PREDICCIÓN DE TRÁFICO DE VÍDEO VBR CON RED NEURONAL RECURRENTE

Carol Zuleimy Fernández Rodríguez.
Código: 20142005090
Inteligencia Computacional 2

1. Introducción

Las aplicaciones multimedia y, en particular, los flujos de vídeo de velocidad de bits variable VBR se están convirtiendo en componentes de tráfico importantes en las redes de alta velocidad y se ha recomendado ATM (modo de transferencia asincrónica) como vehículo de transporte para las redes digitales de servicios integrados de banda ancha (BISDN). Una aplicación importante en las redes ATM es proporcionar una transmisión en tiempo real, de baja pérdida y con un retardo mínimo de tráfico de velocidad de bits variable (VBR), que tiene muchas ráfagas, no es estacionario y está correlacionado. Para asegurar una buena calidad de servicio en este tipo de aplicaciones, es necesario conocer de antemano el tráfico para gestionar los recursos de la red, es decir, predecir el tráfico.
En este proyecto, adoptamos la metodología de redes neuronales para predecir el tráfico VBR representado en muestras de dos películas codificadas en MPEG 4, donde a partir de una trama, se predicen las 5 siguientes; se desarrollan y evalúan un conjunto de redes neuronales Recurrentes, donde se varía su arquitectura, funciones de activación y funciones de entrenamiento, para presentar una comparación de los resultados de cada red. Finalmente se presenta la red que se considera que mejor soluciona el problema, se compara con los mejores resultados de lo presentado en la Entrega 1, el cual se realizó con red neuronal Feed Froward y así se determina cuál es más conveniente de implementar.

2. Planteamiento del problema

La predicción del tráfico de red en redes integradas de banda ancha, como Internet y las redes de modo de transferencia asincrónica (ATM), ha atraído cada vez más actividades de investigación. Dado que diversas aplicaciones multimedia como videoconferencia, video a pedido y audio se vuelven cada vez más importantes en Internet; la asignación y administración del ancho de banda de la red, así como el control del tráfico, se han vuelto más complicados y desafiantes. El control de la red, por un lado, debe admitir diferentes tipos de clases de tráfico con su calidad de servicio requerida y, por otro lado, debe utilizar de manera eficiente los recursos de la red, como el ancho de banda de la red.
En general, hay dos categorías de aplicaciones de video de velocidad de bits variable (VBR), es decir, el video VBR pregrabado y las aplicaciones de video VBR en tiempo real. Para las aplicaciones de video pregrabadas, como video a pedido, todo el perfil de seguimiento de tráfico de video se puede analizar completamente con anticipación y determinar un ancho de banda efectivo cuando se entrega el video. Sin embargo, esto es difícil para situaciones de video en

vivo, porque cualquier transmisión de video en vivo no se conoce de antemano y, por lo tanto, estimar su ancho de banda estático efectivo sería difícil e inexacto.

