AN INTRODUCTION TO DEEP LEARNING FOR ASTRONOMY

Marc Huertas-Company

<u>IAC WINTER School 2018</u>

REFERENCES

SEVERAL SLIDES / INFOS SHOWN HERE ARE INSPIRED/ TAKEN FROM OTHER WORKS / COURSES FOUND ONLINE

- Deep Learning: Do-It-Yourself! [Bursuc, Krzakala, Lelarge]
- DEEPLEARNING.AI [COURSERA, Ng, Bensouda, Katanforoosh]
- MACHINE LEARNING LECTURES [Keck]
- EPFL DEEP LEARNING COURSE [Fleuret]

Thanks to all of them!

SOME PRELIMINARY NOTES

I AM NOT A MACHINE LEARNING RESEARCHER

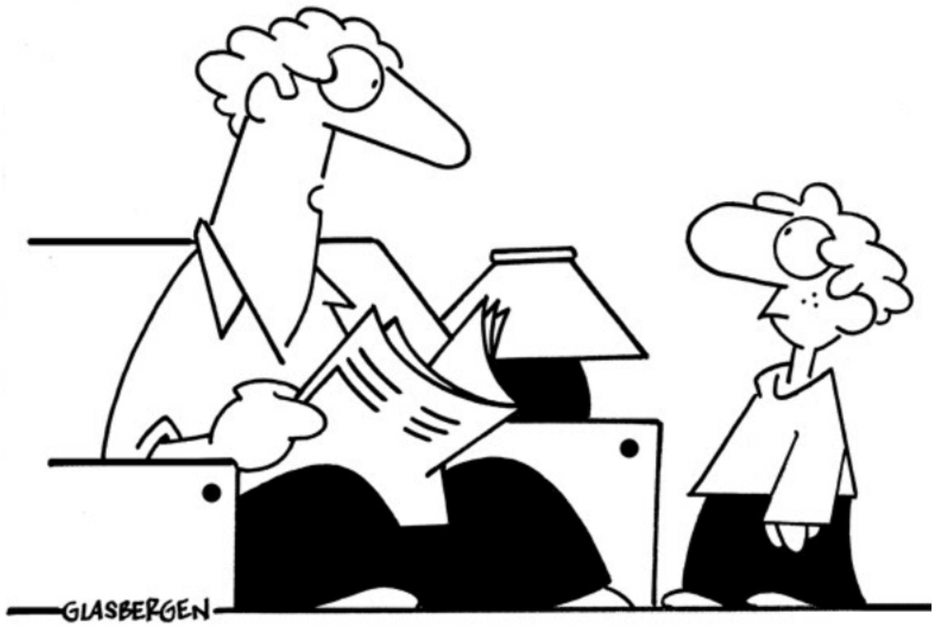
SOME PRELIMINARY NOTES

I AM NOT A MACHINE LEARNING RESEARCHER

ONLY AN ASTRONOMER WHO HAS BEEN USING MACHINE LEARNING FOR THE LAST ~14 YEARS FOR MY RESEARCH

THIS LECTURE IS INTENDED TO PROVIDE A **GLOBAL** UNDERSTANDING OF HOW AI TECHNIQUES WORK AND ESPECIALLY **HOW TO USE THEM FOR YOUR RESEARCH**

WHAT ARE WE GOING TO LEARN?

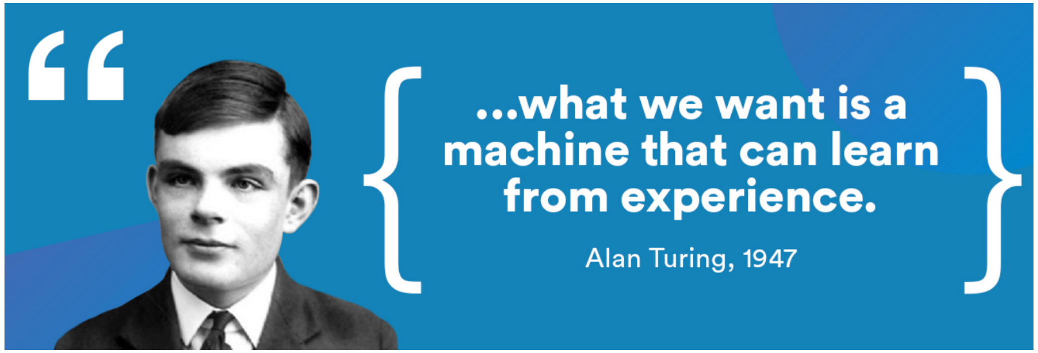


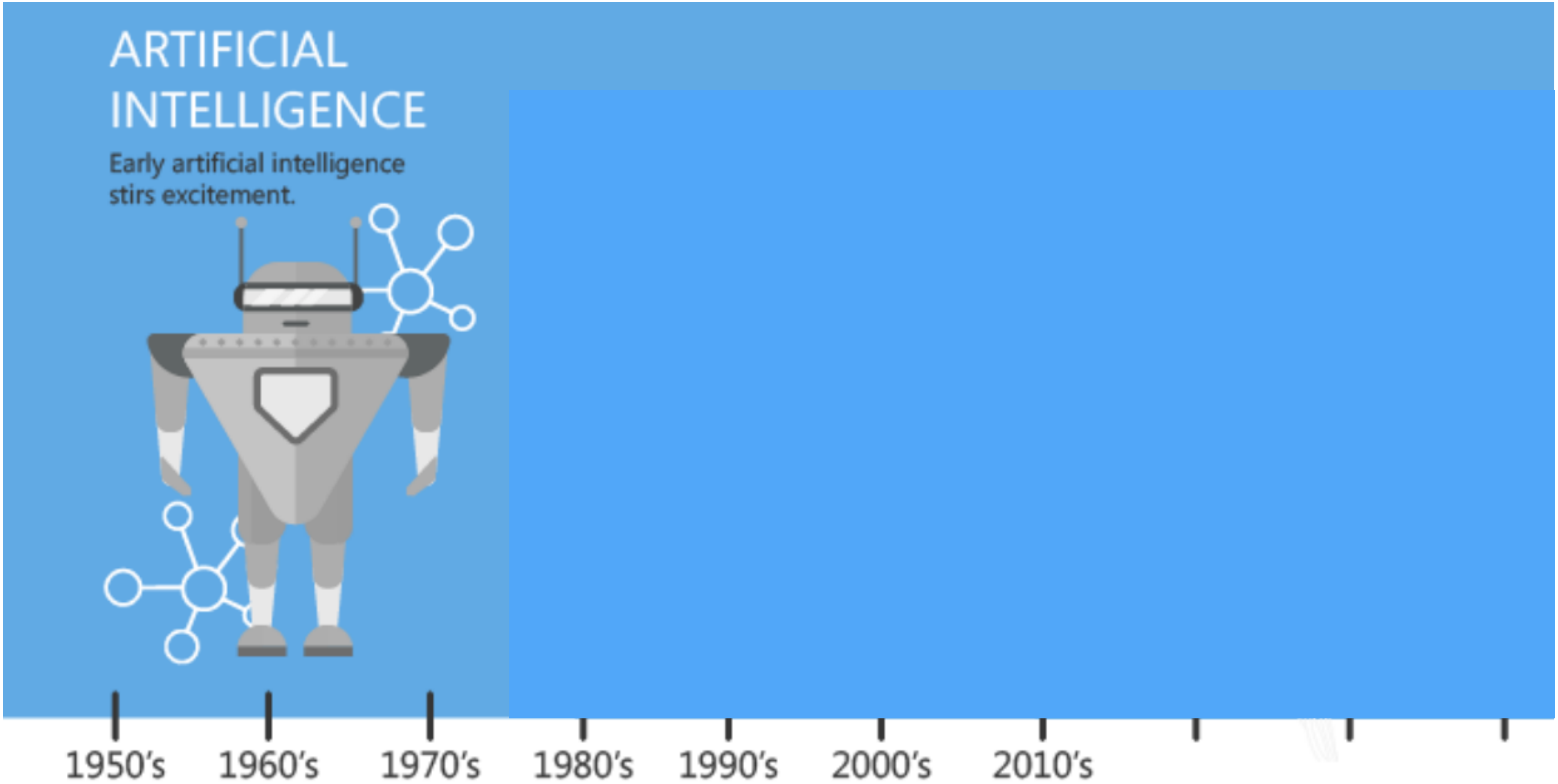
"*Artificial intelligence* is when you get a college degree, but you're still stupid when you graduate."

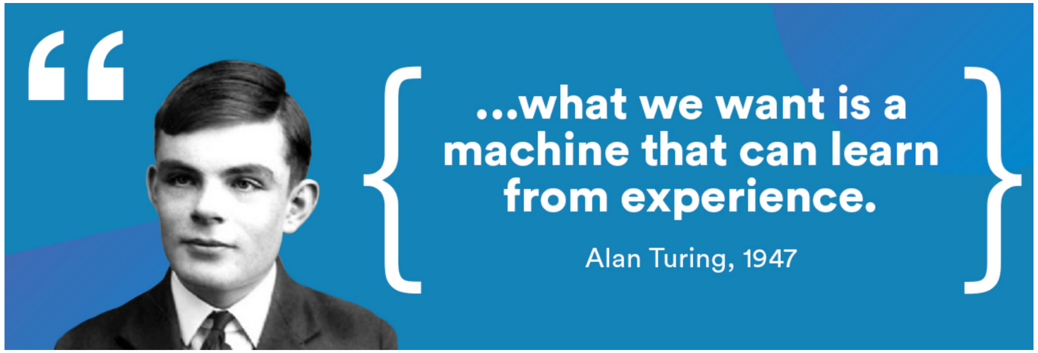
WHAT ARE WE GOING TO LEARN?

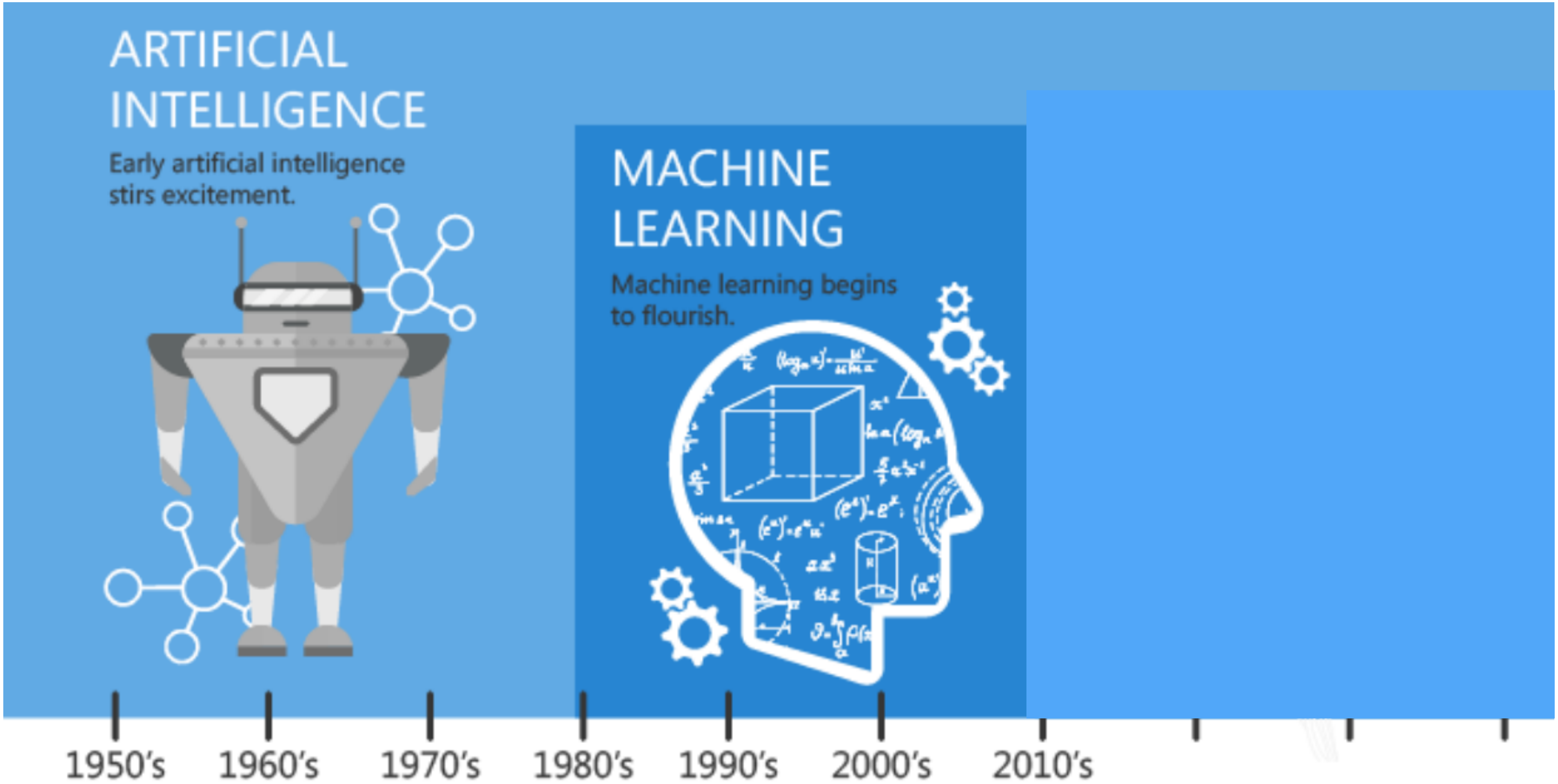
A BUNCH OF SOMETIMES CONFUSING TERMS…

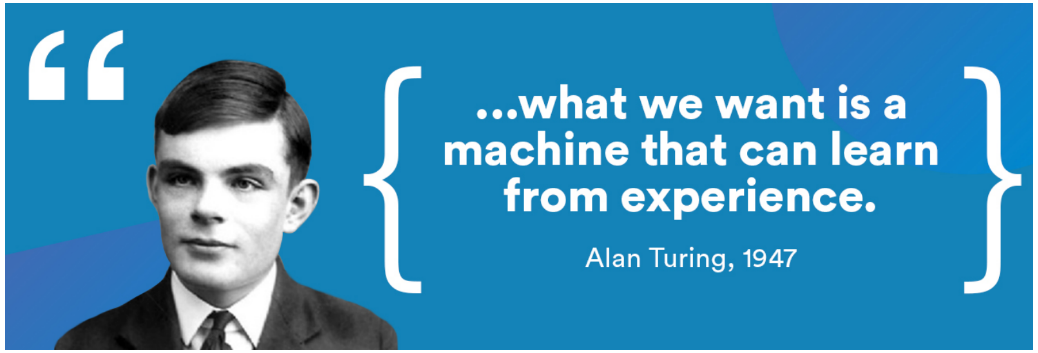
"Artificial intelligence is when you get a college degree, but you're still stupid when you graduate."


...what we want is a machine that can learn from experience.
Alan Turing, 1947

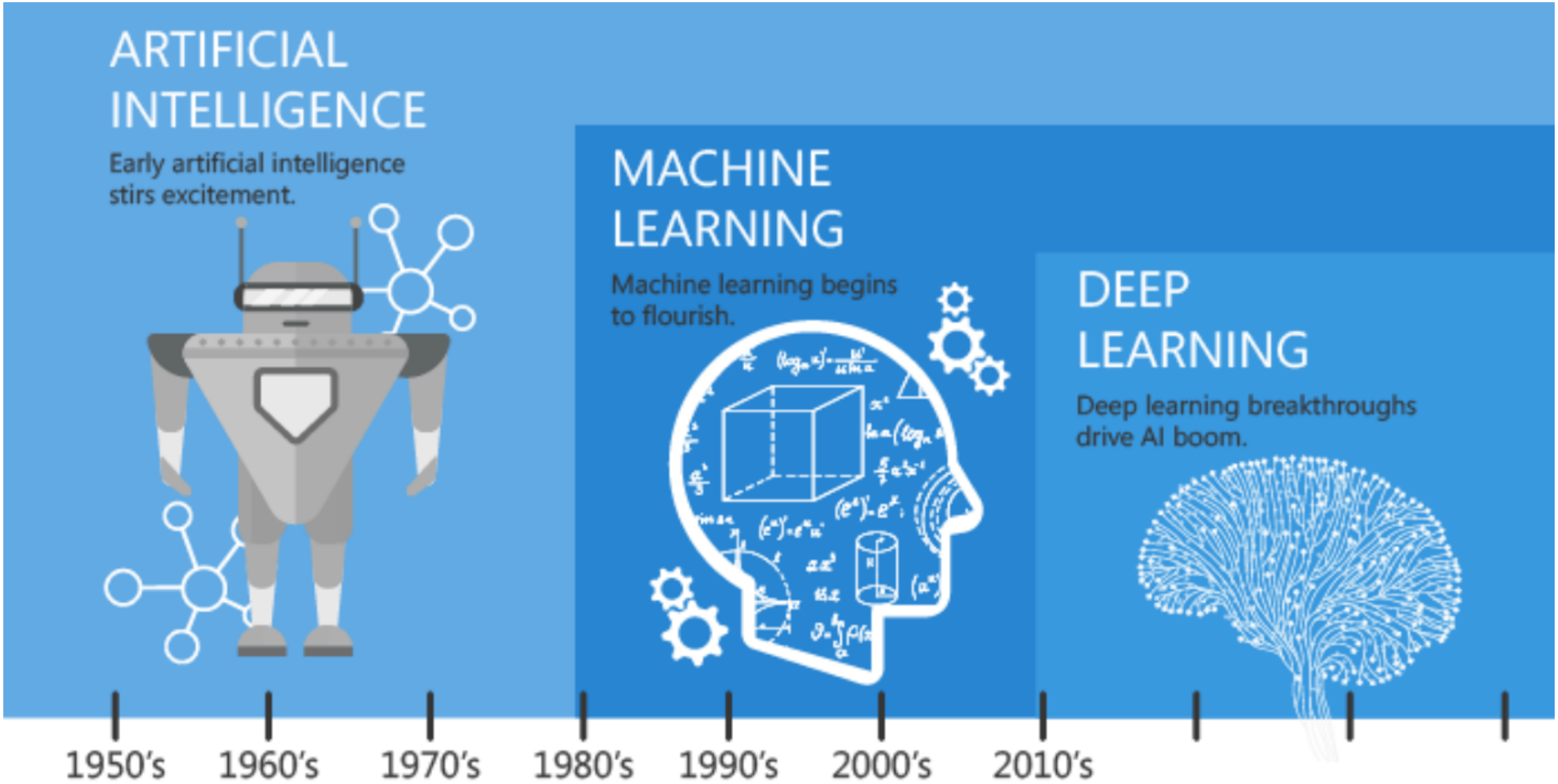

ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
1950's
1960's
1970's
1980's
1990's
2000's
2010's


...what we want is a machine that can learn from experience.
Alan Turing, 1947


ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
1950's
1960's
1970's
1980's
1990's
2000's
2010's



...what we want is a machine that can learn from experience.
Alan Turing, 1947



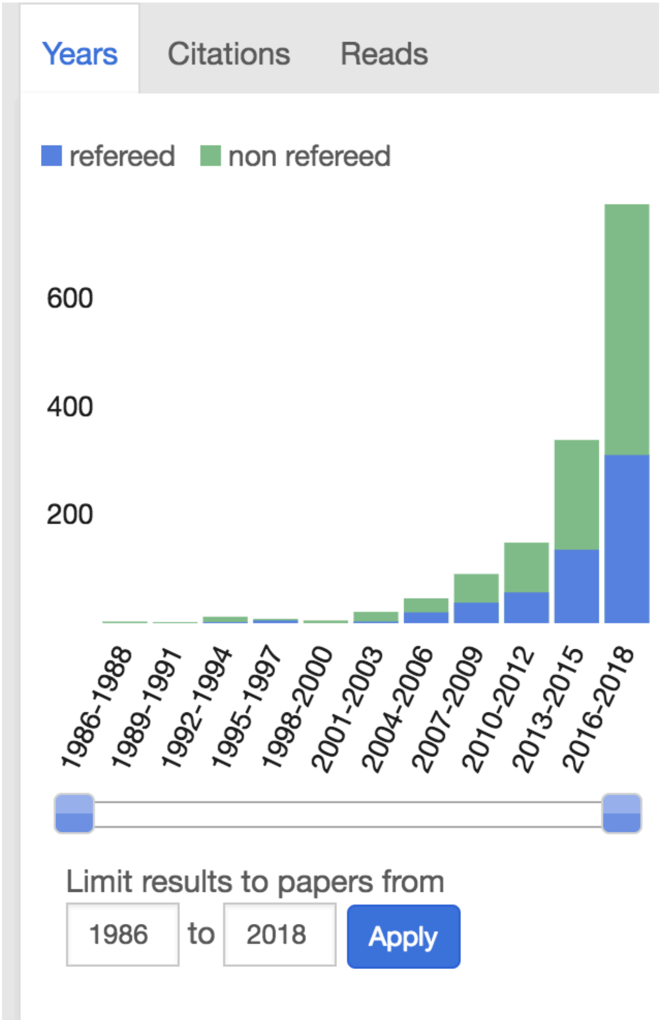
ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950's
1960's
1970's
1980's
1990's
2000's
2010's

AN AMAZING MEDIA ATTENTION

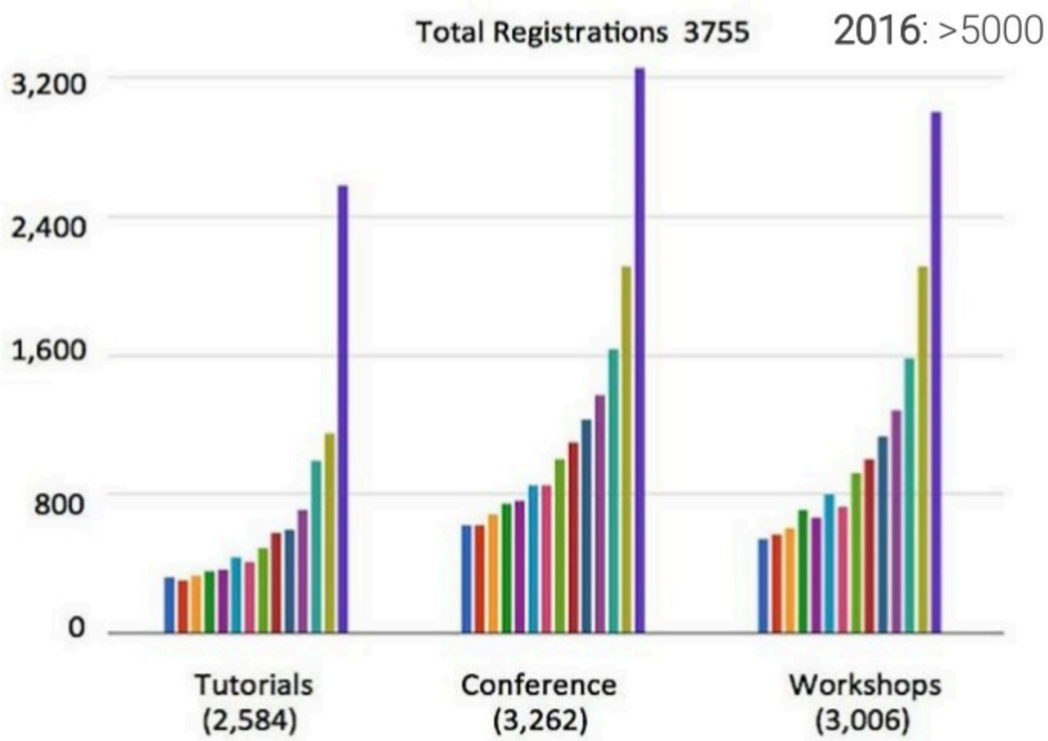
Le CNRS, Inria, l'université PSL et les entreprises Amazon, Criteo, Facebook, Faurecia, Google, Microsoft, NAVER LABS, Nokia Bell Labs, le Groupe PSA, SUEZ et Valeo font converger intérêts académiques et industriels et s'unissent pour créer, à Paris, l'Institut PRAIRIE dont l'objectif est de devenir une référence internationale de l'intelligence artificielle.

AI FEVER?

PUBLICATIONS (ADS)

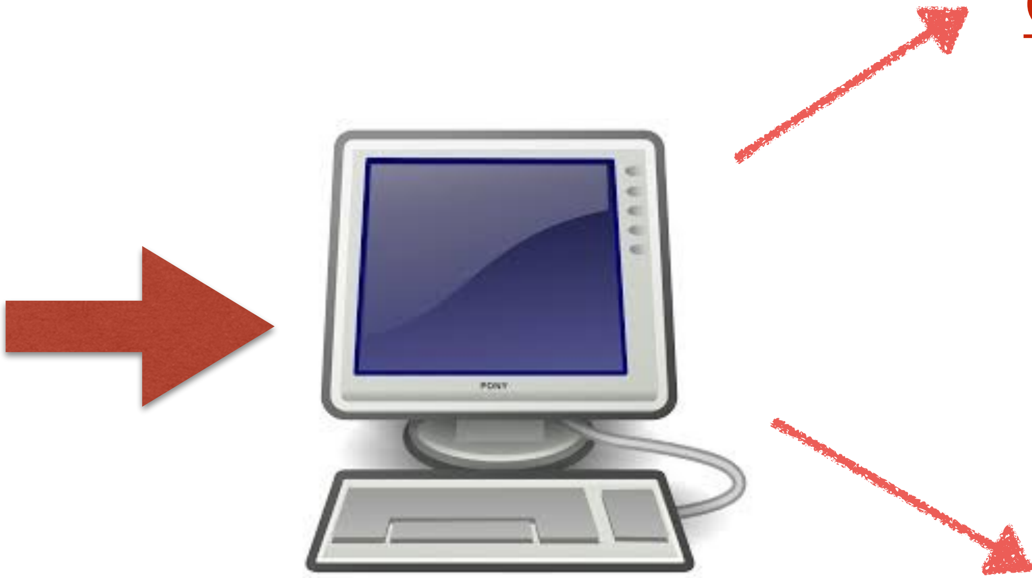


CONFERENCES



Source

BEFORE 2012….

