Università degli Studi di Padova

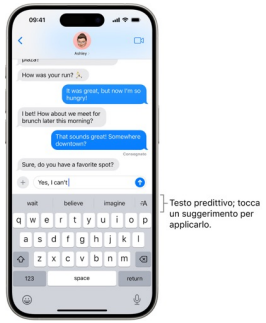
Machine Learning
2024/2025

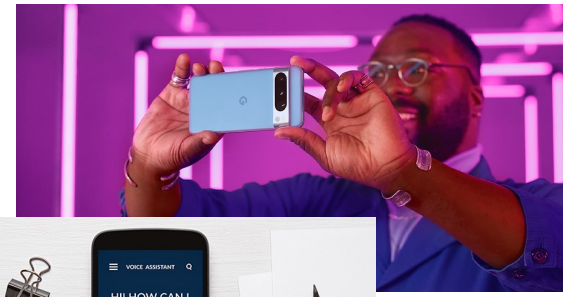
AMCO
ARTIFICIAL INTELLIGENCE, MACHINE LEARNING AND CONTROL RESEARCH GROUP

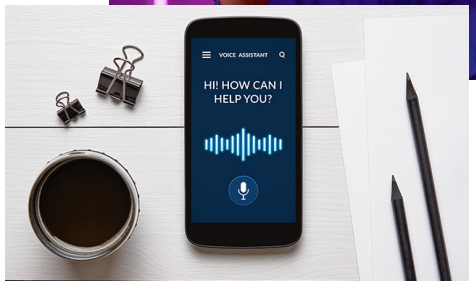
Lecture #36
What's next?

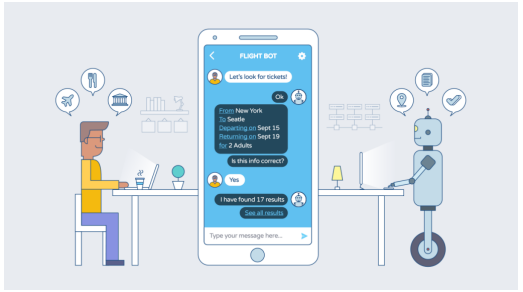
Gian Antonio Susto

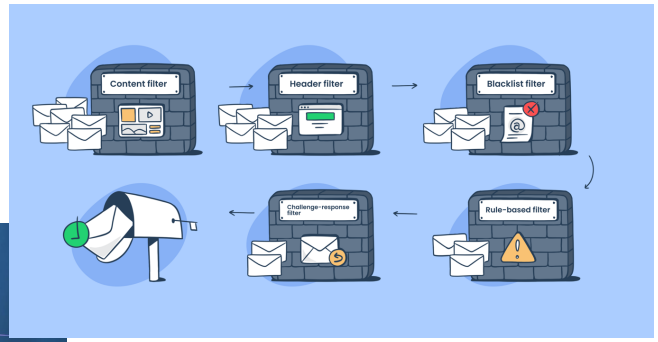
Many technologies that we rely on are based on ML!

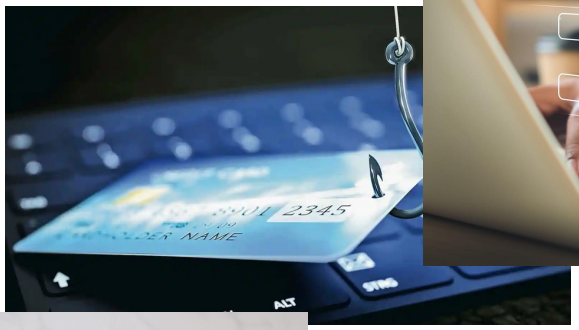


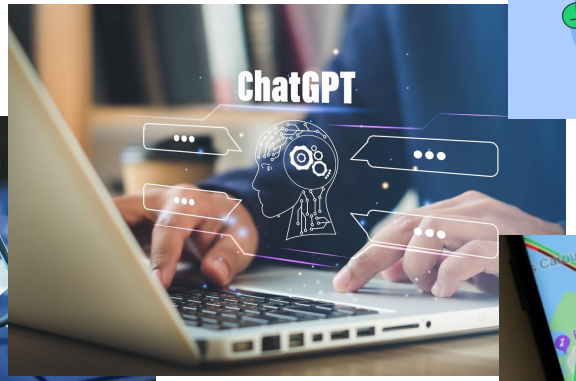


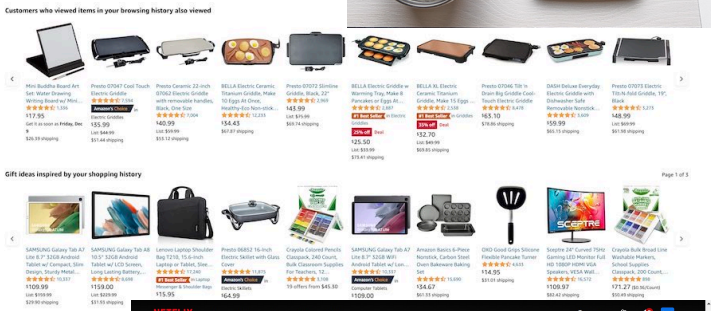


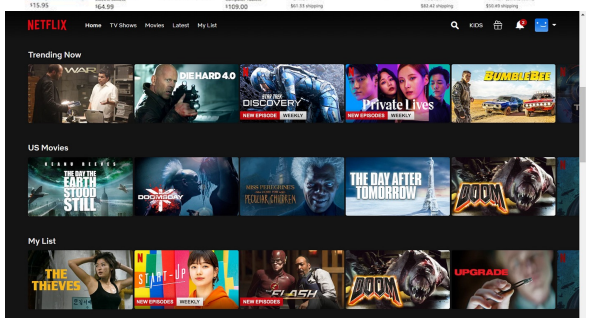


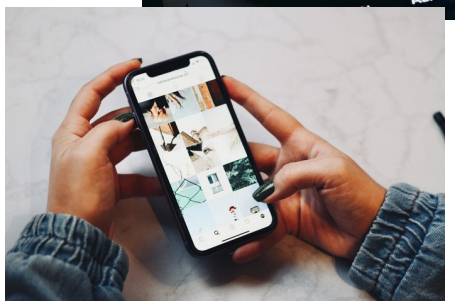


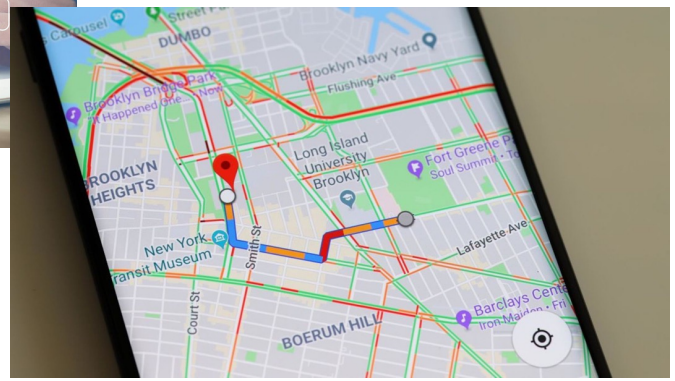


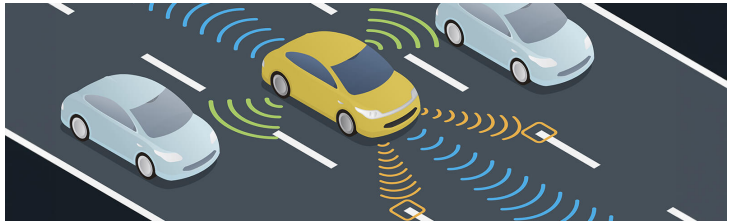






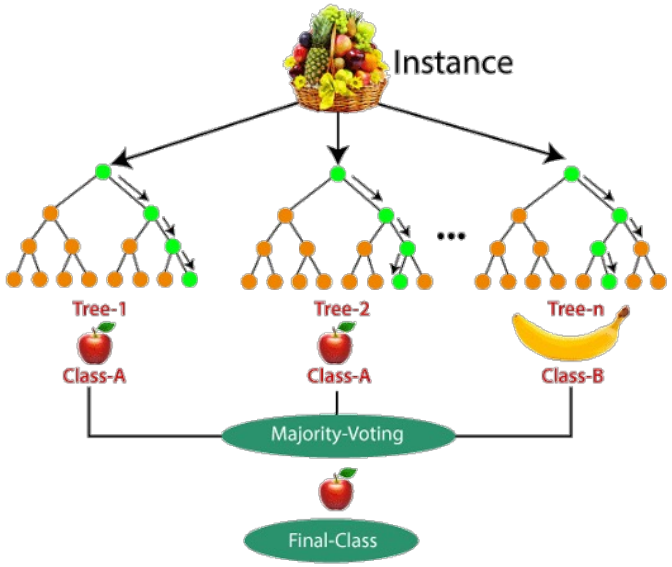


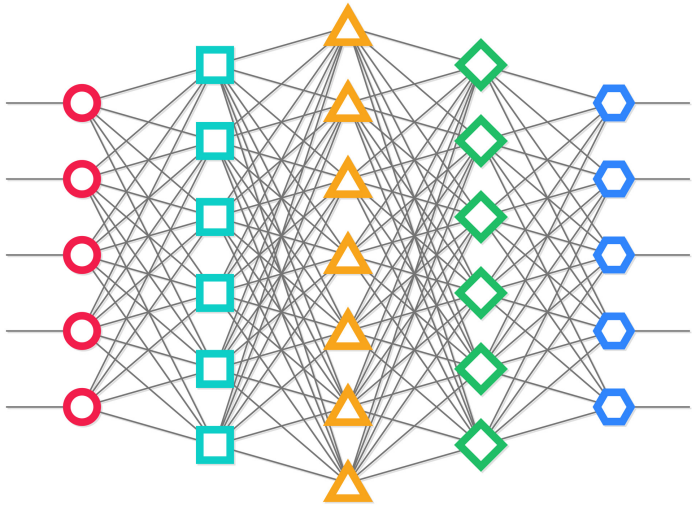


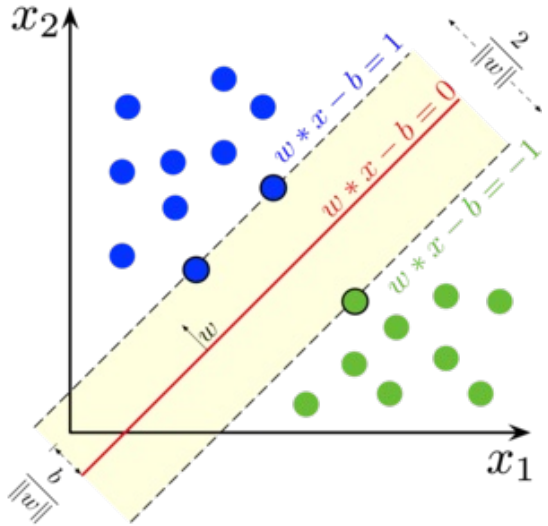


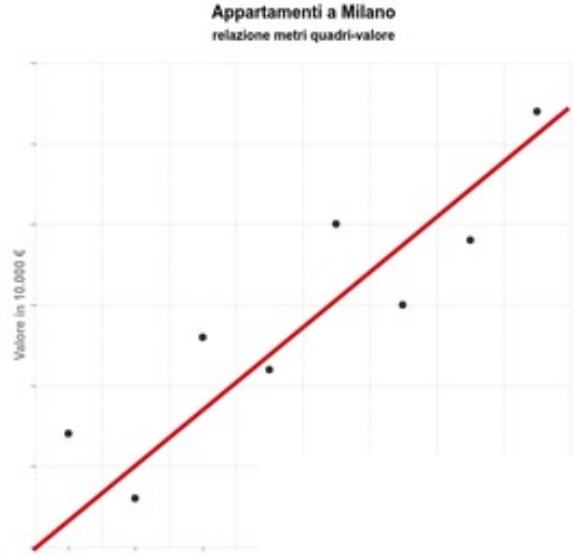
Recap: A Machine Learning pipeline

Recap: many model types...



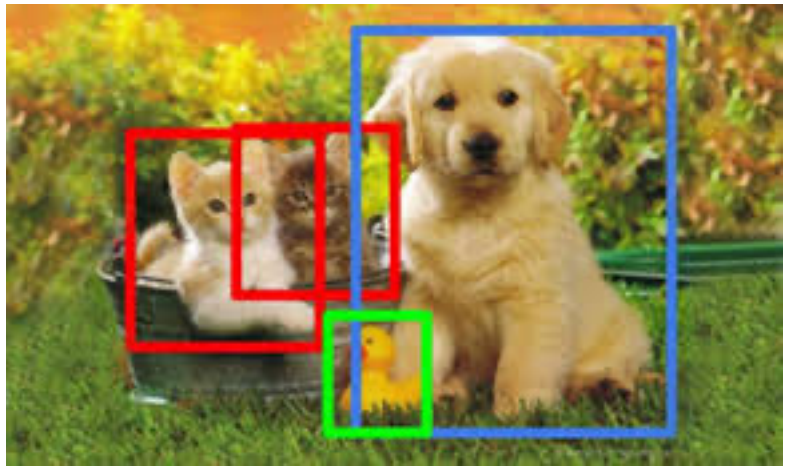


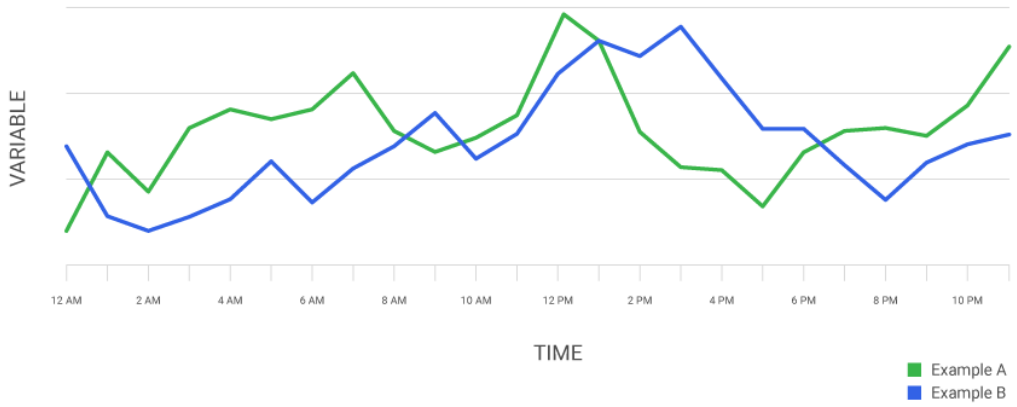




Recap: the data type

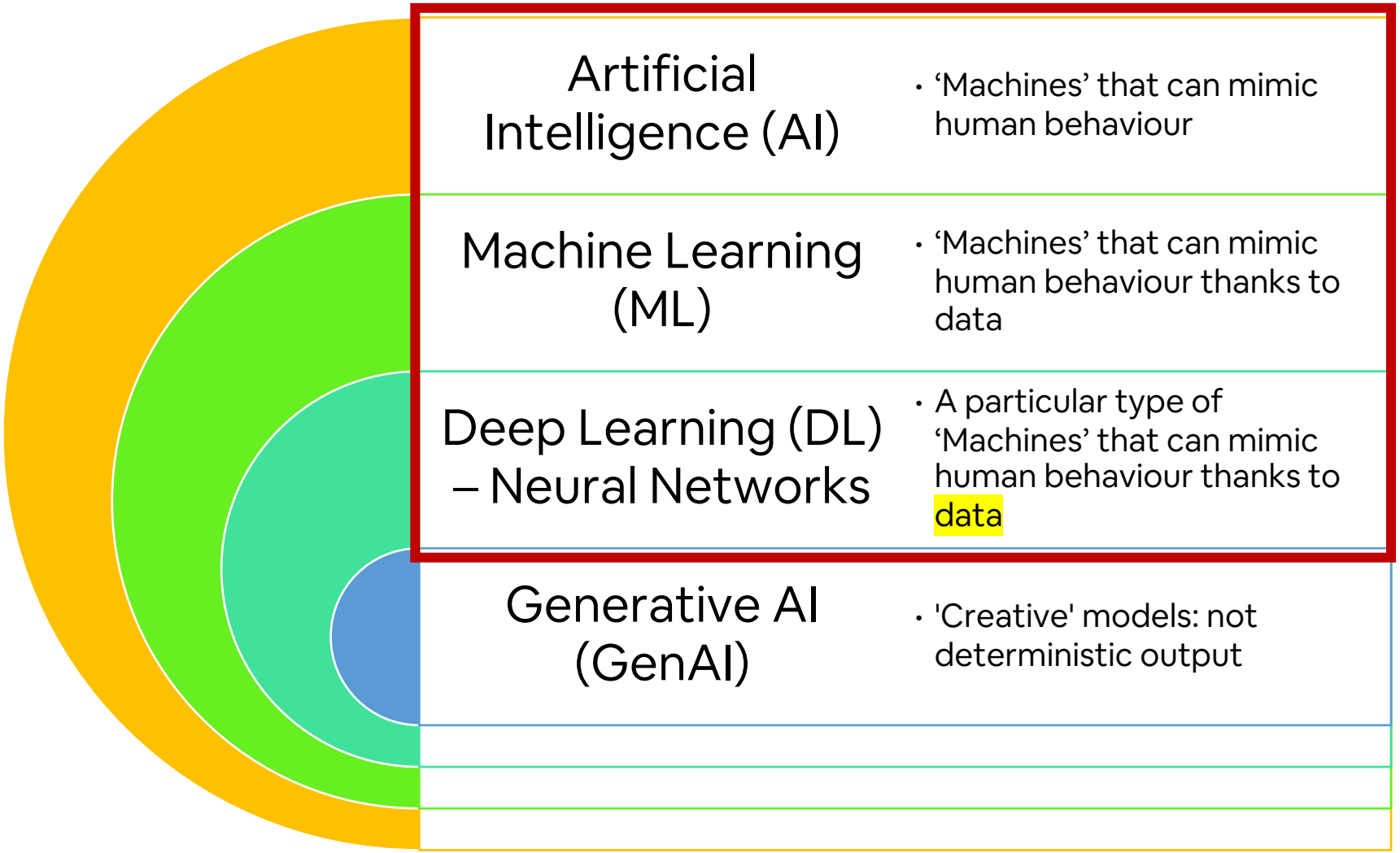
Different tasks (objectives), different models, different data type... and different stages of development!





