AUTOMATIZANDO LA OPERACIÓN DE LA RED DE TRANSPORTE DE ELECTRICIDAD

Anguiano Batanero - Eloy, Data Scientist, Instituto de Ingeniería del Conocimiento/ADIC
Fernández Pascual - Ángela, Professor, Universidad Autónoma de Madrid
Barbero Jiménez - Álvaro, Chief Data Scientist, Instituto de Ingeniería del Conocimiento/ADIC

Resumen: Este artículo presenta una metodología innovadora para la automatización de la operación de la red de transporte de electricidad mediante agentes autónomos basados en aprendizaje por refuerzo. La gestión de redes eléctricas es compleja y requiere un servicio confiable, especialmente con la creciente integración de prosumidores y energías limpias. Este artículo explora la posibilidad de agentes que operan a nivel topológico, optimizando la toma de decisiones y minimizando pérdidas de energía. Al comparar estos agentes con un enfoque de inacción, se evidencia que la automatización puede superar las limitaciones de la inacción, destacando la relevancia de la inteligencia artificial en la gestión de redes eléctricas.

Palabras clave: Aprendizaje, Refuerzo, Red, Transporte, Operación, Agentes, Optimización, Inteligencia, Artificial.

1. INTRODUCCIÓN Y ANTECEDENTES

La red eléctrica como un proceso de decisión de Markov (MDP).

El control de una red de transmisión eléctrica es una tarea compleja que requiere garantizar un servicio estable y confiable, mientras se prioriza el uso de energías limpias y/o abaratar costes. Con la aparición de prosumidores y redes inteligentes, esta tarea se ha ido haciendo cada vez más complicada debido a que la capacidad de análisis de los operadores del sistema tiene fuertes restricciones temporales. Aunque existen soluciones que utilizan conocimiento experto y restricciones de acciones, en este artículo se propone una nueva metodología para desarrollar agentes autónomos basados en aprendizaje por refuerzo capaces de operar la red sin necesidad de dicho conocimiento experto, explorando formas de reducir costos cuando la red no está en riesgo.

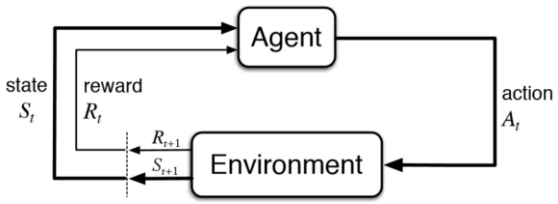

Figura 1. Esquema de iteración de un proceso de decisión de Markov.

Para comenzar a considerar la aplicación de métodos de aprendizaje por refuerzo, es necesario formalizar la operación de una red eléctrica como un proceso de decisión de Markov (MDP). Un proceso de decisión de Markov es un proceso de control estocástico discreto destinado a optimizar la toma de decisiones, en el cual se definen un conjunto de estados S, un conjunto de acciones a tomar A y una función de recompensa R a maximizar. De esta forma, se pretende obtener un agente que encuentre la política de optimización que buscamos que maximice el retorno de recompensas actuales y futuras.

El framework Grid2Op.

Grid2Op (Donnot, 2020) es un framework de código abierto propuesto por Réseau de Transport d'Électricité (RTE) y utilizado en la competencia “Learn to Run a Power Network” (L2RPN) (Marot et al, 2021) para abstraer a los investigadores de la complejidad de la función de transición de estados F, siguiendo algunos de los estándares propuestos por las bibliotecas Gym (Brockman et al, 2016) de OpenAI. En el campo del aprendizaje por refuerzo, es

común referirse a esta abstracción de la función de transición como el "Entorno". Además, este marco proporciona a programadores e investigadores la libertad de establecer la definición de un estado de la red eléctrica s (también llamado observación), así como el conjunto de acciones A que el agente puede tomar y la función de recompensa R entre estados de la red.

El framework Grid2Op hace varias suposiciones sobre el proceso de decisión de Markov que define sobre redes eléctricas y que deben ser explicadas antes de profundizar en más detalle, ya que son clave para varias suposiciones encontradas tanto en la literatura como en este artículo. La primera de ellas se refiere a las condiciones bajo las cuales se determina que un episodio ha terminado, es decir, cuándo se ha alcanzado un estado terminal. Estos estados terminales pueden darse por condición de blackout de la red, violación del criterio N-1 (el estado resultante de la red tras una acción debe ser resistente para que, en caso de eliminar cualquier elemento de la topología de la red obtenida, el resultado siga siendo estable) de redes de transporte haber alcanzado el final de los datos de un caso (crónica) de longitud 2016 pasos en intervalos de 5 minutos.