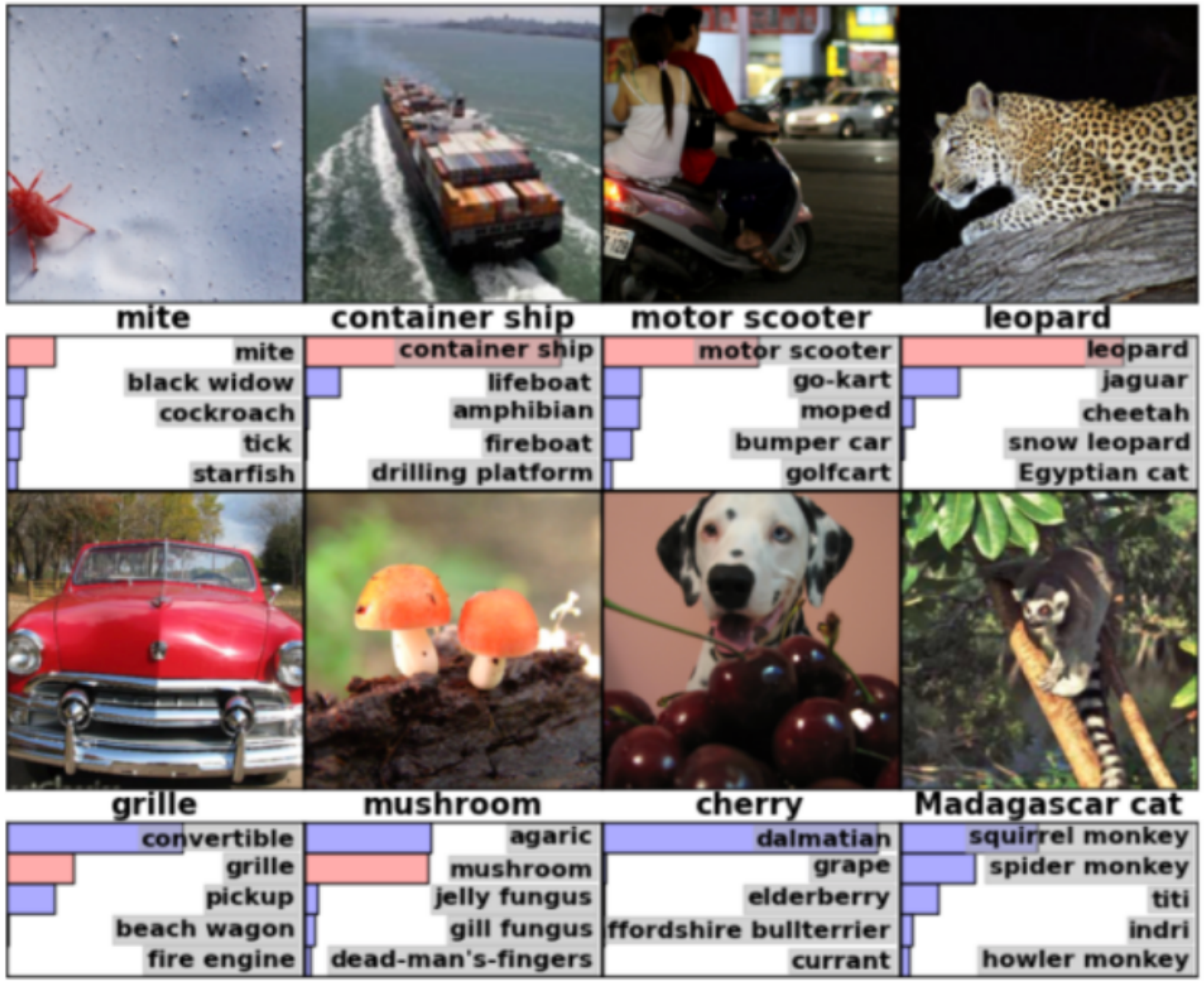


CAT?

DOG?

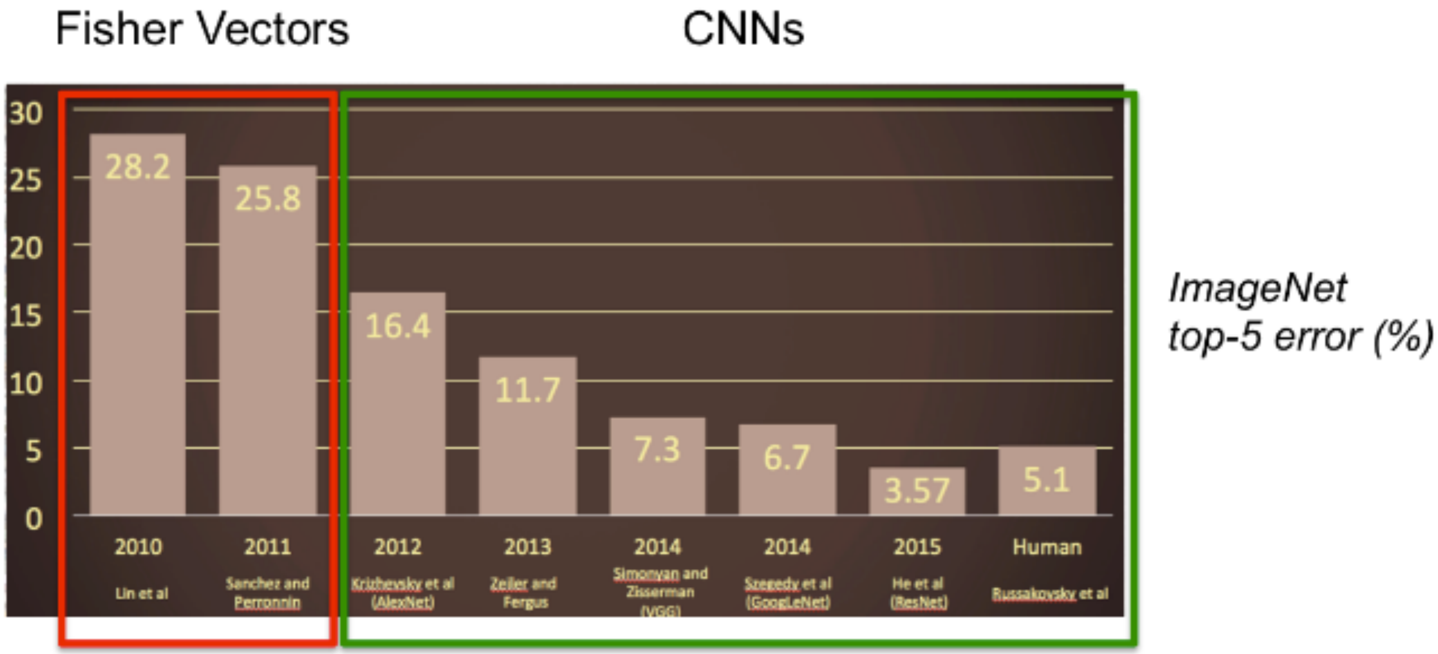
TRIVIAL HUMAN TASKS REMAINED CHALLENGING FOR COMPUTERS

AFTER 2012



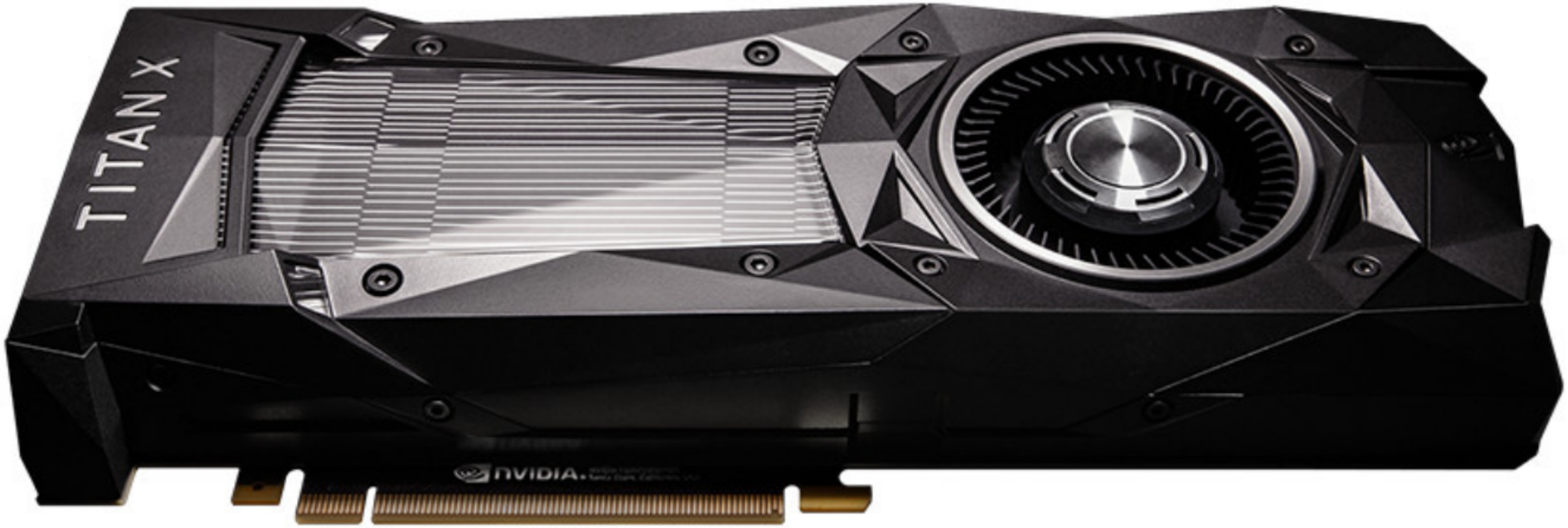
IT HAS BECOME TRIVIAL….

THIS IS A CHANGE OF PARADIGM!

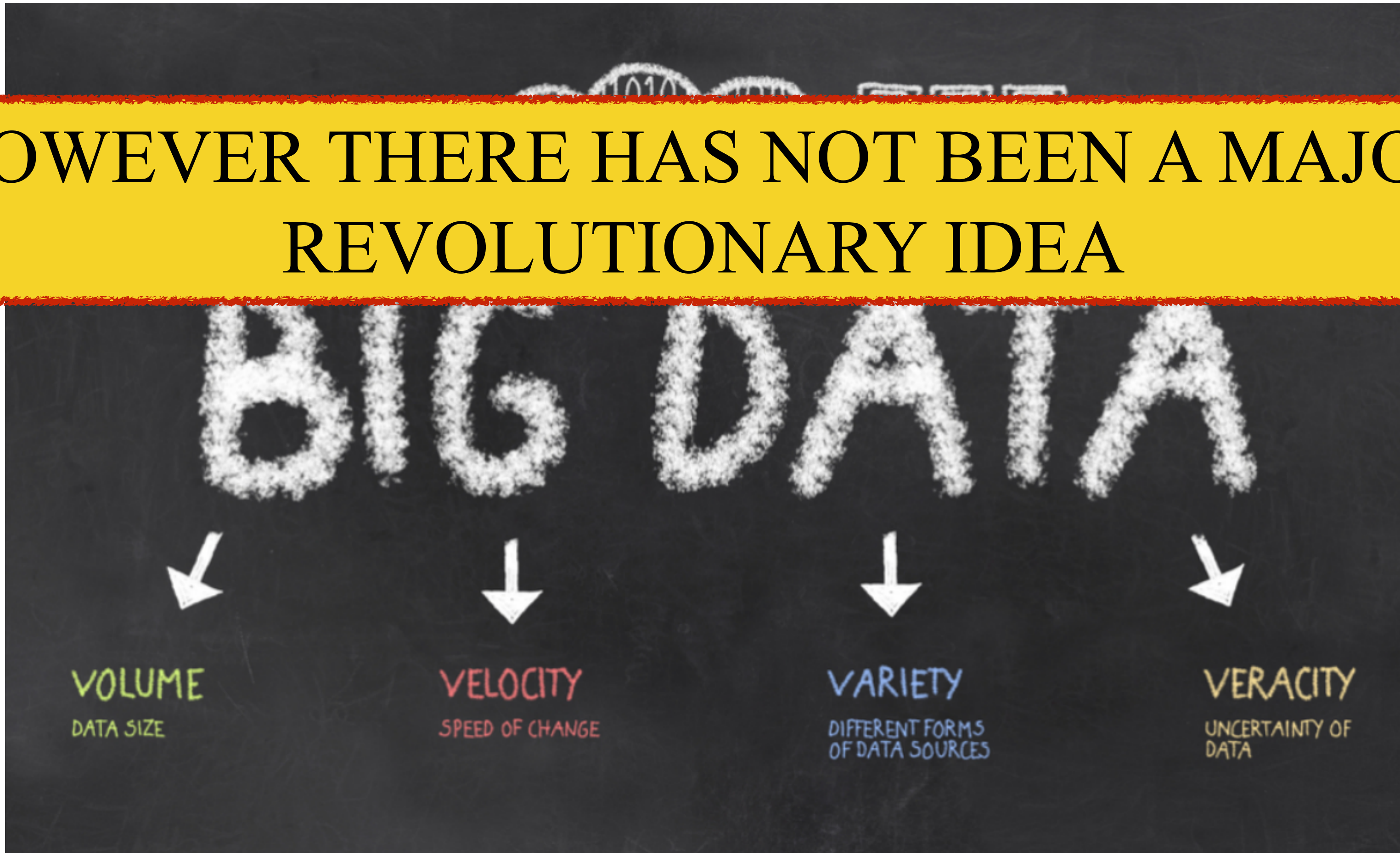


ONE OF THE MAIN REASONS OF THIS BREAKTHROUGH IS THE AVAILABILITY OF VERY LARGE DATASETS TO LEARN

COMBINED WITH THE TECHNOLOGY TO PROCESS ALL THIS DATA



ONE OF THE MAIN REASONS OF THIS BREAKTHROUGH IS THE AVAILABILITY OF VERY LARGE DATASETS TO LEARN



HOWEVER THERE HAS NOT BEEN A MAJOR REVOLUTIONARY IDEA

WHAT ARE WE GOING TO LEARN?

BASICS OF CLASSICAL MACHINE LEARNING
(this is mostly covered by my colleagues)

BASICS OF DEEP LEARNING
(BOTH SUPERVISED AND UNSUPERVISED)

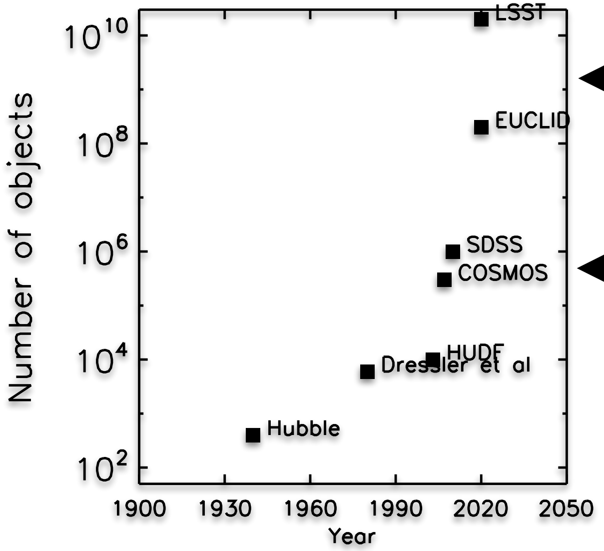
HOPING THAT THIS WOULD BE USEFUL FOR YOUR RESEARCH!

(Apologies in advance for biases on Extra-Galactic Science + imaging)

WHY DO WE NEED THESE TOOLS IN ASTRONOMY?

WHY DO WE NEED THESE TOOLS IN ASTRONOMY?

AS IN MANY OTHER DISCIPLINES THE BIG-DATA REVOLUTION HAS ARRIVED TO ASTRONOMY TOO



BIG-DATA REVOLUTION

we are here

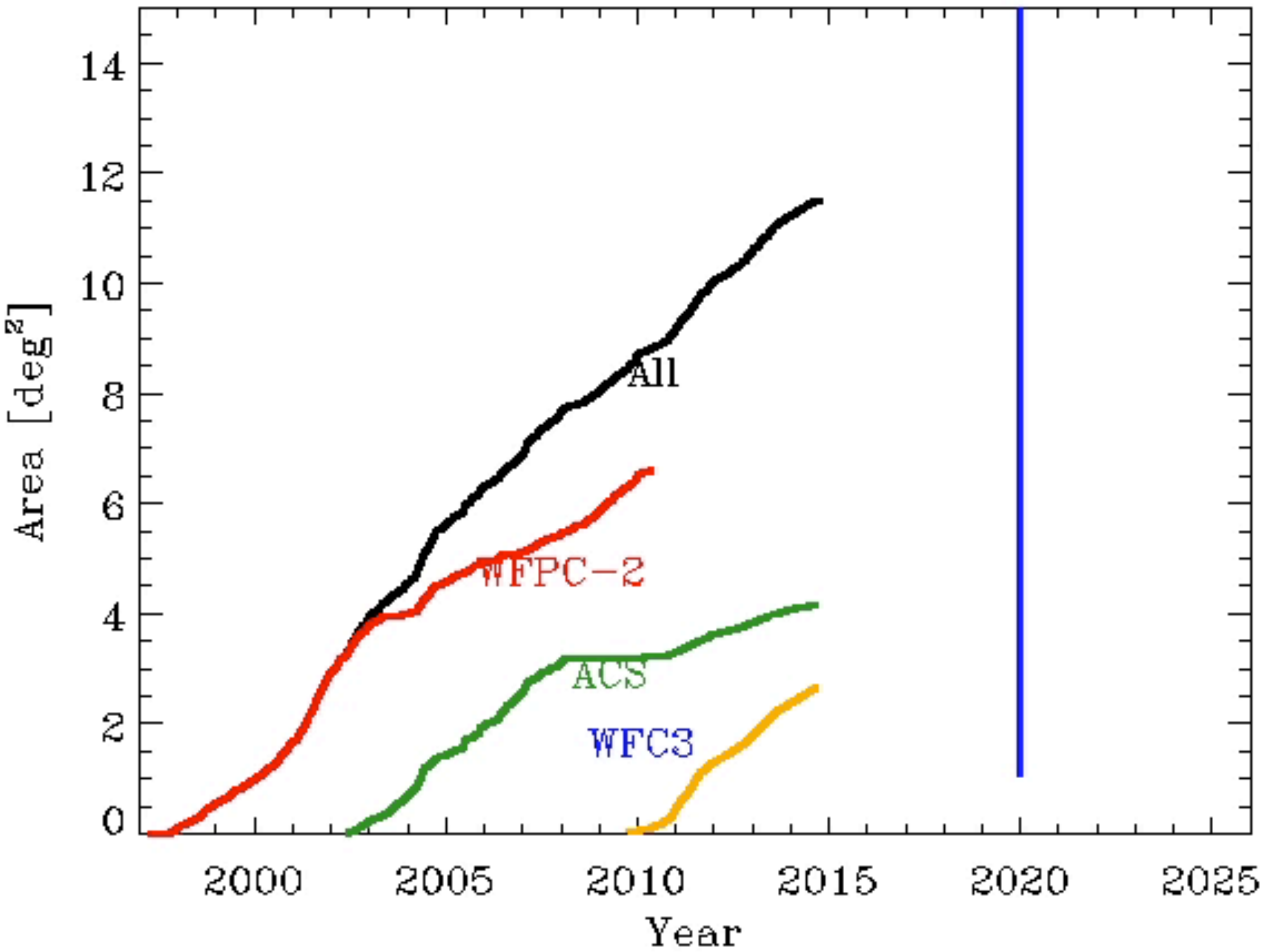
EXTREMELY LARGE IMAGING SURVEYS DELIVERING BILLIONS OF OBJECTS IN 2-5 YEARS

Large Synoptic Survey Telescope

Euclid consortium

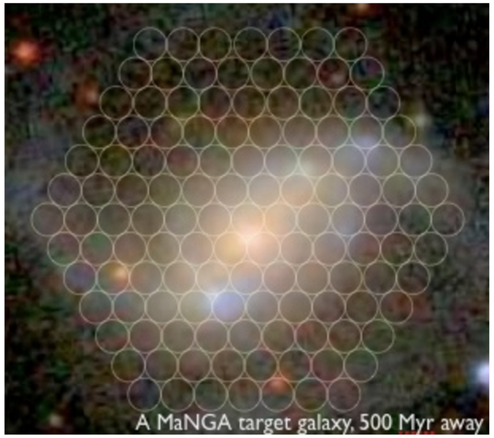
WIDE-FIELD INFRARED SURVEY TELESCOPE
WFIRST
ASTROPHYSICS • DARK ENERGY • EXOPLANETS

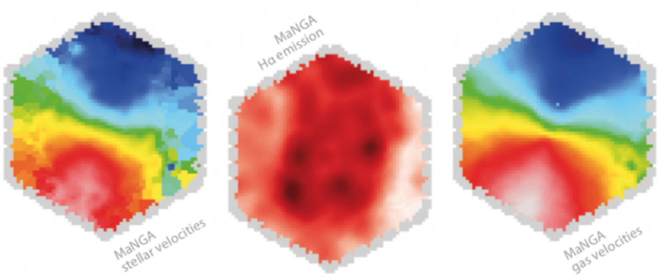
LSST simulation



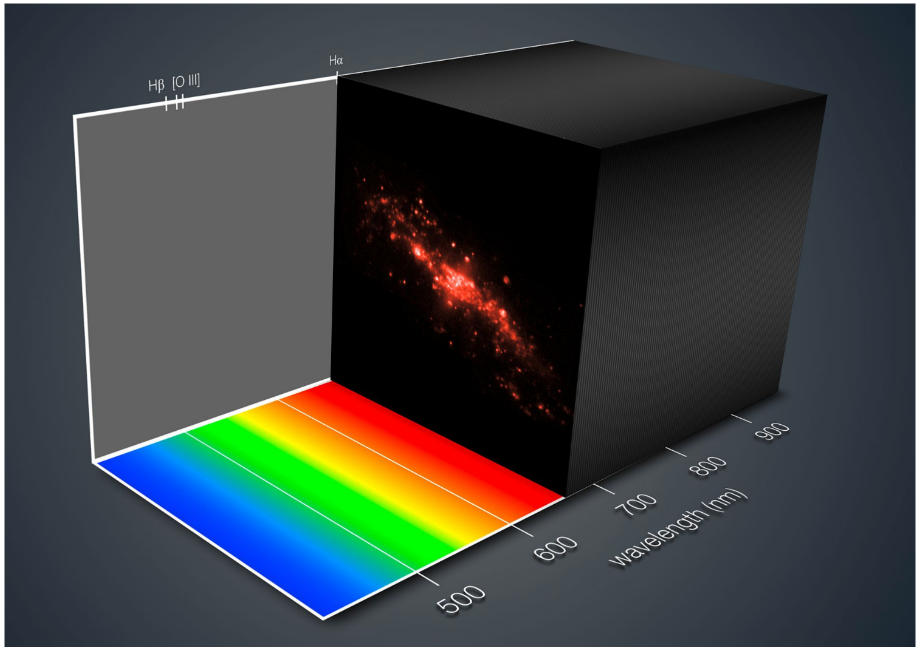
(Thanks to J. Brinchmann)

NOT ONLY VOLUME: AN INCREASING COMPLEXITY OF DATA

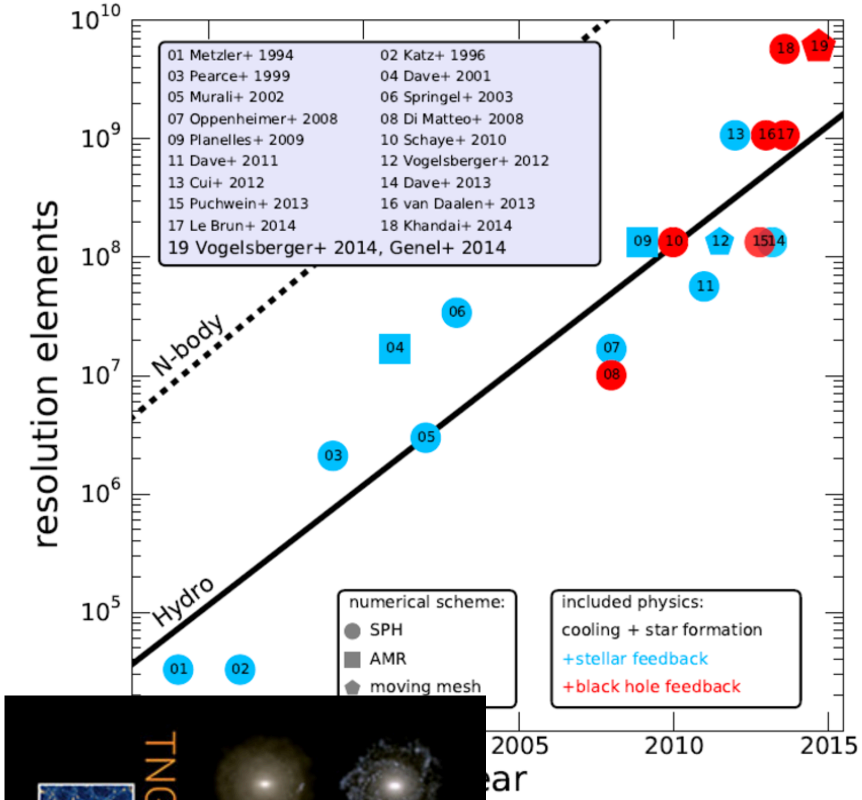




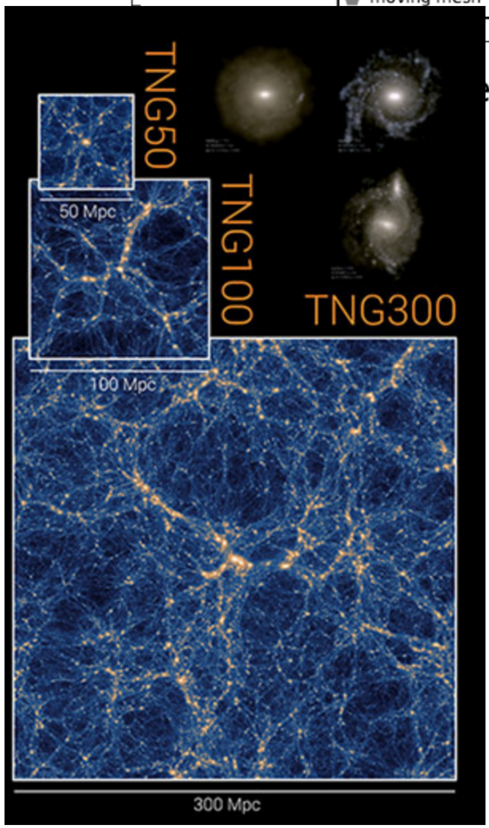
MANGA Survey



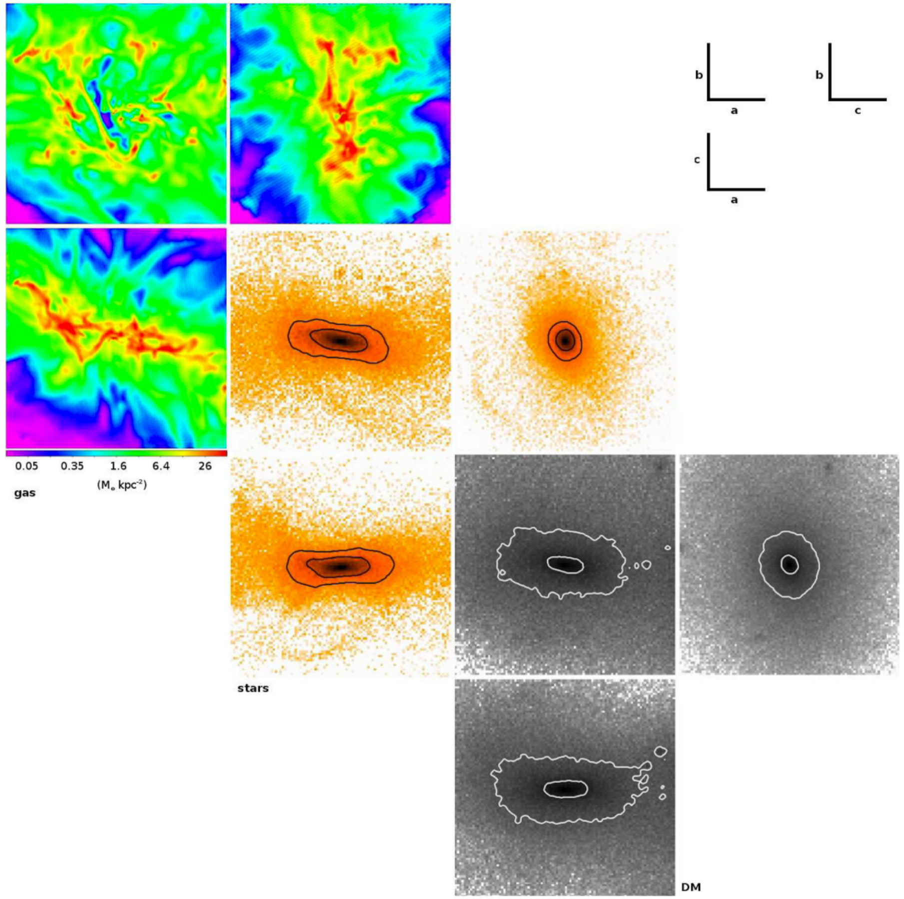
MUSE@VLT



Genel+14



AND ALSO SIMULATIONS!



Ceverino+15

PROGRAM FOR THE WEEK

- **<u>PART I: A VERY QUICK INTRODUCTION TO 'CLASSICAL' MACHINE LEARNING</u>**
  - UNSUPERVISED / SUPERVISED
  - GENERAL STEPS TO "TEACH A MACHINE"
  - "CLASSICAL" CLASSIFIERS

PROGRAM FOR THE WEEK

- **PART II: FOCUS ON ‘SHALLOW’ NEURAL NETWORKS**
  - PERECPTRON, NEURON DEFINITION
  - LAYER OF NEURONS, HIDDEN LAYERS
  - ACTIVATION FUNCTIONS
  - OPTIMIZATION [GRADIENT DESCENT, LEARNING RATES]
  - BACKPROPAGATION

PROGRAM FOR THE WEEK

- **PART III: CONVOLUTIONAL NEURAL NETWORKS**
  - CONVOLUTIONS AS NEURONS
  - CNNs [POOLING, DROPOUT]
  - VANISHING GRADIENT / BATCH NORMALIZATION

PROGRAM FOR THE WEEK

- **PART IV: IMAGE TO IMAGE NETOWRKS + INTRODUCTION TO UNSUPERVISED DEEP LEARNING**
    - NETWORKS FOR IMAGE SEGMENTATION
    - AUTO-ENCODERS
    - GENERATIVE ADVERSARIAL NETOWRKS
    - ANOMALY DETECTION

PROGRAM FOR THE WEEK

- **PART V: SOME PRACTICAL CONSIDERATIONS**
  - HOW DO I SETUP MY CNN?
  - HOW LARGE DO TRAINING SETS NEED TO BE?
  - OPTIMIZING YOUR NET: HYPER PARAMETER SEARCH
  - VISUALIZING CNNs [DECONVNETS, INCEPTIONISM, INTEGRATED GRADIENTS]

HANDS-ON SESSION

WE WILL TRY TO IMPLEMENT SOME OF THE THINGS LEARNED

MORE PRECISELY WE WILL SET UP A DEEP NETWORK TO MEASURE GALAXY ELLIPTICITIES

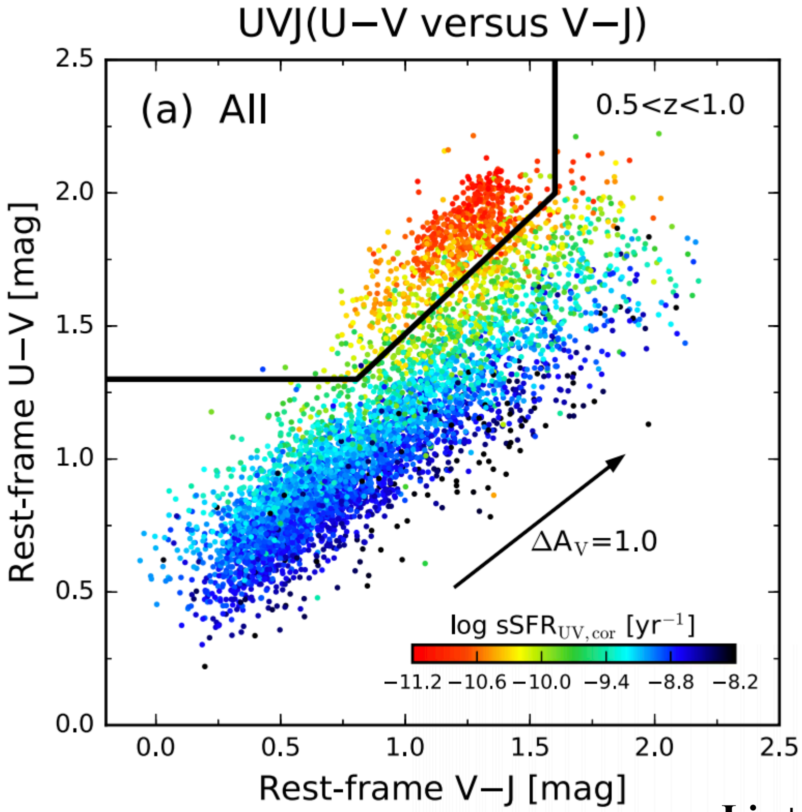
LET'S TRY TO DISCUSS AS MUCH AS POSSIBLE!

SOFTWARE REQUIREMENTS

- PYTHON 3 OR GREATER
- TENSORFLOW FOR DEEP LEARNING
- KERAS - HIGH LEVEL LIBRARY WHICH MAKES GPU CODING TRANSPARENT - SIMPLIFIES THINGS A LOT AND MOST OF THE TIME ENOUGH FOR OUR APPLICATIONS

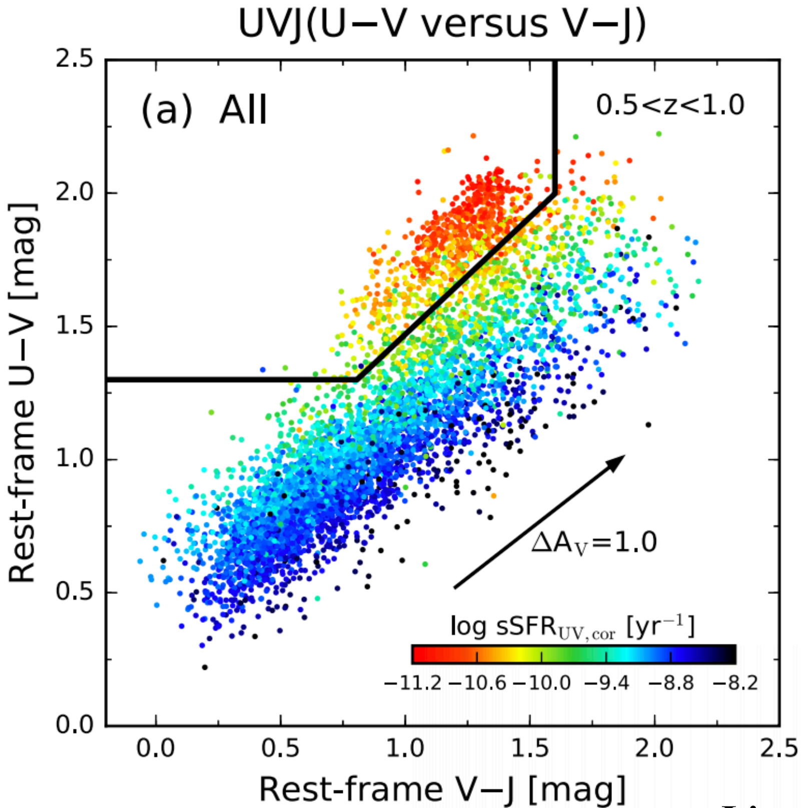
PART I: AN INTRODUCTION TO "CLASSICAL" MACHINE LEARNING

THRE IS NO MAGIC IN MACHINE LEARNING, AND IT IS ACTUALLY PRETTY SIMPLE



Liu+18

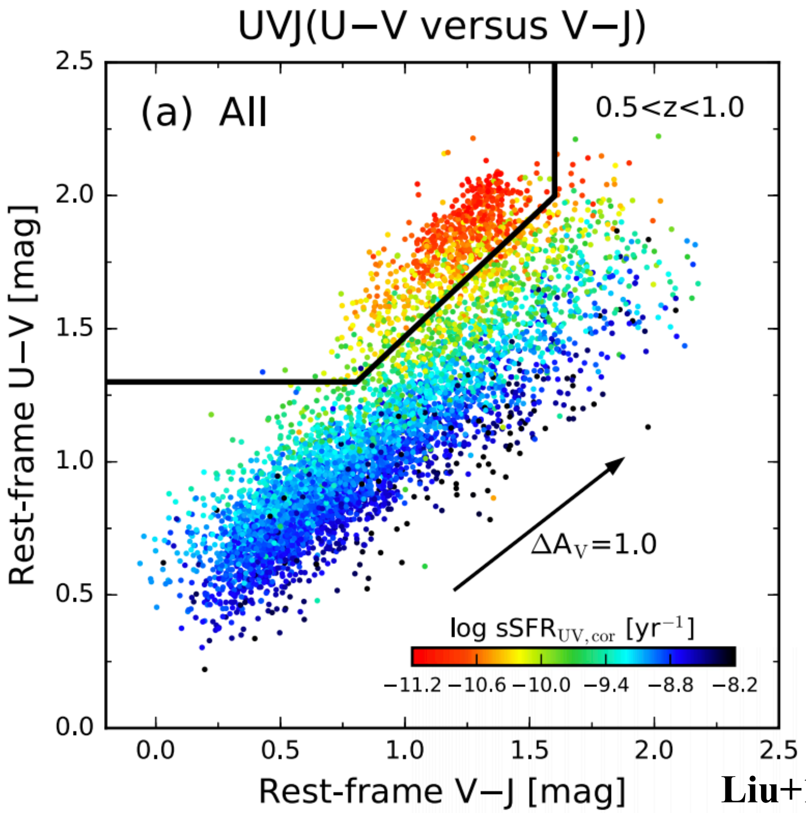
$$f_W(\vec{x}) = \vec{y}$$



Liu+18

$$f_W(\vec{x}) = \vec{y} \longrightarrow$$

LABEL
Q , SF



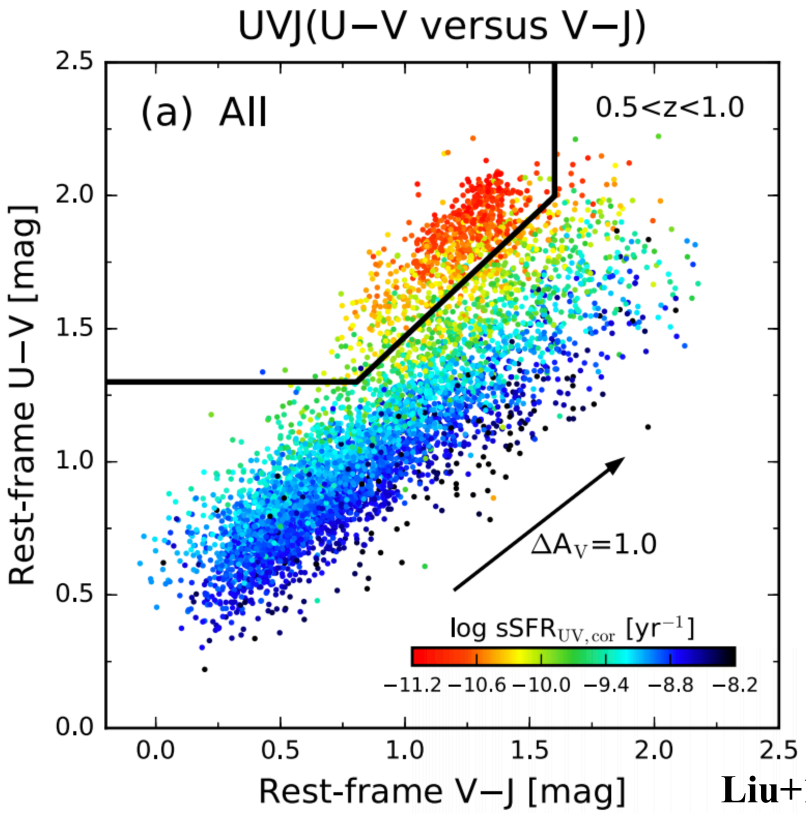
Liu+18

$$f_{\vec{W}}(\vec{x}) = \vec{y}$$

LABEL

Q(0) , SF(1)

(U-V, V-J) FEATURES



Liu+18

$$f_{\vec{W}}(\vec{x}) = \vec{y}$$

LABEL

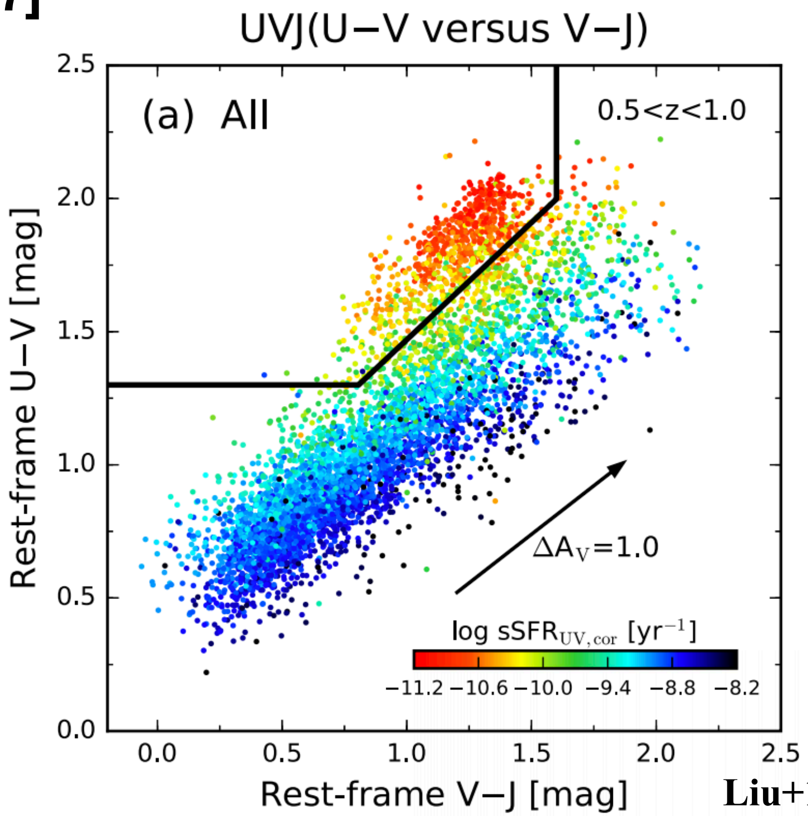
Q(0) , SF(1)

NETWORK FUNCTION

(U-V, V-J) FEATURES

sgn[(u-v)-0.8*(v-j)-0.7]

WEIGHTS



Liu+18

“CLASSICAL”
MACHINE LEARNING

$$f_{\overrightarrow{W}}(\vec{x}) = \vec{y} \longrightarrow$$

LABEL
Q , SF

sgn[(u-v)-W1*(v-j)-W2]

REPLACE THIS BY A GENERAL
NON LINEAR FUNCTION WITH SOME PARAMETERS W

WHAT DOES MACHINE LEARNING DO?

the machine is told what to look for

SUPERVISED

- Classification
- Regression

the machine is NOT told what to look for

UN-SUPERVISED

- Clustering
- Generative (deep learning)

WHAT DOES MACHINE LEARNING DO?

<u>the machine is told what to look for</u>

$\boxed{\text{SUPERVISED}}$

Classification

Regression

<u>[LECTURES BY BIEHL]</u>

<u>the machine is NOT told what to look for</u>

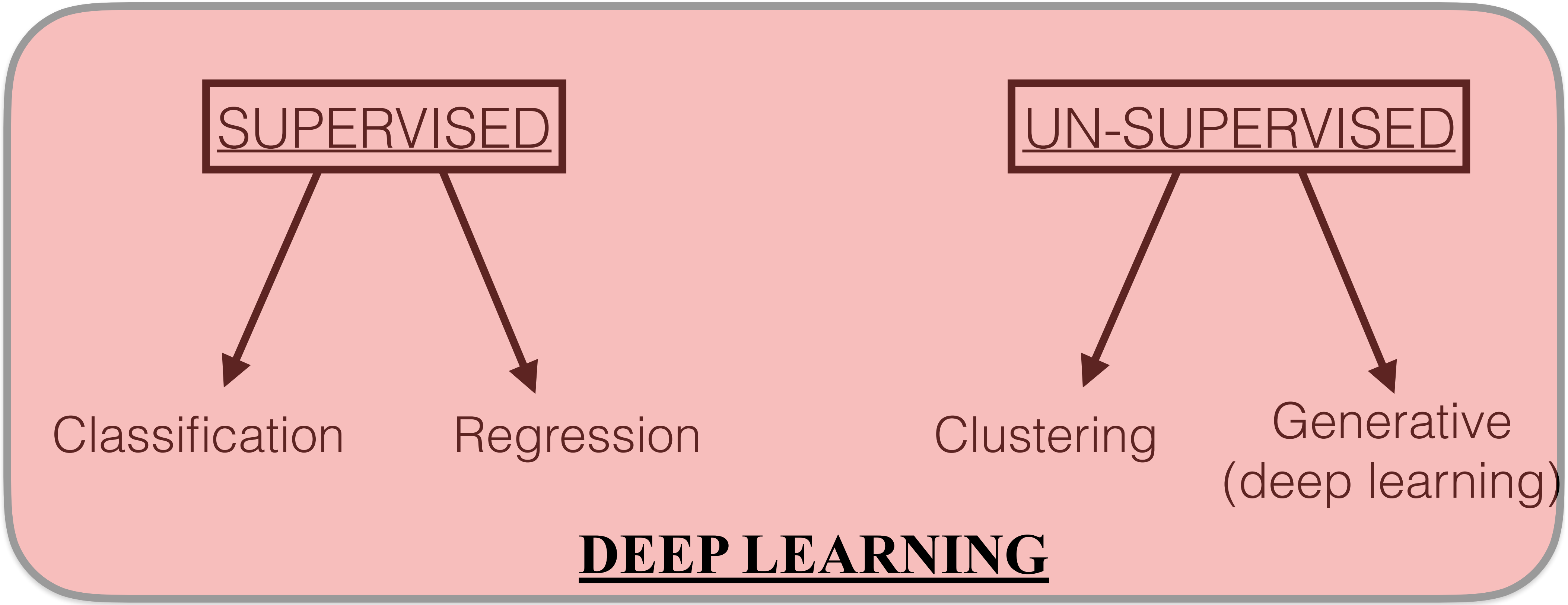
$\boxed{\text{UN-SUPERVISED}}$

Clustering

Generative
(deep learning)

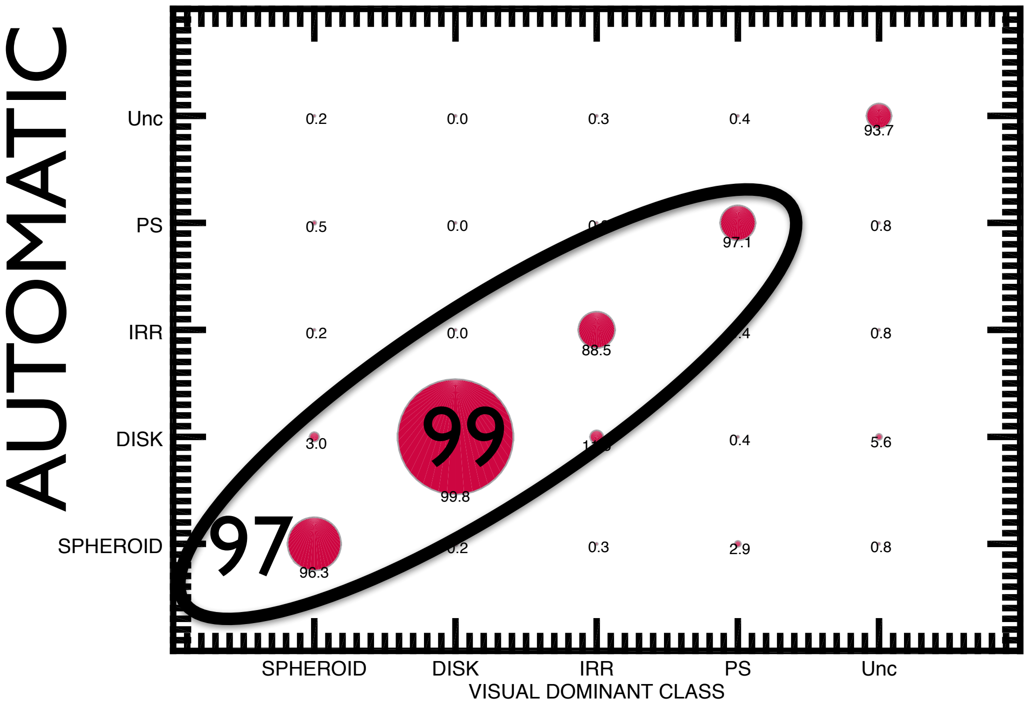
<u>[LECTURES BY BARON]</u>

WHAT DOES MACHINE LEARNING DO?



LET’S HAVE A LOOK AT SOME EXAMPLES OF DEEP LEARNING APPLIED…

"OUR CATS AND DOGS": GALAXY MORPHOLOGY




VISUAL

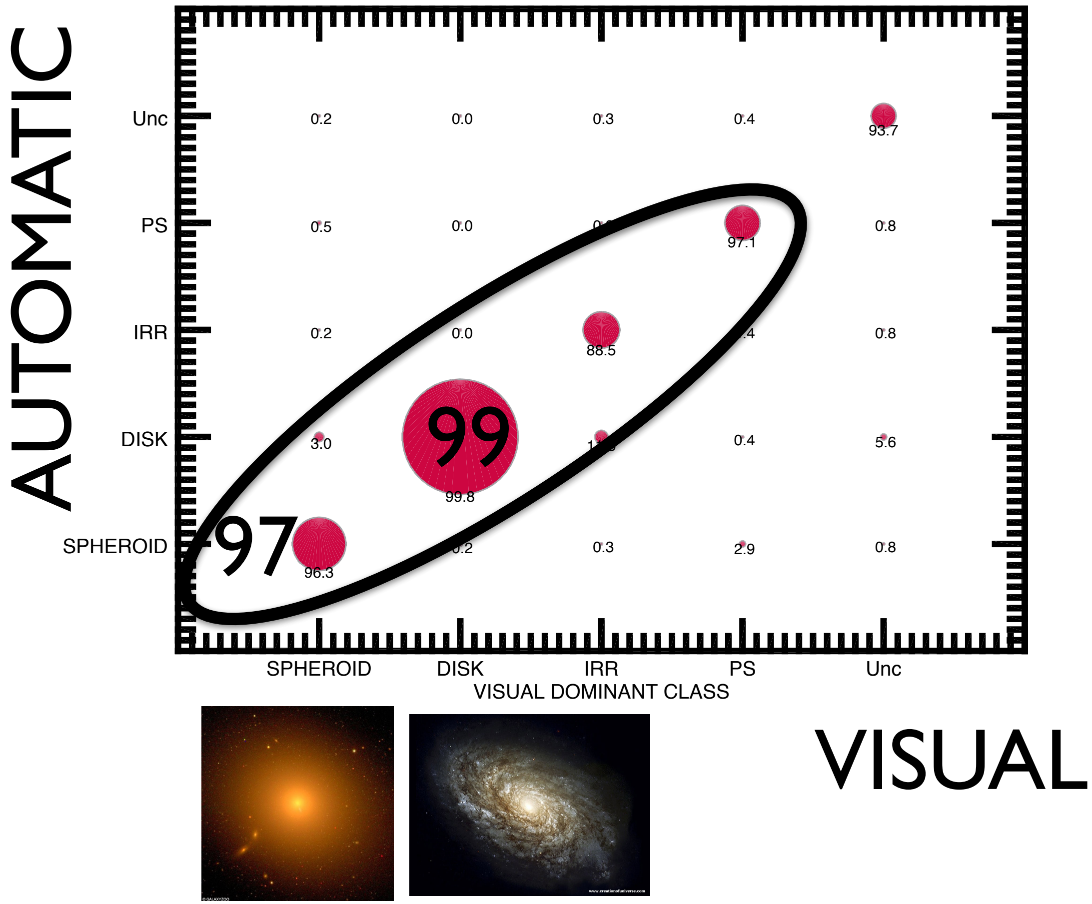


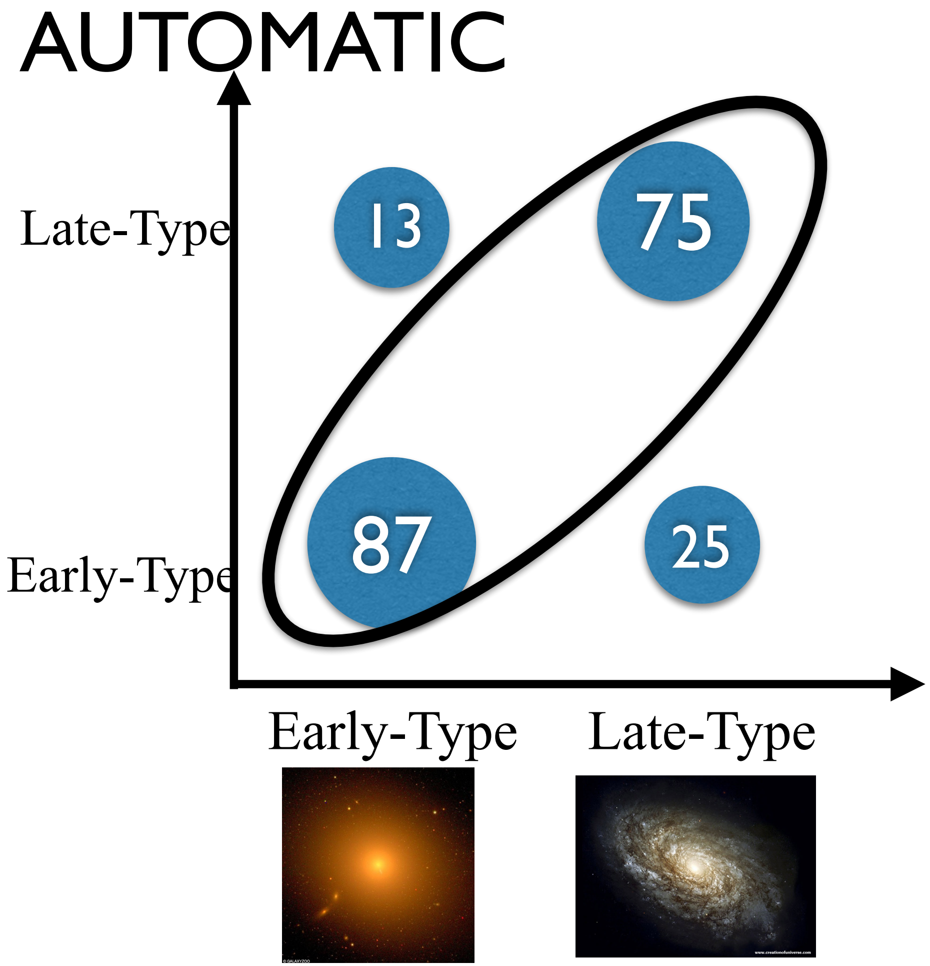

CNNs

DEEP LEARNING SOLVES THE PROBLEM OF GALAXY MORPHOLOGICAL CLASSIFICATION?

