Predictor de equipo ganador en Counter Strike: Global Offensive

Vargas, Leandro; Eberle, Gerónimo; Priemer, Eric

*Inteligencia Computacional, Ingeniería en Informática, Facultad de Ingeniería y Ciencias Hídricas, Universidad Nacional del Litoral.*

***Resumen. Dentro de los enfrentamientos en las partidas de Counter Strike: Global Offensive a nivel profesional, hay mucha incertidumbre sobre el posible resultado del partido en cuestión. Por ejemplo se dan partidos donde un equipo con 70% de posibilidades de victoria, pierde, o viceversa. Debido a esto, se motiva el estudio de poder predecir a partir de un Perceptrón Multicapa el ganador de un partido, teniendo en cuenta estos casos de incerteza. El mismo será entrenado con datos de un torneo en particular. Se realizará un pre-procesamiento de datos descargados de una base de datos de internet y se incluirán los datos ya pre-procesados de 2 equipos, más datos de un mapa específico que se jugará. El modelo neuronal deberá dar como salida el resultado de quién será el equipo vencedor de ese partido en particular. Se realizaron pruebas de validación cruzada, y pruebas con datos con características históricas de cada equipo y sin histórico. El resultado final muestra una tasa de acierto alta, favorable para datos con características históricas.***

***Palabras clave: Perceptrón Multicapa- CS:GO - Predictor - incertidumbre.***

**I. INTRODUCCIÓN**

Dentro de los denominados E-Sports (Deportes electrónicos en inglés), se encuentra el juego **Counter Strike: Global Offensive** (a partir de ahora **CS:GO**), el cual como cualquier otro deporte, presenta situaciones de incertidumbre respecto al resultado de un partido futuro. Un equipo con la mayor probabilidad de victoria puede perder, o un equipo que parecía iba a ser derrotado fácilmente, dio pelea y resultó ganador. El propósito de este trabajo, es evaluar y determinar, si estos casos pueden ser correctamente predecidos por un Perceptrón Multicapa (a partir de ahora MLP de su acrónimo MultiLayer Perceptron en inglés), utilizando como características, datos estadísticos respecto a su historia global, y también su historia en los partidos pasados de un torneo en particular, en el cual se quiere predecir sus resultados.

La idea de utilizar MLP para resolver este trabajo y los datos estadísticos utilizados fueron inspirados a partir de [1], donde se utilizan datos históricos entre los equipos a disputar el partido, datos previos históricos, datos previos entre ellos, datos sobre la ubicación donde se jugará, el ranking de cada equipo, puntos a favor y en contra, historico de puntos a favor y en contra, y la habilidad de cada jugador, que por medio de estas entradas y un MLP, predice resultados para partidos de distintos deportes, con una tasa de acierto de entre 57% a 68%. En [2] se utilizan los históricos de enfrentamiento entre equipos, tiros al arco, posesión del balón, entre otros, estos datos son utilizados para pasarlos como características normalizados a un MLP, el cual alcanzó un 75% de acierto .

Siguiendo con las ideas de [1], y los datos que se pudieron obtener, utilizamos los datos estadísticos e históricos que encontramos a nuestra disposición sobre un torneo en particular de CS:GO. Estos datos tuvieron que pasar por una etapa de preprocesamiento para que sean legibles por el MLP. El cual de antemano se sabía iba a tener como entradas las características obtenidas de los distintos equipos, y como salida un perceptrón de decisión para determinar el equipo ganador. Quedará como parte de este estudio determinar la cantidad de capas ocultas de la red, y como objetivo inicial una tasa de aciertos de entre el 60% y 75%

