

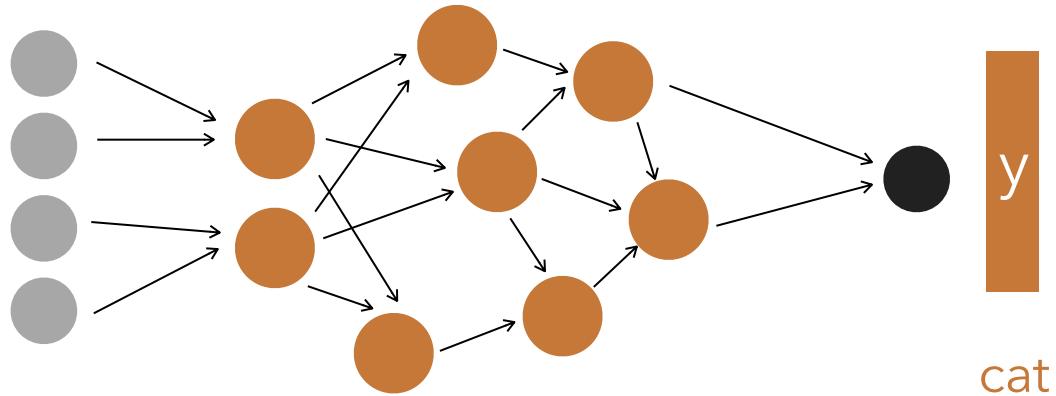
Arquitectura Básica de Redes Neuronales

Expresividad y Profundidad de la Red



Expresividad de la Red

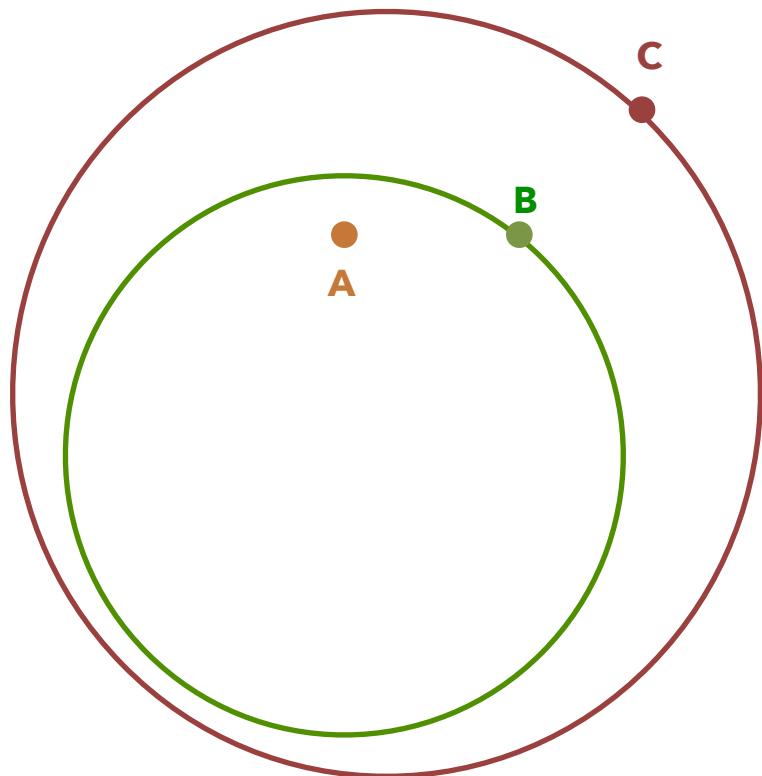
- Hemos dicho que una red neuronal permite implementar **una función entre el espacio de entrada X y el espacio de salida Y**. Queremos usar esa función para resolver problemas de aprendizaje, es decir para reproducir las respuestas que vemos en un conjunto de ejemplos.



- Es natural entonces preguntarse: **¿Qué funciones podemos implementar con el modelo? ¿Cualquiera? ¿O hay un límite?**

Error de Aproximación

- Esto se relaciona íntimamente con lo que aprendizaje automático se denomina **error de aproximación del modelo**.



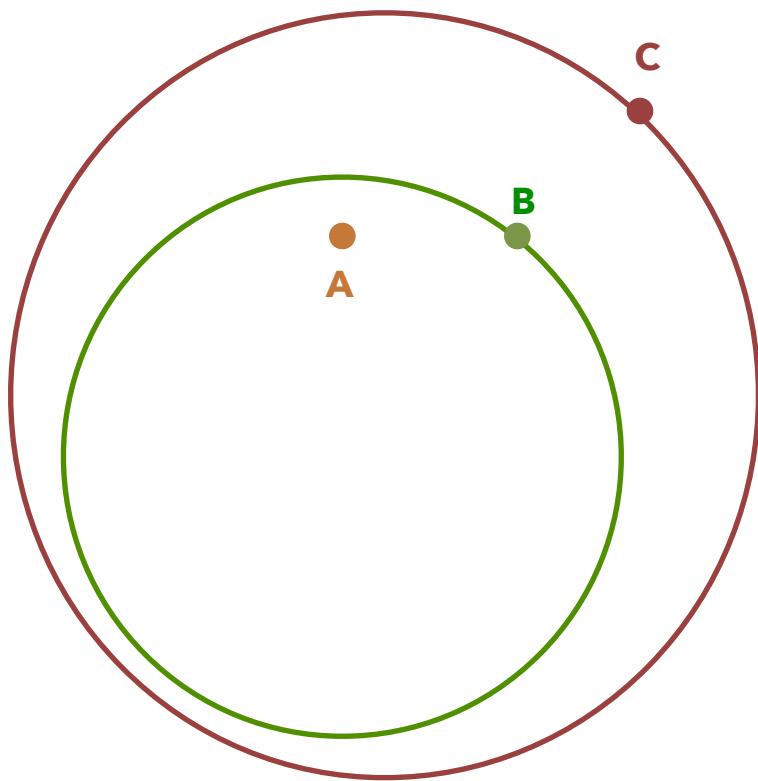
C: Función deseada

B: Mejor función que la red puede implementar

A: Función que la red aprende desde el conjunto (finito) de ejemplos

Error de Estimación

- Hay que tener cuidado con espacios de funciones demasiado expresivos cuando se tiene un número pequeño de ejemplos. Con frecuencia ocurre que estos modelos tienen un mayor **error de estimación**.



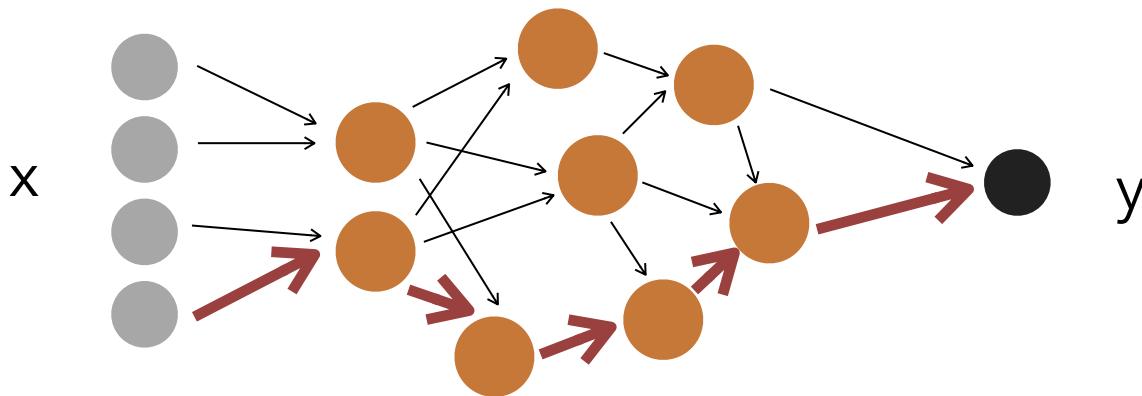
C: Función deseada

B: Mejor función que la red puede implementar

A: Función que la red aprende desde el conjunto (finito) de ejemplos

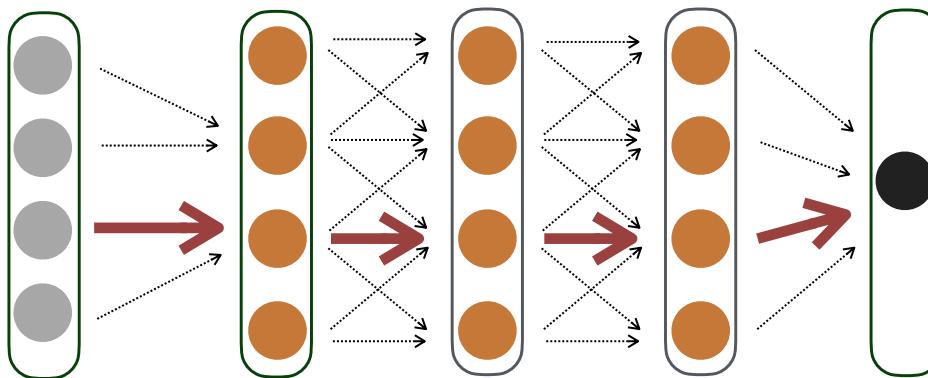
Profundidad y Expresividad de la Red

- En este capítulo queremos resumir algunos resultados clásicos sobre la expresividad de una red neuronal relacionándolos con la idea de profundidad de la red, definida como el camino más largo en el grafo de computación.



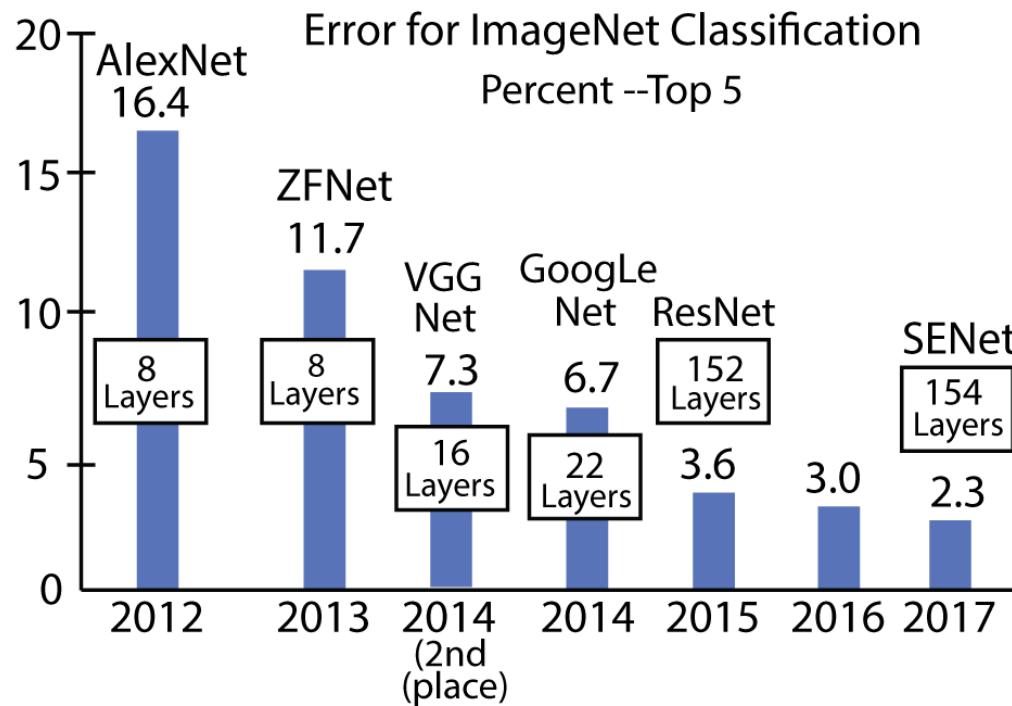
Profundidad y Expresividad de la Red

- Claramente, para redes FF, la profundidad es simplemente el número de capas ocultas (+1).



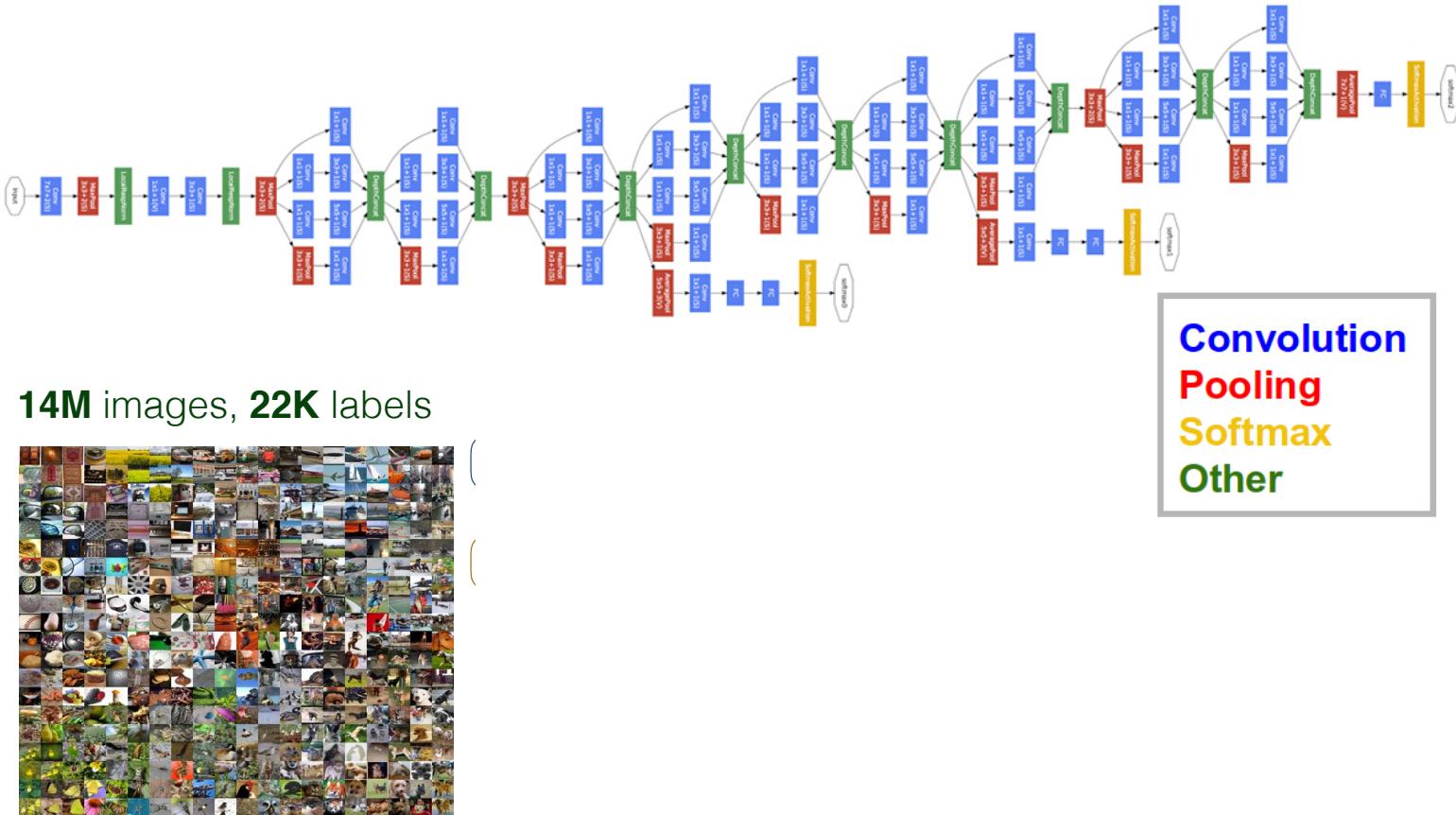
Profundidad y Expresividad de la Red

- Este link es interesante porque la eficacia que se ha visto en los últimos años de las redes neuronales en muchos problemas de inteligencia artificial se ha logrado usando redes muy muy profundas (en comparación con las redes que se usaban antes de la era deep learning).