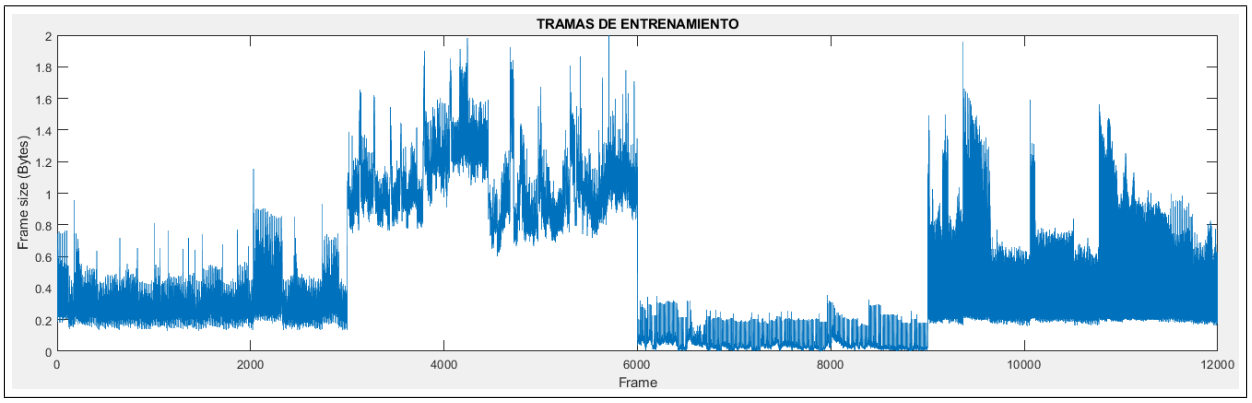

Figura 1: Muestra tráfico VBR para entrenamiento de la red

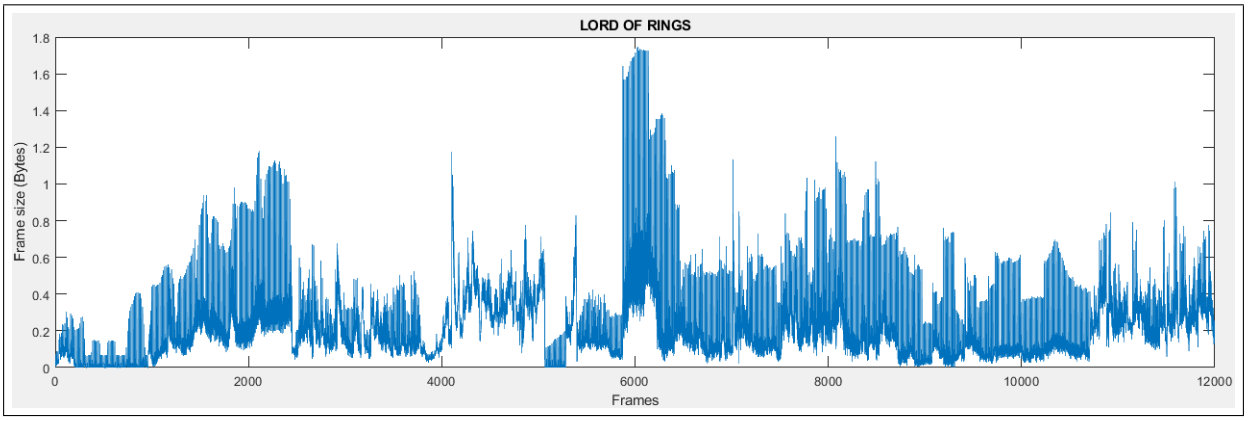

Figura 2: Muestra de tráfico VBR para prueba de red entrenada

Es por ello que la predicción del tráfico es importante para mejorar el funcionamiento de estas redes. Sin embargo, el tráfico de video VBR presenta altas fluctuaciones de tasa de bits durante períodos de tiempo cortos, lo que dificulta la predicción. Las redes neuronales se pueden utilizar eficazmente para superar este problema.
En la literatura, cuando se trata de problemas de análisis de secuencias, ya sea de texto, voz o vídeo como en este caso, se ha demostrado que las Redes Nuronales Recurrentes son las indicadas dadas sus propiedades de memoria para generar la salida o activación, ya que este tipo de red no sólo la entrada actual sino la activación generada en la iteración previa. Por lo que se buscará el mejor resultado usando este tipo de red neuronal, usando diferentes arquitecturas, funciones de activación y funciones de entrenamiento.

3. Desarrollo

3.1. Obtención de los datos

Los datos que se utilizarán serán muestras de tráfico de vídeo, las cuales están codificadas en MPEG-4. MPEG-4 comprende un patrón alterno de tres cuadros (intracuadro), (predictivo) y (bidireccional) con diferentes propiedades. Estos tres tipos de fotogramas se fusionan de forma determinista para formar la secuencia de vídeo MPEG-4 agregada. El tamaño de los fotogramas de vídeo varía drásticamente a medida que se genera la secuencia, lo que genera un tráfico de velocidad de bits variable (VBR).

Las dos muestras de tráfico de vídeo utilizadas, se codificaron en la Universidad Estatal de Arizona y se descargaron las tramas de su repositorio disponible en [1]. Se descargaron tramas de vídeo como se describen en el Cuadro 1, en el caso de los datos de entrenamiento se tuvieron en cuenta diferentes tipos de tráfico para dar versalitidad a los datos, es por ellos que se tuvieron en cuenta vídeos de películas, noticias y documentales, para finalmente validar con la película Lord of Rings, siendo estos los datos que se utilizaron para validar en la Entrega 1. Tanto los datos de entrenamiento como los de validación, se muestran en las Figuras 1 y 2.

TRAMAS DE ENTRENAMIENTO	**NÚMERO DE TRAMAS**
Silence of the Lambs	3000
NBC News	3000
Matrix 1	3000
Tokyo Olympics 1964	3000
TOTAL TRAMAS	12000
TRAMA DE VALIDACIÓN	**NÚMERO DE TRAMAS**
Lord of the Rings	12000

Cuadro 1: Tramas de vídeo utilizadas

3.2. Transformación de los datos para la Red Neuronal

La predicción de tráfico de solo una trama adelante, es una predicción de tráfico a muy corto plazo. Cabe señalar que, al momento de administrar de ancho de banda, el proceso de reasignación de ancho de banda en sí mismo necesita tiempo de cálculo y otros recursos, por lo que no es práctico realizar dicha reasignación dinámica de ancho de banda con demasiada frecuencia. En otras palabras, no es práctico debido a tanta sobrecarga. En vista de esto, la predicción de tráfico a corto plazo de una trama por delante no es suficiente, así que, como primer intento de realizar la predicción a un plazo más amplio, se realizará a 5 tramas a partir de una trama actual.
Para ello se deben transformar los datos de forma tal, que el conjunto sea de una entrada y cinco salidas, es decir, que las muestras de 12000 tramas se deben dividir en 6 filas:

