

Deep Learning

01 Introducción

Miguel A.
Castellanos

**“Cualquier tecnología
suficientemente avanzada es
indistinguible de la magia”**

Arthur C. Clarke

Lo que se ha conseguido en Tecnología parece “magia”

Las aplicaciones parece que funcionan “mágicamente”

Habrà cosas en el curso que usaremos de forma “mágica”



DeepL Traductor

DeepL Pro

Planes y precios

Traducir texto

Traducir archivos .docx y .pptx

Traducir de cualquier idioma

Tradu

Escribe o pega el texto aquí.

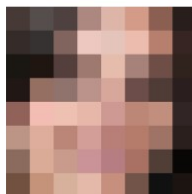
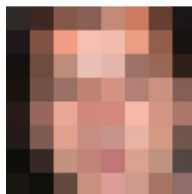
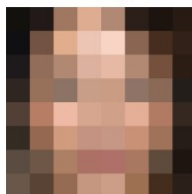
Arrastra hasta aquí un archivo para su traducción. Formatos disponibles: Microsoft Word (.docx) y PowerPoint (.pptx).

Combinaciones populares del Traductor de DeepL con español: alemán-español, español-francés e inglés-español.

8 × 8 input

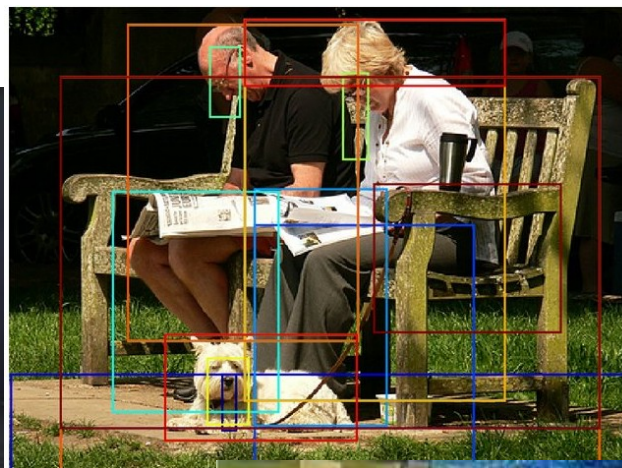
32 × 32 samples

ground truth



Our Result



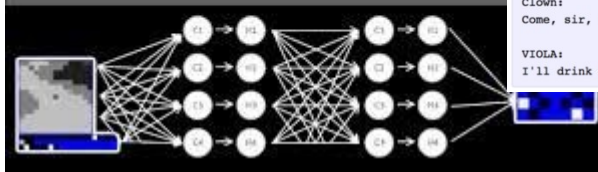
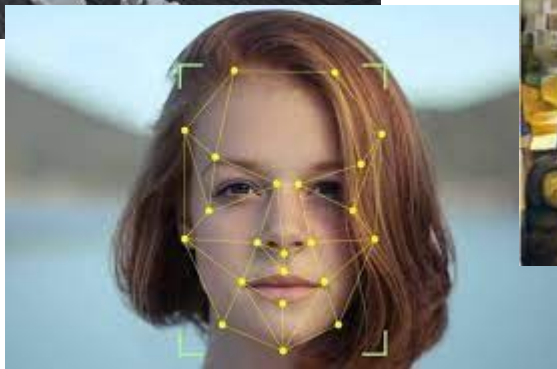
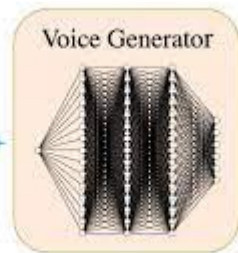


groundtruth

(dog, has, nose) (676)
 (bench, on, concrete) (40)
 (woman, wearing, pant) (822)
 (woman, holding, magazine) (5)
 (man, reading, newspaper) (56)
 (man, wearing, glasses) (2489)
 (woman, wearing, glasses) (882)
 (dog, has, face) (128)
 (woman, sitting on, bench) (357)
 (man, sitting on, bench) (653)
 (dog, next to, woman) (4)
 (bench, has, arm) (58)

prediction

(dog, has, nose) (True)
 (bench, on, ground) (False)
 (woman, holding, leg) (False)
 (woman, holding, paper) (False)
 (man, holding, book) (False)
 (man, wearing, glasses) (True)
 (wc
 (do
 (wc
 (ma
 (do
 (be



PANDARUS:

Alas, I think he shall be come approached and the day
 When little strain would be attain'd into being never fed,
 And who is but a chain and subjects of his death,
 I should not sleep.

Second Senator:

They are away this miseries, produced upon my soul,
 Breaking and strongly should be buried, when I perish
 The earth and thoughts of many states.

DUKE VINCENTIO:

Well, your wit is in the care of side and that.

Second Lord:

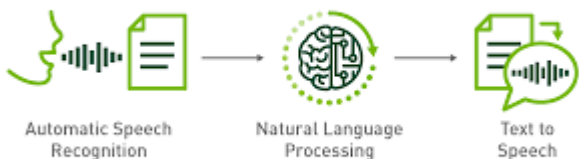
They would be ruled after this chamber, and
 my fair nues begun out of the fact, to be conveyed,
 Whose noble souls I'll have the heart of the wars.

Clown:

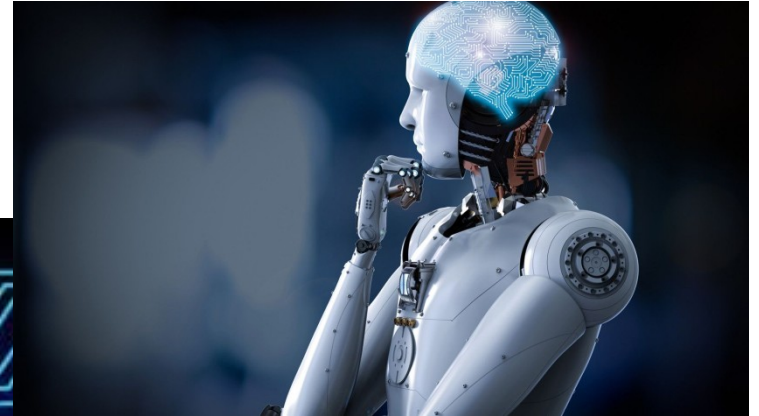
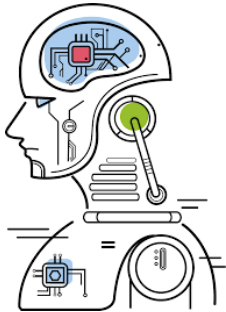
Come, sir, I will make did behold your worship.

