



Argentina  
programa  
4.0



Universidad  
Nacional  
de San Martín

---

# Módulo 3

# Aprendizaje Automático



Argentina  
programa  
4.0



Universidad  
Nacional  
de San Martín

# Módulo 3

# Aprendizaje Automático

Semana 11.

Pantallazo de tópicos avanzados:

*attention mechanisms, transformers, LLMs, reinforcement learning  
Stable diffusion*

# Aprendizaje Automático Generativo: Imágenes, texto...

# Contenidos del módulo

## ML Clásico

- Árboles de Decisión
- Métodos de Ensemble
  - Bagging / Pasting → Random Forest
  - Boosting
- Support Vector Machines

## Deep Learning

- Redes Neuronales
- Redes Neuronales Convolucionales
- Auto-Encoders / Auto-Encoders Variacionales
- Redes Neuronales Recurrentes (LSTM, otras)
- Extras:
  - Generative Adversarial Networks (GAN)
- Grandes modelos de lenguaje (LLMs)

# Contenidos del módulo

## Deep Learning

## ML Clásico

- Árboles de Decisión
- Métodos de Ensemble
  - Bagging / Pasting → Random Forests
  - Boosting
- Support Vector Machines

- Redes Neuronales
- Redes Neuronales Convolucionales
- Auto-Encoders / Auto-Encoders Variacionales
- Redes Neuronales Recurrentes (LSTM, otras)
- Extras:
  - Generative Adversarial Networks (GAN)
  - Reinforcement Learning

# La revolución del aprendizaje profundo

## Attention Is All You Need

### UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS

Alec Radford & Luke Metz  
indico Research  
Boston, MA  
[{alec, luke}@indico.io](mailto:{alec, luke}@indico.io)

Soumith Chintala  
Facebook AI Research  
New York, NY  
[soumith@fb.com](mailto:soumith@fb.com)

Noam Shazeer\*  
Google Brain  
[noam@google.com](mailto:noam@google.com)

Niki Parmar\*  
Google Research  
[nikip@google.com](mailto:nikip@google.com)

Jakob Uszkoreit  
Google Research  
[usz@google.com](mailto:usz@google.com)

Aidan N. Gomez\* †  
University of Toronto  
[aidan@cs.toronto.edu](mailto:aidan@cs.toronto.edu)

Lukasz Kaiser\*  
Google Brain  
[lukasz.kaiser@google.co](mailto:lukasz.kaiser@google.co)

Illia Polosukhin\* ‡  
[illia.polosukhin@gmail.com](mailto:illia.polosukhin@gmail.com)

### Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein  
Stanford University  
  
Eric A. Weiss  
University of California, Berkeley  
  
Niru Maheswaranathan  
Stanford University  
  
Surya Ganguli  
Stanford University



# Texto

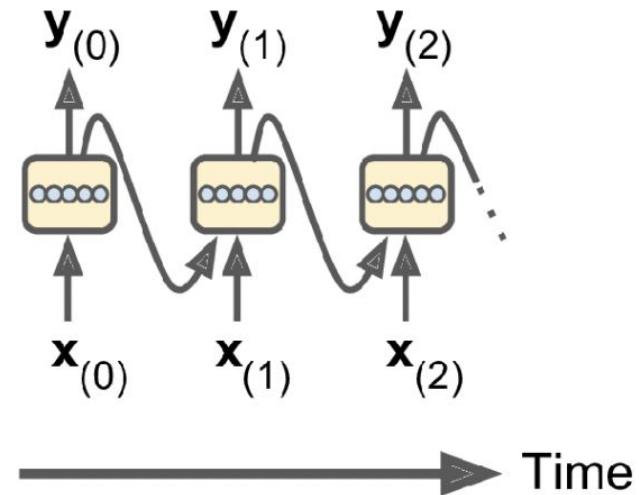
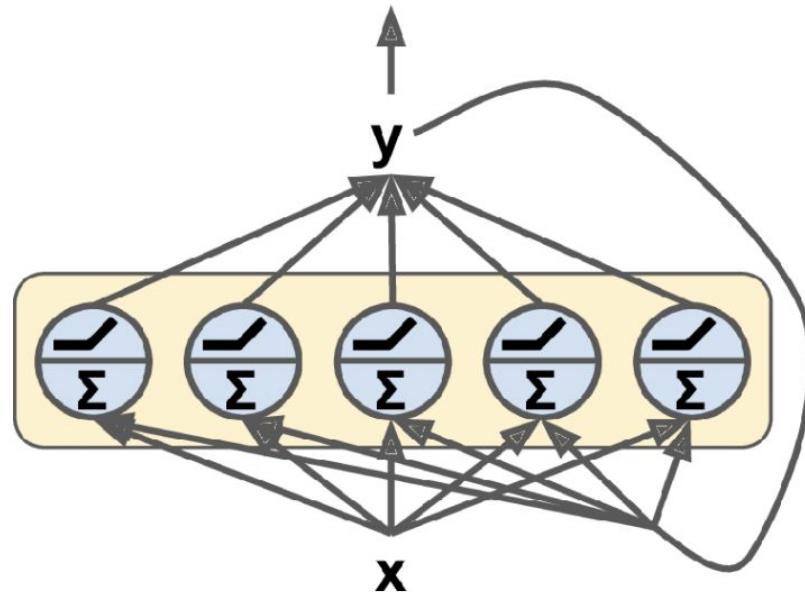
# Procesamiento del Lenguaje Natural (NLP)



# Series temporales: redes recurrentes

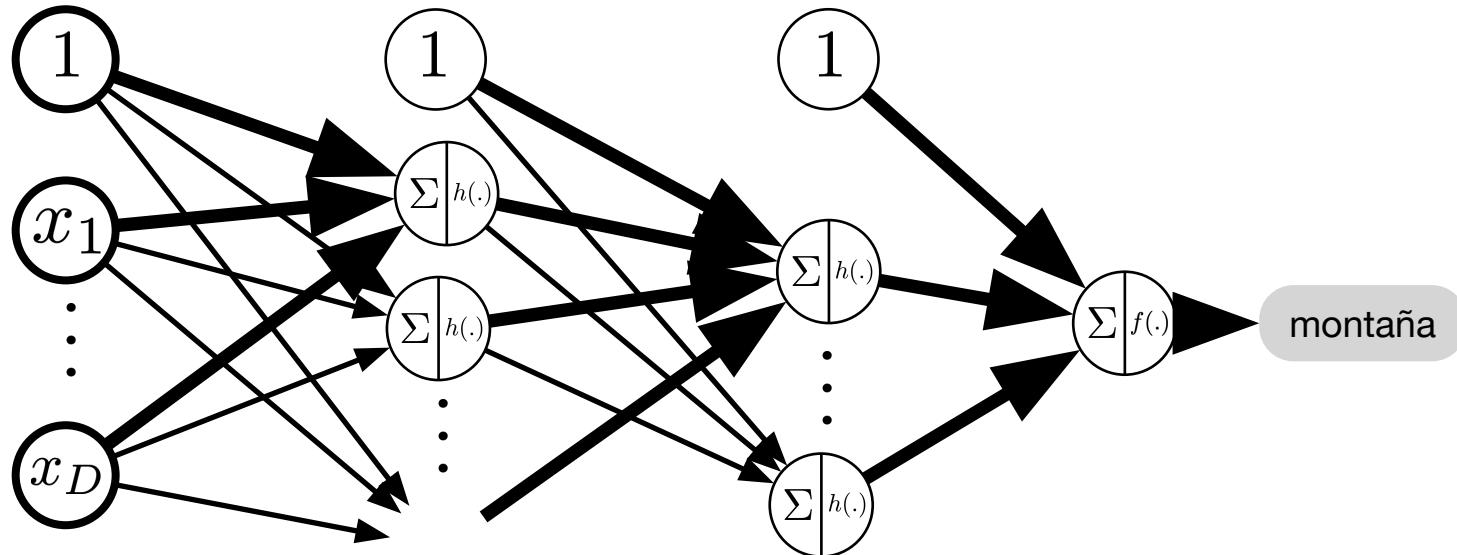
Se transmite, en cada fase del entrenamiento, el estado anterior

Capa de muchas neuronas, desenrollada en el tiempo

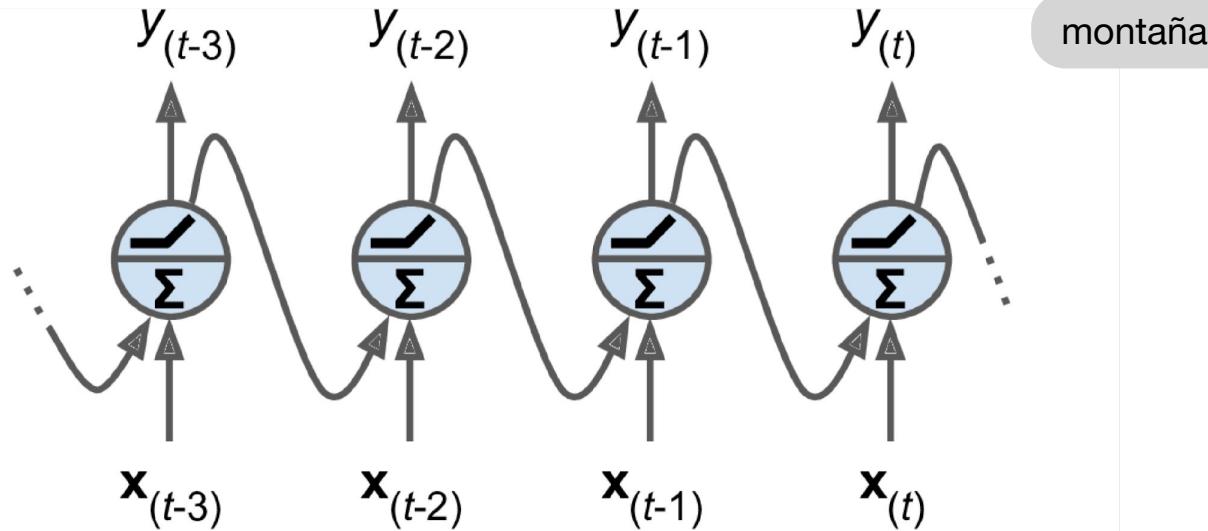


# Secuencias

“Para las vacaciones estamos pensando en ir a la ...”



# Redes Neuronales Recurrentes



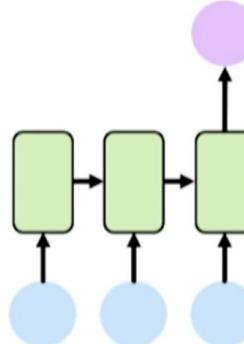
“Para las vacaciones estamos pensando en ir a la ...”

# Series temporales/secuencias

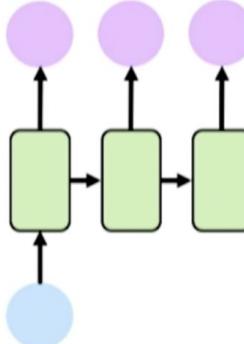


One to One

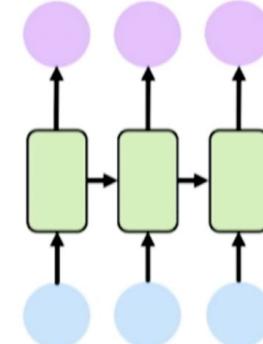
Estático. Correlaciones  
sólo espaciales (e.g. CNN)



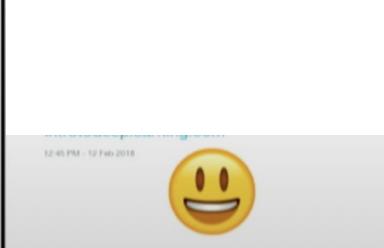
Secuencia a vector



Vector a secuencia



Secuencia a secuencia



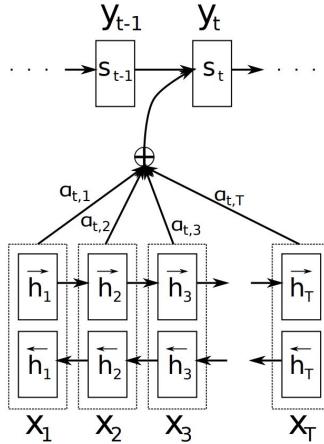
"A baseball player throws a ball."



