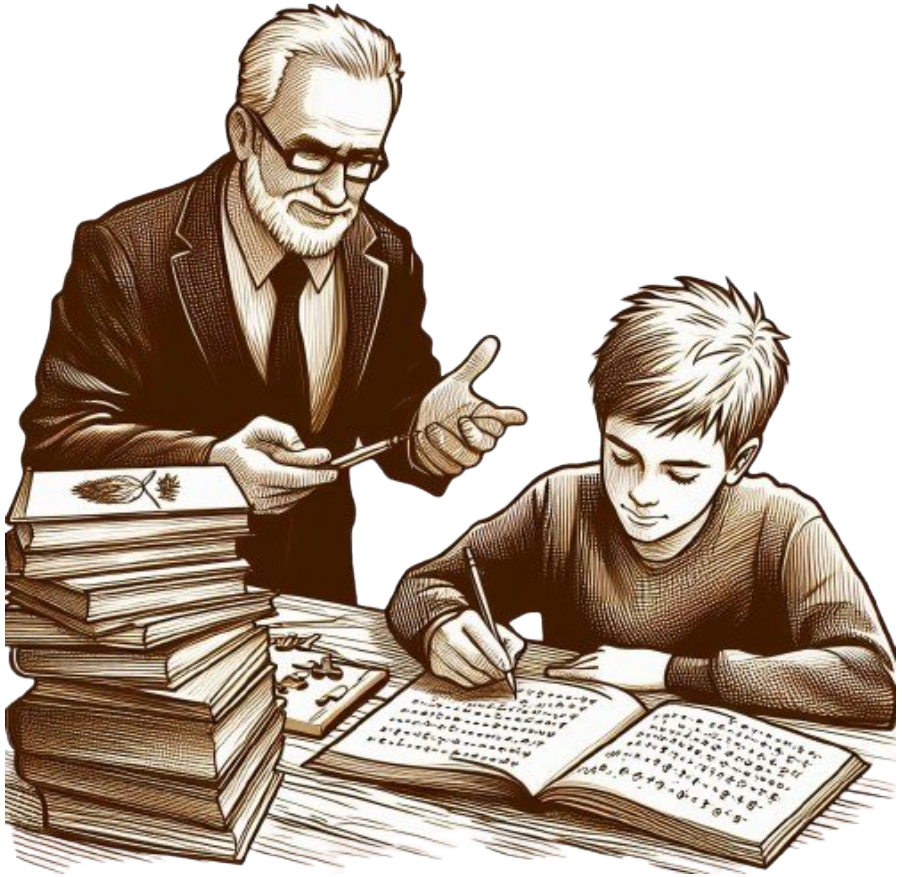


Hands-on Introduction to Deep Learning

Methodology

1. What **work** can be done to **improve** the **data** used for training?
2. How can a model be **evaluated**?
3. Is it possible to make the training more **robust**?
4. Can we **benefit** from an **already** trained model?
5. Bonus: Good **practices**



Learning from exercise with a teacher to guide us



Applying what we learn to the real world

Training & Generalization

https://arxiv.org/pdf/1710.08864.pdf

CIFAR-10

AllConv



SHIP
CAR (99.7%)

NiN



HORSE
FROG (99.9%)

VGG16



DEER
AIRPLANE (85.3%)

ImageNet

BVLC AlexNet



Cup (16.48%)
Soup Bowl (16.74%)



Bassinet (16.59%)
Paper Towel (16.21%)

Generalization Lack Issue: One Pixel Attack

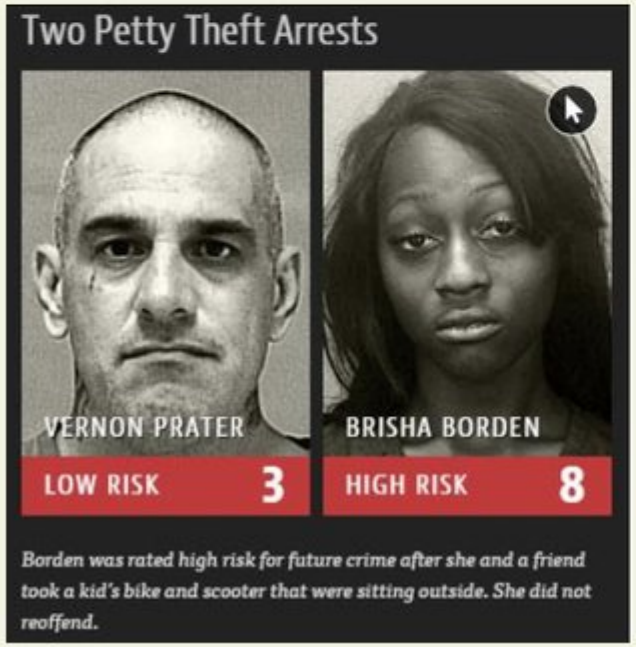
PROPUBLICA

May 25 | 2016

MACHINE BIAS

What Algorithmic Injustice Looks Like in Real Life

A computer program rated defendants' risk of committing a future crime. These are the results.



Le Monde

25 août | 2026

PIXELS

Biais racistes de Gemini : Google « travaille à des améliorations »

« Je suis seule avec un Algérien », « je suis seule avec un Italien » : deux questions simples, susceptibles de générer des réponses stéréotypées, voire racistes.

Par Nour Mohammedi

Les Echos

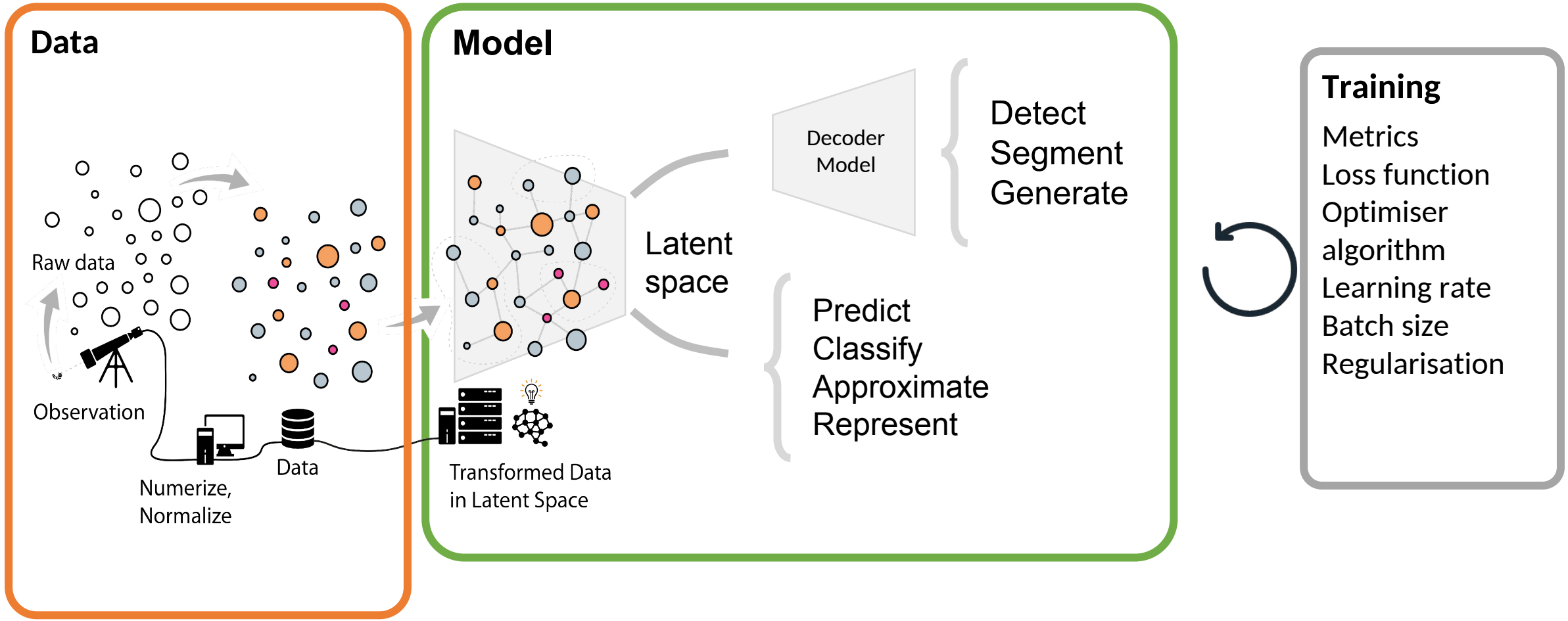
October 13 | 2018

ENTREPRISES

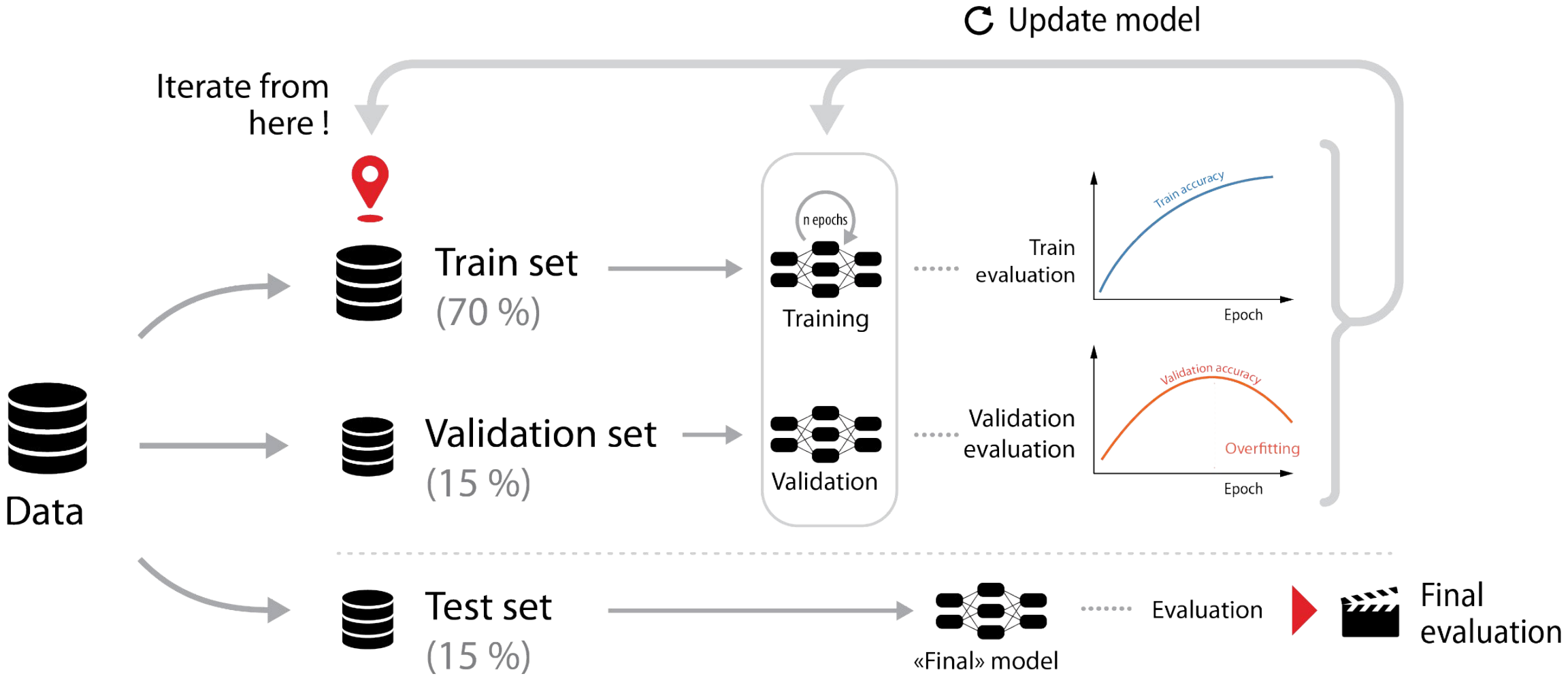
Quand le logiciel de recrutement d'Amazon discrimine les femmes

En 2014, le géant du e-commerce a voulu confier ses candidatures à un algorithme, mais celui-ci a commencé à écarter les profils féminins.

Generalization Lack Issue : Discrimination



Deep Learning Pipeline Principle



Data splitting

Dataset is the culprit ?!



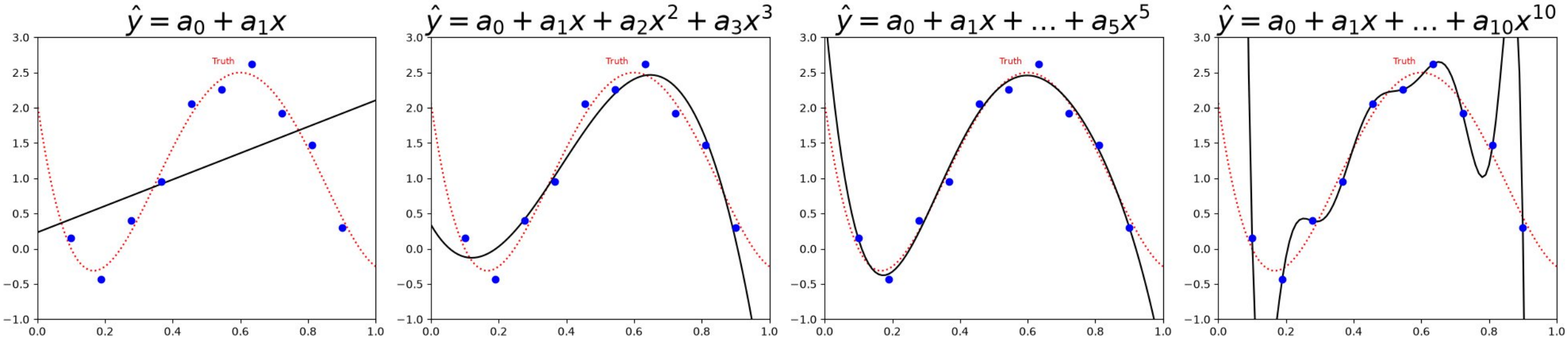
Augment It, Transform It !!!

Super Regularizator !!



Use his Super Powers !!

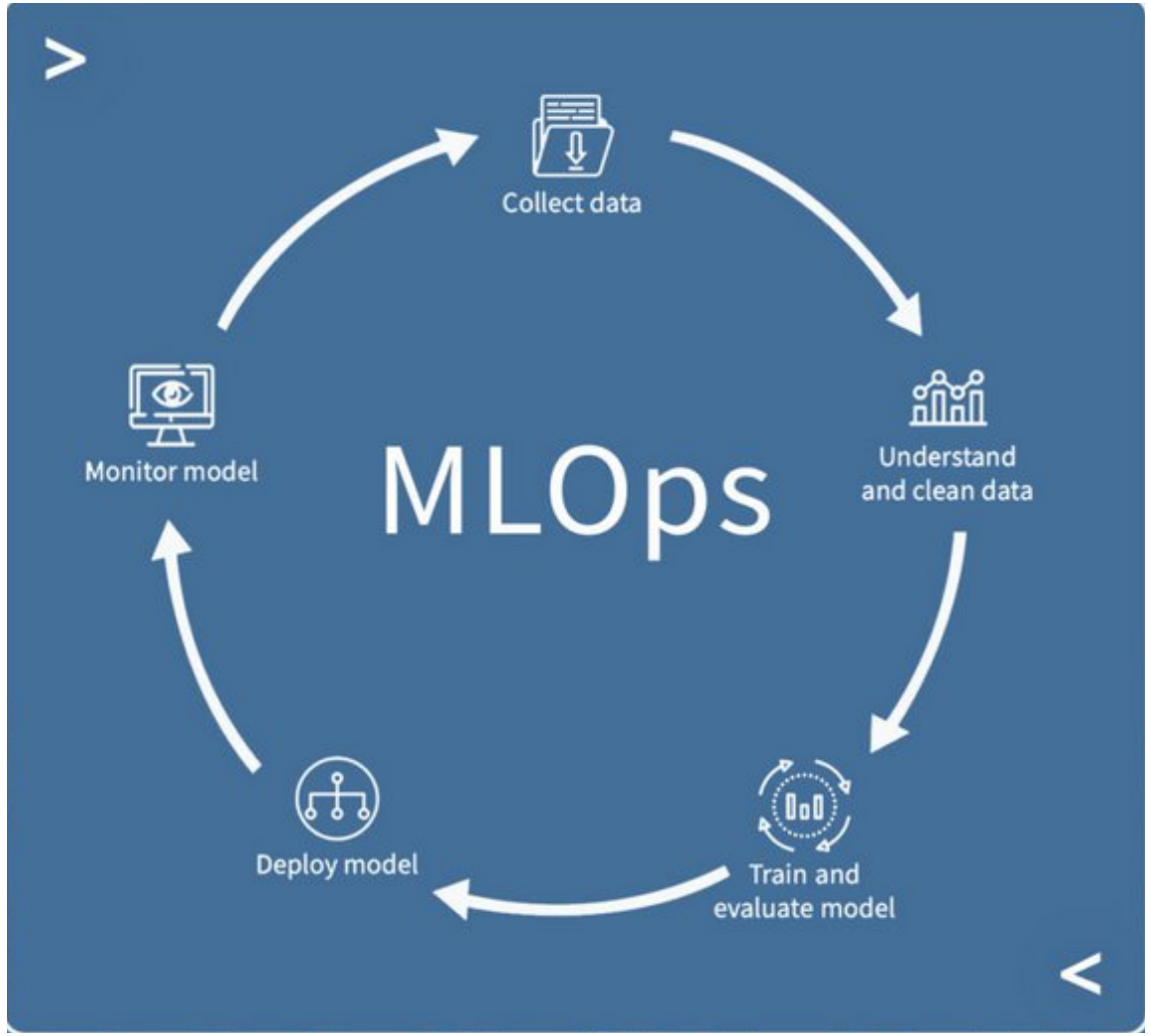
For a Better Generalization...



Underfitting and Overfitting

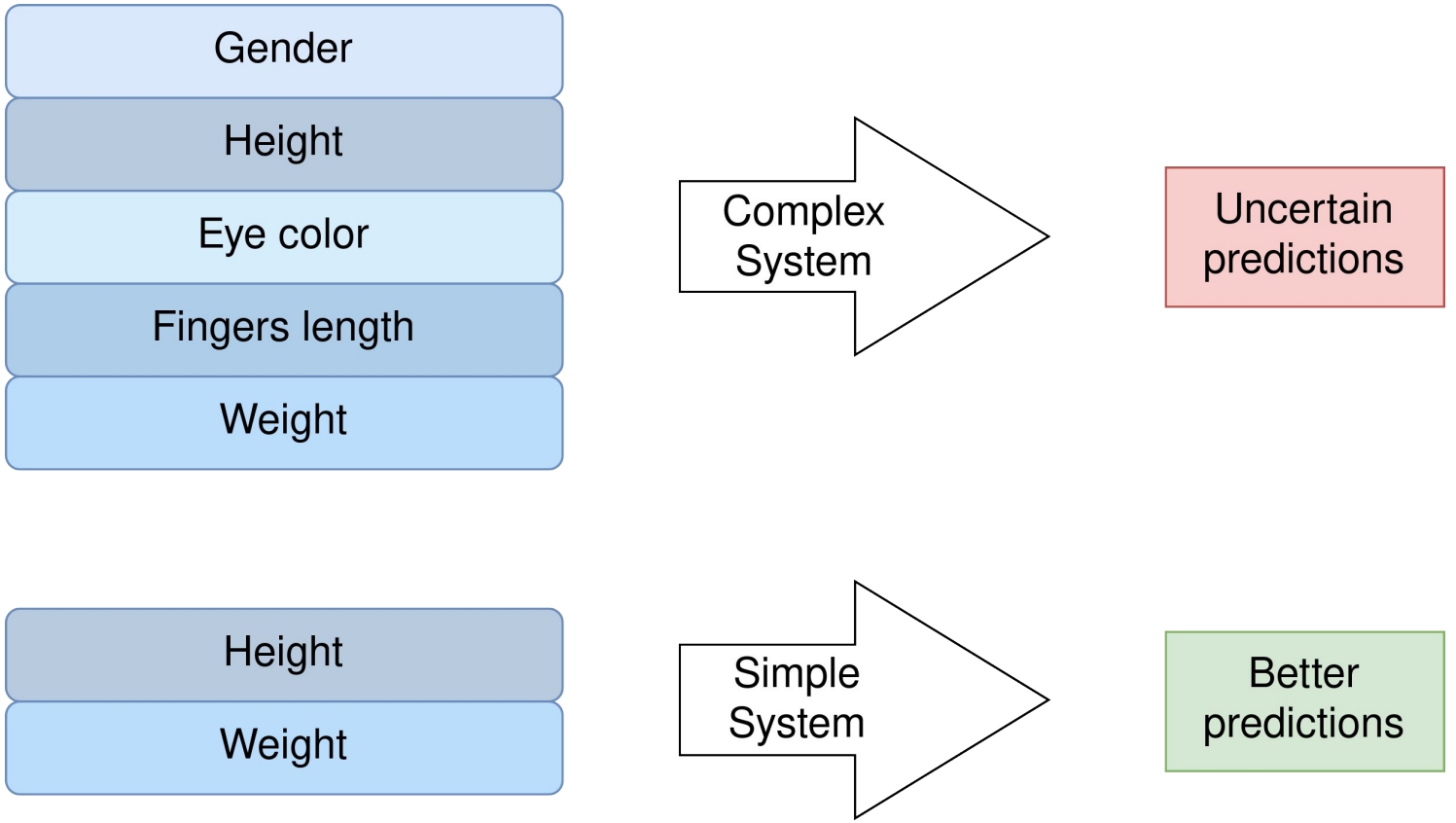
1. What **work** can be done to **improve** the **data** used for training?
2. How can a model be **evaluated**?
3. Is it possible to make the training more **robust**?
4. Can we **benefit** from an **already** trained model?
5. Bonus: Good **practices**