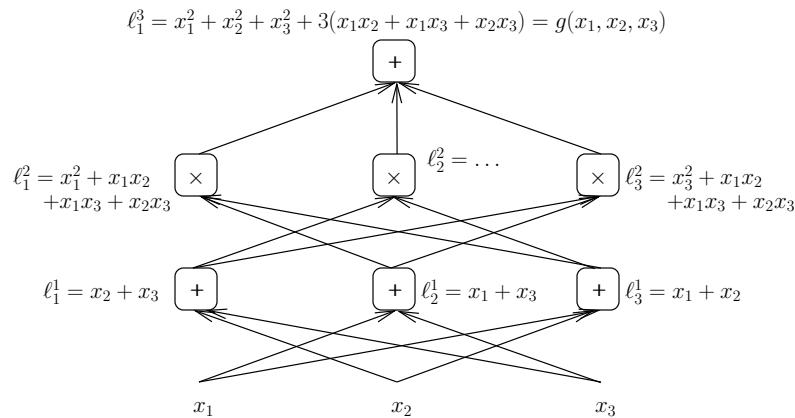
Profundidad y Expresividad de la Red

- GoogleNet, el primer modelo en obtener accuracy super-humana en un desafío de reconocimiento de objetos, tiene 22 capas.



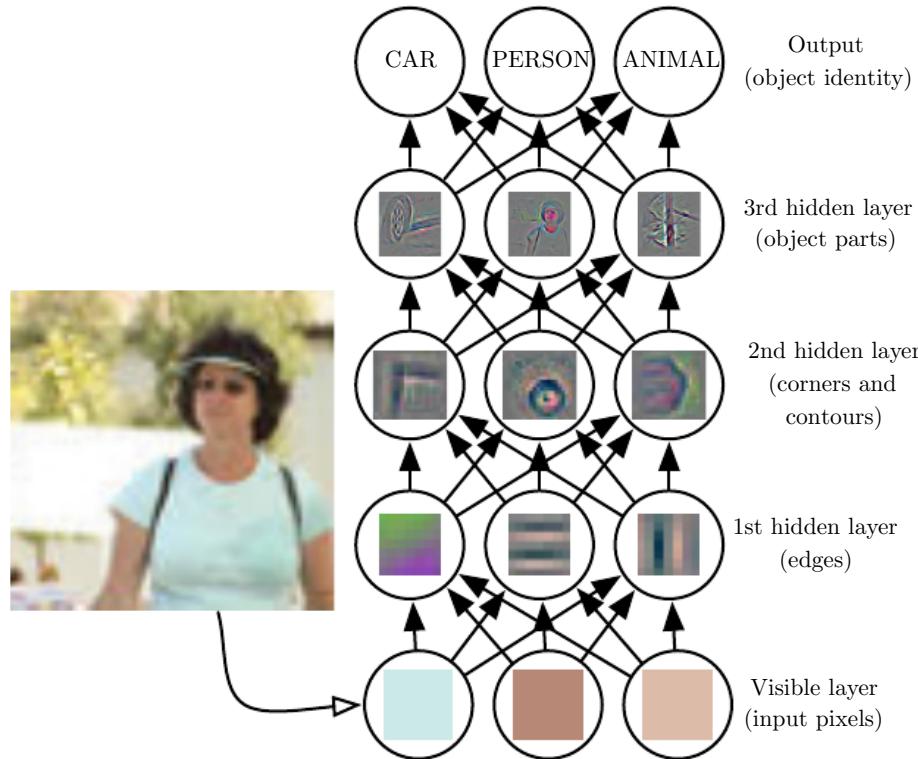
Beneficios Intuitivos de la Profundidad

- Una forma intuitiva de entender las ventajas de la profundidad es recordar uno de los principios básicos en diseño de algoritmos: estructurar la computación en varios niveles permite reutilizar cálculos.



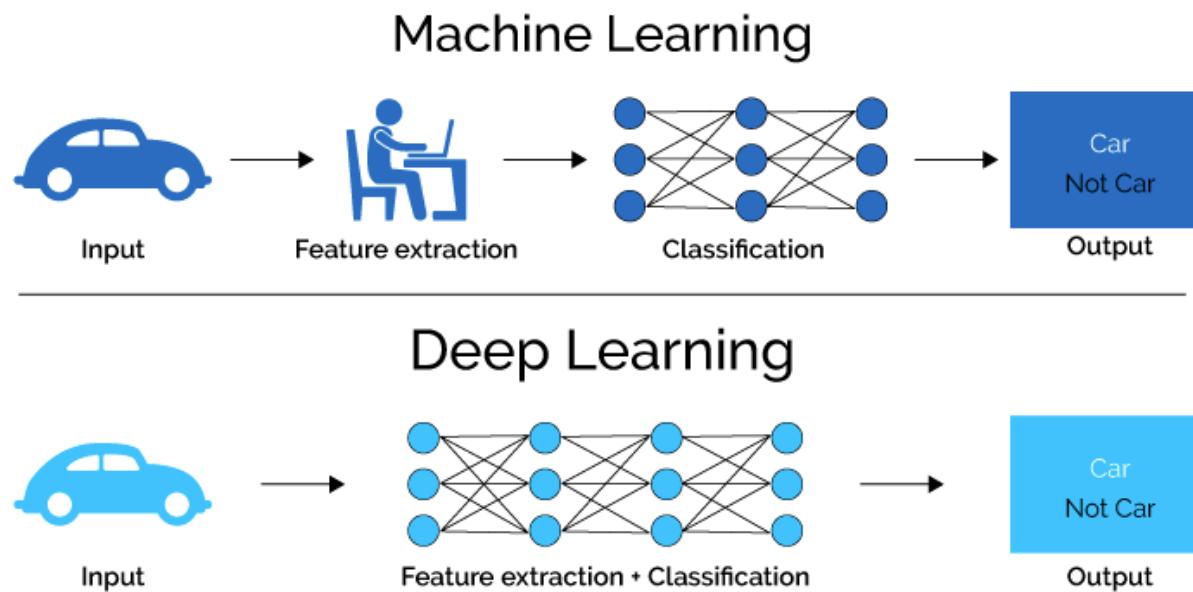
Beneficios Intuitivos de la Profundidad

- Estructurar la computación en varios niveles permite a la red componer un concepto de mayor complejidad a partir de conceptos más primitivos.



Beneficios Intuitivos de la Profundidad

- Otra forma de entender las ventajas de la profundidad, es recordando una de las principales limitaciones de los métodos clásicos de aprendizaje automático: dependen fuertemente de la calidad de la ingeniería de atributos que pueda hacer un humano.



Beneficios Intuitivos de la Profundidad



A Few Useful Things to Know about Machine Learning

Pedro Domingos

Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350, U.S.A.
pedrod@cs.washington.edu

8. FEATURE ENGINEERING IS THE KEY

At the end of the day, some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used. If you have many independent features that each correlate well with the class, learning is easy. On the other hand, if the class is a very complex function of the features, you may not be able to learn it. Often, the raw data is not in a form that is amenable to learning, but you can construct features from it that are. This is typically where most of the effort in a machine learning project goes. It is often also one of the most interesting parts, where intuition, creativity and “black art” are as important as the technical stuff.

Beneficios Intuitivos de la Profundidad

- El patrón de activación de las capas ocultas de una red puede verse como una **representación** alternativa del dato de entrada, que la red optimiza para que la última capa logre resolver las tareas de aprendizaje.

Scaling Learning Algorithms towards AI

Yoshua Bengio (1) and Yann LeCun (2)

(1) Yoshua.Bengio@umontreal.ca

Département d'Informatique et Recherche Opérationnelle
Université de Montréal,

(2) yann@cs.nyu.edu

The Courant Institute of Mathematical Sciences,
New York University, New York, NY

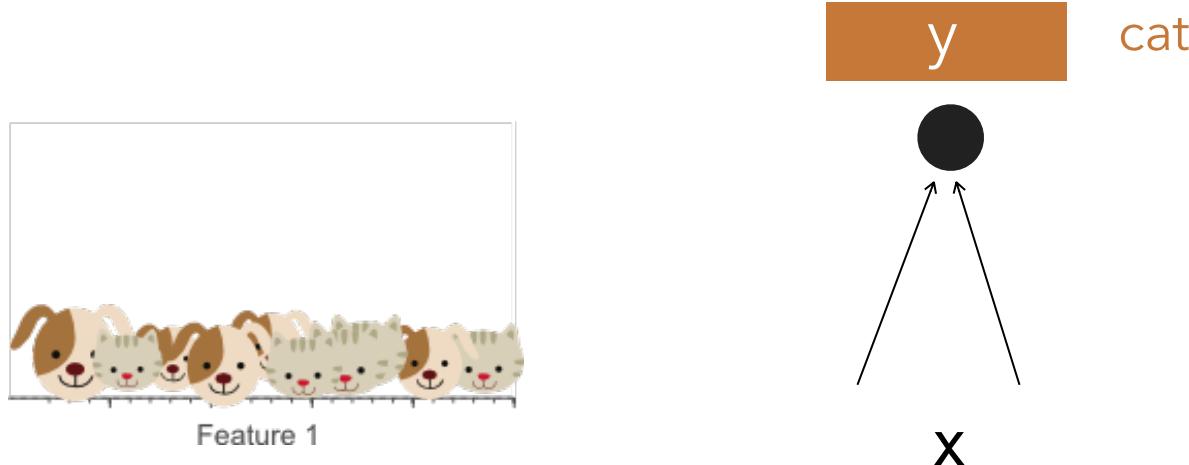


The International Conference on Learning Representations (ICLR) is the premier gathering of professionals dedicated to the advancement of the branch of artificial intelligence called representation learning, but generally referred to as deep learning.

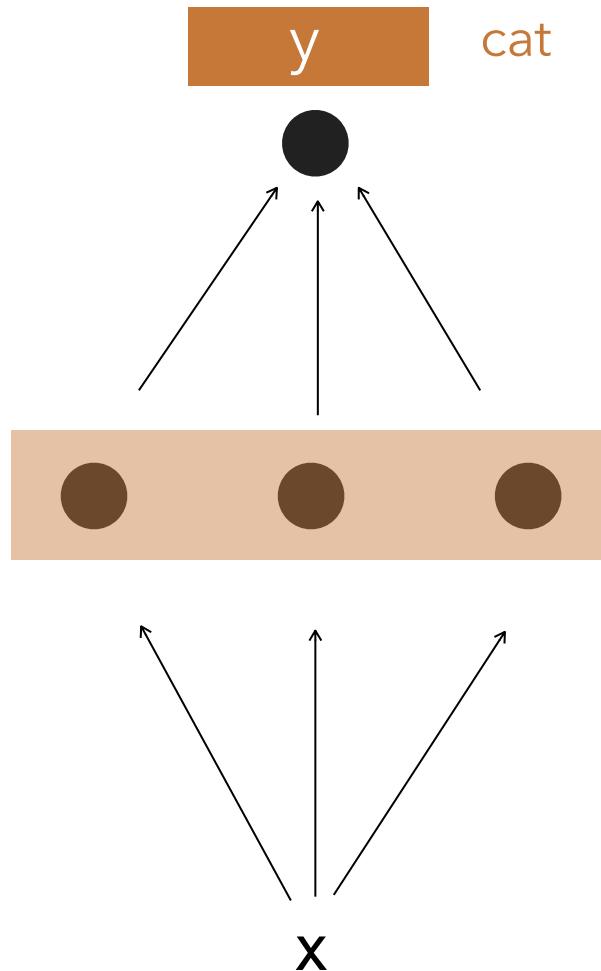
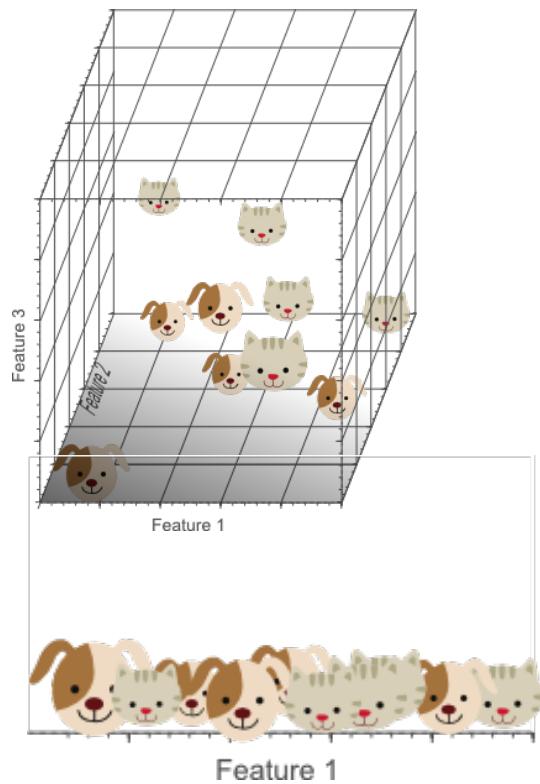
ICLR is globally renowned for presenting and publishing **cutting-edge research on all aspects of deep learning used in the fields of artificial intelligence, statistics and data science, as well as important application areas such as machine vision, computational biology, speech recognition, text understanding, gaming, and robotics.**

Beneficios Intuitivos de la Profundidad

- El patrón de activación de las capas ocultas de una red puede verse como una representación alternativa del dato de entrada, que la red optimiza para que la última capa logre resolver las tareas de aprendizaje.