VIOLA:

I'll drink it.



Todo lo anterior es debido al desarrollo en los últimos años de la ***Inteligencia Artificial (IA)***



No entramos en cuestiones filosóficas sobre la IA, pero una pequeña mención:

- Una IA fuerte
- Una IA débil

“Por ahora” lo único que le interesa al **Deep Learning** es una IA **débil** que resuelve problemas técnicos

“por ahora”

Dos tipos de IA:

Simbólica:

Alto nivel


Lógica

Conocimiento explícito

Funcionalismo

Ejemplos: Redes semánticas,
heurísticos, sistemas expertos

Filosofía, Psicología
Cognitiva,
Pensamiento, Lenguaje



Sub-simbólica:

Bajo nivel


Matemáticas, ecuaciones

Conocimiento implícito

Conexionismo

Ejemplos: matrices, optimización
numérica, Deep learning, Algoritmos
genéticos, machine learning

Matemáticos,
Ingenieros, Ciencias de
la Computación



- En los 80, 90 y 2000 triunfaba la primera pero desde 2010 la paliza de la segunda es tremenda.
 - Ahora hay una IA basada en *Machine Learning* y *Deep Learning*.
-

- Por supuesto, las cosas pueden cambiar (no a corto plazo).
- No quiere decir que lo simbólico sea falso, solo que el modelo “*cognitivo humano*” no está siendo útil.
- Sigue siendo importante saber “cómo” pensamos.

← Ciencia

Tipos de Aprendizaje automático:

Las que vamos a estudiar



- **Supervisado**: para el conjunto de entrenamiento es posible saber si la red lo hace bien o mal, y con ese conocimiento se ajustan los valores de la red. De cada entrada se conoce su valor en y , su clasificación o su etiqueta. La red se ajusta por la *diferencia entre y e y'* .
- **No supervisado**: No tenemos la información anterior, la red no se puede ajustar por la diferencia entre el resultado y la realidad. Son técnicas de agrupamiento y *clustering*, por ejemplo, las SOM de *Kohonen*.
- **Por refuerzo**: Cada algoritmo es recompensado en función de sus resultados en múltiples tareas de manera que el algoritmo va actualizándose para optimizar las recompensas. Por ejemplo: *Programación dinámica*.

Un poco de historia:

- Para revisar la historia recomiendo este enlace:
<https://telefonicatech.com/blog/una-breve-historia-del-machine-learning>
- Solo mencionar a los padres de la neurona artificial: **McCulloch and Pitts**



Una amistad improbable pero fructífera

https://www.eldiario.es/hojaderouter/ciencia/walter-pitts-mcculloch-pioneros-cibernetica-inteligencia-artificial_1_4320200.html

Y a **Rosenblatt**, creador del perceptron (1969) que es el modelo de neurona más utilizado

- A partir del 2008 aparece el Deep Learning y la explosión de aplicaciones
 - Nuevos algoritmos
 - Nuevo hardware (GPUs)
- 2017. Aparecen los Transformers
 - Nuevas estrategias de Aprendizaje (Atención)
- Actualidad, explosión de todas las IAs Generativas
 - Imagen
 - Vídeo
 - NLP
 - Chatbots, ChatGPT, etc.



¿Tan importante va a ser la IA en el futuro?

- Lo más probable es que sí, están para quedarse
- Si primero fue la automatización del trabajo físico esto puede ser la automatización del trabajo intelectual
- Va a suponer fuertes implicaciones sociales y laborales (y probablemente sobreviviremos a ellos)



La nueva electricidad:

“Just as electricity transformed almost everything 100 years ago, today I actually have a hard time thinking of an industry that I don’t think AI will transform in the next several years”

Andrew Ng

Ventajas:

Aprendizaje adaptativo: aprenden a partir de un conjunto de experiencias. Por tanto no necesitan experto.

Auto-organización: crean su propia representación de la información que reciben durante el entrenamiento.

Tolerancia a fallos: pueden trabajar con información parcial debido a que codifican la información de modo distribuido y redundante.

Flexibilidad: pueden adaptarse a un nuevo entorno sin necesidad de ser reprogramadas, gracias a su capacidad de aprendizaje

Tiempo real: pueden implementarse en máquinas paralelas, alcanzándose velocidades de operación elevadas

Generalización: una vez que una red aprende, es capaz de clasificar objetos desconocidos. También son capaces de “saber que no saben”

Información imprecisa : Pueden trabajar con información probabilística, ruidosa o inconsistente

Inconvenientes:

Elecciones arbitrarias sobre la arquitectura de la red

Ajuste de hiper-parámetros por ensayo y error

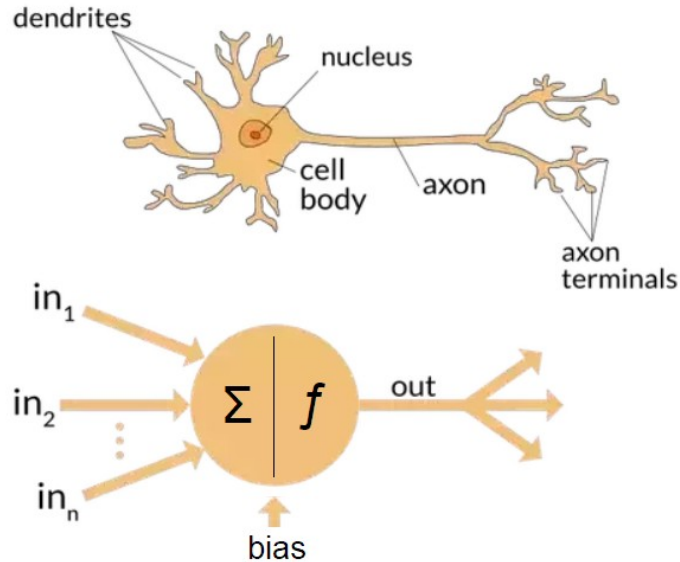
Aprendizaje lento y a veces dificultoso: entrenar es un arte