https://posit.co/blog/mlops-readiness/



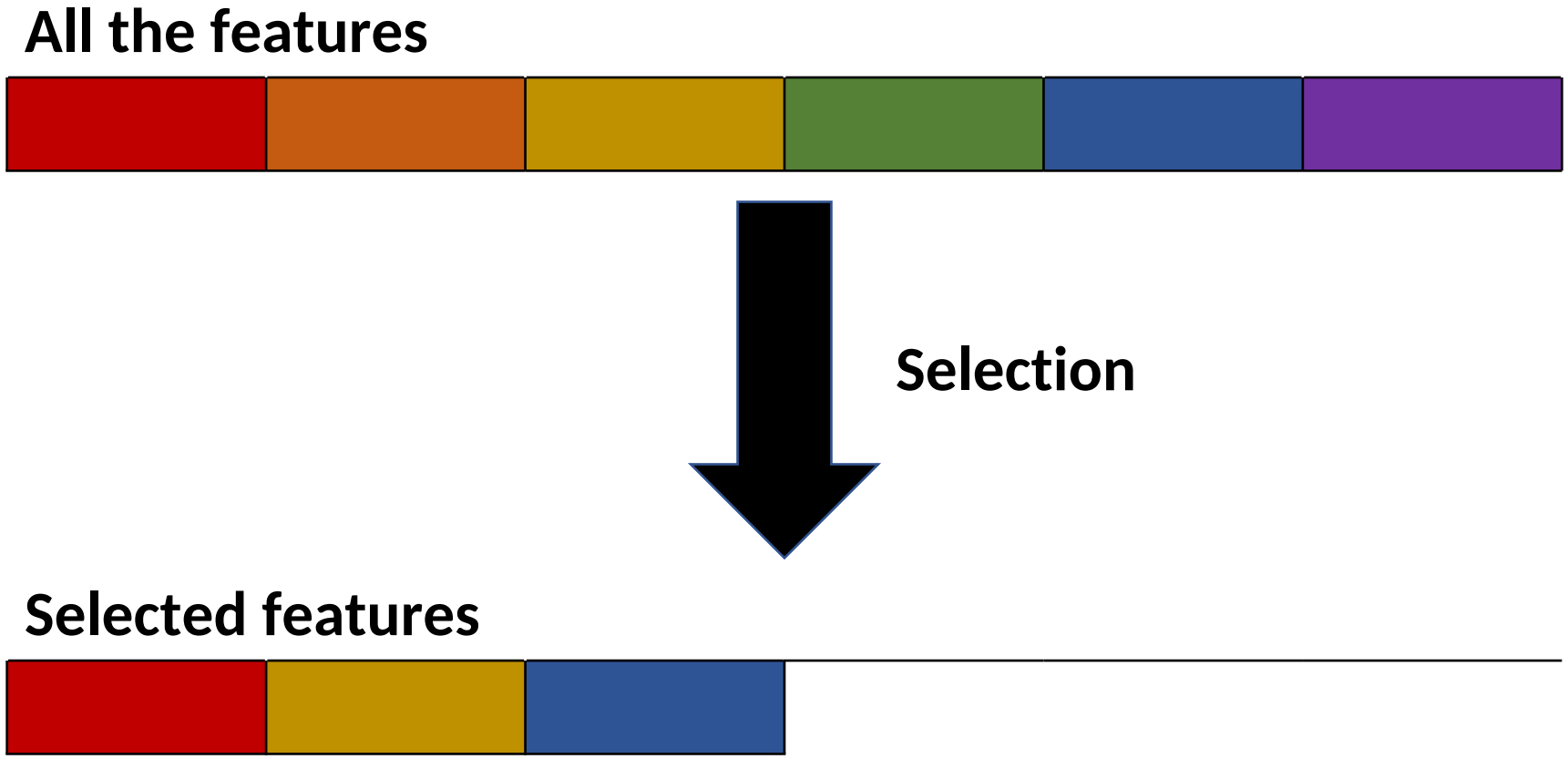
50 to 80% time spent on data

Model development



Disease risk prediction system

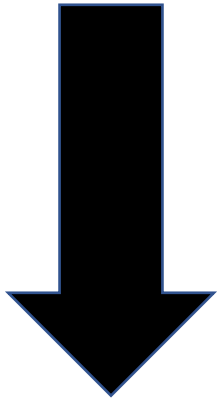
Features selection



Features selection

All the features



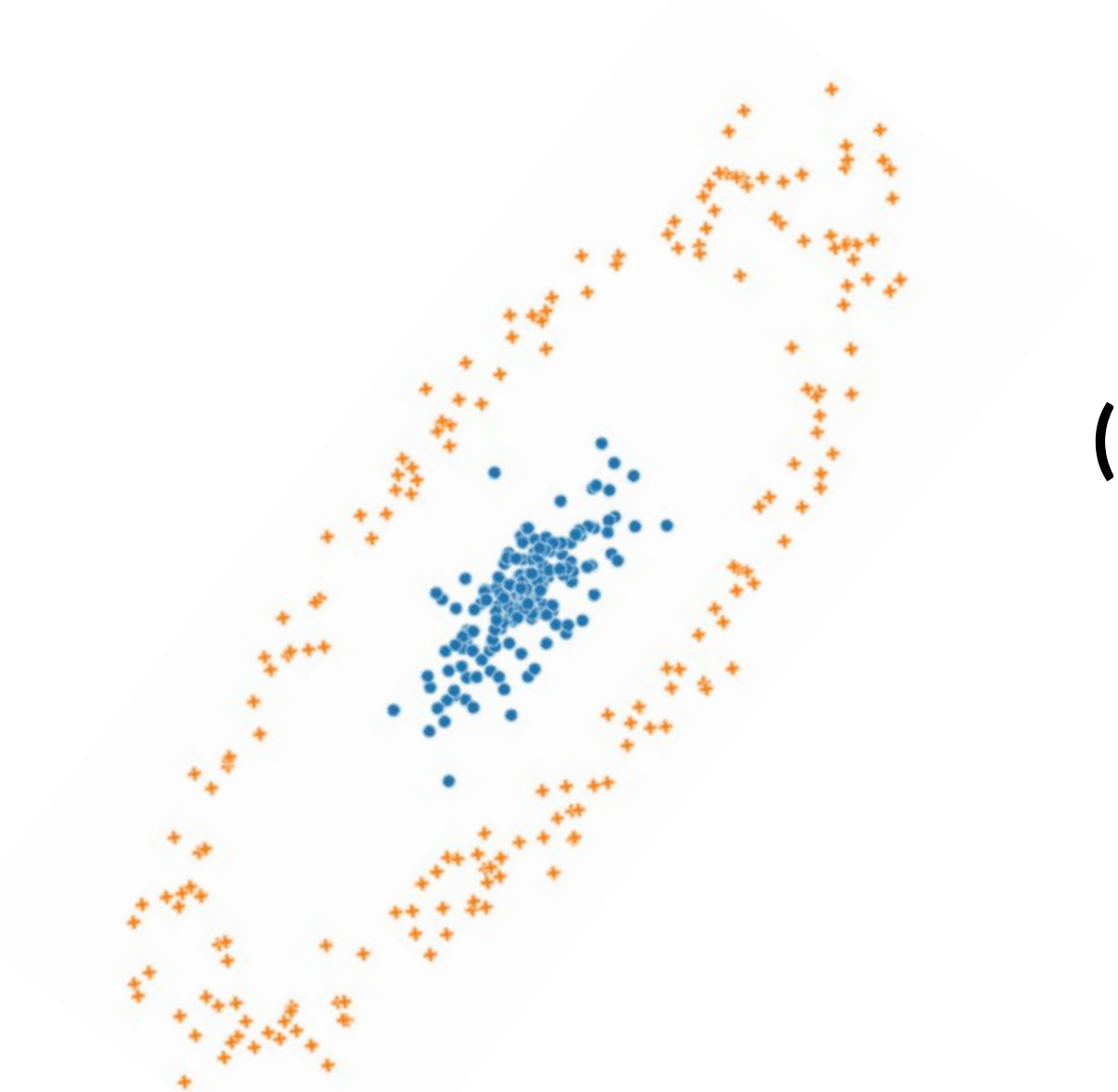


Transformation

Transformed features



Feature transformation

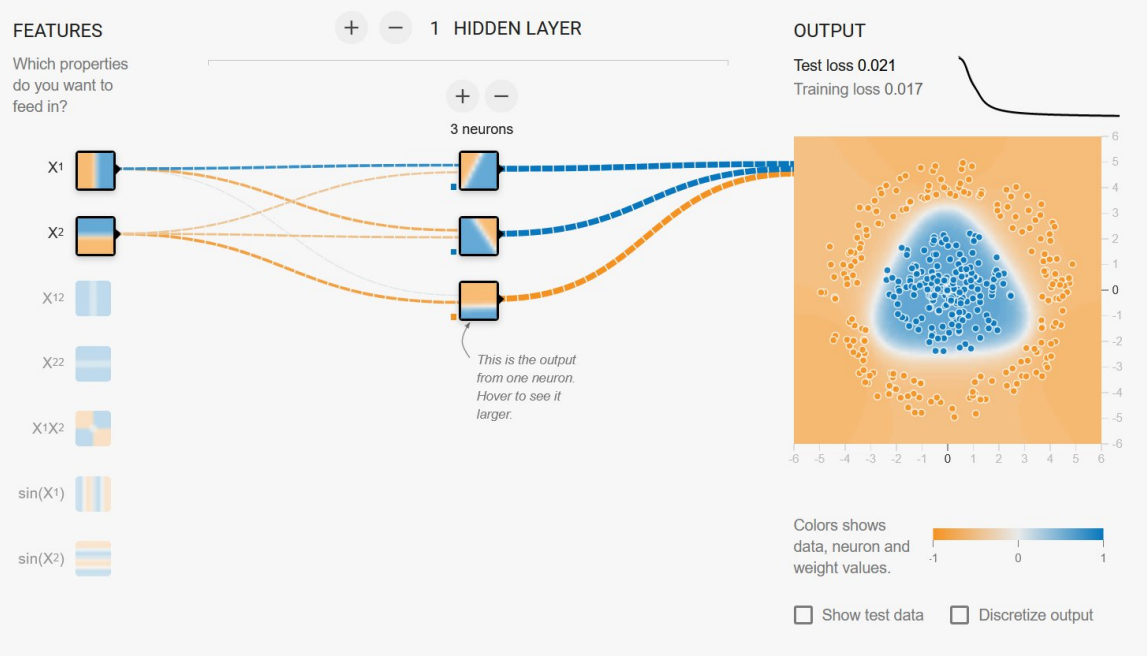


(x , y) → Cartesian coordinates: Complex relation between x and y

(r , θ) → Polar coordinates: Simple relation with r (radius) and θ (angle)

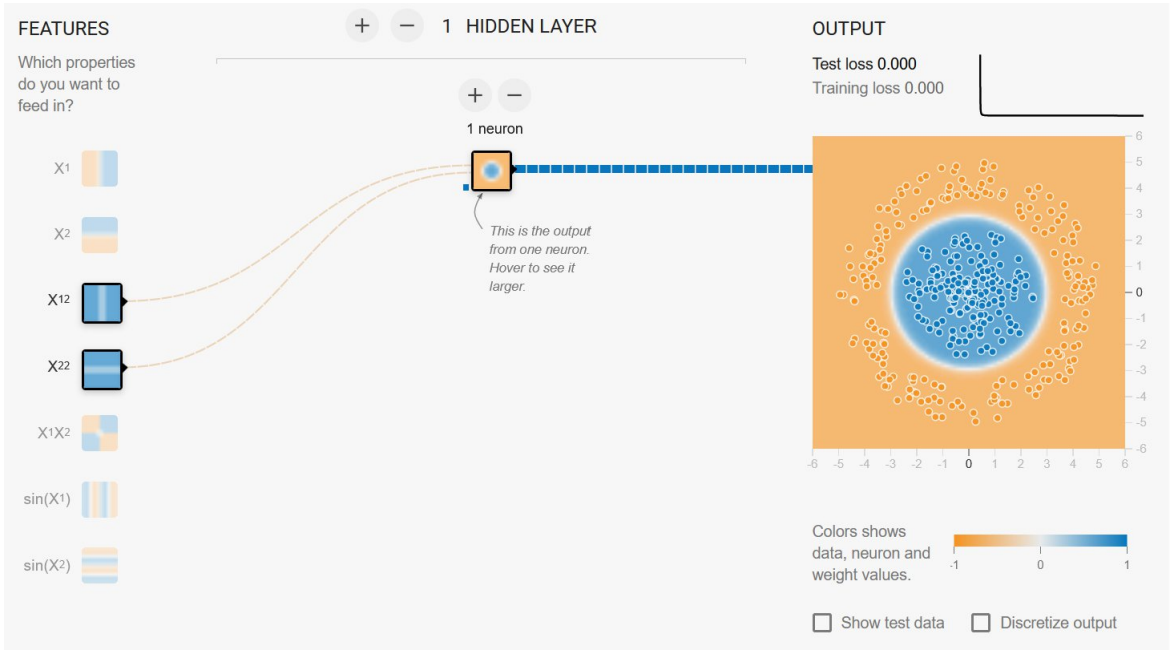
Feature transformation - Example

https://playground.tensorflow.org



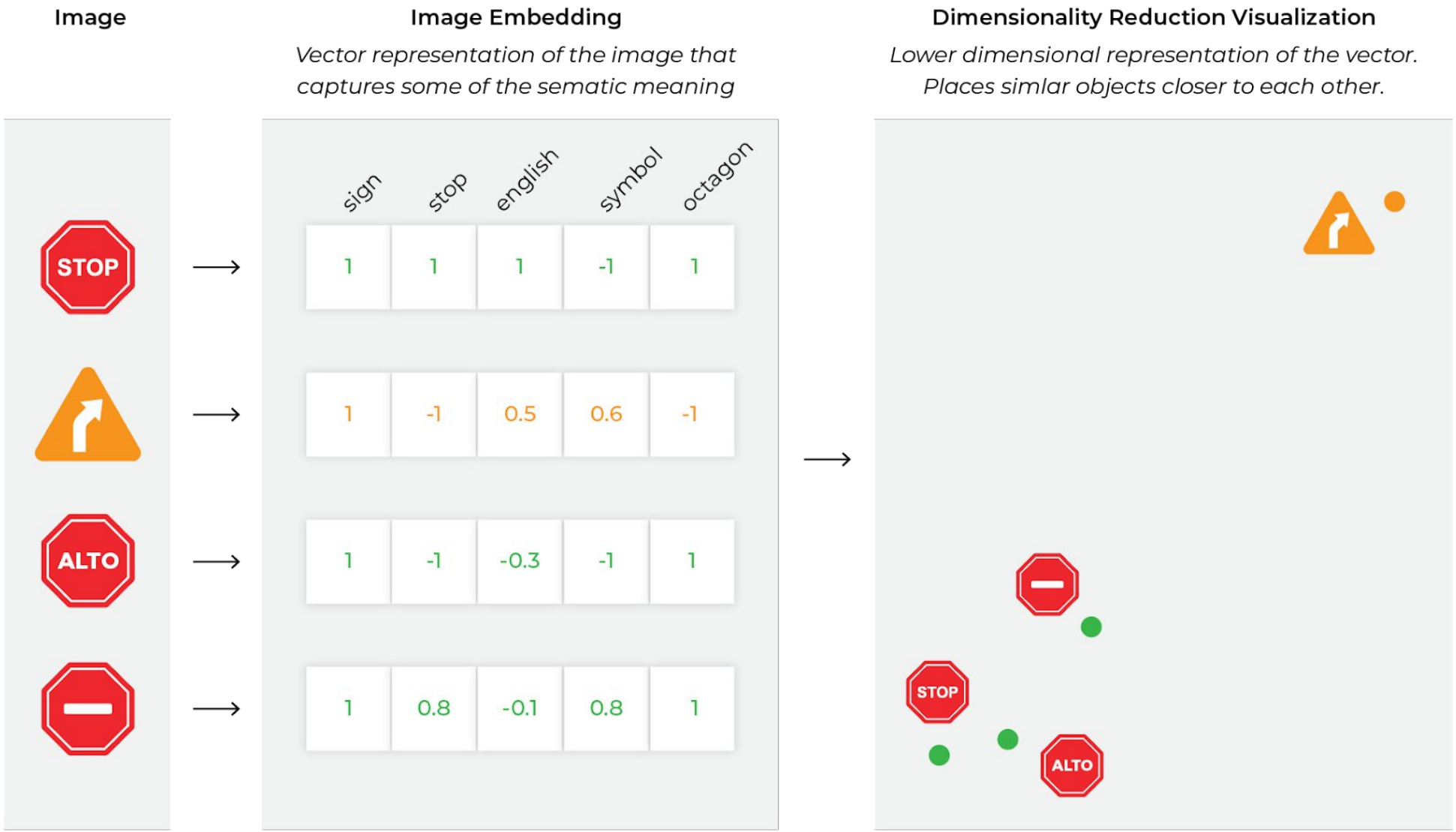
Données de position (x, y)
Réseau à 3 neurones

Données transformées
Réseau à 1 neurone

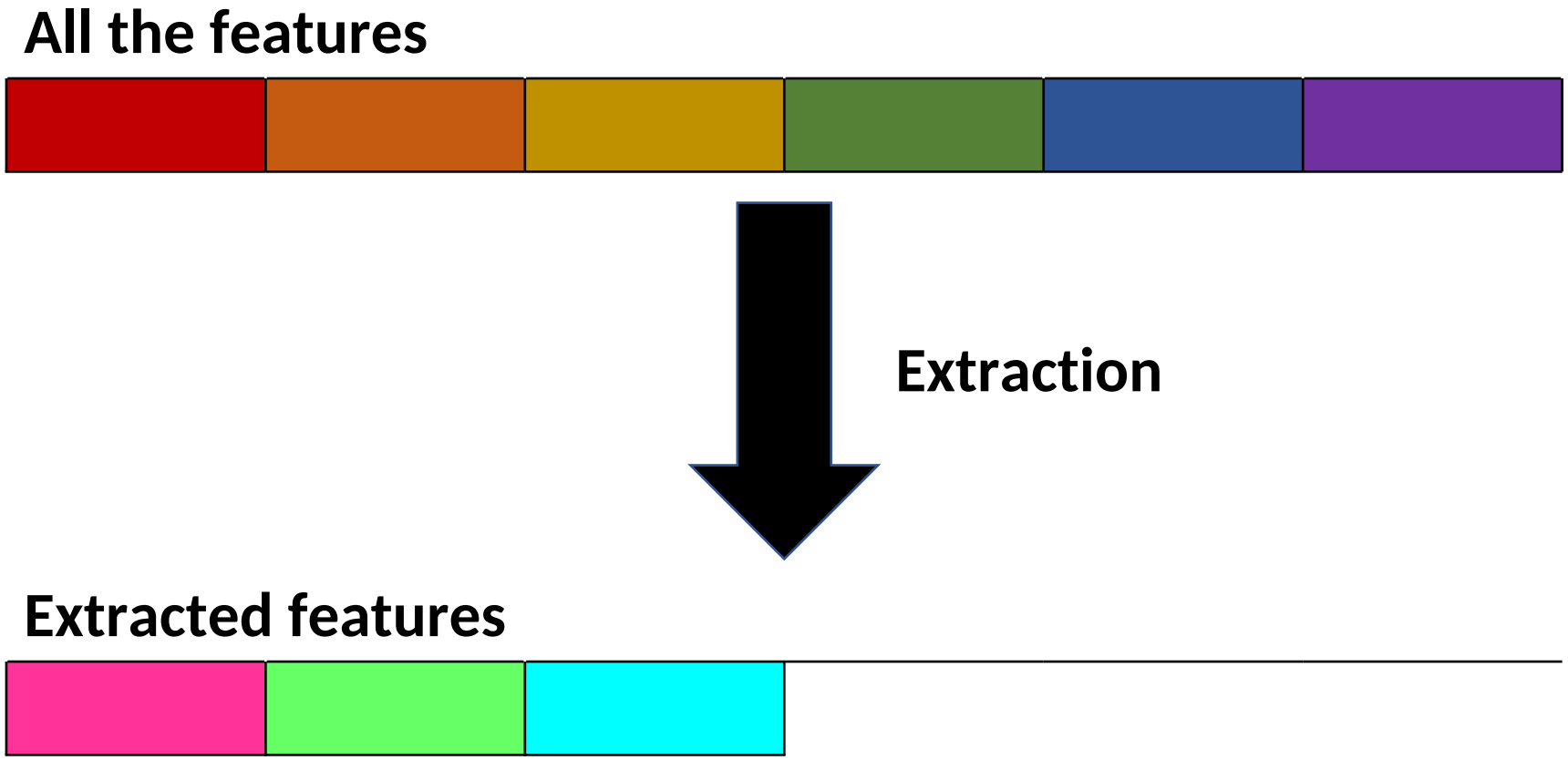


Feature transformation - Example

https://arize.com/blog-course/embeddings-meaning-examples-and-how-to-compute/



Features extraction



Features extraction



All the features

Selection

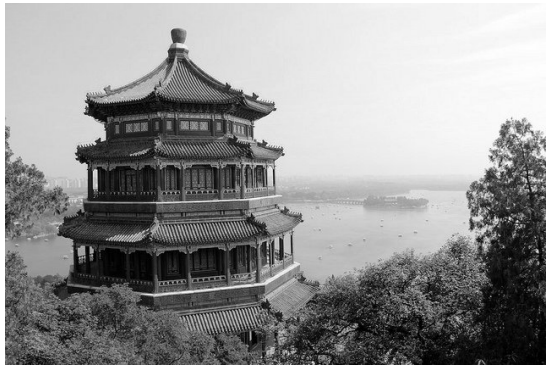
Selected features



All the features

Transformation

Transformed features



All the features

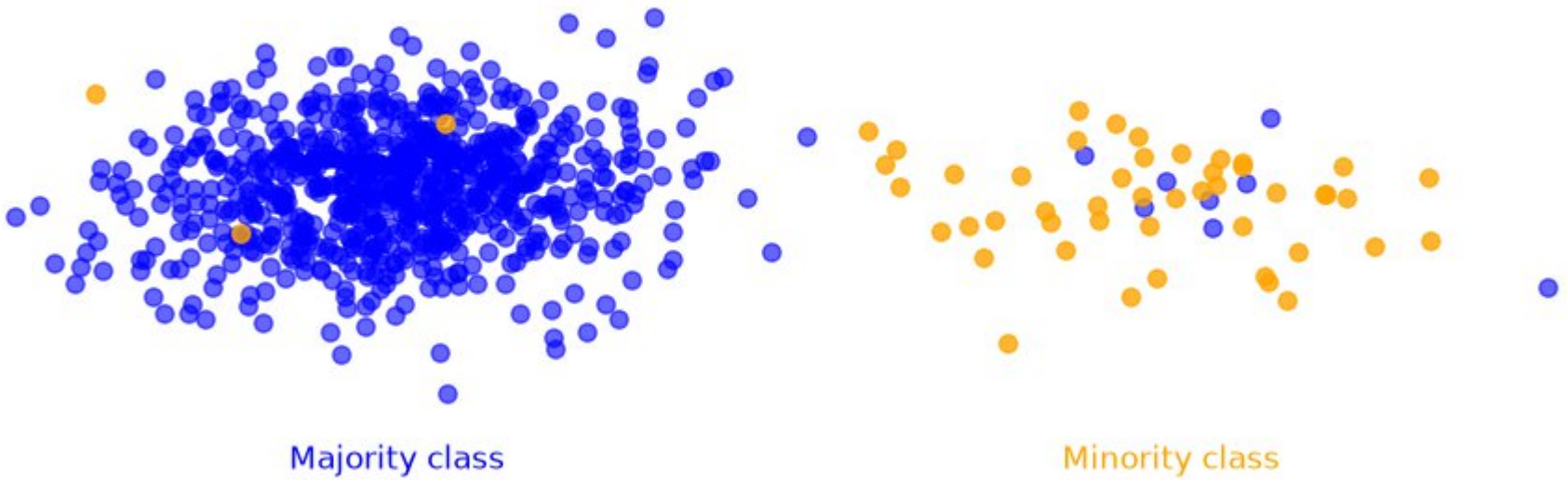
Extraction

Extracted features

```
[  6208.56   -160.27    304.41    661.01     52.87
    56.86   -125.47     21.15     39.49     34.21
                ...
     5.89     19.53     -6.33     17.98     -0.35
    16.14      4.38      9.64      8.03     -7.29 ]
```

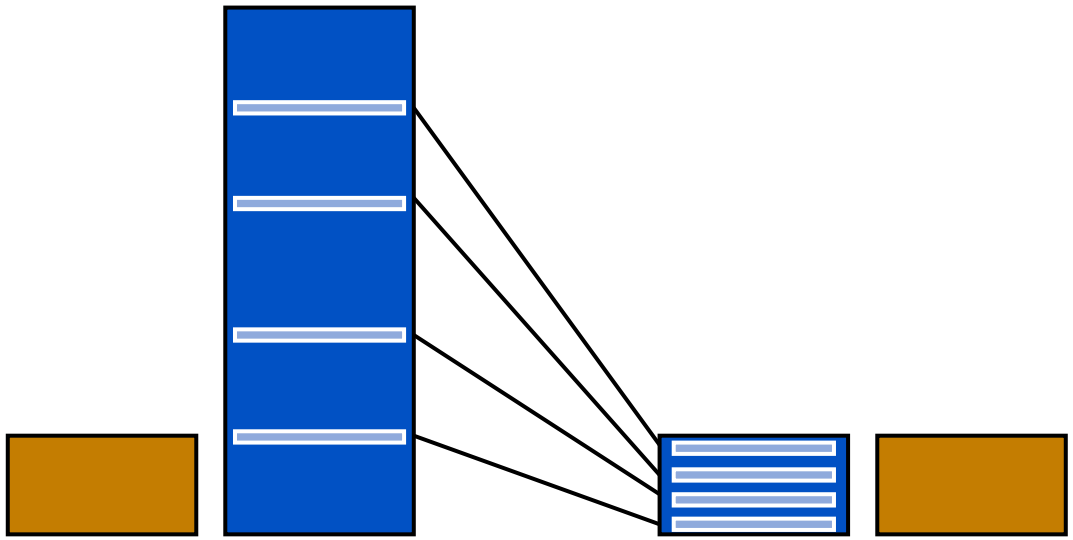
Features engineering - takeaway



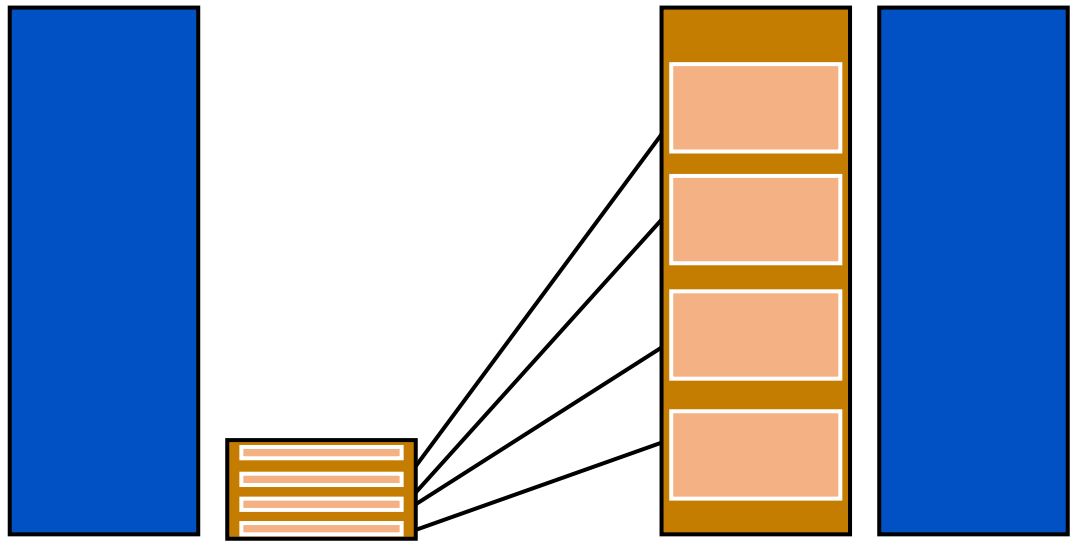


Balance the classes

Undersampling

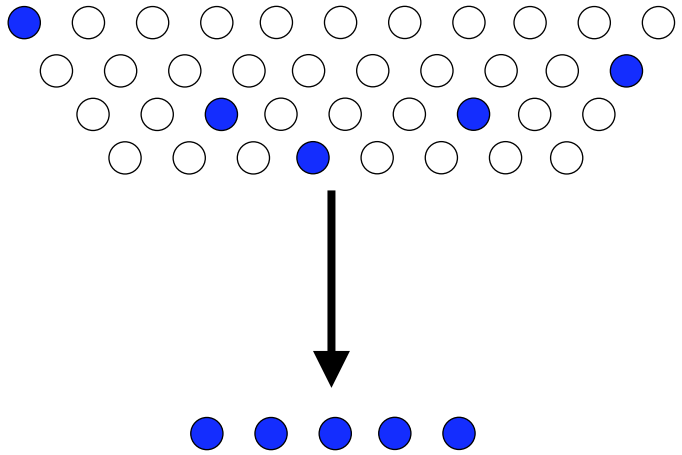


Oversampling

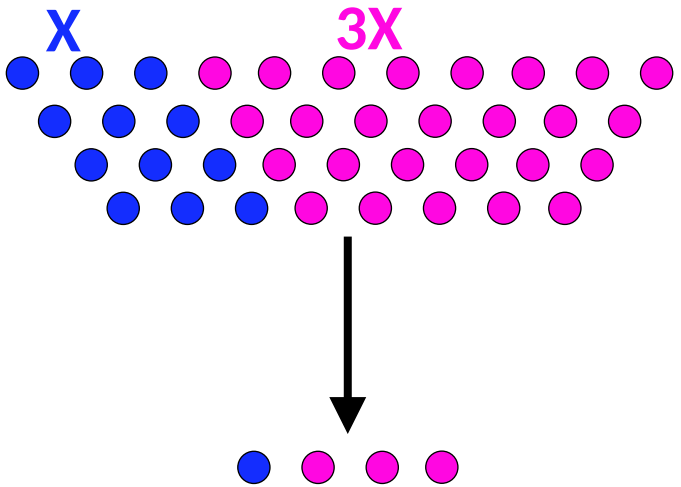


Balance the classes

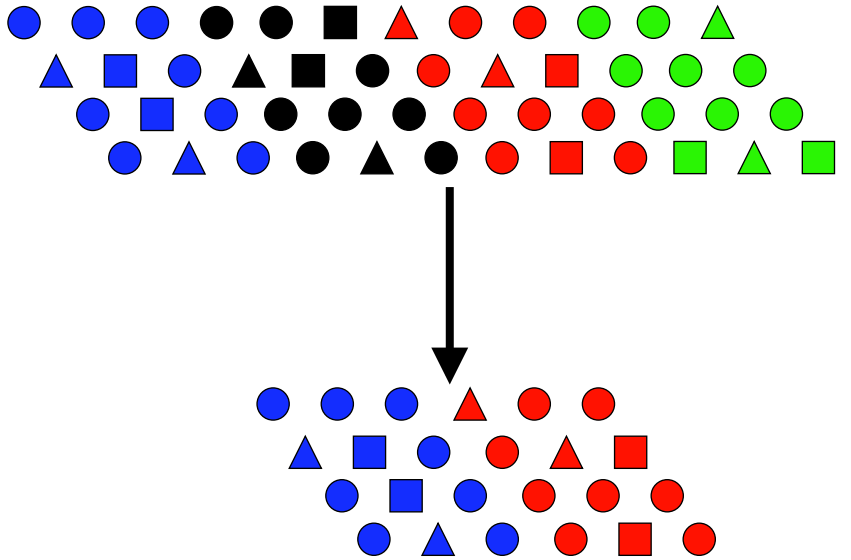
https://towardsdatascience.com/8-types-of-sampling-techniques-b21adcdd2124



Random Sampling



Stratified Sampling

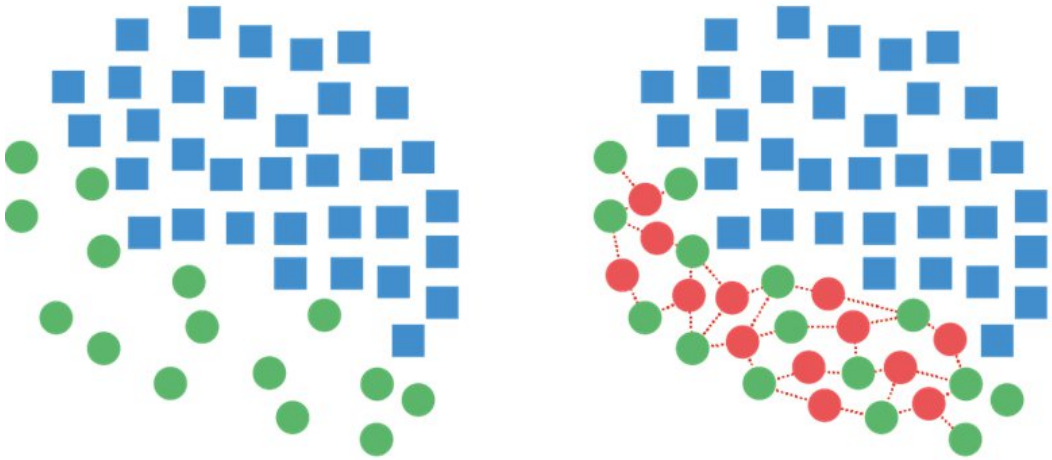


Cluster Sampling

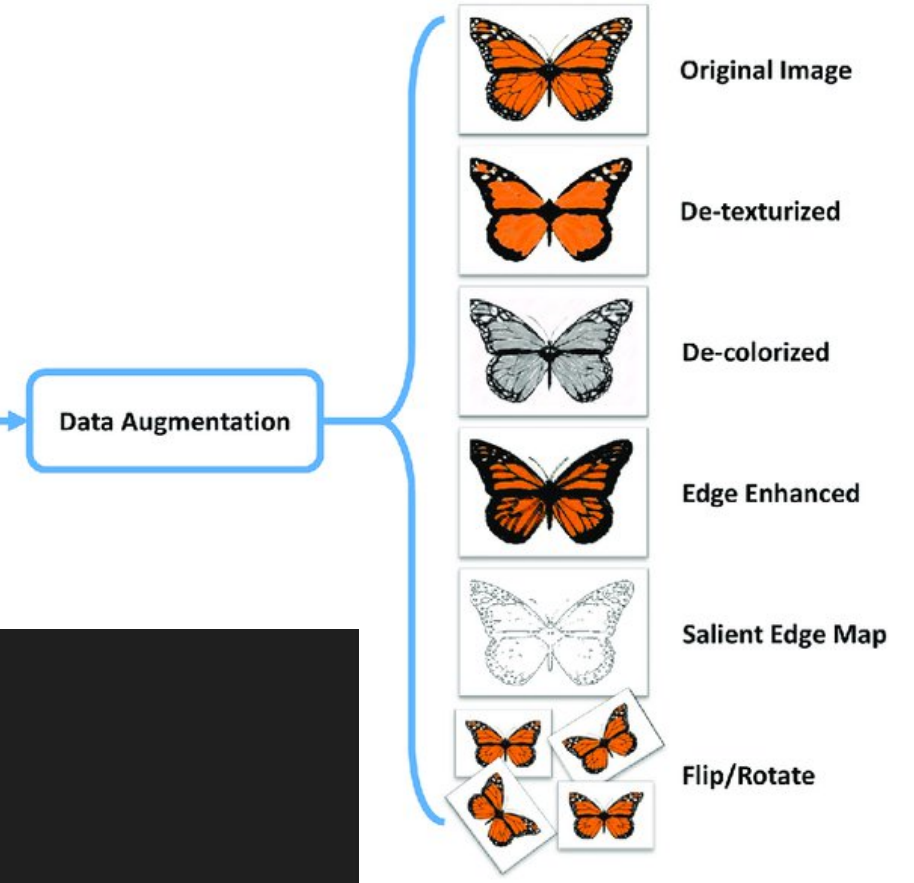
Under sampling

https://pub.towardsai.net/maximizing-the-impact-of-data-augmentation-effective-techniques-and-best-practices-c4cad9cd16e4
https://medium.com/@parthdholakiya180/smote-synthetic-minority-over-sampling-technique-4d5a5d69d720
https://pytorch.org/vision/0.9/transforms.html

Synthetic Minority Oversampling Technique



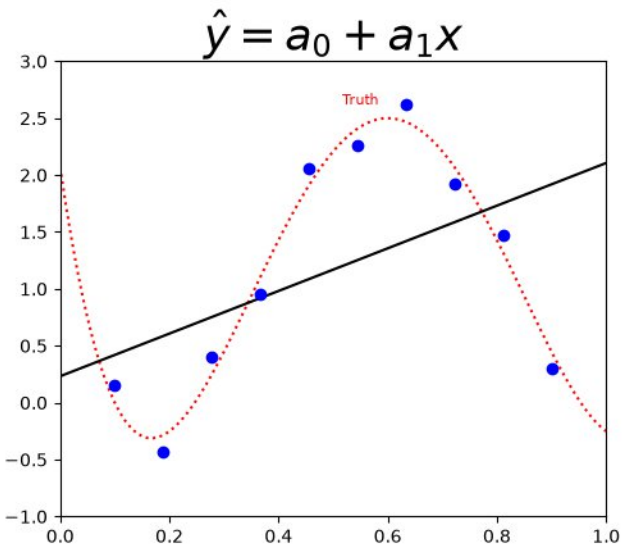


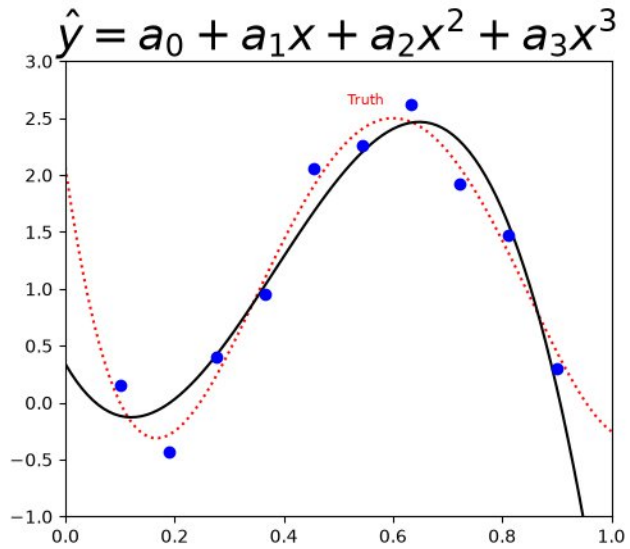


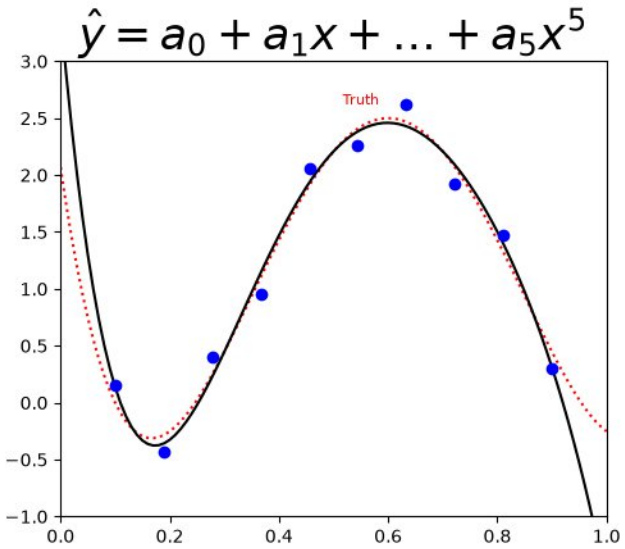

```python
# Define the transformations
transform = transforms.Compose([
    transforms.RandomHorizontalFlip(),
    transforms.RandomRotation(10),
    transforms.RandomResizedCrop(224),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])
```

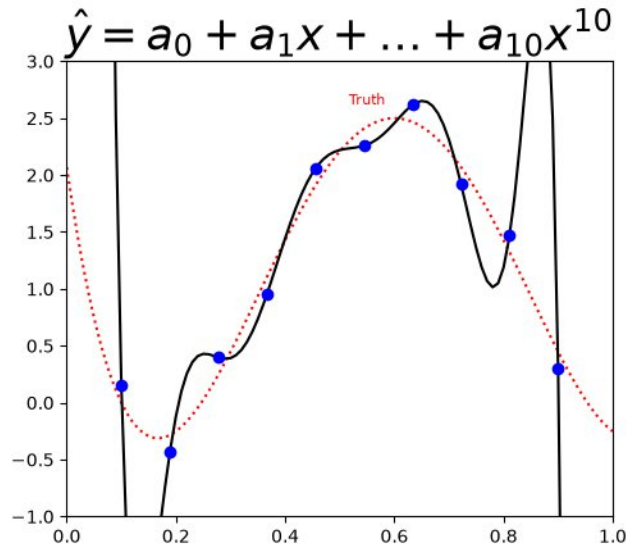
Over-Sampling/Data augmentation

1. What **work** can be done to **improve** the **data** used for training?
2. How can a model be **evaluated**?
3. Is it possible to make the training more **robust**?
4. Can we **benefit** from an **already** trained model?
5. Bonus: Good practices









Bias variance trade-off





Noise or bias ?



Bias



Noise

Noise or bias ?

There is no correlation between my model predictions and the label



Bias

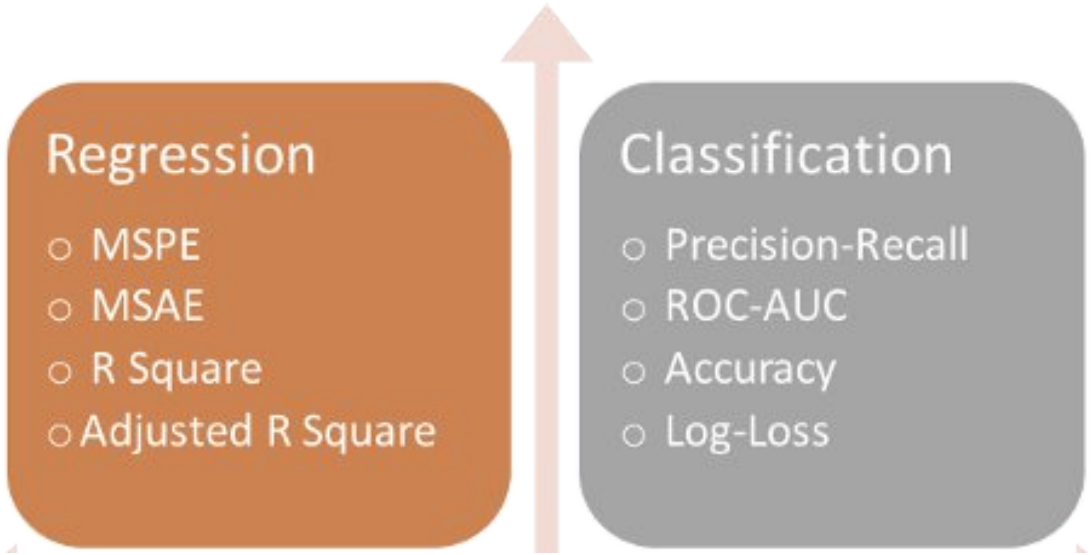
My model says « It's a cat » no matter the image I give it.



Noise

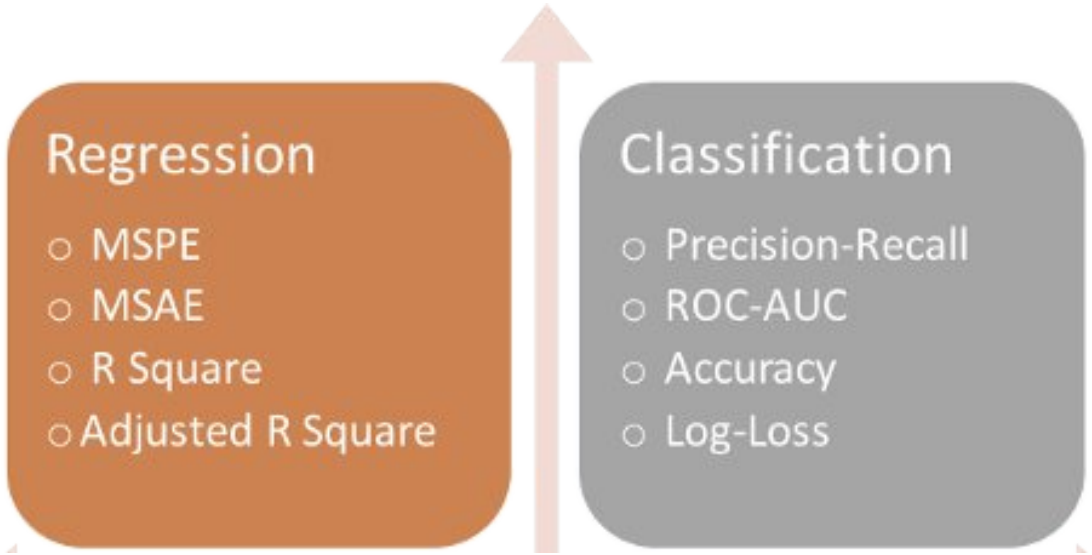
My model is not consistent in its answers and is mostly wrong.

Noise or bias ?



- **What is a metric?**
- **Is it different from a loss function?**

Metrics



- **What is a metric?**
- **Is it different from a loss function?**
  - **Differentiability**
  - **Training vs evaluation**
  - **Interpretability/Meaning**
  - **Number of samples**

Metrics

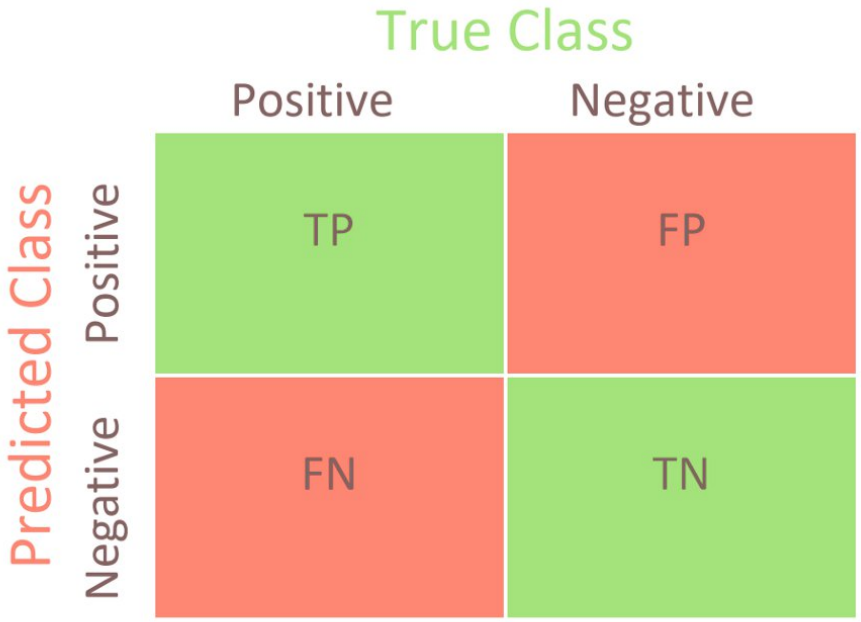




All negative : accuracy = 99% ✔ ?

New system for illness detection – the accuracy is not a good metric for this case

Metrics - Bad Choice



Confusion matrix

precision = $\frac{TP}{TP + FP}$

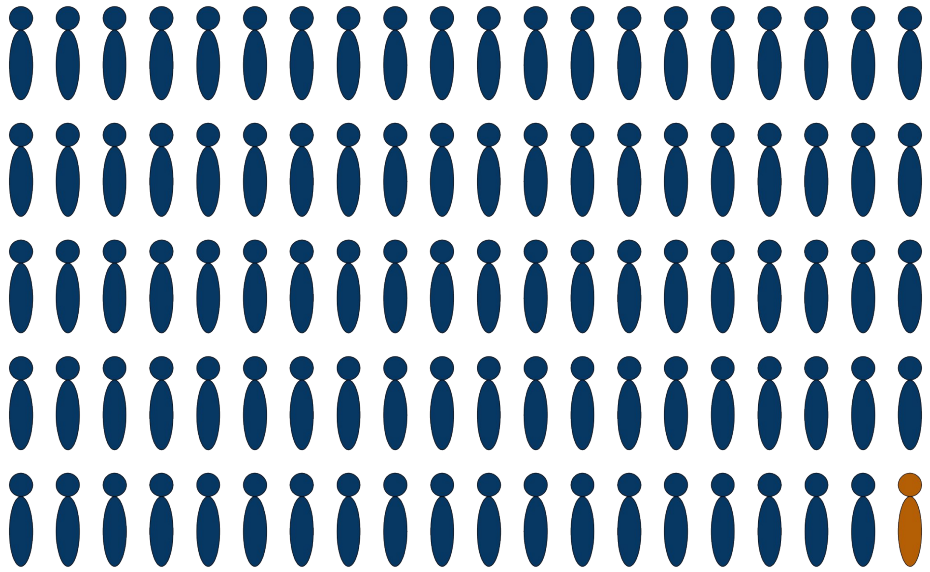
Above all positive prediction, how many are positive data

---

recall = $\frac{TP}{TP + FN}$

Above all positive data, how many have been predicted positive

Metrics - Precision/Recall





Predicted class \ True class	0	1
0	0	
1		99

$$Accuracy = \frac{99}{100} = 99\%$$

$$Precision = \frac{0}{0+0} = NaN$$

$$Recall = \frac{0}{0+1} = 0$$

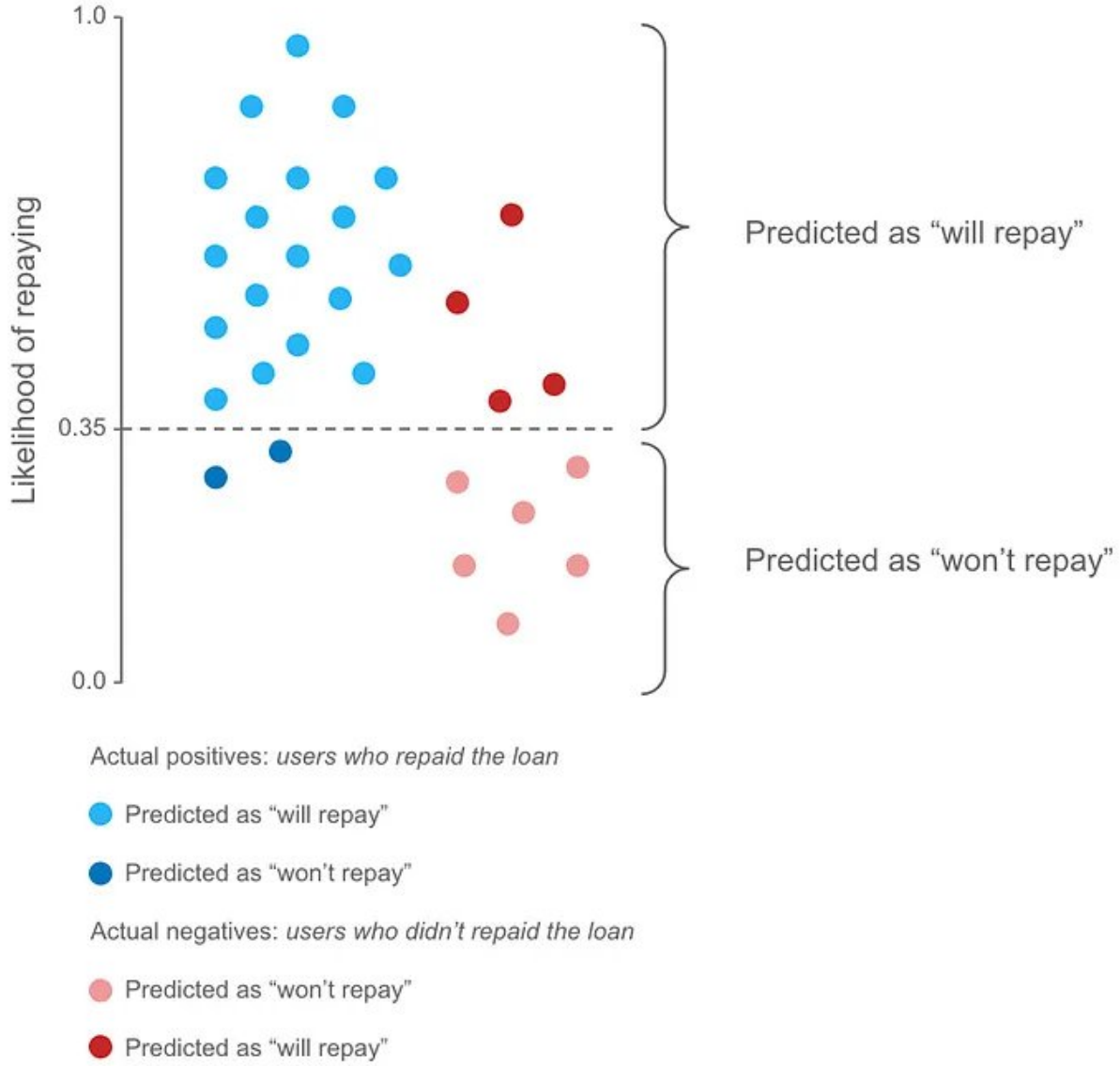


$$\text{precision} = \frac{TP}{TP+FP}$$

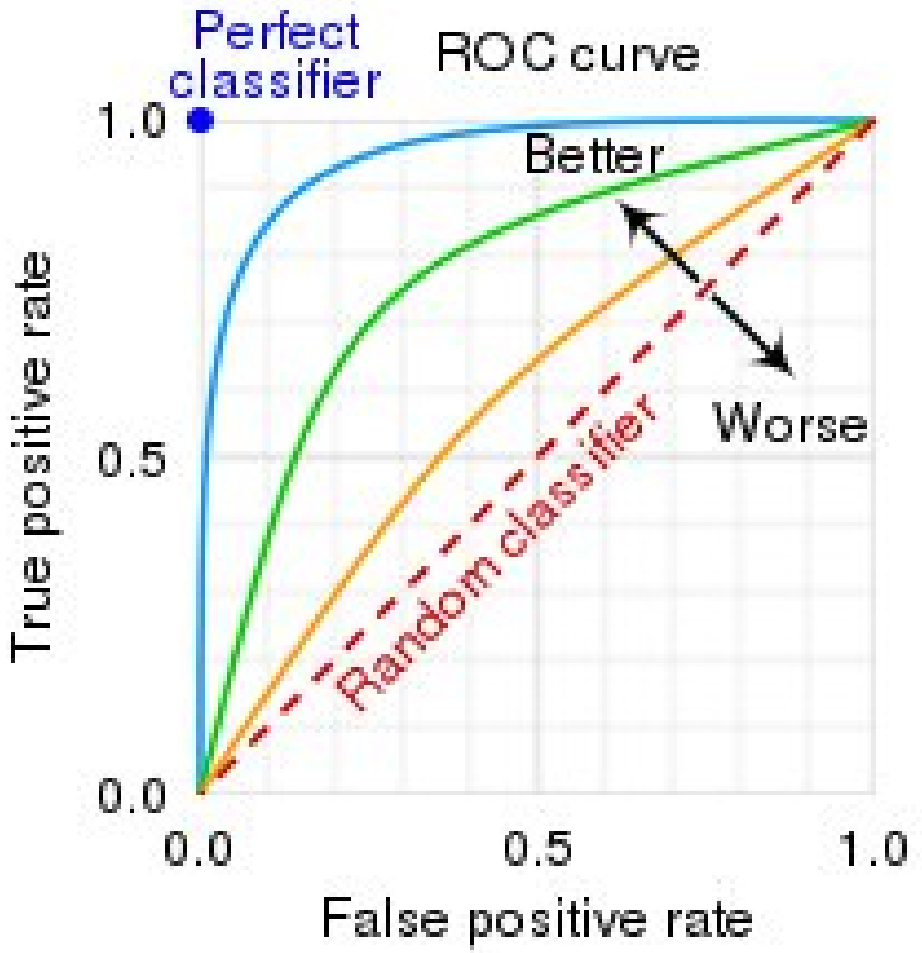
$$\text{recall} = \frac{TP}{TP+FN}$$

Metrics - Precision/Recall

https://towardsdatascience.com/understanding-the-roc-curve-in-three-visual-steps-795b1399481c



Metrics - ROC curve



Variation of the acceptance threshold of a class to obtain the curve

$$TPR = \frac{TP}{TP + FN}$$

Above all positive data, how many have been predicted positive

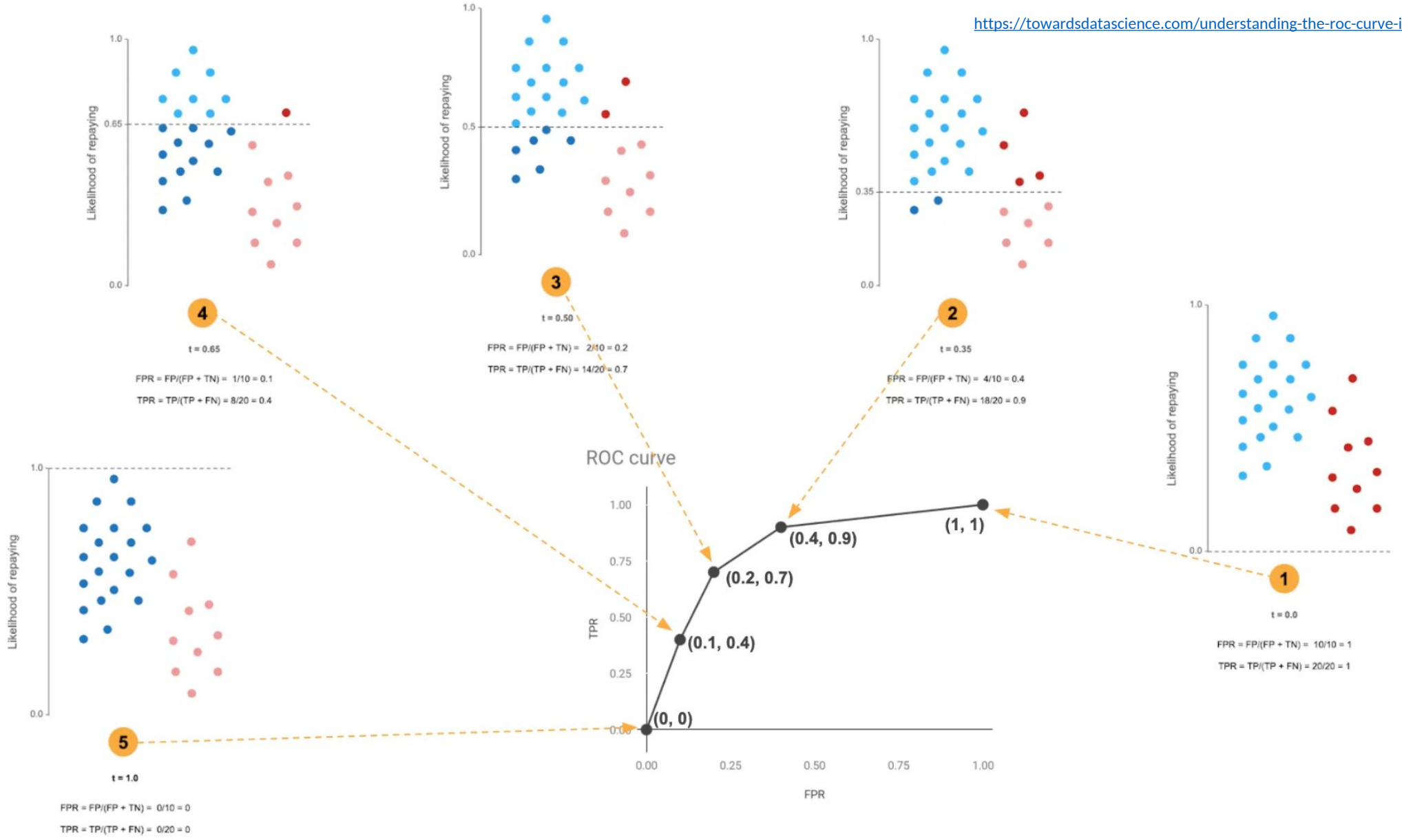
$$FPR = \frac{FP}{FP + TN}$$

Above all negative data, how many have been predicted positive

NB : ROC = Receiver Operating Characteristics

Metrics - ROC curve

https://towardsdatascience.com/understanding-the-roc-curve-in-three-visual-steps-795b1399481c



Metrics - ROC curve

https://huggingface.co/docs/transformers/perplexity

Perplexity :
Is model surprised to see this text?

$$\mathrm{PPL}(X) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta(x_i|x_{<i})\right\}$$

This person is innocent, she needs a lawyer

“ssistance”	- 1.5%
“ lawyer”	- 0.5%
“ car”	- 0.05%
…	
“ sea”	- 0.0001%

Metrics - Language models evaluation

https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

⚠ Perplexity is not All you Need !!

