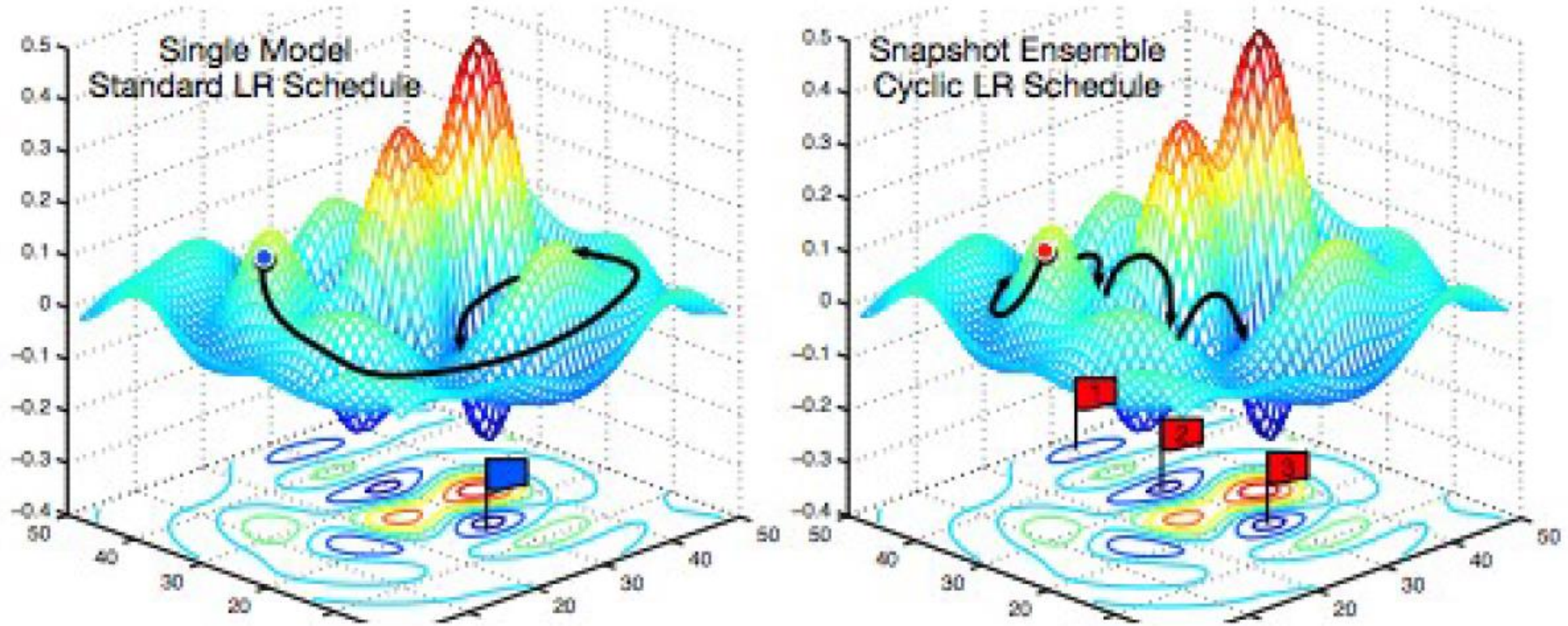


Flip between learning rates.

If what you find is a good local minima, you will be stable even if you increase the learning rate suddenly

https://medium.com/38th-street-studios/exploring-stochastic-gradient-descent-with-restarts-sgdr-fa206c38a74e

Neat idea

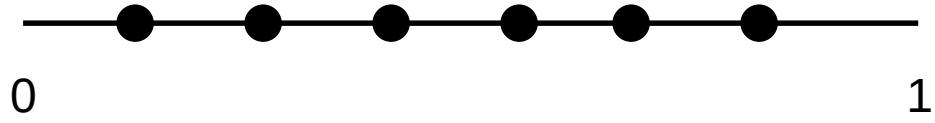


Good case for ensembles

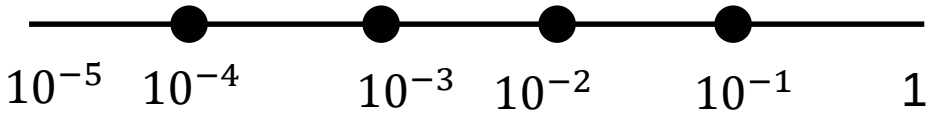
BUILDING A LINEAR MODEL

Framework

- Problem defines the inputs and outputs
- Pick default loss and regularization functions.
- Search for a decent learning rate through a random search in the log space (generate 100 random rates and pick the best after 100 mini batches)



Linear searches are bad as you focus only in one order of magnitude.



Logarithmic searches allow you to explore several orders of magnitude.

Fixing the weak parameters

- Activation, mini batch size, loss and regularization functions

- Pick the best LR from previous page and default Weight decay
  - Activation is defined for regression, multi class classification. For classification either randomly choose or experiment on 50 mini batches (64 size)/epoch whichever is low and pick the best
  - Loss and regularization: Pick what you like or experiment as before. But, one after another
  - Mini batch: Experiment between 32, 64, 128, 256 and 512 up to 50 minibatches and observe the loss. Pick the best.

For that Weight decay

- For the best LR of the previous slide, pick the weight decay

- For all these parameters, find LR range.
- SGDR within the range (3 cycles). Capture the weights at the lowest point
- Ensemble all three neural nets

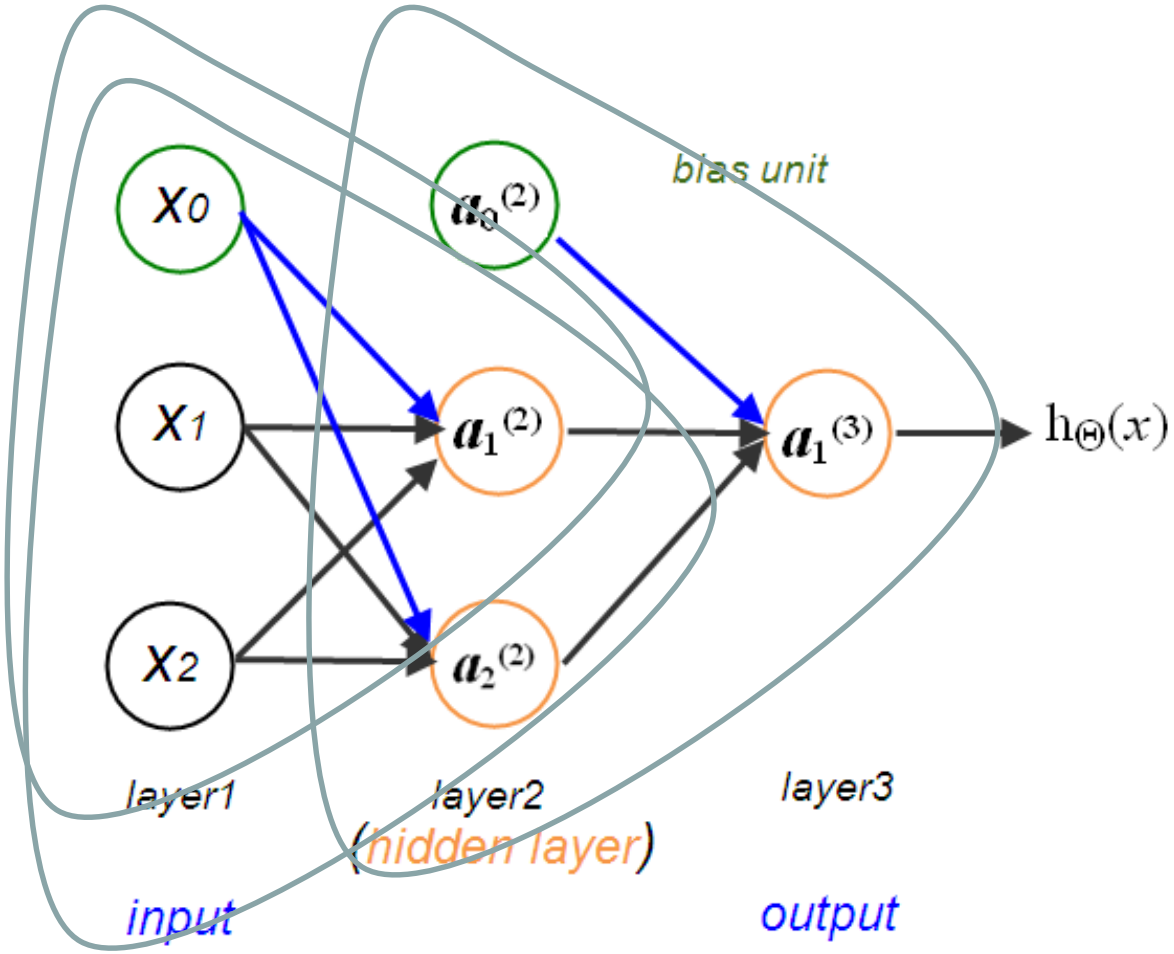
Understanding neural networks

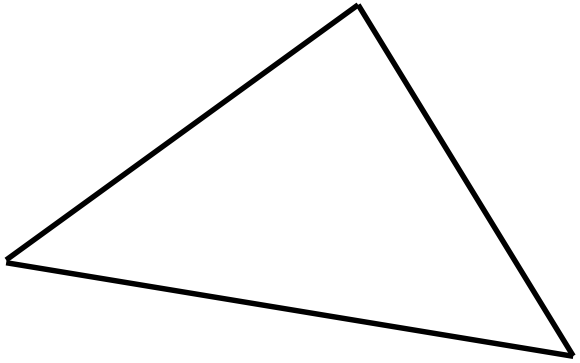
- Perceptron
  - Provides direct interpretations based on the sign of the weight.
  - If weight is positive, y increases with x else it decreases
  - Magnitude of the weight indicates the impact of that variable

Adding non linearity to the system

MLP

MLP





Ensemble?

