A comparison between recurrent neural architectures for real-time nonlinear prediction of speech signals

Juan Antonio Pérez-Ortiz

Jorge Calera-Rubio

Mikel L. Forcada

Departament de Llenguatges i Sistemes Informàtics

Universitat d'Alacant, E-03071 Alacant, Spain

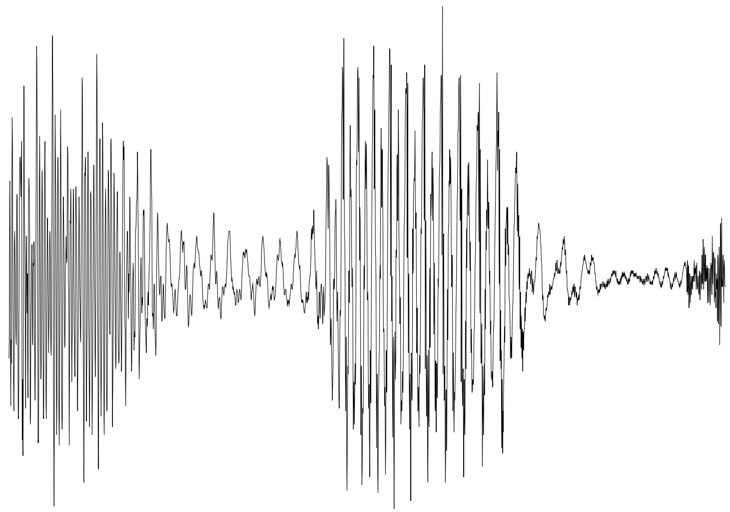
Summer 2001

Contents

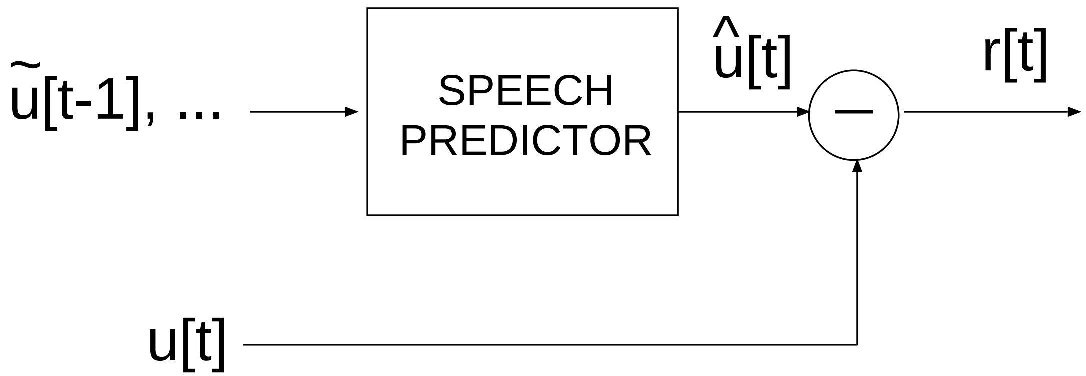
- Predicting speech.
- Some approaches:
  - * Simple linear approaches.
  - * Specialized recurrent neural network.
- Our study: classical recurrent architectures.
- Experiments.
- Conclusions.

Speech prediction

- No preprocessing: we work on sampled waveform $u[t]$.
- Importance of real-time speech prediction.
- Assumption: $u[t]$ may be estimated from $u[t-1], u[t-2], \ldots$.



Predictive coding



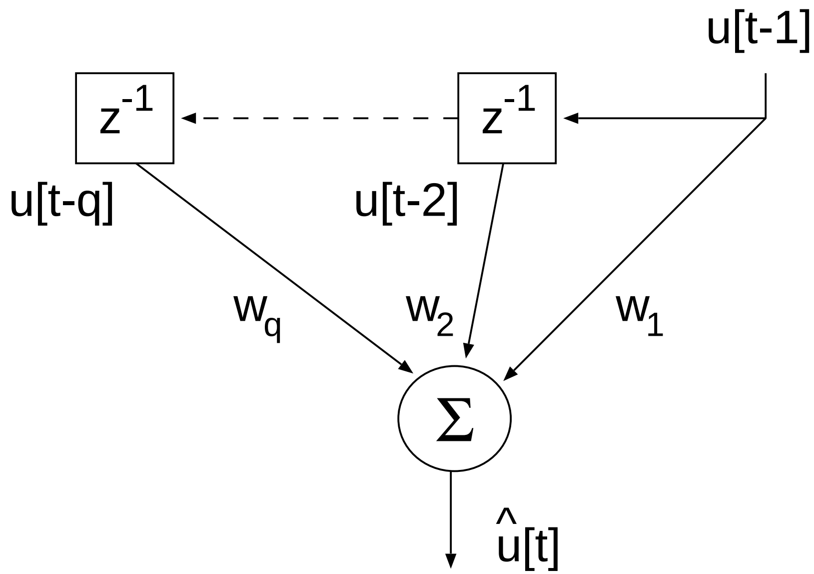
- $r[t]$ is sent instead of $u[t]$.
- Bit rate is lowered if predictor is accurate.
- Predicting is compressing.