Necesidad masiva de datos

Ausencia de explicación de las decisiones

Imposibilidad de interpretar la red, solo se evalúa el resultado

Las RNA es una analogía de una **neurona**



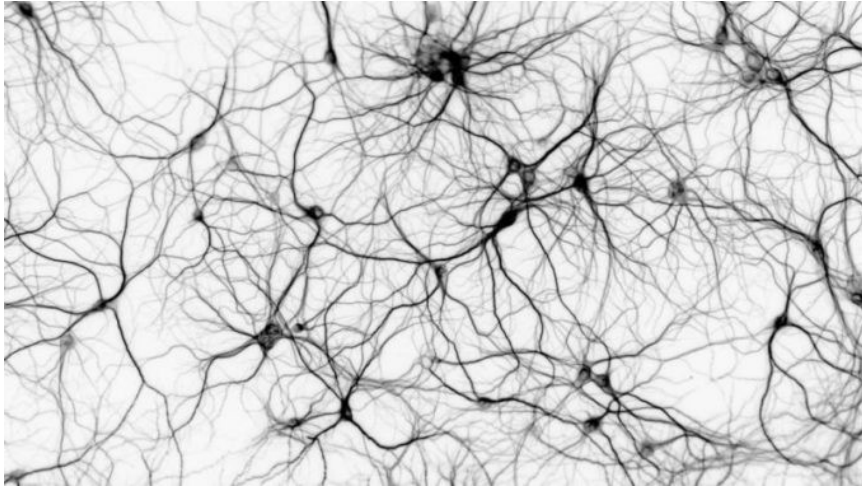
- Hay unas conexiones sinápticas que la activan (entrada)
- Hay una respuesta (salida)

Una neurona es muchísimo **más compleja**, una RNS es solo una simplificación

El modelo matemático puede **no ajustarse** lo suficiente.

- Respuesta gradual, pero no lineal. Efectúan una suma no lineal de las entradas. Para algunas funciones de activación esto se cumple.
- La neurona biológica produce una secuencia de pulsos, no un nivel constante. Lo que cambia según su estado es la frecuencia de activación de la neurona (firing rate).
- Las neuronas biológicas son asíncronas, no existe un reloj central. Esto puede simularse.
- El impulso transmitido por una sinapsis puede variar de modo estocástico

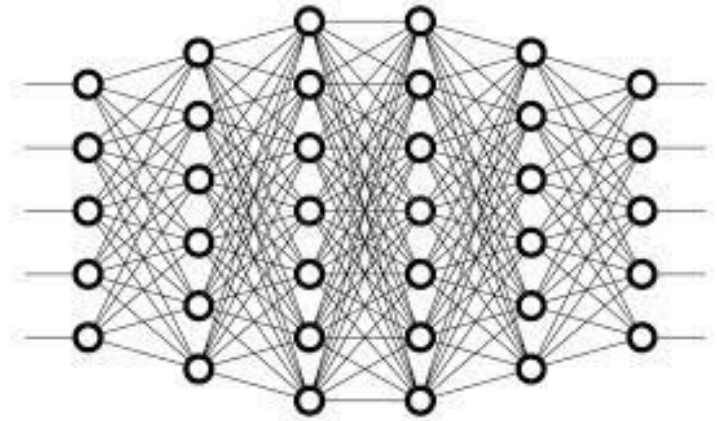
Y un **cerebro**



10^{11} neuronas con 10^{14} conexiones

Cada neurona está conectada a otras 1000 - 200000

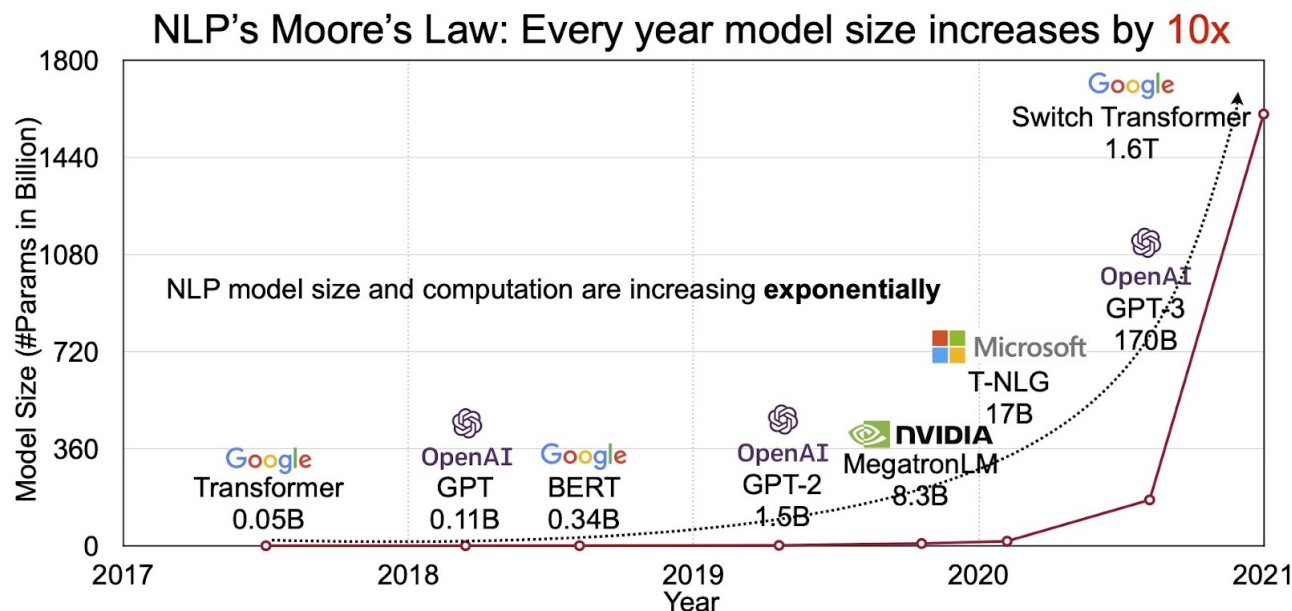
La tasa máxima de activación es de unos 1000 impulsos por minuto.



