Proyecto Computación Blanda: Ajedrez con machine learning

UTP | Pereira

SISTEMAS Y COMPUTACIÓN

MAYO DE 2020

2020

# CONTENIDO

[1 CONTENIDO 1](#_30j0zll)

[2 PRESENTACIÓN 2](#_1fob9te)

[3 MARCO TEÓRICO 3](#_3znysh7)

[4 OBJETIVOS 4](#_2et92p0)

[4.1 General 4](#_tyjcwt)

[4.2 Específicos 4](#_3dy6vkm)

[5 DISEÑO APLICACIÓN 5](#_1t3h5sf)

[5.1 Modelo General 5](#_4d34og8)

[5.2 Arquitectura 5](#_2s8eyo1)

[5.3 Herramientas 5](#_17dp8vu)

[5.4 Ejemplo Código Básico 5](#_1ksv4uv)

[6 resultados 6](#_44sinio)

[6.1 Presentación 6](#_3rdcrjn)

[6.2 Productos Entregables 6](#_26in1rg)

[7 CONCLUSIONES 7](#_lnxbz9)

[7.1 Conclusiones Proyecto 7](#_35nkun2)

# PRESENTACIÓN

**Presentación del proyecto**

Numerosos estudios publicados revelaron que varios investigadores han intentado construir un programa que aprende a jugar juegos cognitivos, dado que tienen poco o ningún conocimiento previo sobre la regla del juego. Una máquina de juego de ajedrez habitual explora a fondo las posibilidades de movimiento desde la configuración de un tablero de ajedrez para elegir cuál es el siguiente mejor movimiento. La técnica de búsqueda de fuerza bruta utilizada por el motor de ajedrez Deep Blue ha tenido un gran impacto en el terreno de la inteligencia artificial, pero aún se encuentra hambriento de recursos. Este documento, con el concepto de Redes Neuronales Artificiales, presenta un enfoque muy simple y eficiente para desarrollar un motor de ajedrez inteligente que pueda ayudar e insinuar el posible movimiento dentro del juego utilizando la técnica de computación evolutiva y adaptativa para aprender de los grandes maestros humanos.

**AUTORES: NOMBRES Y CÓDIGOS**

Santiago Ramírez Moreno - 1088343413

Oscar Mauricio Giraldo Herrera - 1

# MARCO TEÓRICO

**Contexto:**

Los enfoques clásicos del juego de ajedrez por máquinas han evolucionado a lo largo de los años. Nicolás Lassabe et al mencionaron en su trabajo que, el primer enfoque fue evaluar y almacenar, para cada configuración de tablero de ajedrez, el mejor movimiento para jugar. Este enfoque simplista e idealista requería grandes comparaciones y evaluaciones. Von Neumann y Morgenstern (1944) propusieron el mínimo como método para decidir qué movimiento hacer en el ajedrez. Un juego de dos jugadores como el ajedrez se puede representar con un árbol. Cada nodo es una posición en el tablero, comenzando en la raíz con la posición inicial del juego. Los vértices son los movimientos posibles, que conducen a los siguientes nodos (posiciones).

**Metodología**

Esto ilustra la secuencia de flujo durante la fase de entrenamiento donde los movimientos de entrada tomados del juego de gran maestro se codifican usando el esquema de codificación de tablero que se analiza más adelante. Los movimientos codificados se introducen en la red neuronal diseñada donde el sistema está entrenado para producir la salida. El sistema aprende los patrones (movimientos y estrategia) que genera el gran maestro humano durante sus juegos de campeonato.

**Aprendizaje de diferencia temporal en el dominio del ajedrez**

El aprendizaje por diferencia temporal (TO) comprende una familia de enfoques para la predicción en casos en los que el evento a predecir puede retrasarse por un número desconocido de pasos de tiempo. En el contexto del juego, los métodos TD se han aplicado con frecuencia para aprender funciones que predicen el resultado final de los juegos. Dichas funciones se utilizan como funciones de evaluación de la junta.

# OBJETIVOS

## General

Implementar un sistema machine learning para un ajedrez

## Específicos

|  |  |
| --- | --- |
| No | Objetivo Específico |
| 1 | Investigar sobre machine learning |
| 2 | Documentar los requisitos necesarios para la construcción del software |
| 3 | Diseñar una estructura que cumpla con los requisitos documentados |
| 4 | Implementar un sistema con base al diseño y la documentación planteada |
| 5 |  |

# DISEÑO APLICACIÓN

En los siguientes apartados se presenta una visión sistémica del modelo general del proyecto de monografía presentado en este documento.

## Modelo General

**Prueba del sistema en un entorno entrenado:**

El sistema diseñado cuando se probó con la secuencia de movimiento entrenado produjo el resultado como se muestra. Se encuentra que el sistema se comportó muy bien para la secuencia entrenada de movimientos.

**Caso de prueba II:**

Escenario (fase intermedia del juego) Después de que el juego progresa, las piezas en el tablero de ajedrez se desarrollan dando como resultado un crecimiento exponencial en la posibilidad de que cada una de las piezas se mueva dentro del tablero. Durante dicho escenario, se descubre que el resultado del sistema es muy simple y efectivo, ya que se da cuenta e insinúa el movimiento intermedio de una pieza en particular en el tablero que se moverá a continuación.

