

UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO

ESCUELA DE POSGRADO

MAESTRÍA EN CIENCIAS



PLAN DE TESIS

---

---

TITULO DE LA TESIS

---

---

*Tesis presentado por:*

Nombre1 Nombre2 ApellidoP ApellidoM

*Asesor:*

Dr. Nombre del asesor

Cusco - Peru

2024

## **Informe de Similitud**

## Dedicatoria

*Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.*

## **Agradecimientos**

*Aqui va el agradecimiento*

## Índice General

<b>Dedicatoria</b> . . . . .	<b>I</b>
<b>Agradecimientos</b> . . . . .	<b>II</b>
<b>Resumen</b> . . . . .	<b>VII</b>
<b>Abstract</b> . . . . .	<b>VIII</b>
<b>Palabras clave</b> . . . . .	<b>IX</b>
<b>I Planteamiento del Problema</b> . . . . .	<b>1</b>
1.1 Situación problemática . . . . .	1
1.2 Formulación del problema . . . . .	1
1.2.1 Problema general . . . . .	1
1.2.2 Problemas específicos . . . . .	1
1.3 Justificación de la investigación . . . . .	1
1.4 Objetivos de la investigación . . . . .	1
1.4.1 Objetivo general . . . . .	1
1.4.2 Objetivos específicos . . . . .	1
<b>II MARCO TEÓRICO CONCEPTUAL</b> . . . . .	<b>2</b>
2.1 Bases teóricas . . . . .	2
2.1.1 Redes neuronales artificiales . . . . .	2
2.1.2 Deep learning . . . . .	2
2.1.3 Estructura . . . . .	5
2.2 Aprendizaje profundo . . . . .	8
2.2.1 Perceptrón multicapa [MLP] . . . . .	8
2.2.2 Non-Linearities . . . . .	10
2.3 Marco conceptual (palabras clave) . . . . .	11
2.3.1 Tablas . . . . .	11

2.4	Antecedentes empíricos de la investigación (estado del arte)	11
2.5	Hipótesis *	11
2.5.1	Hipótesis general *	11
2.5.2	Hipótesis específicas *	11
2.6	Identificación de variables e indicadores *	11
2.7	Operacionalización de variables *	11
<b>III</b>	<b>METODOLOGÍA</b>	<b>12</b>
3.1	Ámbito de estudio: localización política y geográfica	13
3.2	Tipo y nivel de investigación	13
3.3	Unidad de análisis	13
3.4	Población de estudio	13
3.5	Tamaño de muestra	13
3.6	Técnicas de selección de muestra	13
3.7	Técnicas de recolección de información	13
3.8	Técnicas de análisis e interpretación de la información	13
3.9	Técnicas para demostrarla verdad o falsedad de las hipótesis planteadas	13
<b>IV</b>	<b>RESULTADOS Y DISCUSIÓN</b>	<b>14</b>
4.1	Procesamiento, análisis, interpretación y discusión de resultados	14
4.2	Pruebas de hipótesis *	14
4.3	Presentación de resultados	14
<b>V</b>	<b>CONCLUSIONES Y RESULTADOS</b>	<b>15</b>
	<b>ANEXOS</b>	<b>18</b>

# Índice de tablas

1	Results from Classifying MNIST . . . . .	10
2	Tabla comparativo de resultados . . . . .	11

## Índice de figuras

1	Neurona o Nodo: Unidad basica de Deep Learning . . . . .	5
2	Linear Function . . . . .	7
3	To turn logits into probabilities, the activation function was chosen to be softmax . . . . .	7
4	Basic two layer Neural Network . . . . .	9
5	Dos redes neuronales totalmente conectadas(Li y Karpathy, 2016)	9

## Resumen

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

## Abstract

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

## Palabras clave

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

## **Capítulo I**

### **Planteamiento del Problema**

#### **1.1 Situación problemática**

#### **1.2 Formulación del problema**

##### **1.2.1. Problema general**

##### **1.2.2. Problemas específicos**

#### **1.3 Justificación de la investigación**

#### **1.4 Objetivos de la investigación**

##### **1.4.1. Objetivo general**

##### **1.4.2. Objetivos específicos**

## **Capítulo II**

### **MARCO TEÓRICO CONCEPTUAL**

#### **2.1 Bases teóricas**

##### **2.1.1. Redes neuronales artificiales**

Son modelos matemáticos inspirados en las estructuras y procesos del cerebro humano. Están diseñadas para aprender y mejorar sus capacidades a través del entrenamiento con grandes cantidades de datos.

