

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS
MAESTRÍA EN CIENCIA DE DATOS

**”Modelo de predicción del torque
y temperatura en motores
síncronos de imanes permanentes
mediante redes neuronales no
lineales”**

Marco Antonio Obregón Flores

Profesor:

José Alberto Benavides Vazquez

13 de marzo de 2023

Abstract

Este artículo se enfoca en la predicción del torque y la temperatura del estator en motores síncronos de imanes permanentes mediante modelos de regresión. Se discute el modelo de regresión no lineal basado en redes neuronales, también conocido como red neuronal artificial (ANN), como uno de los modelos más efectivos y ampliamente utilizado. La ANN puede aprender relaciones no lineales entre entradas y salidas mediante el ajuste de los pesos de las conexiones entre las neuronas, lo que la hace adecuada para la predicción de datos complejos y no lineales como la temperatura y el torque en motores síncronos de imanes permanentes. Se presentan diferentes algoritmos de optimización, incluyendo el algoritmo de retropropagación y el algoritmo de Levenberg-Marquardt, y se discute la aplicación de técnicas de regularización como la regularización bayesiana para evitar el sobreajuste del modelo. Los resultados muestran que la variable de torque es la más relevante en el modelo, ya que presenta el mejor ajuste y menor error en todas las métricas evaluadas (RMSE, MAE y MSE). Además, el coeficiente de determinación (R^2) indica que es la variable con mayor capacidad para explicar la variación en los datos observados. Se recomienda enfocarse en el análisis de esta variable debido a sus buenos resultados en términos de ajuste y precisión.

1 Introducción

La temperatura es una de las variables más críticas en la operación de los motores eléctricos, ya que puede afectar su rendimiento y vida útil. En particular, en los motores síncronos de imanes permanentes, las altas temperaturas en el devanado del estator, el diente del estator y el yugo del estator pueden afectar la eficiencia y la estabilidad del motor. Por lo tanto, es importante medir y controlar estas temperaturas durante la operación del motor.

En la actualidad, existen diversas técnicas para medir las temperaturas en diferentes partes del motor, como el uso de termopares y unidades de termografía. Además, se pueden utilizar modelos de regresión para predecir estas temperaturas a partir de otras variables medidas en el motor, como la corriente, el voltaje y la velocidad.

Existen varias variables que influyen en la temperatura del estator, como la temperatura del refrigerante (**coolant**), la temperatura del devanado del estator (**stator_winding**) medida con termopares, la temperatura del diente del estator (**stator_tooth**) medida con termopares, la velocidad del motor (**motor_speed**), la corriente del componente d en coordenadas dq (**i.d**), la corriente del componente q en coordenadas dq (**i.q**), la temperatura de imán permanente (**pm**) medida con termopares y transmitida de forma inalámbrica a través de una unidad de termografía, y la temperatura del yugo del estator (**stator_yoke**) medida con termopares.

En este artículo, nos enfocamos en la predicción del torque y la temperatura del estator en motores síncronos de imanes permanentes mediante modelos de regresión. En particular, se discute el modelo de regresión no lineal basado en redes neuronales, también conocido como red neuronal artificial (ANN), como uno de los modelos más efectivos y ampliamente utilizado. La ANN puede aprender relaciones no lineales entre entradas y salidas mediante el ajuste de los pesos de las conexiones entre las neuronas, lo que la hace adecuada para la predicción de datos complejos y no lineales como la temperatura y el torque en motores síncronos de imanes permanentes.

2 Metodología Propuesta

En el desarrollo del modelo de predicción se consideraron los factores que afectan el torque del motor y la temperatura del estator, como la velocidad del motor, el voltaje y la corriente de los componentes d y q, y las temperaturas ambiental y del refrigerante. Por lo tanto, en este artículo se consideraron siete parámetros de entrada: la temperatura ambiente, la temperatura del refrigerante, el componente d del voltaje, el componente q del voltaje, la velocidad del motor, el componente d de la corriente y el componente q de la corriente. Los parámetros de salida considerados son el torque del motor y las temperaturas en la superficie del imán permanente, el yugo del estator, el diente del estator y el devanado del estator.

