¿Que son las redes neuronales? Una introducción informal

Julio Waissman Vilanova

Departamento de Matemáticas
Universidad de Sonora
para **Botón Rojo**

Plan de la presentación

¿Para que estudiar las redes neuronales?

Para entender como funciona el cerebro .

El cerebro es una cosa bastante complicada, con partes, que si uno las empieza a manipular, el propietario muere generalmente. Por eso los métdos de simulación computacional son muy importantes.

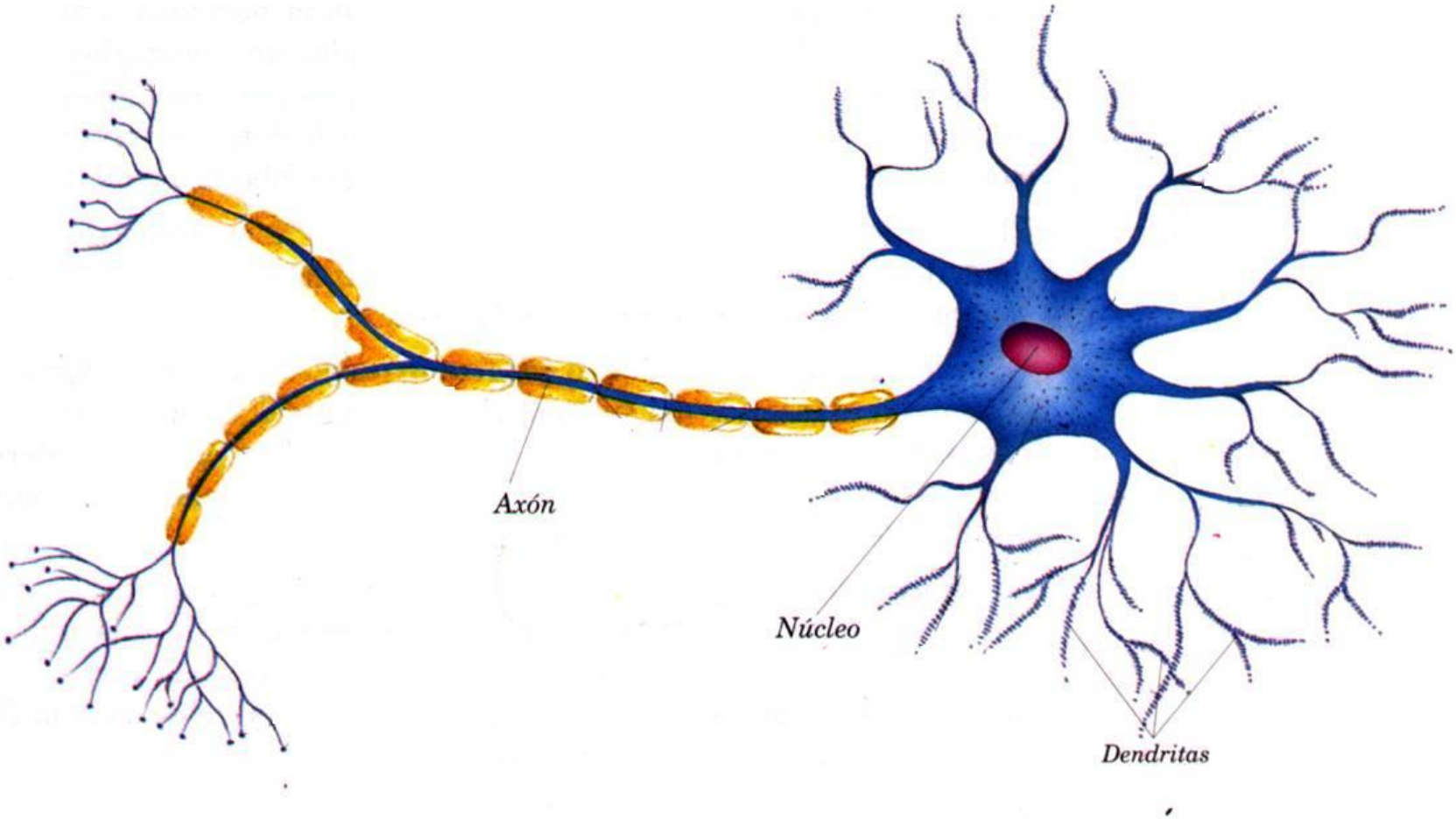
Para entender un tipo de computo paralelo .

Computo paralelo basado en la aplicación masiva de operaciones simples. .

Para resolver problemas prácticos usando aprendizaje .

A pesar que las redes neuronales no son un modelo preciso del cerebro, se pueden utilizar para resolver una gran cantidad de problemas prácticos.

Estructura de una neurona



Sinapsis

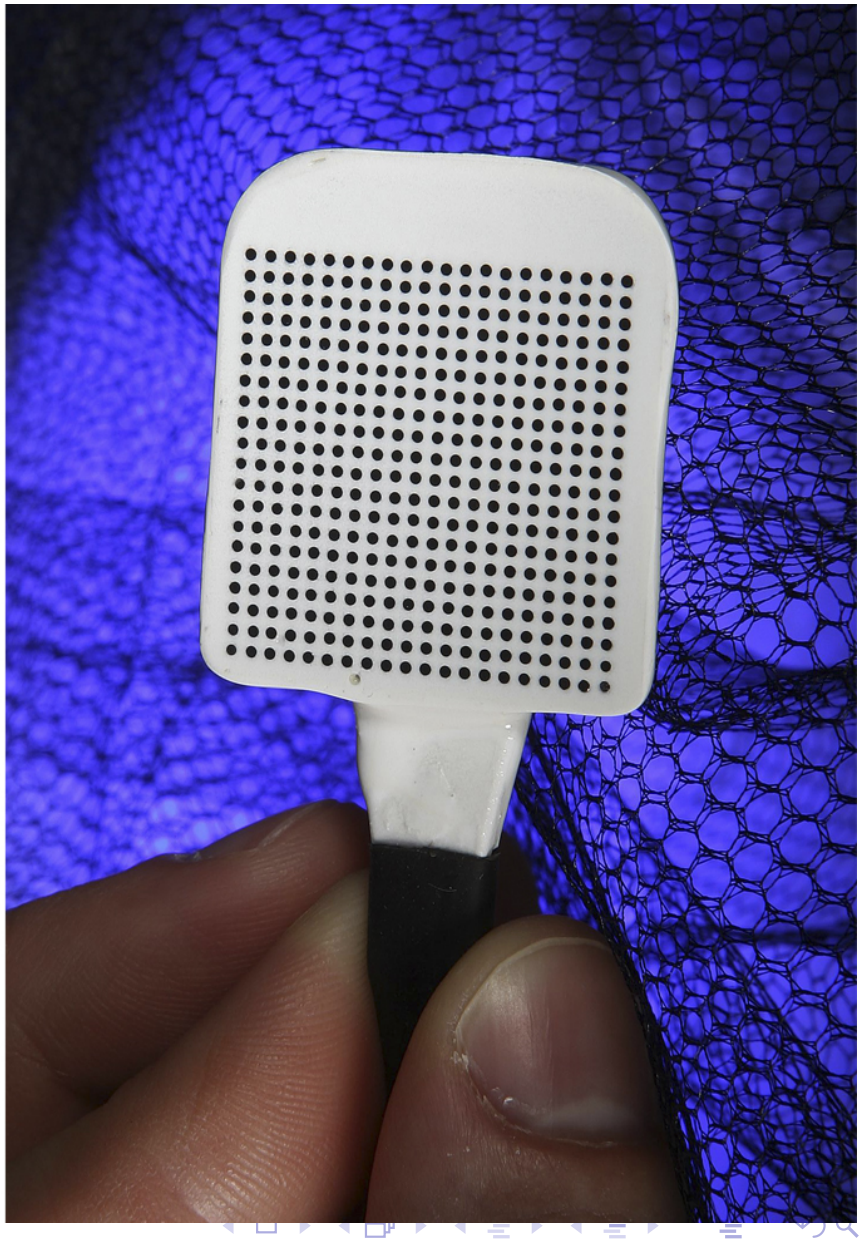
- Cuando un impulso electrico viaja a través de un axón a una sinapsis, se transmite a través de un líquiso neurotransmisor.
- El impulso se transmite por difusión a las neuronas receptoras.
- La efectividad de las sinapsis puede ser cambiada (peso sináptico).
- La sinapisi es lenta, pero usa muy poca energía, y se adapta a partir de señales disponibles localmente.

¿Como funciona el cerebro?

- Solo unas pocas neuronas se conectan con receptores o actuadores.
- Cada neurona recibe estimulos de otras neuronas.
- El efecto de cada entrada de otra neurona es controlado por un peso sináptico.
- El peso sináptico se adapta de forma que la red en forma global aprende a realizar operaciones útiles.
- Tenemos aproximadamente 10^{11} neuronas, cada una con 10^4 pesos sinápticos.

Modularidad del cerebro

Modularidad del cerebro



Modelos simples de una neurona

- Para modelar cosas hay que idealizarlas.
- Eliminar detalles complicados no esenciales para la comprensión del fenómeno.
- Vale la pena realizar modelos comprensibles de cosas que sabemos son incorrectas (sin olvidar que son incorrectas).
- Asumamos que las neuronas transmiten valores reales y no impulsos.

Neuronas lineales

$$y = b + \sum_i x_i w_i$$

- Si hacemos aprender a esta neurona, podemos probar con neuronas más complicadas.
- y es la salida, b es el sesgo, x_i la i–ésima entrada y w_i el peso en la i–ésima entrada.

Neuronas binarias

$$z = \sum_i x_i w_i$$

$$y = \begin{cases} 1 & \text{si } z \geq \theta \\ 0 & \text{en otro caso} \end{cases}$$

- Propuestas por McCulloch y Pitts en 1943.
- Se asume que cada impulso es como el valor de verdad de una proposición lógica.
- El parámetro θ es un umbral.

Neuronas binarias

Representación alternativa

$$z = b + \sum_i x_i w_i$$

$$y = \begin{cases} 1 & \text{si } z \geq 0 \\ 0 & \text{en otro caso} \end{cases}$$

- La neurona binaria es equivalente a aplicar una función no lineal a la neurona binaria.
- A esta función no lineal se le conoce como función de activación

Neuronas lineales rectificadas

ReLU

$$z = b + \sum_i x_i w_i$$

$$y = \begin{cases} z & \text{si } z \geq 0 \\ 0 & \text{en otro caso} \end{cases}$$

- Relativamente reciente, muy rápido de calcular.
- Muy eficiente en problemas de aprendizaje profundo.

Neuronas sigmoidales

Logística

$$z = b + \sum_i x_i w_i$$

$$y = \frac{1}{1 + e^{-z}}$$

- Devuelve un valor real acotado, imitando una neurona binaria.
- Fácil de derivar (lo que es útil para el aprendizaje).
- Las más usadas en modelos de pequeña y mediana escala.

Neuronas estocásticas

Estocástica

$$z = b + \sum_i x_i w_i$$

$$P[s = 1] = \frac{1}{1 + e^{-z}}$$

- Mismas ecuaciones que la logística, pero diferente significado.
- Probabilidad de tener un impulso en un intervalo de tiempo corto.
- La salida es una distribución de Poisson.

Arquitecturas principales de redes neuronales

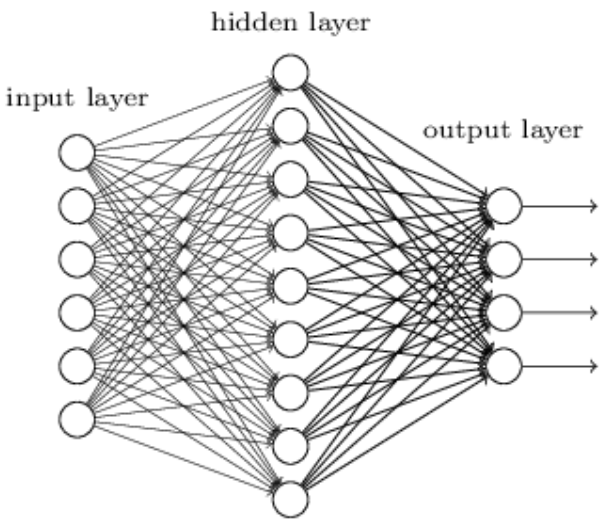
- Por arquitectura, consideramos la forma en que se conectan las neuronas entre si. Tambien se le conoce como topología de la red neuronal.
- Dos arquitecturas generales básicas: *alimentación hacia adelante*, y *recursivas*.
- Para este curso nos centraremos en las de alimentación hacia adelante.

Redes con alimentación hacia adelante

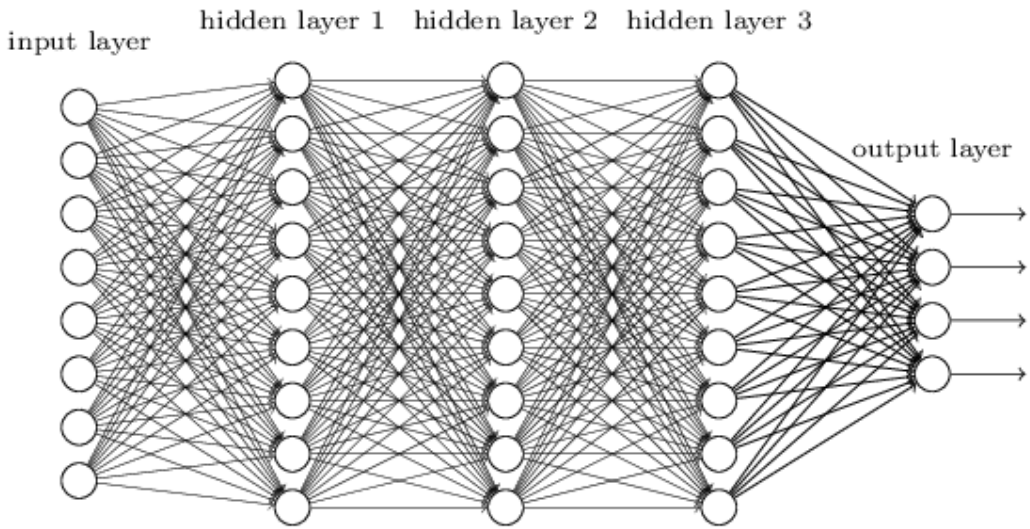
- Redes más comunes en aplicaciones prácticas
- La primer capa es la entrada y la última la salida
- Si tienen más de una capa oculta se les conoce como redes neuronales profundas (*Deep neural network*).
- Calculan una serie de transformaciones de los datos de entrada.
- La función de activación de cada neurona es no lineal.

