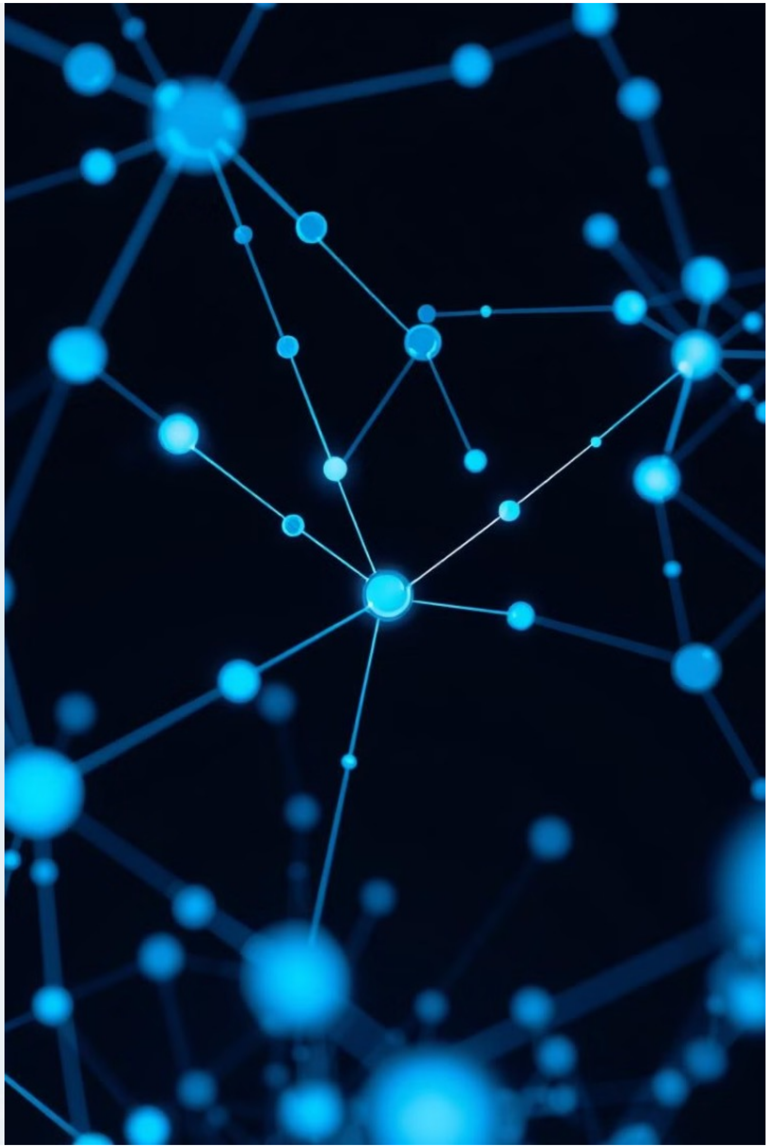
Introduction to AI

Lecture 5: Deep Learning in Practice

by Hong-Han Shuai

National Yang Ming Chiao Tung University



Syllabus

Week	Date	Contents
1	9/2	Lecture 1: Class Overview and Unsupervised Learning (HW#1)
2	9/9	Lecture 2: Traditional Classification-Part 1
3	9/16	Lecture 3: Traditional Classification-Part 2
4	9/23	Lecture 4: Neural Networks Basics (HW#2)
5	9/30	Hands-on Tutorials on PyTorch
6	10/7	Lecture 5: Deep Learning in Practice
7	10/14	Lecture 6: Introduction to Natural Language Processing
8	10/21	Midterm

slido

Please download and install the
Slido app on all computers you use

Join at slido.com
#1893184

ⓘ Start presenting to display the joining instructions on this slide.

Deep Learning Applications

An article published in Nature'19

ARTICLE

https://doi.org/10.1038/s41586-019-1119-1

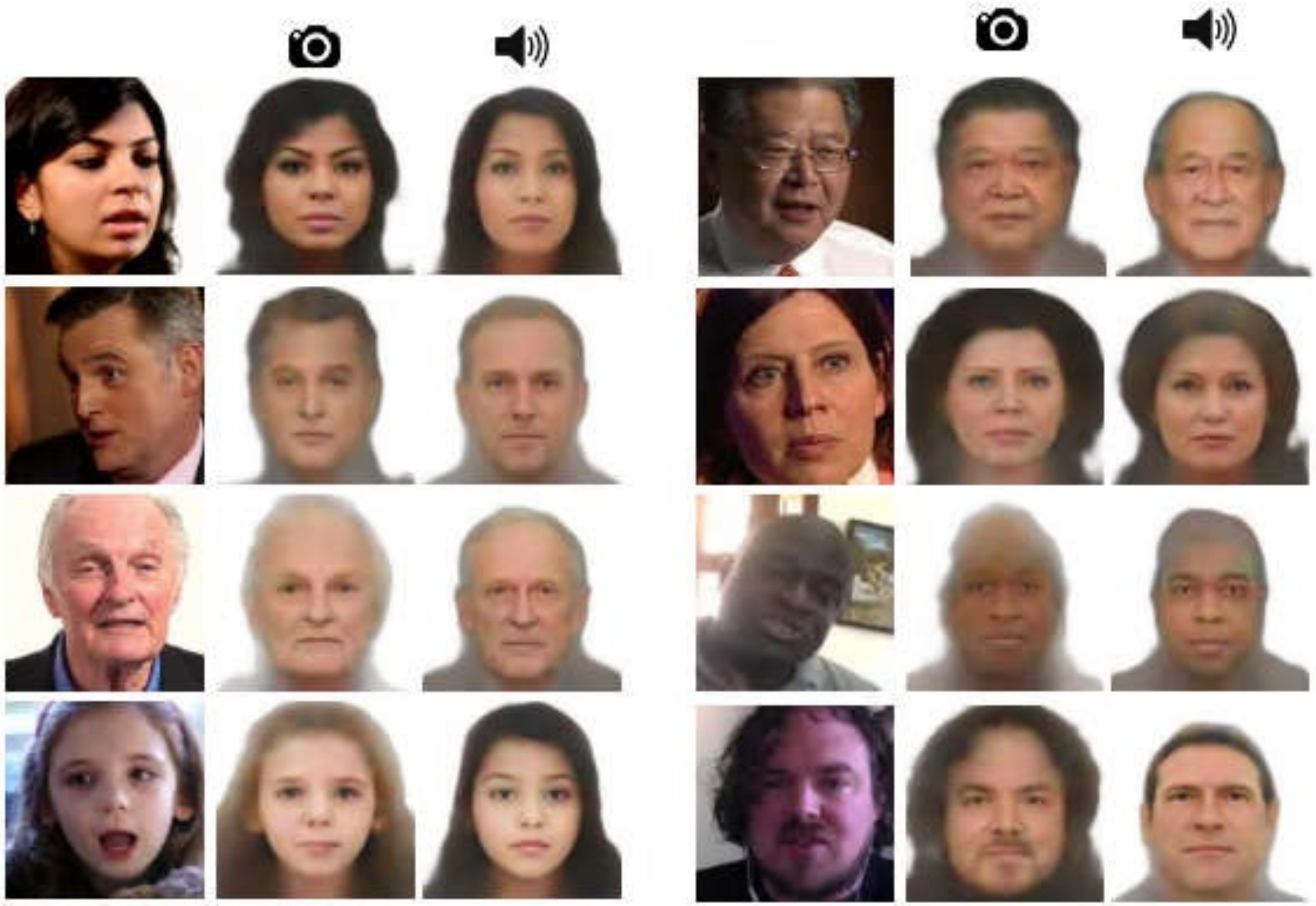
Speech synthesis from neural decoding of spoken sentences

Gopala K. Anumanchipalli[1,2,4], Josh Chartier[1,2,3,4] & Edward F. Chang[1,2,3]*

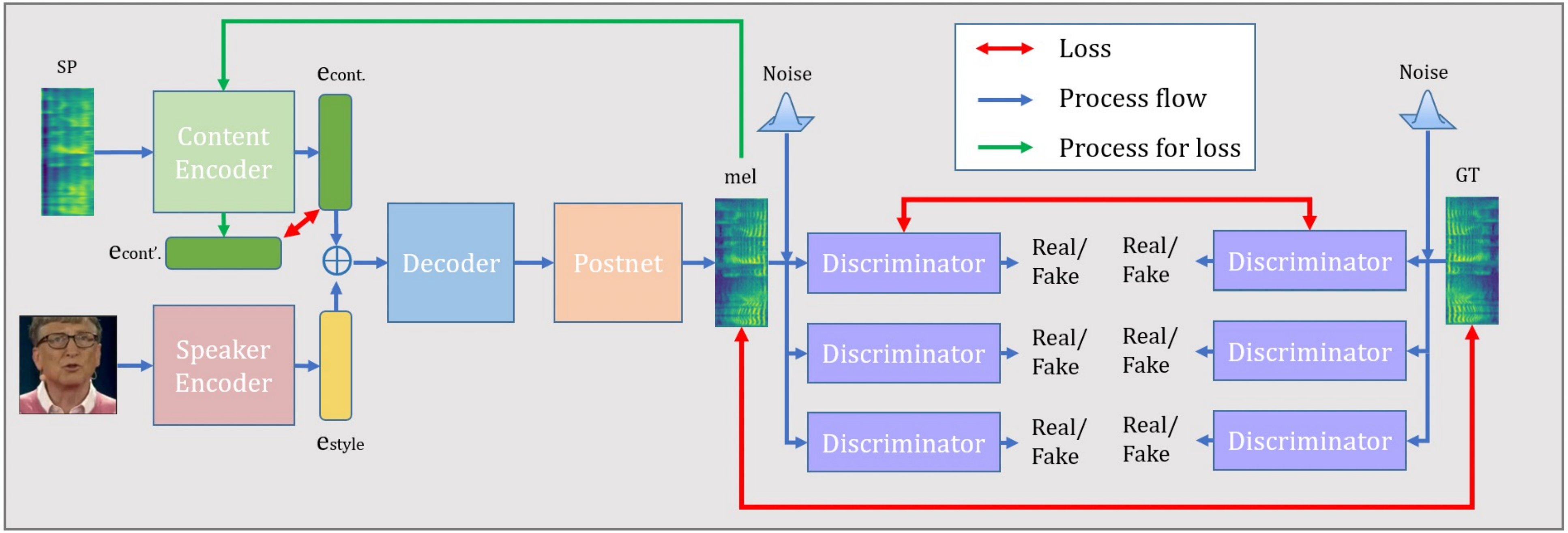
Technology that translates neural activity into speech would be transformative for people who are unable to communicate as a result of neurological impairments. Decoding speech from neural activity is challenging because speaking requires very precise and rapid multi-dimensional control of vocal tract articulators. Here we designed a neural decoder that explicitly leverages kinematic and sound representations encoded in human cortical activity to synthesize audible speech. Recurrent neural networks first decoded directly recorded cortical activity into representations of articulatory movement, and then transformed these representations into speech acoustics. In closed vocabulary tests, listeners could readily identify and transcribe speech synthesized from cortical activity. Intermediate articulatory dynamics enhanced performance even with limited data. Decoded articulatory representations were highly conserved across speakers, enabling a component of the decoder to be transferrable across participants. Furthermore, the decoder could synthesize speech when a participant silently mimed sentences. These findings advance the clinical viability of using speech neuroprosthetic technology to restore spoken communication.

你覺得當前最有潛力的深度學習應用是什麼？

Voice2Face



Zero-Shot Face-Based Voice Conversion: Bottleneck-Free Speech Disentanglement in the Real-World Scenario (AAAI 2023)



Artificial Intelligence

Cognition and Intelligence

Creation

Generative Modeling: Sample Generation

Training Data
(CelebA)

Sample Generator
(Karras et al, 2017)

(Goodfellow 2018)

Progressive Growing of GANs for Improved Quality, Stability, and Variation

Tero Karras
NVIDIA

Timo Aila
NVIDIA

Samuli Laine
NVIDIA

Jaakko Lehtinen
NVIDIA
Aalto University

3.5 Years of Progress on Faces

(Brundage et al, 2018)

(Goodfellow 2018)

<2 Years of Progress on ImageNet

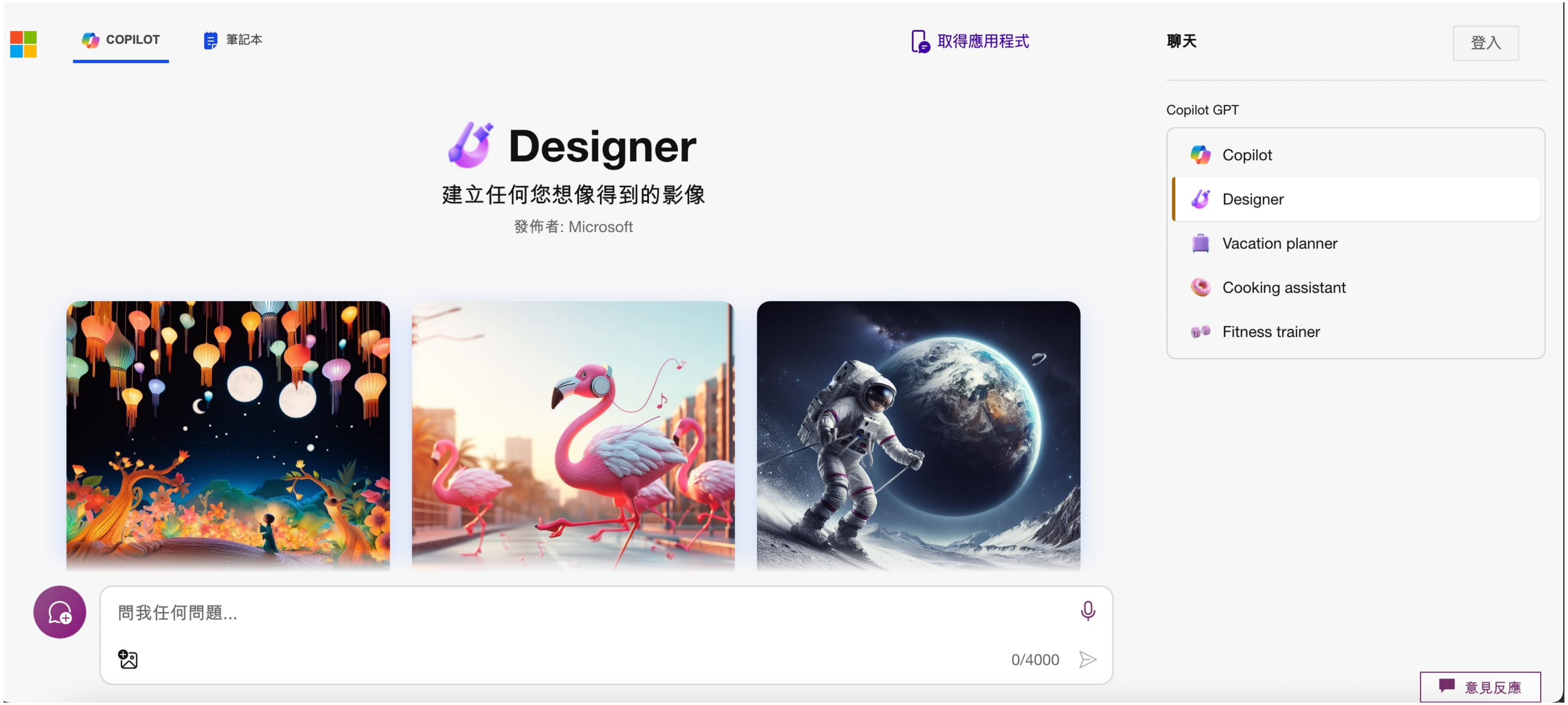
Odena et al
2016

Miyato et al
2017

Zhang et al
2018

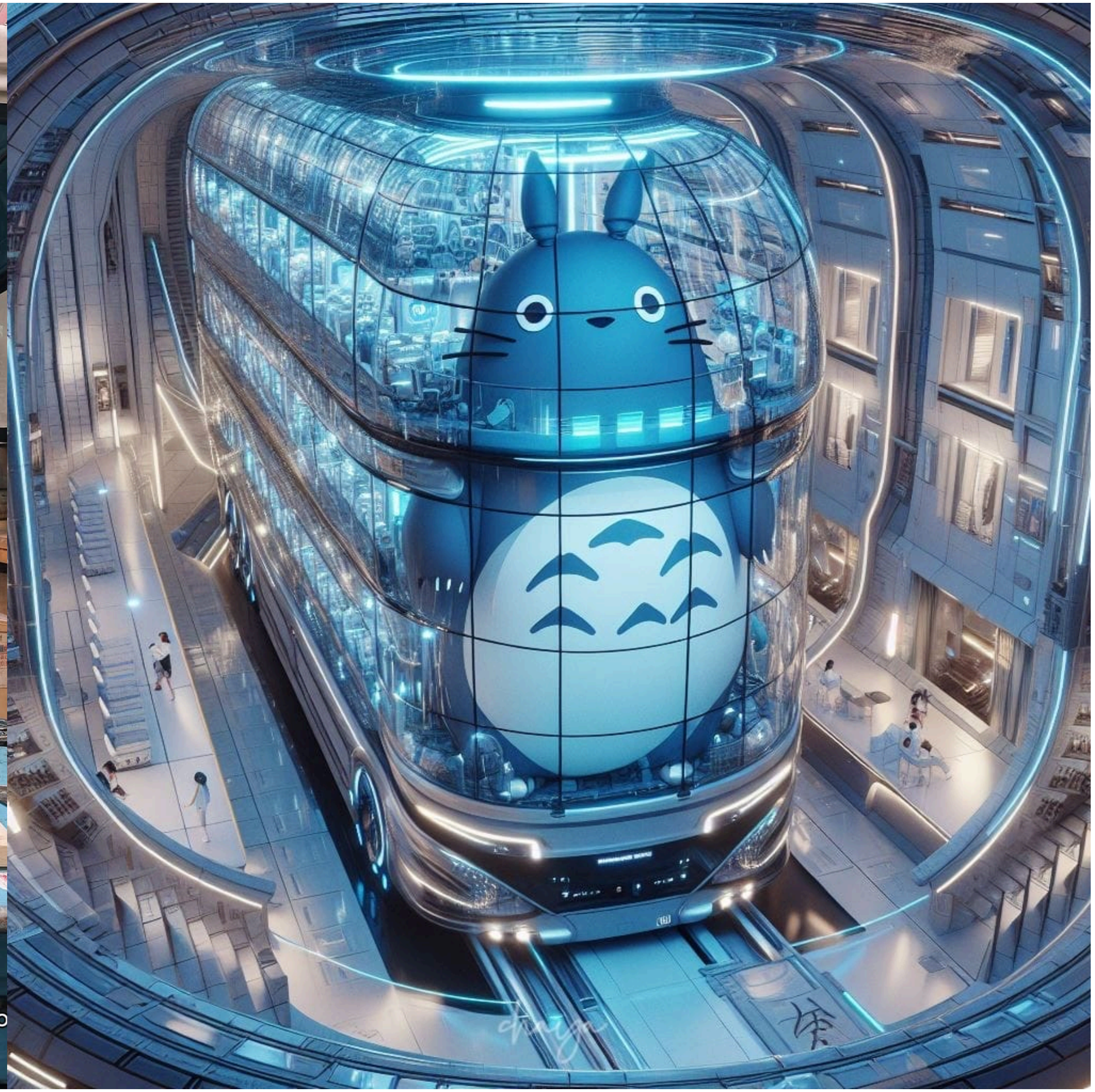
(Goodfellow 2018)