Or

Spectral?

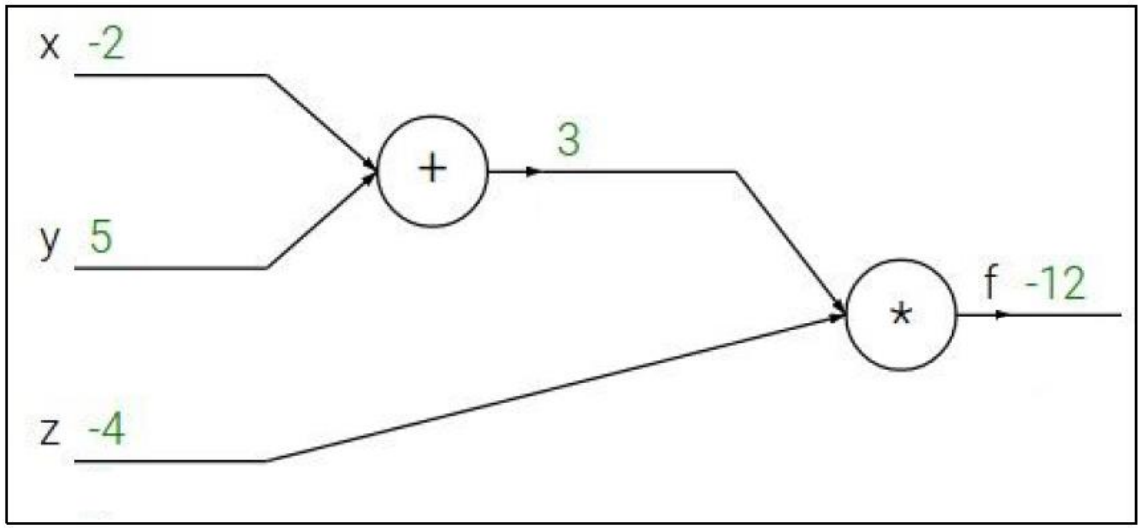
An interesting way to understand MLP
What are hidden layers?

- They are non-linear sums of the inputs
- So, they are not linearly dependent on inputs (they are non-linearly dependent)
- They are engineered features of the original inputs!
  - Feature engineering is so difficult because for each type of data and each type of problem, different features do well
  - Neural networks can potentially ***build features hierarchically***

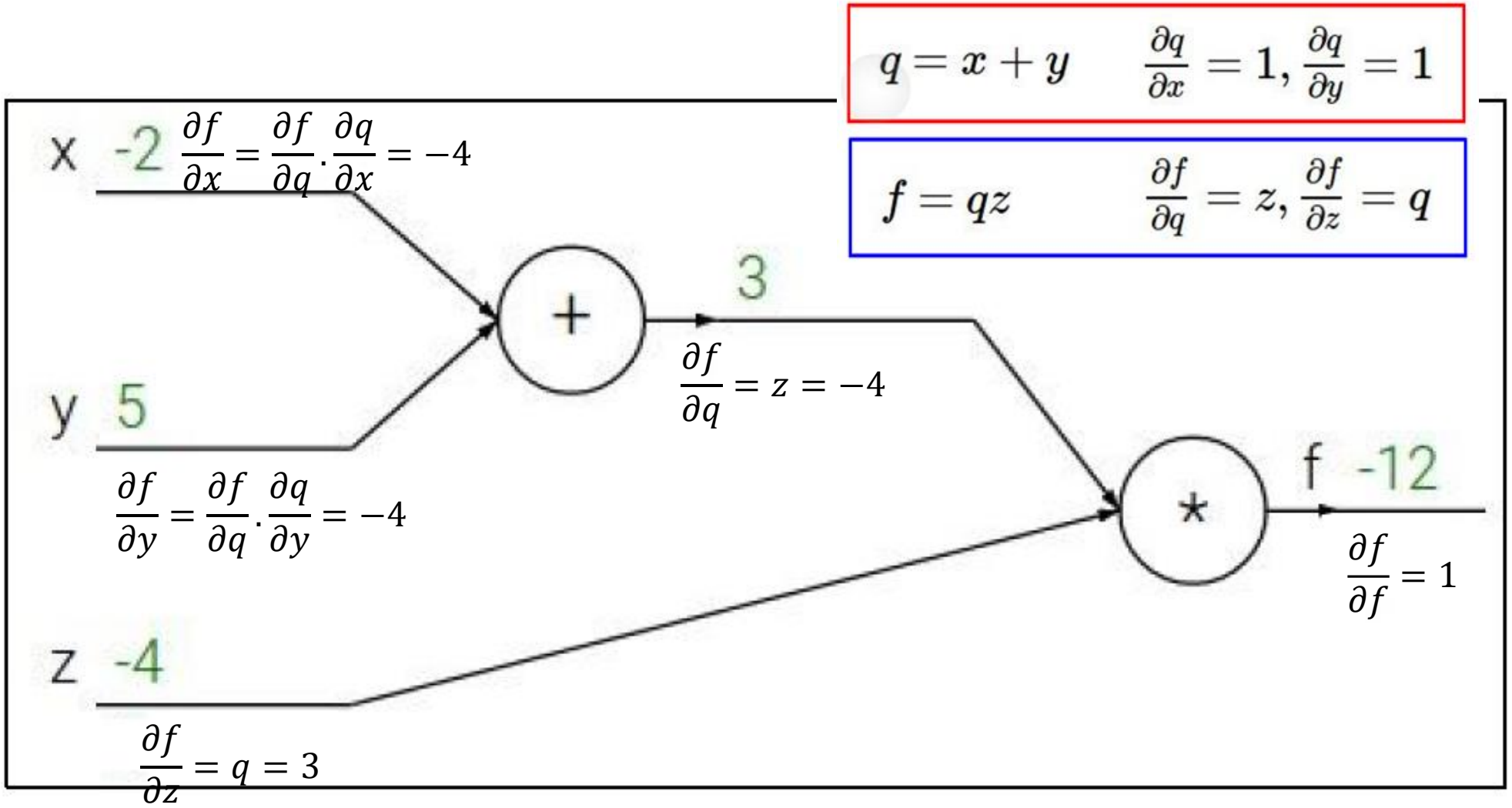
Back propagation algorithm

There were no training algorithms for multi layered networks until 80s. Geoff Hinton, then a young professor at Carnegie-Mellon University, solved it with back-propagation algorithms. He used chain rule of differentiation to propagate the error backwards and update the weights.

