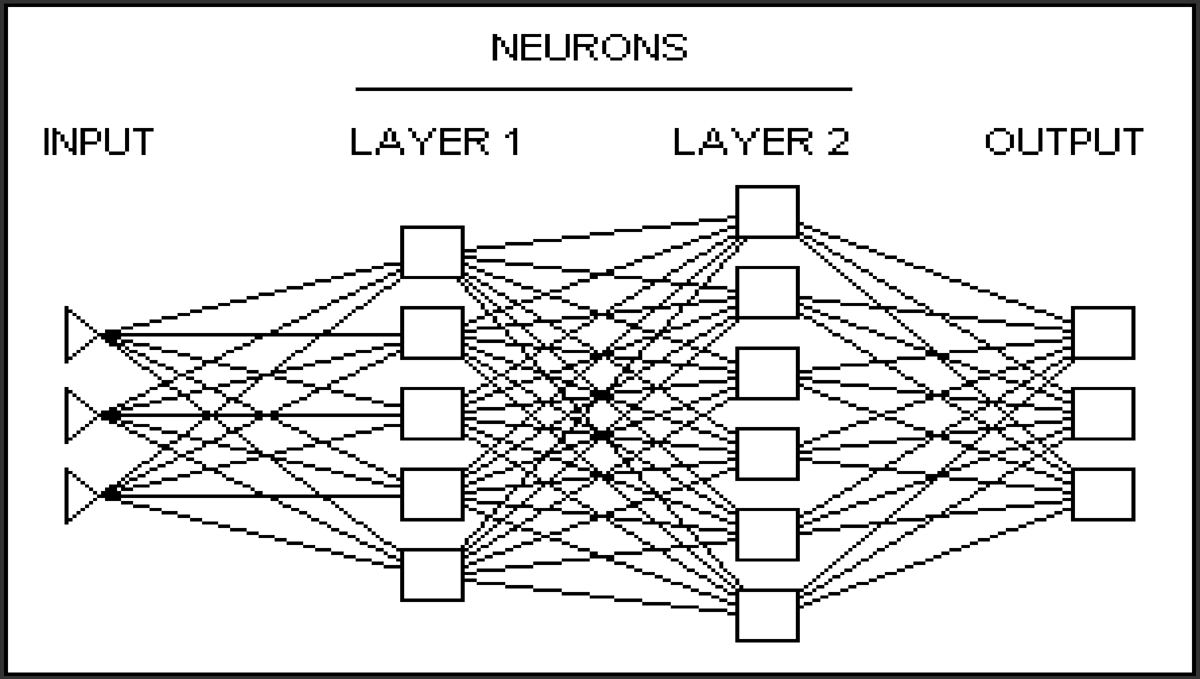
Phoneme Recognition using Reservoirs.

- 2010 paper in NIST [1]
- Main author: Fabian Triefenbach
  - @ Digital Speech and Signal Processing lab
  - ties to Reservoir Lab
  - Ghent University, Belgium

Intro to Reservoir Computation

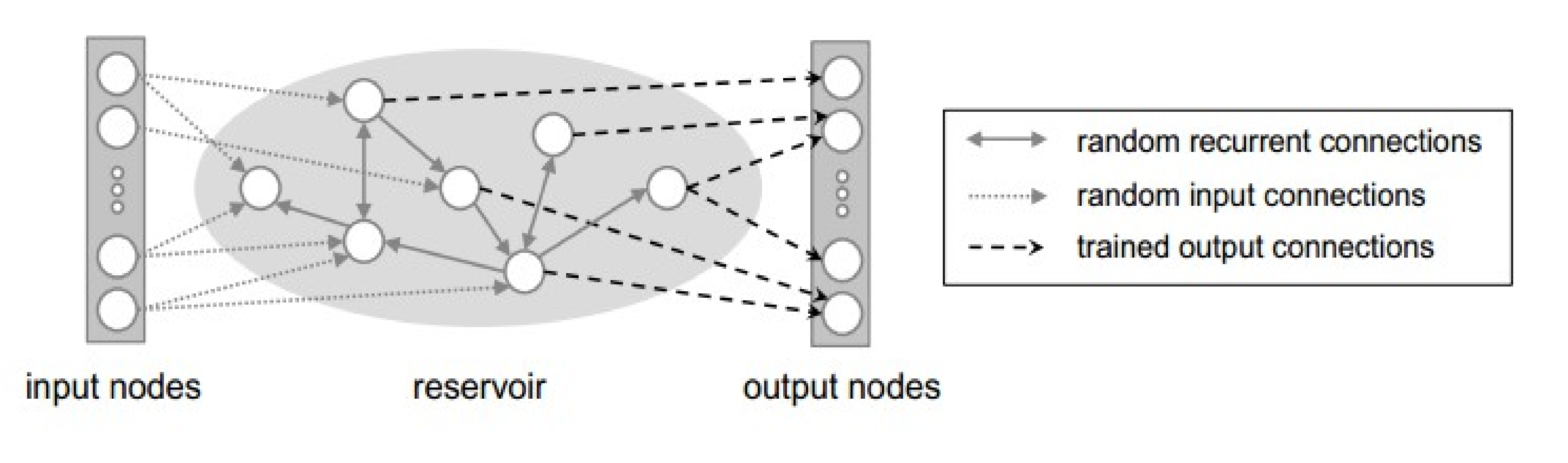
- Biologically inspired neural network
  - Term coined in 2005, evolved from Echo State Networks, Liquid State Machines [2]
- Research topics involving reservoirs:
  - Robotics [3]
  - Machine Learning [4]
  - Neuroscience [5]

