Feature set analysis for chess ЭUИИ networks

Tesis de Licenciatura

Martín Emiliano Lombardo

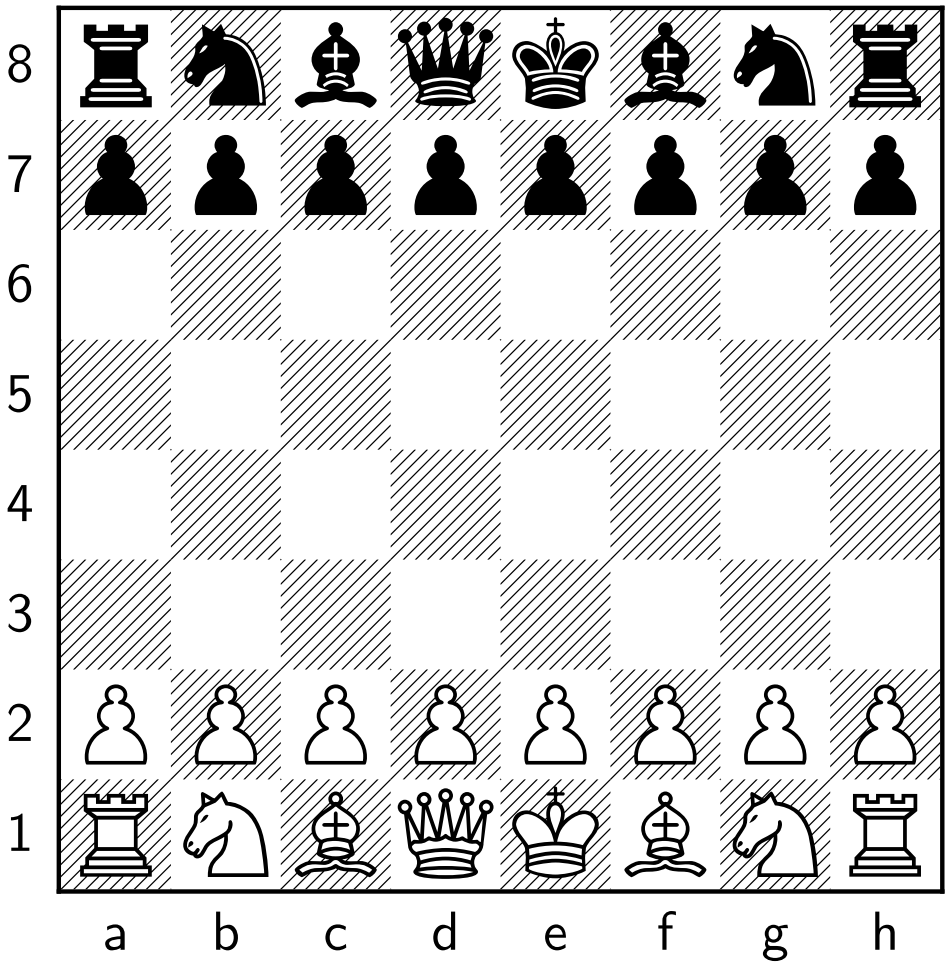
Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