CONJUNTO	**DIMENSIÓN**
ENTRADA	1 Fila - 2000 Columnas
SALIDA	5 Filas - 2000 Columnas

Cuadro 2: Dimensión de los datos

3.3. Construcción Red Neuronal Recurrente

Una red neuronal recurrente se ve bastante similar a una red neuronal tradicional, excepto que se agrega un estado de memoria a las neuronas. Para ello Usan el concepto de recurrencia, donde para generar la salida o activación, la red usa no sólo la entrada actual sino la activación generada en la iteración previa, en pocas palabras, las redes neuronales recurrentes usan un cierto tipo de memoria para generar la salida deseada. Es por ello que son capaces de procesar diferentes tipos de secuencias como vídeos, porque son secuencias que llevan un orden y la siguiente imagen en la secuencia de vídeo dependerá de la imagen anterior e incluso estan relacionadas con aquellas que se presenten más adelante en la secuencia.

La Figura 3 muestra la arquitectura de una red Recurrente con cada capa representando las observaciones en un cierto tiempo t, donde Xt son las entradas, Ot las salidas y St es la activación que se transmite a la siguiente neurona.

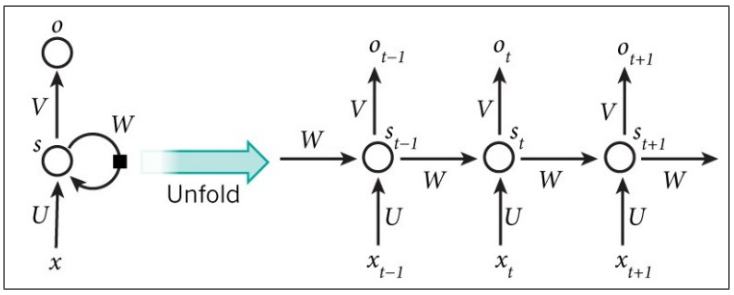

Figura 3: Red Feed Recurrente

Las redes neuronales que se construyen en este proyecto, se realizan en MATLAB, ya que posee herramientas, funciones y toolboxes especializadas para trabajar redes neuronales.

De acuerdo a las funciones de MATLAB con las que se pueden configurar las redes, en aras de experimentar con diferentes modelos para obtener el mejor, se creará un conjunto de redes neuronales donde se variará su estructura de capas y neuronas por capa, para dos funciones de activación diferentes y así mismo para tres funciones de entrenamiento, tal somo se describe en la Figura 4.

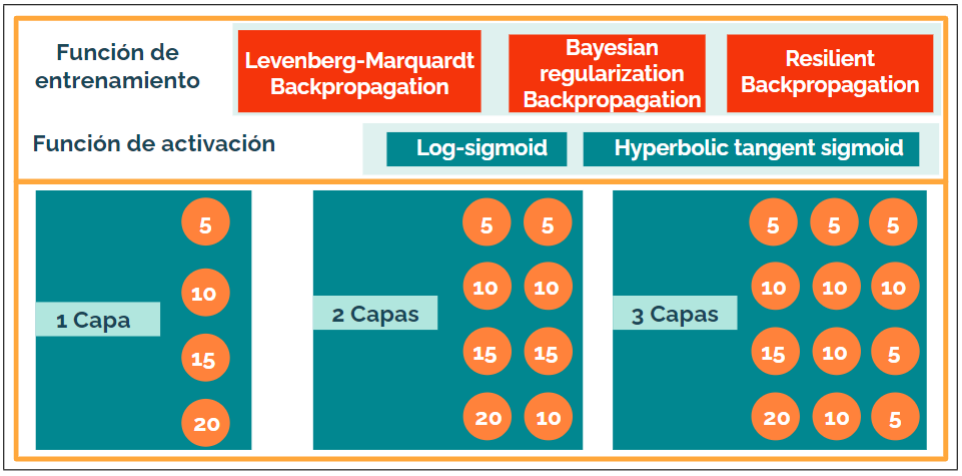

Figura 4: Metodología de experimentación

3.4. Entrenamiento de la red

Además de las características descritas anteriormente, se ajustarán unos parámetros de la red que estarán fijos en todos los casos, tales como la arquitectura, cuya función *layrecnet* define la red neuronal recurrente, la división de los datos y criterios de parada del entrenamiento:

```
%ARQUITECTURA
net=layrecnet(1:2,[10 10],'trainlm');
%Division de datos
net.divideParam.trainRatio = 0.7; %Entrenamiento
net.divideParam.valRatio   = 0.2; %Validación
net.divideParam.testRatio  = 0.1; %Test
%Parámetros de entrenamiento (criterios de parada)
net.trainParam.max_fail = 10;      %Fallos máximos de validación
net.trainParam.epochs =   1000;    %Número máximo de épocas a entrenar
net.trainParam.goal =     0;       %Objetivo de rendimiento
net.trainParam.min_grad = 1e-10;   %Gradiente de rendimiento mínimo
```

