Aprendizaje en máquina aplicado a la detección de fallas mecánicas

Departamento de Ingeniería Eléctrica

Sección de Electrónica del Estado Sólido

Examen de Grado de Maestro en Ciencias

Ing. Luis Elias Salgado Solano

Asesores: Dr. Felipe Gómez Castañeda
Dr. José Antonio Moreno Cadenas

5 de Diciembre del 2022

Contenido

- 1.- Justificación
- 2.- Objetivos
- 3.- Base de datos
- 4.- Metodología propuesta
- 5.- Resultados
- 6.- Conclusiones
- 7.- Perspectivas

Justificación

- En el desempeño de maquinaria rotatoria, una sola falla en un rodamiento puede causar tiempo muerto y pérdida de producción.
- Los procedimientos para extraer información con tratamiento del tipo Gaussiano, son difíciles de implementar en sistemas de diagnóstico en tiempo real.
- Las técnicas de Aprendizaje en Máquina (Machine Learning) han demostrado ser eficientes para la detección de fallas en máquinas rotatorias y son factibles de implementar en hardware específico (Neuromórfico).

Industrias de aplicación con sistemas de máquinas rotatorias

- Transporte urbano, construcción, manufactura, automotriz, agricultura, petroquímica, entre otras.

Contenido

- 1.- Justificación
- 2.- Objetivos
- 3.- Base de datos
- 4.- Metodología propuesta
- 5.- Resultados
- 6.- Conclusiones
- 7.- Perspectivas

Objetivos:

- Experimentar el desempeño de algoritmos de optimización y arquitecturas especificas para llevar a cabo el trabajo de diagnóstico de fallas en **tiempo real**, basado en métodos de Aprendizaje en Máquina.
- Probar experimentalmente que los porcentajes de exactitud (Accuracy) para el sistema de Aprendizaje en Máquina propuesto, son satisfactorios.
- Hacer una propuesta inicial de implementación en hardware de este sistema para su aplicación en IoT y Big Data.

Contenido

- 1.- Justificación
- 2.- Objetivos
- 3.- Base de datos
- 4.- Metodología propuesta
- 5.- Resultados
- 6.- Conclusiones
- 7.- Perspectivas

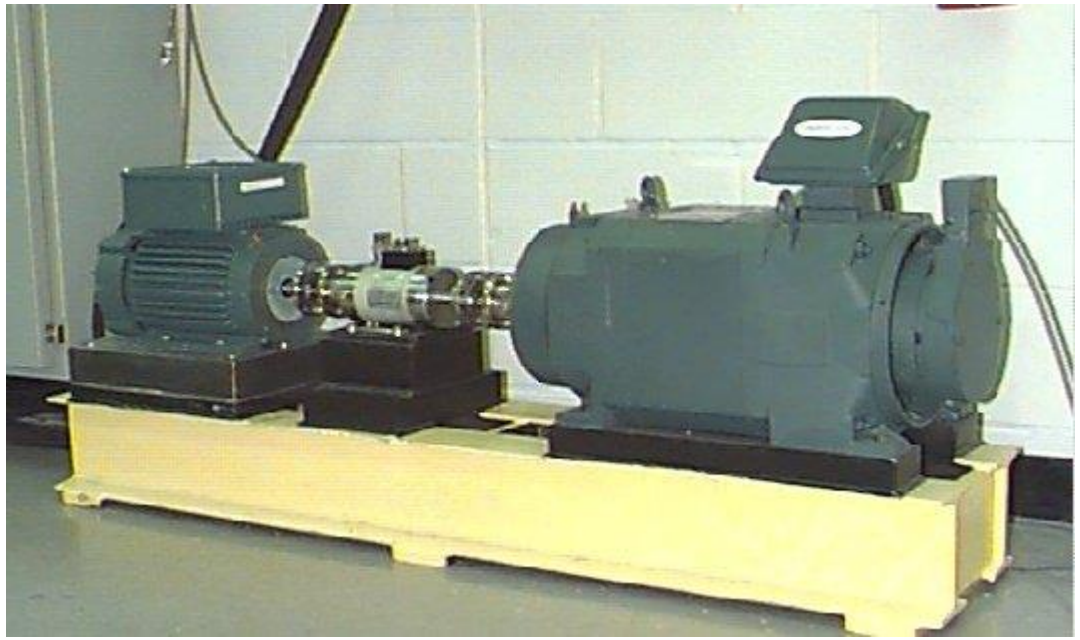
Base de datos de sistemas rotatorios

Descargada del sitio de la Universidad *Case Western Reserve*, patrocinada por el Centro Científico de *Rockwell Automation*, y la oficina de investigación naval de EEUU.

Base de datos

Banco de pruebas

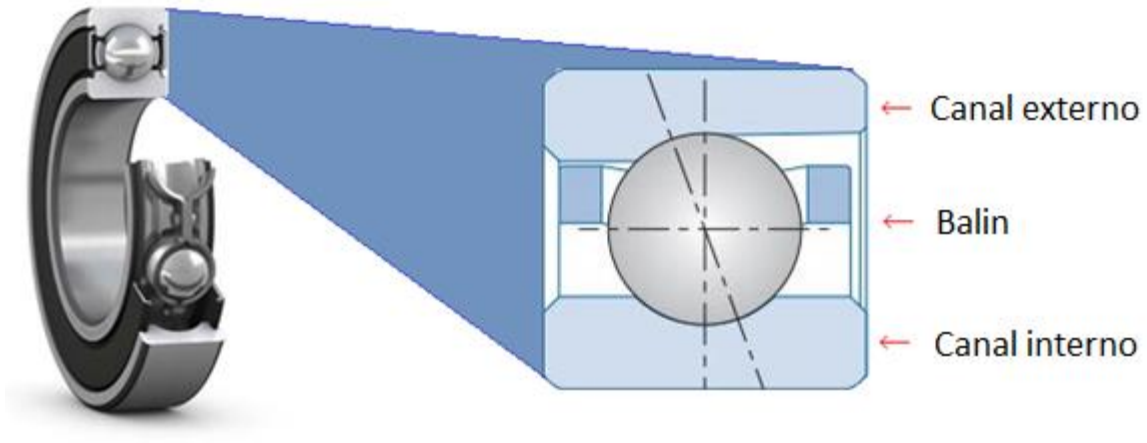
- Consiste de un motor de **2HP** (izquierda), un transductor/codificador de torque (centro), un dinamómetro(derecha); controlado mediante un mando electrónico.
- En este trabajo se tomó la base de datos correspondiente a la condición experimental con carga mecánica de **0HP**, con razón de muestreo de **12KS/s**.



Base de datos

Inducción de falla en los rodamientos.

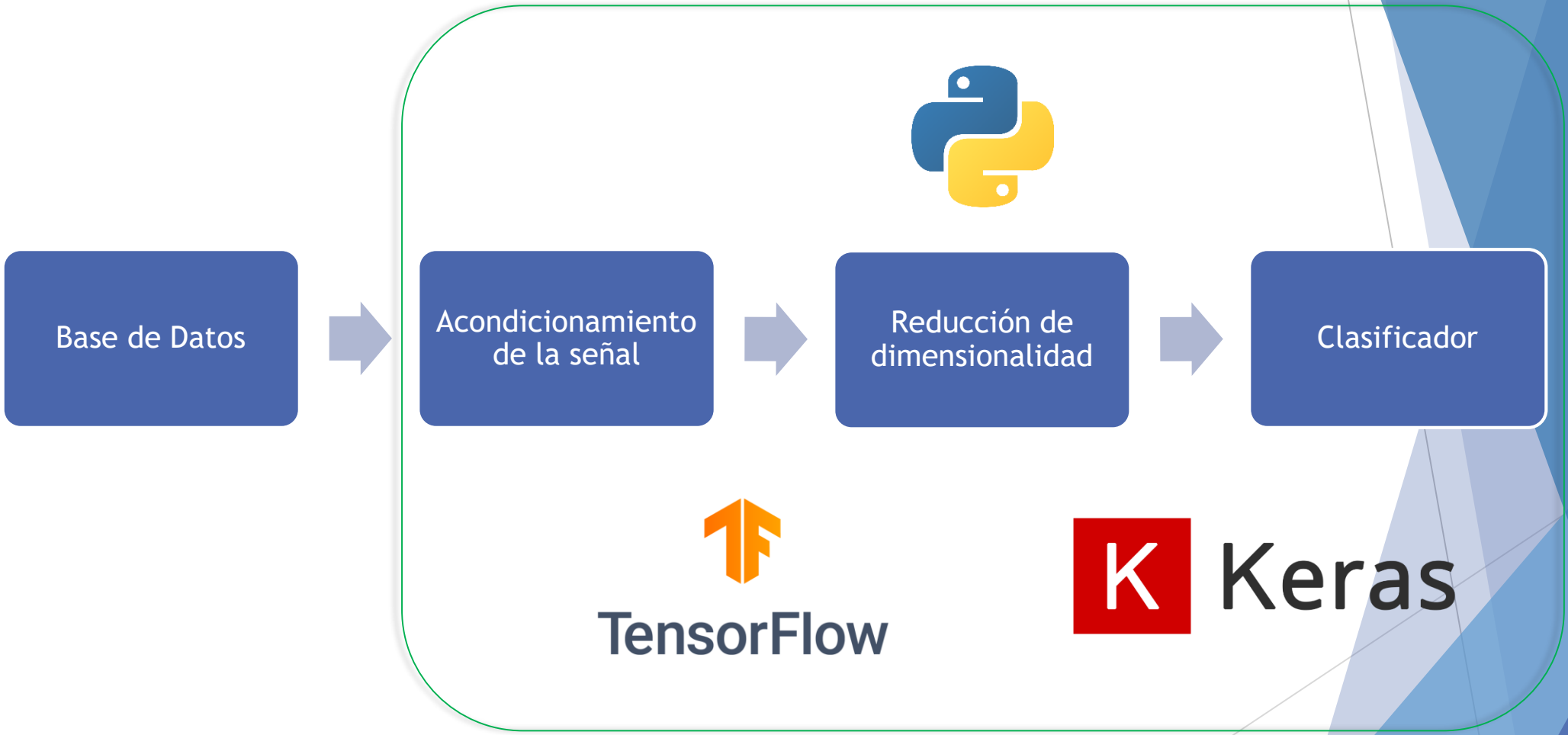
- Los rodamientos de prueba soportan el eje del motor.
- Se introdujeron fallas de un solo punto en los rodamientos de prueba, mediante descarga eléctrica mecanizada, también conocida como electro-erosión, con diámetros de 7mil(0.1778mm), 14mil(0.3556mm) y 21mil(0.5334mm).
- Las fallas fueron inducidas en 3 lugares: En el anillo interno, externo y el balín.
- El rodamiento usado en el experimento es un rodamiento rígido de balines, de la marca SKF y modelo 6205-2RS JEM.