2024

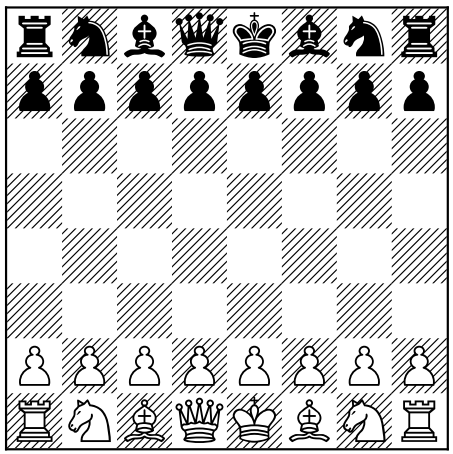
Introducción

Ajedrez

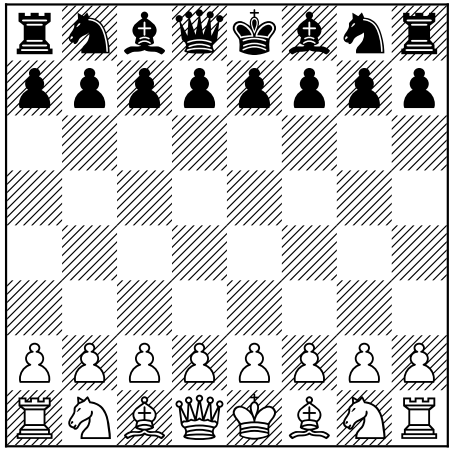
- Dos jugadores
- Suma cero

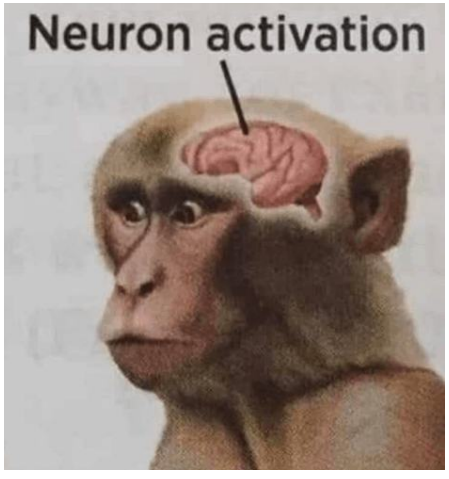


Humano vs. Computadora



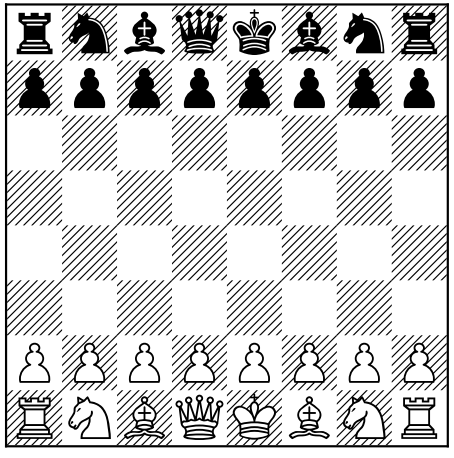
Humano vs. Computadora