Redes con alimentación hacia adelante

"Non-deep" feedforward neural network



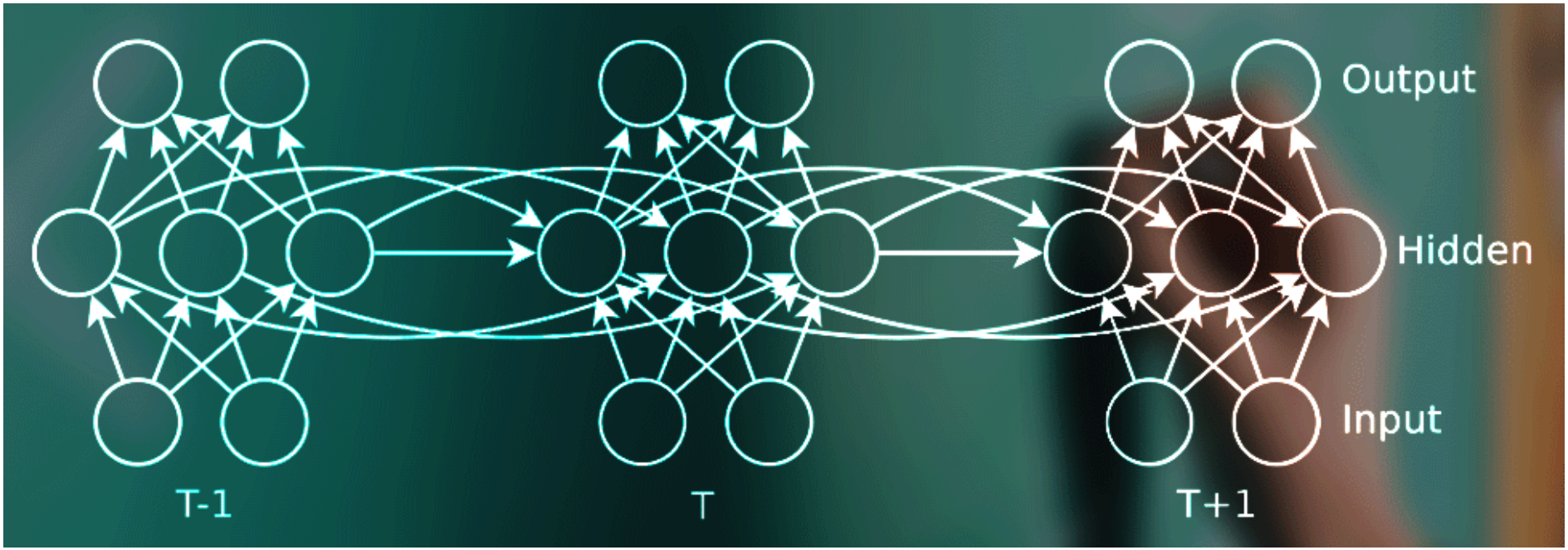
Deep neural network



Redes recurrentes

- Existen ciclos en sus conexiones
- Solo se pueden analizar en función del tiempo
- Dinámicas complicadas y dificultad en el aprendizaje
- Biológicamente son un poco más realistas

Redes recurrentes



Aplicación de la redes recurrentes

- Método natural para modelar información en forma secuencial
- Habilidad para recordar información

Ejemplo (Ilya Sutskever, 2011)

In 1974 Northern Denver had been overshadowed by CNL, and several Irish intelligence agencies in the Mediterranean region. However, on the Victoria, Kings Hebrew stated that Charles decided to escape during an alliance. The mansion house was completed in 1882, the second in its bridge are omitted, while closing is the proton reticulum composed below it aims, such that it is the blurring of appearing on any well-paid type of box printer.

Redes recurrentes simétricas

- Redes recurrentes, donde todas las conexiones son en ambas direcciones y con el mismo peso en ambas direcciones.
- Un caso *simple* de analizar de redes recurrentes.
- Si no hay unidades ocultas se conocen como redes de Hopfield.
- Si tiene unidades ocultas se conocen como máquinas de Boltzmann.

El preceptrón

Y ahora regresemos a la más simple de las redes neuronales hacia adelante: El perceptrón

- Red neuronal hacia adelante, con solo dos capas (no hay capa oculta)
- Las neuronas se modelan como *neuronas binarias*

