

Análisis Arquitectónico y Experimental de un Clasificador de Posiciones de Ajedrez Basado en Redes Neuronales Convolucionales (CNN)

Por: Jonathan Betancur, Samuel Rendón, Valentina Morales.

1. Introducción y Fundamentos de las Redes Neuronales Convolucionales (CNN)

1.1. Contexto y la Necesidad de Redes Especializadas para Estructuras Tensoriales

El campo del *Deep Learning* ha sido transformado por la capacidad de las Redes Neuronales Profundas (DNN) para procesar datos complejos y de alta dimensión. Específicamente, la Red Neuronal Convolucional (por sus siglas, CNN) surgió como la arquitectura fundamental y el paradigma dominante para el análisis de información que puede ser organizada en una estructura de cuadrícula o tensorial, lo que incluye imágenes, pero también otras formas de datos estructurados espacialmente.

Una CNN es un tipo especializado de algoritmo de aprendizaje profundo diseñado para analizar datos en formato de cuadrícula, imitando la estructura y el funcionamiento jerárquico de la corteza visual humana. Esta arquitectura se distingue por su rendimiento superior con entradas que exhiben dependencias espaciales o de vecindad, como ocurre al codificar la Notación Forsyth-Edwards (por sus siglas: FEN) de un tablero de ajedrez en un tensor de características

1.2. Componentes Arquitectónicos Clave y su Significado Matemático en el Contexto FEN

La elección de la CNN se basa en su estructura especializada para procesar datos de tipo cuadrícula, incluso cuando la entrada inicial es texto en formato FEN, que se transforma en un mapa de características del tablero de 8x8 casillas. Una CNN se compone típicamente de tres tipos principales de capas que trabajan en conjunto para aprender los patrones de posición del ajedrez

1.2.1. *Convolutional Layer*

La capa convolucional constituye el núcleo de la arquitectura. En lugar de escanear píxeles o imágenes, los filtros (*kernels*) se aplican sobre la codificación tensorial del tablero para extraer patrones de vecindad posicional. Estos filtros realizan la operación de convolución para generar mapas de características

El propósito principal es extraer características pertinentes de la posición, como la detección de la estructura de peones, el ataque/defensa entre piezas adyacentes, o el control central. En el contexto del ajedrez, las capas iniciales identifican rasgos posicionales simples (ej., un peón en la quinta fila), mientras que las capas más profundas sintetizan estos rasgos en conceptos de alto nivel, discriminando entre posiciones que tienen

una ventaja decisiva frente a aquellas que están en un equilibrio complejo.

1.2.2. ReLU

Tras cada operación de convolución, se aplica la Función de Activación de Unidad Lineal Rectificada (ReLU) al mapa de características, introduciendo la no linealidad en el modelo. Esto es vital, ya que permite que la red aprenda mapeos complejos y modele las relaciones estratégicas sofisticadas en el ajedrez que no son linealmente separables.

1.2.3. Pooling Layer

La capa de agrupación (comúnmente Max-Pooling) reduce la dimensionalidad espacial de los mapas de características. Aunque reduce la información de posición precisa, minimiza la complejidad computacional, mejora la eficiencia y limita el riesgo de sobreajuste (*overfitting*). En el tablero, la capa de *pooling* podría ayudar a la red a enfocarse en la característica más relevante de una región determinada (ej., la pieza más fuerte o el punto de ataque/defensa principal).

1.2.4. Fully Connected Layers

Las capas finales (FC) reciben los vectores de características de alto nivel extraídos por las capas convolucionales y de *pooling*. Su función es mapear estas características al espacio de salida final, produciendo las probabilidades para las clases de evaluación objetivo: Ventaja, Empate o Desventaja.

2. Fundamentos del Entrenamiento de Redes Neuronales Profundas

El entrenamiento de la CNN busca el conjunto óptimo de pesos y sesgos que minimizan el error de predicción. Este proceso se articula a través de la propagación hacia adelante, la medición del error, la propagación hacia atrás y el ajuste de parámetros.