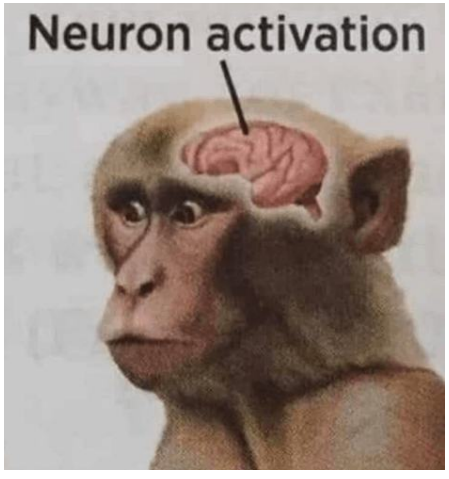




e2e4

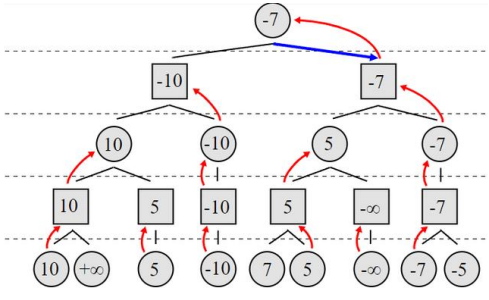
Humano vs. Computadora





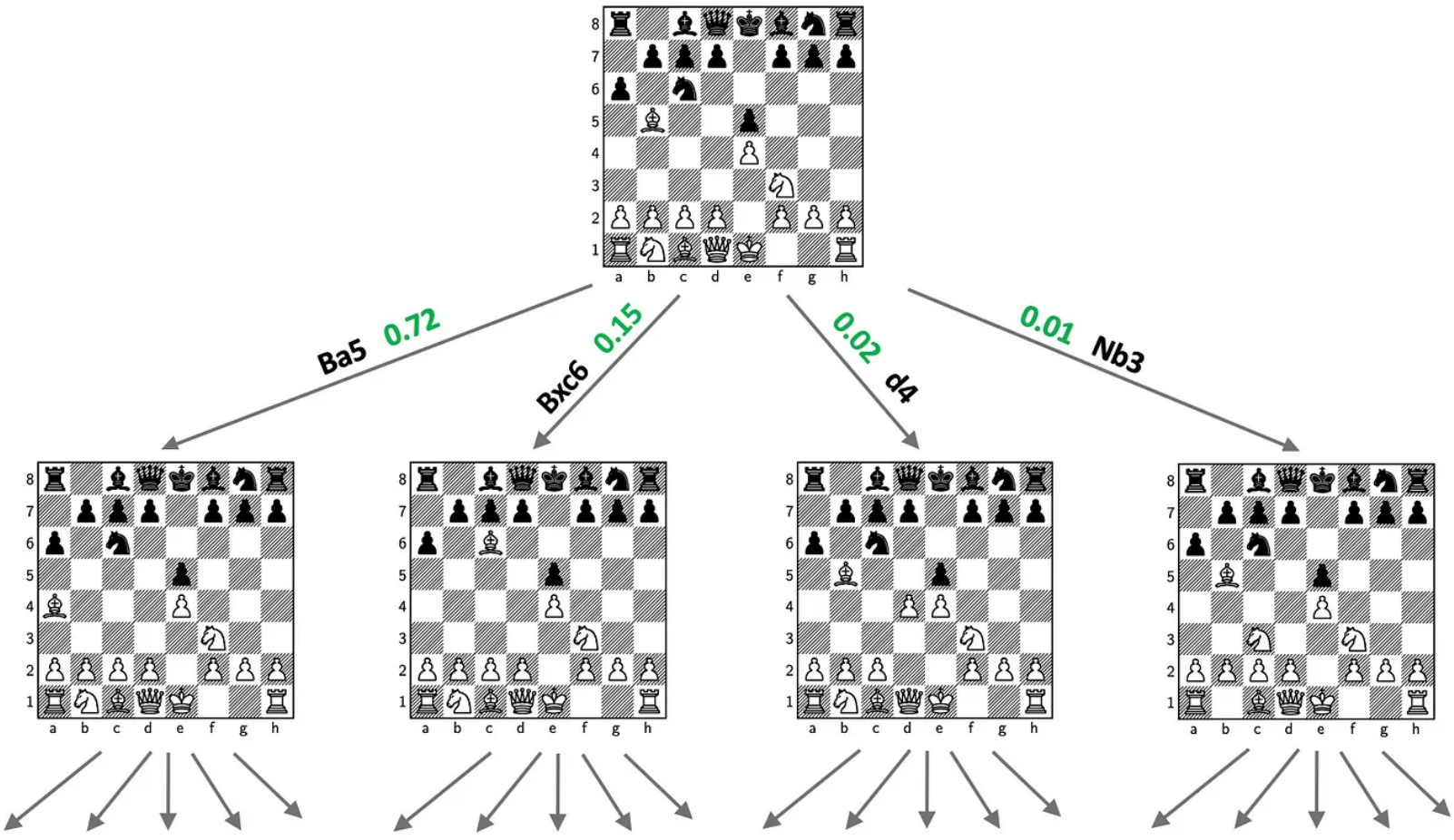
e2e4

Chess Engine



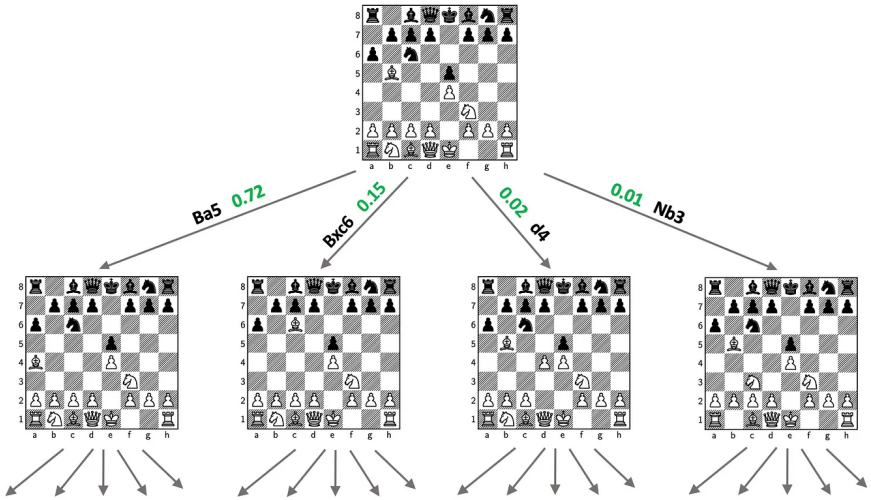
e2e4

Ajedrez como árbol



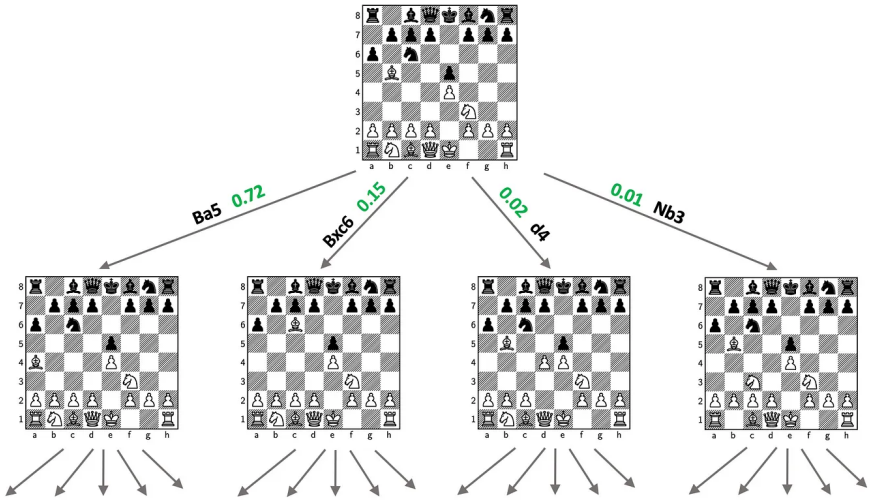