39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K
37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	0	40	United-States	<=50K
49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-family	Black	Female	0	0	16	Jamaica	<=50K
52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	45	United-States	>50K
31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Female	14084	0	50	United-States	>50K
42	Private	159449	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	5178	0	40	United-States	>50K
37	Private	280464	Some-college	10	Married-civ-spouse	Exec-managerial	Husband	Black	Male	0	0	80	United-States	>50K
30	State-gov	141297	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	Asian-Pac-Islander	Male	0	0	40	India	>50K
23	Private	122272	Bachelors	13	Never-married	Adm-clerical	Own-child	White	Female	0	0	30	United-States	<=50K
32	Private	205019	Assoc-acdm	12	Never-married	Sales	Not-in-family	Black	Male	0	0	50	United-States	<=50K
40	Private	121772	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husband	Asian-Pac-Islander	Male	0	0	40	?	>50K
34	Private	245487	7th-8th	4	Married-civ-spouse	Transport-moving	Husband	Amer-Indian-Eskimo	Male	0	0	45	Mexico	<=50K
25	Self-emp-not-inc	176756	HS-grad	9	Never-married	Farming-fishing	Own-child	White	Male	0	0	35	United-States	<=50K
32	Private	186824	HS-grad	9	Never-married	Machine-op-inspct	Unmarried	White	Male	0	0	40	United-States	<=50K
38	Private	28887	11th	7	Married-civ-spouse	Sales	Husband	White	Male	0	0	50	United-States	<=50K
43	Self-emp-not-inc	292175	Masters	14	Divorced	Exec-managerial	Unmarried	White	Female	0	0	45	United-States	>50K
40	Private	193524	Doctorate	16	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	60	United-States	>50K
54	Private	302146	HS-grad	9	Separated	Other-service	Unmarried	Black	Female	0	0	20	United-States	<=50K
35	Federal-gov	76845	9th	5	Married-civ-spouse	Farming-fishing	Husband	Black	Male	0	0	40	United-States	<=50K
43	Private	117037	11th	7	Married-civ-spouse	Transport-moving	Husband	White	Male	0	2042	40	United-States	<=50K
59	Private	109015	HS-grad	9	Divorced	Tech-support	Unmarried	White	Female	0	0	40	United-States	<=50K

Recap: The keywords



Recap: The ’frameworks’

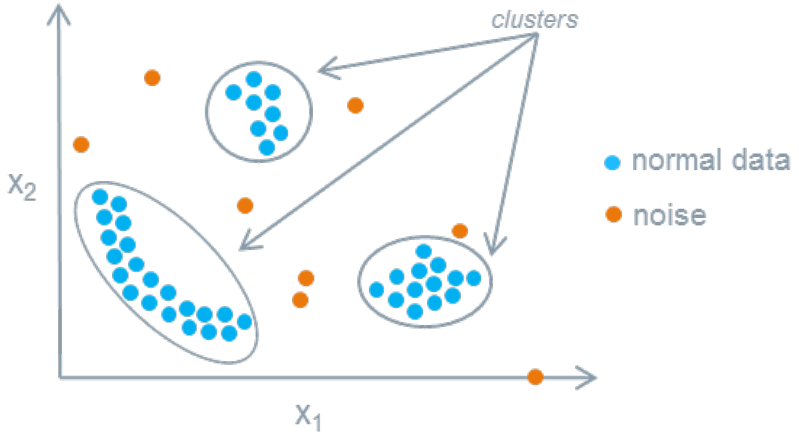
Supervised Learning

Setup: Observation of the environment

Data: (x,y)

Task: learn a map from inputs x to outputs y

Unsupervised Learning



Setup: Observation of the environment

Data: x (no labels)

Task: learn patterns in input data

Reinforcement Learning

Setup: Interaction with the environment

Data: (state,action, rewards)

Task: learn policies that maximize rewards

Disclaimer: today’s lecture

Elements of GenAI

Data Generation

- Generative Models example:

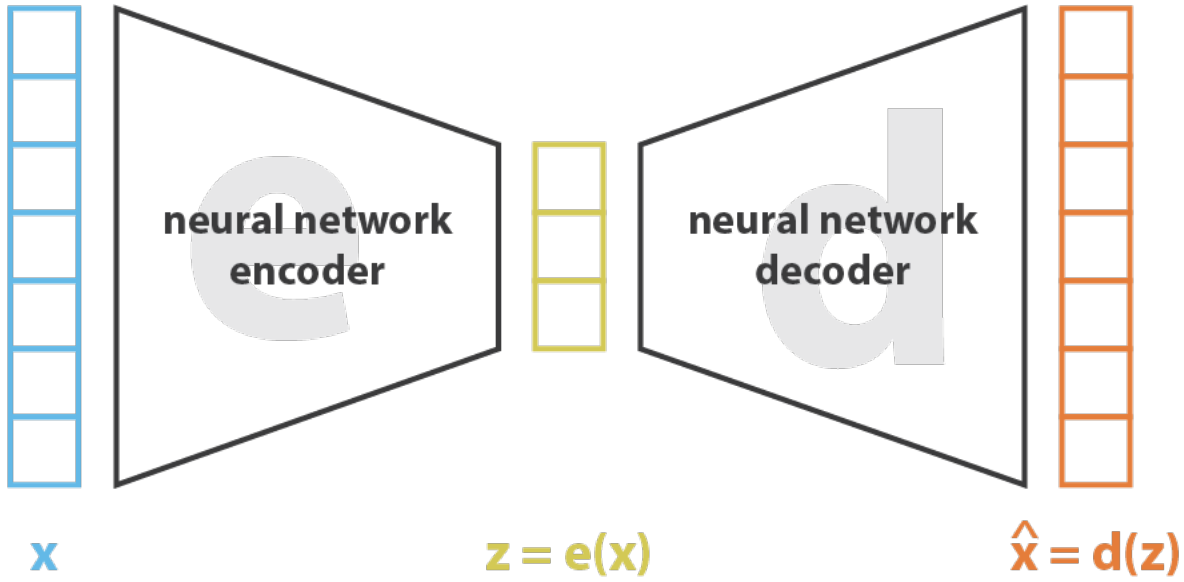
Variational Autoencoder

- Generative Models aims at learning useful representations and to generate new samples from a complex distribution that they model where the data are sampled from

https://thispersondoesnotexist.com/

Autoencoders

- Deterministic models trained using error backpropagation
- Input and Output are the same data: we force a network to be able to reconstruct such data with the limitation of having a ‘bottleneck’ (code) of limited size

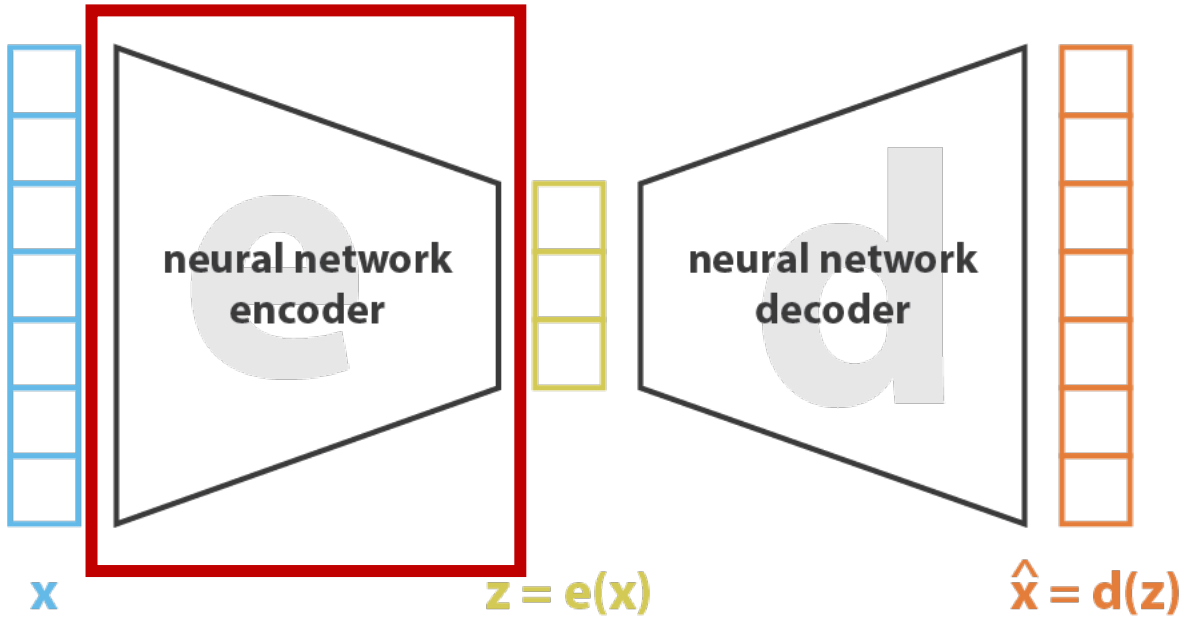


$$\text{loss} = \| x - \hat{x} \|^2 = \| x - d(z) \|^2 = \| x - d(e(x)) \|^2$$

Autoencoders

- Deterministic models trained using error backpropagation
- Input and Output are the same data: we force a network to be able to reconstruct such data with the limitation of having a 'bottleneck' (code) of limited size

The encoder provides a low dimensional representation of the input

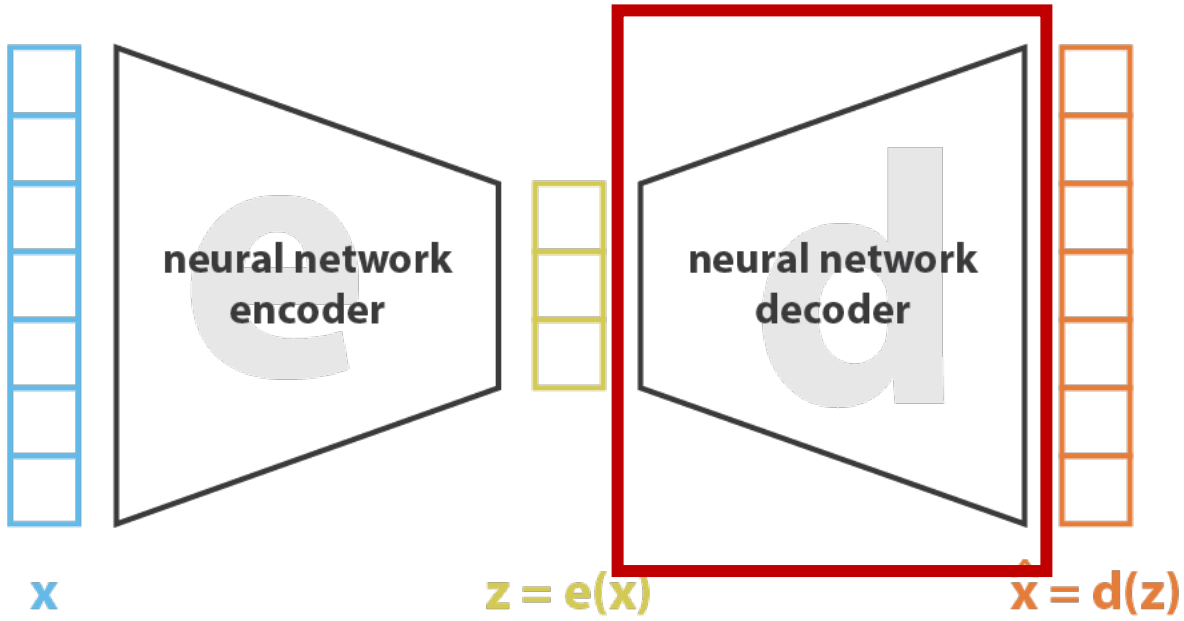


$$\text{loss} = \| x - \hat{x} \|^2 = \| x - d(z) \|^2 = \| x - d(e(x)) \|^2$$

Autoencoders

- Deterministic models trained using error backpropagation
- Input and Output are the same data: we force a network to be able to reconstruct such data with the limitation of having a ‘bottleneck’ (code) of limited size

The decoder reconstructs the input from its compressed representation



$$\text{loss} = \| x - \hat{x} \|^2 = \| x - d(z) \|^2 = \| x - d(e(x)) \|^2$$

Variational Autoencoder (VAE)

- In standard autoencoders, the latent space can be extremely irregular (close points in latent space can produce very different – often meaningless – patterns over visible units) so usually we cannot implement a generative process that simply samples a vector from the latent space and passes it through the decoder
- Possible fix: make the mapping probabilistic!

1. The encoder returns a distribution over the latent space instead of a single point
2. The loss function has an additional regularisation term in order to ensure a “better organization” of the latent space

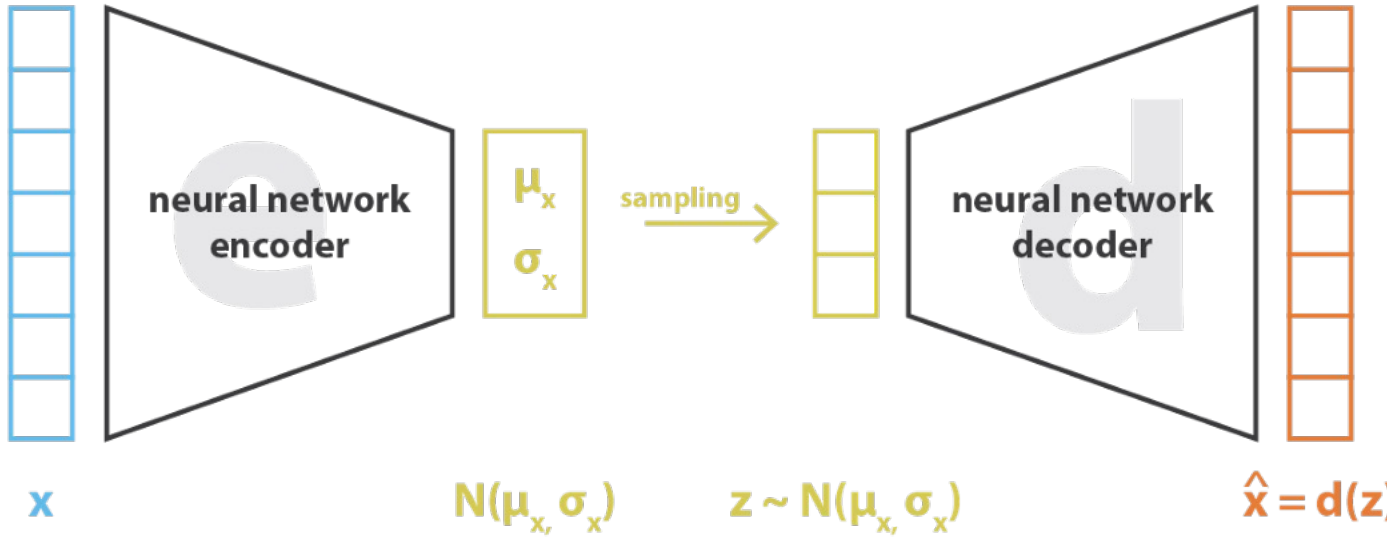
https://arxiv.org/abs/1312.6114

Variational Autoencoder (VAE)

- The encoded distribution is chosen to be a multivariate Gaussian, so that the encoder can be trained to estimate the means and covariance matrix
- This way we can regularize the loss function by forcing the latent distribution to be as close as possible to a standard Normal distribution

KL Divergence:

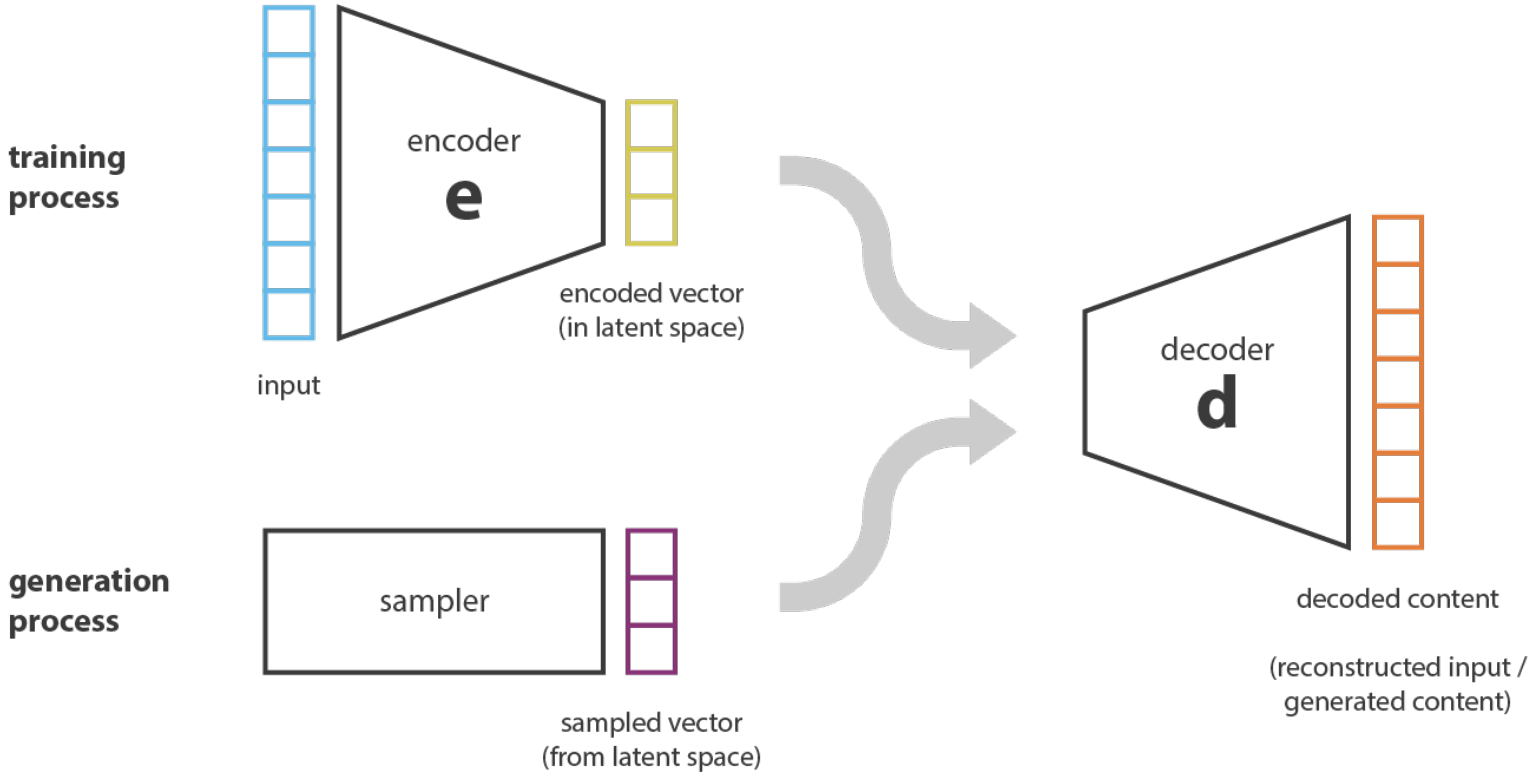
$$D_{\mathrm{KL}}(P \parallel Q) = \int_{x_a}^{x_b} P(x) \log\left(\frac{P(x)}{Q(x)}\right) dx$$



$$\text{loss} = \|x - \hat{x}\|^2 + KL[N(\mu_x, \sigma_x), N(0, I)] = \|x - d(z)\|^2 + KL[N(\mu_x, \sigma_x), N(0, I)]$$

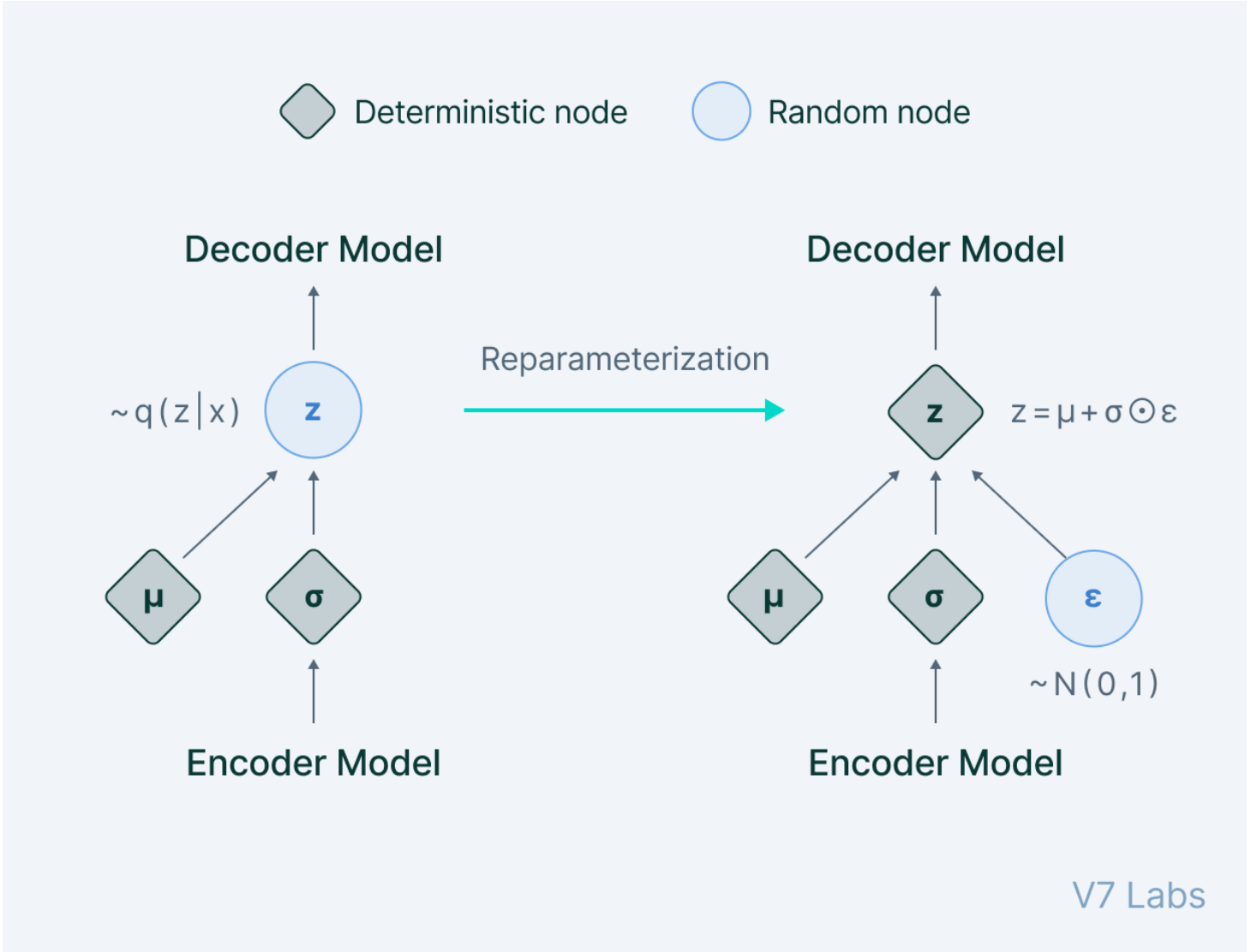
Variational Autoencoder (VAE)

- The encoded distribution is chosen to be a multivariate Gaussian, so that the encoder can be trained to estimate the means and covariance matrix
- This way we can regularize the loss function by forcing the latent distribution to be as close as possible to a standard Normal distribution



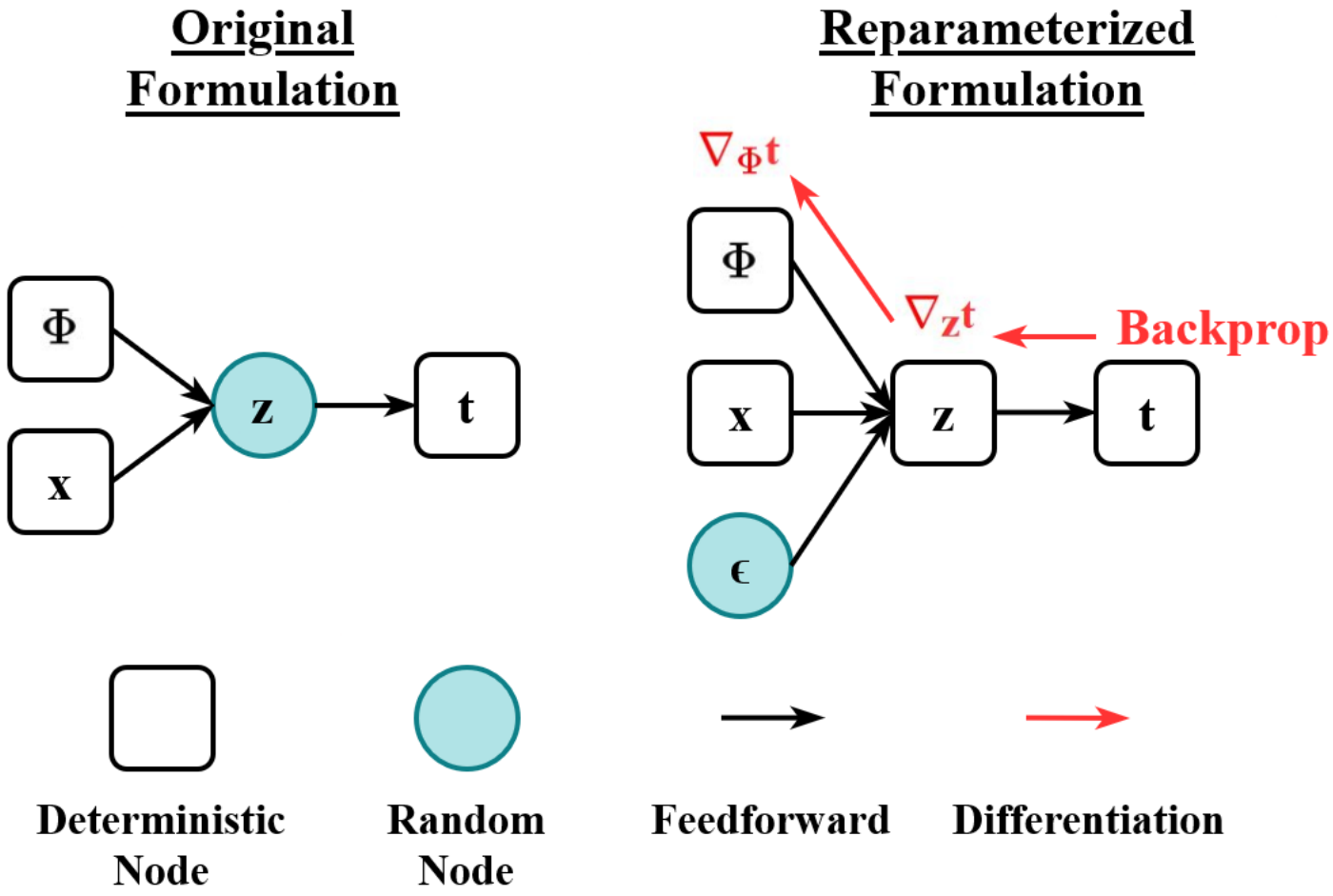
Reparametrization trick

- The latent representation is now defined by two vectors (means and covariance), so the encoder network has two (possibly partially overlapping) branches
- The covariance could just be a square matrix; however, to reduce computational complexity we assume that the multivariate Gaussian has a diagonal covariance matrix (i.e., latent variables are independent)
- Sampling is a discrete process, and we cannot use backpropagation! We need to re-parameterize z to make it differentiable



Reparametrization trick

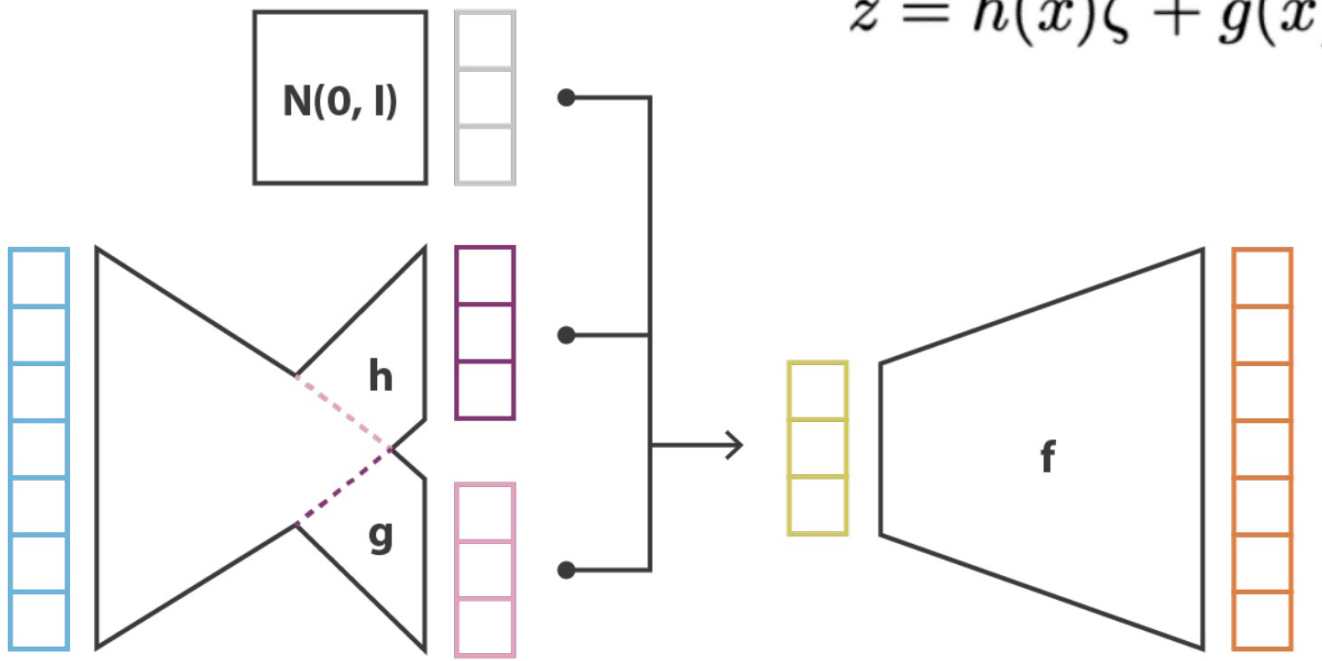
- The latent representation is now defined by two vectors (means and covariance), so the encoder network has two (possibly partially overlapping) branches
- The covariance could just be a square matrix; however, to reduce computational complexity we assume that the multivariate Gaussian has a diagonal covariance matrix (i.e., latent variables are independent)
- Sampling is a discrete process, and we cannot use backpropagation! We need to re-parameterize z to make it differentiable



A. Testolin 'Neural Networks and Deep Learning'

Reparametrization trick

$$z = h(x)\zeta + g(x) \qquad \zeta \sim \mathcal{N}(0, I)$$



$$\text{loss} = C \,||\, x - \hat{x} \,||^2 + KL[\, N(\mu_x, \sigma_x), N(0, I) \,] = C \,||\, x - f(z) \,||^2 + KL[\, N(g(x), h(x)), N(0, I) \,]$$

Variational Autoencoder (VAE)

- The regularization term indeed promotes the creation of a gradient over the latent representations, which allows to generate samples varying smoothly!

Disentangled VAE: β-VAE

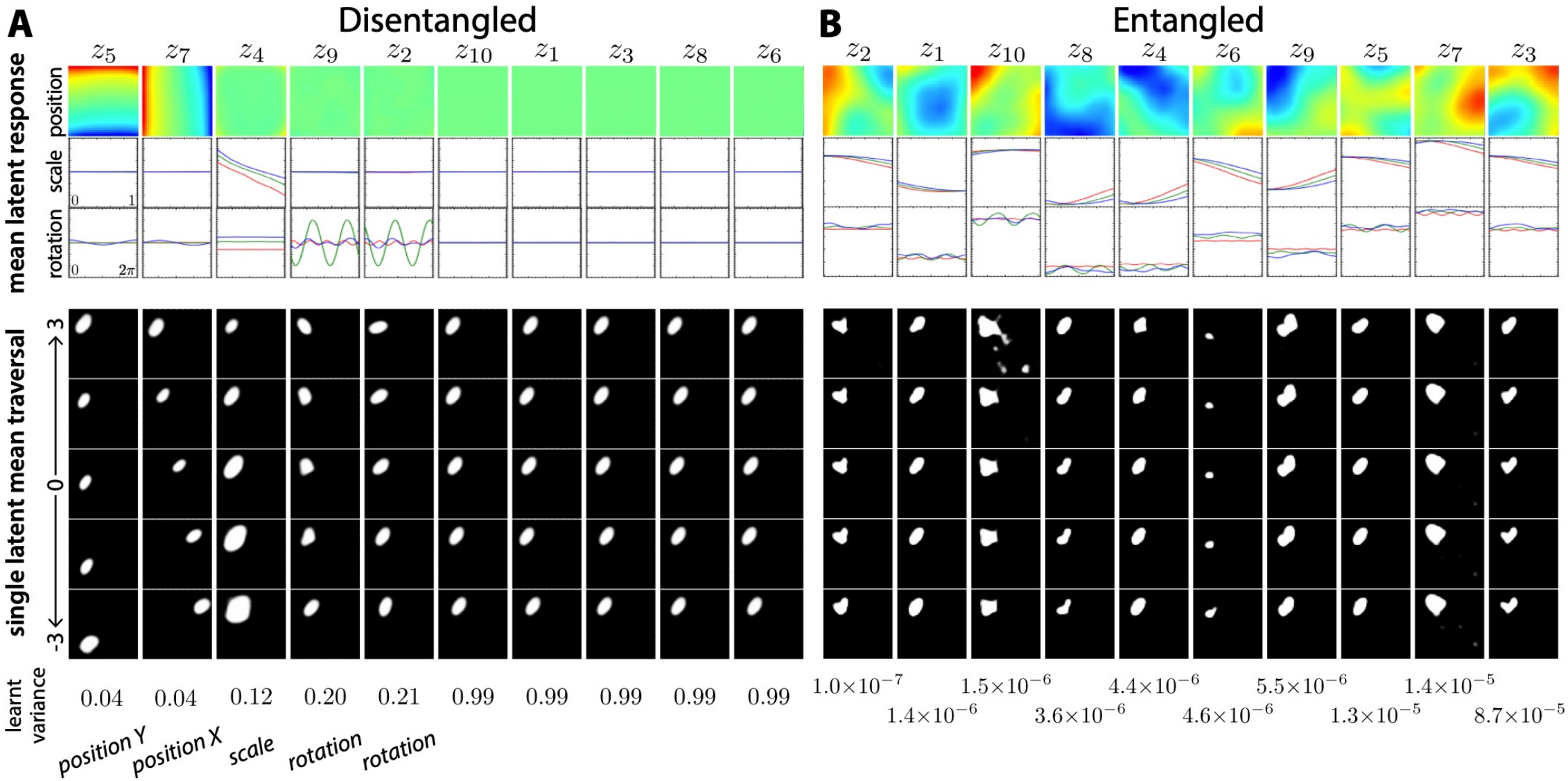
- VAE can be further extended to promote learning of more disentangled representations, which in some cases might encode independent latent factors of variation in the data distribution
- The final goal would be to have single latent units of z sensitive to changes in single generative factors (e.g., color of the hair) while being relatively invariant to changes in other factors (e.g., color of the skin)
- Basic idea: introduce a penalization term in the KL-divergence using a hyperparameter β > 1 that balances latent channel capacity and independence constraints with reconstruction accuracy (the higher the β, the more disentangled should be the representation)

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{x}^{(i)}) = -\beta D_{KL}\left(q_{\phi}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) \,||\, p(\boldsymbol{z})\right) + \mathbb{E}_{q_{\boldsymbol{\phi}}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})\left[log\, p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z})\right]$$