Speech predictors

- Linear predictors.
- Nonlinear predictors:
  * In this work, we focus on recurrent neural networks.
  * They are supposed to take into account speech nonlinearities.

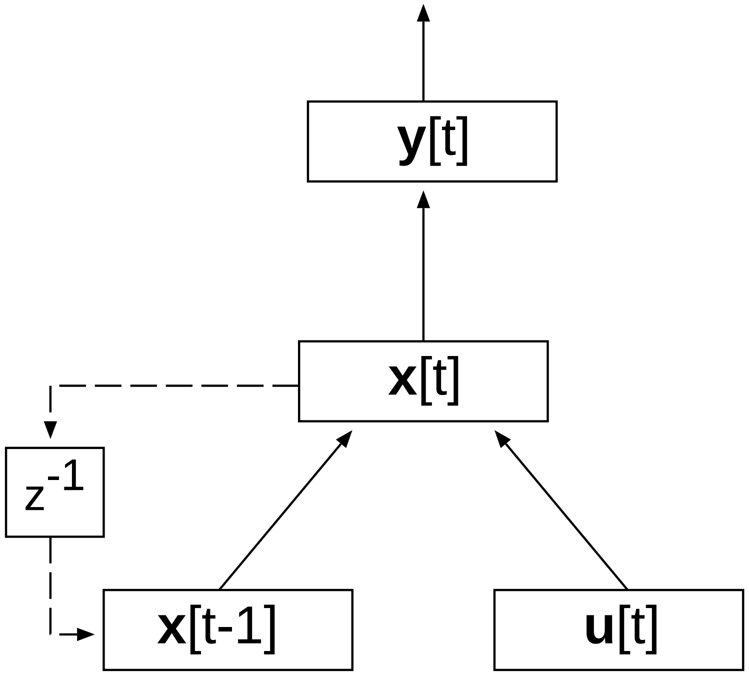
Linear predictors

- Simple predictor taken as baseline: $\hat{u}[t] = u[t-1]$.
- Finite impulse response filter (FIR):



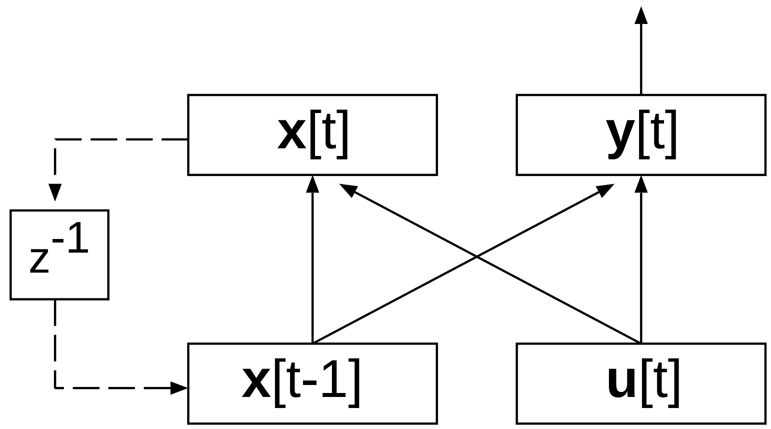
Elman's simple recurrent network

- Elman (1990).



Recurrent error propagation network (REPN)

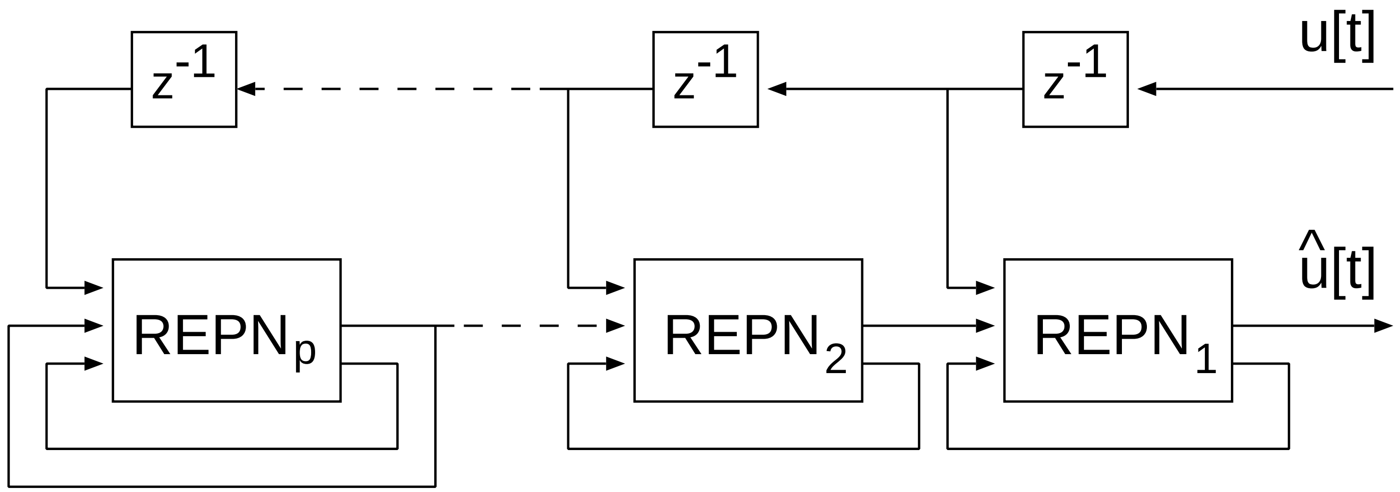
- Robinson and Fallside (1991).