Las conexiones se autoorganizan en respuesta a sus entradas

El error en alguno de sus elementos no es esencial, pues hay un alto nivel de redundancia

Los LLMs (*Large Language Models*) empiezan a tener dimensiones similares al cerebro



Estos modelos son más *verborreicos* que inteligentes. Han sido entrenados con TODO lo escrito por la humanidad (literalmente) y sin embargo un niño de 3 años que solo ha leído *el pirata patapalo* y el *pollito Pepe* es más inteligente que estos modelos

Ojo! Eso ya **no es importante**

El objetivo **no es simular** un comportamiento biológico

El objetivo **es resolver problemas** prácticos usando *mega-ecuaciones*

Quien esté interesado **puede consultar:**

- <https://www.humanbrainproject.eu/>



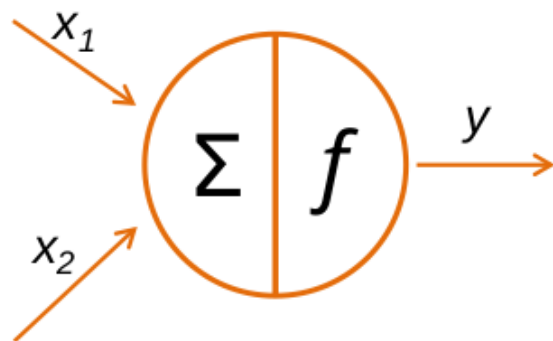
- <https://www.thevirtualbrain.org>



THEVIRTUALBRAIN.

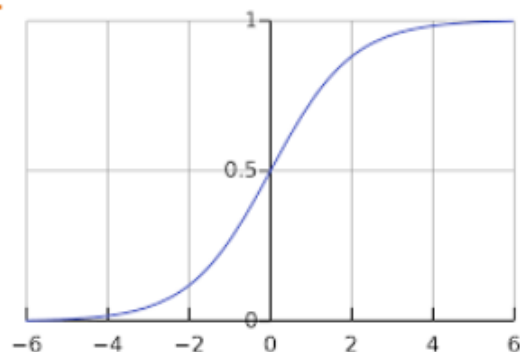
Toda neurona tiene dos componentes:

- Una **suma ponderada** de la entrada **z**
- Una **función de activación $f(z)$**



$$z = w_1x_1 + w_2x_2 + b$$

$$y = f(z)$$



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

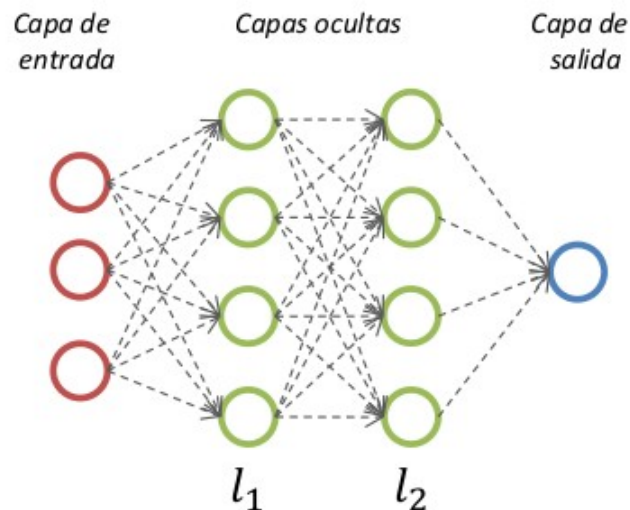
Si te fijas, no es más que una función logística (sigmoide) de las que se usan en TRI y en la **regresión logística**

$$Z = \sum W X + b$$

$$A = y = f(Z)$$

Se usa A en vez de y para indicar que no es la neurona final

Las redes neuronales se organizan en capas (*Hidden Layers*)



dimensiones

$$\left\{ \begin{array}{l} W^l: (n^l, n^{l-1}) \\ b^l: (n^l, 1) \\ Z^l: (n^l, m) \\ A^l: (n^l, m) \end{array} \right.$$

$$\begin{aligned} Z^{[1]} &= W^{[1]}X + b^{[1]} \\ A^{[1]} &= f^{[1]}(Z^{[1]}) \\ Z^{[2]} &= W^{[2]}A^{[1]} + b^{[2]} \\ A^{[2]} &= f^{[2]}(Z^{[2]}) \\ Z^{[3]} &= W^{[3]}A^{[2]} + b^{[3]} \\ Y' &= f^{[3]}(Z^{[3]}) \\ C &= L(Y, Y') \end{aligned}$$

$$\begin{aligned} Z^l &= W^l A^{l-1} + b^l \\ A^l &= f^l(Z^l) \end{aligned}$$

Siendo:

W^l : matriz de coeficientes del layer l

b^l : vector de sesgos del layer l

Z^l : suma ponderada del layer l

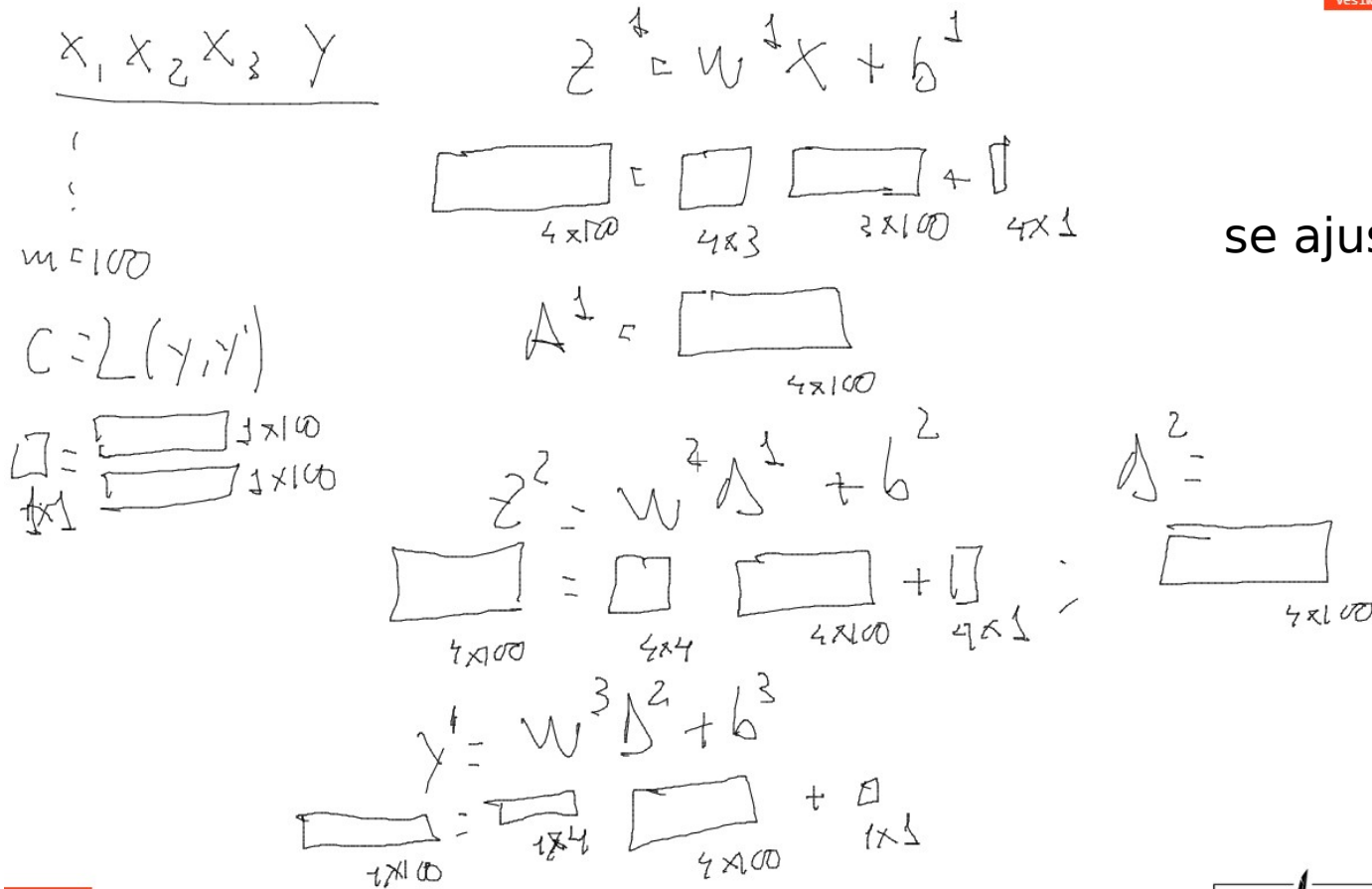
A^l : Activación del layer l

L : función de coste

n^l : n° neuronas en capa l

m : n° elementos en el set de aprendizaje

Ejemplo de las **matrices de la red anterior**:



Vesiani

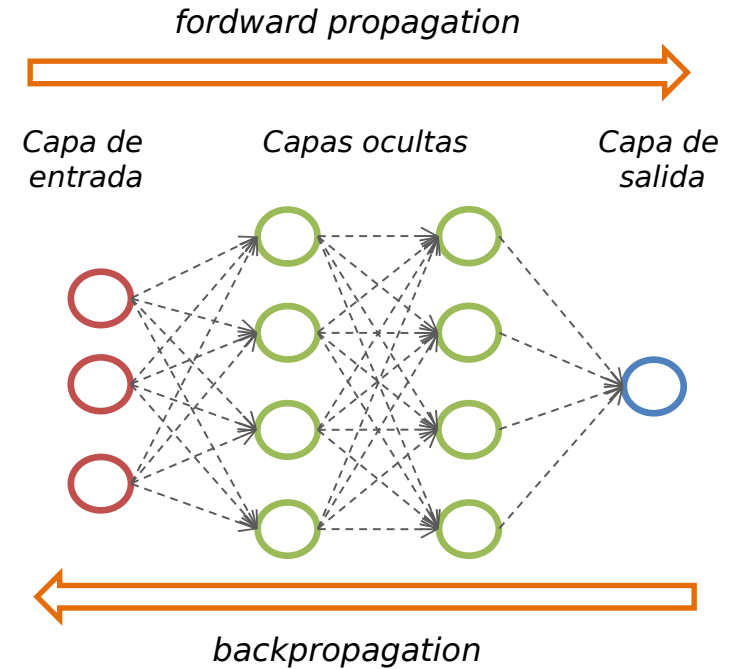
NO AGOBIARSE!

Todas estas matrices se ajustan automáticamente usando **Keras**

Procedimiento en una red neuronal:

- Se inicializan aleatoriamente los pesos
- Se hace un **forward propagation** con los datos
- Se calcula una función de coste
- Se hace un **backpropagation** que actualiza y mejora los pesos
- Se repite todo hasta que el valor de la función de coste es mínimo
- Se **evalúa la calidad** de los resultados con **otro set de datos**

¿Por qué no se hace en estadística?

