

Sistema de Predicción de Tráfico

Descripción del Problema

- Objetivo: Determinar la ruta más óptima hacia universidades en Lima Metropolitana.
- Consideraciones:
 - Día
 - Hora del día
 - Nivel de tráfico en tiempo real
 - Condiciones de tránsito

FATE Framework

- Fairness: Trato imparcial hacia los upecinos.
- Accountability: Auditorías regulares y documentación completa.
- Transparency: Explicación clara del proceso de toma de decisiones.
- Ethics: Garantiza equidad en acceso a información de congestión.

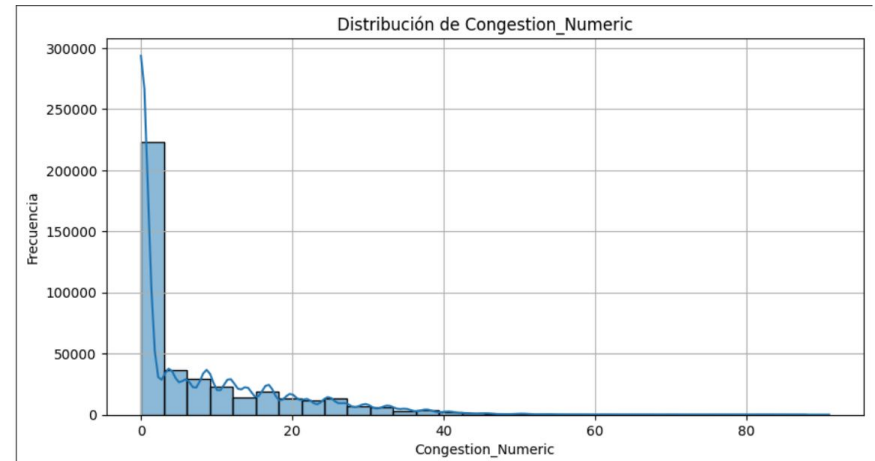
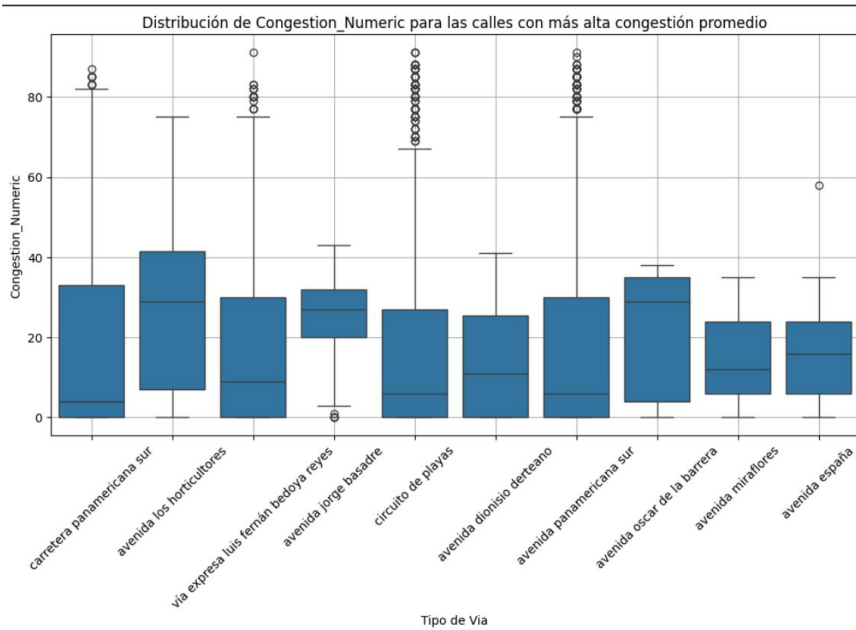
Dataset y Herramientas

- API MapBox: 500,000 consultas disponibles.
- Datos principales:
 - Dia, Hora, Coordenadas de la calle, Nombre de la calle
 - Congestión numérica (0-100)