Contenido

- 1.- Justificación
- 2.- Objetivos
- 3.- Base de datos
- 4.- Metodología propuesta
- 5.- Resultados
- 6.- Conclusiones
- 7.- Perspectivas

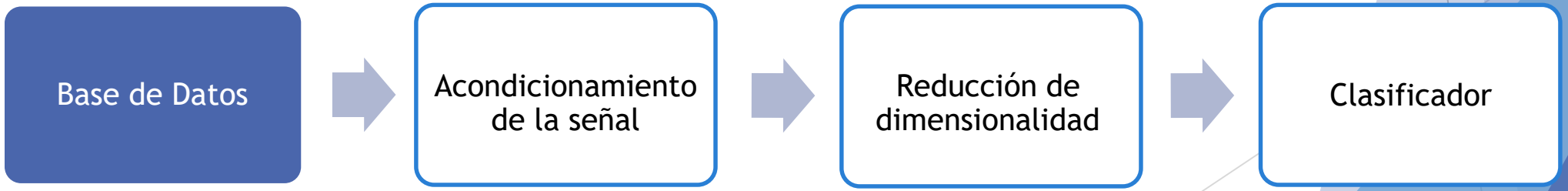
Metodología propuesta



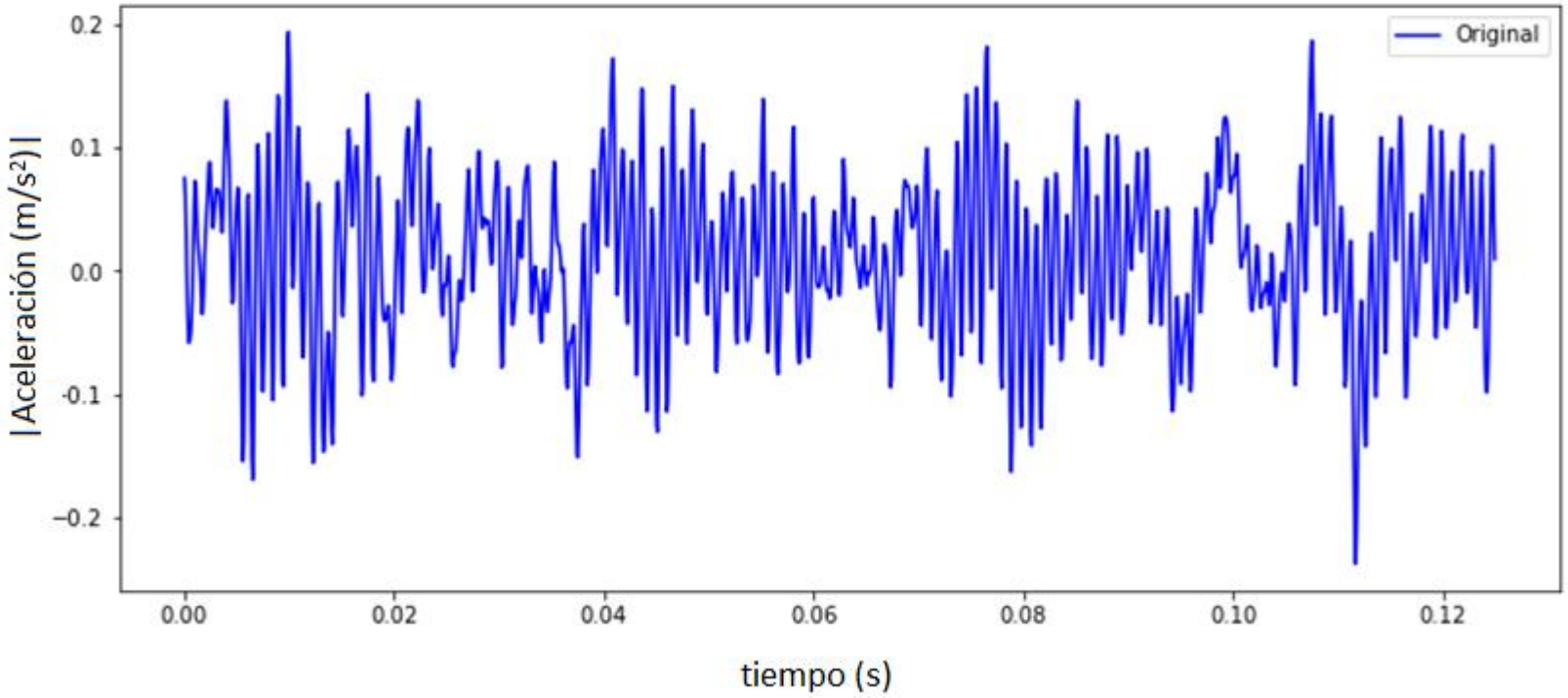
Base de datos

- La base de datos se compone de vectores de una sola fila, con mas de 100 x 10^3 muestras cada uno. Para poder llevar a cabo el entrenamiento fue necesario descomponer el vector en un arreglo de vectores de 100 muestras cada uno.

Archivo (0.007”)	Número de muestras	Número de vectores
Normal	243938	2439
Falla en canal interno	121265	1212
Falla en canal externo	121991	1219
Falla en el balín	122571	1225



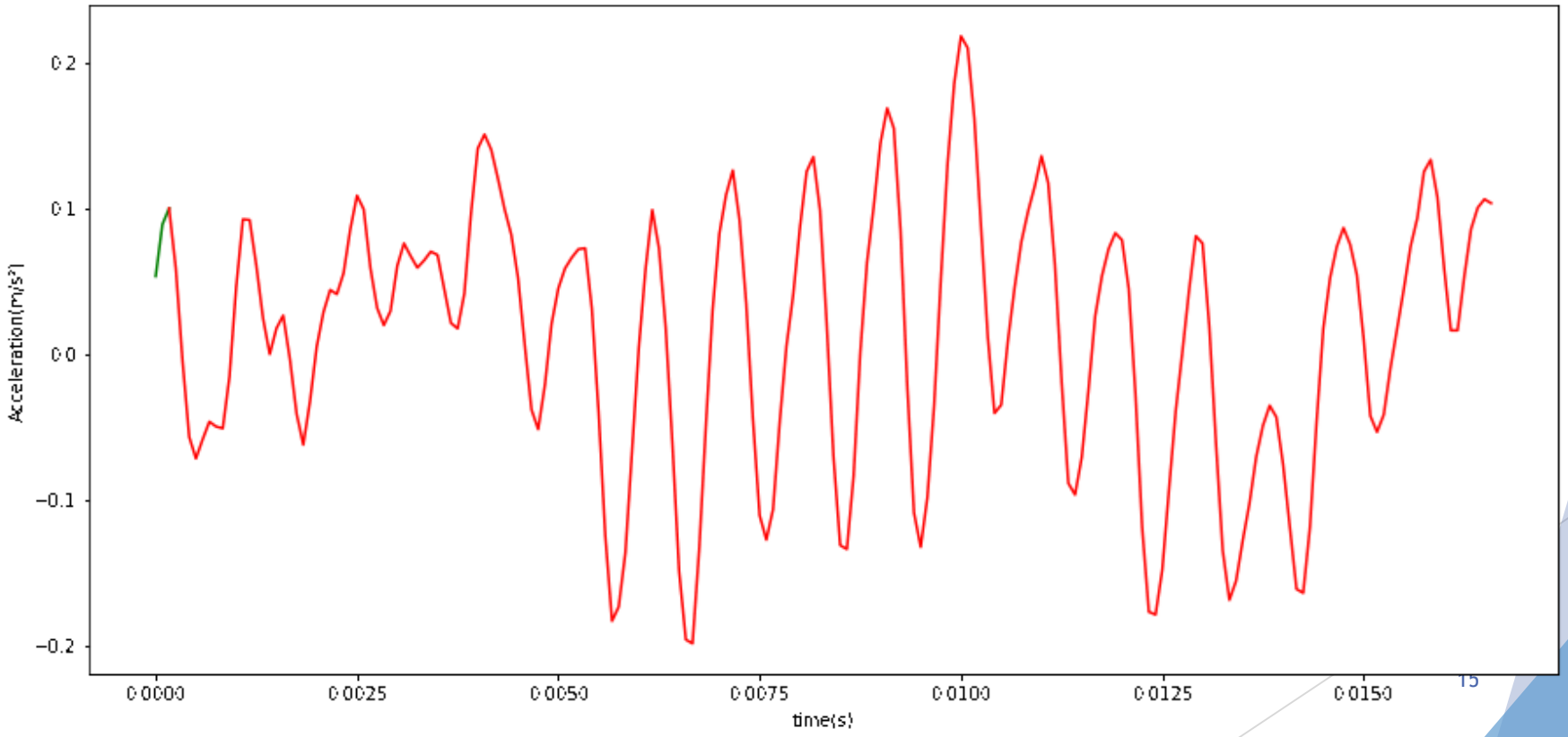
Acondicionamiento de la señal



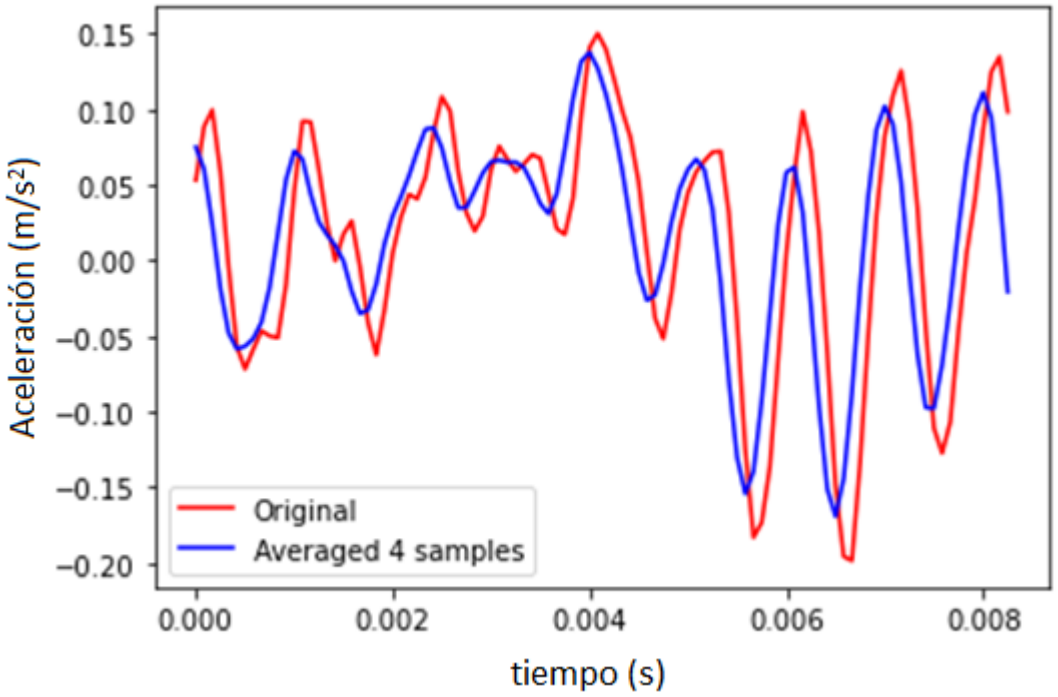
Base de Datos → **Acondicionamiento de la señal** → Reducción de dimensionalidad → Clasificador

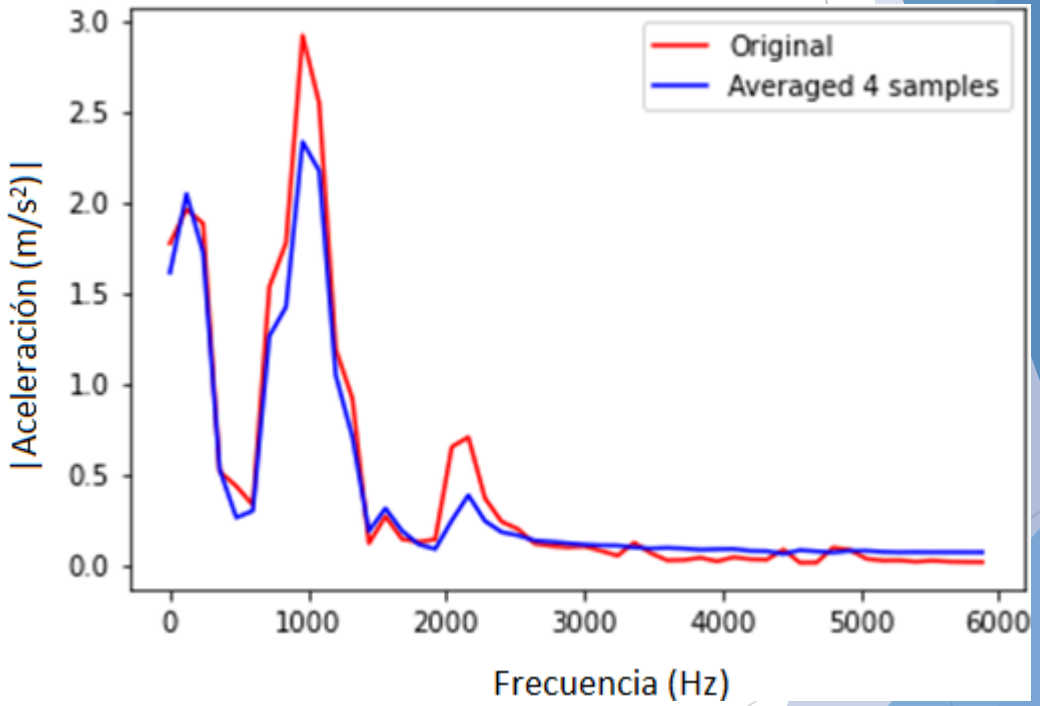
Etapa de filtrado recurrente

- Experimentalmente, se demostró que un promedio en los vectores de entrenamiento mejora el desempeño del autoencoder. Esto fue debido a que los métodos de optimización basados en gradiente obtienen derivadas numéricas más estables.