**II. MÉTODOS**

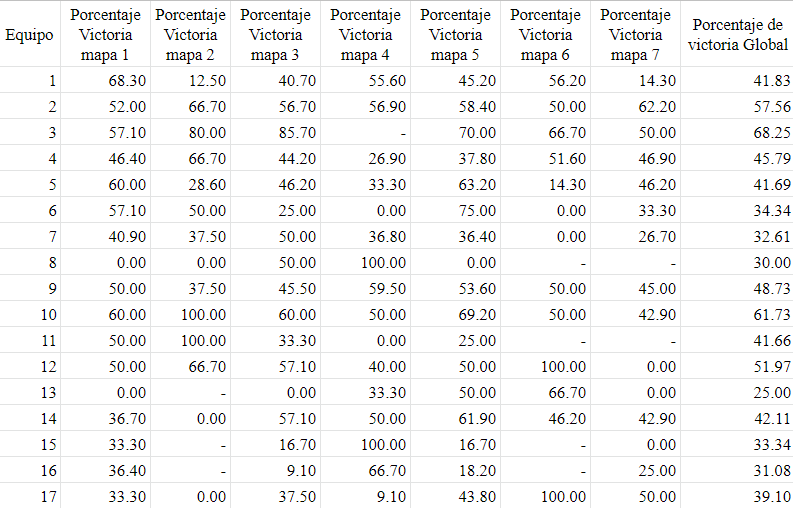
Para la resolución de este trabajo, se realizó una búsqueda de información sobre los equipos y torneos, luego un preprocesamiento de estos datos obtenidos y se utilizó un MLP en el cual se tomarán como entradas distintas características de cada uno de los equipos que participan en un determinado partido.

*A. Datos*

Los datos fueron descargados de la página web hltv.org utilizando una Interfaz de Programación de Aplicaciones ( cuyo acrónimo en inglés es API, Application Programing Interface) programada en TypeScript por Stanislav Iliev [3]. Con esta API se descargaron los datos del torneo “ESEA MDL season 35 north america” los cuales serán utilizados tanto para entrenamiento como para test. De aquí se obtendrá la información de todas las fechas que se jugaron

Tabla I

Datos Históricos por equipo



en el torneo cada una con los 2 equipos participantes, el mapa en el cual se llevó a cabo el encuentro y el resultado final del partido.

Por otro lado también se obtuvo las estadísticas por separado de cada uno de los equipos que intervienen en los partidos de este torneo en la cual se proporcionaba el porcentaje histórico de victorias de ese equipo y el porcentaje histórico de victorias de ese equipo en cada uno de los mapas que se jugarán en ese torneo.

*B. Pre procesamiento de datos*

Se realizó una codificación binaria (del inglés one-hot encoding) de todos los mapas que se iban a jugar en ese torneo, que eran un total de 7:

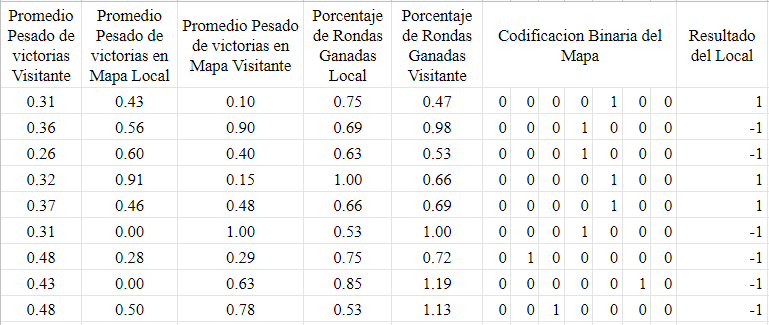
* overpass 0000001
* vértigo 0000010
* inferno 0000100
* nuke 0001000
* mirage 0010000
* dust\_2 0100000
* train 1000000

Para preprocesar los datos y convertirlo en las futuras características del perceptrón se realizaron 2 tareas diferentes.

* La primera fue calcular el porcentaje de victoria de cada equipo tanto general como de cada mapa solo con los datos de entrenamiento de ese torneo, es decir se cuentan los partidos ganados del bloque de entrenamiento y se divide por la cantidad de partidos, y lo mismo para los mapas en particular.
* La segunda fue hacer un promedio ponderado tanto del porcentaje de victoria general de ese torneo con el porcentaje histórico general del equipo como del porcentaje de victoria de cada mapa en ese torneo con el porcentaje de victorias histórico de cada mapa.

En Tabla I, se muestra para cada equipo los datos estadísticos obtenidos para cada mapa, y su desempeño

Tabla II  
Datos de entrada al MLP con histórico



global. Figurando ‘**-’** en caso de no haber jugado ningún partido en ese mapa, **0** si no obtuvo ninguna victoria en ese mapa, o el porcentaje correspondiente. Una vez recopilados estos datos se realiza el promedio pesado para cada par de datos estadísticos (histórico y del torneo actual) tanto para el local como visitante. Como se ve en las ecuaciones siguientes a modo ejemplo mostrando solo para el local:

Siendo correspondiente a estadísticas del tipo actual tanto para el local como para el visitante, tomado de a pares con su respectivo valor histórico, el cual corresponde a en la ecuación y el valor pesado de los datos. Por ejemplo =porcentaje de victoria,= porcentaje de victoria historico.

Tener en cuenta que estos mismos porcentajes calculados en el entrenamiento, son pasados a los datos de test, detectando que equipos disputan, y pasando de estos datos a cada uno, es decir, no hay un cálculo de porcentaje para los partidos de test, dado que se suponen, aún no sucedieron.

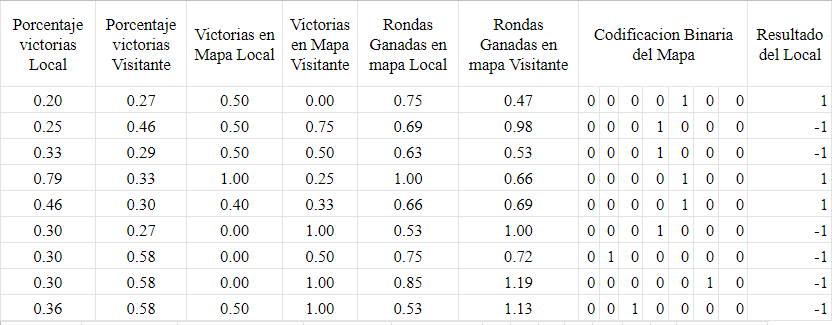
Por último, en el resultado del Local se buscó quien tenía un número mayor de rondas ganadas en ese encuentro y en base a eso se agregó 1 si el que mayor número de rondas ganadas era el equipo 1 y -1 si era el equipo 2.

En Tabla II, se muestran algunos elementos resultantes de aplicar los datos estadísticos y los pesos, al histórico y al torneo actual, para cada una de las columnas. Entre ellas promedio de victorias global, del mapa en cuestión, y de rondas ganadas para ese mapa. Todo esto tanto para local como para visitante.

*C. Estructura del MLP*

En principio se determinó la cantidad de entradas en base a las características obtenidas y la salida sería un perceptrón el cual de 1 en caso de que el equipo local sea el ganador, y -1 caso contrario. Para la capa oculta se realizó un proceso iterativo, donde se evaluó la calidad de la solución en base al monitoreo en la etapa de entrenamiento hasta encontrar el número óptimo para la capa oculta.

Tabla III  
Datos de entrada al MLP sin histórico:



Luego para cada partido del entrenamiento se busca la información correspondiente de la matriz de datos y se completan las entradas al perceptrón de la siguiente manera:

* WinRate equipo 1
* WinRate equipo 2
* WinRate mapa X equipo 1
* WinRate mapa X equipo 2
* WinRate de rondas en mapa X equipo 1
* WinRate de rondas en mapa X equipo 2
* One-hot encoding mapa X
* Salida de referencia

Las características finales que ingresan al perceptrón se ilustran en Tabla III, la cual contiene los datos calculados anteriormente, pero considerando que no hay datos históricos, y que el peso asociado a los datos calculados actuales, tienen peso 1.

Los datos vistos en la Tabla II contienen las características finales que ingresan al perceptrón, pero esta vez con datos históricos. Es importante notar la diferencia en los valores, dado que son promedios ponderados entre el histórico y el porcentaje del torneo actual.

Como función de activación del MLP se utilizó la función sigmoidea entre -1 y 1 para poder comparar el signo de esta con el valor de referencia de la salida deseada.

donde es la salida del perceptrón y es un parámetro para modificar la velocidad de variación entre -1 y 1.

El método de reducción de error fue el de gradiente descendiente, y se realizaron pruebas de este mismo método con y sin el término de momento.

**III. RESULTADOS**

Para el entrenamiento se dividió todo el set de datos de 153 partidos en:

-80% Entrenamiento.

-20% Test.

A su vez, del 80% de entrenamiento se guardó un 10% para monitoreo con la intención de evitar el sobre entrenamiento.

