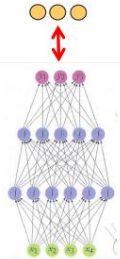
— Training: algorithm —

ÿ To operate a neural network, its parameters must first be determined; and to obtain the parameters, it must be trained: which training algorithm should be adopted? What are the main options regarding the network's hyperparameters and their configuration?

Þ Training



q Pârametros (recap)

ÿ weights of connections between neurons ÿ determined by training

q **Training**

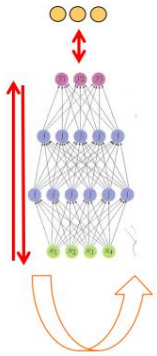
ÿ **The goal** is **to find the function** that maps the examples into respective desired labels

ÿ that is, find the weights that support such a function

ÿ **we look** for weights that **minimize error** or loss

ÿ **training** is an **optimization process**

Training algorithm



q 1. Forward pass
ÿ make prediction

q 2. Estimate error/loss ÿ compare with desired value

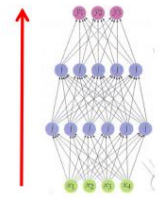
q 3. Backward pass ÿ adjust the weights

q Repeat the above steps until the parameter values stabilize (without over-adjustment: see below)

Þ Step 1: Forward pass para a frente ("forward pass")

q Only in the initial step:
Assign initial weights to connections between neurons (possibly randomly)

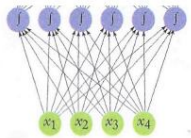
q **Compute the prediction over the network**
Go from the input layer to the output layer, taking an example training (or a batch of them) as input



Network Computing (recap) 1/2

q Entrance

ÿ vector ***x*** = [x1, ..., xn]



q Weights

ÿ matrix ***W*** with *n* neurons of the source layer in the rows and *m* neurons of the target layer in the columns

layer origin

[w11, ..., w1m]

[....]

[wn1, ..., wnm]

layer target

ÿ W[i,j] is the connection weight from source neuron i to target neuron j

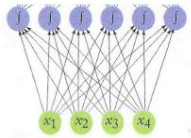
5

Network Computing (recap) 2/2

q ***h*** = *g* (x ***W*** + ***b***) (fully connected layer)

ÿ ***h*** is output vector

each component with the output of each neuron



ÿ *g* is nonlinear activation function

e.g. ReLU – Rectifier Linear Unit, *g(x)* = max(0,x)

ÿ ***x*** *is input vector and* ***W*** *is transition matrix*

ÿ ***x W*** =

[x1, ..., xn].[w11, ..., w1m] = [x1.w11+...+xn.wn1, ..., x1.w1m+ ...+xn.wnm]

[....]

[wn1, ..., wnm]

ÿ ***b*** *is bias term*

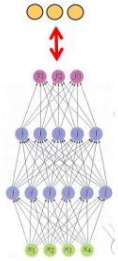
6

X is an input vector and W is a transition matrix

Þ Step 2: Estimate the loss

q **Loss function L(y^,y)**

ÿ error/loss occurs when the prediction **y^** is different from the correct output **y**



ÿ function maps the two vectors to a scalar

Which two vectors?

ÿ The goal of training is to minimize loss: less is better

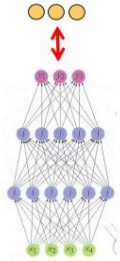
7

Loss

q Examples of commonly used functions

ÿ For binary classification problems

- Loss of binary articulation
- Binary cross-entropy
- ...

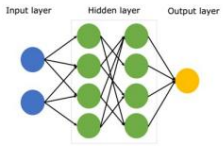


ÿ For multi-class classification problems

- Multi-class articulation loss
- Categorical cross-entropy
- ...

8

Binary classification 1/2



q Loss of binary articulation (“hinge”)

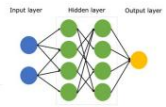
ÿ also known as marginal loss

ÿ binary classification: prediction y^ is a scalar and **correct outputs are -1 or +1**

ÿ correct classification: y^ and y have the same sign ÿ
zero loss when prediction y^ and output y have the same sign and |y^|>=1

ÿ L(y^,y) = max(0, 1-yy^)

Binary classification 2/2

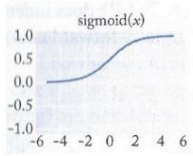


q Binary cross-entropy

ÿ also known as logistic loss ÿ **correct outputs are 0 or 1**

ÿ prediction y^ in [0,1] after y^ is passed through the sigmoid function ÿ correct classification: y^<0.5 and y=0, or y^>=0.5 and y=1

ÿ L(y^,y) = - log y^ = - if y = 1
log (1 – y^) if y = 0



Intermezzo: softmax function (recap)

q *softmax(x[i])* = ex[i] / ÿj ex[j]

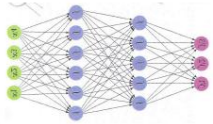
ÿ after application to the components of the vector **x :**

ÿ each component with value in [0,1]
ÿ the sum of all components results in 1

ÿ component values interpretable as a probability distribution

ÿ normalizing function

Multi-class classification 1/2



q Loss of multi-class hinge

ÿ correct output will be a one-hot **y** vector : **components a 0 except one** ÿ
correct class: t = argmax_i y[i]

