Parcial #1

Computacion Blanda - UTP
Nicolas Amaya
Cristian Rodriguez
Victor Betancourth

Lista de tareas

Tarea 1

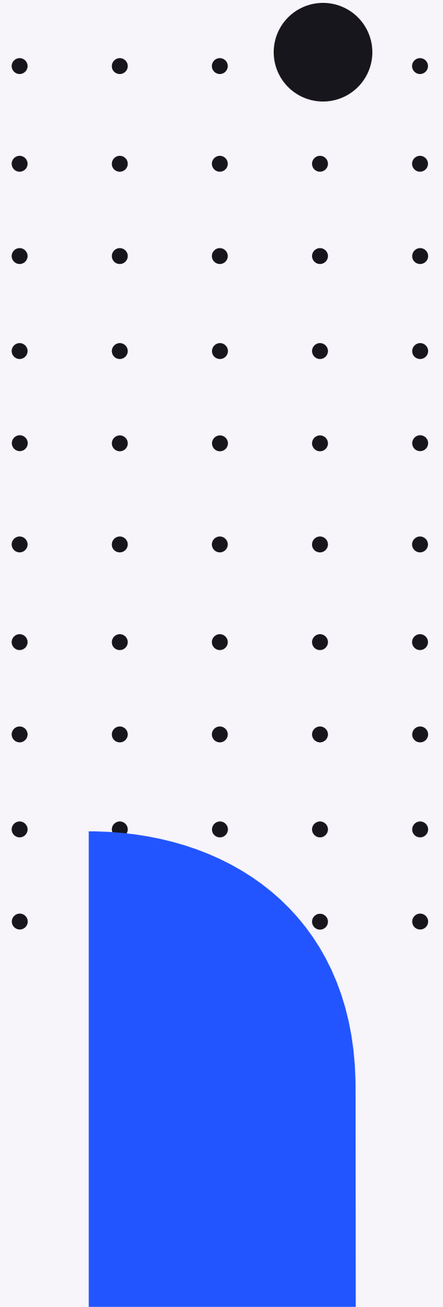
Realizar un paper sobre el perceptron y propagación hacia atras

Tarea 2

Aplicacion de la logica difusa en un problema

Tarea 3

Sustentacion



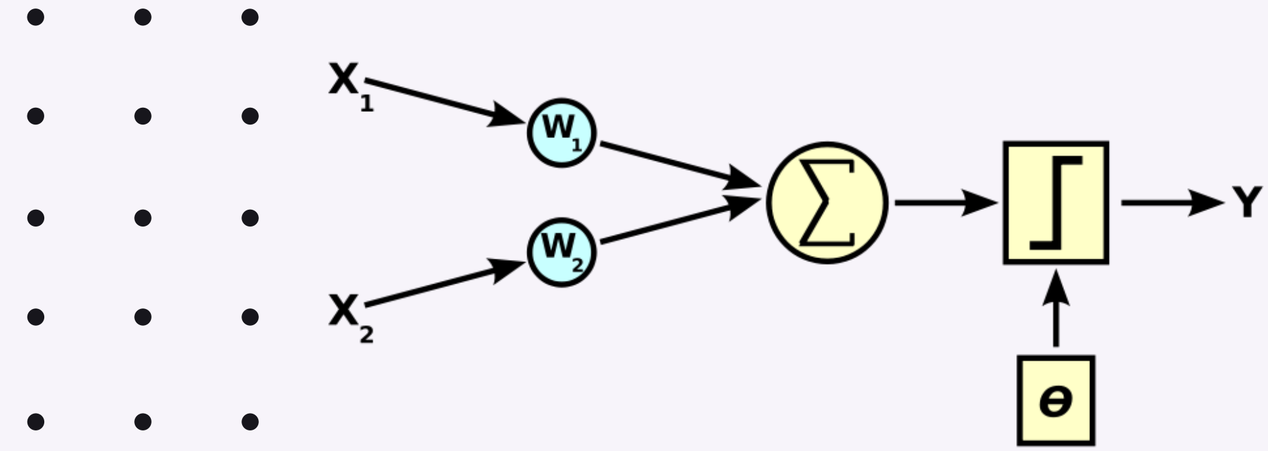
Paper

Como primer paso lo que realizamos fue el paper donde expresamos de manera clara y concisa el funcionamiento de un perceptron simple, un perceptrón multicapa, y finalmente el algoritmo de propagación hacia atrás