**Prueba del sistema en un entorno desconocido**

El sistema diseñado cuando se probó en el nuevo entorno produjo los resultados que son muy aceptables durante la fase inicial del juego al insinuar algo de la posición correcta de la pieza que se moverá a continuación (por ejemplo, 5755) Cuando el alfil fue movido de la posición 61 a la 34, el sistema sugirió mover el peón de la posición 67, pero se encontró bloqueado por un caballero posicionado en el 66. Aunque insinuó el movimiento inválido pero el sistema fue lo suficientemente inteligente como para sugerir la pieza. Ser movido a continuación en el entorno desconocido.

**Diseño del sistema**

Prueba del sistema en un entorno desconocido El sistema diseñado cuando se probó en el nuevo entorno produjo los resultados que son muy aceptables durante la fase inicial del juego al insinuar algo de la posición correcta de la pieza que se moverá a continuación (por ejemplo, 5755) Cuando el alfil fue movido de la posición 61 a la 34, el sistema sugirió mover el peón de la posición 67, pero se encontró bloqueado por un caballero posicionado en el 66. Aunque insinuó el movimiento inválido pero el sistema fue lo suficientemente inteligente como para sugerir la pieza. ser movido a continuación en el entorno desconocido.

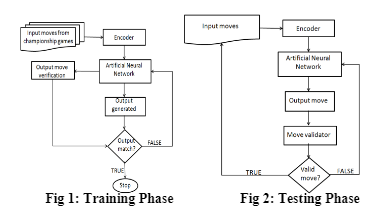
**Implementación**

La notación algebraica es un método para registrar y describir los movimientos en un juego de ajedrez. Cada cuadrado del tablero de ajedrez se identifica por un par de coordenadas único, una letra y un número. Cada cuadrado tiene una identificación única de la letra del archivo seguido del número de rango. Teniendo en cuenta el hecho de que la red neuronal artificial solo comprende los valores numéricos de entrada, esas anotaciones algebraicas deben procesarse previamente.

## Arquitectura

**Diseño del sistema**

El sistema se entrena hasta que produce el resultado con un error mínimo. La Figura muestra el flujo International Journal of Computer Applications® (IJCA) (0975 - 8887) Conferencia Nacional cum Taller sobre Bioinformática y Biología Computacional, NCWBCB- 2014 del sistema diseñado durante la fase de prueba. Aquí el sistema entrenado se prueba con los movimientos que no se entrenaron anteriormente, es decir, movimientos desconocidos alimentados por el usuario en un entorno desconocido, donde el sistema se comporta como un generador de movimientos. Los movimientos así generados se validan para verificar si son movimientos válidos o no. También verifica el movimiento sugerido para cumplir con las reglas del juego.



## Herramientas

Se usaran las herramientas para desarrollo Python

## Ejemplos de aplicación tomados de la literatura existente

1. Red neuronal profunda de extremo a extremo para el aprendizaje automático en ajedrez

2. Dominar el ajedrez y el shogi mediante el auto-juego con un algoritmo de aprendizaje de refuerzo general

3. Aprender a evaluar las posiciones de ajedrez con redes neuronales profundas y búsqueda anticipada limitada

# RESULTADOS

## Presentación

Construir un sistema machine learning aplicado en el juego ajedrez

## Productos Entregables

En esta sección se da entender sobre los entregables del proyecto los cuales son:

|  |  |
| --- | --- |
| No | Productos Entregables |
| 1 | Sistema de machine learning aplicado en el ajedrez |
| 2 | Documentación del sistema |
| 3 |  |
| 4 |  |
| 5 |  |

# CONCLUSIONES

## Conclusiones Proyecto

|  |  |
| --- | --- |
| No | Conclusiones |
| 1 | Tener un conocimiento sobre el manejo del machine learning |
| 2 | Aprender a construir un sistema de machine learning |
| 3 | Documentar el conocimiento aplicado en el sistema de machine learning |
| 4 |  |
| 5 |  |

# BIBLIOGRAFÍA

- Thomas S. Anantharaman. A Statistical Study of Selective Min-Max Search in Computer Chess.

PhD thesis, Carnegie Mellon University, School of Computer Science, Pittsburgh, PA, 1990.

Technical Report CMU-CS-90-173.

- R. E. Bellman. Dynamic Programming. Princeton University Press, Princeton, NJ, 1957.

- Hans J. Berliner, Gordon Goetsch, Murray S. Campbell, and Carl Ebeling. Measuring the Performance potential of chess programs. Artificial Intelligence, 43:7-20, 1990.

- Justin A. Boyan. Generalization in reinforcement learning: Safely approximating the value

Function. In G. Tesauro, D. Touretzky, and T. Leen, editors, Advances in Neural Information Processing Systems 7, San Mateo, CA, 1995. Morgan Kaufmann. (To appear).

- Gerald Dejong and Raymond Mooney. Explanation-based learning: An alternative view. Machine Learning, 1(2): 145-176, 1986.

- Michael Gherrity. A Game-Learning Machine. PhD thesis, University of California, San Diego, 1993.

- Tom M. Mitchell, Rich Keller, and Smadar Kedar-Cabelli. Explanation-based generalization: A Unifying view. Machine Learning, 1 (1):47-80, 1986.

- Tom M. Mitchell and Sebastian Thrun. Explanation based learning: A comparison of symbolic

# FIRMA DEL DOCUMENTO

Firmas de los autores de la monografía, indicando el MES y el DÍA de la firma:

(FIRMA DIGITAL)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NOMBRE:

CÓDIGO:

(FIRMA DIGITAL)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NOMBRE:

CÓDIGO:

(FIRMA DIGITAL)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NOMBRE:

CÓDIGO: