

Aprendizaje de Redes Neuronales Recurrentes con instancias de longitud variable. Aplicaciones a la resistencia antiviral del VIH

Isel Grau¹, Isis Bonet^{2*}

RESUMEN. La predicción de la resistencia del Virus de Inmunodeficiencia Humana es un problema de clasificación de secuencias que intenta predecir la susceptibilidad de una mutación a un fármaco determinado partiendo de la información del código genético de la proteína que éste inhibe. Las redes neuronales recurrentes ofrecen buenos resultados en la predicción de la resistencia del virus ante los inhibidores de la proteasa. La transcriptasa reversa tiene como característica distintiva la variabilidad de la longitud de sus secuencias, por lo que se propone una modificación de las redes neuronales recurrentes para adaptar los tiempos de la red de forma dinámica en función del tamaño de cada caso, y de esta forma sólo aprender de la información relevante de cada secuencia, ignorando los fragmentos no codificados. En el siguiente trabajo se utiliza un modelo de redes neuronales recurrentes que permite obtener mejores resultados en el problema de clasificación de resistencia a 11 inhibidores de la proteína transcriptasa reversa del Virus de Inmunodeficiencia Humana.

INTRODUCCIÓN

El Virus de Inmunodeficiencia Humana (VIH) es un retrovirus con alta capacidad de mutación que afecta considerablemente a la humanidad causando más de 3 millones de muertes al año, a pesar de los avances en el desarrollo de terapias para combatirlo. Por estas razones se hace necesario el estudio de la resistencia antiviral de los fármacos ya existentes, con el objetivo de utilizar de manera adecuada cada uno, o combinaciones de ellos, ante la aparición de nuevas mutaciones del virus.

Desde el punto de vista molecular el VIH es una poliproteína que está conformada entre otras por la proteasa y la transcriptasa reversa, dos enzimas que intervienen en el proceso de replicación del virus. Particularmente la transcriptasa reversa es la encargada de copiar el código genético del virus en el ADN de la célula, lo que

propicia la creación de nuevas copias del virus. No es extraño entonces que muchos fármacos antirretrovirales estén diseñados para inhibir la función de esta proteína.

Existen dos formas experimentales de estimar la susceptibilidad de una cepa del VIH ante un fármaco: las pruebas fenotípicas y las pruebas genotípicas. Los resultados de la unión de estos dos experimentos, en pares genotipo-fenotipo suman una gran cantidad de información disponible en bases de datos reconocidas como Stanford HIV Resistance Database lo que ha generado la aplicación de una gran variedad de métodos estadísticos y de inteligencia artificial para intentar predecir la susceptibilidad del virus a determinado fármaco a partir de su código genético.

Como una característica interesante del VIH está la variabilidad en sus secuencias, por la cantidad de mutaciones que presenta. En el caso particular de la proteasa la cantidad de inserciones y eliminaciones no es significativa. Sin embargo, las secuencias de las mutaciones de la

¹ Centro de Estudios de Informática, Universidad Central "Marta Abreu" de Las Villas, Cuba *isisb@uclv.edu.cu
Disponible en el sitio web: <http://hivdb.stanford.edu/>

transcriptasa reversa sí presentan muchas inserciones y eliminaciones en su código genético, lo que provoca que las secuencias tengan longitudes muy variables y complejiza la selección de un clasificador apropiado para el problema a tratar. Teniendo en cuenta esta característica de las secuencias del VIH, en (Bonet, 2008, Salazar, 2005) se obtuvieron buenos resultados en la aplicación de un modelo de Red Neuronal Recurrente (RNR) para el pronóstico de la resistencia del VIH ante inhibidores de la proteasa como un problema de clasificación.

Respecto a la transcriptasa reversa en la literatura revisada se utilizan clasificadores comunes con entradas de secuencias de longitud fija pero aprendiendo con sólo una parte de la transcriptasa reversa o describiendo la secuencias con descriptores químico biológicos (Rhee *et al.*, 2006, Vermeiren *et al.*, 2004, Kierczak *et al.*, 2009).

