Latent Structure in Chess Positions

Juli Garbulsky[1] and Juan Pablo Pinasco[1,2]

[1] Universidad de Buenos Aires, Facultad de Ciencias Exactas y Naturales, Departamento de Matemática, Av Cantilo s/n, Pabellón I, Ciudad Universitaria (1428) Buenos Aires, Argentina
[2] Instituto de Investigaciones Matemáticas A. Santaló (IMAS), UBA-CONICET, Av Cantilo s/n, Pabellón I, Ciudad Universitaria (1428) Buenos Aires, Argentina
juliangarbulsky@gmail.com
jpinasco@dm.uba.ar

Abstract. In this work, we use autoencoders to understand latent representations of chess positions. We utilized a database of Lichess games filtered by player rating and time control, and represented each position as a multi-channel tensor. We trained an autoencoder to reconstruct these positions and then analyzed the latent space using Principal Component Analysis (PCA). Our results show that the first components capture relevant game information, such as castling and opening characteristics. We also conducted an experiment based on board symmetries that suggests the model does not memorize positions in a trivial way. These findings indicate that autoencoders can capture meaningful structures in chess and provide an interpretable window into the knowledge of the game.

Keywords: chess, autoencoders, interpretability, game theory

Estructura latente en posiciones de ajedrez

Abstract. En este trabajo entrenamos autoencoders para entender representaciones latentes de posiciones de ajedrez. Utilizamos una base de datos de partidas de Lichess filtradas por nivel de juego y control de tiempo, y representamos cada posición como un tensor de múltiples canales. Entrenamos un autoencoder para reconstruir dichas posiciones y posteriormente analizamos el espacio latente mediante Análisis de Componentes Principales (PCA). Nuestros resultados muestran que las primeras componentes capturan información relevante del juego, tales como el enroque y las características de la apertura. Realizamos también un experimento basado en simetrías del tablero que sugiere que el modelo no memoriza posiciones de forma trivial. Estos hallazgos indican que los autoencoders pueden capturar estructuras significativas en el ajedrez y ofrecen una ventana interpretable al conocimiento del juego.

Keywords: ajedrez, autoencoders, interpretabilidad, teoría de juegos

1 Introducción

El ajedrez es un juego altamente estructurado con una complejidad combinatoria inmensa, y el problema de construir algoritmos que jueguen correctamente, planteado en su momento por Shannon (Shannon, 1950), ha despertado gran curiosidad no sólo de científicos o ajedrecistas, sino también del gran público, con hitos como DeepBlue venciendo a Kasparov (Newborn, 2011), o el estilo de juego tan particular de AlphaZero (Silver et al., 2018).

Sin embargo, existe una diferencia fundamental en cómo los humanos y las máquinas juegan al ajedrez. Los humanos razonan de manera intuitiva y conceptual: un gran maestro no calcula millones de variantes, sino que reconoce patrones aprendidos a lo largo de años de experiencia, identifica debilidades estructurales, evalúa la iniciativa, y anticipa planes a largo plazo. Su pensamiento es holístico y se basa en conceptos abstractos tales como "seguridad del rey", "control del centro" o "ventaja de espacio".

Los programas tradicionales (como Rebel, Stockfish, y otros) operan mediante una búsqueda ordenada en el árbol de movidas posibles, optimizada con métodos como el alpha-beta pruning, guiada por funciones de evaluación que combinan heurísticas explícitas (material, movilidad, estructura de peones, etc.). Aunque extremadamente potentes, su estilo se basa en el poder de cálculo, utilizando el conocimiento humano codificado explícitamente.

En contraste, los programas basados en técnicas modernas de inteligencia artificial (como AlphaZero y Leela Chess Zero) utilizan redes neuronales que internalizan una evaluación implícita de las posiciones y una política de movimientos. Su estilo resulta notablemente más críptico que el de los motores usuales, y realizan muchas veces movidas inexplicables desde la valoración humana tradicional de las posiciones, si bien a largo plazo se encuentran justificaciones basadas en los argumentos clásicos: "dinamismo", "provocar debilidades", "ganar espacio", etc..

Esto nos plantea una pregunta en el espíritu de la Hipótesis de la Representación Platónica (Huh et al., 2024): ¿existe un espacio latente donde las posiciones de ajedrez se agrupan en función de conceptos generales? Dada la amplitud de la misma, y las diversas formas de intentar buscar una respuesta, en este trabajo nos enfocamos en un experimento sencillo: entrenamos un autoencoder con posiciones de la apertura de partidas de ajedrez, buscando que las reconstruya luego de representarlas en un espacio de menor dimensionalidad. El objetivo es explorar si el espacio latente de las representaciones captura características relevantes del juego de manera interpretable, y si emergen conceptos abstractos.

Un análisis de este espacio vía el Análisis de Componentes Principales (PCA) nos muestra que es posible vincular componentes con conceptos ajedrecísticos. Además, vemos que el autoencoder no se limita a memorizar la ubicación de las piezas ya que se observa una distinción entre posiciones reales y posiciones obtenidas reflejando el tablero e intercambiando los colores de las piezas, lo cual nos habla, también, de la asimetría del juego en la etapa inicial de la partida.

El trabajo está organizado de la siguiente manera. En la sección Metodología se describen los datos de entrenamiento, su codificación, y la arquitectura del au-

toencoder. En Resultados analizamos la calidad de la reconstrucción, estudiamos el espacio latente mediante PCA, y discutimos la asimetría de blancas y negras. Finalmente, en Conclusiones discutimos los resultados obtenidos y mencionamos algunas líneas de trabajo futuro.

2 Metodología

Datos. Construimos una base de datos con posiciones extraídas de partidas jugadas en la plataforma Lichess (Lichess, 2026). Se filtraron las partidas considerando únicamente aquellas en las que ambos jugadores tienen un ELO de por lo menos 2100 y con controles de tiempo de al menos 180 segundos, jugadas durante mayo de 2019. Las posiciones corresponden al movimiento 20 (diez de cada jugador), cuando el juego se encuentra todavía en la fase de apertura, descartando partidas que finalizaron en menos movimientos. Luego, eliminamos las posiciones repetidas, para evitar data leakage.

Al aplicar estos filtros quedaron aproximadamente medio millón de posiciones. De estas, se seleccionaron aleatoriamente unas 100.000 posiciones para el entrenamiento, y de las restantes, otras 100.000 para testeo.

Representación de posiciones. Cada posición de ajedrez se representa como un tensor de dimensiones 8 x 8 x 13, donde 12 canales corresponden a un tipo de pieza (6 por color: rey, dama, caballo, alfil, torre, peón) y uno extra indica casillas vacías. En los primeros 12 canales, la celda toma el valor 1 si la pieza correspondiente ocupa esa casilla y 0 en caso contrario, mientras que en el último, la celda toma el valor 1 si la casilla está vacía y 0 en caso contrario. Así, cada casilla tiene exactamente un 1 en algún canal, y tiene 0 en todos los demás. En la Figura 1 se muestra un ejemplo de un tablero y su correspondiente codificación.

Esta codificación no se corresponde con las habituales (Chole et al., 2021, Kapicioglu et al., 2020), y ocupa mucho menos espacio que una imagen, si bien no considera metadata útil que se suele incluir (por ejemplo quién juega, si es posible enrocar, si hay capturas al paso, etc.).

Autoencoder. La arquitectura del autoencoder consiste en:

- Entrada y salida: sus dimensiones coinciden entre ellas y con la del vector de la codificación (832).
- Espacio latente: vector de 20 dimensiones. Codificador: colocamos desde la entrada capas ocultas reduciendo progresivamente su dimensión a la mitad hasta alcanzar la del espacio latente. Es decir, utilizamos cinco capas ocultas de tamaños 416, 208, 104, 52, y 26.
- Decodificador: repetimos la estructura del codificador pero en orden inverso.

Todas las capas son densas (totalmente conectadas), con función de activación ReLU, excepto la última capa del codificador y la última del decodificador, donde se utiliza la función sigmoide. Esto permite obtener salidas acotadas entre

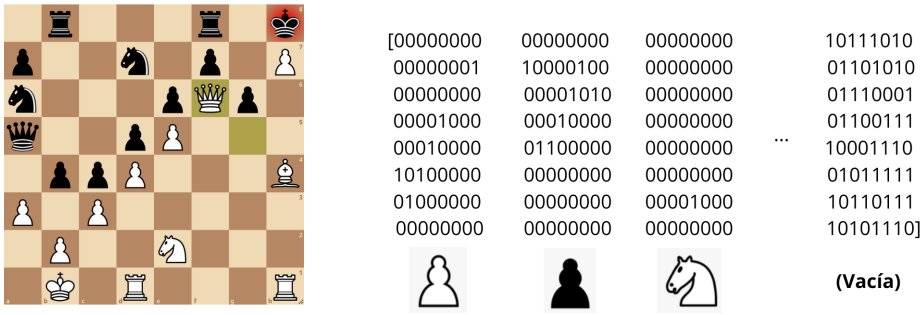

Fig. 1. Ejemplo de la codificación de un tablero de ajedrez.

0 y 1, lo cual es deseable dado que las entradas están compuestas sólo por ceros y unos, y que puede resultar conveniente garantizar que la representación en el espacio latente sea acotada.

3 Resultados

3.1 Calidad de la reconstrucción

Para la reconstrucción, analizamos en cada casilla qué canal arrojó un valor más alto de activación en la salida del decodificador, y se coloca la pieza (o el lugar vacío) correspondiente a ese canal. Evaluamos el modelo entrenado sobre unas 100.000 posiciones de testeo para estimar la capacidad de generalización del autoencoder a datos no vistos. Para cuantificar la calidad de la reconstrucción, consideramos dos métricas complementarias.

En primer lugar, utilizamos el error cuadrático medio (MSE, por sus siglas en inglés), calculando la norma 2 para la diferencia entre la codificación de la posición de entrada y la salida del decodificador.

En segundo lugar, promediamos el número de casillas mal reconstruidas (NCMR) para cada tablero, contando el número de casillas donde difieren la entrada y la reconstrucción de la salida. Esta métrica resulta particularmente relevante en el contexto del ajedrez, ya que captura errores en la ubicación de las piezas más allá de la magnitud del error numérico. Los valores obtenidos fueron:

- MSE: 0.01466.
- NCMR: 7.55.

3.2 Interpretación del espacio latente

Para estudiar la estructura interna del espacio latente generado por el encoder, aplicamos Análisis de Componentes Principales (PCA) sobre las representaciones latentes de las posiciones de testeo. Esta técnica permite proyectar los vectores latentes en componentes ortogonales ordenadas según la varianza que explican. Si bien originalmente los autoencoders se consideraron una alternativa a PCA para aprender el espacio latente de un conjunto de datos dado Baldi and Hornik, 1989, a su vez PCA resultó ser una herramienta clave en interpretabilidad del espacio latente Huben et al., 2024; Magri and Doan, 2022.

Exploramos la interpretabilidad mediante la visualización de las representaciones latentes proyectadas sobre las dos primeras componentes. En particular, consideramos la posición de los reyes blanco y negro, distinguiendo entre configuraciones asociadas al enroque corto, la posición inicial y otras configuraciones.

