TEHNICI PREDICTIVE ÎN PROCESAREA SEMNALULUI VORBIT

MARINA CIDOTA
Universitatea Bucureşti, Facultatea de Matematică şi Informatică,
email cmarina@k.ro

Rezumat: Semnalul vorbit este nestaționar şi din acest motiv nu poate fi modelat cu ajutorul modelelor lineare parametrice clasice (*AR, MA, ARMA*). Aplicarea rețelelor neurale recurente în domeniul predicției seriilor dinamice este potrivită pentru invățarea şi recunoaşterea structurii neliniare a semnalului vorbit. Prezentăm trei tipuri de algoritmi de antrenare a rețelelor recurente pentru predicție şi rezultatele testelor efectuate pe două clase de semnale, rostite atât de femei cât şi de bărbați în diferite dialecte ale limbii engleze.

Cuvinte cheie: tehnici predictive în procesarea semnalului vorbit, rețele reecurente neurale

1. INTRODUCERE

Tradițional, predicția seriilor dinamice s-a realizat cu ajutorul modelelor ***A**uto**R**egressive*, ***M**oving **A**verage* sau ***ARMA*** ai căror parametri sunt estimați fie în bloc fie secvențial cu algoritmul ***L**east **M**ean **S**quare*. Problema este că aceste modele sunt liniare şi nu pot procesa semnale neliniare sau nestaționare. Rețelele neurale sunt un instrument puternic de rezolvare a problemelor ale căror soluții necesită cunoştințe greu de specificat, dar pentru care există foarte multe exemple. Deoarece prin predicția seriilor dinamice înțelegem deducerea comportamentului viitor pe baza exemplelor comportamentului trecut, înseamnă că aceasta este o aplicație potrivită pentru o rețea neurală. Putem privi sistemele adaptive de predicție ca fiind nucleul procesării semnalelor digitale pentru că de fapt orice filtru adaptiv încearcă să prezică răspunsul dorit. Un astfel de sistem constă din:

- o mulțime de ponderi ajustabile
- un bloc de calcul al erorii (diferența dintre răspunsul dorit şi răspunsul sistemului)
- un algoritm de control (învățare) pentru ajustarea ponderilor

şi poate fi reprezentat în figura de mai jos:

Analele Universității SPIRU HARET, Seria Matematică-Informatică, nr. 1 , 2005, ISSN 1841 - 7833

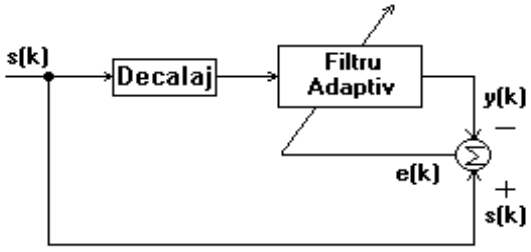

Fig.1 Configurația unui sistem de predicție

Sunt două clase de filtre adaptive:

- cele fără *feedback*, pentru care ieşirea depinde de intrarea curentă şi de intrările de la *p* momente din trecut (***F**inite **I**mpulse **R**esponse* sau structuri ***MA***)
- cele cu *feedback* pentru care ieşirea depinde atât de intrările cât şi de ieşirile trecute (***I**nfinite **I**mpulse **R**esponse* sau structuri ***AR*** sau ***ARMA***)

Forma generală a unui filtru *IIR* neliniar este dată de ecuația

$$y(k)=\Theta\left(\sum_{i=1}^{p}a_i y(k-i)+\sum_{j=1}^{q}b_j e(k-j)\right)+e(k), \tag{1}$$

unde *y(k)* reprezintă ieşirea, *e(k)* intrarea iar $\Theta(\cdot)$ este o funcție diferențiabilă neliniară. Vom numi acest model ***NARMA**(p,q) (Nonlinear **ARMA**) iar predictorul corespunzător este dat de*

$$\hat{y}(k)=\Theta\left(\sum_{i=1}^{p}a_i y(k-i)+\sum_{j=1}^{q}b_j \hat{e}(k-j)\right), \tag{2}$$

rezidurile fiind $\hat{e}(k-j)=y(k-j)-\hat{y}(k-j), j=1,2,\ldots,q.$

Vom prezenta în continuare câteva implementări neurale ale unor filtre *IIR*.

2. ARHITECTURI NEURALE RECURENTE PENTRU PREDICȚIE

O rețea cu o bogată reprezentare a ieşirilor trecute este o rețea recurentă complet conectată, cunoscută sub numele de rețea *Williams-Zipser* (vezi Fig. 2). Constă din trei nivele: de intrare, de procesare şi de ieşire. Pentru fiecare neuron *i*, $i=1,2,\ldots, N$, elementele u_j, $j=1,2, \ldots, p+1+N$ ale vectorului de intrare $\boldsymbol{u}$ sunt ponderate şi însumate pentru a produce activarea internă a fiecărui neuron, care apoi este trecută prin funcția de activare neliniaăa Φ pentru a obține ieşirea neuronului *i*, y_i. Vom considera în continuare funcția neliniară Φ ca fiind funcția logistică

$$\Phi(v) = \frac{1}{1+e^{-\beta v}}\ , \beta > 0, \tag{3}$$

care este crescătoare şi ia valori în intervalul (0,1).
Ecuațiile următoare descriu rețeaua din figura 2

$$\begin{aligned} &y_i(k) = \Phi(v_i(k)),\ \ i = 1,2,\ldots,N, \\ &v_i(k) = \sum_{l=1}^{p+1+N} w_{i,l}(k) u_l(k), \\ &u_i^T(k) = [s(k-1),\ldots,s(k-p),1, y_1(k-1), y_2(k-1),\ldots, y_N(k-1)]. \end{aligned} \tag{4}$$

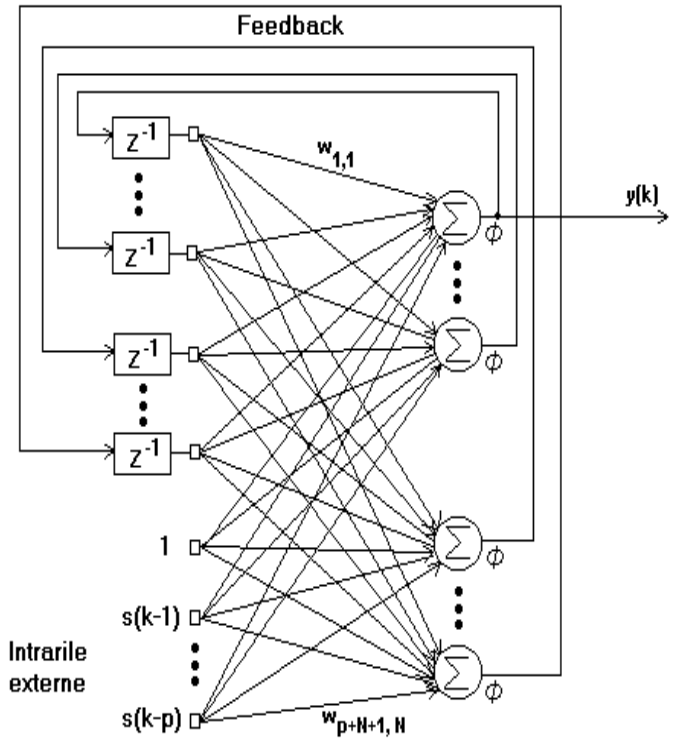

Fig.2 Rețea recurentă Williams-Zipser

Un proces general NARMA(p,q) poate fi exprimat prin

$$\hat{s}(k)=\Phi\left(\sum_{i=1}^{p} w_{1,i}(k)s(k-i)+w_{1,p+1}(k)+\sum_{j=p+2}^{p+q+1} w_{1,j}(k)\hat{e}(k+j-2-p-q)\right.$$
$$\left.+\sum_{l=p+2+q}^{p+q+N} w_{1,l}(k)y_{l-p-q}(k-1)\right), \tag{5}$$

şi poate fi implementat printr-o reţea recurentă *Williams-Zipser* pentru care intrarea este

$$s(k-1),\ldots,s(k-p),1,\hat{e}(k-1),\ldots,\hat{e}(k-q),y_2(k-1),\ldots,y_N(k-1). \tag{6}$$

Intrările reţelei conţin termenii erorii de predictie $\hat{e}(k-1),\ldots,\hat{e}(k-q)$, ceea ce face ca învăţarea să fie un proces dificil. Algoritmul de învăţare recurentă în timp real (***R**eal **T**ime **R**ecurrent **L**earning*) a fost conceput pentru a minimiza eroarea pătratică instantanee de predictie *e(k)* şi din acest motiv nu poate fi aplicat direct acestei reţele deoarece intrarea conţine termeni ai erorii de predicţie. Din acest motiv trebuie găsită o altă reprezentare echivalentă a reţelei *NARMA(p,q)* astfel încât să putem aplica algoritmul de învăţare *RTRL*. Dacă înlocuim $\hat{e}(k)$ cu $s(k)-y_1(k)$ şi regrupăm termenii obţinem

$$\hat{s}(k)=\Phi\left(\sum_{i=1}^{p} w_{1,i}(k)s(k-i)+w_{1,p+1}(k)+\sum_{j=p+2}^{p+q+1} w_{1,j}(k)y_1(k+j-2-p-q)\right.$$
$$\left.+\sum_{l=p+2+q}^{p+q+N} w_{1,l}(k)y_{l-p-q}(k-1)\right), \tag{7}$$

şi sub această formă poate fi implementat cu ajutorul reţelei *Williams-Zipser*.
Cea mai simplă arhitectură neurală *NARMA* este perceptronul recurent prezentat în figura 3. Ieşirea perceptronului este dată de

$$y(k)=\Phi(net(k)),$$
$$net(k)=\sum_{j=1}^{M} w_j(k)s(k-j)+w_{M+1}(k)+\sum_{m=1}^{N} w_{m+M+1}(k)y(k-m). \tag{8}$$

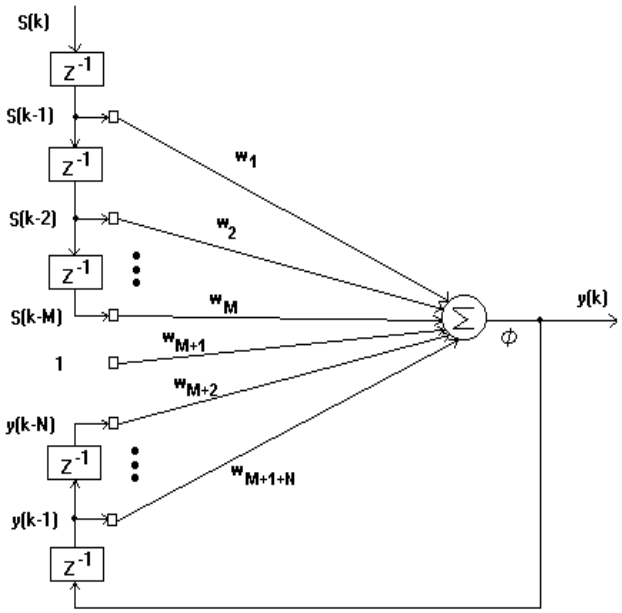

Fig.3 Percepton recurent NARMA

3. **ALGORITMI DE ÎNVĂȚARE PENTRU REȚELE NEURALE RECURENTE (*RNN*)**

3.1 **Algoritmul real time recurrent learning (RTRL)**

Vom prezenta algoritmul *RTRL* de instruire a rețelei din figura 2, algoritmul pentru instruirea rețelei din figura 3 fiind un caz simplificat al acestuia. Învățarea *RTRL* are la bază ideea minimizării erorii pătratice instantanee, adică

$$\min\left(E(k)\right) = \min\left(\frac{1}{2}e^2(k)\right), \tag{9}$$

unde $e(k) = s(k) - y_1(k)$. Pentru a actualiza matricea ponderilor *w*, avem nevoie de

$$\Delta w_{n,l}(k) = -\frac{\eta}{2}\frac{\partial}{\partial w_{n,l}(k)}e^2(k) = -\eta e(k)\frac{\partial e(k)}{\partial w_{n,l}(k)}. \tag{10}$$

Deoarece intrarea *s(k)* nu depinde de elementele matricei *w*, gradientul erorii devine

$$\frac{\partial e(k)}{\partial w_{n,l}(k)} = -\frac{\partial y_1(k)}{\partial w_{n,l}(k)}. \tag{11}$$

Dar $y_1(k) = \Phi(v_1(k))$, rezultă că

$$\begin{aligned} \frac{\partial y_1(k)}{\partial w_{n,l}(k)} &= \Phi'(v_1(k)) \frac{\partial v_1(k)}{\partial w_{n,l}(k)} \\ &= \Phi'(v_1(k)) \left(\sum_{\alpha=1}^{N} \frac{\partial y_\alpha(k-1)}{\partial w_{n,l}(k)} w_{1,\alpha+p+1}(k) + \delta_{n,1} u_l(k) \right). \end{aligned} \tag{12}$$

Dacă rata de învățare η este suficient de mică, atunci putem presupune că

$$\frac{\partial y_\alpha(k-1)}{\partial w_{n,l}(k)} \approx \frac{\partial y_\alpha(k-1)}{\partial w_{n,l}(k-1)}. \tag{13}$$

Făcând notațiile

$$\pi_{n,l}^{j} = \frac{\partial y_j(k)}{\partial w_{n,l}(k)} \quad 1 \le j, n \le N, 1 \le l \le p+1+N, \tag{14}$$

obținem următoare formulă recursivă

$$\pi_{n,l}^{j}(k+1) = \Phi'(v_j(k)) \left(\sum_{m=1}^{N} w_{j,m+p+1}(k) \pi_{n,l}^{m}(k) + \delta_{n,j} u_l(k) \right) \tag{15}$$

cu condițiile inițiale $\pi_{n,l}^{j}(0) = 0, \quad 1 \le j, n \le N, 1 \le l \le p+1+N.$

Deci ponderile se reactualizează după formula

$$w_{n,l}(k+1) = w_{n,l}(k) + \Delta w_{n,l}(k) \quad 1 \le n \le N, 1 \le l \le p+1+N. \tag{16}$$

3.2 **Algoritmul extended recursive least squares (ERLS)**

Algoritmul *ERLS* are la bază ideea filtrului *Kalman* extins. Filtrul *Kalman* constă dintr-un set de ecuații matematice care modelează un mecanism recursiv pentru estimarea stării unui proces prin minimizarea erorii medii pătratice. Filtrul este foarte puternic în câteva privințe: poate estima stările trecută, prezentă sau

viitoare chiar când natura exactă a sistemului modelat este necunoscută. Filtrul *Kalman* tratează problema generală a estimării stării unui proces discret guvernat de o ecuație diferențială liniară. Dar semnalul vorbirii nu este un proces liniar şi pentru estimarea lui vom considera filtrul *Kalman* extins. Vom considera următoarele ecuații ale filtrului:

$$\begin{aligned} w(k) &= w(k-1) + q(k), \\ s(k) &= h(w(k)) + r(k), \end{aligned} \tag{17}$$

unde *w(k)* este vectorul ponderilor de dimensiune *N*(p+1+N)*, *s(k)* este observația (semnalul) la momentul *k*, *q(k)* este zgomot alb *Gaussian*, *q~N(0,Q)*, *r(k)* reprezintă zgomotul (alb *Gaussian*) observației, *r~N(0,C)* iar *h* este o funcție neliniară diferențiabilă.

Vom liniariza funcția neliniară $h(\cdot)$ folosind dezvoltarea *Taylor* în jurul estimației lui *w(k)* bazată pe datele anterioare, notată cu $\hat{w}(k \mid k-1)$:

$$h(w(k)) \approx h(\hat{w}(k \mid k-1)) + \nabla h^T (w(k) - \hat{w}(k \mid k-1)), \tag{18}$$

unde prin ∇h^T înțelegem

$$\nabla h^T = \frac{\partial h(\hat{w}(k \mid k-1))}{\partial \hat{w}(k \mid k-1)} = H(k). \tag{19}$$

Ecuația observației va avea următoarea formă:

$$s(k) = H(k)w(k) + r(k) + [h(\hat{w}(k \mid k-1)) - H(k)\hat{w}(k \mid k-1)]. \tag{20}$$

Cu notațiile de mai sus, ecuațiile finale de reactualizare a vectorului ponderilor *w* pentru algoritmul *ERLS* devin:

$$\begin{aligned} K(k) &= Q(k-1)H^T(k)[C + H(k)Q(k-1)H^T(k)]^{-1}, \\ \hat{w}(k) &= \hat{w}(k-1) + K(k)[x(k) - h(\hat{w}(k-1))], \\ M(k) &= (I - K(k)H(k))Q(k-1). \end{aligned} \tag{21}$$

In cazul rețelei din figura 2 avem $h(w(k))=y_1(k)$.