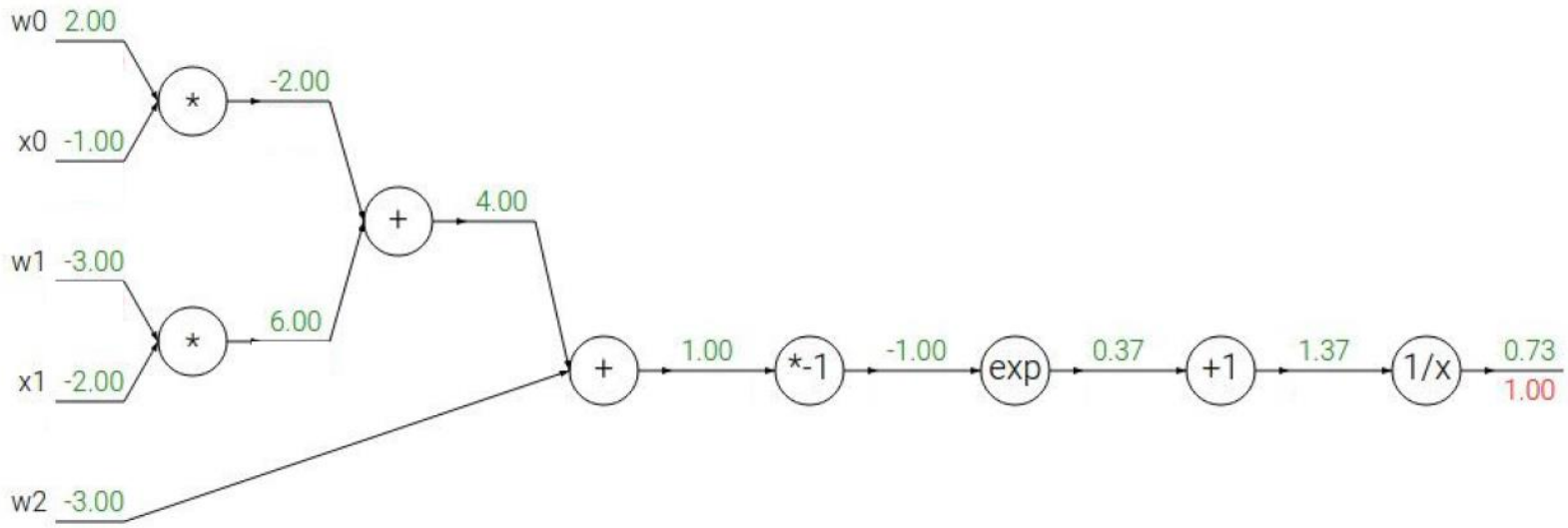
$$f(x, y, z) = (x + y)z \qquad x = -2, y = 5, z = -4$$



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


$q = x + y$ $\frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$
$f = qz$ $\frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$
x -2
$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \cdot \frac{\partial q}{\partial x} = -4$
y 5
$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \cdot \frac{\partial q}{\partial y} = -4$
+
3
$\frac{\partial f}{\partial q} = z = -4$
z -4
$\frac{\partial f}{\partial z} = q = 3$
*
f -12
$\frac{\partial f}{\partial f} = 1$

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



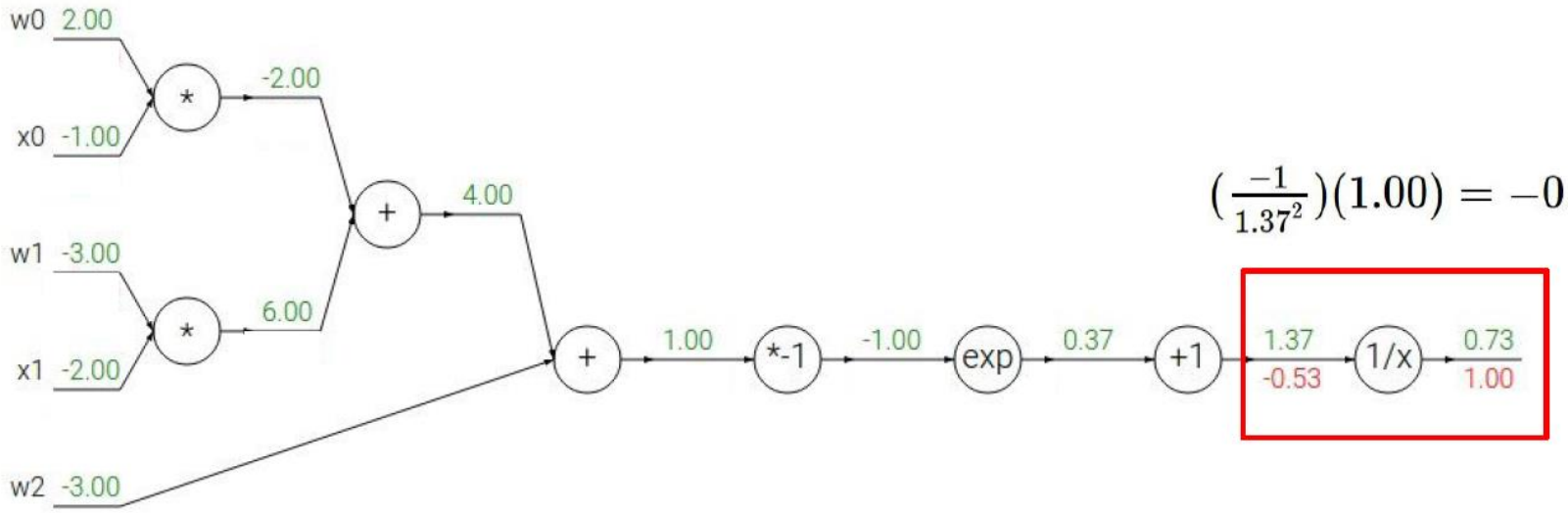
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

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

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

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

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



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

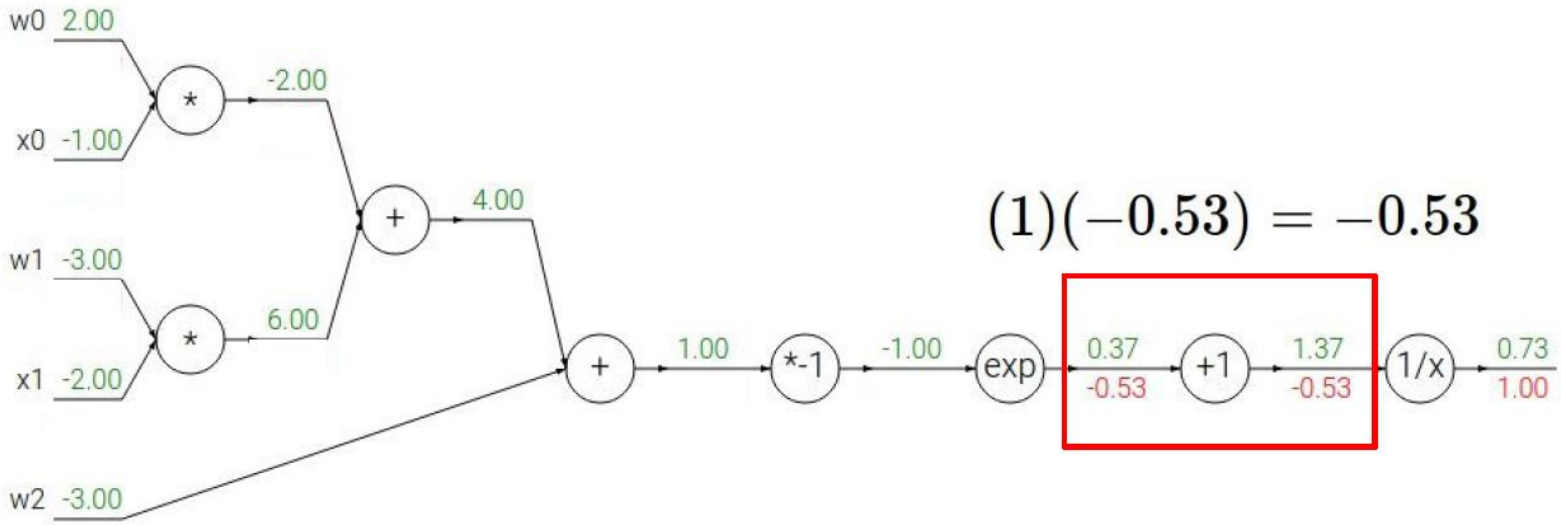
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