MHC+15b

“OUR CATS AND DOGS”: GALAXY MORPHOLOGY





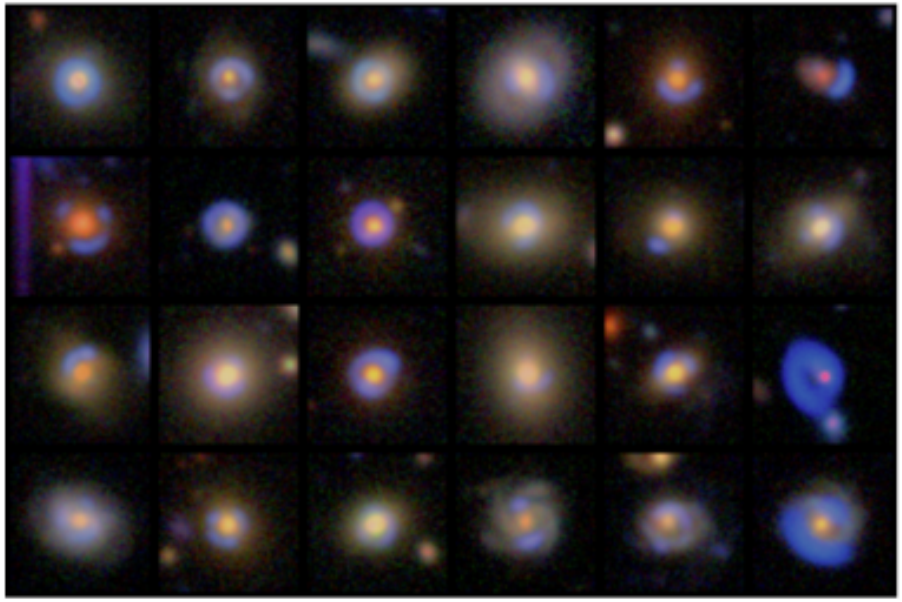
SVMs

CNNs

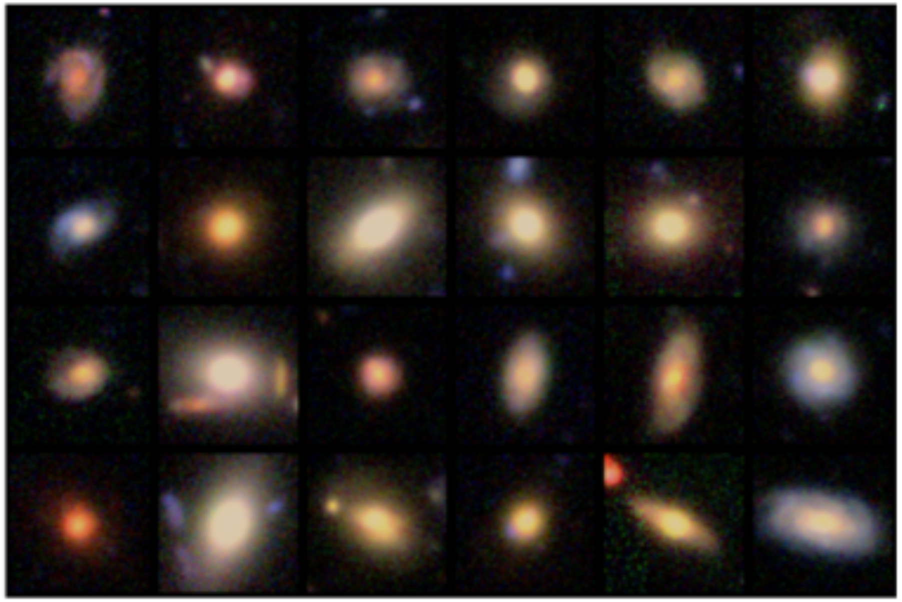
DEEP LEARNING SOLVES THE PROBLEM OF GALAXY MORPHOLOGICAL CLASSIFICATION?

MHC+15b

CLASSIFICATION: LENS FINDER



LENS



NON-LENS

Jacobs+17

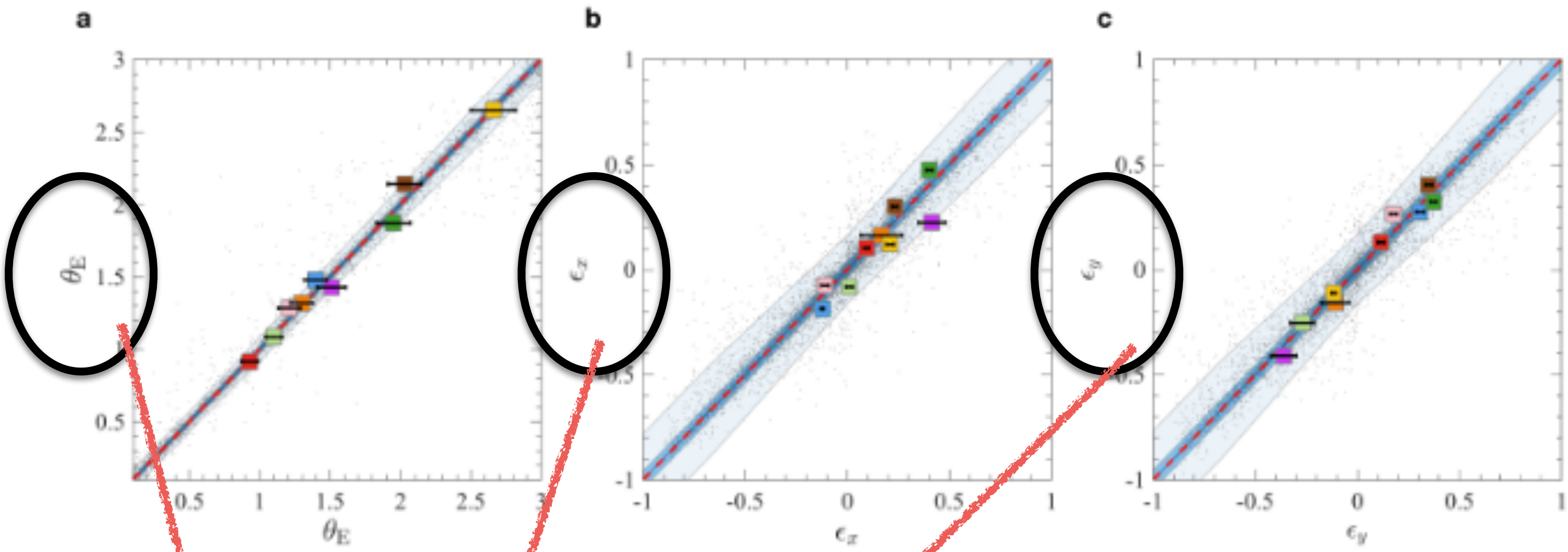
CLASSIFICATION: LENS FINDER

Name	type	AUROC	TPR_0	TPR_{10}	short description
CMU-DeepLens-ResNet-ground3	Ground-Based	0.98	0.09	0.46	CNN
CMU-DeepLens-Resnet-Voting	Ground-Based	0.98	0.02	0.10	CNN
LASTRO EPFL	Ground-Based	0.97	0.07	0.11	CNN
CAS Swinburne Melb	Ground-Based	0.96	0.02	0.08	CNN
AstrOmatic	Ground-Based	0.96	0.00	0.01	CNN
Manchester SVM	Ground-Based	0.93	0.22	0.35	SVM / Gabor
Manchester-NA2	Ground-Based	0.89	0.00	0.01	Human Inspection
ALL-star	Ground-Based	0.84	0.01	0.02	edges/gradiants and Logistic Reg.
CAST	Ground-Based	0.83	0.00	0.00	CNN / SVM
YattaLensLite	Ground-Based	0.82	0.00	0.00	SExtractor
LASTRO EPFL	Space-Based	0.93	0.00	0.08	CNN
CMU-DeepLens-ResNet	Space-Based	0.92	0.22	0.29	CNN
GAMOCLASS	Space-Based	0.92	0.07	0.36	CNN
CMU-DeepLens-Resnet-Voting	Space-Based	0.91	0.00	0.01	CNN
AstrOmatic	Space-Based	0.91	0.00	0.01	CNN
CMU-DeepLens-ResNet-aug	Space-Based	0.91	0.00	0.00	CNN
Kapteyn Resnet	Space-Based	0.82	0.00	0.00	CNN
CAST	Space-Based	0.81	0.07	0.12	CNN
Manchester1	Space-Based	0.81	0.01	0.17	Human Inspection
Manchester SVM	Space-Based	0.81	0.03	0.08	SVM / Gabor
NeuralNet2	Space-Based	0.76	0.00	0.00	CNN / wavelets
YattaLensLite	Space-Based	0.76	0.00	0.00	Arcs / SExtractor
All-now	Space-Based	0.73	0.05	0.07	edges/gradiants and Logistic Reg.
GAHEC IRAP	Space-Based	0.66	0.00	0.01	arc finder

Jacobs+17

Metcalf+18

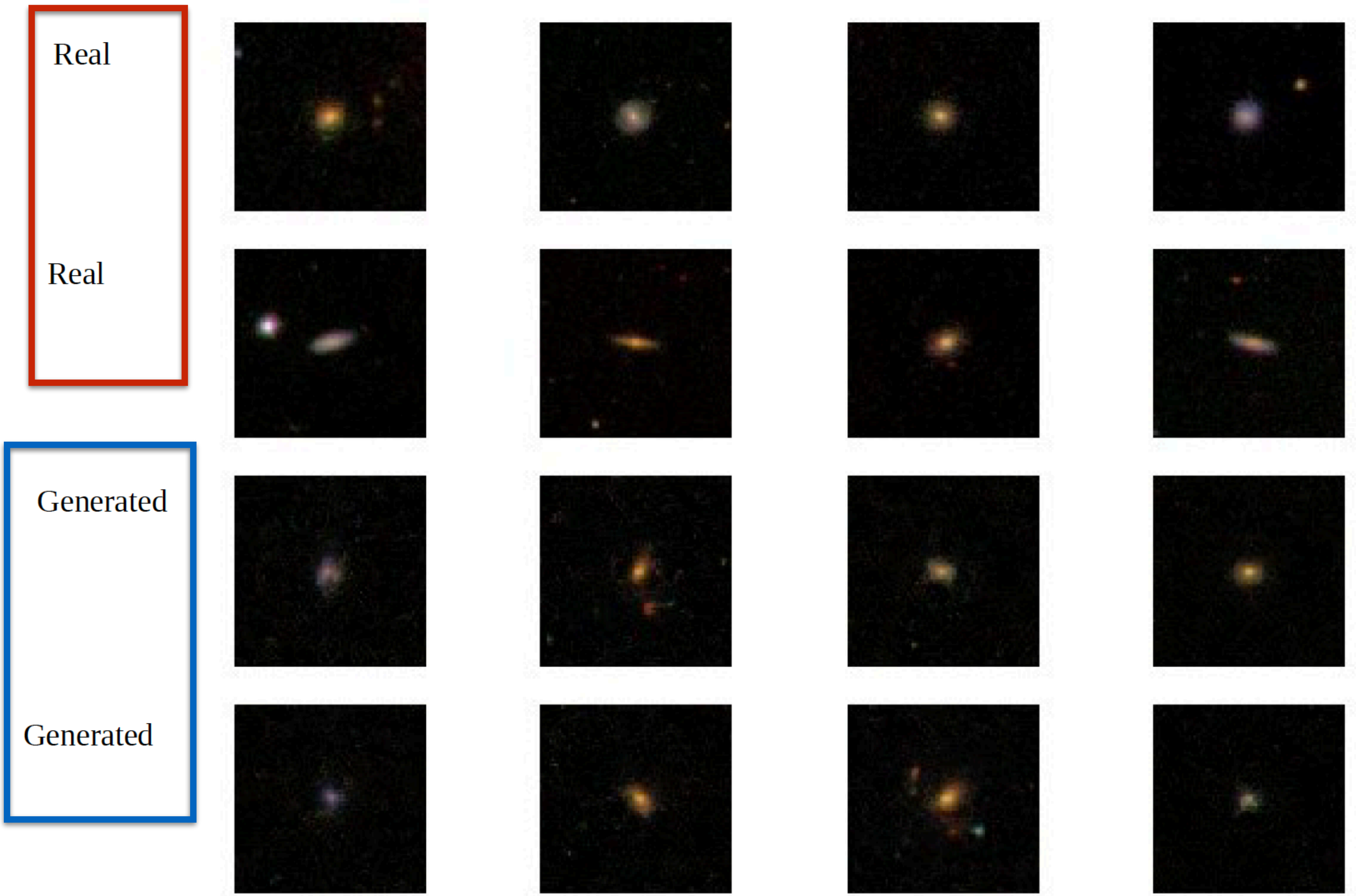
REGRESSION



Hezaveh+17, Nature

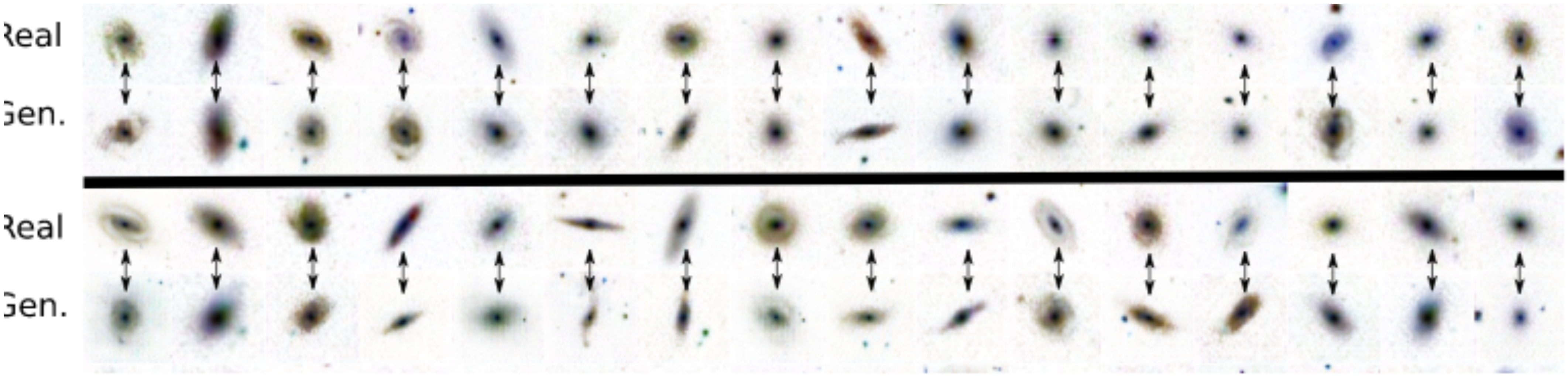
REGRESSION ON
STRONG LENSES PARAMETERS

GENERATIVE MODELS
(UNSUPERVISED)



Margalef,MHC+19

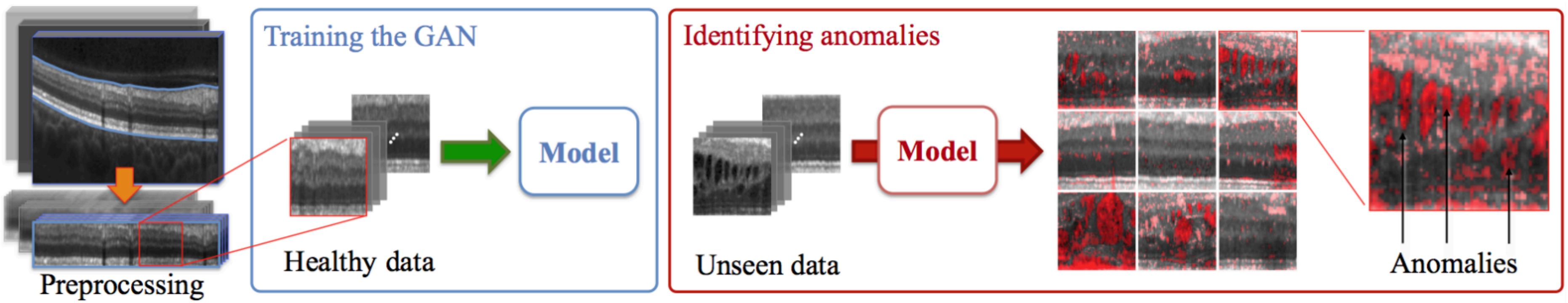
GENERATIVE MODELS
(UNSUPERVISED)



Generation of realistic galaxy images

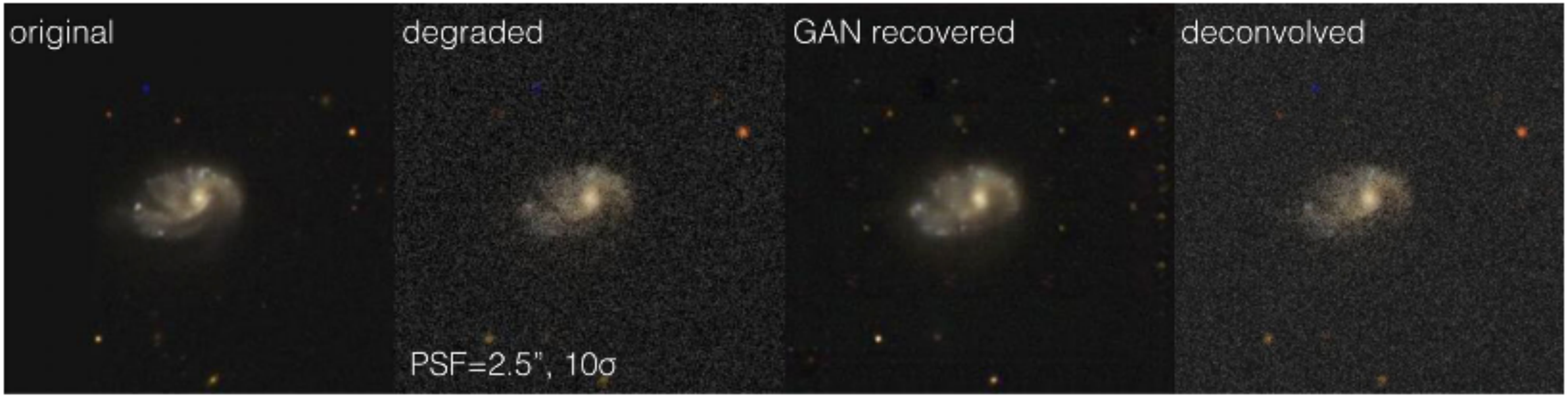
Ravanbakhsh+16

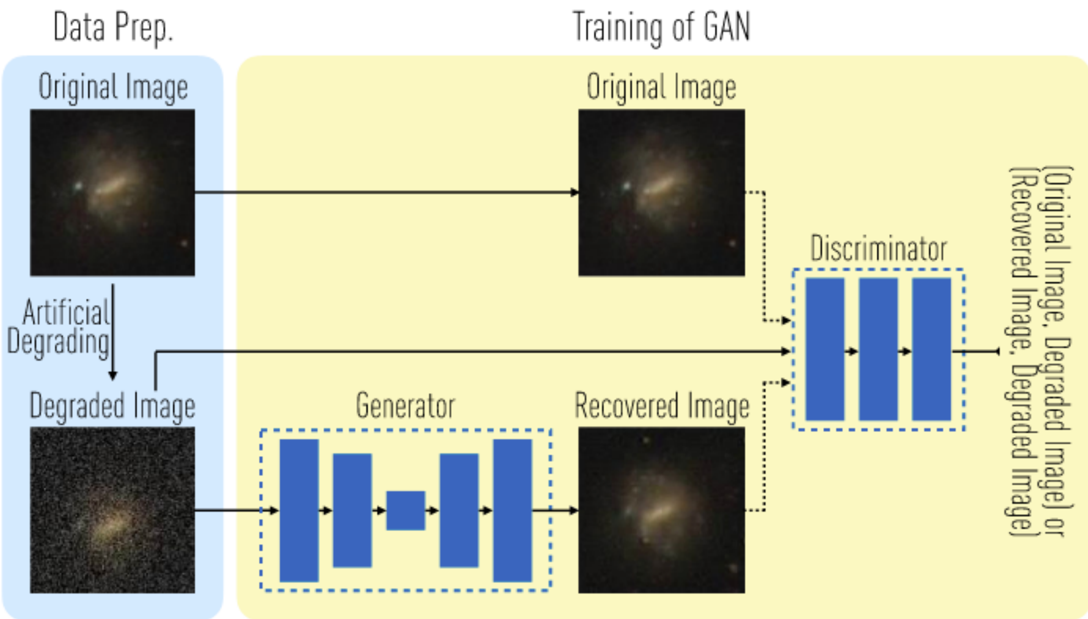
GENERATIVE MODELS TO BOOST DISCOVERY



Schlegl+17

GENERATIVE MODELS (UNSUPERVISED)





Schawinsky+17

SUPERVISED LEARNING

Given a dataset with <u>known labels</u> (measurements) - find a function that can assign (predict) measurements for an unlabeled dataset

<u>Training set</u>

$$(\vec{x_1}, \vec{x_2}, \vec{x_3}, ..., \vec{x_n})$$

<u>Measurements</u>
<u>(colors, fluxes, spectra indices…)</u>

$$(\vec{y_1}, \vec{y_2}, \vec{y_3}, ..., \vec{y_n})$$

<u>Label</u>
<u>(morphology, object type, transit …)</u>

SUPERVISED LEARNING

Given a dataset with known labels (measurements) - find a function that can assign (predict) measurements for an unlabeled dataset

Training set

$$(\vec{x_1}, \vec{x_2}, \vec{x_3}, \ldots, \vec{x_n})$$

$$(\vec{y_1}, \vec{y_2}, \vec{y_3}, \ldots, \vec{y_n})$$

$$\rightarrow \quad f_W(\vec{x}) = \vec{y}$$

?

SUPERVISED LEARNING

Unlabeled set

$$(\vec{x_1}', \vec{x_2}', \vec{x_3}', ..., \vec{x_n}')$$

$$(\vec{y_1}', \vec{y_2}', \vec{y_3}', ..., \vec{y_n}')$$

Training set

$$(\vec{x_1}, \vec{x_2}, \vec{x_3}, ..., \vec{x_n})$$

$$(\vec{y_1}, \vec{y_2}, \vec{y_3}, ..., \vec{y_n})$$

$$f_W(\vec{x}) = \vec{y}$$

?

$$(\vec{x_1}, \vec{x_2}, \vec{x_3}, ..., \vec{x_n})$$ $\vec{x} \in \mathbb{R}^d$

$$(\vec{y_1}, \vec{y_2}, \vec{y_3}, ..., \vec{y_n})$$ $\vec{y} \in \mathbb{R}$ $\vec{y} \in \mathbb{N}$

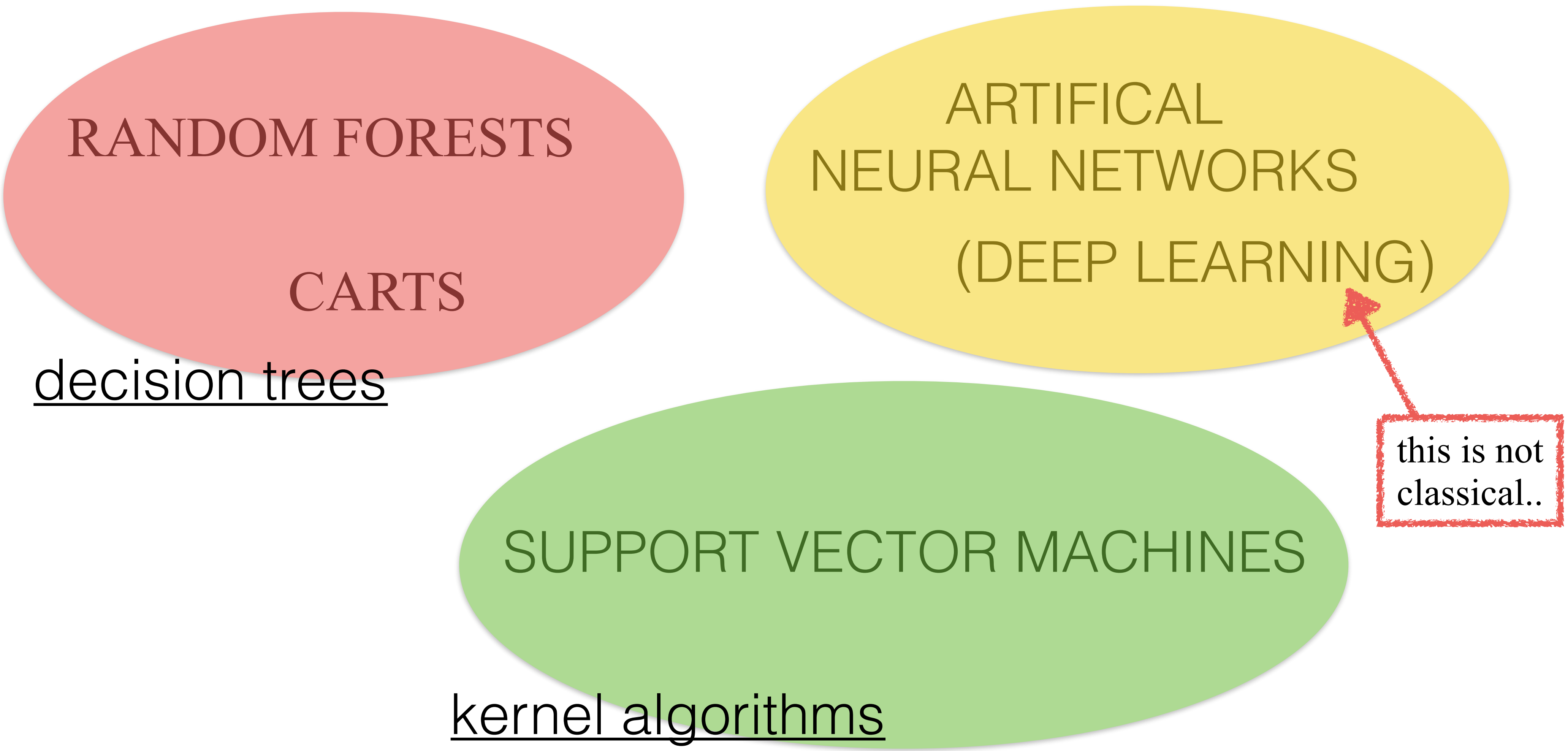
GENERAL GOAL: Find a (non-linear) function that outputs the correct class / measurement for a given input object:

$$f_W(\vec{x})$$

Number of parameters - can be large

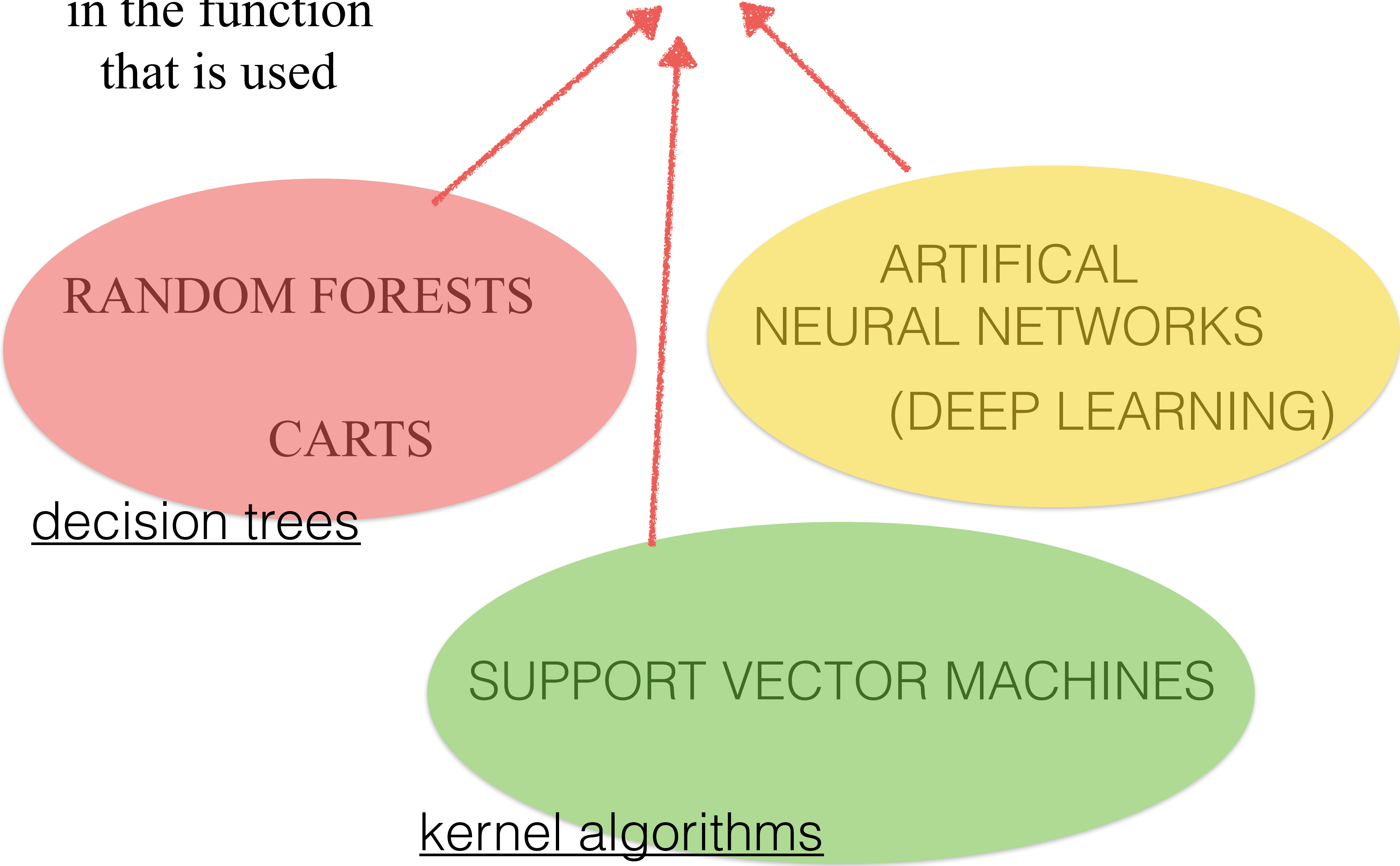
It is translated into a minimization problem : find W such as the prediction error is minimal <u>over all unseen vectors</u>

Different "classical" supervised machine learning methods



$$f_W(\vec{x})$$

The differences are in the function that is used



We need two key elements

1. A LOSS FUNCTION

2. A MINIMIZATION OR OPTIMIZATION ALGORITHM

We need two key elements

1. A LOSS FUNCTION

2. A MINIMIZATION OR OPTIMIZATION ALGORITHM

THIS IS COMMON TO ALL MACHINE LEARNING ALGORITHMS

1. DEFINE A LOSS FUNCTION

$$loss(F_W(.), \vec{x_i}, \vec{y_i})$$

For example: $(F_W(\vec{x_i}) - \vec{y_i})^2$ Quadratic loss function

2. MINIMIZE THE EMPIRICAL RISK

$$\Re_{empirical}(W) = \frac{1}{N}\sum_{i}^{N}[loss(W, \vec{x}, \vec{y})]$$

MINIMIZE THE RISK

EMPIRICAL RISK?