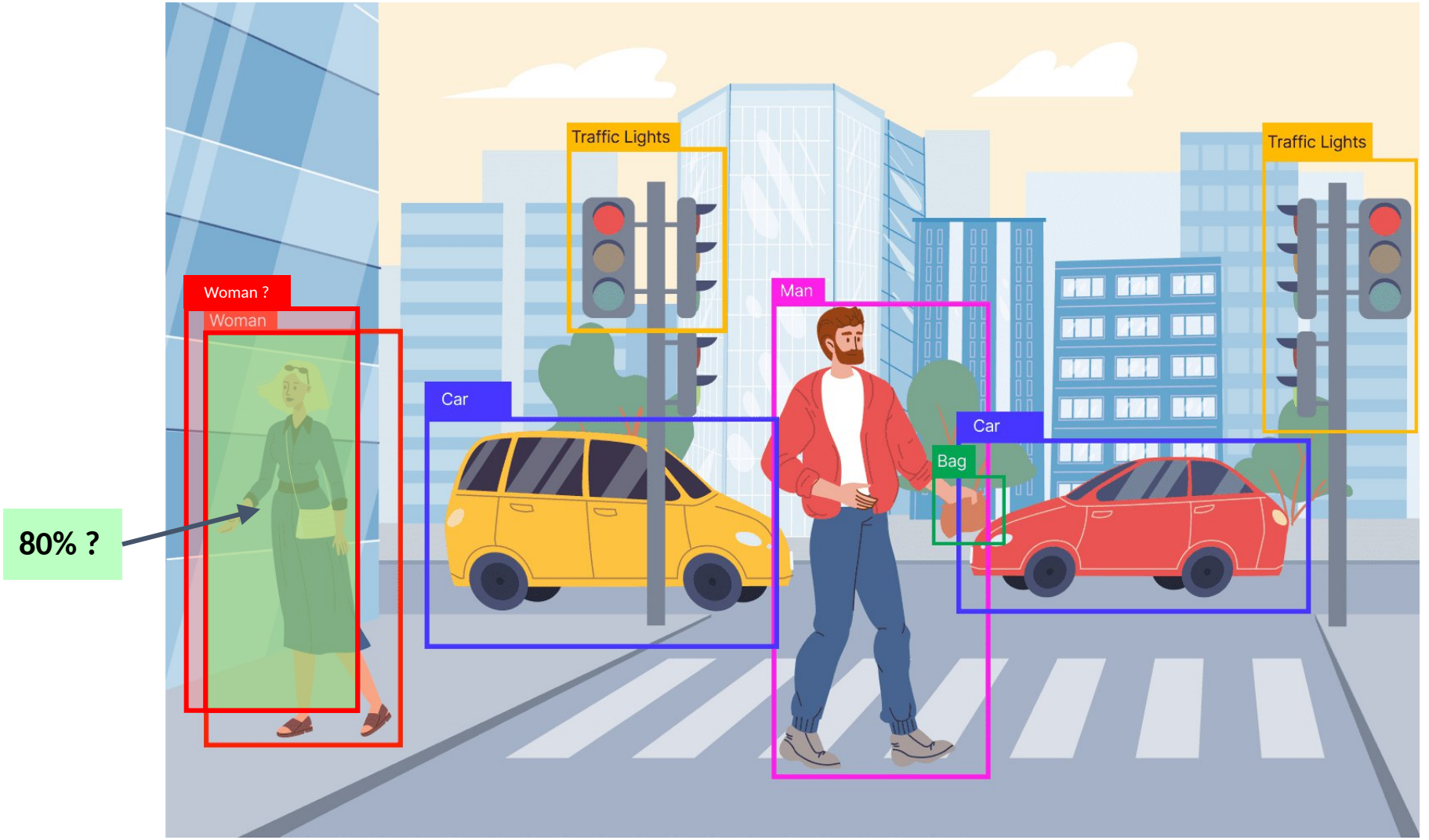
T	Model	Average	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K	Architecture	#Params (B)
🟦	moreh/MoMo-70B-lora-1.8.6-DPO	77.29	70.14	86.03	77.4	69	84.37	76.8	LlamaForCausalLM	72.29
🟦	moreh/MoMo-70B-lora-1.8.4-DPO	76.23	69.62	85.35	77.33	64.64	84.14	76.27	LlamaForCausalLM	72.29
🔶	TomGrc/FusionNet_7Bx2_MoE_14B	75.91	73.55	88.84	64.68	69.6	88.16	70.66	MixtralForCausalLM	12.88
🔶	Weyaxi/Helion-4x34B	75.48	69.71	85.28	77.33	63.91	84.37	72.25	MixtralForCausalLM	113.66
🔶	one-man-army/UNA-34Beagles-32K-bf16-v1	75.41	73.55	85.93	76.45	73.55	82.95	60.05	LlamaForCausalLM	34.39
🔶	Weyaxi/Cosmosis-3x34B	75.39	69.71	85.18	77.25	63.82	84.14	72.25	MixtralForCausalLM	87.24
🔶	Weyaxi/Bagel-Hermes-2x34b	75.1	69.8	85.26	77.24	64.82	84.77	68.69	MixtralForCausalLM	60.81
⭕	jondurbin/bagel-dpo-34b-v0.2	74.69	71.93	85.25	76.58	70.05	83.35	60.96	LlamaForCausalLM	34.39
🔶	jondurbin/nontoxic-bagel-34b-v0.2	74.69	72.44	85.64	76.41	72.7	82.48	58.45	LlamaForCausalLM	34.39
🔶	moreh/MoMo-70B-LoRA-V1.4	74.67	69.2	85.07	77.12	62.66	83.74	70.2	LlamaForCausalLM	72.29
🟦	udkai/Turdus	74.66	73.38	88.56	64.52	67.11	86.66	67.7	MistralForCausalLM	7.24

Be sure to look at Language models evaluation benchmarks which suite your needs.

The evaluation is complicated, there are many different tasks
It can depends on the targeted application
It is hard to be fair/objective without knowing the training data

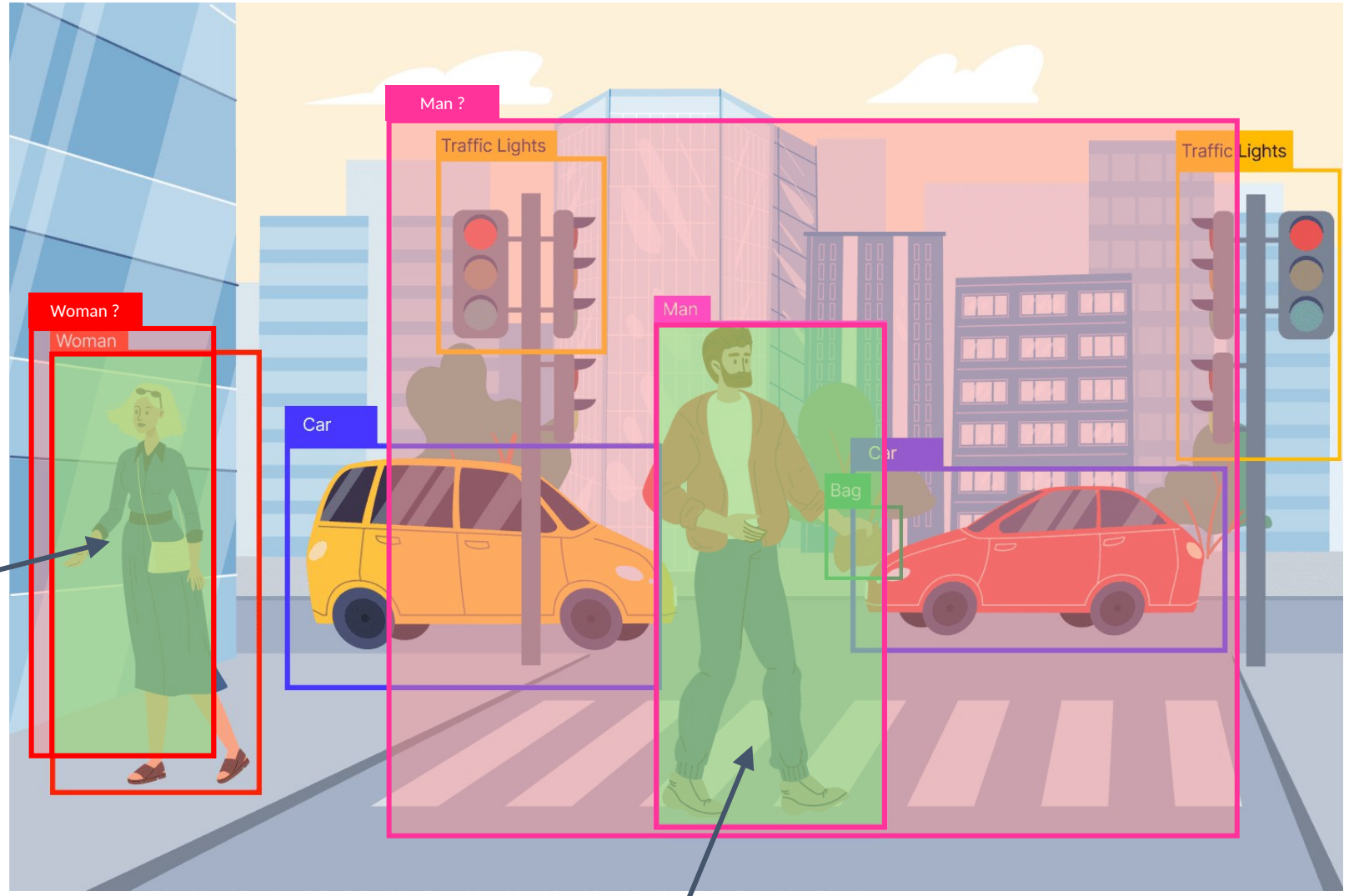
Metrics - Language models evaluation

https://deeplobe.ai/exploring-object-detection-applications-and-benefits/



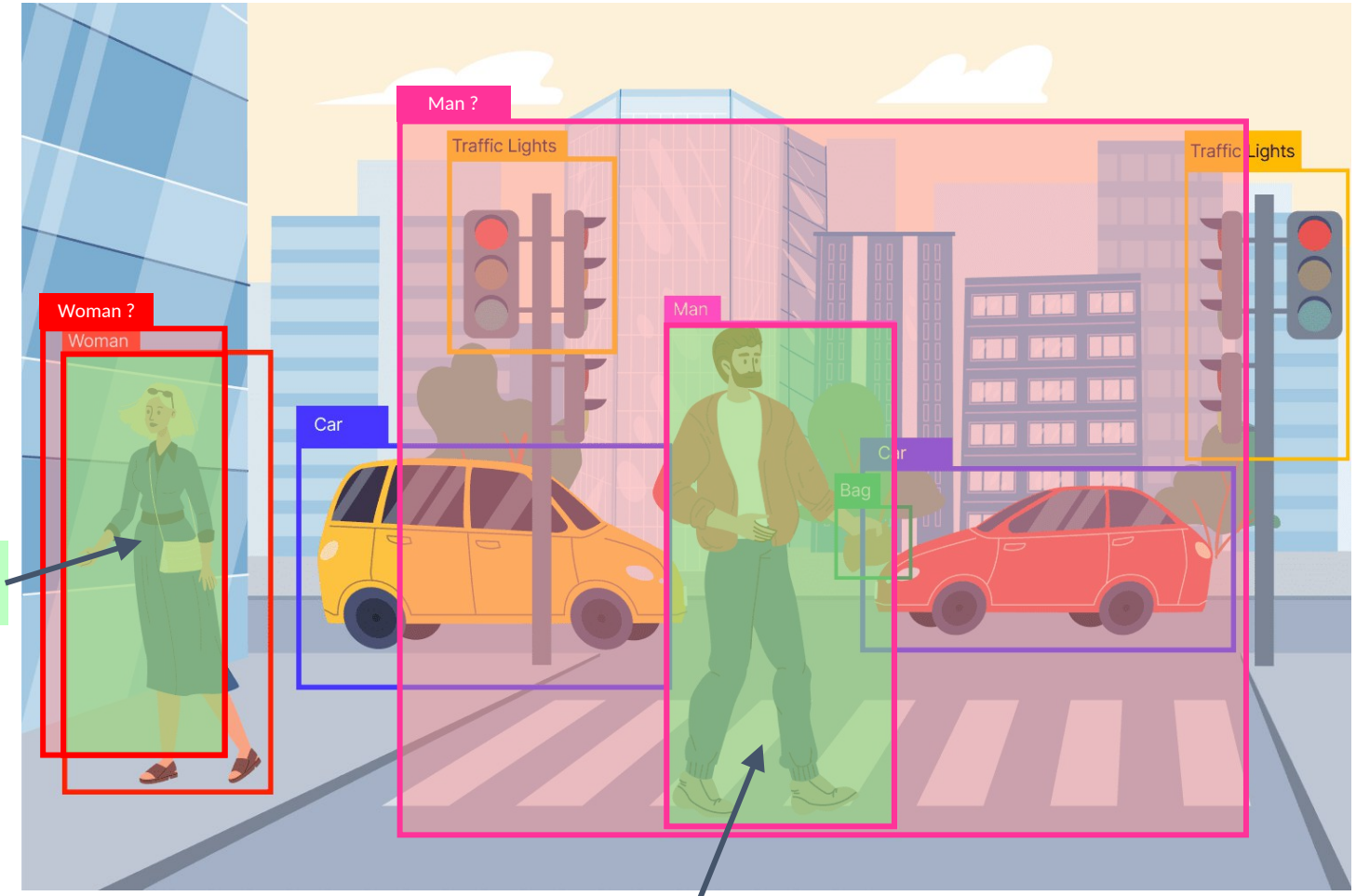
Metrics - Intersection over Union

https://deeplobe.ai/exploring-object-detection-applications-and-benefits/

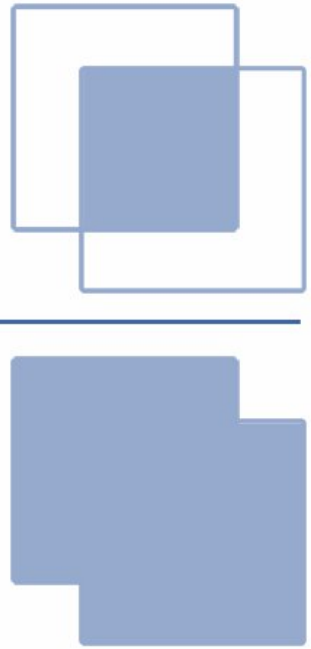


Metrics - Intersection over Union

https://deeplobe.ai/exploring-object-detection-applications-and-benefits/



$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



Intersection over Union

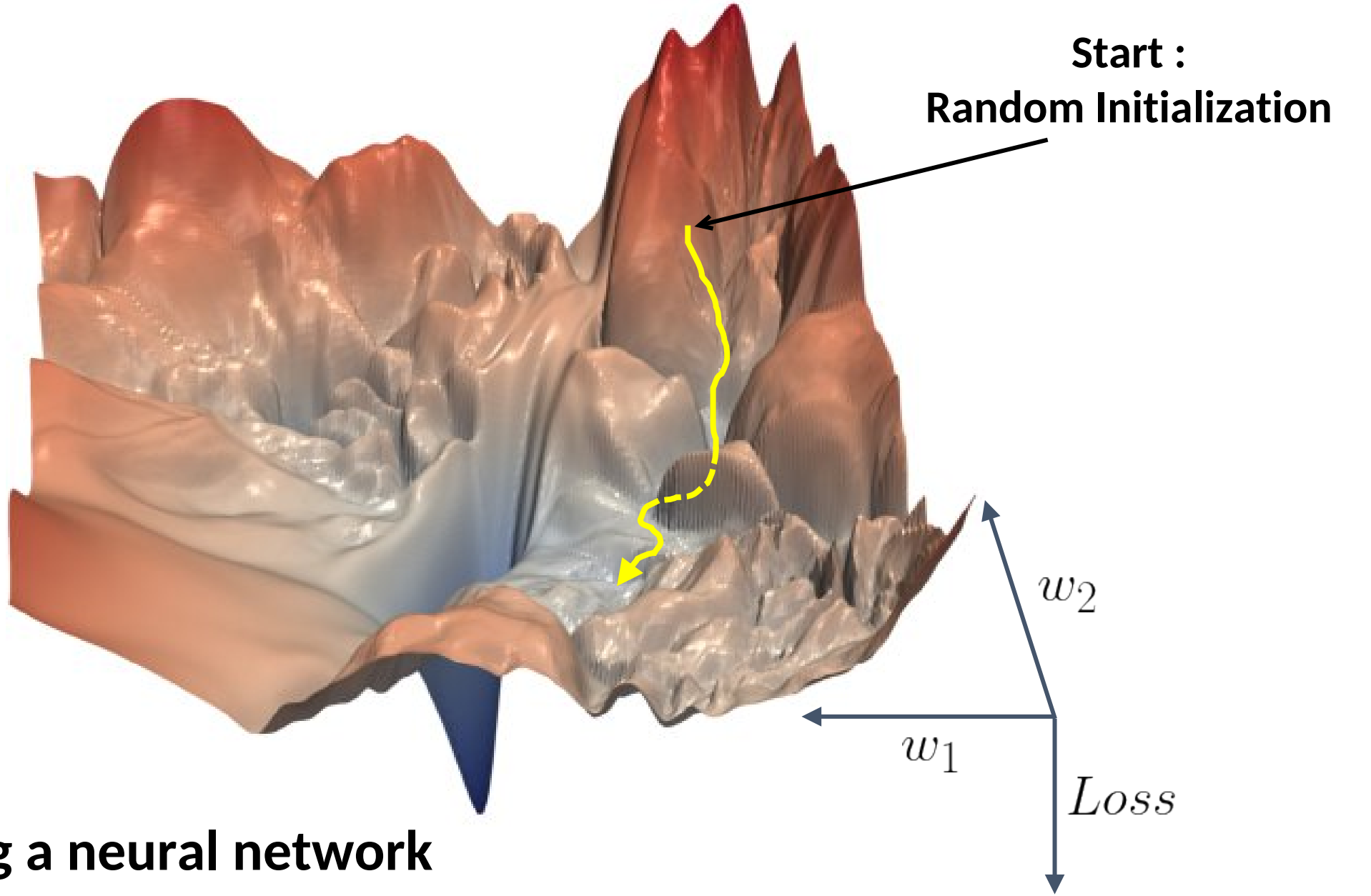
Metrics - Intersection over Union

1. What **work** can be done to **improve** the **data** used for training?
2. How can a model be **evaluated**?
3. Is it possible to make the training more **robust**?
4. Can we **benefit** from an **already** trained model?
5. Bonus: Good practices

https://losslandscape.com/

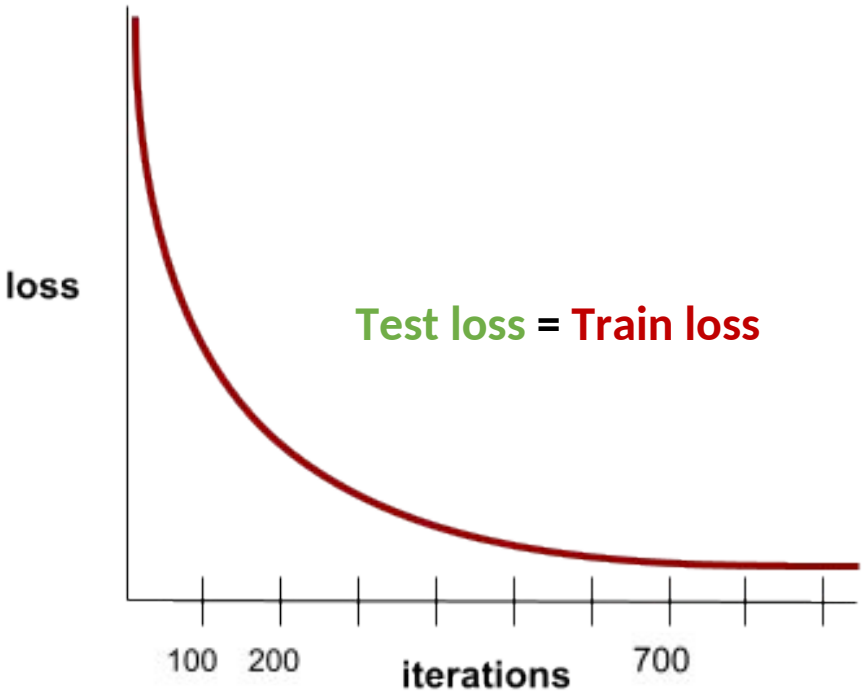
Gradient Descent





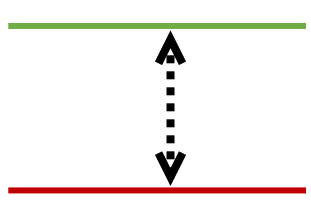
Training a neural network

Expectation

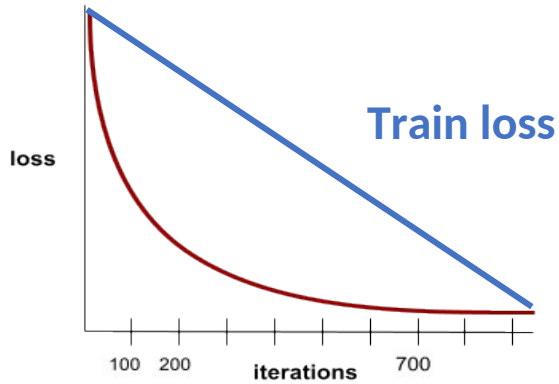


Test loss = Train loss

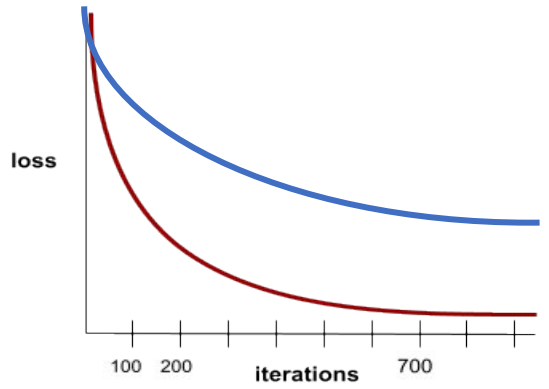
Reality



Generalization issue

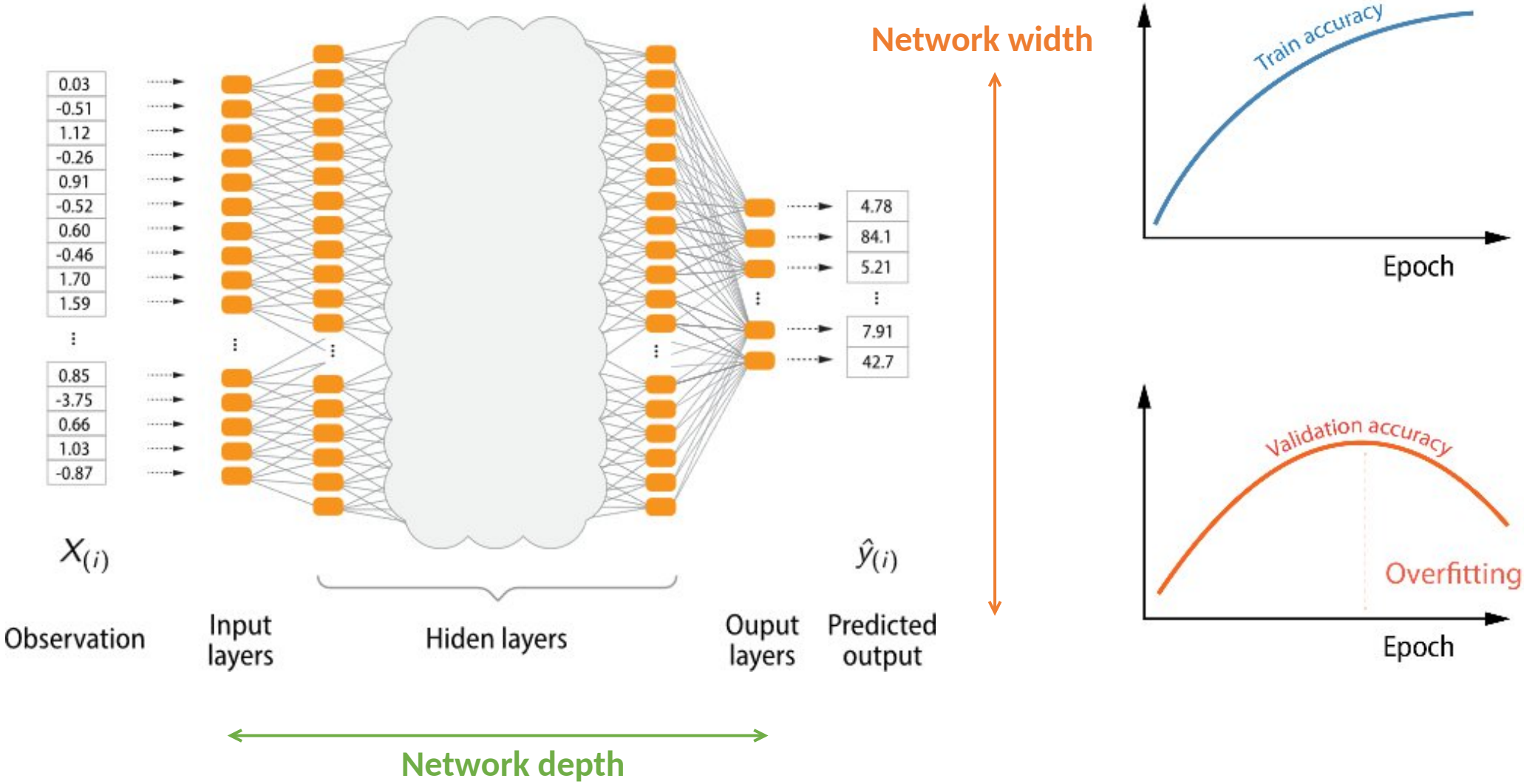


Slow convergence



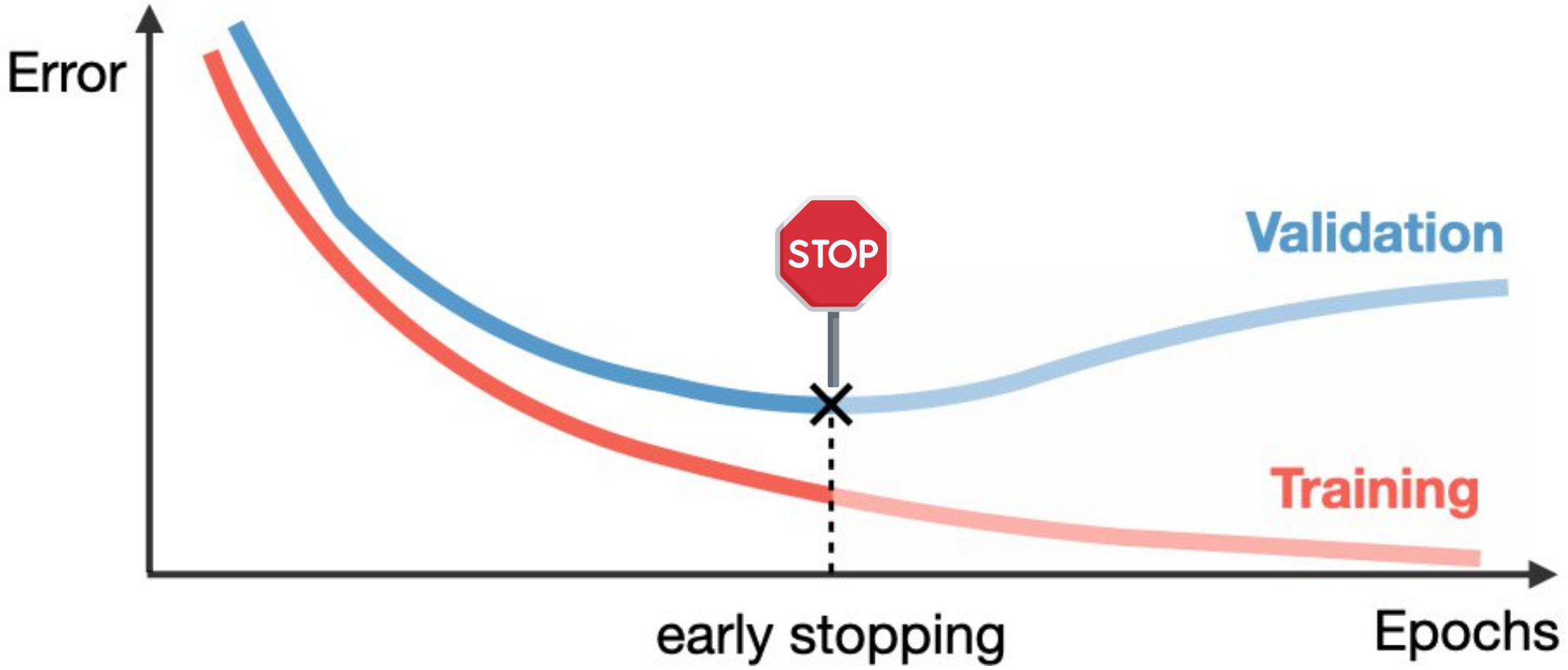
Convergence toward a local minimum

What happens during training ?