Otro de los principales desafíos de estos sistemas es el gran número de acciones posibles y la complejidad de muestrear una acción coherente con el estado actual de la red. Para abordar esto, Grid2Op proporciona información sobre si la acción ha sido categorizada en cuatro tipos distintos: correcta, ilegal, ambigua o errónea.

Una acción correcta corresponde a una acción que puede ser ejecutada completamente. Este tipo de acción es lo que buscamos que nuestro agente genere. Por el contrario, una acción errónea sería aquella que lleva a un estado irresoluble. Aunque es importante evitar tales acciones, aún pueden ocurrir debido a complicaciones matemáticas internas inherentes a cualquier simulador de flujo de carga de red eléctrica. Es importante la distinción entre los tipos de acciones ilegales y acciones ambiguas. En este sentido, las acciones ilegales son aquellas que no son factibles dadas las circunstancias actuales de la red eléctrica (por ejemplo, reconectar una línea eléctrica que está en mantenimiento), mientras que las acciones ambiguas son aquellas que no tienen sentido por sí mismas (por ejemplo, conectar el elemento 999 de la red, que no existe). Debemos asegurar un buen diseño del espacio de acciones para que no incluya acciones ambiguas ni ilegales, ni siquiera en las fases de exploración aleatoria típicas de este tipo de estrategias.

2. DESCRIPCIÓN DE LA SOLUCIÓN.

La solución propuesta en este artículo se compone de varias partes. Para entender completamente la propuesta es necesario hablar de cómo se codifica un estado de la red (a lo que se le denomina observación), qué función es necesario maximizar (función de recompensa), qué algoritmo y qué metodología se ha utilizado para obtener los agentes objetivo y qué capacidades tiene dicho agente (las acciones que es capaz de realizar).

Observación.

La formalización matemática de un estado de red en algo entendible para el agente es una parte crucial de la configuración de la solución. En nuestro caso, hemos hecho que cada observación del estado de la red se componga pon un vector N dimensional que llamaremos “FlatObs” compuesto por las siguientes variables:

- Módulos de voltajes de cada productor de energía (generadores).
- Módulos de potencia, componentes de potencia activa y componentes de potencia reactiva de cada generador y carga de la red eléctrica.
- Capacidad de cada línea eléctrica (rho), que representa el flujo de corriente en relación con el límite térmico de la línea.
- Estado de cada línea (conectada/desconectada).

Recompensa.

La recompensa es el valor que el agente busca maximizar en sus interacciones con el entorno. En problemas de control, como la gestión de redes eléctricas, la duración de los episodios depende del rendimiento del agente. Por ello, nos interesa una función de recompensa positiva para que el agente mantenga la red funcionando el mayor tiempo

posible, pero además queremos también que la magnitud de la recompensa sea más o menos alta en función de los costes de gestión de la red, favoreciendo la minimización de los costes de operación.

Teniendo en cuenta las restricciones que tiene que tener esta función en el marco que nos atañe y el objetivo de abaratar costes de gestión de red, una aproximación bastante directa sería la minimización de las pérdidas que se obtienen ante un escenario de red. Para ello definimos la siguiente ecuación

$$R(s_t, a_{t-1}) = \begin{cases} \frac{Loss_{max}(s_t) - Loss_{act}(s_t, a_{t-1})}{Loss_{max}(s_t)}, & \text{si } Loss_{max}(s_t) \neq 0 \\ 1, & \text{si } Loss_{max}(s_t) = 0 \end{cases}$$

teniendo en cuenta que la pérdida se define como aquella energía generada que no llega a las cargas, teniendo en cuenta el coste del marginal en cada instante de tiempo. Por tanto, la actual sería calcular el resultado de nuestro estado mientras que para la pérdida máxima sería calcular el caso en el que ninguna carga recibiera energía.

Acciones, algoritmo y arquitectura del modelo.

Los desarrollos de este trabajo van enfocados a obtener agentes que operen únicamente a nivel topológico, es decir, que únicamente operen conectando y desconectando los elementos de la red (generadores, cargas, extremos de líneas) a los distintos buses de cada subestación. Esto se debe a que suelen ser las acciones preferentes a la hora de operar una red eléctrica por su bajo impacto económico.

