

GIANLUCA CARUCCI

# deeplearning from zero to hero

---

»AI&ML CONF - MICROSOFT HOUSE  
27 SETTEMBRE 2019



[fbk.com/caruccigianluca](https://www.facebook.com/caruccigianluca)



[gianlucacarucci5](https://www.instagram.com/gianlucacarucci5)



[www.linkedin.com/in/rucka](https://www.linkedin.com/in/rucka)

# Neural Networks for Machine Learning

## derivatives of a logistic neuron

**About this course:** Learn about artificial neural networks and how they're being used for machine learning, as applied to speech and object recognition, image segmentation, modeling language and human motion, etc. We'll emphasize both the basic algorithms and the practical tricks needed to get

“good” results.

▼ More

**Created by:** University of Toronto



**Taught by:** Geoffrey Hinton, Professor  
Department of Computer Science

$$\frac{d}{dz} \left( \frac{e^{-z}}{(1+e^{-z})^2} \right) = \frac{1}{(1+e^{-z})} \left( \frac{e^{-z}}{(1+e^{-z})} \right) - \frac{e^{-z}(-1)}{(1+e^{-z})^2} =$$

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



not a good start

A black and white photograph of a woman with long, light-colored hair, seen from behind and slightly to the side. She is wearing a dark, sleeveless top and a wide-brimmed hat. Her head is bowed, and she appears to be crying or deeply distressed. The background is blurred, showing what might be foliage or a garden.

is there another way?

W A R N I N G

S P O I L E R

A L E R T





# Gianluca Carucci

*Software Engineer*

"Now is the time to examine how we build AI responsibly and avoid a race to the bottom. This requires both the private and public sectors to take action."

- Satya Nadella





COMMUNITY

communities

Ugidotnet | Scala Milano Meetup | Marketers



look mum, no code!

STARTED

PROGRAMMER'S GUIDE

TUTORIALS

PERFORMANCE

MOBILE

## How to Retrain Inception's Final Layer for New Categories

# transfer learning<sup>1</sup>

Modern object recognition models have millions of parameters and can take weeks to fully train.

Transfer learning is a technique that shortcuts a lot of this work by taking a fully-trained model for a set of categories like ImageNet, and retrains from the existing weights for new classes. In this example we'll be retraining the final layer from scratch, while leaving all the others untouched. For more information on the approach you can see this paper on Decaf.

Though it's not as good as a full training run, this is surprisingly effective for many applications, and can be run in as little as thirty minutes on a laptop, without requiring a GPU. This tutorial will show you how to run the example script on your own images, and will explain some of the options you have to help control the training process.

[How to Retrain Inception's Final Layer for New Categories](#)

[TensorFlow for Poets](#)

Note: A version of this tutorial is also available as a [codelab](#).

Before you start, you must install tensorflow.

Contents

Training on Flower

Bottlenecks

Training

Visualizing the Retraining with TensorBoard

Using the Retrained Model

Training on Your Own Categories

Creating a Set of Training Images

Training Steps

Distortions

Hyper-parameters

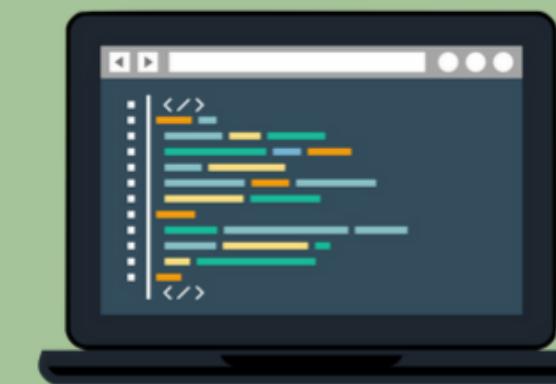
Training, Validation

Testing Sets

Other Model Architectures



slides



code



environment



# the simpsons detector

```
root@cc5a496bf40e:/code/simpson_retraining# ./retrain.sh
INFO:tensorflow:Looking for images in 'marge'
INFO:tensorflow:Looking for images in 'homer'
INFO:tensorflow:Looking for images in 'bart'
INFO:tensorflow:Looking for images in 'lisa'
...
INFO:tensorflow:2018-02-25 09:26:27.689515: Step 0: Train accuracy = 53.0%
INFO:tensorflow:2018-02-25 09:26:27.689750: Step 0: Cross entropy = 1.367082
INFO:tensorflow:2018-02-25 09:26:28.583996: Step 0: Validation accuracy = 33.0% (N=100)
...
INFO:tensorflow:2018-02-25 09:26:49.359180: Step 30: Train accuracy = 65.0%
INFO:tensorflow:2018-02-25 09:26:49.359404: Step 30: Cross entropy = 1.172318
INFO:tensorflow:2018-02-25 09:26:49.871993: Step 30: Validation accuracy = 69.0% (N=100)
...
INFO:tensorflow:2018-02-25 09:35:37.061001: Step 1000: Train accuracy = 95.0%
INFO:tensorflow:2018-02-25 09:35:37.061221: Step 1000: Cross entropy = 0.311776
INFO:tensorflow:2018-02-25 09:35:37.600067: Step 1000: Validation accuracy = 87.0% (N=100)
...
INFO:tensorflow:2018-02-25 09:44:03.139348: Step 2000: Train accuracy = 95.0%
INFO:tensorflow:2018-02-25 09:44:03.139593: Step 2000: Cross entropy = 0.286085
INFO:tensorflow:2018-02-25 09:44:03.538799: Step 2000: Validation accuracy = 83.0% (N=100)
...
INFO:tensorflow:2018-02-25 10:00:39.674607: Step 3999: Train accuracy = 97.0%
INFO:tensorflow:2018-02-25 10:00:39.674859: Step 3999: Cross entropy = 0.175444
INFO:tensorflow:2018-02-25 10:00:40.073648: Step 3999: Validation accuracy = 90.0% (N=100)
INFO:tensorflow:Final test accuracy = 90.4% (N=261)
```

```
root:/code/simpson_retraining# ./evaluate.sh /data/simpson/test_set/0.jpg^
bart 0.63594097
marge 0.3191907
lisa 0.034744244
homer 0.010124116
```

# the simpsons detector

## summary

```
root@cc5a496bf40e:/code/simpson_retraining# ./retrain.sh
INFO:tensorflow:2018-02-25 09:26:27.689515: Step 0: Train accuracy = 53.0%
INFO:tensorflow:2018-02-25 09:26:27.689750: Step 0: Cross entropy = 1.367082
INFO:tensorflow:2018-02-25 09:26:28.583996: Step 0: Validation accuracy = 33.0% (N=100)
...
INFO:tensorflow:2018-02-25 10:00:39.674607: Step 3999: Train accuracy = 97.0%
INFO:tensorflow:2018-02-25 10:00:39.674859: Step 3999: Cross entropy = 0.175444
INFO:tensorflow:2018-02-25 10:00:40.073648: Step 3999: Validation accuracy = 90.0% (N=100)
INFO:tensorflow:Final test accuracy = 90.4% (N=261)
```

```
root:/code/simpson_retraining# ./evaluate.sh /data/simpson/test_set/0.jpg`  
bart 0.63594097  
marge 0.3191907  
lisa 0.034744244  
homer 0.010124116
```



# common scenarios

6. Se disservos que tensas, mentiras, e que praticas a tua  
7. Mas se andarmos na luz, conto de na luz está, ou  
comitâmos um com os outros, e o santo de Jesus Cristo, seu  
Filho, nos purifica de todo pecado.

8. Se disservos que não tensas pecado nenhuns, engana  
nos-nos a nós mesmos, e não há verdade em nós.

9. Se confessarmos os nossos pecados, ele é fiel e justo para  
nos perdoar os pecados, e nos purifica de toda mentira.

10. Se disservos que não pecamos, fazemo-lo mentira, e a  
sua palavra não está em nós.

**2** Meus filhos, estas coisas vos escrevo para que não pe  
quem. Se, porém, alguém pecar, temos um Advogado para  
com o Pai, Jesus Cristo, o justo.

1. Ele é a propriação pelos nossos pecados, e não somente  
pelos nossos, mas também pelos de todo o mundo.

3. E visto sabemos que o conhecemos, se guardarmos os seu  
mandamentos.

4. Aquela que dir: Eu o conheço, e não guarda os seus man  
damentos, é mentiroso, e onde não está a verdade.

5. Mas qualquera que guarda a sua palavra, o amor de Deus  
não tem se verdadeiramente aperfeiçoado. E visto conhecer  
mos que estamos nele.