# Avances disruptivos

2014

Mecanismos  
de atención



Published as a conference paper at ICLR 2015

## NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau

Jacobs University Bremen, Germany

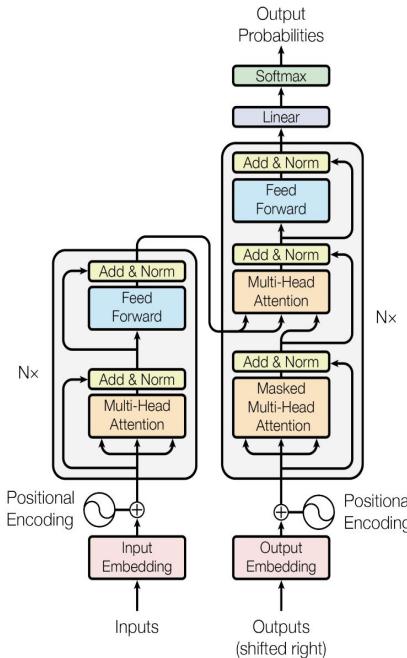
KyungHyun Cho    Yoshua Bengio\*

Université de Montréal

# Avances disruptivos

2017

Transformers



## Attention Is All You Need

Ashish Vaswani\*  
Google Brain  
avaswani@google.com

Noam Shazeer\*  
Google Brain  
noam@google.com

Niki Parmar\*  
Google Research  
nikip@google.com

Jakob Uszkoreit\*  
Google Research  
usz@google.com

Llion Jones\*  
Google Research  
llion@google.com

Aidan N. Gomez\* †  
University of Toronto  
aidan@cs.toronto.edu

Lukasz Kaiser\*  
Google Brain  
lukaszkaiser@google.com

Illia Polosukhin\* ‡  
illia.polosukhin@gmail.com

# Métodos avanzados: mecanismos de atención y transformers

- Procesamiento del lenguaje natural

- Traducciones

- Contexto

- Cada palabra se asocia a un número  
(¡puede considerar todas las palabras!)

- *Word Embedding*: vector de características lingüísticas

- Vectores en un espacio de menor dimensión: semanticamente parecidos, tendrán embeddings parecidos: vector de tamaño fijo (¡obtenidos también por ML!)

- *Position Embedding*: forma de guardar posición de las palabras

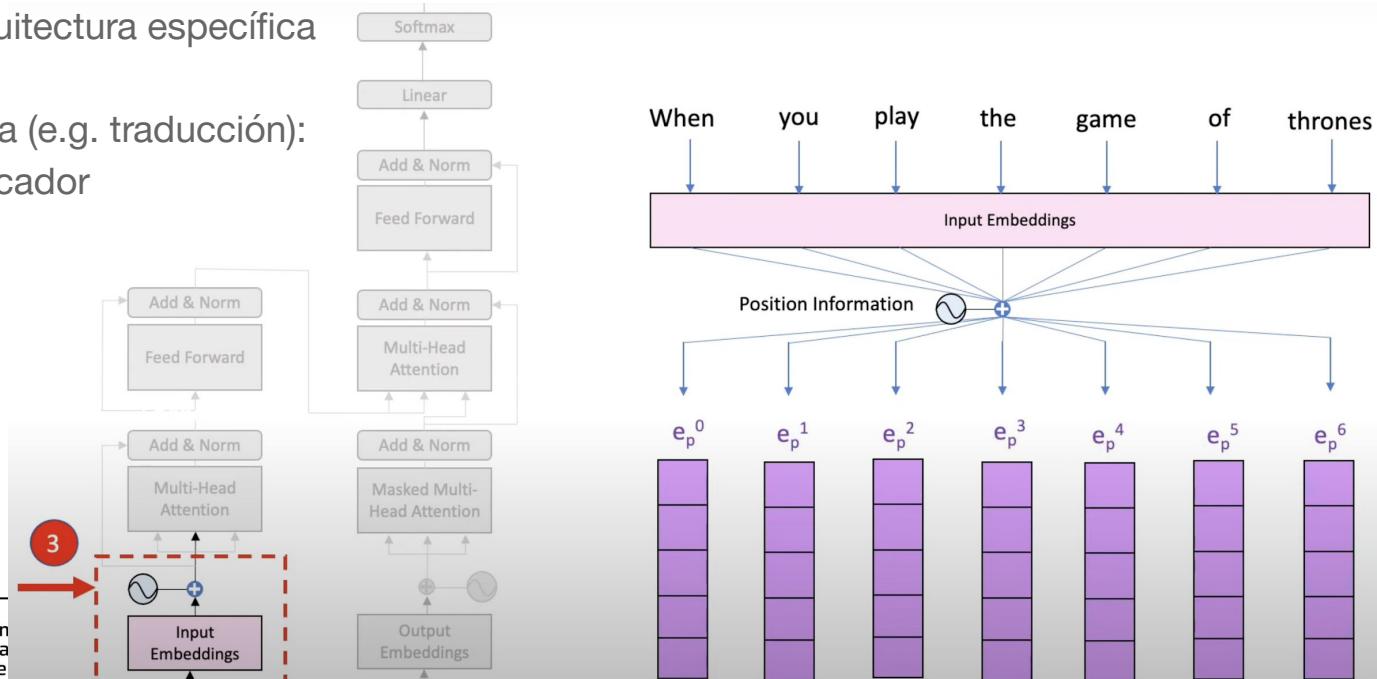
- Vector de igual dimensión que se combina con *Word Embedding*

Position Embeddings	
When	pos = 1
$e_0$	$p_0$
0.42	0.82
0.31	0.79
0.73	0.76
0.36	0.74
0.99	0.71
you	pos = 2
$e_1$	$p_0$
0.87	-0.33
-0.64	-0.25
0.81	-0.17
0.26	-0.09
-0.35	-0.02
play	pos = 3
$e_2$	$p_0$
0.02	0.27
0.01	0.39
-0.24	0.50
-0.07	0.60
0.00	0.69
the	pos = 4
$e_3$	$p_0$
0.38	-0.78
0.16	-0.87
0.01	-0.94
0.09	-0.98
0.00	-1.00

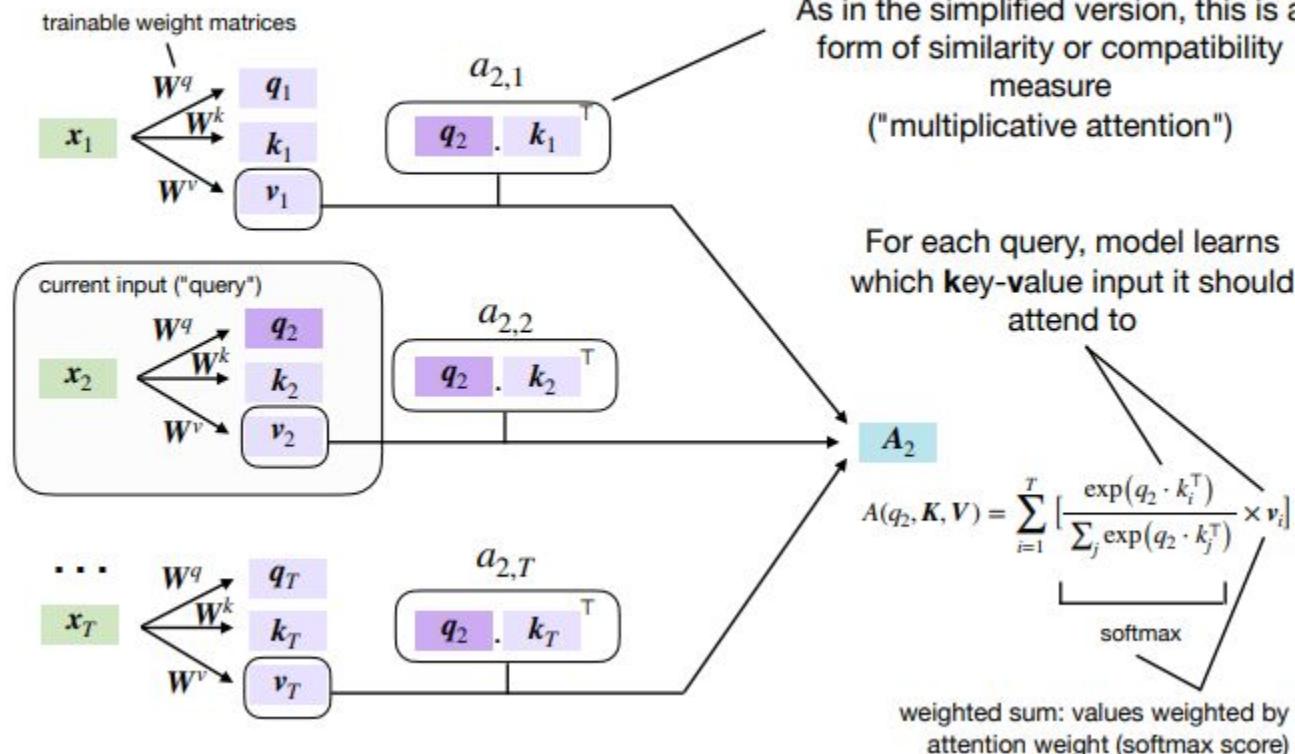
$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000} \frac{2i}{d}\right)$$

# Métodos avanzados: mecanismos de atención y transformers

- Procesamiento del lenguaje natural
- Traducciones
- Contexto
- Red neuronal con arquitectura específica (y no recurrente)
- Secuencia a secuencia (e.g. traducción): codificador y decodificador



# Mecanismos de auto-atención



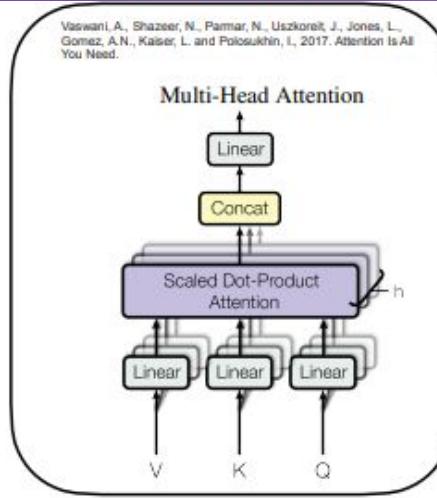
# Autoatención Multi-Head

$$x \times \begin{matrix} W_1^q \\ W_2^q \\ W_3^q \end{matrix} = \begin{matrix} Q_1 \\ Q_2 \\ Q_3 \end{matrix}$$

$$x \times \begin{matrix} W_1^k \\ W_2^k \\ W_3^k \end{matrix} = \begin{matrix} K_1 \\ K_2 \\ K_3 \end{matrix}$$

$$x \times \begin{matrix} W_1^v \\ W_2^v \\ W_3^v \end{matrix} = \begin{matrix} V_1 \\ V_2 \\ V_3 \end{matrix}$$

$$\dots \quad A_1 \quad \text{concat} \quad A_2 \quad A_3 \longrightarrow \begin{matrix} A \end{matrix} \times \begin{matrix} W_o \end{matrix}$$



# Transformers

Componentes llave de la arquitectura *Transformer*:

- Mecanismo de Autoatención
- Autoatención *Multi-Head*
- Codificación Posicional
- Capas Apiladas
- Conexiones Residuales
- Normalización de Capa
- Arquitectura Codificador-Decodificador
- Autoatención Enmascarada
- Redes Neuronales Feedforward por Posición
- Capa de Salida

Rendimiento notable en diversas tareas de NLP: traducción automática, generación de texto, respuesta a preguntas, análisis de sentimientos.