##### **2.1.2. Deep learning**

El aprendizaje profundo es un campo tecnológico de rápido crecimiento que a menudo se describe como un potencial ilimitado. Este artículo describe su historia, el motivo de la explosión de popularidad y cómo funciona. Se explorará un ejemplo de clasificación de imágenes de dígitos escritos a mano (MNIST) utilizando una red completamente conectada y una red neuronal convolucional.

A continuación, una breve descripción de las herramientas necesarias para que el lector implemente su propia red. Finalmente, una visión del estado del arte que están desarrollando empresas como Google, Facebook y Baidu.

#### **2.1.2.1. Historia**

Los primeros algoritmos similares al aprendizaje profundo fueron inventados por Ivakhnenko y Lapa en 1965. Se realizó mucho trabajo e innovación en los años 1980 (las redes neurales convolucionales de Fukushima) y 1990 (LeNet de LeCun). Dettmers (2015b) Muchas de estas técnicas todavía están disponibles. utilizado hoy en día. Sin embargo, en aquel entonces las computadoras eran lentas y los conjuntos de datos pequeños. Los investigadores no encontraron muchas aplicaciones para las redes neuronales (NN), por lo que durante la década de 2000 la investigación decayó. No fue hasta los últimos años que NN resurgió.

El gran cambio se debió a una mayor potencia computacional y a un mayor número de datos. Primero fue la introducción de la unidad de procesamiento de gráficos (GPU). Las GPU aumentaron la velocidad de procesamiento computacional en un factor de 1000 en el lapso de 10 años. Dettmers (2015b) La

segunda razón para el regreso de NN fue el aumento exponencial de los datos. La tecnología nos ha permitido almacenar más datos e Internet nos ha permitido compartir esos datos.

La combinación de tecnología exponencial y datos ha permitido que el aprendizaje profundo rompa récord tras récord.

- *Reconocimiento de voz*: En 2009, Microsoft y la Universidad de Toronto mejoraron el reconocimiento de voz en un 30 % utilizando DL. Markoff (2012)
- *Computer Vision*: Hay una competencia anual llamada ImageNet donde los equipos compiten para clasificar una biblioteca de 14 millones de imágenes en 20,000 categorías. En 2012, Alex Krizhevsky y Geoff Hinton presentaron un algoritmo de aprendizaje profundo, AlexNet, que alcanzó una tasa de error del 15 % (un 40 % mejor que el estado del arte). Krizhevsky et al. (2010)
- *Descubrimiento de fármacos*: En 2011, Geoff Hinton y un equipo de la Universidad de Toronto ganaron el "Merck Molecular Activity Challenge" para el descubrimiento automático de fármacos. Utilizaron el aprendizaje

profundo para determinar qué molécula tenía más probabilidades de ser un fármaco eficaz. Lo sorprendente fue que nadie en el equipo tenía experiencia en química, biología o ciencias biológicas y lo hicieron en dos semanas.

Markoff (2012)

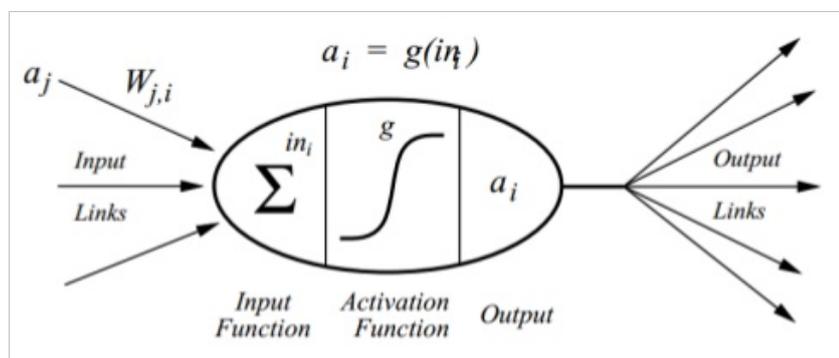
### 2.1.3. Estructura

#### 2.1.3.1. Neurona o Nodo

El componente básico de una red es la neurona o nodo (Figura 1). Toma algunos datos de entrada, aplica una función lineal a esas entradas calculando una suma ponderada y aplica una función de activación a esa suma.