Etapa de filtrado recurrente





Reducción de dimensionalidad

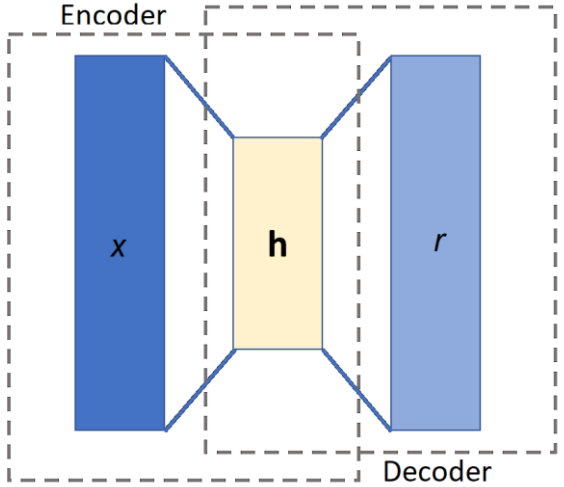
El autoencoder es una arquitectura fundamental en el aprendizaje no supervisado en Deep Learning, así como para las tareas de Reducción de Dimensionalidad.

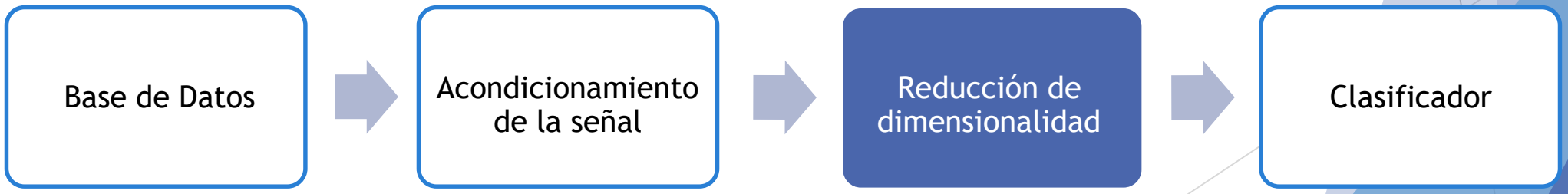
El Autoencoder reconstruye la entrada en la salida con la menor distorsión posible. Este proceso se introdujo en la década de los 80s con el enfoque de: "*backpropagation without a teacher*".

El autoencoder esta descrito por:

$$h = f(\boldsymbol{x}) \text{ encoder}$$
$$r = g(h) \text{ decoder}$$

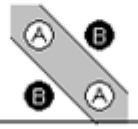
Al término del entrenamiento: $r \approx x$,
ésto es conocido como Reconstrucción.

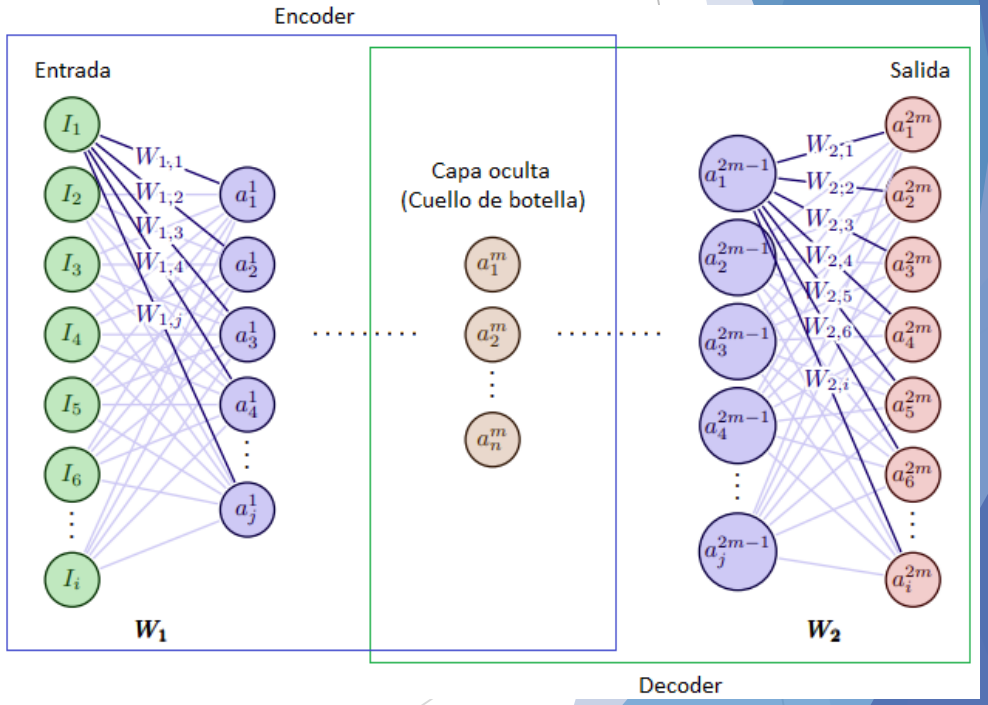




Entrenamiento de Autoencoder

- Con *Back-Propagation*, las capas de una red son entrenadas de manera iterativa desde la capa de salida hacia la entrada; esto introduce el problema de que en una red la influencia del entrenamiento en las capas cercanas a la entrada sea menor respecto a las cercanas a la salida.
- Por lo tanto para evitar este problema, se sigue el procedimiento de Transferencia de Aprendizaje y Ajuste Fino.

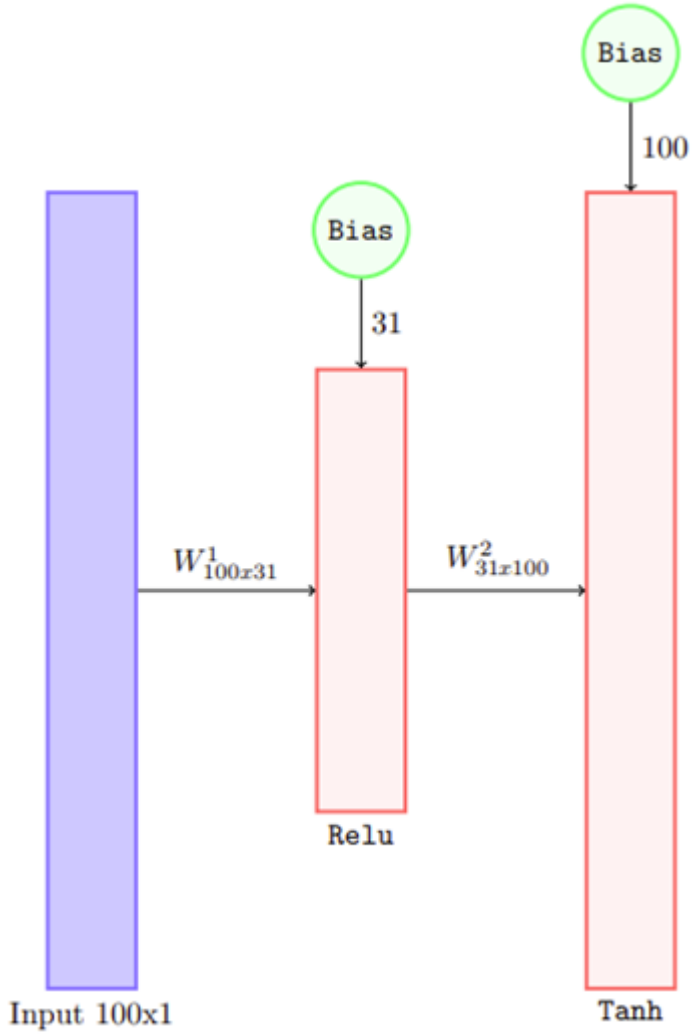
Estructura	Regiones de Desición	Problema de la XOR	Clases con Regiones Mezcladas	Formas de Regiones más Generales
1 Capa 	Medio Plano Limitado por un Hiperplano			
2 Capas 	Regiones Cerradas o Convexas			
3 Capas 	Complejidad Arbitraria Limitada por el Número de Neuronas			



Transferencia de Aprendizaje y Ajuste Fino para este trabajo

Primera etapa:

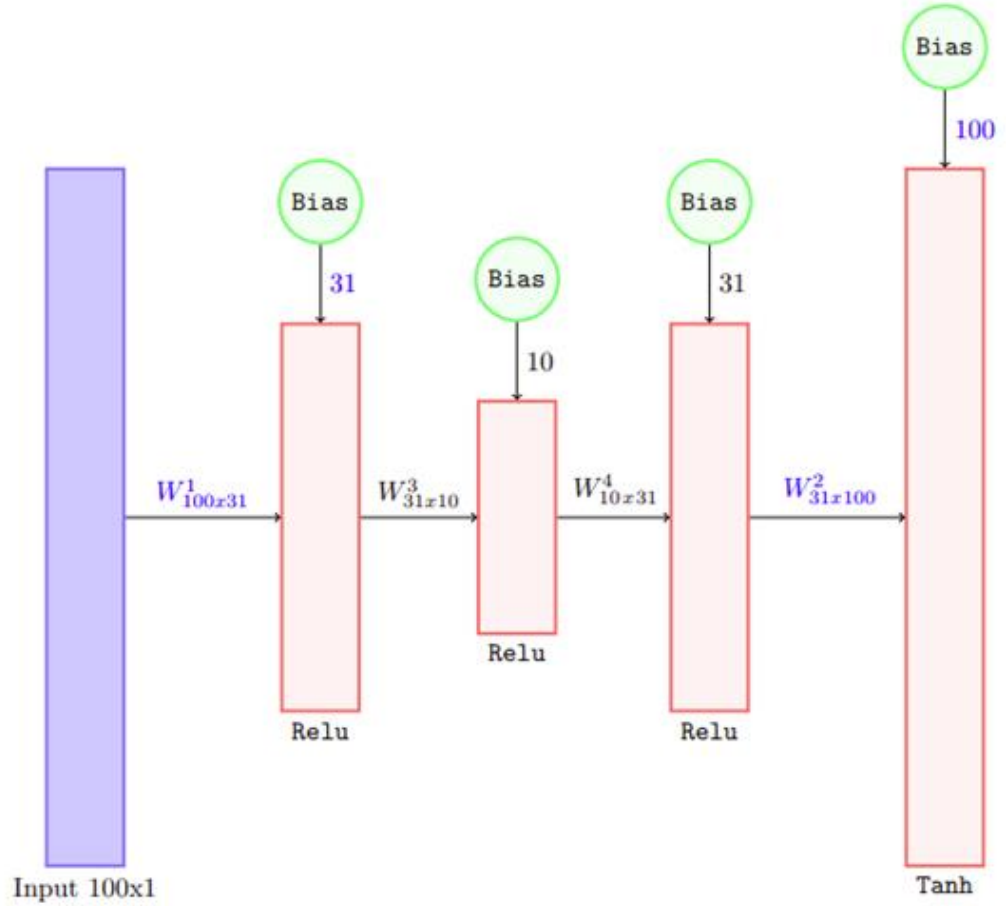
Se crea un Autoencoder con una capa oculta de 31 neuronas y una capa de salida de 100.