6. Aquela que diz que está nele, também deve andar como  
de andou.

7. Amados, não vos escrevo mandamento novo, mas um  
mandamento antigo, que desde o princípio investe. Este man  
damento antigo é a palavra que ouvistes.

8. Vós, que estais em mim, porque as trevas vão passar,  
vou vos mandar uma coisa nova.

10. Aquela que não tem a lei, não tem nenhuma trans  
gressão da lei.

11. Mas aquele que comete pecado, de modo que  
nas trevas permanece, e não nas luzes, é aquele que  
ram o mundo.

12. Pondo isto em vista, que a lei é de trans  
gressão.

13. desde que o mundo existiu.

14. Em vista disso, vos escrevo, e a sua palavra é a  
principia. Porque a palavra de Deus está em vós, e não no mundo.

15. Não amea o mundo, e não deixes que o mundo  
ama o mundo, o mundo é do diabo.

16. Por todo o que há de mau, pertence ao mundo.  
a concupiscencia dos olhos e a concupiscencia do  
mundo.

17. Ora, o mundo passa, e a sua concupis  
cencia que há a vontade de Deus permanece para  
sempre.

18. Filhos, está é a ultima hora, e como o  
mundo astucioso, na ultima hora tem surgido. Pe  
remos que é a ultima hora.

Models and examples built with TensorFlow

1,819 commits

12 branches

2 releases

304 contributors

Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

a-dai Merge pull request #3414 from a-dai/master ...	Latest commit d903d5e 13 hours ago
official Merge pull request #3400 from asimshankar/mnist-eager ...	6 days ago
research Merge pull request #3404 from m-dalton/... ...	13 hours ago
samples Fix one more Distributioned_Tensorflow ...	21 hours ago
tutorials six.moves works with python2/3	14 days ago
.gitignore Added PyCharm to .gitignore	6 months ago
.gitmodules Move the research models into a research subfolder (#2430)	5 months ago
AUTHORS Spatial Transformer model	2 years ago
CODEOWNERS Add a-dai to CODEOWNERS for adversarial_text.	6 days ago
CONTRIBUTING.md Fixing small typo	8 months ago
ISSUE_TEMPLATE.md Update ISSUE_TEMPLATE.md	15 days ago
LICENSE Update LICENSE	2 years ago
README.md Update Readme to link to master research models (#3318)	17 days ago
WORKSPACE Consolidate privacy/ and differential_privacy/.	a year ago

README.md

## TensorFlow Models

This repository contains a number of different models implemented in TensorFlow:

The official models are a collection of example models that use TensorFlow's high-level APIs. They are intended to be well-maintained, tested, and kept up to date with the latest stable TensorFlow API. They should also be reasonably optimized for fast performance while still being easy to read. We especially recommend newer TensorFlow users to start here.

## TensorFlow Models<sup>2</sup>

The research models are a large collection of models implemented in TensorFlow by researchers. They are not officially supported or available in release branches; it is up to the individual researchers to maintain the models and/or provide support on issues and pull requests.

The samples folder contains code snippets and smaller models that demonstrate features of

- audioset: Models and supporting code for use with AudioSet.
- autoencoder: various autoencoders.
- brain\_coder: Program synthesis with reinforcement learning.
- cognitive\_mapping\_and\_planning: implementation of a spatial memory based mapping and planning architecture for visual navigation.
- compression: compressing and decompressing images using a pre-trained Residual GRU network.
- delf: deep local features for image matching and retrieval.
- differential\_privacy: privacy-preserving student models from multiple teachers.
- domain\_adaptation: domain separation networks.
- gan: generative adversarial networks.
- adversarial\_text: neural network for generating text via a variational autoencoder.
- image\_caption: see convolutional networks for computer vision.
- learning\_to\_remember\_rare\_events: a large-scale life-long memory module for use in deep learning.
- Ifads: sequential variational autoencoder for analyzing neuroscience data.
- lm\_1b: language modeling on the one billion word benchmark.
- namigner: recognize and generate names.
- neural\_gpu: highly parallel neural computer.
- neural\_programmer: neural network augmented with logic and mathematic operations.
- next\_frame\_prediction: probabilistic future frame synthesis via cross convolutional networks.
- object\_detection: localizing and identifying multiple objects in a single image.
- pcl\_rl: code for several reinforcement learning algorithms, including Path Consistency Learning.
- ptn: perspective transformer nets for 3D object reconstruction.
- qa\_kg: module networks for question answering on knowledge graphs.
- real\_nvp: density estimation using real-valued non-volume preserving (real NVP) transformations.
- rebar: low-variance, unbiased gradient estimates for discrete latent variable models.
- resnet: deep and wide residual networks.
- skip\_thoughts: recurrent neural network sentence-to-vector encoder.
- slim: image classification models in TF-Slim.
- street: identify the name of a street (in France) from an image using a Deep RNN.
- swivel: the Swivel algorithm for generating word embeddings.
- syntaxnet: neural models of natural language syntax.
- tcn: Self-supervised representation learning from multi-view video.

common solutions  
ready to go models<sup>2</sup>

Inserisci algoritmi intelligenti nelle tue app, nei siti Web e nei bot per vedere, ascoltare, parlare, comprendere e interpretare le esigenze dei tuoi utenti tramite i metodi di comunicazione naturali. Trasforma subito il tuo business con l'intelligenza artificiale.

[Prova Servizi cognitivi gratuitamente >](#)

Esplora i servizi cognitivi: [Directory](#) [Prezzi](#) [Documentazione](#)

Annuncio

Vinci premi straordinari creando il tuo prossimo progetto con Servizi cognitivi. >

# common solutions

Usa l'intelligenza artificiale per risolvere problemi di business



Decisione

Crea app che espongono raccomandazioni per un processo decisionale informato ed efficiente.

## Cloud Services

Riconosci, identifica, aggiungi sottotitoli, indicizza e modera immagini, video e contenuti di tipo input penna.



Voce

Converti il parlato in testo e il testo in parlato simile al linguaggio naturale. Traduci da una lingua a un'altra e consenti la verifica voce e il riconoscimento del parlante.



Ricerca

Aggiungi le API di ricerca Bing alle tue app e sfrutta la possibilità di esaminare miliardi di pagine Web, immagini, video e notizie con una singola chiamata all'API.

## Object Detection

- + Face Detection



Face

common solutions  
ready to go libraries or tools<sup>3</sup>

<sup>3</sup> - [OpenCV-Python](#)

- [Spacy](#)

- [gensim](#)



## Get things done

spaCy is designed to help you do real work — to build real products, or gather real insights. The library respects your time, and tries to avoid wasting it. It's easy to install, and its API is simple and productive. We like to think of spaCy as the Ruby on Rails of Natural Language Processing.

### Similarities

• Return file on disk.  
(`corpus.mm`)

• Working with 200 dimensions.  
(`spc(0:n=200)`)

• Count space and index (0..  
`ity[1:n]`|`another\_corpus`))