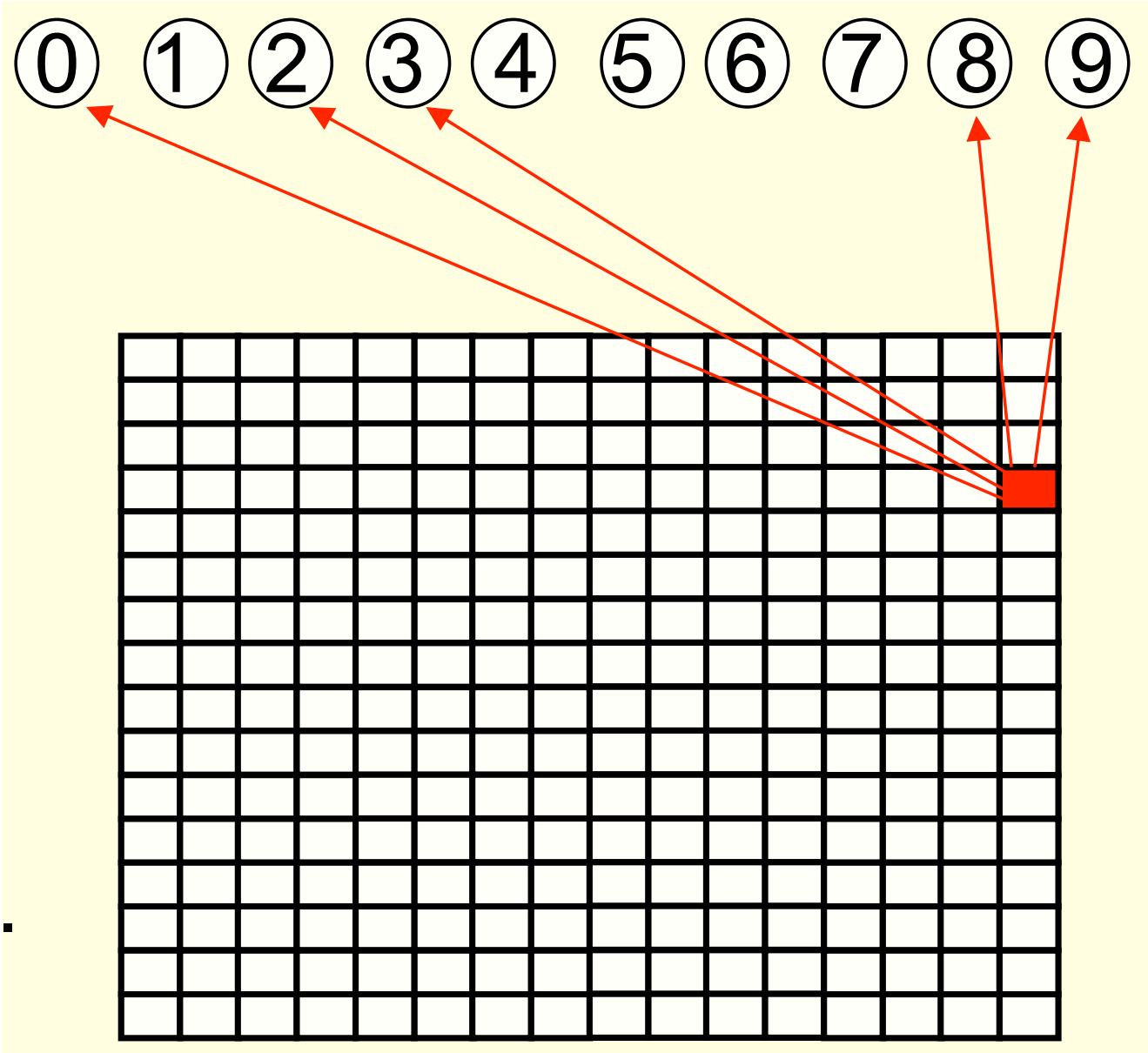
El Perceptrón

- Popularizado por Rosenblatt a principios de los 60.
- Parecían tener gran poder de aprendizaje en la época.
- En 1969, Minsky y Papert hacen pedazos al perceptrón.
- Actualmente es usado en problemas con varios millones de entradas.

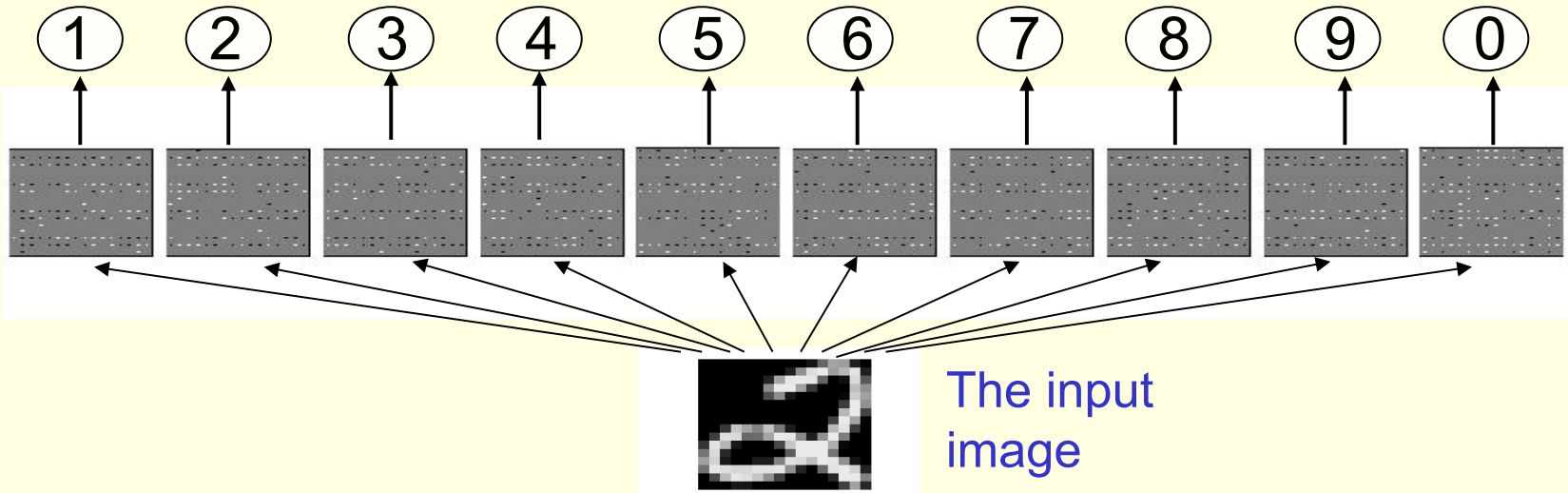
Ejemplo simple de un perceptrón

- Red simple para reconocer formas escritas a mano.
- Dos capas de neuronas:
  - Neuronas en la primer capa representan intensidad de pixeles.
  - Neuronas en la segunda capa representan formas conocidas.
- Un pixel vota por una forma solo si tiene tinta.
- Cada pixel puede vota por cada forma.
- La forma más votada es seleccionada.