También se utilizan en visión por computadora (por ejemplo, en la generación de subtítulos de imágenes).

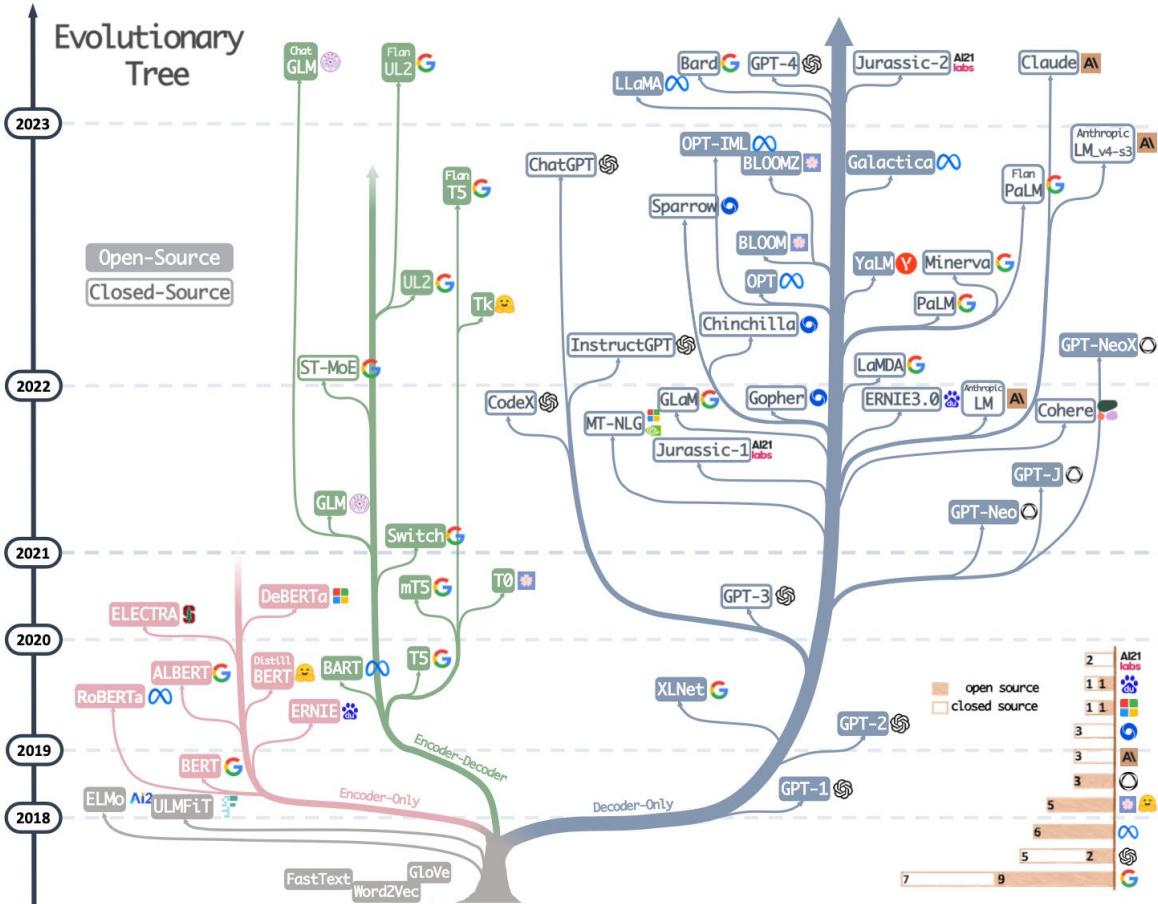
Usados en BERT, GPT y T5

# Redes “famosas” / Celebrities

## Procesamiento del lenguaje natural

- Generative Pre-trained Transformer (GPT-N)
- BERT: Bidirectional Encoder Representations from Transformers (Red de google creada en 2018)
- LLaMA (Large Language Model Meta AI)
- LaMDA (Language Model for Dialogue Applications)
- Muchos otros....
- Parecen generalizar de forma extraordinaria...
- Polémica de sentimientos, auto-percepción, etc.

# Grandes Modelos de Lenguaje (LLMs)



# Grandes Modelos de Lenguaje (LLMs)

(actualizado a marzo de 2023)

• BERT 340M

• GPT-1 117M

• GPT-2 1.5B

11B T5

11B Plato-XL

11B Macaw

Cohere

GPT-NeoX-20B

20B

52.4B

Megatron-11B

ruGPT-3

GPT-3  
175B

Jurassic-1  
178B

MT-NLG  
530B

Cedille

Fairseq

Anthropic-LM

LaMDA  
LaMDA 2  
Bard  
137B

GPT-J

6B  
9.4B

BlenderBot2.0

52B  
RL-CAI  
Claude

Gopher  
280B

Luminous  
200B

CM3  
VLM-4  
mGPT

13B  
10B  
13B

BLOOM  
BLOOMZ  
176B

11B  
Flan-T5

Kosmos-1  
Atlas

11B  
NLLB

1.6B\*  
ChatGLM-6B

GPT-4  
Undisclosed  
\*

7B  
Alpaca  
Toolformer

6.7B\*  
LLaMA

65B\*  
OPT-175B

BB3  
OPT-IML  
175B

20B\*  
Galactica

10B\*  
VIMA  
WeLM

10B\*

UL2  
20B

10B

NOOR

● SeeKeR

2.7B

Z-Code++

710M\*

20B\*

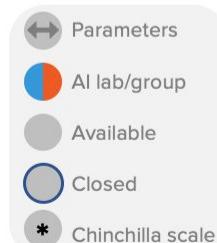
Gato

1.2B

FIM

6.9B\*

\*



Beeswarm/bubble plot, sizes linear to scale. Selected highlights only. \*Chinchilla scale means T:P ratio > 15:1. <https://lifearchitect.ai/chinchilla/> Alan D. Thompson. March 2023. <https://lifearchitect.ai/>



LifeArchitect.ai/models

# Los últimos vertiginosos meses



Oct. 2022

ChatGPT

Feb. 2023

OpenAssistant

Feb. 2023

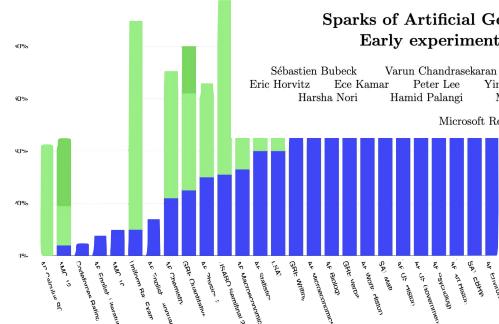
Llama

Creemos que podemos crear una revolución.

De la misma forma que Stable Diffusion ayudó al mundo a crear arte e imágenes de nuevas maneras, queremos mejorar el mundo proporcionando una IA conversacional asombrosa.

Mar. 2023

Pesos de Llama  
filtrados



GPT-4

Sparks of AGI  
paper

Pause Giant AI Experiments: An Open Letter

We call on all AI labs to immediately pause for at least 6 months the training of AI systems more powerful than GPT-4.



Carta Future of  
Life

Avances técnicos y  
chatbots por doquier



Julio 2023

Llama 2, licencia  
comercial

Stanford  
Alpaca



# Loros estocásticos (¿peligrosos?)

El término “loros estocásticos” fue acuñado en marzo de 2021.

## On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?



Emily M. Bender\*

ebender@uw.edu

University of Washington  
Seattle, WA, USA

Angelina McMillan-Major

aymm@uw.edu

University of Washington  
Seattle, WA, USA

Timnit Gebru\*

timnit@blackinai.org

Black in AI  
Palo Alto, CA, USA

Shmargaret Shmitchell

shmargaret.shmitchell@gmail.com

The Aether

applied with great success to a wide variety of tasks [e.g. 2, 149].

However, no actual language understanding is taking place in LM-driven approaches to these tasks, as can be shown by careful manipulation of the test data to remove spurious cues the systems are leveraging [21, 93]. Furthermore, as Bender and Koller [14] argue from a theoretical perspective, languages are systems of signs [37], i.e. pairings of form and meaning. But the training data for LMs is only form; they do not have access to meaning. Therefore, claims about model abilities must be carefully characterized.

As the late Karen Spärck Jones pointed out: the use of LMs

## Riesgos

- Costos ambientales.
- Refuerzo de sesgos y visiones hegemónicas.
- Resultados engañosos por la forma de evaluación
- Esfuerzos de investigación mal dirigidos

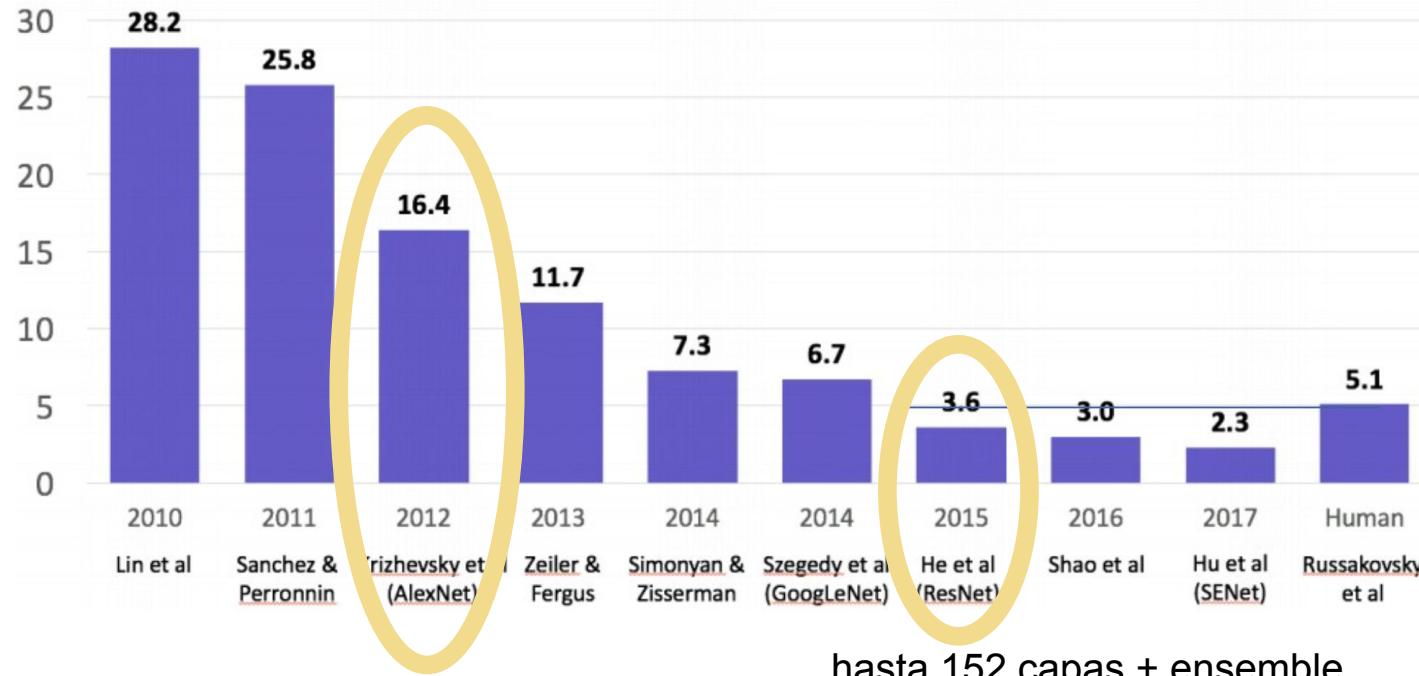
# Discusión

- La I.A. es un campo amplio, multidisciplinario, que excede al aprendizaje automático. Su objetivo es la creación de “agentes” racionales, cuyas acciones lleven a cumplir sus objetivos.
- Esta visión es problemática, si el objetivo está dado a priori. Particularmente, a medida que los sistemas se vuelven más potentes y se acercan a la I.A. General.
- El Aprendizaje Profundo es una rama de Machine Learning basada en el entrenamiento de modelos de redes neuronales de diversas arquitecturas (feed-forward, CNNs, RNNs, Transformers, ...)
- Los Grandes Modelos de Lenguaje recientes se basan en la tecnología de los Transformers (2017), y les permite generar grandes cantidades de texto coherente
- ChatGPT y chatbots usan estas tecnologías, que no están exentas de problemas y riesgos.
- Como siempre, en la medida que aumenta el poder de la herramienta, la responsabilidad debería crecer.

# Imágenes

# La revolución del aprendizaje profundo

Large Scale Visual Recognition Challenge (ILSVRC) -  
ImageNet

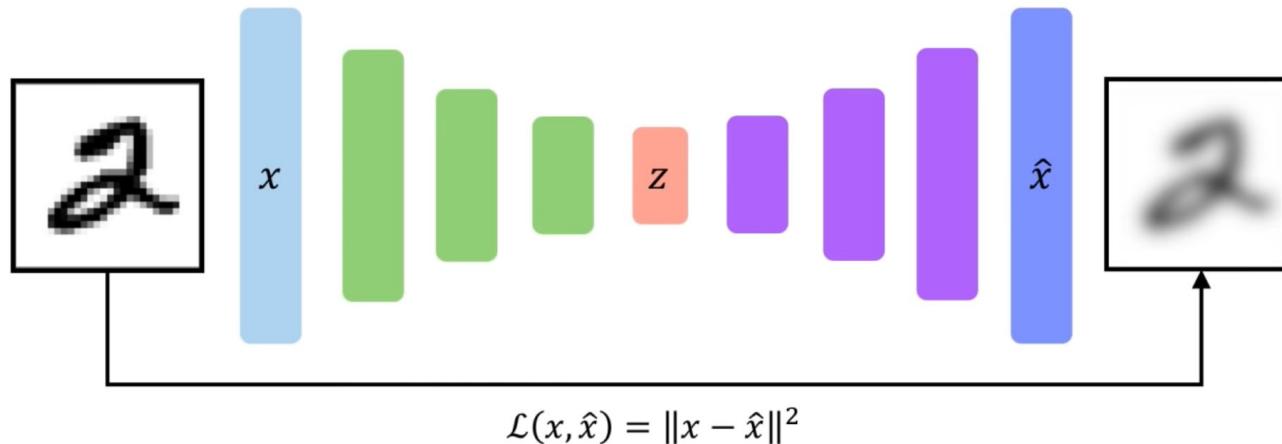


# Autoencoder con CNN

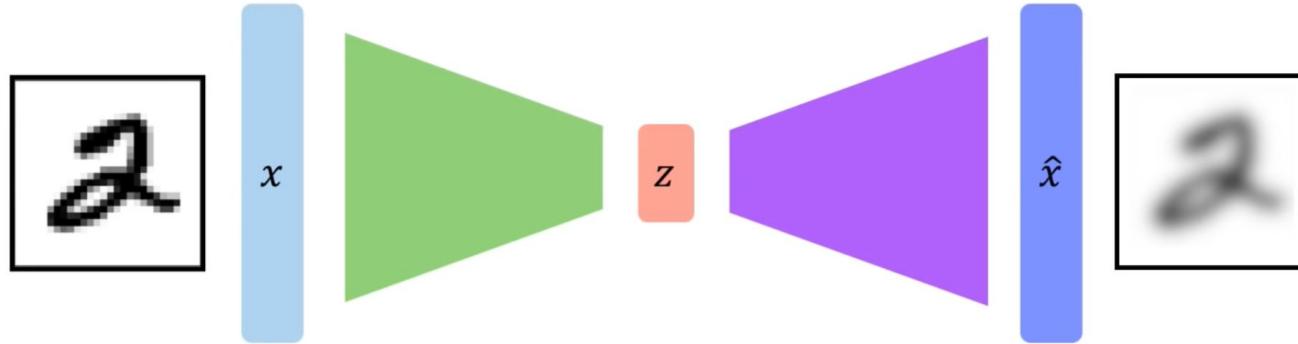
Objetivo de un *autoencoder*: aprender una representación latente para un conjunto de datos mediante codificación y decodificación

¿Cómo podemos aprender este espacio latente?

Entrenar el modelo para utilizar estas características para reconstruir los datos originales

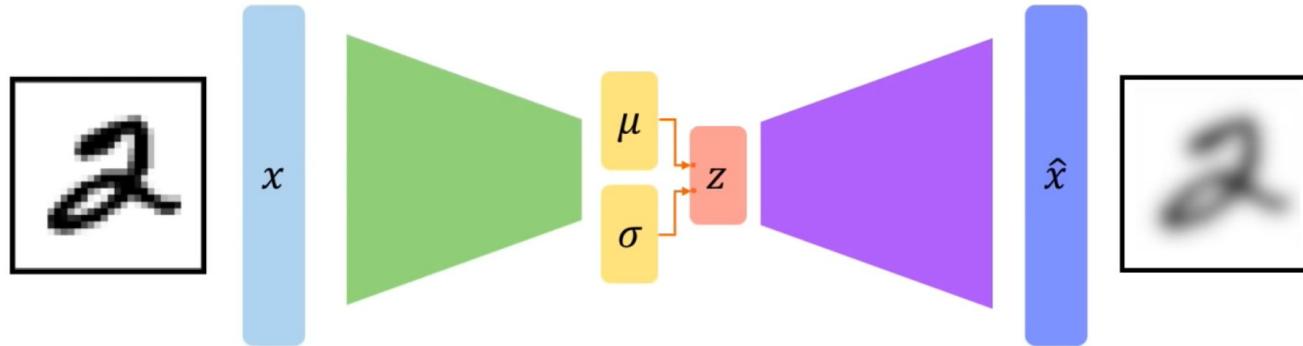


# Autoencoders deterministas



# Autoencoders variacionales

- Distribución de las variables latentes (por ejemplo, media y varianza)



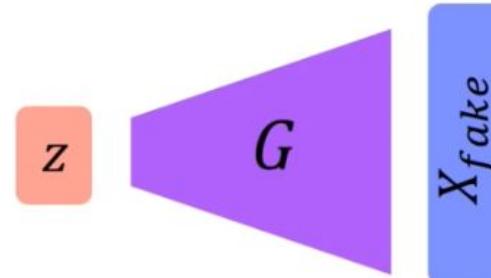
- Representación más suave de los datos
- Mejora de la calidad de la reconstrucción
- Posibilidad de **crear** nuevas imágenes similares a los datos de entrada

# Redes Generativas Adversárias

## *Generative Adversarial Networks (GANs)*

- ¿Cómo empezar de la nada?
- Cómo mejorar sobre VAE específicamente para la generación de datos

**generador:** convierte unas pocas variables en una imitación de los datos



# Redes Generativas Adversarias

## *Generative Adversarial Networks (GANs)*

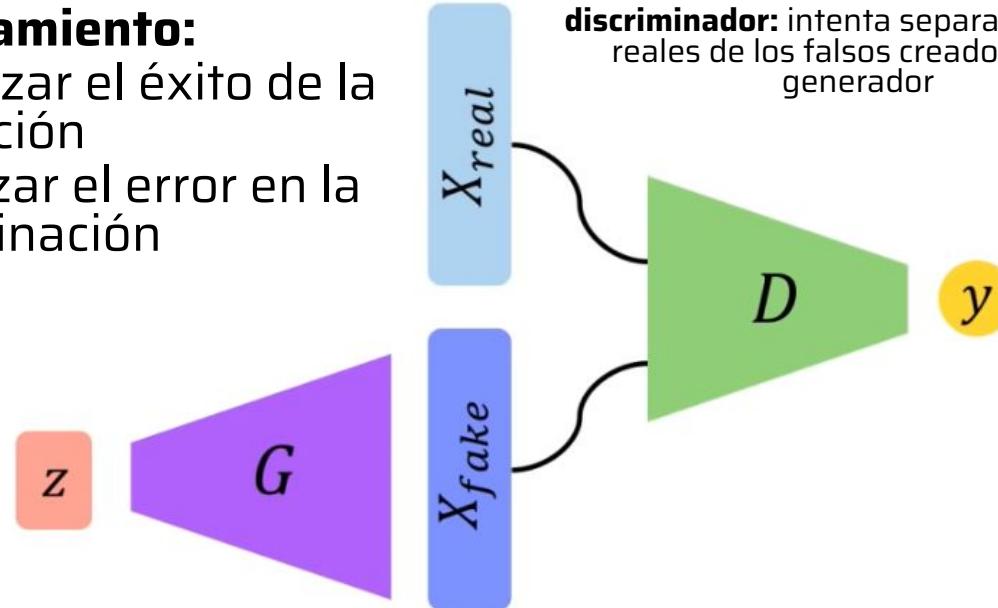
**Solución:** ¡hacer que dos redes compitan entre sí!