El procedimiento para obtener el proceso de predicción consiste en los siguientes pasos:

1. **Preprocesamiento de datos:** Se realiza una limpieza y preparación de los datos, eliminando valores faltantes o inconsistentes y normalizando los datos si es necesario.
2. **Selección de características:** Se seleccionan las características relevantes que se utilizarán en el modelo de predicción. En este caso, se considerarán las variables mencionadas en la introducción.
3. **División de datos:** Se divide el conjunto de datos en dos partes, una para entrenar el modelo y otra para probar su rendimiento.
4. **Entrenamiento del modelo:** Se selecciona el modelo de regresión no lineal basado en redes neuronales y se entrena con el conjunto de datos de entrenamiento. Se utilizan diferentes algoritmos de optimización, como el algoritmo de retropropagación o el algoritmo de Levenberg-Marquardt, y se aplican técnicas de regularización como la regularización bayesiana para evitar el sobreajuste del modelo.

5. **Evaluación del modelo:** Se evalúa el rendimiento del modelo utilizando el conjunto de datos de prueba. Se calculan métricas como el error cuadrático medio (MSE) y el coeficiente de determinación (R^2) para evaluar la precisión del modelo. Además, también se utilizan métricas como el error absoluto medio (MAE) y la raíz del error cuadrático medio (RMSE) para evaluar el desempeño del modelo en términos de la magnitud del error. Todas estas métricas permiten una evaluación rigurosa y completa del modelo en términos de su capacidad para predecir los resultados esperados.

Error cuadrático medio (MSE). La fórmula matemática del MSE es:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Donde y_i es el valor real de la variable a predecir, \hat{y}_i es el valor predicho por el modelo y n es el número total de observaciones.

Error de raíz cuadrada media (RMSE). La fórmula matemática del RMSE es:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

donde n es el número de observaciones, y_i son los valores observados, y \hat{y}_i son los valores predichos.

Error absoluto medio (MAE). La fórmula matemática del MAE es:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

donde n es el número de observaciones, y_i son los valores observados, y \hat{y}_i son los valores predichos.

Coeficiente de determinación (R^2): La fórmula matemática del R^2 es:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Donde y_i es el valor real de la variable a predecir, \hat{y}_i es el valor predicho por el modelo, \bar{y} es la media de los valores reales y n es el número total de observaciones.

6. **Optimización del modelo:** Si el modelo no alcanza el rendimiento deseado, se realizan ajustes en la selección de características, hiperparámetros o algoritmos de optimización hasta que se logre un rendimiento aceptable.

7. **Uso del modelo para predicción:** Una vez que se ha optimizado el modelo, se puede utilizar para predecir el torque y la temperatura del estator en motores síncronos de imanes permanentes basados en las entradas de las variables mencionadas en la introducción.

3 Resultados y Discusión

3.1 Descripción de los datos

Los registros corresponden a mediciones realizadas a un motor síncrono de imán permanente (PMSM), los cuales fueron muestreados a una frecuencia de 2 Hz. El conjunto de datos contiene múltiples sesiones de medición, las cuales se pueden distinguir por el identificador de perfil (`profile_id`) y tienen una duración variable de entre una y seis horas. En total, se registraron 185 horas de operación del motor.

Table 1: Variables del conjunto de datos

Variable	Descripción
<code>u_q</code>	Tensión en el eje q del motor
<code>coolant</code>	Temperatura del refrigerante del motor
<code>stator_winding</code>	Temperatura del devanado del estator del motor
<code>u_d</code>	Tensión en el eje d del motor
<code>stator_tooth</code>	Temperatura del diente del estator del motor
<code>motor_speed</code>	Velocidad del motor
<code>i_d</code>	Corriente en el eje d del motor
<code>i_q</code>	Corriente en el eje q del motor
<code>pm</code>	Temperatura del imán permanente del motor
<code>stator_yoke</code>	Temperatura del yugo del estator del motor
<code>ambient</code>	Temperatura ambiente durante la medición
<code>torque</code>	Par del motor
<code>profile_id</code>	Identificador de la sesión de medición

Cabe destacar que el motor es excitado por ciclos de conducción diseñados a mano, que establecen una velocidad de referencia y un par de referencia. Las corrientes y voltajes en coordenadas d/q son resultado de una estrategia de control estándar que intenta seguir la velocidad y el par de referencia, y las variables de velocidad y torque son las cantidades resultantes logradas por esa estrategia, derivadas de las corrientes y voltajes establecidos. La mayoría de los ciclos de conducción corresponden a caminatas aleatorias en el plano velocidad-par, con el fin de imitar ciclos de conducción del mundo real de manera más precisa que las excitaciones y rampas de subida y bajada constantes.