MS Copilot

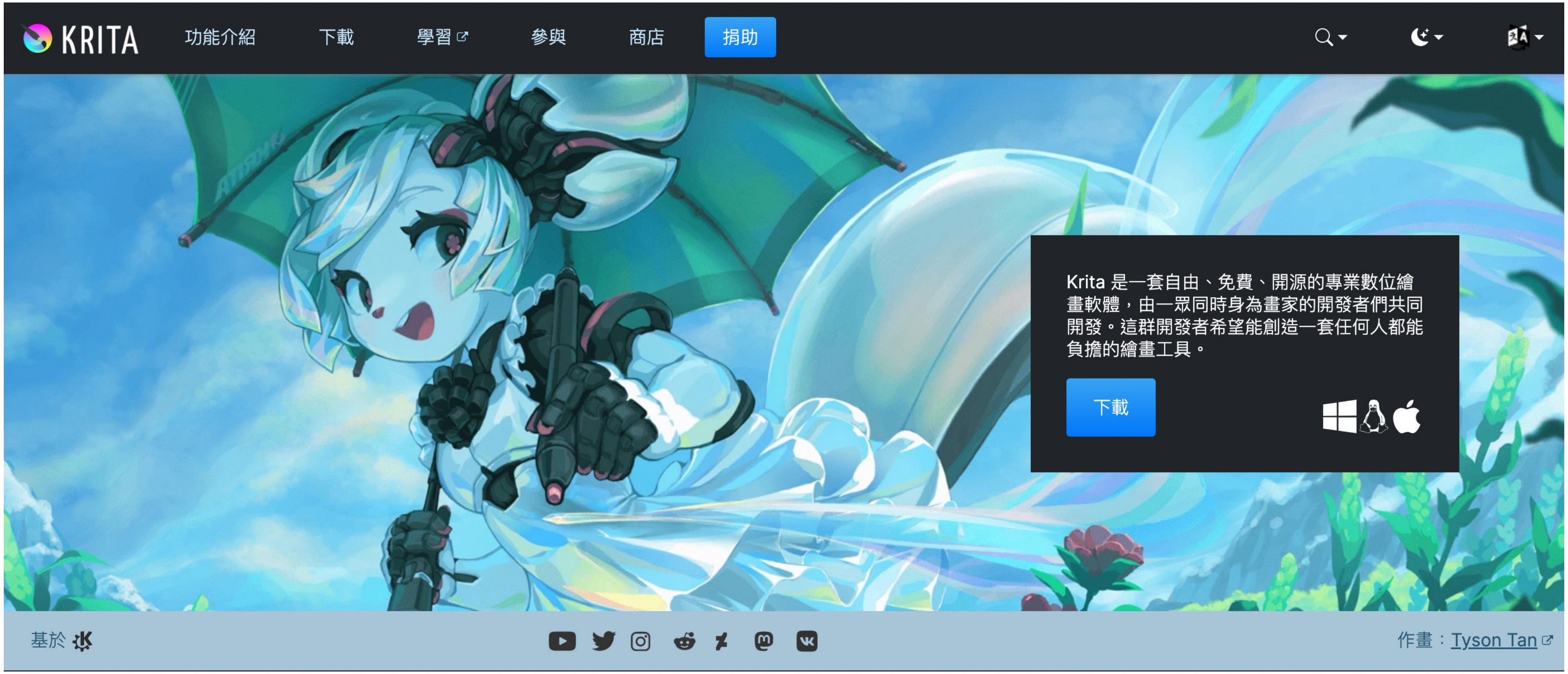


https://copilot.microsoft.com/


18 6/ 68
Pet Pixels Studio



Krita



https://krita.org/zh-tw/


PC: 林博川


TONY的老照片修復


AI
SUNO
Music
SUNO
Suno
AI 作曲大師
https://suno.com/

https://ai.meta.com/research/movie-gen/

Prompt: A medium shot, historical adventure setting: Warm lamplight illuminates a cartographer in a cluttered study, poring over an ancient, sprawling map spread across a large table. Cartographer: "According to this old sea chart, the lost island isn't myth! We must prepare an expedition immediately!"-

中景鏡頭，歷史冒險場景：
溫暖的燈光照亮了一位地圖製作者，他坐在雜亂的書房中，專注地研究一張鋪滿整張大桌子的古老海圖。
地圖製作者說：「根據這張古老的航海圖，那座失落的島嶼並不是神話！我們必須立刻準備探險行動！」

https://deepmind.google/models/veo/

Gemini **Storybook**

Gemini | Free for Students | What Gemini Can Do | Subscriptions | About Gemini | Try Gemini

For the stories only you could imagine

Turn cherished memories, inside jokes, and complex concepts into a story you can read, listen to, print, and share.

https://gemini.google/overview/storybook/

Tech 科技新聞 A.I. A.I. 生成

Grok 聊天機械人 Ani 日本爆紅　洗腦秘技改變女友性格

Mi

愛模擬飛行
上的宅，眼
醇跟美味是
以騙，但絕
貨才是錢，
4.踩單車，是
的話語，不

LOVERSE
利用規約
プライバシー
よくある質問
ログイン
「もう恋しない」なんて
つまんなすぎる
ONLY 1
マッチングアプリ※
あの恋を超える恋
今すぐ、誰でも。
お相手はAI。そのままのあなたで。
登録なしで見てみる
登録ずみの方はこちら

Virtual Try-On

Motivation | In real-world…

https://www.youtube.com/watch?v=eOiiMB4mmYU

Motivation | But researches…

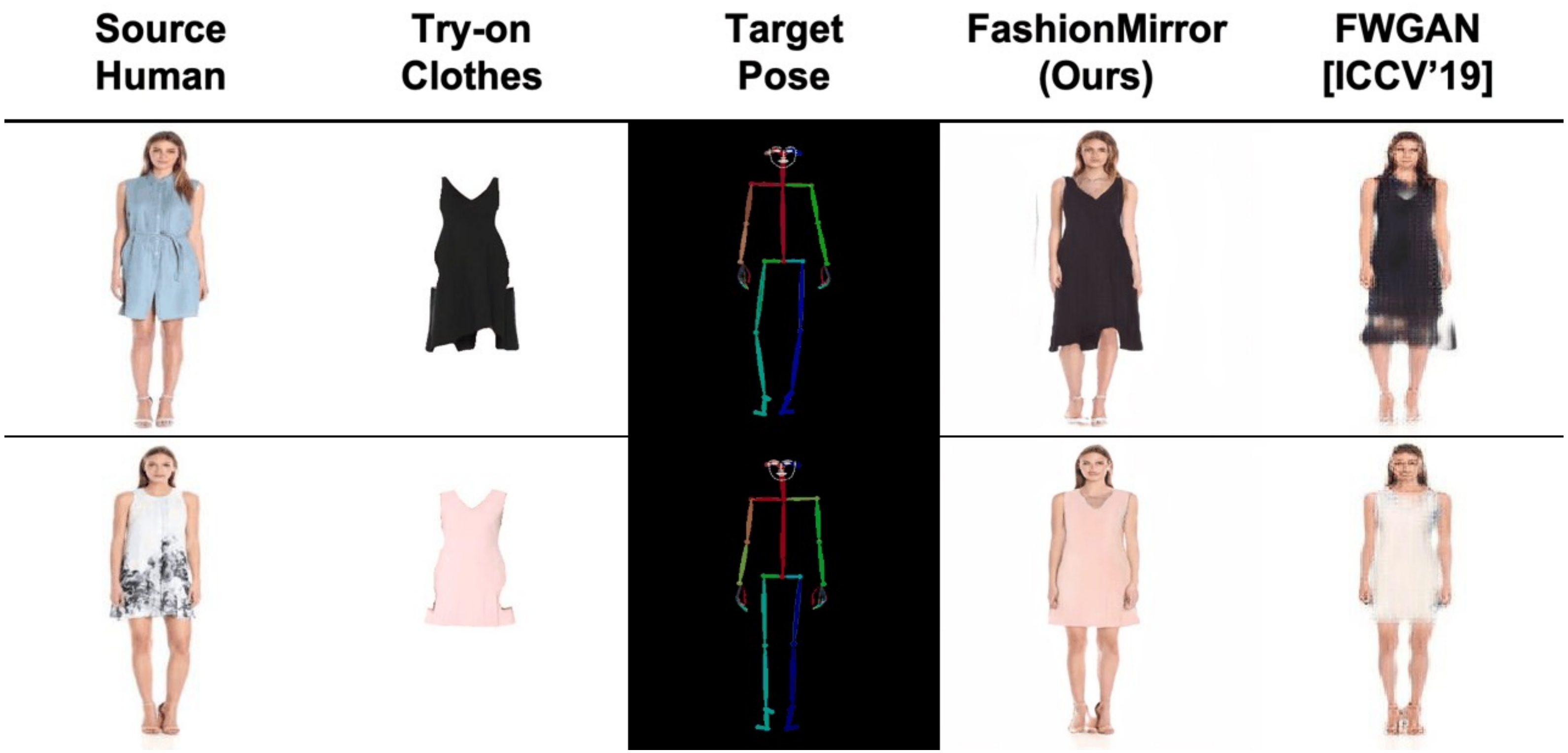
Wang et al., "Toward characteristic-preserving image-based virtual try-on network," in *ECCV*, 2018

Yang et al., "Towards photorealistic virtual try-on by adaptively generating ↔ preserving image content," in *CVPR,* 2020.

Ge et al., "Parser-Free Virtual Try-on via Distilling Appearance Flows," in *CVPR*, 2021

Pose Transformation ?

Virtual Try-on with Sequential Template Poses



Virtual Try-On

ɔure 1. Our method outperforms the existing SOTAs in generating low-artifacts images while preserving garment details.

Makeup Transfer

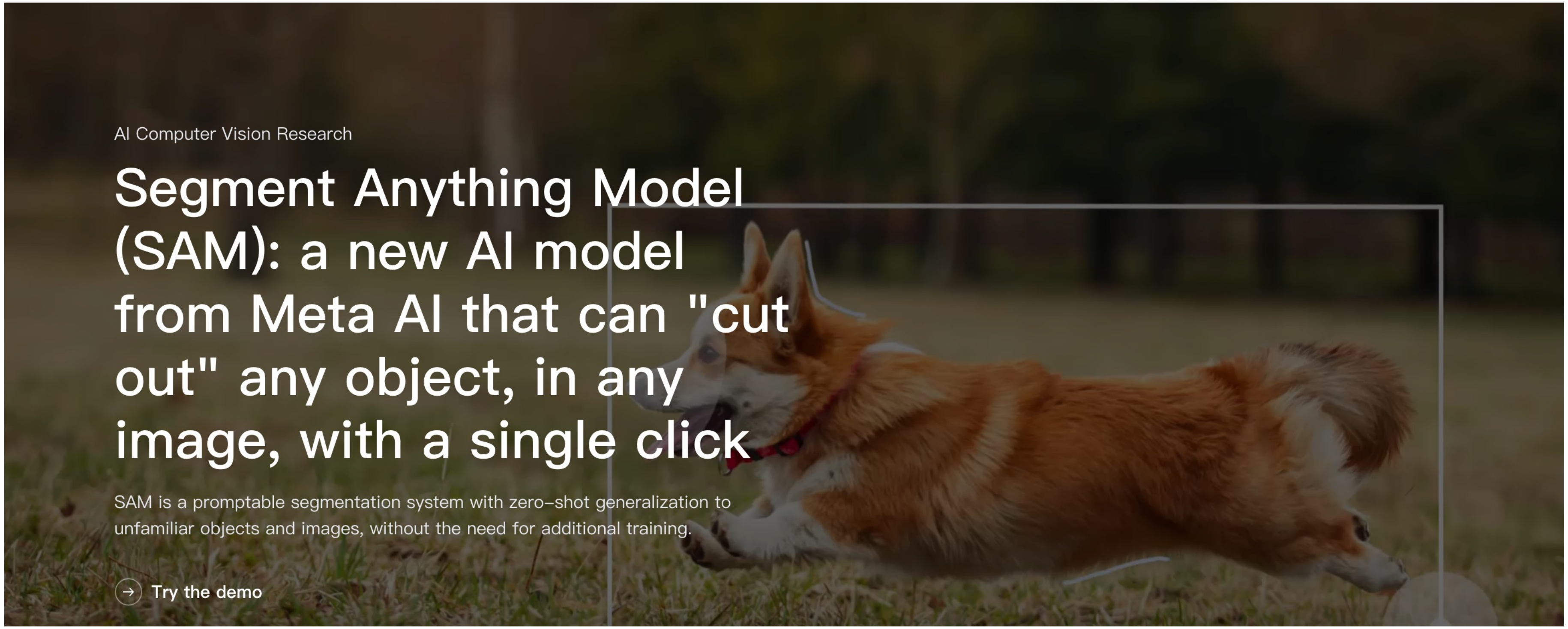
Fig. 1: Our proposed approach achieves a range of applications, including (a) beauty filter, (b) makeup removal, (c) text modification, (d) makeup transfer, (e) scale transfer, (f) component transfer, and (e) multi-makeup transfer.

MULTI-MODAL LLM

[redacted]

Segment Anything

Research by Meta AI



Prompt it with interactive points and boxes.

Automatically segment everything in an image.

Generate multiple valid masks for ambiguous prompts.

SAM can take input prompts from other systems, such as in the future taking a user's gaze from an AR/VR headset to select an object.

Bounding box prompts from an object detector can enable text-to-object segmentation.

Do not edit
How to change the design

Any new ideas?