• Unknown documents

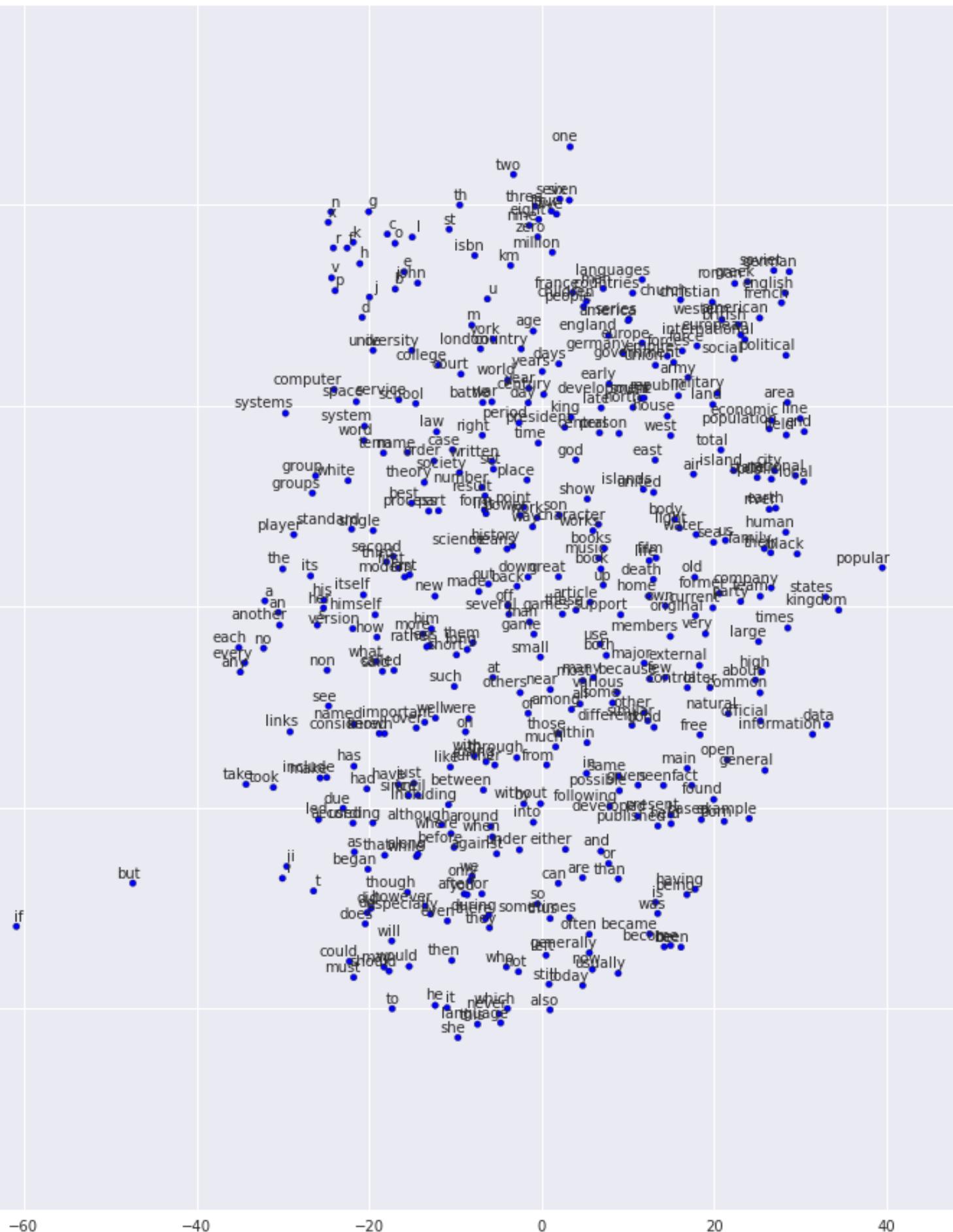
Gensim is

Available statistics

Analyze plain-text

Retrieve semantic

# Word Algebra<sup>4</sup>



```
from gensim.models.KeyedVectors import load_word2vec_format

model = load_word2vec_format(
    'GoogleNews-vectors-negative300.bin',
    limit=2000000, binary=True)

#calculate: (Rome - Italy) + France = ?
model.similar_by_vector(
    model['Rome'] - model['Italy'] + model['France'],
    topn=1)
#[('Paris', 0.7167540192604065)]

#calculate: (doctor - man) + woman = ?
model.most_similar(
    model['doctor'] - model['man'] + model['woman'],
    topn=1)
#[('nurse', 0.7127888798713684)]

#calculate: (gone - go) + eat = ?
model.most_similar(
    positive=['gone', 'eat'], negative=['go'],
    topn=1)
#[('eaten', 0.7462186217308044)]
```

<sup>4</sup> Credits: [The dark side of deep learning - Simone Scardapane - Codemotion 2017](#)

calculate: (Rome - Italy) + France = ?

```
model['Rome'] - model['Italy'] + model['France']
```

```
[(u'Paris', 0.7167540192604065)]
```

calculate: (gone - go) + eat = ?

model['gone'] - model['go'] + model['eat']

[(u'eaten', 0.7462186217308044)]

# common solutions

## Visual Studio ML Model builder



## Datasets

1. MNIST Handwritten digits
2. Google House Numbers from street view
3. CIFAR-10 and CIFAR-100
4. IMAGENET
5. Tiny Images 80 Million tiny images<sup>6</sup>.

common solutions

6. Flickr Data 100 Million Yahoo dataset
7. Berkeley Segmentation Dataset 500
8. UC Irvine Machine Learning Repository
9. Flickr 8k
10. Flickr 30k
11. Microsoft COCO
12. VQA

ready to go dataset<sup>5</sup>

13.<sup>5</sup> - [Awesome deep learning#datasets \(Github\)](#)

14. - [List of datasets for machine learning research \(Wikipedia\)](#)

- [Deep Learning datasets](#)

15. AVPRK Pathfinder
16. Air Freight - The Air Freight data set is a ray-traced image sequence along with ground truth segmentation based on

# RECIPES

## what else?

RECIPE  
Pork Chalupas

from Grams  
cooking time 6-8 hours  
2-3 # pork roast  
(I remove fat after roast begins to be  
tender or you can do it before cooked)  
1 lb. pinto beans  
clove garlic  
chile pwd.  
chopped  
1 cube beef bouillon  
or canned broth  
Jalapeno (opt.)  
ONER