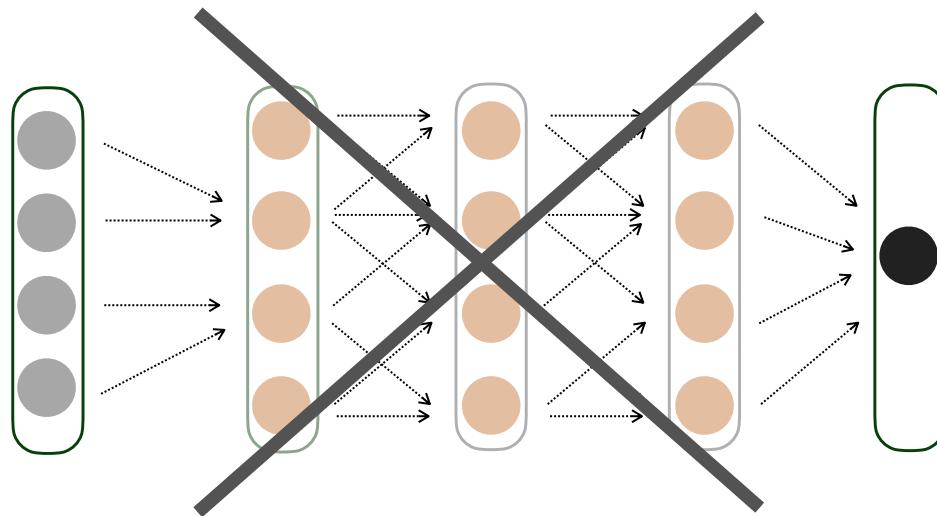


Beneficios Intuitivos de la Profundidad



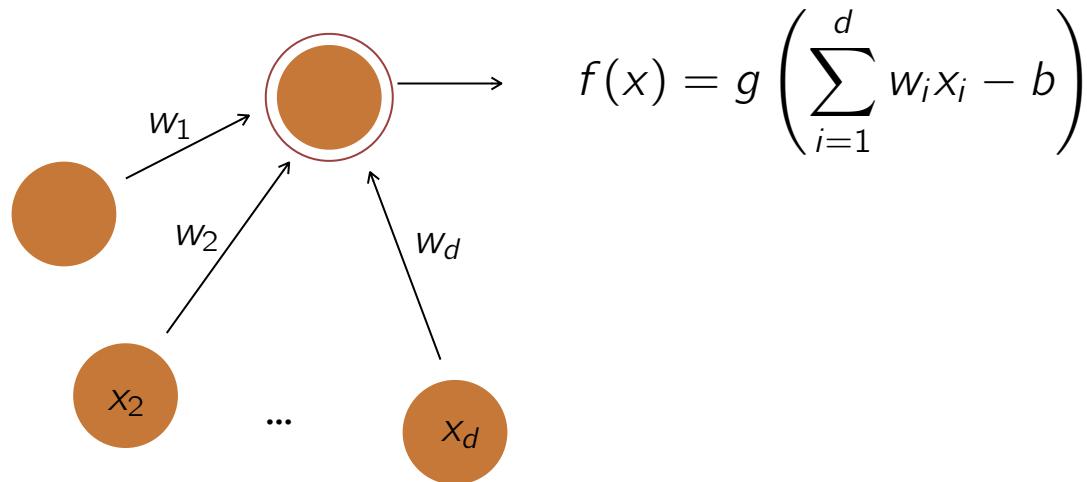
Beneficios de la Profundidad

- Una forma de determinar si la profundidad tiene un efecto que podamos medir matemáticamente consiste en remover las capas ocultas de una red y estudiar cuántos perdemos en términos de capacidad de aproximación.



Expresividad de una Neurona (MP)

- Si partimos analizando las capacidades de una neurona MP, podemos descubrir rápido que ésta puede implementar **un número muy limitado de funciones.**



El Problema de la Paridad

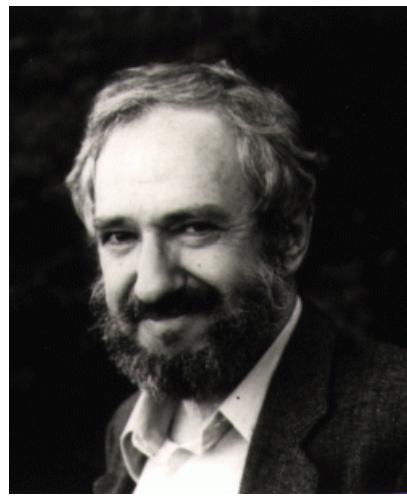
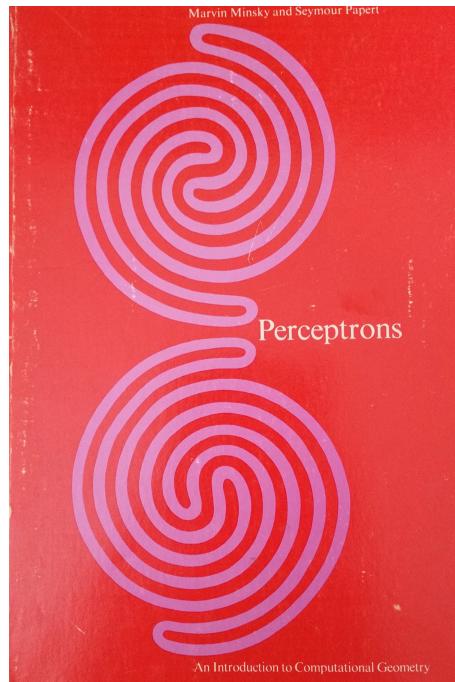
- La situación partió observándose en el caso de entradas y salidas binarias, con el célebre ejemplo de la **función de paridad (XOR en varias variables)**. En este caso, la neurona debiese retornar 0 si el número de 1's en el input es par y 1 en otro caso.

$$f(x) = \bigoplus_{i=1}^d x_i = \left(x_1 \bigoplus \left(\left(\left(x_2 \bigoplus \dots \bigoplus x_{d-1} \right) \bigoplus x_d \right) \right) \right)$$

- **Es posible demostrar que no existe una neurona MP capaz de calcular esta función.**

El Problema de la Paridad

- La primera demostración formal de este resultado las dieron Minsky & Pappert en el famoso libro de 1959 (*Perceptrons*), que apagó fuertemente el interés por las redes neuronales en su tiempo.



Límites de una Neurona (MP)

- El resultado es hoy bastante evidente si consideramos la situación en \mathbb{R}^d
- Notemos que la activación de la neurona MP se puede escribir como

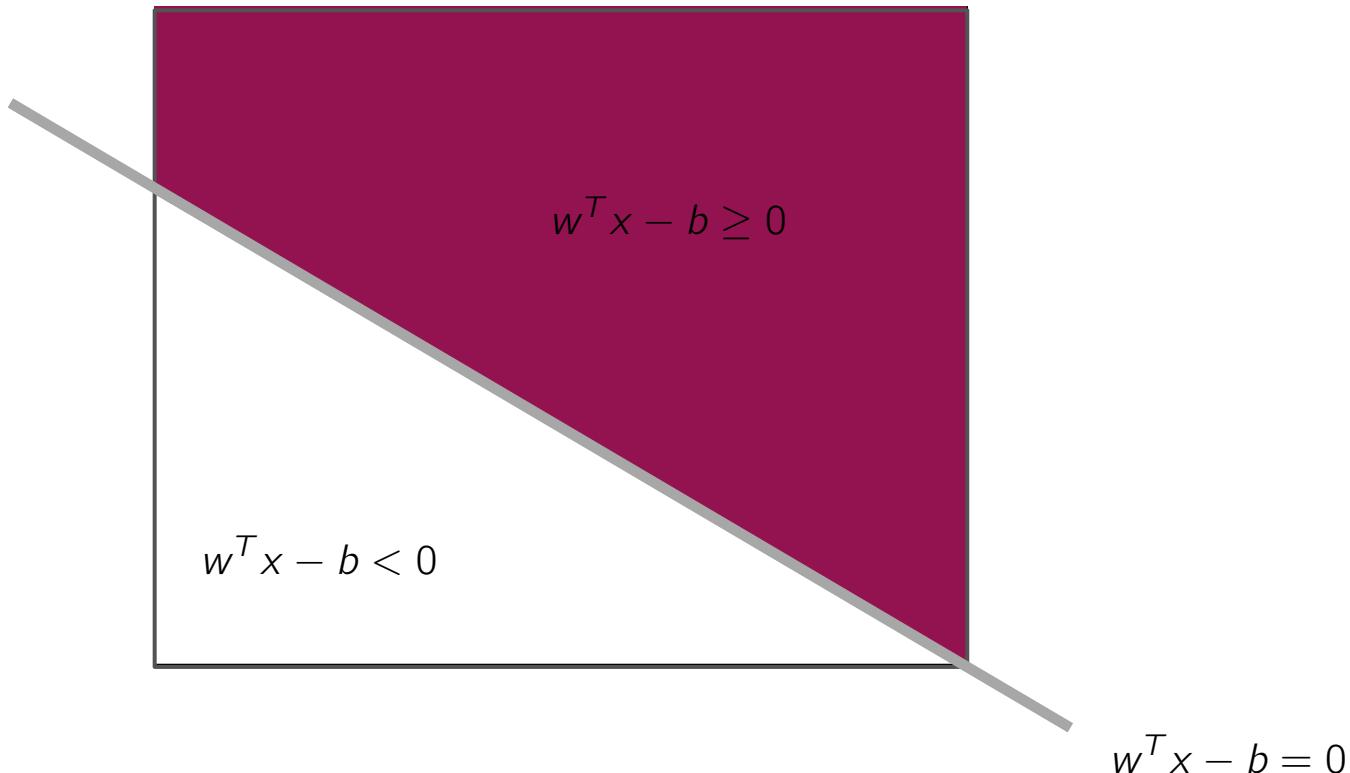
$$f(x) = \begin{cases} 1 & \sum_i w_i x_i - b \geq 0 \\ 0 & \sum_i w_i x_i - b < 0 \end{cases}$$

o equivalentemente ...

$$f(x) = \begin{cases} 1 & w^T x - b \geq 0 \\ 0 & w^T x - b < 0 \end{cases}$$

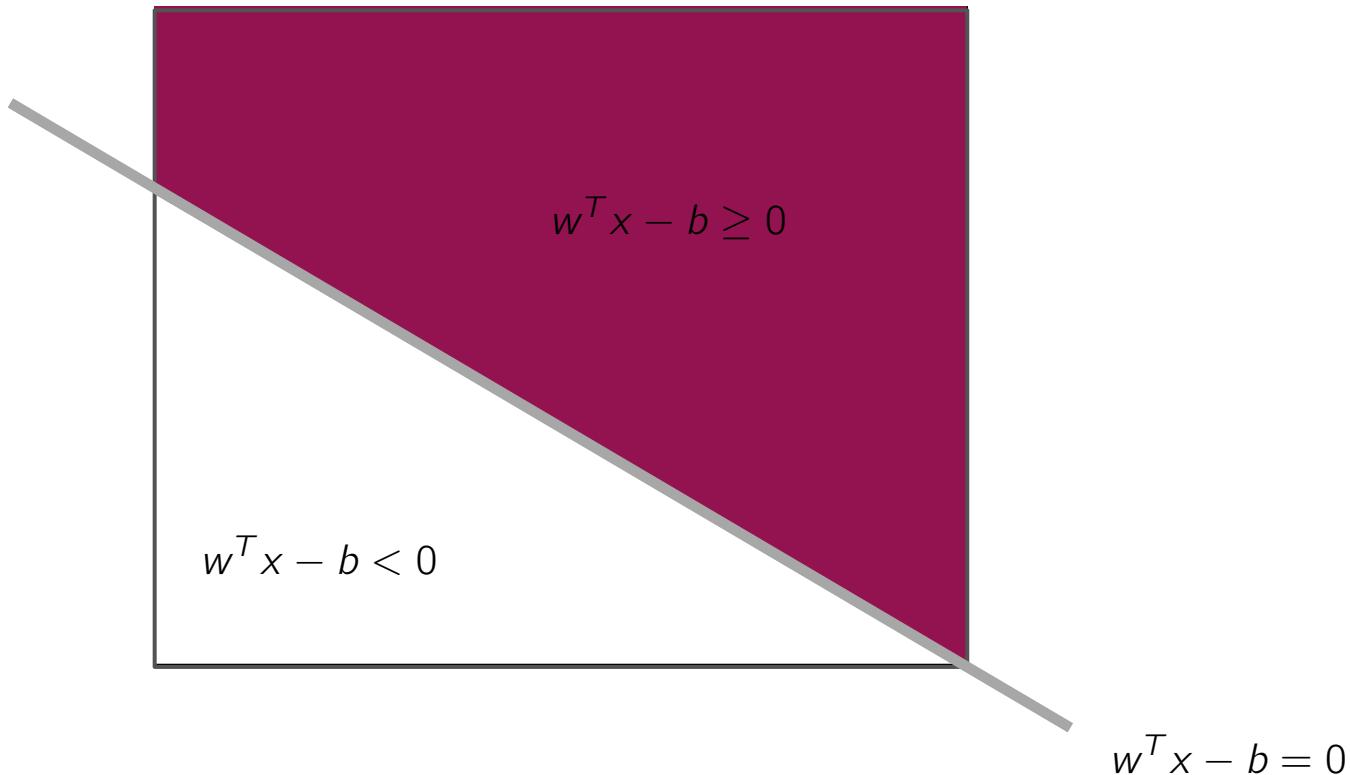
sin embargo, la ecuación $w^T x - b = 0$ no es otra cosa que la ecuación de un hiperplano! (una "línea" en \mathbb{R}^d).

Límites de una Neurona (MP)



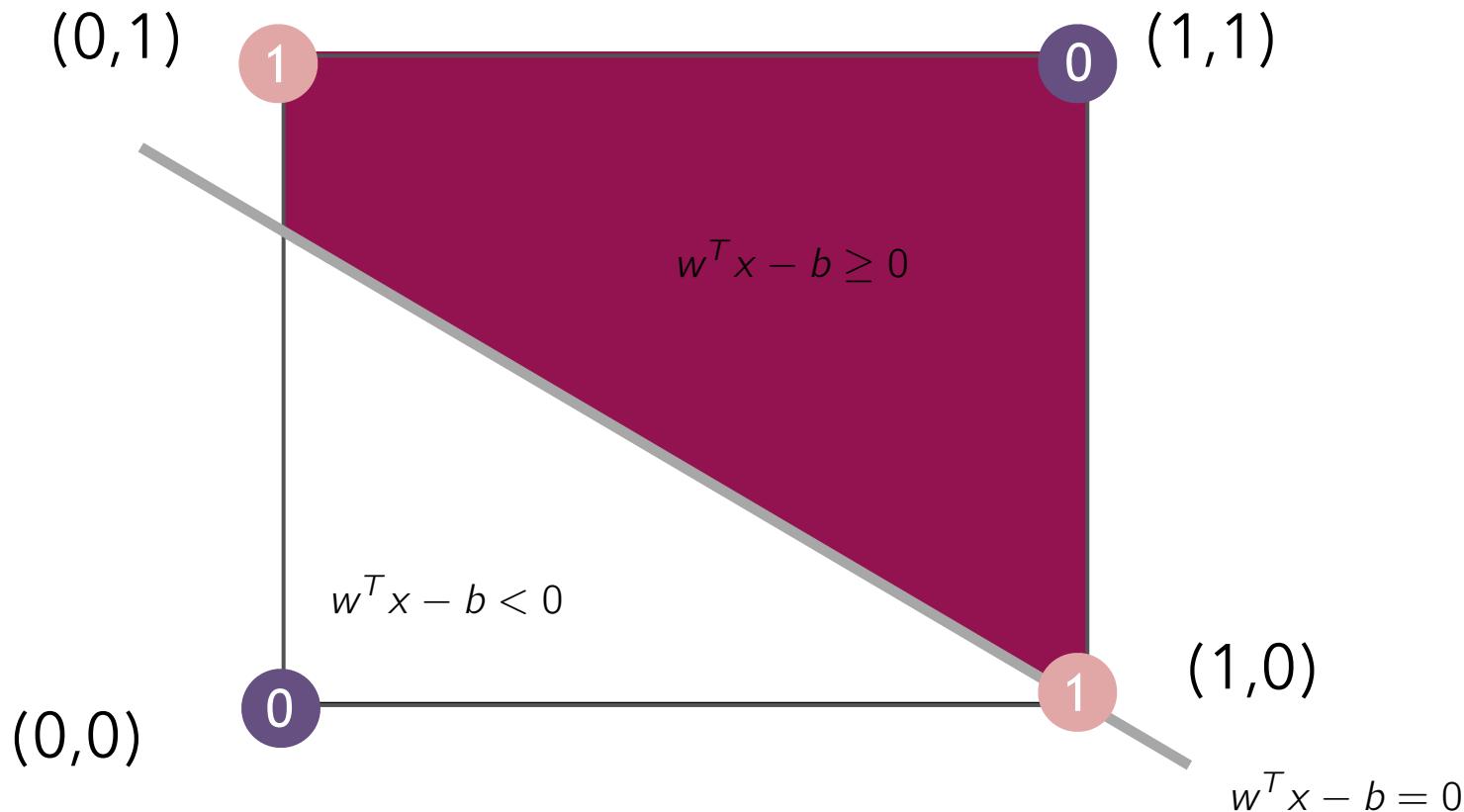
La neurona se activa en un “lado” de hiperplano y en el otro está inactiva.

Límites de una Neurona (MP)



En otras palabras, la neurona MP implementa un **discriminador lineal** en \mathbb{R}^d

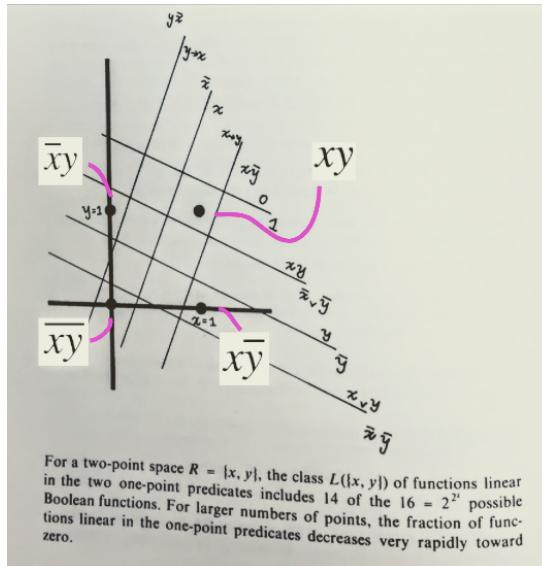
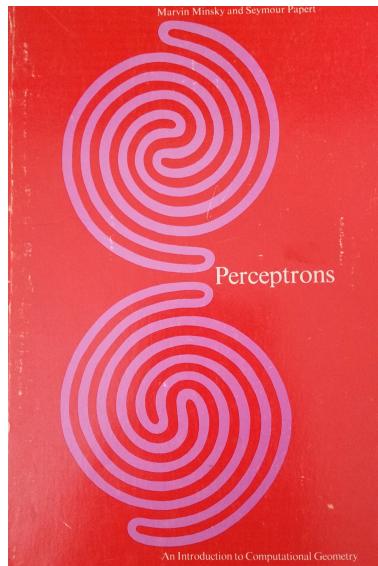
Límites de una Neurona (MP)



Como los conjuntos $\{(0,0); (1,1)\}$ y $\{(1,0),(0,1)\}$ no son linealmente separables,
no es posible encontrar una neurona MP que implemente la función XOR.

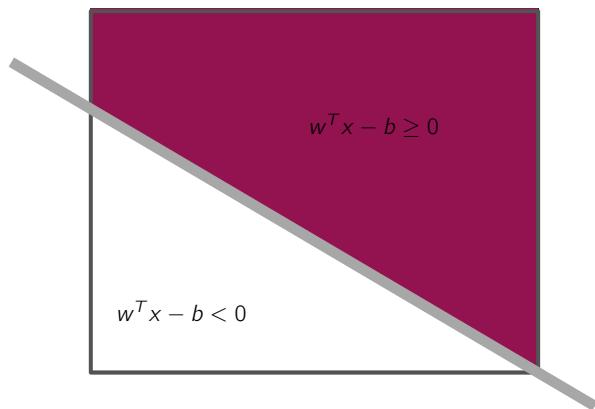
Límites de una Neurona (MP)

- La función de paridad y su negación son las únicas 2 funciones booleanas de 2 variables que una neurona MP no puede implementar.
- Sin embargo, como demuestran Minsky y Papert, la fracción de funciones booleanas lineales en sus argumentos **decrece rápidamente a 0** en mayores dimensiones.



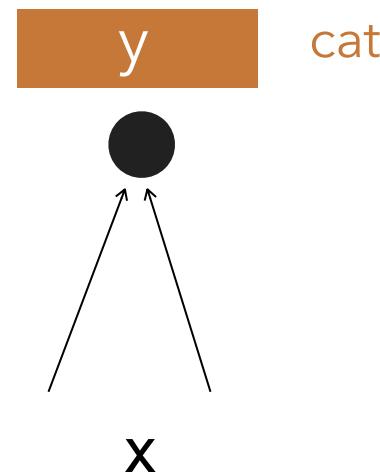
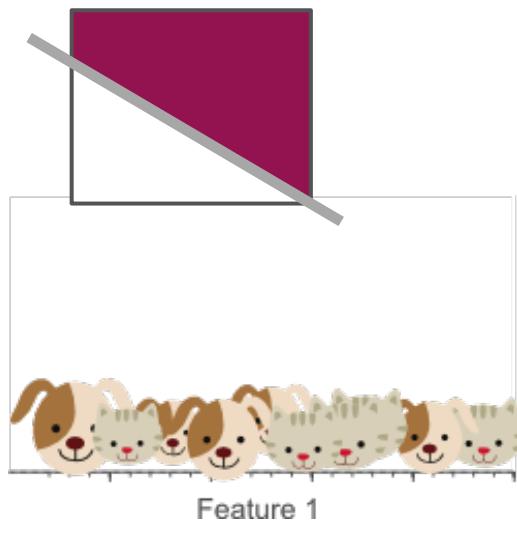
Límites de una Neurona (MP)

- Una situación similar se presenta en \mathbb{R}^d .
- Como la neurona MP implementa un discriminante lineal, su “capacidad” como clasificador es idéntica a la de un regresor logístico o la de una SVM lineal, dos clasificadores lineales clásicos.



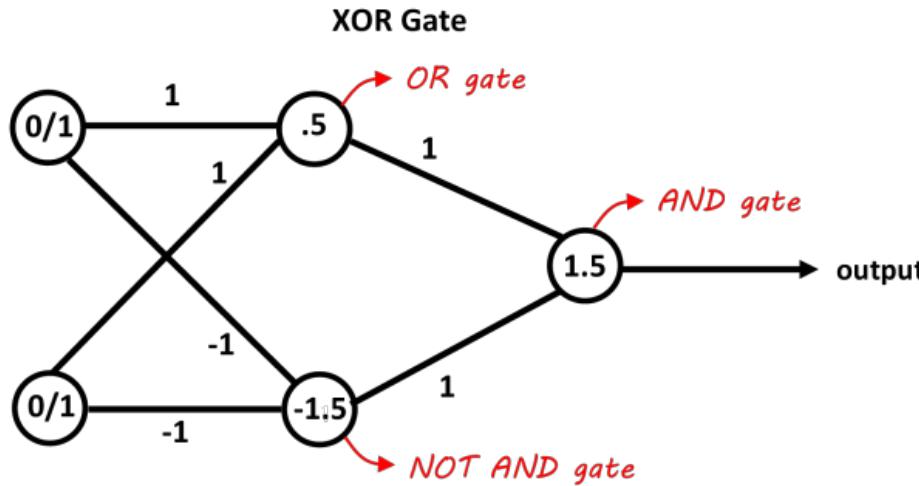
Límites de una Neurona (MP)

- Si entrenamos una neurona con ejemplos representados de manera que no son linealmente separables, la unidad no podrá distinguir adecuadamente entre ambas clases.



Límites de una Red de Neuronas

- **¿Puede mejorarse esta capacidad expresiva usando redes de neuronas en vez de 1 neurona individual?**
- De hecho, ya hemos visto que para $d=2$, es posible construir una red de 3 neuronas MP que implementa e XOR.



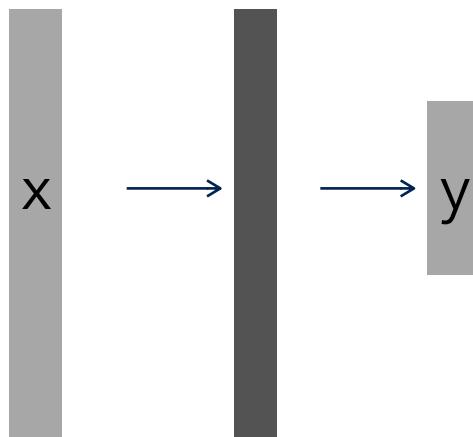
(Es posible demostrar de hecho que 3 es el número mínimo de neuronas necesarias)

Universalidad de una Red Neuronal

- Es sorprendente verificar que, en efecto, si consideramos redes de neuronas en vez de neuronas individuales, **basta sólo 1 capa oculta para obtener un aproximador universal.**

Teorema

Una red feed-forward binaria con 1 capa oculta puede implementar cualquier función booleana.



Universalidad de una Red Neuronal

- La demostración es relativamente sencilla y se basa en la siguiente observación (o resultado): **toda función booleana de d-variables se puede reducir a lo que se denomina forma normal disyuntiva (FND)**, es decir, se puede escribir como disyunción de un número finito de cláusulas que contienen sólo conjunciones de variables posiblemente negadas.

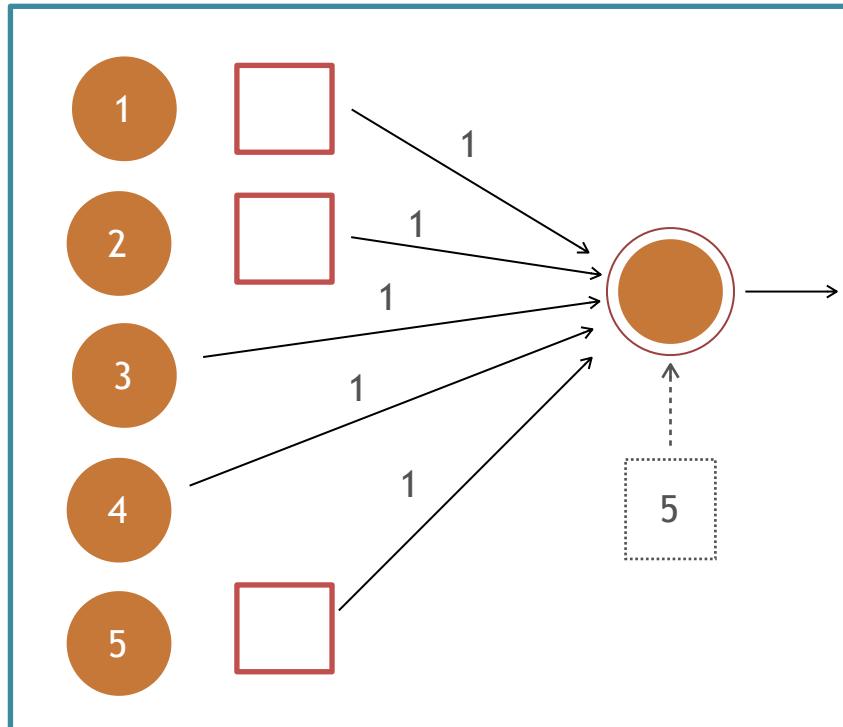
X ₁	X ₂	X ₃	X ₄	X ₅	Y
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1

$$Y = \bar{X}_1 \bar{X}_2 X_3 X_4 \bar{X}_5 + \bar{X}_1 X_2 \bar{X}_3 X_4 X_5 + \bar{X}_1 X_2 X_3 \bar{X}_4 \bar{X}_5 + \\ X_1 X_2 X_3 \bar{X}_4 X_5 + X_1 \bar{X}_2 X_3 X_4 X_5 + X_1 X_2 \bar{X}_3 \bar{X}_4 X_5$$

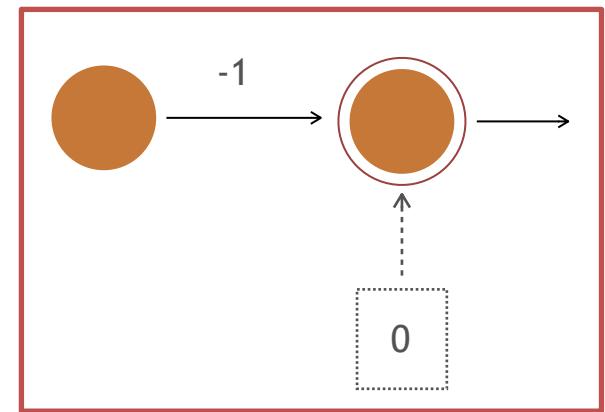
Universalidad de una Red de Neuronas

- Como podemos hacer conjunciones y negar, es fácil construir una mini-red que compute las cláusulas requeridas por la FND

$$Y = \overline{\bar{X}_1 \bar{X}_2 X_3 X_4 \bar{X}_5} + \bar{X}_1 X_2 \bar{X}_3 X_4 X_5 + \bar{X}_1 X_2 X_3 \bar{X}_4 \bar{X}_5 + \\ X_1 X_2 X_3 \bar{X}_4 X_5 + X_1 \bar{X}_2 X_3 X_4 X_5 + X_1 X_2 \bar{X}_3 \bar{X}_4 X_5$$



NOT

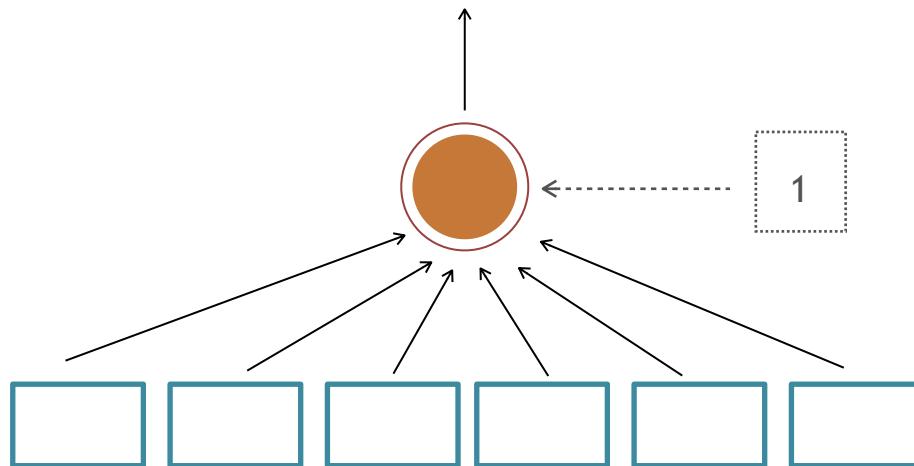


4 neuronas (+5)

Universalidad de una Red de Neuronas

- Como podemos hacer disyunciones, es fácil construir una neurona que ensamble las sub-redes que corresponden a las cláusulas FND

$$Y = \bar{X}_1 \bar{X}_2 X_3 X_4 \bar{X}_5 + \bar{X}_1 X_2 \bar{X}_3 X_4 X_5 + \bar{X}_1 X_2 X_3 \bar{X}_4 \bar{X}_5 + \\ X_1 X_2 X_3 \bar{X}_4 X_5 + X_1 \bar{X}_2 X_3 X_4 X_5 + X_1 X_2 \bar{X}_3 \bar{X}_4 X_5$$



Universalidad de una Red Neuronal

- Obtenemos así el resultado:

Teorema

Una red feed-forward binaria con 1 capa oculta puede implementar cualquier función booleana.