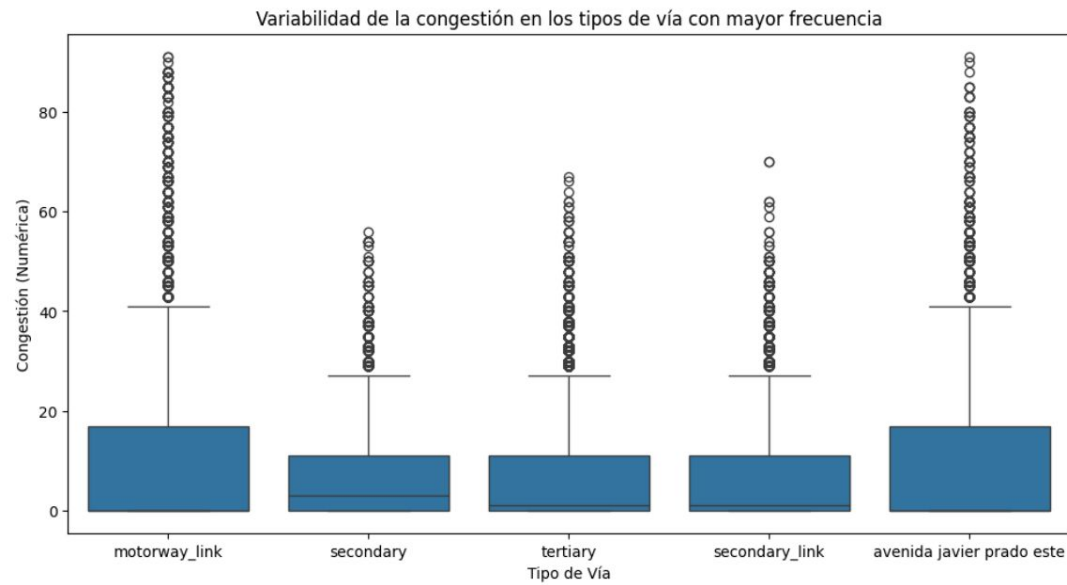
Entrenamiento del Modelo

- Inputs:
 - Hora del día
 - Día de la semana
 - Ubicación (Longitud y Latitud)
 - Calle específica (ex: Javier Prado Este)
- Output:
 - Predicción de congestión Numerica(0-100)