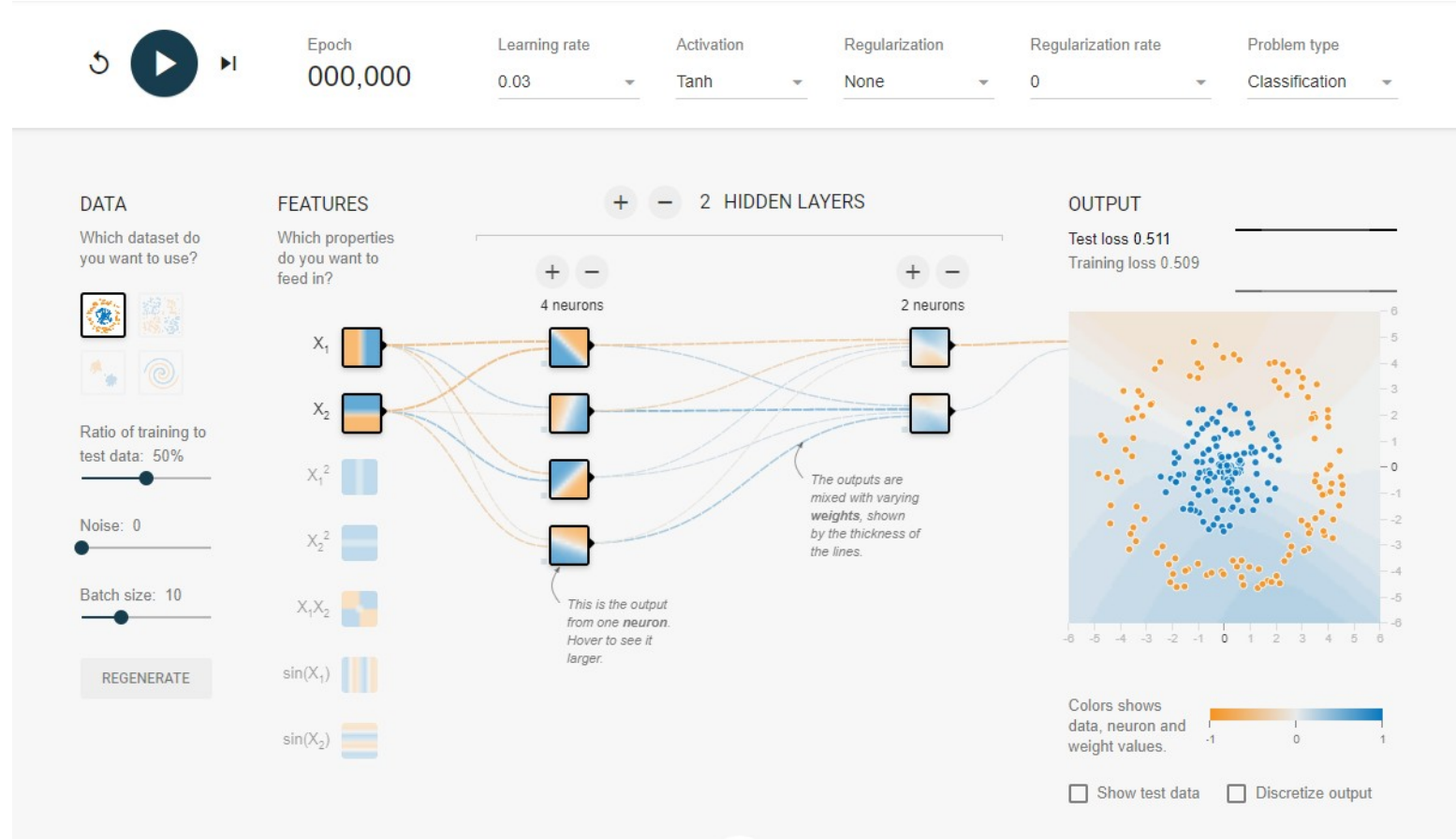


Qué necesitamos para hacer *Deep Learning*

- Un conjunto de datos
- Una red neuronal
- Una función de activación
- Una función de coste
- Un algoritmo de aprendizaje

Playground (<https://playground.tensorflow.org>)

Todo lo que hemos visto hasta ahora se puede mostrar en *playground*. Es una herramienta didáctica creada por *tensorflow* (google). Nos va a servir para entender la complejidad de las capas en *deep learning*.



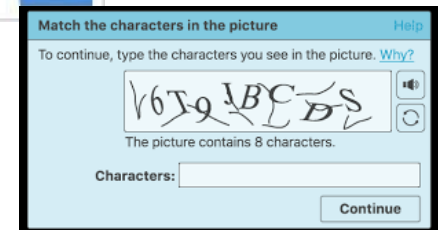
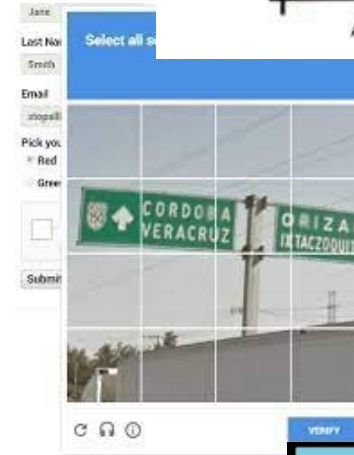
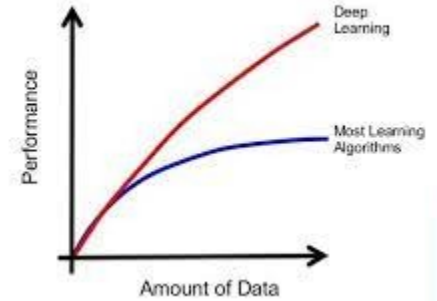
El Conjunto de datos o *dataset*:

- Se necesita muchos datos para entrenar redes
- Eso puede ser un inconveniente para aplicarse en algunas áreas (pe . Experimentos)
- Cada vez existen más y más datos porque todo se registra:
 - Cámaras
 - Móviles
 - Redes sociales

¿Para qué creéis que se usan los captcha?

- Existen infinidad de sitios donde descargar datos para entrenar nuestras redes: Kaggle.com, image-net.org

BIG DATA & DEEP LEARNING



La red neuronal:

Las redes neuronales pueden crearse de infinitas maneras combinando su *arquitectura* y sus *hiperparámetros*

- La *arquitectura* viene determinada por la estructura de la red: número de neuronas, número de capas, conexiones entre las capas, función de activación, etc.
- Los *parámetros* son los valores de los coeficientes (los W y b)
- Los *hiperparámetros* son valores que controlan el comportamiento de la red: el tipo de optimizador, la tasa de aprendizaje, el tipo de regularización, etc.

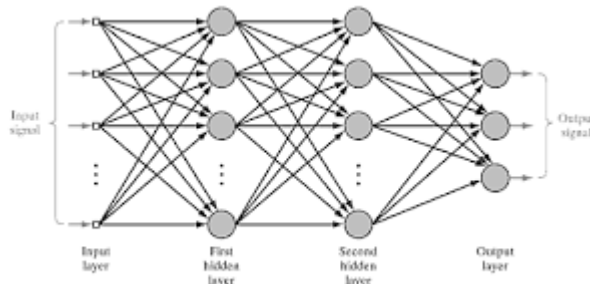
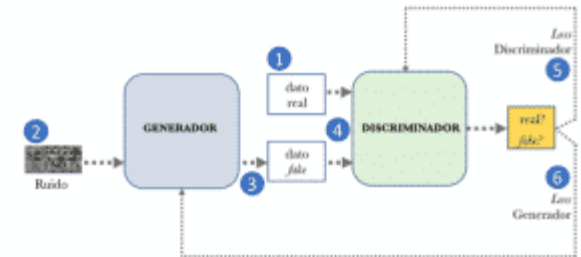
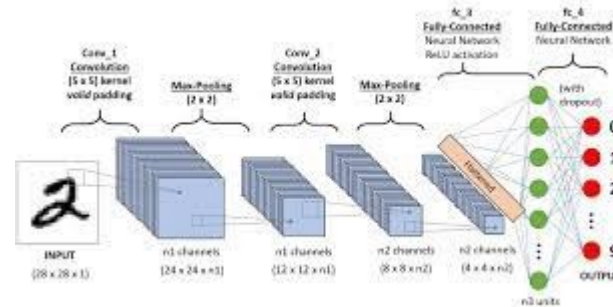