2.1. El Ciclo de Aprendizaje: Propagación Hacia Adelante y Hacia Atrás

2.1.1. Forward Pass

El proceso inicia con el paso hacia adelante, donde los datos de entrada (la codificación tensorial de la posición FEN) atraviesan la red, generando una predicción de salida (probabilidades para Ventaja, Empate, Desventaja).

2.1.2. Medición del Error: Loss Function

La función de pérdida cuantifica la discrepancia entre la salida prevista de la red y la etiqueta de verdad (el valor real, es decir, si la posición fue clasificada como Ventaja, Empate o Desventaja). Esta función produce un valor escalar que guía el algoritmo de optimización. Para este problema de clasificación multicategoría, la función de Entropía Cruzada (*Cross-Entropy Loss*) es la elegida, ya que mide la diferencia entre las probabilidades de clase predichas y las etiquetas de clase verdaderas.

2.1.3. Propagación Hacia Atrás

Una vez calculado el error, el algoritmo de retropropagación propaga este error hacia atrás a través de las capas de la red. Aplicando la regla de la cadena, se determinan las derivadas parciales de la función de pérdida con respecto a cada peso y sesgo en la red. Este cálculo indica cuánto contribuyó cada parámetro al error total, facilitando el ajuste.

2.2. El Rol del Optimizador y el Descenso de Gradiente

El optimizador actualiza los parámetros del modelo de forma iterativa para minimizar la función de pérdida. El concepto central de estos algoritmos es el Descenso de Gradiente (*Gradient Descent*, GD).

El Descenso de Gradiente ajusta los pesos y sesgos de la red en la dirección que reduce más rápidamente el error. El Mini-batch Gradient Descent es la implementación más común, ya que utiliza un subconjunto más pequeño de datos (bien llamado *mini-batch*) en cada iteración, ofreciendo una velocidad de convergencia superior al Descenso de Gradiente por lotes (BGD) y mayor estabilidad que el Descenso de Gradiente Estocástico (SGD).

3. Justificación Arquitectónica: CNN frente a Otros Paradigmas de DNN

Para una tarea de evaluación posicional basada en la estructura del tablero (FEN), la selección de la arquitectura CNN sobre otras Redes Neuronales Profundas (DNN) se justifica por su capacidad inherente para explotar la estructura espacial y local del ajedrez.

3.1. Perceptrones Multicapa (MLP) y su Inadecuación Espacial

Los Perceptrones Multicapa son inadecuados para esta tarea porque requieren aplanar la codificación tensorial FEN en un vector único. Al hacer esto, se pierde por completo la estructura espacial bidimensional del tablero y la información de vecindad de las celdas.

Además, un MLP sufre de un número excesivo de parámetros debido a que cada nodo de la capa de entrada estaría conectado a la primera capa oculta. La CNN supera esto mediante la reutilización de pesos.

3.2. Redes Recurrentes (RNN) y la Dimensión Temporal

Aunque el ajedrez es secuencial en movimientos, la clasificación de una posición estática es un problema puramente espacial. Las Redes Neuronales Recurrentes están diseñadas para datos secuenciales. Aplicar una RNN a una posición FEN requeriría una vectorización secuencial (por ejemplo, de posiciones de A1 hasta H8) que no está optimizada para capturar la interacción espacial simultánea de todas las piezas, una propiedad que la CNN maneja eficientemente.

3.3. Ventajas Intrínsecas de la CNN para Datos de Cuadrícula

La CNN capitaliza las propiedades de los datos de cuadrícula de manera eficiente:

1. Reutilización de Pesos: Un único filtro convolucional se aplica a todas las regiones del tablero codificado. Esto significa que si un filtro aprende un patrón estratégico específico (ej., un rey enrocado seguro) en un flanco, esos mismos pesos se reutilizan para detectar patrones similares en cualquier

otra parte del tablero. Esta propiedad reduce drásticamente el número de parámetros a entrenar.

2. Localidad de Conexiones: Las neuronas de las capas convolucionales se conectan solo a una pequeña región espacial en un parche de la matriz FEN codificada. Esto permite que el modelo se enfoque en características relevantes que existen dentro de la vecindad inmediata de una pieza, capturando interacciones locales.

Tabla 1: Comparativa Estructural y Aplicativa de Arquitecturas DNN

Arquitectura	Tipo de Dato Óptimo	Manejo de Localidad Espacial	Invariancia a Traslación	Densidad de Conexiones
Perceptrón Multicapa (MLP)	Datos tabulares, Vectores planos	Nulo	Baja/Nula	Alta (Totalmente conectada)
Red Neuronal Convolucional (CNN)	Datos de tipo cuadrícula (FEN codificado)	Alta (Filtros compartidos)	Inherente (Por Pooling) ²	Baja (Conexiones locales)
Red Neuronal Recurrente (RNN)	Datos Secuenciales (Texto, Series Temporales)	No aplicable	N/A	Variable

4. Aplicación Específica: El Clasificador CNN de Posiciones FEN

El objetivo fundamental del modelo CNN es la Evaluación Global de Posición, lo que implica clasificar una posición FEN de ajedrez en una de las tres clases estratégicas: Ventaja, Empate o Desventaja.

Este modelo opera transformando primero la notación FEN en una representación tensorial del tablero, que es la entrada que procesan las capas convolucionales. Para determinar las etiquetas de verdad, se requiere un criterio robusto. La literatura sobre motores de ajedrez utiliza el concepto de Evaluación de Centipeón (scp) como medida de valor posicional.⁶

El etiquetado de las clases (Ventaja, Empate, Desventaja) se establece comúnmente mediante umbrales en la evaluación de centipeón:

- Ventaja: Posiciones donde la evaluación scp es superior a 1.5
- Desventaja: Posiciones donde la evaluación scp es inferior a -1.5
- Empate: Posiciones donde la evaluación scp se encuentra entre -1.5 y 1.5.

Este proceso permite transformar una métrica continua (scp) en un problema de clasificación discreta, haciendo que el modelo se enfoque en el juicio estratégico del estado de la partida, en lugar de una evaluación numérica exacta.

5. Resultados Experimentales, Hallazgos Cruciales y Desafíos

5.1. Desempeño General y Análisis de Métricas

Los resultados experimentales del clasificador de posiciones FEN, evaluados con un conjunto de datos total de 42240 posiciones, indican una precisión global del 68%. A continuación, se presenta el reporte de clasificación detallado:

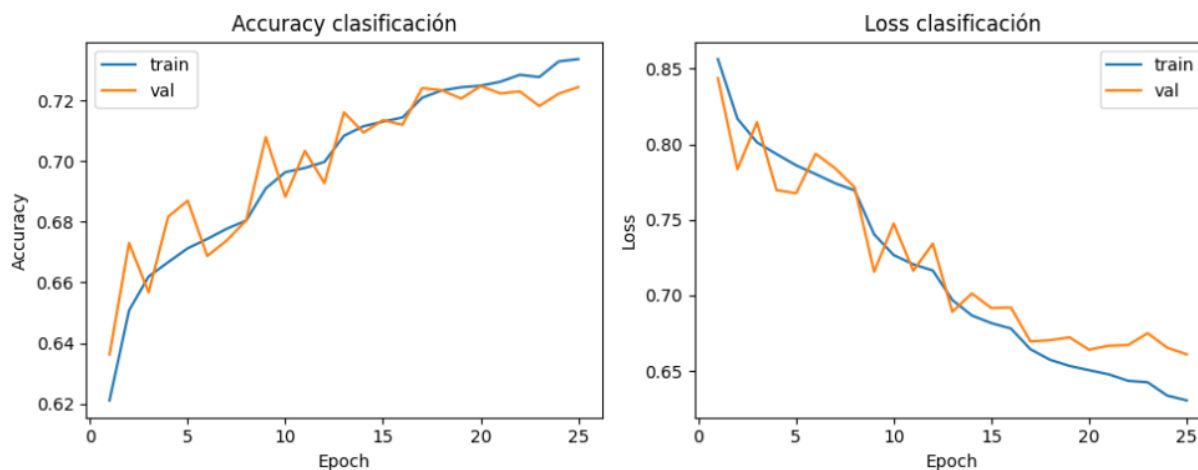


Tabla 2: Reporte de Clasificación (Ventaja, Empate, Desventaja)