ÿ network output is a vector **y^** = y^[1], **…,** y^[m] ÿ
predicted class: k = argmax_i y^[i]

ÿ **L(y^,y)** = max (0, 1 - (y^[t] - y^[k])) com t !=k

Multi-class classification 2/2

Categorical cross-entropy

ÿ also known as negative log likelihood ÿ correct output will be a vector **y** = y[1], **…,** y[m] representing the multinomial distribution over the labels

item may belong to different classes with some degree of certainty

ÿ network output is a vector **y^** = y^[1], **…,** y^[m] after being transformed by the softmax function (each value in [0,1] and all added together results in 1)

ÿ **L(y^,y)** = – ÿi y[i] log (y^[i])

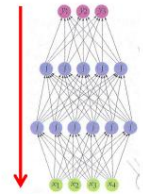
Þ Step 3: Backward pass

(“backward pass”)

q Back- **propagation**

ÿ compute the gradient of each parameter relative to the loss that was estimated

ÿ move the parameters in the opposite direction to the sign of the gradient



interlude: networks calculate composite functions (recap)

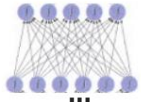
q Multi-layer neural network

ÿ is a composite function that involves the successive

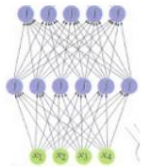
ÿ multiplication of vectors by matrices,

ÿ passing vectors through non-linear functions,

ÿ and sum with bias vectors:



ÿ ***y^*** = g(... g(g(xW1 + b1)W2 + b2)W3 + **b3) ...)Wn** + **bn** ÿ ***y^*** = ***r(x)***



ÿ **which, during training, is composed with the function of loss:** *L* ***(r(x), c(x))***

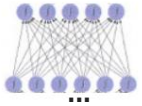
where ***c(x)*** = ***y*** the correct classification for each ***x***

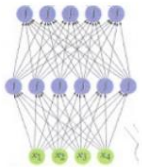
Optimization based on 1/4 gradient

q Stochastic gradient descent

(“stochastic gradient descent – SGD”)

ÿ 1. compute the gradient g of each parameter with respect to the loss





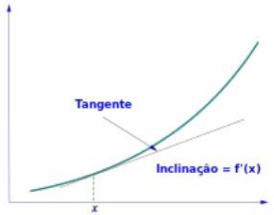
Interlude: Gradient

q Derivative of a function

ÿ the derivative function f' of a function f measures the sensitivity of your output to change your entry

ÿ the derivative function of a unary function in your entry is the slope of the tangent line to the graph at that point

ÿ the sign of the derivative function, negative or positive, indicates whether the object function is decreasing or increasing.



Barcoso

9a.17

17

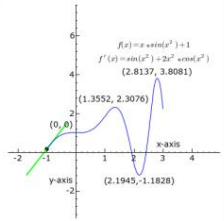
Gradient-based optimization 2/4

q Objective: minimize the loss function

ÿ the loss function measures the error, i.e. the distance between the prediction and the correct output

ÿ the goal of training is to find the weights that minimize the loss, i.e. that make the distance between the prediction and the correct output zero

ÿ that is, the objective is to adjust the weights of such so that the output of the (loss) function calculated by the network is zero



Barcoso

9a.18

18

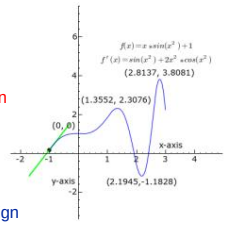
Gradient-based optimization 3/4

q Objective: "counteract the sign of the gradient"

ÿ that is, the objective is to adjust the network weights in the opposite direction to the (de)growth of the function of loss

ÿ that is, in terms of gradient, the goal is change each weight in the opposite direction to the sign of its gradient

ÿ that is, if the gradient is negative the (loss) function is decreasing, an increase in weight will contribute to increasing the prediction value and thus making the loss continue to decrease and therefore approaching zero (and vice versa)



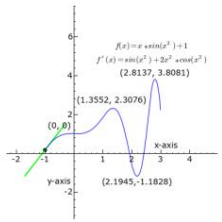
Barcoso

9a.19

19

q **N.B.**

q This graph was used in the previous slides only to help form intuitions about the relevance of the derivative to the optimization process; does not represent "the" loss function curve, nor any curve especially relevant to deep learning



Barcoso

9a.20

20

Optimization based on 4/4 gradient

q Stochastic gradient descent

("stochastic gradient descent – SGD")

ÿ 1. compute the gradient g of each parameter with respect to the loss

ÿ 2. move each parameter p in the opposite direction to its gradient:
replace p with what results from its adjustment by the gradient
after this is weighted by a learning rate t:
p ç p – t.g

q Variants [advanced topics]

ÿ Gradients are accumulated: SGD+Momentum, Nesterov Momentum

ÿ Adaptive learning rate: AdaGrad, AdaDelta, RMSProp, Adam 9a.21

Bracco

21

- Conclusion -

q Index

ÿ Training algorithm

ÿ Forward pass computation

ÿ Loss functions

ÿ Backward pass computation

What hyperparameters should I configure a model with (to train it)?

Bracco

9a.22

22