ENERGY EFFICIENCY

Utilizando IA para determinar la carga de calefacción y refrigeración (eficiencia energética) según los parametros de un edificio.

Integrantes

Pablo Cruz - 21000116
David Blanck - 21000617
Julio Del Valle-21000492

Escanéame

INTRODUCCIÓN

La inteligencia artificial y Machine Learning son una nueva forma de resolver problemas, en el siglo 21, si se involucra suficiente matemática.

La Eficiencia energética se enfoca en la cantidad de carga calorífica que necesita un edificio en base a sus características físicas, es decir cuanta calefacción o refrigeración necesita. El atractivo de este problema es en base al uso de inteligencia artificial al cuantificar y encapsular todas estas características.

Este trabajo consiste en una solución para la eficiencia energética de los edificios, utilizando Machine learning. Nuestra solución busca resolver un problema de regresión, dónde determinamos 2 valores importantes.

DATA SET

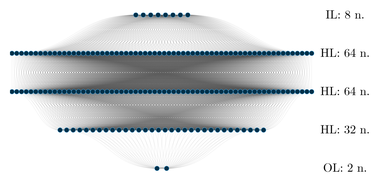
El dataset fue creado por Angeliki Xifara y Atanasios Tsanas y este se obtuvo de 12 formas de edificios diferentes simulados en Ecotec, en los cuales se modificaron distintos parámetros de los edificios hasta obtener 768 formas de edificios. Por esa razon el dataset se conforma de 768 instancias y 8 features, además de dos variables dependientes. A continuación se muestran las todas variables con su respectiva descripción.

Variable	Descripción
X1	Relación volumen/área total de la edificación.
X2	Total de superficies exteriores.
X3	Superficies verticales exteriores.
X4	Superficie superior de la edificación.
X5	Distancia vertical máxima.
X6	Dirección cardinal de la edificación.
X7	Superficies de vidrio.
X8	Distribución de superficies de vidrio.
Y1	Energía para calentar.
Y2	Energía para enfriar.

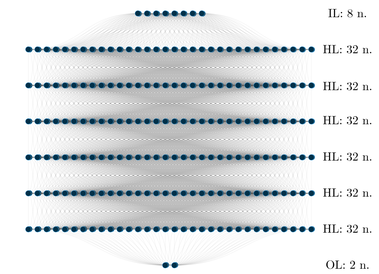
MODELOS

IL: Input Layer
HL: Hidden Layer
OL: Output Layer

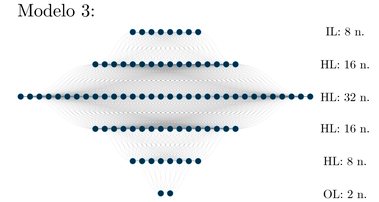
Modelo 1:



Modelo 2:



Modelo 3:



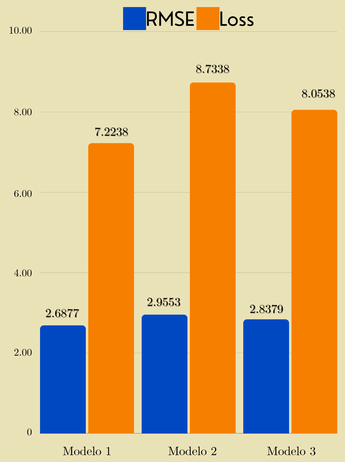
RESULTADOS

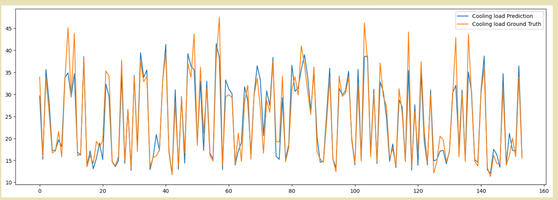
Tras ejecutar los 3 modelos planteados, se obtuvo que el modelo con mejores resultados fue el M1, el cual tenia menos capas neuronales que los otros, mientras que si observamos los demás modelos entre más capas tenían, peores eran los resultados obtenidos de Loss y RMSE, ya que el modelo 2 tuvo los peores resultados y era el que más capas neuronales tenía.

El hecho de que los modelos M2 y M3 no sean tan buenos como el M1 se puede deber a que se estaba dando un caso de overfitting debido a que los modelos se volvieron demasiado complejos al tener muchas capas neuronales.