Disentangled VAE: β-VAE

https://arxiv.org/pdf/1606.05579.pdf

https://arxiv.org/pdf/1804.03599.pdf

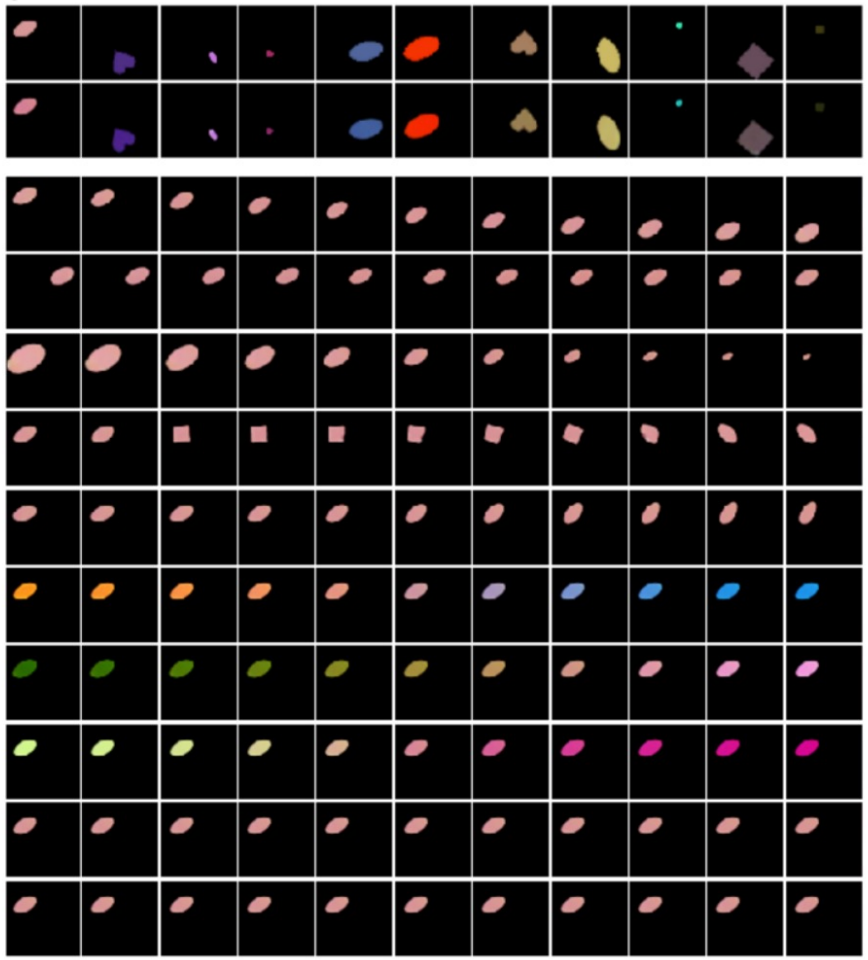


Disentangled VAE: β-VAE

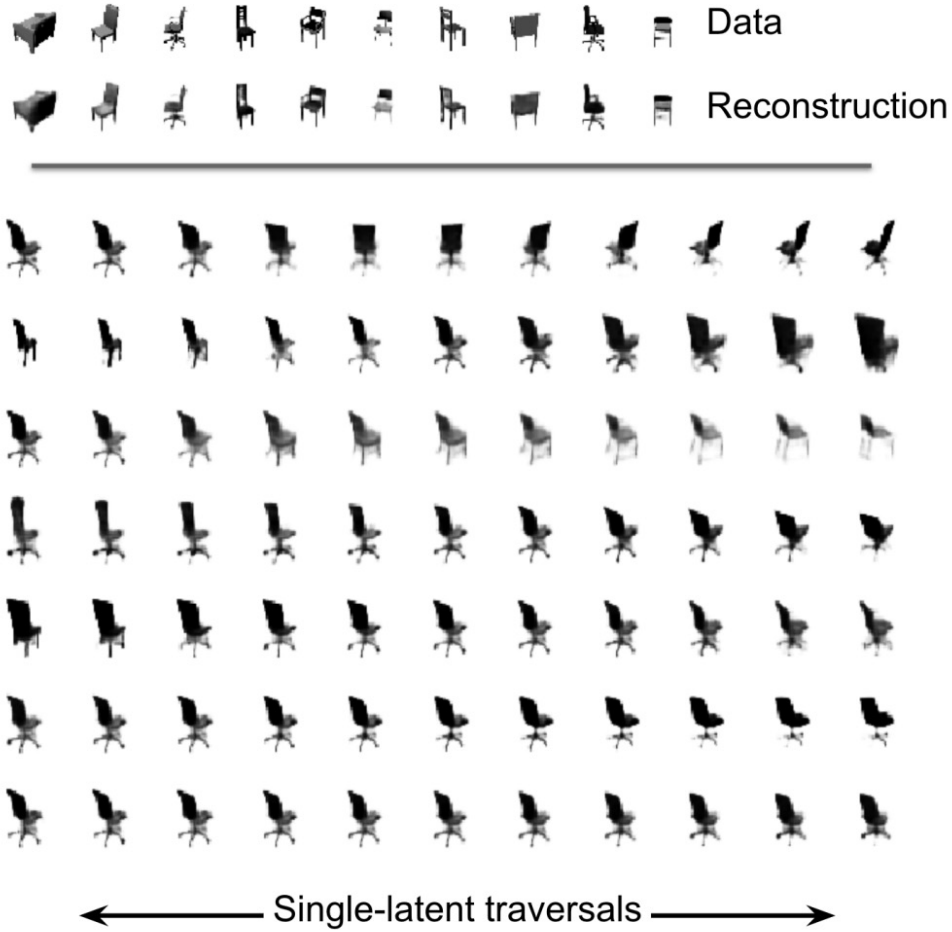
https://arxiv.org/pdf/1606.05579.pdf

https://arxiv.org/pdf/1804.03599.pdf

(a) Coloured dSprites



(b) 3D Chairs



Elements of Reinforcement Learning

Machine Learning by itself is not enough

- For example, ML algorithms may sense the environment in a self-driving car, but you need control and planning
- RL can be seen as the intersection of control and planning (more on this later)

Machine Learning without a supervisor

- ML solutions typically require a ‘supervisor’ that provides meaningful data of the phenomenon we want to characterize

Machine Learning without a supervisor

- ML solutions typically require a ‘supervisor’ that provides meaningful data of the phenomenon we want to characterize

The Matrix (1999)

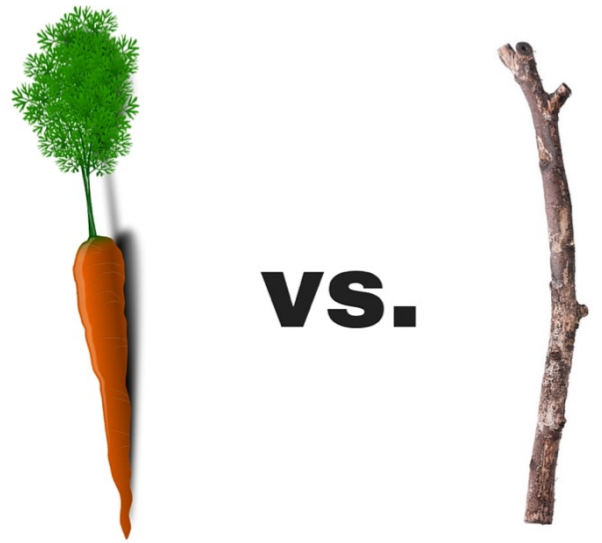
Machine Learning without a supervisor

- ML solutions typically require a ‘supervisor’ that provides meaningful data of the phenomenon we want to characterize

The Matrix (1999)

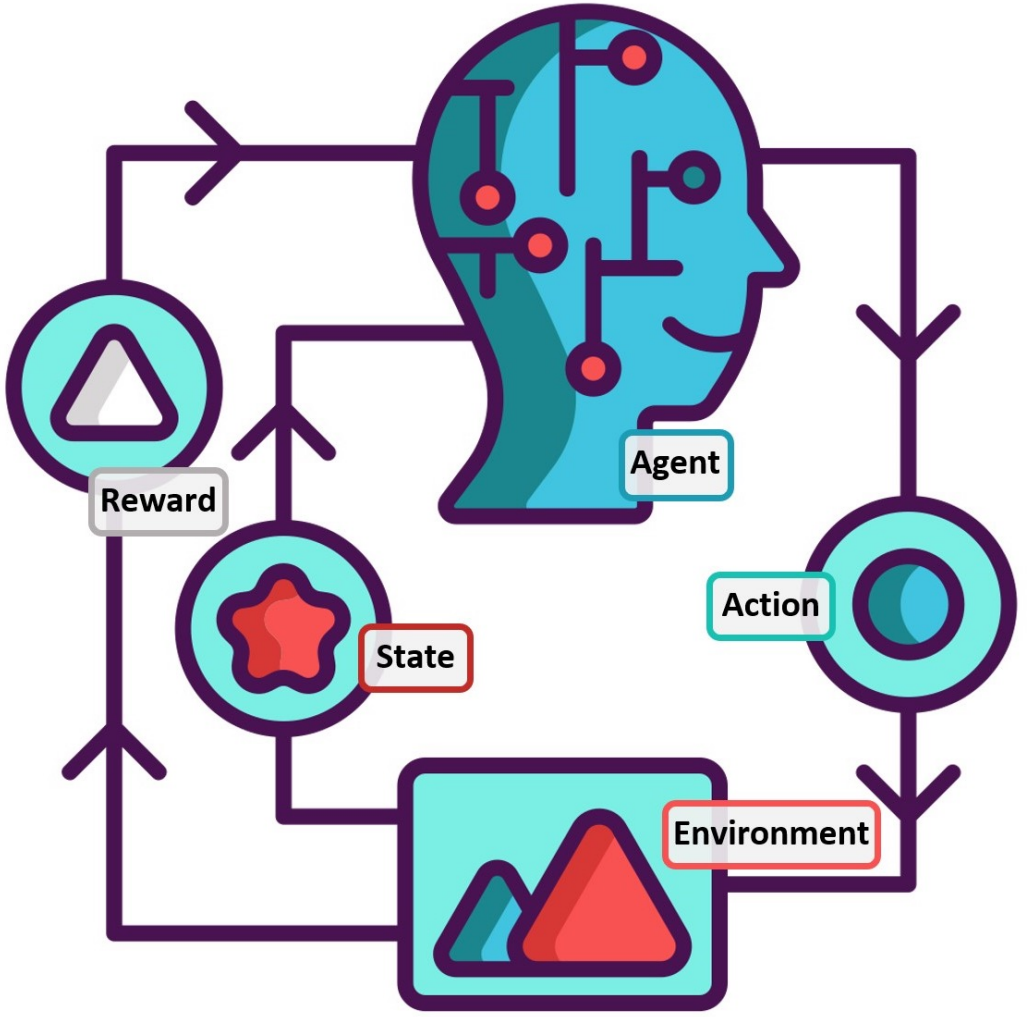
'Limits' of Supervised Learning

- In many cases, data are not available without interacting with the world
- Collected data may not be explicitly associated with actions: a lot of scenarios we get 'partial' information of how good a 'task' was executed, in the form of rewards (more on this later)
- In many cases, time matters: we don't always have independent identical distributed (i.i.d.) data (where rows can be swapped)
- In many settings, supervised learning do not envision long-term scenarios
- It's difficult to learn along the way (but we have continual learning...)



RL Formalism: The elements…

- An agent: the entity aiming at ‘solving a task’
- A set of states in which the agent can be
- A set of actions (that could depend on the state) that can be taken by the agent
- An environment with which the agent interacts and that could provide, at each time, rewards for state/state-action pair
- An agent could be more interested in long-term rewards, more than short-term…



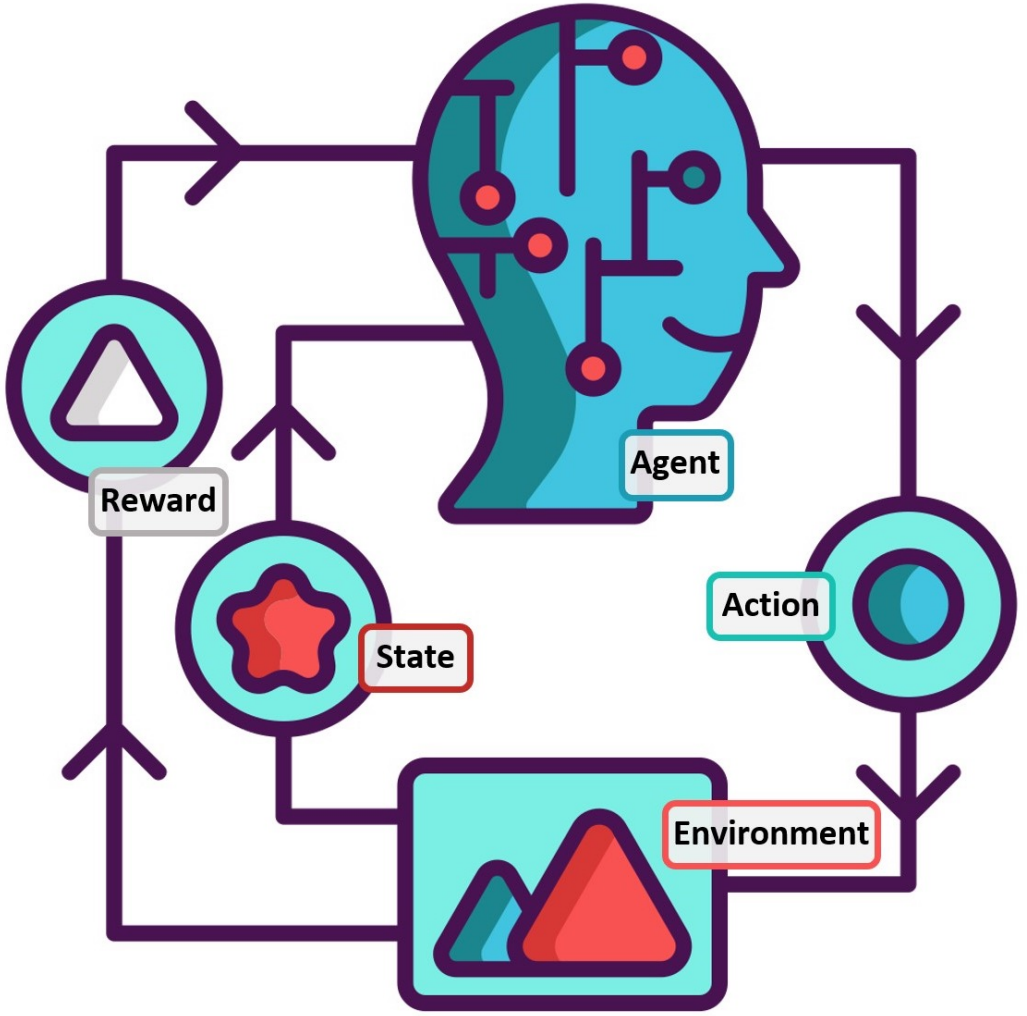
… in movies terms:

- In Groundhog Day, Phil (played by Bill Murray) enters a time-loop and revive the same day over and over again. During these ‘days’ he tries to make Rita (played by Andie MacDowell) to fall in love with him
- Phil is the agent: he plays many ‘episodes’ from which he can learn
- He tries different actions (be nice, be funny, …) from different states (at the restaurant, the park, …)
- The environment is composed by Lisa, the town, the people in the town…
- During his experience Phil collects data and learn how to maximize rewards (’love’ from Rita) from the environment

Groundhog Day (1993)

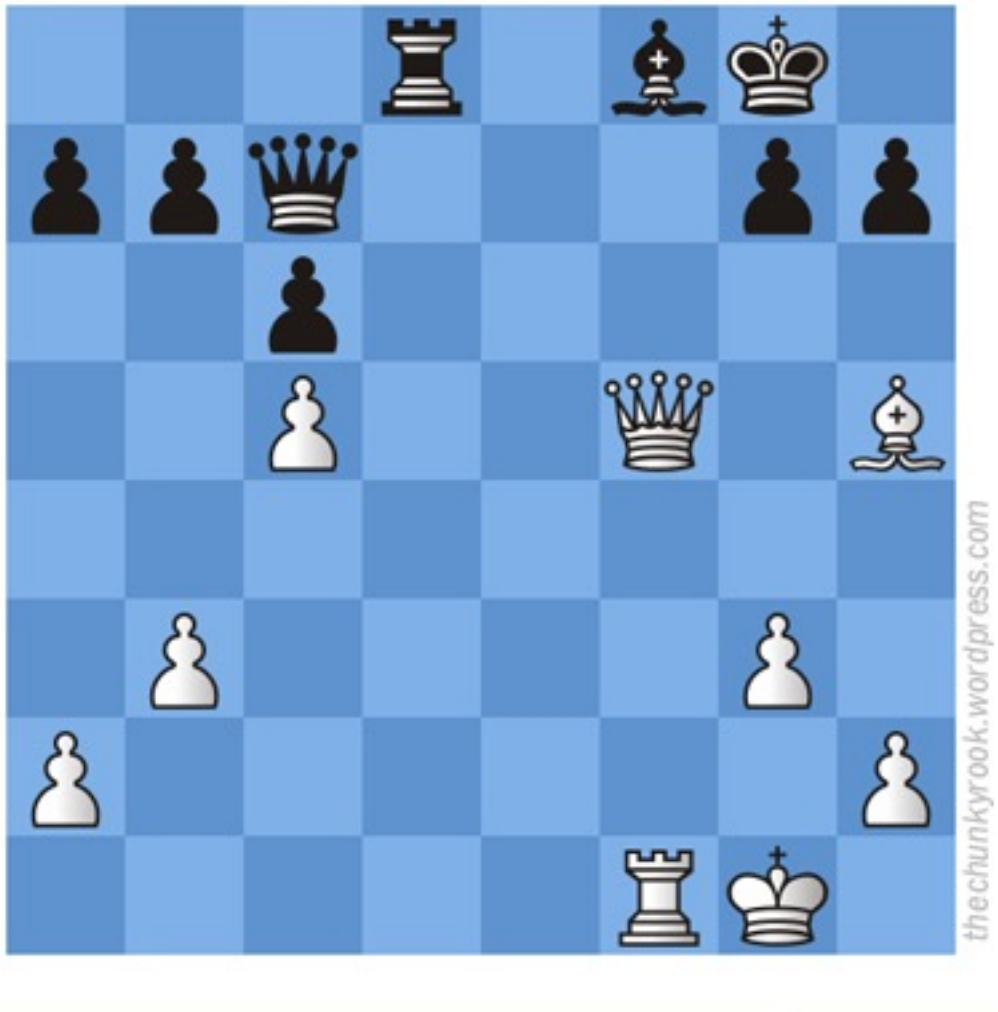
The RL Problem: data/information are gathered by Interacting with the Enviroment

- The agent could not know in advance a thing about the environment and which rewards and next state a given action from a state could lead to
- Such things will be learned by interacting with the environment!