Ilustración del problema

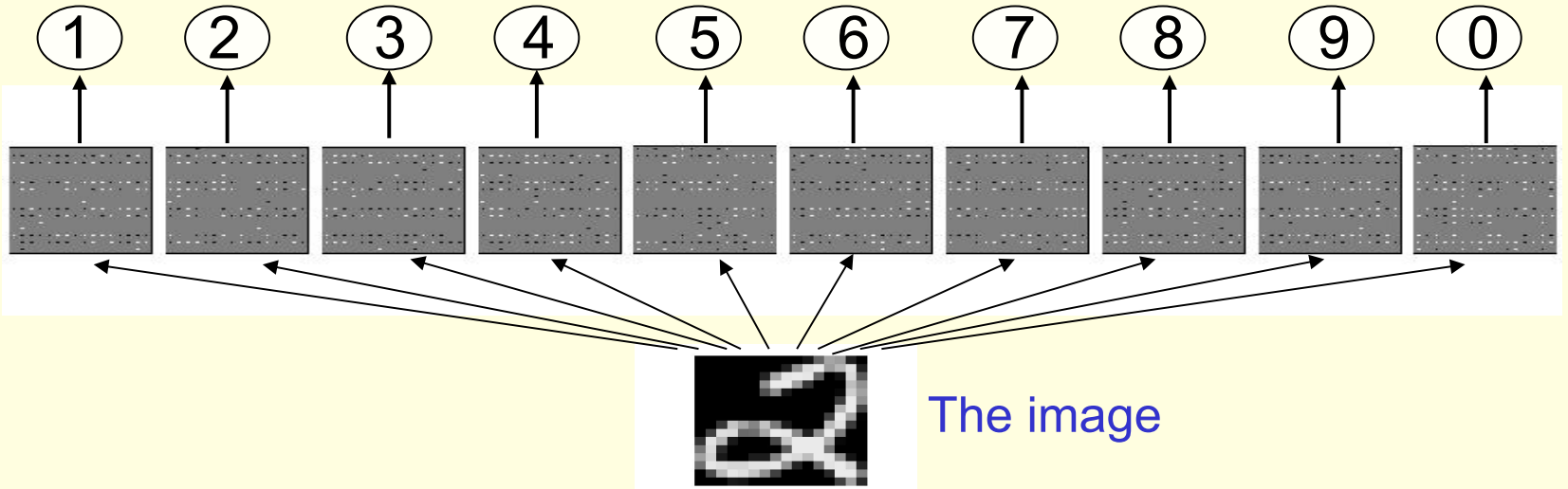


Representación de pesos



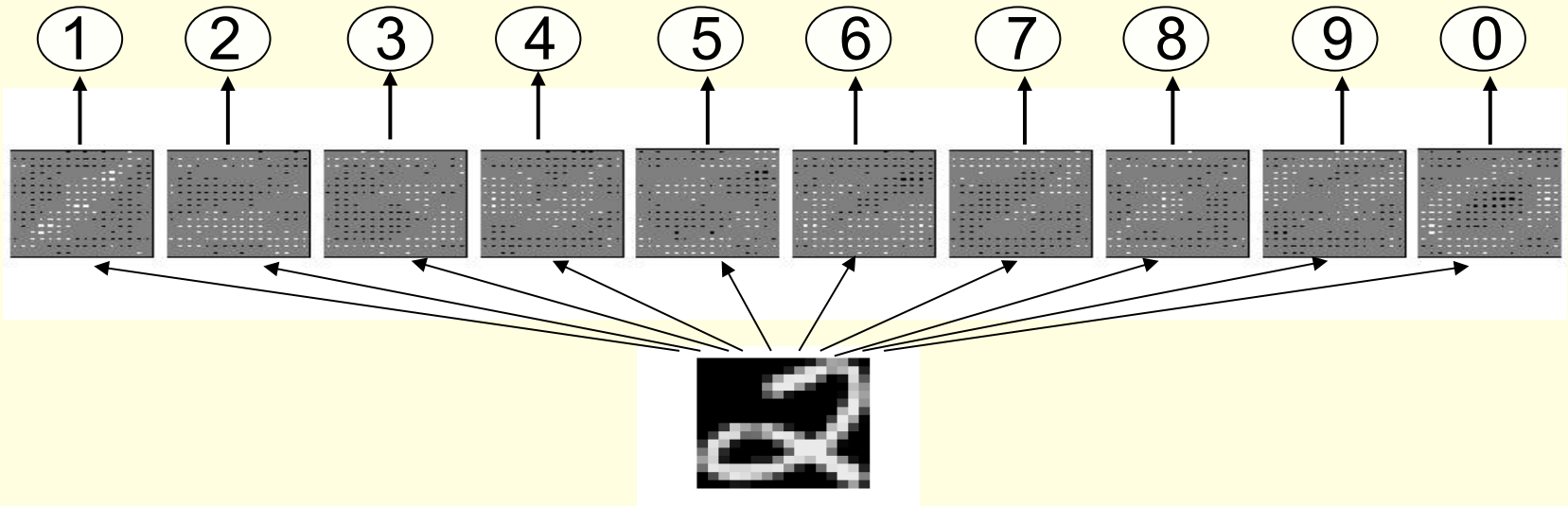
- El color (blanco o negro) representa el signo del voto.
- El área representa la intensidad (peso) del voto.

Aprendizaje



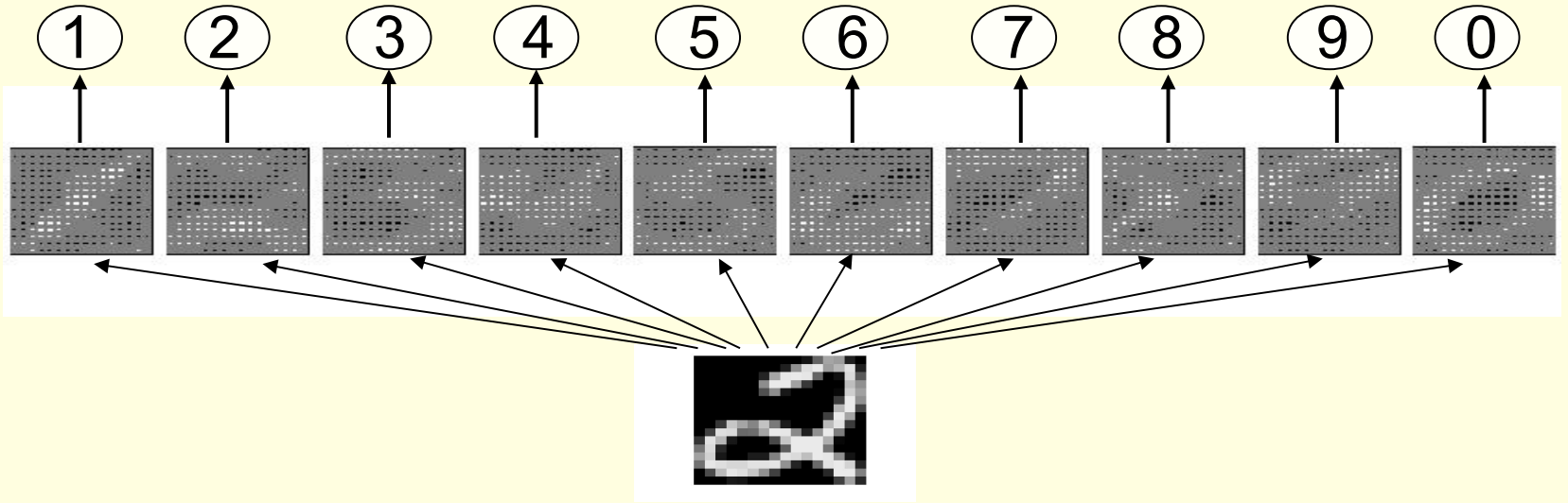
- Muestra un ejemplo conocido a la red, e incrementa el peso de los pixeles activos para la forma correcta.
- Decrementa el peso de los pixeles activos para la forma seleccionada por la red neuronal.