3.2 Preprocesamiento de datos

El código utiliza una red neuronal artificial, que es un método de aprendizaje automático que simula el funcionamiento del cerebro humano. El objetivo es predecir algunas variables de salida a partir de unas variables de entrada usando los datos de un archivo csv.

El código sigue los siguientes pasos:

1. Cargar los datos desde Google Colab y leerlos con pandas.
2. Definir las variables predictoras y la variable a predecir.
3. Dividir los datos en conjunto de entrenamiento y prueba, usando el 70% de los datos para entrenar y el 30% para probar.
4. Escalar los datos usando la clase `MinMaxScaler`, que transforma los valores al rango $[0, 1]$.
5. Definir el modelo de red neuronal usando la clase `Sequential`, que permite crear capas de neuronas de forma secuencial.
6. Añadir las capas ocultas usando la clase `Dense`, que crea neuronas totalmente conectadas con una función de activación (`relu`) y una regularización (`L1L2`) para evitar el sobreajuste. También se usa la clase `Dropout`, que elimina aleatoriamente algunas neuronas durante el entrenamiento para mejorar la generalización.
7. Añadir la capa de salida usando la clase `Dense`, que crea neuronas totalmente conectadas con una función de activación lineal, que es adecuada para problemas de regresión.
8. Definir el optimizador y la función de pérdida usando la clase `Adam` y el error cuadrático medio (`mse`), respectivamente.

3.3 Diseño de experimentos

El diseño óptimo de Box-Behnken es una técnica estadística para encontrar la mejor combinación de variables que afectan a un proceso. En este caso, se quiere encontrar la combinación óptima de variables para obtener las mejores mediciones de rendimiento del motor eléctrico.

Para aplicar el diseño óptimo de Box-Behnken en Python, se usó una librería llamada `pyDOE2`. Esta librería tiene una función que se llama `"bbdesign"` que se puede usar para generar la matriz de diseño de Box-Behnken.