Transferencia de Aprendizaje y Ajuste Fino para este trabajo

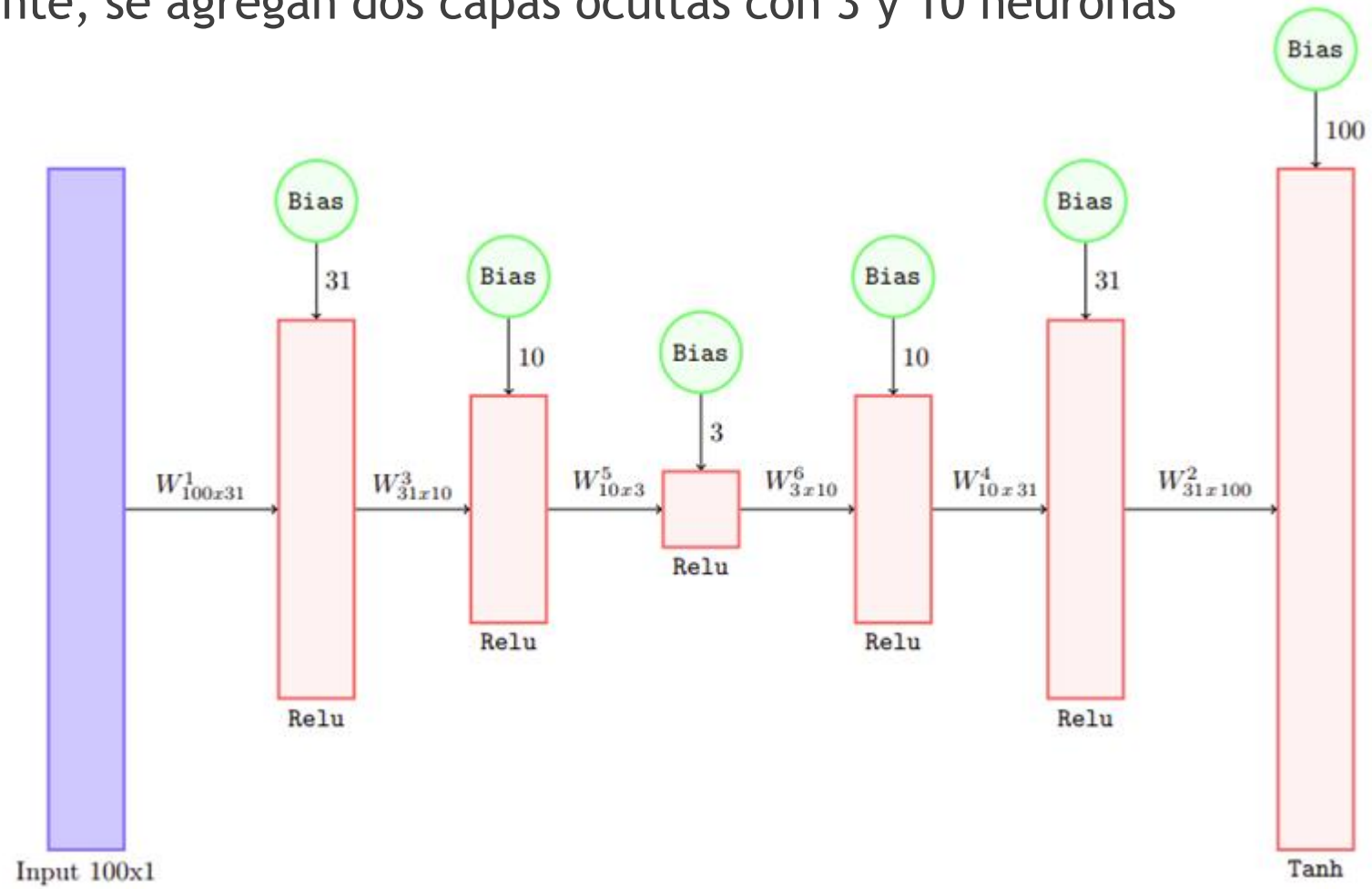
Segunda etapa:

Al autoencoder de la primera etapa, se agregan dos capas ocultas, una de 10 neuronas y otra de 31.



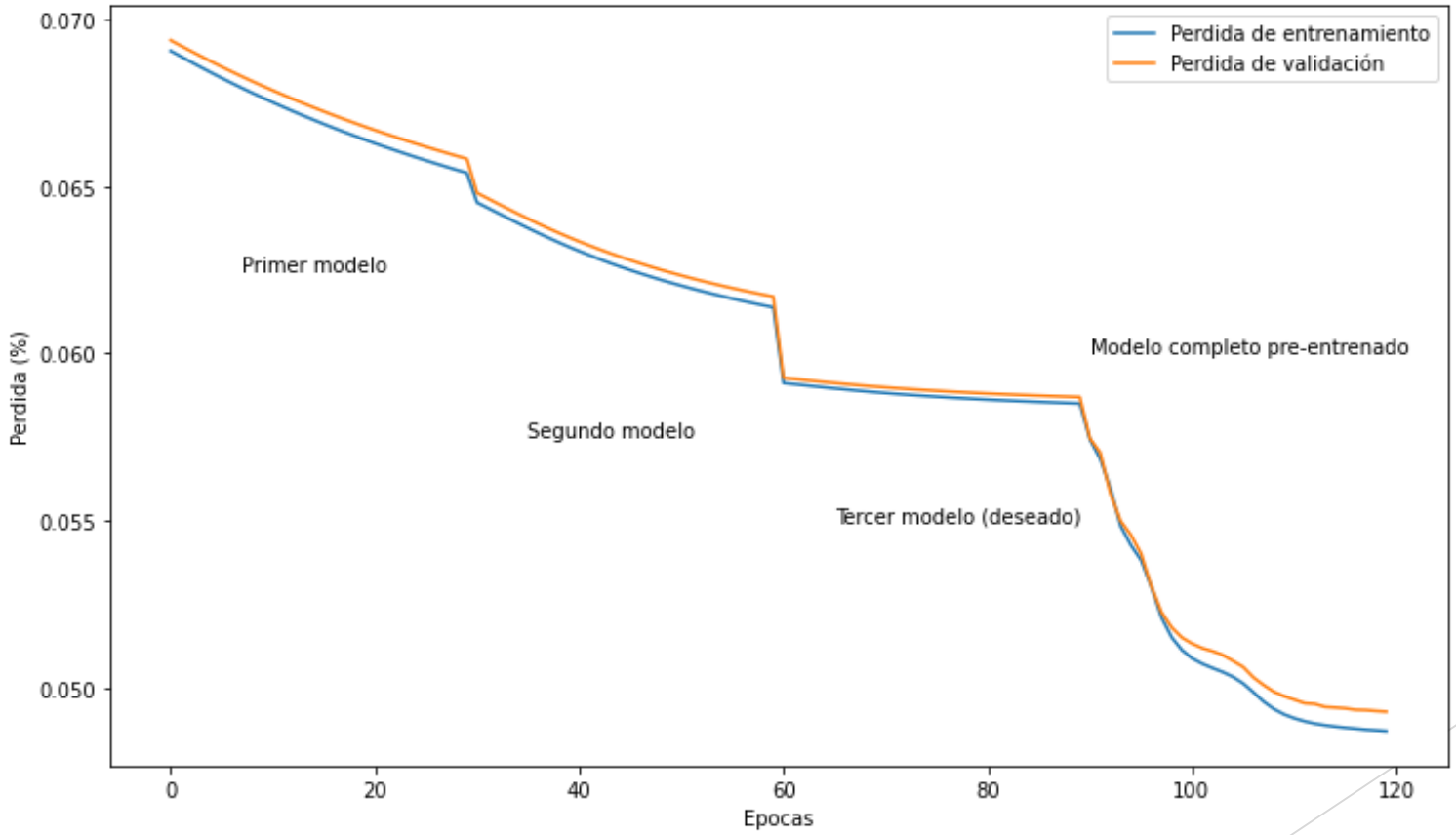
Transferencia de Aprendizaje y Ajuste Fino para este trabajo

Finalmente, se agregan dos capas ocultas con 3 y 10 neuronas



Transferencia de Aprendizaje y Ajuste Fino para este trabajo

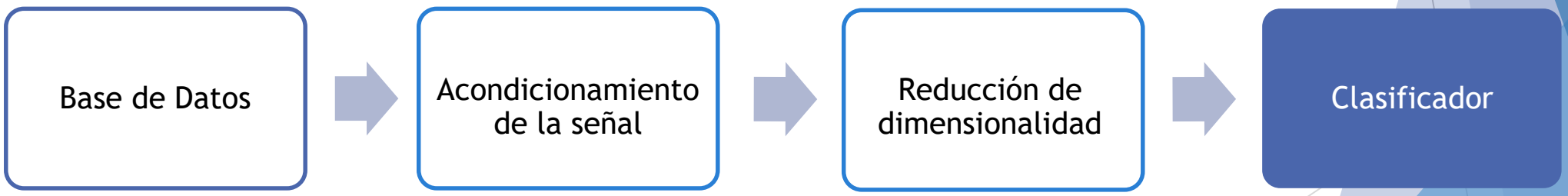
Gráficas: de entrenamiento y de validación.
-La gráfica de entrenamiento evalúa como el modelo esta aprendiendo.
-La gráfica de validación evalúa el desempeño del modelo con los datos de prueba.



Clasificador: Extreme Learning Machine

La arquitectura del modelo Extreme Learning Machine corresponde a una red neuronal de una sola capa oculta (Single Layer Feedforward Network, **SLFN**), que elige aleatoriamente los **pesos de la capa oculta** y determina analíticamente los **pesos de la capa de salida**.

En teoría, este algoritmo tiende a proveer un buen desempeño de generalización, mientras mantiene una velocidad de aprendizaje extremadamente rápida.