### Entrenamiento:

Maximizar el éxito de la generación

Minimizar el error en la discriminación

**discriminador:** intenta separar los datos reales de los falsos creados por el generador



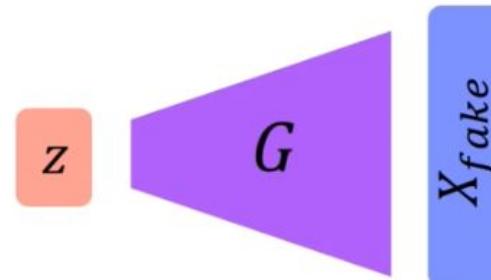
**generador:** convierte el ruido en una imitación de los datos para intentar engañar al discriminador

# Redes Generativas Adversarias

## *Generative Adversarial Networks (GANs)*

- Una vez entrenada la red generativa, puede utilizarse para generar nuevos datos
- Si en lugar de variables aleatorias en el espacio latente, entendemos la representación de la variable, podemos crear imágenes con **características específicas**

**generador:** convierte el ruido en una imitación de los datos para intentar engañar al discriminador



# Redes Generativas Adversárias

## *Generative Adversarial Networks (GANs)*

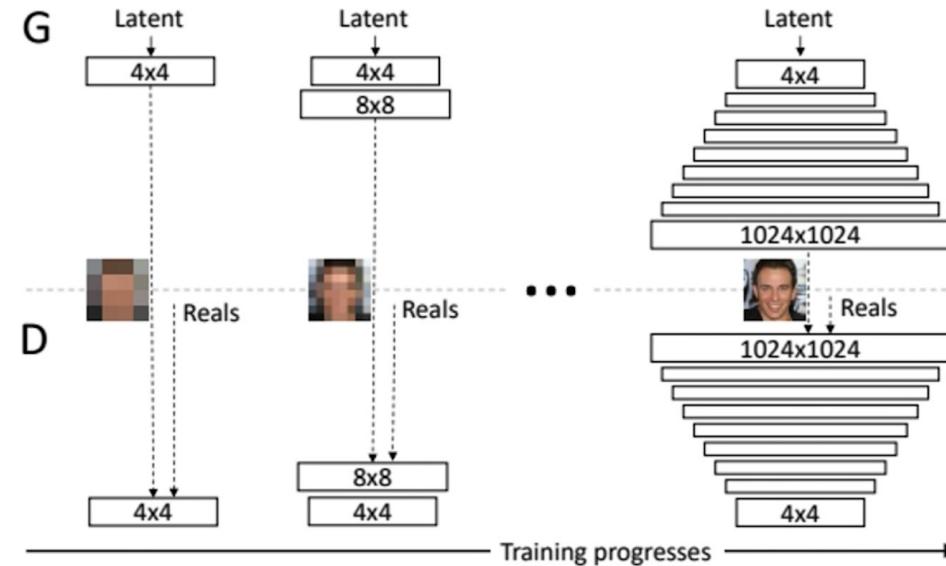
- Una vez entrenada la red generativa, puede utilizarse para generar nuevos datos
- Si en lugar de variables aleatorias en el espacio latente, entendemos la representación de la variable, podemos crear imágenes con **características específicas**



# Crecimiento progresivo de las GANs

- Construir de forma iterativa más detalles en las instancias que se generan

- añadiendo progresivamente capas de mayor resolución espacial (imágenes)
- Construcción progresiva del generador y de los discriminadores
- Imágenes sintéticas muy bien resueltas

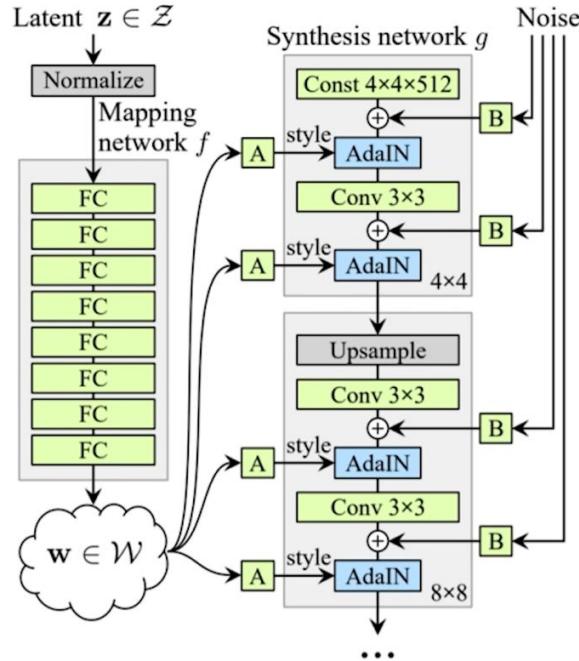


Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen, <https://arxiv.org/abs/1710.10196>

Generar caras eligiendo atributos: <https://dash.gallery/dash-gan-editor/>

# Style GAN

## Transferencia de estilo



<https://thispersondoesnotexist.com/>

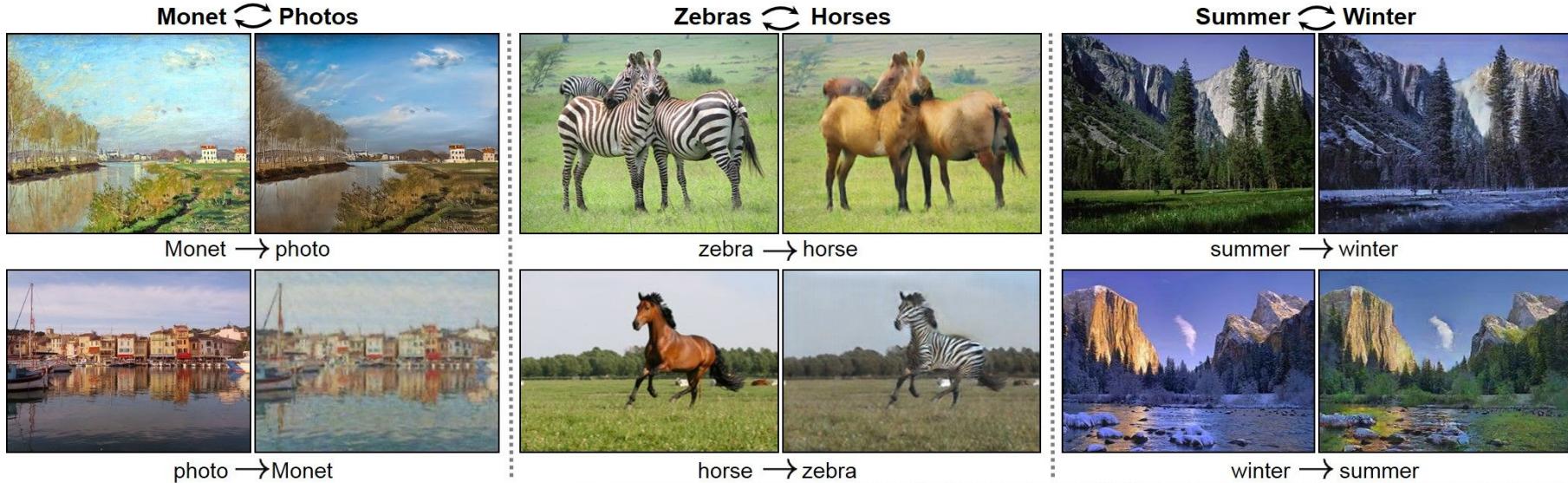


201

9

# Otros desarrollos

- Traducción por parejas (*paired translation*)
  - coloración: <https://deepai.org/machine-learning-model/colorizer>
- CycleGAN: dos generadores y dos discriminadores trabajando al mismo tiempo



<https://junyanz.github.io/CycleGAN/>

# Modelos de difusión



Search...

Help | Advanced

Computer Science > Machine Learning

[Submitted on 12 Mar 2015 ([v1](#)), last revised 18 Nov 2015 (this version, v8)]

## Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, Surya Ganguli

A central problem in machine learning involves modeling complex data-sets using highly flexible families of probability distributions in which learning, sampling, inference, and evaluation are still analytically or computationally tractable. Here, we develop an approach that simultaneously achieves both flexibility and tractability. The essential idea, inspired by non-equilibrium statistical physics, is to systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process. We then learn a reverse diffusion process that restores structure in data, yielding a highly flexible and tractable generative model of the data. This approach allows us to rapidly learn, sample from, and evaluate probabilities in deep generative models with thousands of layers or time steps, as well as to compute conditional and posterior probabilities under the learned model. We additionally release an open source reference implementation of the algorithm.

Subjects: [Machine Learning \(cs.LG\)](#); Disordered Systems and Neural Networks ([cond-mat.dis-nn](#)); Neurons and Cognition ([q-bio.NC](#)); Machine Learning ([stat.ML](#))

Cite as: [arXiv:1503.03585](#) [cs.LG]

(or [arXiv:1503.03585v8](#) [cs.LG] for this version)

<https://doi.org/10.48550/arXiv.1503.03585>

### Submission history

From: Jascha Sohl-Dickstein [[view email](#)]

[[v1](#)] Thu, 12 Mar 2015 04:51:37 UTC (5,395 KB)

[[v2](#)] Thu, 2 Apr 2015 06:48:02 UTC (5,397 KB)

[[v3](#)] Wed, 29 Apr 2015 06:00:20 UTC (5,403 KB)

[[v4](#)] Wed, 13 May 2015 01:57:49 UTC (5,409 KB)

[[v5](#)] Wed, 20 May 2015 03:19:10 UTC (4,586 KB)

# Modelos de difusión

In the case of Gaussian diffusion, we learn<sup>2</sup> the forward diffusion schedule  $\beta_{2\dots T}$  by gradient ascent on  $K$ . The variance  $\beta_1$  of the first step is fixed to a small constant to prevent overfitting. The dependence of samples from  $q(\mathbf{x}^{(1\dots T)} | \mathbf{x}^{(0)})$  on  $\beta_{1\dots T}$  is made explicit by using ‘frozen noise’ – as in (Kingma & Welling, 2013) the noise is treated as an additional auxiliary variable, and held constant while computing partial derivatives of  $K$  with respect to the parameters.

For binomial diffusion, the discrete state space makes gradient ascent with frozen noise impossible. We instead choose the forward diffusion schedule  $\beta_{1\dots T}$  to erase a constant fraction  $\frac{1}{T}$  of the original signal per diffusion step, yielding a diffusion rate of  $\beta_t = (T - t + 1)^{-1}$ .

## 2.5. Multiplying Distributions, and Computing Posteriors

Tasks such as computing a posterior in order to do signal denoising or inference of missing values requires multiplication of the model distribution  $p(\mathbf{x}^{(0)})$  with a second distribution, or bounded positive function,  $r(\mathbf{x}^{(0)})$ , producing a new distribution  $\tilde{p}(\mathbf{x}^{(0)}) \propto p(\mathbf{x}^{(0)}) r(\mathbf{x}^{(0)})$ .

Multiplying distributions is costly and difficult for many techniques, including variational autoencoders, GSNs, NADEs, and most graphical models. However, under a diffusion model it is straightforward, since the second distribution can be treated either as a small perturbation to each

### 2.5.2. MODIFIED DIFFUSION STEPS

The Markov kernel  $p(\mathbf{x}^{(t)} | \mathbf{x}^{(t+1)})$  for the reverse diffusion process obeys the equilibrium condition

$$p(\mathbf{x}^{(t)}) = \int d\mathbf{x}^{(t+1)} p(\mathbf{x}^{(t)} | \mathbf{x}^{(t+1)}) p(\mathbf{x}^{(t+1)}). \quad (17)$$

We wish the perturbed Markov kernel  $\tilde{p}(\mathbf{x}^{(t)} | \mathbf{x}^{(t+1)})$  to instead obey the equilibrium condition for the perturbed distribution,

$$\tilde{p}(\mathbf{x}^{(t)}) = \int d\mathbf{x}^{(t+1)} \tilde{p}(\mathbf{x}^{(t)} | \mathbf{x}^{(t+1)}) \tilde{p}(\mathbf{x}^{(t+1)}), \quad (18)$$

$$\frac{p(\mathbf{x}^{(t)}) r(\mathbf{x}^{(t)})}{\tilde{Z}_t} = \int d\mathbf{x}^{(t+1)} \tilde{p}(\mathbf{x}^{(t)} | \mathbf{x}^{(t+1)}) \cdot \frac{p(\mathbf{x}^{(t+1)}) r(\mathbf{x}^{(t+1)})}{\tilde{Z}_{t+1}}, \quad (19)$$

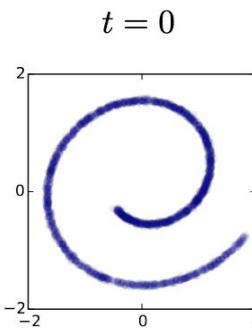
$$p(\mathbf{x}^{(t)}) = \int d\mathbf{x}^{(t+1)} \tilde{p}(\mathbf{x}^{(t)} | \mathbf{x}^{(t+1)}) \cdot \frac{\tilde{Z}_t r(\mathbf{x}^{(t+1)})}{\tilde{Z}_{t+1} r(\mathbf{x}^{(t)})} p(\mathbf{x}^{(t+1)}). \quad (20)$$