La variante de tratar solamente una parte de la secuencia de la transcriptasa reversa puede obviar información importante, incluso puede dejarse de analizar algunas posibles mutaciones. Precisamente el sentido de este trabajo es diseñar y entrenar un modelo de RNR con bases de inhibidores de la transcriptasa reversa en la cual se mantengan las secuencias con su tamaño original, conservando la información de todas sus posiciones y que permita la predicción de resistencia del VIH ante inhibidores de esta proteína.

MÉTODOS

Redes Neuronales Artificiales

Las redes neuronales artificiales se agrupan dentro de las técnicas conexionistas de la Inteligencia Artificial y constituyen una de las especialidades más ampliamente difundidas. Estas herramientas matemáticas para la modelación de problemas, permiten obtener las relaciones funcionales subyacentes entre los datos involucrados en problemas de clasificación, reconocimiento de patrones, regresiones, etc. (Hammer and Villmann, 2003).

Una red neuronal puede ser caracterizada por el modelo de la neurona, el esquema de conexión que presentan sus neuronas, o sea su topología, y el algoritmo de aprendizaje emplea-

do para adaptar su función de cómputo a las necesidades del problema particular.

En los últimos años se han producido una amplia variedad de arquitecturas de redes neuronales, sin embargo, la mayoría de ellas se encuentran ubicadas en dos grandes grupos: las redes multicapa de alimentación hacia adelante (Feed-Forward Neuronal Networks, FFN) y las redes recurrentes (Recurrent Neuronal Networks, RNR) (Hilera and Martínez, 1995).

Redes Neuronales Recurrentes

Las RNRs se diferencian de las redes FFN en que éstas admiten conexiones hacia atrás, o sea, pueden formar ciclos en el grafo que describe sus conexiones, lo que permite que la red sea capaz de guardar una memoria de los estados anteriores para su uso en el cálculo de las salidas del estado actual. Ésta es la característica esencial que convierte a este tipo de redes en una herramienta de amplio uso en tareas de reproducción de señales y análisis de secuencias, donde se reflejan relaciones causales en el tiempo y el espacio respectivamente (Pearlmuter, 1990, Baldi, 2002).

En la Figura 1 se muestra un ejemplo simple de una RNR de dos capas con conexiones recurrentes en el tiempo. Esta topología puede emplearse para el análisis de secuencias donde existe influencia entre las posiciones iniciales, medias y finales de la proteína.

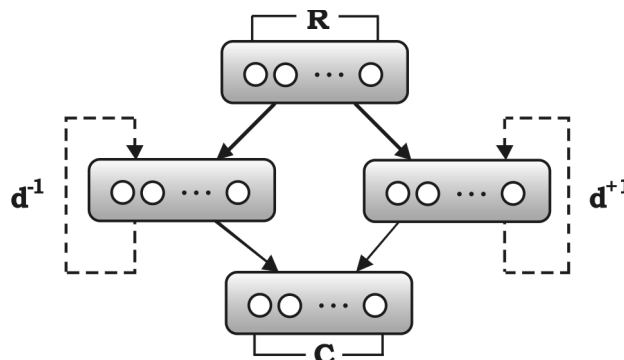


Fig. 1 Red Neuronal Recurrente de dos capas con conexiones hacia atrás y hacia adelante en el tiempo.

De otra manera, el estado computado con la entrada provista a la red en el tiempo t , es usado junto a la entrada provista en el tiempo $t+1$ para calcular el nuevo estado y la salida en $t+1$. Los

operadores de desplazamiento (shift operators) $d+1$ representados en las conexiones recurrentes señalan la sincronización de las dependencias entre tiempos. Un exponente positivo advierte que la red requiere el estado computado en etapas anteriores y un exponente negativo, que se requiere el estado de etapas posteriores.

Algoritmo de aprendizaje Backpropagation Through Time

El algoritmo de entrenamiento Backpropagation Through Time (BPTT) para una RNR es una adaptación del algoritmo Backpropagation para redes FFN. Éste es uno de los algoritmos más utilizados en la literatura y justamente es el que se implementa para el modelo propuesto. Este algoritmo realiza la actualización del estado de activación de cada neurona en los momentos puntuales de tiempo t_1, t_2, \dots, t_n .

El primer paso de BPTT consiste en el proceso de desdoblamiento (unfolding) donde la red neuronal que se obtiene no es más que el resultado de la replicación de la red recurrente original t veces (t es la cantidad de tiempos). En este proceso cada conexión replicada comparte su valor w_{ij} en todas las etapas. La red desplegada no es más que una red feed-forward, y es susceptible de la aplicación del algoritmo Backpropagation para su entrenamiento. Precisamente ésta es la idea subyacente del algoritmo BPTT (Figura 2).

Para la aplicación de la primera etapa de Backpropagation (proceso forward) el cálculo de la salida y_i en el tiempo t de cada neurona se puede definir como:

$$y_i(t) = f(x_i(t)) \quad (1)$$

$$x_i(t) = \sum_{j \in H} y_j(t) w_{ij} + \sum_{j \in I} x_j^{in} w_{ij} + \sum_{j \in M} y_j(t - \tau_{ij}) w_{ij} \quad (2)$$

Donde f se refiere a la función de activación del modelo de la neurona, H denota los índices de las neuronas ocultas, I los índices de las neuronas de entrada, x_{in}^j es la j -ésima neurona de entrada, M denota los índices de las neuronas que almacenan la información de estados anteriores de la red (memoria de estado), $\tau_{ij} \geq 0$ es una magnitud entera que refiere el desplazamiento de las conexiones recurrentes en el tiempo.

Luego, en el proceso de retropropagación del error cada neurona j es caracterizada por una magnitud de error δ_j . Para las neuronas de la capa de salida esta magnitud se calcula teniendo en cuenta la diferencia entre el valor obtenido y el valor esperado (3) mientras que para las neuronas ocultas se calcula teniendo en cuenta el error de las neuronas sucesoras a la neurona en cuestión (4):

$$\delta_j(t) = (d_j - y_j) y_j (1 - y_j) \quad (3)$$

$$\delta_j(t) = y_j (1 - y_j) \sum_{i \in \text{Suc}(j)} w_{ji} \delta_i \quad (4)$$

Por su parte la actualización de los pesos se realiza de acuerdo a la formula (5) donde se calcula el cambio necesario en los pesos teniendo en cuenta la tasa de convergencia α .

$$\Delta w_{ij}^{e+1} = \alpha \delta_j y_j \quad (5)$$

En la fórmula, e se refiere al ordenamiento sucesivo con el que se realiza las actualizaciones de los pesos en el proceso de entrenamiento.

Una vez aplicado el algoritmo *Backpropagation* en la red FFN obtenida corresponde la aplicación de la segunda etapa del algoritmo BPTT: el plegamiento de la red (*folding*). En esta etapa la red se contrae para obtener la red recurrente original, donde los pesos de las conexiones son resultado del promedio de las conexiones equivalentes en cada tiempo de la red FFN anterior.

Como resultado del proceso de entrenamiento con BPTT se obtiene una RNR con el conocimiento de las base de casos que se utilizó para el aprendizaje, representado en los pesos de las conexiones, que son los parámetros libres que permiten adaptar cierta topología a un problema de aplicación particular.

Modificaciones a BPTT para trabajar con instancias de longitud variable

El problema de clasificación de secuencias de la transcriptasa reversa del VIH se distingue por la característica de que las instancias de las bases de casos tienen longitudes variables. Al alinear estas secuencias de longitudes variables y rellenar con valores nulos los segmentos que no

contienen información, el algoritmo BPTT tradicional antes propuesto no ofrece buenos resultados, ya que los valores perdidos se traducen como ceros y este valor no constituye un neutro ni para el modelo ni para el contexto del problema de aplicación. Por lo que es necesaria la adaptación de la RNR para que sea capaz de ignorar los segmentos nulos de cada secuencia y adaptar la cantidad de tiempos de la red al tamaño real de la instancia de forma dinámica dentro del proceso de entrenamiento.