# ingredients



tensorflow

TensorFlow

# deep learning in 5 steps



**STAR  
WARS**

# 1. define the problem

regression

classification

# 2. prepare your data

---

choose/create dataset

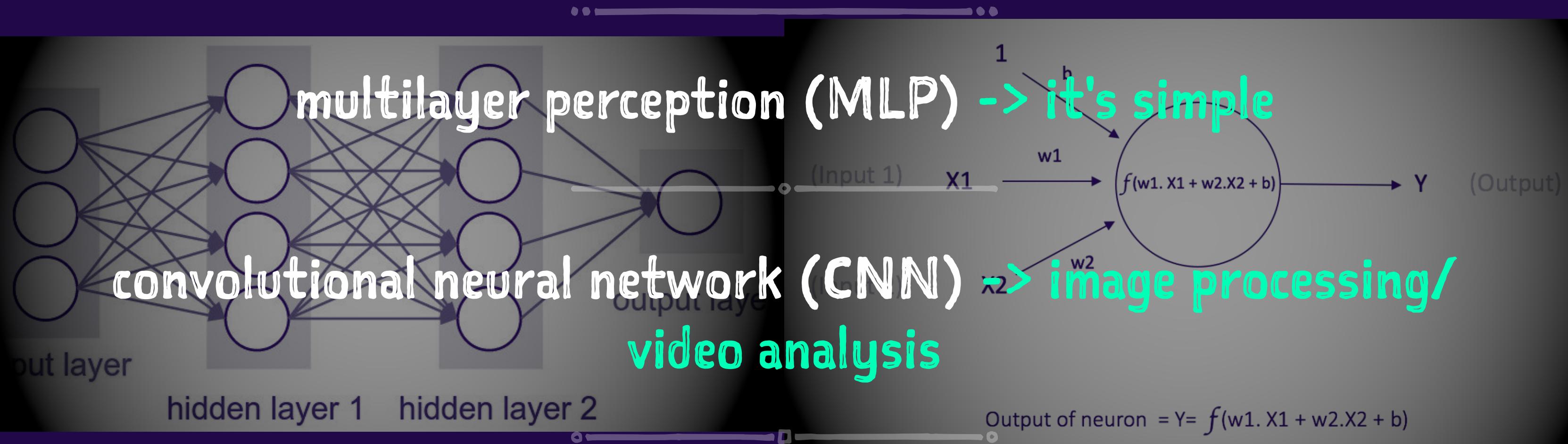
---

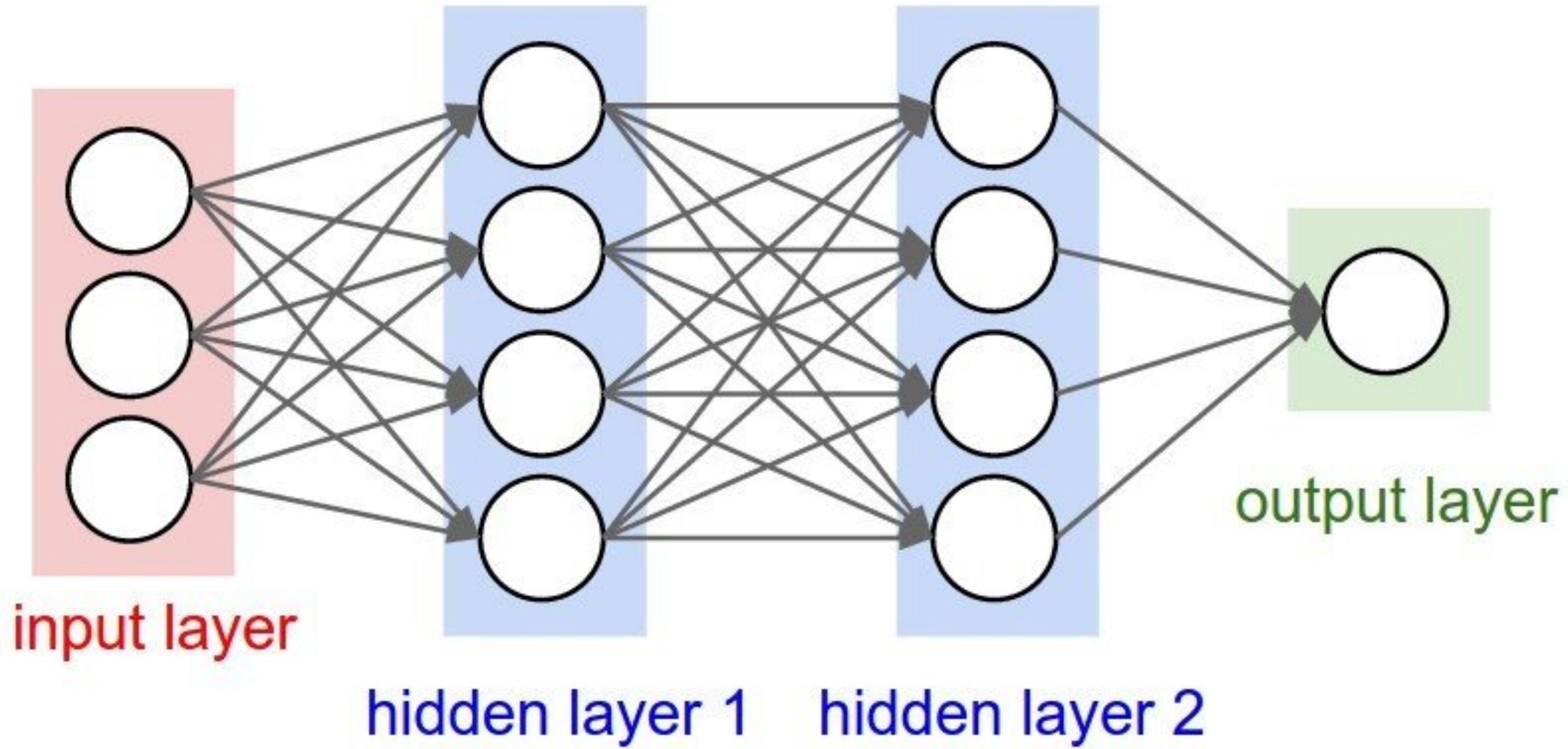
clean data

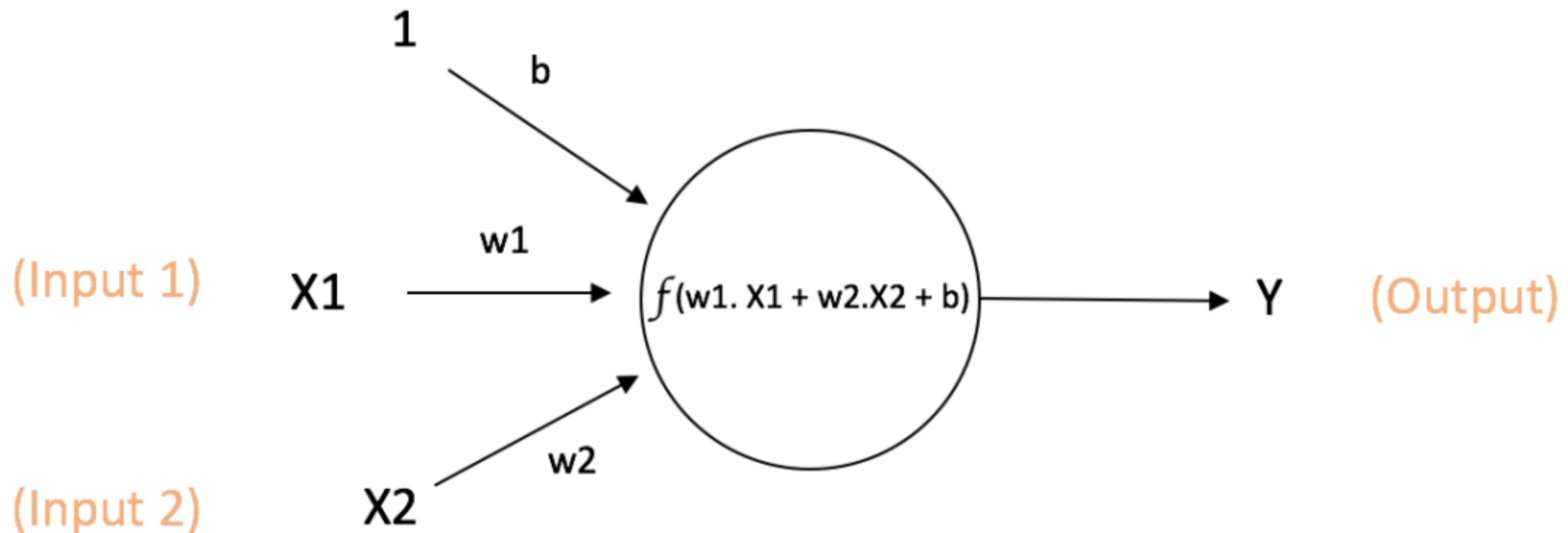
---

normalize data

# 3. define the model







Output of neuron =  $Y = f(w_1 \cdot X_1 + w_2 \cdot X_2 + b)$

# Neural Networks

Color Guided Matrix Multiplication for a Binary Classification Task  
with  $N = 4$

**Input Layer**

bias X1 X2

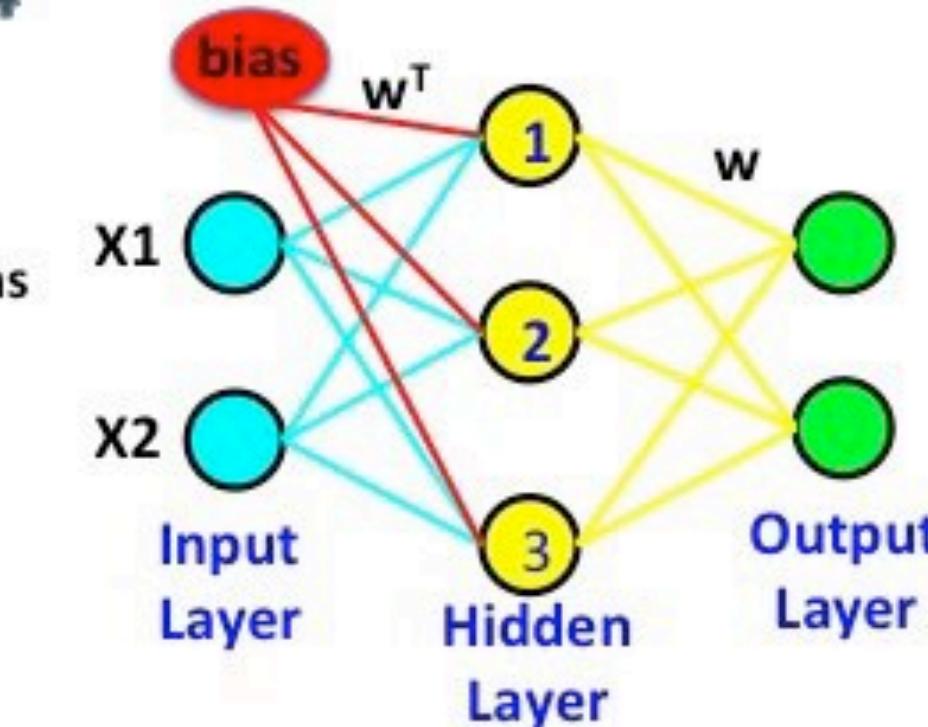
$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} .5 & .5 & .5 \\ .5 & .5 & .5 \\ .5 & .5 & .5 \end{bmatrix} =$$

$4 \times 3$

**Weights  $w^T$  (transposed)**

Go to Hidden Nodes

$3 \times 3$



**Hidden Layer**

Bias

$$= \begin{bmatrix} 1 & 1 & 1 \\ .5 & .5 & .5 \\ .5 & .5 & .5 \\ 1 & 1 & 1 \end{bmatrix} \cdot \frac{1}{1 + e^{-(wx+b)}}$$