The Slido app must be installed on every computer you're presenting from

slido

TRAINING ISSUES

USEFUL TOOLS

- **Gradient Descent:**
  - Batch gradient descent
  - Stochastic gradient descent
  - Mini-batch gradient descent
- **Adaptive:**
  - Adagrad
  - Adadelta
  - RMSprop
  - Adam
- **Scheduler**
  - CosineAnnealing
  - ExponentialLR

ADAGRAD — ADAPTIVE GRADIENT ALGORITHM

- Adapts learning rate individually for each parameter
- Accumulates squared gradients to scale updates

$$g_t = \nabla_\theta L_t, \quad G_t = G_{t-1} + g_t^2$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} g_t$$

✓ Works well for sparse features (e.g., NLP)

✗ Learning rate decreases monotonically (can become too small)

ADADELTA — REDUCING ADAGRAD'S DECAY PROBLEM

- Solves Adagrad's 'infinitely decreasing learning rate' issue
- Uses exponentially decaying average of squared gradients

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1-\rho) g_t^2$$

$$\Delta\theta_t = -\frac{\sqrt{E[\Delta\theta^2]_{t-1} + \epsilon}}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

$$E[\Delta\theta^2]_t = \rho E[\Delta\theta^2]_{t-1} + (1-\rho)(\Delta\theta_t)^2$$

✓ No explicit learning rate needed, robust hyperparameter behavior

RMSPROP — EFFICIENT ONLINE ADAPTIVE LEARNING

- Maintains exponential moving average of squared gradients

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

- Typical values: γ=0.9, η=0.001

✓ Stabilizes training for RNNs or non-stationary tasks

✓ Prevents learning rate collapse

ADAM — ADAPTIVE MOMENT ESTIMATION

- Combines Momentum (1st moment) and RMSprop (2nd moment)

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

Bias correction:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Parameter update:

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

✓ Default optimizer, fast convergence, handles sparse gradients

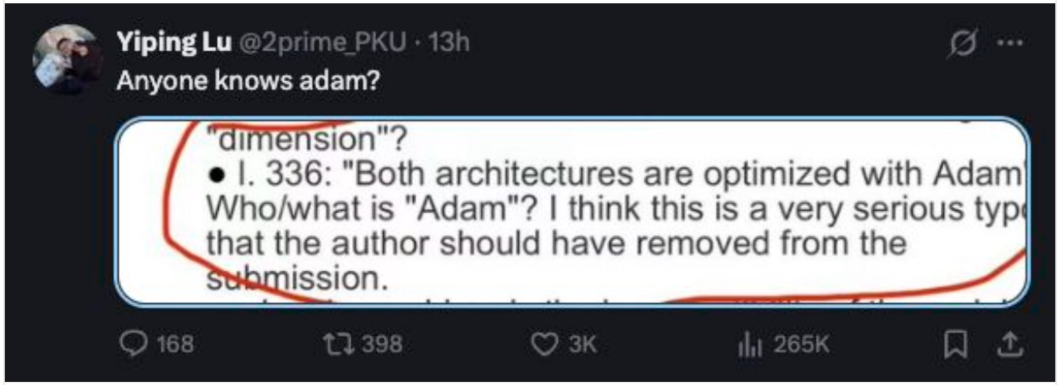
Yexi Jiang
Learner & Problem Solver | Visit yexijiang.substack.com/
3 週

Who is Adam?!

NeurIPS 2025 review season has officially gone off the rails.
“Both architectures are optimized with Adam.
Who/what is ‘Adam’? I think this is a very serious typo…”

This real review comment has now gone viral.
A NeurIPS reviewer genuinely asked who Adam is (the optimizer, not a co-author).
It was flagged as a very serious typo.

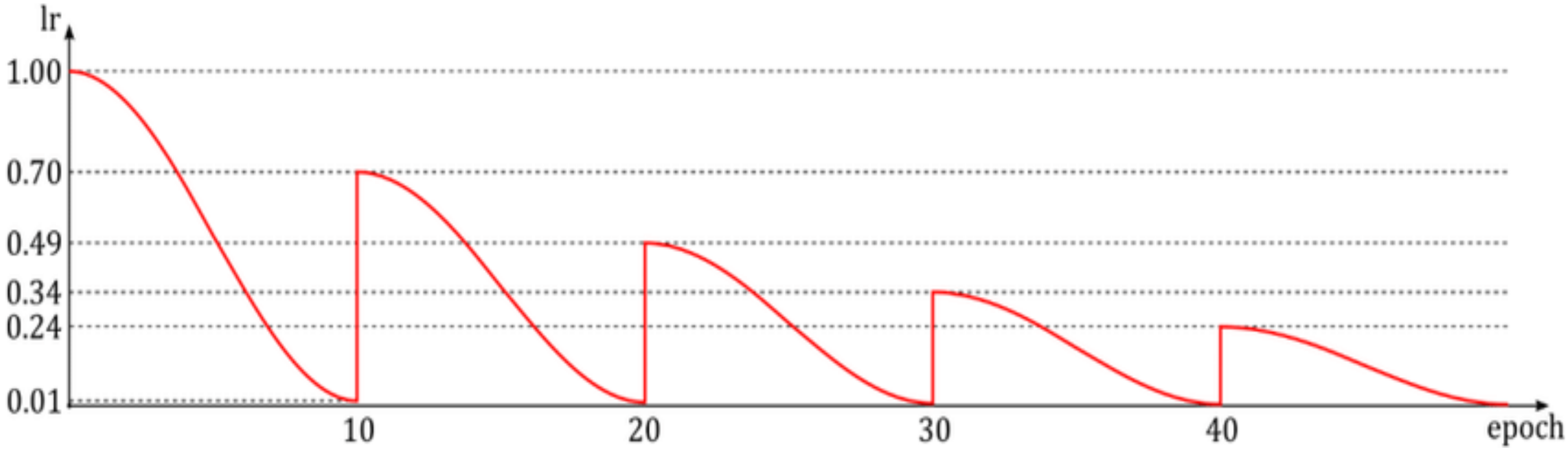
Amusing about this review and concern about the quality of the paper review process.



3,415 · 153 則回應

https://www.linkedin.com/posts/yxjiang_who-is-adam-neurips-2025-review-season-activity-7354524930631233536-De4l

COSINE ANNEALING



IMPLEMENTATION DETAILS

```
import torch model = [Parameter(torch.randn(2, 2, requires_grad=True))]
optimizer = torch.optim.AdamW(model.parameters(), lr=learning_rate, weight_decay=0.01, amsgrad=False)
scheduler = torch.optim.lr_scheduler.CosineAnnealingLR(optimizer, T_max=1000, eta_min=0, last_epoch=-1, verbose=False)
for epoch in range(20):
   for input, target in dataset:
      optimizer.zero_grad()
      output = model(input)
      loss = loss_fn(output, target)
      loss.backward()
      optimizer.step()
   scheduler.step()
```

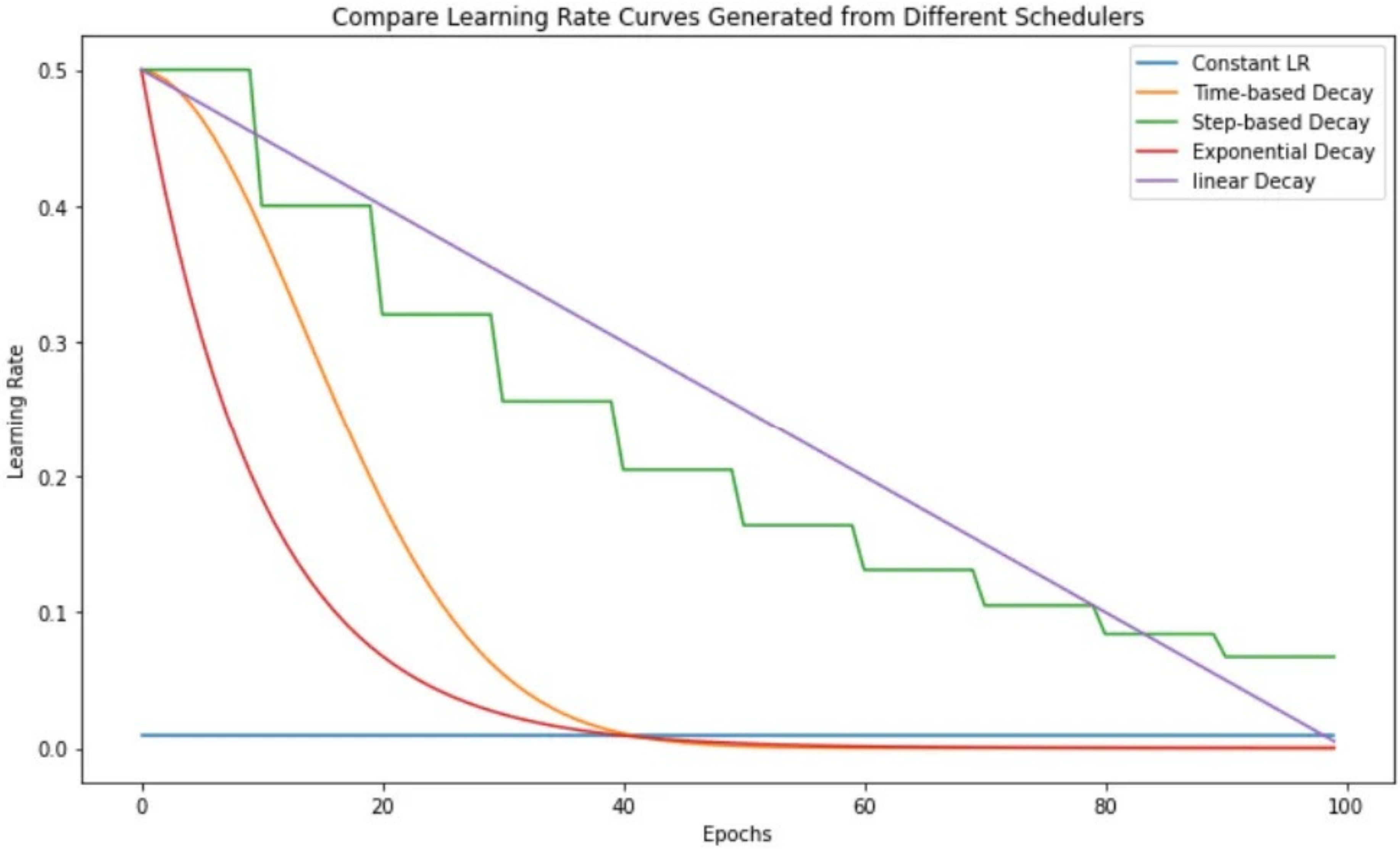
場景	T_max	eta_min	備註
小型 dataset	50–100	0	一次完整下降週期即可
大型 dataset	500–1000	1e-6	平滑收斂、穩定訓練
Fine-tuning	10–30	1e-5	適合短訓練階段

T_max

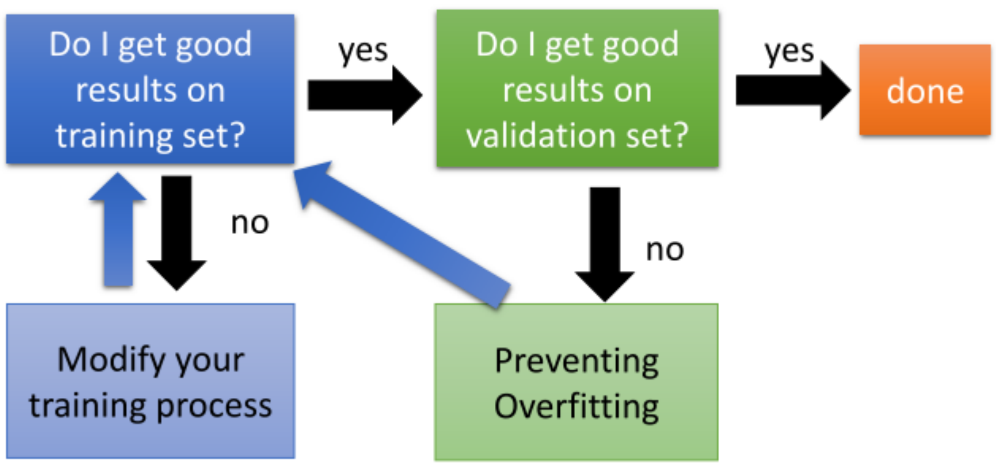
完整一個餘弦週期所需的 epoch 數。

這是最關鍵的參數，決定學習率會多久「降到底」一次。

•若 T_max=1000，表示過了 1000 次 scheduler.step() 之後，學習率會從初始值 lr_max 降到 eta_min。


Compare Learning Rate Curves Generated from Different Schedulers
Constant LR
Time-based Decay
Step-based Decay
Exponential Decay
linear Decay
Learning Rate
Epochs
0.0
0.1
0.2
0.3
0.4
0.5
0
20
40
60
80
100

Recipe for Learning



➢Your code usually do not have bug at this situation.

FOR EXAMPLE

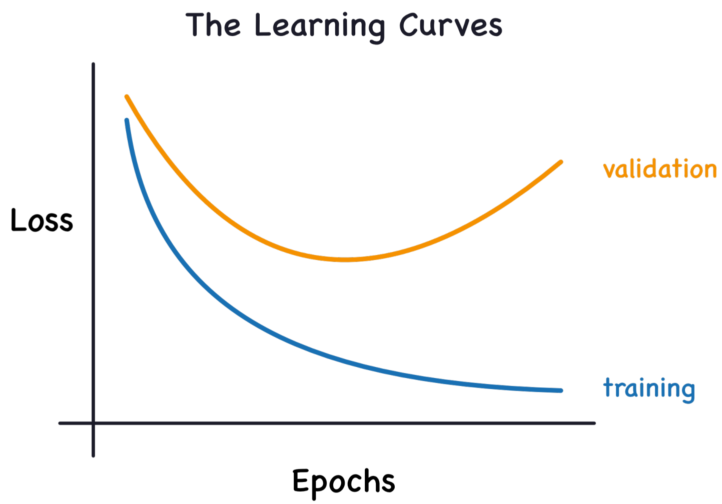
	MAE↓	MSE↓
Ours	0.9	1.4

FOR EXAMPLE

	MAE↓	MSE↓
Baseline	0.8	1.1
Ours	0.9	1.4

PROVIDE THE CLUES

- Baselines
- Training Curve
- Visualization
- Failure case analysis
- Ablation Study
- Gradient Analysis



Overfitting issue

Universal approximation theorem

9 languages

Article Talk | Read Edit View history Tools

From Wikipedia, the free encyclopedia

> This article **may be too technical for most readers to understand**. Please help improve it to make it understandable to non-experts, without removing the technical details. *(July 2023) (Learn how and when to remove this message)*

In the mathematical theory of artificial neural networks, **universal approximation theorems** are theorems[1][2] of the following form: Given a family of neural networks, for each function f from a certain function space, there exists a sequence of neural networks $\phi_1, \phi_2, \ldots$ from the family, such that $\phi_n \to f$ according to some criterion. That is, the family of neural networks is dense in the function space.

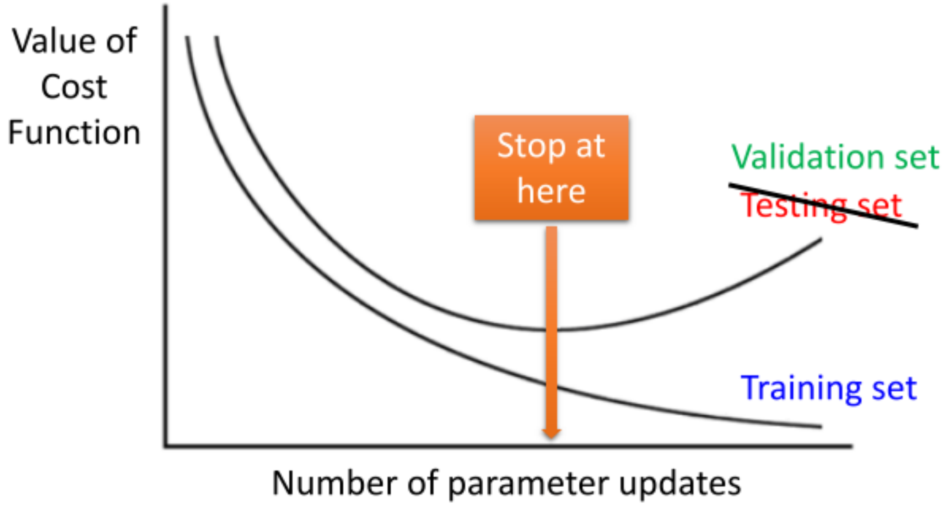
The most popular version states that feedforward networks with non-polynomial activation functions are dense in the space of continuous functions between two Euclidean spaces, with respect to the compact convergence topology.

Tricks

- Early Stopping
- Weight Decay
- Dropout
- Data Augmentation

Early Stopping

How many parameter updates do we need?



Tricks

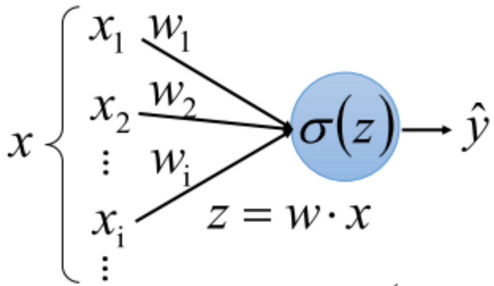
- Early Stopping
- Weight Decay
- Dropout
- Data Augmentation

Weight Decay

- The parameters closer to zero is preferred.

Training data:

$$\{(x, \hat{y}), \ldots\}$$



Testing data:

$$\{(x', \hat{y}), \ldots\} \qquad x' = x + \varepsilon$$

$$\begin{aligned} z' &= w \cdot (x + \varepsilon) \\ &= w \cdot x + w \cdot \varepsilon \\ &= z + w \cdot \varepsilon \end{aligned}$$

To minimize the effect of noise, we want w close to zero.