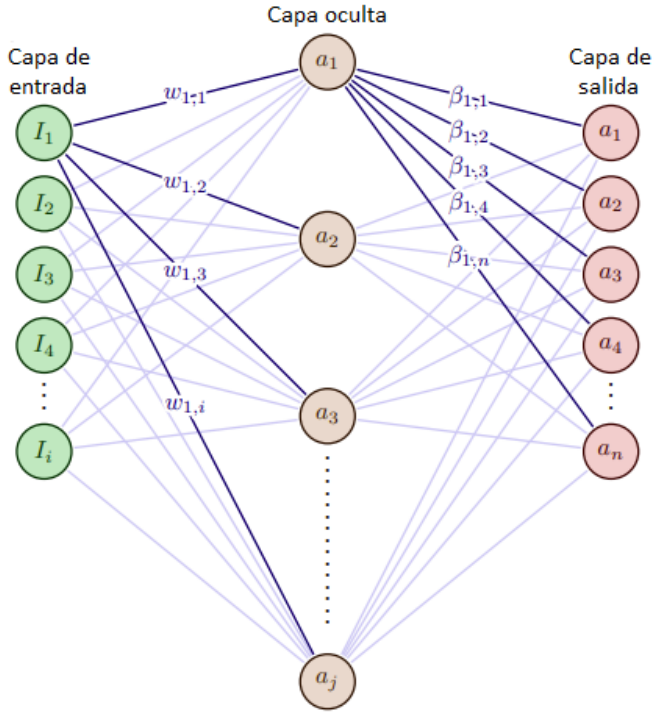


Arquitectura ELM

Pesos de la capa oculta:
Elegidos aleatoriamente

$$W_{1,1}, W_{1,2} \ \dots \ W_{1,j}$$
$$W_{2,1}, W_{2,2} \ \dots \ W_{2,j}$$

$$W_{i,1}, W_{i,2} \ \dots \ W_{i,j}$$



Pesos de la capa de salida:
Determinados mediante Moore Penrose

$$\beta_{1,1}, \beta_{1,2} \ \dots \beta_{1,n}$$
$$\beta_{2,1}, \beta_{2,2} \ \dots \beta_{2,n}$$

$$\beta_{j,1}, \beta_{j,2} \ \dots \beta_{j,n}$$

Entrenamiento de los pesos de salida

- 3.1.- Mediante la Ecuación 2.12, se calcula $\boldsymbol{H}$.

$$\mathbf{H} = \begin{bmatrix} g\left(w_1^T x_1 + b_1\right) & \cdots & g\left(w_d^T x_1 + b_d\right) \\ \vdots & \ddots & \vdots \\ g\left(w_1^T x_N + b_1\right) & \cdots & g\left(w_d^T x_N + b_d\right) \end{bmatrix} \in \mathbb{R}^{Nxd}, \quad (2.12)$$

- 3.2.- Mediante la Ecuación 2.14, se calcula $\boldsymbol{Y}$, mediante $B(2^{m-1})$.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix} = \begin{bmatrix} y_{11} & \cdots & y_{1m} \\ \vdots & \ddots & \vdots \\ y_{N1} & \cdots & y_{Nm} \end{bmatrix} \in \mathbb{R}^{Nxm}, \quad (2.14)$$

- 3.3 Se calcula la norma mínima de mínimos cuadrados, primeramente obteniendo $\boldsymbol{B}$, de la Ecuación 2.15, posteriormente se obtiene el vector de pesos β_i de $B^T = [\beta_1, \ldots, \beta_d]$.

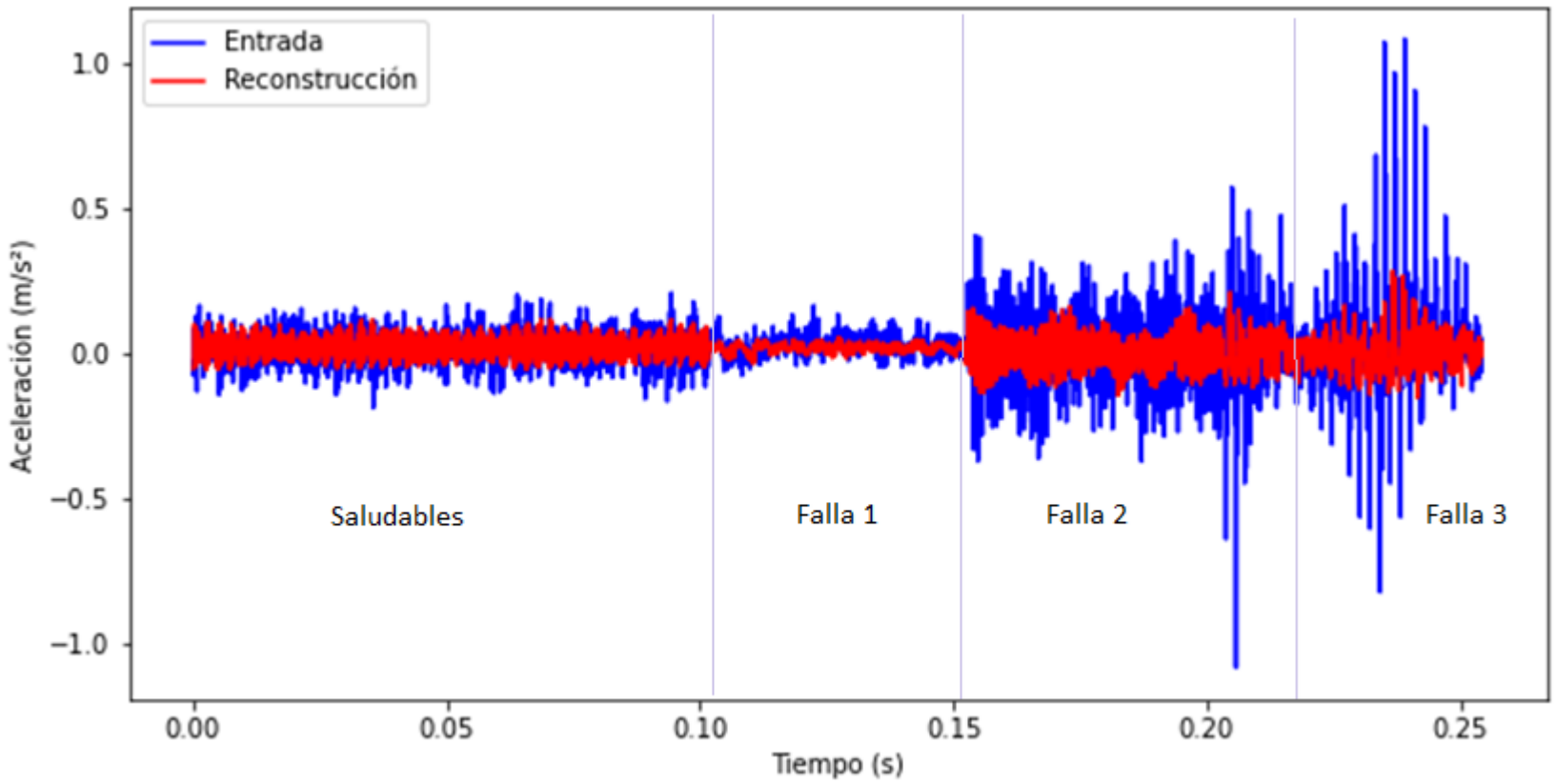
$$\mathbf{B} = \mathbf{H}^{\dagger}\mathbf{Y} \quad (2.15)$$

- Donde $\mathbf{H}^{\dagger}$ es la matriz pseudoinversa de Moore Penrose.

Contenido

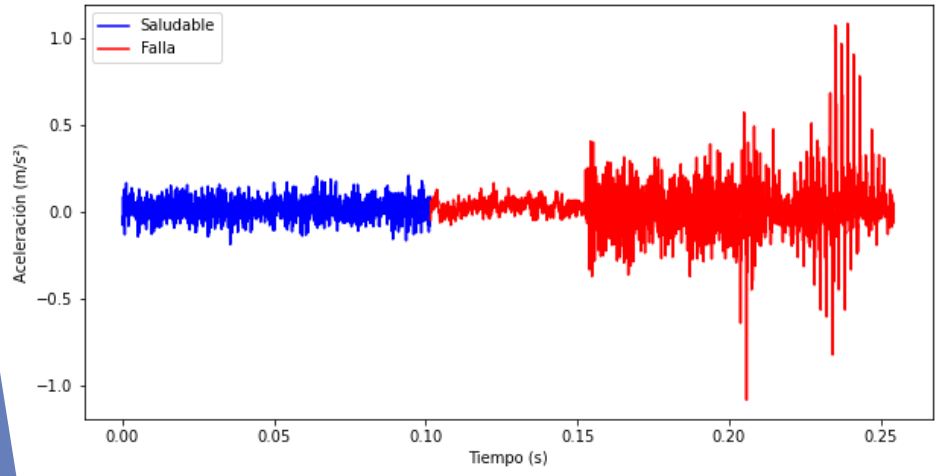
- 1.- Justificación
- 2.- Objetivos
- 3.- Base de datos
- 4.- Metodología propuesta
- 5.- Resultados
- 6.- Conclusiones
- 7.- Perspectivas