Figura 5: Arquitectura, división de los datos y criterios de parada

La entrada de la muestra de tráfico que se usará para entrenar la red, la cual se muestra en la Figura 1, se denotará como x y las 5 salidas como t; de esta forma, utilizando la función

train(net,x,t), se ingresan los datos mencionados.
Finalmente, para observar los resultados de la red entrenada con los datos ingresados, se usa la función *sim(net,x)*, para simular la red con los datos de entrada *x*.

3.5. Validación de la red entrenada

La muestra de tráfico que se usará para validar la red entrenada, será una muestra distinta a la usada en el entrenamiento, para verificar que funciones bien ante datos nuevos, en este caso se usará la de *Lord of the Rings*, la cual se muestra en la Figura 2, además fue la que se usó en la Entrega 1, por lo que es adecuado usarla y así comparar. Para ello, sólo se usa la función *sim(net,xx)*, para simular la red entrenada anteriormente, con los datos de entrada de la muestra de validación, que se denota como *xx*.

3.6. Funciones de desempeño

El entrenamiento, cuyo objetivo es que la red neuronal sea capaz de reproducir el comportamiento subyacente en los datos aportados, consiste básicamente en la minimización de una función de coste o error, lo que equivale a que la salida de la red, se aproxima a la salida en los datos. Entre las funciones más usadas para problemas de este tipo, así como en la literatura de referencia, están el MSE y el MAE, los cuales están definidos en sus respectivas ecuaciones.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(yi - yi')^2 \tag{1}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|yi - yi'| \tag{2}$$

Donde n es el número de datos *yi* representa los datos observados y *yi'* representa los que predice la red neuronal.
Para la optimización de la red neuronal, un método muy usado es el del descenso del gradiente, donde calculan la variación del error al variar cada uno de los parámetros y luego modifican todos los parámetros de la red neuronal obteniendo un error menor. Se puede decir que es una búsqueda en serie de la solución o mínimo global. MATLAB cuenta con las funciones de entrenamiento, las cuales realizan dicho procedimiento y obtienen el gradiente resultante del entrenamiento de la red. Las que se usarán en este caso se ilustran en la Figura 4, en MATLAB, son denotadas como: *trainlm, trainbr y trainrp*, respectivamente.

4. Resultados

Se recopilaron todos los resultados de MSE, MAE y gradiente para la trama de entrenamiento y los resultados de MSE y MAE para la trama de validación. Para todos los casos se realizaron 5 experimentos, cuyos resultados al final se promediaron para obtener un único resultado. Las redes neuronales se variaron en cuanto a número de capas, capas por neurona, funciones de transferencia y funciones de entrenamiento, tal como se ilustró en la Figura 4. Finalmente, se graficaron los resultados con la trama de entrenamiento:

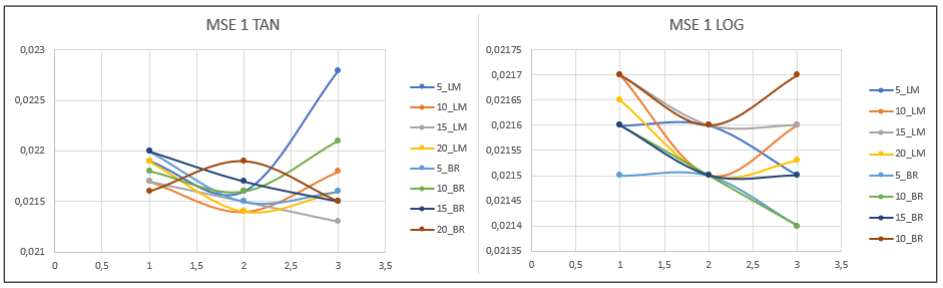

Figura 6: MSE trama de entrenamiento de todas las redes con función de transferencia Tangencial y Logarítmica

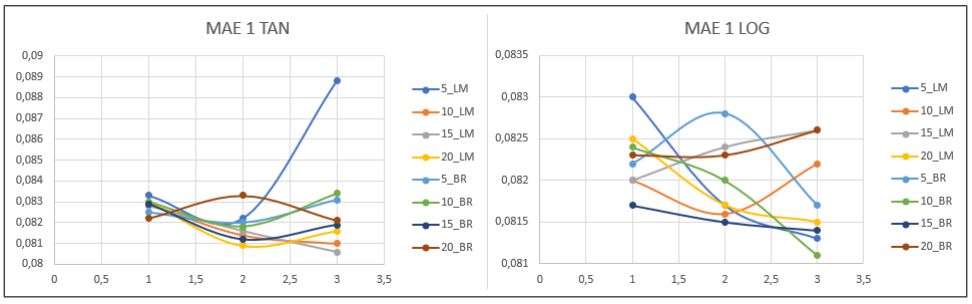

Figura 7: MAE trama de entrenamiento de todas las redes con función de transferencia Tangencial y Logarítmica