The RL Problem: Planning and Long-term Rewards

- The goal of the agent is to maximize long term rewards, not necessarily at all steps
- Planning is fundamental in RL
- Ex. In chess the ‘true’ reward came when winning a game: at each steps of the game, event the losing of a piece can be ‘optimal’ in order to get a better state for reaching the target



What is Reinforcement Learning (RL)?

Reinforcement Learning (RL) is both:

- a research area (sub-field of Machine Learning -> more in the following)
- a learning problem/paradigm concerned with learning to control a system (with many unknown elements) so as to maximize a numerical performance measure

Many real-world problems are better formalized in the RL fashion instead of the classical control or supervised learning formalization...

Some problems that can be formalized in a RL fashion

- Autonomous agents (self-driving cars, drones, robots...)
- Games
- HVAC (Heating, Ventilating, Air Conditioning) energy optimization
- Trading and Portfolio management
- Online advertising & Recommendation systems (news, items, ...)
- Healthcare, biology

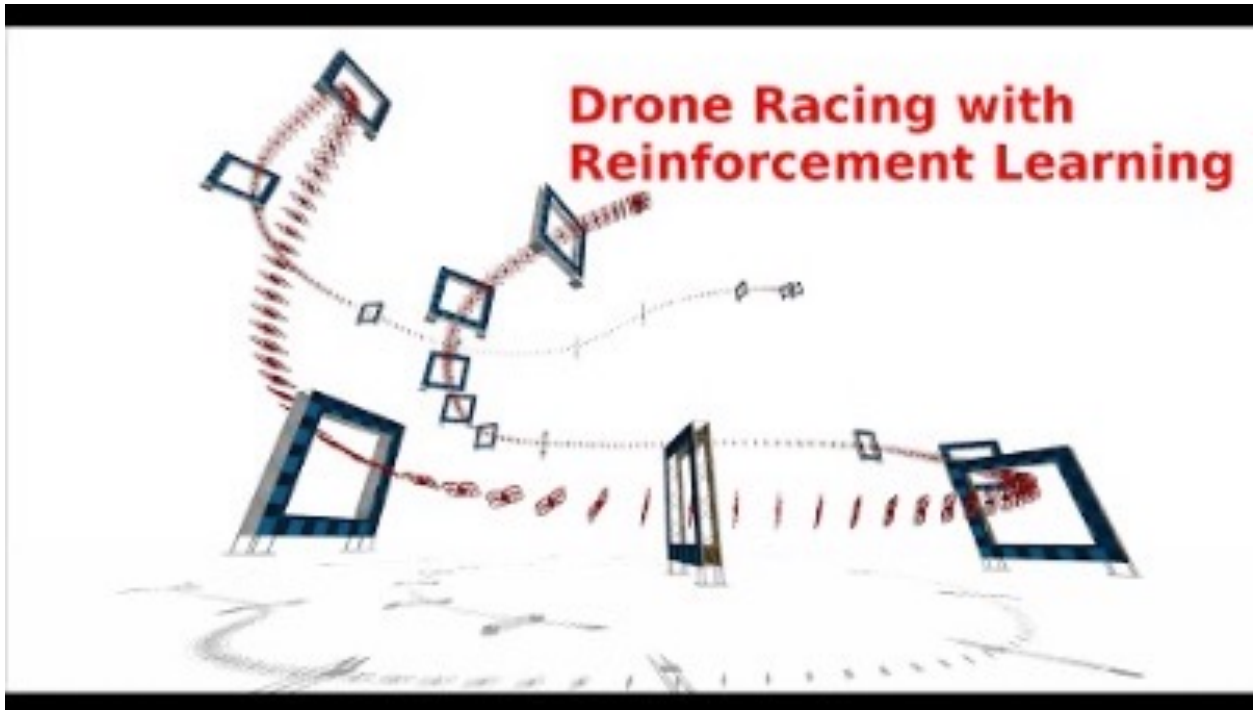
....

Aerospace Control Labs @ MIT (2015)
https://www.youtube.com/watch?v=opsmd5yuBF0

Some problems that can be formalized in a RL fashion

- Autonomous agents (self-driving cars, drones, robots…)
- Games
- HVAC (Heating, Ventilating, Air Conditioning) energy optimization
- Trading and Portfolio management
- Online advertising & Recommendation systems (news, items, …)
- Healthcare, biology

….



UZH Robotics and Perception Group @ ETH (2021)
https://www.youtube.com/watch?v=Hebpmadjqn8

Some problems that can be formalized in a RL fashion

- Autonomous agents (self-driving cars, drones, robots...)
- Games
- HVAC (Heating, Ventilating, Air Conditioning) energy optimization
- Trading and Portfolio management
- Online advertising & Recommendation systems (news, items, ...)
- Healthcare, biology

....

Hybrid Robotics @ Berkeley (2021)
https://www.youtube.com/watch?v=6tn-owW6iME
https://arxiv.org/abs/2103.14295

Some problems that can be formalized in a RL fashion

- Autonomous agents (self-driving cars, drones, robots...)
- Games
- HVAC (Heating, Ventilating, Air Conditioning) energy optimization
- Trading and Portfolio management
- Online advertising & Recommendation systems (news, items, ...)
- Healthcare, biology

....

Some problems that can be formalized in a RL fashion

- Autonomous agents (self-driving cars, drones, robots...)
- Games
- HVAC (Heating, Ventilating, Air Conditioning) energy optimization
- Trading and Portfolio management
- Online advertising & Recommendation systems (news, items, ...)
- Healthcare, biology

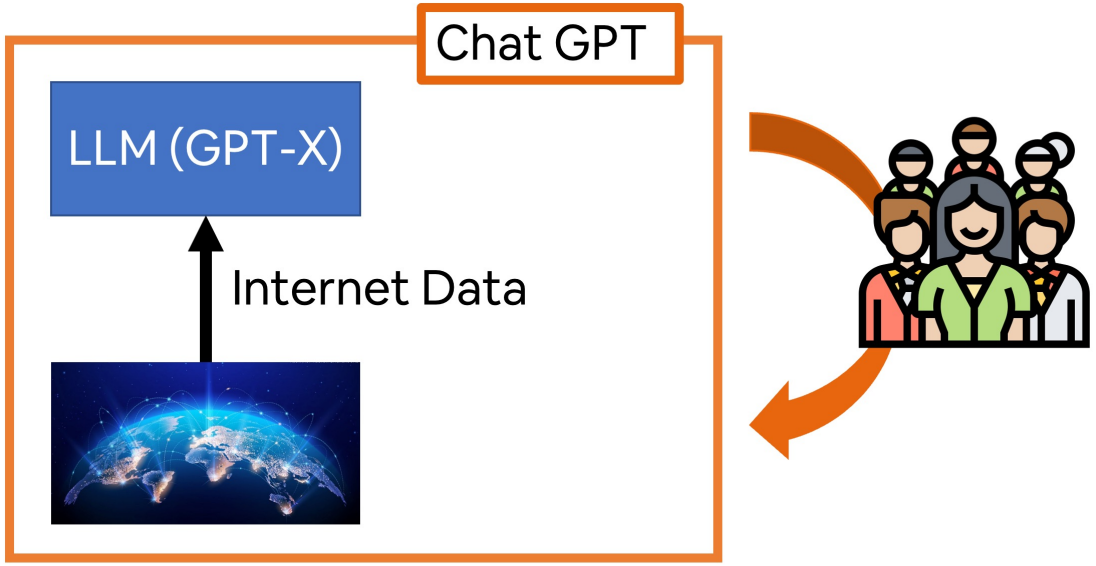
....

Some problems that can be formalized in a RL fashion

- Autonomous agents (self-driving cars, drones, robots...)
- Games
- HVAC (Heating, Ventilating, Air Conditioning) energy optimization
- Trading and Portfolio management
- Online advertising & Recommendation systems (news, items, ...)
- Healthcare, biology
- Chatbots

....

Reinforcement Learning from human Feedback (RLHF)

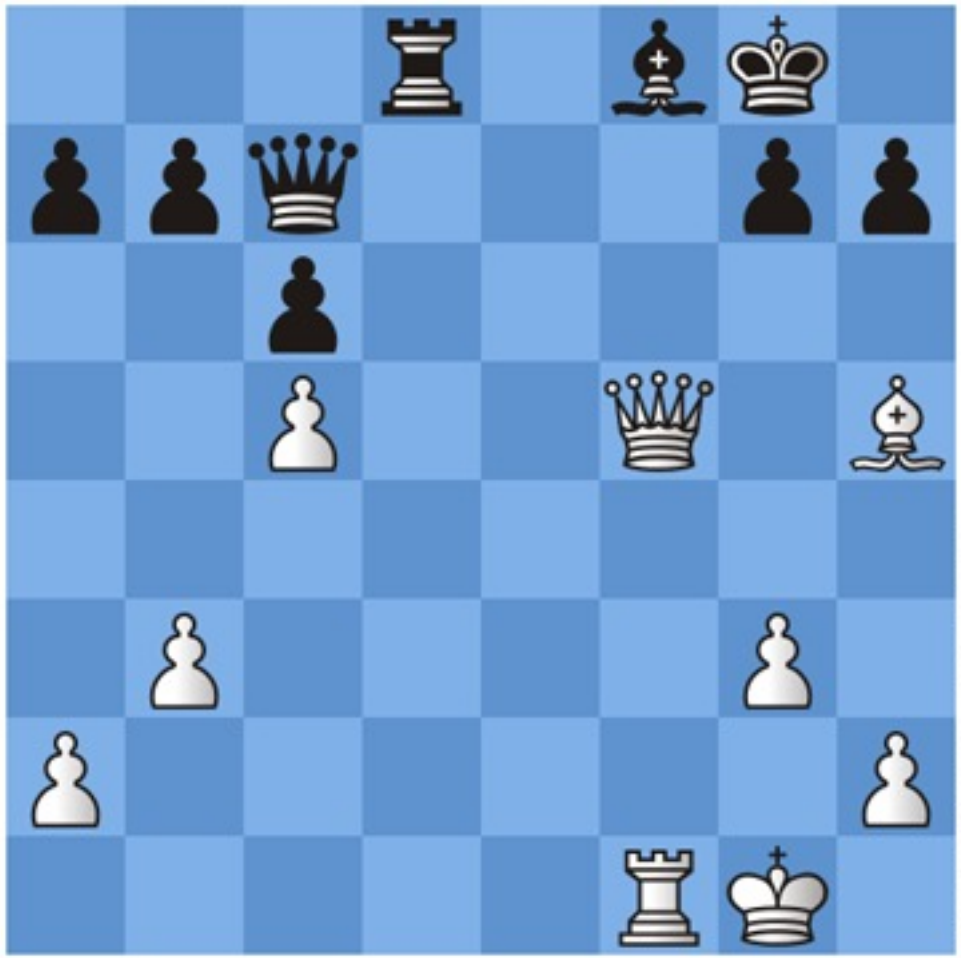


Rewards: examples

1. Autonomous agents (self-driving cars, drones, robots...)
   - +/- for/for not following desired trajectory
   - - for each time step taken for reaching a target position
   - ------ for crashing
2. Games
   - +/- win/lose
   - +/- A reward proportional to the score achieved
3. HVAC (Heating, Ventilating, Air Conditioning) energy optimization
   - - for the energy spent and/or for the user discomfort
4. Trading and Portfolio management
   - +/- proportional to € gained/lost
5. Online advertising & Recommendation systems (news, items, ...)
   - +/- for ad/recommendation followed/ignored

Rewards: sequential decision making

- Agent's goal: select actions to maximize cumulative rewards/total future rewards
- Actions may have long-term consequences & rewards may be delayed
- Immediate rewards can be sacrificed to gain more long-term rewards: this is relevant both in the planning and in the 'training'



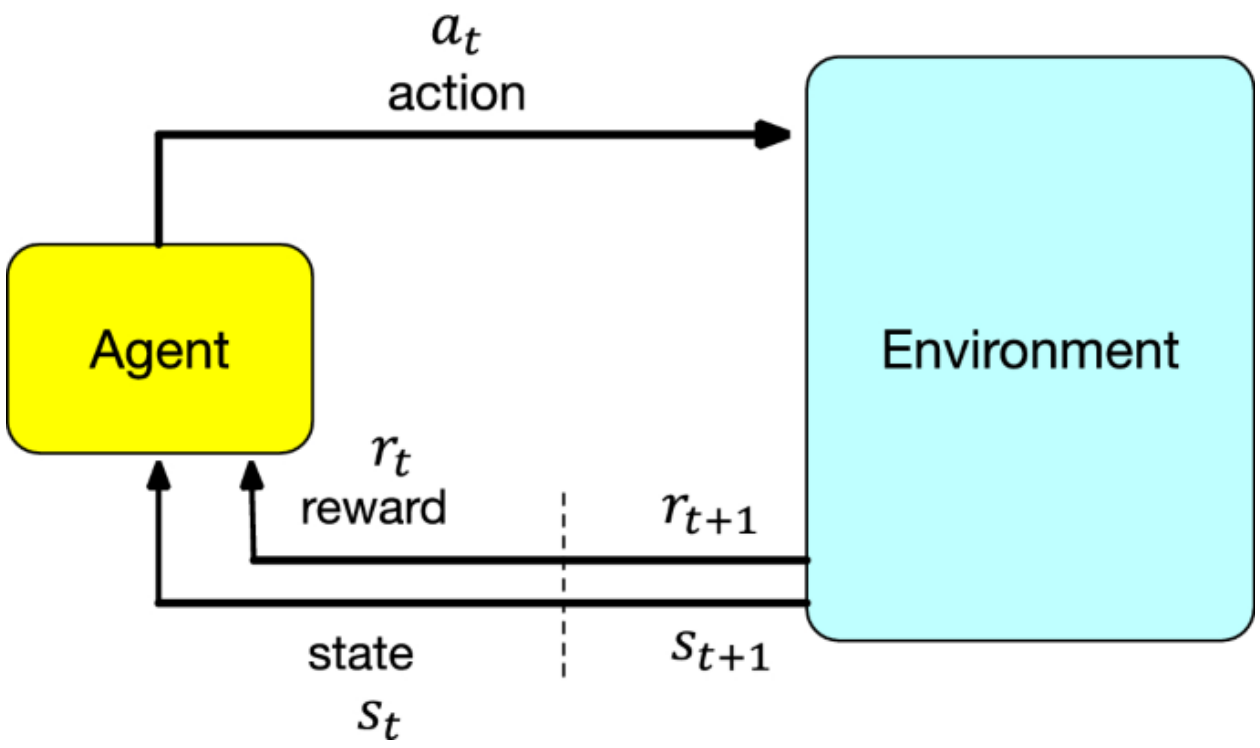
The Agent and the Environment

At each step t the agent (something that we can program/control):

- Is in the state S_t, an exhaustive description of the system (agent and environment) at time t
- Execute action A_t (that is a feasible action from state S_t)

The environment, based of the state/action pair (S_t, A_t):

- provides a reward R_{t+1}
- 'move' the agent in state S_{t+1}



Maximizing the accumulated rewards

Inspired by Alberto Testolin

- The sum of accumulated rewards is usually called the *return*

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

- Finite-horizon case: we aim at summing rewards up to a certain point
- Infinite-horizon case: need to introduce a discount rate, because we aim at maximizing rewards for the entire agent life. The agent tries to maximize the discounted long-term expected reward:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- The presence of delayed rewards makes the task more complex: it is hard to link the actual reward with the past actions that led to it (credit assignment problem)

Maximizing the accumulated rewards

Inspired by Alberto Testolin

- The sum of accumulated rewards is usually called the *return*

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

- Finite-horizon case: we aim a
certain point

- Infinite-horizon case: need to
because we aim at maximizing rewards for the entire agent life. The agent tries to maximize the discounted long-term expected reward:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

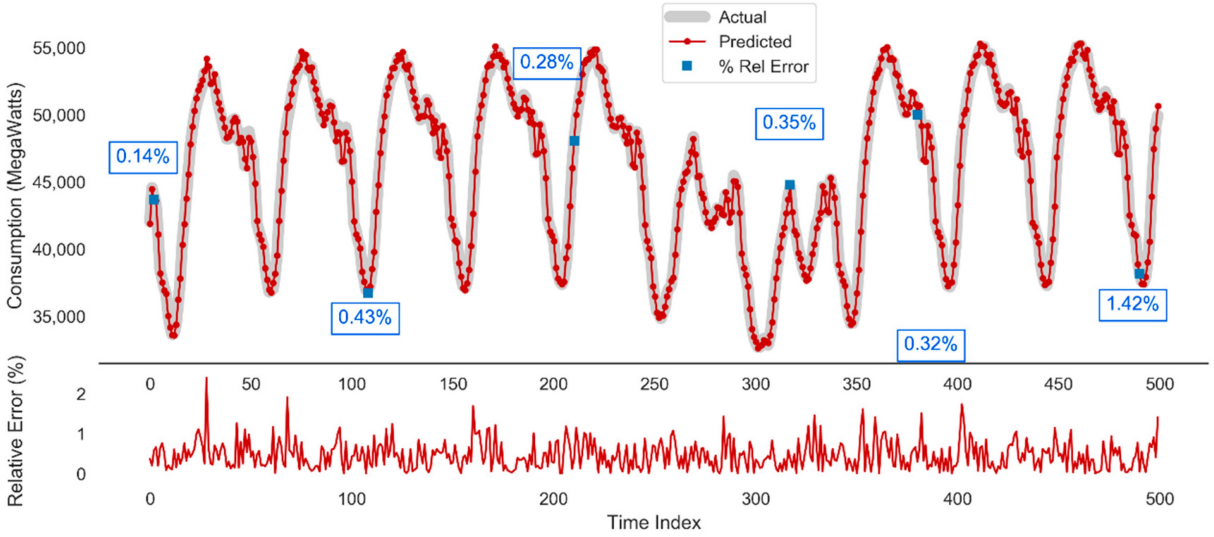
While typically ML is about minimizing a loss, RL is about maximizing the cumulative reward!