Equation 20 will be satisfied if

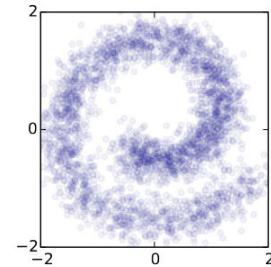
$$\tilde{p}(\mathbf{x}^{(t)} | \mathbf{x}^{(t+1)}) = p(\mathbf{x}^{(t)} | \mathbf{x}^{(t+1)}) \frac{\tilde{Z}_{t+1} r(\mathbf{x}^{(t)})}{\tilde{Z}_t r(\mathbf{x}^{(t+1)})}. \quad (21)$$

# Modelos de difusión

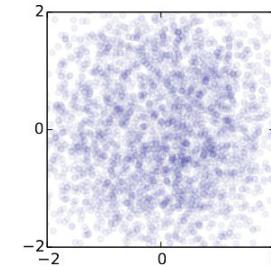
$q(\mathbf{x}^{(0 \cdots T)})$



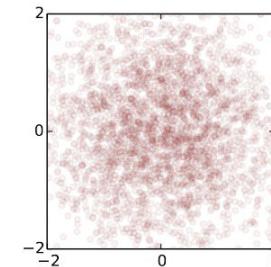
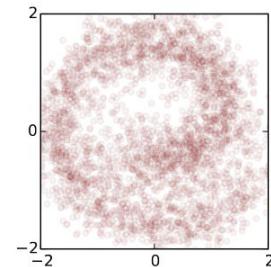
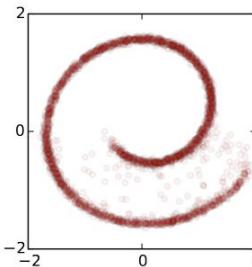
$t = \frac{T}{2}$



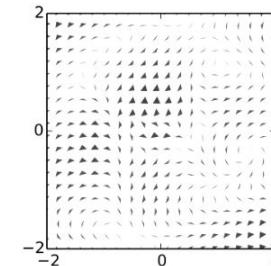
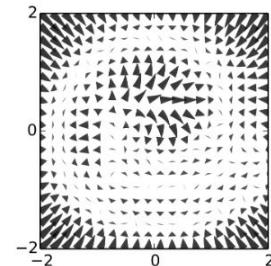
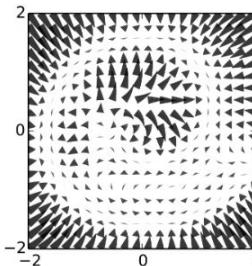
$t = T$



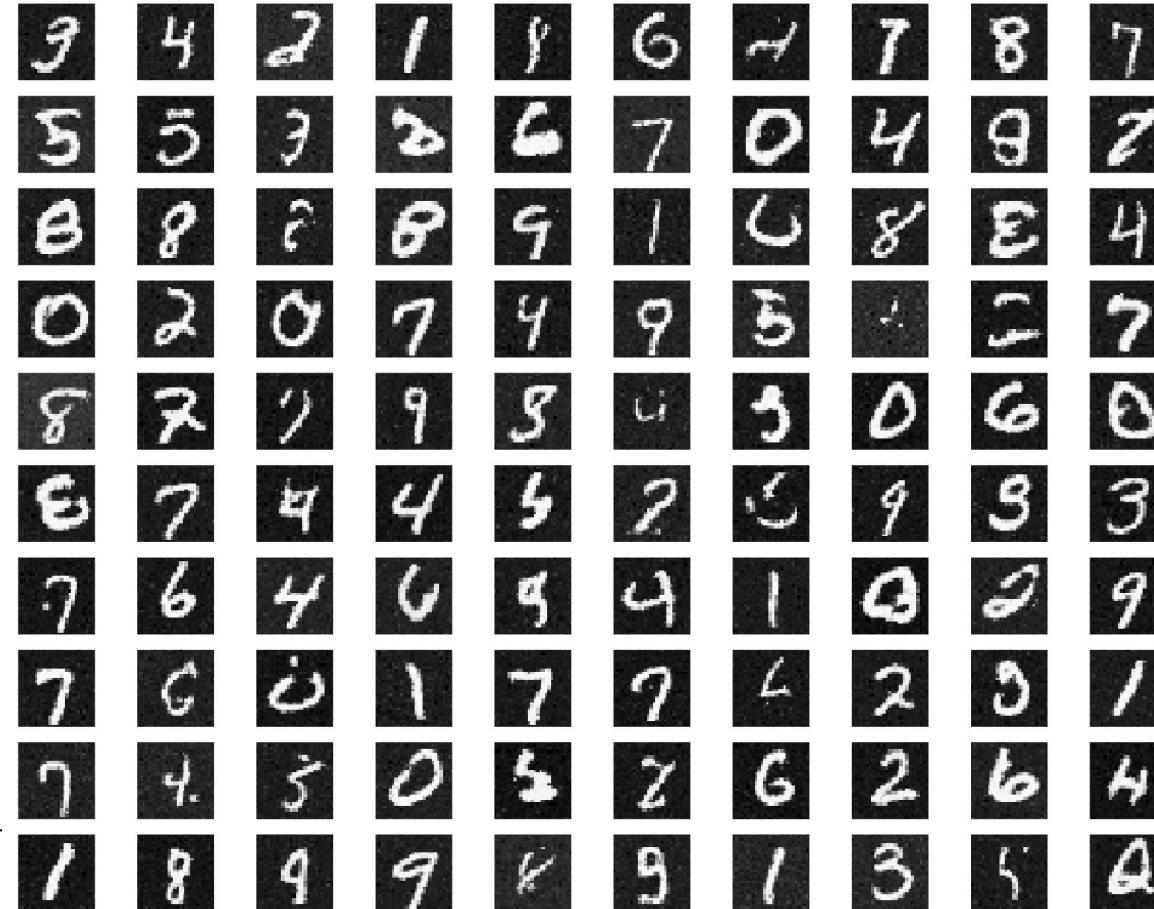
$p(\mathbf{x}^{(0 \cdots T)})$



$\mathbf{f}_\mu(\mathbf{x}^{(t)}, t) - \mathbf{x}^{(t)}$

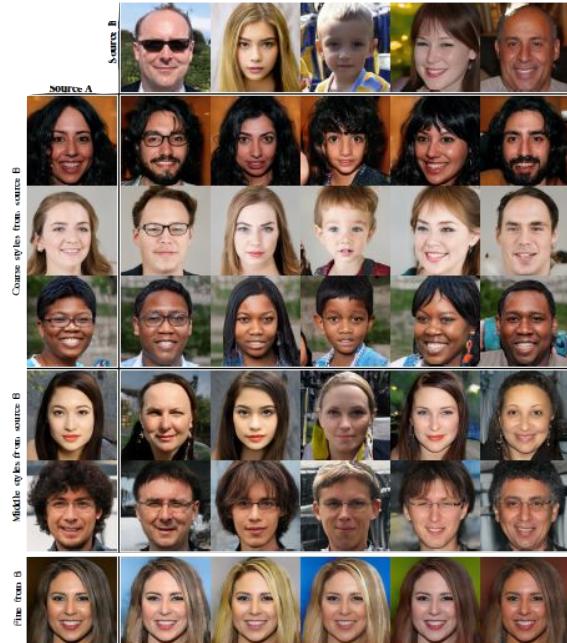


# Modelos de difusión



# Modelos generativos

StyleGAN 3



DALL-E 2  
Stable  
Diffusion



# De lo general a lo especializado

Módulos 2 y 3: datos genéricos/no estructurados (e.g. redes neuronales totalmente conectadas)

*Deep Learning*: muchas capas, pero también redes más especializadas

Procesamiento de imágenes (clasificación, regresión): Imágenes CNN y sus variantes

- Arquitecturas específicas: ResNet, Inception, EfficientNet + nuevos conceptos (vision transformers, normalizing flows)
- Modelos generativos: VAE, GAN y sus variantes, modelos de difusión (vedettes Dall-E, Midjourney or LeonardoAI)

Series temporales: redes recurrentes

Procesamiento del lenguaje natural: recurrentes, transformers

La vedette: LLMs!

Grandes redes, nuevas ideas!



Argentina  
programa  
4.0



Universidad  
Nacional  
de San Martín

---

# Módulo 3

# Aprendizaje Automático

Semana 11.  
Aprendizaje por refuerzo - *Reinforcement learning*

# Aprendizaje por refuerzo

# Aprendizaje Supervisado

## Entrenamiento (y validación)

(después de definir y pre-procesar los datos, elegir método, hiperparámetros, etc.)



función de pérdida (*loss*): comparación de la salida con el target  
entrenamiento = minimización de la *loss*

# Aprendizaje Supervisado

## Entrenamiento (y validación)

(después de definir y pre-procesar los datos, elegir método, hiperparámetros, etc.)



$$\| \text{salida} - \text{target} \|^2 + \text{regularización}$$

entrenamiento = minimización de la

*loss*

# Aprendizaje Supervisado

## Entrenamiento (y validación)

(después de definir y pre-procesar los datos, elegir método, hiperparámetros, etc.)



En ese proceso: evitar sobreajuste, mínimos locales, estimar los errores (e.g. validación cruzada: separación en entrenamiento y validación), usar metricas (e.g. AUC), elegir el *working point*, ajustar hiperparámetros (probar con distintos métodos)

# Aprendizaje Supervisado

## Uso



# Aprendizaje por refuerzo / Reinforcement Learning

- Hasta ahora: **utilizar** o **crear** datos
  - a partir de datos etiquetados o sin etiquetar, hacer inferencias o aprender a generar nuevos datos
- ¿Es ésta la forma **natural** de aprender?
  - la mayor parte del aprendizaje implica **intento y error**
  - **actuamos** sobre los datos, generando **nuevos** datos
- ¡Aprender es también **ser capaz de interactuar con los datos!**

# Aprendizaje por refuerzo / Reinforcement Learning

- Rama del aprendizaje automático que trata de cómo aprender estrategias de control para **interactuar** con un entorno complejo
- Marco para aprender a interactuar con el entorno a partir de la **experiencia**
- Inspirado en la biología: **refuerza** el buen comportamiento con recompensas

# Aprendizaje supervisado x no supervisado x por refuerzo

## Aprendizaje supervisado

- **Datos:**  $(x, y)$

$x$  son los datos,  $y$  las etiquetas

- **Objetivo:** aprender la función que mapea

$$x \rightarrow y$$

- **Ejemplos:** clasificación, regresión, detección de objetos, segmentación semántica, etc.

## Aprendizaje no supervisado

- **Datos:**  $x$

$x$  son los datos

- **Objetivo:** aprender la estructura **oculta** o **subyacente** de los datos

- **Ejemplos:** agrupación, reducción de la dimensionalidad, **autoencoders**, etc.

## Aprendizaje por refuerzo

- **Datos:** pares **estado-acción**

- **Objetivos:** maximizar las recompensas futuras a lo largo de muchos pasos de tiempo

- **Ejemplos:** : juegos, coches autónomos, etc.