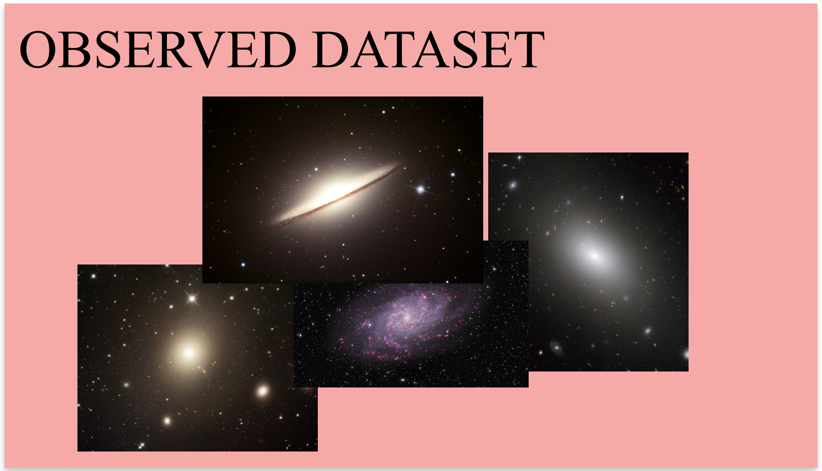
$$\Re_{empirical}(W) = \frac{1}{N}\sum_{i}^{N}[loss(W, \vec{x}, \vec{y})]$$

WE ARE MINIMIZING WITH RESPECT TO A FINITE NUMBER OF OBSERVED EXAMPLES

EMPIRICAL RISK?

$$\Re_{empirical}(W) = \frac{1}{N}\sum_{i}^{N}[loss(W,\vec{x},\vec{y})]$$

WE ARE MINIMIZING WITH RESPECT TO A FINITE NUMBER OF OBSERVED EXAMPLES



EMPIRICAL RISK?

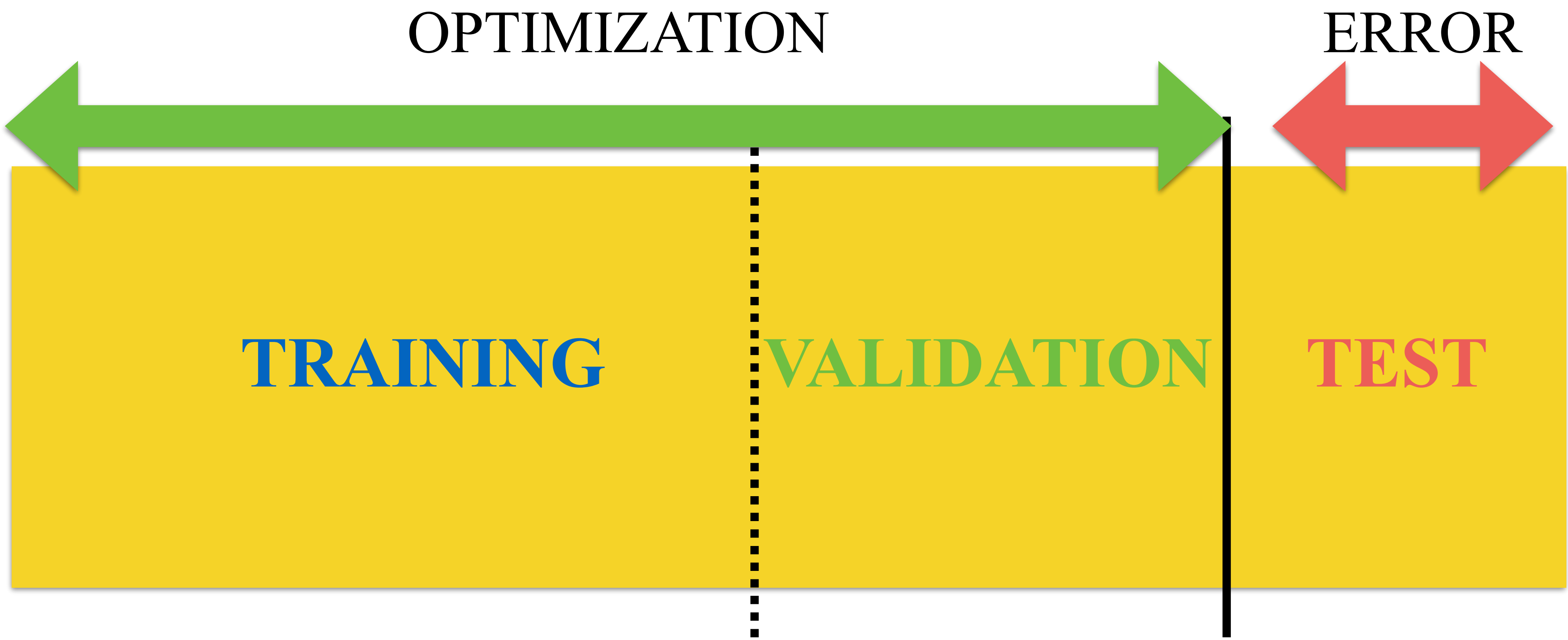
$$\Re_{empirical}(W) = \frac{1}{N}\sum_{i}^{N}[loss(W, \vec{x}, \vec{y})]$$

WE ARE MINIMIZING WITH RESPECT TO A FINITE NUMBER OF OBSERVED EXAMPLES

ALL "GALAXIES IN THE UNIVERSE"

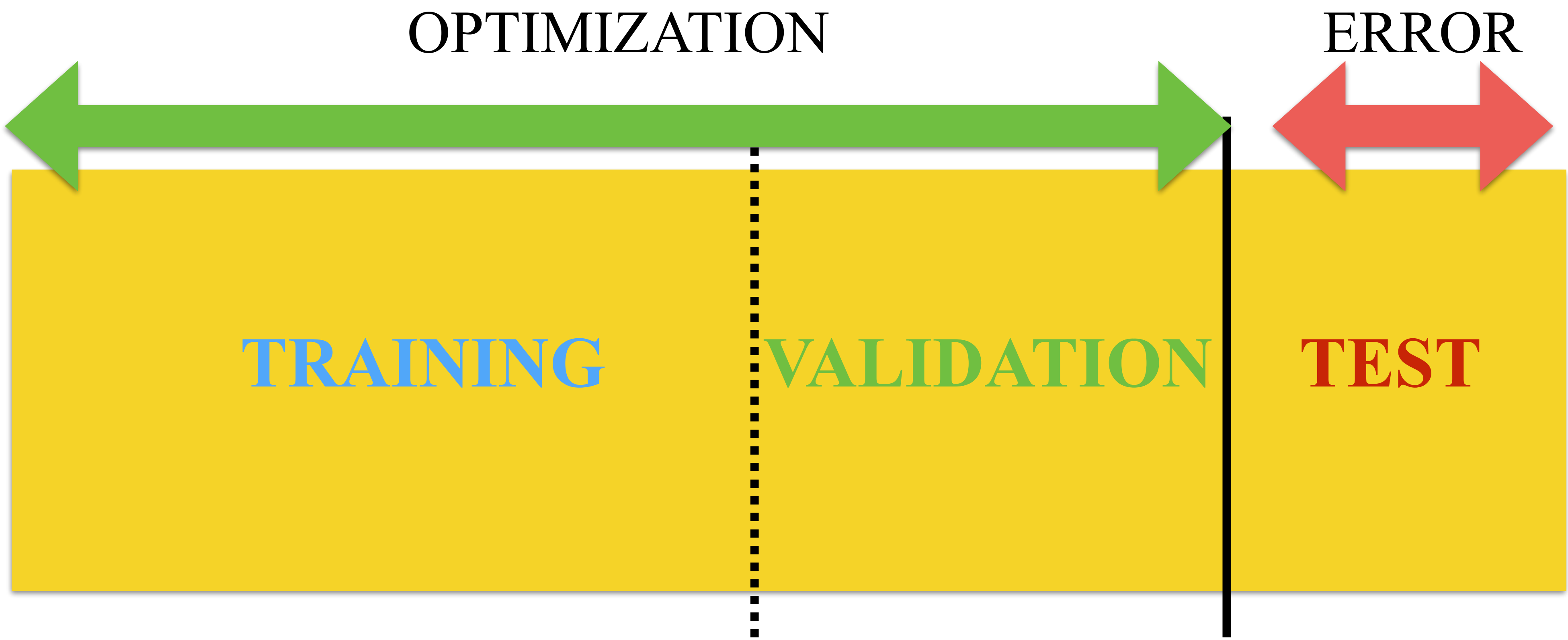
OBSERVED DATASET

In practice



training set: use to train the classifier
validation set: use to monitor performance in real time - check for overfitting
test set: use to train the classifier

In practice



NO CHEATING! NEVER USE TRAINING TO VALIDATE YOUR ALGORITHM!

The algorithm used to minimize is called OPTIMIZATION

THERE ARE SEVERAL OPTIMIZATION TECHNIQUES

Optimization

THERE ARE SEVERAL OPTIMIZATION TECHNIQUES

THEY DEPEND ON THE MACHINE LEARNING ALGORITHM

Optimization

THERE ARE SEVERAL OPTIMIZATION TECHNIQUES

THEY DEPEND ON THE MACHINE LEARNING ALGORITHM

NEURAL NETWORKS USE THE GRADIENT DESCENT AS WE WILL SEE LATER

$$W_{t+1} = W_t - \lambda_h \triangledown f(W_t)$$

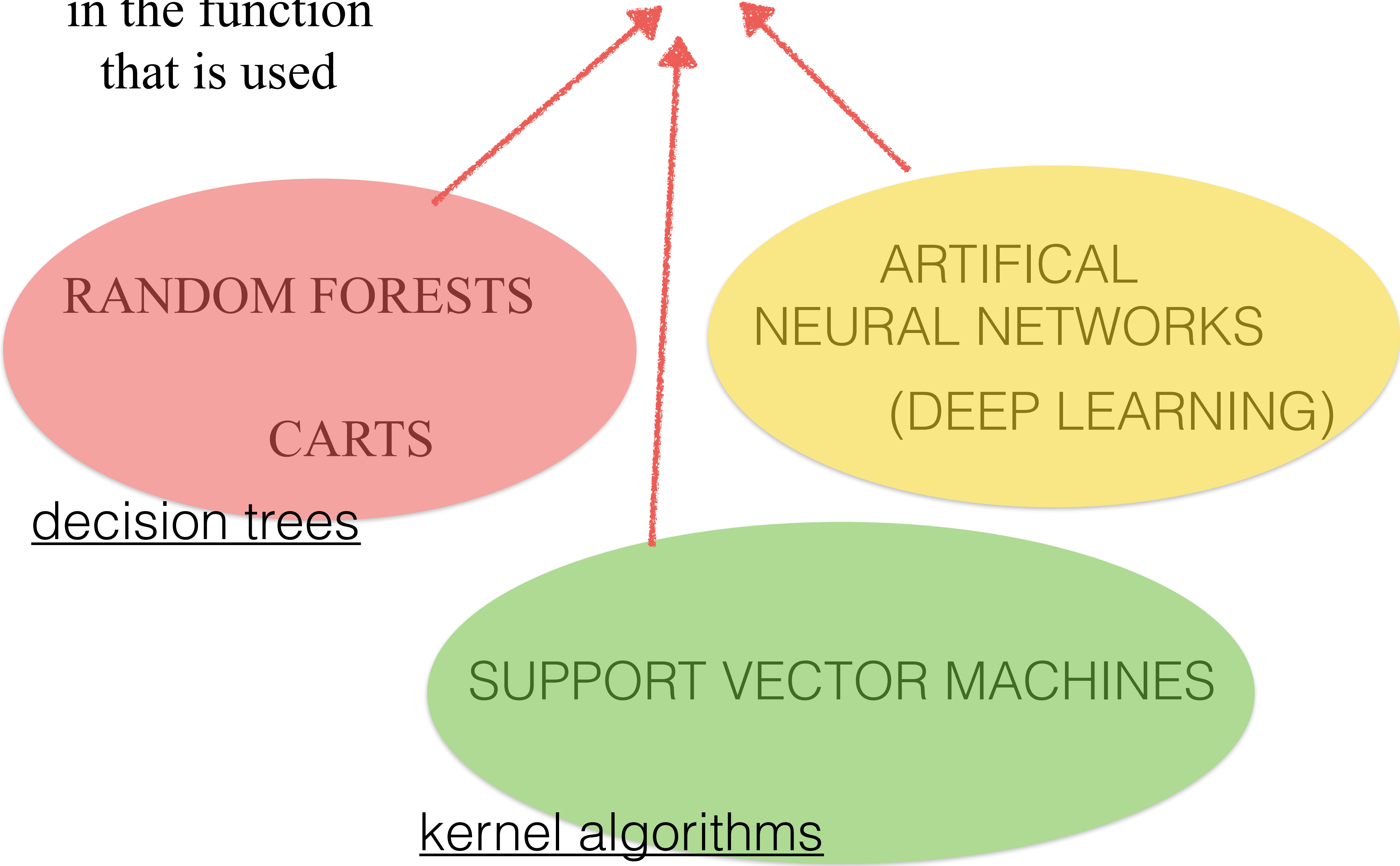
weights to be learned

epoch

learning rate

$$f_W(\vec{x})$$

The differences are
in the function
that is used

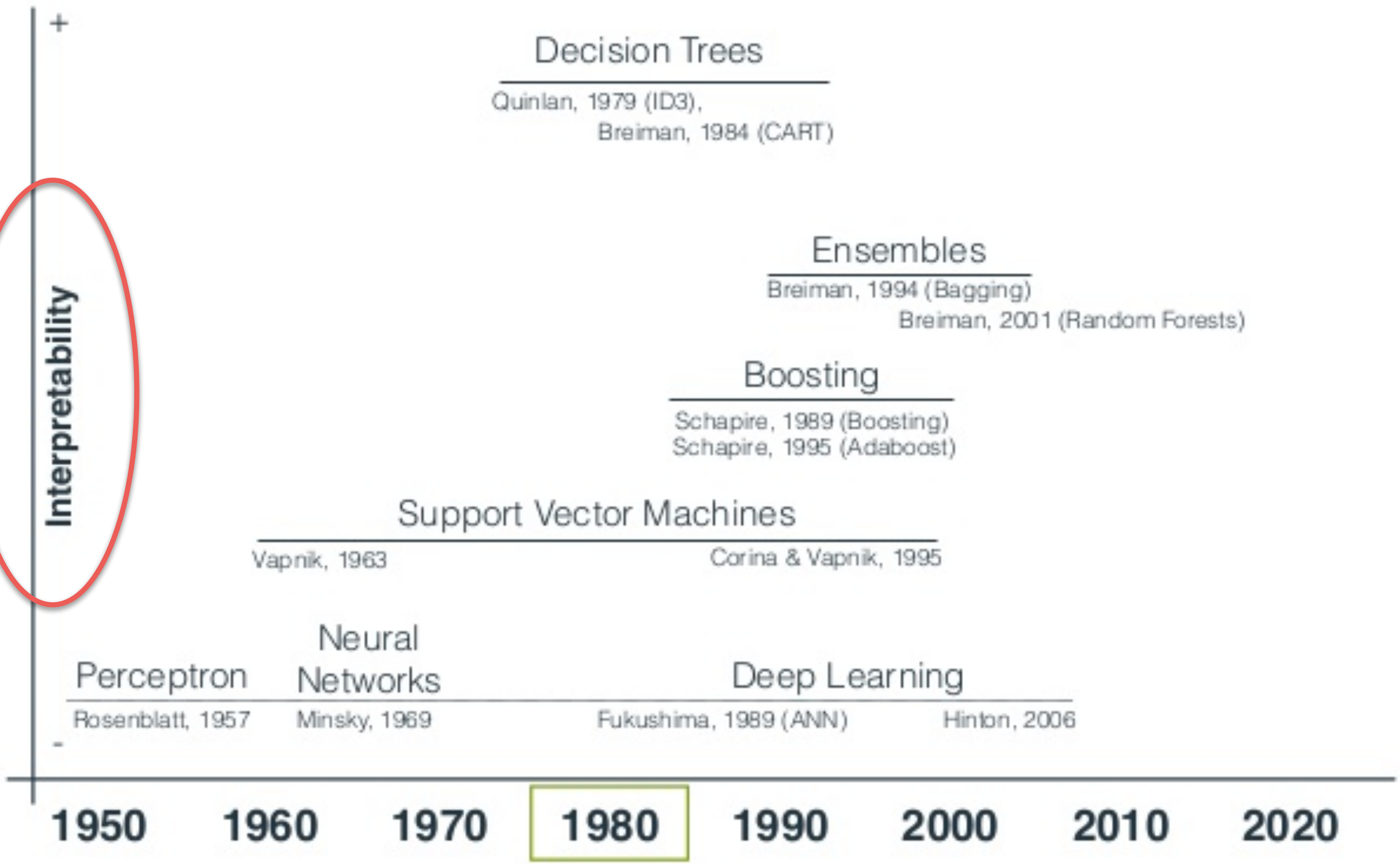


HOW TO CHOOSE YOUR CLASSICAL CLASSIFIER?

NO RULE OF THUMB - REALLY DEPENDS ON APPLICATION

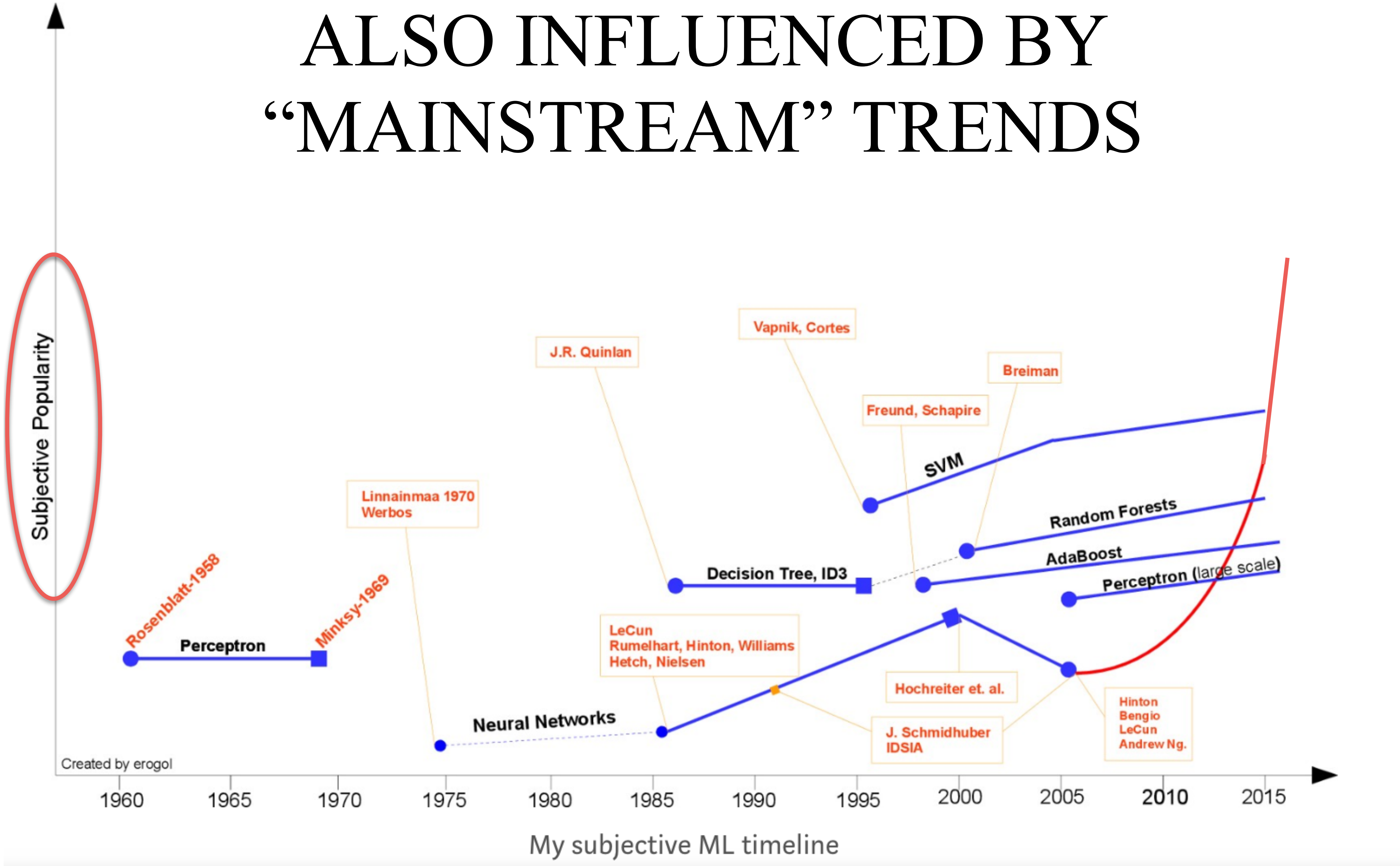
ML METHOD	++	—	Python
CARTS / RANDOM FOREST	Easy to interpret ("White box") Litte data preparation Both numerical + categorical	Over-complex trees Unstable Biased tress if some classes dominate	sklearn.ensemble.RandomForestClassifier sklearn.ensemble.RandomForestRegressor
SVM	Easy to interpret + Fast Kernel trick allows no linear problems	not very well suited to multi-class problems	sklearn.svm sklearn.svc
NN	seed of deep-learning very efficient with large amount of data as we will see	more difficult to interpret computing intensive	sklearn.neural_network.MPL_CLassifier sklearn.neural_network.MPL_Regressor

CAN DEPEND ON YOUR MAIN INTEREST



credit

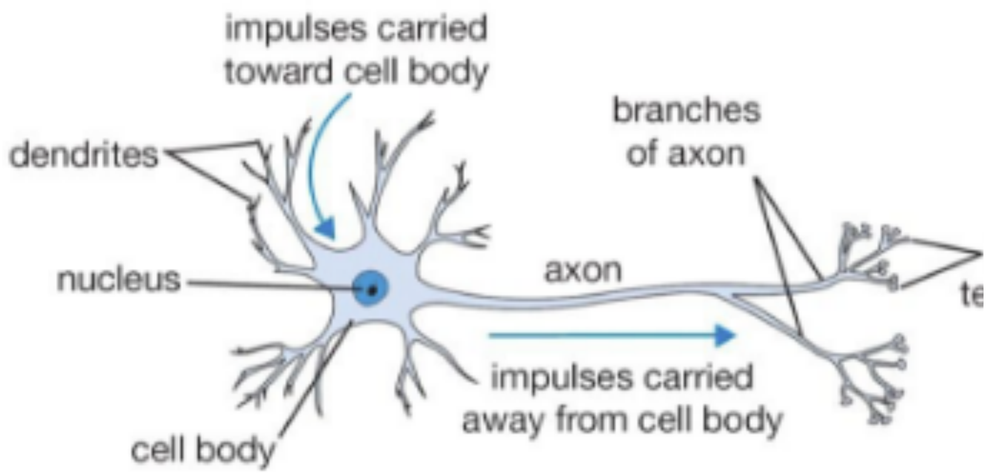
ALSO INFLUENCED BY "MAINSTREAM" TRENDS



Source

PART II: A FOCUS ON "SHALLOW" NEURAL NETWORKS

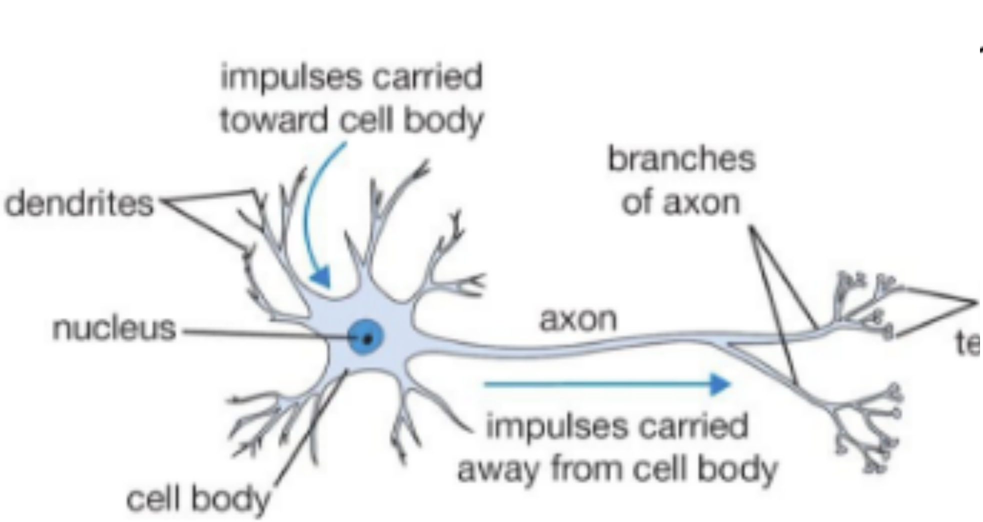
THE NEURON

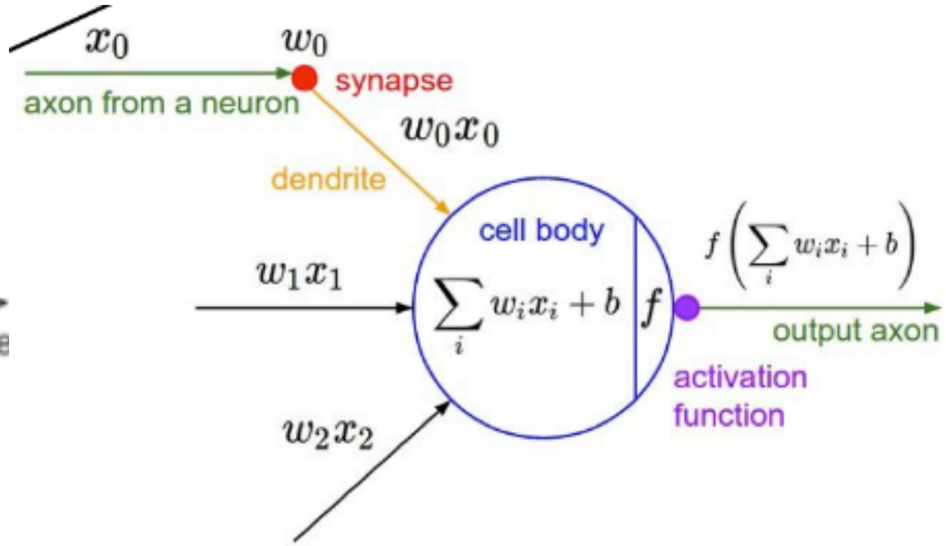


INSPIRED BY NEURO - SCIENCE?

Credit: Karpathy

THE NEURON





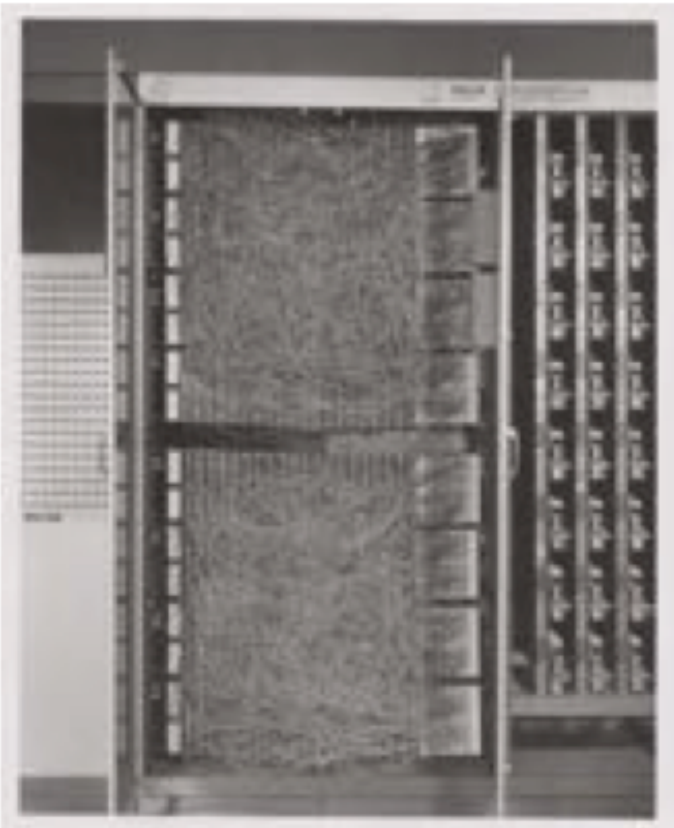
INSPIRED BY NEURO - SCIENCE?

Credit: Karpathy

Mark I Perceptron

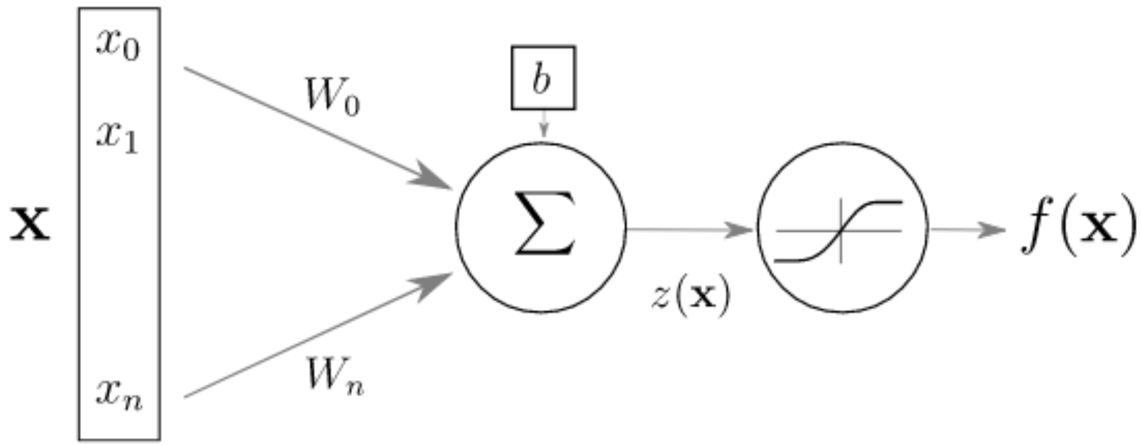
FIRST IMPLEMENTATION OF NEURAL NETWORK [Rosenblatt, **1957!**]

INTENDED TO BE A MACHINE (NOT AN ALGORITHM)



it had an array of 400 photocells, randomly connected to the "neurons". Weights were encoded in potentiometers, and weight updates during learning were performed by electric motors

TODAY'S ARTIFICIAL NEURON



$$z(\vec{x}) = \vec{W}.\vec{x} + b \qquad f(\vec{x}) = g(\vec{W}.\vec{x} + b)$$

Pre-Activation

Weights

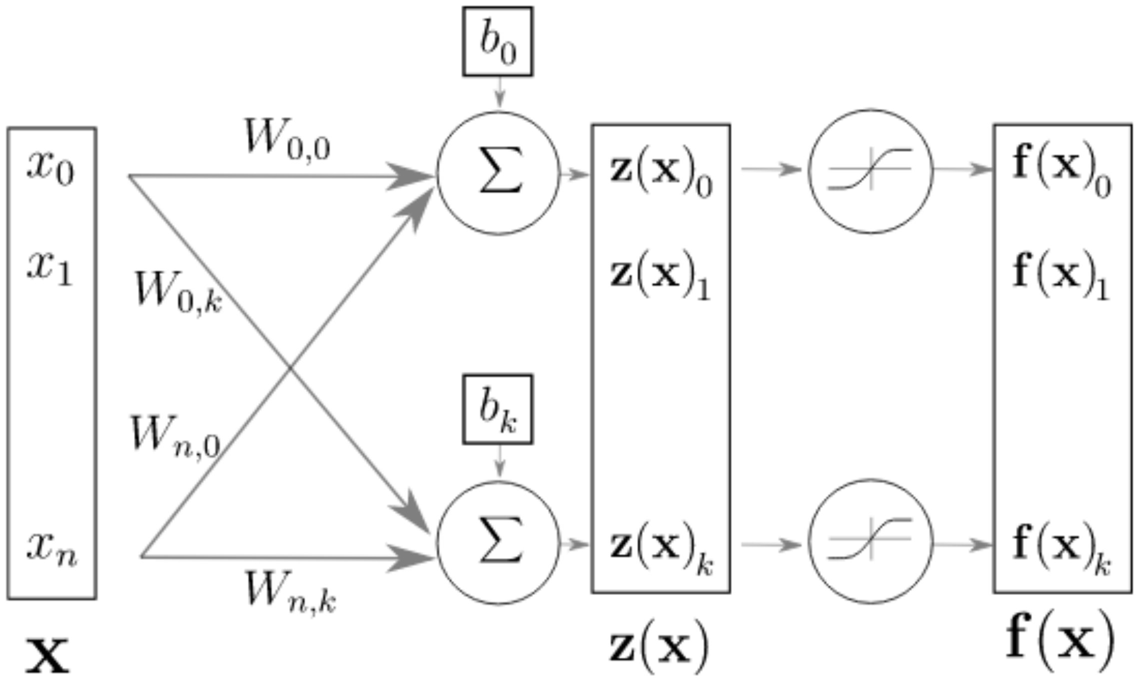
Bias

Activation Function

Input

Output

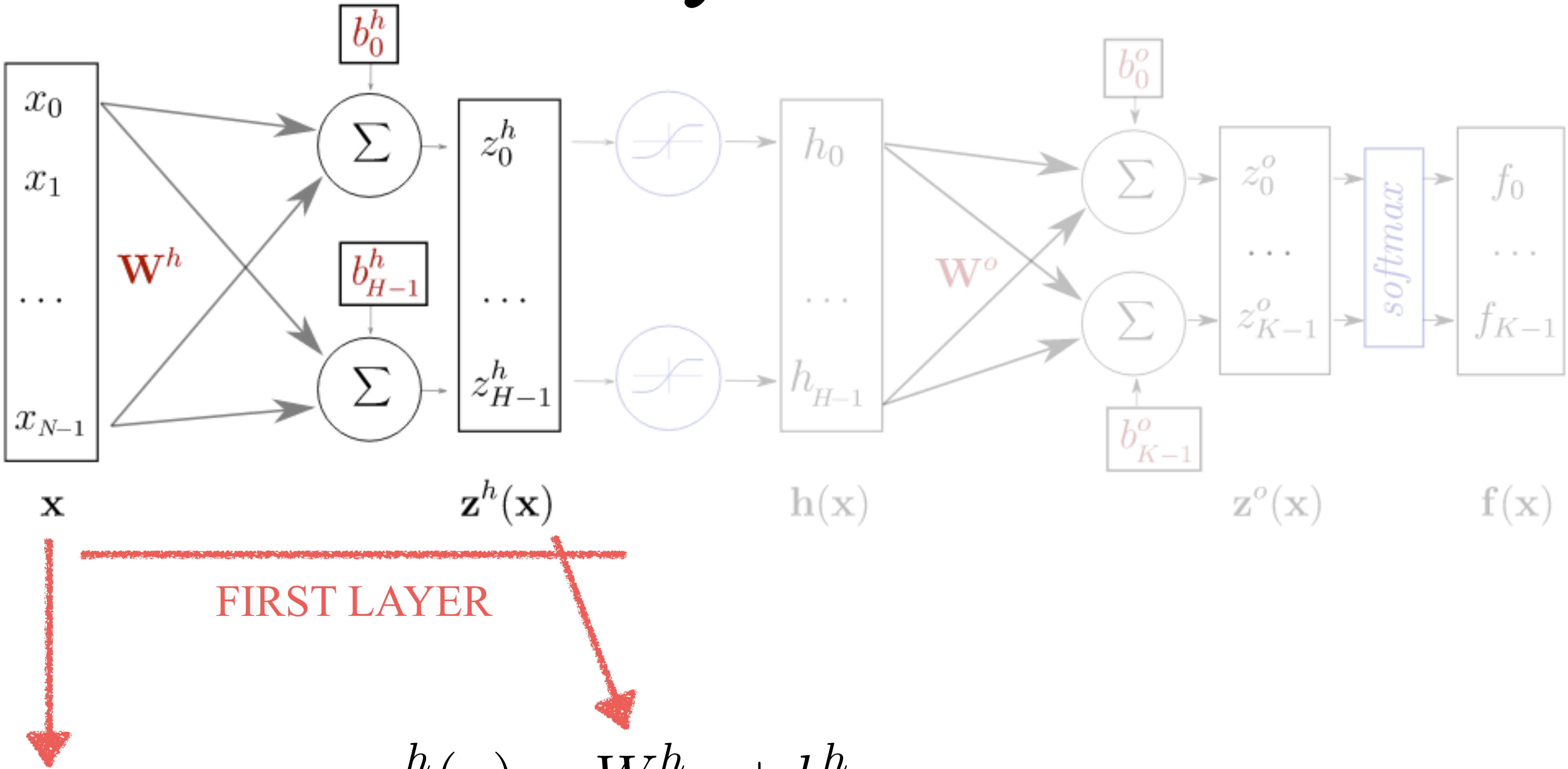
LAYER OF NEURONS



$$f(\vec{x}) = g(\mathbf{W}.\vec{x} + \vec{b})$$

SAME IDEA. NOW **W** becomes a matrix and **b** a vector

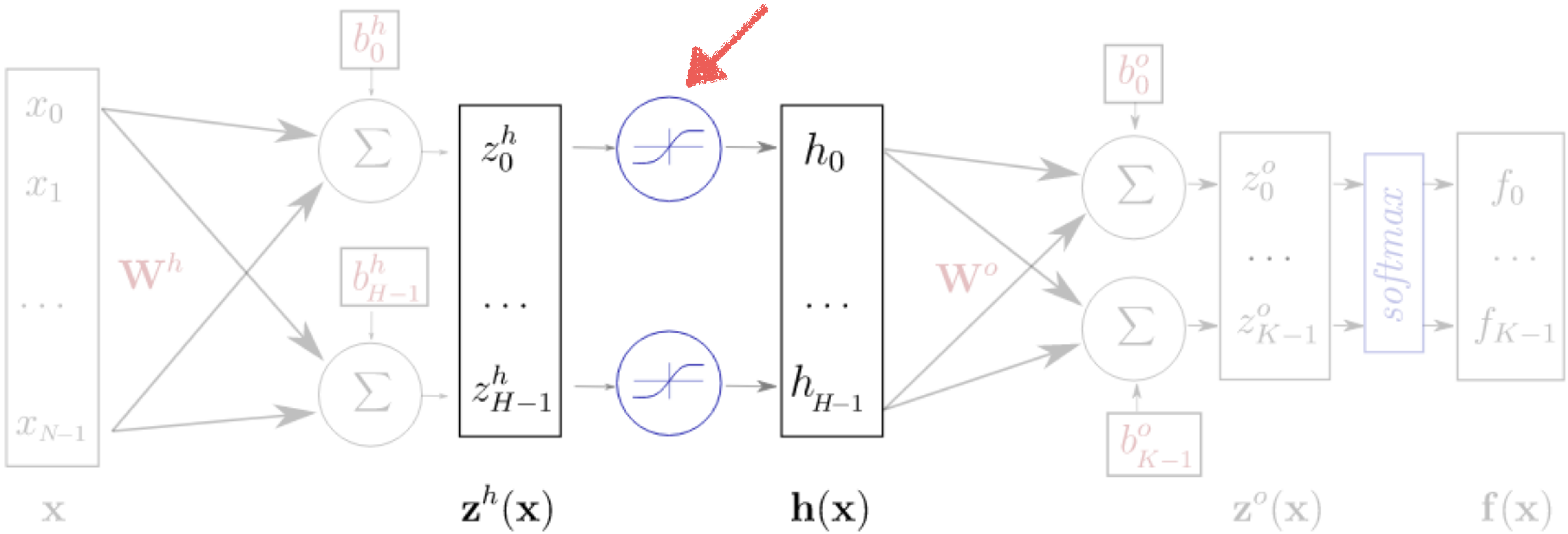
Hidden Layers of Neurons



INPUT

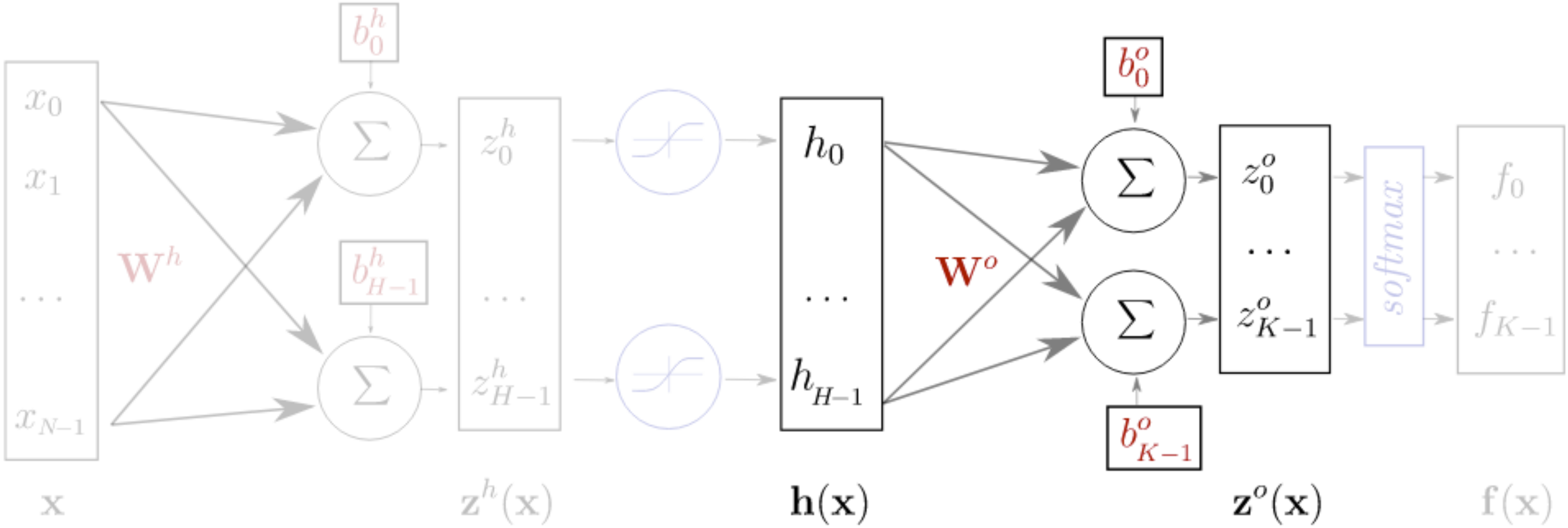
$$z^h(x) = W^h x + b^h$$

ACTIVATION FUNCTION



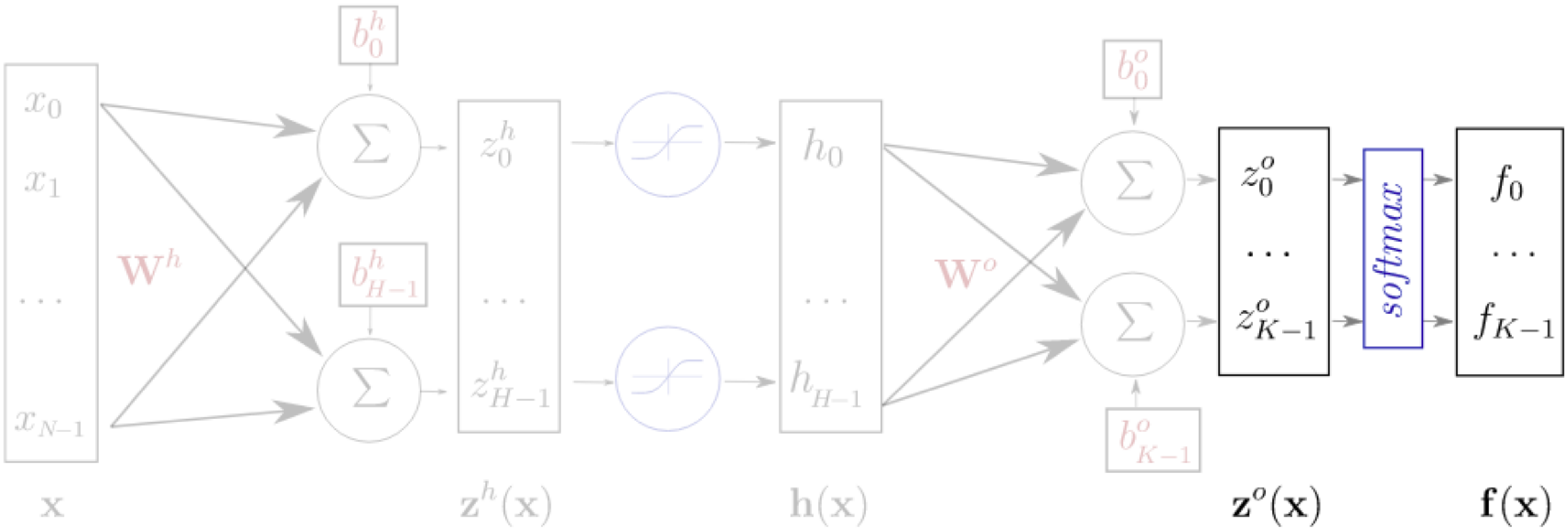
HIDDEN LAYER

$$h(x) = g(z^h(x)) = g(W^h x + b^h)$$



OUTPUT LAYER

$$z^0(\mathbf{x}) = W^0 h(\mathbf{x}) + b^0$$



PREDICTION LAYER

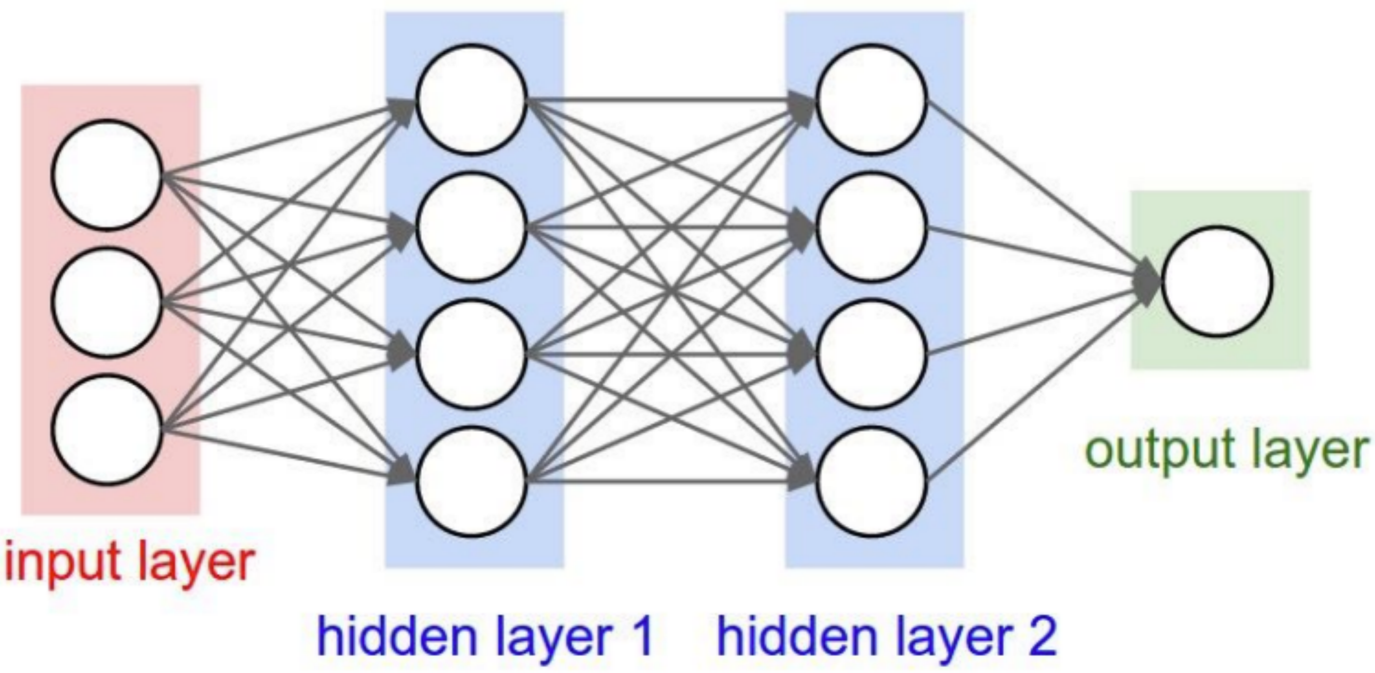
$$f(\mathbf{x}) = softmax(\mathbf{z^0})$$

“CLASSICAL” MACHINE LEARNING

$$f_{\vec{W}}(\vec{x}) = \vec{y} \longrightarrow$$

LABEL
Q , SF

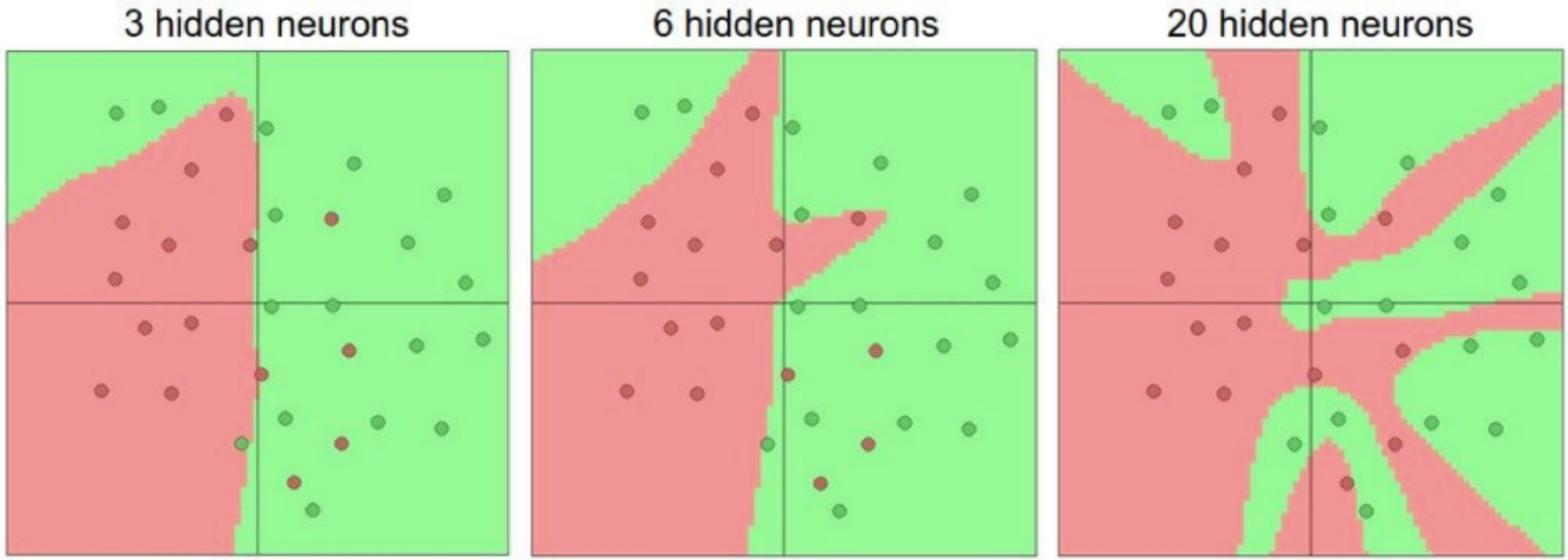
REPLACE THIS BY A GENERAL NON LINEAR FUNCTION WITH SOME PARAMETERS W



$$p = g_3(W_3 g_2(W_2 g_1(W_1 \vec{x_0})))$$

NETWORK FUNCTION

WHY HIDDEN LAYERS?