Pentru calcularea gradientului vom presupune că $\hat{w}(k \mid k-1) \approx \hat{w}(k-1 \mid k-1) = \hat{w}(k-1)$ şi ăn aceste condiții $H(k) = \frac{\partial h(\hat{w}(k-1))}{\partial \hat{w}(k-1)}$. Elementele matricei gradient se calculează recurent astfel:

$$\frac{\partial y_j(k)}{\partial w_{n,l}(k)} \approx \Phi'\left(v_j(k)\right)\left[\sum_{\alpha=1}^{N} \frac{\partial y_\alpha(k-1)}{\partial w_{n,l}(k-1)} w_{j,\alpha+p+1}(k) + \delta_{n,j} u_l(k)\right]. \qquad (22)$$

Algoritmul *ERLS* prezentat diferă de algoritmul *RTRL* doar prin formula de reactualizare a ponderilor, structura rețelei neurale fiind aceeaşi.

3.3 **Algoritmi de învățare a posteriori**

Pentru a obține un predictor neliniar *a posteriori*, să considerăm pentru simplitate perceptronul recurent *NARMA* a cărui ieşire este:

$$y(k) = \Phi\left(u^T(k) w(k)\right), \qquad (23)$$

unde $u^T(k) = \left[s(k-1),\ldots,s(k-M),1,y(k-1),y(k-2),\ldots,y(k-N)\right]$. Deoarece vectorul ponderilor actualizat *w(k+1)* este disponibil înainte de introducerea vectorului de intrare *u(k+1)*, putem calcula ieşirea *a posteriori*:

$$\bar{y}(k) = \Phi\left(u^T(k) w(k+1)\right). \qquad (24)$$

Erorile corespunzătoare la ieşire sunt:

$$\begin{aligned} e(k) &= s(k) - y(k) \quad \text{eroarea a priori}, \\ \bar{e}(k) &= s(k) - \bar{y}(k) \quad \text{eroarea a posteriori}, \end{aligned} \qquad (25)$$

unde *s(k)* este semnalul dorit. Ieşirea *a posteriori* poate fi folosită pentru a obține vectorul de intrare *a posteriori*

$$\bar{u}^T(k) = \left[s(k-1),\ldots,s(k-M),1,\bar{y}(k-1),\bar{y}(k-2),\ldots,\bar{y}(k-N)\right] \qquad (26)$$

care poate înlocui vectorul de intrare *a priori* pentru calcularea ieşirii şi actualizarea ponderilor.
Vom prezenta o variantă a metodei coborârii pe gradient astfel încât eroarea *a posteriori* să fie mai mică decât eroarea *a priori*:

$$
\begin{aligned}
&w(k+1) = w(k) - \eta \nabla_w E(k), \\
&e(k) = s(k) - \Phi\left(u^T(k) w(k)\right), \\
&\bar{e}(k) = s(k) - \Phi\left(u^T(k) w(k+1)\right), \\
&\text{cu conditia ca } |\bar{e}(k)| \le \gamma |e(k)|, \quad 0 < \gamma < 1.
\end{aligned} \tag{27}
$$

iar $E(k) = \frac{1}{2} e^2(k).$

În cazul *NARMA*, ecuația de reactualizare a ponderilor este:

$$w(k+1) = w(k) + \eta(k) e(k) \pi(k), \tag{28}$$

unde $\pi^T(k) = \left[\pi_1(k), \pi_2(k), \ldots, \pi_{M+N+1}(k)\right]$ *iar* $\pi_i(k) = \frac{\partial y(k)}{\partial w_i(k)}.$

Putem scrie eroarea *a posteriori* sub forma:

$$\bar{e}(k) = s(k) - \Phi\left(u^T(k) w(k)\right) - \left[\Phi\left(u^T(k) w(k+1)\right) - \Phi\left(u^T(k) w(k)\right)\right]. \tag{29}$$