EDA

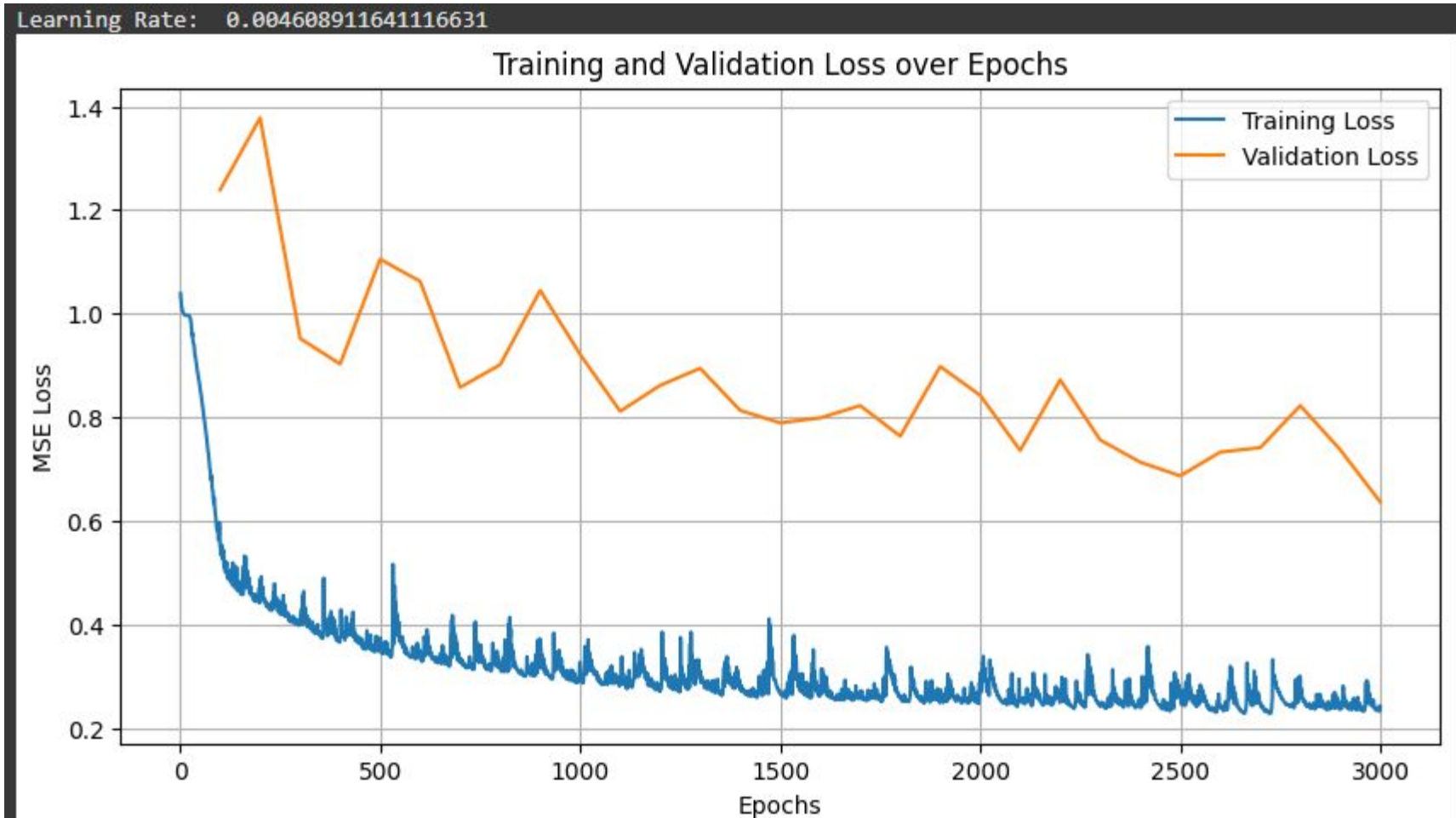


EDA

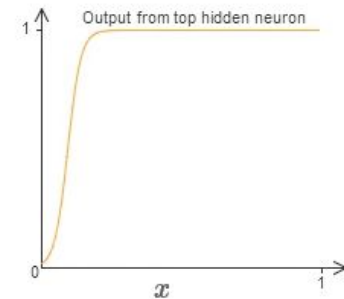
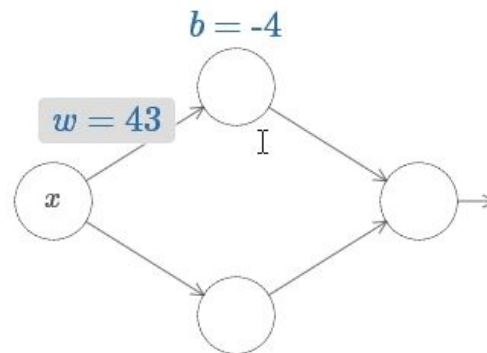
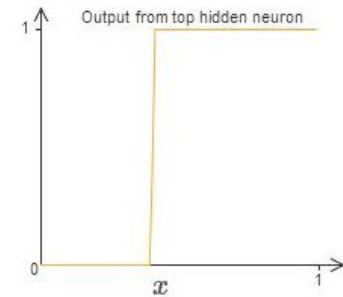
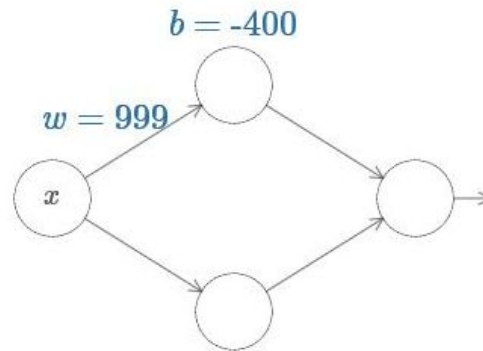
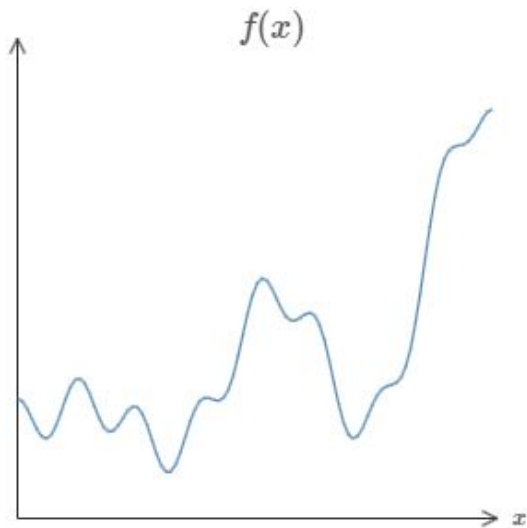