- El detalle del teorema es que no especifica cuántas neuronas podrían requerirse. Es posible mostrar que en el peor caso este número es exponencial en d !
- Por ejemplo, la función de paridad tiene una FND de 2^{d-1} términos.
- Es posible mostrar que ese es el número máximo de términos para una función booleana cualquiera (después de aplicar un proceso de simplificación denominado reducción de la FND).



Universalidad de una Red Neuronal

- Un resultado similar se encuentra cuando consideramos inputs continuos y redes con funciones de activación continuas.

Teorema (Cybenko, 1989)

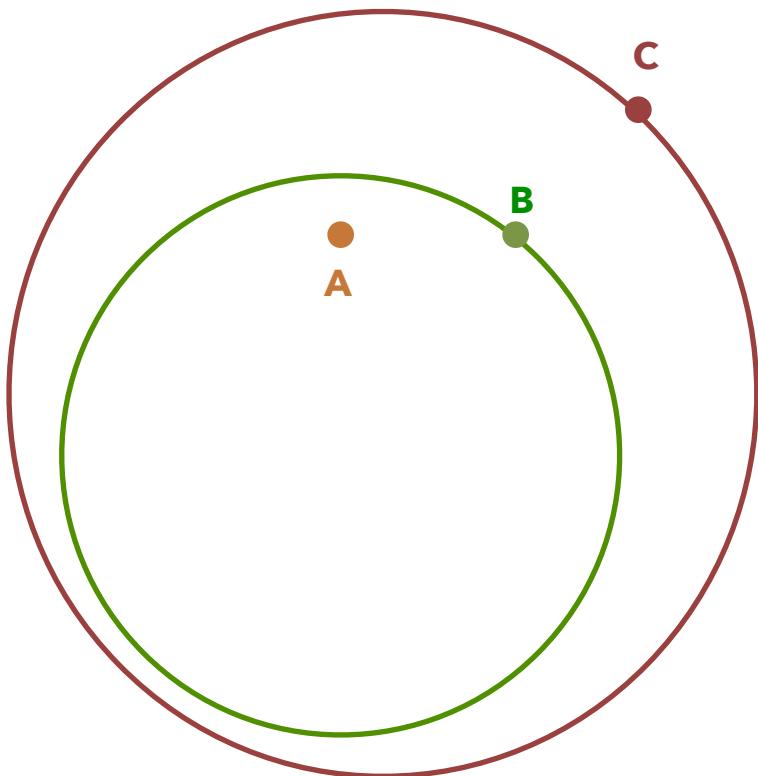
Una red feed-forward con 1 capa oculta de neuronas sigmoidales puede aproximar arbitrariamente bien cualquier función continua en $[0,1]^d$. Es decir, para cualquier función $f \in C([0,1]^d)$ y $\forall \epsilon$, existe una red neuronal con 1 capa oculta de neuronas sigmoidales $f_{\text{ANN}}(x)$ tal que

$$|f_{\text{ANN}}(x) - f(x)| < \epsilon$$

Cybenko, G. (1989) "Approximations by superpositions of sigmoidal functions", *Mathematics of Control, Signals, and Systems*, 2(4), 303–314. doi:10.1007/BF02551274



Universalidad de una Red Neuronal



C: Función deseada

B: Mejor función que la red puede implementar

A: Función que la red aprende desde el conjunto (finito) de ejemplos



Cybenko, G. (1989) "Approximations by superpositions of sigmoidal functions", *Mathematics of Control, Signals, and Systems*, 2(4), 303–314. doi:10.1007/BF02551274

Universalidad de una Red Neuronal

- Hornik en 1991 extiende el resultado de Cybenko considerando funciones de activación no necesariamente sigmoidales con las siguientes propiedades:
 - es continua
 - no es constante
 - es acotada



Approximation Capabilities of Multilayer Feedforward Networks

KURT HORNIK

Technische Universität Wien, Vienna, Austria

(Received 30 January 1990; revised and accepted 25 October 1990)

Abstract—We show that standard multilayer feedforward networks with as few as a single hidden layer and arbitrary bounded and nonconstant activation function are universal approximators with respect to $L^p(\mu)$ performance criteria, for arbitrary finite input environment measures μ , provided only that sufficiently many hidden units are available. If the activation function is continuous, bounded and nonconstant, then continuous mappings can be learned uniformly over compact input sets. We also give very general conditions ensuring that networks with sufficiently smooth activation functions are capable of arbitrarily accurate approximation to a function and its derivatives.

- Notemos que ese resultado excluye ReLU!

Universalidad de una Red Neuronal

- Varias versiones actuales del teorema permiten el uso de funciones de activación no acotadas como ReLU.



Neural Network with Unbounded Activation Functions is
Universal Approximator

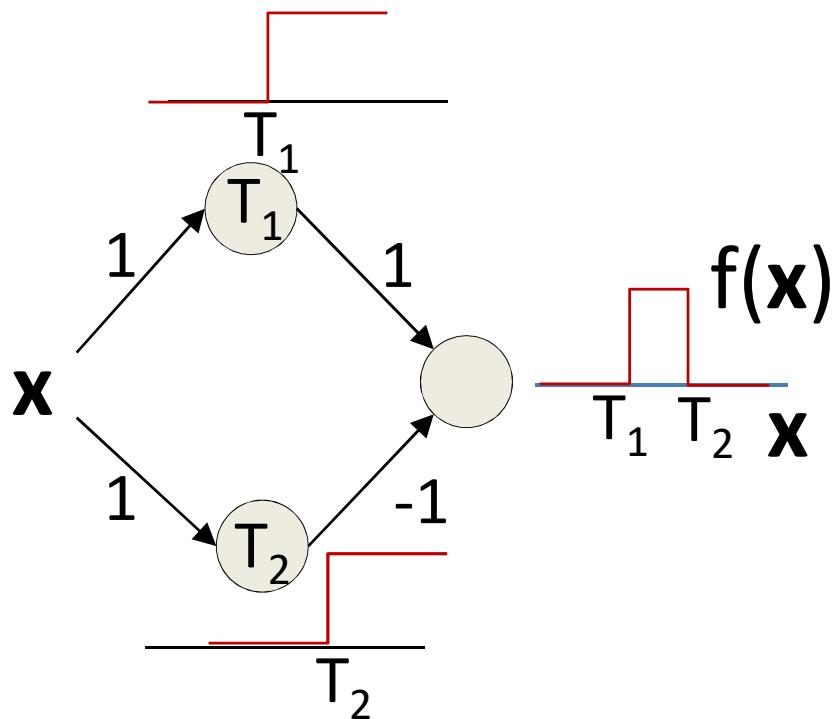
Sho Sonoda^{*1} and Noboru Murata¹

¹Faculty of Science and Engineering, Waseda University

- Lo que permite a la red "crecer" desde discriminantes lineales a una familia mucho más grande es la presencia de una capa oculta de neuronas.

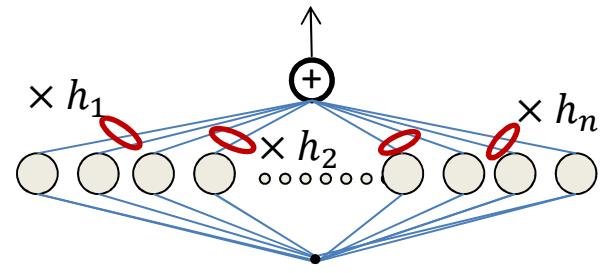
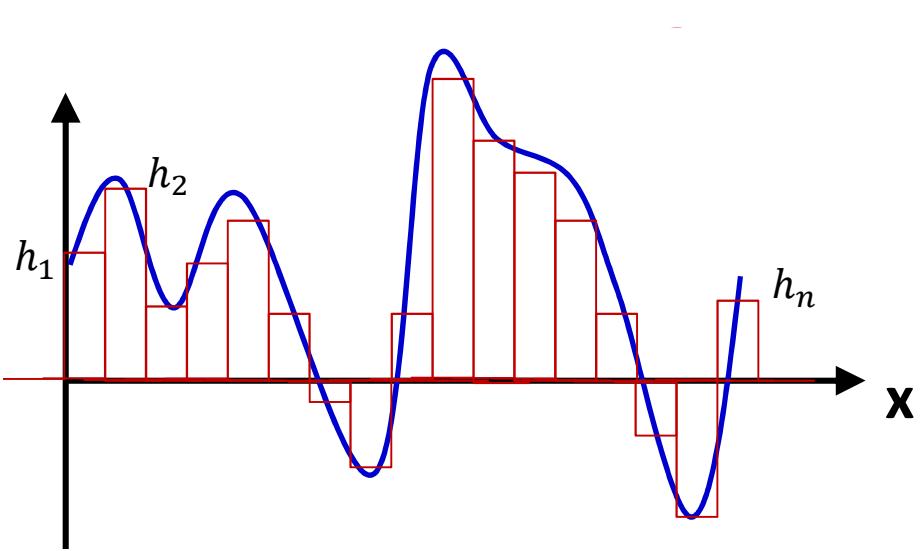
Intuición en Regresión

- Combinando las salidas de 2 o más neuronas ocultas podemos crear un pequeño “detector” de una región de dominio de \mathbf{x} que escalado apropiadamente permite aproximar la respuesta (\mathbf{y}) en esa región.



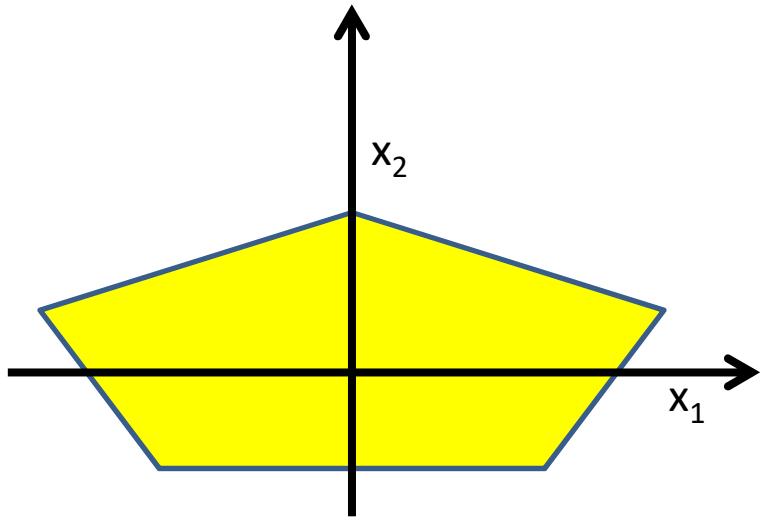
Intuición en Regresión

- Combinando estos detectores en capas superiores podemos aproximar cualquier función continua arbitrariamente bien.



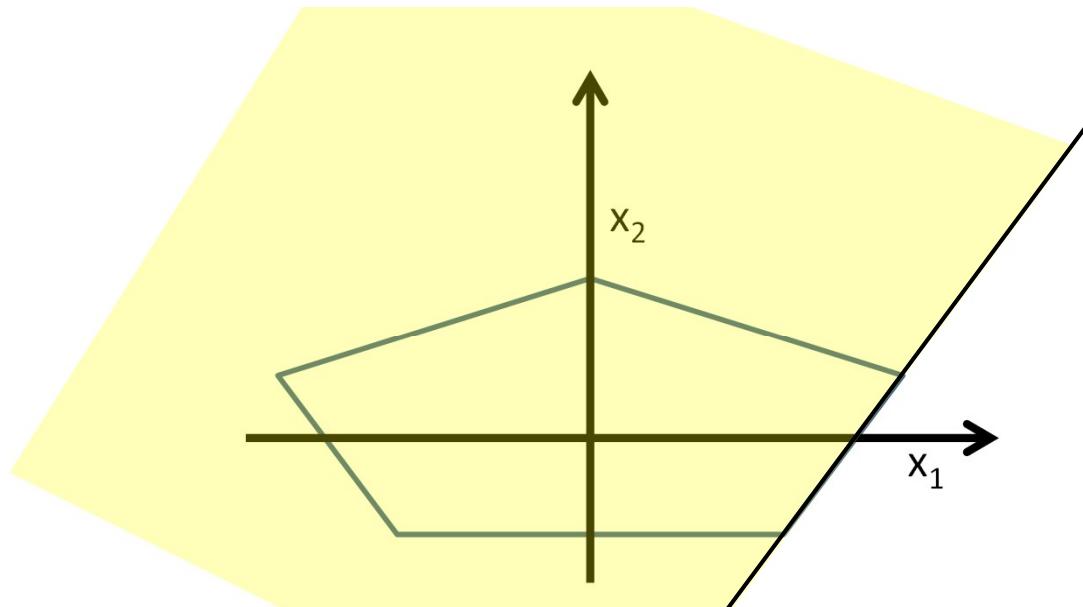
Intuición en Clasificación

- Combinando las regiones que detecta una neurona MP en el espacio, podemos componer regiones arbitrariamente complejas.