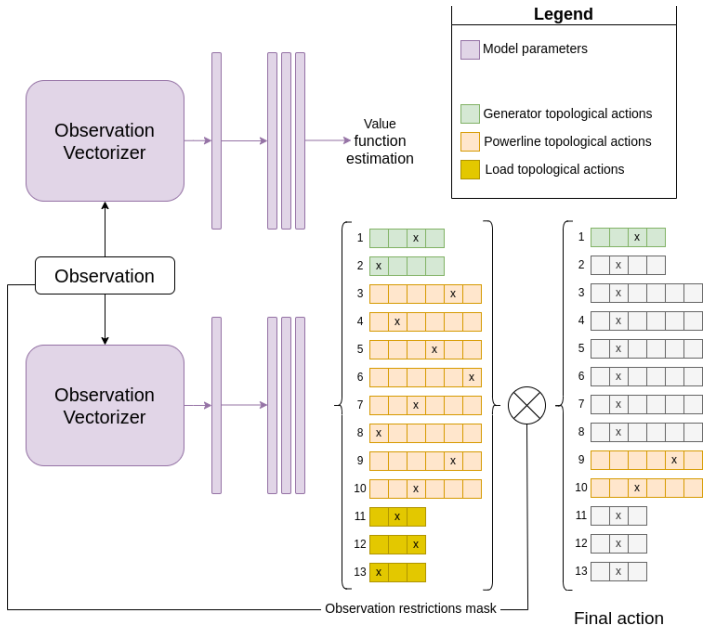

Figura 2. Arquitectura de operación del agente. Se puede observar cómo tiene las dos salidas necesarias para el algoritmo PPO y cómo existe una lógica para enmascarar (forzar la dimensión de inacción) la acción final según el estado u observación actual.

Atendiendo a esta restricción, cada uno de los elementos de una red será una dimensión sobre la cual el agente tendrá que escoger una opción, y las elecciones que podrá realizar dependerán de la naturaleza de dicho elemento y teniendo en cuenta que es necesario añadir una acción extra a cada elemento que representa la "inacción" sobre dicho elemento. Esta necesidad nace de evitar las acciones ilegales, y es necesario rellenar la dimensión asociada a esa acción para que el agente asocie esa decisión al valor de inacción de salida.

Es por esto que, en el caso que nos atañe en el que cada subestación está configurada con dos barras, la dimensión de la acción asociada a un generador tendrá 4 posibilidades (desconexión, inacción, conexión bus 1 y conexión bus 2), la de una línea tendrá 6 (desconexión, inacción y la combinatoria de conexión de cada extremo a los dos buses de cada subestación que conecta) y en el caso de una carga tendrá 3 (inacción, conexión a bus 1 y conexión a bus 2) ya que según la definición del entorno no se plantean situaciones de Value Of Lost Load (VOLL). En la Figura 2 se observa que para el nuestro caso particular ("rte_case5_example") el entorno presenta una topología de red de 5 subestaciones, las cuales conectan 13 elementos: 2 generadores, 8 líneas y 3 cargas.

Gracias a la definición de este espacio de acciones, podemos aplicar un algoritmo de Proximal Policy Optimization (Schulman et al, 2017) en su variante enmascarable (Huang & Ontañón, 2020) con un modelo simple de red de perceptrón multicapa que optimice tanto la selección de las acciones (la política del agente) como la esperanza de futuras recompensas al aplicar dichas acciones (la función valor).

Metodología.

Como en cualquier estrategia basada en aprendizaje automático, es necesario definir distintos conjuntos para el proceso de aprendizaje y de validación o test de tu agente. Para ello, vamos a especificar la metodología utilizada en cada una de las fases del agente.

Para empezar, en su fase de entrenamiento, se han establecido todas las crónicas de la 01 a la 20 excepto la 17 y la 19 como el conjunto de entrenamiento. Adicionalmente, de cada una de las crónicas se ha puesto un fin de episodio de 864 pasos de 5 minutos (3 días en total) pese a tener una duración total de crónica de 2016 pasos. También, para añadir variabilidad a estos casos, se ha decidido que el comienzo de los episodios sea en distintos puntos de cada crónica, poniendo como saltos de 0 a 4 días del comienzo de la crónica. De esta forma, la combinatoria indicada dará una variación de casos de 18 crónicas y 5 saltos, lo cual suponen 90 casos de los cuales el agente deberá gestionar los 3 días siguientes en lapsos de 5 minutos. La selección del caso a exponer al agente se hará de forma uniforme entre los 90 supuestos y entrenaremos cada uno de los 5 agentes por configuración (para comprobar la estabilidad) con 6M de observaciones.