Resultados del Entrenamiento

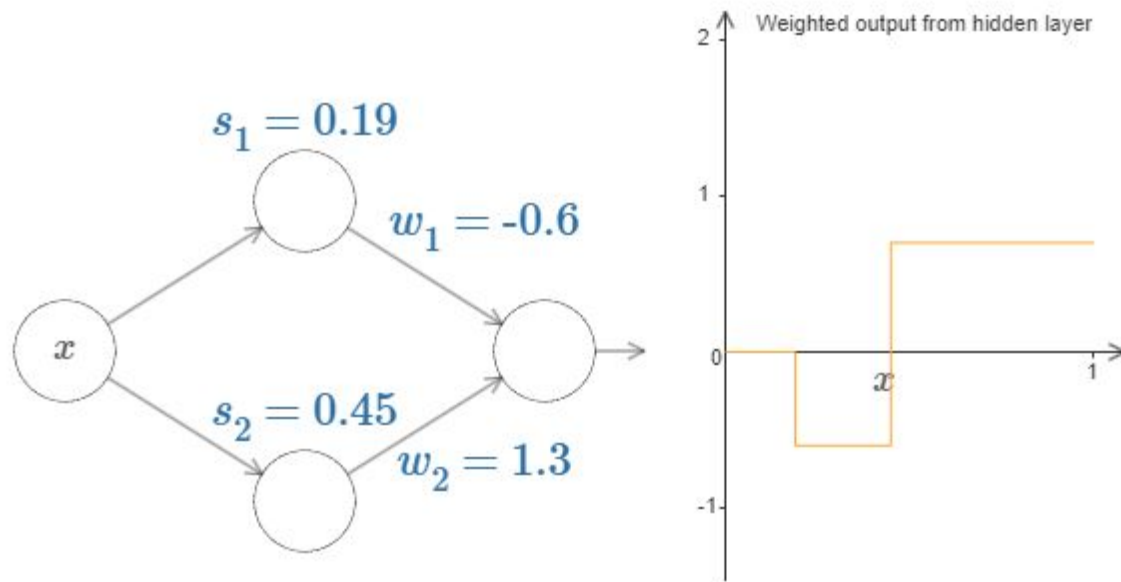
- Mejores hiper parámetros:
- - Learning Rate: 0.004608
- - Capas ocultas: 14
- - Neuronas por capa: 26
- Error de validación: 0.63