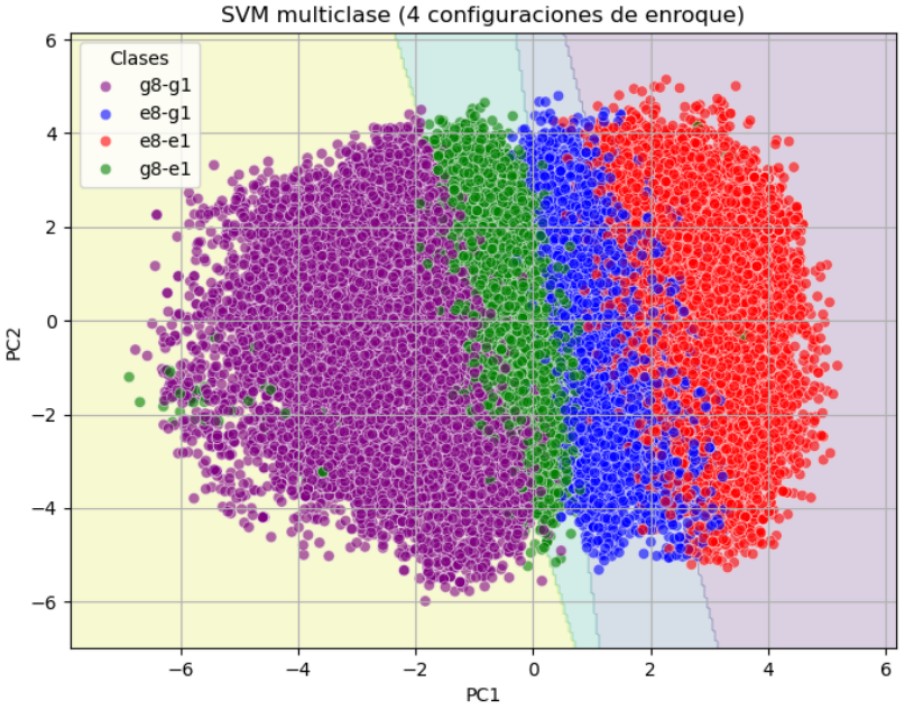

Fig. 2. Proyección en las dos primeras componentes de las posiciones con los reyes en su casilla original o en la del enroque corto. Fondo: separación de las clases utilizando SVM.

La Figura 2 muestra la proyección de las posiciones en el plano definido por las dos primeras componentes principales, coloreadas según cinco clases que

combinan la ubicación de ambos reyes: (e8, e1), (e8, g1), (g8, e1), (g8, g1), y un conjunto residual que agrupa el resto de las configuraciones. Se observa que las primeras cuatro clases, correspondientes a configuraciones típicas de los reyes en las casillas del enroque corto o en sus casillas originales, definen regiones bien diferenciadas.

Table 1. Classification report del modelo SVM entrenado sobre los datos de testeo.

	Precision	**Recall**	**F1**	**Support**
e8-e1	0.86	0.91	0.89	21276
e8-g1	0.82	0.76	0.79	14767
g8-e1	0.86	0.81	0.84	15598
g8-g1	0.94	0.96	0.95	36608
Accuracy			0.89	88249

Para confirmar esta separación, entrenamos un clasificador SVM multiclase sobre estas representaciones bidimensionales, considerando únicamente las cuatro clases principales. El modelo logra una alta precisión de clasificación (ver Tabla 1), lo que indica que la información relativa al estado de enroque de ambos jugadores es en gran medida linealmente separable en las primeras dos componentes principales.

Finalmente, en la Figura 3 mostramos las posiciones clasificadas según la apertura jugada. Se observa una clara división en tres grandes grupos:

- posiciones con e4-e5, típicas de aperturas abiertas (tales como Ruy López, Italiana, Sicilianas abiertas) con un coeficiente negativo en la segunda coordenada,
- posiciones donde el negro realiza un fianchetto de rey, y sólo un bando ocupa el centro, típico de defensas indias (India de Rey, Pirc, Benoni, Holandesa) con un coeficiente negativo de la primera coordenada,
- posiciones con d4-d5, típicas de aperturas cerradas y defensas semiabiertas (Peón Dama, Eslava, Caro-Kann, Francesa) con un coeficiente positivo en la segunda coordenada.