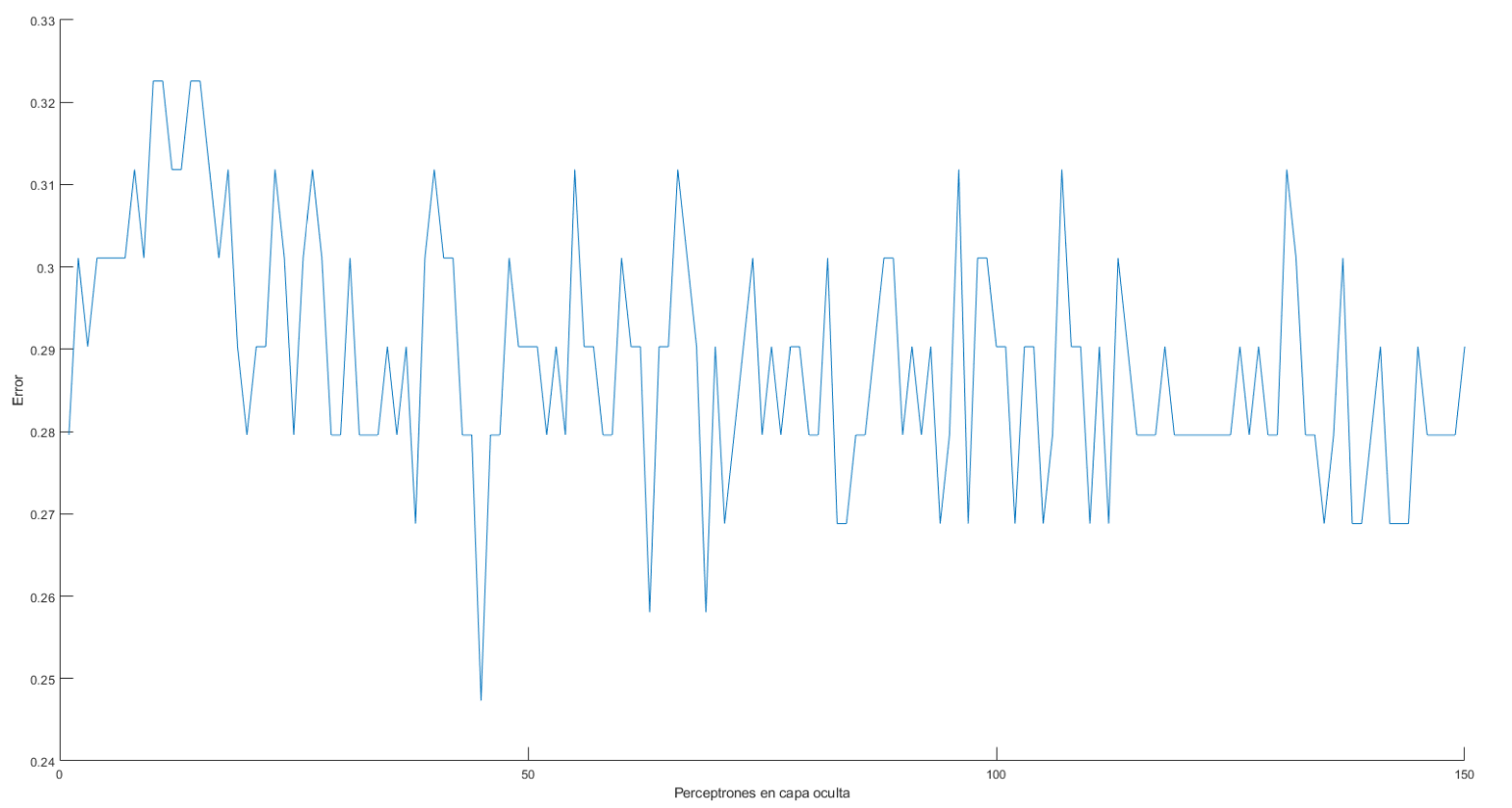


Figura 1: Error en el monitoreo para n capas ocultas.

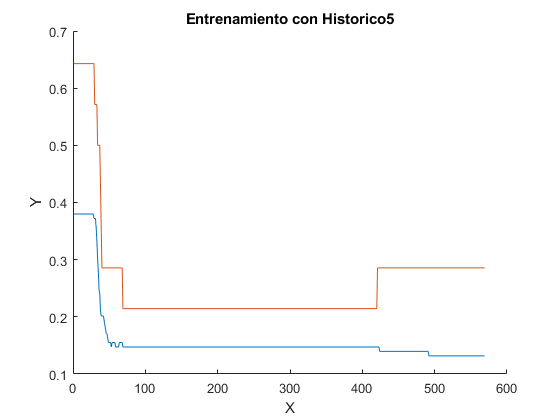


Figura 2: Error en entrenamiento y monitoreo para cada época con histórico.