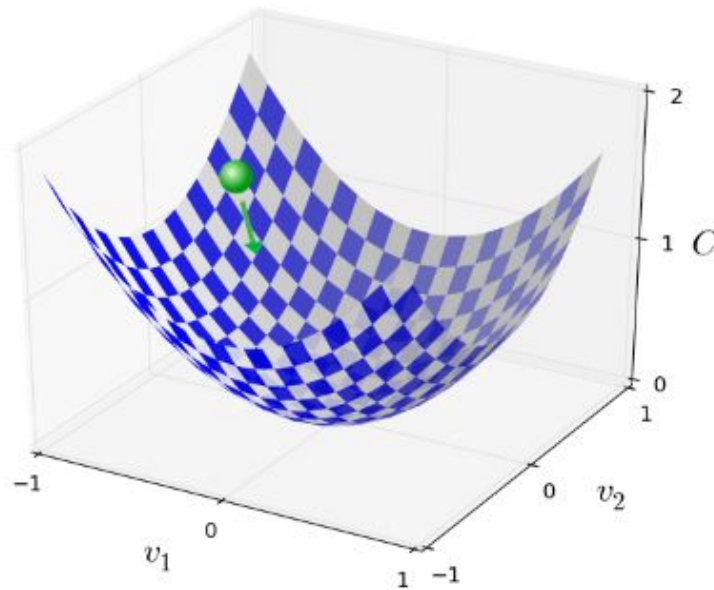
Representación Matemática



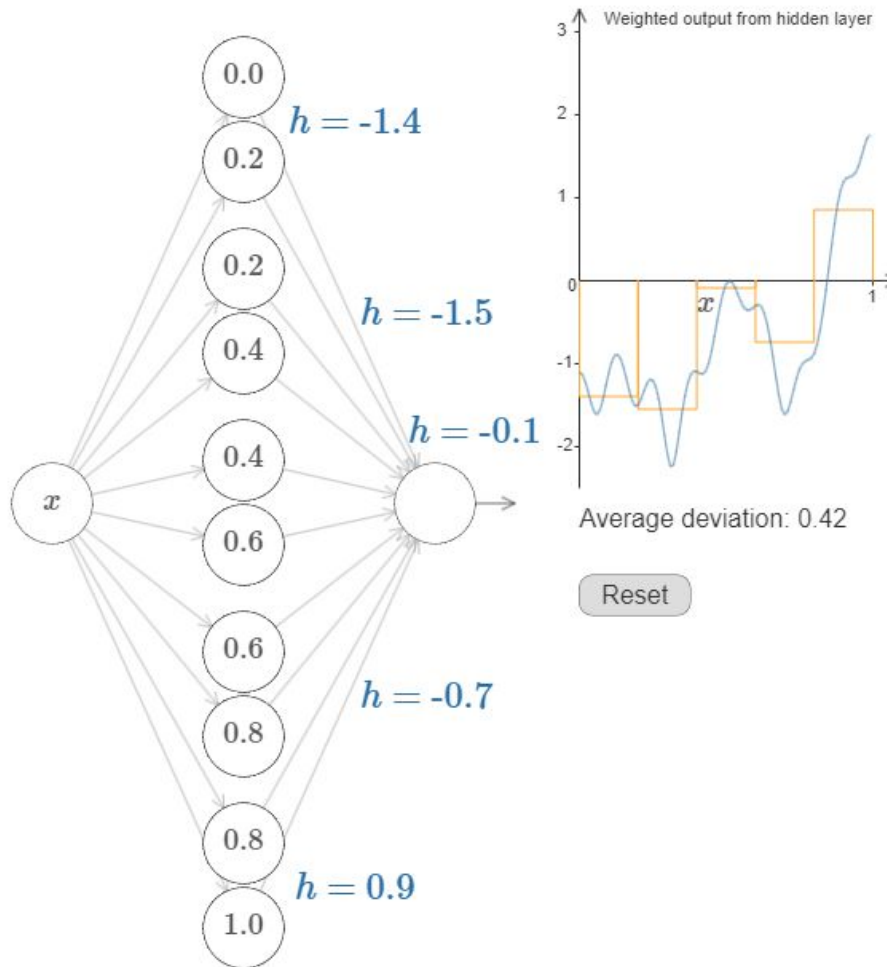
Representación Matemática



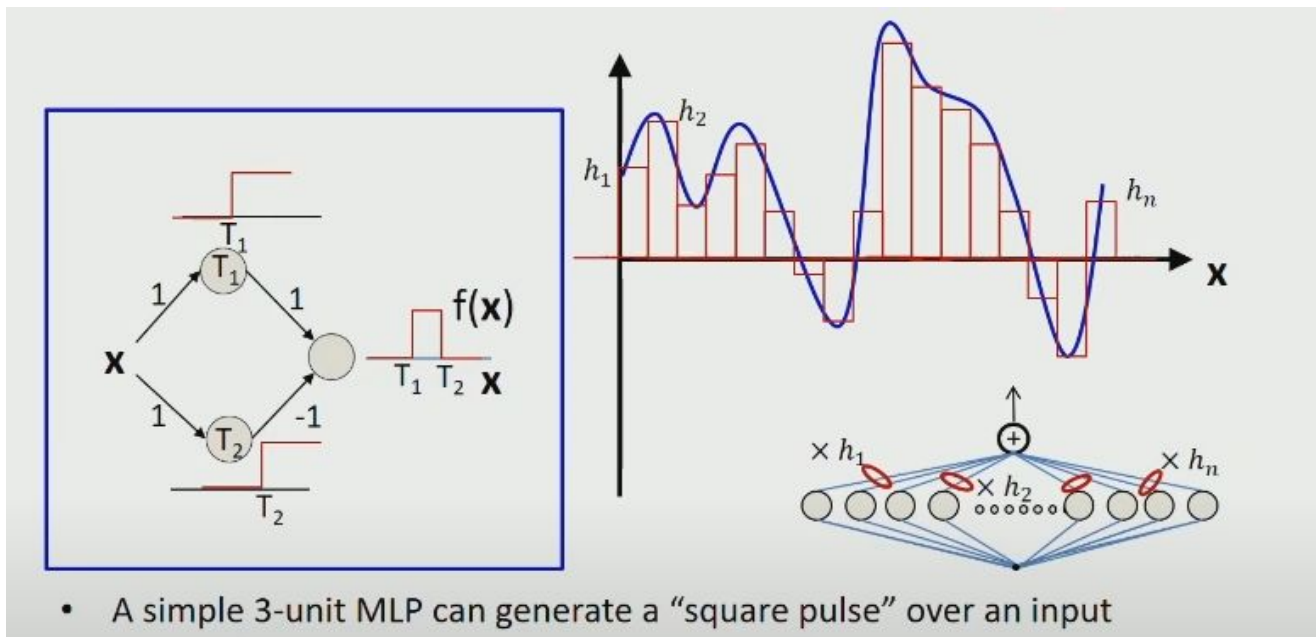
Representación Matemática



Representación Matemática



Representación Matemática



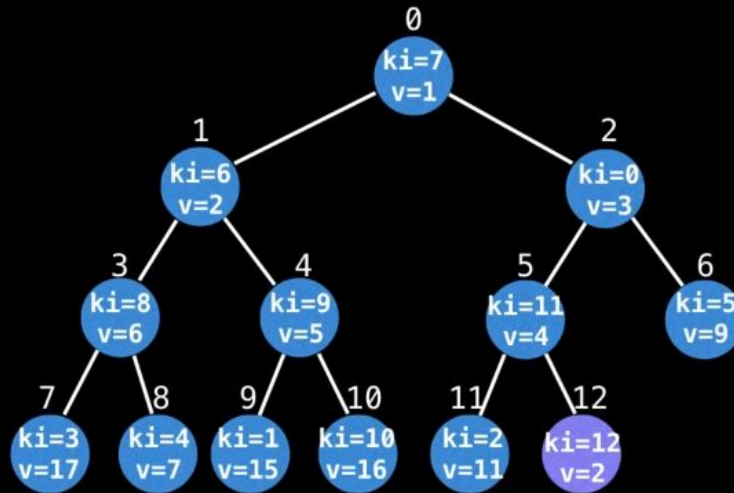
Index Priority Queue

IPQ as a binary heap

Operation	Indexed Binary Heap PQ
<code>delete(ki)</code>	$O(\log(n))$
<code>valueOf(ki)</code>	$O(1)$
<code>contains(ki)</code>	$O(1)$
<code>peekMinKeyIndex()</code>	$O(1)$
<code>pollMinKeyIndex()</code>	$O(\log(n))$
<code>peekMinValue()</code>	$O(1)$
<code>pollMinValue()</code>	$O(\log(n))$
<code>insert(ki, value)</code>	$O(\log(n))$
<code>update(ki, value)</code>	$O(\log(n))$
<code>decreaseKey(ki, value)</code>	$O(\log(n))$
<code>increaseKey(ki, value)</code>	$O(\log(n))$

