



Comunidad de  
Supercómputo



Σύνταγμα  
Syntagma

## Neuroevolución

# Una alternativa para simplificar los modelos asociados al Aprendizaje Profundo

Presenta: Mtro. José-Clemente Hernández-Hernández  
jclementehdzhdz@gmail.com

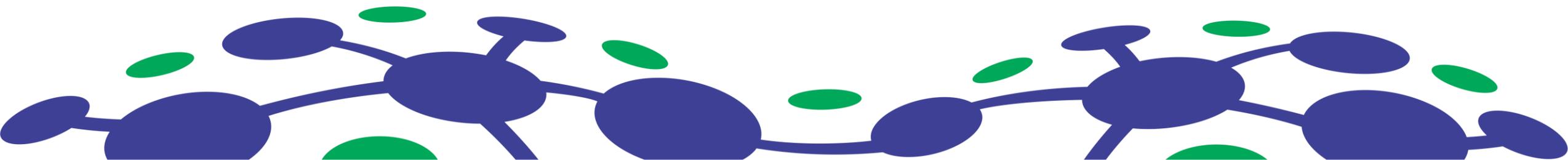
Instituto de Investigaciones en Inteligencia Artificial  
Universidad Veracruzana

03 de agosto del 2022

# Itinerario

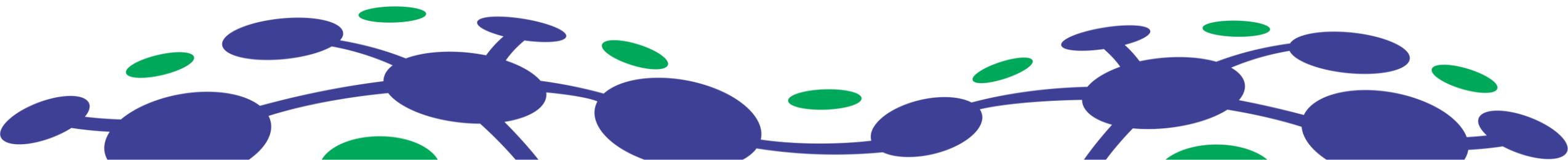
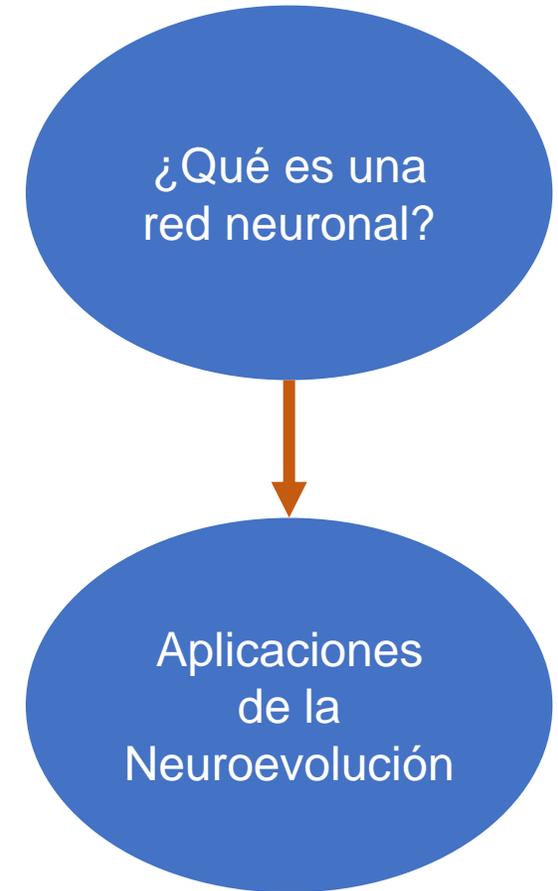
¿Qué es una red neuronal?

Aplicaciones de la Neuroevolución



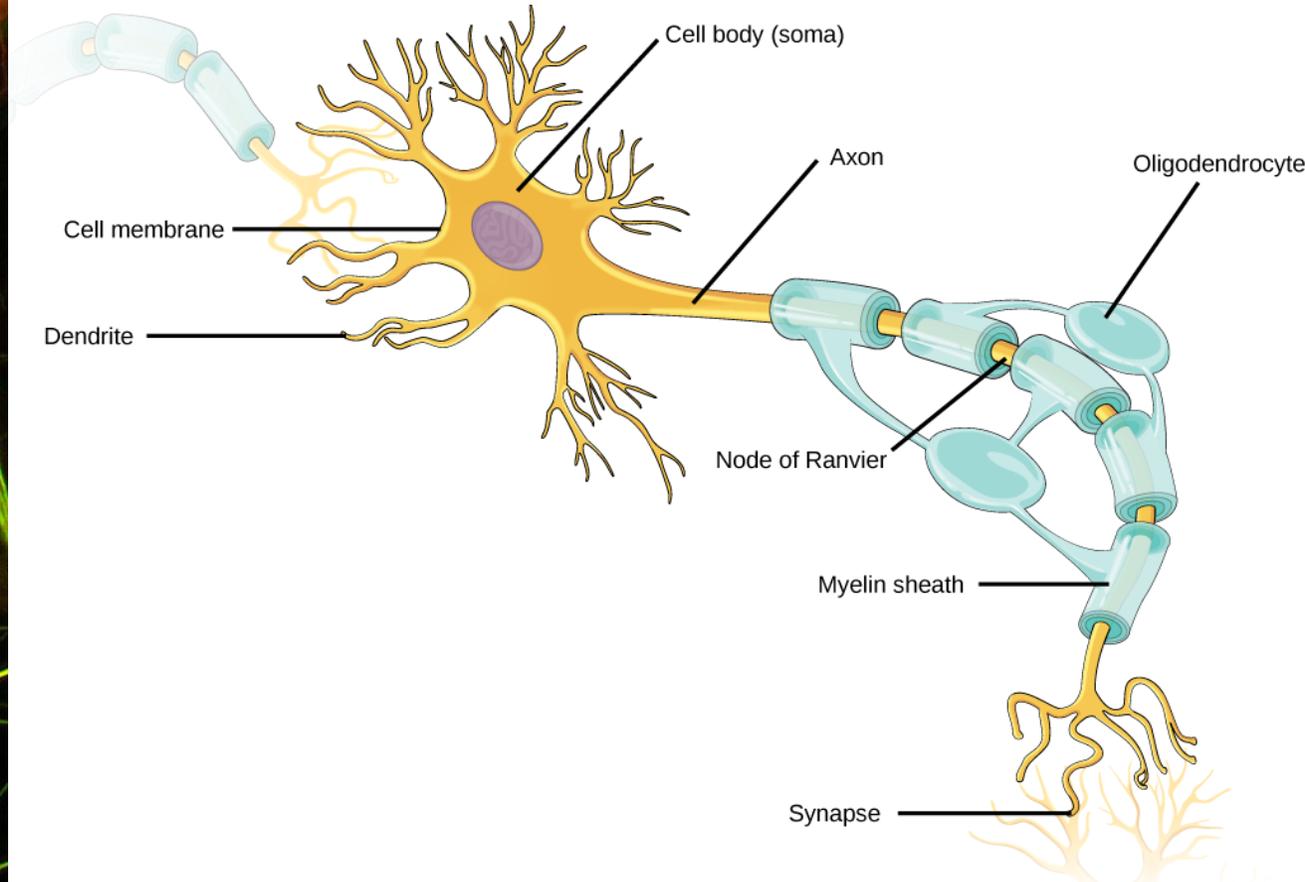
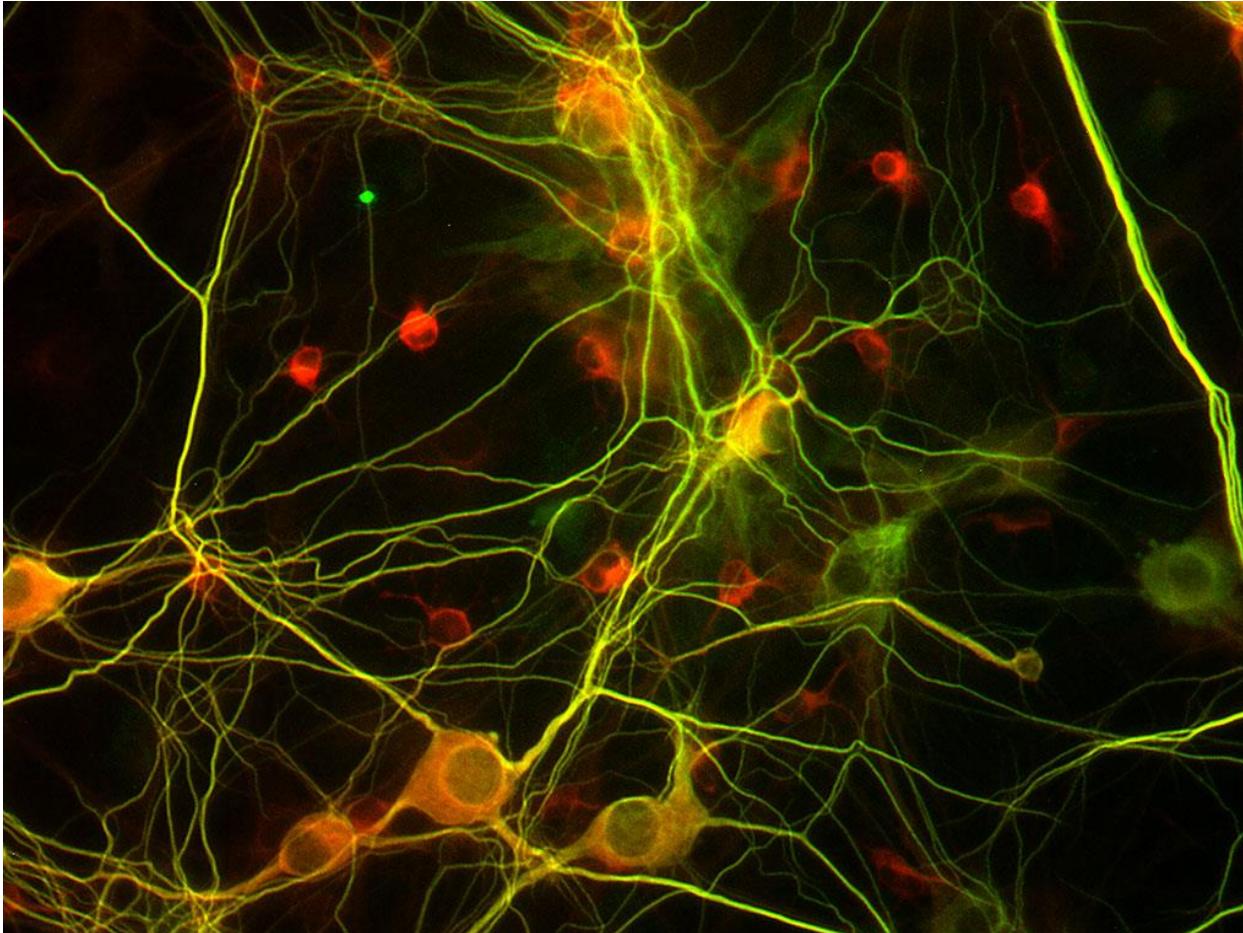
# Agenda

- ¿Qué es una neurona?
- Un poco de historia de las redes neuronales artificiales
- Tipos de redes neuronales artificiales
- Revisión de modelos de Aprendizaje Profundo
  
- ¿Qué es la Computación Evolutiva?
- Algoritmos de Cómputo Evolutivo
- ¿Qué es la Neuroevolución?
- Aplicaciones de la Neuroevolución



# La neurona biológica

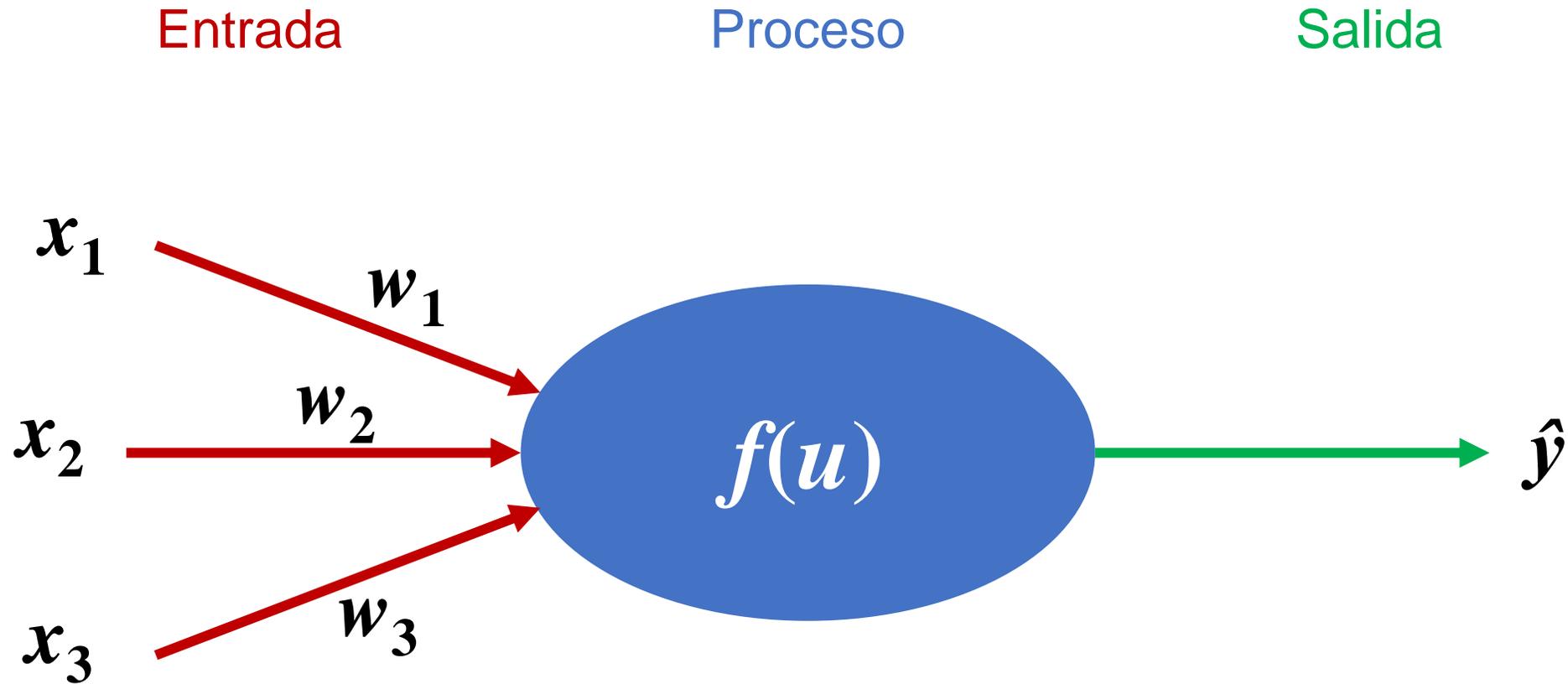
## Introducción



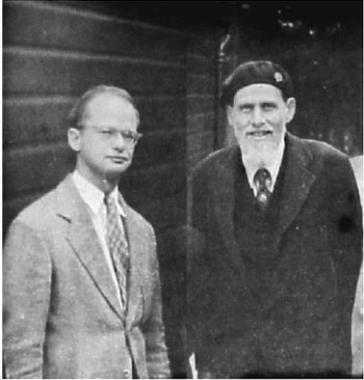
Neuronas del hipocampo de rata  
Fuente: <https://bit.ly/3PTNnnn> (bioscience.lonza.com)

Partes de una neurona  
Fuente: <https://bit.ly/3OxhQ9P>

# La neurona artificial

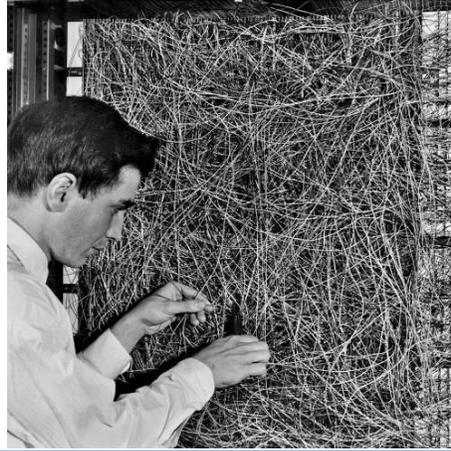


# Un poco de historia...



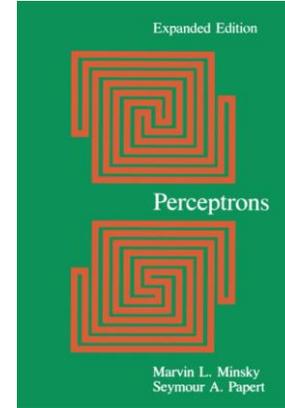
1943

Warren McCulloch y Walter Pitts desarrollan los primeros modelos de neuronas artificiales



1958

Frank Rosenblatt realizó el primer modelo de un perceptrón como un dispositivo electrónico

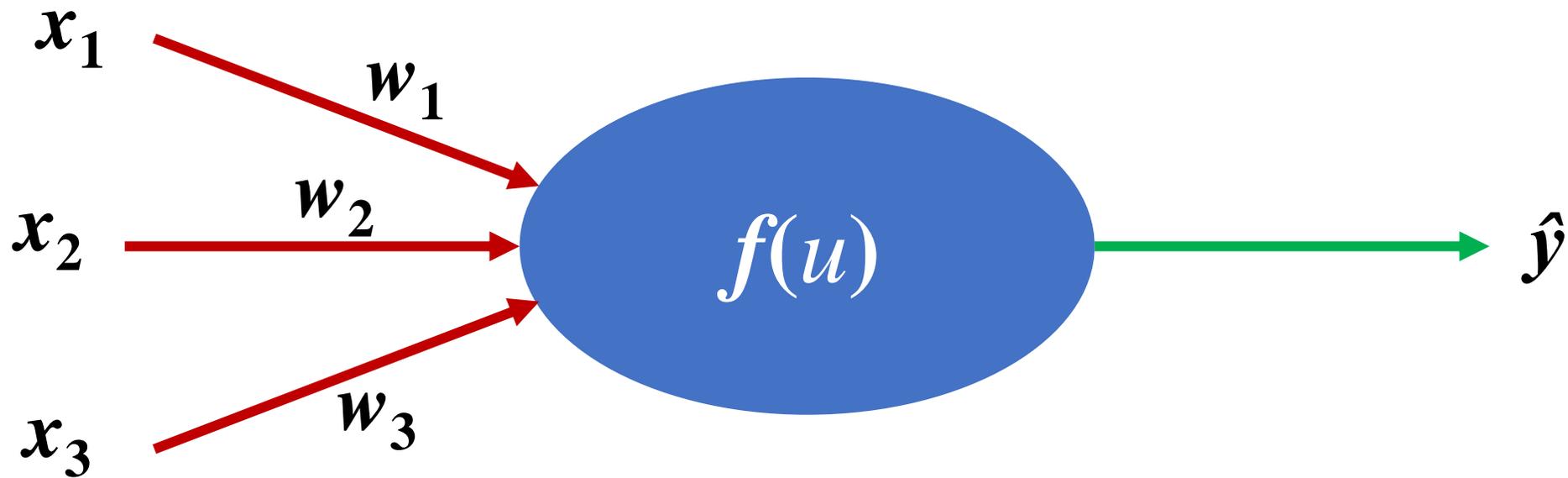


1969

Marvin Minsky y Seymour Papert publican Perceptrons

# La función

$$u = \sum x_i w_i \quad f(u) = \begin{cases} 1, & \text{si } u \geq 0 \\ 0, & \text{si } u < 0 \end{cases}$$



# Más historia...



1974

Paul Werbos desarrolla el algoritmo de retropropagación (backpropagation)



1980

Kunihiko Fukushima introduce Neocognitron, inspiración de las redes convolucionales

$$\frac{\partial E}{\partial W}$$

1986

Rumelhart y otros, redescubren el algoritmo de Werbos y lo aplican en las redes neuronales

Wang, Haohan, y Bhiksha Raj. (2017). On the Origin of Deep Learning. arXiv:1702.07800, arXiv. arXiv.org, <http://arxiv.org/abs/1702.07800>.

# Otras funciones de activación

Sigmoide

$$f(\mathbf{u}) = \frac{1}{1 + e^{-u}}$$

Tangente hiper.

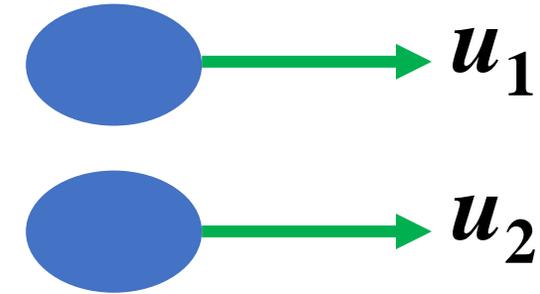
$$f(\mathbf{u}) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

ReLU

$$f(\mathbf{u}) = \max(0, u)$$

Softmax

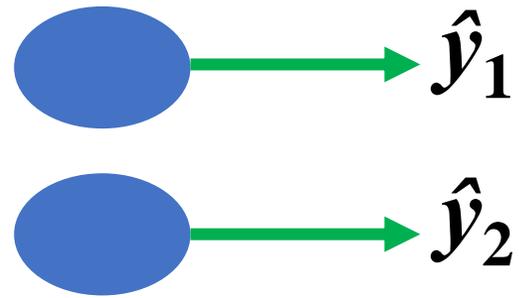
$$f(\mathbf{u}_i) = \frac{e^{u_i}}{\sum e^{u_j}}$$



# Funciones de pérdida

Salida de la red

Salida esperada



$y_1$

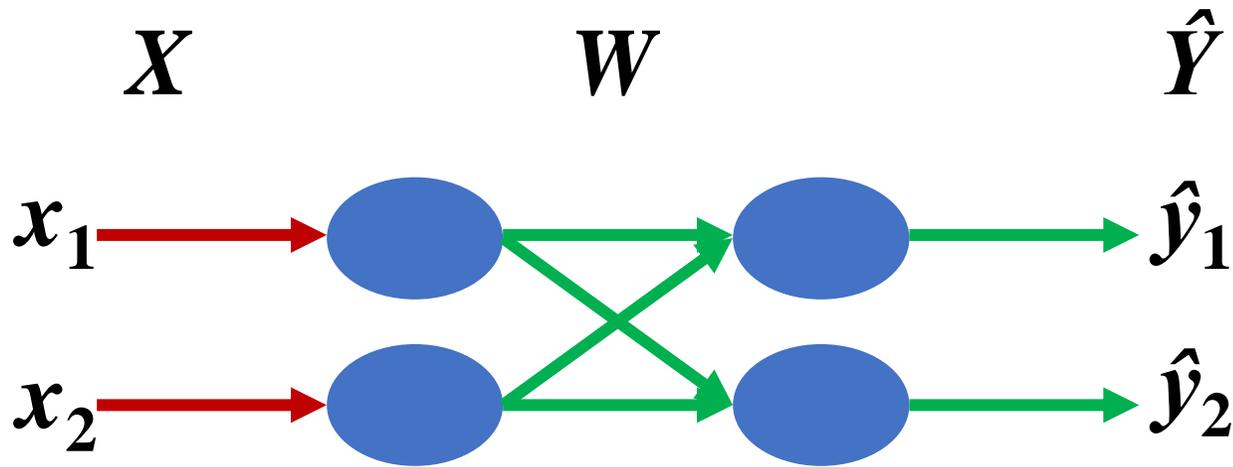
$y_2$

$$E(\hat{Y}) = \frac{1}{n} \sum (\hat{y}_i - y_i)^2$$

# Retropropagación

$$u(X) = W^T X$$

$$E(f(u(X)))$$



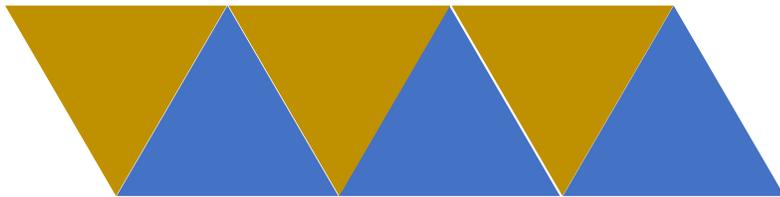
$$\frac{\partial E}{\partial W}$$

$$E(f(u(f(u(f(u(X)))))))$$



# Uso de las redes neuronales

- Procesamiento de imágenes
- Procesamiento de texto
- Reconocimiento de patrones



- Normalmente entrenadas un cierto número de épocas utilizando el algoritmo de retropropagación, y dividiendo la base de datos en conjuntos de prueba y entrenamiento
- Se usan para aprendizaje supervisado, no supervisado, por refuerzo y semi supervisado, entre otros
- El Aprendizaje Profundo es el uso de redes neuronales de tamaños más grandes



# Redes neuronales

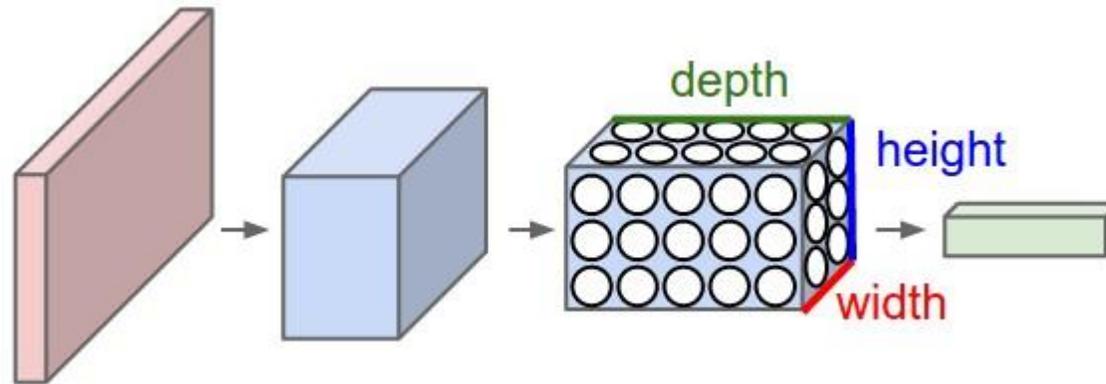
Tipos y modelos

- Redes neuronales feed-forward y fully-connected
- Redes neuronales convolucionales
- Redes neuronales recurrentes y LSTM
- Redes neuronales autoencoders y encoder-decoders
- Redes neuronales generativas
- Redes neuronales de tipo transformer



# Redes neuronales convolucionales

- Se agregan filtros convolucionales y filtros de pooling
- Se conforman por capas convolucionales y una de clasificación fully-connected
- Normalmente utilizadas para procesamiento de imágenes

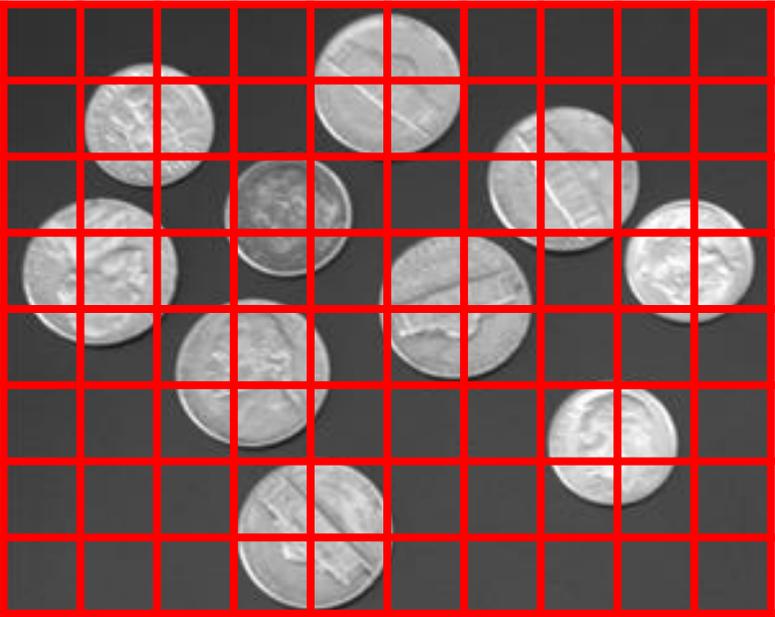


Red neuronal convolucional

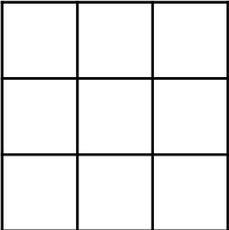
Fuente: <https://cs231n.github.io/convolutional-networks/>

# Una imagen y un filtro convolucional

*I*



*f*



# Campo receptivo

*f*


*f*

4	6	6
2	8	4
4	5	3



# Operación de convolución

*f*

2	2	2	7	7	3	6	3	3	3
6	1	9	8	3	9	9	8	5	1
4	9	2	4	1	4	5	7	2	2
8	6	2	4	2	6	6	6	8	9
1	7	9	9	3	4	8	7	3	8
4	8	4	8	1	7	5	4	9	2
6	5	8	2	2	7	1	1	3	1
6	7	5	1	1	4	9	8	9	9

*f*

4	6	6
2	8	4
4	5	3

$$2(4) + 2(6) + 2(6) + 6(2) + 1(8) + 9(4) + 4(4) + 9(5) + 2(3) = 155$$

155									



# “Brinco” del filtro

*f*

2	2	2	7	7	3	6	3	3	3
6	1	9	8	3	9	9	8	5	1
4	9	2	4	1	4	5	7	2	2
8	6	2	4	2	6	6	6	8	9
1	7	9	9	3	4	8	7	3	8
4	8	4	8	1	7	5	4	9	2
6	5	8	2	2	7	1	1	3	1
6	7	5	1	1	4	9	8	9	9

*f*

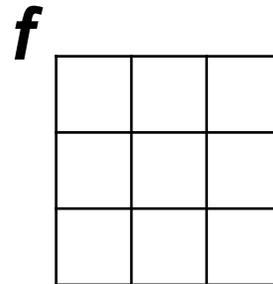
4	6	6
2	8	4
4	5	3

$$4(4) + 9(6) + 2(6) + 1(2) + 3(8) + 1(4) + 8(4) + 9(5) + 9(3) = 216$$

155	226	217	197	235	250	223	152
240	202	176	180	230	286	266	197
220	216	172	157	190	249	242	231
242	258	205	180	204	261	259	263
261	282	232	169	208	212	188	204
246	242	138	149	191	193	223	216

# El filtro de pooling

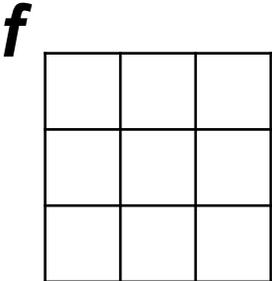
155	226	217	197	235	250	223	152
240	202	176	180	230	286	266	197
220	216	172	157	190	249	242	231
242	258	205	180	204	261	259	263
261	282	232	169	208	212	188	204
246	242	138	149	191	193	223	216



Max ó Avg

# Operación de pooling

155	226	217	197	235	250	223	152
240	202	176	180	230	286	266	197
220	216	172	157	190	249	242	231
242	258	205	180	204	261	259	263
261	282	232	169	208	212	188	204
246	242	138	149	191	193	223	216



Max

240	226	235	286	286	286
258	258	230	286	286	286
282	282	232	261	261	263
282	282	232	261	261	263



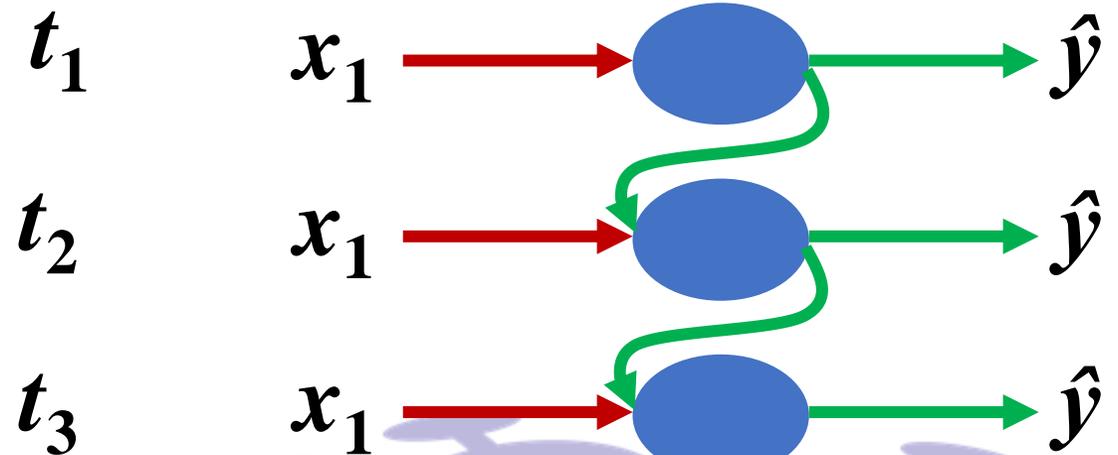
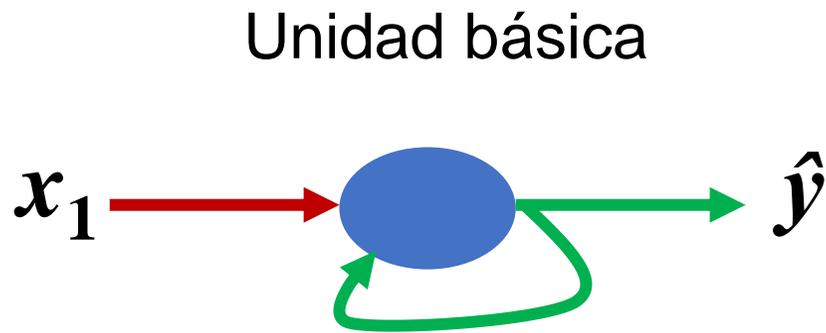
# Modelos convolucionales

- DenseNet
- ResNet
- Redes generativas
- TextCNN
- U-Net
- LeNet
- VGG
- AlexNet
- CLIP (ResNet50 + Vision Transformer)



# Redes neuronales recurrentes

- Una red recurrente tiene neuronas con conexiones hacia sí mismas
- Son utilizadas para análisis de tiempo y procesamiento del lenguaje natural



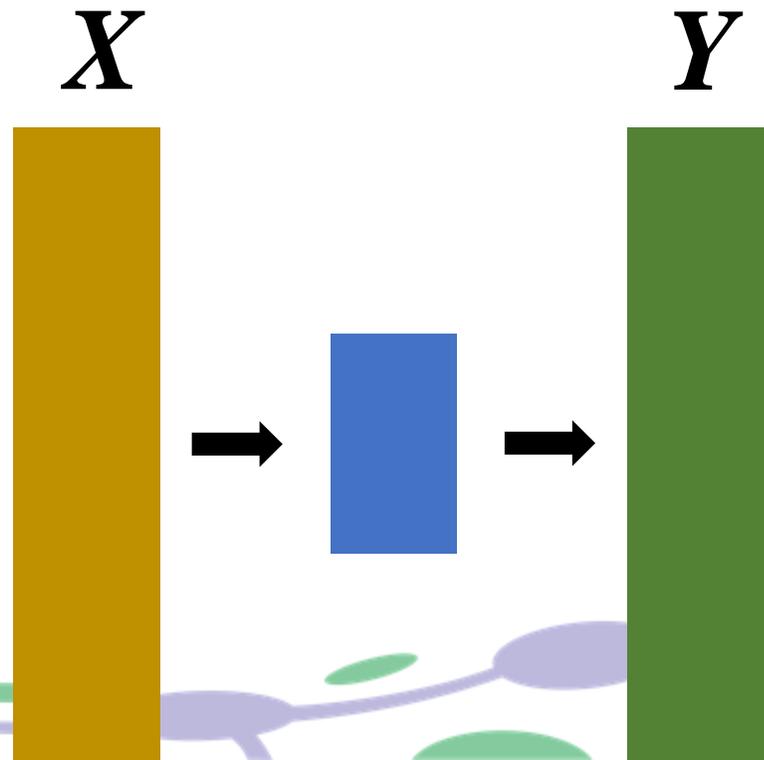
# Modelos recurrentes

- LSTM bidireccional
- LSTM integrada en otros modelos como en los convolucionales
- ELMo



# Redes neuronales de tipo autoencoder

- Utilizadas para comprimir información
- Representan datos en dimensiones más pequeñas
- Arquitecturas que se pueden adaptar con diferentes tipo de neuronas y funciones



# Modelos de autoencoders

- Word2Vec
- GloVe
- Skip-Gram y C-BOW

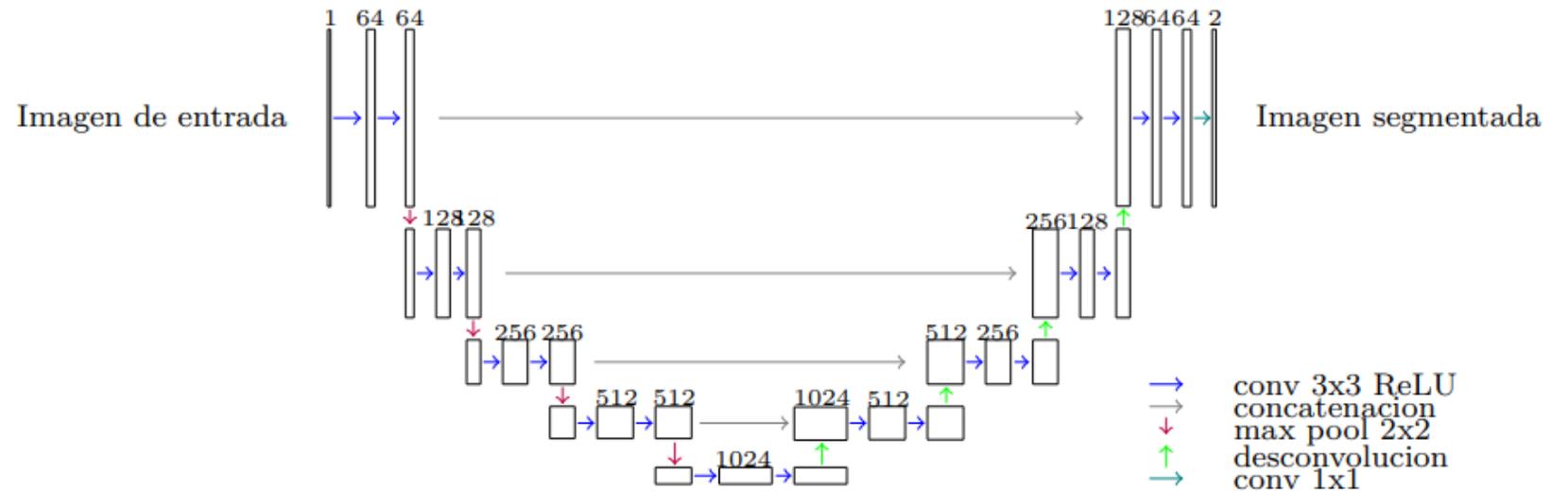


# Redes neuronales encoder-decoder

- Son diferentes a los autoencoders
- Los autoencoders son un tipo especial de arquitectura encoder-decoder

Modelos:

- U-Net
- Dalle-2  
(CLIP + CLIP inverso)
- Transformers



Arquitectura U-Net  
Imagen tomada de:

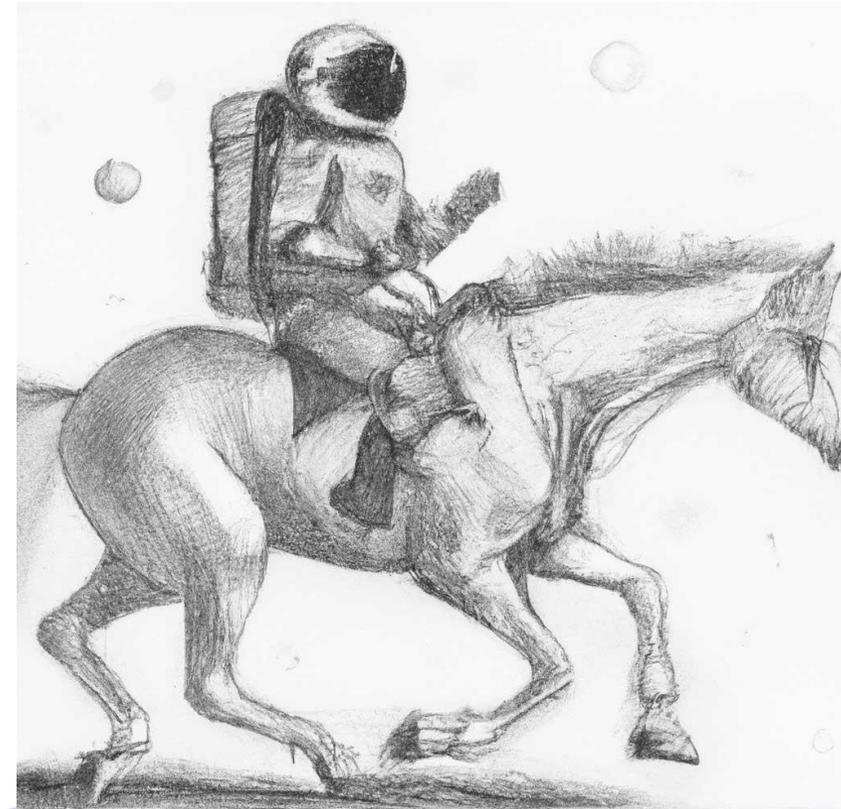
<https://www.uv.mx/mia/files/2022/07/tesis-JoseAntonioFuentesTomas-2022.pdf>

# Redes neuronales generativas

- Utilizadas para generar datos de manera artificial
- Imágenes o texto, o lo que se nos ocurra

Modelos:

- GANs
- DCGANS
- Dalle-2
- GPTs



Astronauta montando a caballo

Fuente: <https://openai.com/dall-e-2/#demos>

# Redes neuronales transformers

Tipos y modelos

- Incluyen neuronas de auto-atención
- Son modelos que han llevado al Aprendizaje Profundo a aumentar el número de parámetros de sus arquitecturas, y a hacer cosas más “interesantes”
- Normalmente utilizados para procesamiento del lenguaje natural



# Modelos de transformers

- Vision Transformer
- BERT
- Modelo de lenguaje de GPT-3



# Sumario de modelos y sus parámetros

Parámetros

Nombre	Número de parámetros	Autores
Dalle-2	$3.5 \times 10^9$	Ramesh <i>et al.</i> (2022) [1]
GPT-3	$175 \times 10^9$	Brown <i>et al.</i> (2020) [2]
BERT	$345 \times 10^6$	Devlin <i>et al.</i> (2018) [3]
ResNet18	$11 \times 10^6$	Oh <i>et al.</i> (2020) [4]
ResNet50	$23 \times 10^6$	He <i>et al.</i> (2015) [5]
VGG-16	$148 \times 10^6$	Civit-Masot <i>et al.</i> (2020) [6]
TextCNN (adaptada)	73203	Kim (2014) [7]
U-Net	$31 \times 10^6$	Ronneberger <i>et al.</i> (2015) [8]
Red neuronal McCulloch-Pitts	Para el AND: 3	

# ¿Qué significa mejorar?

Parámetros

- Los modelos de Aprendizaje Profundo están aumentando su tamaño considerablemente
- Pero mejorar, en este ámbito, no siempre significa que el modelo tenga que aumentar el número de parámetros
- ¿Cómo sabemos cuál es el mejor número de parámetros?



# Aumentar el número de parámetros...

Parámetros

Si el número de parámetros sigue aumentando, el entrenamiento de estos modelos tendría consecuencias no solo en el tiempo requerido para entrenar y en los recursos computacionales, sino en el cambio climático



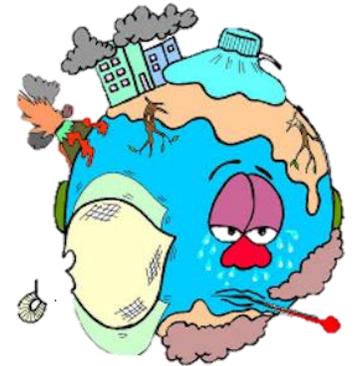
[Esta foto](#) de Autor desconocido está bajo licencia [CC BY](#)

# Ejemplo

Parámetros

GPT-3 fue entrenado durante un estimado de 28,000 días GPU. Y en promedio, esto emitiría un aproximado de 84,738.48kgCO<sub>2</sub>eq gases de efecto invernadero

Esto es equivalente a viajar en coche a través de aproximadamente 704,000km



Anthony, L.F.W., Kanding, B., Selvan, R. (2020). Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models, <http://arxiv.org/abs/2007.03051>

# Alternativas para reducir CO<sub>2</sub>

Parámetros

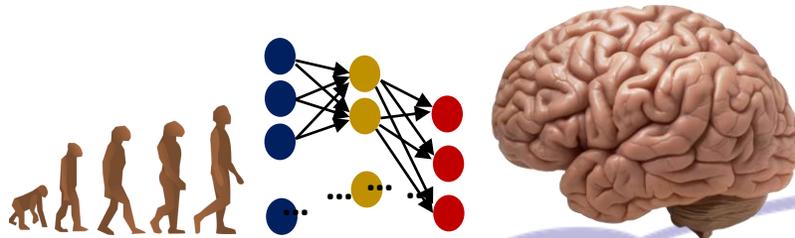
En el mismo artículo mencionan algunas técnicas para efficientar los modelos de Aprendizaje Profundo con respecto a las emisiones de gas:

- Entrenar modelos en regiones donde se emita menor cantidad de CO<sub>2</sub>
- Entrenar modelos cuando en un cierto periodo se emita menos CO<sub>2</sub>
- Técnicas para encontrar hiperparámetros de manera más rápida y menos costosa
- Elección de mejores características de hardware

# Una alternativa más...

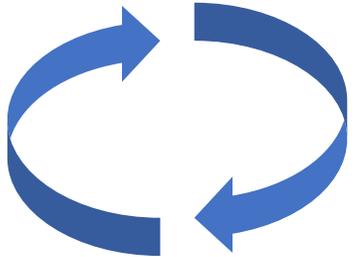
Es posible optimizar el número de parámetros entrenables de las redes neuronales. Y otros parámetros como las funciones de activación y tamaño de las capas de la red. Esto se puede hacer a través de

# NEUROEVOLUCIÓN

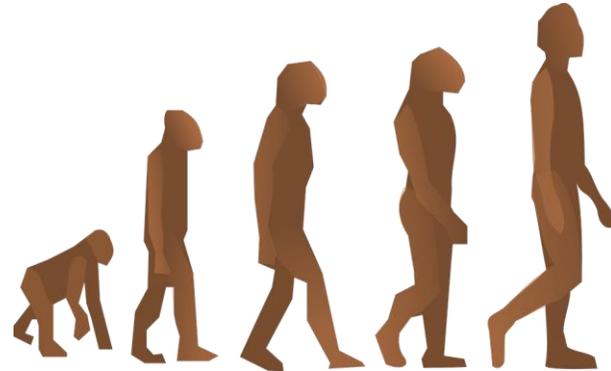


# Cómputo Evolutivo

Neuroevolución



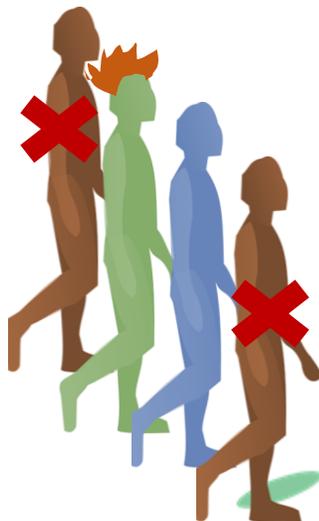
Varias generaciones



Principio de la Evolución



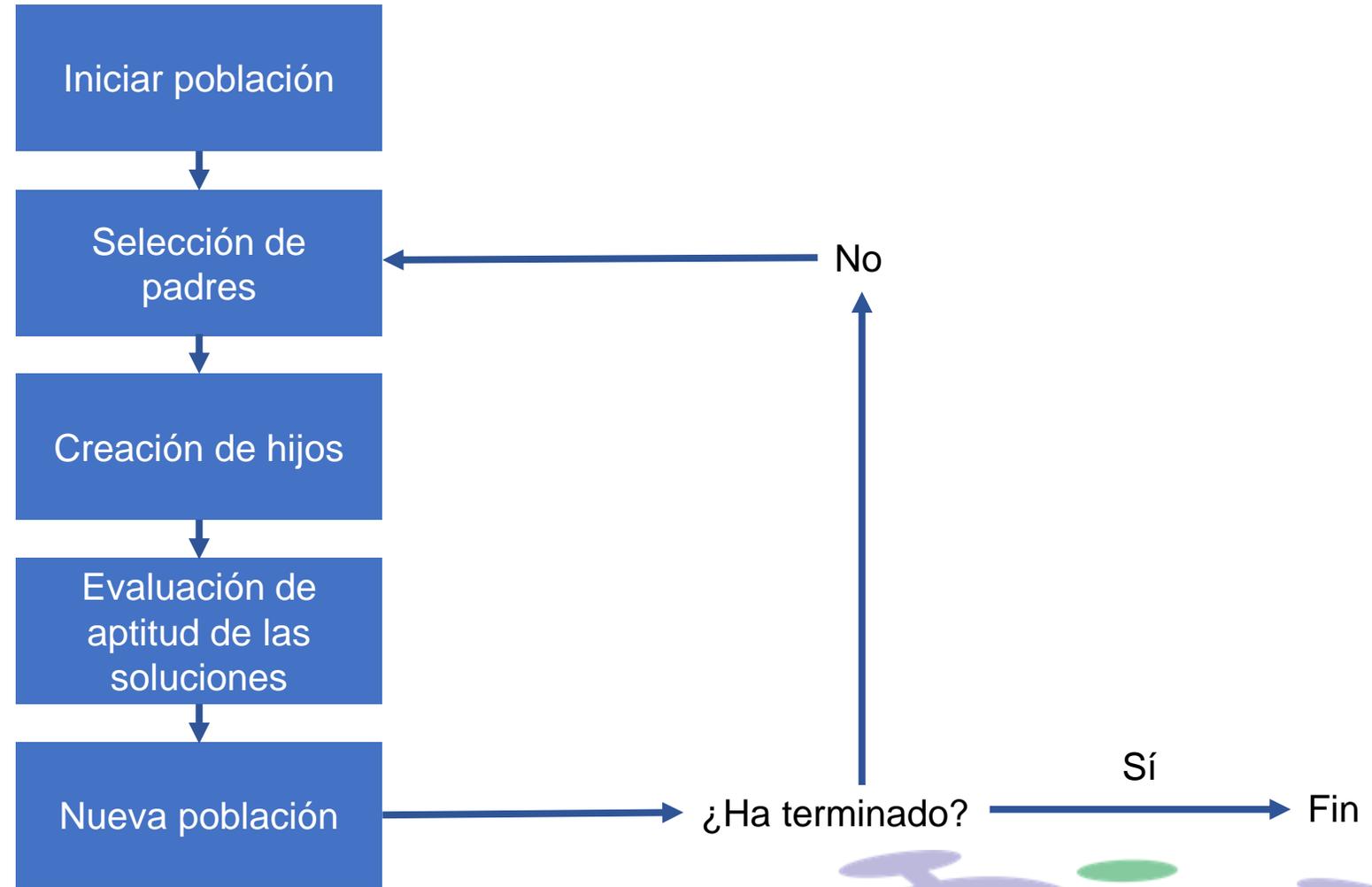
Variaciones aleatorias



Selección natural



# Dentro de la computadora...



# Para resolver problemas...

Problemas de optimización o búsquedas

$$\max f(x)$$

ó

$$\min f(x)$$



# Usando una codificación

Codificaciones (genotipo) de individuos para diferentes problemas:

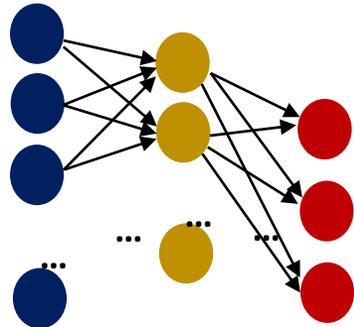
- Binaria 

1	0	1	1	0
---	---	---	---	---
- Continua 

0.33	0.12	0.03	0.9	0.56
------	------	------	-----	------
- Permutaciones 

1	5	4	2	3
---	---	---	---	---
- De grupos 

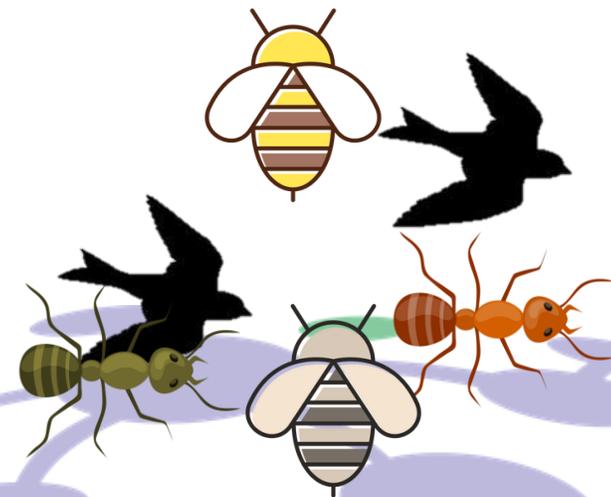
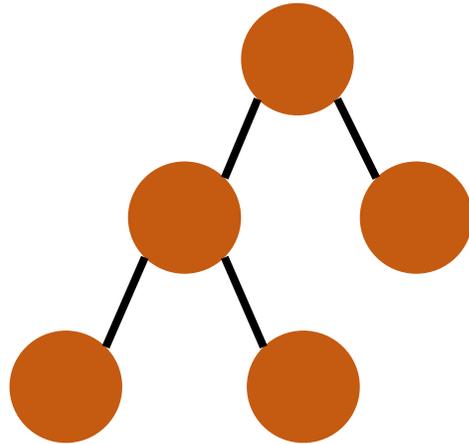
1, 6	5, 2	4	9, 7, 8	3
------	------	---	---------	---
- Y codificaciones especiales propias de los problemas



# Técnicas de Cómputo Evolutivo

Neuroevolución

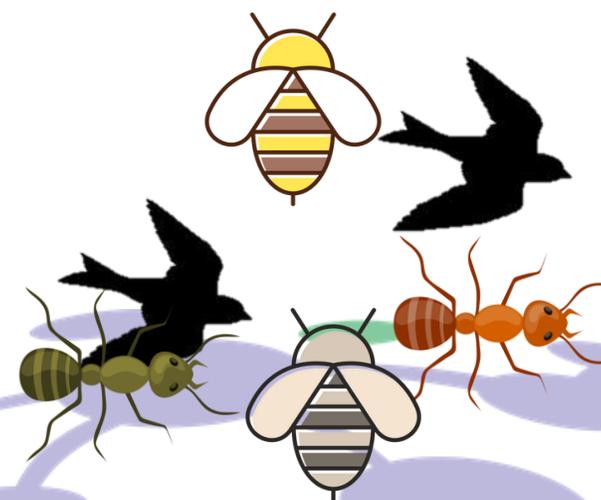
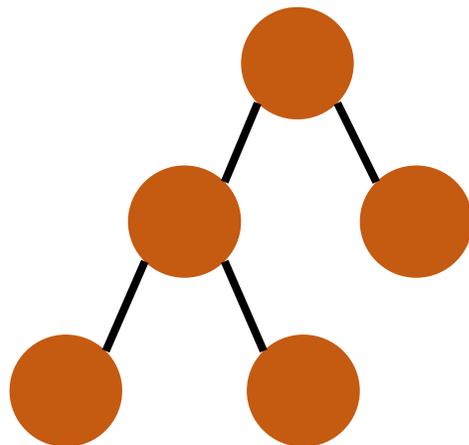
- Algoritmos Genéticos
- Estrategias Evolutivas
- Programación Genética
- Programación Evolutiva
- Inteligencia Colectiva



# Técnicas de Cómputo Evolutivo

Neuroevolución

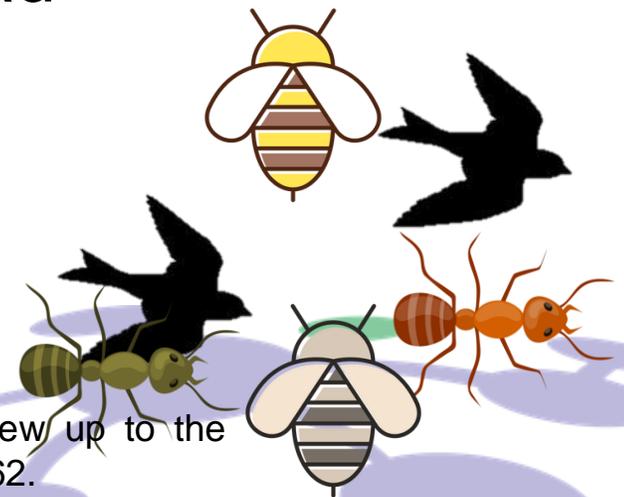
- **Algoritmos Genéticos**
- Estrategias Evolutivas
- **Programación Genética**
- Programación Evolutiva
- **Inteligencia Colectiva**



# Particle Swarm Optimization (PSO)

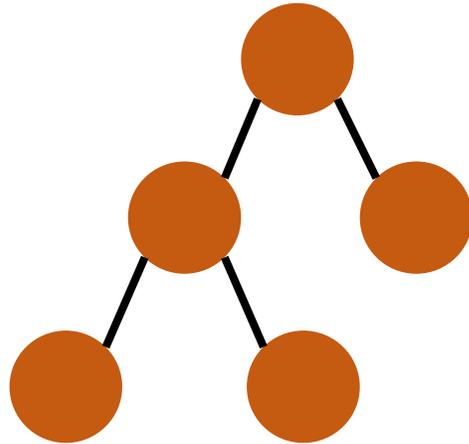
El algoritmo de Optimización por Enjambre de Partículas, tiene las siguientes características:

- Su población son partículas
- Cada una de éstas almacena su mejor posición y la mejor del enjambre en un espacio continuo
- Cada partícula, además, tiene una velocidad
- Generacional



La Programación Genética, se caracteriza por lo siguiente:

- Su codificación es un árbol
- Dos elementos son fundamentales en este tipo de algoritmos: elementos terminales y funciones



# Algoritmos Genéticos

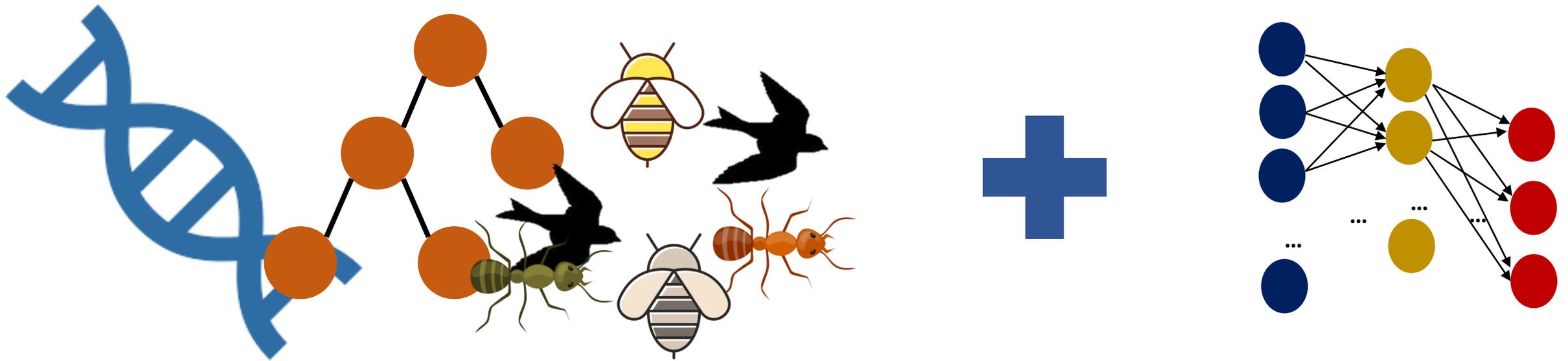
Inspirados en el Principio de la Evolución Natural:

- Consiguen realizar un proceso de búsqueda con distintas codificaciones
- Existe la selección y reproducción de individuos



# Neuroevolución

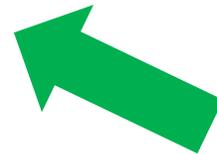
La Neuroevolución (NE) es el uso de algoritmos de Computación Evolutiva para la búsqueda de las mejores arquitecturas de redes neuronales y, algunas veces, de sus pesos



Stanley, K. O., Clune, J., Lehman, J. & Miikkulainen, R. (2019). Designing neural networks through neuroevolution. *Nature Machine Intelligence*, 1(1), 24-35.  
<https://doi.org/10.1038/s42256-018-0006-z>

Un algoritmo PSO (DCGAN-PSO) utilizado para la evolución de GANs para la generación de imágenes biomédicas:

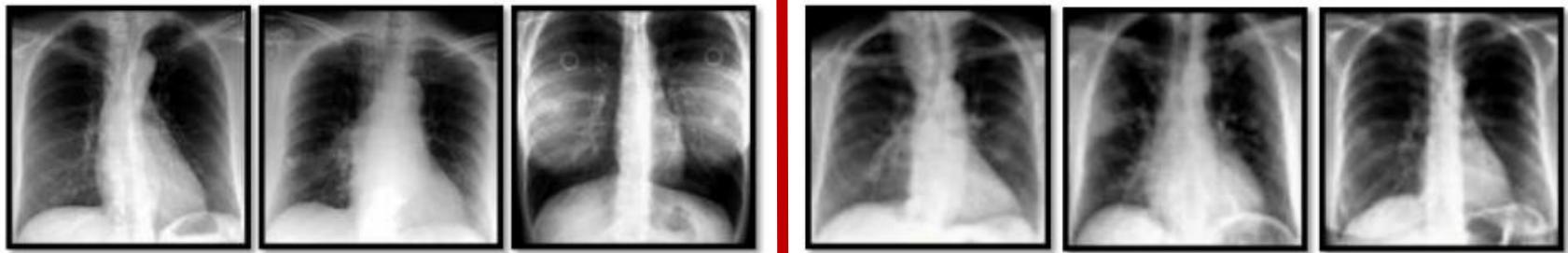
- Es la primera vez que se utiliza un PSO para GANs
- Las imágenes generadas mediante las GANs usando PSO son mejores en cuanto a calidad que las generadas por una GAN hecha a mano
- **La prematura convergencia del PSO reduce el tiempo de búsqueda de la arquitectura**



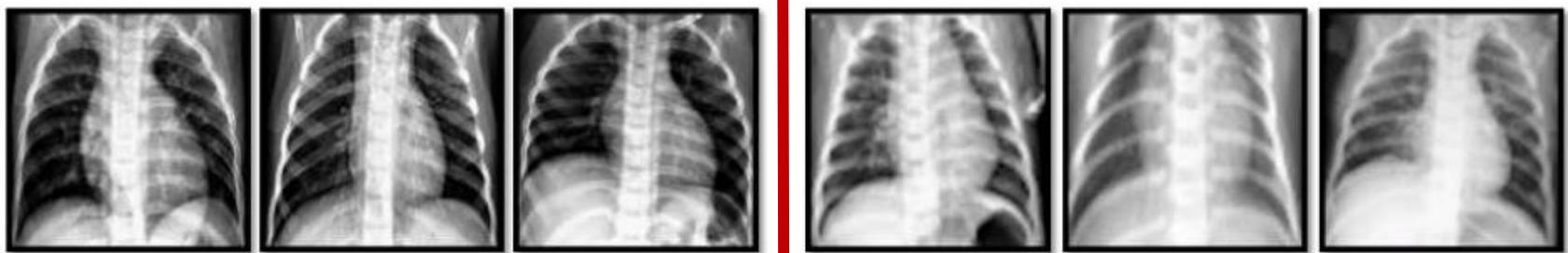
# Algunas imágenes resultantes

Aplicaciones

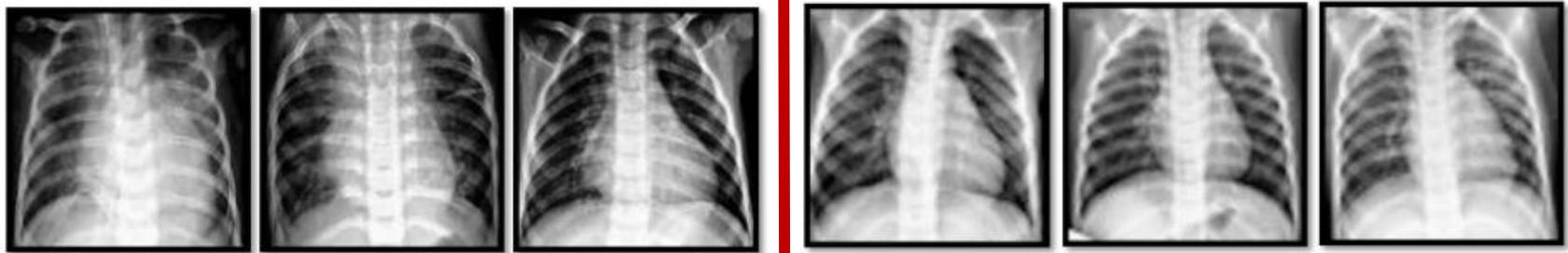
COVID-19



Neumonía



Sanos



Ejemplo de imágenes sintetizadas  
Imagen tomada de: <https://bit.ly/3oQPn4l>

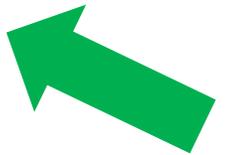
Programación Genética (NASGP-Net) para la segmentación de imágenes de placentas y retinas:

- Se usó para evolucionar arquitecturas U-Net
- Se encontraron arquitecturas con hasta un 98% menos complejas que la original
- Su función de evaluación incluye el número de parámetros
- Las redes se entrenan con retropropagación
- **Reducción de tiempo de entranamiento de 96 a 24 horas**



Clasificación de imágenes con CNNs evolucionadas con un Algoritmo Genético (DeepGA):

- Se realizó un estudio para probar diferentes codificaciones [9]
- **Se encontraron arquitecturas más pequeñas que las iniciales, que pueden ser usadas en hospitales donde no cuentan con GPU**
- Su función de evaluación incluye el número de parámetros
- Las redes se entrenan con retropropagación

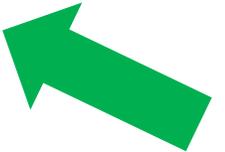


Clasificación de texto en diferentes polaridades de sentimientos con NEAT:

- NEAT [10] es un algoritmo genético con el cual inició la historia de la Neuroevolución, a principios del milenio
- **Se encontraron arquitecturas más pequeñas que las creadas a mano**
- Su función de evaluación **NO** incluye el número de parámetros
- **Y la actualización de pesos se realiza mediante variaciones genéticas**

TextCNN evolucionada con un Algoritmo Genético (DeepNEWT):

- Es la primera vez que se evolucionan pesos y arquitecturas de una CNN con variaciones genéticas al mismo tiempo
- **Se encontraron arquitecturas hasta en un 96% más pequeñas que la creada a mano**
- Su función de evaluación **NO** incluye el número de parámetros



# Conclusiones y trabajo futuro

La Neuroevolución ha demostrado que puede apoyar en la reducción de la complejidad de las redes neuronales

Por otro lado, es necesario ejecutar más experimentos, lo que conlleva manejar cómputo de alto desempeño...



# Referencias

1. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C. and Chen, M. (2022). Hierarchical Text-Conditional Image Generation with CLIP Latents. arXiv. <https://arxiv.org/abs/2204.06125>
2. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. Advances in Neural Information Processing Systems 2020-December (2020)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference 1(Mlm), 4171–4186 (2019), <https://arxiv.org/abs/1810.04805>
4. Oh, Y., S. Park, and J. C. Ye (2020). “Deep Learning COVID-19 Features on CXR using Limited Training Data Sets”. In: IEEE Transactions on Medical Imaging 39.8, pp. 2688– 2700. DOI: <https://.www.doi.org/10.1109/TMI.2020.2993291>
5. He, K. et al. (2016). “Deep Residual Learning for Image Recognition”. In: 2016 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 770–778. <https://www.doi.org/10.1109/CVPR.2016.90>

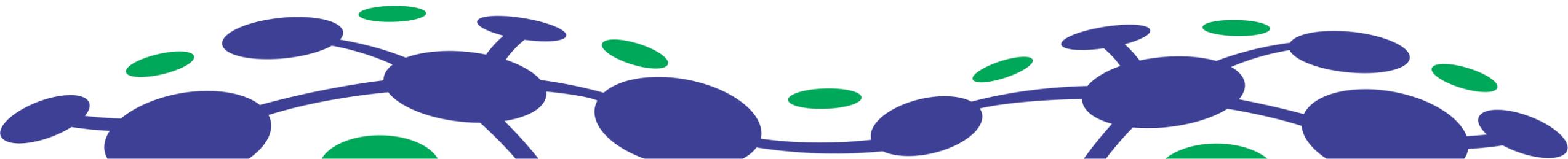
# Referencias

6. Civit-Masot, J. et al. (2020). "Deep Learning System for COVID-19 Diagnosis Aid Using X-ray Pulmonary Images". In: Applied Sciences 10.13. <https://doi.org/10.3390/app10134640>
7. Kim, Y.: Convolutional neural networks for sentence classification. EMNLP 2014 – 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference pp. 1746–1751 (2014). <https://doi.org/10.3115/v1/d14-1181>
8. Ronneberger, O., Fischer, P. y Brox, T. U-net: Convolutional networks for biomedical image segmentation en International Conference on Medical image computing and computer-assisted intervention (2015), 234-241. [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
9. G. -A. Vargas-Hákim, E. Mezura-Montes and H. -G. Acosta-Mesa, "A Review on Convolutional Neural Network Encodings for Neuroevolution," in *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 1, pp. 12-27, Feb. 2022, <https://www.doi.org/10.1109/TEVC.2021.3088631>
10. Stanley, K. O. & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2), 99-127. <https://doi.org/10.1162/106365602320169811>

**GRACIAS POR SU  
ATENCIÓN**



**¿PREGUNTAS?**





Comunidad de  
Supercómputo



Σύνταγμα  
Syntagma

## Neuroevolución

# Una alternativa para simplificar los modelos asociados al Aprendizaje Profundo

Presenta: Mtro. José-Clemente Hernández-Hernández  
jclementehdzhdz@gmail.com

Instituto de Investigaciones en Inteligencia Artificial  
Universidad Veracruzana

03 de agosto del 2022