Una medida adicional de variación de supuestos es el oponente. El oponente es un agente externo no entrenable que se encarga de variar los estados de los elementos de la red según las limitaciones indicadas para aumentar la robustez del agente. En este informe se incluirán los rendimientos de entrenamiento y de test de la opción de incluir un oponente aleatorio que inhabilite líneas de forma aleatoria y el efecto de no incluir dicho oponente.

Por otro lado, a la hora de testear la metodología de entrenamiento propuesta, se pondrá a cada uno de los agentes a testearse contra las crónicas vistas y no vistas durante el entrenamiento (la 17 y la 19) y bajo el supuesto de que existe o no un oponente para comprender la influencia de cada supuesto del entrenamiento para la generalización y robustez del agente objetivo.

3. RESULTADOS OBTENIDOS.

Debido a la complejidad inherente a la gestión de una red eléctrica, una comparativa que tiene sentido es compararse contra la inacción absoluta, ya que en muchos casos es una opción más que viable. Por ello, nuestra comparativa irá siempre acompañada contra agentes "DoNothing" como ejemplo a batir y para dar contexto a nuestros resultados. En nuestros resultados debemos ver si la maximización de la recompensa especificada conlleva una mejora en la longitud de los episodios, comprobando las curvas de recompensa y longitud de episodios tanto en entrenamiento como en test en cada uno de los casos propuestos.

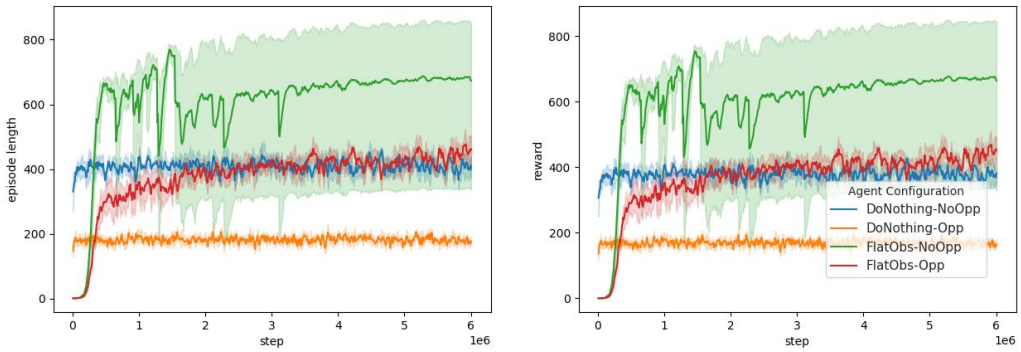

Figura 3. Rendimientos de longitudes de episodios (izquierda) y recompensas (derecha) de los entrenamentos de los agentes "DoNothing" y "FlatObs" con sus variantes con y sin oponente.

Se han ejecutado 5 agentes para las configuraciones de agentes "DoNothing" de referencia y los agentes "FlatObs", tanto con injerencias de un oponente aleatorio en el entrenamiento como sin ellas. Es evidente apreciar que los entrenamientos sin oponentes consiguen unas longitudes y recompensas mayores a costa de ofrecer menos observaciones menos variadas al agente, por lo que se podría caer en un sobreentrenamiento. También se puede observar cómo la presencia de un oponente, pese a obtener resultados no tan abultados como sin la presencia del mismo, parece que estabiliza los rendimientos obtenidos a en el ciclo de vida de dichos agentes. Ahora cabe preguntarse si estas diferencias se mantienen en la ejecución de cada uno de los casos sin influencia de la exploración propia del entrenamiento de agentes de este estilo.

