

Desarrollo de una metodología para control dinámico de motores con Machine Learning

Antonio Eduardo Álvarez Núñez¹, María del Carmen Santiago Díaz¹,
Ana Claudia Zenteno Vázquez¹, María Catalina Rivera Morales²,
María Dolores Guevara Espinosa², Gustavo Trinidad Rubín Linares¹

¹ Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

² Benemérita Universidad Autónoma de Puebla,
Facultad de Ingeniería Química,
México

{maricarmen.santiago, ana.zenteno,
gustavo.rubin}@correo.buap.mx

Resumen. En robótica el problema de la precisión en el control de posición de los diferentes sistemas motrices requiere de la evaluación de múltiples variables en tiempo real a fin de no ocasionar una inestabilidad en el robot por el tiempo de procesamiento. En éste trabajo se propone una metodología basada en el aprendizaje automático de los sistemas motrices a fin de poder generar acciones de control con antelación a su medición y procesamiento, para lo cual se propone un sistema de segundo orden y su respuesta a perturbaciones como señales escalón y rampas.

Palabras clave: Aprendizaje automático, control dinámico, asistentes inteligentes.

Development of a Methodology for Dynamic Motor Control with Machine Learning

Abstract. In robotics, the problem of precision in the position control of the different motor systems requires the evaluation of multiple variables in real time in order not to cause instability in the robot due to processing time. In this paper, a methodology based on automatic learning of motor systems is proposed in order to be able to generate control actions in advance of their measurement and processing, for which a second-order system and its response to disturbances such as step signals are proposed. and ramps.

Keywords: Machine learning, dynamic control, intelligent assistants.

1. Introducción

Inteligencia artificial (IA) se refiere a sistemas o máquinas que imitan la inteligencia humana para realizar tareas y pueden mejorar iterativamente a partir de la información que recopilan. La IA se manifiesta de varias formas. Algunos ejemplos son:

- Los chatbots utilizan la IA para comprender más rápido los problemas de los clientes y proporcionar respuestas más eficientes.
- Los asistentes inteligentes utilizan la IA para analizar información crítica proveniente de grandes conjuntos de datos de texto libre para mejorar la programación.
- Los motores de recomendación pueden proporcionar recomendaciones automatizadas para programas de TV según los hábitos de visualización de los usuarios.

La IA se ha convertido en un término general para las aplicaciones que realizan tareas complejas que antes requerían aportes humanos, como la comunicación online con los clientes o jugar al ajedrez. El término a menudo se usa indistintamente con sus subcampos, que incluyen el aprendizaje automático y el aprendizaje profundo. Sin embargo, hay ciertas diferencias.

Por ejemplo, el machine learning se centra en la creación de sistemas que aprenden o mejoran su rendimiento en función de los datos que consumen. Es importante tener en cuenta que, aunque todo machine learning es IA, no toda la IA es machine learning [1].

1.1. Machine Learning

El Machine Learning, o aprendizaje automático, es una rama de la inteligencia artificial que permite que las máquinas aprendan sin estar programadas para este propósito específico. Una habilidad esencial para crear sistemas que no solo sean inteligentes, sino autónomos y capaces de identificar patrones en los datos para convertirlos en predicciones [2].

1.2. Tipos de Machine Learning

El aprendizaje automático clásico a menudo se clasifica según la forma en que un algoritmo aprende a ser más preciso en sus predicciones. Hay cuatro enfoques básicos: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semi supervisado y aprendizaje por refuerzo. El tipo de algoritmo que los científicos de datos eligen usar depende del tipo de datos que quieran predecir [3].

- **Aprendizaje supervisado:** en este tipo de aprendizaje automático, los científicos de datos proporcionan algoritmos con datos de entrenamiento etiquetados y definen las variables que quieren que el algoritmo evalúe para las correlaciones. Se especifica tanto la entrada como la salida del algoritmo.

Tabla 1. Parámetros de la data base.

Tiempo de simulación	Muestreo	Cantidad de entradas escalón
10 s	100	50

Tabla 2. Planta con entradas seno.

	Con ruido	Sin ruido
Normalizado	Método regresión lineal MSE: 214263771407263412925759488.0000	Método regresión lineal MSE: 5.1602
	k-Nearest Neighbors para regresión multi-output MSE: 27.1421	k-Nearest Neighbors para regresión multi-output MSE: 27.1174
	Decision Tree para regresión multi-output MSE: 89.9494	Decision Tree para regresión multi-output MSE: 141.6744
	Red neuronal multi-output MSE: 2935.1144	Red neuronal multi-output MSE: 2604.0644
No normalizado	Método regresión lineal MSE: 4730340379753366876585984.0000	Método regresión lineal MSE: 125.3017
	k-Nearest Neighbors para regresión multi-output MSE: 133.0015	k-Nearest Neighbors para regresión multi-output MSE: 132.9581
	Decision Tree para regresión multi-output MSE: 174.3104	Decision Tree para regresión multi-output MSE: 174.3605
	Red neuronal multi-output MSE: 8.8519	Red neuronal multi-output MSE: 5.4136