Un pequeño vistazo

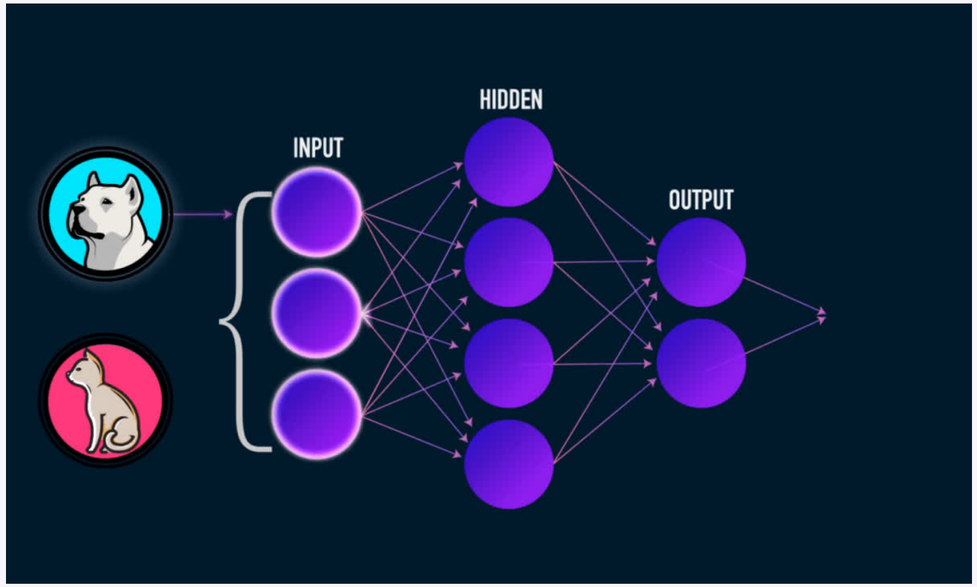
Perceptron

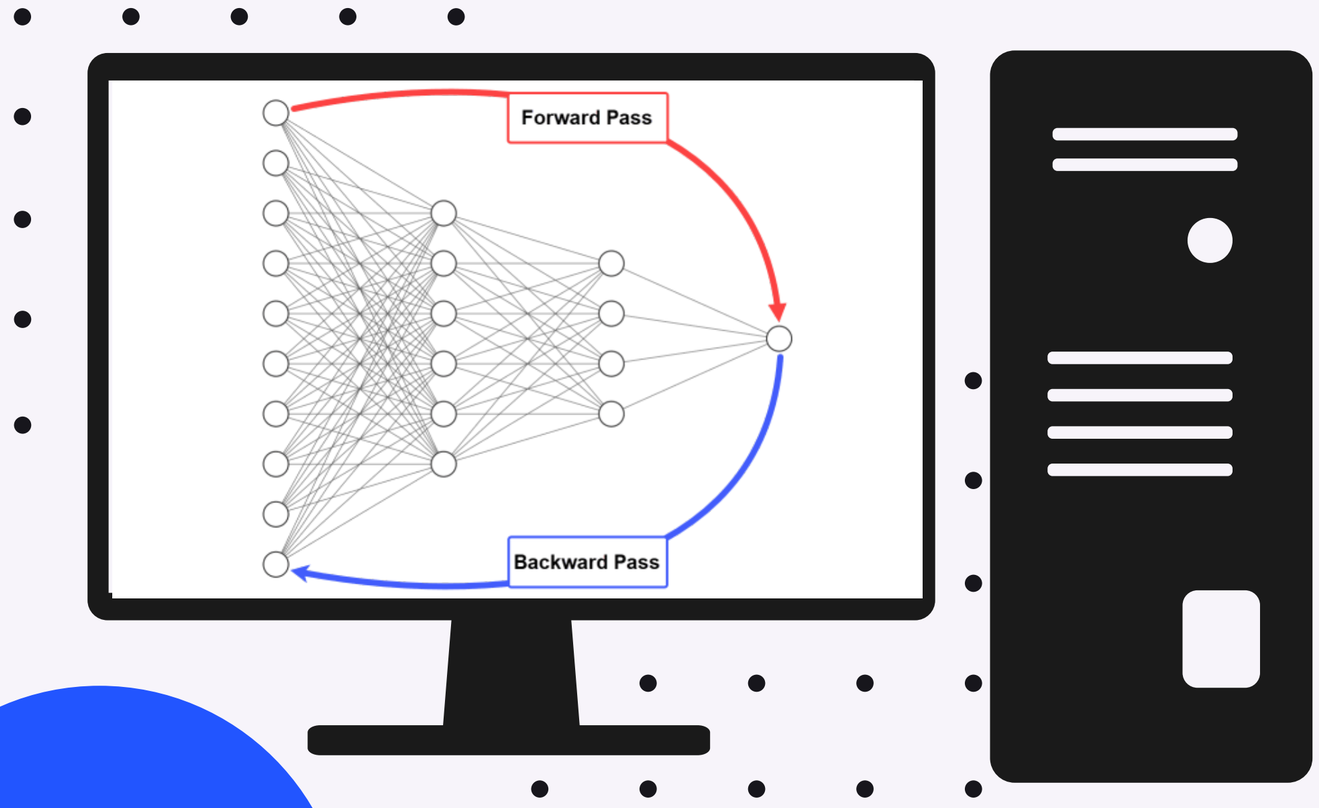
el perceptrón simple es la red neuronal artificial más antigua. consiste en un clasificador binario o discriminador lineal, esto quiere decir que a partir de un entrenamiento con datos el perceptrón es capaz de reconocer patrones y tomar decisiones.



Perceptron Multicapa

El perceptrón multicapa es una red neuronal artificial (RNA) formada por múltiples capas, de tal manera que tiene capacidad para resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón





BACKPROPAGATION

El algoritmo de backpropagation nos indica cuanto de culpa tiene cada neurona del error global cometido.

La forma en que como se calcula la culpa que tiene cada neurona en el error es lo que da sentido al nombre de backpropagation, ya que primeramente calcula la culpa del error de cada neurona de la última capa y lo va propagando hacia atrás para ver cuanta culta tienen el resto.
Se podría decir que pondera el reparto del error para cada una de las neuronas de la red.

El algoritmo de backpropagation determina la culpa del error, calculando las derivadas parciales de la función de coste con respecto a cada una de las variables.

Aplicaciones de la tematica

Aplicación 1

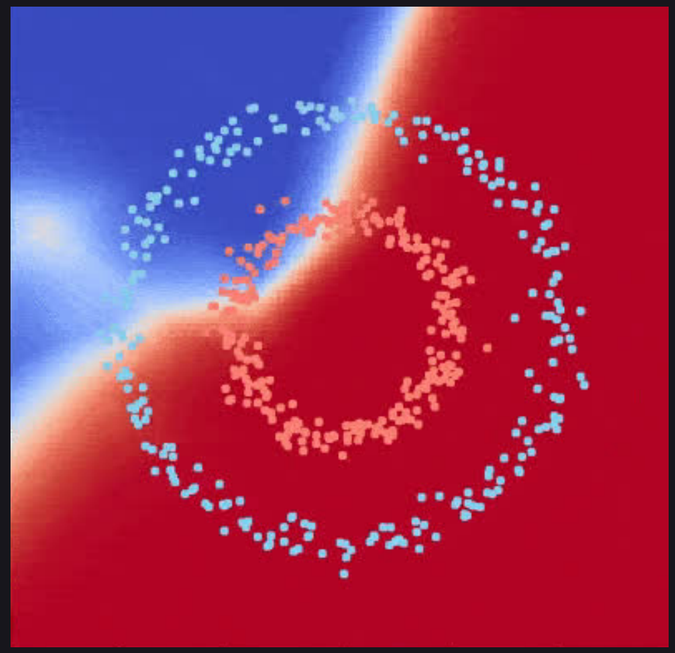
- Predicción de sucesos y simulaciones: Producción de los valores de salida esperados en función de los datos entrantes.

Aplicación 2

- Reconocimiento y clasificación: Asociación de patrones y organización de conjuntos de datos en clases predefinidas.

Aplicación 3

- Procesamiento de datos y modelización: Validación, agregación y análisis de datos. Diseño y búsqueda de fallos en sistemas de software complejos.



Complemento al paper

Implementamos una red neuronal paso a paso sin libreria utilizando lo visto en el paper. Luego, desarrollamos otra red neuronal pero esta vez utilizando una libreria.

- **Programación completa de la red neural**

  Dificultad: Alta

- **Uso de la libreria de Tensorflow**

  Dificultad: Media

implementación red neuronal

Se implemento una red neuronal de clasificación de dos grupos. Esta se entrena con una entrada aleatoria de dos circulos. Esta implementa tres pasos principales de entrenamientos. El primero "Un paso hacia adelante" utilizando la función sigmoide. El segundo paso, es evaluar el error mediante una función de coste. Finalmente, se hace una "Propagación hacia atras" para autoajustas los paramentros del problema (peso y bias)

1

Evaluar

Se realiza una propagación hacia adelante y se encuentra el error

2

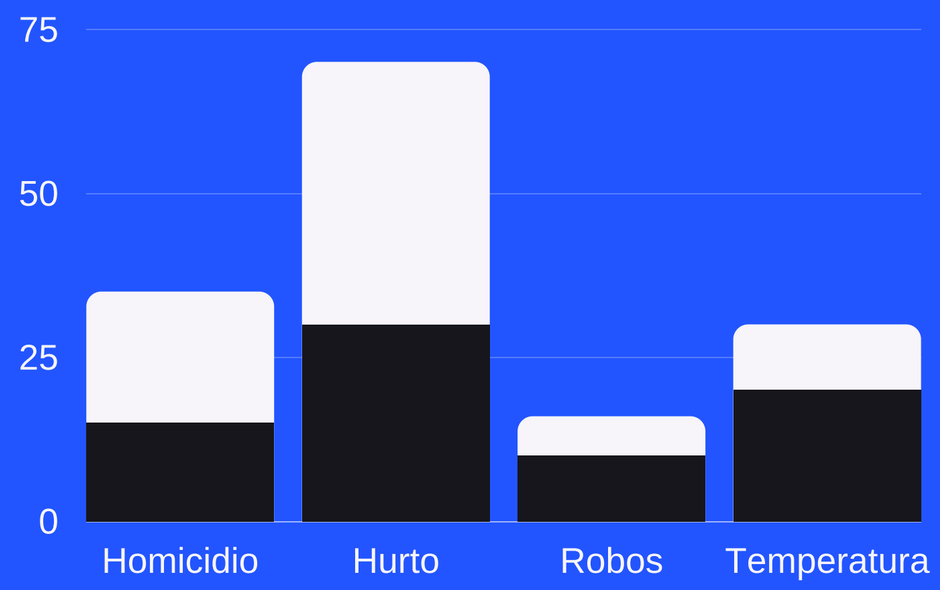
Calcular

Se aplica la derivada de la funcion de coste para encontrar la variación de los parametros

3

Actualizar

Se realiza el ajuste de los parametros



Lógica difusa

Realizamos nuestra propia implementación sobre lógica difusa basandonos en *"Programa_40_control_difusion_API"* proporcionado por el docente

Criminalistica

Mediante lógica difusa hicimos una clasificación por: hurto, robo, homicidio y también la temperatura como factor influyente para determinar el indice de criminalidad.
Para ello, realizamos una aplicación del algoritmo con datos extraidos de una excel con fin de simular el mundo real, en esta situación especifica Pereira.

Comentarios

Algoritmo lógica difusa

Para ejecutar correctamente el algoritmo de lógica difusa deberá subir el excel que contiene los datos en colab

Presentación

si desea visualizar la presentación de manera interactiva dirigir al enlace que esta en la descripción del repositorio de github

Red neural

El algoritmo donde implementamos se encuentra en el github