Weight Decay

- New cost function to be minimized
  - Find a set of weight not only minimizing original cost but also close to zero

$$\mathrm{C}'(\theta) = \underbrace{C(\theta)}_{\text{Original cost (e.g. minimize square error, cross entropy …)}} + \lambda \frac{1}{2} \underline{\|\theta\|^2} \rightarrow \text{Regularization term:}$$

$$\theta = \left\{ \mathrm{W}^1, \mathrm{W}^2, \ldots \right\}$$

$$\|\theta\|^2 = \left(w_{11}^1\right)^2 + \left(w_{12}^1\right)^2 + \ldots$$

$$+ \left(w_{11}^2\right)^2 + \left(w_{12}^2\right)^2 + \ldots$$

(not consider biases. why?)

Weight Decay

$$\|\theta\|^2 = \left(w_{11}^1\right)^2 + \left(w_{12}^1\right)^2 + \ldots$$
$$+ \left(w_{11}^2\right)^2 + \left(w_{12}^2\right)^2 + \ldots$$

- New cost function to be minimized

$$\mathrm{C}'(\theta) = C(\theta) + \lambda \frac{1}{2}\|\theta\|^2$$

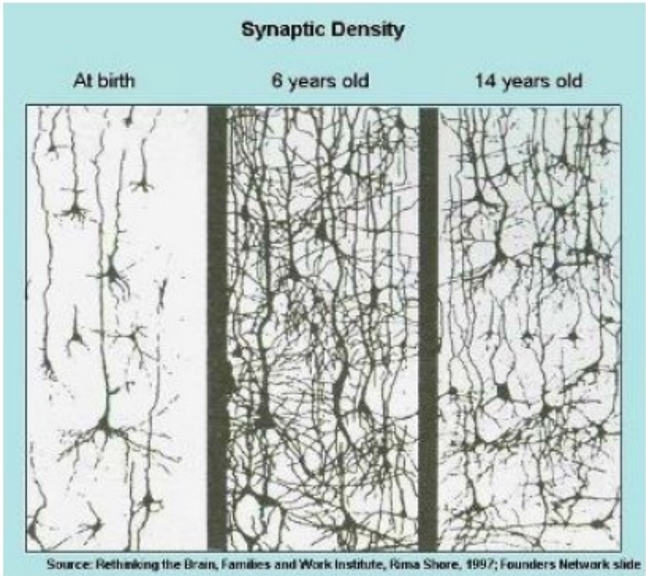
Gradient: $\frac{\partial \mathrm{C}'}{\partial w} = \frac{\partial \mathrm{C}}{\partial w} + \lambda w$

Update: $w^{t+1} \rightarrow w^t - \eta \frac{\partial \mathrm{C}'}{\partial w} = w^t - \eta\left(\frac{\partial \mathrm{C}}{\partial w} + \lambda w^t\right)$

$$= \underbrace{(1 - \eta\lambda) w^t}_{\text{Smaller and smaller}} - \eta \underline{\frac{\partial \mathrm{C}}{\partial w}}$$

Weight Decay

- Our Brain



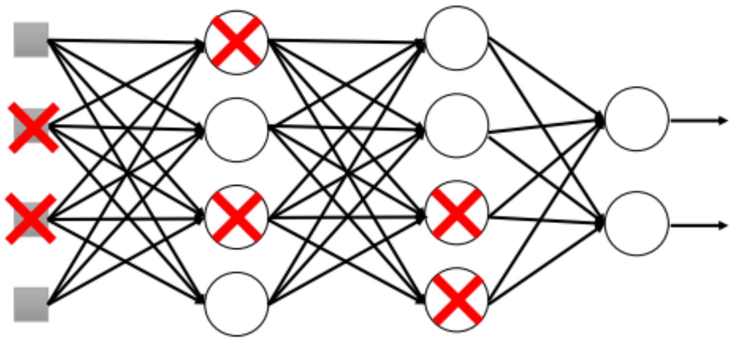
Tricks

- Early Stopping
- Weight Decay
- **Dropout**
- Data Augmentation

Dropout

$$\theta^t \leftarrow \theta^{t-1} - \eta \nabla C_x(\theta^{t-1})$$

Training:

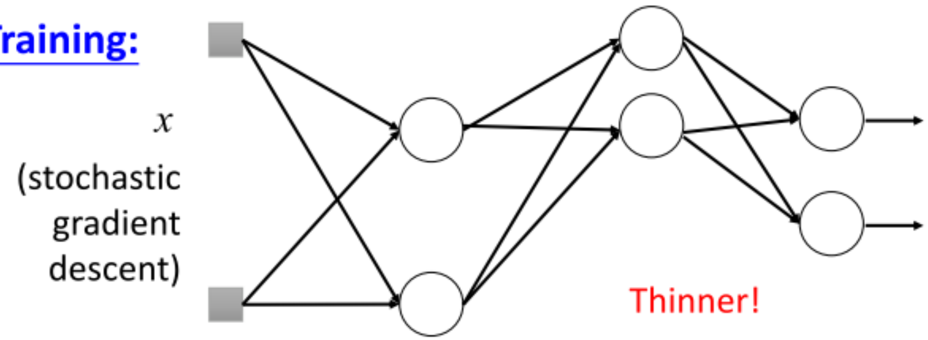


- In each ***iteration***
  - Each neuron has p% to dropout

Dropout

$\theta^t \leftarrow \theta^{t-1} - \eta \nabla C_x(\theta^{t-1})$

Training:

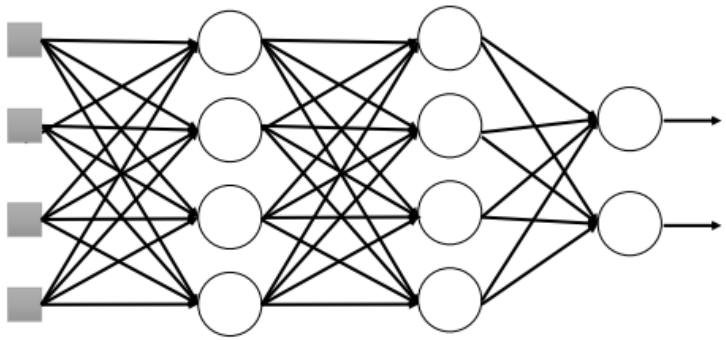


- In each ***iteration***
  - Each neuron has p% to dropout
    - **The structured of the network is changed.**
  - Using the new network for training

For each iteration, we resample the dropout neurons

Dropout

Testing:



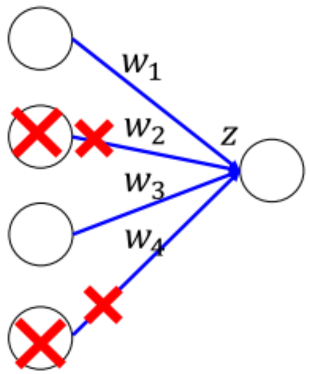
- **No dropout**
  - If the dropout rate at training is p%, all the weights times (1-p)%
  - Assume that the dropout rate is 50%. If $w_{ij}^{l} = 1$ from training, set $w_{ij}^{l} = 0.5$ for testing.

Dropout
- Intuitive Reason

- Why the weights should multiply (1-p)% (dropout rate) when testing?

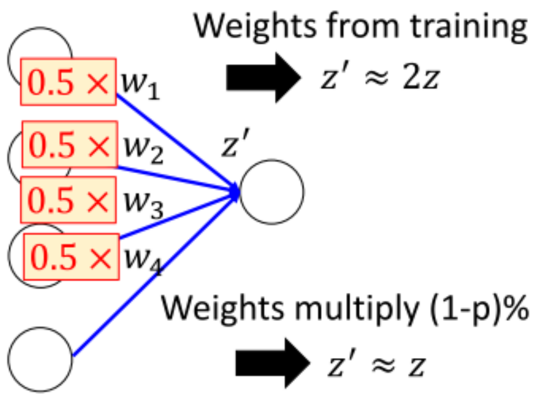
Training of Dropout

Assume dropout rate is 50%



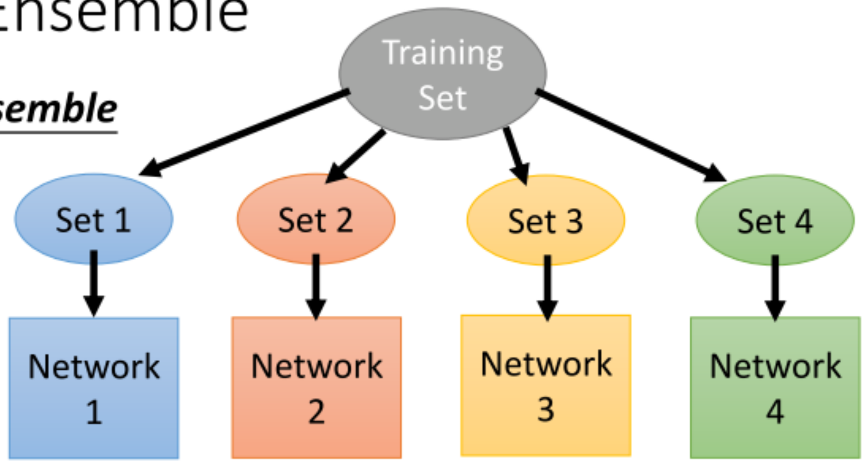
Testing of Dropout

No dropout



Dropout
- Ensemble

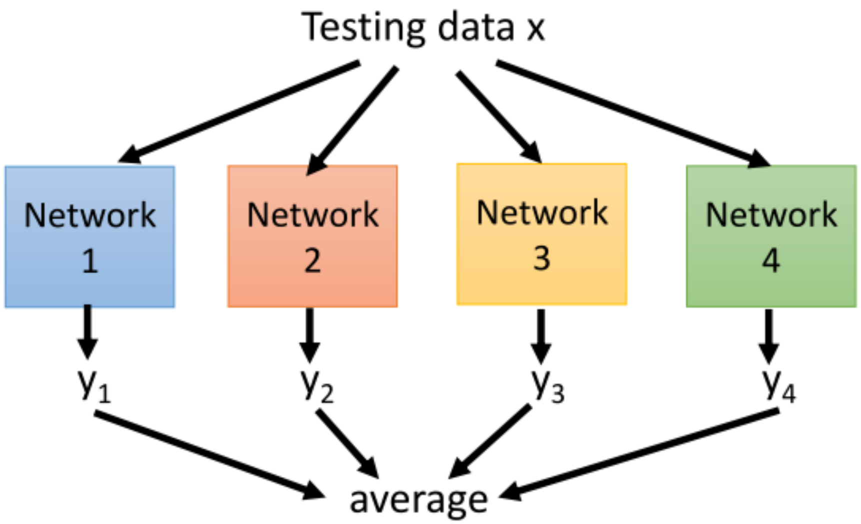
Ensemble



Train a bunch of networks with different structures

Dropout
- Ensemble

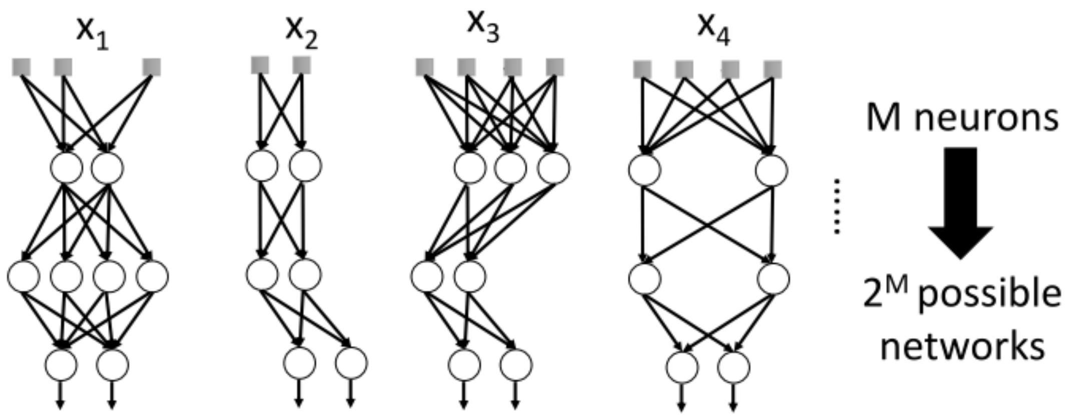
Ensemble



Dropout
- Ensemble

***Dropout* ≈ *Ensemble*.**

Training of Dropout

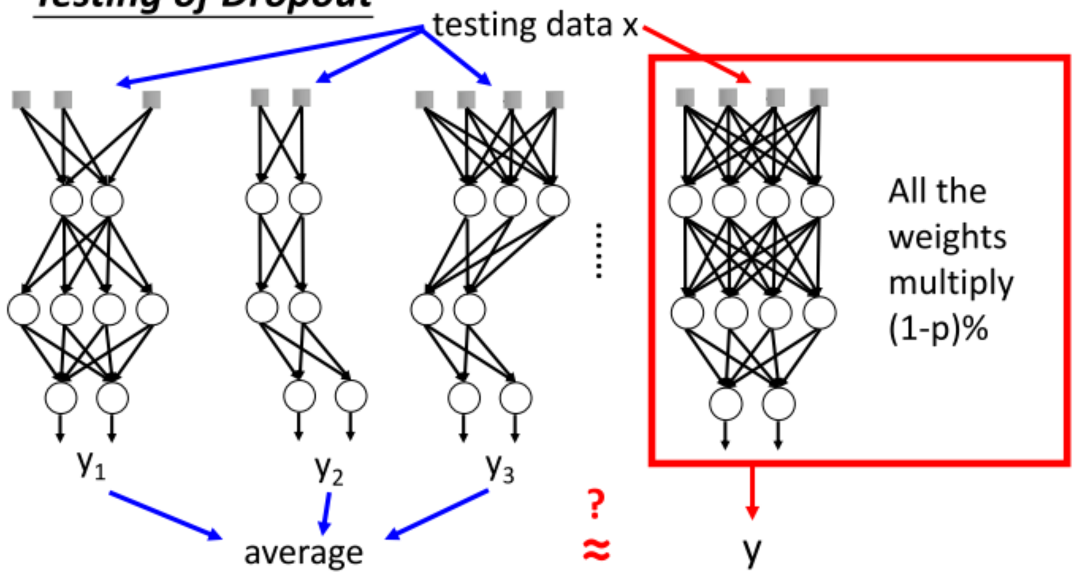


- Using one data to train one network
- Some parameters in the network are shared

Dropout
- Ensemble

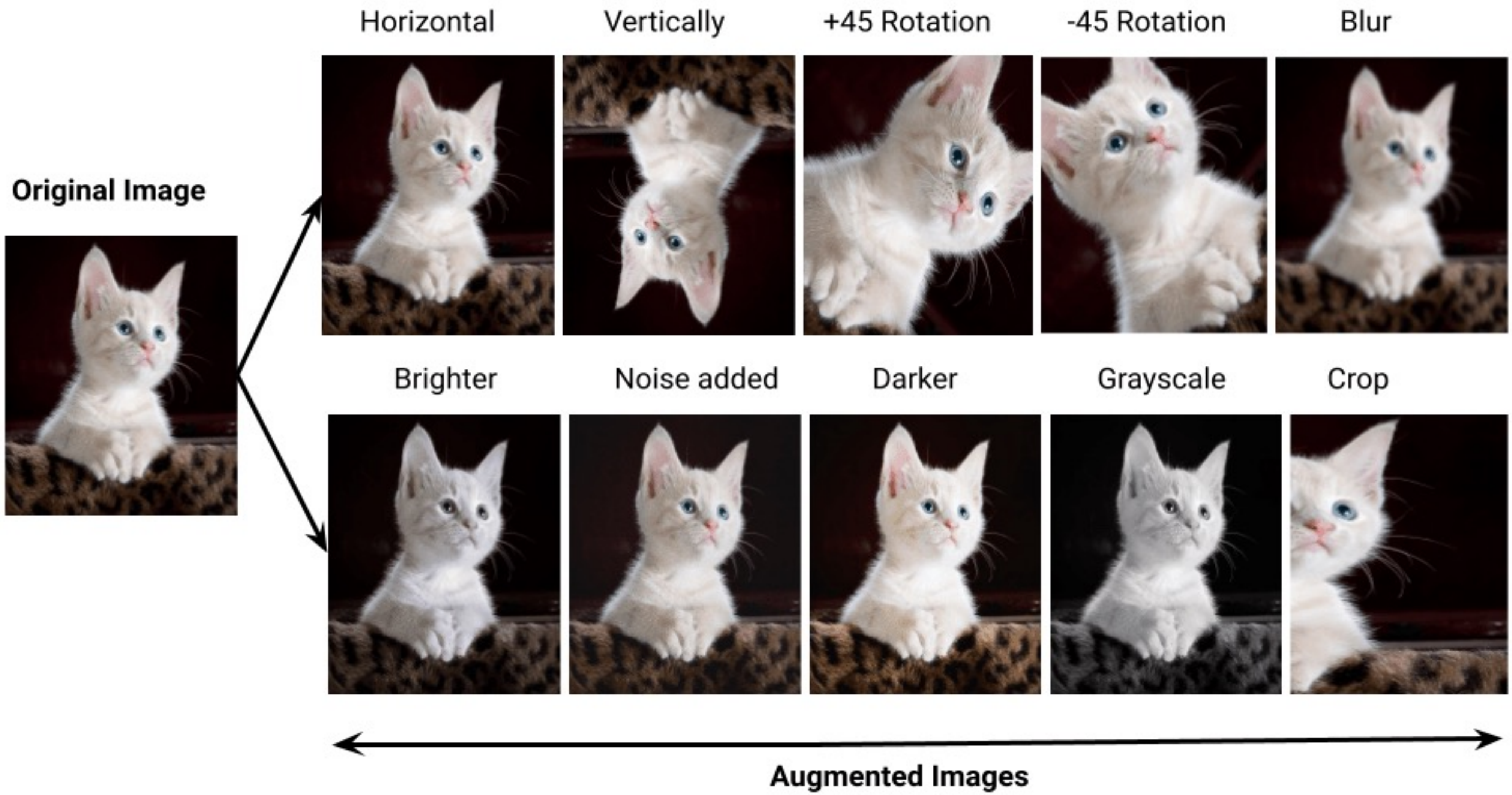
Dropout ≈ ***Ensemble***.