Motores de ajedrez (Chess Engines)

- Exploran el árbol de juego (Minimax, MCTS, etc.)



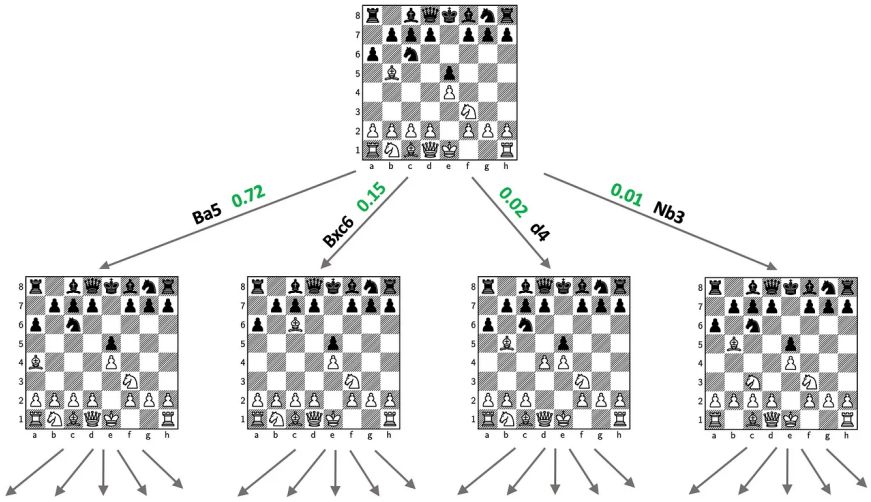
Motores de ajedrez (Chess Engines)

- Exploran el árbol de juego (Minimax, MCTS, etc.)
- Utilizan funciones de evaluación en las hojas

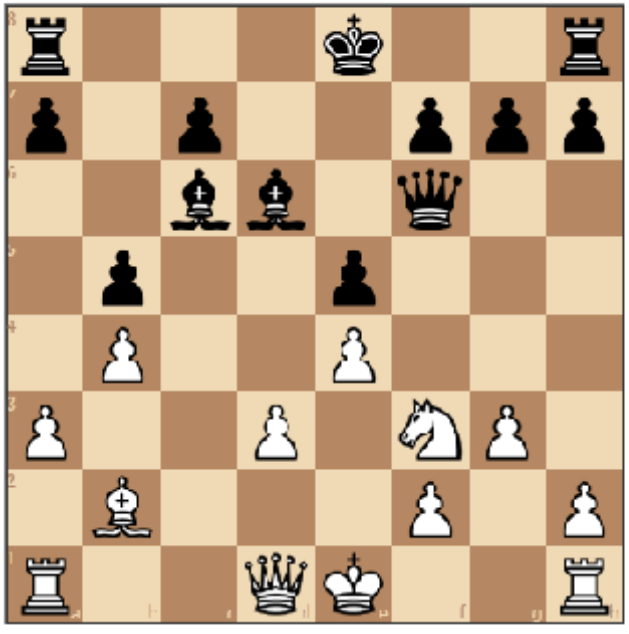


Motores de ajedrez (Chess Engines)

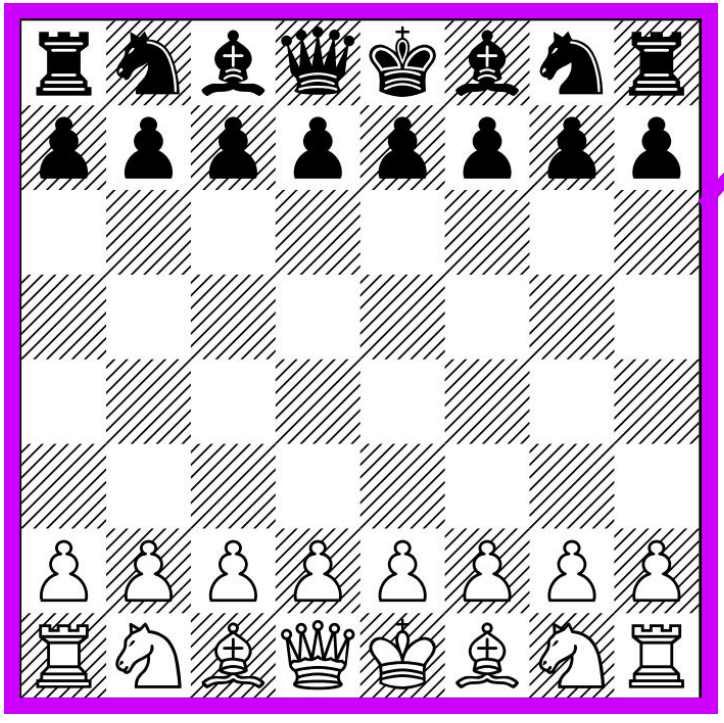
- Exploran el árbol de juego (Minimax, MCTS, etc.)
- Utilizan funciones de evaluación en las hojas
- La evaluación se propaga hacia arriba, según el algoritmo



Función de evaluación

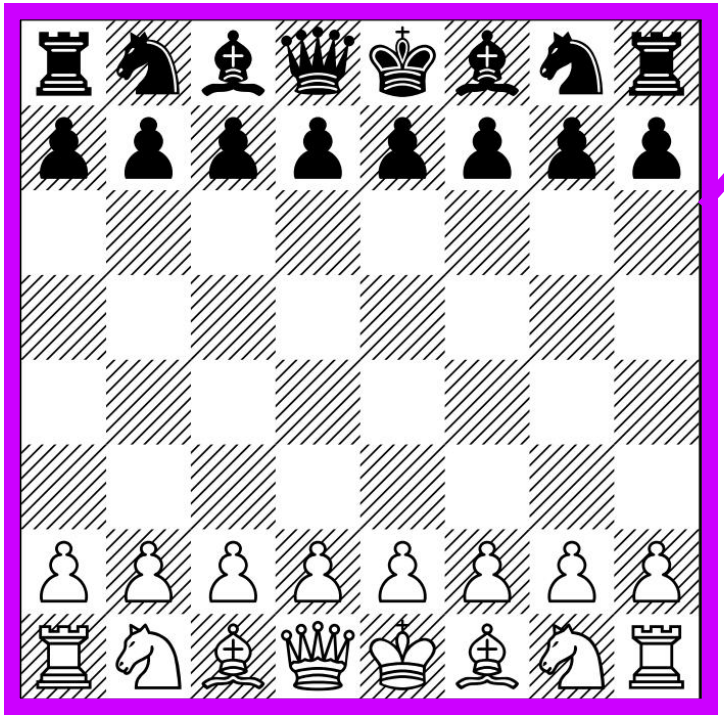
$$f\left(\right.$$  $$\left.\right) = 5$$

(adelanto) Feature set: ¿Cómo transformar la posición a un vector?



$$f(?) = ?$$

(adelanto) Feature set: ¿Cómo transformar la posición a un vector?



feature set!

$$f(?) = ?$$

Motores de ajedrez (breve historia)

Plan

asdasd

- Text visible on slide 1

asdasd

Plan

asdasd

- Text visible on slide 1
- Text visible on slide 2

asdasd

Plan

asdasd

- Text visible on slide 1
- Text visible on slide 2
- Text visible on slide 3

asdasd

Plan

asdasd

- Text visible on slide 1
- Text visible on slide 2

- Text visible on slide 4

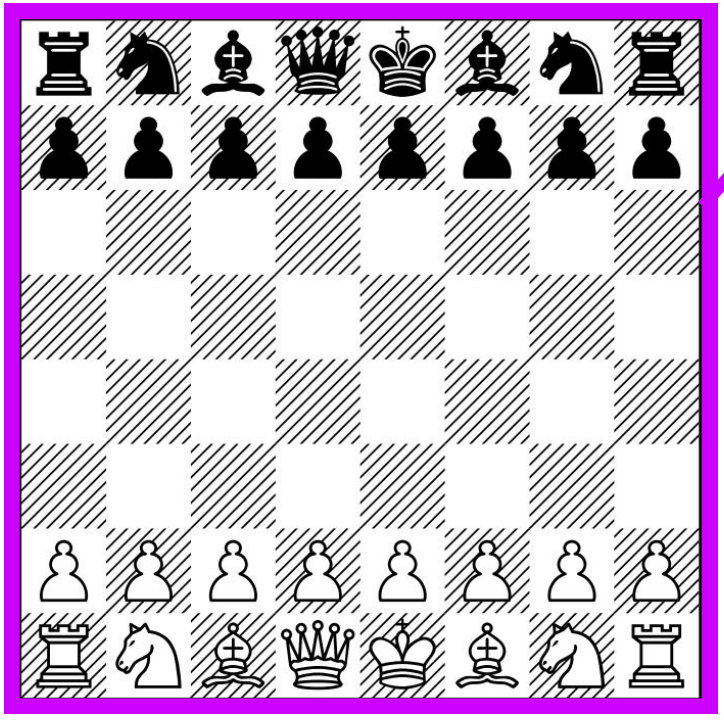
asdasd

Contenido

Engine

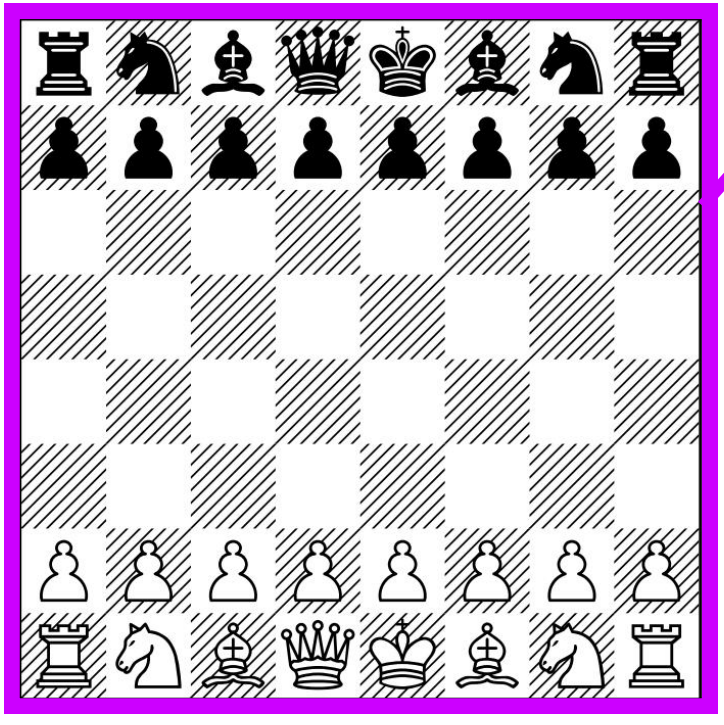
Feature set

¿Cómo transformar la posición a un vector?



