

1. Introducción

En el mundo actual, el cambio climático y consumo energético en edificaciones es un tema que ha tomado relevacia debido a la concientización que como sociedad hemos tomado. Por lo tanto, es importante desarrollar herramientas que optimicen el uso de recursos en estas nuevas edificaciones. Y como equipo creemos firmemente que el análisis de eficiencia energética mediante modelos predictivos, detectando patrones de edificios ya existentes, puede contribuir significativamente a la reducción del impacto ambiental.



En este proyecto, buscamos aplicar técnicas de *Inteligencia Artificial* para predecir el consumo energético de edificios enfocándonos en la carga térmica de la calefacción y refrigeración. Haciendo uso de estas predicciones, podremos analizar la eficiencia energética en un edificio en base a varias características de la construcción, y con ello se podría obtener información sobre el cosumo energético de edificios existentes y facilitar el diseño de edificios nuevos que sean más sostenibles.

2. Descripción del data set

- **Nombre.** Energy Efficiency
- **Fuente.** UCI Machine Learning Repository

Para la recopilación de los datos de este dataset, se simularon usando el software *Ecotect*, y con ello se crearon 768 configuraciones distintas de edificios utilizando variables como el área acristalada, la distribución de los cristales, la orientación el edificio, la altura, entre otros parámetros.

Dentro de las características del dataset podemos encontrar que consta de:

- **768 muestras**
- **8 Variables Explicativas** (variables tienen algún efecto en el resultado)
 - **Relative Compactness.** (Compacidad del edificio)
 - **Surface Area.** (Área superficial)
 - **Wall Area.** (Área de paredes)
 - **Roof Area.** (Área del techo)
 - **Overall Height.** (Altura total)
 - **Orientation.** (Orientación del edificio)
 - **Glazing Area.** (Porcentaje de acristalamiento)
 - **Glazing Area Distribution.** (Distribución del acristalamiento)
- **2 Variables Explicadas** (resultado final que queremos entender)
 - **Heating Load.** (Carga térmica de calefacción)
 - **Cooling Load.** (Carga térmica de refrigeración)

Variable	Mínimo	Máximo	Promedio	Desviación Estándar
Relative Compactness	0.62	0.98	0.76	0.11
Surface Area	514.5	808.5	671.7	88.1
Wall Area	245.0	416.5	318.5	43.6
Roof Area	110.2	220.5	176.6	45.2
Overall Height	3.5	7.0	5.25	1.75
Orientation	2	5	3.5	1.1
Glazing Area	0.0	0.4	0.23	0.13
Glazing Area Distribution	0	5	2.8	1.5
Heating Load	6.0	43.1	22.3	10.1
Cooling Load	10.9	48.0	24.6	9.5

Table 1. Resumen estadístico de las variables del dataset Energy Efficiency

3. Metodología

Utilizamos el dataset *Energy Efficiency* del repositorio UCI. Primero, cargamos los datos de un archivo Excel, verificamos y exploramos estos datos con librerías como *pandas* y *seaborn*. Luego los dividimos en un conjunto de entrenamiento y prueba con una proporción 80/20 (80% de los datos para entrenamiento y 20% para prueba). Para la predicción de las cargas de calefacción y refrigeración, entrenamos una red neuronal utilizando *TensorFlow*. Normalizamos las variables explicativas para mejorar el rendimiento del modelo. Finalmente, evaluamos la precisión de las predicciones usando métricas de regresión como el error cuadrático medio y el coeficiente de determinación R^2

4. Resultados

Red	Capas	Activaciones	Métrica 1	Métrica 2
NN1	128 -> 64 -> 2	Ocultas: Relu. Salida: Linear	Loss: 0.006948	RMSE: 0.08335
NN2	256 -> 128 -> 64 -> 2	Ocultas: Relu. Salida: Linear	Loss: 0.0073	RMSE: 0.0853
NN3	1	Sin activación	MSE 10.1257	MAE: 2.4368

Durante el desarrollo del proyecto, uno de los principales problemas encontrados fue evitar el sobreajuste (*overfitting*), ya que los primeros entrenamientos mostraban una pérdida de validación que comenzaba a aumentar después de pocas épocas, mientras que el loss seguía disminuyendo. Esto indicaba que el modelo solo estaba memorizando los datos en lugar de generalizar correctamente (estaba teniendo *overfitting*).

Para corregir este problema, probamos varios conceptos distintos:

- Introdujimos capas de *Dropout* (con tasas del 20% y 30%) para reducir la dependencia en neuronas específicas.
- Usamos *Batch Normalization*, lo que permitió una mejor estabilidad y velocidad en el entrenamiento.
- Aplicamos la técnica *Early Stopping*, deteniendo automáticamente el entrenamiento si la pérdida de validación no mejoraba durante 10 épocas consecutivas.

Otro problema importante fue ajustar correctamente la arquitectura de las redes. Probamos varias configuraciones:

- Una versión más simple con una sola capa densa como *baseline*.
- Una arquitectura con dos capas ocultas y menos regularización.
- Y una arquitectura más compleja con más capas, *dropout* y normalización, que fue la que entregó los mejores resultados.

El razonamiento detrás de estos ajustes fue encontrar un equilibrio entre la capacidad del modelo para aprender patrones un poco complejos y su capacidad de generalizar datos. Comparamos las métricas de desempeño MSE, MAE y RMSE en los datos de validación y elegimos las tres mejores para el póster científico.

5. Conclusiones

Los modelos desarrollados lograron predecir de manera precisa las cargas térmicas (de calefacción y refrigeración) en función de los parámetros constructivos de los edificios. Las distintas arquitecturas exploradas permitieron comparar la influencia de la profundidad de la red, el uso de técnicas de regularización y normalización, y la selección de métricas.

Se observó que el modelo más complejo, con múltiples capas ocultas, normalización por lotes y regularización mediante dropout, fue el que obtuvo el mejor desempeño en términos de error cuadrático medio (MSE) y error absoluto medio (MAE), sin evidenciar sobreajuste. Esto demuestra que, si bien los modelos simples pueden ofrecer resultados aceptables, el uso de técnicas adicionales puede mejorar sustancialmente la capacidad predictiva de las redes neuronales.

6. Mejoras a futuro

- Aumentar el tamaño del dataset, utilizando técnicas de generación de datos sintéticos o integrando simulaciones adicionales para mejorar la capacidad generalizadora del modelo.
- Explorar otros algoritmos de regresión como *Random Forest*, *XGBoost* o *Support Vector Regression* para comparar su desempeño con las redes neuronales.
- Optimizar los hiperparámetros mediante técnicas automatizadas como *Grid Search* o *Bayesian Optimization*.
- Implementar una validación cruzada más robusta para evaluar la estabilidad del modelo ante diferentes particiones del conjunto de datos.

7. Visita el PDF