El primer paso del BPTT para tiempos dinámicos consiste en caracterizar cada neurona como válida o inválida de la siguiente forma:

- Una neurona de entrada es inválida si el valor del rasgo (posición de la secuencia para el problema de aplicación) es nulo.
- Una neurona oculta es inválida si todas las conexiones que entran a ella provienen de neuronas inválidas.
- Las neuronas de salida siempre son válidas.

Posteriormente, en el proceso de actualización de los pesos sólo se tendrán en cuenta aquellas conexiones cuyas neuronas son válidas, de esta forma se conserva el conocimiento que las instancias anteriores aportaron a la red y la instancia actual sólo aporta la información consistente. Además si todas las neuronas de la capa de entrada de un tiempo determinado son inválidas la salida de este tiempo se ignora, adaptando así la cantidad de tiempos al tamaño de la instancia de forma dinámica dentro del proceso de entrenamiento. De esta manera en cada iteración del algoritmo (cada vez que entra una nueva instancia a la red para ser procesada) se va a activar la parte de la red que propaga información relevante.

En la Figura 2 se muestra el ejemplo de una RNR desplegada con una topología de tres tiempos con dos bloques de contexto: uno con recurrencia al pasado y otro con recurrencia al futuro, para una secuencia de entrada dividida en tres partes, una para cada tiempo. La instancia posee 12 rasgos donde las primeras 4 posiciones de la secuencia no están codificadas (tienen valores nulos). Siguiendo los criterios anteriores todas las neuronas de la primera capa de entrada son inválidas porque los valo-

res de los atributos asociados a ella son cero. Las neuronas de las capas ocultas que están conectadas directamente a las entradas inválidas también se afectan propiciando que la salida del tiempo $t = 1$ no se tome en cuenta para procesar la salida general del modelo. El algoritmo termina cuando el error medio de entrenamiento es menor que un umbral especificado o cuando alcanza el máximo número de iteraciones.

Funciones de agregación de salidas

Luego de definir la serie de pasos del algoritmo para procesar una instancia, resta definir cuál es la salida final de la red. Como se muestra en la Figura 3 la salida es un vector de varios componentes que contiene las probabilidades de pertenencia a cada clase por cada uno de los tiempos de la red. Pueden existir diferentes variantes para agregar estos valores para obtener una única salida. En este trabajo se tienen en cuenta cuatro de ellas:

1. *Average*: calcula un promedio de las probabilidades de pertenencia a cada clase de todas las salidas.
2. *Weighted Average*: calcula el promedio anterior pero ponderando las salidas de cada tiempo en función de la cantidad de neuronas válidas que posee su entrada, es decir, mientras más neuronas válidas tenga un tiempo más peso se le da a su salida en el cálculo del promedio.
3. *Mode*: calcula la moda de las salidas de todos los tiempos, de esta manera se oferta el valor que fue más frecuente.
4. *Middle*: devuelve el criterio de la salida del tiempo que se encuentra en el medio de la red si este es válido, para topologías compuestas por pocos tiempos (aproximadamente entre 3 y 7) puede ofrecer buenos resultados porque precisamente en esta zona de la red es donde mayor información se acumula proveniente de todas las partes de la secuencia.

Es importante señalar que al igual que en el proceso de entrenamiento sólo se tendrán en cuenta los criterios de salida de aquellos tiempos que sean válidos, independientemente de la función de agregación que se utilice.

Secuencia de una mutación de la transcriptasa reversa

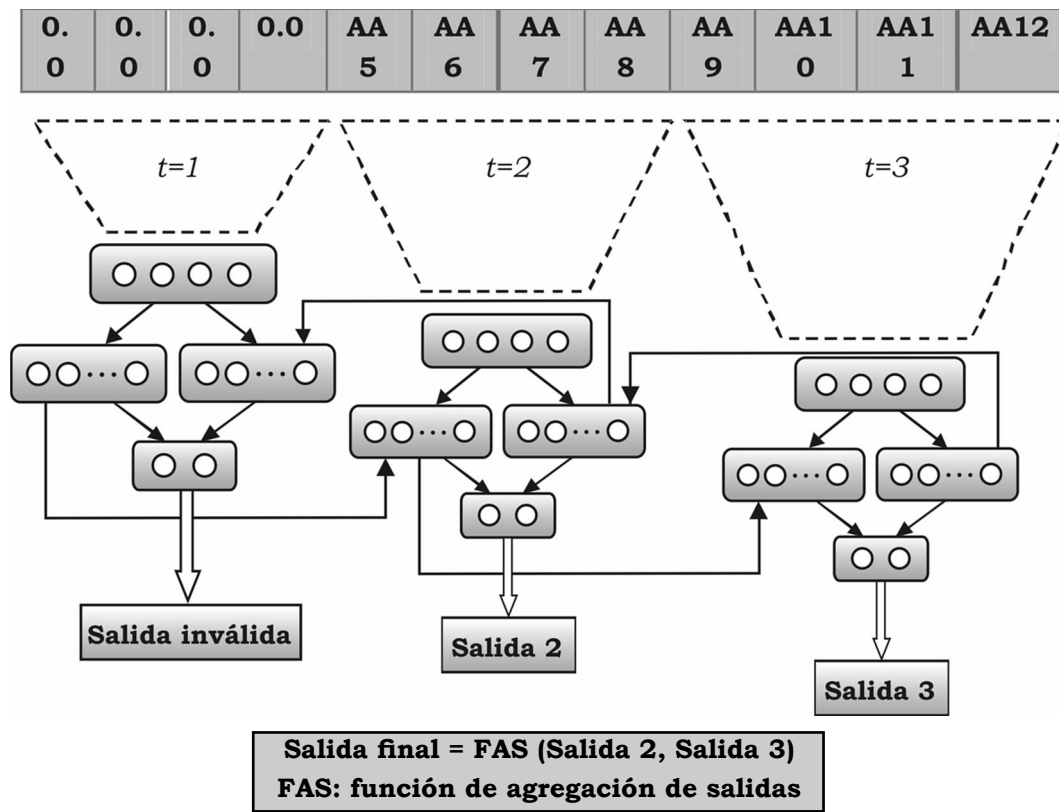


Fig. 2 RNR desplegada que ajusta los tiempos dinámicamente de acuerdo al tamaño de la instancia.

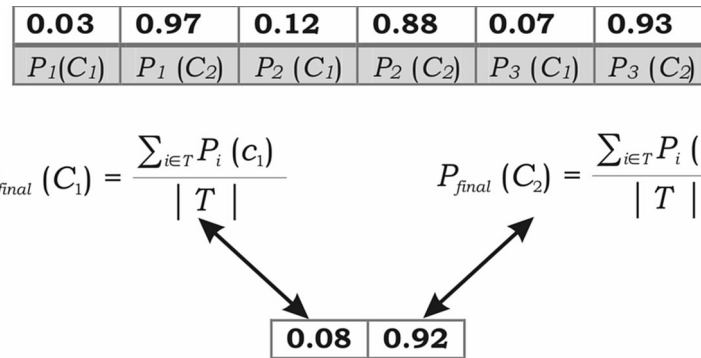


Fig. 3 Función de agregación de salida Average.

RESULTADOS OBTENIDOS

Se realizó un experimento donde se utilizaron 11 bases de casos obtenidas en Stanford HIV Resistance Database que representan la resistencia de mutaciones de la transcriptasa reversa a los siguientes fármacos inhibidores: lamivudina (3TC), abacavir (ABC), zidovudina (AZT), estavudina (d4T), zalcitabina (ddC), didanosina (ddI), delavirdina (DLV), efavirenz (EFV), emtricitabina

(FTC), nevirapina (NPV) y tenofovir (TDF). Previamente se alinearon todas las secuencias de las bases para hacer corresponder las posiciones del código genético de cada cepa del virus, cada posición de los fragmentos no secuenciados se denotan por el número cero.

En el experimento se analizaron los resultados del aprendizaje de la variante original del algoritmo BPTT utilizando la función de agrega-

ción de salidas average contra los resultados de la aplicación del BPTT con tiempos dinámicos utilizando las cuatro funciones de agregación de salidas implementadas: average, weighted average, mode y middle. Con este propósito se aplicaron 20 procesos de cross-validation de 10 folds para cada una de las bases ofreciendo los siguientes resultados (Figura 4).