$4 \times 3$

**Sigmoid Function**

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

**Weights**

$$= \begin{bmatrix} .2 & .1 \\ .4 & .1 \\ .4 & .1 \end{bmatrix}$$

$3 \times 2$

**Output Layer**

$$= \begin{bmatrix} 1 & .3 \\ .5 & .15 \\ .5 & .15 \\ 1 & .3 \end{bmatrix}$$

$4 \times 2$

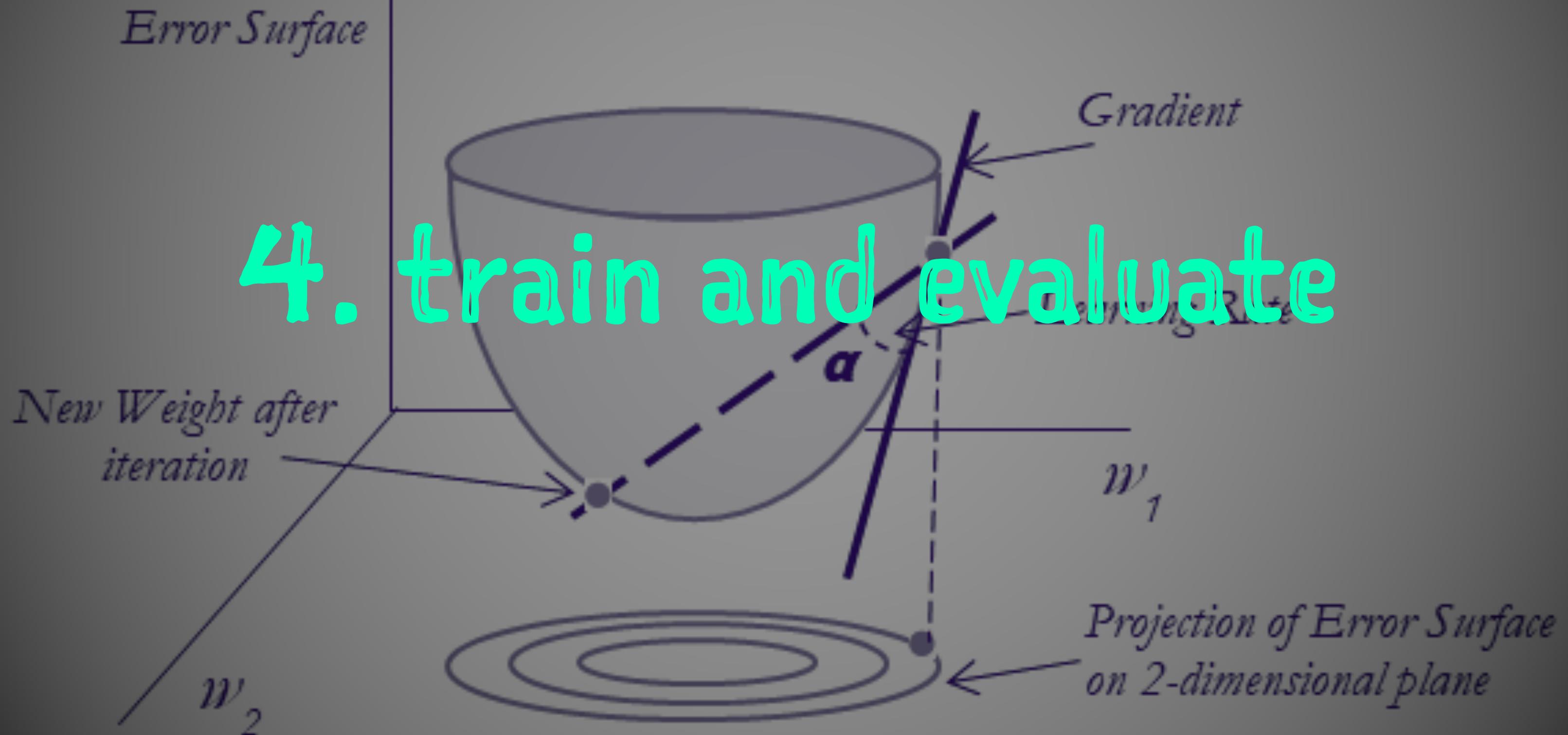
**Sigmoid Function**

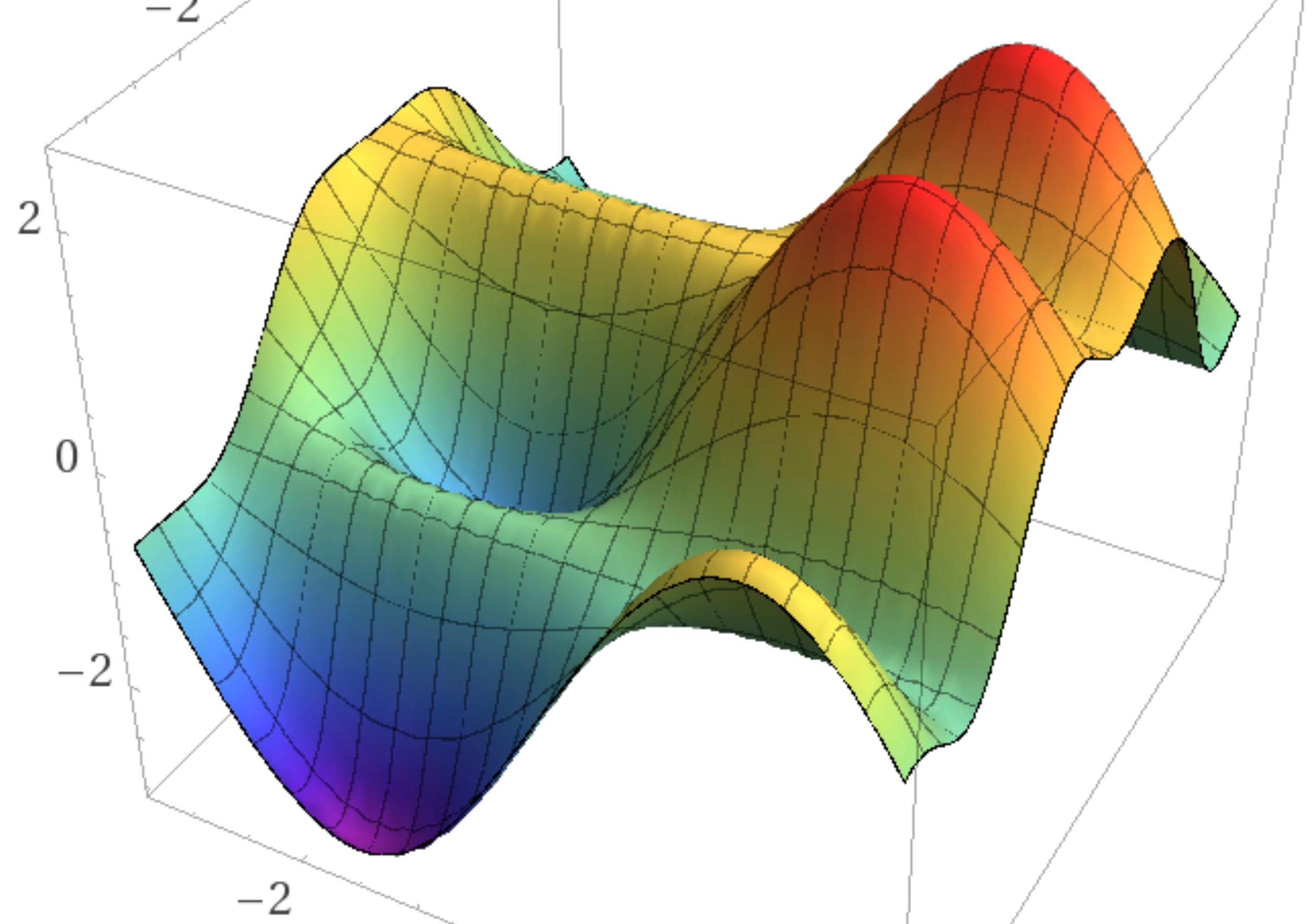
$$g(x) = \frac{1}{1 + e^{-(wx+b)}}$$

**Output**

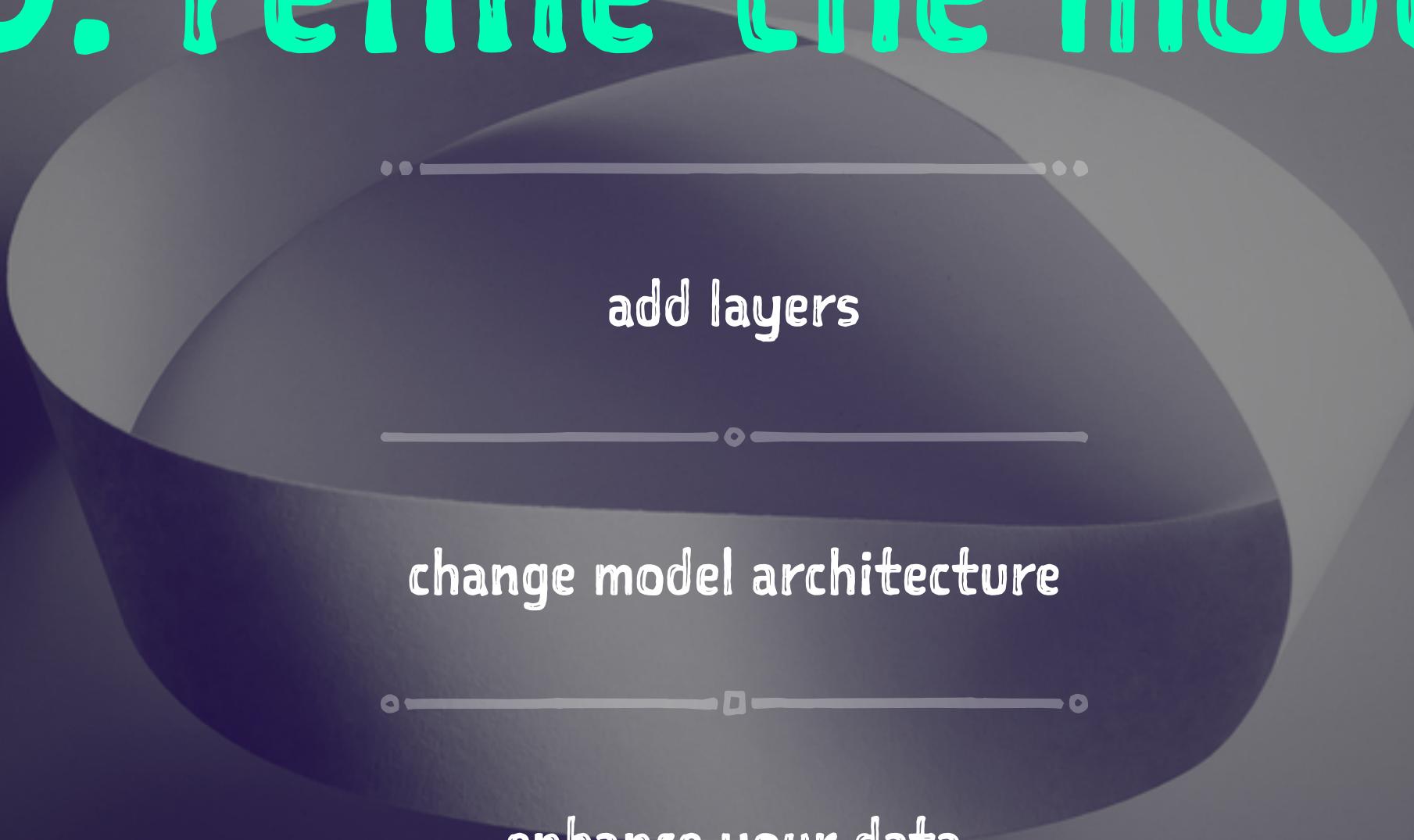
$$= \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$$

# *Gradient Descent*





# 5. refine the model



add layers

change model architecture

enhance your data

A small, light-colored wooden doghouse with a dark, textured roof. The house has a single entrance door and a smaller window-like opening near the peak. The interior is dark and appears to be made of a different material than the exterior.