- **Aprendizaje no supervisado:** este tipo de aprendizaje automático implica algoritmos que se entrenan con datos no etiquetados. El algoritmo escanea a través de conjuntos de datos en busca de cualquier conexión significativa. Los datos con los que se entrenan los algoritmos, así como las predicciones o recomendaciones que generan, están predeterminados.
- **Aprendizaje semi supervisado:** este enfoque de aprendizaje automático implica una combinación de los dos tipos anteriores. Los científicos de datos pueden alimentar un algoritmo mayormente etiquetado como datos de entrenamiento, pero el modelo es libre de explorar los datos por sí mismo y desarrollar su propia comprensión del conjunto de datos.
- **Aprendizaje por refuerzo:** los científicos de datos suelen utilizar el aprendizaje por refuerzo para enseñar a una máquina a completar un proceso de varios pasos para el que existen reglas claramente definidas. Los científicos de datos programan un algoritmo para completar una tarea y le dan señales positivas o

negativas a medida que descubre cómo completar una tarea. Pero en su mayor parte, el algoritmo decide por sí mismo qué pasos tomar en el camino.

Los datos multivariados ocurren en una variedad de disciplinas, por ejemplo, en la investigación biomédica, las ciencias sociales, o la econometría. Se dice que los datos son multivariados si la respuesta no solo consta de una variable, sino de $d \geq 2$ variables de salida, digamos $\overline{Y} \in \mathbb{R}$. Entonces, a menudo estamos interesados en encontrar una relación funcional entre la salida \overline{Y} y algunas características variables $\overline{X} \in \mathbb{R}$, es decir, queremos realizar un análisis de regresión multivariable.

A diferencia de la regresión múltiple uní variable ($d = 1$), que también incluye características múltiples de \overline{X} , la regresión multivariable trata de especificar la relación de varias variables de resultado con \overline{X} simultáneamente.

El objetivo de tales análisis multivariados es que la consideración de las posibles dependencias entre los resultados pueda conducir a procedimientos con mejor potencia (en caso de inferencia) o precisión (en caso de predicción) en comparación con análisis univariados separados [1].

2. Metodología

La regresión es una tarea de modelado predictivo que implica predecir una salida numérica dada alguna entrada.

Es diferente de las tareas de clasificación que implican predecir una etiqueta de clase.

Por lo general, una tarea de regresión implica predecir un solo valor numérico. Aunque, algunas tareas requieren predecir más de un valor numérico. Estas tareas se conocen como regresión de salida múltiple o regresión de salida múltiple para abreviar.

En la regresión de múltiples salidas, se requieren dos o más salidas para cada muestra de entrada y las salidas se requieren simultáneamente. La suposición es que las salidas son una función de las entradas [2].

Para que funcionen mejor muchos algoritmos de Machine Learning usados en Data Science, hay que normalizar las variables de entrada al algoritmo. Normalizar significa, en este caso, comprimir o extender los valores de la variable para que estén en un rango definido. Sin embargo, una mala aplicación de la normalización, o una elección descuidada del método de normalización puede arruinar tus datos, y con ello tu análisis [3].

Existen técnicas de normalización que comprime los datos de entrada entre unos límites empíricos (el máximo y el mínimo de la variable). Esto quiere decir que, si existe ruido, éste va a ser ampliado [3].

Para ver el efecto de normalizar los datos y el tener ruido en la señal que se medirá un futuro directamente desde los sensores del robot, se decidió hacer cuatro casos de estudio:

1. Implementar los algoritmos con el conjunto de datos sin normalizar y sin ruido.
2. Implementar los algoritmos con el conjunto de datos sin normalizar y con ruido.
3. Implementar los algoritmos con el conjunto de datos normalizados y sin ruido.
4. Implementar los algoritmos con el conjunto de datos sin normalizar y con ruido.

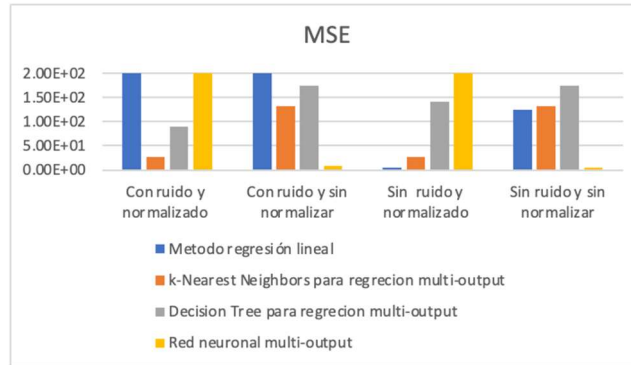


Fig. 1. Comparación de los errores cuadráticos medio.

Tabla 3. Comparación de la red neuronal con entrada escalón en las tres implementaciones.