FIGURE 4.1 Architectural graph of a multilayer perceptron with two hidden layers



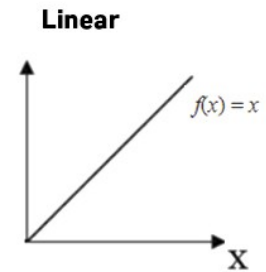
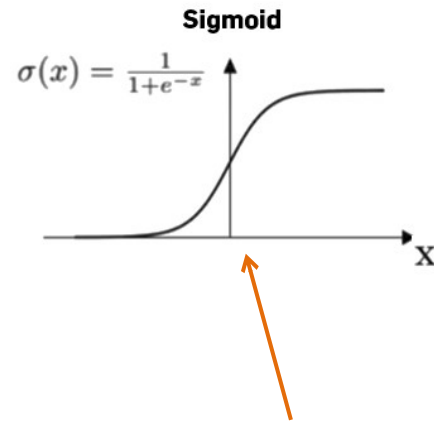
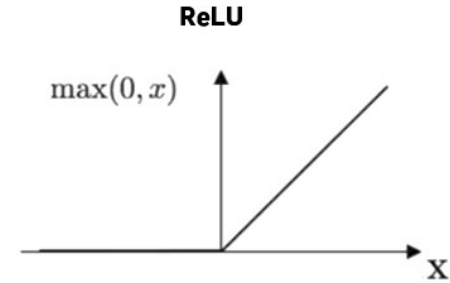
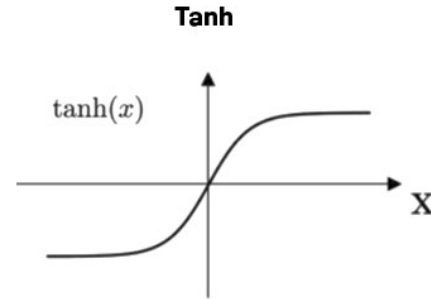
Función de activación:

La *suma ponderada* es siempre igual pero se pueden definir *diferentes* funciones de activación

Las más comunes son:

- Sigmoid
- Tanh
- Relu
- Linear
- Softmax

Lo normal es que una red *combine* varios tipos dependiendo de la capa
Todas deben ser *derivables*



[https://www.wolframalpha.com/
input/?i=sigmoid+function](https://www.wolframalpha.com/input/?i=sigmoid+function)

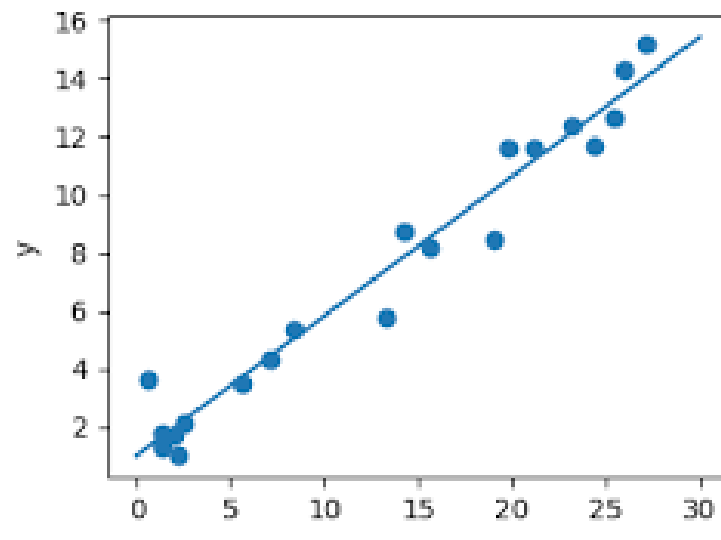
Función de coste o pérdida (*loss function*)

En las redes supervisadas es una función que calcula el error que cometemos, la diferencia entre y e y' :

En función de ese valor, se actualizarán todos los pesos utilizando algún tipo de algoritmo de aprendizaje (por ejemplo, el *descenso del gradiente*)

Dependiendo de la naturaleza de y e y' se utilizan *distintas funciones de pérdida*:

- Datos continuos: *mse*
- Categóricos binarios: *entropía binaria*
- Categóricos múltiples: *entropía categórica*



Algoritmo de aprendizaje:

Todos parten de la función de pérdida y van cambiando los valores de los pesos para minimizarla. Es lo que se llama el *backpropagation*.

El más conocido es el *descenso del gradiente*, todos usan la misma lógica o son derivados de él.

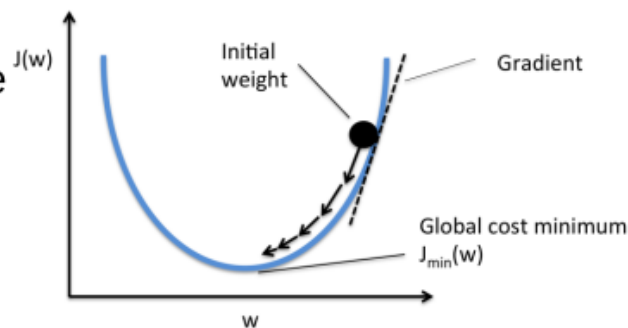
Un gradiente es una derivada, se busca minimizar los errores buscando los valores de W y b que hacen mínimo el valor de coste

Como existen múltiples capas esa derivada se va propagando hacia atrás a través de las capas

Se basa en la *regla de la cadena* (derivadas encadenadas): Si tenemos una función de una función, se puede calcular la derivada en forma de cadena:

Si f es una combinación de g : $f(g(x))$ entonces:

$$\frac{\partial f(g)}{\partial x} = \frac{\partial f(g)}{\partial g} \frac{\partial g}{\partial x}$$



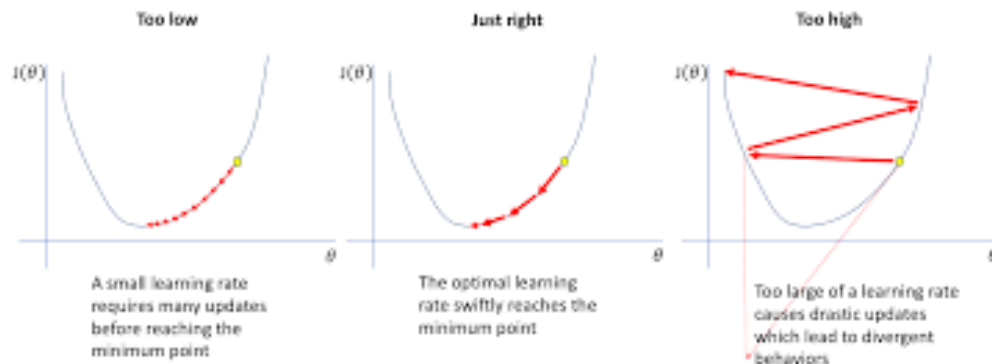
Existen muchos algoritmos de aprendizaje:

- *Descenso del gradiente*
- *Descenso del gradiente estocástico*
- *RMSprop*
- *AdaGrad*
- *Adadelta*
- *Adam*
- *Adamax*
- *Nadam*

Cada uno de ellos tienen uno o más parámetros que controlan su funcionamiento

- Alfa: Tasa de aprendizaje (*learning rate*)
- Beta: Descenso en la tasa de aprendizaje (*learning rate decay*)

El más utilizado en la actualidad es el *Adam*



Debe descender rápido pero ser capaz de evitar mínimos locales

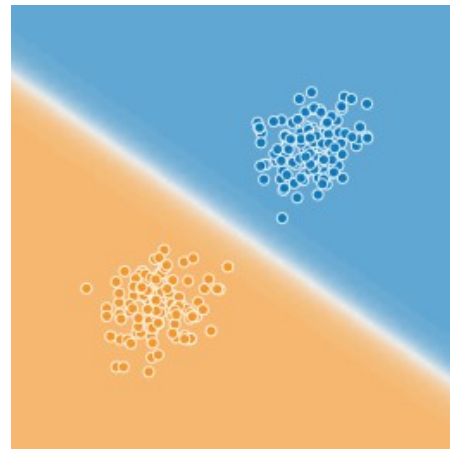
Práctica 01

Entender un
perceptron multicapa
utilizando
playground

Tarea 1: Dos pelotas

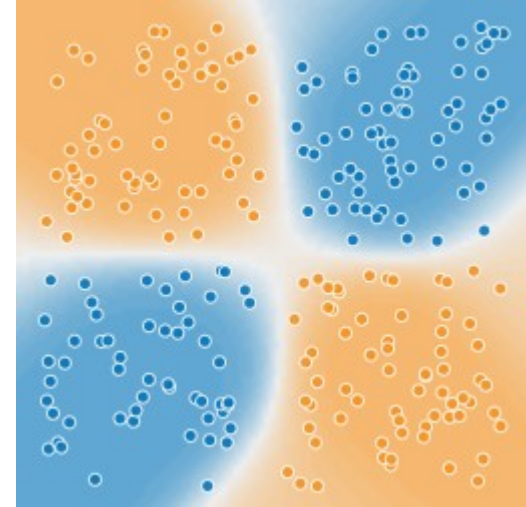
- No tocar nada de los *hiperparámetros* de aprendizaje
- Lo ponemos en clasificación, que es más visual, es un clasificador binario (equivalente a una logística)
- Como función de activación: ***sigmoid***
- Datos: **dos pelotas** (tercer conjunto de datos, inferior izquierda)
- Utilizamos solamente información de **x1 y x2**
- Buscamos la **combinación mínima** de capas y neuronas para resolver el problema
- Vemos que **1 capa y 1 neurona es suficiente**
- Escribimos la ecuación:

$$z = w_1 x_1 + w_2 x_2 + b$$
$$y' = \sigma = \frac{1}{1 + e^{-z}}$$



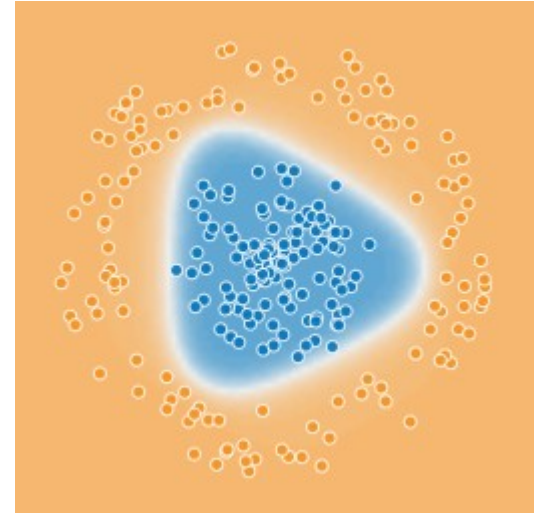
Tarea 2: Cuatro cuadrados

- Datos: segundo conjunto de datos, superior derecha
- Objetivo: obtener el *test loss* más bajo posible con el número menor de unidades (no importa el número de capas)



Tarea 3: Datos circulares

- Datos: primer conjunto de datos, superior izquierda
- Objetivo: obtener el *test loss* más bajo posible con el número menor de unidades (no importa el número de capas)
- Objetivo: Una vez que hemos encontrado la menor solución, probar a eliminar neuronas en la primera capa y meter más capas.



Tarea 4: Datos en espiral

- Datos: primer conjunto de datos, superior izquierda
- Objetivo: obtener el *test loss* más bajo posible con el número menor de unidades (no importa el número de capas)
- No se puede conseguir porque necesitas más información: Introducid *sin(x1)*, *sin(x2)* y *tanh* como función de activación
- Explicaciones:
 - <http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/>
 - <https://ai.stackexchange.com/questions/1987/how-to-classify-data-which-is-spiral-in-shape>