Index Priority Queue

Insertion



Name	ki
Anna	0
Bella	1
Carly	2
Dylan	3
Emily	4
Fred	5
George	6
Henry	7
Isaac	8
James	9
Kelly	10
Laura	11
Mary	12

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
vals	3	15	11	17	7	9	2	1	6	5	16	4	2	-1	-1
pm	2	9	11	7	8	6	1	0	3	4	10	5	12	-1	-1
im	7	6	0	8	9	11	5	3	4	1	10	2	12	-1	-1

The heap invariant is not satisfied ATM since node at index 12 has a value less than node 5.

Arquitectura Técnica

- Red Neuronal:
 - Función de activación: ReLU
 - Optimizador: Adam
 - Inicialización de pesos: Distribución uniforme
- Algoritmos:
 - Redes Neuronales Artificiales
 - SMA* con Index Priority Queue

Ventajas del Sistema

- Predicción en tiempo real de la congestión de las principales avenidas para llegar de una sede a otra sede de la UPC.
- Optimización de rutas con SMA*.
- Mejores tiempos de respuesta con Index Priority Queue.
- Eficiencia en consumo de memoria con SMA* y Index Priority Queue (Actualización por Indexación).

Limitaciones

- Solo se usa las avenidas principales para llegar de una sede a otra.
- Costo computacional alto para poder predecir la congestión en más calles.
- Nuestro sistema no tiene GPS.
- Solo se considera las sedes de UPC.
- Se desconoce la congestión de algunas zonas

Validación de resultados y pruebas

Rutas UPC

Encuentra la mejor ruta entre dos sedes de la UPC :D

Punto de inicio

UPC Monterrico

Punto final

UPC San Miguel

Hora de salida

08/11/2024 19:01

Información de la ruta:

Distancia total: 16.49 km
Tiempo estimado: 151 minutos
Hora de salida: 19:01
Hora estimada de llegada: 21:32

Calcular Ruta

Sedes UPC disponibles:

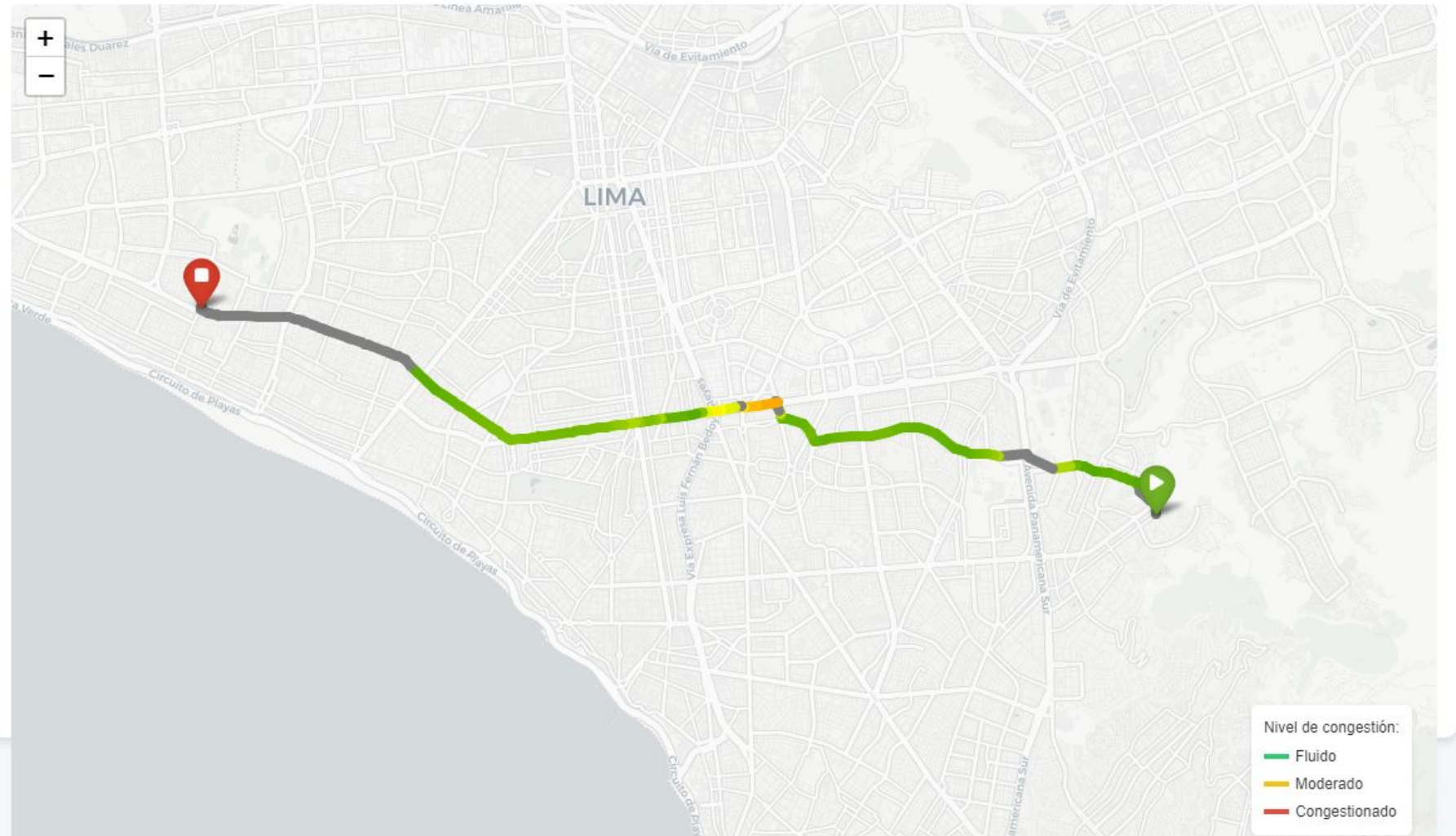
UPC San Isidro

UPC San Miguel

UPC Monterrico

UPC Villa

Validación de resultados y pruebas



Mapa de la zona de estudio en Bogotá, Colombia, mostrando la Avenida Javier Prado Este y la Avenida Nicolás Arriola. Se indica el nivel de congestión: Fluido (verde), Moderado (amarillo) y Congestionado (rojo). La congestión moderada se observa en la Avenida Javier Prado Este, especialmente en el tramo que cruza la Avenida Nicolás Arriola.

Conclusiones

Conclusiones

Esta aplicación permitirá a los estudiantes planear sus rutas de forma eficiente. Con una predicción basada en datos históricos los estudiantes tienen la posibilidad de planificar con anticipación, de esta forma evitarán contratiempos causados por el tráfico.

Este proyecto demuestra que la integración de datos históricos, combinada con algoritmos avanzados como redes neuronales y SMA*, puede ofrecer soluciones eficientes a problemas cotidianos como el tráfico en ciudades complejas.

Al momento de planificar la arquitectura es importante investigar acerca de otras arquitecturas creadas para una solución similar y leer acerca del desempeño de los modelos como una guía.

La integración de herramientas de planificación, como el uso de la API de MapBox, destaca la importancia de las plataformas abiertas y colaborativas para agilizar el desarrollo de proyectos tecnológicos en entornos urbanos.