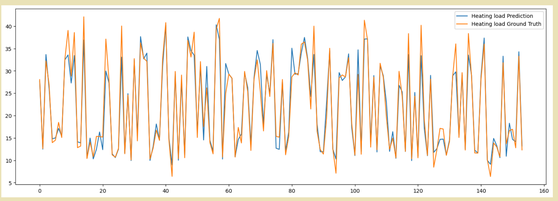
Modelo	Batch Size	Hidden Layers	Loss	RMSE
M1	64	3	7.2238	2.6877
M2	64	6	8.7338	2.9553
M3	16	4	8.0538	2.8379

Tabla de resultados por modelo.





Predicción vs Valor Real de Cooling Load (Y1) para el Modelo 1



Predicción vs Valor Real de Heating Load (Y2) para el Modelo 1

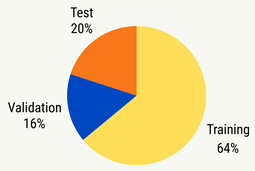
METODOLOGÍA

Para resolver este problema de Regresión se diseñaron 3 arquitecturas de redes neuronales para elegir el mejor de los modelos observando las métricas de validación. Cada una de estas redes neuronales tiene un diseño densamente conectado, y están hechas en Python, que ofrece herramientas optimizadas para el Machine Learning. Las librerías utilizadas son TensorFlow, Pandas, Keras, SkLearn. Para la visualización de los datos se utilizó matplotlib, seaborn, entre otras librerías básicas como numpy.

Pasos previos relacionados con el Dataset: antes de de entrenar nuestro modelo, es importante tomar en cuenta cuestiones de normalización de los datos, para así tener los mejores resultados posibles. En este caso optamos por un pre-porcesamiento de los datos sencillo ya que la información proporcionada por el dataset es puntual y no requiere mayor manipulación; se hizo una identificación y separación de las variables explicativas y explicadas, y una normalización de maximos y minimos para cada una de las variables explicativas. Luego, se separaron los datos en un set de entrenamiento del 80% de datos y un set de testeo con el 20% restante. Total de datos: 768; set de entrenamiento: 614; set de testeo: 154. Y del set de entrenamiento tomamos el 20% para el set de validación; entonces, de los 614 de entrenamiento, 123 se usan para validación, que es el 16% del total de datos.

Los modelos: Cada uno de los modelos evaluados cuenta con una input layer de 8 neuronas, ya que contamos con 8 features y una output layer de 2 neuronas para cada variable dependiente. En donde ocurre la distinción es en las capas densas de cada modelo.

- Modelo 1:
  - Capas densas: 3 capas, de 64, 64 y 32 neuronas respectivamente, con función de activación relu. Se puso a entrenar con límite de 150 epochs pero se detuvo en 68 por el EarlyStopping Callback.
- Modelo 2:
  - Capas densas: 6 capas, de 32 neuronas cada una, con función de activación relu. Se puso a entrenar con límite de 150 epochs pero se detuvo en 53 por el EarlyStopping Callback.
- Modelo 3:
  - Capas densas: 4 capas, de 16, 32, 16 y 8 neuronas respectivamente, con función de activación LeakyRelu. Se puso a entrenar con límite de 150 epochs pero se detuvo en 42 por el EarlyStopping Callback.



MEJORAS A FUTURO

Dentro de este apartado tenemos la posibilidad de discutir aspectos a mejorar en este proyecto. Los principales aspectos que identificamos a mejorar son los siguientes:

- Problema: otra aproximación para resolver este problema es tratarlo como si fuera uno de clasificación, en lugar de uno de regresión. si tomamos esto en cuenta es otro factor que pudiera mejorar los resultados.
- Arquitectura: la arquitectura de los 3 modelos es parecida, en cuanto a diseño. para profundizar más en esto, es posible modificar hiperparámetros cómo el batch size, o trabajar con una red que no esté completamente conectada.

CONCLUSIONES

- Para este dataset específico, el mejor modelo fue el que tenía menos capas pero con una mayor cantidad de neuronas por capa.
- Es importante tener un EarlyStopping Callback para evitar el overfitting; gracias a este los 3 modelos se detuvieron antes del número de epochs especificado y tuvieron muy buenos resultados.
- Un dataset más extenso pudo ayudar a tener resultados más acertados, siendo 614 datos muy pocos para entrenamiento del modelo.