overfitting



tip 1:

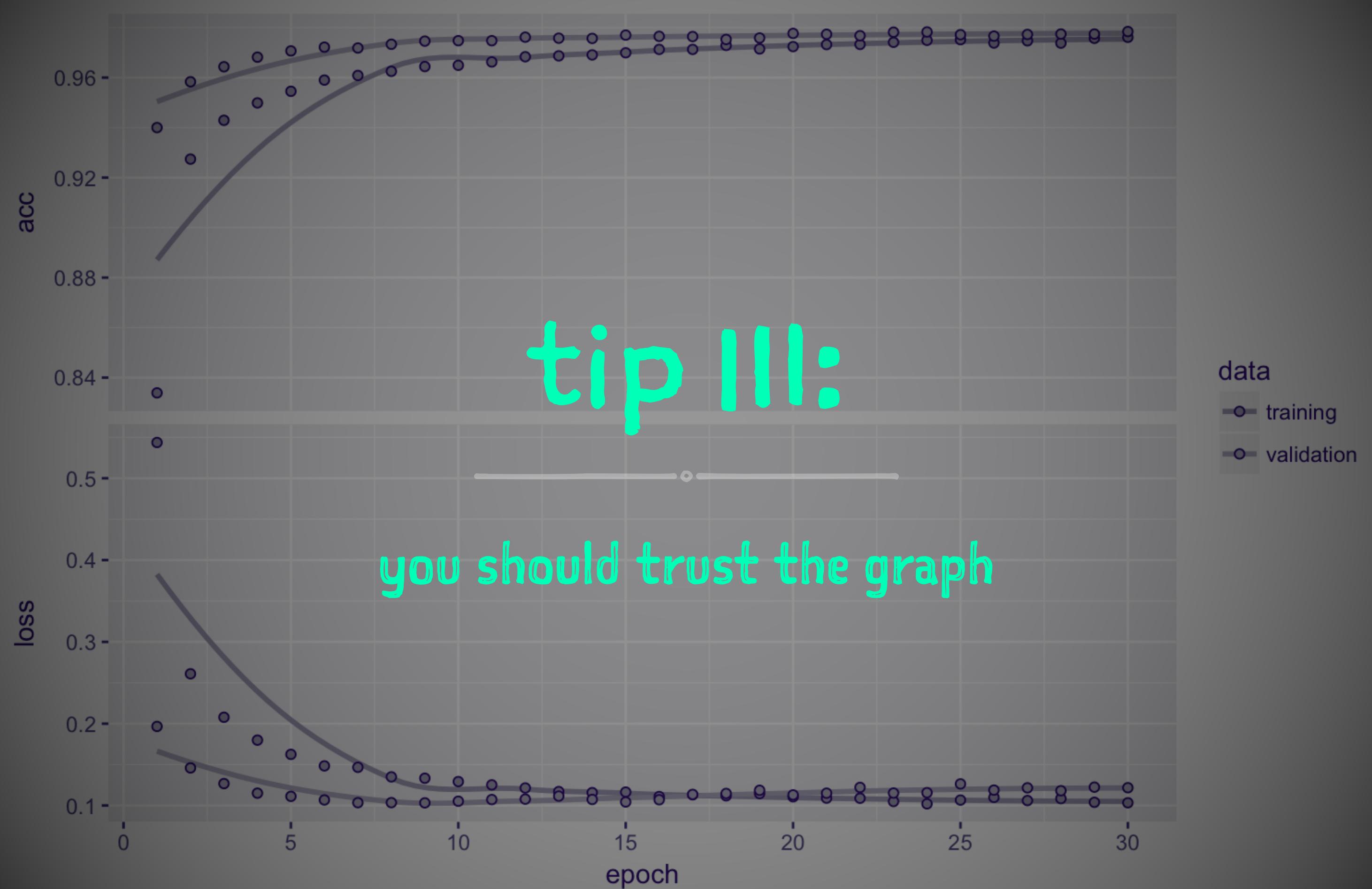
---

your dataset is the key

please, start with the simplest model possible  
please, start with the simplest mo

tip!!





A black and white photograph of a woman with long hair, seen from behind, sitting on a beach. She is wearing a plaid jacket and is sitting on a patterned blanket. She is looking out at the ocean. A white horizontal line with two small circles at each end is positioned above her head.

tip IV:

patient

AFC

TUNING

REO —  OFF

NO —  ON

FM STEREO

M-AM TUNER

TUNER INPUT

FM 98 92

104

108

MHz

0 10 20 30 40

AM 530 600 700 800

1000

1200 1400 1600

kHz

ST

... . . .