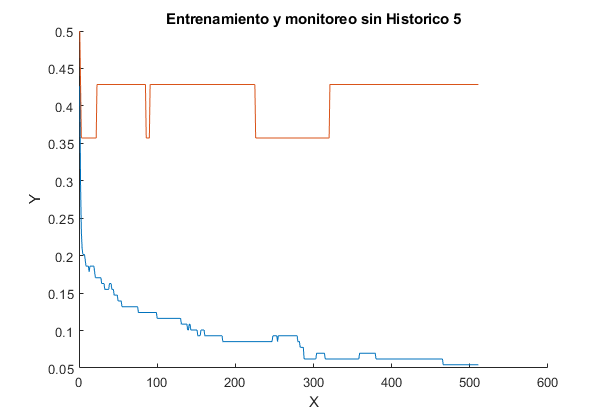


Figura 3: Error en entrenamiento y monitoreo para cada época sin histórico.

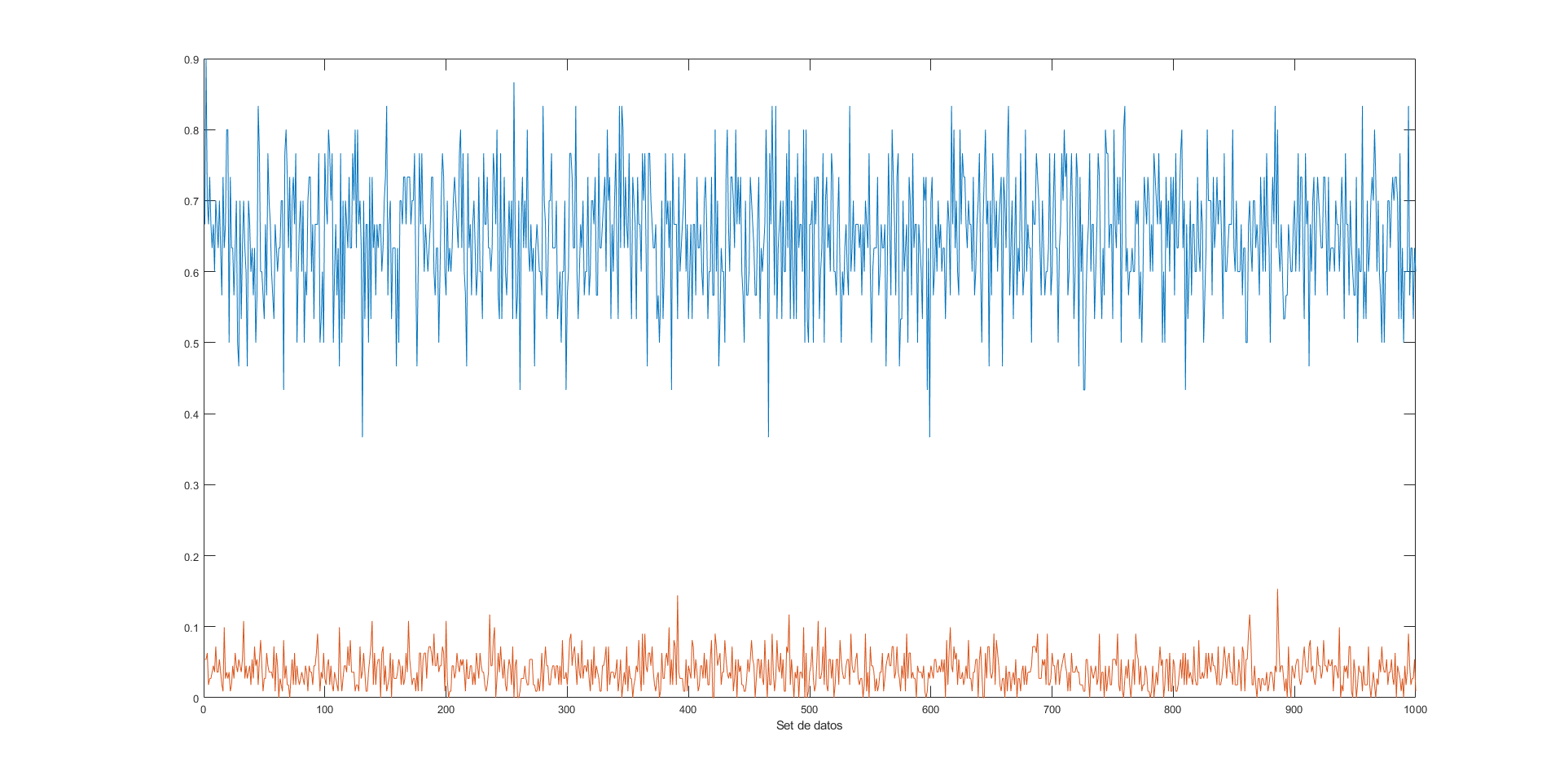


Figura 4: Tasa de aciertos y error