3.3 Asimetría blancas-negras

Con el objetivo de evaluar si el autoencoder memoriza sólo la ubicación de las piezas, o si, por el contrario, aprende regularidades estructurales del juego, realizamos un experimento basado en simetrías del tablero: generamos nuevas instancias mediante la reflexión vertical del tablero y el intercambio de colores de las piezas. Evaluamos la capacidad de reconstrucción del modelo sobre estas posiciones transformadas y comparamos con los resultados obtenidos sobre las posiciones originales. El desempeño del modelo difiere entre ambos conjuntos:

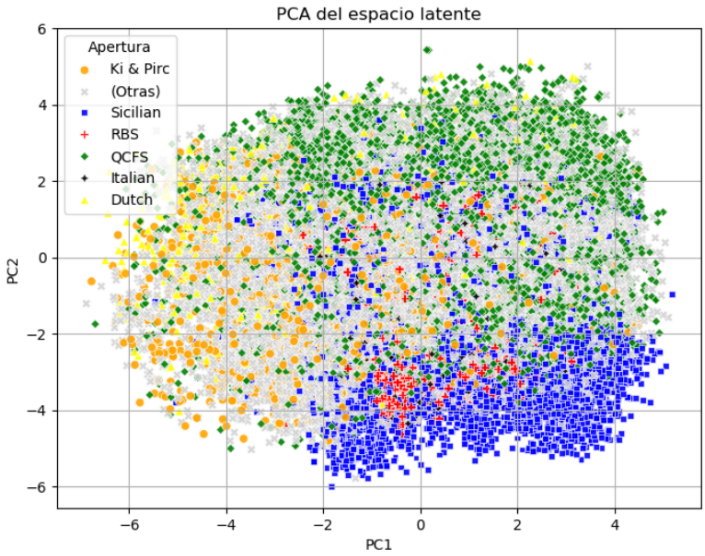

Fig. 3. Proyección en las dos primeras componentes de las posiciones indicando la apertura o defensa jugada. QCFS indica Peón Dama, Caro-Kann, Francesa, y Eslava. RBS indica Ruy Lopez, Berlin y Escocesa. Ki indica India de Rey.

se obtuvo un MSE de 0.0175 y un NCMR de 9.17. La diferencia es significativa, ya que el estadístico que se obtiene para un test asintótico para las medias del MSE es T = 129.3, lo cual indica que aprende regularidades en las posiciones.

4 Conclusiones

En este trabajo mostramos que los autoencoders pueden aprender representaciones latentes significativas de posiciones de ajedrez. El análisis mediante PCA permite identificar componentes interpretables, como el enroque y características de la apertura, y revela un espacio latente que captura conceptos estratégicos del juego.

El experimento basado en simetrías sugiere, por un lado, que el ajedrez presenta asimetrías inherentes, consistentes con el hecho de que las blancas y las negras no desempeñan roles equivalentes en la partida, y, por otro, que el autoencoder no se limita a memorizar la ubicación de las piezas, sino que captura aspectos posicionales más profundos.

El análisis de las componentes principales también refleja, en cierta medida, la configuración de los peones centrales, y del desarrollo de las piezas. Si bien este aspecto no se explora en detalle en el presente trabajo, sugiere que el modelo representa posiciones basándose en distintos factores fundamentales del ajedrez para conseguir una reducción de la dimensión. En el espíritu de la Hipótesis

de la Representación Platónica (Huh et al., 2024), el espacio latente aprendido por el autoencoder representa una aproximación al *espacio latente del ajedrez*, entendido como la clasificación ideal de posiciones en términos conceptuales. Esta convergencia se evidencia en que las direcciones principales pueden interpretarse utilizando conceptos desarrollados por los jugadores humanos (importancia del enroque, formas de ocupar o atacar el centro, desarrollo de piezas), sugiriendo que existe un modelo estadístico común de *qué define un tipo de posición*.

