Applications Of Neural Networks To Telecommunications 3, pp. 104-111
Lawrence Erlbaum Associates , Publishers, New Jersey, 1995

Generation of ATM Video Traffic Using Neural Networks

E. Casilari, A. Reyes, A. Díaz-Estrella, F. Sandoval
Dpto. Tecnología Electrónica
E.T.S.I. Telecomunicación,Universidad de Málaga, Campus de Teatinos, 29071 Málaga, Spain
casilari@dte.uma.es
arcadio@dte.uma.es

Abstract

A new model to generate Asynchronous Transfer Mode (ATM) video traffic is presented. The model, implemented on neural networks, is capable of accurately adjusting the autocorrelation and probability distribution functions of a given video traffic. This adjustment is performed by capturing the projected conditioned histogram of the real traffic, so that the neural model will be able to yield a simulated as a function of an input white noise. Using neural networks we benefit from their inherent capacities for working in real time, because of their parallelism, and interpolating unknown functions. Results are presented for a real MPEG video source.

1 Introduction

The aim of the future broadband integrated service digital network (B-ISDN), based on asynchronous transfer mode (ATM), is supporting a wide variety of multimedia services, with different statistical characteristics and quality of service requirements (delay, losses). Among the most emerging services, we have the video and image transmissions, which include many types of visual media, as still image, video-conferencing, broadcast TV, HDTV, etc.

The statistical multiplexing that ATM provides is specially appropriate to transmit these traffics in real time efficiently, as long as this packet transfer mode optimizes the use of the available bandwidth in the network. However, due to the particular complexity of video statistical characteristics, some issues, as the modelling and analysis of this traffic, are still open in the ambit of ATM communications. Thus, finding an universal model for simulating packet video traffic is a difficult task because the traffic flow depends not only on the used encoding methods (JPEG, MPEG, etc.), but also on the nature (resolution, motioned scenes, etc.) of the image that is being transmitted.

So, different solutions such as Markov chains [1] [2], autorregresive(AR, ARMA), gamma distributed models, have been proposed to model video traffic. The general objective of these models is to approximate the probability density distribution, as well as the autocorrelation function of a real video traffic, since the apparition of correlation between frames is an essential characteristic that distinguishes VBR video from other types of traffics.

But most of the proposed models are suitable just for modelling simple video scenes, with low complexity, or, otherwise, they focus their approximation on one of those two statistical functions, assuming that the other one follows a determined pattern (decreasing autocorrelations, gaussian probability distributions). In this letter a new model for the generation of ATM traffic video is presented.

The model, implemented on neural networks, is capable of accurately adjusting the functions of autocorrelation and probability distribution of a given video traffic. This adjustment is performed by capturing the projected conditioned histogram of the real video traffic, so that the neural model will be able to yield a simulated video signal at its output, just as a function of an input white noise. Moreover, using neural networks we benefit from their inherent capacities for working in real time, because of their parallelism, and interpolating unknown functions. These interpolations avoid the need of storing the histogram and substitute the searching in matrices of other histogram-based methods [3].

2 Theoretical Formulation

The final objective of a traffic emulator is to generate a random series s'[n], with a physical meaning (e.g.: time between cells, number of generated cells in a period of time), which imitates the behaviour of a real traffic stream s[n]. This imitation can be performed for different time levels regarding the source activity (cell level, burst level, call level). Our model carries out its approximation for the burst level, taking into account the fixed period corresponding to a frame. Modelling the source for the cell level is not so relevant, because the existence of shaping and buffers in the network deforms the traffic flow for short-term considerations. So the real traffic stream s[n] indicates the number of cells that each video frame contains.

In order to simulate this real traffic, the flow s'[n] must fit the curves of the autocorrelation function $R_s[n]$ as well as the probability distribution $F_s(x)$ of s[n]. Higher order statistics can be neglected [4] as they are not representative of the traffic for queuing solutions. The proposed emulator performs the adjust of both functions by fitting the projected conditioned histogram of the real video traffic. If we define:

$$F_s(x) = \Pr ob\{s[n] < x \,/\, s[n-d(1)], s[n-d(2)], \ldots s[n-d(i)], \ldots, s[n-d(K)]\}$$

as the distribution function of s[n] conditioned to K previous samples determined by an index d(i) ($i\in[0,K]$) denoting the lag, this function could be empirically obtained by calculating a K+1 dimensional histogram, consisting in a transition matrix.