Aprendizaje



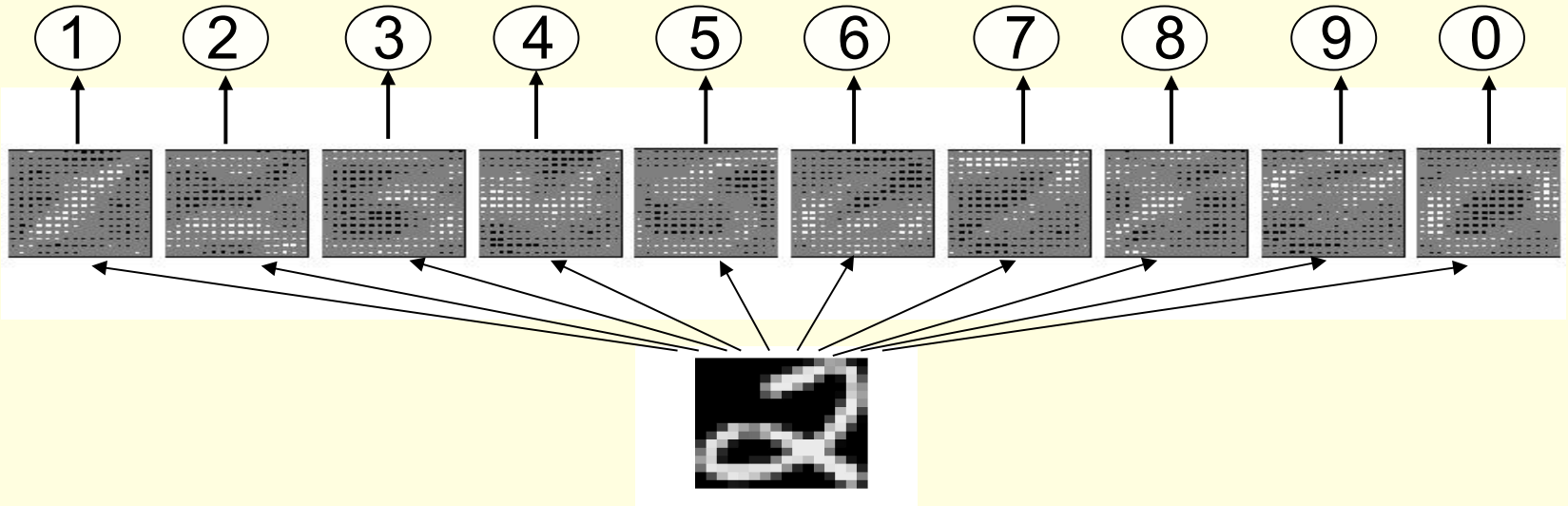
The image

Aprendizaje



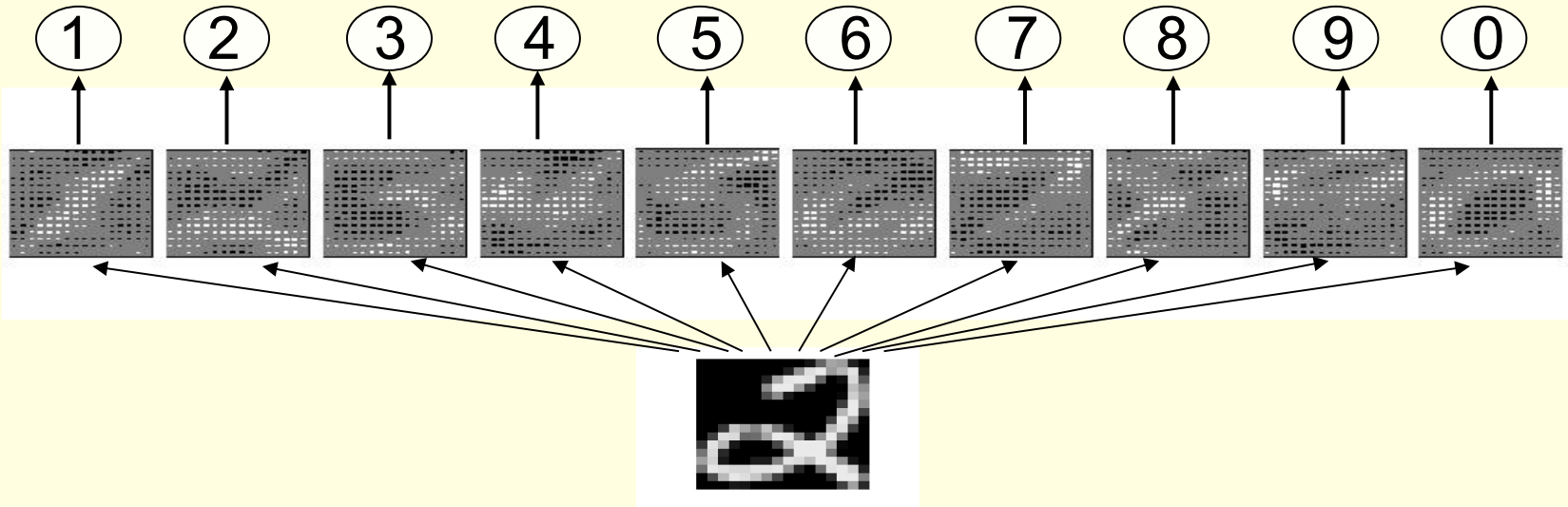
The image

Aprendizaje



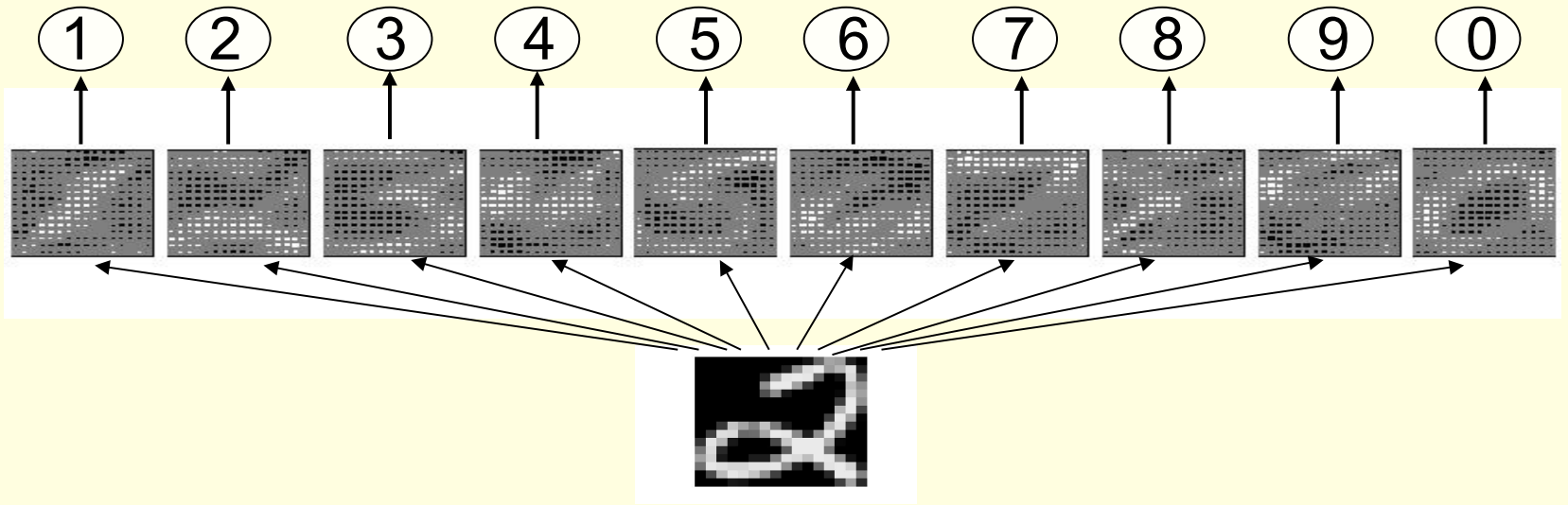
The image

Aprendizaje



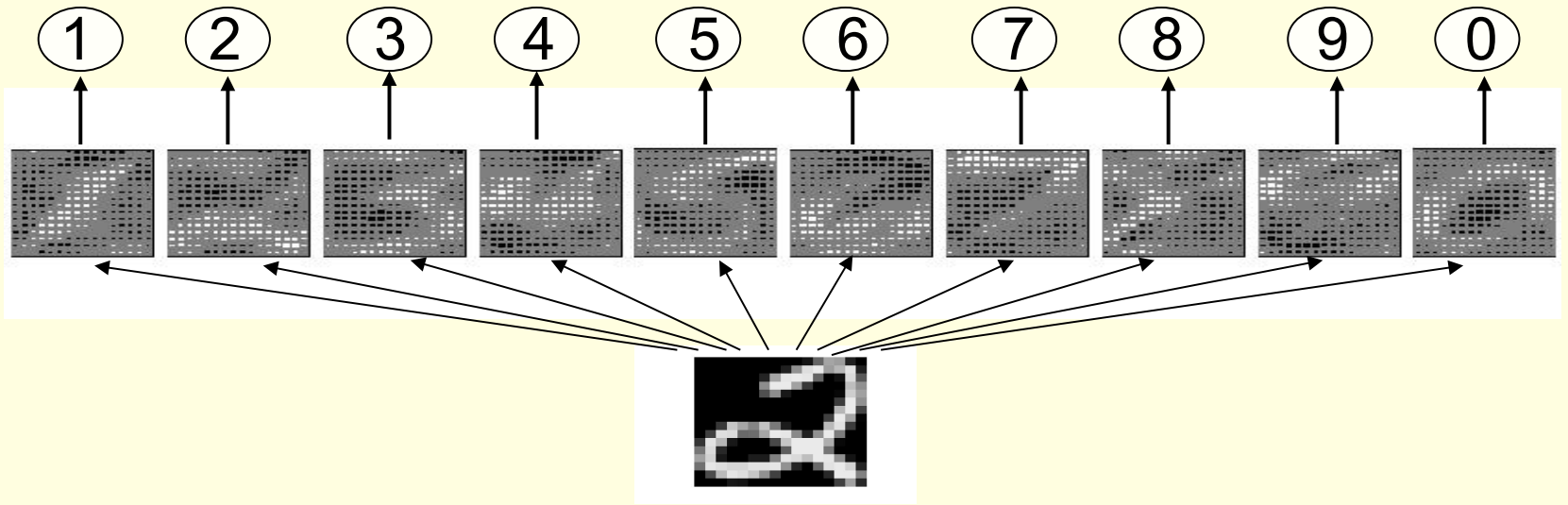
The image

Aprendizaje



The image

Aprendizaje

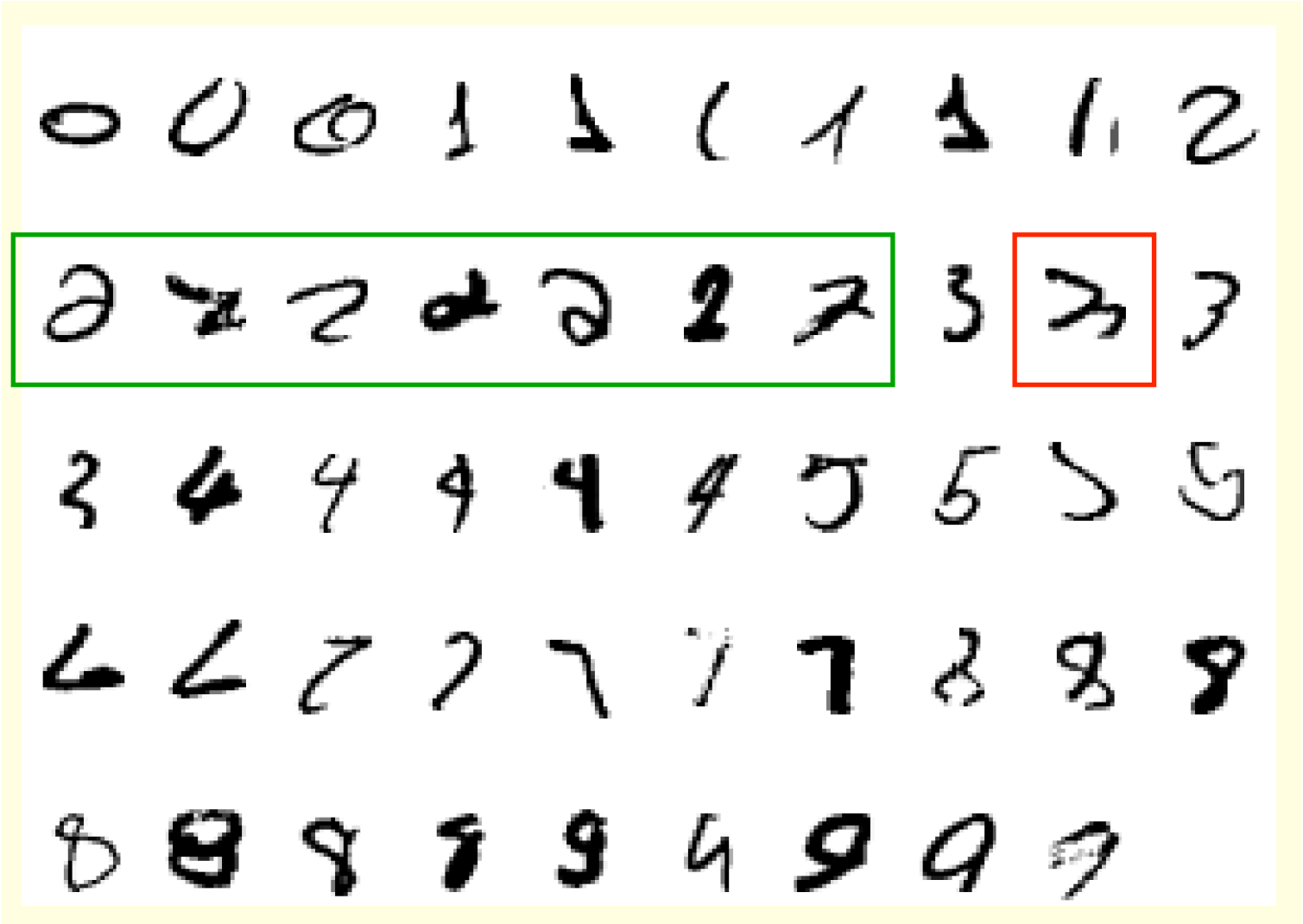