Así mismo, se grafican los de la trama de validación:

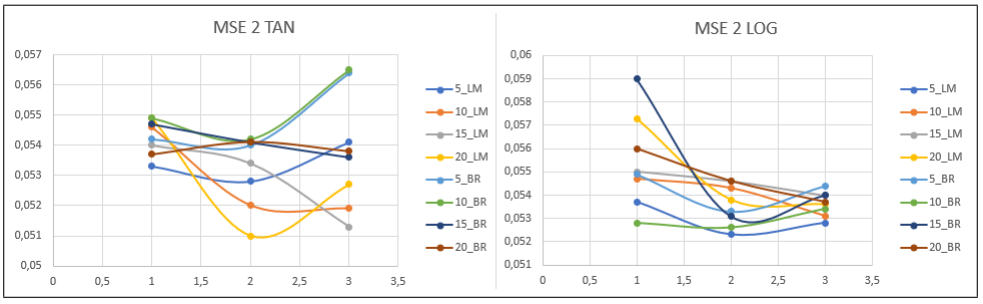

Figura 8: MSE trama de validación de todas las redes con función de transferencia Tangencial y Logarítmica

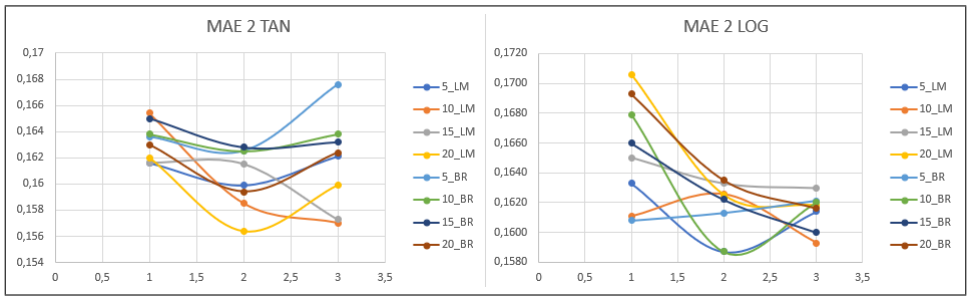

Figura 9: MAE trama de validación de todas las redes con función de transferencia Tangencial y Logarítmica

En las gráficas ilustradas anteriormente, se compilan los resultados de todas las redes con los diferentes números de neuronas y funciones de entrenamiento; tanto para la función de transferencia tangencial y logarítmica, donde el eje x indica el núcero de capas de cada una.

De acuerdo a ello, lo más notable de los resultados, es que con 2 y 3 capas hubo un mejor desempeño en general, sin embargo, el tiempo de entrenamiento para 3 capas es demasiado alto en comparación con 2 capas, por lo tanto, en ese sentido se tendrán en cuenta los mejores resultados con dos capas. A partir de ello, se visualizaron las neuronas que obtuvieron los errores numéricamente más bajos en general, es decir, en su MSE y MAE para las dos tramas simuladas. Las dos redes que obtuvieron los mejores resultados son las siguientes:

4.1. Mejores Soluciones

RED 1		**RED 2**	
Número de capas	2	**Número de capas**	2
Neuronas por capa	10-10	**Neuronas por capa**	20-10
Función entrenamiento	Levenberg Marquardt	**Función entrenamiento**	Levenberg Marquardt
Función de transferencia	Tangencial	**Función de transferencia**	Tangencial
MSE trama entrenamiento	2.14 %	**MSE trama entrenamiento**	2.13 %
MSE trama validación	5.2 %	**MSE trama validación**	5.1 %
MAE trama entrenamiento	8.14 %	**MAE trama entrenamiento**	8.09 %
MAE trama validación	15.85 %	**MAE trama validación**	15.64 %
Tiempo entrenamiento	77 s	**Tiempo entrenamiento**	180 s

Cuadro 3: Resultados mejores redes

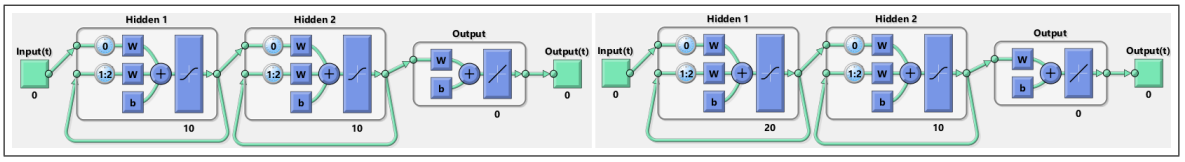

Figura 10: Arquitectura de las mejores redes

De acuerdo a lo descrito en el Cuadro 3, los errores entre ambas redes son muy similares, sólo que para la red con 20-10 neuronas, el tiempo de entrenamiento es mucho mayor; es por ello que teniendo en cuenta el tiempo y la simplicidad del modelo en cuanto a su arquitectura, se tomará la red de 10-10 neuronas como la mejor solución en esta experimentación. Sin embargo, el tiempo que toma la red seleccionada, tampoco es muy bueno y no la hace adecuada para una aplicación de predicción de vídeo en tiempo real.