The estimation of this histogram requires to divide the signal s[n] in a number N of discrete levels. The larger this number N is chosen, the more accurate the histogram is, but more samples of the traffic are needed for a reliable and representative histogram.

The obtained distribution $F_s(x)$ is a continuous function ranging from 0 to 1. Thus it can be regarded as an application of the K+1 dimensional space {s[n],s[n-d(1)],... s[n-d(i)],... s[n-d(K)]} in the interval $\xi\in$ [0,1].

$$F_s\!:\!\{s[n], s[n-d(1)], s[n-d(2)], \ldots s[n-d(K)]\} \xrightarrow{F_s} \xi \in [0,1]$$

Knowing that $F_s(x)$ is a monotonously increasing function between 0 and 1, the inverse function F_s^{-1} can be calculated, so that we could obtain the current sample s[n] as a function of the past traffic {s[n],s[n-d(1)],... s[n-d(i)],... s[n-d(K)]} and a random noise ξ, uniformly distributed between 0 y 1.

$$F_s^{-1}\ :\{\xi, s[n-d(1)], s[n-d(2)], \ldots s[n-d(K)]\} \xrightarrow{F_s^{-1}} s[n]$$

An analytical expression for F_s^{-1} is obviously not possible and consequently it has to be achieved form the transition matrix of the histogram. This method implies the need of searching in a multidimensional matrix. Therefore it is not appropriate to work in real time.

To avoid this solution we propose the neural scheme depicted in figure 1. As the learning of inverse functions is one of the inherent abilities of multilayer perceptrons, we utilize a neural network to approximate F_s^{-1}. For this purpose a set of learning patterns is selected from the conditioned histogram that has been

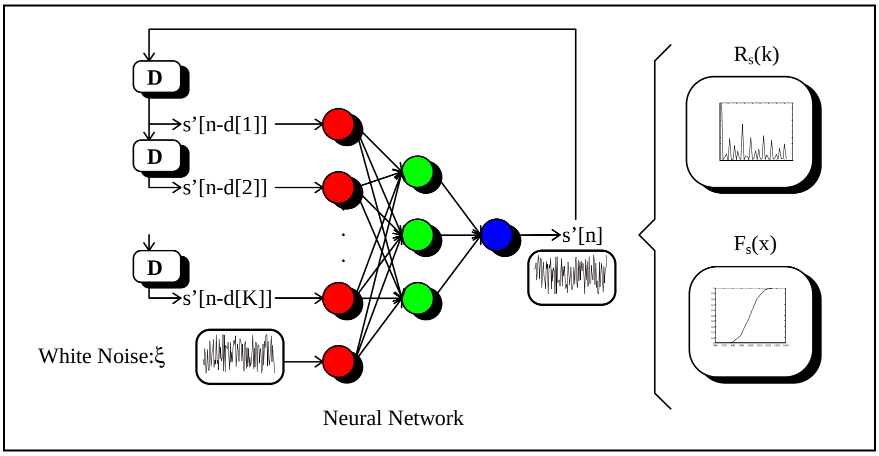


Figure 1: Neural Generator. The input of the neural net is a white noise and its previous outputs. This way, the probability distribution function F_s and autocorrelation function R_s are fitted

previously estimated from the actual signal s[n]. The inputs for these patterns are the delayed samples {s[n],s[n-d(1)],... s[n-d(i)],... s[n-d(K)]} and the noise ξ, and the output the present sample s[n].

The final result, after the training phase has concluded, is that the network will be able to yield a simulated signal s'[n] just as a function of a noise and some past states of the same traffic s'[n]. This signal s'[n] will adjust the probability distribution $F_s(x)$ of s[n] with an accuracy that is determined by the number N of levels in which the signal has been divided, as well as it will approximate the autocorrelation function $R_s[n]$ in those points corresponding to the considered lags. This matching is specially interesting for video signals since video traffic always exhibits a strong correlation.

4 Simulation and Numerical Results

We utilize our model to imitate MPEG codified video. At present, the MPEG family of coding algorithms is widely used in broadband video communications. The compression that MPEG performs reduces both the spatial and temporal redundancy of the video stream. This is achieved by using three types of frame:

I frames, which only eliminate spatial redundancy, and P and B frames, which also reduce temporal redundancy by motion compensation.

MPEG algorithm normally arranges these three types of frames in a fixed periodic sequence, e.g. IBBPBBIBBPBBIBBPBB..., whose period (IBBPBB) is called a group of picture GOP (Group of Blocks).

In this paper a MPEG-I video source is contemplated to evaluate the performance of the proposed scheme. In particular the source s[n], indicating the number of ATM cells per GOP, is the film "*Star Wars*" which lasts for about 97 minutes and contains 14511 gops of 12 frames. The histogram of s[n], conditioned to the previous sample s[n-1], is estimated taking into consideration 30 levels. The neural network was designed with 2, 10 and 1 neurons in its input, hidden and output layers, respectively. After selecting the learning patterns, the training phase of the neural network is performed. A backpropagation algorithm was used and the mean quadratic error, normalizing between 0 and 1, was lower than $2x10^{-5}$. Once the learning is accomplished, the network generated a sequence of 20000 samples of the simulated traffic s'[n] using as an input a white noise, uniformly distributed between 0 and 1. The statistics of this traffic s'[n] are compared with those of the actual traffic s[n] in figures 2, 3 and 4.

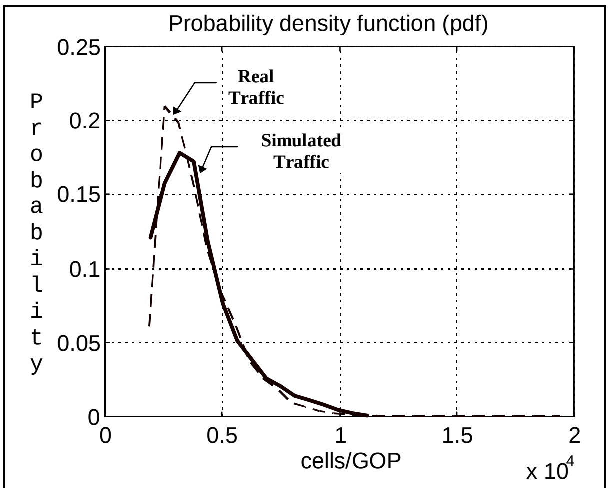


Figure 2. Comparison of pdf
of the real traffic and the model

From these figures we can see that the simulated traffic s'[n] correctly fits the curve of the probability density and distribution of s[n] while it approximates the autocorrelation for the first lag, as the histogram was estimated considering the previous sample s[n-1]. Figure 4 also shows that real traffic exhibits a strong autocorrelation (R_s>0.25) for high values of the lags while the autocorrelation $R_{s'}$ of the simulated signal fastly decays as the lag increases. This long term dependence of R_s is due to the existence of scenes within the film with a similar activity level. This *fractal* characteristic of the traffic must not be neglected as it can determine the behaviour in a queue [5]. So, in further studies, a more complete model of the traffic should contemplate the characterization of these scene-changes.

4 Conclusions

A neural model for the generation of ATM video traffic has been proposed. The model uses a neural network to learn the conditioned histogram of a given traffic in such a way that it is able to generate a traffic which fits the probability distribution and some points of the autocorrelation function of the real signal. Using neural networks we avoid the necessity of searching in transition matrices of other histogram based models. Moreover, neural networks are adequate tools for working in real time so they can be used to develop ATM traffic emulators.

On the other side, although the study has been focused on video traffic, because of its correlation characteristics, the proposed model can simulate any random series. Thus it could be extended to other types of ATM traffic sources.

Acknowledgement

This work has been partially supported by the Spanish Comisión Interministerial de Ciencia y Tecnología (CICYT), Project No. TIC96-0743-PB.

We also wish to express our gratitude to M. Garret (Bellcore) for releasing the MPEG-I traces of the film *Star Wars*.

References

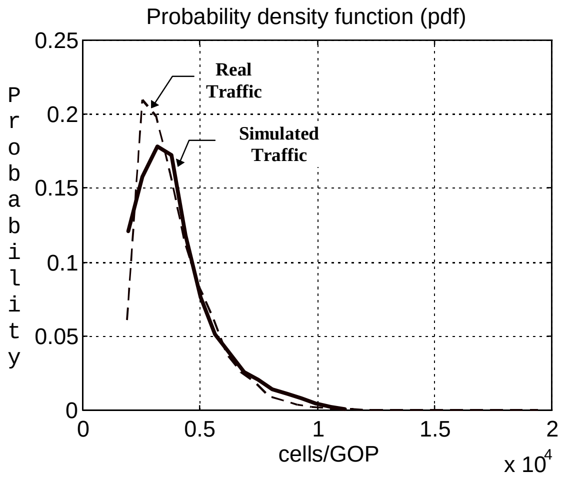
[1] B. Maglaris, D. Anastassious, P. Sen, G. Karlsson, and J. D. Roberts, “Performance Models of Statistical Multiplexing In Packet Video Communications”, *IEEE Trans. on Communications*, Vol. 36, No. 7, pp. 834-843, July 1988.

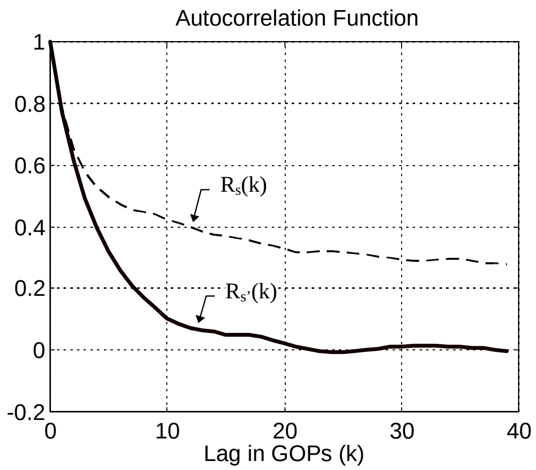
[2] N. Ohta, *Packet Video*, Artech House, 1994.

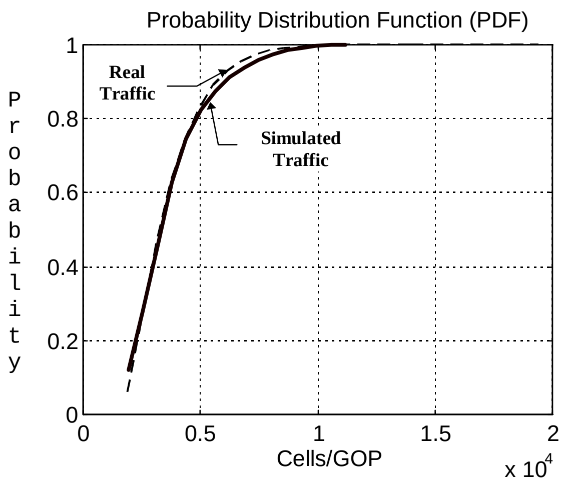
[3] J.L. Wu, Y.W. Chen, and K.C. Jiang, “Two Models for Variable Bit Rate MPEG Sources”, *IEICE Trans. on Communications*, Vol. E78-B, No. 5, pp. 773-745, May 1995.

[4] L.A. Kulkarni, and S.Q. Li, “Traffic modeling: matching the power spectrum and distribution”, in *Proceedings. of GLOBECOM’95*, pp. 1701-1706, November 1995.

[5] M- Conti, E. Gregori and A. Larson, “Study of the Impact of MPEG-1 Correlations on Video-Sources Statistical Multiplexing”, *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 7, pp. 1455-1471, September 1996.


Probability density function (pdf)
Probability
0.25
0.2
0.15
0.1
0.05
0
0
0.5
1
1.5
2
cells/GOP
x 10^4
Real Traffic
Simulated Traffic


Autocorrelation Function
1
0.8
0.6
0.4
0.2
0
-0.2
0
10
20
30
40
Lag in GOPs (k)
Rs(k)
Rs'(k)


Probability Distribution Function (PDF)
Probability
1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
Cells/GOP
x 10^4
Real Traffic
Simulated Traffic