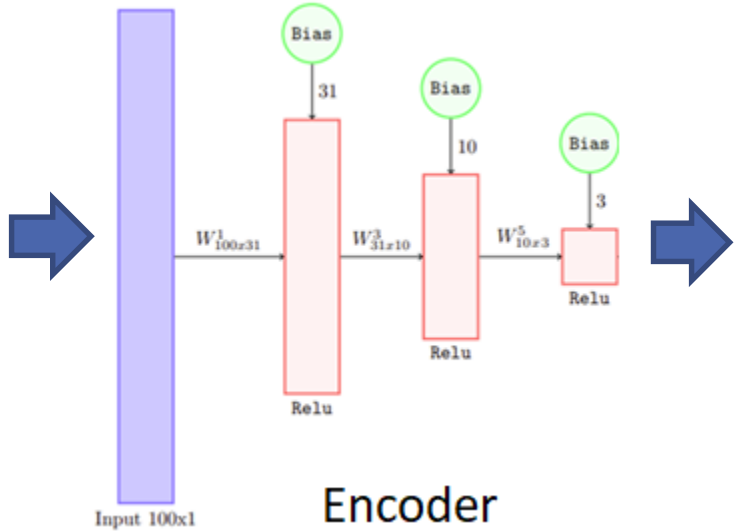
Reconstrucción

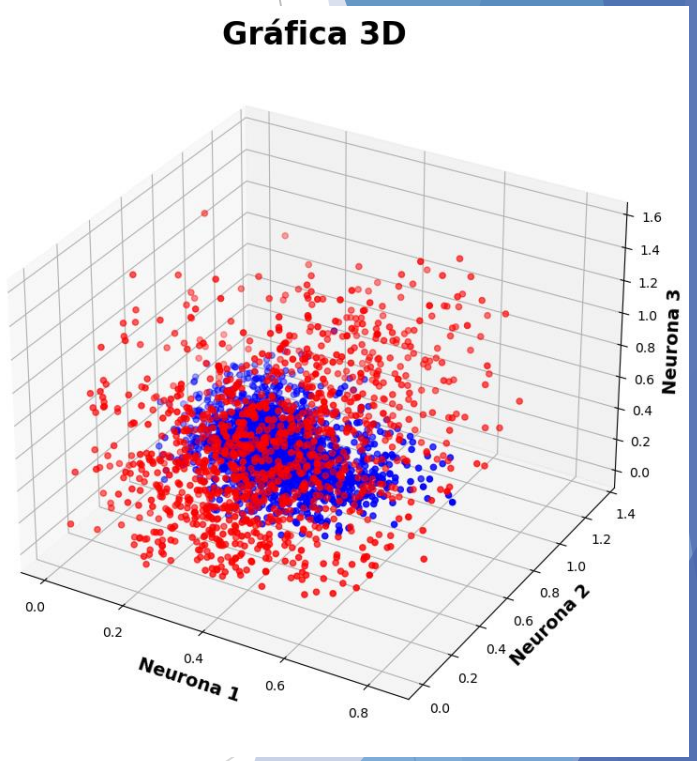


Reconstrucción de la entrada

Información del encoder

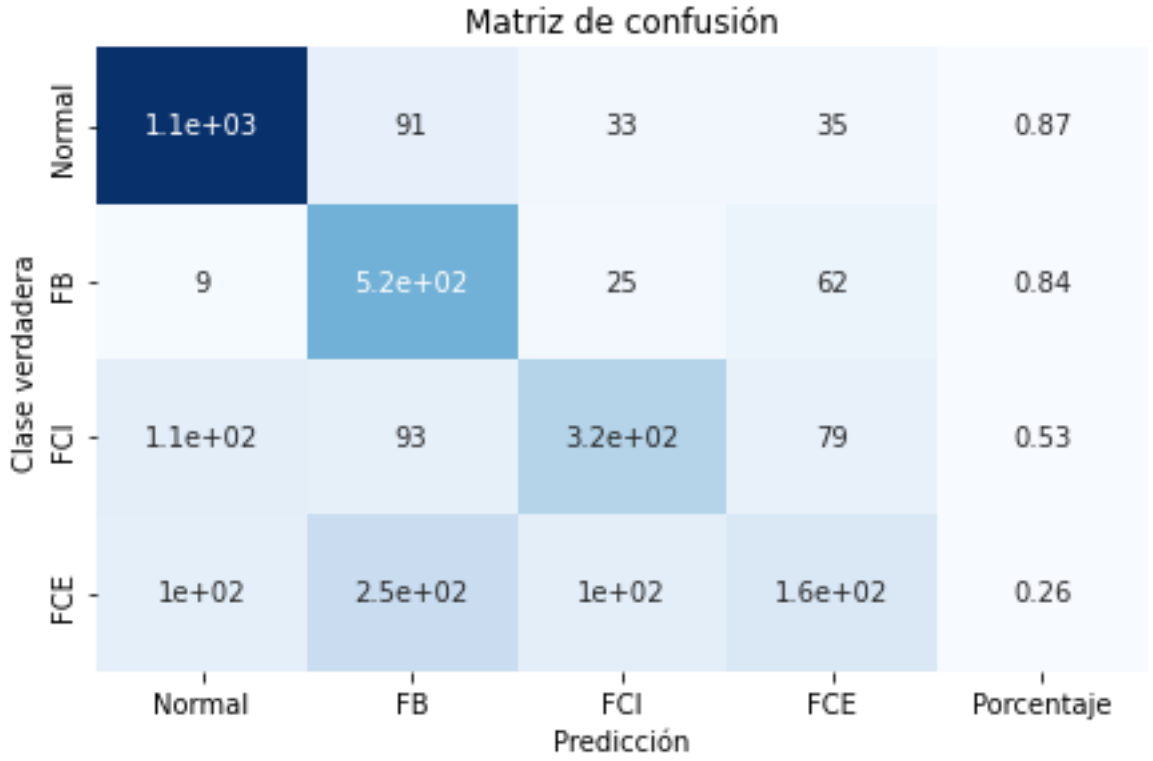




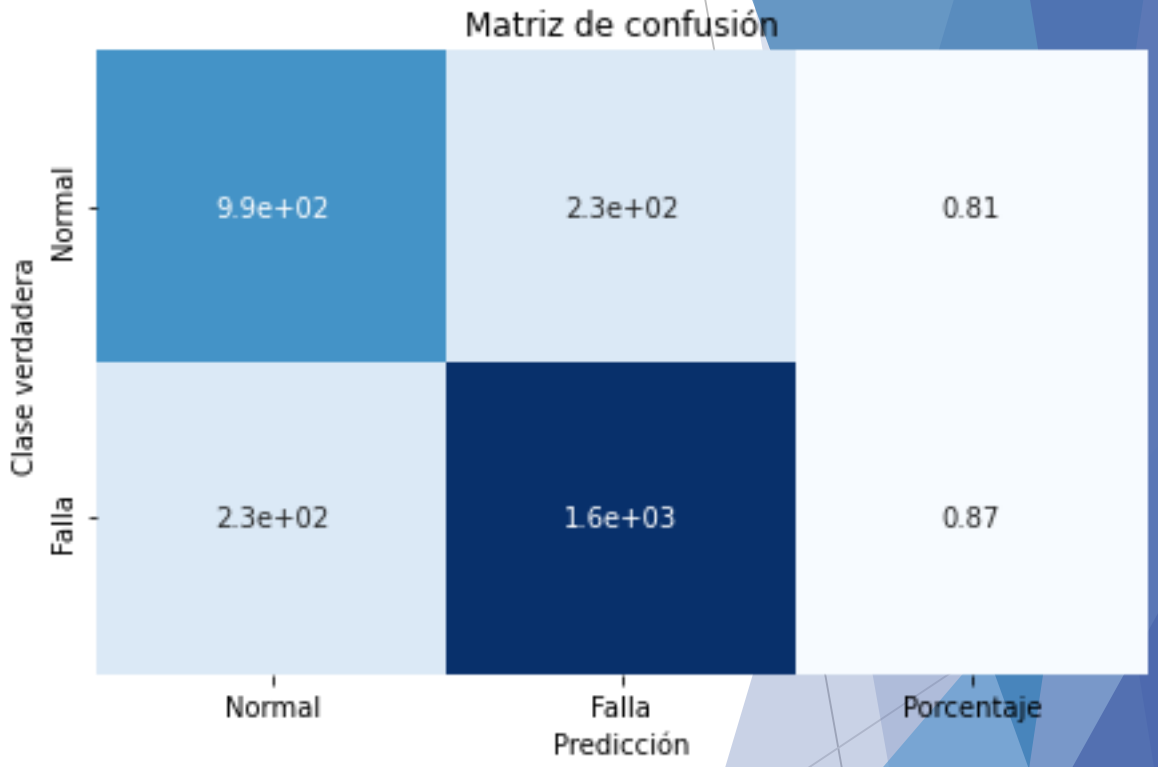


Reducción de Dimensionalidad:
de 100 a 3

Clasificación

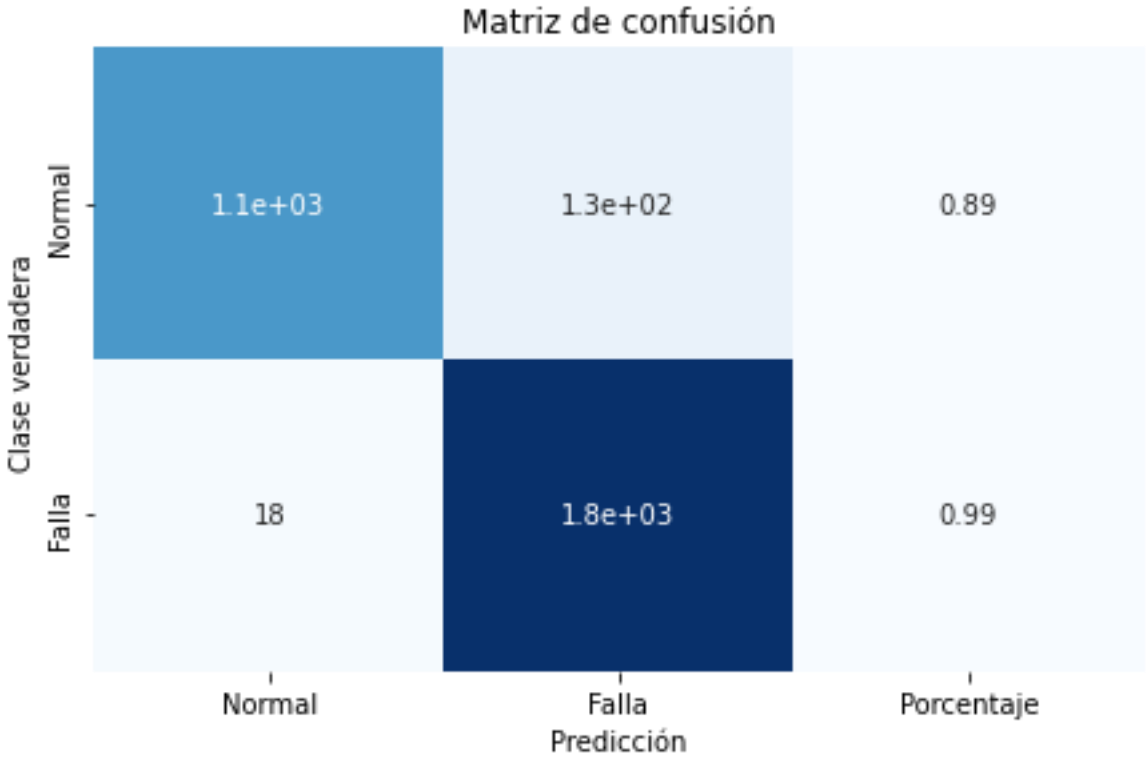


Matriz de confusión para 4 clases

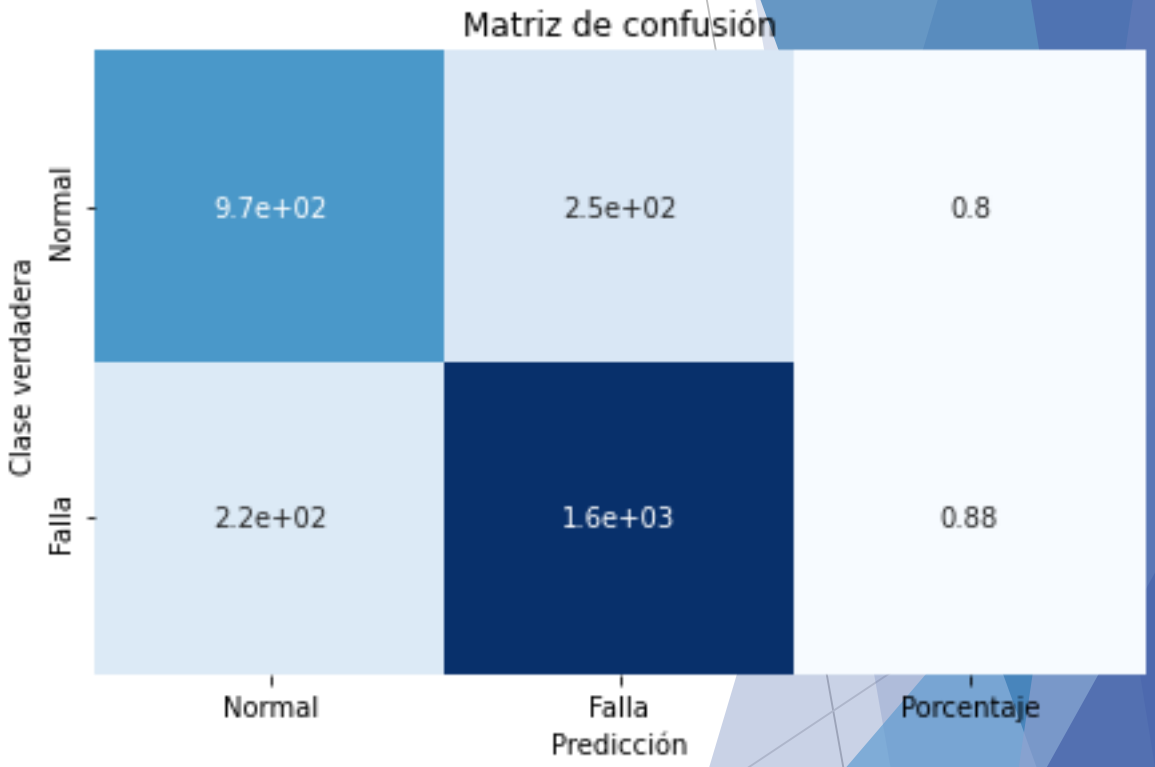


Matriz de confusión a 2 clases
Para diámetro de falla de 0.007”