Como se puede observar en la Figura 4, para la mayoría de las bases las variantes de BPTT para tiempos dinámicos ofrecen mejores resultados. Particularmente la modificación que utiliza la función de agregación de salidas average alcanza los mejores porcentajes de clasificación, en algunas bases supera en 10 puntos porcentuales al promedio alcanzado por la variante original de BPTT.

En la Figura 5 se observa un promedio de los porcentos de clasificación alcanzados por cada algoritmo en todas las bases, lo cual refuerza las diferencias entre los resultados alcanzados, los cuales, como promedio, refuerzan el planteamiento anterior de que los resultados con tiempos dinámicos son mejores.

Promedio de los porcentos alcanzados para todas las bases por cada algoritmo

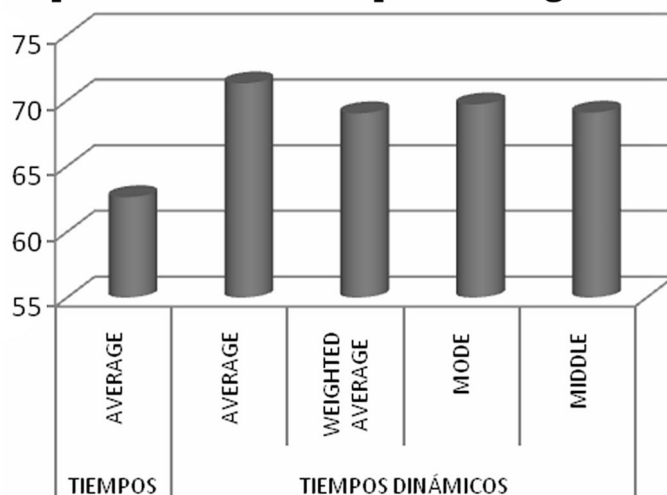


Fig. 5 Promedios de los porcentos de casos bien clasificados para la variante tradicional de BPTT y las cuatro modificaciones propuestas para trabajar con instancias de longitud variable.

Promedios de los porcentos de clasificación obtenidos para las 11 bases con cada una de las modificaciones del algoritmo

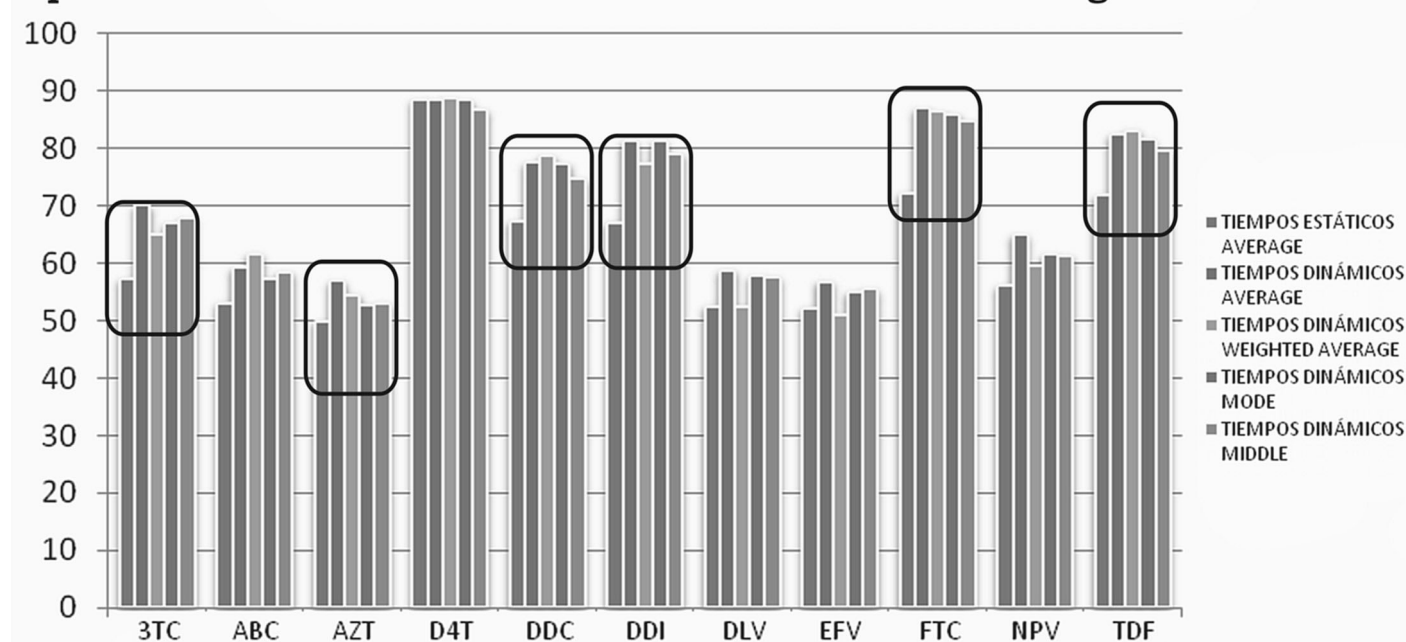


Fig. 4 Promedios de los porcentos de casos bien clasificados para la variante tradicional de BPTT y las cuatro modificaciones propuestas para trabajar con instancias de longitud variable.

CONCLUSIONES

En este trabajo se implementó una RNR utilizando el algoritmo de entrenamiento *BPTT* y se incorporó a la plataforma Weka como un nuevo clasificador. Además se adaptó el algoritmo de entrenamiento para que permita ajustar dinámicamente los tiempos de la red y de esta manera ser capaz de trabajar con instancias de longitud variable.

La modificación propuesta se utilizó en el entrenamiento de RNRs para predecir la resistencia de secuencias de la transcriptasa reversa ante 11 fármacos antirretrovirales del VIH. Los resultados obtenidos se compararon con los de la variante inicial del BPTT, resultando los de la modificación propuesta significativamente mejores.

Referencias Bibliográficas

- Baldi, P. 2002. New Machine Learning Methods for the Prediction of Protein Topologies, In: P. Frasconi and R. Shamir (eds.) Artificial Intelligence and Heuristic Methods for Bioinformatics, IOS Press.
- Bonet, I. 2008. Modelo para la clasificación de secuencias, en problemas de la bioinformática, usando técnicas de inteligencia artificial. PhD, Universidad Central "Marta Abreu" de Las Villas.
- Hilera, J. R. & Martínez, V. J. 1995. Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones, Madrid / Mexico, RA-MA / Addison Wesley Iberoamericana.
- Kierczak, M., Ginalski, K., Damiński, M., Koronacki, J., Rudnicki, W. & Komorowski, J. 2009. A Rough Set-Based Model of HIV-1 Reverse Transcriptase Resistance. Bioinformatics and Biology Insights.
- Pearlmutter, B. 1990. Dynamic Recurrent Neural Networks. DARPA Research.
- Rhee, S., Taylor, J., Wadhera, G., Ben-hur, A., Brutlag, D. & Shafer, R. 2006. Genotypic predictors of human immunodeficiency virus type 1 drug resistance. Proceedings of the National Academy of Sciences of the United States of America.
- Ronald, J. W. & David, Z. 1995. Gradient-based learning algorithms for recurrent networks and their computational complexity. In: Chauvin, Y. & Rumelhart, D. E. (eds.) Backpropagation: theory, architectures, and applications. Lawrence Erlbaum Associates, Inc.
- Salazar, S. 2005. NEngine v 1.0: Una Herramienta Software para Redes Neuronales Recurrentes. Lic, Universidad Central "Marta Abreu" de Las Villas.
- Vermeiren, H., Van Den Bulcke, T., Van Marck, H., Lecocq, P., Van Houtte, M. & Bachelier, L. 2004. Application of multiple linear regression modelling to the quantitative prediction of HIV-1 drug susceptibility phenotype from viral genotype. XIII International HIV Drug Resistance Workshop.