Podría argumentarse que la restricción a cierto momento de la partida, o a un rango de ELO específico, son limitaciones de este trabajo. Para la primera, observemos que los conceptos críticos de una posición de ajedrez dependen mucho de la altura de la partida a la que está. Esperaríamos una representación completamente diferente de la posición en el medio juego, donde ya se han definido estructuras de peones y existen temas clásicos que catalogan las posiciones (peones aislados, doblados, colgantes, pareja de alfiles), como así también el dominio de columnas y diagonales. Lo mismo ocurre en el final, típicamente clasificado en función de los tipos de piezas presentes. Respecto al rango de ELO, experimentos en Garbulsky, 2025 para posiciones que provienen de partidas de jugadores con bajo ELO, mostraron valores más altos del MSE y NCMR.

Como trabajo futuro nos interesa explorar la reducción de dimensionalidad del espacio latente utilizando sólo PCA sobre los vectores obtenidos al codificar, sin pasar por un autoencoder. Los experimentos muestran que las componentes decaen linealmente, aportando todas prácticamente la misma información. Sin embargo, utilizando `kernelPCA` se obtiene un decaimiento no lineal, a costa de perder interpretabilidad por el uso de kernels. Vale destacar que la codificación elegida mapea las posiciones a vectores con exactamente 64 unos, con lo cual viven en la intersección de la superficie de la bola euclídea de radio 8, con la superficie de la bola asociada a la norma $\|cdot\|_1$ de radio 64.

Por otro lado, nos proponemos explorar modelos más complejos, considerando autoencoders específicos para niveles de ELO, y para las distintas fases del juego. En el espíritu de Omori and Tadepalli, 2024; Tijhuis et al., 2023 consideramos que podría utilizarse para predecir el ELO de un jugador, según los errores en la reconstrucción. Más ambicioso, nos proponemos estudiar la dinámica temporal de las partidas mediante secuencias de posiciones en un único espacio latente, y analizar los tipos de trayectorias.

Acknowledgments. Agradecemos al Ing. M. Bergerman su sugerencia de codificar los espacios en blanco, que resuelve elegantemente el problema de reconstruir la posición según la máxima activación. Agradecemos también a Lichess por compartir las partidas jugadas en su plataforma. Este trabajo está parcialmente financiado por PIP 11220200102851CO-CONICET.

References

Baldi, P., & Hornik, K. (1989). Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, *2*(1), 53–58.

Chole, V., Gote, N., Ujwane, S., Hedaoo, P., & Umare, A. (2021). Design and development of game playingsystem in chess using machine learning. *International Research Journal of Engineering and Technology*, *8*.

Garbulsky, J. (2025). Ajedrezzinni latentinni: Autoencoders para interpretar el espacio latente del ajedrez [Last accessed 2026/06/26]. https://lcd.exactas.uba.ar/ajedrezzinni-latentinni-autoencoders-para-interpretar-el-espacio-latente-del-ajedrez/

Huben, R., Cunningham, H., Smith, L., Ewart, A., & Sharkey, L. (2024). Sparse autoencoders find highly interpretable features in language models. *International Conference on Learning Representations*, *2024*, 7827–7845.

Huh, M., Cheung, B., Wang, T., & Isola, P. (2024). The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*.

Kapicioglu, B., Iqbal, R., Koc, T., Andre, L. N., & Volz, K. S. (2020). Chess2vec: Learning vector representations for chess. *arXiv preprint arXiv:2011.01014*.

Lichess. (2026). Lichess.org [Last accessed 2026/06/26]. https://lichess.org

Magri, L., & Doan, A. K. (2022). On interpretability and proper latent decomposition of autoencoders. *arXiv preprint arXiv:2211.08345*.

Newborn, M. (2011). *Beyond deep blue: Chess in the stratosphere*. Springer Science & Business Media.

Omori, M., & Tadepalli, P. (2024). Chess rating estimation from moves and clock times using a cnn-lstm. *International Conference on Computers and Games*, 3–13.

Shannon, C. E. (1950). Xxii. programming a computer for playing chess. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *41*(314), 256–275.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, *362*(6419), 1140–1144.

Tijhuis, T., Blom, P. M., & Spronck, P. (2023). Predicting chess player rating based on a single game. *2023 IEEE Conference on Games (CoG)*, 1–8.