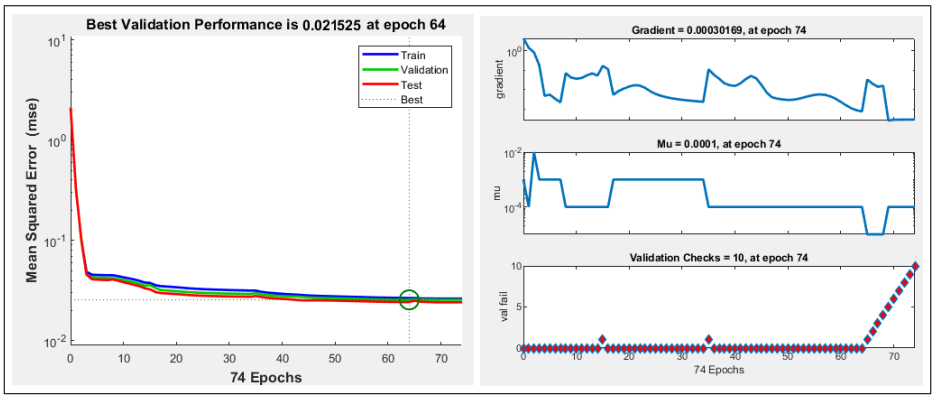

Figura 11: Performance y gradiente mejor red

Los resultados de las tramas de simulación se vuelven a acomodar de forma tal de que vuelva a ser de una sola fila y así poder graficarlos sobre las tramas originales para comparar. Al ser una trama demasiado larga, se hacen dos acercamientos para observar mejor los resultados:

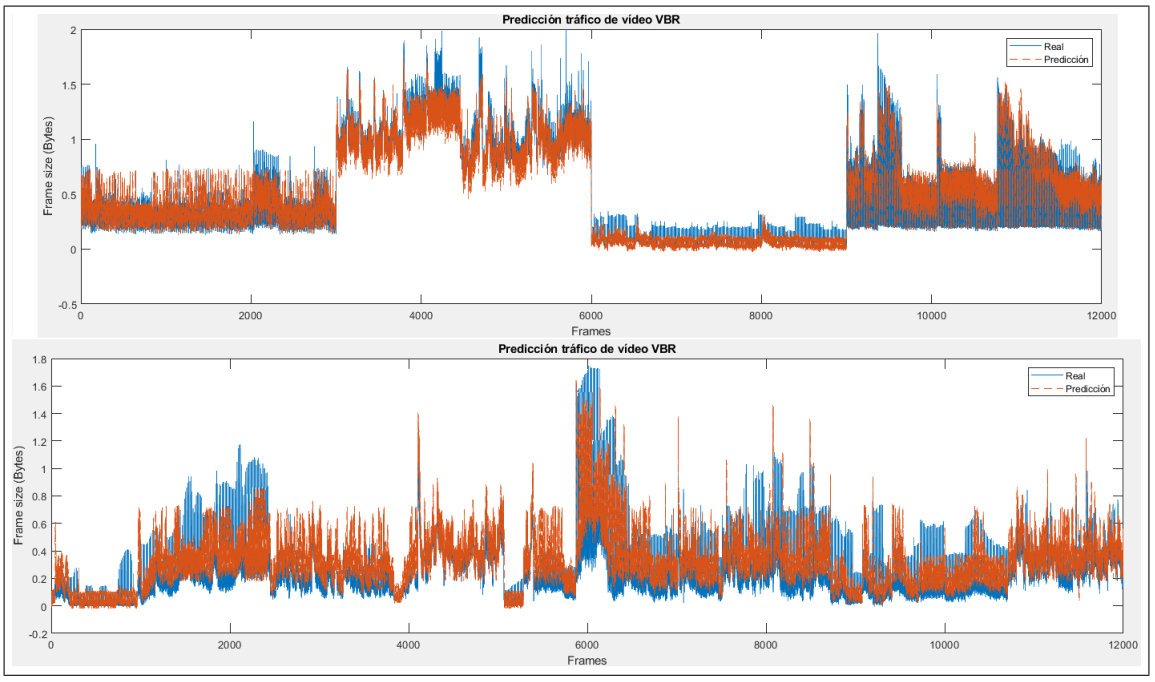

Figura 12: Comparación de resultados de las dos tramas con las originales

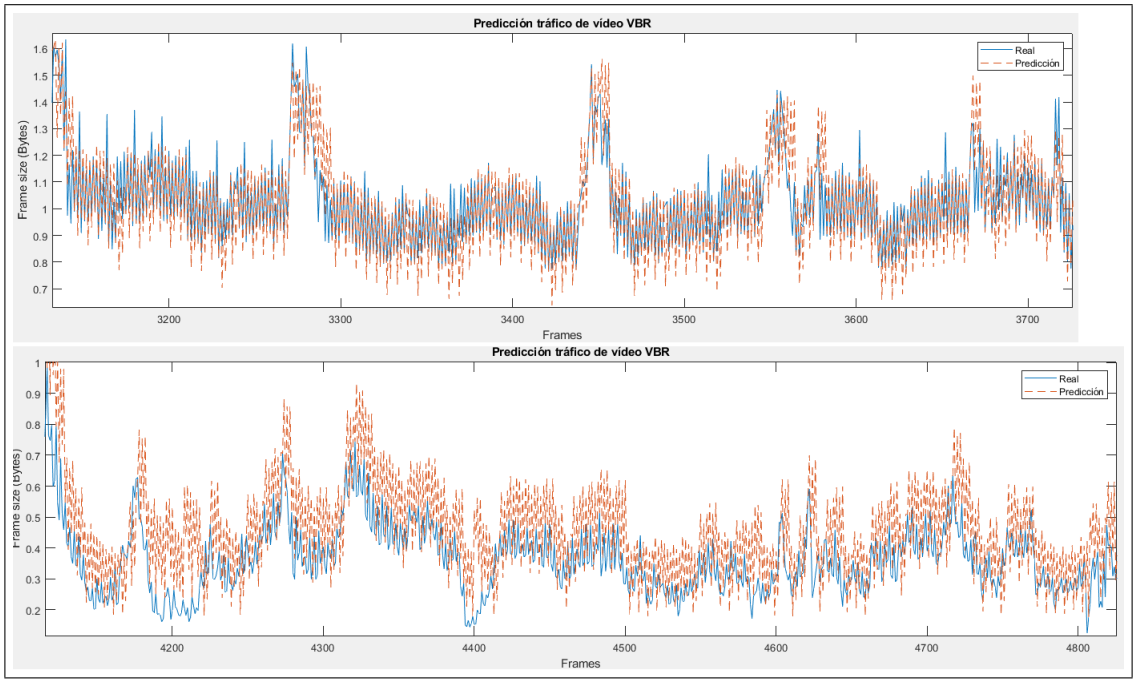

Figura 13: Acercamiento 1

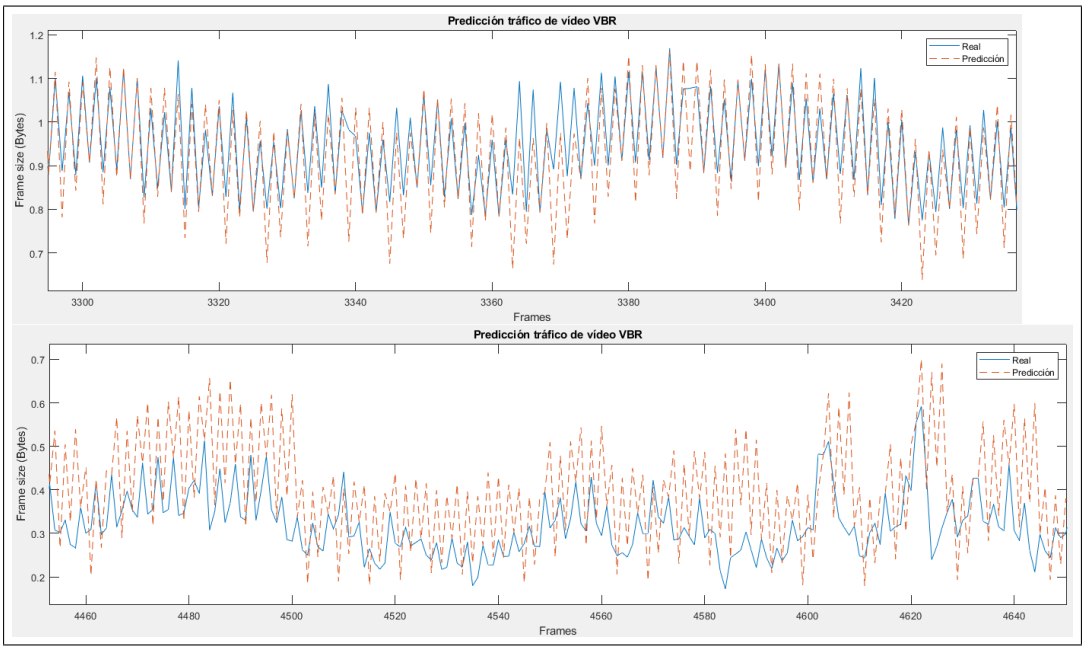

Figura 14: Acercamiento 2

En vista de la Figura 11, a pesar de ser los mejores resultados, no logra predecir adecuadamente las zonas donde el tráfico varía con más amplitud, sin embargo, sigue la forma del tráfico en general de buena manera aunque no sea exacto. Tanto en la Figura 12 como en la 13, se observa que los resultados de la trama usada en el entrenamiento (la ubicada en la parte superior) son más cercanos a la trama real que la trama usada de validación, cuyas razones son evidentes, siendo esta última trama, datos nuevos para la red entrenada.

5. Comparación con resultados de Entrega 1

Para finalizar, se hace una comparación entre la mejor red de esta entrega de red neuronal Recurrente, con la mejor red Feed Forward de la entrega 1, para determinar cuál de las dos resulta mejor para el proyecto realizado. Valga aclarar que son resultados a partir de un promedio entre 5 experimentos.

PARÁMETROS	**RED FEED FORWARD**	**RED RECURRENTE**
Arquitectura	2 capas: 10-10 neuronas	2 capas: 10-10 neuronas
Función de entrenamiento	Bayesian Regularization	Levenberg Marquardt
Función de transferencia	Tangencial	Tangencial
MSE Entrenamiento	1.36 %	2.14 %
MSE Validación	5.25 %	5.2 %
MAE Entrenamiento	4.32 %	8.14 %
MAE Validación	14.55 %	15.85 %
Tiempo entrenamiento	0s-1s	52 s

Cuadro 4: Comparación merores redes Entrega 1 y 2

En vista de la comparación realizada, en cuanto al los errores, ambas redes tienen un desempeño similar, aunque de parte de la red Feed Forward hay errores un poco más bajos en los datos de validación; sin embargo, el detalle que marca la diferencia es el tiempo de entrenamiento que toma cada una, ya que la red Recurrente se demora demasiado para ser utilizada en una aplicación de predicción en tiempo real, en cambio, la red Feed Forward es

muy rápida e ideal para este caso en particular. Es por ello que entre ambas redes, la que mejor responde al problema de este proyecto es la red Feed Forward. Sin embargo el error sigue siendo elevado y debe mejorarse.

6. Conclusiones

Se propuso un conjunto de 72 redes neuronales Recurrentes con diferentes configuraciones, para la predicción de tráfico de Tasa de Bits Variable (VBR), de las cuales se observó un mejor comportamiento en las configuraciones de 2 capas.
El entrenamiento de una red neuronal recurrente debe prolongarse para cada paso temporal, lo que es muy costoso en tiempo de proceso y memoria RAM, lo cual no lo hace una red indicada para la aplicación que se busca en este proyecto, por lo que entre las dos entregas, fueron mejores los resultados con Feed Forward.
Es necesario seguir explorando más combinaciones de redes, así como otras configuraciones en sus parámetros tanto en arquitectura, como en la búsqueda funciones de entrenamiento que se depempeñen mejor, ya que, aún se observa que falta una mejor predicción en las zonas donde el tamaño entre tramas es más variable. Puede que sea necesario aumentar el tamaño de datos de entrenamiento y añadir muchos más tipos de tráfico para que el entrenamiento sea más robusto y pueda generalizar mejor.
Por otra parte, el reto también está en realizar una predicción a un plazo más amplio que 5 tramas, sin embargo, la obtención de una red que logre la predicción adecuadamente será más compleja que la actual, por lo tanto, es el objetivo a seguir una vez se haya mejorado la predicción tratada en este proyecto.

Referencias

[1] P. Seeling, M. Reisslein, and B. Kulapala. Network Performance Evaluation with Frame Size and Quality Traces of Single-Layer and Two-Layer Video: A Tutorial. IEEE Communications Surveys and Tutorials, Vol. 6, No. 3, pages 58-78, Third Quarter 2004.

[2] Ramakrishnan, N., Soni, T. (2018, December). Network traffic prediction using recurrent neural networks. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 187-193). IEEE.

[3] Prasad, S. C., Prasad, P. (2014). Deep recurrent neural networks for time series prediction. arXiv preprint arXiv:1407.5949.

[4] Liang, Y. (2004). Real-Time VBR Video Traffic Prediction for Dynamic Bandwidth Allocation. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), 34(1), 32–47.

[5] Moh, W. M., Chen, M.-J., Chu, N.-M., Liao, C.-D. (1995). Traffic prediction and dynamic bandwidth allocation over ATM: a neural network approach. Computer Communications, 18(8), 563–571.

[6] Gavade, J. D., Kharat, P. K. (2011). Neural network based approach for MPEG video traffic prediction. 3rd International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2011).