Testing of Dropout



Tricks

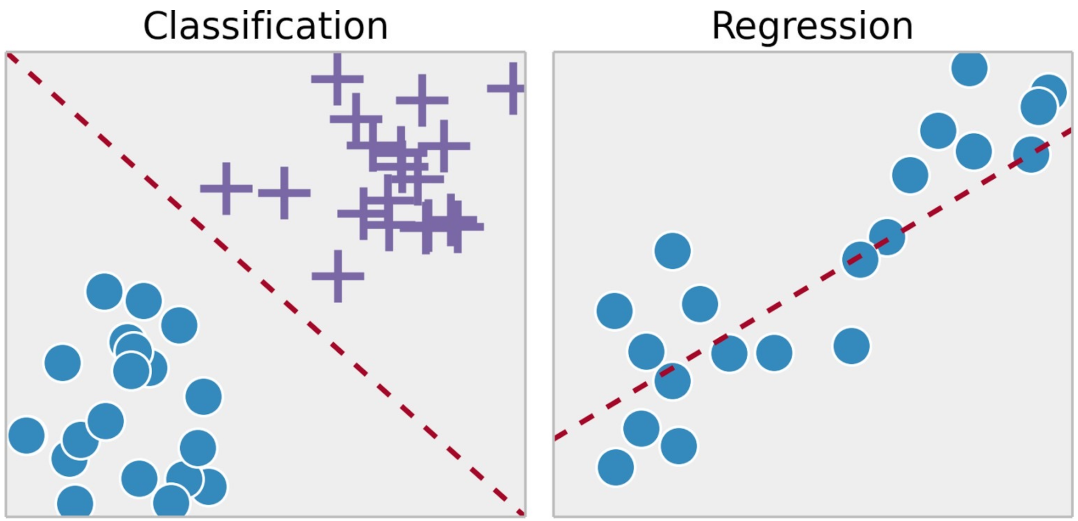
- Early Stopping
- Weight Decay
- Dropout
- Data Augmentation



https://ubiai.tools/what-are-the-advantages-anddisadvantages-of-data-augmentation-2023-update/

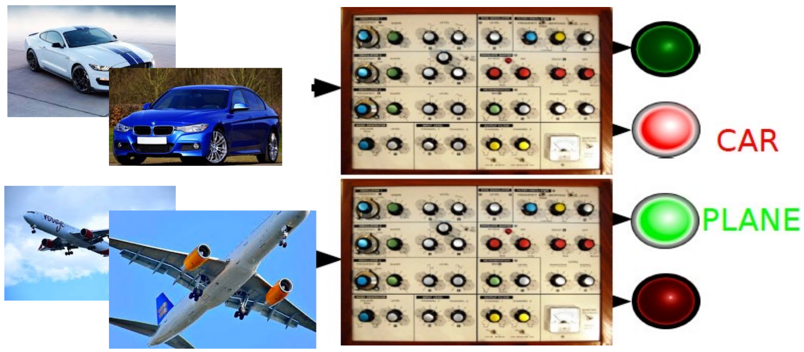
BASIC INTRODUCTION OF 12 KINDS OF LEARNING PROBLEMS

1. Supervised Learning


Classification
Regression

SUPERVISED LEARNING WORKS BUT REQUIRES MANY LABELED SAMPLES

- Training a machine by showing examples instead of programming it
- When the output is wrong, tweak the parameters of the machine
- Works well for:
  - Speech→words
  - Image→categories
  - Portrait→ name
  - Photo→caption
  - Text→topic



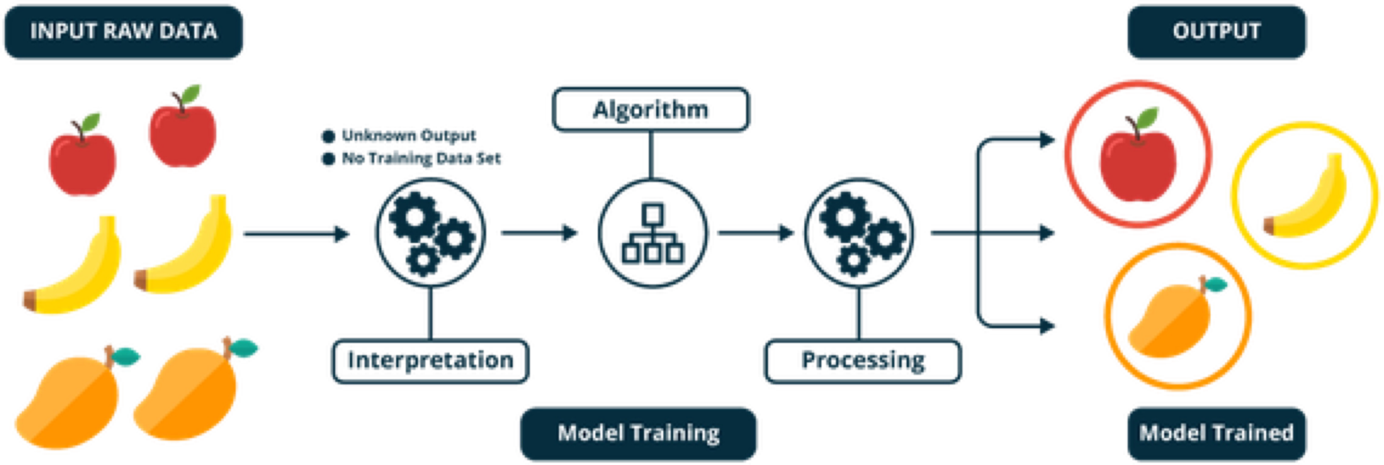
SUPERVISED DL WORKS AMAZINGLY WELL, WHEN YOU HAVE DATA

And services like Facebook, Instagram, Google, Youtube,… are built around it.

Content understanding, filtering, ranking, translation, accessibility….

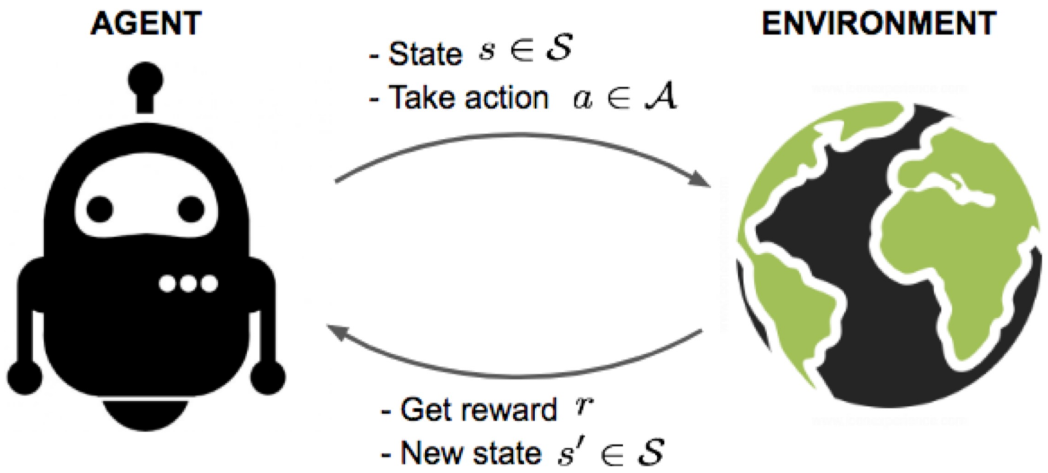
Content understanding, filtering, ranking, translation, accessibility….

2. UNSUPERVISED LEARNING


INPUT RAW DATA
Unknown Output
No Training Data Set
Algorithm
Interpretation
Processing
Model Training
OUTPUT
Model Trained

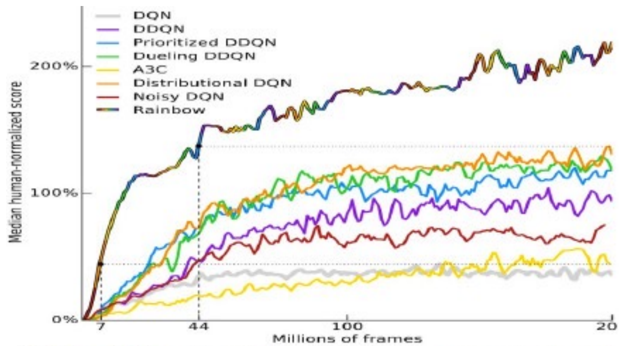
3. REINFORCEMENT LEARNING

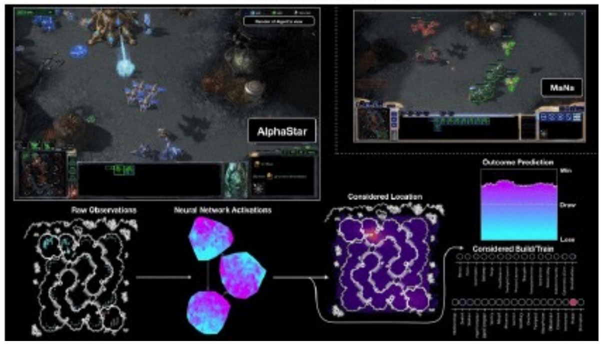
Recap: Reinforcement Learning

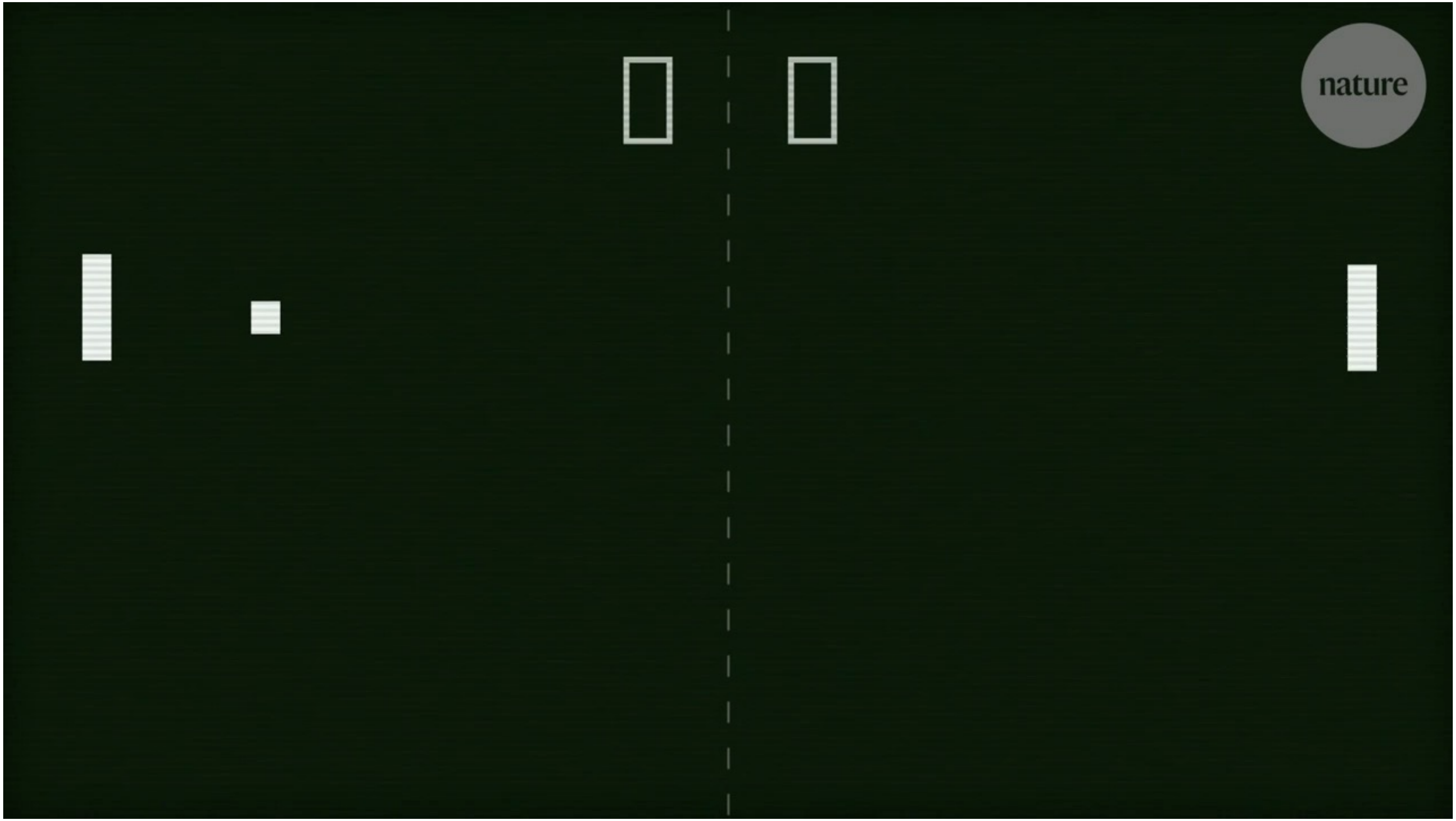


Reinforcement Learning: works great for games and simulations.

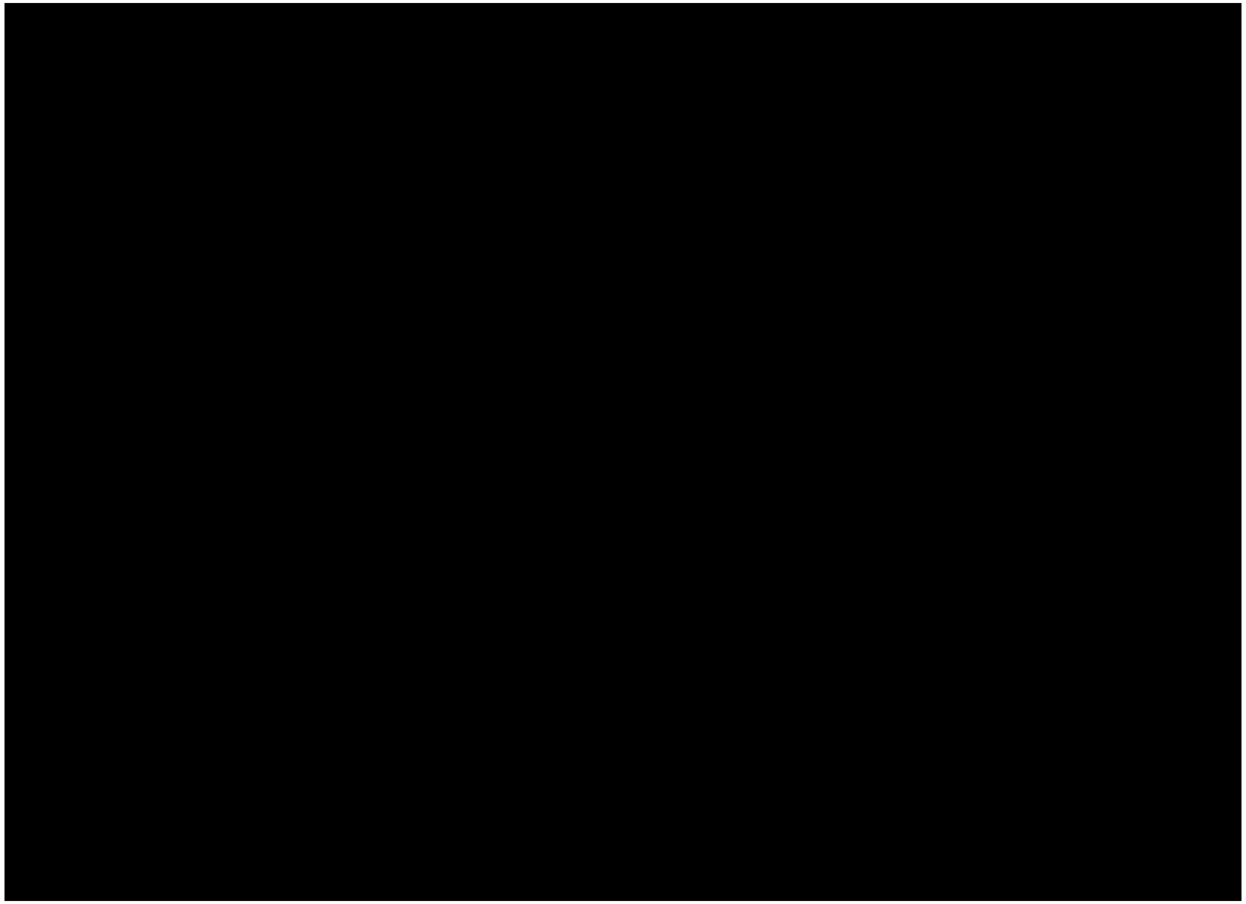
- 57 Atari games: takes **83 hours equivalent real-time** (18 million frames) to reach a performance that humans reach in 15 minutes of play.
  - [Hessel ArXiv:1710.02298]
- Elf OpenGo v2: 20 million self-play games. (2000 GPU for 14 days)
  - [Tian arXiv:1902.04522]
- StarCraft: AlphaStar 200 years of equivalent real-time play
  - [Vinyals blog post 2019]
- OpenAI single-handed Rubik's cube
  - 10,000 years of simulation