# Human-level control through deep reinforcement learning

Volodymyr Mnih<sup>1\*</sup>, Koray Kavukcuoglu<sup>1\*</sup>, David Silver<sup>1\*</sup>, Andrei A. Rusu<sup>1</sup>, Joel Veness<sup>1</sup>, Marc G. Bellemare<sup>1</sup>, Alex Graves<sup>1</sup>, Martin Riedmiller<sup>1</sup>, Andreas K. Fidjeland<sup>1</sup>, Georg Ostrovski<sup>1</sup>, Stig Petersen<sup>1</sup>, Charles Beattie<sup>1</sup>, Amir Sadik<sup>1</sup>, Ioannis Antonoglou<sup>1</sup>, Helen King<sup>1</sup>, Dharshan Kumaran<sup>1</sup>, Daan Wierstra<sup>1</sup>, Shane Legg<sup>1</sup> & Demis Hassabis<sup>1</sup>

The theory of reinforcement learning provides a normative account<sup>1</sup>, deeply rooted in psychological<sup>2</sup> and neuroscientific<sup>3</sup> perspectives on animal behaviour, of how agents may optimize their control of an environment. To use reinforcement learning successfully in situations approaching real-world complexity, however, agents are confronted with a difficult task: they must derive efficient representations of the environment from high-dimensional sensory inputs, and use these to generalize past experience to new situations. Remarkably, humans and other animals seem to solve this problem through a harmonious combination of reinforcement learning and hierarchical sensory processing systems<sup>4,5</sup>, the former evidenced by a wealth of neural data revealing notable parallels between the phasic signals emitted by dopaminergic neurons and temporal difference reinforcement learning algorithms<sup>3</sup>. While reinforcement learning agents have achieved some successes in a variety of domains<sup>6–8</sup>, their applicability has previously been limited to domains in which useful features can be handcrafted,

agent is to select actions in a fashion that maximizes cumulative future reward. More formally, we use a deep convolutional neural network to approximate the optimal action-value function

$$Q^*(s, a) = \max_{\pi} \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots | s_t = s, a_t = a, \pi],$$

which is the maximum sum of rewards  $r_t$  discounted by  $\gamma$  at each time-step  $t$ , achievable by a behaviour policy  $\pi = P(a|s)$ , after making an observation ( $s$ ) and taking an action ( $a$ ) (see Methods)<sup>19</sup>.

Reinforcement learning is known to be unstable or even to diverge when a nonlinear function approximator such as a neural network is used to represent the action-value (also known as  $Q$ ) function<sup>20</sup>. This instability has several causes: the correlations present in the sequence of observations, the fact that small updates to  $Q$  may significantly change the policy and therefore change the data distribution, and the correlations between the action-values ( $Q$ ) and the target values  $r + \gamma \max_{a'} Q(s', a')$ .

# Control a nivel humano mediante el aprendizaje profundo por refuerzo

La teoría del aprendizaje por refuerzo ofrece una explicación normativa, **profundamente arraigada en las perspectivas psicológicas y neurocientíficas del comportamiento animal**, sobre cómo los **agentes** pueden optimizar su control de un **entorno**. Sin embargo, para utilizar con éxito el aprendizaje por refuerzo en situaciones cercanas a la complejidad del mundo real, los agentes se enfrentan a una tarea difícil: deben derivar **representaciones eficientes del entorno a partir de entradas sensoriales de alta dimensión**, y utilizarlas para **generalizar** la experiencia pasada a nuevas situaciones.

Sorprendentemente, los seres humanos y otros animales parecen resolver este problema mediante una combinación armoniosa de sistemas de aprendizaje por refuerzo y de procesamiento sensorial jerárquico, lo que se pone de manifiesto en una gran cantidad de datos neuronales que revelan notables paralelismos entre las señales fáscicas emitidas por las neuronas dopamínergicas y los algoritmos de aprendizaje por refuerzo por diferencia temporal. Aunque los agentes de aprendizaje por refuerzo han logrado algunos éxitos en una variedad de dominios, su aplicabilidad se ha limitado previamente a dominios en los que se pueden crear características útiles a mano, o a dominios con espacios de estado completamente observados y de baja dimensión. Aquí utilizamos los recientes avances en el **entrenamiento de redes neuronales profundas** para desarrollar un nuevo agente artificial, denominado **red Q profunda**, que puede aprender **políticas** exitosas directamente a partir de entradas sensoriales de alta dimensión utilizando el **aprendizaje de refuerzo de punta a punta**. Probamos este agente en el desafiante dominio de los juegos clásicos de Atari 2600. Demostramos que el agente de la red Q profunda, **recibiendo sólo los píxeles y la puntuación del juego como entradas, fue capaz de superar el rendimiento de todos los algoritmos anteriores y alcanzar un nivel comparable al de un probador de juegos humano profesional en un conjunto de 49 juegos, utilizando el mismo algoritmo, la arquitectura de la red y los hiperparámetros**. Este trabajo tiende un puente entre las entradas sensoriales de alta dimensión y las acciones, dando como resultado el primer agente artificial capaz de aprender a sobresalirse en un conjunto diverso de tareas desafiantes.

# Conceptos clave del aprendizaje por refuerzo



**Recompensa:** retroalimentación que mide el éxito o el fracaso de la acción de los agentes

**Ejemplo:** coche autodirigido

**Agente:** vehículo

**Estado:** cámaras

**Acción:** controlar el volante, acelerar, etc.

**Recompensa:** distancia recorrida

# Clave: la función $Q$

**Recompensa total:** suma descontada de todas las recompensas obtenidas a partir del momento  $t$

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots$$

La función  $Q$  captura la **recompensa total futura esperada** que recibe un agente en el estado  $s$  al ejecutar una determinada acción  $a$ :

$$Q(s_t, a_t) = \mathbb{E}[R_t | s_t, a_t]$$

¿Cómo tomar acciones dado  $Q$ ?

El agente necesita una política para inferir la mejor acción a tomar en su estado  $s$

La política debe elegir una acción que maximice la recompensa futura

---

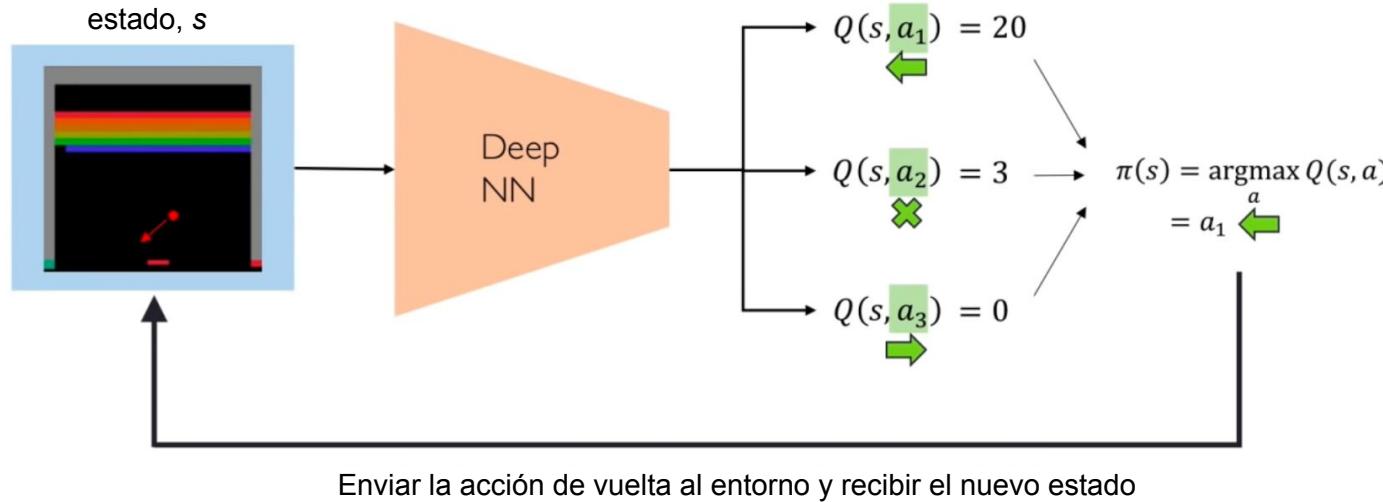
$$\text{política } \pi^*(s) = \operatorname{argmax}_a Q(s, a)$$

---

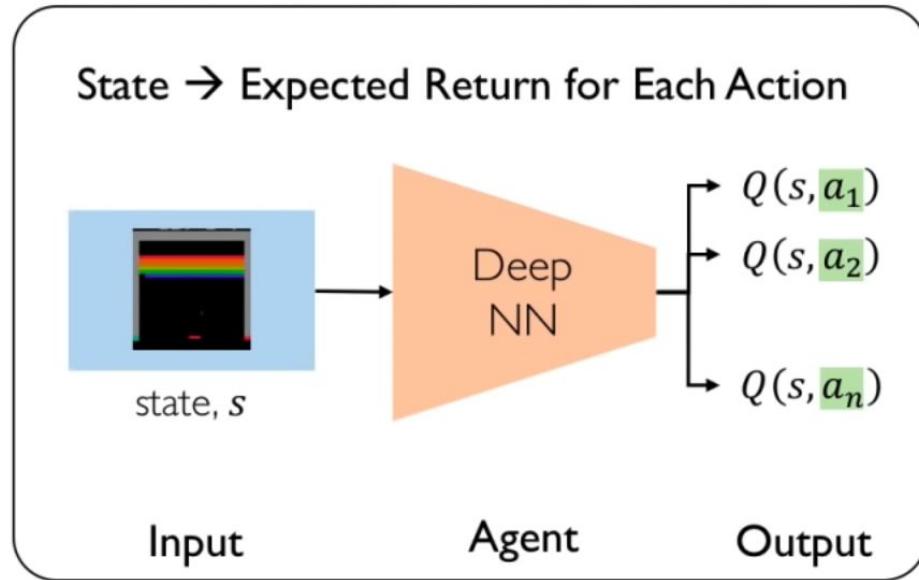


# Redes Q profundas (DQN): Entrenamiento

Utilizar una red neuronal para aprender la función  $Q$  y luego utilizarla para inferir la política óptima



# Redes Q profundas (DQN): Entrenamiento



Aprendizaje  
del valor

$$\textbf{Q-loss} \quad \mathcal{L} = \mathbb{E} \left[ \left\| \underbrace{\left( r + \gamma \max_{a'} Q(s', a') \right)}_{\text{target}} - \underbrace{Q(s, a)}_{\text{predicted}} \right\|^2 \right]$$

# Problemas con el *Q*-aprendizaje

- **Complejidad:**

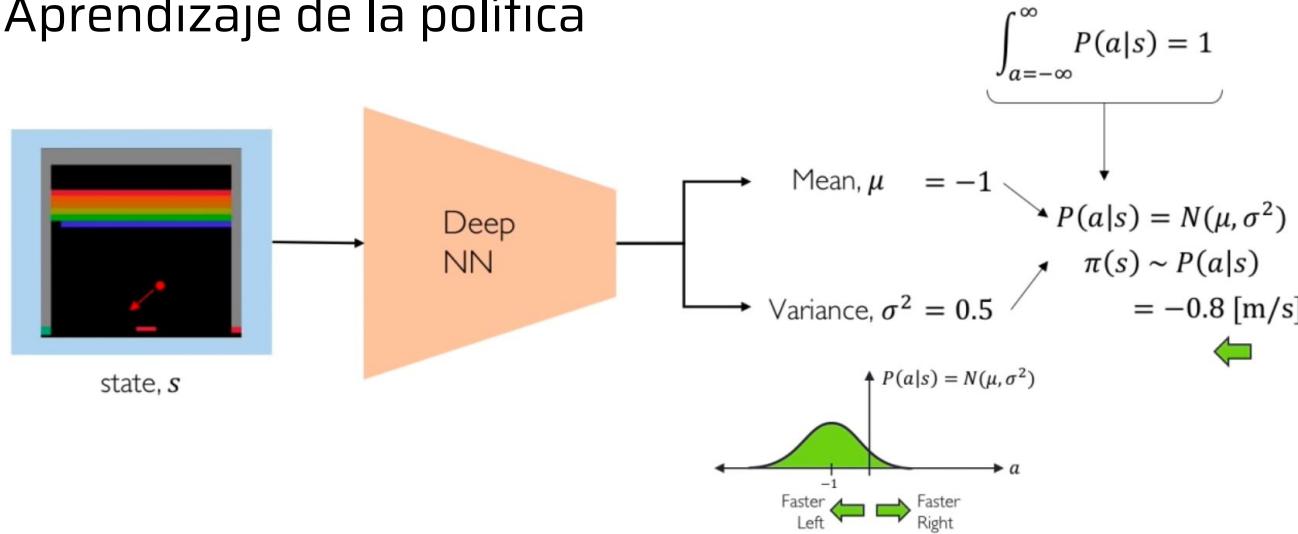
- puede modelar escenarios donde el espacio de acción es discreto y pequeño
- no puede manejar espacios continuos