Model architecture and overfitting

https://wandb.ai/wandb_fc/french/reports/L-extraction-d-informations-partir-de-documents--Vmlldzo0OTI0OTU



Basic way : Early stopping

$$\Theta_{t+1} = \Theta_t - \eta \nabla_\Theta [\mathcal{L}(\hat{y}_i, y_i)]$$

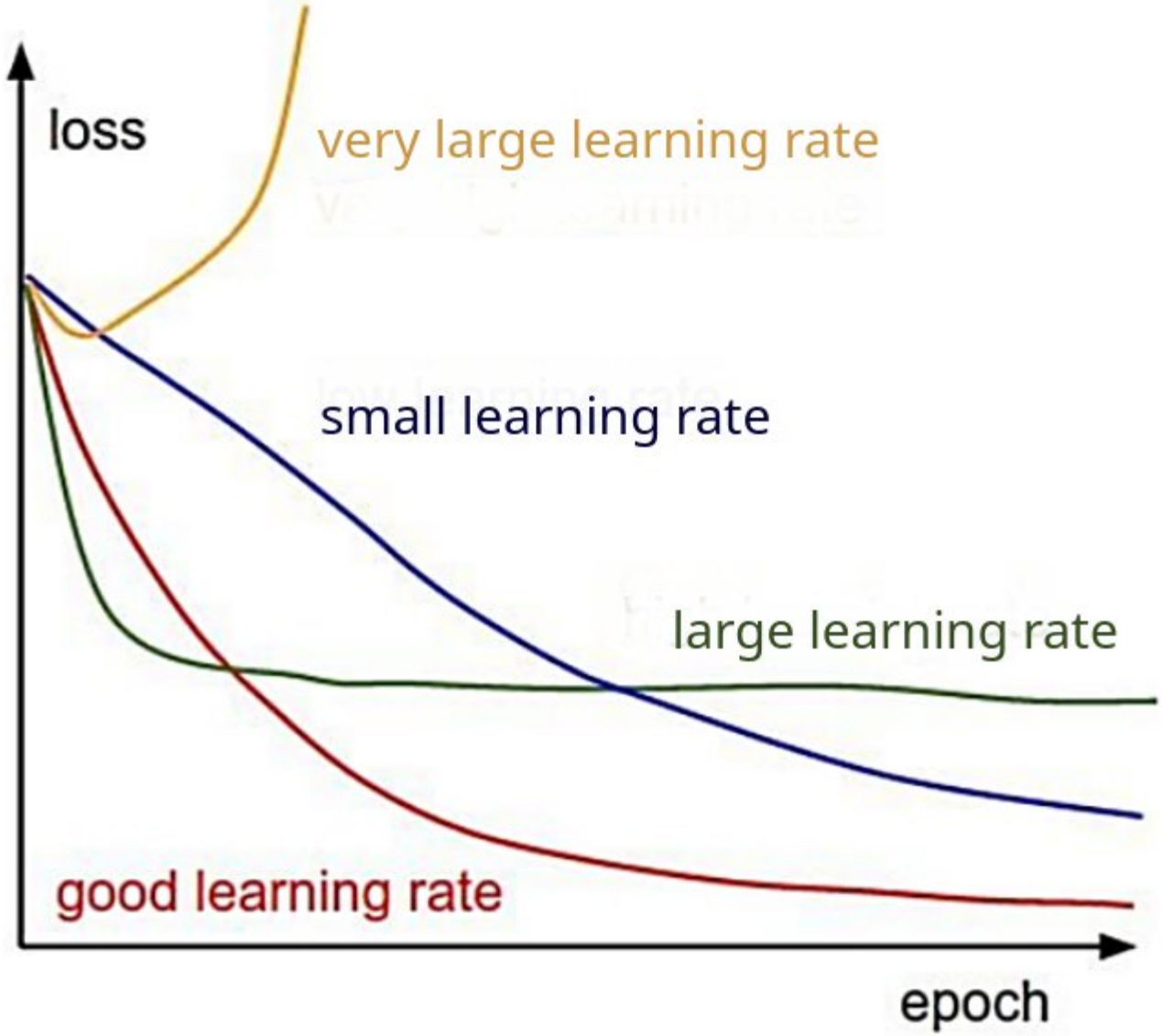
Updated weights = Weights before update − Learning rate * Gradient [Cost function (Prediction , Label)]

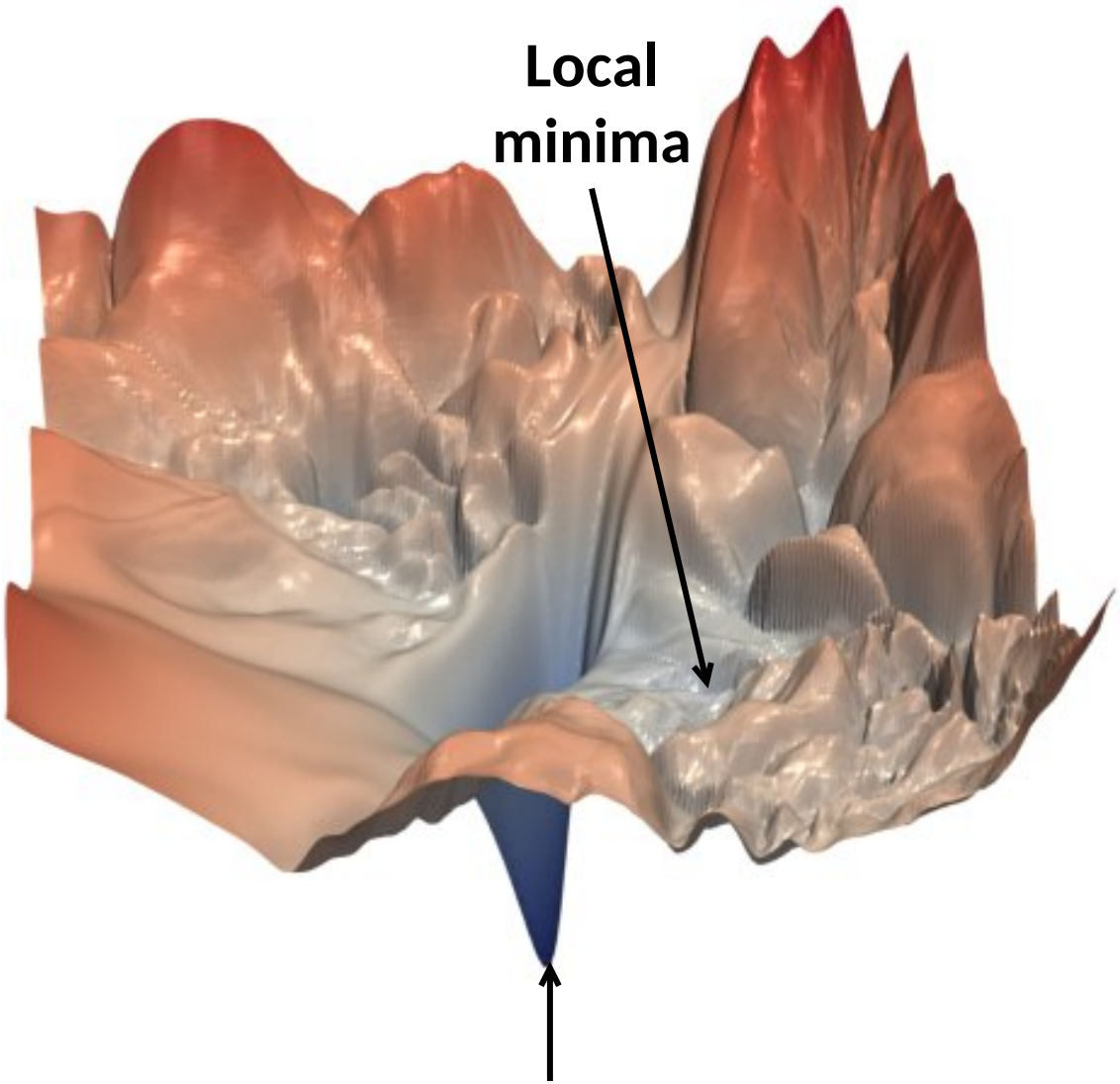
Weight update equation

Regularization - Loss function and weight update

Hyper-parameters

- Learning rate
- Regularization
- Optimizer
- Model architecture
- Batch size
- …

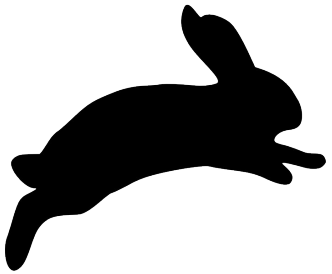




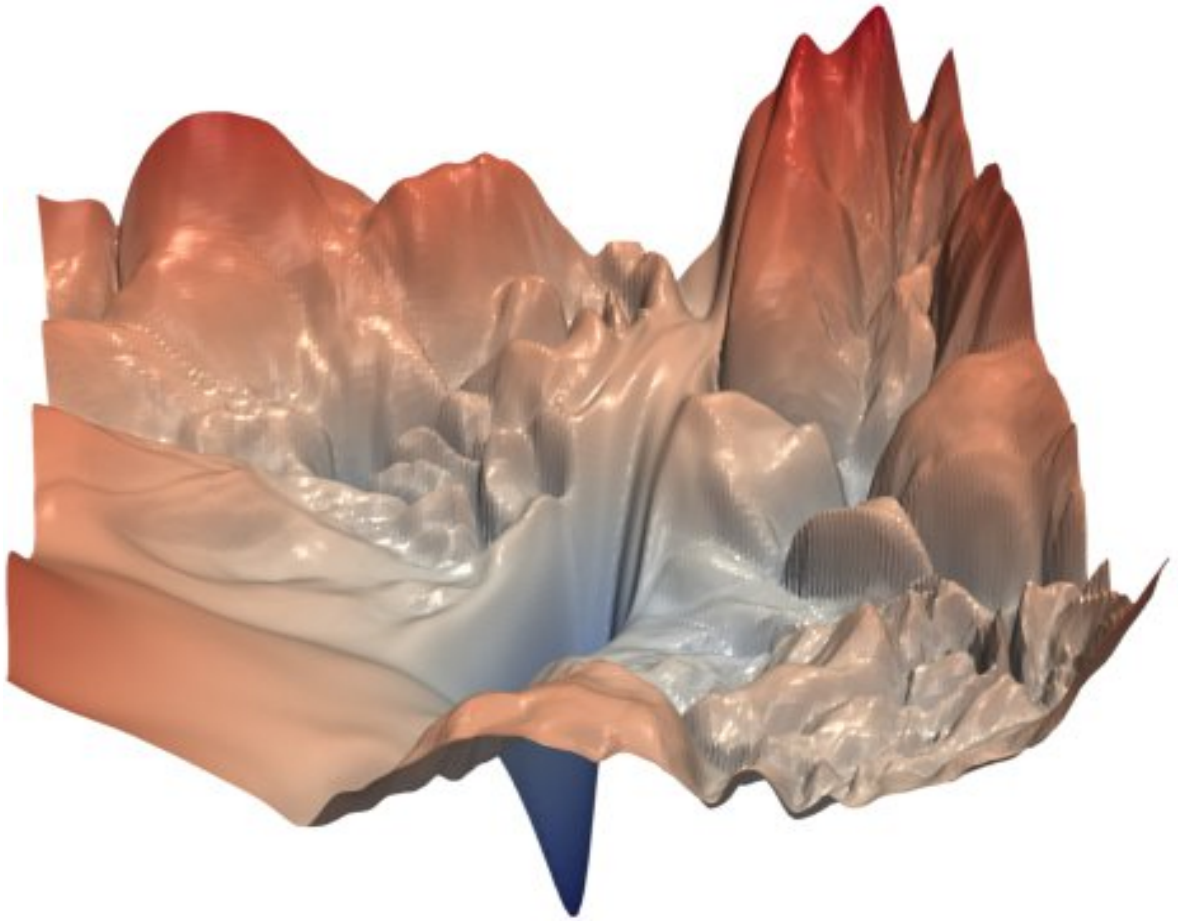
Learning rate

$$\underbrace{\theta}_{\text{Gradient}} = \theta - \underbrace{\eta}_{\text{LR}} \underbrace{\frac{\partial J(\theta)}{\partial \theta}}_{\text{Gradient}}$$

High Learning-rate



+ Larger steps (faster convergence)
+ Escape local minima
+ Good for early steps
- Risk skipping acceptable minima
- Risk "bouncing" over minima

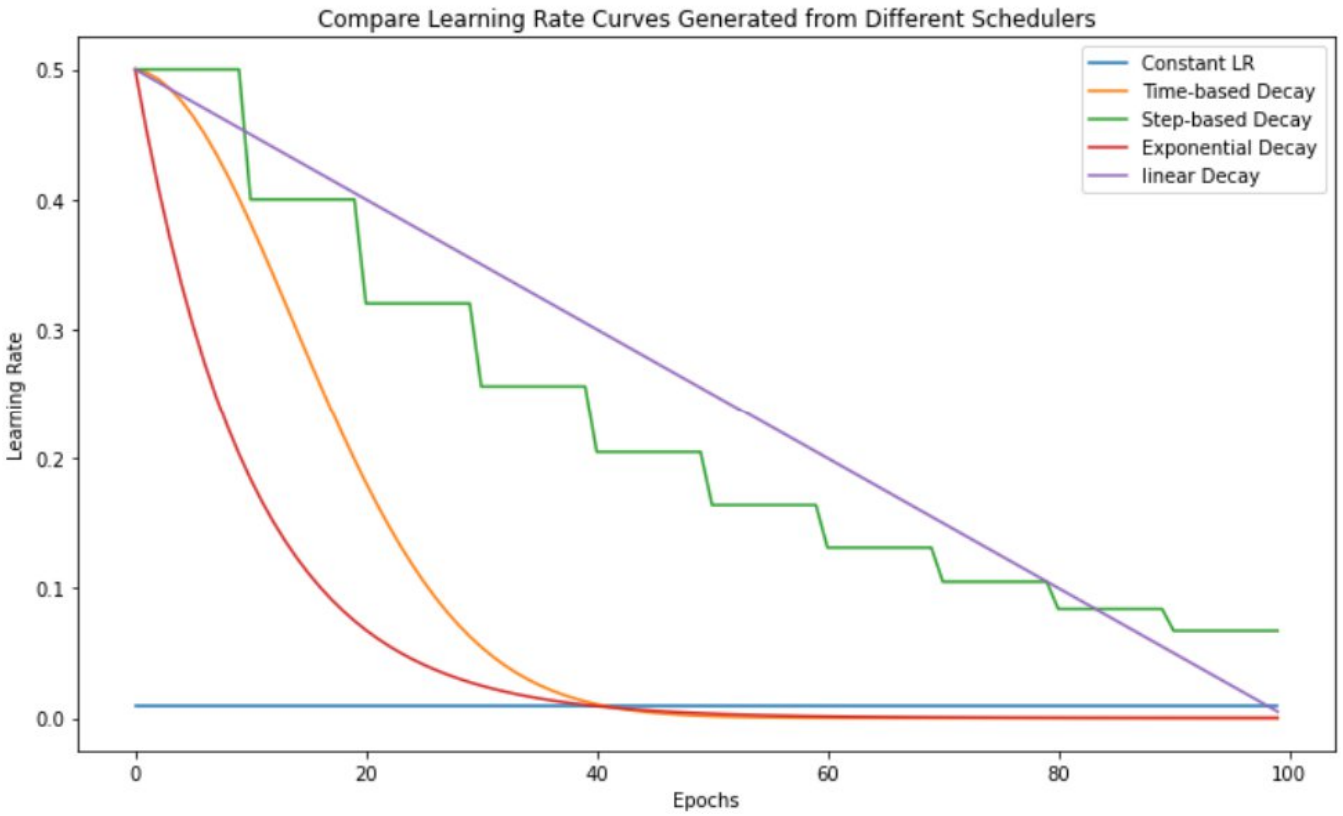


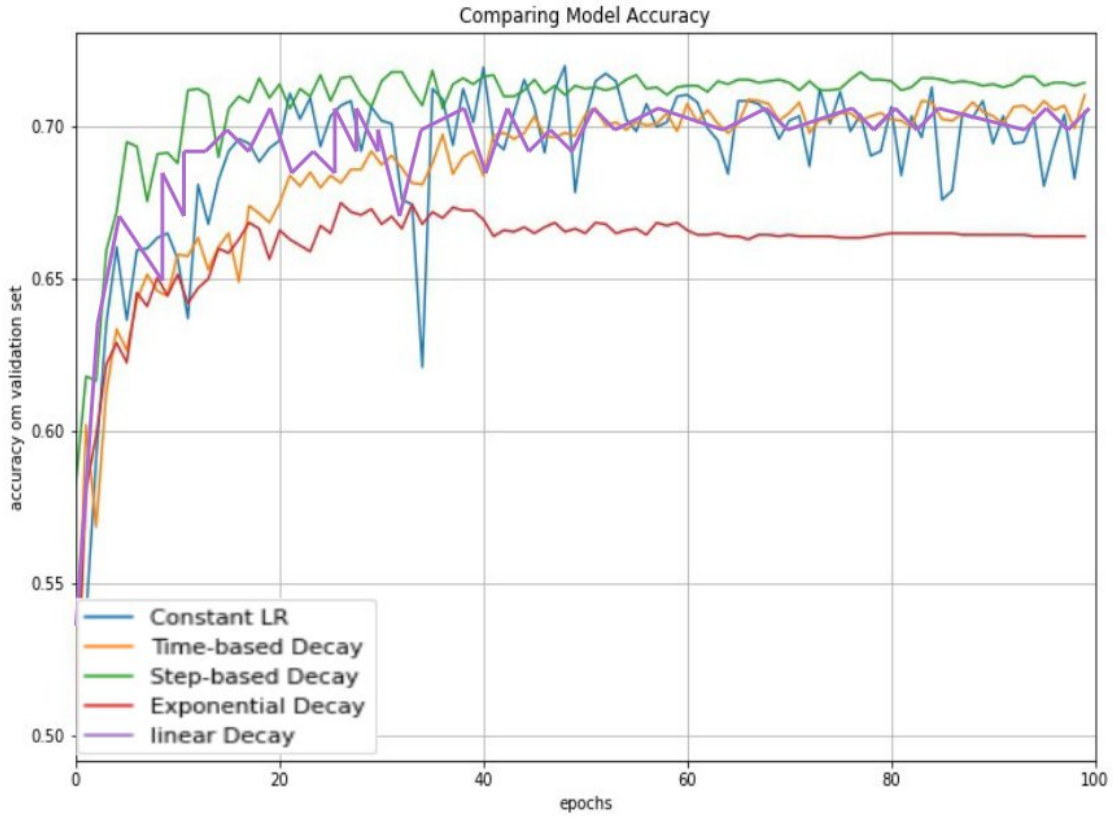
Small Learning-rate



+ Don't skip local minima
+ Converge towards bottom of minima
+ Good for later steps
- Smaller steps (slower convergence)
- Risk getting stuck in local minima

Learning rate

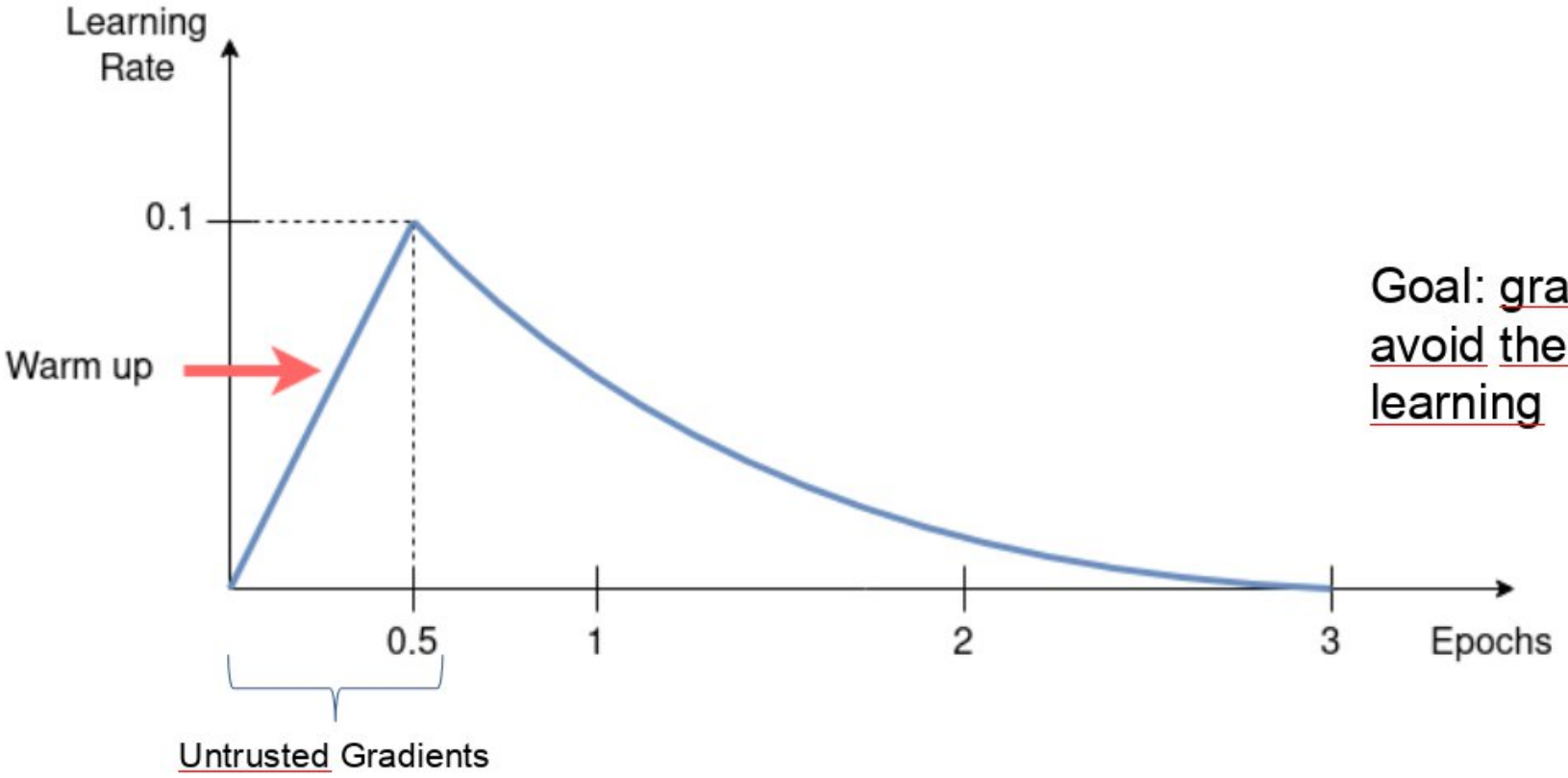




Learning rate scheduler : decay

$$\Theta_{t+1} = \Theta_t - \eta \nabla_\Theta [\mathcal{L}(\hat{y}_i, y_i)]$$

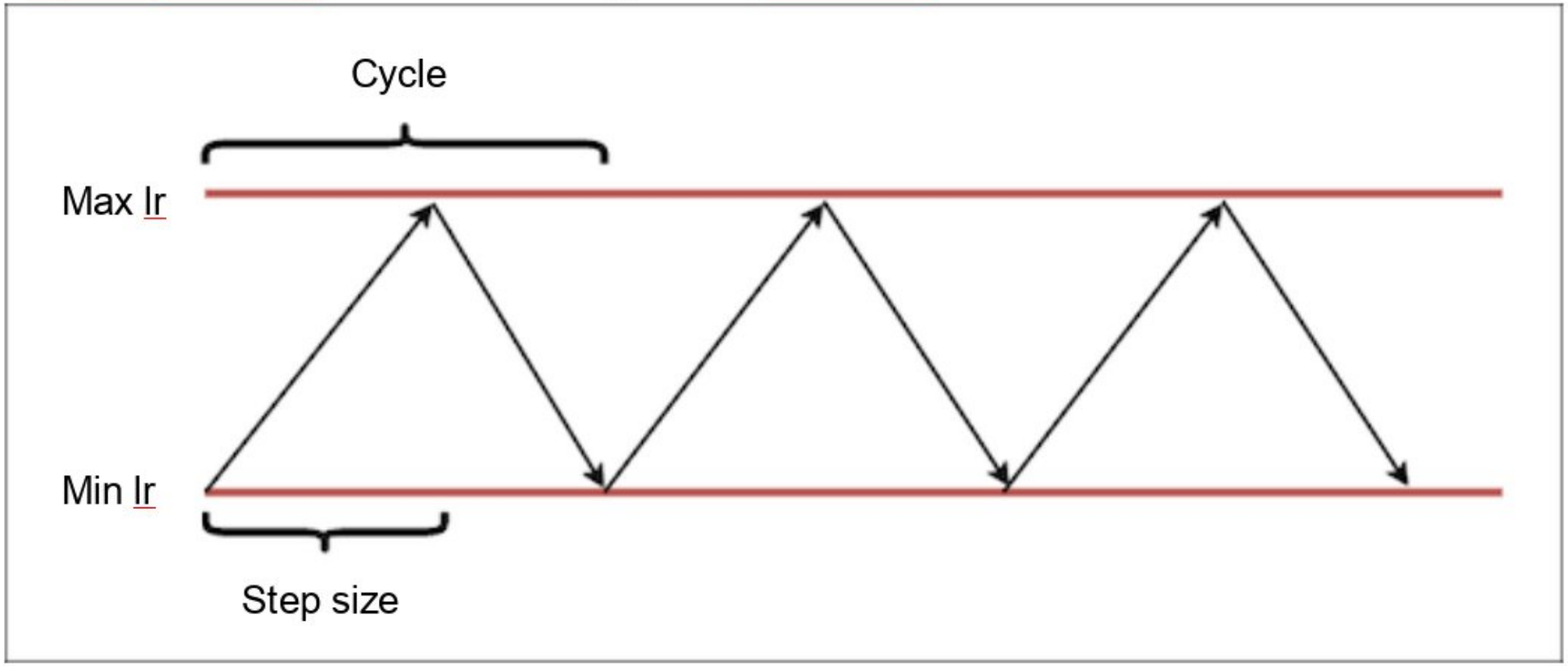
Problems : The first iterations have too much effect on the model (**significant losses**, high gradients, etc.)
A high learning rate can cause strong instability or divergence



Goal: gradually increase the learning rate to avoid the risk of divergence at the start of learning

Learning rate scheduler : warmup

Cyclical Learning Rates for Training Neural Networks - Leslie N. Smith 2017

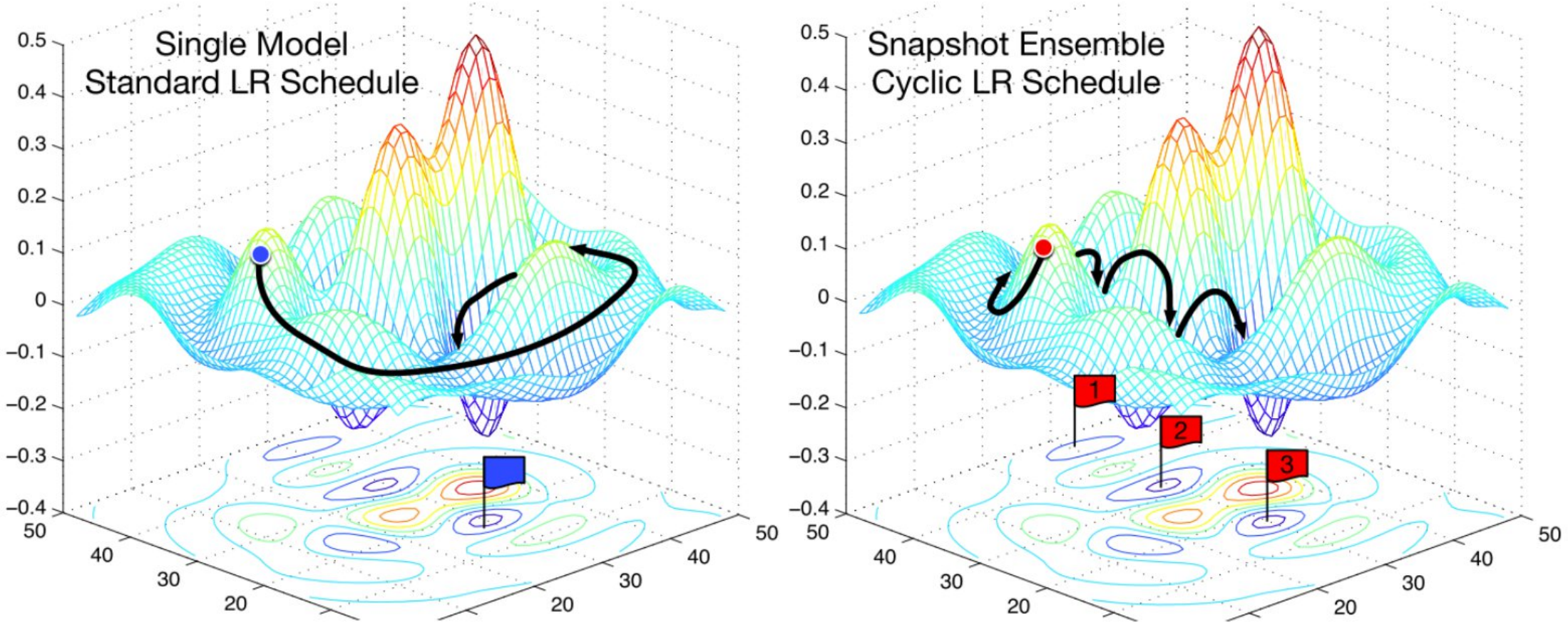


Paramètres :

- Step_size = x * epoch (2 ≤ x ≤ 10)
- Base_lr -> min convergence value
- max_lr -> max convergence value

Succession of warmups and learning rate decays

Cyclic Learning Rate Scheduler



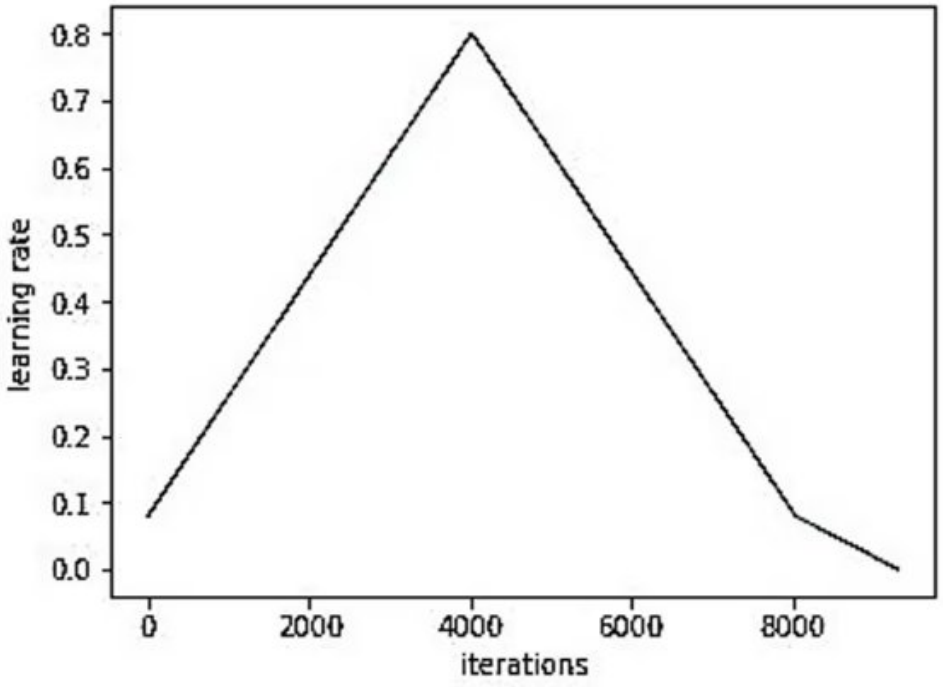
SNAPSHOT ENSEMBLES: TRAIN 1, GET M FOR FREE

Gao Huang, Yixuan Li, Geoff Pleiss

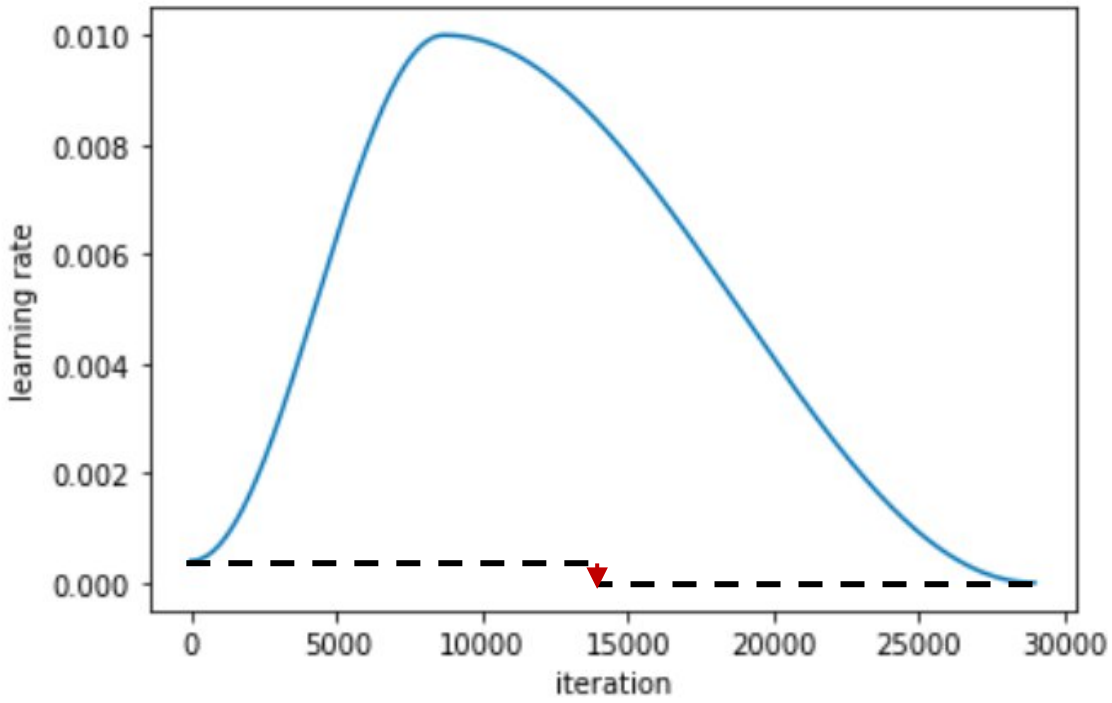
Cyclic Learning Rate

A disciplined approach to neural network hyper-parameters - *Leslie N. Smith*

Proposition initiale

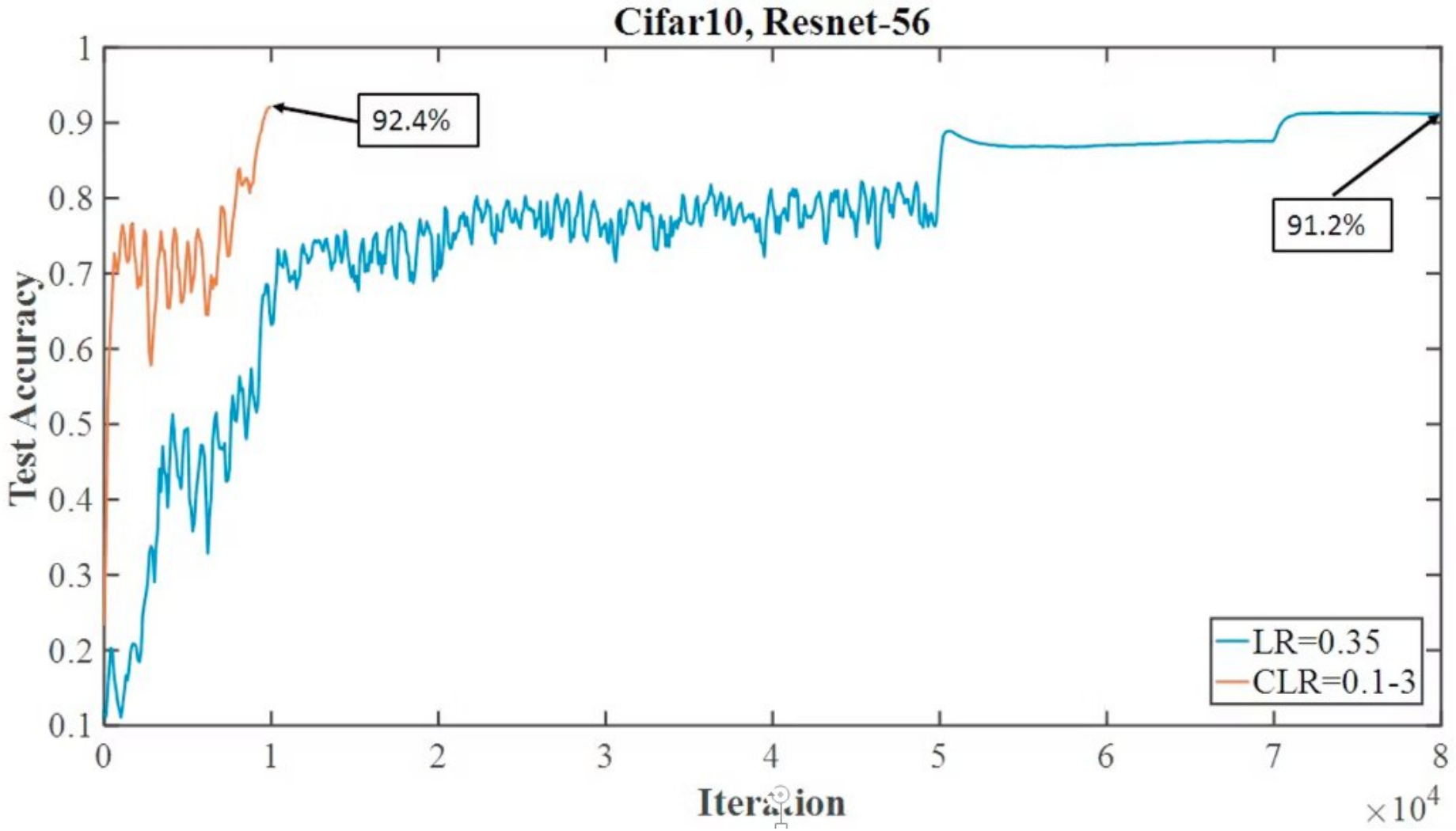


cosine annealing : Recommandation par FastAI



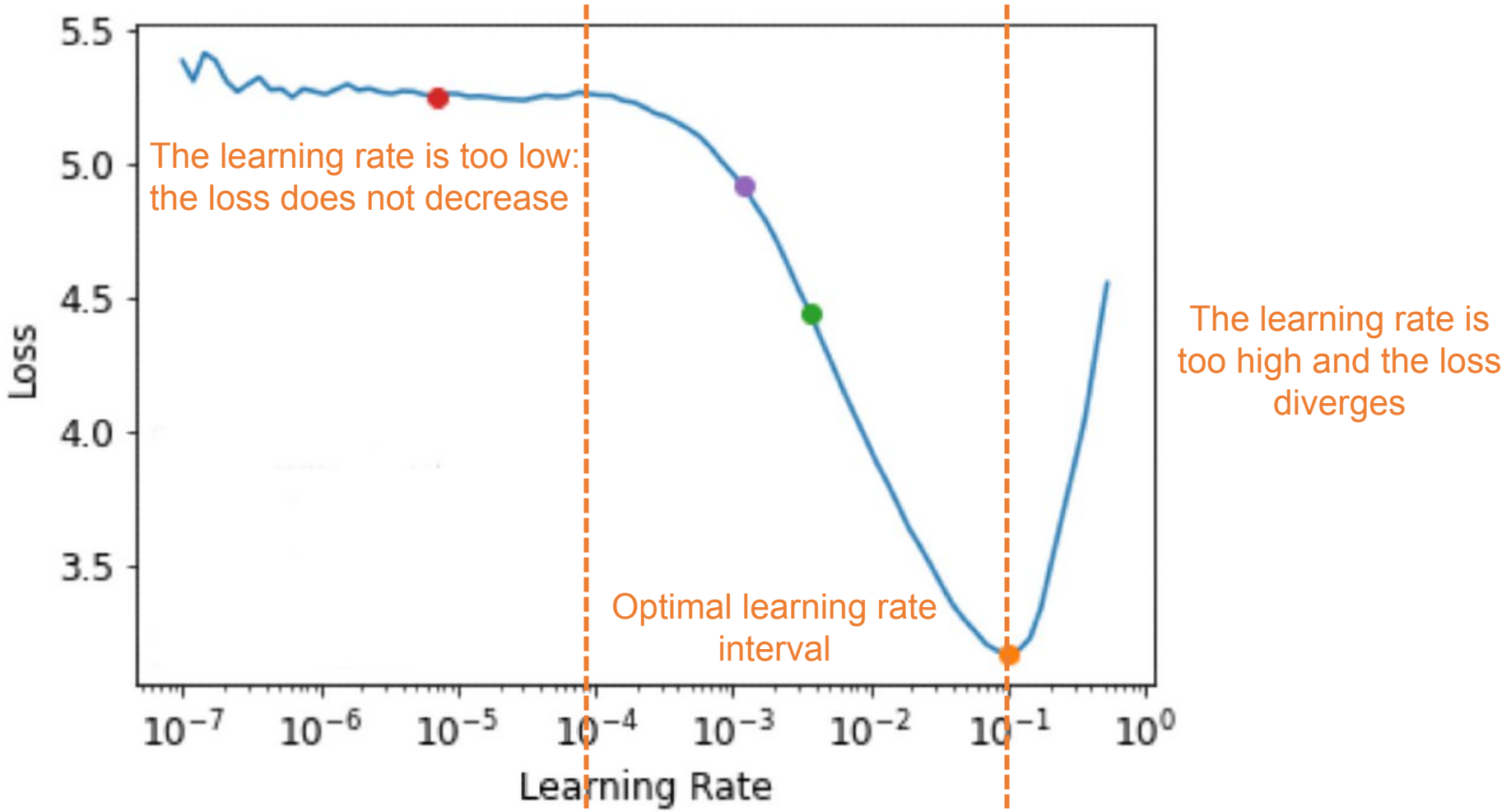
One Cycle is enough

Faster convergence for equivalent final precision



Cyclic learning rate scheduler

Goal: Find the **optimal learning rate** values for your model, particularly for **the maximum value** of a *cyclic scheduler*



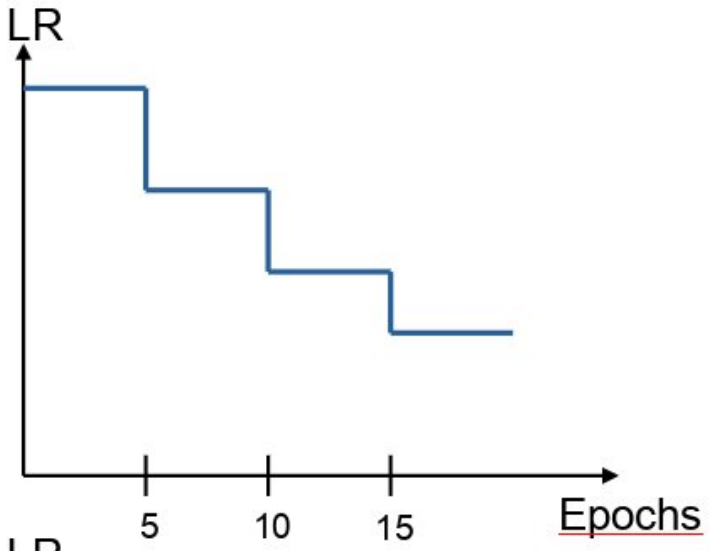
Learning Rate Finder

Each **scheduler** has *its own settings*

```
import torch.optim as opt

scheduler = opt.lr_scheduler.StepLR(optimizer, step_size=5, gamma=0.1)

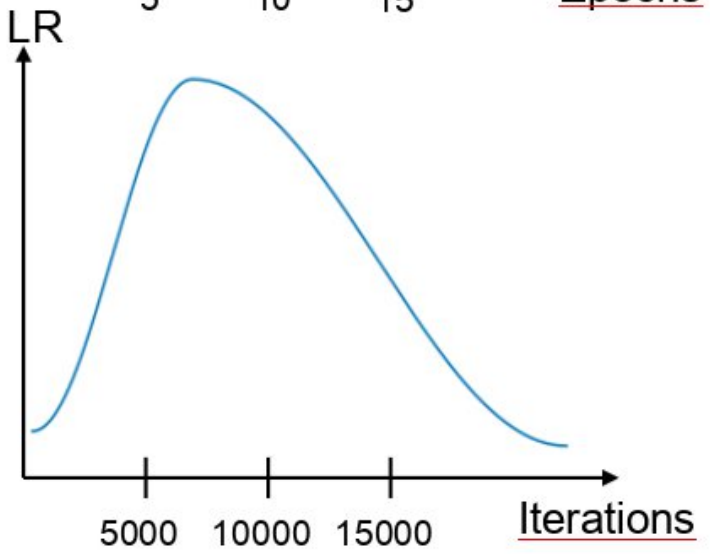
for epoch in range(100):
        train(...)
        validate(...)
        scheduler.step()
```




```
import torch.optim as opt

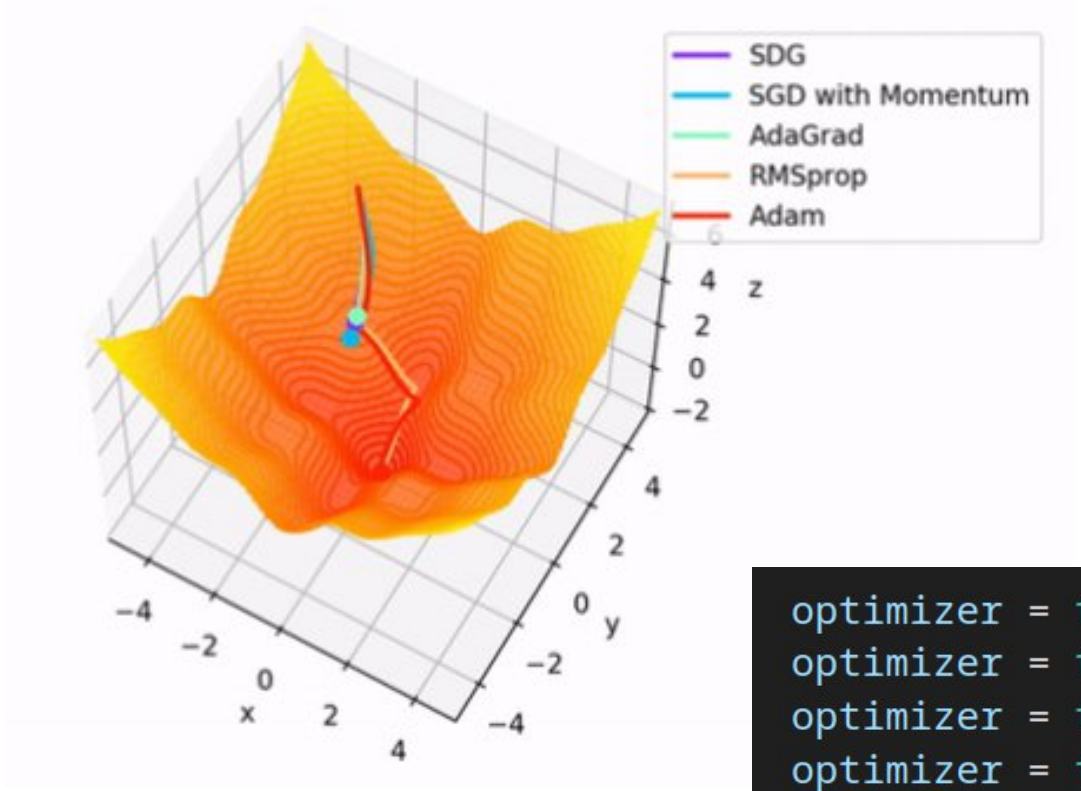
scheduler = opt.lr_scheduler.CyclicLR(optimizer, base_lr=0.01, max_lr=0.1)

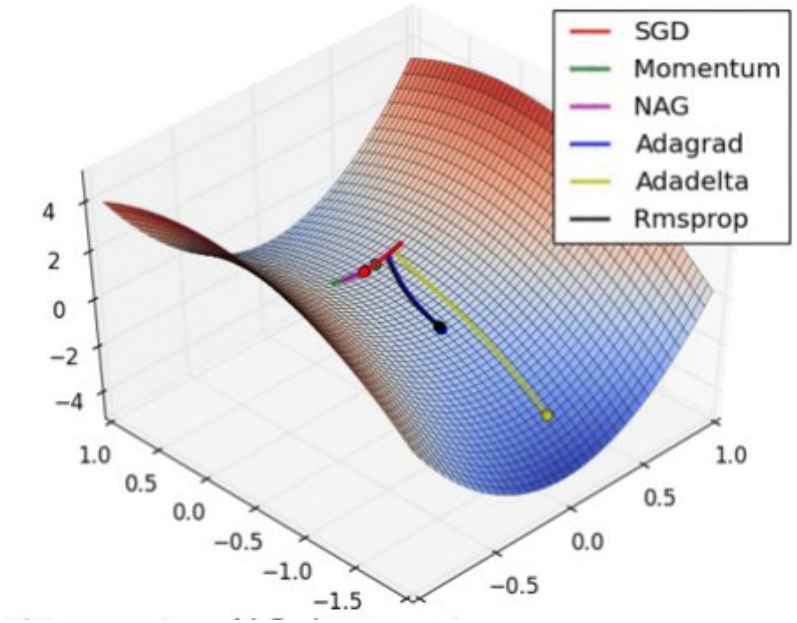

for epoch in range(10):
        for batch in data_loader:
                train_batch(...)
                scheduler.step()
```



Learning rate scheduler implementation

https://www.makerluis.com/content/images/2023/12/Advanced_Optimizations_Saddle_optim-1.gif
https://medium.com/@LayanSA/complete-guide-to-adam-optimization-1e5f29532c3d



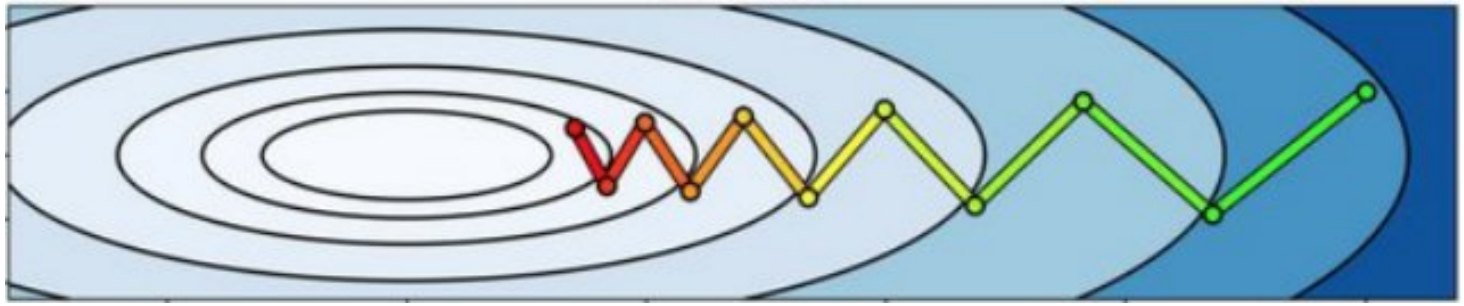



```
optimizer = torch.optim.Adadelta
optimizer = torch.optim.Adagrad
optimizer = torch.optim.Adam
optimizer = torch.optim.AdamW
optimizer = torch.optim.Adamax
optimizer = torch.optim.ASGD
optimizer = torch.optim.LBFGS
optimizer = torch.optim.NAdam
optimizer = torch.optim.RAdam
optimizer = torch.optim.RMSprop
optimizer = torch.optim.Rprop
optimizer = torch.optim.SGD
```

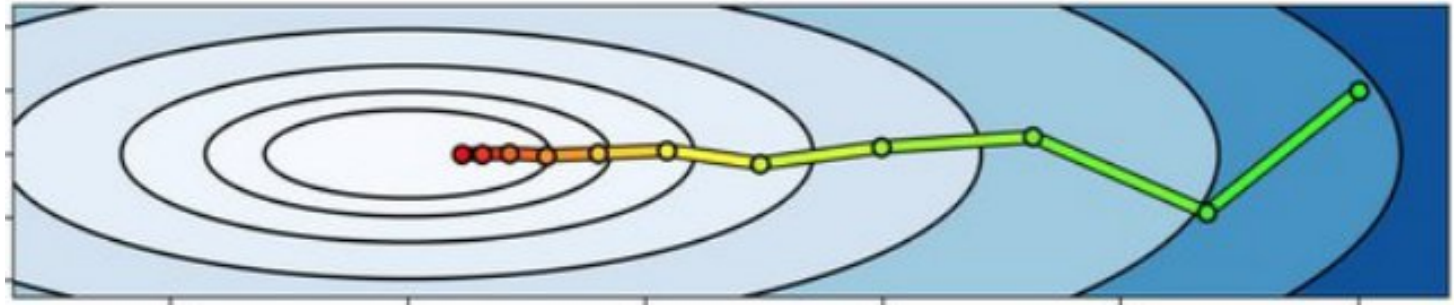
Optimizers

https://ai.stackexchange.com/questions/10013/what-is-a-bad-local-minimum-in-machine-learning
https://jermwatt.github.io/machine_learning_refined/notes/3_First_order_methods/3_8_Momentum.html

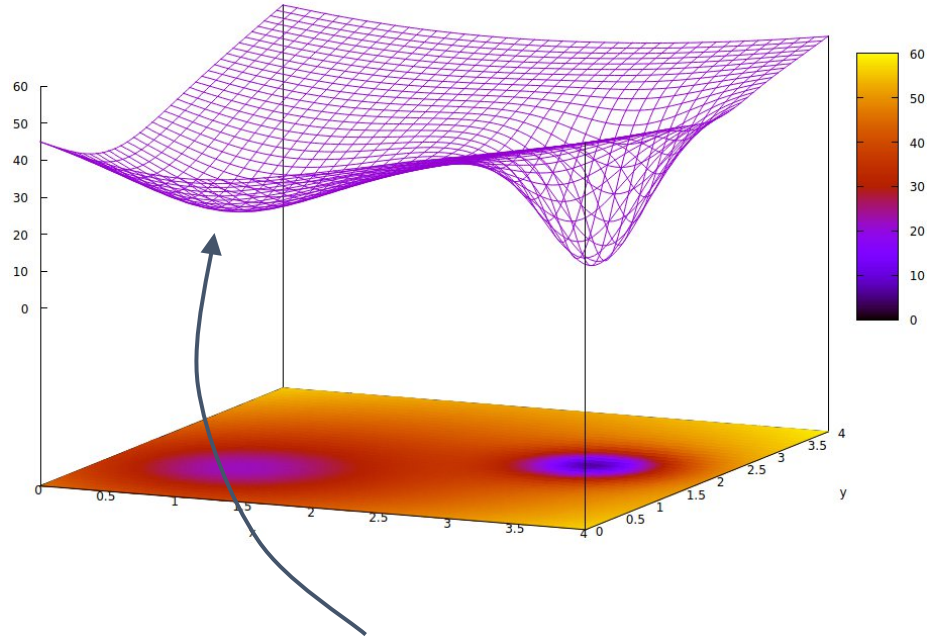
Convergence



Without momentum



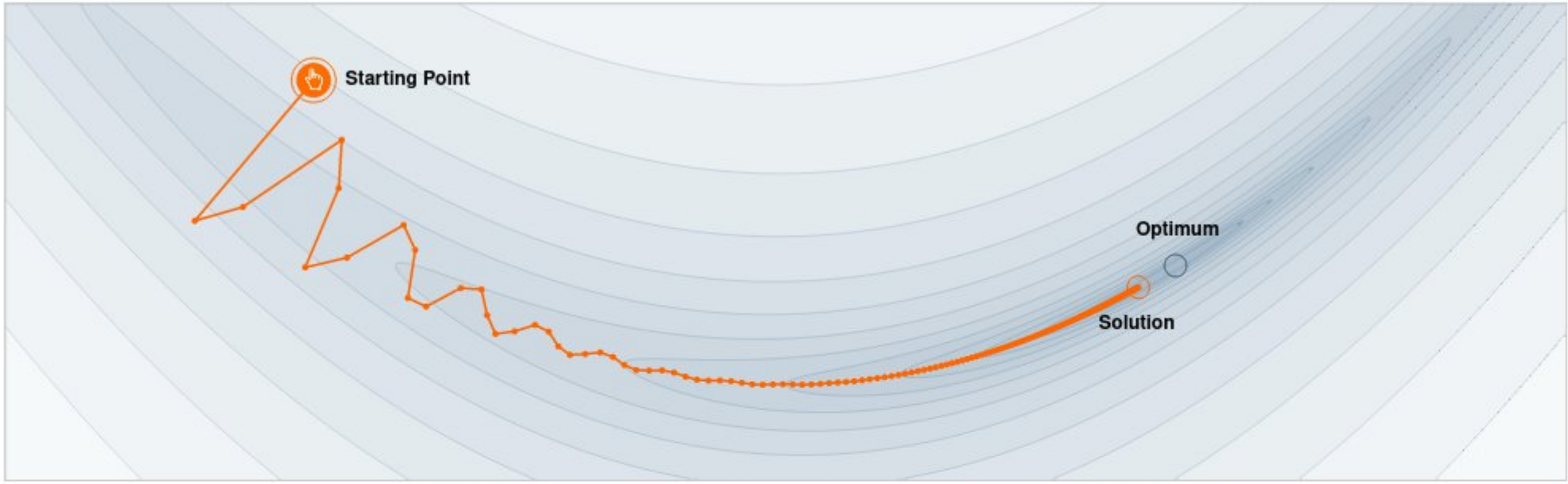
With momentum



Local minimum

Optimizers : Momentum

https://distill.pub/2017/momentum/



Step-size α = 0.02

0 0.003 0.006

Momentum β = 0.99

0.00 0.500 0.990

We often think of Momentum as a means of dampening oscillations and speeding up the iterations, leading to faster convergence. But it has other interesting behavior. It allows a larger range of step-sizes to be used, and creates its own oscillations. What is going on?

GABRIEL GOH
UC Davis

April. 4
2017

Citation:
Goh, 2017

Why momentum works ?

$$\Theta_{t+1} = \Theta_t - \eta \nabla_\Theta [\mathcal{L}(\hat{y}_i, y_i) + \lambda R(\Theta_t)]$$

Updated weights = Weights before update − Learning rate * Gradient [Cost function (Prediction , Label) + Regularization rate * Regularization function (Weights before update)]

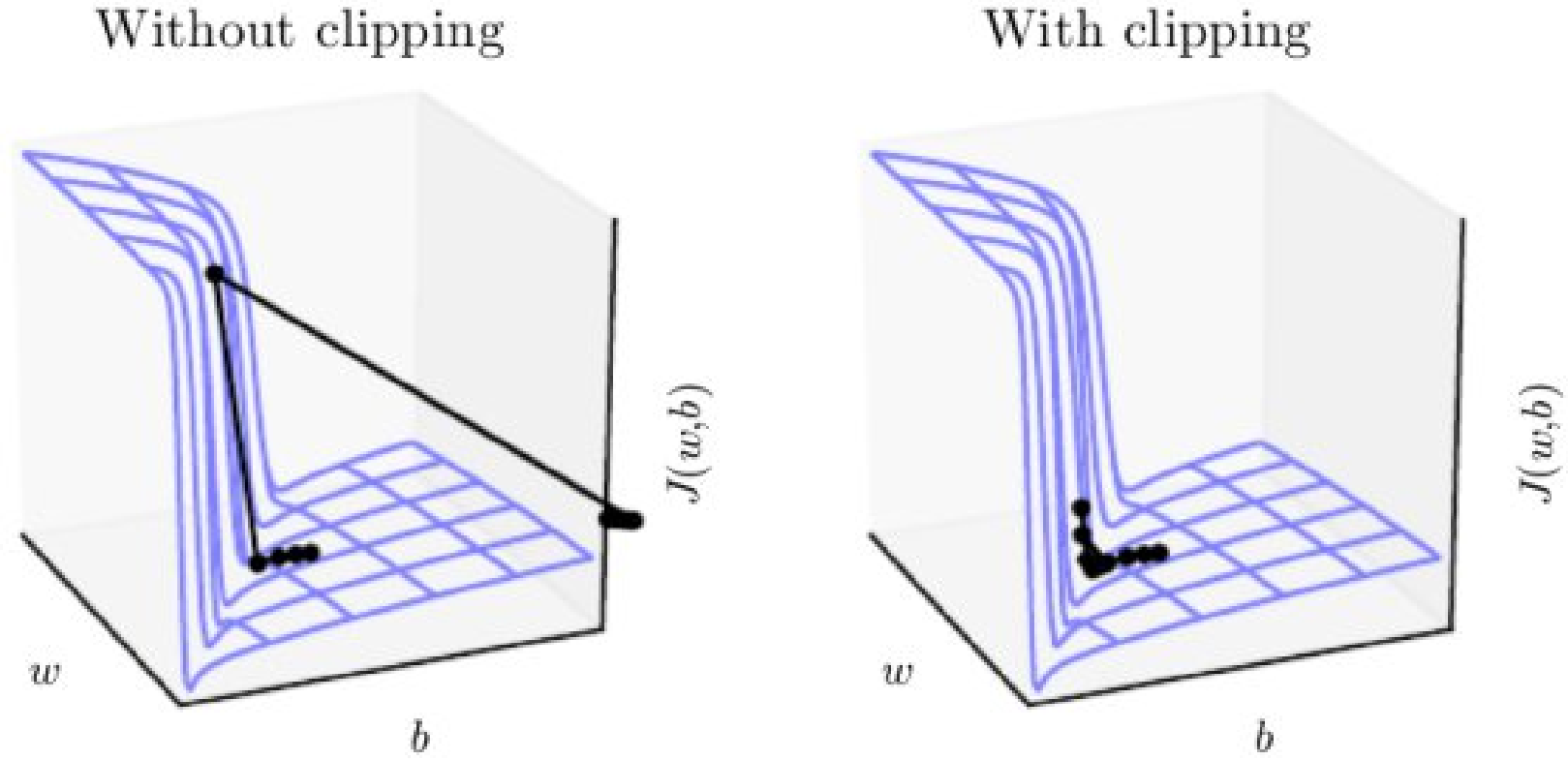
Weight update equation

- L1 Regularization
- L2 Regularization
- Max norm Regularization
- Regularization with the cost function
- Dropout

L1 : LASSO	L2 : Ridge
$\lvert\Theta\rvert$	Θ^2

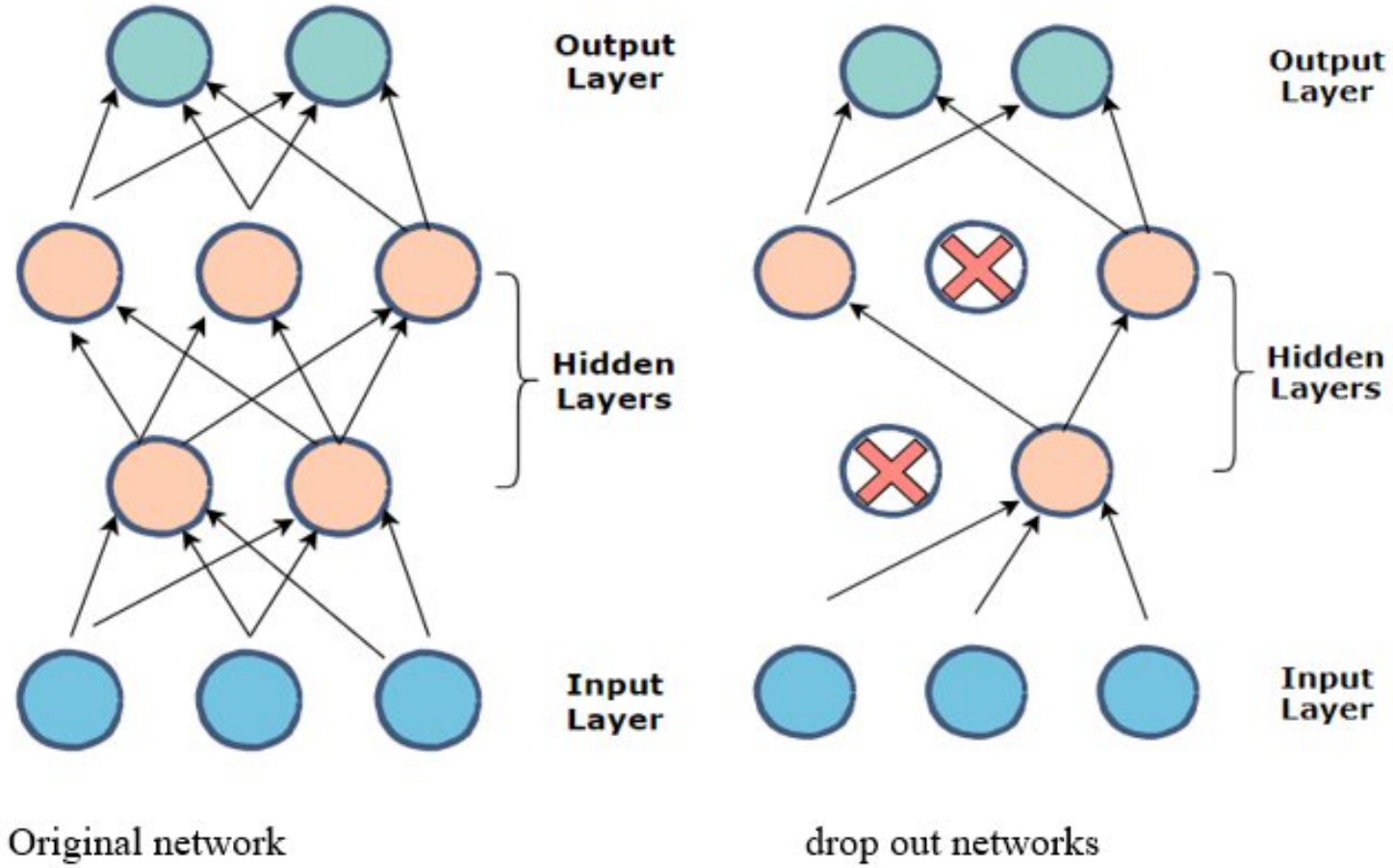
Regularization : L1 and L2

https://neptune.ai/blog/understanding-gradient-clipping-and-how-it-can-fix-exploding-gradients-problem



Regularization : Gradient clipping

https://medium.com/unpackai/introduction-of-dropout-and-ensemble-model-in-the-history-of-deep-learning-a4c2a512dcca



Original network

drop out networks

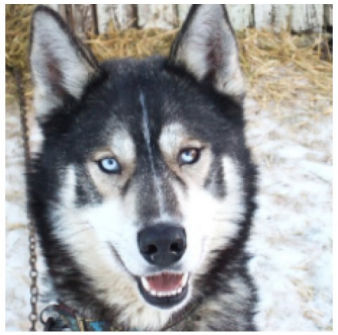
Regularization : Dropout

Source :
pytorch.org
data-flair.training
Right for the Right Reason: Making Image Classification Robust





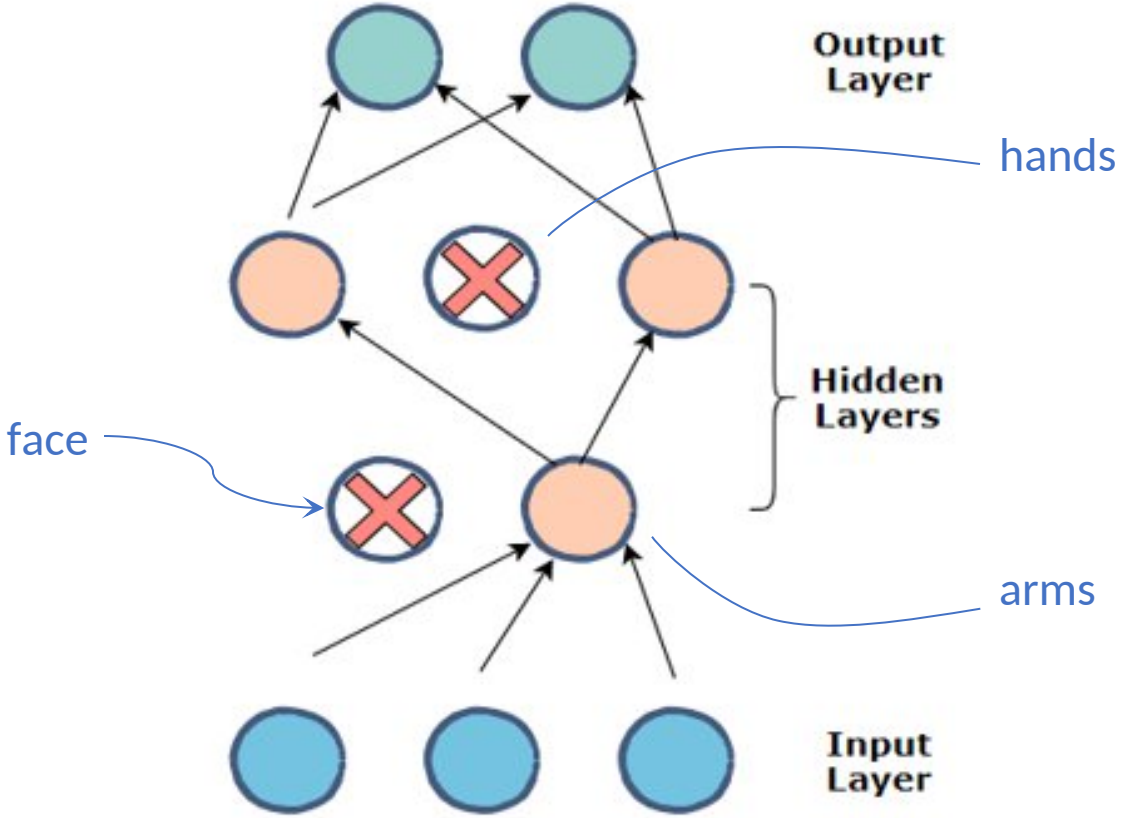
Human = face ?



(a) Husky classified as wolf



(b) Explanation



drop out networks

Regularization : Dropout

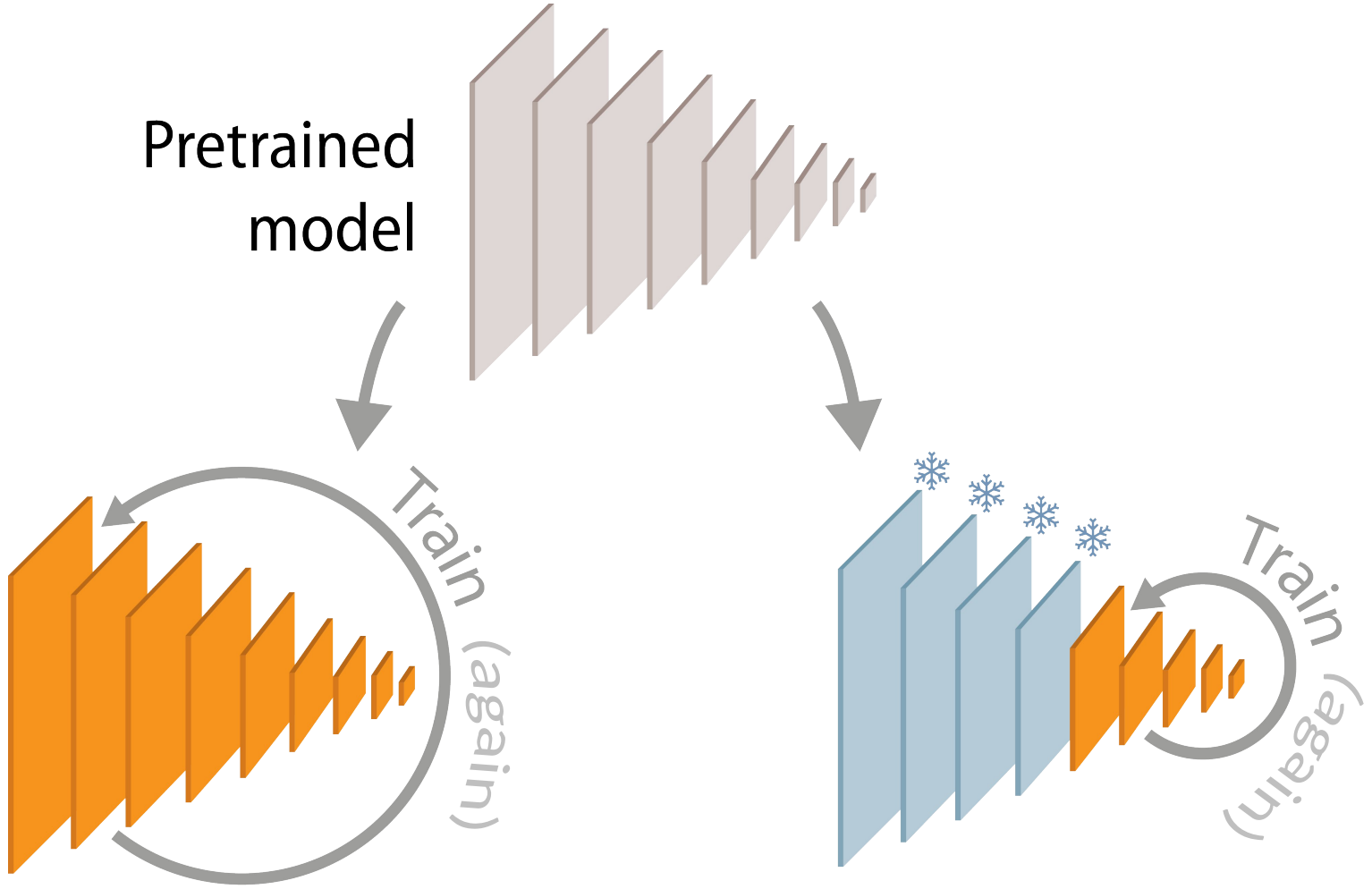
A neural network that **converges and generalizes correctly*** generally has weights **that tend to 0**.
*(neither underfitting nor overfitting)



$$\text{Loss} = L_\theta + \lambda \sum \theta^2$$

Weight Decay

1. What **work** can be done to **improve** the **data** used for training?
2. How can a model be **evaluated**?
3. Is it possible to make the training more **robust**?
4. Can we **benefit** from an **already** trained model?
5. Bonus: Good practices