#### Figura 1

*Neurona o Nodo: Unidad básica de Deep Learning*



*Nota:* tomado de (Bakó et al., 2021)

La función lineal se define como

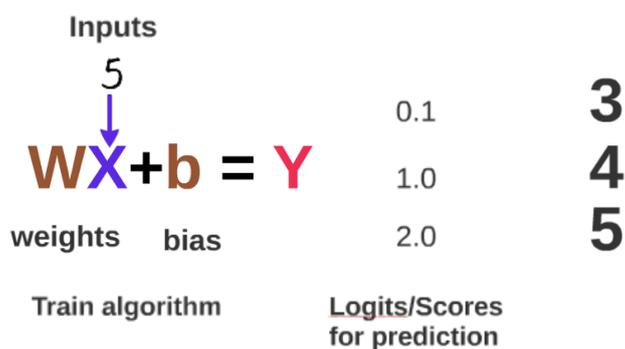
$$WX + b = Y \quad (2.1)$$

donde  $X$  denota un vector de entrada,  $W$  denota una matriz de pesos,  $b$  denota los sesgos y  $Y$  denota las *puntuaciones* o *logits*. El entrenamiento se realiza tratando de encontrar los pesos y sesgos que sean buenos para predecir la clase correcta. Por ejemplo, tomemos un modelo que intenta aprender dígitos escritos a mano con una entrada como la imagen de un "5" escrito a mano. La función lineal (Fig 2) toma esa entrada y genera logits. Al principio estos resultados no significan mucho. La tarea consiste en determinar la probabilidad de que la imagen pertenezca a cada clase (dígito). La forma de convertir logits en probabilidades es aplicar un softmax como nuestra función de activación, consulte la figura 3.

La función softmax genera las probabilidades de que la imagen pertenezca a cada clase (la más probable es cercana a 1 y la menos probable es cercana a 0). La técnica de codificación One-hot se utiliza para convertir cada etiqueta en un vector de membresía de clase. Este vector tiene el valor 1 para la clase correcta

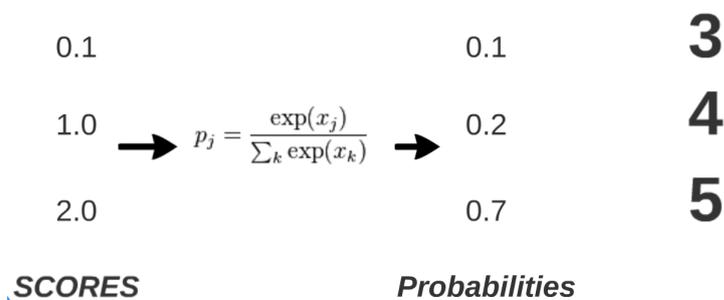
**Figura 2**  
*Linear Function*

## Linear Model



**Figura 3**  
*To turn logits into probabilities, the activation function was chosen to be softmax*

## Activation Function: "Softmax"



y 0 para el resto de entradas. En el ejemplo anterior, el cinco es la etiqueta correcta, por lo que el vector codificado en caliente es [0,0,1].

Ahora hay dos vectores, uno del clasificador (las probabilidades) y otro que representa la etiqueta correcta (vector codificado).

## 2.2 Aprendizaje profundo

### 2.2.1. Perceptrón multicapa [MLP]

Para convertir el clasificador logístico en una red, se conecta una segunda neurona entre la neurona actual y la entrada (Fig 4). Esto se denomina red neuronal de dos capas (la capa de entrada no se cuenta).

Las capas son el componente básico de más alto nivel de una red. La nueva capa se llama Capa Oculta porque sus valores de salida no son visibles para la salida de la red. La capa oculta le da al modelo la oportunidad de representar los datos de una forma más sencilla.