Clase	precision	recall	f1-score	support
Ventaja	0.68	0.54	0.60	12650
Empate	0.72	0.81	0.76	20176
Desventaja	0.60	0.62	0.61	9414
accuracy			0.68	42240

macro avg	0.67	0.66	0.66	42240
weighted avg	0.68	0.68	0.68	42240

	precision	recall	f1-score	support
Ventaja	0.68	0.54	0.60	12650
Empate	0.72	0.81	0.76	20176
Desventaja	0.60	0.62	0.61	9414
accuracy			0.68	42240
macro avg	0.67	0.66	0.66	42240
weighted avg	0.68	0.68	0.68	42240

5.2. Hallazgo Crítico: El Desequilibrio de Clases

El análisis de la tabla revela un hallazgo crucial relacionado con la distribución y el rendimiento de las clases:

1. Dominio de la Clase "Empate": La clase "Empate" es la mayoritaria, con 20176 posiciones de soporte, lo que representa aproximadamente el 47.7% del conjunto de datos de validación.
2. Sesgo del Modelo: El modelo muestra su mejor desempeño en esta clase, con la precisión 0.72 y el *recall* 0.81 más altos, resultando en un excelente F1-score de 0.76.
3. El alto *recall* 0.81 significa que el modelo es muy bueno para identificar correctamente las posiciones de empate, lo que probablemente contribuye a inflar la precisión global del 68%
4. Dificultad en Clases Minoritarias: Las clases "Ventaja" y "Desventaja" son minoritarias y muestran un rendimiento significativamente menor.
 - La clase Desventaja es la más pequeña con 9414 posiciones y es la que peor predice, con la *precisión* más baja 0.60 y un F1-score de 0.61. Esto indica que el modelo a menudo clasifica erróneamente las posiciones de desventaja como "Empate", o que tiene dificultades para aprender las características únicas de una posición perdedora.
 - La clase Ventaja muestra un *recall* particularmente bajo 0.54, lo que significa que de todas las posiciones que realmente son Ventaja, solo la mitad son identificadas correctamente por el modelo.

Conclusión del Hallazgo: El principal desafío no es la capacidad de la CNN para procesar la estructura FEN, sino su sensibilidad al desequilibrio de clases. La red tiene una fuerte tendencia a predecir la clase mayoritaria ('Empate'), lo que resulta en un buen rendimiento general, pero un reconocimiento deficiente de las situaciones de ventaja o desventaja.

6. Conclusiones y Alcances Futuros del Modelo CNN

6.1. Transición de la Clasificación a la Evaluación Posicional

El modelo ha logrado un punto de partida viable con una precisión del 68% en la evaluación de posición, demostrando que la arquitectura CNN es adecuada para la extracción de patrones espaciales a partir de la codificación FEN. Sin embargo, para que el modelo pase de ser un clasificador a un **evaluador posicional competente**, es necesario mejorar la precisión en las clases minoritarias de "Ventaja" y "Desventaja".

6.2. Estrategias Futuras para Mejorar la Robustez y Contextualización

Para abordar la debilidad ante el desequilibrio de clases y optimizar el juicio estratégico, se proponen las siguientes líneas de trabajo futuro:

6.2.1. Mitigación del Desequilibrio de Clases

Es imperativo aplicar técnicas para mejorar el reconocimiento de las clases minoritarias:

- Ponderación de Clases: Asignar un peso mayor en la función de pérdida a los errores de clasificación de las clases "Ventaja" y "Desventaja". Esto penaliza más al modelo cuando se equivoca en las clases menos frecuentes.
- Estrategias de Muestreo: Utilizar sobremuestreo de las clases minoritarias, es decir: ventaja y desventaja) o submuestreo de la clase mayoritaria, osea, empate, para equilibrar la distribución de soporte en el conjunto de entrenamiento.

6.2.2. Integración de *Features* Auxiliares con *Input Engineering*

A pesar de la codificación FEN, la CNN se beneficia de *features* que codifican el contexto estratégico global.

- Ventaja Material: La diferencia de material es un factor primordial en ajedrez. El modelo debe ser complementado con la inyección de la diferencia de material (ej., el delta de la suma de valores de las piezas) en las capas totalmente conectadas finales. Esto ayuda a la red a tomar decisiones basadas en el factor estratégico más básico.
- Información Meta-FEN: Incorporar información auxiliar extraída directamente de la notación FEN, como el enroque disponible, el turno a mover, o la existencia de peones pasados.

6.2.3. Exploración de Arquitecturas Avanzadas para Dependencias de Largo Alcance

Aunque la CNN maneja la localidad, la evaluación de piezas como la Dama, Torre y Alfil depende de la influencia que ejercen a lo largo de todo el tablero. Para capturar mejor estas dependencias en el tensor FEN, se recomienda explorar arquitecturas más profundas y avanzadas.

- Redes con Conexiones Residuales: Estas redes permiten que la información fluya sin obstáculos a través de muchas capas, facilitando la propagación de gradientes y la extracción de patrones posicionales muy profundos.
- Mecanismos de Atención: Estos mecanismos permiten a la red ponderar la importancia de las celdas distantes, simulando de manera efectiva la capacidad de las piezas globales para influir en todo el tablero, lo que es esencial para una evaluación posicional precisa.

6.3. Desafíos de Escalabilidad y Búsqueda

Dada la inmensidad del espacio de posibles posiciones de ajedrez 10^{40} , el clasificador CNN debe

integrarse en un sistema de búsqueda más amplio. Al proporcionar una evaluación heurística de posición rápida (Ventaja, Empate, Desventaja), la CNN puede funcionar como una función de evaluación eficiente dentro de un motor de ajedrez, optimizando la búsqueda y haciendo el análisis estratégico más factible.

Citas

1. [P] chess-cv: Clasificador de piezas de ajedrez basado en CNN - Reddit, fecha de acceso: noviembre 6, 2025, https://www.reddit.com/r/MachineLearning/comments/1nywei8/p_chesscv_cnnbased_chess_piece_classifier/?tl=es-419
2. Visión por Computadora + TensorFlow CNN para detectar piezas de ajedrez de capturas de pantalla de tableros de ajedrez en línea : r/MachineLearning - Reddit, fecha de acceso: noviembre 6, 2025, https://www.reddit.com/r/MachineLearning/comments/45ucs3/computer_vision_tensorflow_cnn_to_detect_chess/?tl=es-419
3. Masterclass: IA (Inteligencia Artificial) en ajedrez - YouTube, fecha de acceso: noviembre 6, 2025, <https://www.youtube.com/watch?v=QBZyXK3JNVs>
4. Tipos de posiciones en Ajedrez y cómo jugarlas, fecha de acceso: noviembre 6, 2025, <https://ajedrezpractico.com/estrategia/tipos-de-posiciones/>
5. Predicting Moves in Chess using Convolutional Neural Networks - CS231n, fecha de acceso: noviembre 6, 2025, <https://cs231n.stanford.edu/reports/2015/pdfs/ConvChess.pdf>
6. Advantages of Recurrent Neural Networks over basic Artificial Neural Networks, fecha de acceso: noviembre 6, 2025, <https://datascience.stackexchange.com/questions/32169/advantages-of-recurrent-neural-networks-over-basic-artificial-neural-networks>
7. ¿Qué es la retropropagación? - IBM, fecha de acceso: noviembre 6, 2025, <https://www.ibm.com/es-es/think/topics/backpropagation>
8. Convolutional Neural Networks (CNN) (Image Classification) - YouTube, fecha de acceso: noviembre 6, 2025, <https://www.youtube.com/watch?v=ns2L2T6wvAY>
9. Computer Vision - MATLAB & Simulink - MathWorks, fecha de acceso: noviembre 6, 2025, <https://la.mathworks.com/discovery/computer-vision.html>
10. When to use MLP, CNN or RNN?. When to Use Multilayer Perceptrons? | by Vatsal Raval, fecha de acceso: noviembre 6, 2025, <https://medium.com/@iamvatsalraval/when-to-use-mlp-cnn-or-rnn-4aadba56073e>