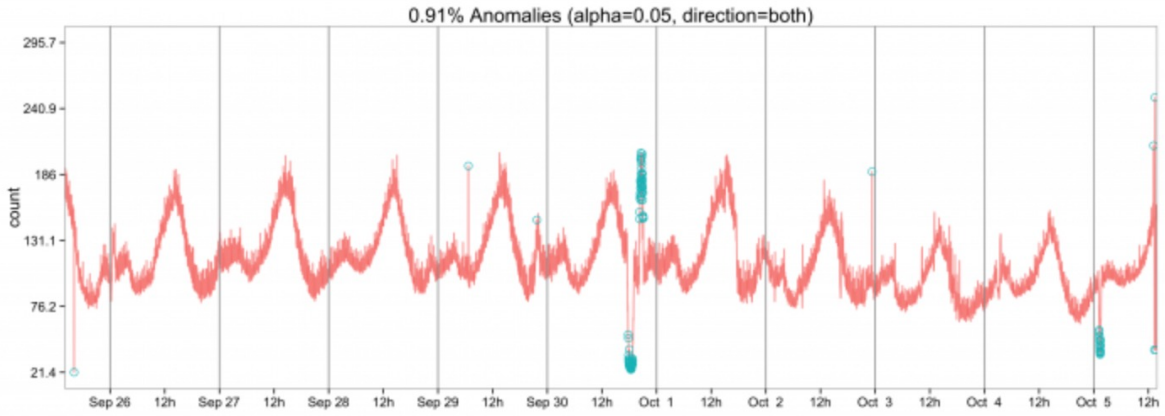
$$\pi^* = \operatorname{argmax} G$$

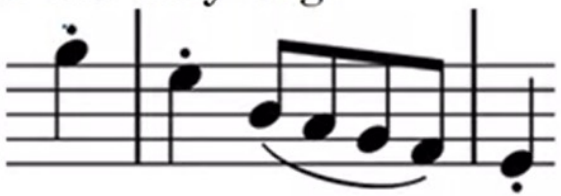
- The presence of delayed rewards makes the task more complex: it is hard to link the actual reward with the past actions that led to it (credit assignment problem)

Elements of Sequence Learning

Learning sequences matters!





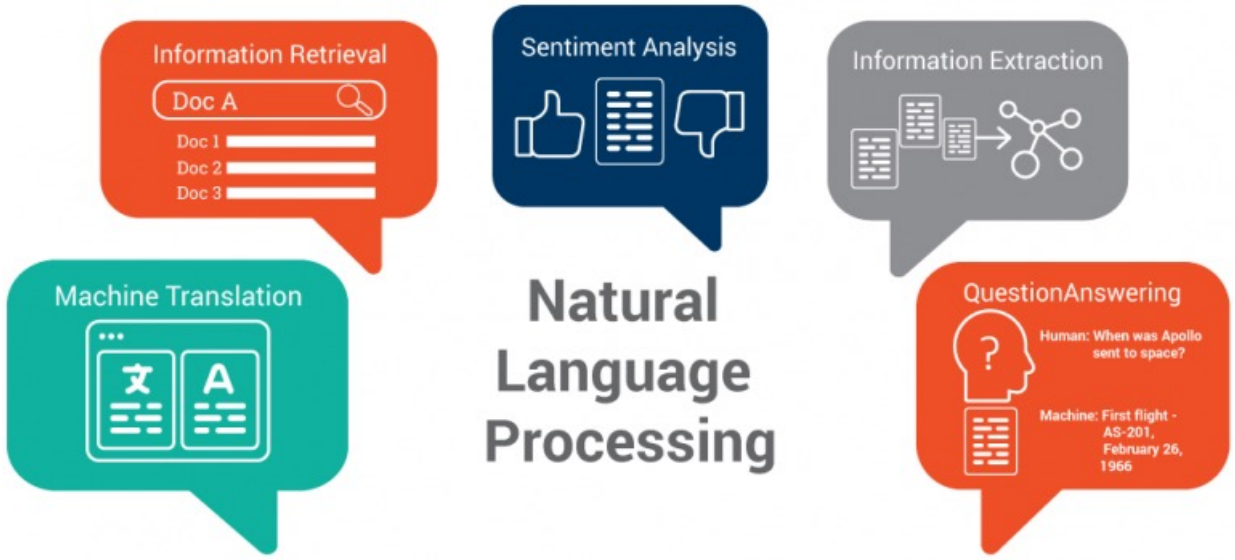
Speech recognition		"The quick brown fox jumped over the lazy dog."
Music generation	∅	
Sentiment classification	"There is nothing to like in this movie."	★☆☆☆☆
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	AGCCCCTGTGAGGAACTAG
Machine translation	Voulez-vous chanter avec moi?	Do you want to sing with me?
Video activity recognition		Running
Name entity recognition	Yesterday, Harry Potter met Hermione Granger.	Yesterday, Harry Potter met Hermione Granger.

A. Ng 'Sequential Learning' https://www.coursera.org/specializations/deep-learning

Sequence learning requirements

To model sequences, we need to:
1. Handle variable-length sequences
2. Track long-term dependencies
3. Maintain information about order
4. Share parameters across the sequence

Deep Learning for sequence modeling has been disruptive in NLP!



From MIT *Introduction to Deep Learning* http://introtodeeplearning.com

An example: NLP

To model sequences, we need to:

1. Handle variable-length sequences
2. Track long-term dependencies
3. Maintain information about order
4. Share parameters across the sequence

Q: “Do you like cats?”

A1: “I love cats!”

VS

A2: “Dogs or cats? even if the dogs are super cute and very friendly, a kitten that purrs is unmatchable…”

An example: NLP

To model sequences, we need to:

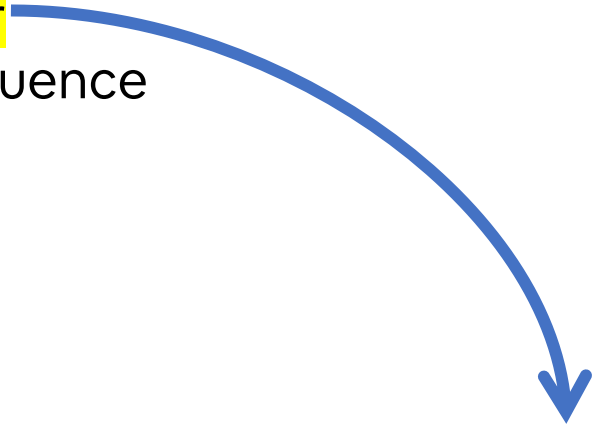
1. Handle variable-length sequences
2. Track long-term dependencies
3. Maintain information about order
4. Share parameters across the sequence

“Lisboa is were I grew up, but now I live in Sweden. I speak fluent _____”

An example: NLP

To model sequences, we need to:

1. Handle variable-length sequences
2. Track long-term dependencies
3. Maintain information about order
4. Share parameters across the sequence



“The food was good, not bad at all.”

VS

“The food was bad, not good at all.”

An example: NLP

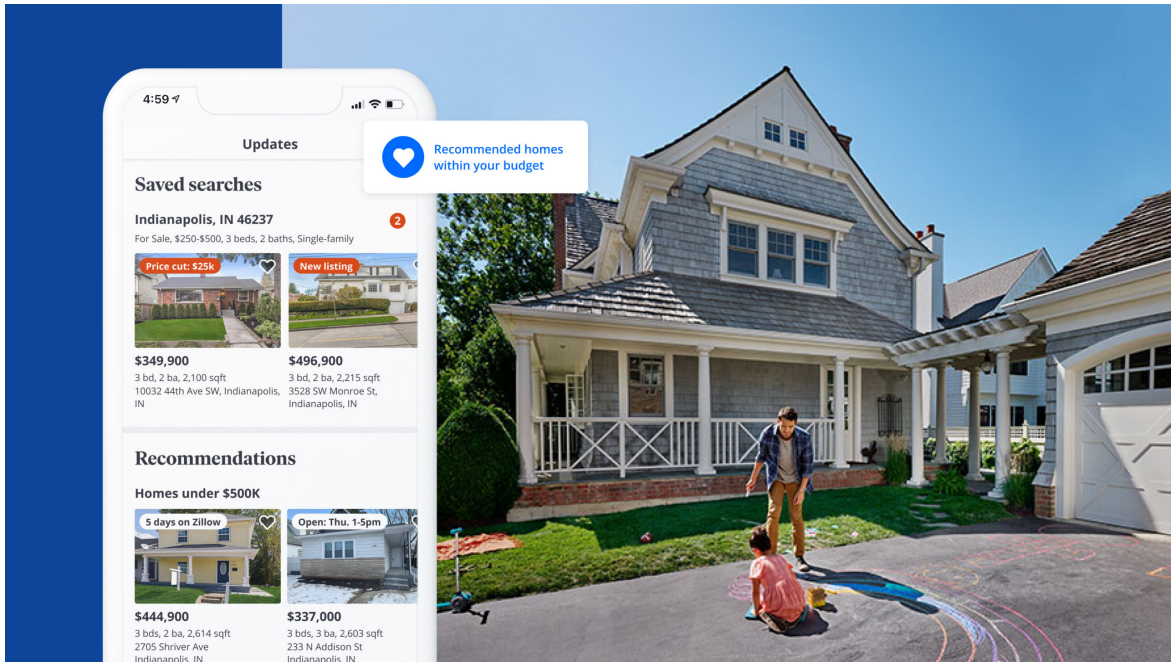
To model sequences, we need to:

1. Handle variable-length sequences
2. Track long-term dependencies
3. Maintain information about order
4. Share parameters across the sequence

Once the models learns the meaning of a word, it can recognize it in subsequent occurrences.

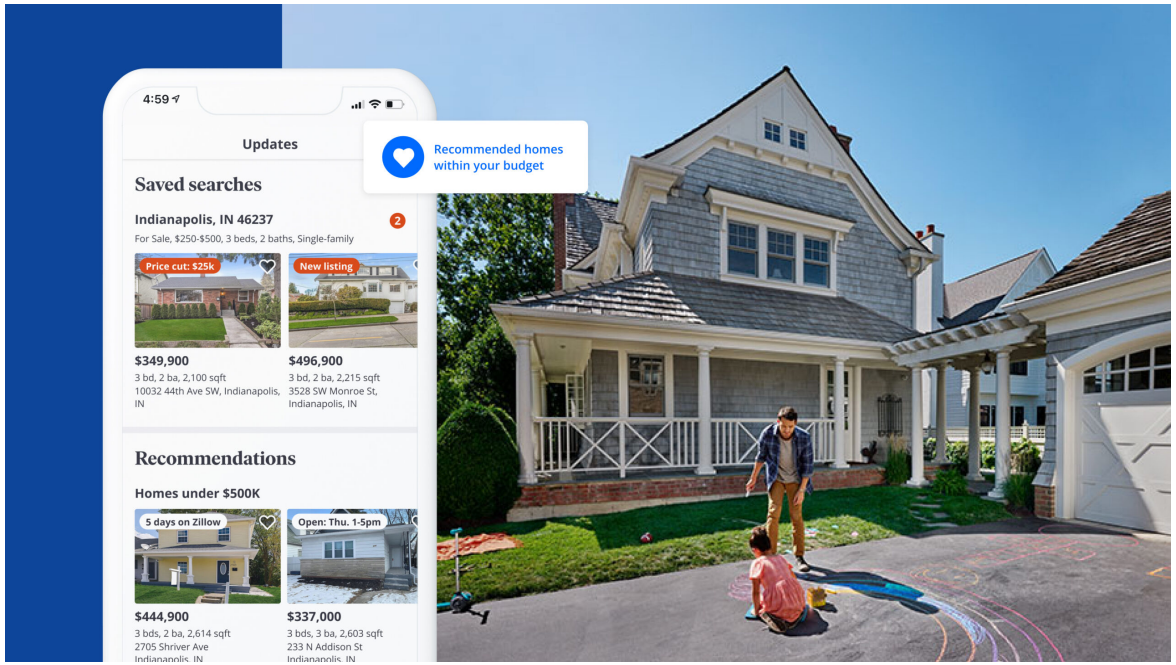
Dealing with time series is hard

Zillow



Dealing with time series is hard

Zillow



THE INDUSTRY

Zillow Torched $381 Million Overpaying for Houses. Spectacular.

BY ALEX KIRSHNER

NOV 03, 2021 • 1:14 PM

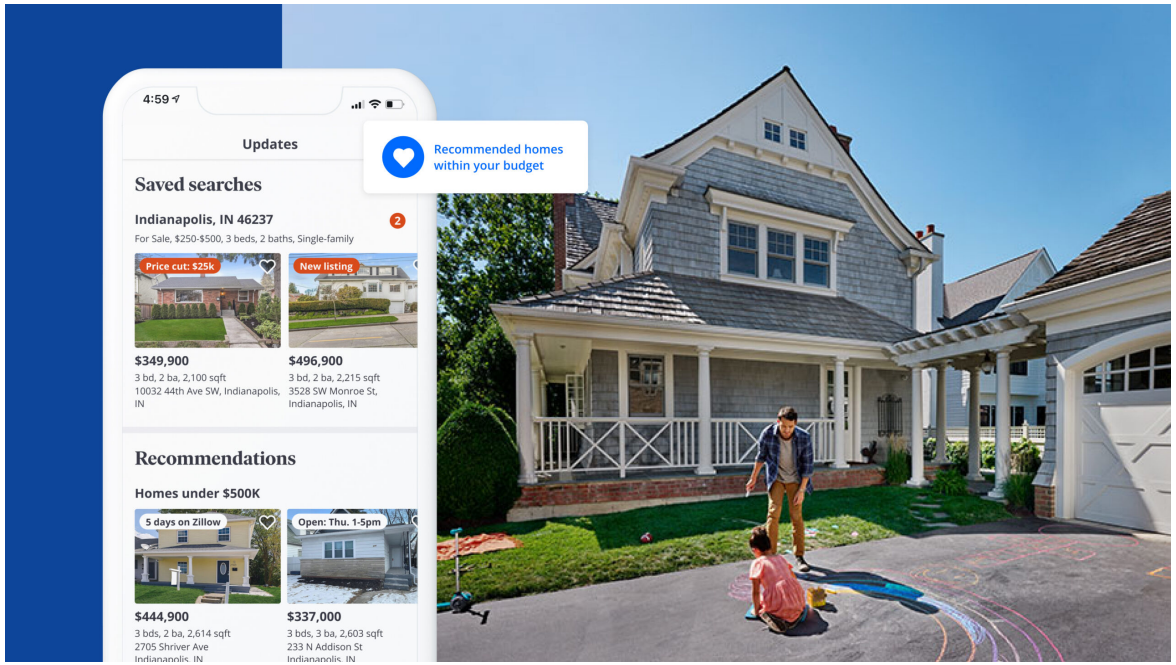
Zillow AI Goes Crazy. Causes $8 Billion Drop in Market Cap, a $304 Million Operating Loss, and 2,000+ Jobs

By **John Mecke**

JAN 9, 2022 Facebook Prophet & Zillow AI, Zillow AI DIsaster

Dealing with time series is hard

Zillow



THE INDUSTRY

Zillow Torched $381 Million Overpaying for Houses. Spectacular.

BY ALEX KIRSHNER

NOV 03, 2021 • 1:14 PM

Zillow AI Goes Crazy. Causes $8 Billion Drop in Market Cap, a $304 Million Operating Loss, and 2,000+ Jobs

By **John Mecke**

JAN 9, 2022 Facebook Prophet & Zillow AI, Zillow AI DIsaster

PROPHET

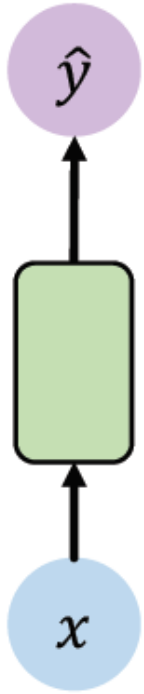
Docs GitHub

Forecasting at scale.

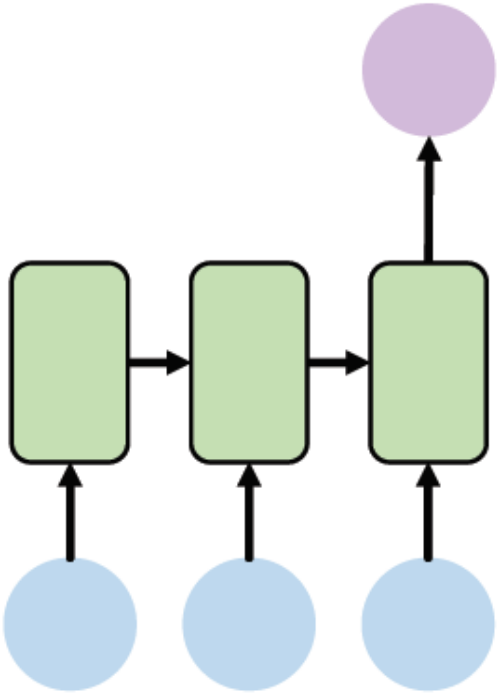
Prophet is a forecasting procedure implemented in R and Python. It is fast and provides completely automated forecasts that can be tuned by hand by data scientists and analysts.

INSTALL PROPHET | GET STARTED IN R | GET STARTED IN PYTHON | READ THE PAPER

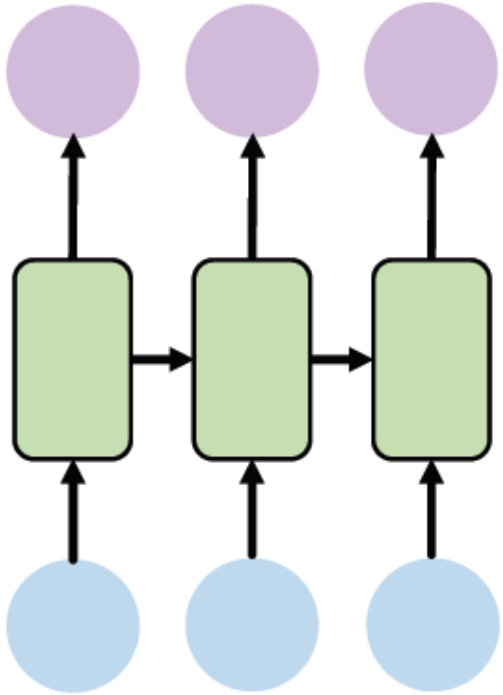
Recurrent Neural Networks (RNNs)



One to One
"Vanilla" neural network



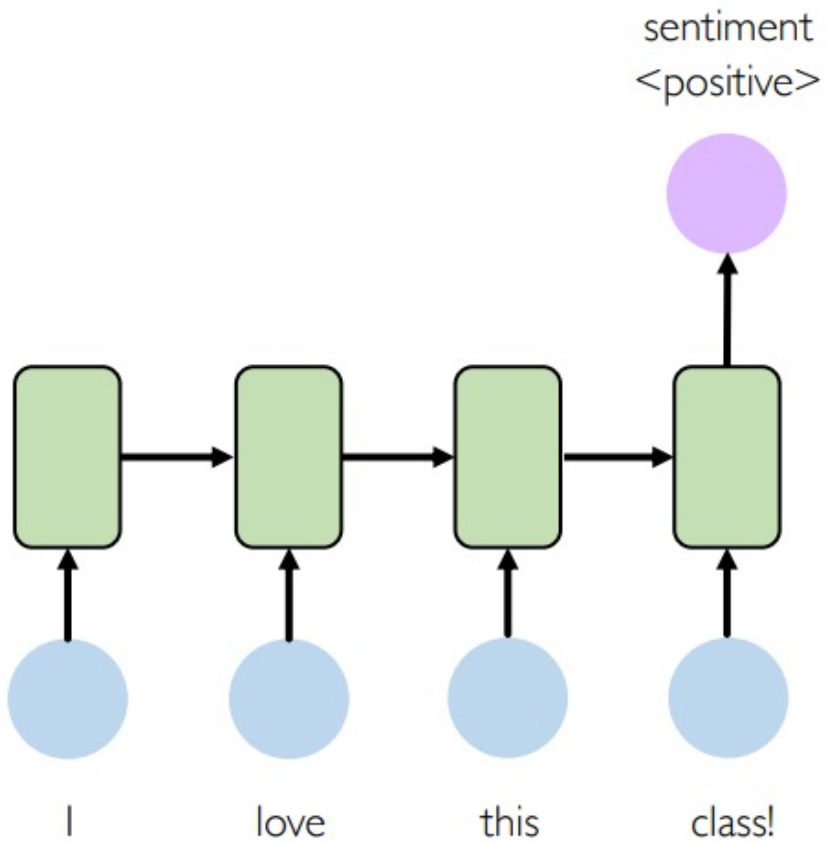
Many to One
Sentiment Classification



Many to Many
Named Entity Recognition

From MIT *Introduction to Deep Learning* http://introtodeeplearning.com

Sequence learning in action: sentiment classification



From MIT *Introduction to Deep Learning* http://introtodeeplearning.com

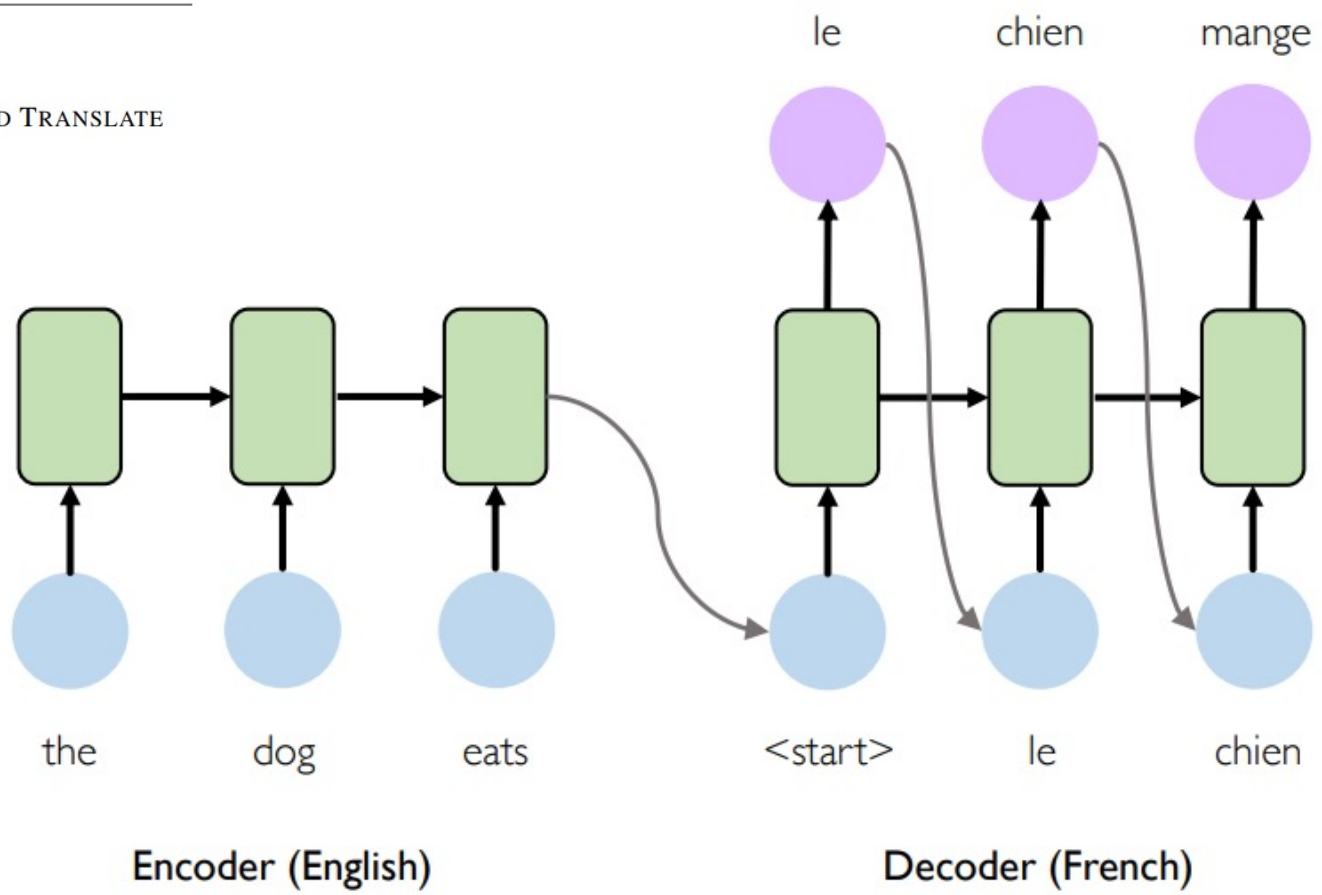
Sequence learning in action: machine translation

Published as a conference paper at ICLR 2015

NEURAL MACHINE TRANSLATION
BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho **Yoshua Bengio***
Université de Montréal



Bahdanau, Dzmitry et al. “Neural Machine Translation by Jointly Learning to Align and Translate.” CoRR abs/1409.0473 (2015):

From MIT *Introduction to Deep Learning* http://introtodeeplearning.com

Dealing with sequences: motivating NLP example(s) and Notation

The cat is on the table

$x^{<1>}$ $x^{<2>}$ $\ldots$ $x^{<6>}$

$y^{<1>}$ $y^{<2>}$ $\ldots$ $y^{<6>}$

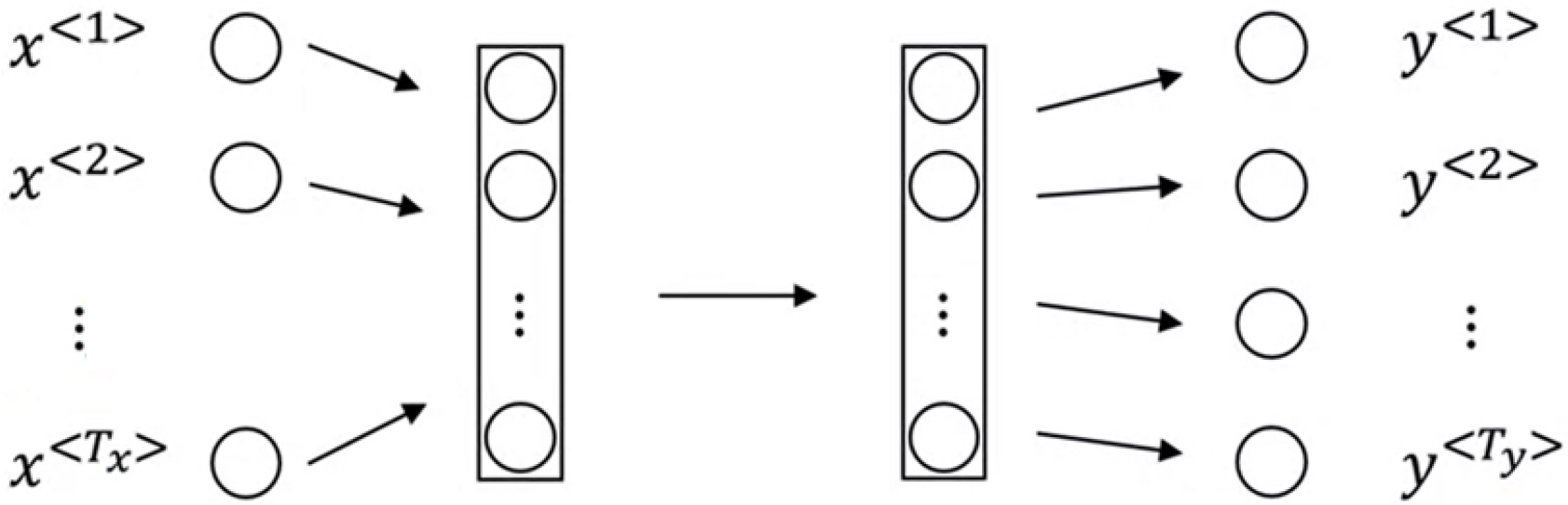
Named entity recognition associated an output to each input (word): it is a classifcation task that associated 1 if the word is a name and 0 otherwise

Notation

$x^{(i)<t>}$ $T_x^{(i)}$

$y^{(i)<t>}$ $T_y^{(i)}$

Why not using a FFNN?

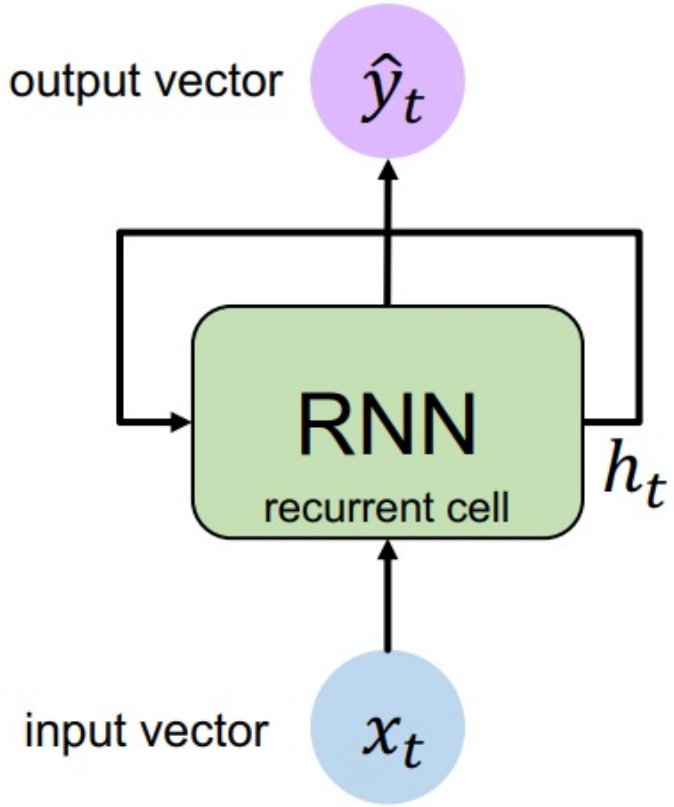


A. Ng 'Sequential Learning' https://www.coursera.org/specializations/deep-learning

Issues:

- Input and outputs can have different lengths for different observations (sequences)
- There is not sharing of the features learned along the network!

Recurrent Neural Networks (RNNs)



Apply a **recurrence relation** at every time step to process a sequence:

$$h_t = f_W(h_{t-1}, x_t)$$

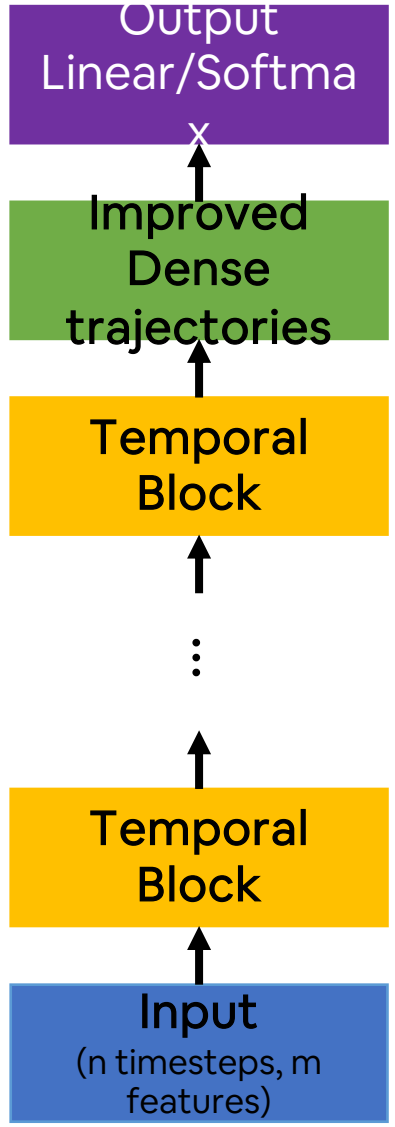
new state — function parameterized by W — old state — input vector at time step t

Note: the same function and set of parameters are used at every time step

From MIT *Introduction to Deep Learning* http://introtodeeplearning.com

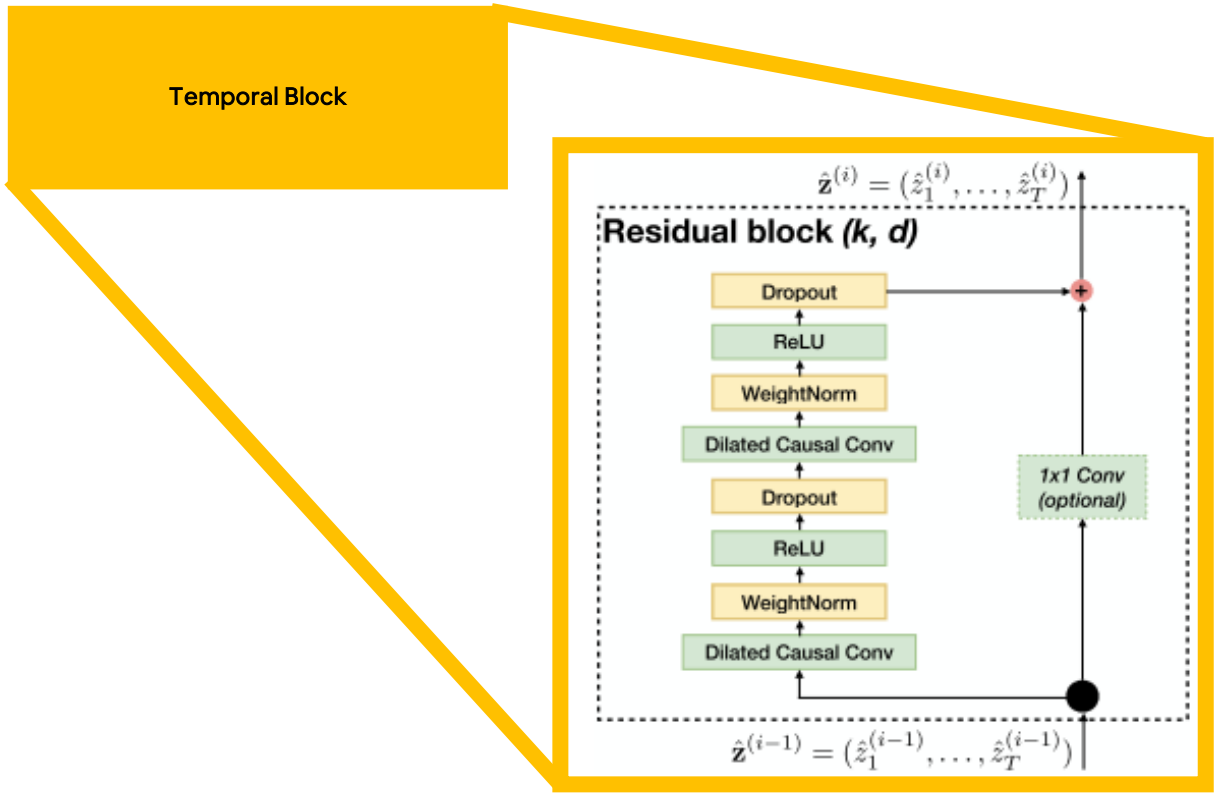
Temporal Convolutional Networks (TCN)

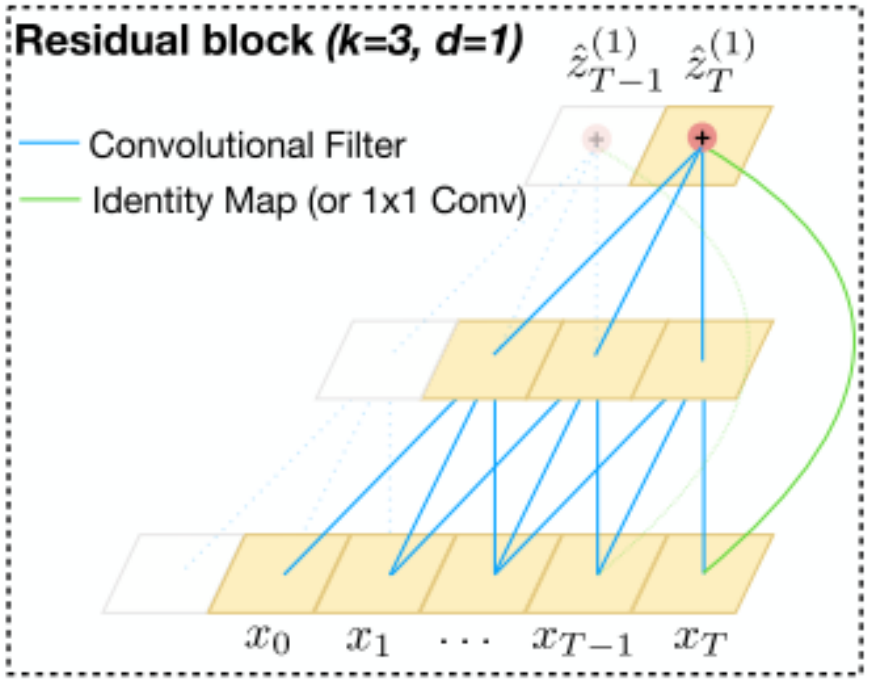
- Fully convolutional architecture used to predict sequences
- Convolutions must be **causal** to ensure that the prediction $P(x_{t+1}|x_1,\ldots,x_t)$ emitted by the model at time step t will not depend on any of the future time steps $x_{t+1},x_{t+2},\ldots,x_{t+n}$.