- **Flexibilidad:**

- La política se calcula de forma determinista a partir de la función  $q$  maximizando la recompensa: no puede aprender políticas probabilísticas

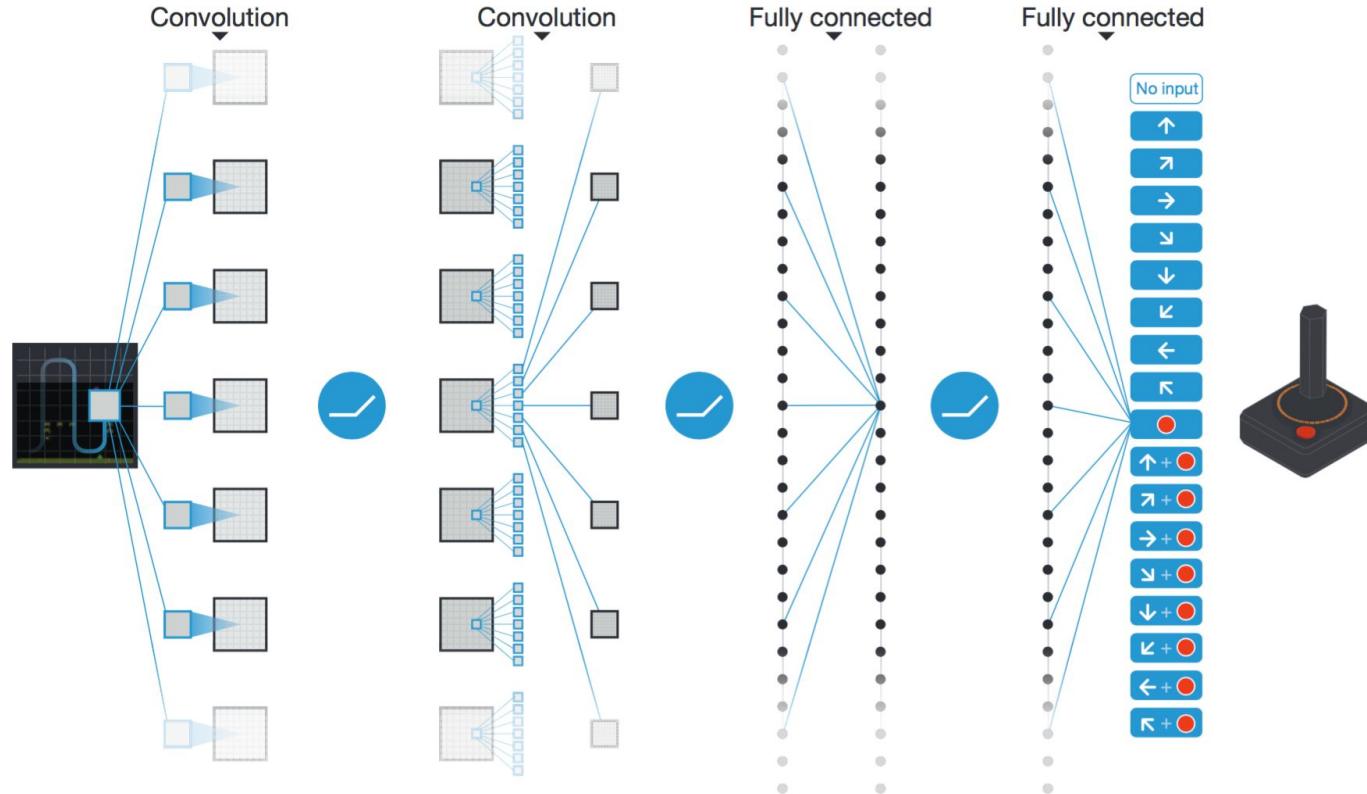
# Gradiente de política: Idea clave

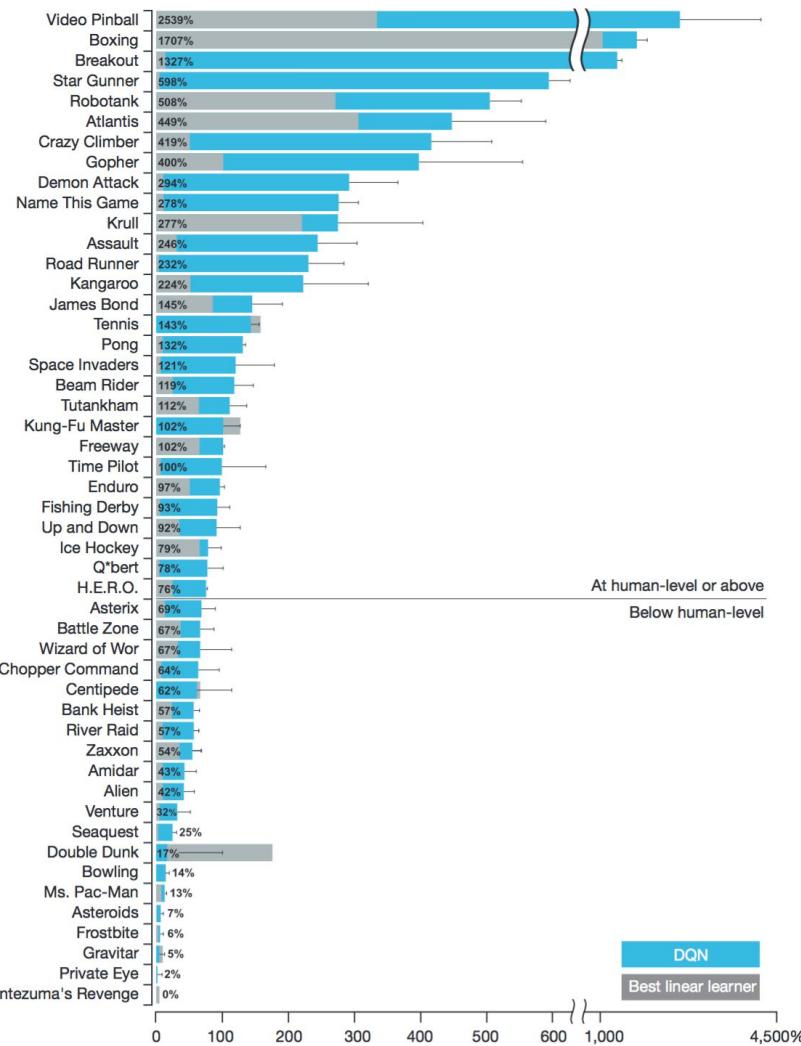
## Aprendizaje de la política



**Gradiente político:** permite modelar un espacio de acción continuo

# La “máquina”





# Resumen del aprendizaje por refuerzo profundo (*Deep Reinforcement Learning*)

- Fundamentos:
  - Agentes que actúan en el entorno
  - Pares estado-acción: maximizar las recompensas futuras
  - Descuento
- Aprendizaje ***Q***
  - Función ***Q***: recompensa total esperada dada ***s, a***
  - Política determinada por la selección de la acción que maximiza la función ***Q***
- Gradiente de la política
  - Aprender y optimizar la política directamente
  - Aplicable a espacios de acción continuos

# Resumen del aprendizaje por refuerzo

- Aprendizaje semi-supervisado ("pérdida externa"), pero cualitativamente muy diferente
- En general el aprendizaje automático solía ser para una tarea específica
- El refuerzo es más flexible en el aprendizaje de las reglas a través de la interacción con el sistema
- Más cerca del concepto de aprendizaje de los animales
- Más cercano a la IA
- Se puede entrenar utilizando jugadas anteriores
- Se puede entrenar de forma puramente autónoma (prueba y error)
- Puede entrenarse mediante simulaciones (¡GAN!) y luego aplicarse al mundo real
- Ejemplos: Go, coches autodirigidos, comercio

# Resumen - Discusión

- Hemos recorrido un largo y arduo camino hasta aquí
- ¡Los logros no han sido menores!
- Es increíble que haya sido posible escudriñar el aprendizaje automático en tan poco tiempo
- Curso práctico de aprendizaje supervisado con ejemplos concretos de código y aplicaciones reales
- Base para aplicaciones reales, en la escalas "industriales" y desarrollos de soluciones propias
- Bases para entender las aplicaciones más sofisticadas, de vanguardia, innovadoras
  - Redes más complejas y especializadas
  - ¡Muchas públicas! [e.g. <https://huggingface.co/>] (otras de código propietario)
  - Requieren supercomputadoras para entrenar (+ conjuntos de datos, etc.), ¡pero no para usar!
- *Deep learning* es quizás la faceta más increíble del aprendizaje automático, lo más próximo de IA y seguramente la rama con más *hype*, pero no es la solución para todos los problemas.
- Aplicaciones visuales y “recreativas” x aplicaciones y desarrollos académicos e industriales
- Desarrollos de los últimos ~< 10 años!
- Muchas veces, lo que llamamos en este curso aprendizaje automático, se lo denomina inteligencia artificial, lo que le dá mucho más *hype*
- *Muchas gracias por la participación de todo/as!*

# Lo que no tocamos en este curso

- Hardware (presente y futuro!)
- Cuestiones éticas (e.g., cezgo algorítmico)
- Cuestiones filosóficas (qué es IA, que és pensar, tener auto-conciencia, etc.)
- Previsiones futurísticas
- Aplicaciones específicas

**Fin**

# Buenas referencias en Transformadores / Transformers

- Transformers - Self-Attention to the rescue:  
<https://www.dominodatalab.com/blog/transformers-self-attention-to-the-rescue>
- Transformer Neural Networks: A Step-by-Step Breakdown:  
<https://builtin.com/artificial-intelligence/transformer-neural-network>
- Attention Is All You Need: <https://arxiv.org/abs/1706.03762>
- Effective Approaches to Attention-based Neural Machine Translation:  
<https://arxiv.org/abs/1508.04025>
- Temporal Fusion Transformer: Time Series Forecasting with Deep Learning — Complete Tutorial  
<https://towardsdatascience.com/temporal-fusion-transformer-time-series-forecasting-with-deep-learning-complete-tutorial-d32c1e51cd91>
- Visual transformer: [https://www.youtube.com/watch?v=HZ4j\\_U3FC94](https://www.youtube.com/watch?v=HZ4j_U3FC94)
- Generadores de imagen:
- How to Run ERNIE-ViLG AI Art Generator in Google Colab Free:  
<https://bytedd.com/how-to-run-ernie-vilg-ai-art-generator-in-google-colab-free/>
- Google's DreamFusion AI Generates 3D Model From Text
- Goodfellow GAN paper: Generative Adversarial Nets, <https://arxiv.org/pdf/1406.2661.pdf>

# Links reinforcement learning

- Alpha Go: <https://www.youtube.com/watch?v=WXuK6gekU1Y>
- Hide & Seek (sin desperdicios): <https://www.youtube.com/watch?v=kopoLzvh5jY>