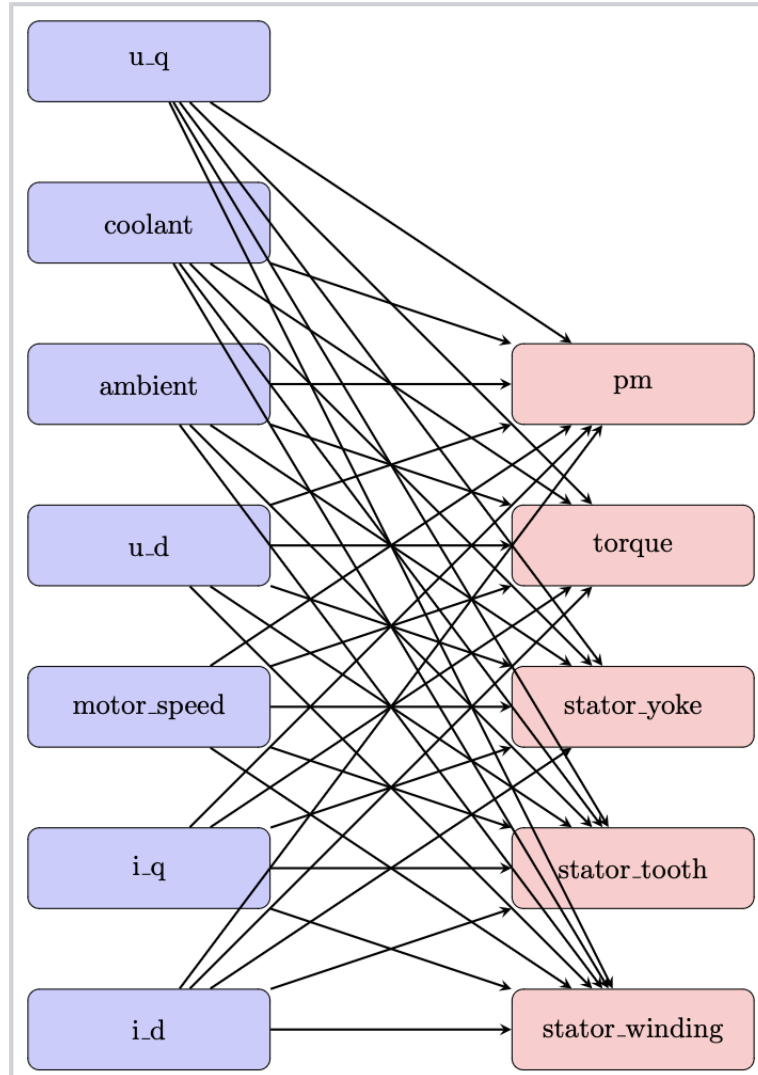
3.4 Desempeño del modelo de predicción

El modelo usa la librería Keras, que es una librería de código abierto para crear redes neuronales artificiales. El código hace lo siguiente:

- Entrena un modelo de red neuronal con los datos de entrenamiento (`train_set`) usando la función `fit` , que recibe como argumentos las variables predictoras, las variables a predecir, los datos de validación (`test_set`), el número de épocas (`epochs`) y el tamaño del lote (`batch_size`).
- Predice las variables a predecir con los datos de prueba (`test_set`) usando la función `predict` , que recibe como argumento las variables predictoras.
- Calcula las métricas de evaluación en el conjunto de prueba usando las funciones de NumPy `mean`, `square`, `sqrt`, `abs` y `sum`. Las métricas son el error cuadrático medio (MSE), la raíz del error cuadrático medio (RMSE), el error absoluto medio (MAE) y el coeficiente de determinación (R^2).

En particular, este modelo representa un motor eléctrico y su relación con varias entradas (`u_q`, `coolant`, `ambient`, `u_d`, `motor_speed`, `i_q` e `i_d`) y salidas (`torque`, `pm`, `stator_yoke`, `stator_tooth` y `stator_winding`). Las entradas representan los diferentes factores que afectan el rendimiento del motor eléctrico, como la temperatura del refrigerante y el entorno, la velocidad y la corriente, y las salidas representan las diferentes medidas de rendimiento del motor eléctrico, como la producción de torque y la magnetización del rotor permanente.

Figure 1: Estructura del modelo



Evaluación de modelos			
Metrica	Variable	Modelo	Valor
R^2	torque	Lineal	0.088
		ANN	0.943500
		Box-Behnken + ANN	0.969545
	pm	Lineal	0.070
		ANN	0.480450
		Box-Behnken + ANN	0.473920
	stator_yoke	Lineal	0.062
		ANN	0.836566
		Box-Behnken + ANN	0.850503
	stator_tooth	Lineal	0.076
		ANN	0.723499
		Box-Behnken + ANN	0.735437
	stator_winding	Lineal	0.091
		ANN	0.650030
		Box-Behnken + ANN	0.654852

Table 2: Tabla de comparativa de métricas

Los modelos evaluados son un modelo de regresión lineal OLS, una red neuronal artificial (ANN) y una combinación de Box-Behnken y ANN. Los resultados muestran que el modelo de regresión lineal OLS se desempeñó peor en todas las variables en comparación con los otros dos modelos. En particular, se puede observar que la combinación de Box-Behnken y ANN tuvo el mejor desempeño en todas las variables, con valores de R^2 que van desde 0.47 hasta 0.97, lo que indica que este modelo tiene la capacidad de explicar una gran parte de la variabilidad en los datos. En general, esta tabla puede ser útil para seleccionar el mejor modelo en función de la variable que se está analizando en el estudio

References

3.5 Artículo científico

1. Jun Li, Thangarajah Akilan, (2022). Global Attention-based Encoder-Decoder LSTM Model for Temperature Prediction of Permanent Magnet Synchronous Motors. IEEE, 11.
2. Smith, L. N. (2018). A disciplined approach to neural network hyperparameters: Part 1 – learning rate, batch size, momentum, and weight decay. arXiv preprint arXiv:1803.09820.
3. Bottou, L., Curtis, F. E., Nocedal, J. (2018). Optimization methods for large-scale machine learning. Siam Review, 60(2), 223-311.

3.6 Libro

1. Hastie, T., Tibshirani, R., Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
2. Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning. MIT press.

3.7 Web

1. Kirgiz, W. (2021). Electric Motor Temperature. Recuperado el 10 de enero de 2023, de <https://www.kaggle.com/datasets/wkirgsn/electric-motor-temperature>