de entrenamiento para los 1000 set de datos con histórico.

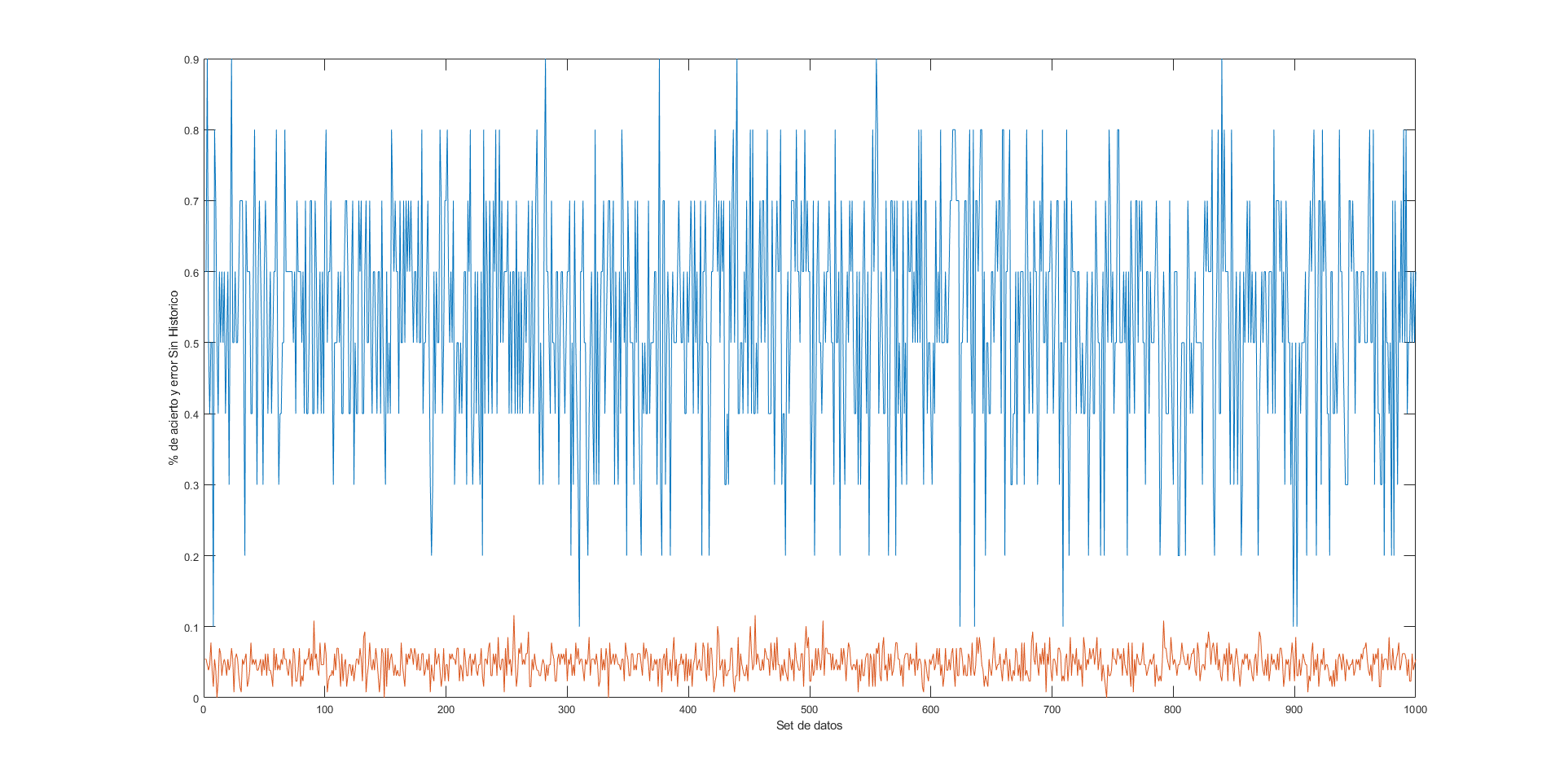


Figura 5: Tasa de aciertos y error de entrenamiento para los 1000 set de datos sin histórico.