Hugging Face

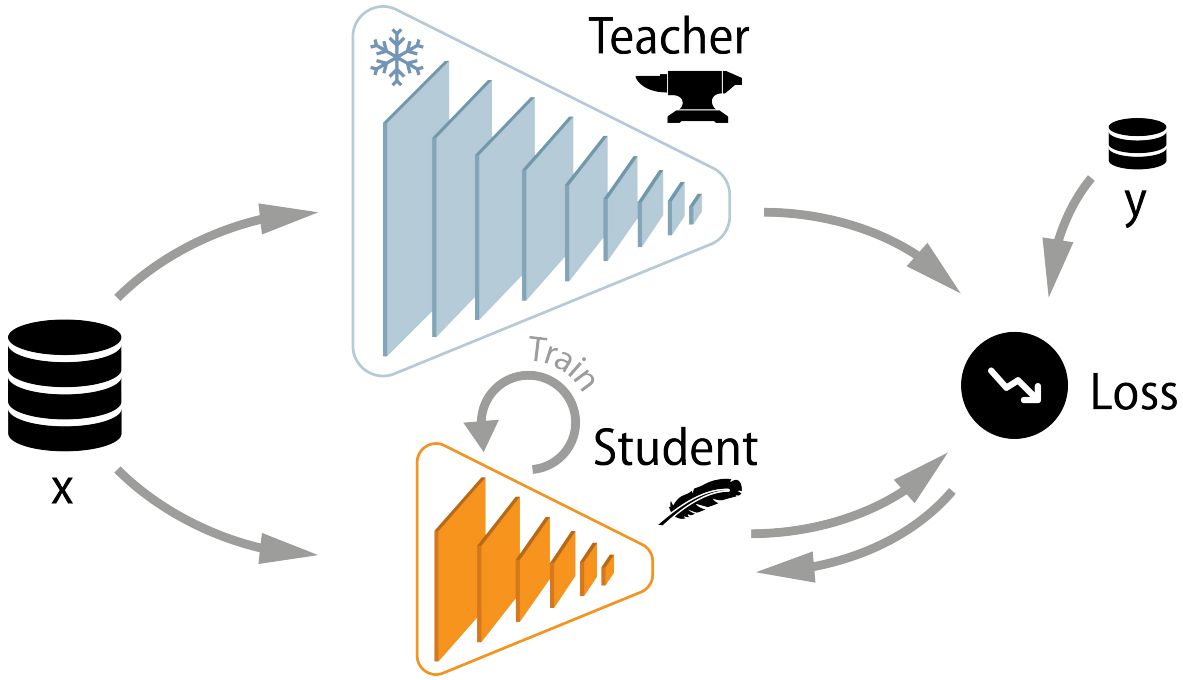
Transformers

Timm

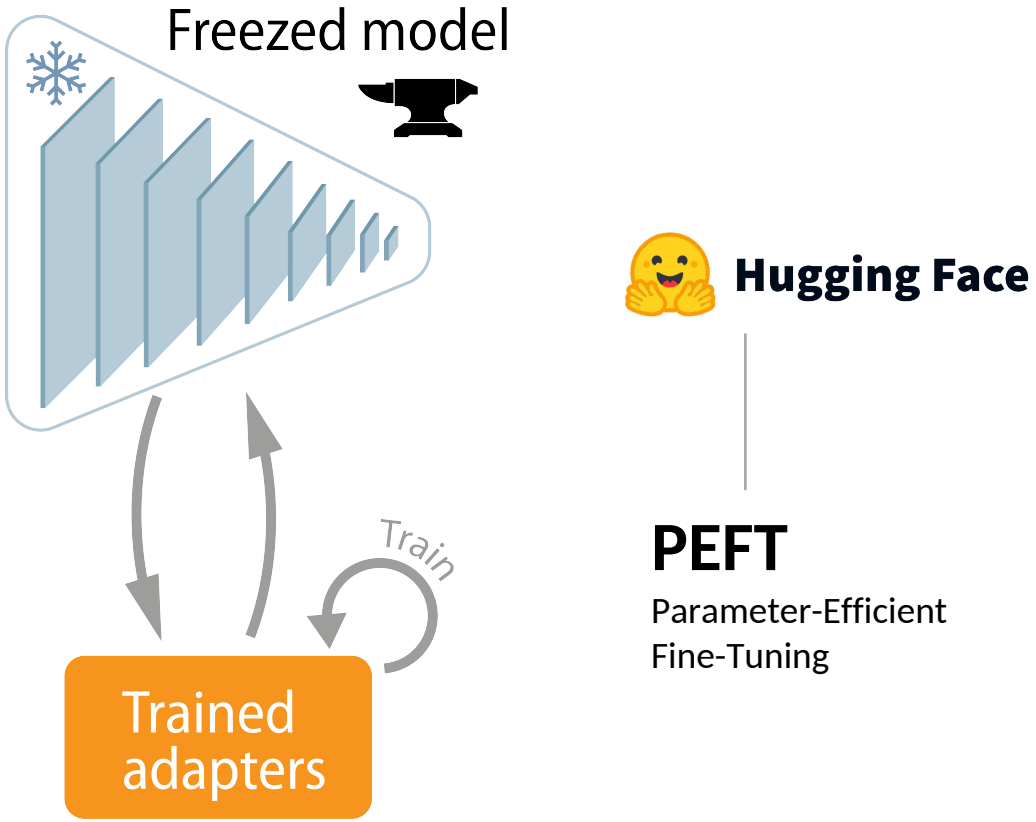
MMCV

Fine tuning & Transfer Learning

https://huggingface.co/



Knowledge distillation



Adapters

Transfer learning

1. What **work** can be done to **improve** the **data** used for training?
2. How can a model be **evaluated**?
3. Is it possible to make the training more **robust**?
4. Can we **benefit** from an **already** trained model?
5. Bonus: Good practices

Hyper-parameters

- Learning rate
- Regularization
- Optimizer
- Model architecture
- Batch size
- …

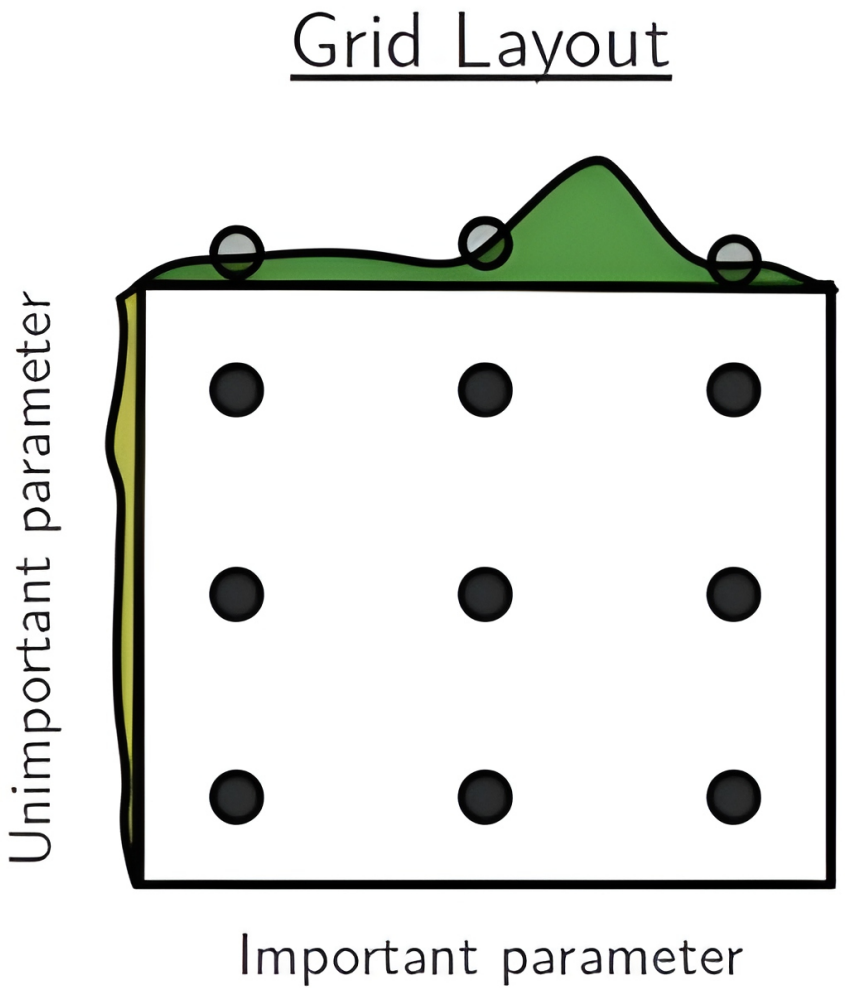
Methods

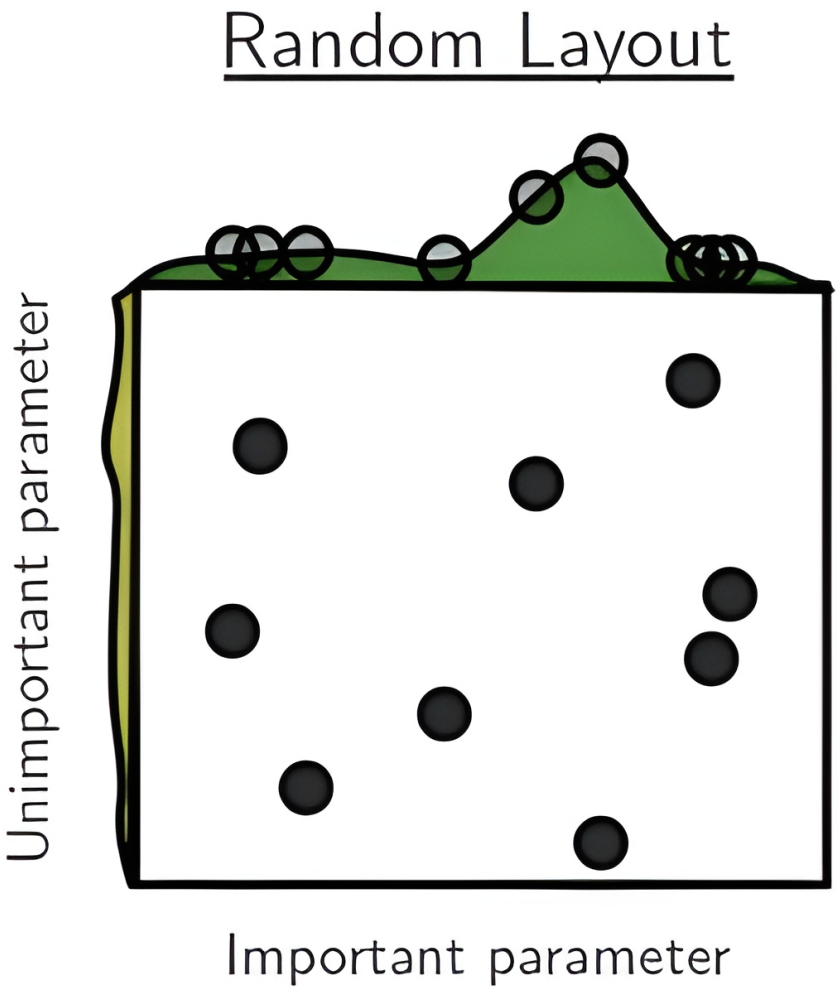
- Manual research
- Grid search
- Random research
- Gradient
- Evolutionary algorithms
- …

HPO (Hyperparameter Optimization)

Find the good hyper-parameters

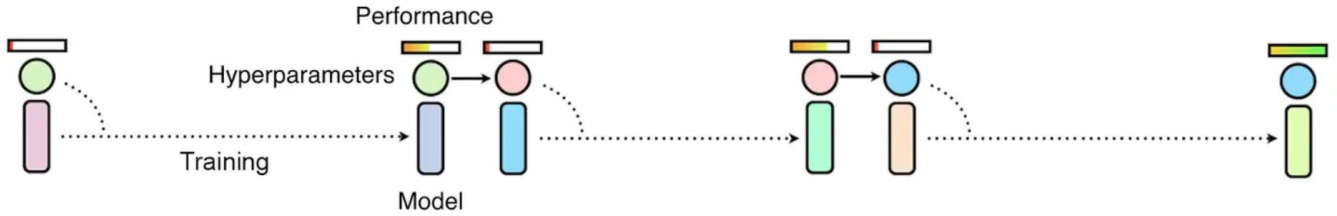
https://dl.acm.org/doi/pdf/10.5555/2188385.2188395

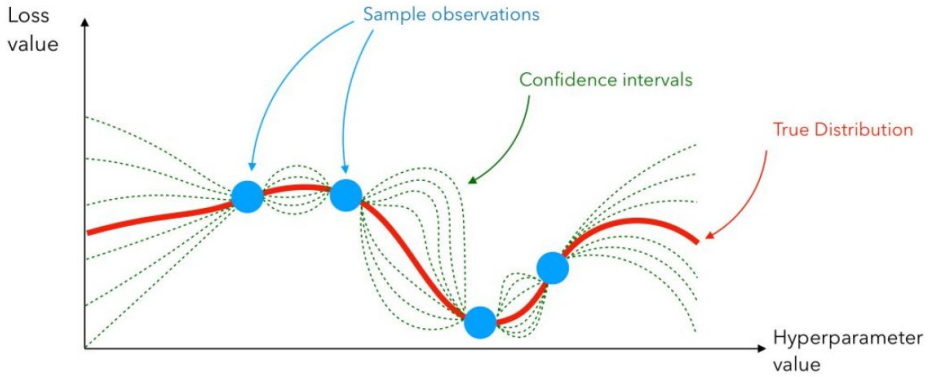


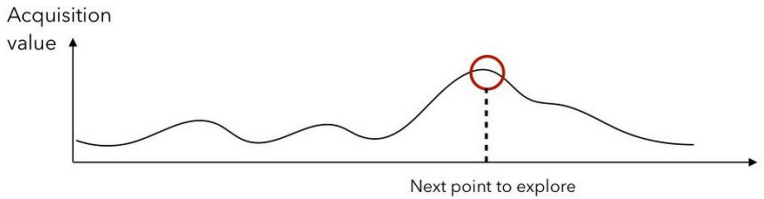


HPO : Grid & Random

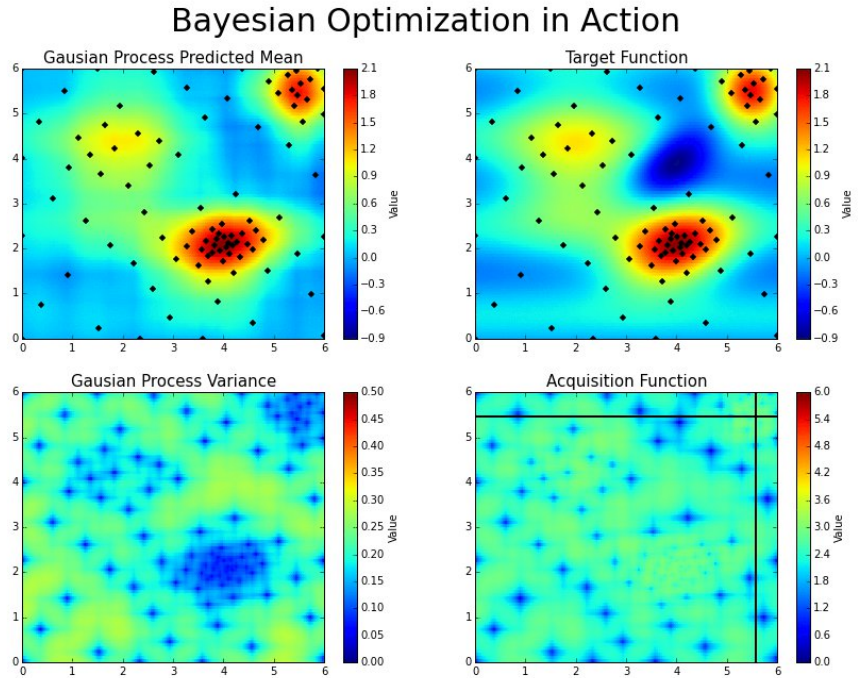
https://maelfabien.github.io/assets/images/bo.gif
https://maelfabien.github.io/machinelearning/Explorium4/#acquisition-function







- Tree Parzen Estimator / Gaussian Process
- Sequential but allows to quickly find the global optimum.
- Proposes a new set of hyper parameters based on the scores obtained by the previous ones tested.

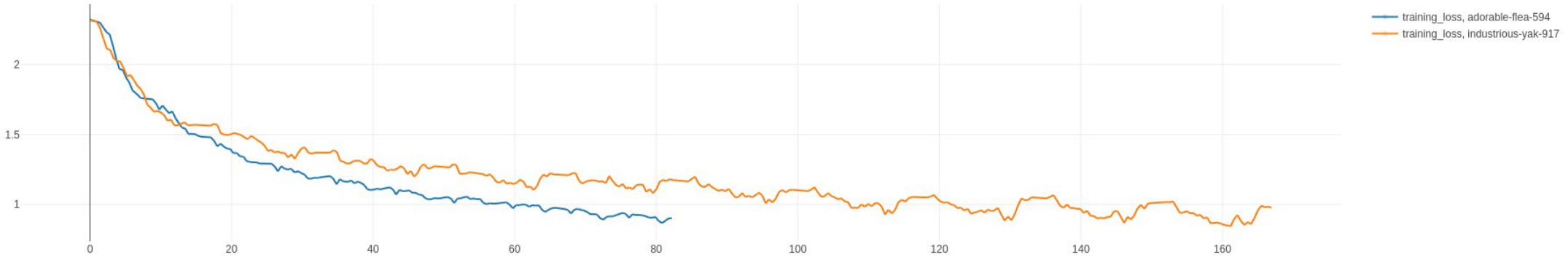


HPO : Bayesian Optimization

Table view | Chart view | metrics.rmse < 1 and params.model = "tree" | Sort: Created | Columns | Refresh

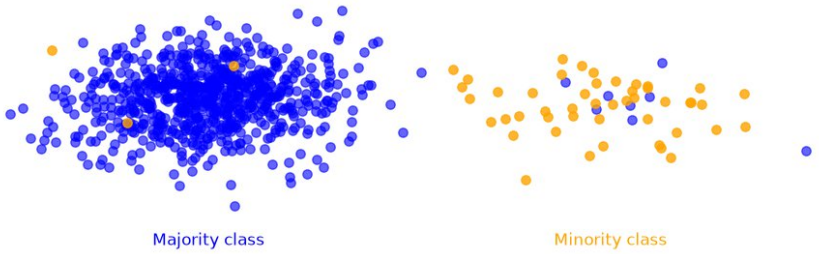
Time created: All time | State: Active

				Metrics		Parameters					
Run Name	Created	Duration	test_accuracy	training_loss	batch_size	epochs	learning_rate	momentum	weight_clampin	world_size
adorable-flea-594	1 day ago	1.5min	58.73	0.935	32	10	0.01	0.9	False	2
righteous-calf-205	1 day ago	1.5min	61.84	0.738	32	10	0.01	0.9	True	2
loud-dog-130	1 day ago	2.8min	55.87	1.303	32	10	0.01	0.9	True	1
industrious-yak-917	1 day ago	2.9min	57.57	0.915	32	10	0.01	0.9	False	1

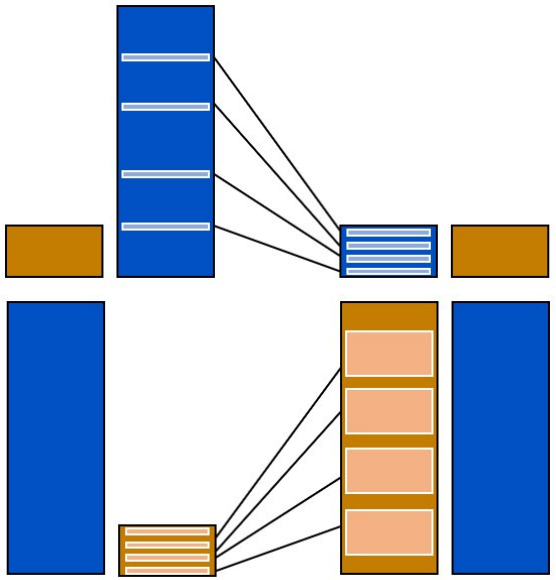


Collaboration & Reproducibility

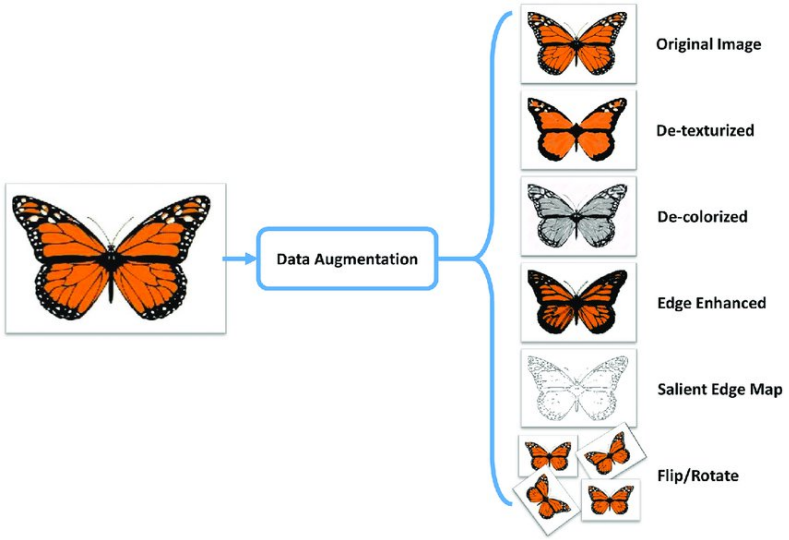
Data: Visualize them, understand them, curate them



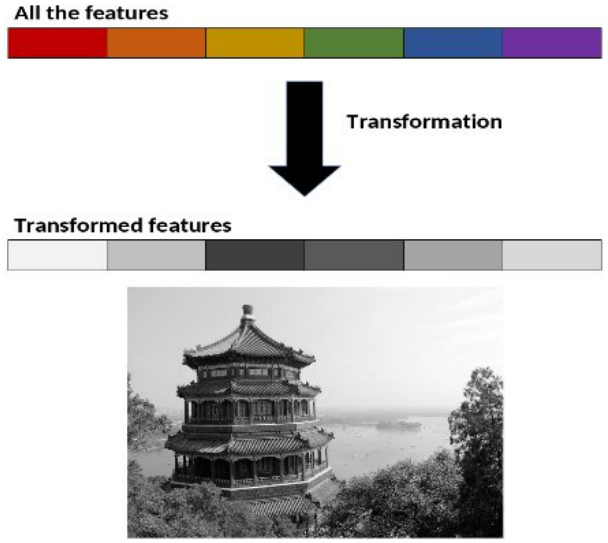
Be **aware** of irregularities (class imbalance, lack of diveristy)



Under/over-sampling



Data augmentation



Features Engineering

Use **data augmentation** techniques to solve them

Conclusion and takeaway

Metrics: they should be chosen accordingly



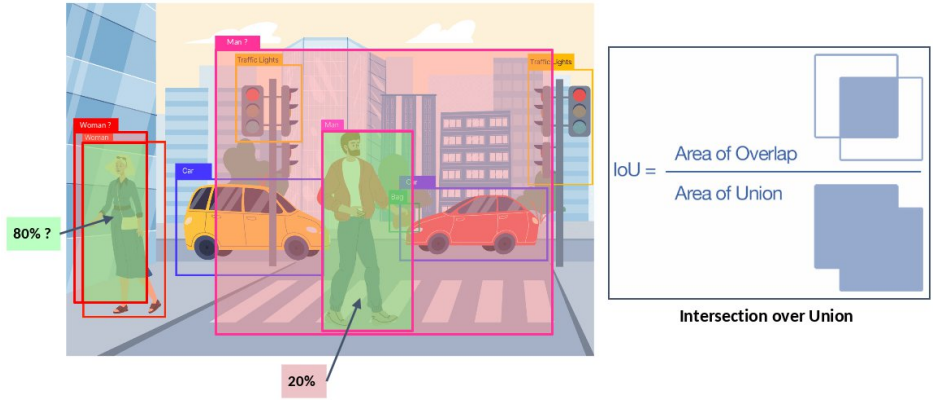
All negative:
accuracy = 99%

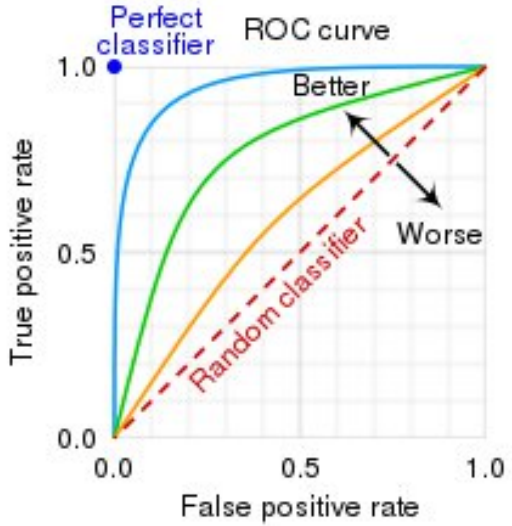
Avoid catch-all, non-representative metrics.



$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{TP + FN}$$





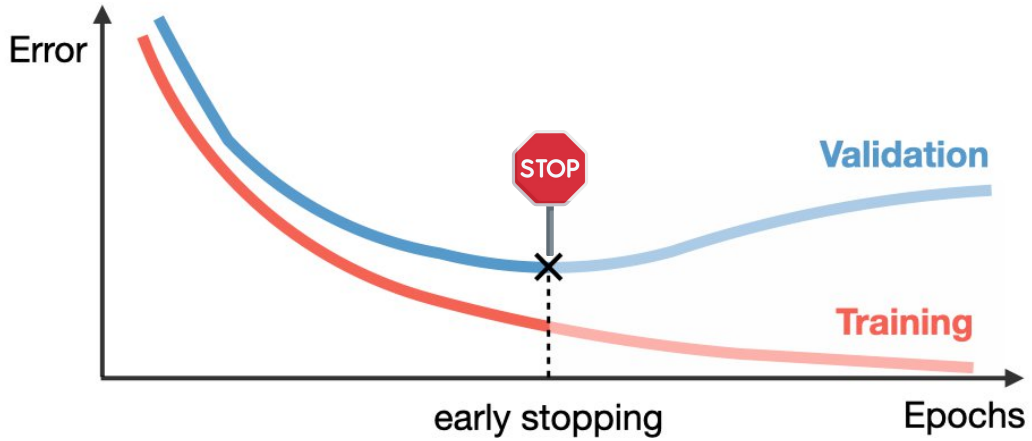
Confusion matrix **Intersection over Union** **ROC Curve**

Use **corresponding** metrics and **combinations** of metrics to identify where the model does and does not perform.

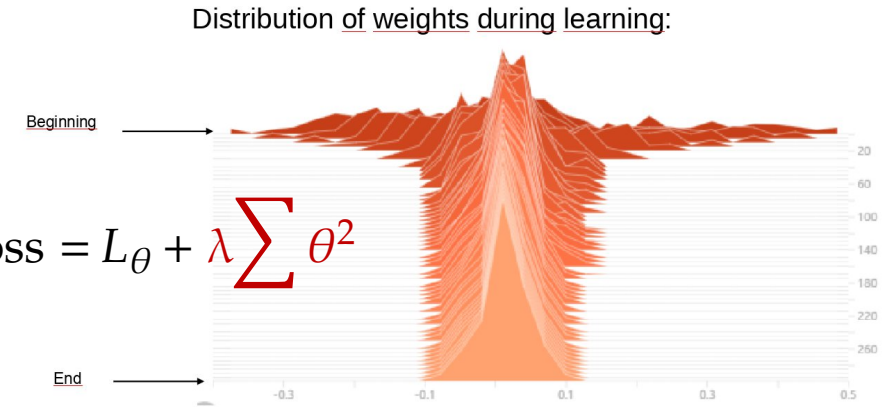
A **loss** is not a metric, and a metric is not a loss. A loss is differentiable and evaluated on a few sample for training purposes.
A metric is a non-differentiable tool for evaluation purposes.

Conclusion and takeaway

Overfitting : A model which performs well on the training set might actually not work under real-world conditions

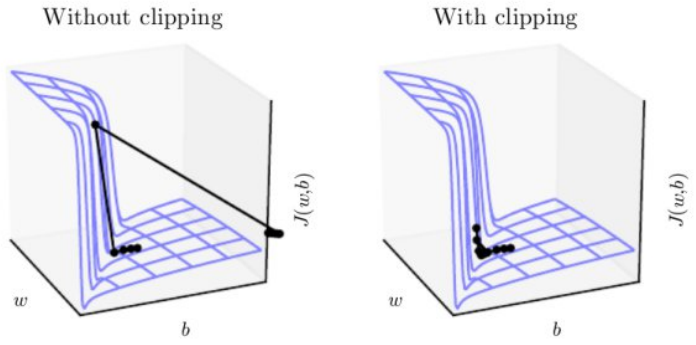


Identify when the model is starting to overfit and stop the training.

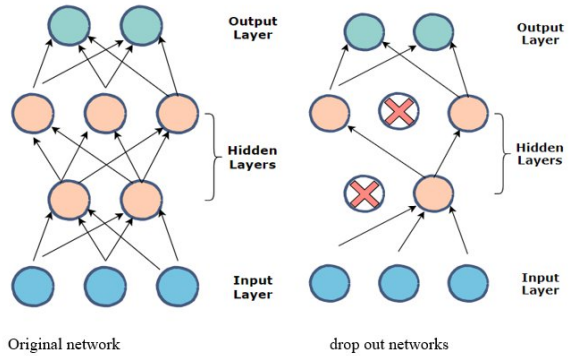


$$\text{Loss} = L_\theta + \lambda \sum \theta^2$$

Weight decay



Gradient clipping



Dropout

Use **regularization** techniques to help prevent overfitting in the first place

Conclusion and takeaway