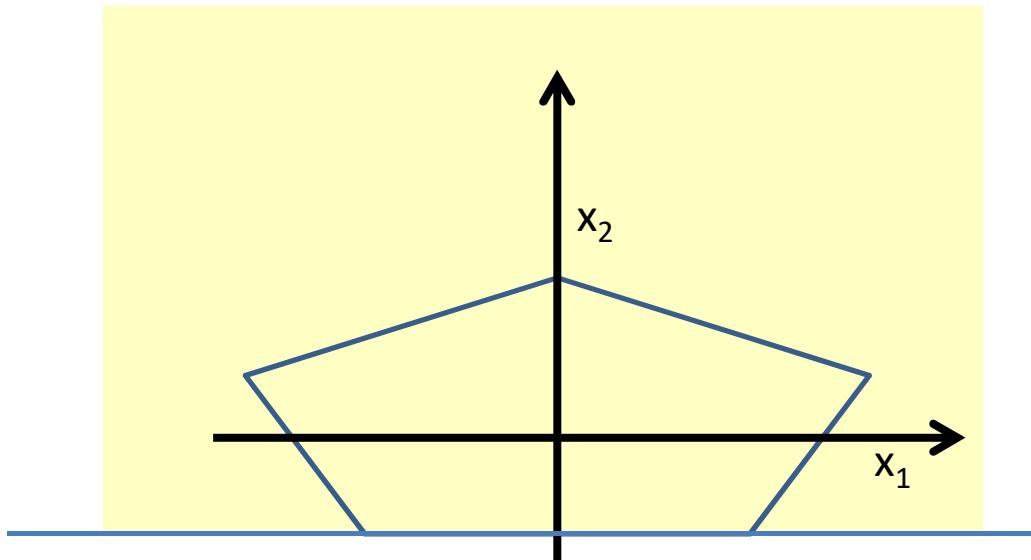
Intuición en Clasificación

- Combinando las regiones que detecta una neurona MP en el espacio, podemos componer regiones arbitrariamente complejas.



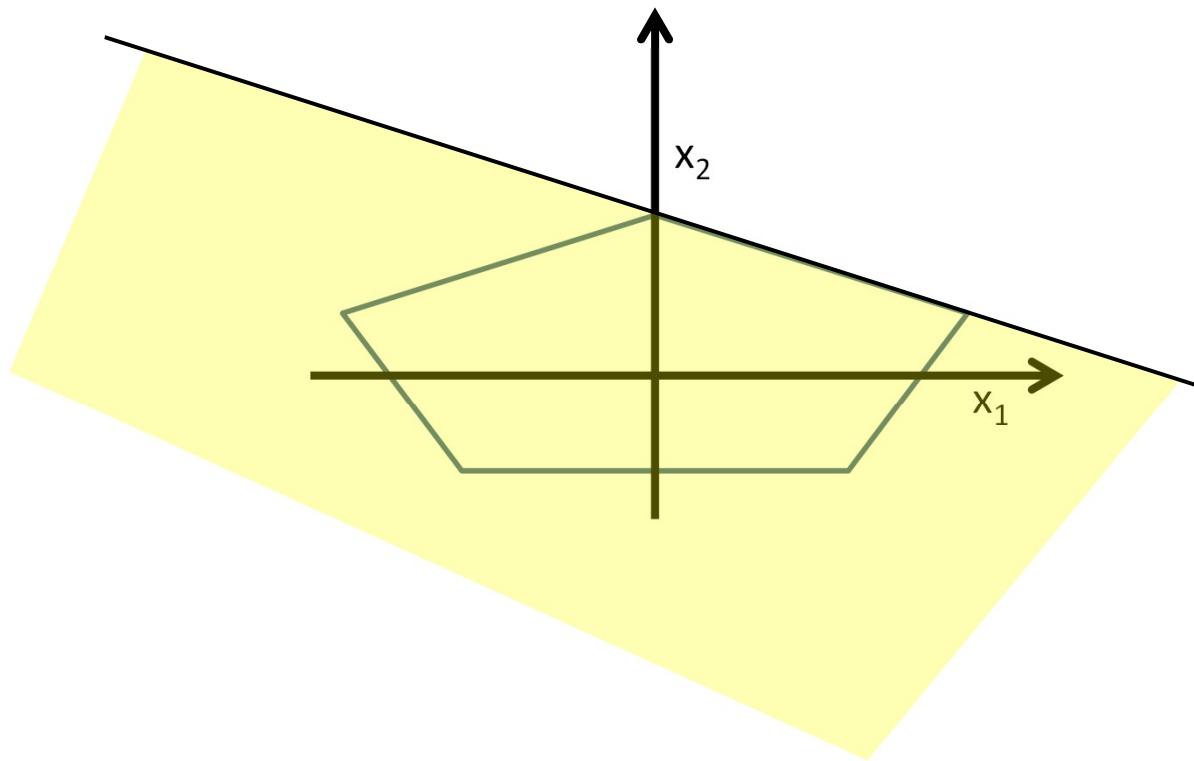
Intuición en Clasificación

- Combinando las regiones que detecta una neurona MP en el espacio, podemos componer regiones arbitrariamente complejas.



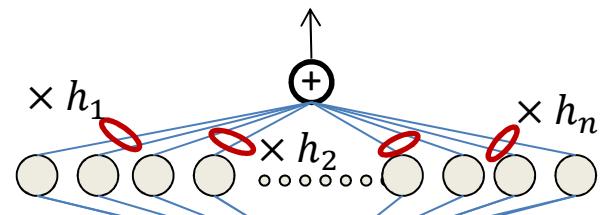
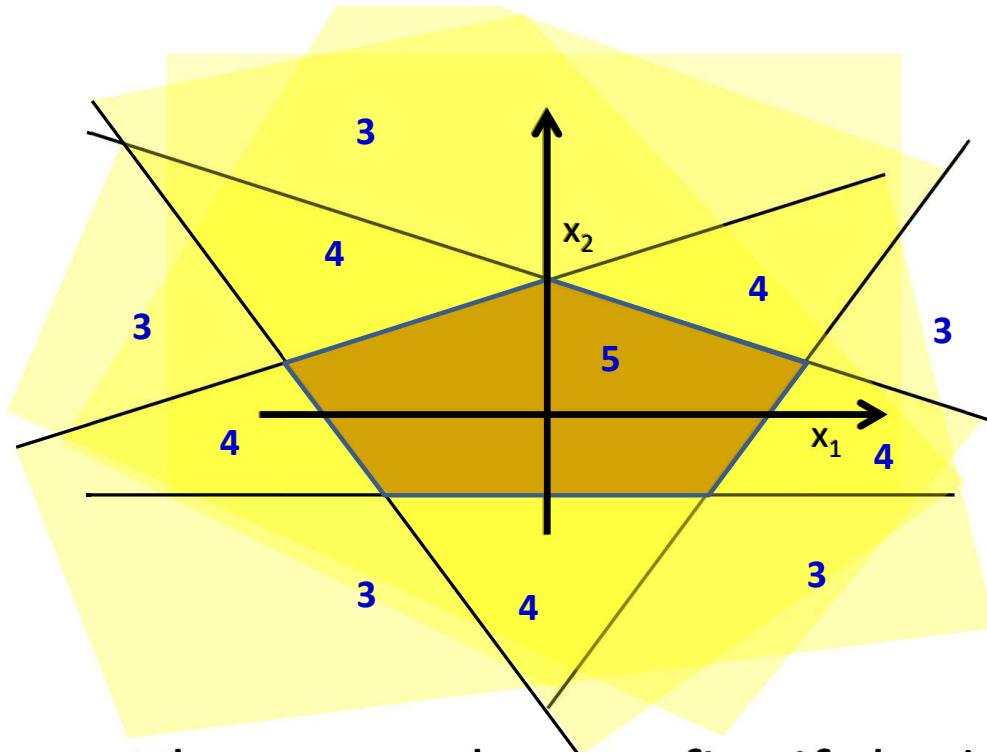
Intuición en Clasificación

- Combinando las regiones que detecta una neurona MP en el espacio, podemos componer regiones arbitrariamente complejas.



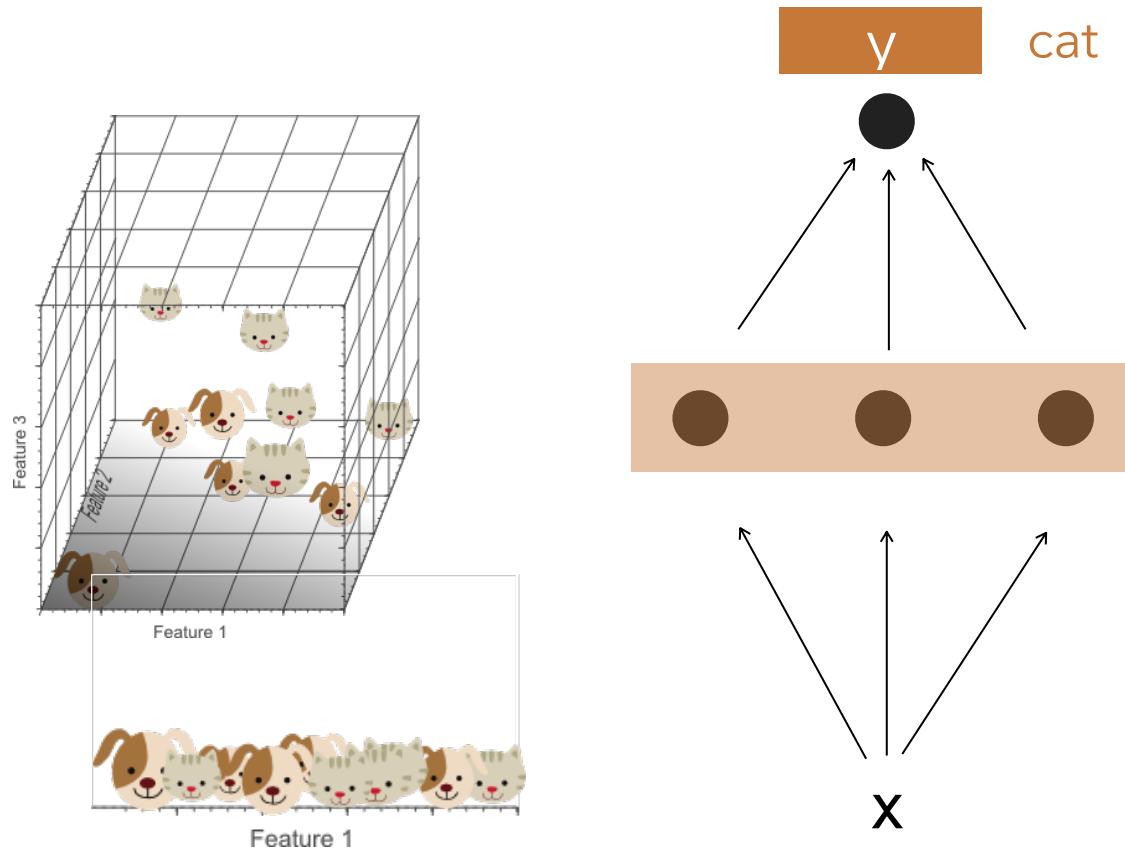
Intuición en Clasificación

- Combinando las regiones que detecta una neurona MP en el espacio, podemos componer regiones arbitrariamente complejas.



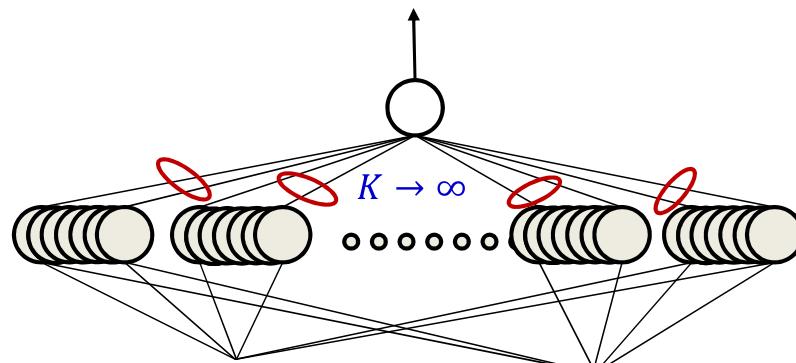
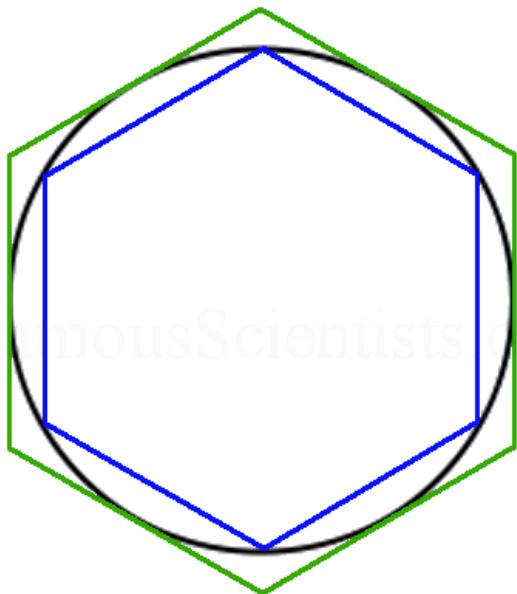
Intuición en Clasificación

- Esto es exactamente lo que decíamos cuando mencionamos la intuición del cambio de representación.



Universalidad de una Red Neuronal

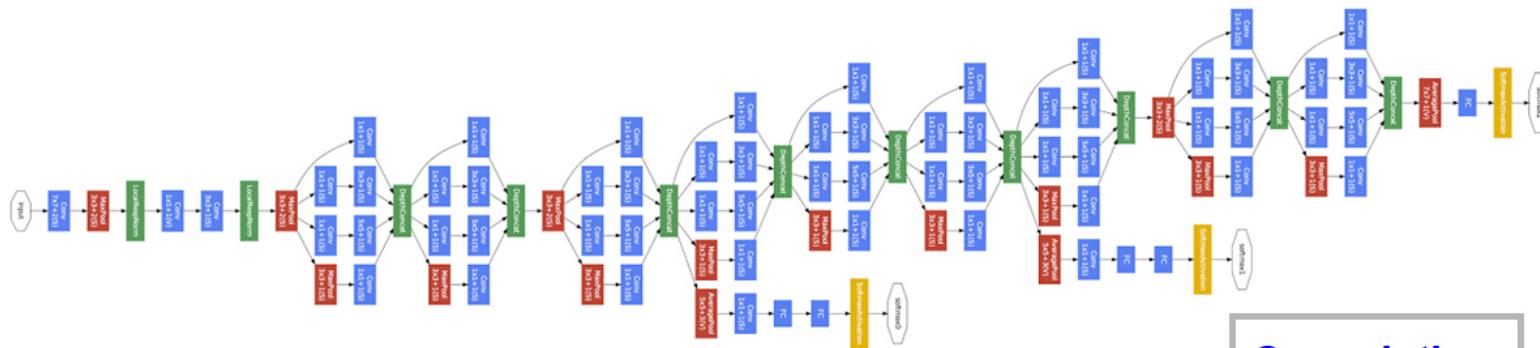
- Nuevamente, la universalidad podría costar un número muy grande de neuronas.



- Un número exponencialmente grande de neuronas se traduce en un número exponencial o super-exponencialmente grande de parámetros.

Profundidad

- Si la profundidad nos permitió obtener aproximados universales, ¿puede ésta ayudarnos a reducir la complejidad de la red en ciertas tareas complejas?



**Convolution
Pooling
Softmax
Other**

- En la práctica, es lo que observamos.