Luego de realizar las iteraciones variando la cantidad de capas ocultas como se nombró en II. C, se llegó a una estructura con 13 perceptrones iniciales o de entrada, 45 en la capa oculta, y 1 en la salida, con 1000 repeticiones de datos desordenados para cada iteración.

Como se ve en la Figura 1, para la cantidad de capas ocultas igual a 45 se produce el menor error en los datos de monitoreo, por lo que queda definida la capa oculta por esta cantidad. Una vez determinado el número de perceptrones en la capa oculta se volvió a correr el entrenamiento y prueba, otra vez para 1000 iteraciones variando el conjunto de entrenamiento y test de manera aleatoria en cada iteración.

Notar en Figura 2 y Figura 3 que el entrenamiento se detiene en la mayoría de los casos, por no mejorar la solución en los datos de monitoreo por más de 500 épocas de entrenamiento. En naranja se ve el error de monitoreo, mientras que en azul el error de entrenamiento.

En la Figura 4 y Figura 5 se puede ver en azul la tasa de aciertos para cada iteración de datos desde la 1 hasta la 1000. Y en naranja el error de entrenamiento para cada una de estas iteraciones. Como se puede observar en el gráfico, la tasa de aciertos es en promedio de un 68% para los datos históricos, y 63% para los datos sin histórico. Notar también que la tasa de acierto está más acotada en la gráfica con datos históricos, mientras que en la gráfica sin históricos hay valores que hasta llegan a un 10% de aciertos. En ambos casos el error de entrenamiento no supera el 0.15%

**IV. CONCLUSIONES**

Como se vio a lo largo del trabajo, se realizaron pruebas con datos históricos y sin datos históricos. Donde, aplicando datos históricos, y dándole un mayor peso que a los datos estadísticos del torneo actual, se obtuvo una buena calidad en la solución.

Si bien la incertidumbre siempre estará presente, la resolución del problema con la incorporación de los datos históricos dio un buen porcentaje de aciertos sobre el total de datos. Desafortunadamente, la base de datos utilizada contaba con una cantidad limitada de elementos, por lo que de haberse contado con una base de datos más amplia, posiblemente se hubieran obtenido mejores resultados o una mejor capacidad de generalización.

De continuar el desarrollo del proyecto, se buscaría ampliar la cantidad de datos (partidos), y la posibilidad de obtener mayor cantidad de datos estadísticos para mejorar la precisión, como pueden ser estadísticas sobre los jugadores del equipo, variaciones en el plantel, datos del mejor jugador de cada equipo, y más datos históricos tanto del equipo, como del enfrentamiento. O datos como en [4], donde se utilizan los enfrentamientos pasados entre los demás equipos, respecto a los dos que jugarán en contra, y esos porcentajes de victorias servirán como entradas para el MLP.

**REFERENCIAS**

|  |  |
| --- | --- |
| [1] | Alan McCabe and Jarrod Trevathan, "Artificial Intelligence in Sports Prediction," Fifth International Conference on Information Technology: New Generations (itng 2008), Las Vegas, NV, 2008, pp. 1194-1197, doi: 10.1109/ITNG.2008.203. |
| [2] | Kou-Yuan Huang, Kai-Ju Chen, "Multilayer Perceptron for Prediction of 2006 World Cup Football Game", *Advances in Artificial Neural Systems*, vol. 2011, Article ID 374816, 8 pages, 2011. <https://doi.org/10.1155/2011/374816> |
| [3] | Stanislav Iliev, Interfaz de Programación de Aplicaciones para extracción de datos.  https://github.com/gigobyte/HLTV. |
| [4] | William J. Knottenbelt, Demetris Spanias, Agnieszka M. Madurska, “A common-opponent stochastic model for predicting the outcome of professional tennis matches”, Computers & Mathematics with Applications,Volume 64, Issue 12, 2012.  <https://doi.org/10.1016/j.camwa.2012.03.005> |