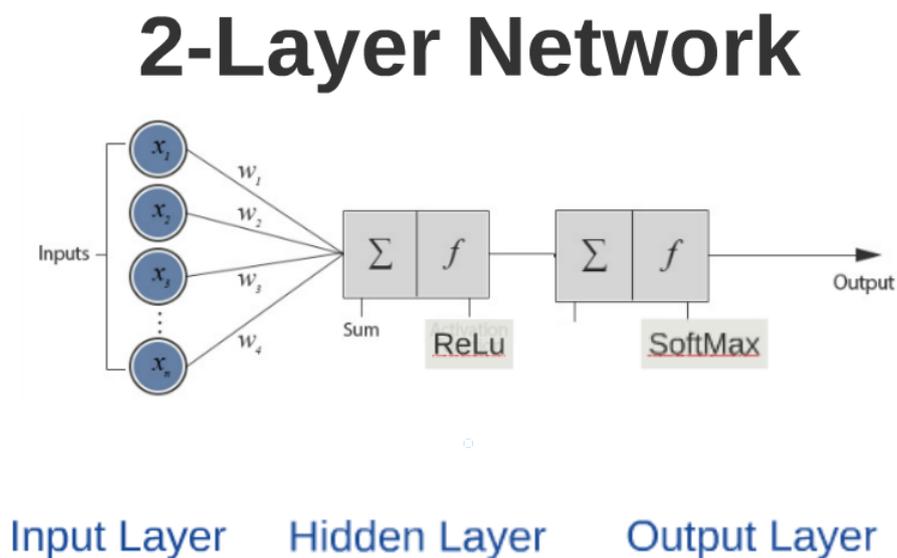
La profundidad de la red está definida por el número de capas ocultas.

In addition to layers, the number of nodes per layer can increase as well. The number of nodes on a layer represents the degree of freedom of that layer.

When the output of every node on one layer is connected to the input of every node on the next layer, the network is called *Fully Connected* or *Dense*.

The size of a network is defined by the number of layers and the number of nodes or parameters. Fig 5a has 2 layers,  $4+2=6$  nodes (do not count input) or  $[3 \times 4] + [4 \times 2] = 20$  weights and  $4+2=6$  biases for a total of 26 parameters. Fig 5b

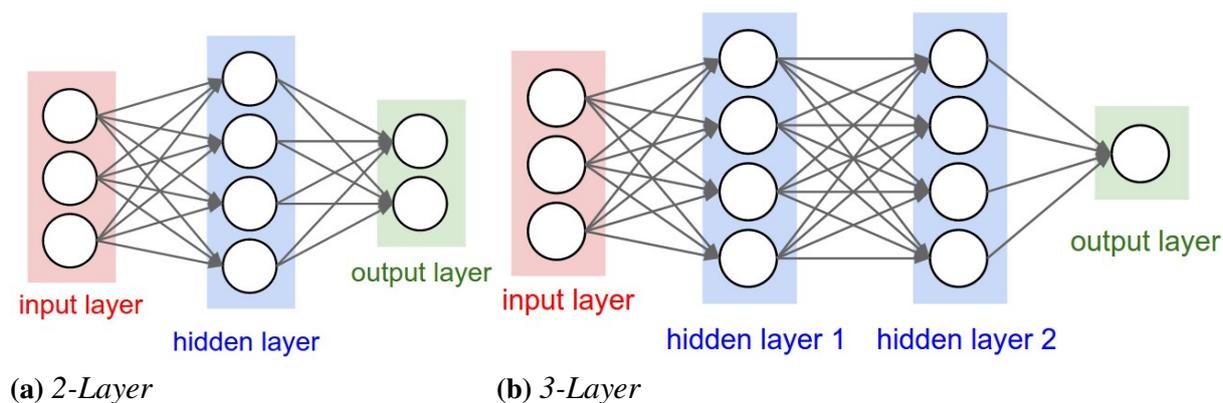
**Figura 4**  
Basic two layer Neural Network



has 3 layers, 9 nodes, and 42 learnable parameters.

Modern convolutional neural networks have 100 million parameters and 20 layers (hence deep learning).

**Figura 5**  
Dos redes neuronales totalmente conectadas (Li y Karpathy, 2016)



Evolving the structure from a single node into a network has allowed the

model more opportunities to represent the data in a simpler way (layers) and more degrees of freedom (nodes). However, the model is still linear. Because of superposition, stacking a 100 purely linear transformations can be simplified to a single layer. The solution is to introduce non-linear functions.

### 2.2.2. Non-Linearities

To preserve the network's structure (and the benefits gained with this structure), each hidden layer is given a non-linear activation function. By adding non-linearity, the entire model is now non-linear and cannot be simplified down to a single transformation. This creates a hierarchy of abstraction that grows in complexity with every layer. Dettmers (2015a) Mellina (2016)

This is the foundation for building deep models.