$$f(?) = ?$$

¿Cómo transformar la posición a un vector?



feature set!

$$f(?) = ?$$

Definición

Un **feature set** S_P se define con un conjunto S y un predicado asociado $P(e)$, donde:

- S es un conjunto de conceptos (rol, color, celda, número, etc.).
- $P(e)$ es un predicado que determina si e está presente (o *activo*) en la posición (implícita).

Definición

Un **feature set** S_P se define con un conjunto S y un predicado asociado $P(e)$, donde:

- S es un conjunto de conceptos (rol, color, celda, número, etc.).
- $P(e)$ es un predicado que determina si e está presente (o *activo*) en la posición (implícita).

- Cada elemento en S_P es un *feature*.

Definición

Un **feature set** S_P se define con un conjunto S y un predicado asociado $P(e)$, donde:

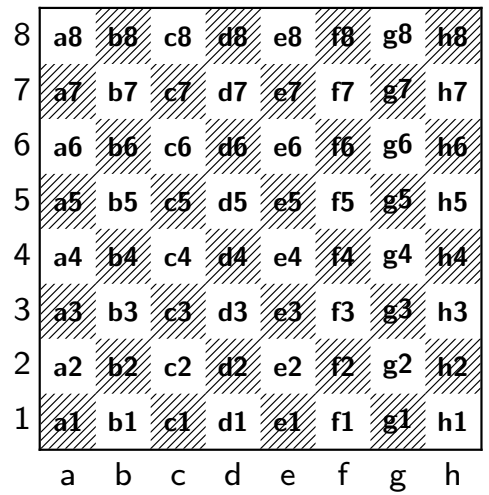
- S es un conjunto de conceptos (rol, color, celda, número, etc.).
- $P(e)$ es un predicado que determina si e está presente (o *activo*) en la posición (implícita).

- Cada elemento en S_P es un *feature*.
- Cada *feature* es un valor en el vector de entrada, valiendo 1 si está *activo* y 0 si no.

Ejemplos de S

Información posicional:

$$\textsc{Files} = \{a, b, ..., h\}$$
$$\textsc{Ranks} = \{1, 2, ..., 8\}$$
$$\textsc{Squares} = \{a1, a2, ..., h8\}$$

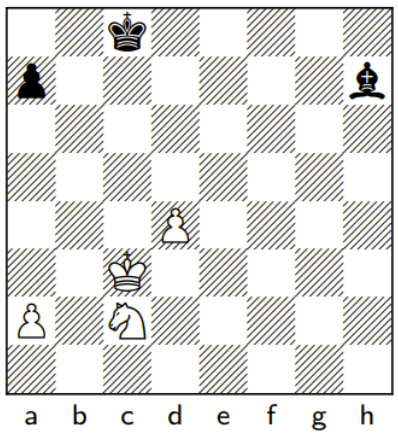


Información sobre las piezas:

$\textsc{Roles} = \{$ ♙ Pawn, ♘ Knight, ♗ Bishop, ♖ Rook, ♕ Queen, ♔ King

$\textsc{Colors} = \{$○ White, ● Black$\}$

Ejemplo completo



	Feature set	
	$(\textsc{Files} \times \textsc{Colors})_P$	$(\textsc{Files} \times \textsc{Roles})_Q$
Active features	⟨a, ○⟩, ⟨a, ●⟩, ⟨c, ●⟩, ⟨c, ○⟩, ⟨d, ○⟩, ⟨h, ●⟩	⟨a, ♙⟩, ⟨c, ♔⟩, ⟨c, ♘⟩, ⟨d, ♙⟩, ⟨h, ♗⟩

$P(\langle f, c \rangle)$: there is a piece in file f with color c.
$Q(\langle f, r \rangle)$: there is a piece in file f with role r.

Operación: Suma $\oplus$ (concatenación)

Hay veces que es útil combinar información de dos *feature sets*

Operación: Suma $\oplus$ (concatenación)

Hay veces que es útil combinar información de dos *feature sets*

$$S_P, T_Q : \text{feature sets}$$

$$S_P \oplus T_Q = (S \cup T)_R$$

$$\text{where } R(e) = \begin{cases} P(e) & \text{if } e \in S \\ Q(e) & \text{if } e \in T \end{cases}$$

Operación: Producto $\times$ (and)

$$S_P \times T_Q = (S \times T)_R$$

$$\text{where } R(\langle e_0, e_1 \rangle) = P(e_0) \ \wedge \ Q(e_1)$$

Feature set: All

La codificación más natural de una posición de ajedrez

$$\textsc{All} : (\textsc{Squares} \times \textsc{Roles} \times \textsc{Colors})_P$$

$P(\langle s, r, c \rangle)$: there is a piece in square s with role r and color c

Feature set: All

La codificación más natural de una posición de ajedrez

$$\textsc{All} : (\textsc{Squares} \times \textsc{Roles} \times \textsc{Colors})_P$$

$P(\langle s, r, c \rangle)$: there is a piece in square s with role r and color c

- Es pequeño: $64 \times 6 \times 2 = 768$ *features*

Feature set: All

La codificación más natural de una posición de ajedrez

$$\textsc{All} : (\textsc{Squares} \times \textsc{Roles} \times \textsc{Colors})_P$$

$P(\langle s, r, c \rangle)$: there is a piece in square s with role r and color c

- Es pequeño: $64 \times 6 \times 2 = 768$ *features*
- Es completo: contiene toda la información de la posición

Feature set: All

La codificación más natural de una posición de ajedrez

$$\textsc{All} : (\textsc{Squares} \times \textsc{Roles} \times \textsc{Colors})_P$$

$P(\langle s, r, c \rangle)$: there is a piece in square s with role r and color c

- Es pequeño: $64 \times 6 \times 2 = 768$ *features*
- Es completo: contiene toda la información de la posición
- Es muy rápido computar cuáles *features* están activas

Feature set: King-All ó "KA"

Los engines modernos usan variaciones del siguiente feature set.
Permite entender la posición en relación a la posición del rey:

$$\text{King-All} = \text{Square}_K \times \text{All}$$

$K(s)$: s is the square of the king of the side to move

Feature set: King-All ó "KA"

Los engines modernos usan variaciones del siguiente feature set. Permite entender la posición en relación a la posición del rey:

$$\textsc{King-All} = \textsc{Square}_K \times \textsc{All}$$

$K(s)$: s is the square of the king of the side to move

- Es grande: $64 \times 768 = 49152$ *features*

Feature set: King-All ó "KA"

Los engines modernos usan variaciones del siguiente feature set. Permite entender la posición en relación a la posición del rey:

$$\text{King-All} = \text{Square}_K \times \text{All}$$

$K(s)$: s is the square of the king of the side to move

- Es grande: $64 \times 768 = 49152$ *features*
- Es muy rápido como All

Feature set: King-All ó "KA"

Los engines modernos usan variaciones del siguiente feature set. Permite entender la posición en relación a la posición del rey:

$$\text{King-All} = \text{Square}_K \times \text{All}$$

$K(s)$: s is the square of the king of the side to move

- Es grande: $64 \times 768 = 49152$ *features*
- Es muy rápido como All
- Entrenarlo require un dataset más grande y lleva más tiempo (no me meto acá)

Resumen

- $\boldsymbol{S}$: set of concepts (roles, colors, squares, files, ranks, etc.).
- $\boldsymbol{P(e)}$: predicate that defines when the feature e is present in the (implicit) position.
- $\boldsymbol{S_P}$: a feature set. Every element in S_P is a feature. Features that satisfy P are *active*.
- $S_P \times T_Q = (S \times T)_R$ where $R(\langle e_0, e_1 \rangle) = P(e_0) \ \wedge \ Q(e_1)$
- $S_P \oplus T_Q = (S \cup T)_R$ where $R(e) = \begin{cases} P(e) & \text{if } e \in S \\ Q(e) & \text{if } e \in T \end{cases}$

NNUE

motivacion comparacion de burns

Training

Experiments

Conclusión

Ajedrez

- asdasd