Clasificación



Matriz de confusión a 2 clases
Para diámetro de falla de 0.014”

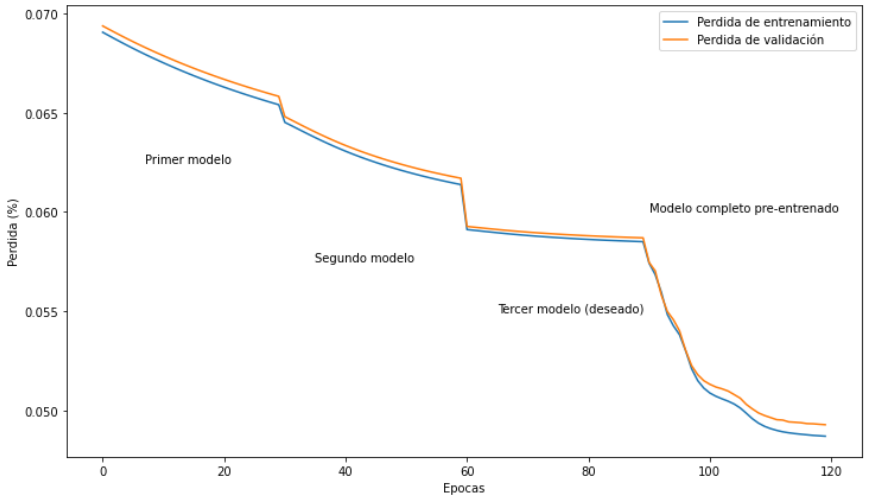


Matriz de confusión a 2 clases
Para diámetro de falla de 0.021”

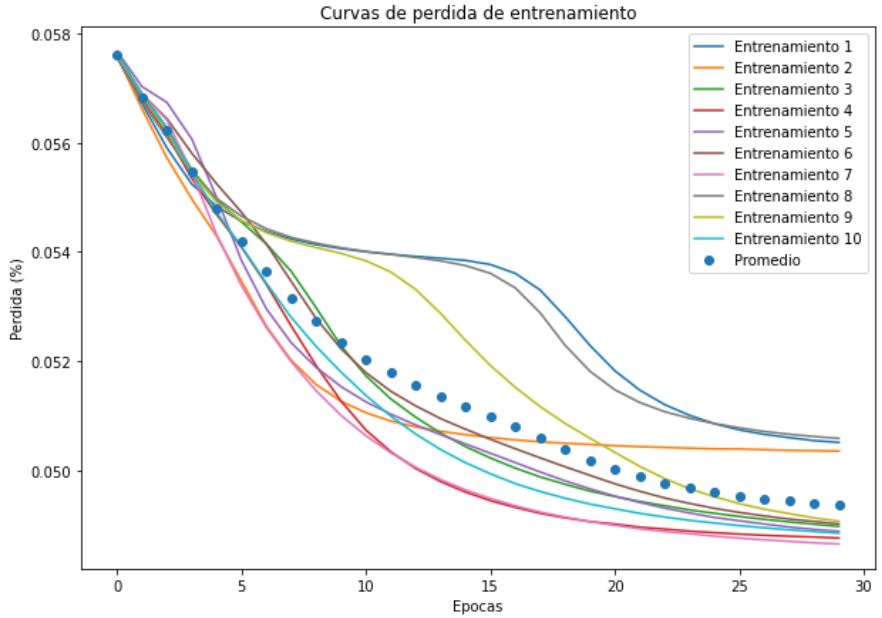
Experimentación de selección de híper-parámetros

Tasa de aprendizaje, algoritmo de optimización, épocas, tamaño de lote.

Transferencia de Aprendizaje y Sintonía Fina.



Back-Propagation



Algoritmo de optimización	Basado en gradiente	Libres de gradiente	Evolutivos e inteligencia de colonia
Método de Newton	X		
Búsqueda de línea	X		
SGD	X		
CG	X		
Adam	X		
Nadam	X		
Nelder-Mead		X	
Levenberg-Marquardt		X	
Algoritmo genético			X
Evolución diferencial			X
Enjambre de partículas			X

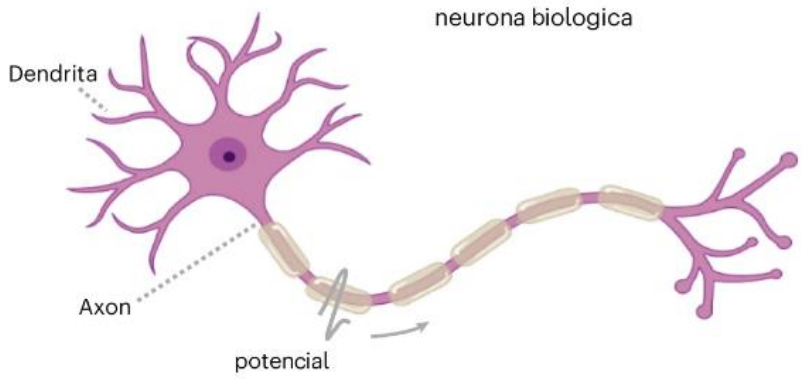
Propuesta de neurona digital en FPGA

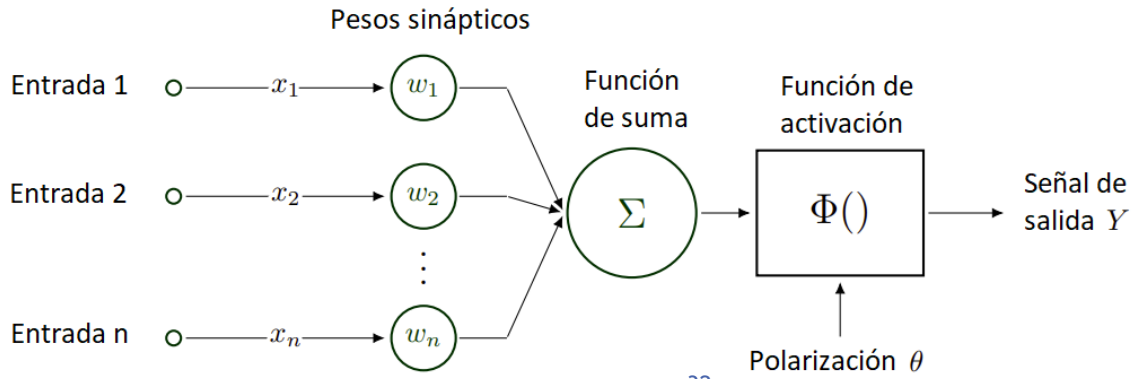
En este trabajo se propuso un sistema digital para una neurona, que comprende las siguientes etapas:

1.- Multiplicación de pesos con entradas y acumulación de productos.

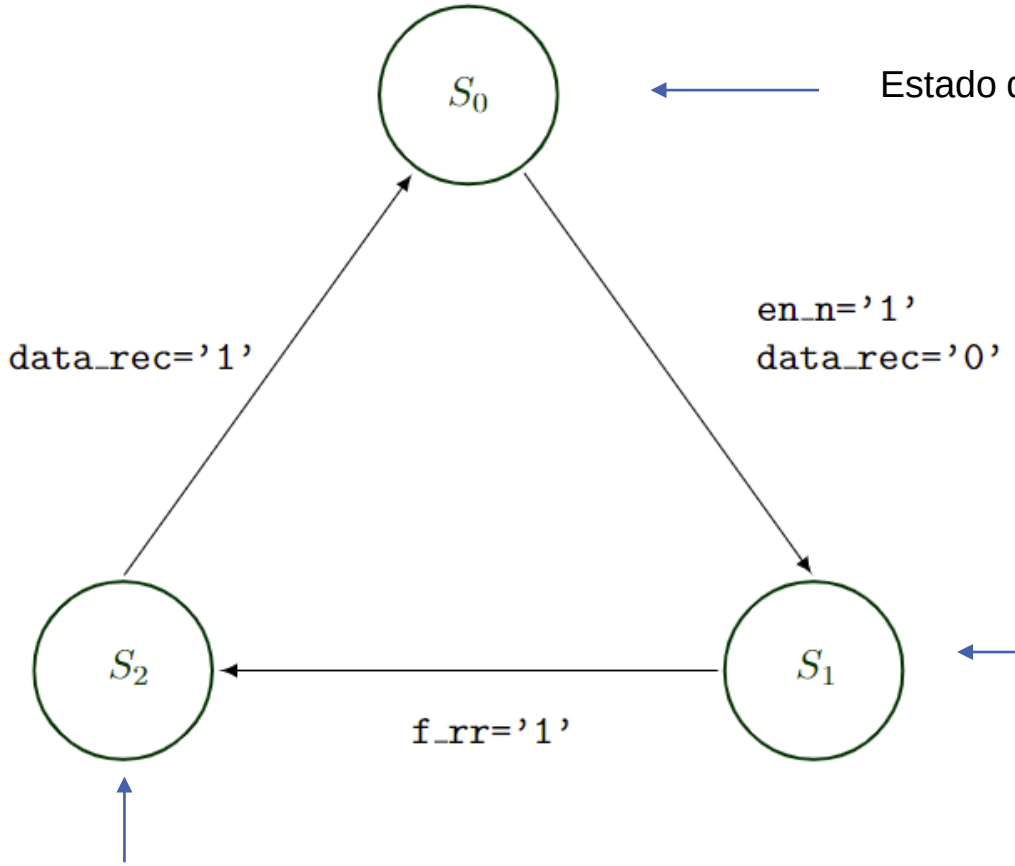
2.- Aplicación de la función de activación y polarización.

3.- Cómputo de salida.