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

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

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

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



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

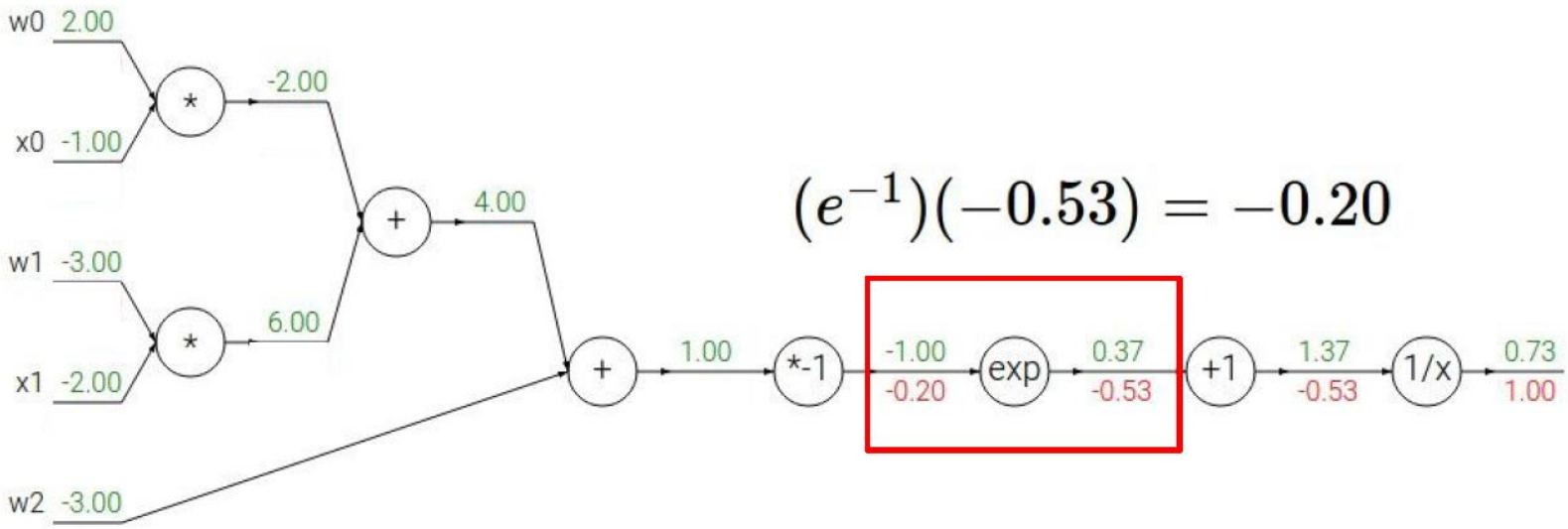
$$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$$

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

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

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

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

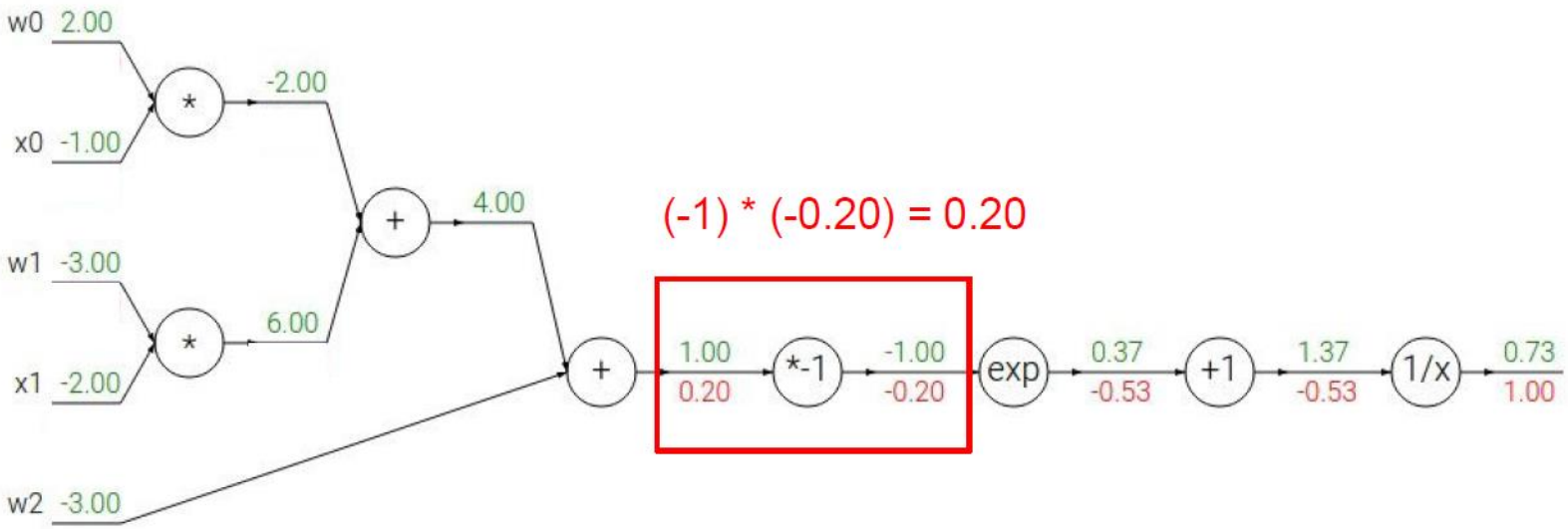


$$(e^{-1})(-0.53) = -0.20$$

$f(x) = e^x \rightarrow \frac{df}{dx} = e^x$	$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$
$f_a(x) = ax \rightarrow \frac{df}{dx} = a$	$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$

Another example:

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



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

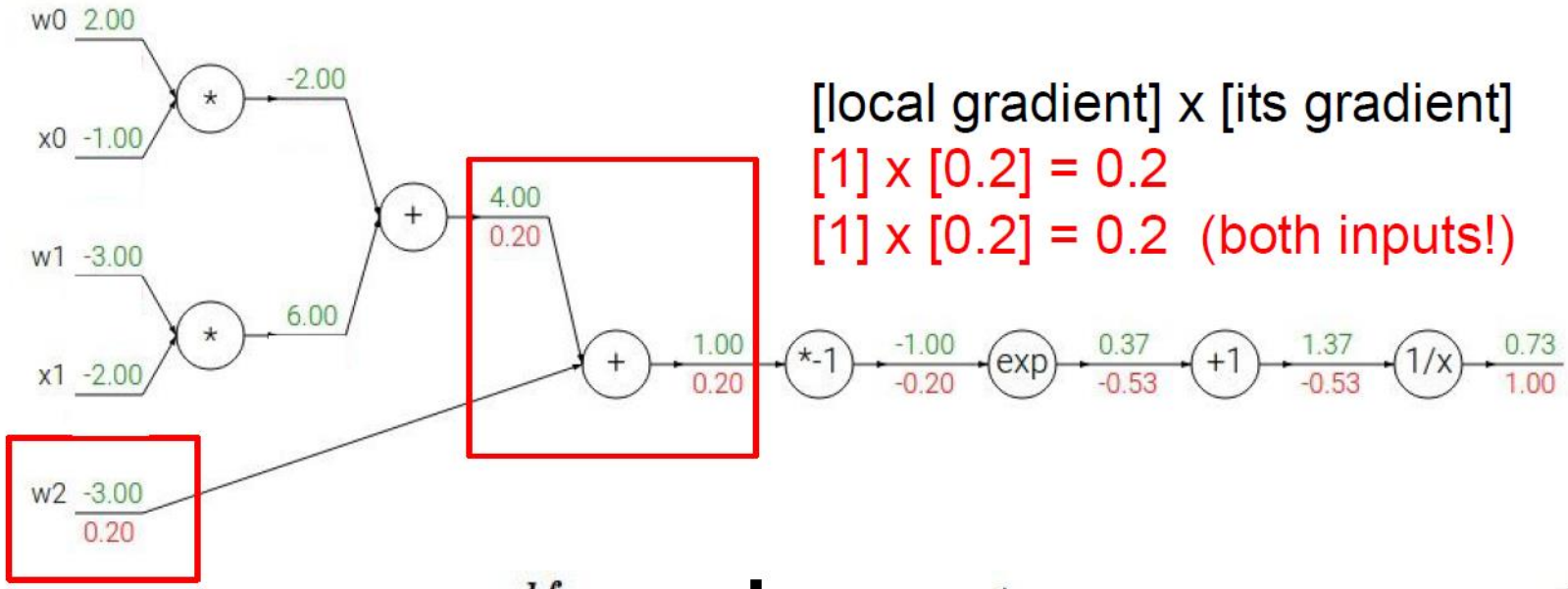
$$f_a(x) = ax \rightarrow \frac{df}{dx} = a$$

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

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

Another example:

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



[local gradient] x [its gradient]

[1] x [0.2] = 0.2

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

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

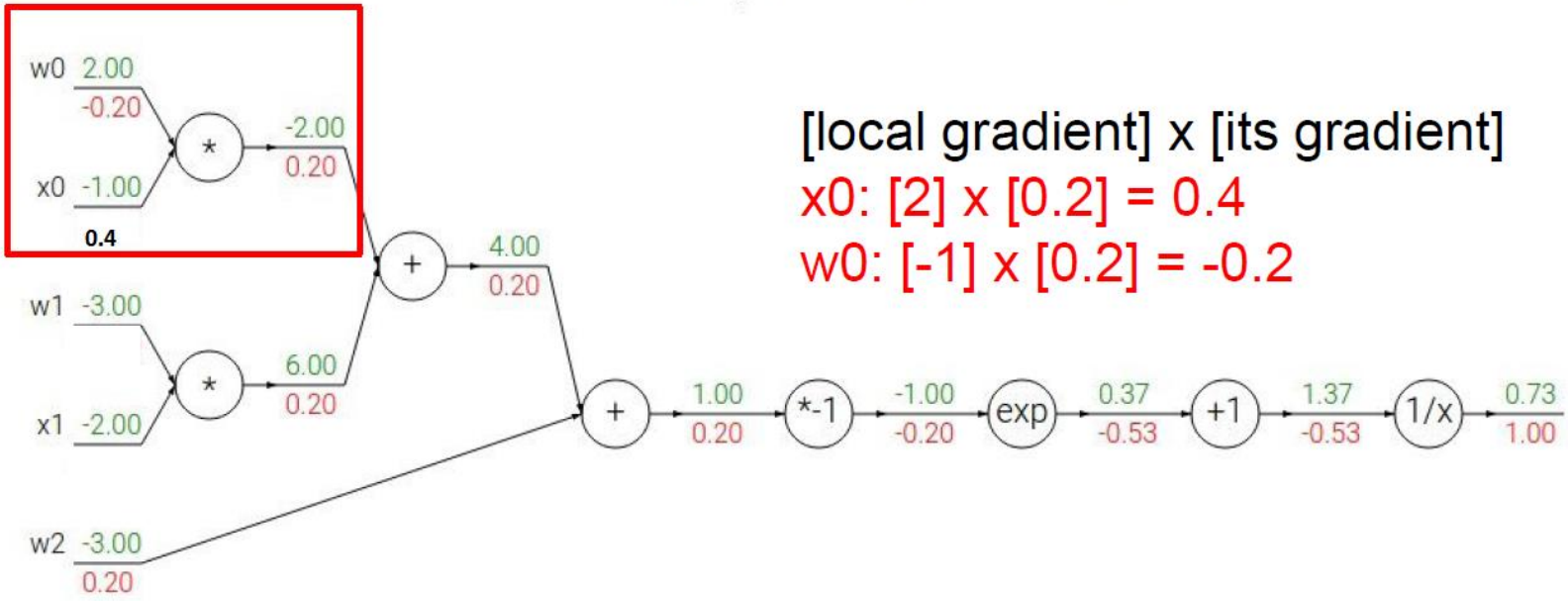
$$f_a(x) = ax \rightarrow \frac{df}{dx} = a$$

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

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

Another example:

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



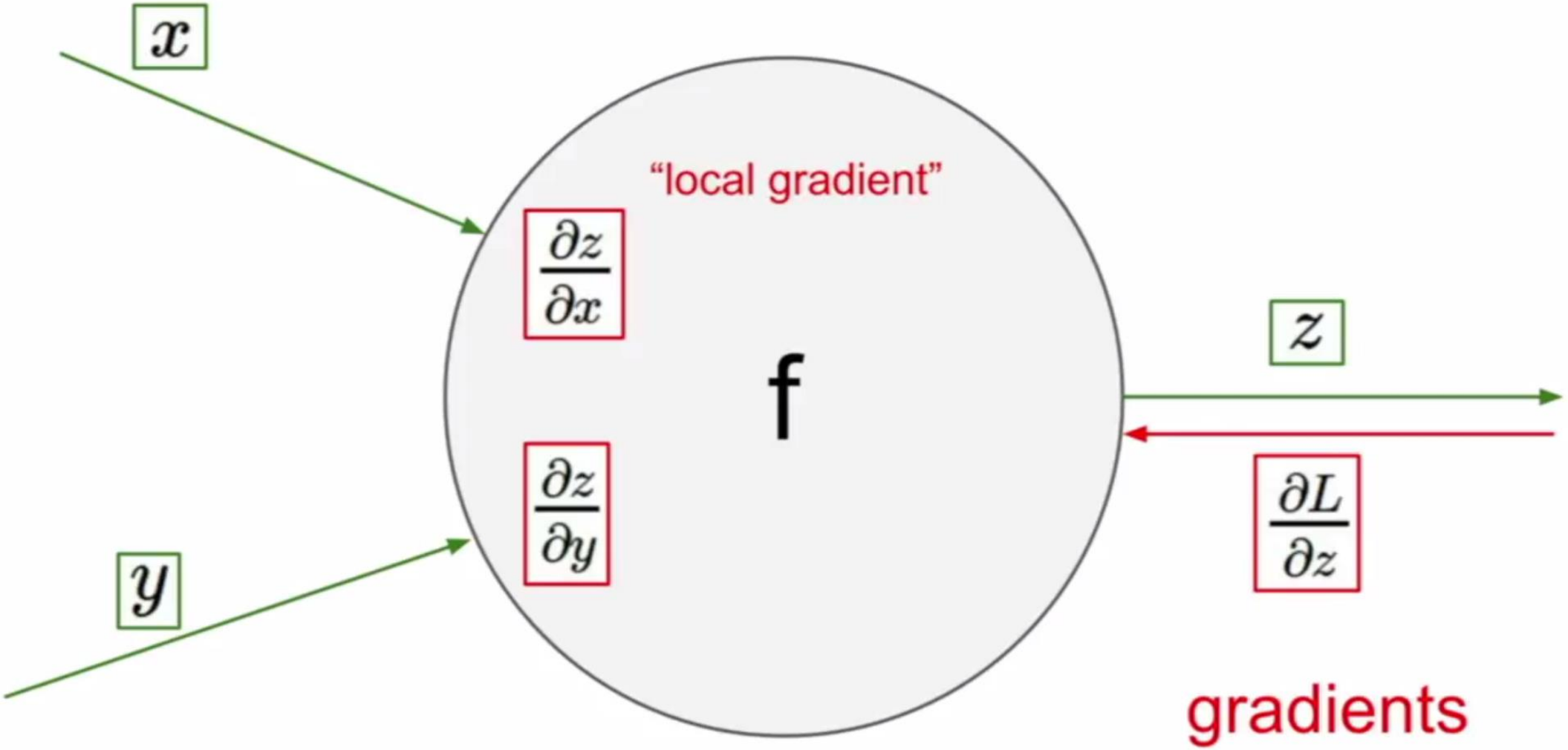
[local gradient] x [its gradient]

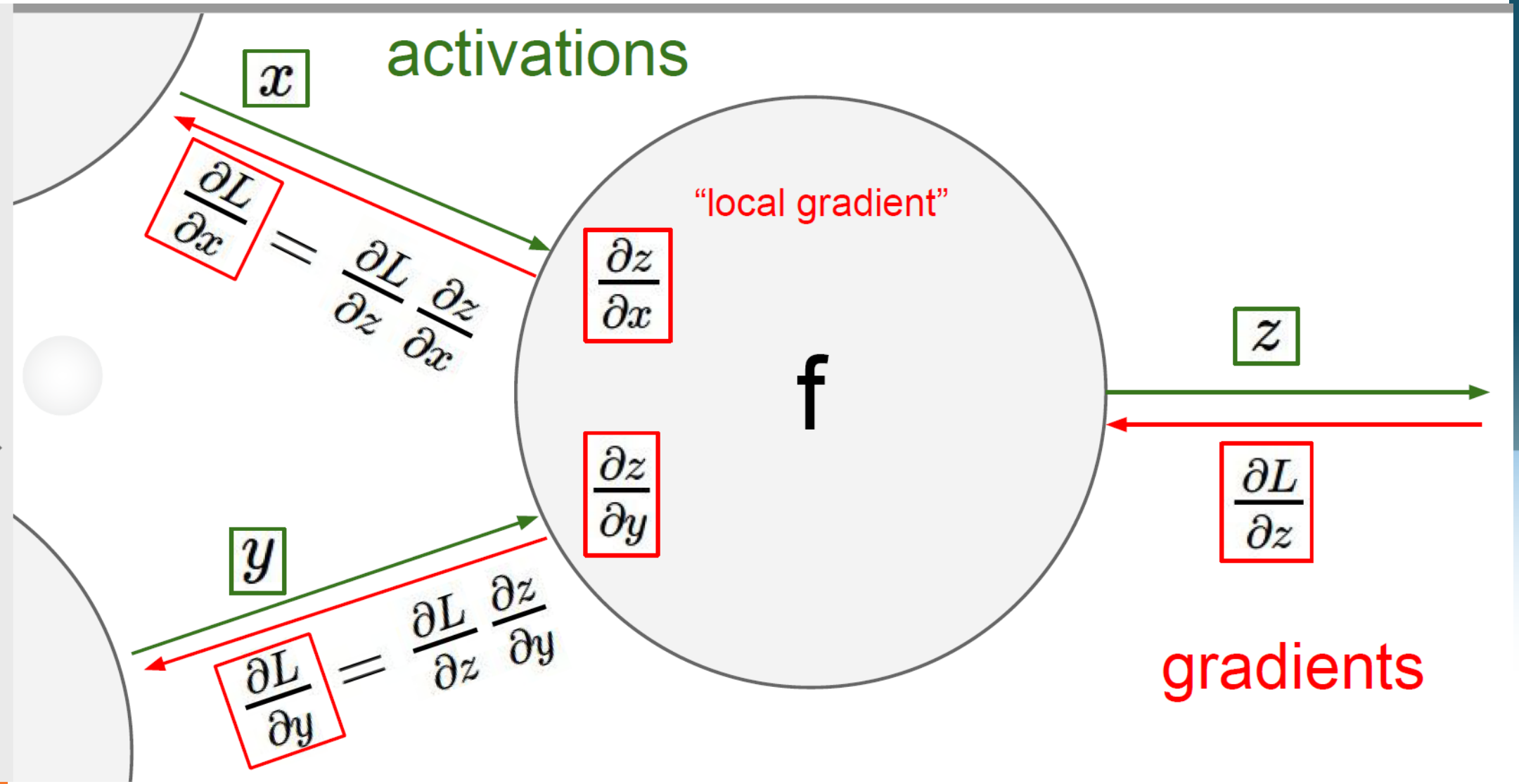
x0: [2] x [0.2] = 0.4

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

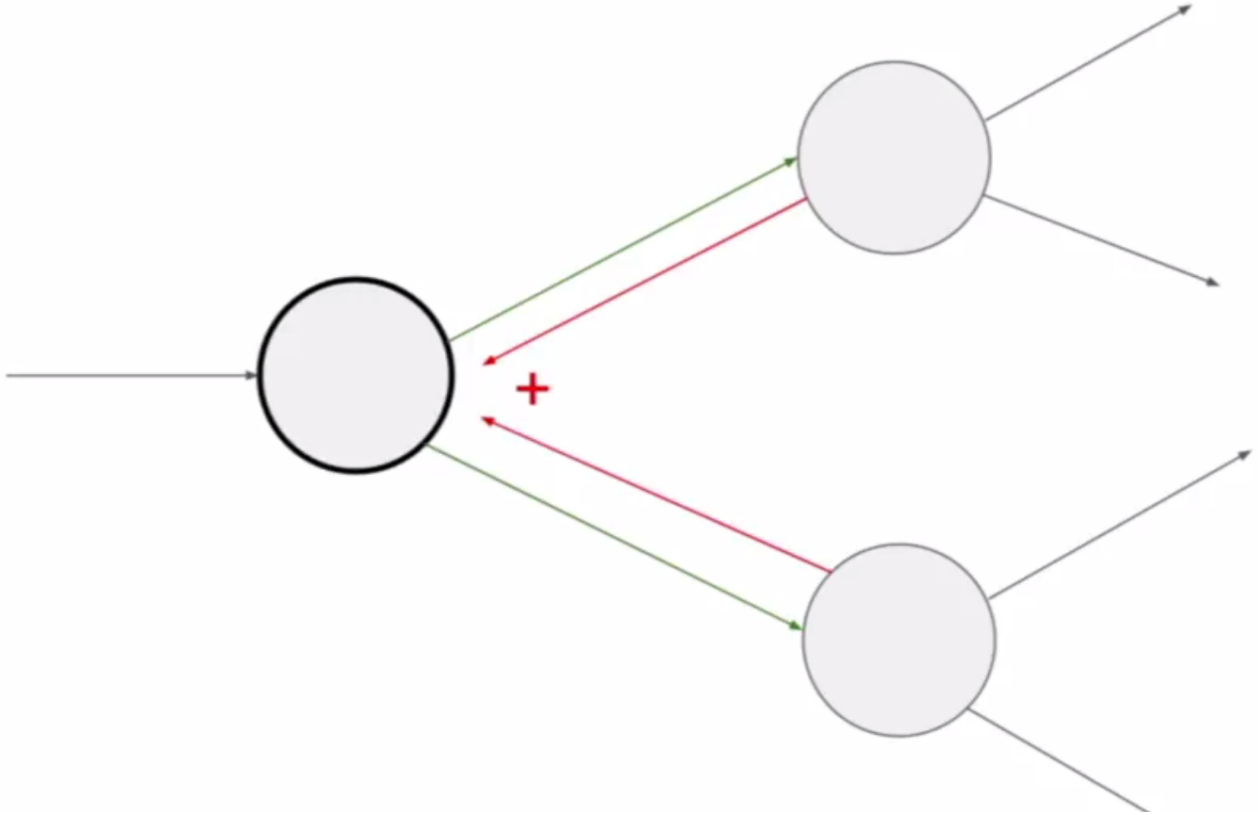
$$f(x) = e^x \rightarrow \frac{df}{dx} = e^x \qquad f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$$

$$f_a(x) = ax \rightarrow \frac{df}{dx} = a \qquad f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$$


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

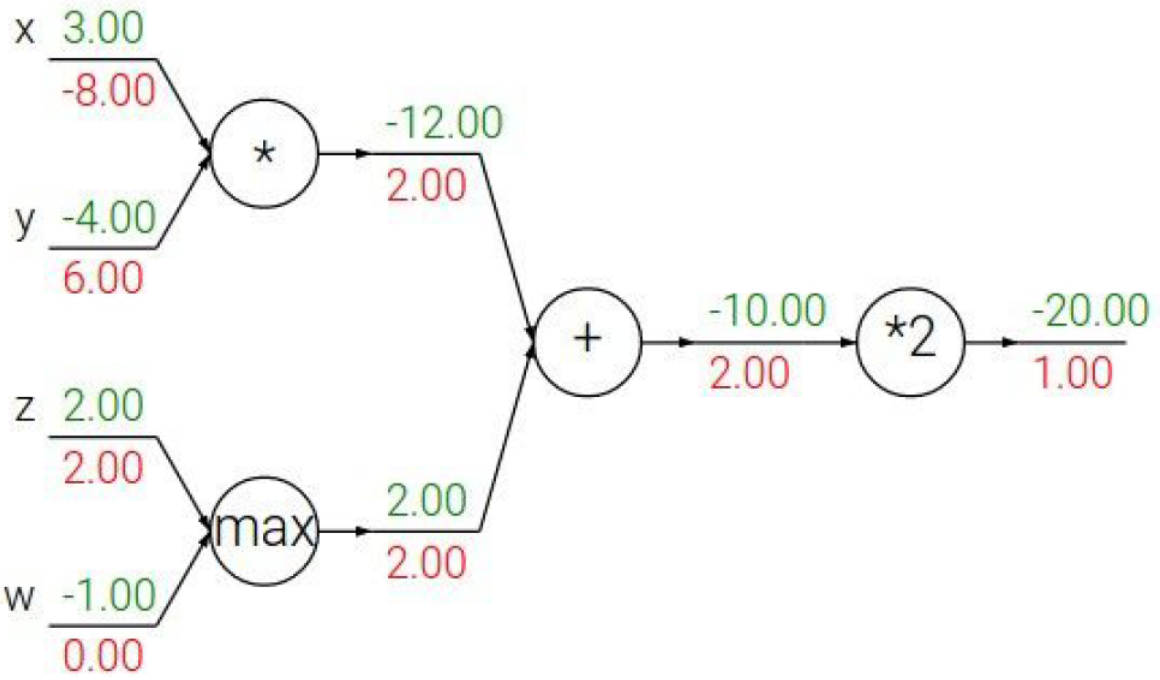

activations
x
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
"local gradient"
$\frac{\partial z}{\partial x}$
f
$\frac{\partial z}{\partial y}$
z
$\frac{\partial L}{\partial z}$
y
$\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z}\frac{\partial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

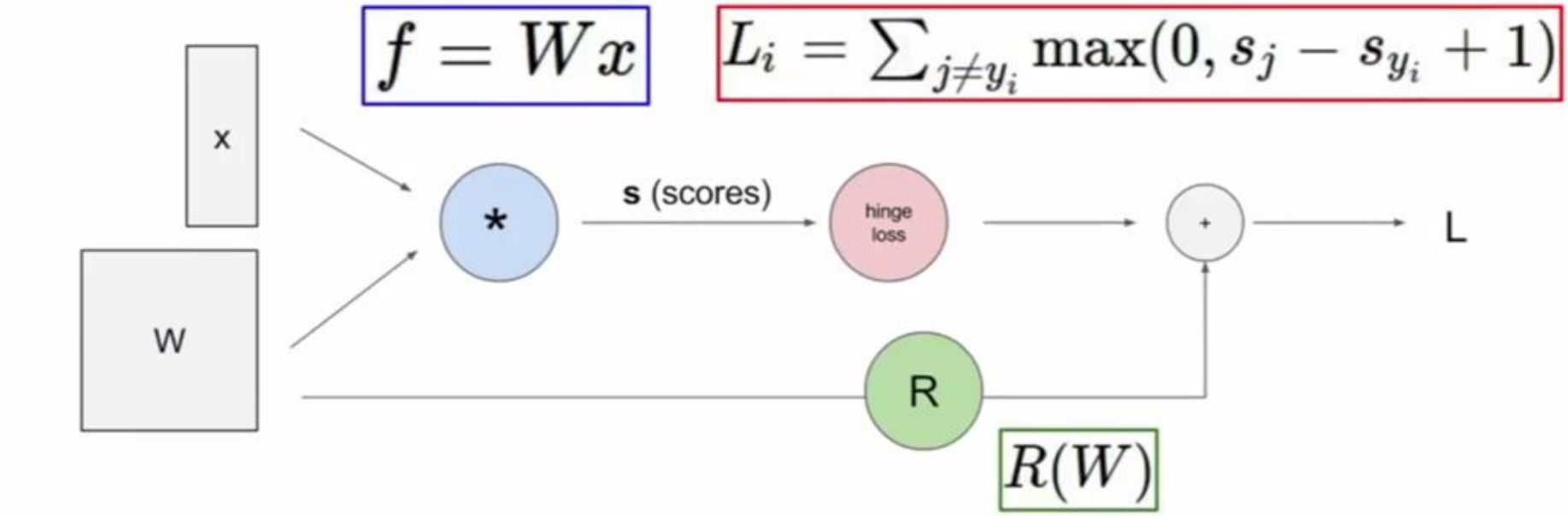
Gradients add at branches



Patterns in backward flow

add gate: gradient distributor
max gate: gradient router
mul gate: gradient… “switcher”?




$f = Wx$
$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$
x
W
*
s (scores)
hinge loss
+
L
R
$R(W)$

ENGINEERING AN MLP

Perceptron or MLP

- Ask the lead for performance requirements
- Start with a perceptron. If it meets, productionize it.
- If it does not meet the accuracy, move on to the next slide

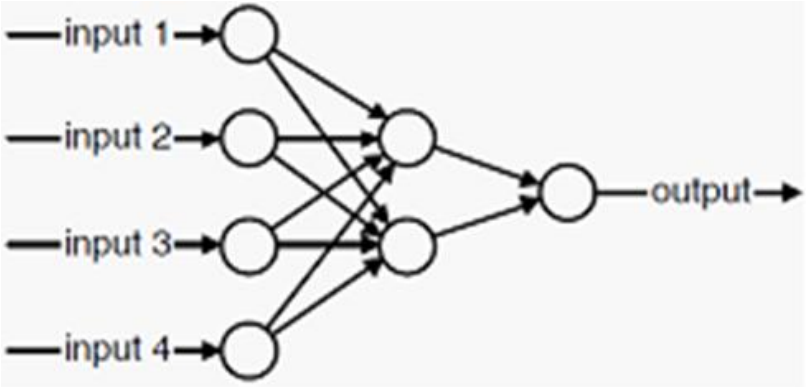
Major differences from architect's point of view for MLP

- Data requirements
- Explicability is gone so visualization strategies differ
- A few more hyper parameters get added

How much data?

- Number of weights * 100 is great. 20X is the least
- In fact, if you are restricted by available data, you build smaller neural networks based on the above thumb rule as explained below

Hidden nodes based on data



$$H * (I + O) + H + O$$

5 input features and 10 units in the hidden network and 1 output, then there are 71 weights in the network.

- The weights should be 1/100th of the amount of training data set
- A NN with 4 input, 2 hidden and 2 outputs require how much of data?

Hyper parameters: Layers

- For structured data, 1 or 2 hidden layers are good.
- So, if I have 20,000 records and 20 inputs for a regression
  - Training will be 15,000, so we can have only 150 to 750 weights
  - 20X10X1 or 20X10X5X1 can be tried
- Rest follows perceptron model discussed before

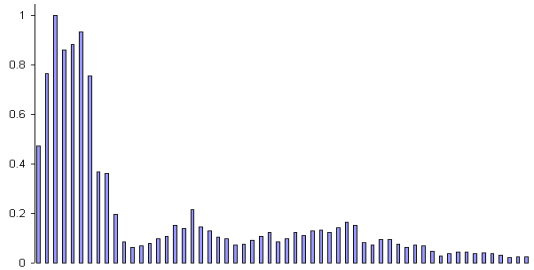
Visualization

- Find the average value for each input. We can think of this average value as the center of the test set.
- Measure the output of the network when all inputs are at their average value.
- Measure the output of the network when each input is modified, one at a time, to be at its minimum and maximum values
- Identify the most influential inputs
- Plot the variation as partial dependency plots

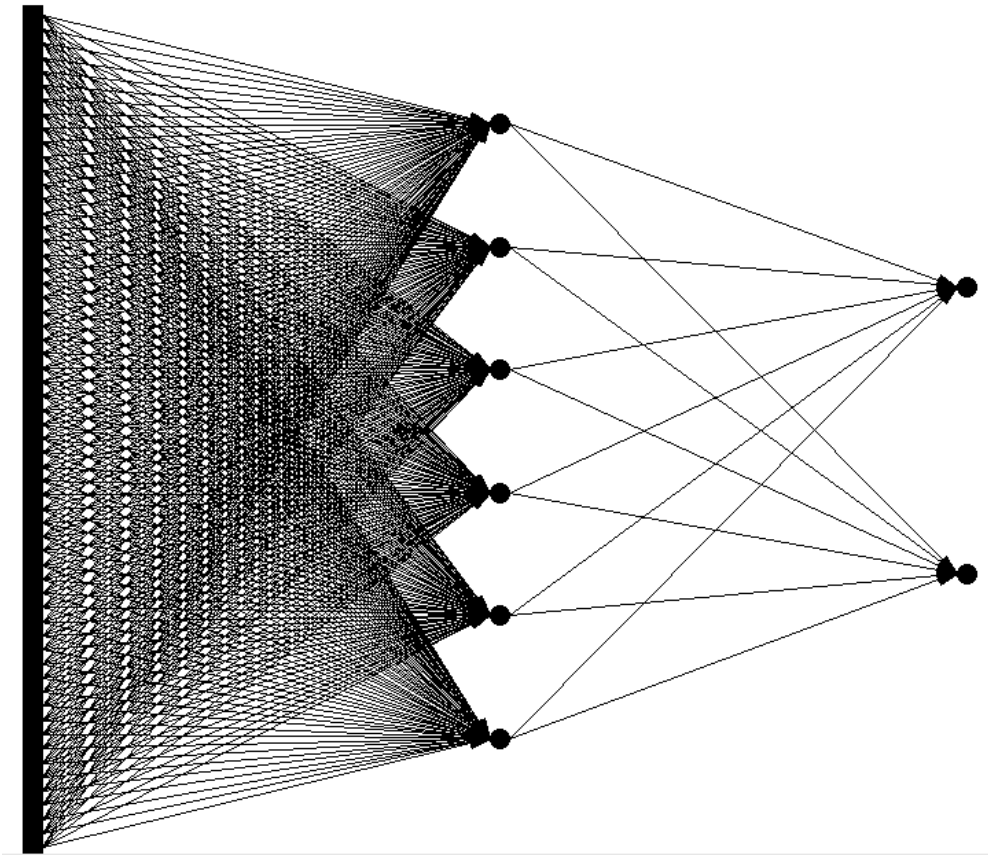
Example: Voice Recognition

- Task: Learn to discriminate between two different voices saying “Hello”

- Data
  - Sources
    - Steve Simpson
    - David Raubenheimer
  - Format
    - Frequency distribution (60 bins)
    - Analogy: cochlea

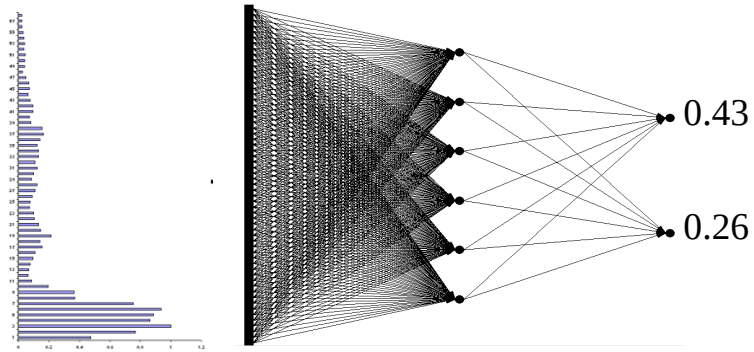


- Network architecture
  - Feed forward network
    - 60 input (one for each frequency bin)
    - 6 hidden
    - 2 output (0-1 for “Steve”, 1-0 for “David”)

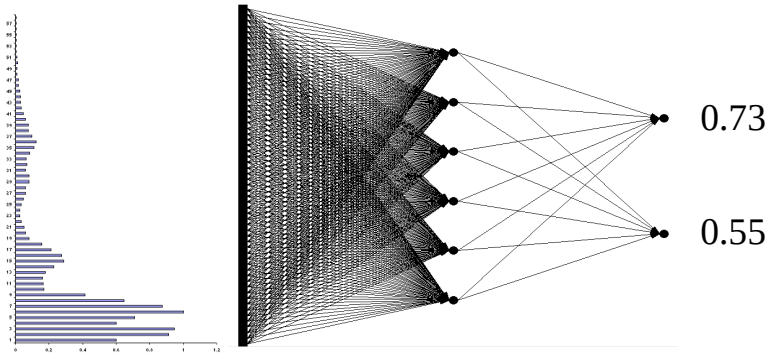


Untrained network

Steve

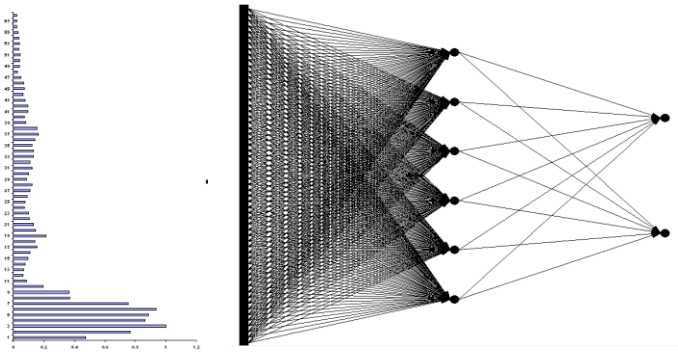


David

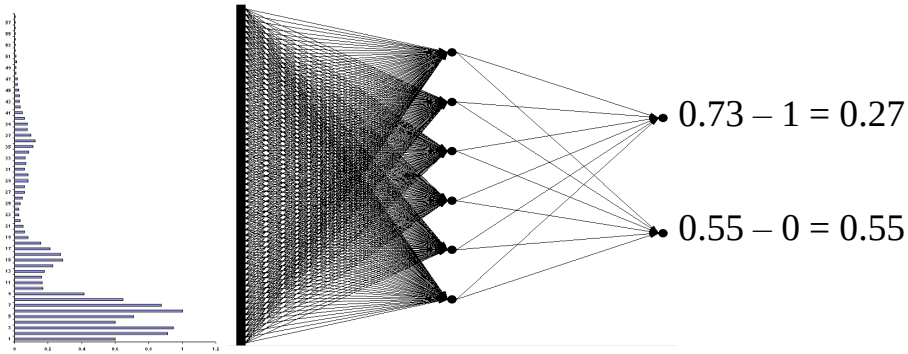


Calculate error

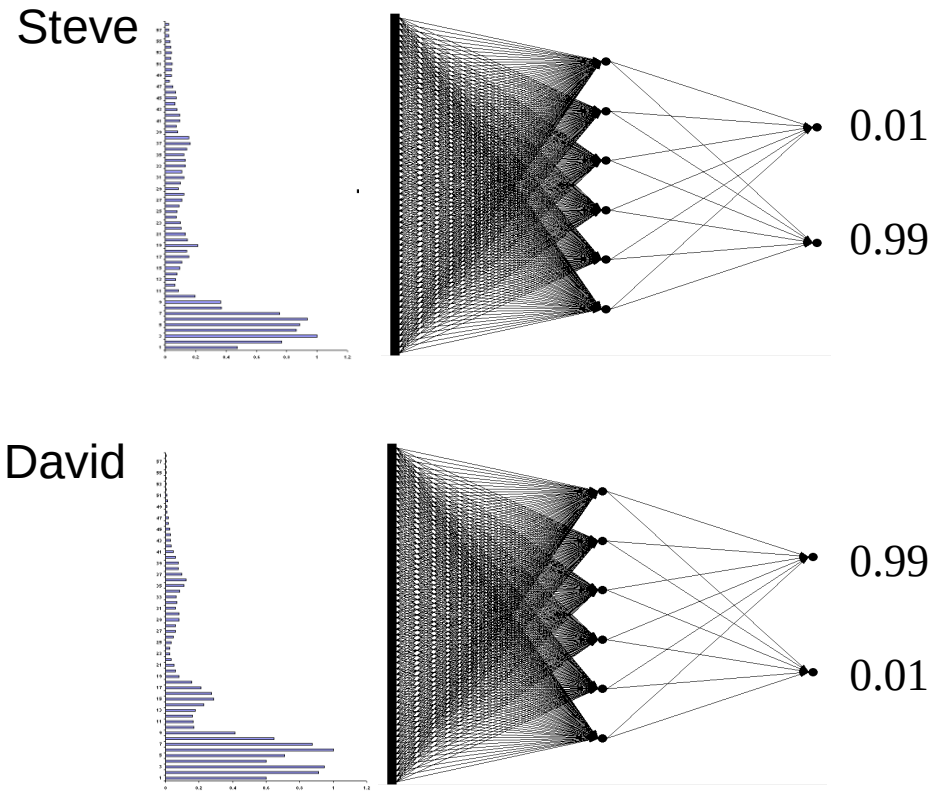
Steve

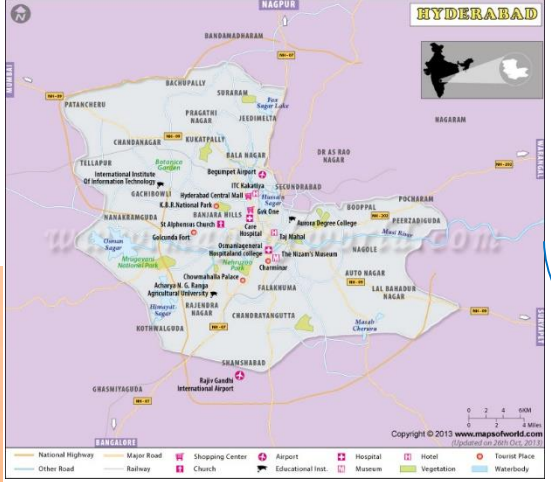


David



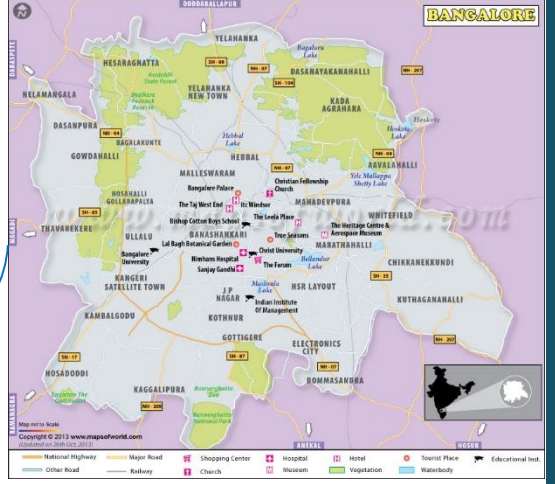
- Performance of trained network
  - Discrimination accuracy between known “Hello”s
    - 100%
  - Discrimination accuracy between new “Hello”’s
    - 100%





INSOFE

Inspire...Educate...Transform.



HYDERABAD

Office and Classrooms

Plot 63/A, Floors 1&2, Road # 13, Film Nagar,
Jubilee Hills, Hyderabad - 500 033
+91-9701685511 (Individuals)
+91-9618483483 (Corporates)

BENGALURU

Office

Incubex, #728, Grace Platina, 4th Floor, CMH Road,
Indira Nagar, 1st Stage, Bengaluru – 560038
+91-9502334561 (Individuals)
+91-9502799088 (Corporates)

Classroom

KnowledgeHut Solutions Pvt. Ltd., Reliable Plaza,
Jakkasandra Main Road, Teacher's Colony, 14th Main
Road, Sector – 5, HSR Layout, Bengaluru - 560102

Social Media

Web:	http://www.insofe.edu.in
Facebook:	https://www.facebook.com/insofe
Twitter:	https://twitter.com/Insofeedu
YouTube:	http://www.youtube.com/InsofeVideos
SlideShare:	http://www.slideshare.net/INSOFE
LinkedIn:	http://www.linkedin.com/company/international-school-of-engineering

This presentation may contain references to findings of various reports available in the public domain. INSOFE makes no representation as to their accuracy or that the organization subscribes to those findings.