Dacă pentru funcția logisticî $\Phi(x) = \frac{1}{1+e^{-\beta x}}$ luăm $0 < \beta < 4$ atunci $\Phi'(x) \le 1 \ \forall x \in \Re$ şi, revenind la expresia erorii *a posteriori*, avem conform teoremei *Lagrange*:

$$\Phi\left(u^T(k) w(k+1)\right) - \Phi\left(u^T(k) w(k)\right) = \alpha(k) u^T(k) \Delta w(k), \tag{30}$$

unde $\alpha(k) = \Phi'(\xi) < 1$, $\xi \in \left(u^T(k) w(k), u^T(k) w(k+1)\right).$
Rezultă că:

$$
\begin{aligned}
\bar{e}(k) &= e(k) - \alpha(k) u^T(k) \Delta w(k) \\
&= e(k) - \alpha(k) \eta(k) u^T(k) \pi(k) e(k) \\
&= \left(1 - \alpha(k) \eta(k) u^T(k) \pi(k)\right) e(k).
\end{aligned} \tag{31}
$$

iar rata de învățare $\eta(k) = \frac{1}{\alpha(k) u^T(k) \pi(k)}$ minimizează expresia de mai sus. Problema este că nu cunoaştem $\alpha(k)$ şi din acest motiv vom încerca să găsim o limită inferioară a erorii $\bar{e}(k)$. Plecând de la egalitatea

$$\Phi\left(u^T(k)w(k+1)\right) = \Phi\left(u^T(k)w(k) + \eta(k)e(k)u^T(k)\pi(k)\right) \tag{32}$$

şi folosind proprietatea funcției logistice $\Phi(a+b) \le \Phi(a) + \Phi(b)\ \forall a, b \in \Re$ deducem că

$$\bar{e}(k) \ge e(k) - \Phi\left(\eta(k)e(k)u^T(k)\pi(k)\right). \tag{33}$$

Dacă $0 < \beta < 4$ avem $|\Phi(x)| < |x|, \forall x \in \Re$ şi obținem în final

$$\bar{e}(k) \ge \left(1 - \eta(k)\left|u^T(k)\pi(k)\right|\right)e(k), \tag{34}$$

iar domeniul permis pentru rata de învățare este:

$$0 < \eta(k) < \frac{1}{\left|u^T(k)\pi(k)\right|}. \tag{35}$$

4. **APLICAȚII**

Pentru testarea algoritmilor prezentați am folosit baza de date *TIMIT* ce conține semnale rostite de 630 de vorbitori în 8 dialecte ale limbii engleze, fiecare înregistrând câte 10 propoziții. Am considerat două clase de semnale, fiecare reprezentând câte o propoziție rostită atât de femei cât şi de bărbați. Prima propoziție *"Those who teach values first abolish cheating"* este rostită de trei femei şi patru bărbați iar a doua propoziție *"The eastern coast is a place for pure pleasure and excitement"* este rostită de o femeie şi şase bărbați. Nu toate aceste persoane vorbesc acelaşi dialect. Fişierele le-am obținut în formatul *WAV* şi le-am prelucrat cu ajutorul soft-ului *HTK* (**H**idden Markov **T**ool**K**it) specializat în procesarea semnalului vorbit. Fişierele al căror nume începe cu "*f*" reprezintî semnale rostite de femei, celelalte (care încep cu "*m*") sunt rostite de bărbați. Ca fişier de antrenare am folosit "*mabc0.txt*" pentru toți algoritmii (vezi fig. 4). Deoarece funcția logistică ia valori în intervalul (0,1), am ajustat amplitudinea semnalelor astfel încât să fie în acelaşi interval (0,1).