**Table 1**  
*Results from Classifying MNIST*

	Epochs	time per Epoch [s]	Accuracy		
			Training Set	Validation Set	Test Set
MLP	10	10	0.9822	0.9828	0.9812
CNN	5	360	0.9928	0.9915	0.9907

**Tabla 2***Tabla comparativo de resultados*

Items	Unidades				
	Unidad 1	unidad 2	unidad 3	unidad 4	unidad 5
Item A	valor	valor	valor	valor	valor
Item B	valor	valor	valor	valor	valor

## **2.3 Marco conceptual (palabras clave)**

### **2.3.1. Tablas**

## **2.4 Antecedentes empíricos de la investigación (estado del arte)**

## **2.5 Hipótesis \***

### **2.5.1. Hipótesis general \***

### **2.5.2. Hipótesis específicas \***

## **2.6 Identificación de variables e indicadores \***

## **2.7 Operacionalización de variables \***



## **Capítulo III**

### **METODOLOGÍA**

**3.1 Ámbito de estudio: localización política y geográfica**

**3.2 Tipo y nivel de investigación**

**3.3 Unidad de análisis**

**3.4 Población de estudio**

**3.5 Tamaño de muestra**

**3.6 Técnicas de selección de muestra**

**3.7 Técnicas de recolección de información**

**3.8 Técnicas de análisis e interpretación de la información**

**3.9 Técnicas para demostrarla verdad o falsedad de las hipótesis  
planteadas**

aquí va una cita Mellit y Kalogirou (2008)

## **Capítulo IV**

### **RESULTADOS Y DISCUSIÓN**

**4.1 Procesamiento, análisis, interpretación y discusión de resultados**

**4.2 Pruebas de hipótesis \***

**4.3 Presentación de resultados**

## **Capítulo V**

### **CONCLUSIONES Y RESULTADOS**

## Referencias Bibliograficas

Bakó, F., Judit, B., y Szigeti, C. (2021, 05). Households' electricity consumption in hungarian urban areas. *Energies*, 14, 2899. doi: 10.3390/en14102899

Dettmers, T. (2015a). *Deep learning in a nutshell: Core concepts*. Descargado 2024-30-06, de <https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-core-concepts/feature-engineering>

Dettmers, T. (2015b). *Deep learning in a nutshell: History and training*. Descargado 2024-30-06, de <https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-history-training/>

Krizhevsky, A., Hinton, G., y cols. (2010). Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7), 1–9.

Li, F.-F., y Karpathy, A. (2016). *Cs231n: Convolutional neural networks for visual recognition*. Descargado 2024-30-06, de <http://cs231n.stanford.edu/>

Markoff, J. (2012). *Scientists see promise in deep learning programs*. Descargado 2024-30-06, de <http://www.nytimes.com/2012/11/24/science/scientists-see-advances-in-deep-learning-a-part-of-artificial-intelligence.html>

Mellina, C. (2016). *Deep learning workshop*.

Mellit, A., y Kalogirou, S. A. (2008). Artificial intelligence techniques for photovoltaic applications: A review. *Progress in energy and combustion science*, 34(5), 574–632.

## **ANEXOS**

### ANEXO 1

### Matriz de consistencia

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES E INDICADORES	METODOLOGÍA
PROBLEMA GENERAL	OBJETIVO GENERAL	HIPÓTESIS GENERAL	VARIABLE INDEPENDIENTE	TIPO DE INVESTIGACIÓN
<ul style="list-style-type: none"> <li>■ Objetivo 1</li> <li>■ Objetivo 2</li> </ul>				
PROBLEMA ESPECÍFICOS	OBJETIVOS ESPECÍFICOS	HIPÓTESIS ESPECÍFICA	VARIABLE DEPENDIENTE	DISEÑO DE LA INVESTIGACIÓN
<ul style="list-style-type: none"> <li>■ Problema 1</li> <li>■ Problema 2</li> </ul>				

## Anexo 2

### Codigo Utilizado

```
1 n = 10000
2 fibo = [0]*(n+1)
3 fibo[1]=1
4 fibo[2]=1
5
6 def fibonacci(n):
7     if fibo[n] != 0:
8         return fibo[n]
9     fibo[n] = fibo[n-1] + fibo[n-2]
10    return fibo[n]
11
12 for i in range(1,n+1):
13     fibonacci(i)
14
15 def count_digits(n):
16     n = str(n)
17     return len(n)
18
19 i = 2
20 while True:
21     if count_digits(fibo[i])>=1000:
22         break
23     i+=1
24 print(i)
```