# bonus track: hyperparameters tuning<sup>6</sup>

scikit learn GridSearchCV, RandomizedSearchCV

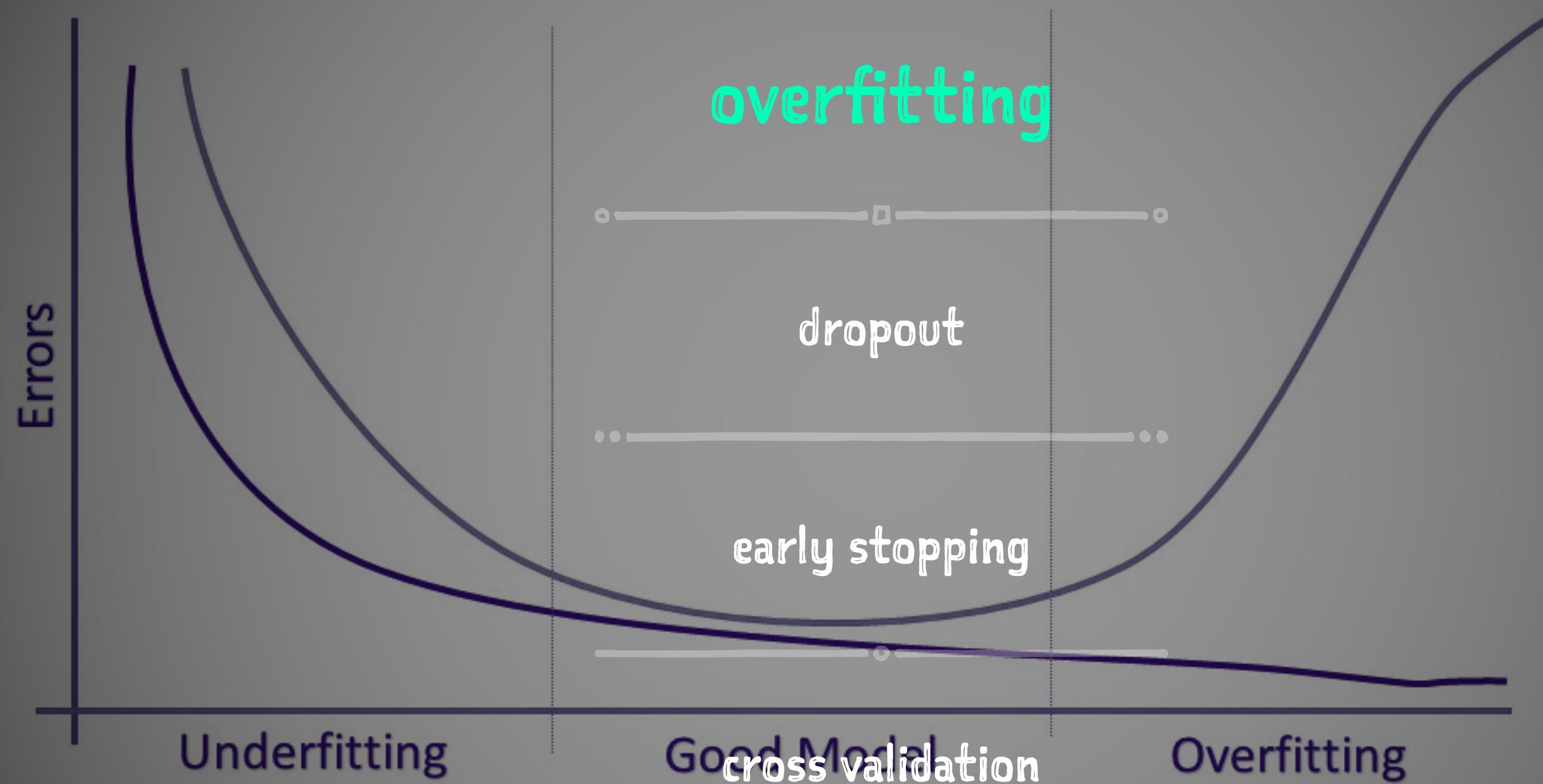
hyperopt

<sup>6</sup> - scikit learn Exhaustive&Random Grid Search

- Hyperopt

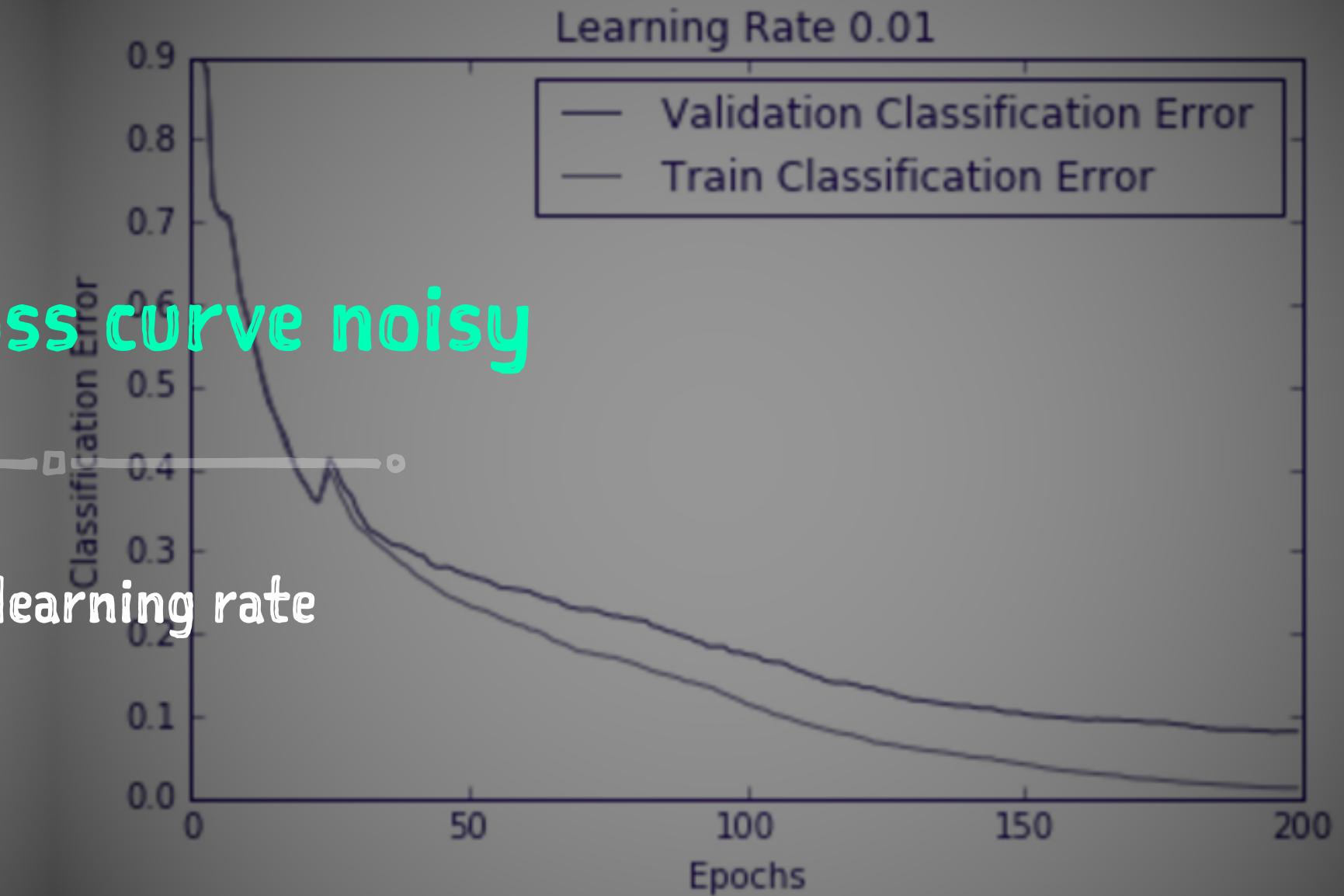
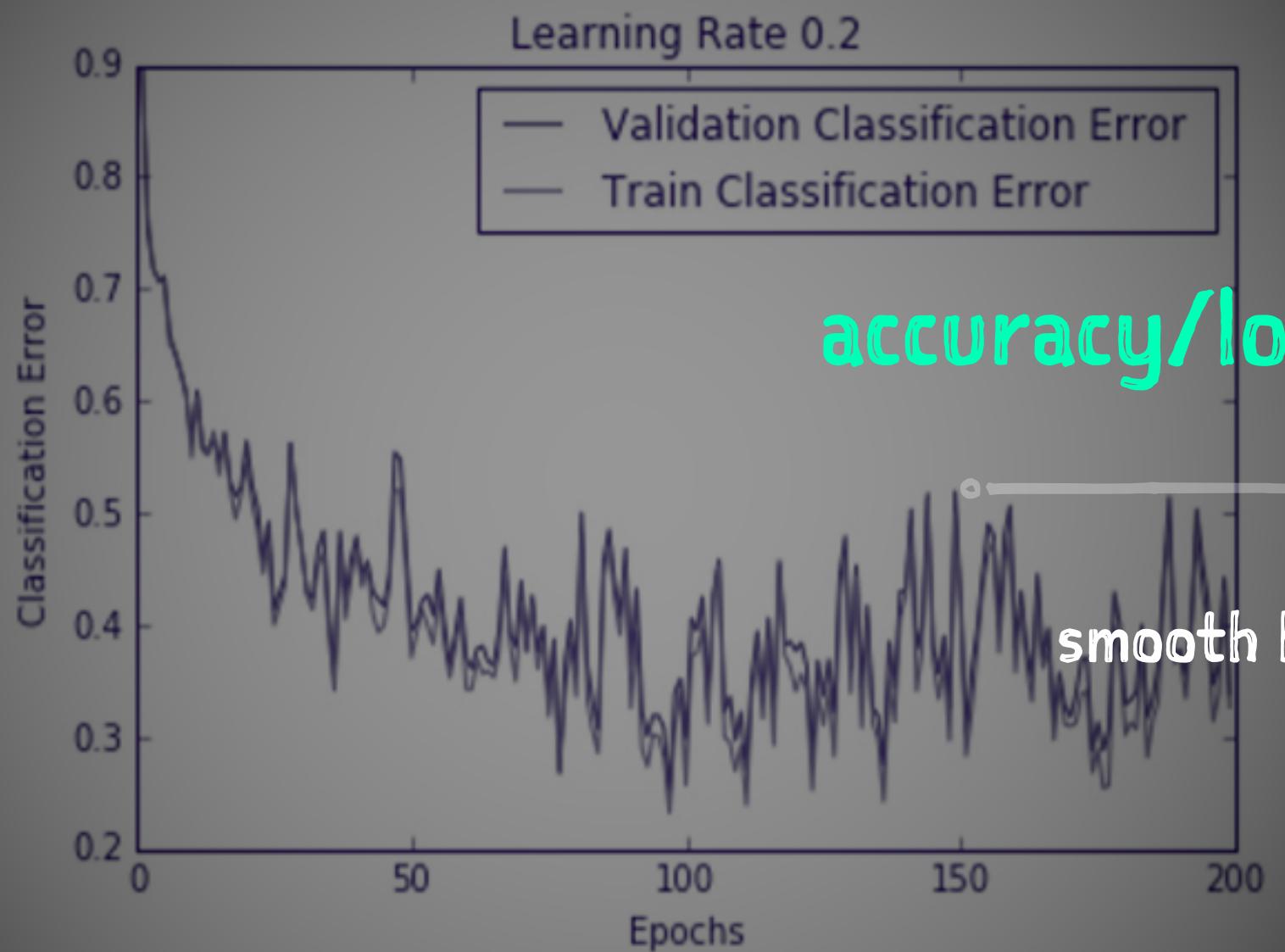
A black and white photograph showing a close-up of a wrench and a nut on a light-colored wooden surface. The wrench is positioned diagonally across the frame, with its handle pointing towards the top right and its jaws towards the bottom left. A single nut lies next to the wrench. The background is slightly blurred, emphasizing the tools in the foreground.