The image

Problemas con el aprendizaje

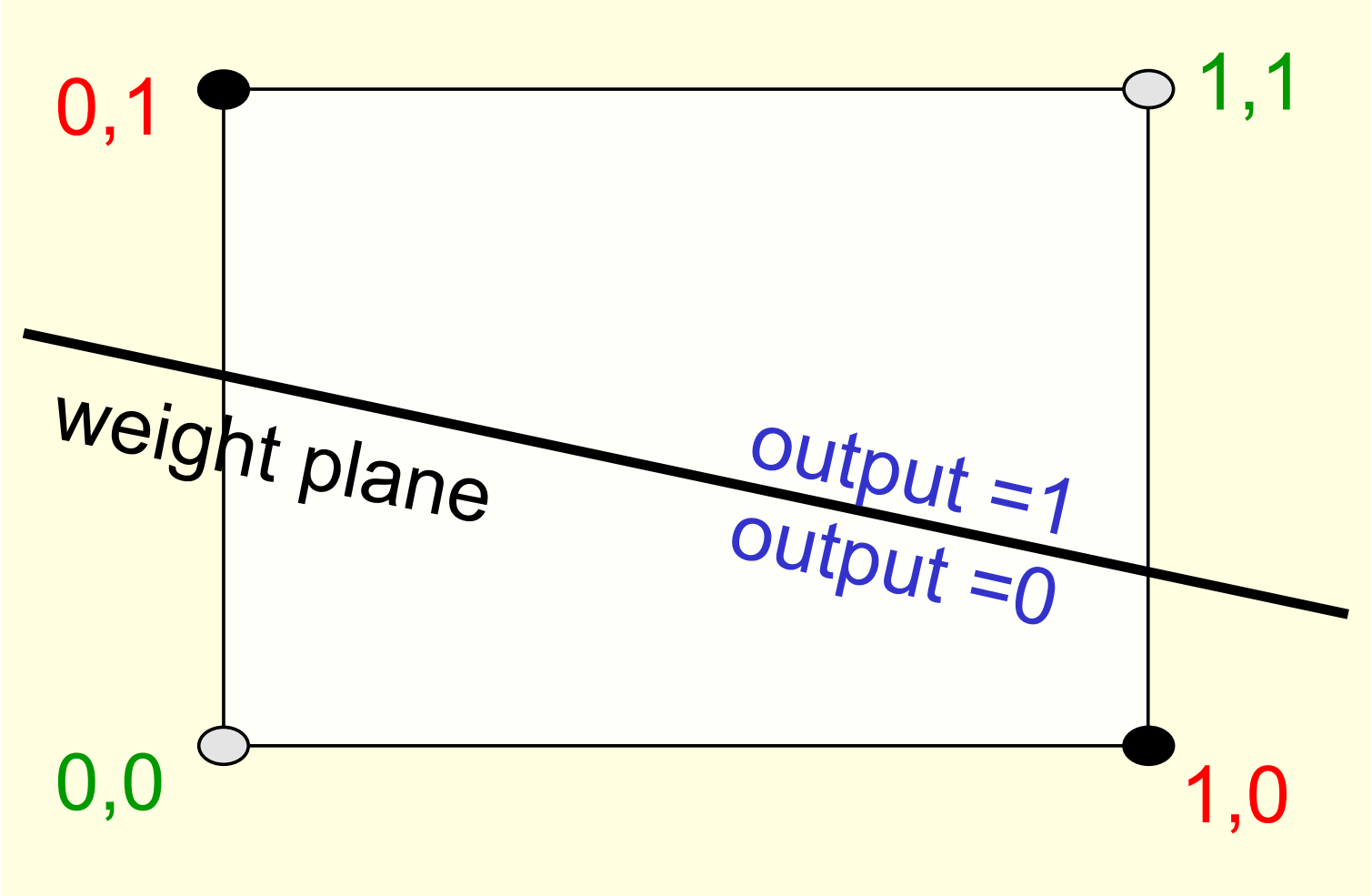
- Una red con solo dos capas es equivalente a un patron rígido por cada forma.
- El ganador es el patron con más traslape.
- Los dígitos escritos a mano son mucho más complejos.
- Es necesario capturar otras características.