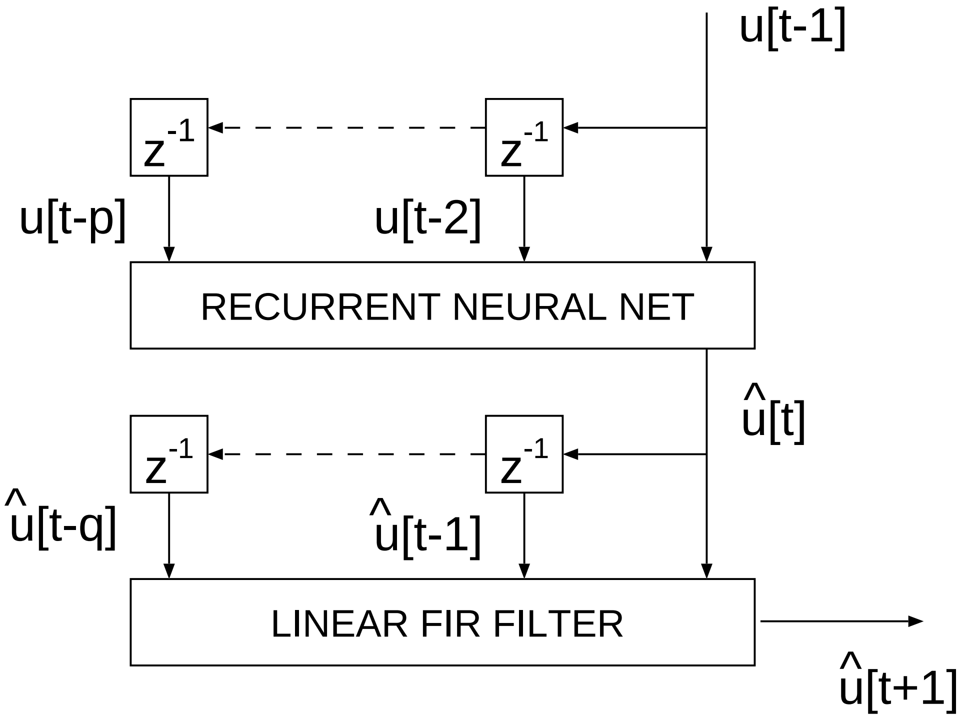
Pipelined recurrent neural network (PRNN)

- Haykin and Li (1995).

- Weights are shared by all the recurrent error propagation networks (REPN) in the PRNN.



Cascade configuration



- Nonlinear block is supposed to linearize the input signal.
- We use general-purpose recurrent networks for the upper block.

Training algorithms

- Linear FIR predictor:
  - * Least-mean-square (LMS).
  - * Recursive least-squares (RLS).
- Recurrent neural networks:
  - * Real-time recurrent learning (RTRL).
  - * Decoupled extended Kalman filter (DEKF).

Experiments

- Results are the average for 7 different weight initializations (except for PRNN).
- Results are similar for different signals.
- The linear FIR filter order is $q = 12$.
- Both neural architectures PRNN and REPN have been chosen to have a similar number of parameters (around 35 adjustable weights).

Prediction quality

- Performance is measured by means of prediction gain G:

$$G = 10 \log_{10} \left(\frac{S_u^2}{S_e^2} \right)$$

  * S_u^2 is the estimated variance of the speech signal $u[t]$.
  * S_e^2 is the estimated variance of the error signal $u[t] - \widehat{u}[t]$

Worst results

Architecture (training)	**G** (dB)
REPN + FIR	≤ 3.99
$\star\ \ \widehat{u}[t+1] = u[t]$	4.61
REPN (RTRL)	5.80
FIR (LMS)	5.82

Best results

Architecture (training)	**G** (dB)
PRNN (RTRL) + FIR (LMS)	7.30
REPN (DEKF)	8.61
PRNN (RTRL) + FIR (RLS)	9.24
$\star$ FIR (RLS)	9.66
PRNN (DEKF) + FIR (RLS)	10.90

Conclusions

- Problems to deal with speech series for classical recurrent networks.
- The decoupled extended Kalman filter partially overcomes some of these limitations.
- Only PRNN trained with the Kalman filter followed by a FIR filter trained by recursive least-squares (RLS) attains better results than a simple FIR filter trained by RLS.