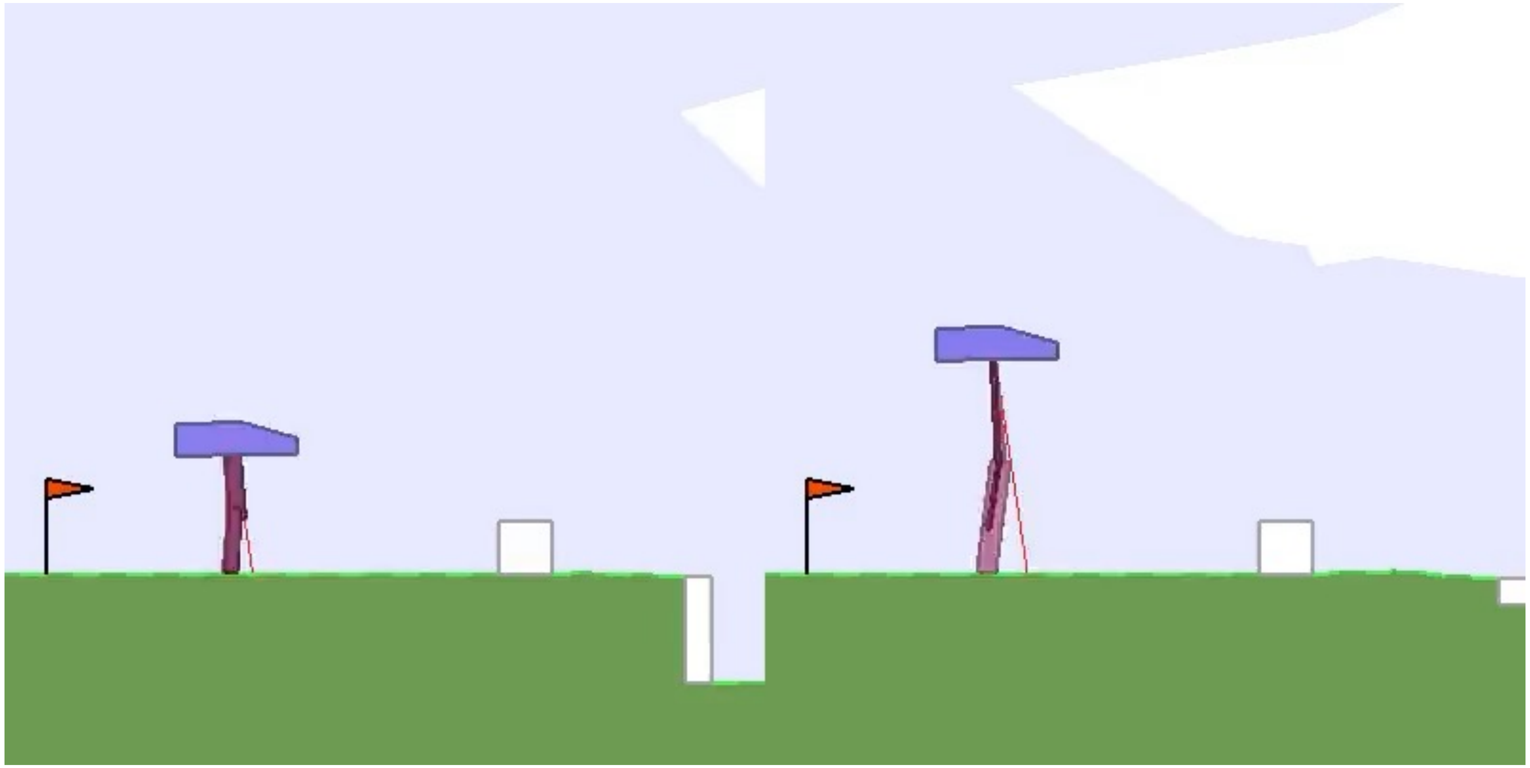


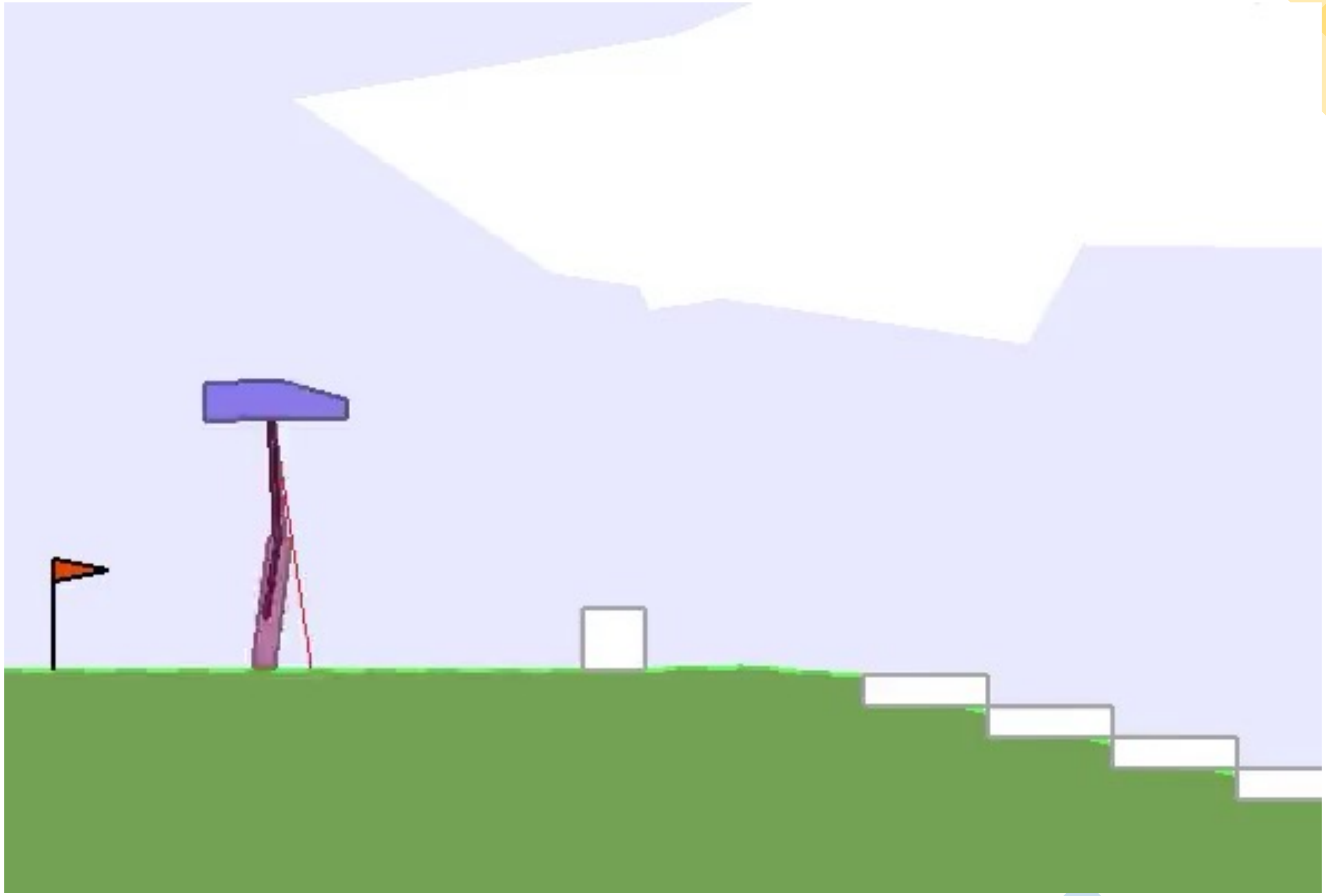

nature

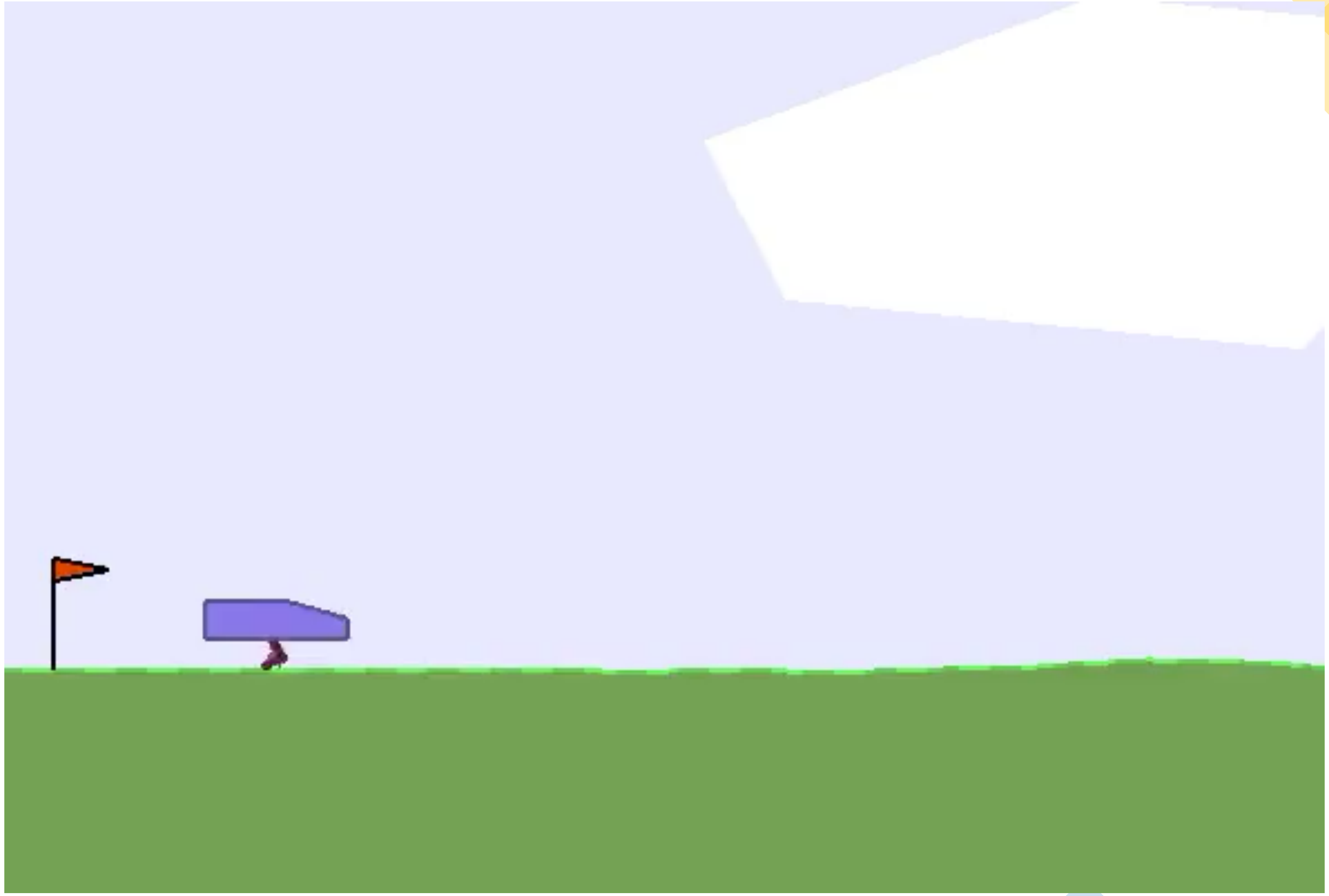
Producing flexible behaviours in simulated environments (DeepMind)

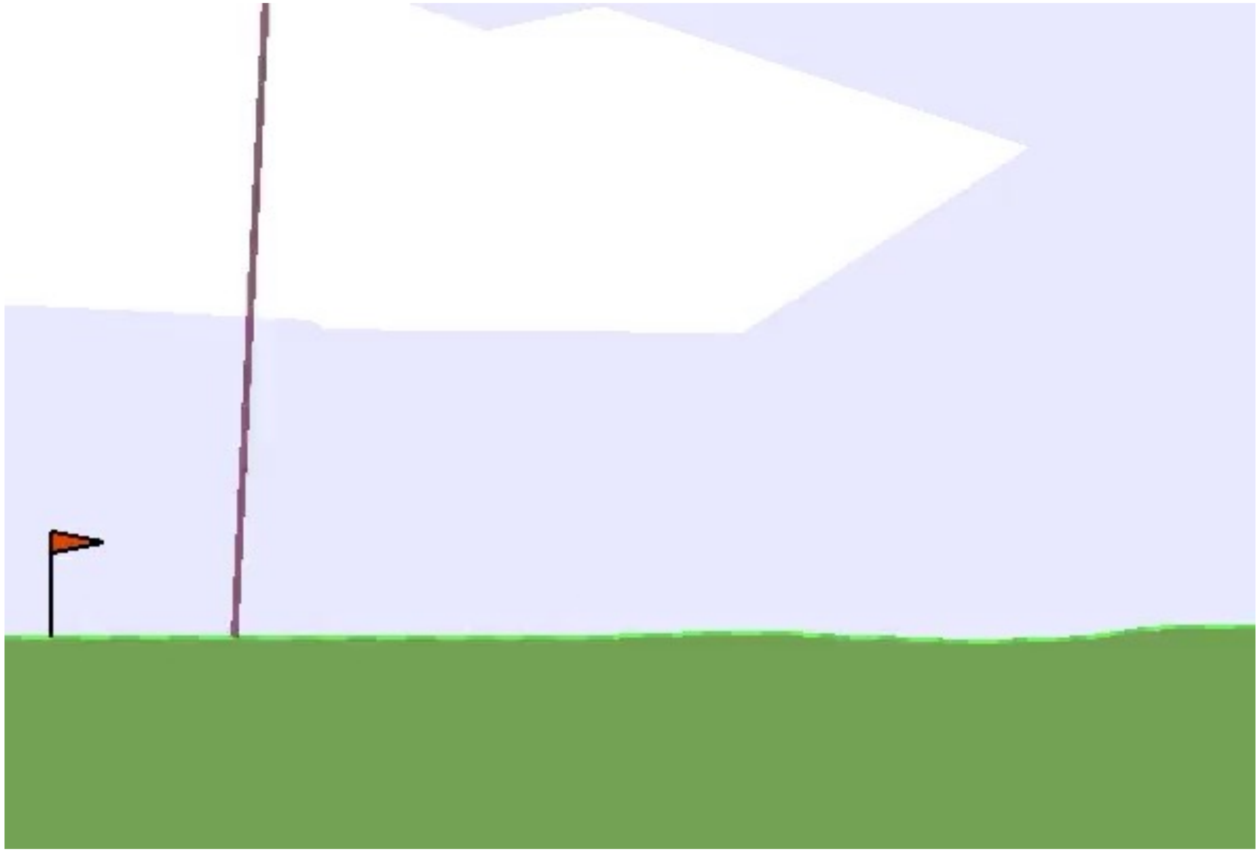


Reinforcement Learning for Improving Agent Design (NIPS'18)







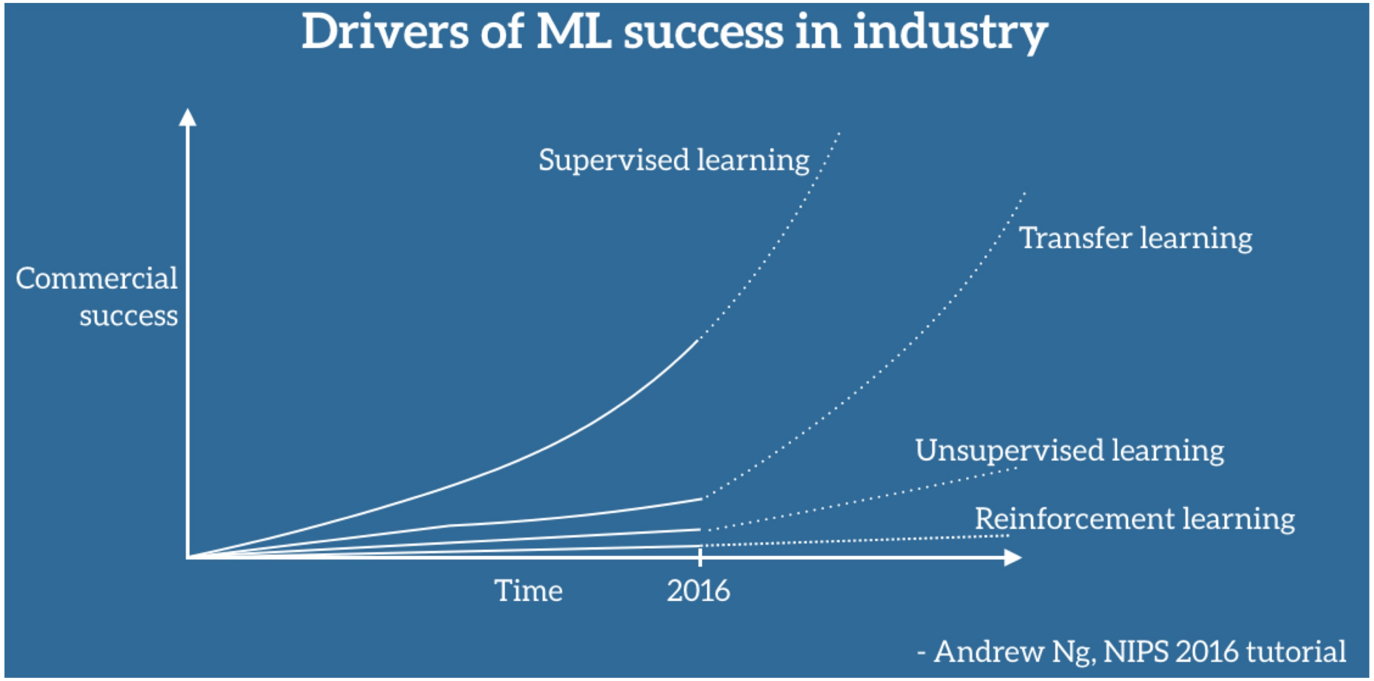


But RL Requires too many trials in the real world.

- Pure RL requires too many trials to learn anything
- it’s OK in a game
- it’s not OK in the real world
- RL works in simple virtual world that you can run faster than real-time on many machines in parallel.

- Anything you do in the real world can kill you
- You can’t run the real world faster than real time

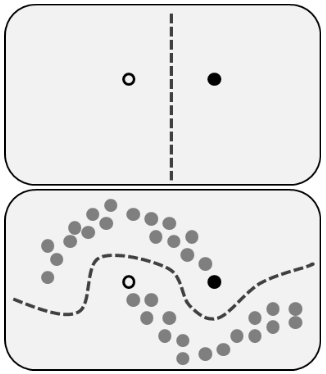
一般人看到的路况

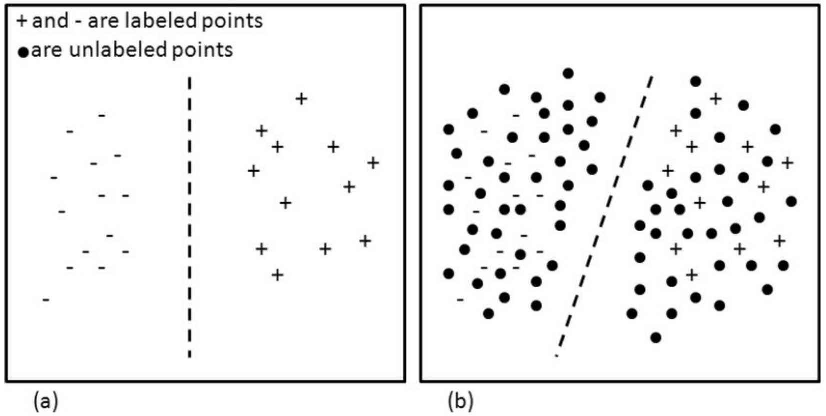

Drivers of ML success in industry
Supervised learning
Transfer learning
Commercial success
Unsupervised learning
Reinforcement learning
Time
2016
- Andrew Ng, NIPS 2016 tutorial

MIXUP PROBLEM

4. SEMI-SUPERVISED LEARNING

LABEL PROPAGATION





5. SELF-SUPERVISED LEARNING

How do humans
and animals learn
so quickly?

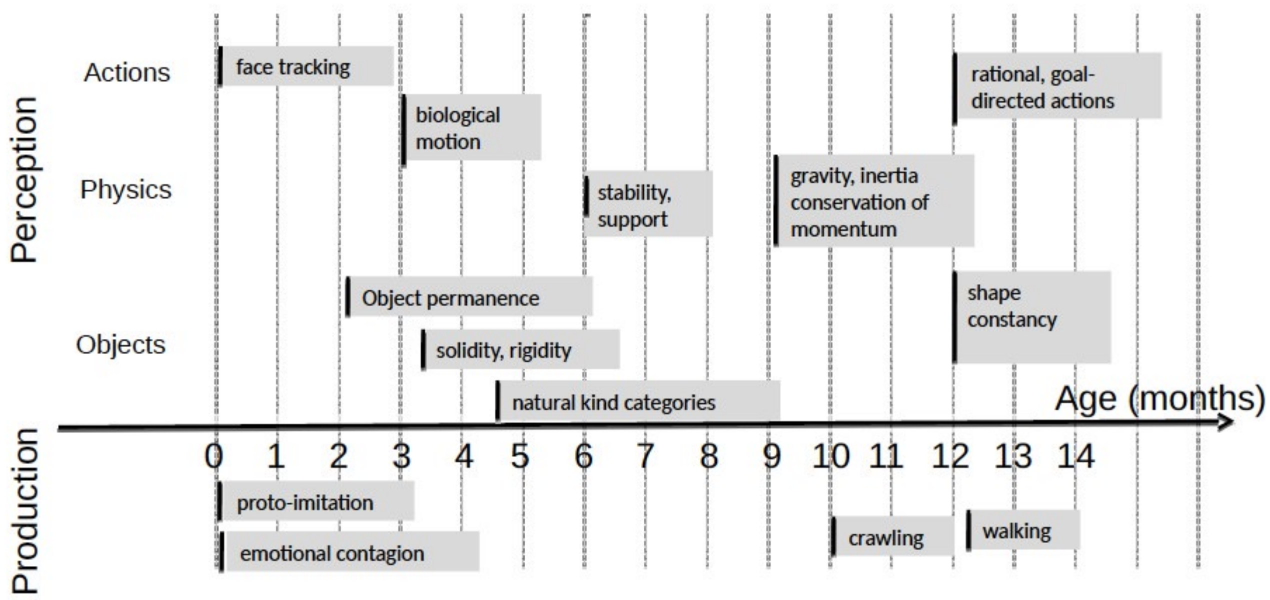
Not supervised.
Not Reinforced.

Babies learn how the world works by observation

- Largely by observation, with remarkably little interaction.

Photos courtesy of Emmanuel Dupoux

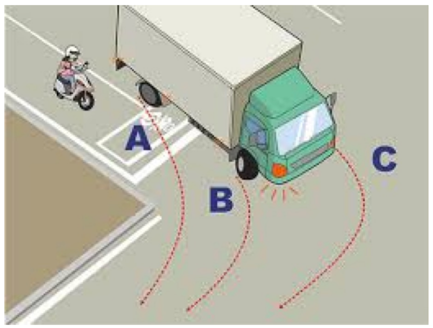
Early Conceptual Acquisition in Infants [from Emmanuel Dupoux]



Magic Trick

Prediction is the essence of Intelligence

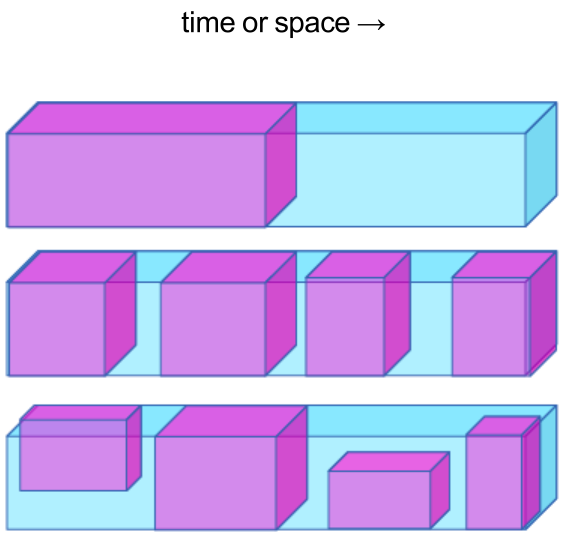
- We learn models of the world by predicting

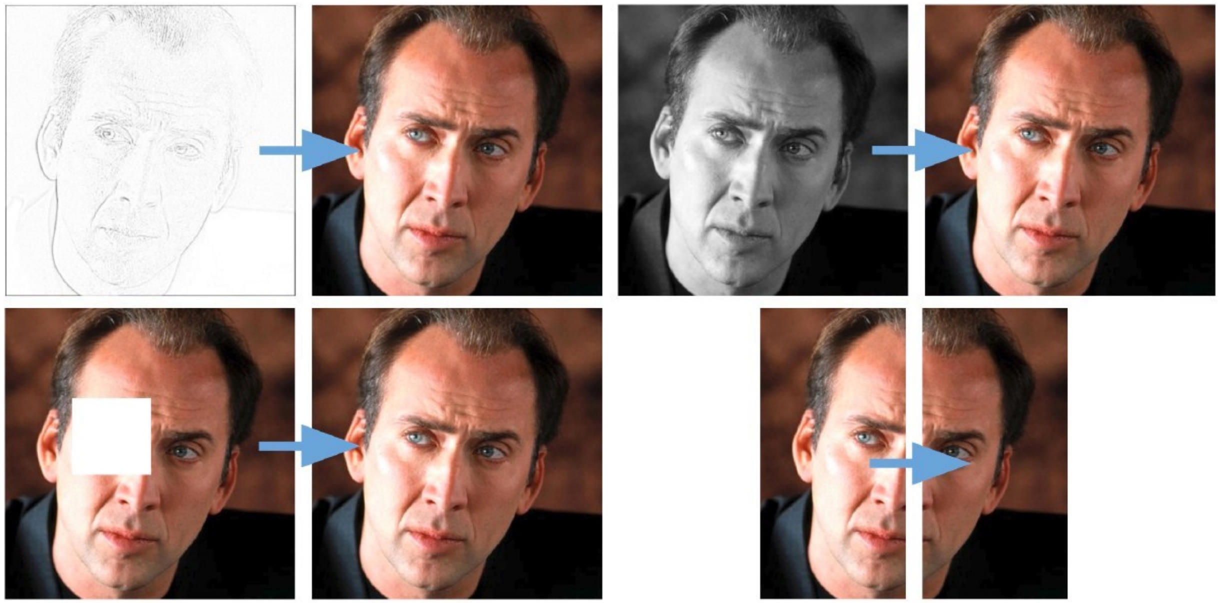


Self-Supervised Learning
= Filling in the Blanks

- Predict any part of the input from any part.
- Predict the future from the past.
- Predict the masked from the visible.
- Predict the any occluded part from all available parts.

- Pretend there is a part of the input you don't know and predict that.
- Reconstruction = SSL when any part could be known or unknown

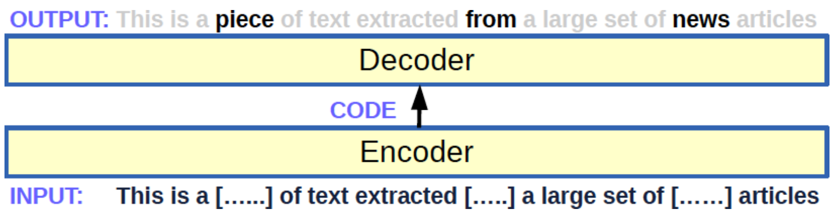




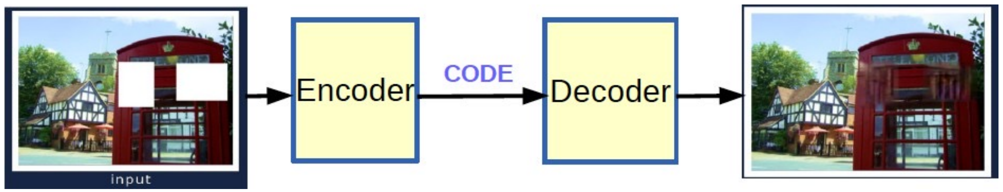
- The most important question that needs to be answered in order to use self-supervised learning is: “what pretext task should you use?”

Self-Supervised Learning: filling in the bl_nks

- Natural Language Processing: works great!



- Image Recognition / Understanding: works so-so



Colorization

https://www.fast.ai/2020/01/13/self_supervised/

Placing image patches in the right place

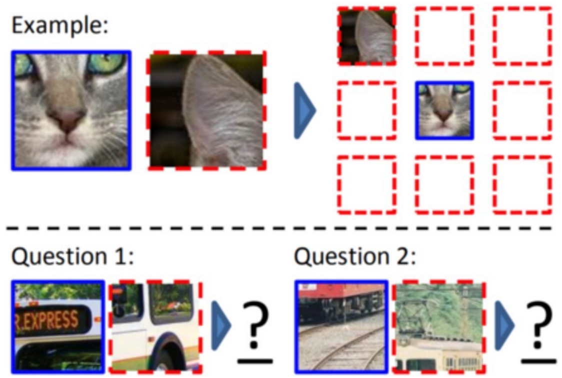
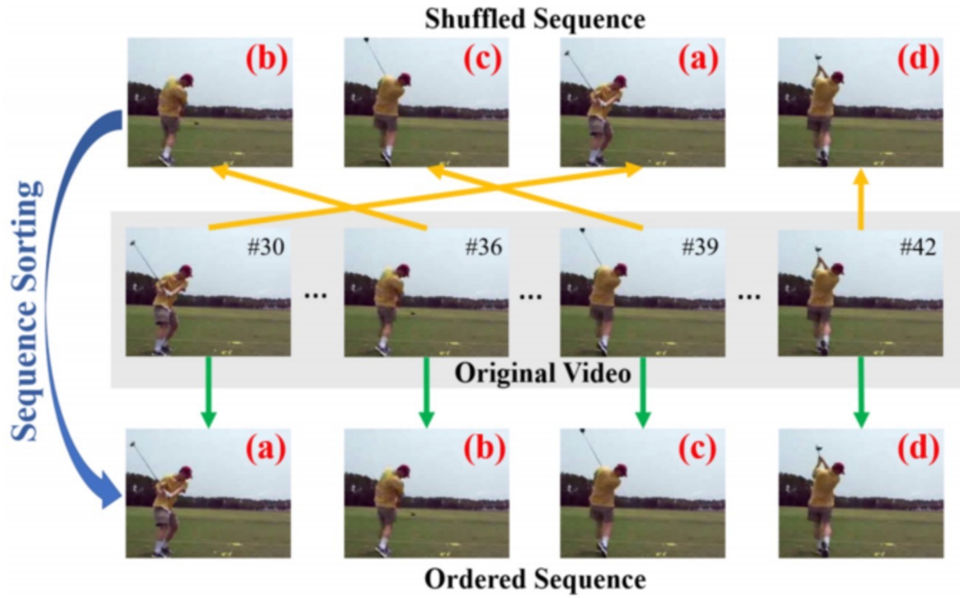


Figure 1. Our task for learning patch representations involves randomly sampling a patch (blue) and then one of eight possible neighbors (red). Can you guess the spatial configuration for the two pairs of patches? Note that the task is much easier once you have recognized the object!