<u>More complex functions allow increasing complexity</u>

Credit: Karpathy

SO LET’S GO DEEPER AND DEEPER!

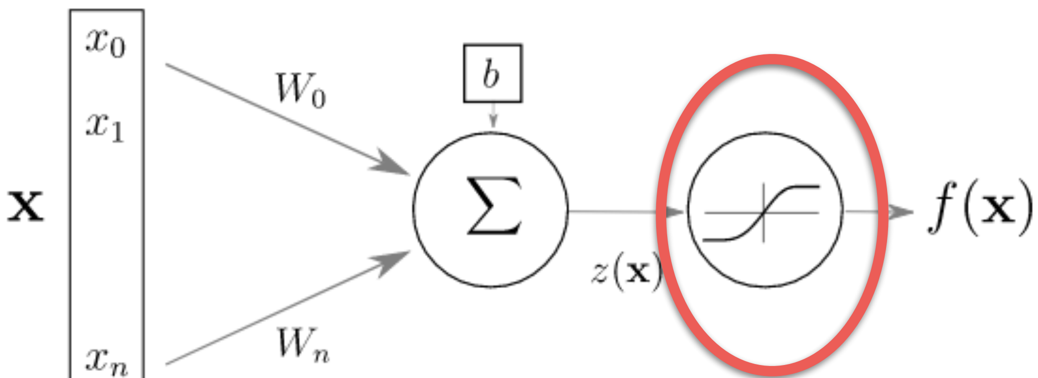
SO LET’S GO DEEPER AND DEEPER!

YES BUT…

NOT SO STRAIGHTFORWARD, DEEPER MEANS MORE WEIGHTS, MORE DIFFICULT OPTIMIZATION, RISK OF OVERFITTING…

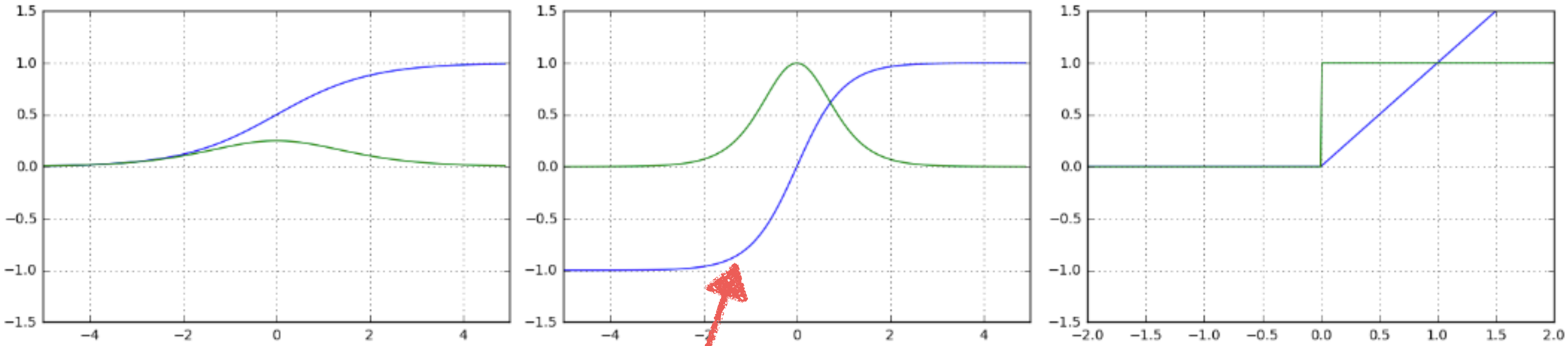
LET’S FIRST EXAMINE IN MORE DETAIL HOW SIMPLE “SHALLOW” NETWORKS WORK

ACTIVATION FUNCTIONS?



ADD NON LINEARITIES TO THE PROCESS

ACTIVATION FUNCTIONS



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

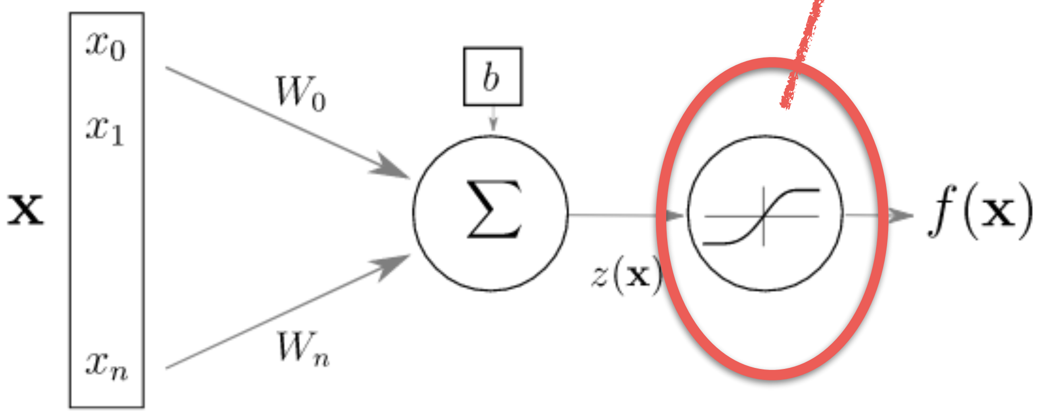
$$\mathrm{sigm}'(x) = \mathrm{sigm}(x)(1 - \mathrm{sigm}(x))$$

$$\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$$

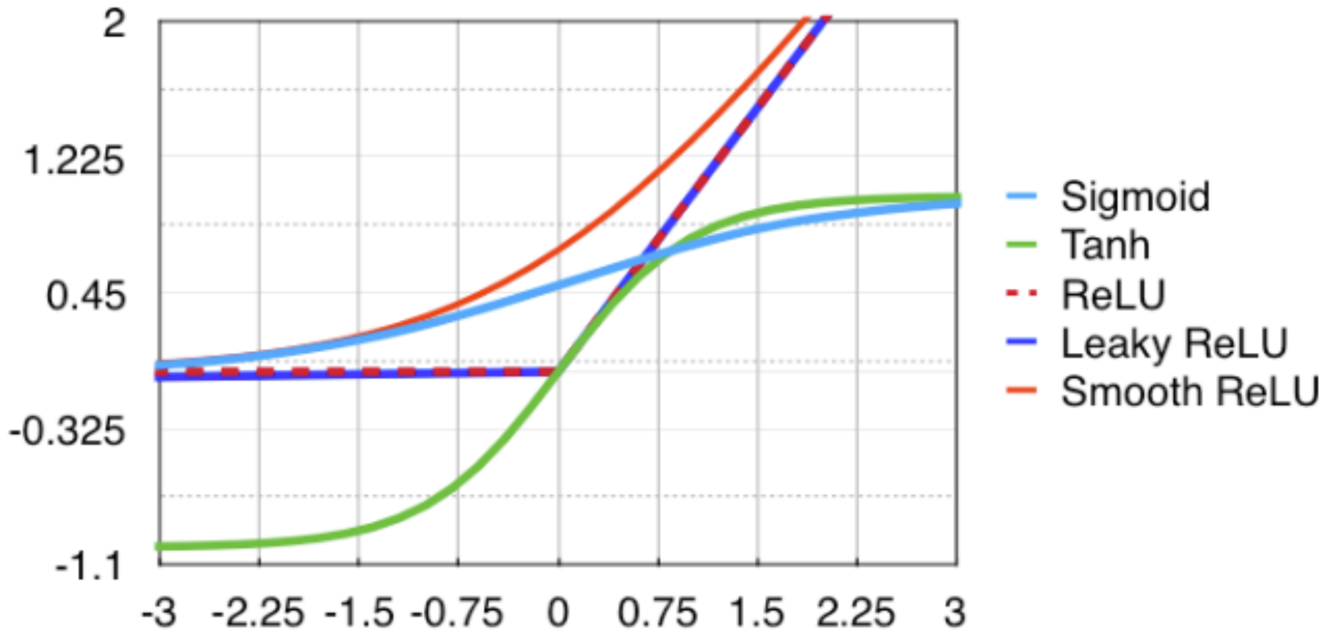
$$\tanh'(x) = 1 - \tanh(x)^2$$

$$\mathrm{relu}(x) = \max(0, x)$$

$$\mathrm{relu}'(x) = 1_{x>0}$$



ACTIVATION FUNCTIONS



Sigmoid: $f(x) = \dfrac{1}{1+e^{-x}}$

Tanh: $f(x) = tanh(x)$

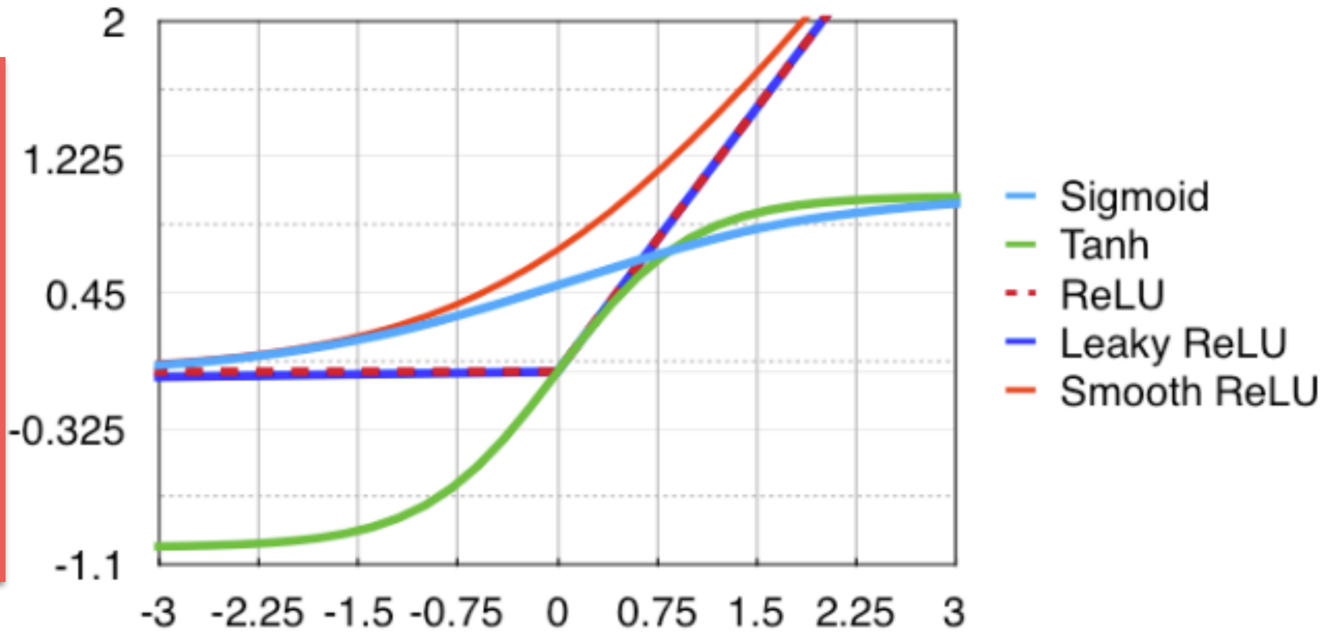
ReLu: $f(x) = max(0, x)$

Soft ReLu: $f(x) = log(1 + e^x)$

Leaky ReLu: $f(x) = \epsilon x + (1 - \epsilon) max(0, x)$

ACTIVATION FUNCTIONS

+ MANY OTHERS!



Sigmoid: $f(x) = \dfrac{1}{1+e^{-x}}$

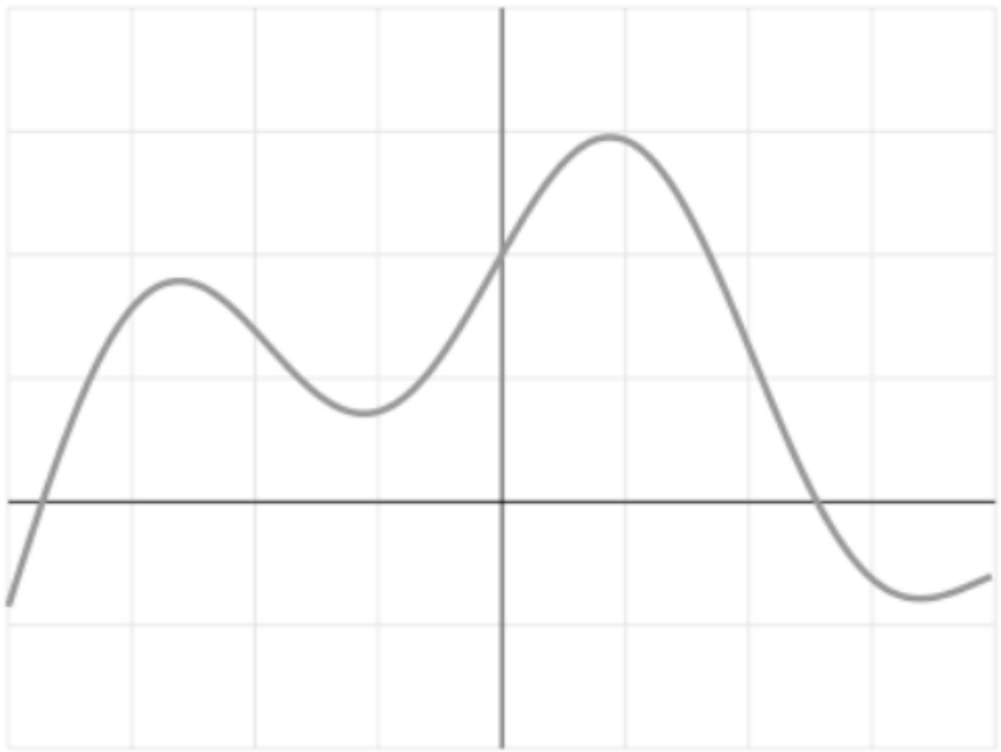
Tanh: $f(x) = tanh(x)$

ReLu: $f(x) = max(0, x)$

Soft ReLu: $f(x) = log(1 + e^{x})$

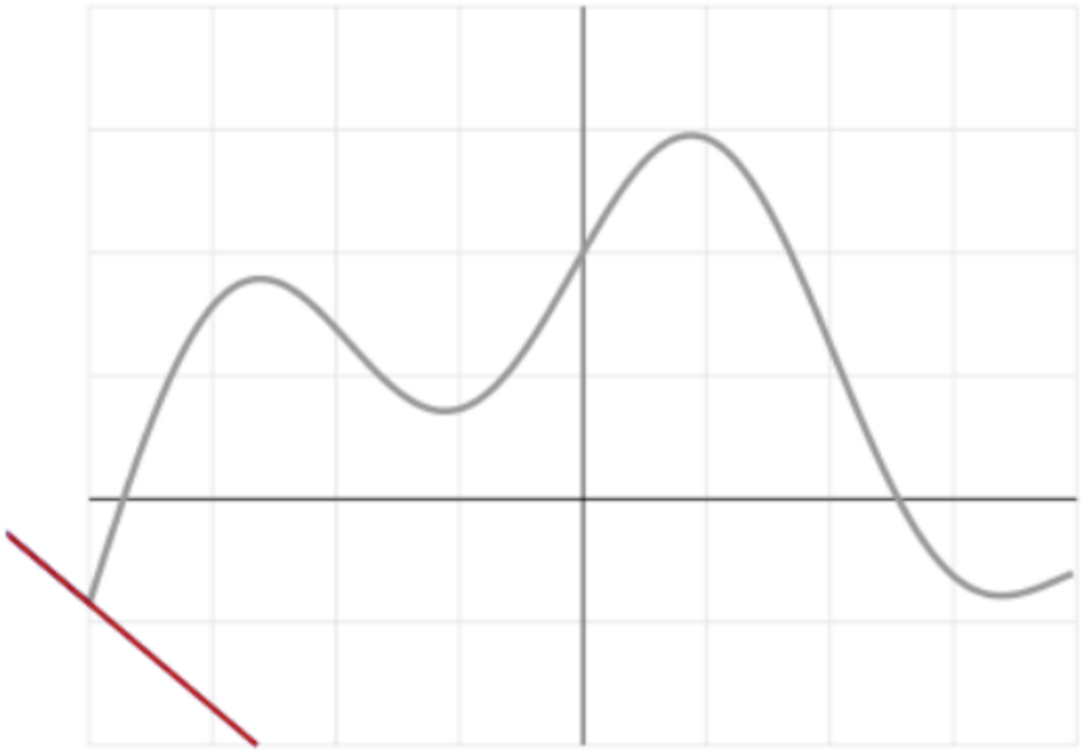
Leaky ReLu: $f(x) = \epsilon x + (1 - \epsilon) max(0, x)$

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



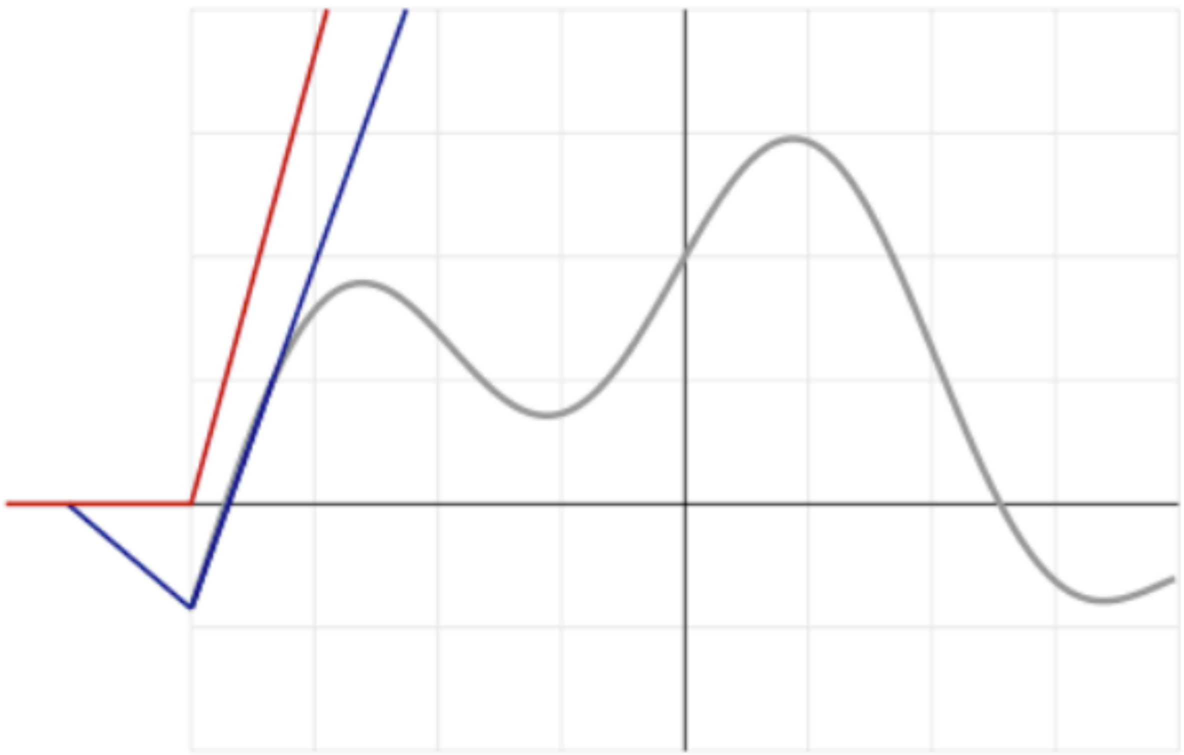
Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



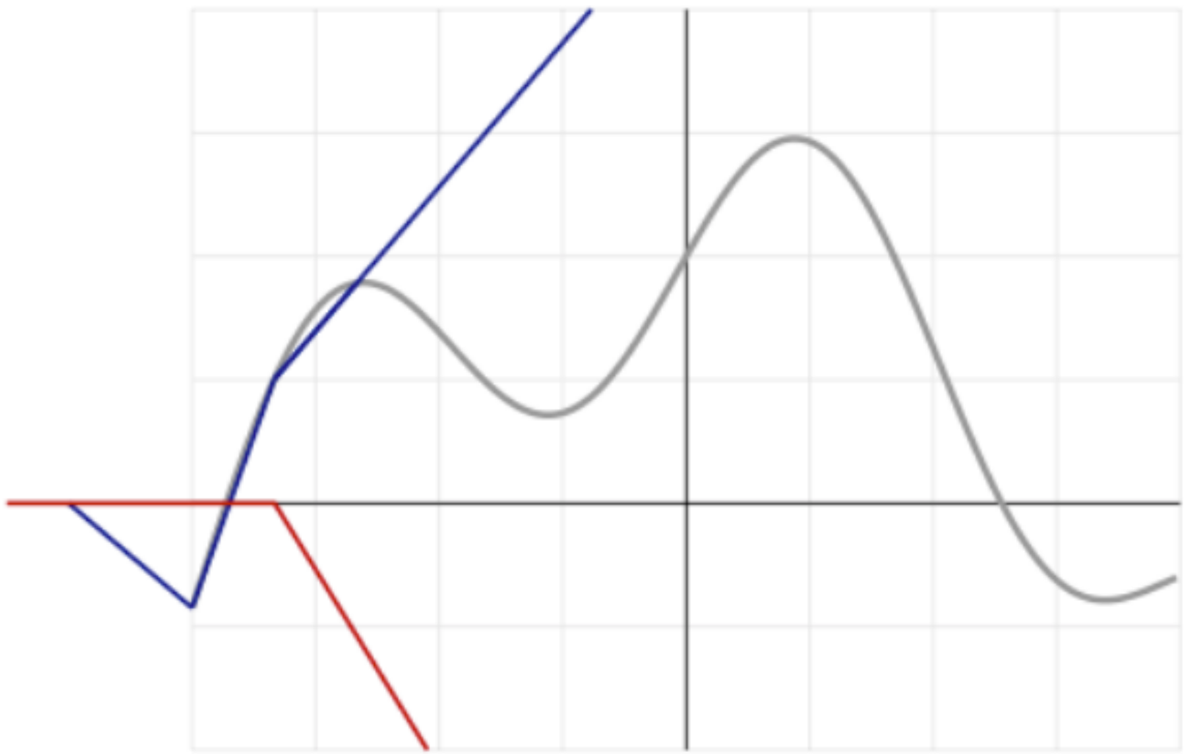
Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



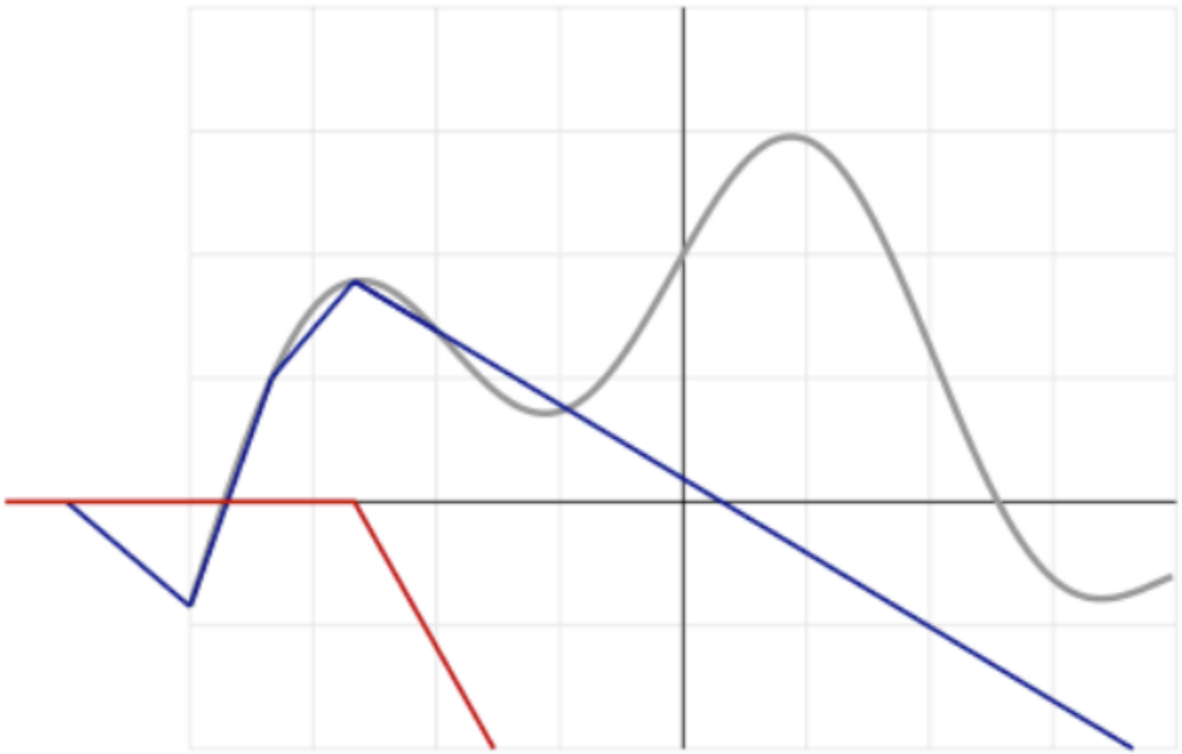
Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



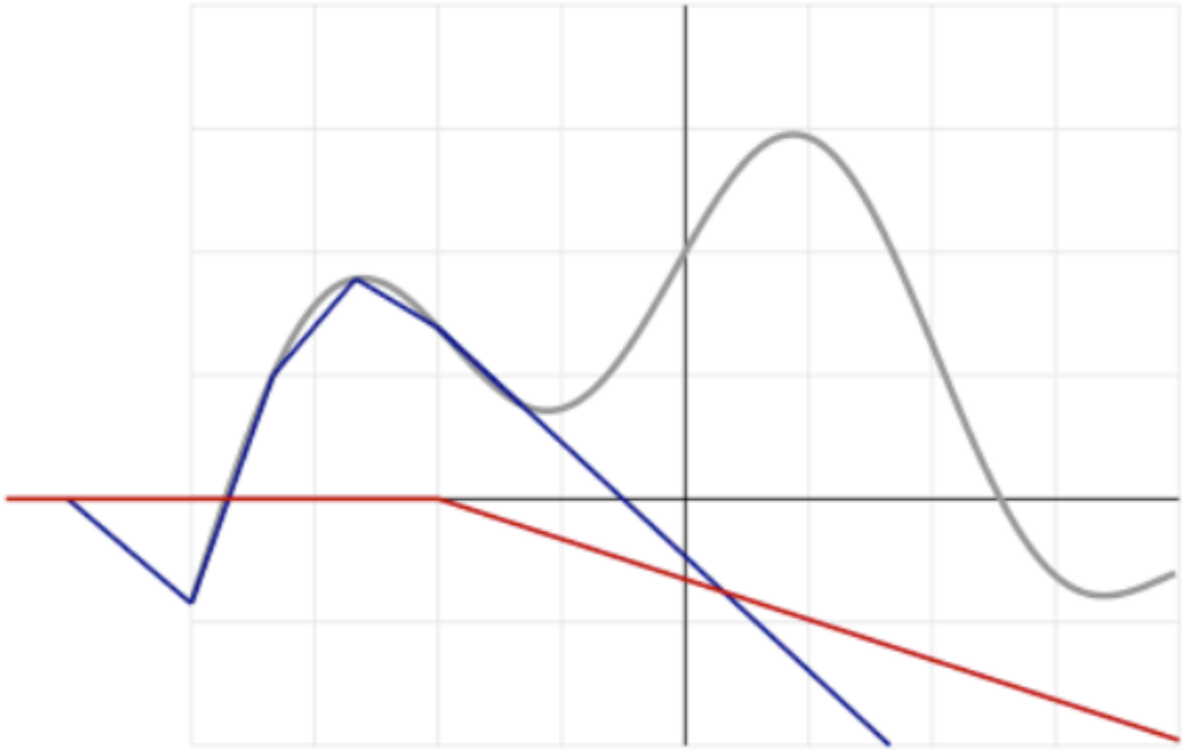
Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