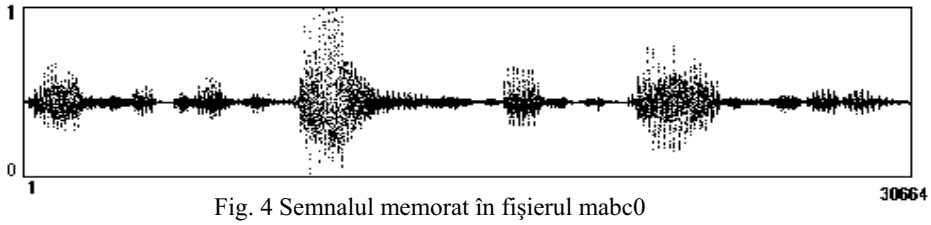

Fig. 4 Semnalul memorat în fişierul mabc0

Pentru a măsura calitatea predictorilor am folosit "*câştigul predicţiei*" definit astfel:

$$R_p = 10\log_{10}\left(\frac{\hat{\sigma}_s^2}{\hat{\sigma}_e^2}\right) d\mathbf{B}, \tag{36}$$

unde $\left\{\hat{\sigma}_s^2\right\}$reprezintă varianta estimată a semnalului $\{s(k)\}$ în timp ce $\left\{\hat{\sigma}_e^2\right\}$este varianta estimată a erorii de predicţie $\{e(k)\}$.

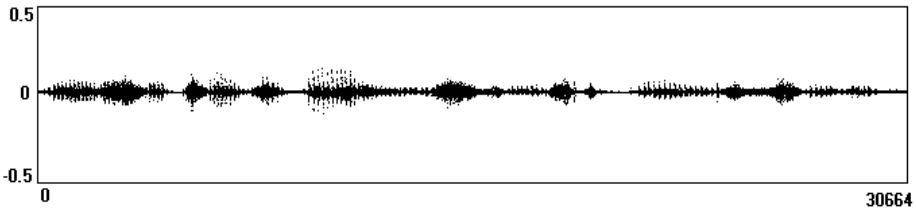

Fig. 5 Eroarea rezultată din testarea fişierului mabc0

In tabelele de mai jos am afişat indicele R_p rezultat din testarea algoritmilor *RTRL*, *NARMA*, *ERLS* şi *NARMA a posteriori* antrenaţi cu parametrii specificaţi în tabele.

	RTRL p=12 N=3 β=2	NARMA M=12 N=3 β=2	ERLS p=10 N=2 β=2	NARMA a posteriori p=12 N=2 β=2
fjlr0	17.46	17.15	18.25	17.04
fkde0	20.59	20.33	21.96	20.26
fleh0	21.62	21.25	22.28	21.23
mabc0	**29.85**	**29.88**	**28.63**	**29.88**
mgag0	25.96	26.07	25.61	26.09
mpmb0	24.80	24.93	25.26	25.04
mrem0	28.80	28.70	27.19	28.63

Tabelul 1 Indicele R_p obţinut prin testarea semnalelor din prima clasă (*"Those who teach values first abolish cheating"*)

	RTRL p=12 N=3 β=2	NARMA M=12 N=3	ERLS p=10 N=2 β=2	NARMA a posteriori

		β=2		p=12 N=2 β=2
fsmm0	15.89	15.69	18.00	15.66
mbma1	20.48	20.60	20.81	20.60
mfxs0	25.10	25.24	24.90	25.25
mmab1	27.66	27.49	27.82	27.50
mmrp0	24.50	24.73	24.92	24.82
mrab0	16.97	16.87	18.82	16.84
mtlb0	18.74	18.65	18.92	18.55

Tabelul 2 Indicele R_p obținut prin testarea semnalelor din a doua clasă (*"The eastern coast is a place for pure pleasure and excitement"*)

BIBLIOGRAFIE

1. Mandic, D.P., Chambers, J.A., *Recurrent Neural Networks for Prediction –Learning Algorithms, Architectures and Stability* , John Wiley & Sons, 2001
2. Welch, G., Bishop, G., *An Introduction to the Kalman Filter*, (internet)
3. Catlin, D., *Estimation, Control and the Discrete Kalman Filter*, Springer Verlag, 1989
4. Young, S.,Evermann, G., *The HTK Book* (internet)

Abstract: Speech signal has statistically non-stationary properties and cannot be processed properly by the use of classical linear parametric models (*AR, MA, ARMA*). The neural network approach to time series prediction is suitable for learning and recognizing the nonlinear nature of the speech signal. We present three types of learning algorithms for recurrent neural networks for prediction and test them on two classes of speech signal spoken by both men and women in different dialects of English language.