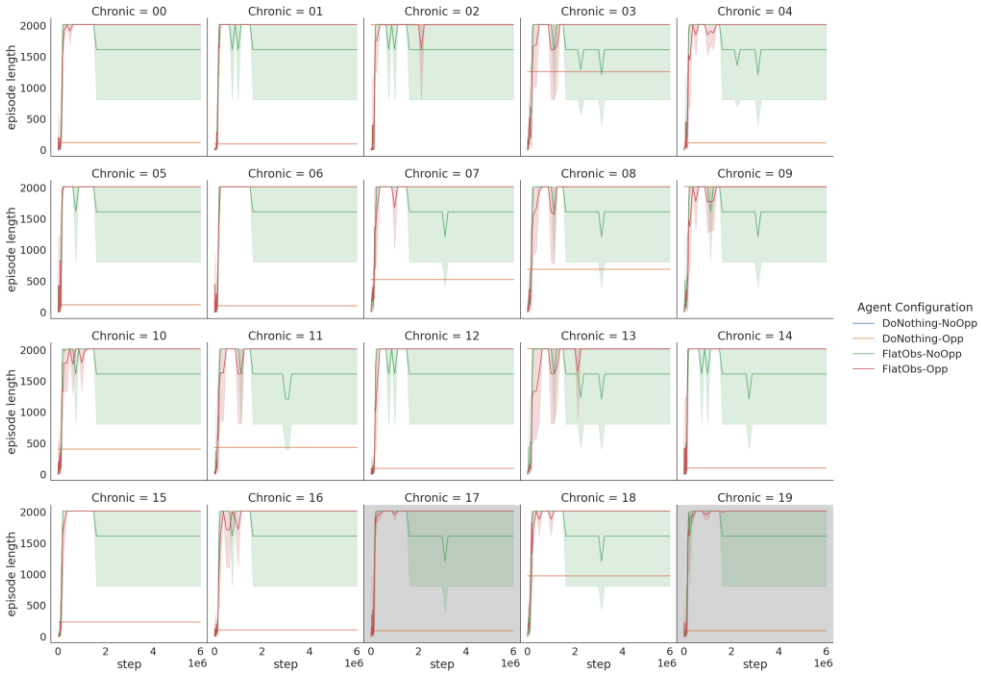

Figura 4. Rendimientos de longitudes de episodios en cada una de las crónicas de los agentes "DoNothing" y "FlatObs" con sus variantes con y sin oponente. Las crónicas 17 y 19 están oscurecidas indicando que ningún agente las ha podido ver durante su entrenamiento.

En la Figura 4 se puede comprobar que en cada uno de los casos (o crónicas) en los que los agentes "DoNothing" conducían a blackout, los que sí que son capaces de entrenar los superan incluso en los casos que no ha visto nunca (crónicas 17 y 19). También, podemos ver cómo las diferencias de rendimiento observadas en el entrenamiento entre

entrenar o no con un oponente no se traducen en una caída de rendimiento del mismo tipo. Por lo tanto, es seguro concluir que la inclusión de un oponente que varíe el entrenamiento ayuda a regularizar los agentes y es una buena práctica que se ha de mantener en futuras investigaciones.

4. CONCLUSIONES Y TRABAJO FUTURO.

Este trabajo presenta una metodología innovadora para automatizar la operación de redes eléctricas mediante agentes autónomos basados en aprendizaje por refuerzo. Se optimiza la toma de decisiones en entornos complejos y dinámicos, mostrando que la inclusión de un oponente aleatorio en el entrenamiento mejora la estabilidad y preparación de los agentes para situaciones imprevistas. En conclusión, este estudio no solo aporta una nueva perspectiva sobre la automatización en la operación de redes eléctricas, sino que también sienta las bases para futuras investigaciones en el campo del aprendizaje por refuerzo aplicado a sistemas complejos, con el objetivo de mejorar la eficiencia y sostenibilidad de la infraestructura energética.

En investigaciones futuras se plantea mejorar la vectorización de la observación en formalizaciones más potentes como puede ser un grafo, así como la adaptación de la arquitectura del modelo para aprovechar esa nueva formalización. Por otro lado, se pretenden explorar nuevas funciones de recompensa que tengan en cuenta más optimizaciones necesarias en una red, así como explorar las implicaciones computacionales de ampliar este método a redes de más subestaciones disponibles en el framework Grid2Op (14 subestaciones, 118 subestaciones, etc.).

5. AGRADECIMIENTOS.

Agradecemos tanto al Instituto de Ingeniería del Conocimiento como a la Universidad Autónoma de Madrid (PID2022-139856NB-I00 fundado por MCIN / AEI / 10.13039 / 501100011033 / FEDER UE) por dejarnos los recursos necesarios para realizar esta investigación, así como por haber recibido su apoyo durante la realización de la misma.

6. REFERENCIAS.

[1] Brockman et al. Openai gym, 2016. CoRR, abs/1606.01540

[2] Huang S., & Ontañón S. ,2020, A Closer Look at Invalid Action Masking in Policy Gradient Algorithms. CoRR, abs/2006.14171.

[3] Marot et al, 2021. Learning to run a Power Network Challenge: a Retrospective Analysis. CoRR, abs/2103.03104.

[4] Schulman et al, 2017,Proximal Policy Optimization Algorithms. CoRR, abs/1707.06347.

[5] B. Donnot. - A testbed platform to model sequential decision making in power systems. https://GitHub.com/rte-france/grid2op, 2020. (11 septiembre 2024)