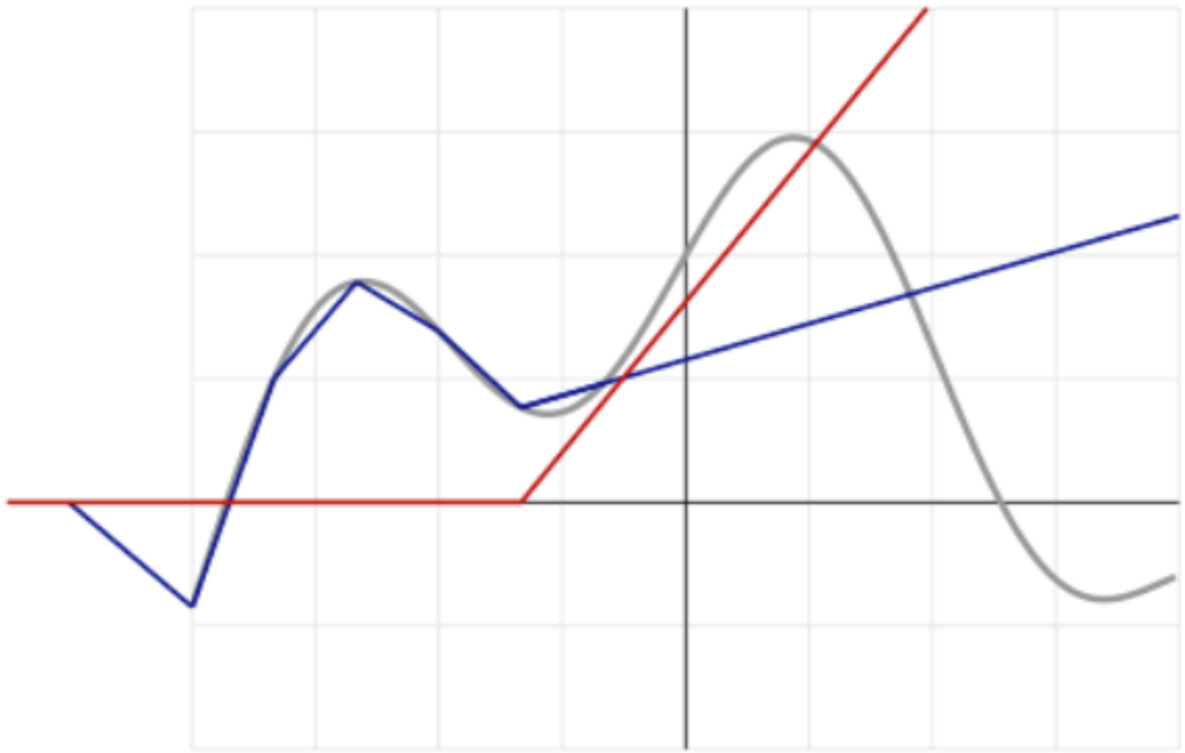
Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



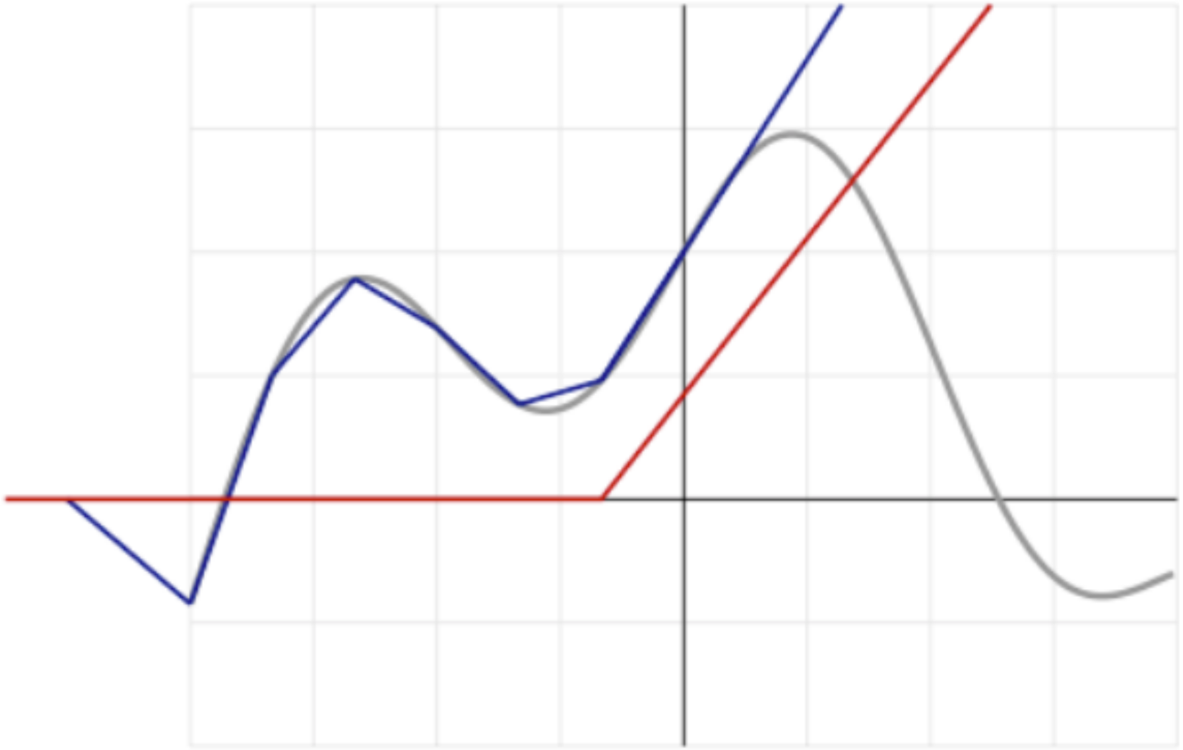
Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



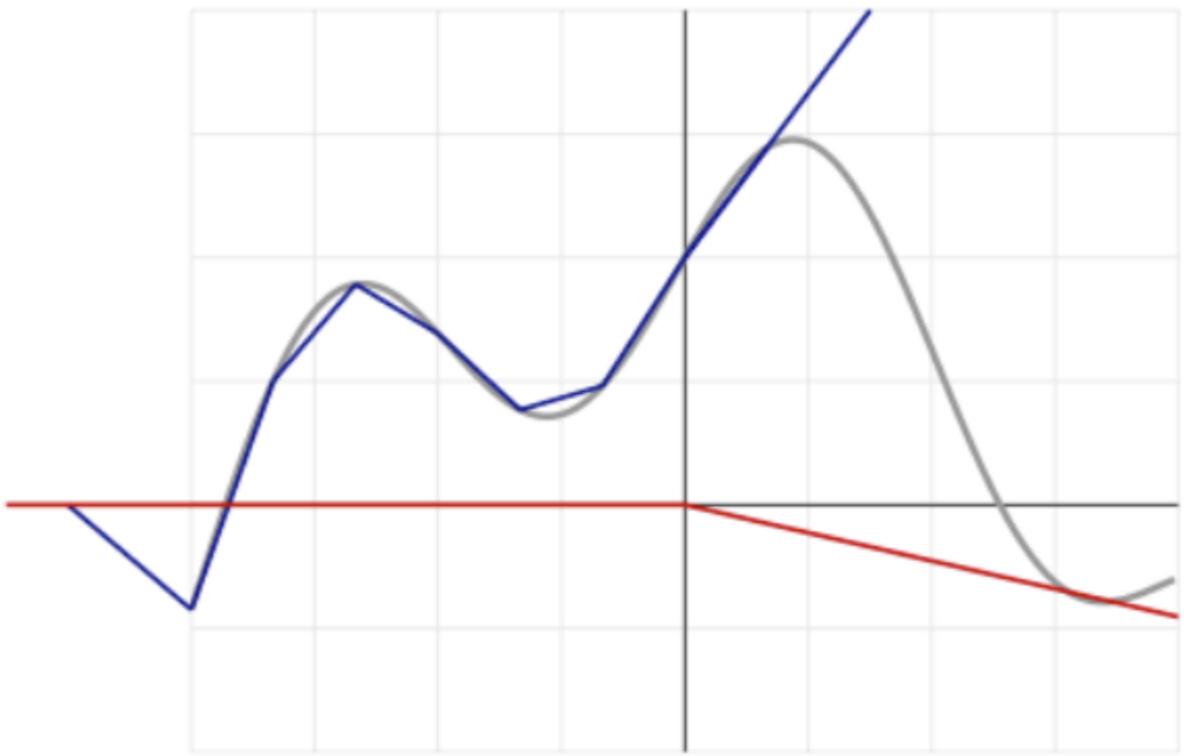
Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



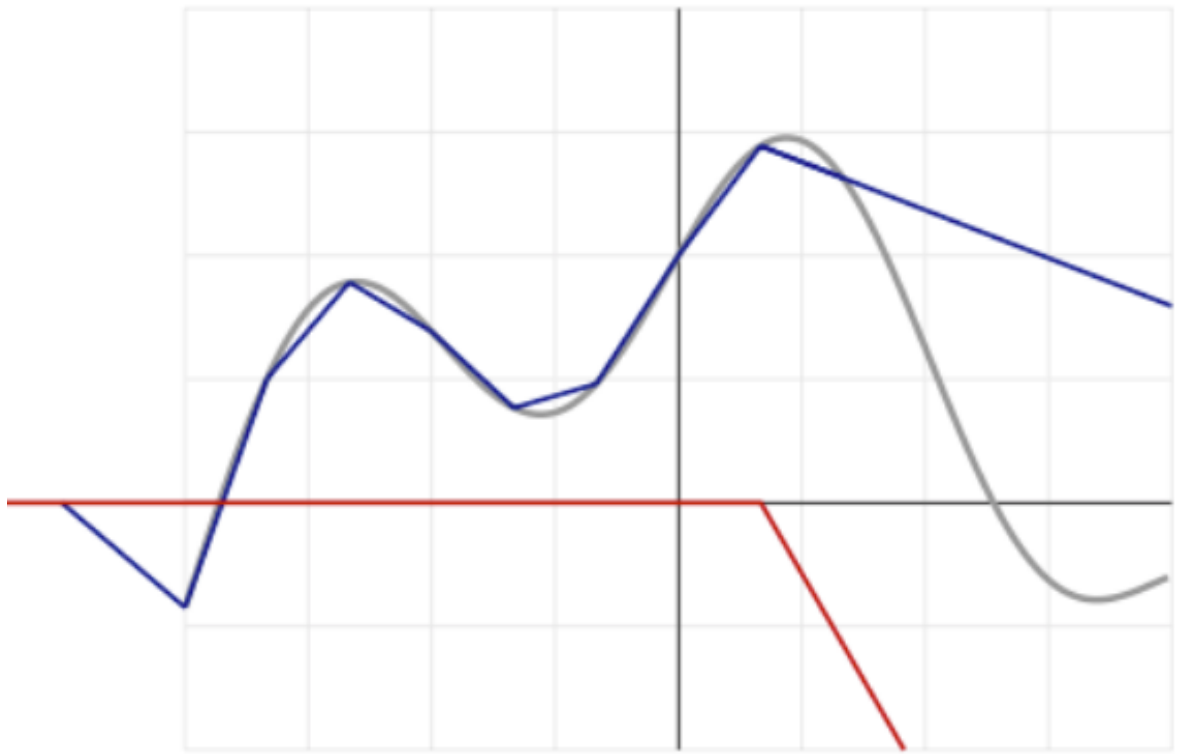
Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



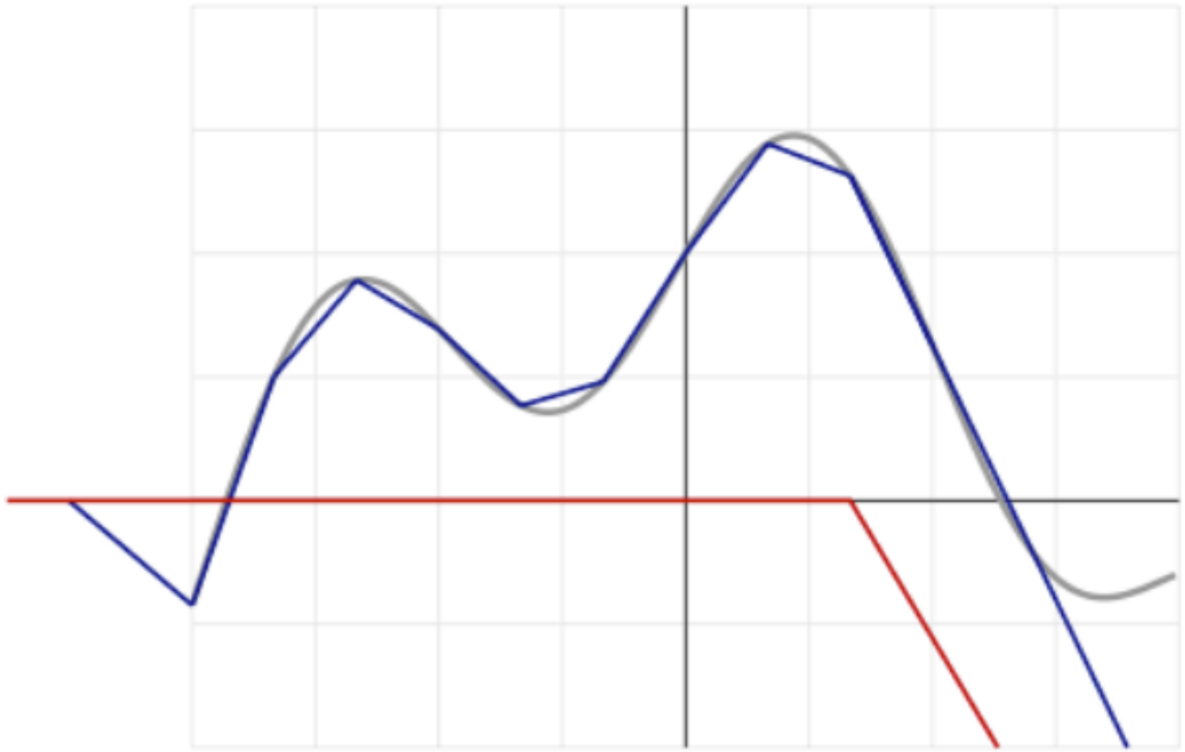
Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



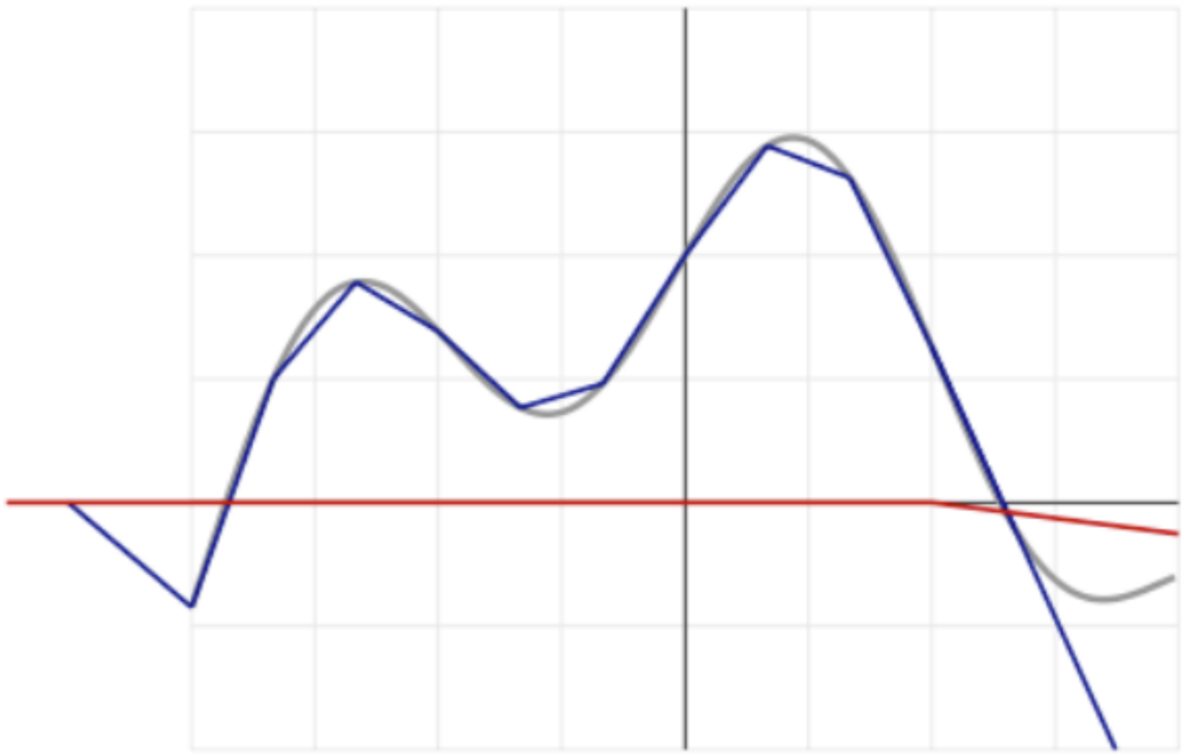
Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



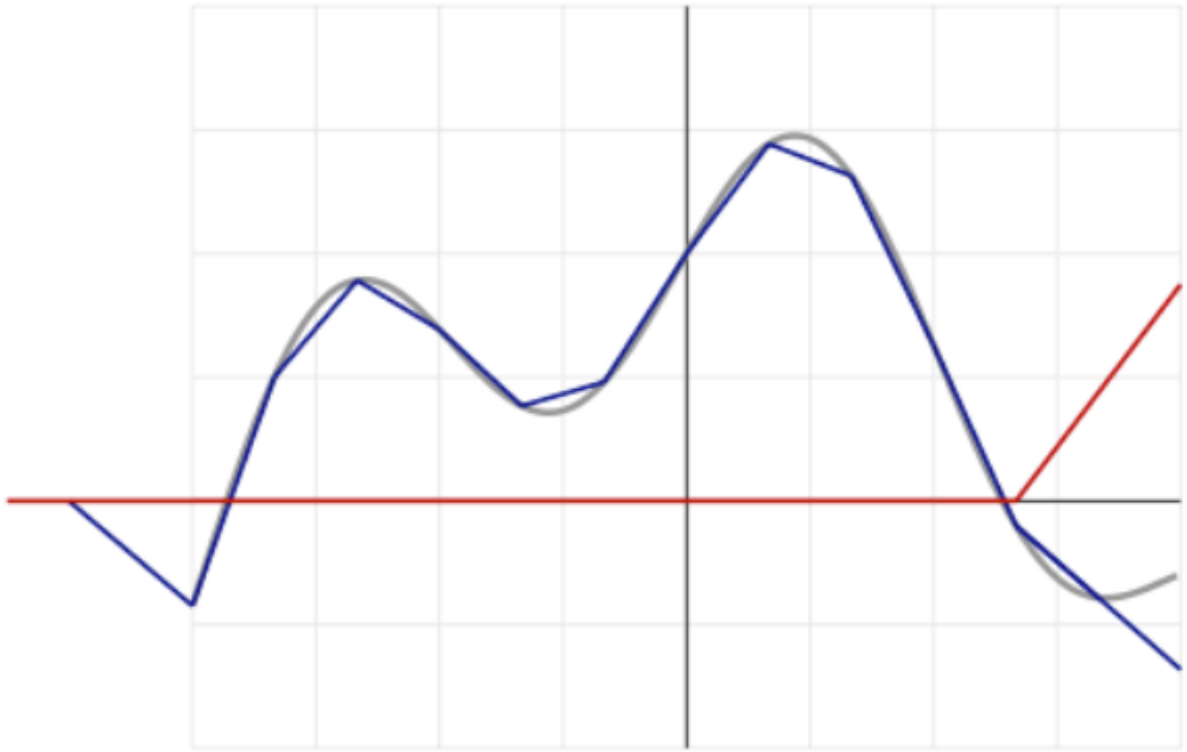
Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



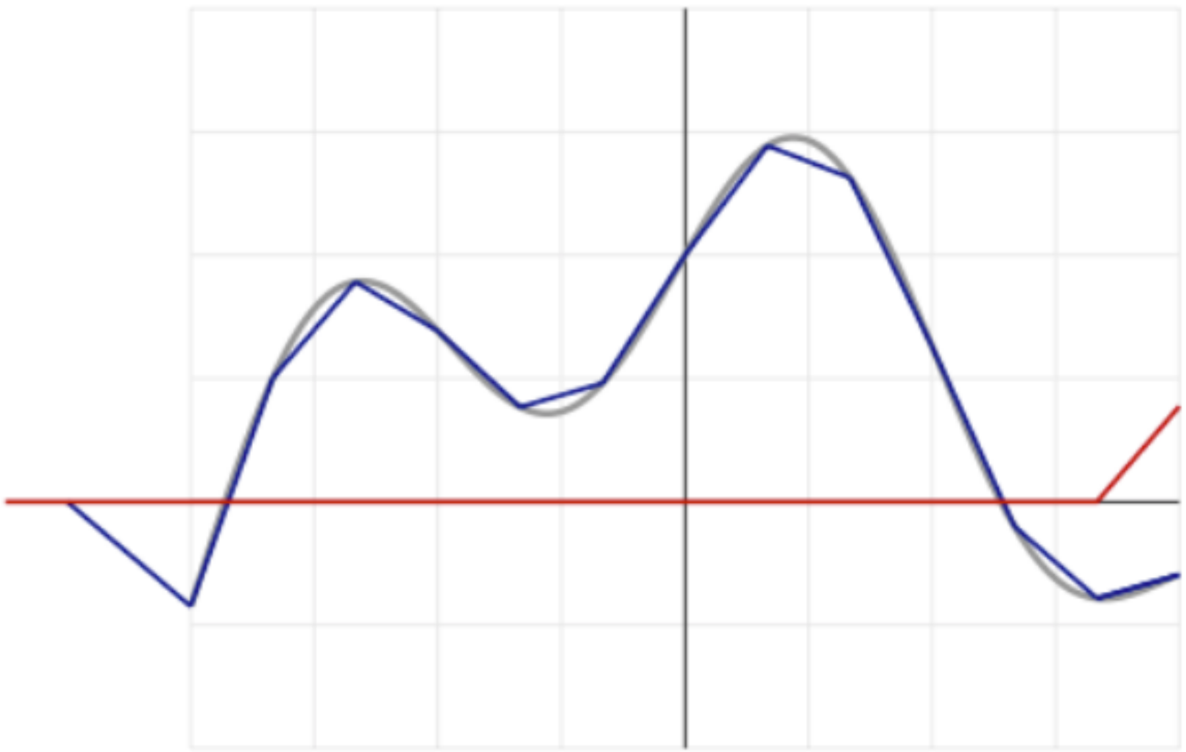
Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



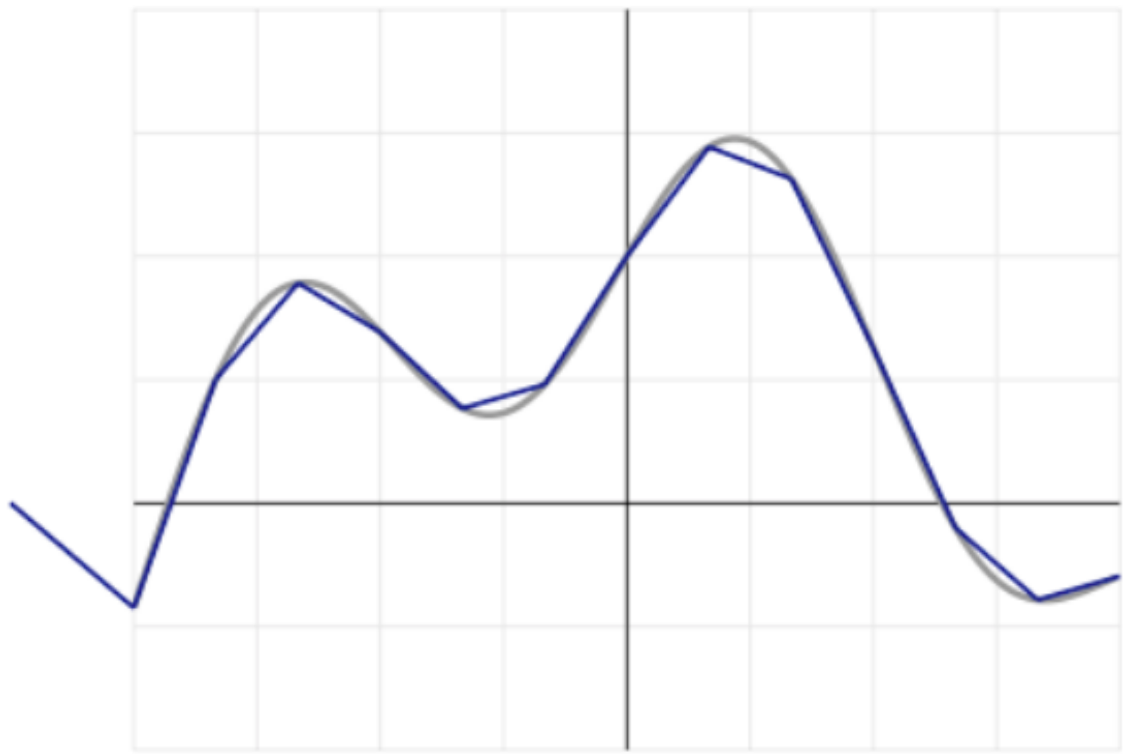
Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

WHAT IS THE MEANING OF THE ACTIVATION FUNCTION?



Any real function in a interval (a,b) can be approximated with a linear combination of translated and scaled ReLu functions

SOFTMAX

A generalization of the SIGMOID ACTIVATION

$$softmax(\mathbf{x}) = \frac{e^{\mathbf{x}}}{\sum_{i=1}^{n} e^{x_i}}$$

THE OUTPUT IS NORMALIZED BETWEEN 0 AND 1

THE <u>COMPONENTS ADD TO 1</u>

CAN BE INTERPRETED AS A PROBABILITY

$$p(Y = c | X = \mathbf{x}) = softmax(z(\mathbf{x}))_c$$

SOFTMAX

A generalization of the SIGMOID ACTIVATION

GENERALLY
USED AS ACTIVATION
OF LAST LAYER

(will come back later)

THE OUT[illegible] AND 1

CAN BE INTERPRETED AS A PROBABILITY

$$p(Y = c|X = \mathbf{x}) = softmax(z(\mathbf{x}))_c$$