El Caso de la Función de Paridad

- Volvamos a considerar el problema de la paridad y su resolución mediante redes neuronales.

$$f(x) = \bigoplus_{i=1}^d x_i = \left(x_1 \bigoplus \left(\left(\left(x_2 \bigoplus \dots \bigoplus \left(x_{d-1} \bigoplus x_d \right) \right) \right) \right) \right) \\ \left(\left(\left(x_1 \bigoplus x_2 \right) \bigoplus \dots \bigoplus x_{d-1} \right) \bigoplus x_d \right)$$

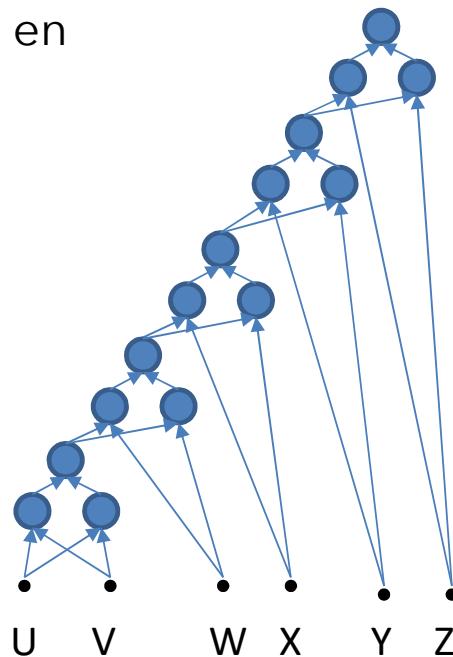
- Hemos visto que la red de 2 capas construida mediante la técnica de la FND puede requerir un número exponencialmente grande de neuronas (en d).

¿Existen redes más profundas que puedan resolver este problema con un número menor de neuronas?



Red Profunda para el Problema de la Paridad

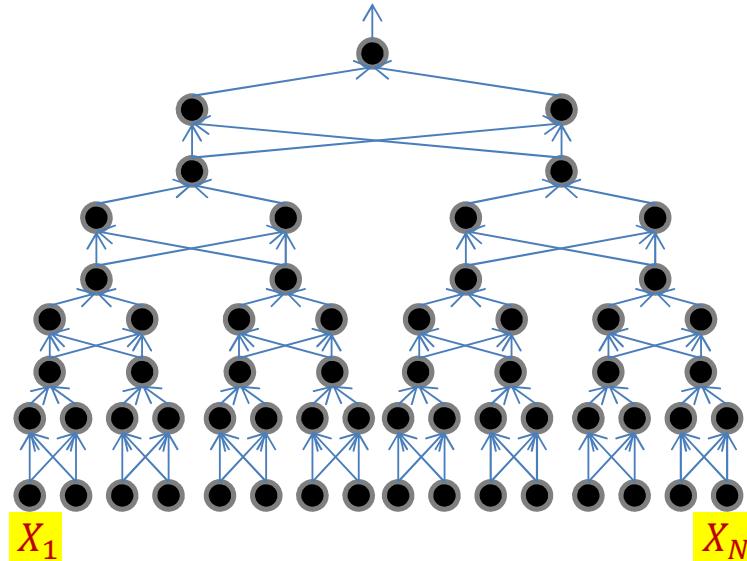
- La definición del XOR sugiere en efecto un diseño en cascada.



- **Esta red usa sólo $O(d)$ neuronas organizadas en $O(d)$ capas!**

Red Profunda para el Problema de la Paridad

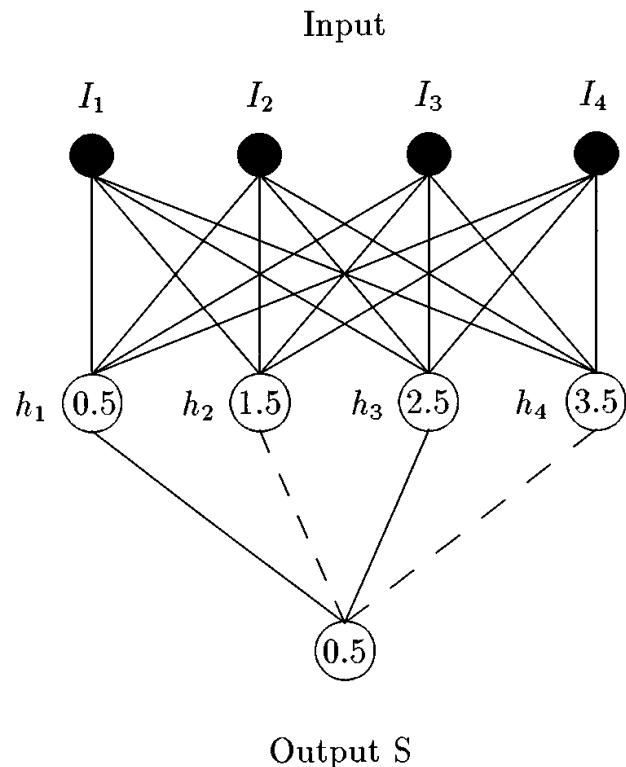
- Organizando el cómputo mediante conectividad local mejoramos aún más el resultado (notemos que el esquema recuerda el truco detrás de la *Transformada Rápida de Fourier*)



- Esta red usa sólo $O(d)$ neuronas organizadas en $O(\log(d))$ capas!**

Red Shallow para el Problema de la Paridad

- Lamentablemente para los entusiastas de la profundidad, es posible encontrar una red mucho menos profunda (1 capa oculta) que resuelve el problema usando también $O(d)$ neuronas.



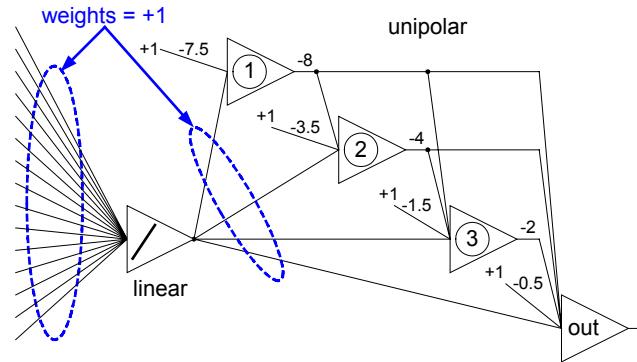
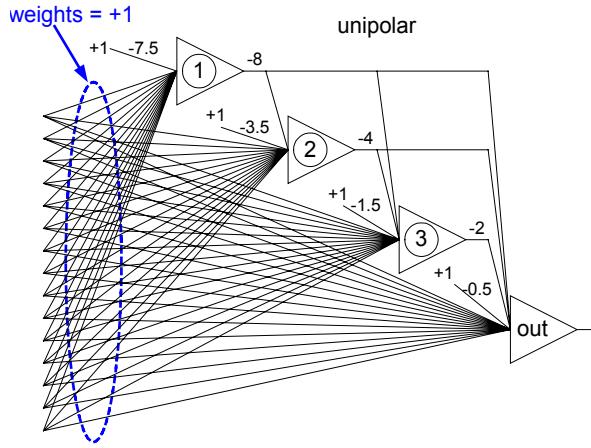
- Notemos que esta red usa más conexiones (parámetros) que su versión profunda.

Leonardo Franco and Sergio Alejandro Cannas. Generalization Properties of Modular Networks: Implementing the Parity Function. 2001.

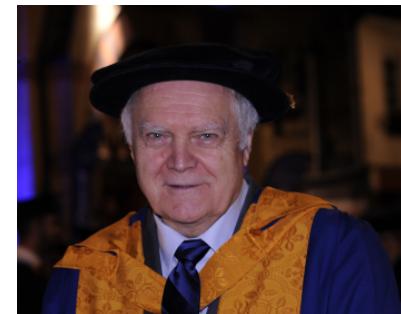


Red Profunda para e Problema de la Paridad

- Felizmente para los entusiastas de la profundidad, es posible diseñar una red profunda que resuelve el problema usando sólo $O(\log(d))$ neuronas!



Wilamowski, B. M., Hunter, D., Malinowski, A. (2003, July). Solving parity-N problems with feedforward neural networks. In Proceedings of the International Joint Conference on Neural Networks, 2003. (Vol. 4, pp. 2546-2551). IEEE.



Red Profunda para el Problema de la Paridad

- El resultado anterior no puede ser igualado por una red de 2 capas. Como muestra *Impagliazzo et al.* en 1997, se requieren al menos \sqrt{d} neuronas!

Conclusión

Existen problemas que una red profunda puede resolver más eficientemente (en términos de neuronas o parámetros utilizados) que cualquier red de 1 sola capa oculta.

Impagliazzo, R., Paturi, R., Saks, M. E. (1997).
Size-Depth Tradeoffs for Threshold Circuits.
SIAM Journal on Computing, 26(3), 693-707.



Red Profunda para el Problema de la Paridad

- Lamentablemente, la observación anterior no puede demostrarse válida siempre (a no ser que consideremos restricciones del problema):

Teorema (Shannon, 1949)

$\forall d$ existen funciones booleanas de d variables, para las cuales no es posible construir un circuito con menos de $2^d/2d$ compuertas.

- Adaptado a redes MP por ejemplo en Neciporuk 1964.

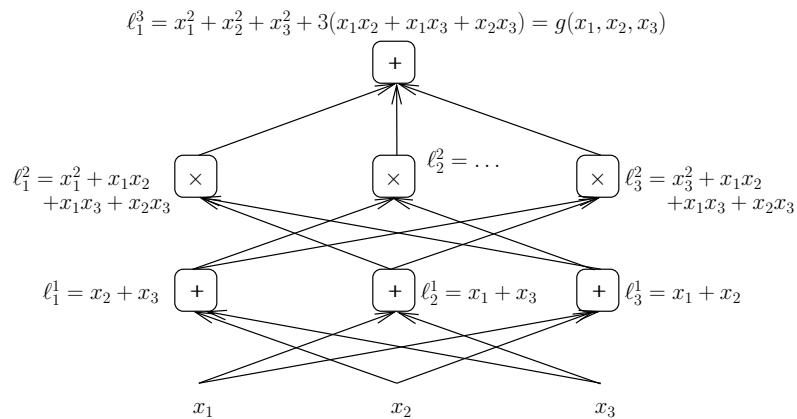
Shannon, Claude E. "The synthesis of two-terminal switching circuits." *The Bell System Technical Journal* 28.1 (1949): 59-98.

Neciporuk, E.I. (1964) The Synthesis of Networks from Threshold Elements. Soviet Mathematics 5, 163-166. English trans]. (1964) Automation Express 7, 27-32 & 35-39



Beneficios de la Profundidad

- En el caso continuo, se ha buscado intensamente caracterizar los beneficios de la profundidad, pero muchos de los resultados valen sólo para sub-clases de redes y funciones.
- Por ejemplo, *Delalleau y Bengio* han estudiado mucho una clase de red denominada *suma-producto* con la siguiente forma:



Beneficios de la Profundidad

- Se ha logrado demostrar resultados como el siguiente:

Teorema

Existe una cierta clase interesante de funciones representable mediante una red suma-producto profunda con $\mathcal{O}(d)$ unidades, para las cuales, una red de 1 sola capa oculta requeriría $\mathcal{O}(2^{\sqrt{d}})$ unidades.

Delalleau, Olivier, and Yoshua Bengio. "Shallow vs. deep sum-product networks." *Advances in neural information processing systems*. 2011.



Beneficios de la Profundidad

- Se ha logrado demostrar resultados como el siguiente:

Teorema

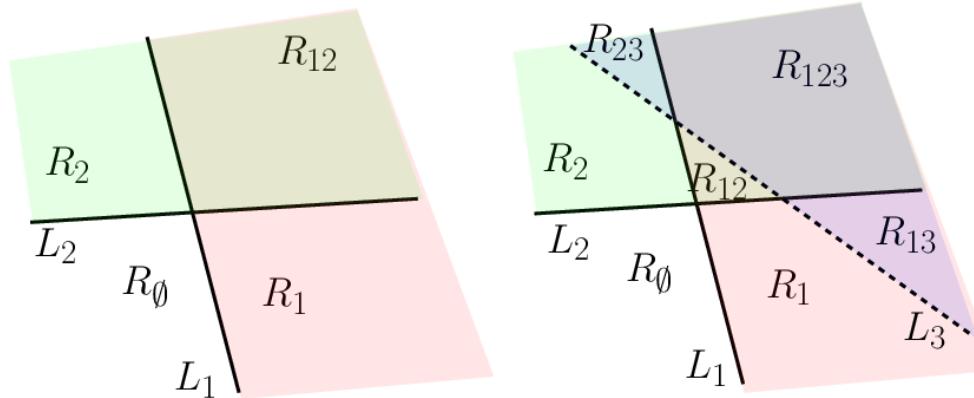
Existe una cierta clase interesante de funciones representable mediante una red suma-producto de profundidad k con $\mathcal{O}(kd)$ unidades, para las cuales, una red de 1 sola capa oculta requeriría $\mathcal{O}((d - 1)^k)$ unidades.

Delalleau, Olivier, and Yoshua Bengio. "Shallow vs. deep sum-product networks." *Advances in neural information processing systems*. 2011.



Beneficios de la Profundidad

- Se han encontrado resultados un poco más generales, considerando el número de **regiones diferentes** que induce la red sobre el espacio de entrada X .
- Por ejemplo, en redes que usan neuronas ReLU es posible contar el **número de regiones lineales** (regiones donde la función implementada es lineal) en que queda dividido X por la red.



Beneficios de la Profundidad

- Pascanu demostró en 2013 el siguiente teorema:

Teorema

El número de regiones lineales en una red de 1 capa oculta con N neuronas ReLU es a lo más $\sum_{j=0}^d \binom{N}{j}$.