Ejemplo de dígitos escritos a mano



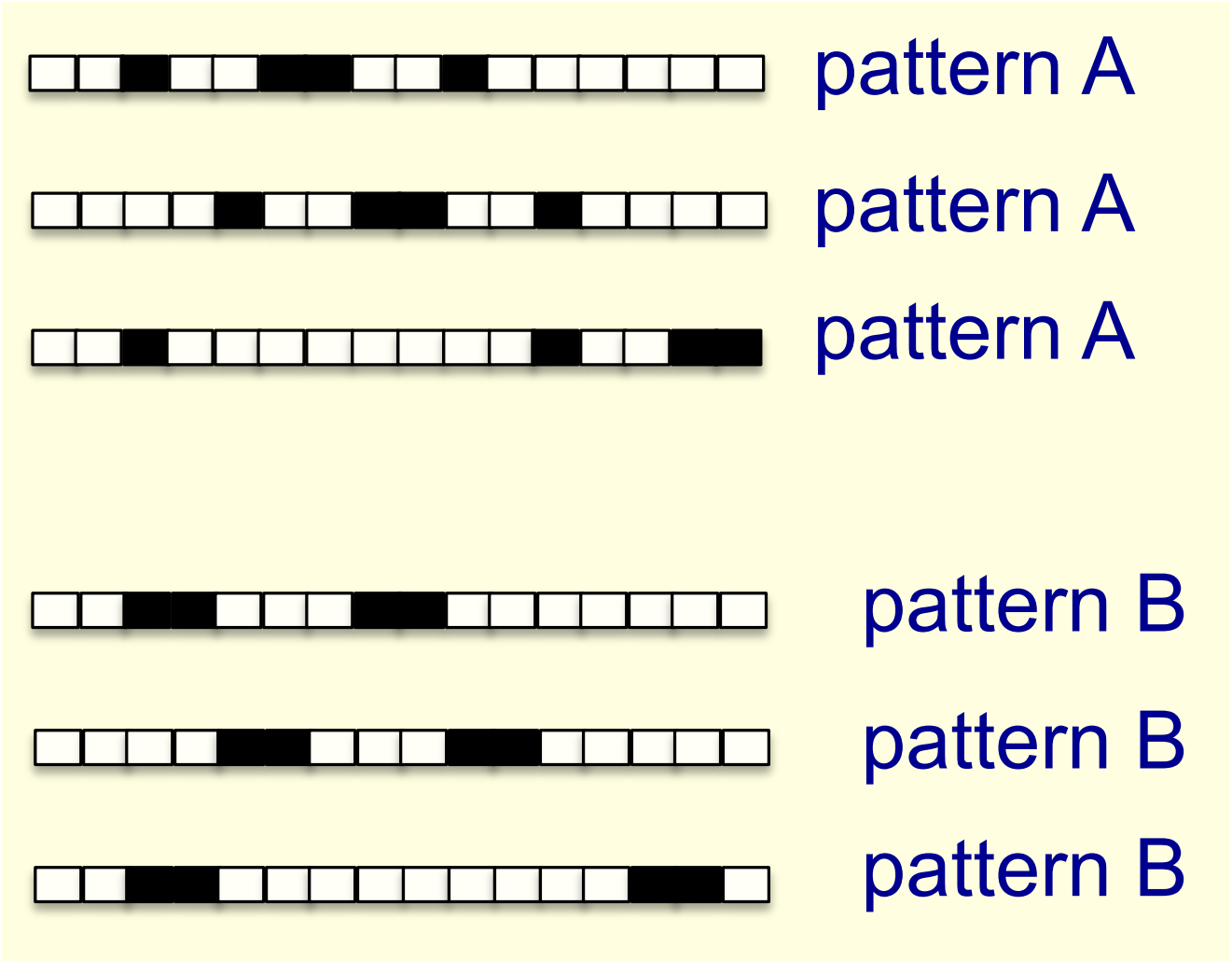
Limitaciones del perceptrón

El perceptrón no puede resolver ni siquiera los problemas más elementales no linealmente separables.



Limitaciones del perceptrón

El perceptrón no puede resolver patrones simples con translación.



Una catastrofe para las redes neuronales

- El objetivo del reconocimiento de patrones es reconocer patrones, a pesar de transformaciones como la translación.
- Para resolver estos problemas, es necesario extraer otro tipo de características para reconocer translaciones, u otro subpatron informativo.
- La parte sabrosa entonces hay que programarla a mano, y le dejamos el problema fácil al algoritmo de aprendizaje.

Aprendizaje con neuronas ocultas

- Redes sin neuronas ocultas solo pueden hacer funciones muy simples entre las entradas y las salidas.
  - Neuronas ocultas lineales no ayudan, el resultado sigue siendo una combinación lineal de las entradas.
  - No linealidades fijas a la salida tampoco son suficientes.
- Es necesario agregar capas de neuronas adaptables y no lineales.
  - Para que esto sea posible es necesario ajustar *todos* los pesos al mismo tiempo, y no solo los de la capa de salida.
  - Aprender pesos en las capas ocultas es equivalente a aprender nuevas características.
  - Problema difícil, ya que nadia da indicaciones de como deben ser esas nuevas características.

¿Que sigue?

Ahora falta ver

- Como aprender en forma genérica (sin capas ocultas)
- Como generalizar esto a una o varias capas ocultas.

Pero primero tomemos un descanso instalando y revisando las características básicas de TensorFlow.