troubleshooting



■ Training Data

■ Test Data



accuracy/loss curve noisy

smooth learning rate

# Resources #1

- [Google Machine Learning Crash Course](#)
- [Learn TensorFlow and deep learning, without a Ph.D. by Martin Görner](#)
- [How to Use Metrics for Deep Learning with Keras in Python](#)
- [Neural Networks for Machine Learning course from Coursera](#)

# Resources #2

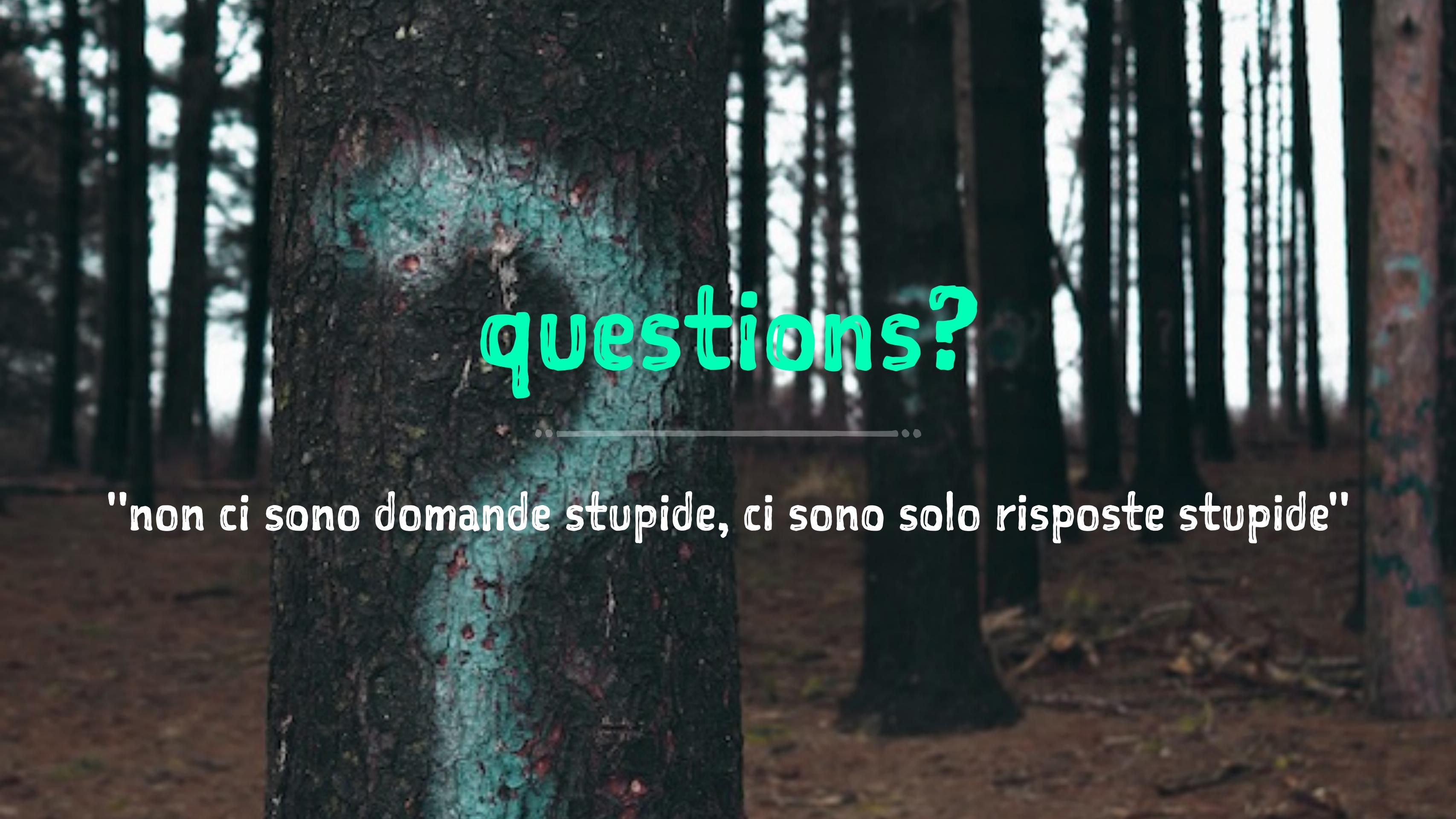
- Machine learning finance
- Keras
- Tensorflow
- rucka/deeplearning docker image

It is a capital mistake  
to theorize before one has  
data

Sherlock Holmes



thank you!



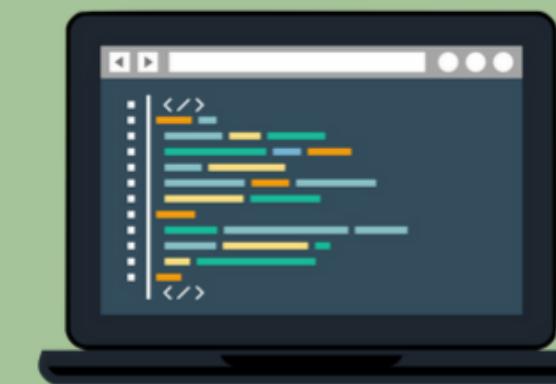
questions?

---

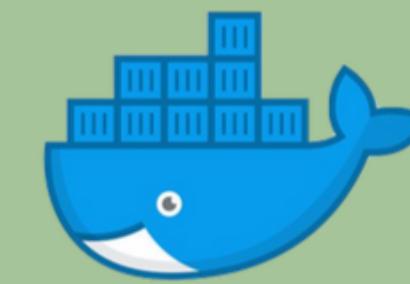
"non ci sono domande stupide, ci sono solo risposte stupide"



slides



code



environment



# >> AI&ML CONF