Razvan Pascanu 2013. *On the Number of Response Regions of Deep Neural Networks with Piecewise Linear Activations.*

Beneficios de la Profundidad

- Montufar demostró en 2014 el siguiente teorema:

Teorema

El número de regiones lineales en una red de K capas ocultas con n_i neuronas ReLU cada una es al menos

$$\left(\prod_{i=1}^{K-1} \left\lfloor \frac{n_i}{d} \right\rfloor^d \right) \sum_{j=0}^d \binom{n_K}{j}$$

Guido Montufar 2014. *On the Number of Linear Regions of Deep Neural Networks.*



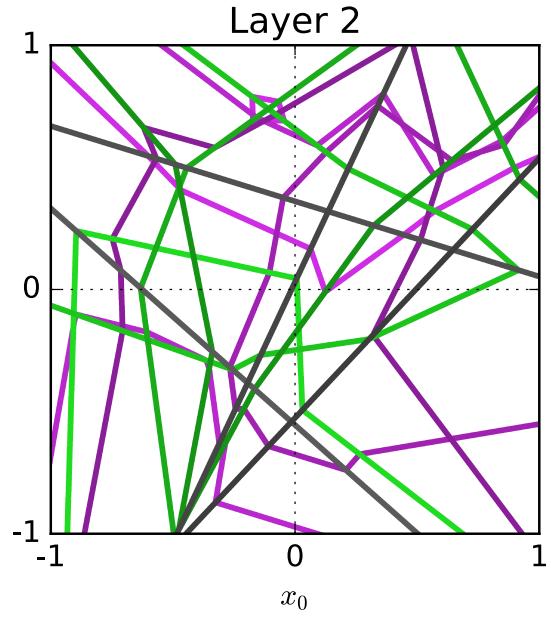
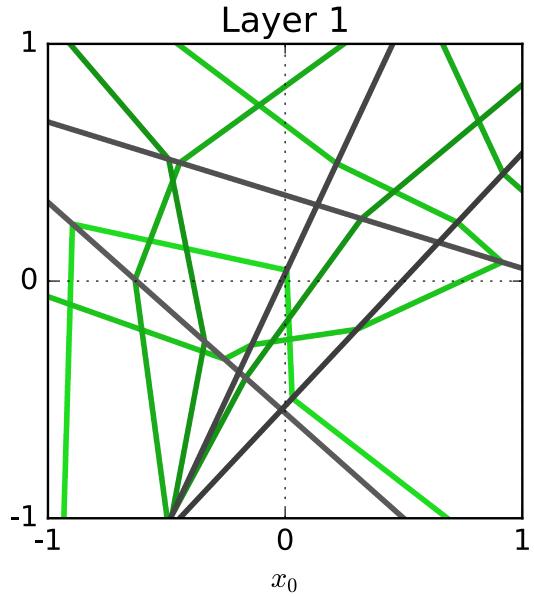
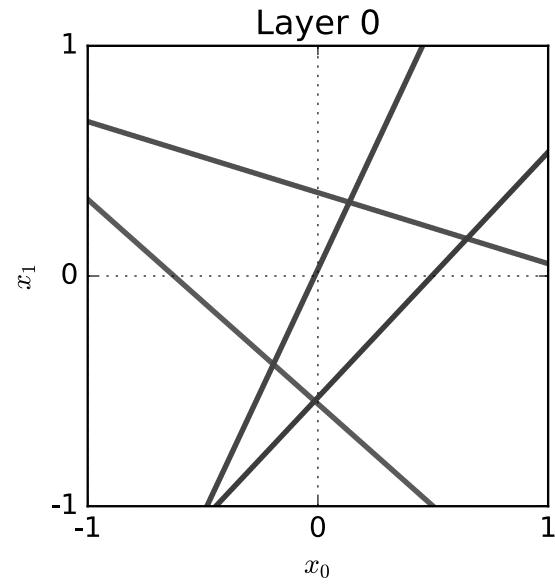
Beneficios de la Profundidad

- Si hacemos $n_i = n > 2d$ para cada capa de la red profunda, tenemos que el número de regiones crece como $2^{d(K-1)}n^d$.
- En cambio, para la red de 1 capa (para ser justos, en este caso le ponemos Kn neuronas), el número de regiones lineales crece como $\mathcal{O}(K^d n^d)$.
- En otras palabras: en una red de 1 capa el número de regiones aumenta a lo más polinomialmente en K y en n . En una red profunda, en cambio aumenta exponencialmente en K y polinomialmente en n . El cuociente entre los dos términos es

$$\theta = \frac{2^{d(K-1)}n^d}{K^d n^d} = \frac{2^{d(K-1)}}{K^d}$$

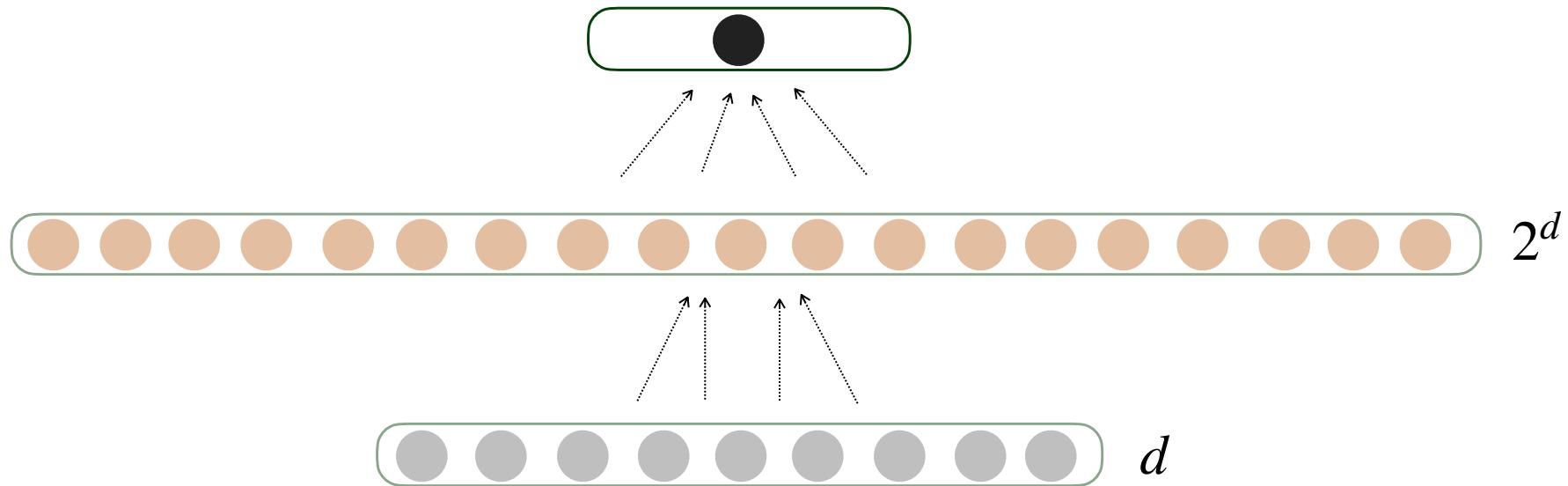
- Para $d=2$ y $K=5$, la red profunda es al menos 10 veces más potente. Para $d=10$ y $K=5$, la red profunda es 100.000 veces más potente.

Intuición



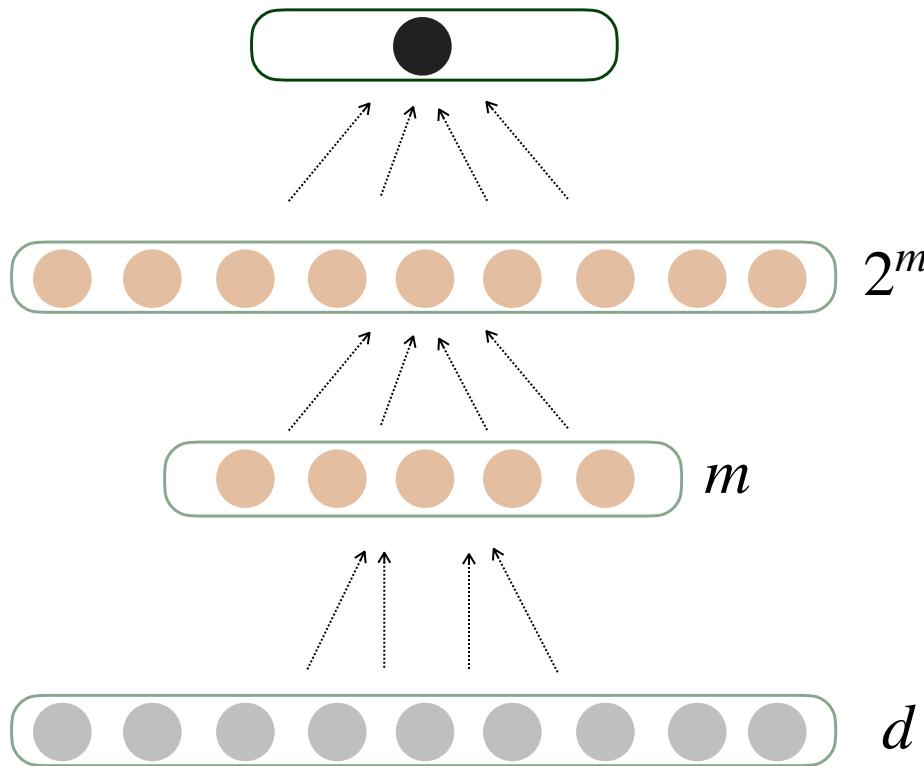
Intuición

- Hemos visto que una red no profunda puede requerir un número exponencialmente grande neuronas en d : la dimensionalidad de input.



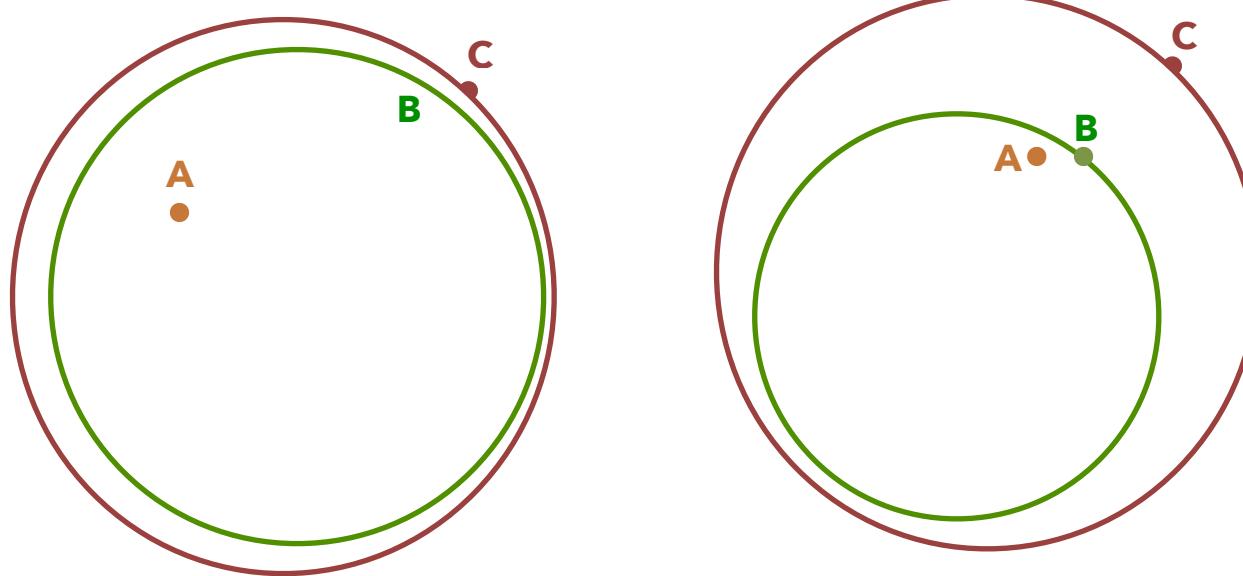
Intuición

- ¿Qué tal si antes reducimos dimensionalidad?



Error de Aproximación versus Error de Estimación

- Entonces, todo parece indicar que una red profunda goza de una mayor capacidad de aproximación que una red no profunda cuando se limita el número de neuronas.



C: Función deseada.

B: Mejor función implementable por una red profunda de N neuronas

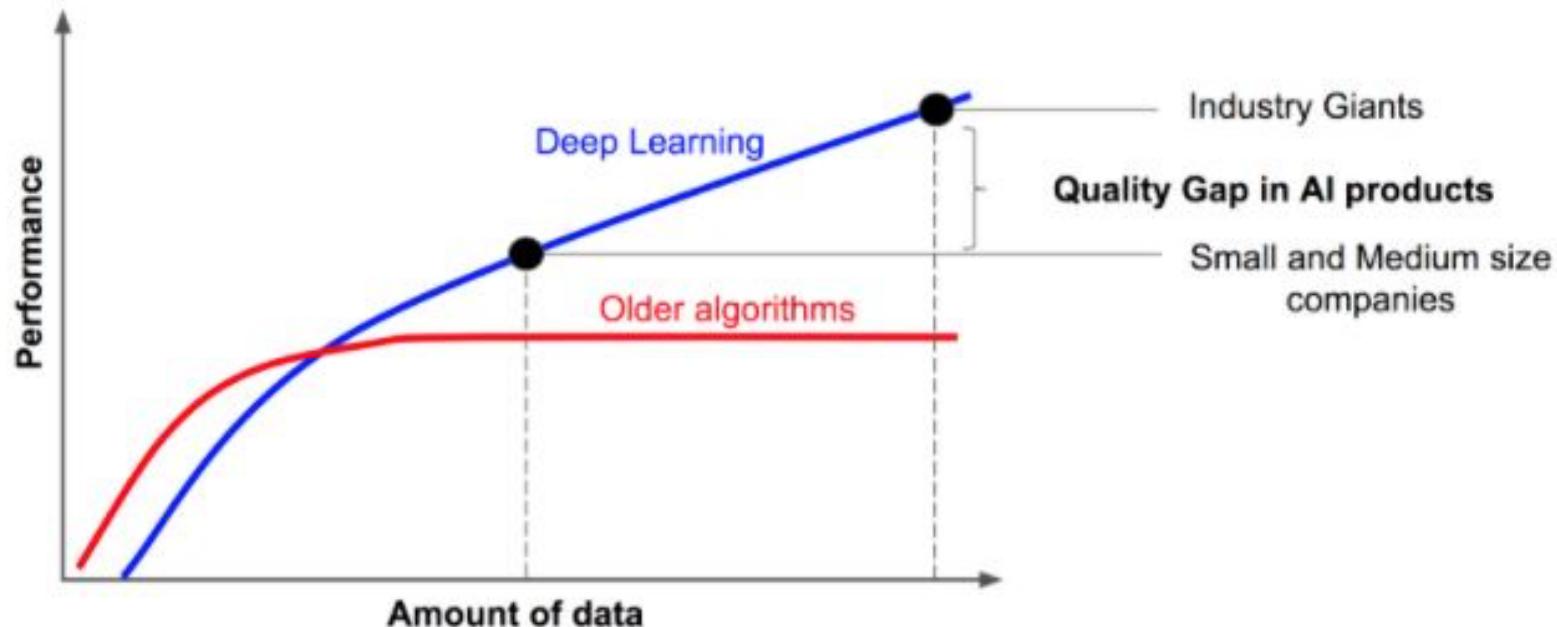
A: Mejor función implementable por una red 1 sólo capa oculta de N neuronas

B': Función aprendida en B con conjunto finito de M ejemplos

A': Función aprendida en A con conjunto finito de M ejemplos

Error de Aproximación versus Error de Estimación

- El mayor poder expresivo (con un número limitado de unidades), le permite a una red profunda aprovechar mejor la disponibilidad de grandes conjuntos de ejemplos.



Entonces ...

- Una red neuronal es un **aproximador universal** tanto si la analizamos en el contexto de inputs-outputs binarios como en el caso continuo, lo que significa que, si tenemos suficientes datos, podemos aprender prácticamente cualquier función.
- El número de neuronas (y parámetros) requerido para la aproximación podría ser exponencialmente grande en la dimensión del vector de entrada.
- La profundidad puede ayudar a reducir esa complejidad.
- Muchos resultados experimentales confirman esta idea y una serie de trabajos recientes ha logrado formalizarla, al menos en parte.