Answer key: Q1: Bottom right Q2: Top center

https://www.fast.ai/2020/01/13/self_supervised/

Placing frames in the right order

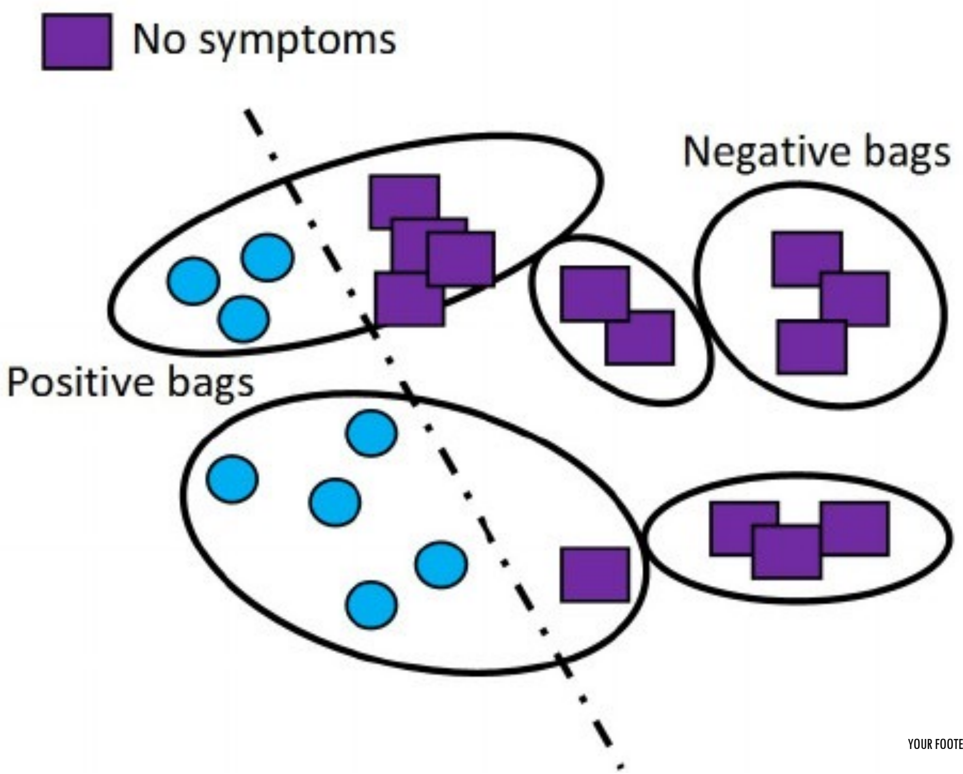


https://www.fast.ai/2020/01/13/self_supervised/

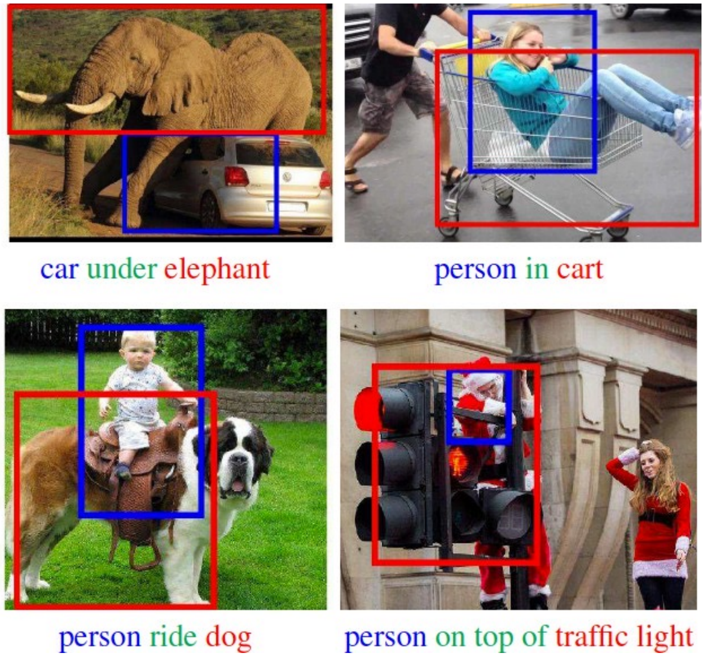
The Next AI Revolution

- Jitendra Malik: “Labels are the opium of the machine learning researcher”

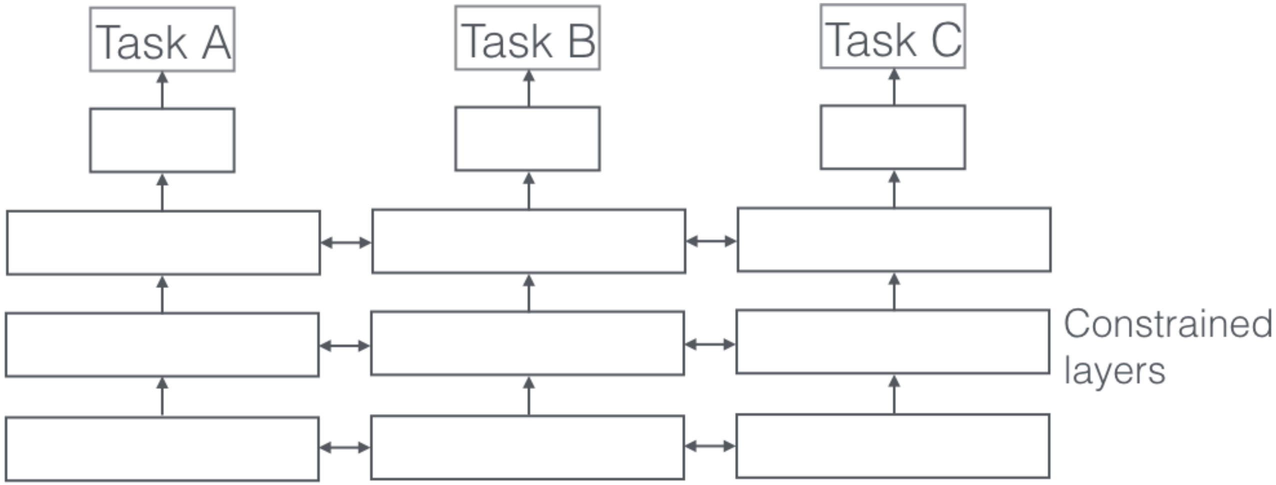
6. MULTI-INSTANCE LEARNING (HIGHLY RELATED TO WEAKLY-SUPERVISED LEARNING)


No symptoms
Negative bags
Positive bags

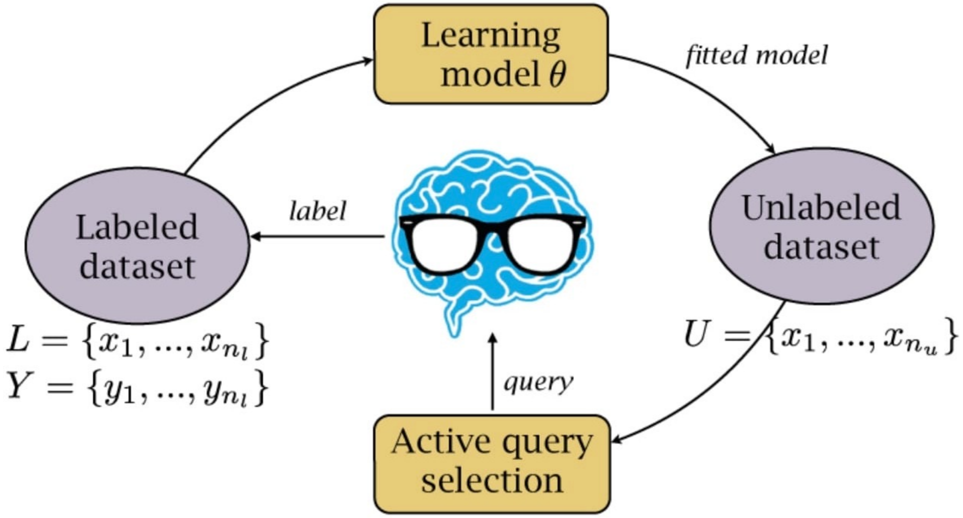
WEAKLY-SUPERVISED LEARNING



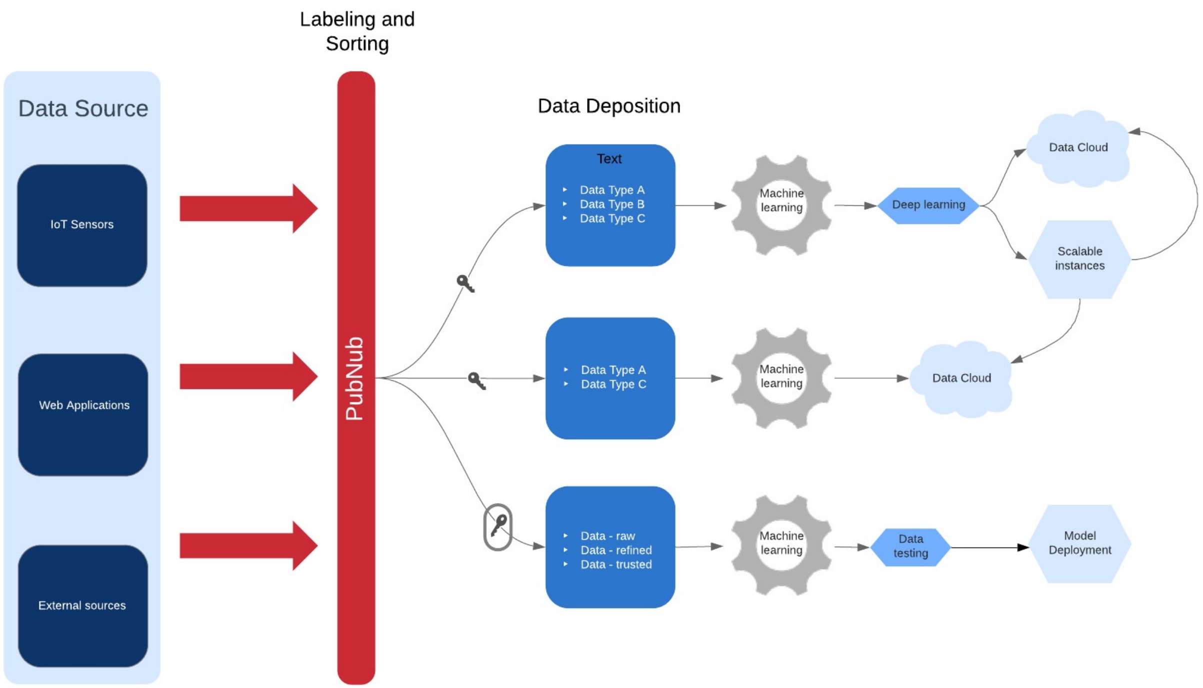
7. MULTI-TASK LEARNING


Task A
Task B
Task C
Constrained
layers

8. ACTIVE LEARNING


Learning
model θ
fitted model
Unlabeled
dataset
$U = \{x_1, ..., x_{n_u}\}$
Active query
selection
query
label
Labeled
dataset
$L = \{x_1, ..., x_{n_l}\}$
$Y = \{y_1, ..., y_{n_l}\}$

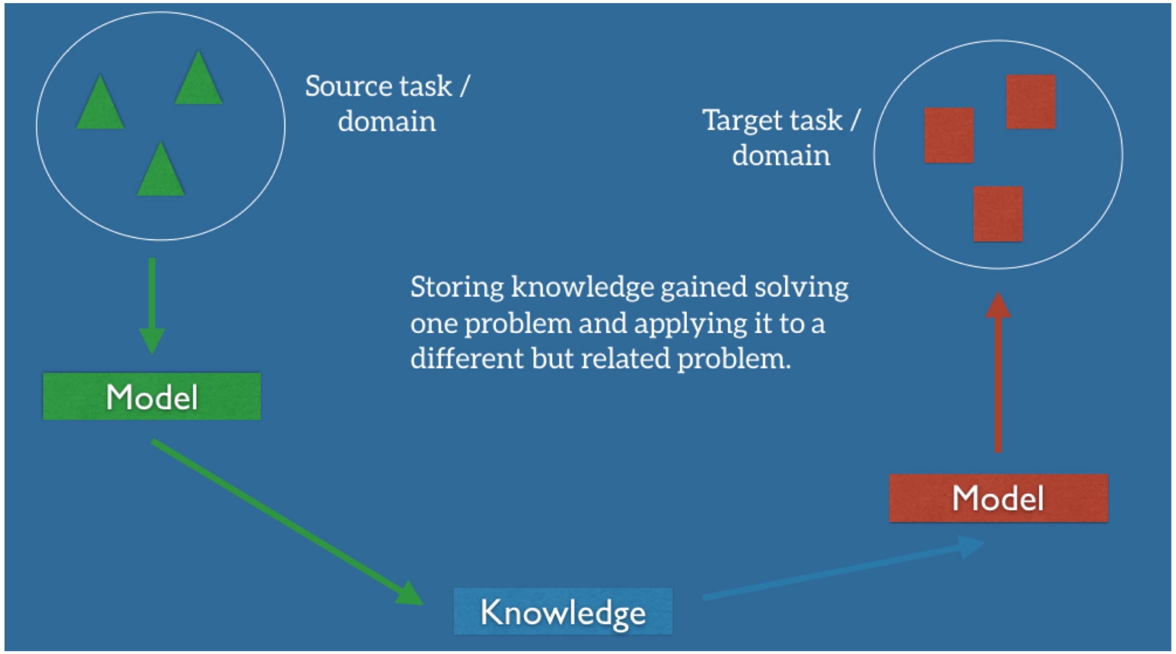
9. ONLINE LEARNING



Labeling and Sorting
Data Source
IoT Sensors
Web Applications
External sources
PubNub
Data Deposition
Text
Data Type A
Data Type B
Data Type C
Machine learning
Deep learning
Data Cloud
Scalable instances
Data Type A
Data Type C
Machine learning
Data Cloud
Data - raw
Data - refined
Data - trusted
Machine learning
Data testing
Model Deployment

10. TRANSFER LEARNING

TRANSFER LEARNING



11. ENSEMBLE Learning

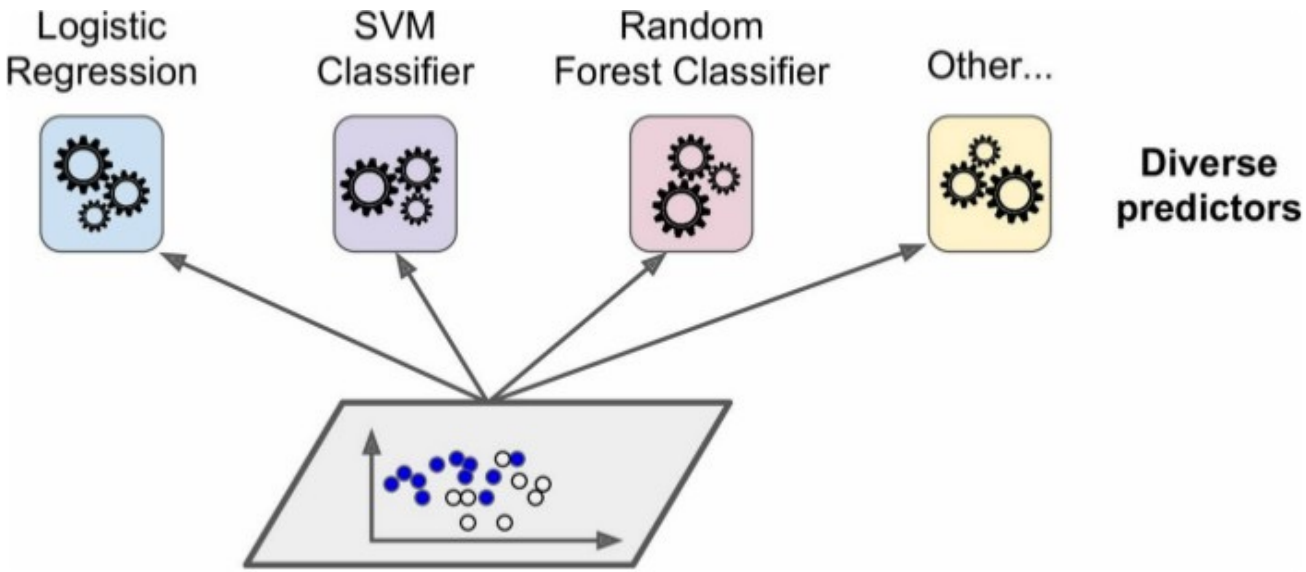
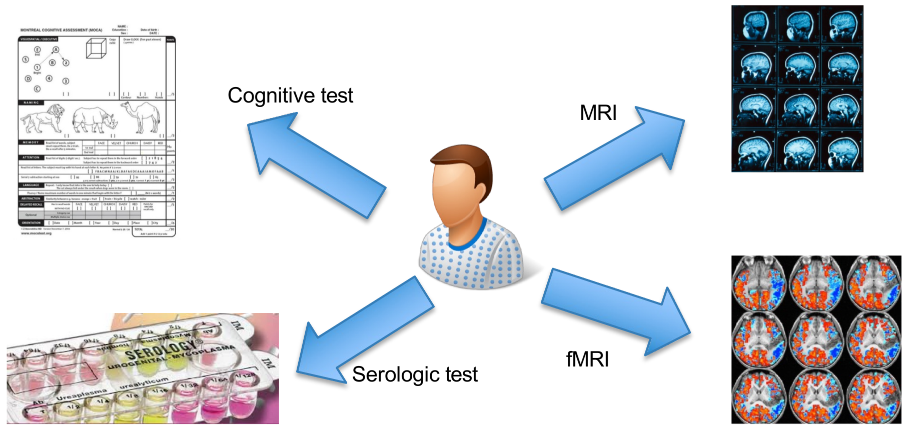


Figure 7-1. Training diverse classifiers

12. Multi-Source Learning

Multi-Source Learning

- Information come from different sources.
- Different sources may provide *compatible* and *complementary* information.



Three challenges for Deep Learning

- Deep Supervised Learning works well for perception
  - When labeled data is abundant.
- Deep Reinforcement Learning works well for action generation
  - When trials are cheap, e.g. in simulation.
- **Three problems the community is working on:**
- 1. Learning with fewer labeled samples and/or fewer trials
  - Self-supervised learning / unsup. learning / learning to fill in the blanks / learning to represent the world before learning tasks
- 2. Learning to reason, beyond “system 1” feed-forward computation.
  - Making reasoning compatible with gradient-based learning.
- 3. Learning to plan complex action sequences
  - Learning hierarchical representations of action plans

https://kahoot.it/

1^{st}-3^{rd}: 1 pt
4^{th} and 5^{th}: 0.5 pts

slido

Please download and install the
Slido app on all computers you use

Audience Q&A

ⓘ Start presenting to display the audience questions on this slide.