Secuencia del proceso digital mediante máquina de estados de Moore

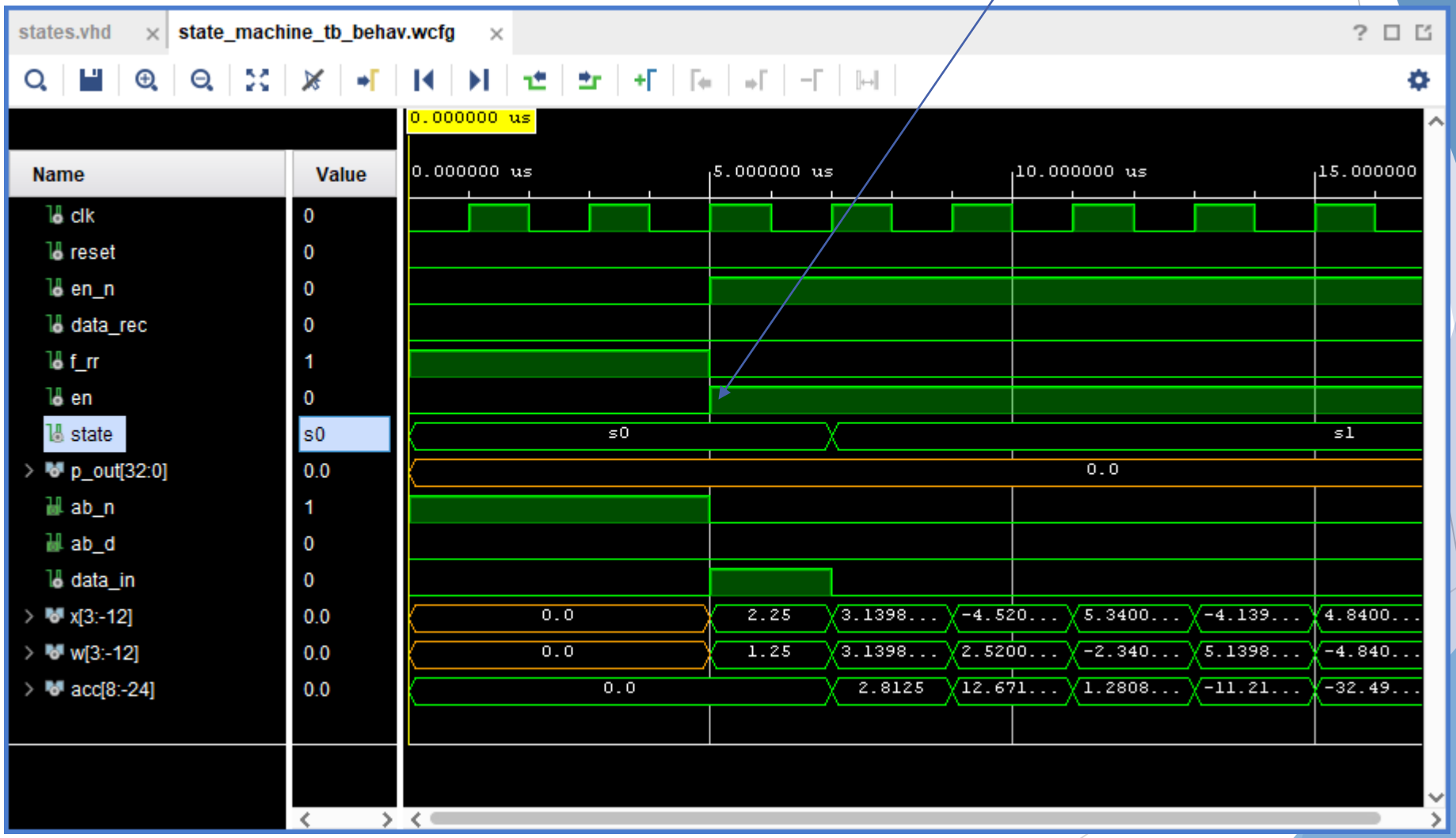


Estado de espera.

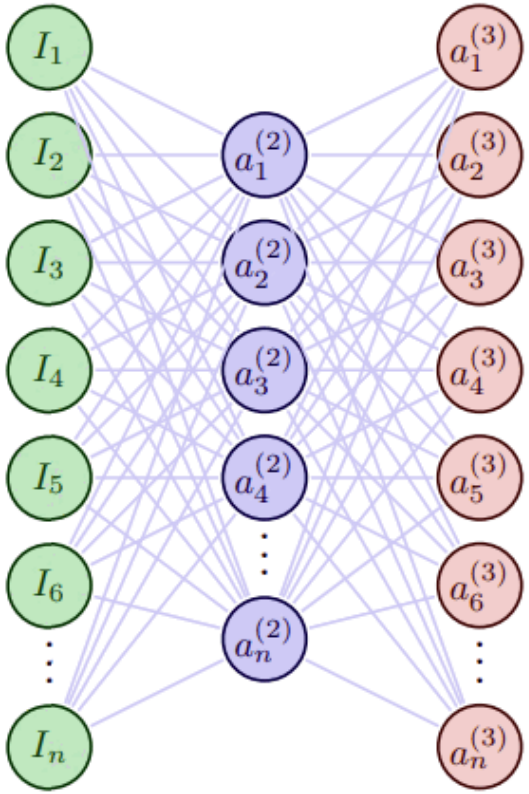
Estado de cómputo

Estado de realimentación

Proceso digital



Arquitectura Final



Contenido

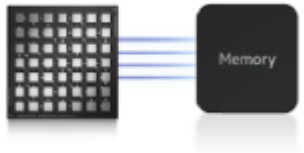
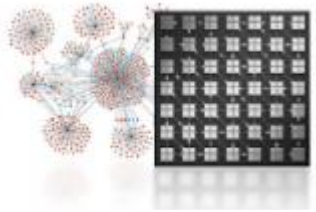
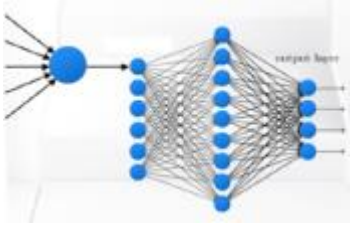
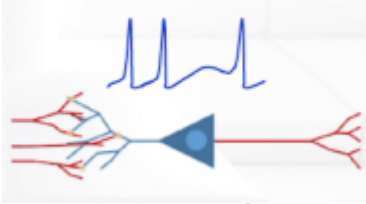
Conclusiones

- En este trabajo se exploraron los procesos de aprendizaje en máquina de un problema complejo (un sistema de rodamiento con fallas) a través de una base de datos. Se concluye que este método es eficiente para abordar problemas de ingeniería de este tipo.
- Las arquitecturas profundas aprenden de una manera mas eficiente, en comparación con aquellas con profundidad somera.
- Es posible implementar redes neuronales en hardware mediante unidades de procesamiento numérico (neurona digital), que están interconectadas y organizadas en capas.

Contenido

- 1.- Justificación
- 2.- Objetivos
- 3.- Base de datos
- 4.- Metodología propuesta
- 5.- Resultados
- 6.- Conclusiones
- 7.- Perspectivas

Esquema de arquitecturas neuro-inspiradas

Cómputo von Newmann	Cómputo Paralelo	Cómputo Neuromórfico
		
Funcionamiento basado en descripción de instrucciones.	**Entrenamiento fuera de línea, usando conjuntos de datos etiquetados.**	**Aprendizaje sobre la marcha mediante reglas de activación neuronal.**
Proceso síncrono.	Proceso síncrono.	Asíncrono, basado en eventos.
		

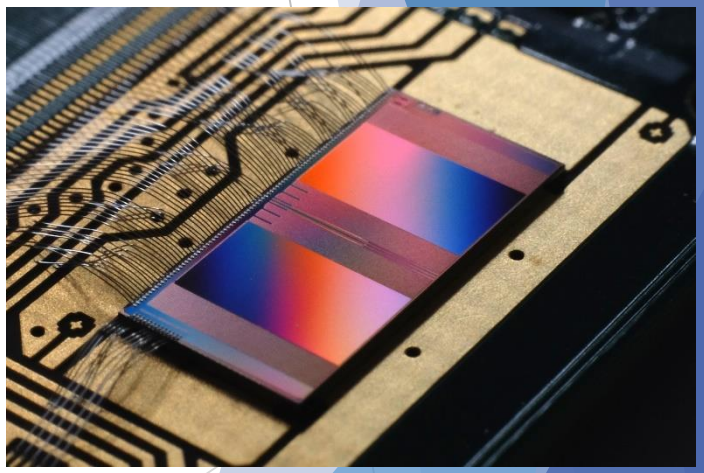
Tendencias

Human Brain Project

189 Publicaciones

La máquina de múltiples núcleos **SpiNNaker,** es un paradigma de procesamiento neuromorfico que está ubicada en Manchester, Reino Unido, con una red basada en paquetes, optimizada para el intercambio de potenciales de activación neuronal.

El Sistema SpiNNaker cuenta con casi 30 000 chips, cada uno basado en arquitectura ARM de dieciocho núcleos y RAM local compartida de 128 Mb. Un solo chip puede simular 16.000 neuronas con ocho millones de sinapsis plásticas funcionando en tiempo real con un consumo energético de 1W.



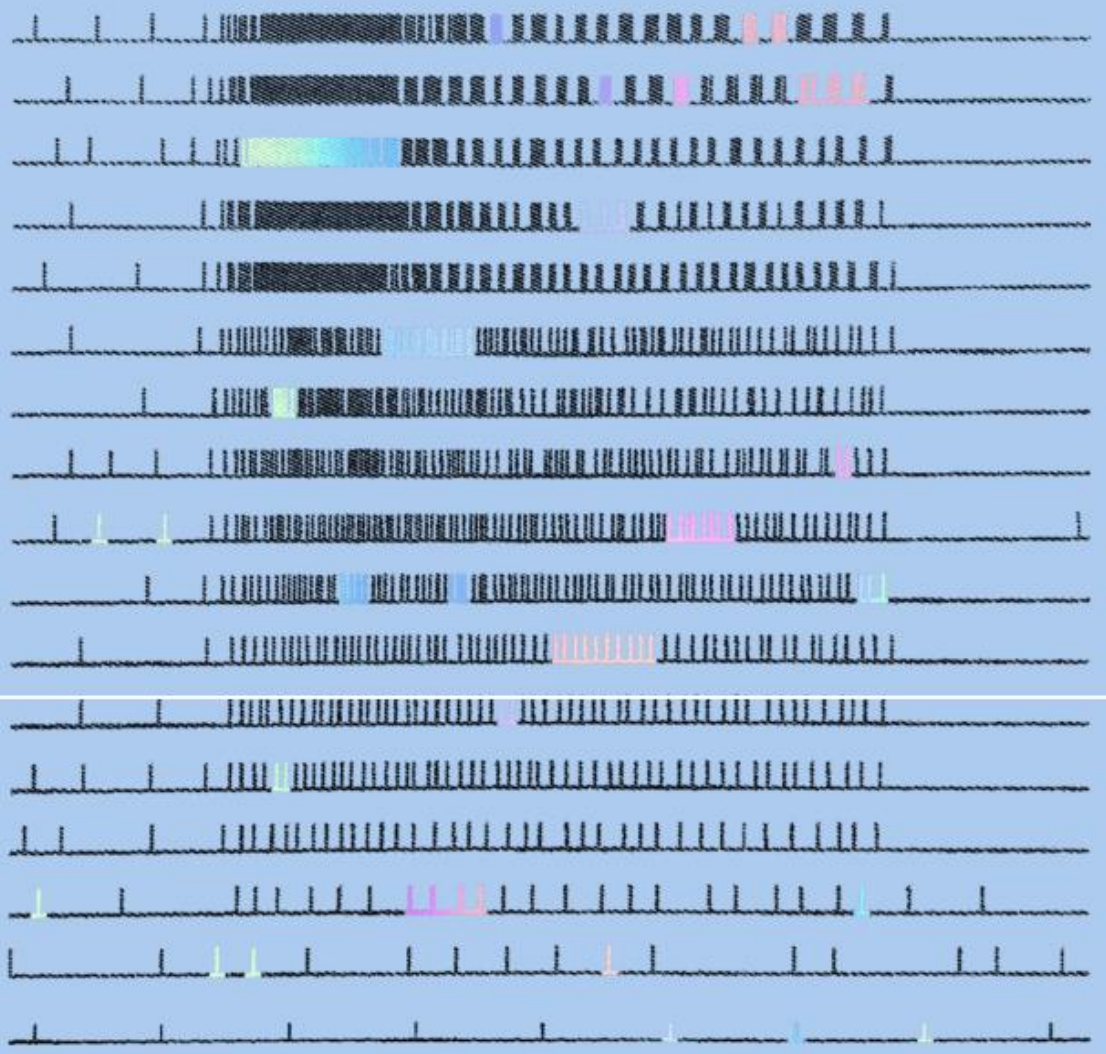
Tendencias

“Loihi 2 y Lava recogen los resultados de varios años de investigación en colaboración con Loihi. Nuestro chip de segunda generación mejora enormemente la velocidad, la programabilidad y la capacidad del procesamiento neuromórfico, ampliando sus usos en aplicaciones de computación inteligente con limitaciones de potencia y latencia. Estamos abriendo Lava para responder a la necesidad de convergencia de software, evaluación comparativa y colaboración entre plataformas en este campo, y para acelerar nuestro progreso hacia la viabilidad comercial”.

NEURALINK

Interfacing with the Brain

Innovation pushing the boundaries of neural engineering.



¡Muchas gracias por su atención!