	Con ruido y normalizado	Con ruido y sin normalizar	Sin ruido y normalizado	Sin ruido y sin normalizar
Red neuronal multi-output (1)	20215,8381	0,4571	18726,291	0,1747
Red neuronal multi-output (2)	2494159,312	0,5066	163.150	0,0981
Red neuronal multi-output (3)	284.105	0,0776	276104,4268	38912,2673

3. Resultados

A continuación, vemos los resultados, primero se dará un vistazo a los algoritmos con la planta recibiendo escalones unitarios y después recibiendo entradas seno.

3.1. Planta con entradas seno

Aquí la predicción es mejor cuando los datos no tienen ruido, y esta sin normalizar. También la red neuronal mejora su comportamiento cuando los datos no se normalizan.

En la tabla podemos ver que el error cuadrático medio (MSE) disminuye cuando los datos no contienen ruido. Por lo tanto, se recomienda que los datos obtenidos de los sensores pasen por un tratamiento de señal para reducir el ruido de estas. Por otro lado, aún no se puede concluir que normalizar los datos afecten la predicción de los modelos, ya que esto también se puede ver afectado por la cantidad de información con la que se entrenan los modelos. Por lo tanto, queda generar una base de datos más extensa para ver si los resultados mejoran.

3.2. Comparación de los resultados de las tres metodologías

En las Fig. 2 y 3 y en la tabla 4 se presentan los resultados de aplicación de las tres metodologías descritas anteriormente.

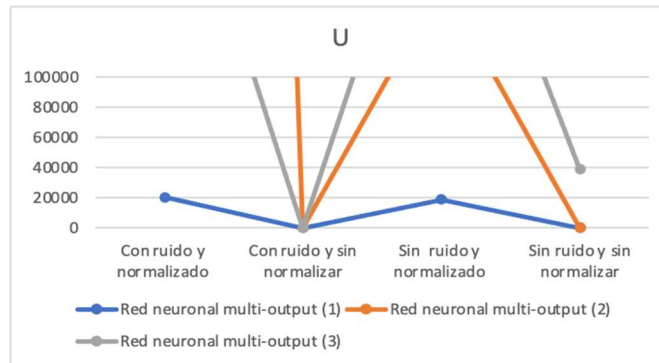


Fig. 2. Comportamiento de la red neuronal en las tres implementaciones.

Tabla 4. Comparación de la red neuronal con entrada escalón en las tres implementaciones.

	Con ruido y normalizado	Con ruido y sin normalizar	Sin ruido y normalizado	Sin ruido y sin normalizar
Red neuronal multi-output (1)	2935,1144	8,8519	2604,0644	5,4136
Red neuronal multi-output (2)	285180,3809	1478,1502	163.150	1105,6508
Red neuronal multi-output (3)	7,225	17,5880	13993,253	13,2334

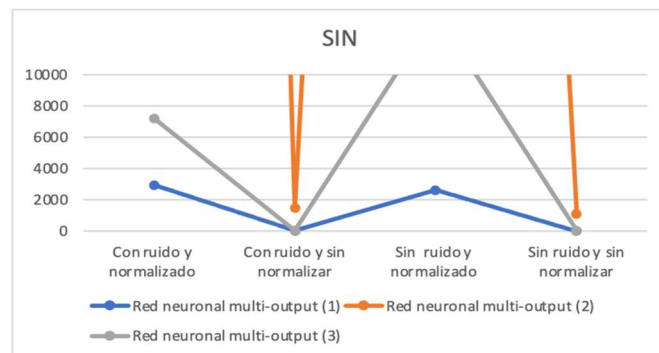


Fig. 3. Comportamiento de la red neuronal en las tres implementaciones

4. Conclusiones

Podemos observar que las predicciones mejoran en el caso donde los datos no son normalizados, también se observa que en la segunda metodología no hizo que las predicciones mejoraran.

Un punto importante por remarcar es que la red neuronal presenta mejor desempeño con la primera metodología, ya que se diseñó la red neuronal para ese caso, para las

siguientes dos metodologías se ocupó la misma red neuronal, los datos mejorarían si se hace una red neuronal para cada caso.

Para tener una red neuronal capaz de predecir cualquier entrada no solo la escalón y seno, es necesario entrenar la red con entradas aleatorias, este es la siguiente metodología que seguir.

References

1. Schmid, L., Gerharz, A., Groll, A., Pauly, M.: Machine learning for multi-output regression: when should a holistic multivariate approach be preferred over separate univariate ones? (2022) doi: 10.48550/ARXIV.2201.05340
2. Brownlee, J.: How to develop multi-output regression models with python. Machine Learning Mastery, Ensemble Learning (2020) machinelearningmastery.com/multi-output-regression-models-with-python/
3. Morante, S.: Precauciones a la hora de normalizar datos en data science. Telefónica Tech (2018) empresas.blogthinkbig.com/precauciones-la-hora-de-normalizar/