- C. Lea, R. Vidal, A. Reiter, G. Hager. Temporal convolutional networks: A unified approach to action segmentation. In European Conference on Computer Vision, pages 47–54. Springer, 2016.
- Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV (2013)

Temporal Convolutional Block

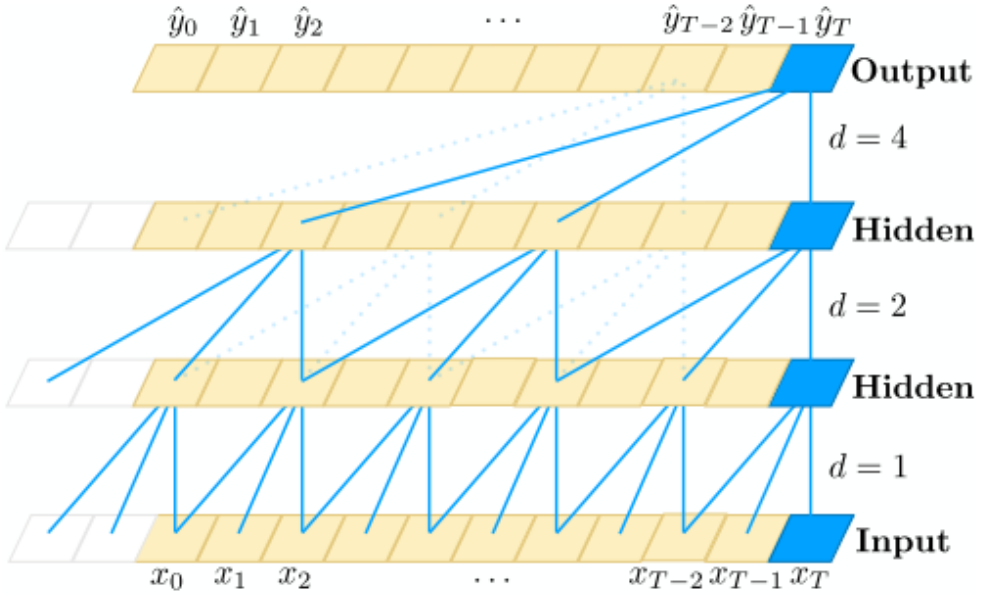




-S. Bai, J. Kolter, V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271, 2018.

Dilated convolution

- it effectively allows the network to **operate on a coarser scale** than standard convolution;
- stacking dilated convolutions allows the network to have **larger receptive field even with a few layers**.

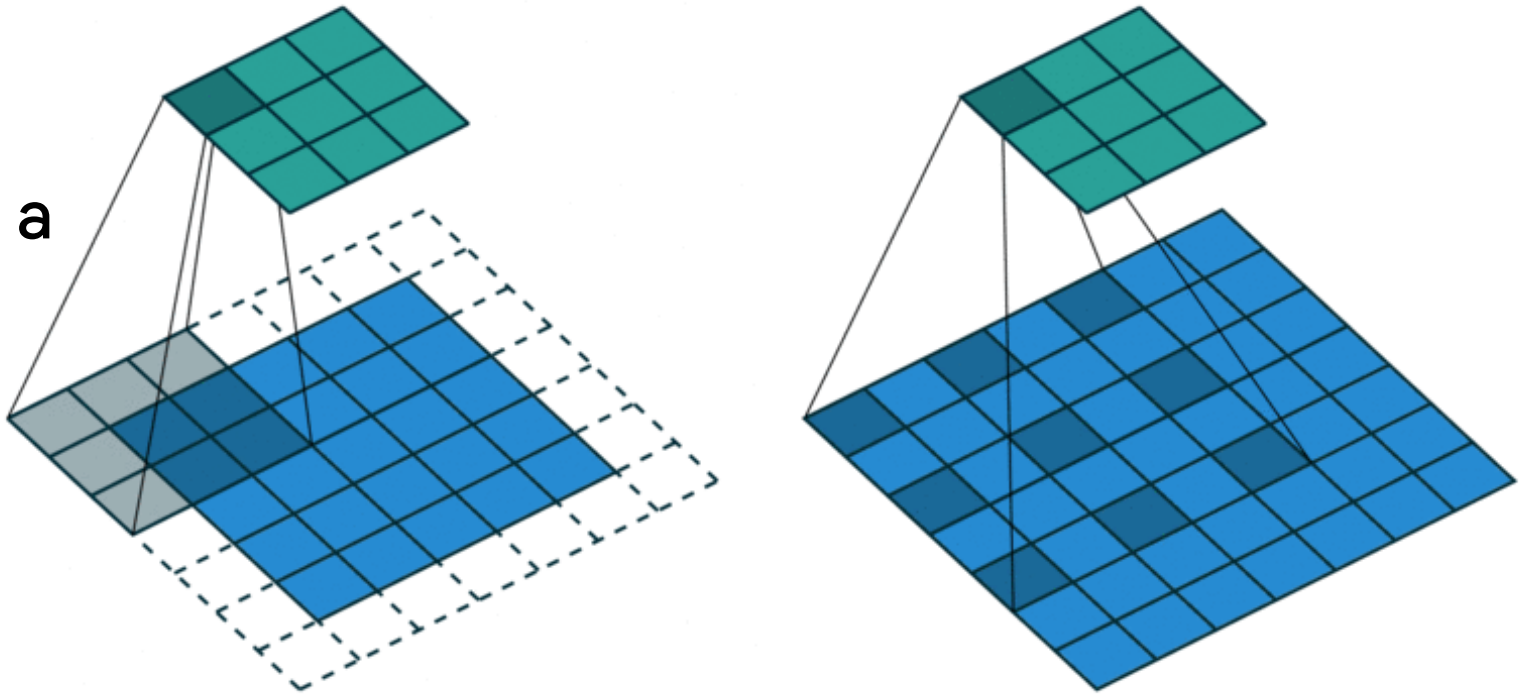


A dilated causal convolution with dilation factor d=1,2,4 and kernel size k=3

-S. Bai, J. Kolter, V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271, 2018.

Dilated convolution

- it effectively allows the network to **operate on a coarser scale** than standard convolution;
- stacking dilated convolutions allows the network to have **larger receptive field even with a few layers**.



-S. Bai, J. Kolter, V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271, 2018.

Other approaches

Mamba: Linear-Time Sequence Modeling with Selective State Spaces

Albert Gu[*1] and Tri Dao[*2]

[1]Machine Learning Department, Carnegie Mellon University

[2]Department of Computer Science, Princeton University
agu@cs.cmu.edu, tri@tridao.me

Abstract

Foundation models, now powering most of the exciting applications in deep learning, are almost universally based on the Transformer architecture and its core attention module. Many subquadratic-time architectures such as linear attention, gated convolution and recurrent models, and structured state space models (SSMs) have been developed to address Transformers' computational inefficiency on long sequences, but they have not performed as well as attention on important modalities such as language. We identify that a key weakness of such models is their inability to perform content-based reasoning, and make several improvements. First, simply letting the SSM parameters be functions of the input addresses their weakness with discrete modalities, allowing the model to *selectively* propagate or forget information along the sequence length dimension depending on the current token. Second, even though this change prevents the use of efficient convolutions, we design a hardware-aware parallel algorithm in recurrent mode. We integrate these selective SSMs into a simplified end-to-end neural network architecture without attention or even MLP blocks (**Mamba**). Mamba enjoys fast inference (5× higher throughput than Transformers) and linear scaling in sequence length, and its performance improves on real data up to million-length sequences. As a general sequence model backbone, Mamba achieves state-of-the-art performance across several modalities such as language, audio, and genomics. On language modeling, our Mamba-3B model outperforms Transformers of the same size and matches Transformers twice its size, both in pretraining and downstream evaluation.

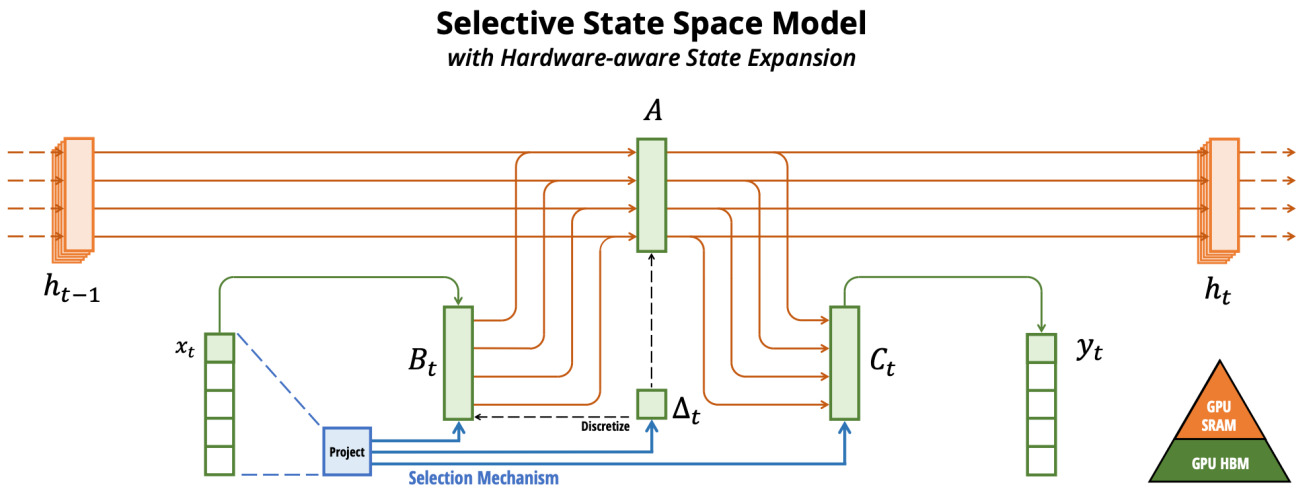


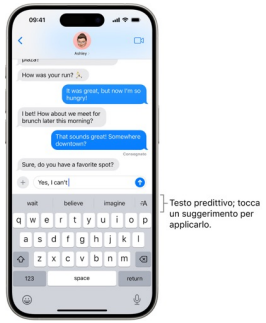
Figure 1: (**Overview**.) Structured SSMs independently map each channel (e.g. $D = 5$) of an input x to output y through a higher dimensional latent state h (e.g. $N = 4$). Prior SSMs avoid materializing this large effective state (DN, times batch size B and sequence length L) through clever alternate computation paths requiring time-invariance: the (Δ, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$) parameters are constant across time. Our selection mechanism adds back input-dependent dynamics, which also requires a careful hardware-aware algorithm to only materialize the expanded states in more efficient levels of the GPU memory hierarchy.

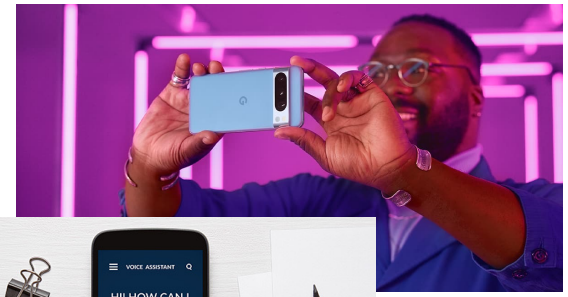
-https://arxiv.org/pdf/2312.00752

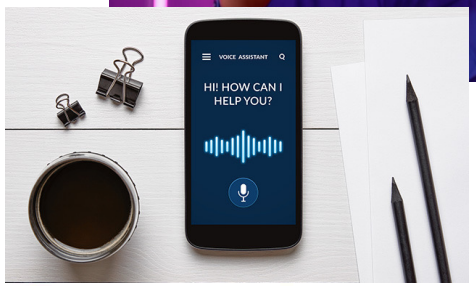
So, it’s time to draw some conclusions!

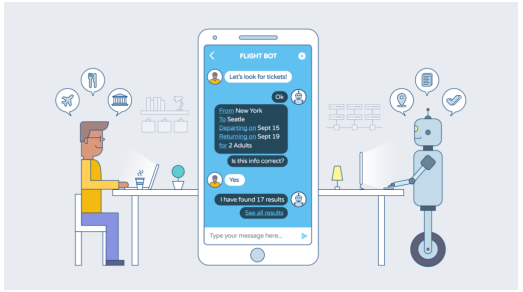
1st edition of the course… feedback is fundamental!

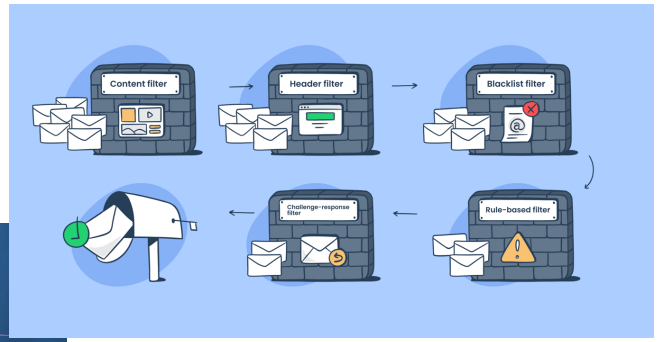
Many technologies... still many to be made!

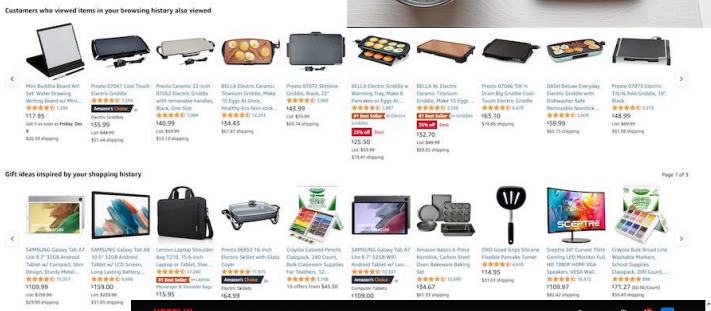


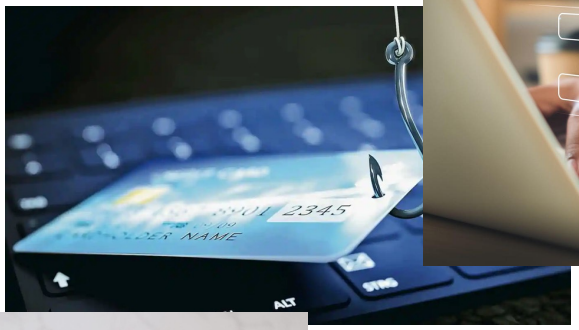


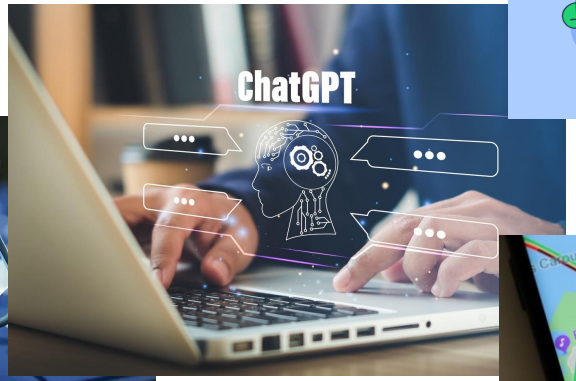


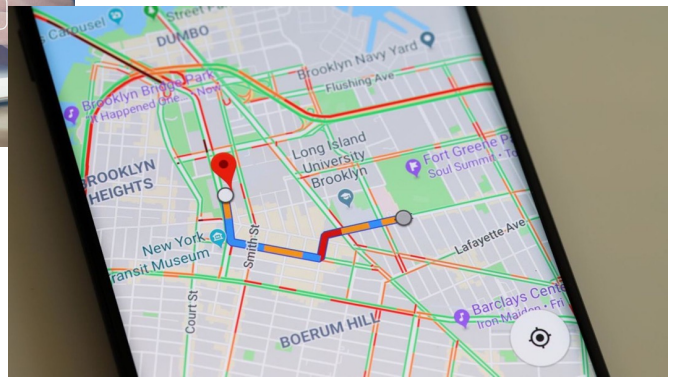


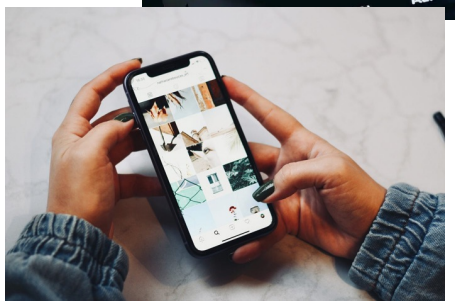


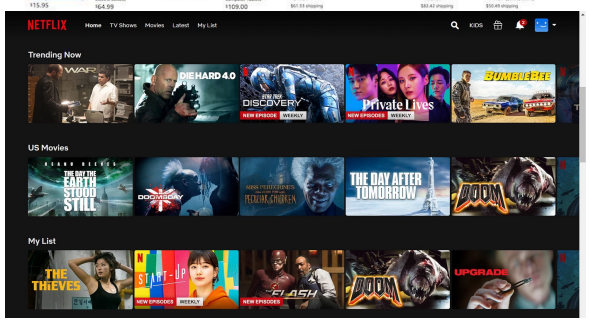


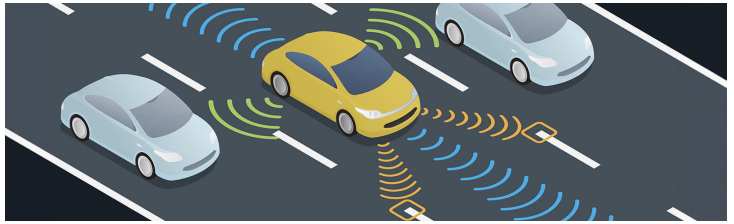












Ad hoc AI: exploiting data available only to single companies...

- Financial: turnover, operating costs, profit margins, balance sheet.
- Sales: sales volumes, average prices, discounts, profitability per product/service.
- Customers: demographic profiles, purchase history, feedback, retention rates.
- Marketing: advertising campaigns, conversion rates, ROI analysis.
- Operations: inventory levels, production times, logistics costs.
- Suppliers: procurement costs and times, supplier performance.
- Human resources: hours worked, productivity, turnover, training.

Thank you!

Gian Antonio Susto