Classical Neural Networks



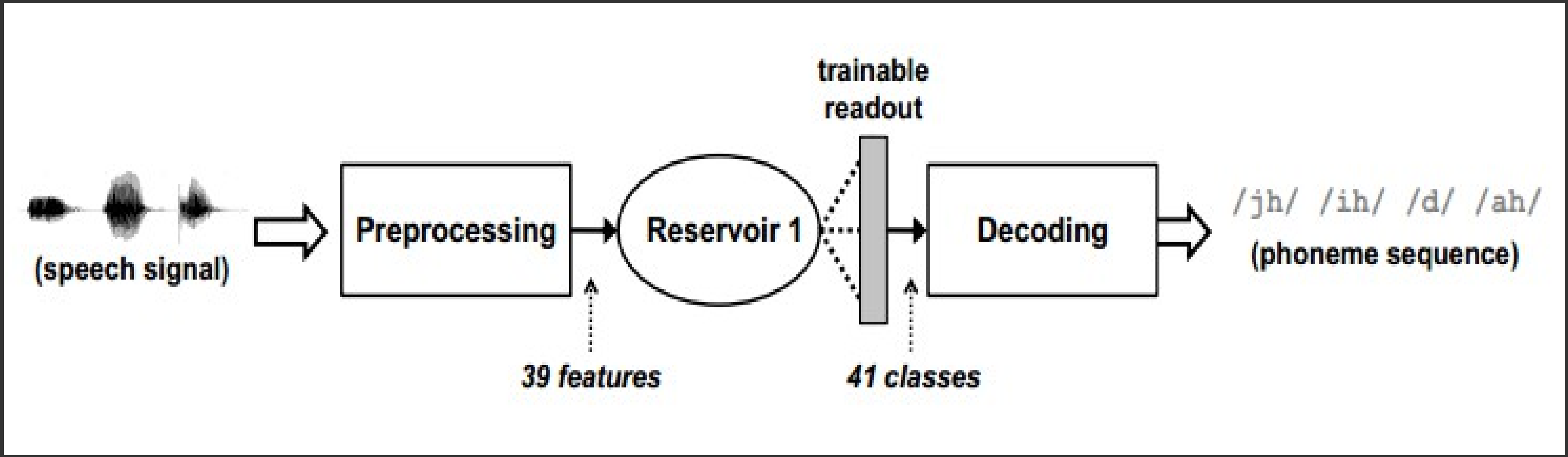
- Usually the weights are learned / trained
  - Ex: back-propagation algorithm
  - This can take a long time to converge

Reservoirs



- Classifiers are linearly trained on the output nodes
  - Unique solution (no local optimum)
  - No iterative process: “one-shot learning”

Phoneme Recognition



- Decoder uses Viterbi algorithm and a phonemic language model (bigram)
- Variant: Hierarchical Extension
  - Multiple layers learn to correct error pattern emerging from lower layers

Reproduction: Preprocessing

- TIMIT Corpus: 630 Speakers, 6100 words fully annotated per time-step
  - MFCC analysis: 25ms Hamming window every 10 ms

    Energy & 12 coefficients + Δs + ΔΔs
  - OpenSMILE used to extract features
- One issue: TIMIT contains 61 different phonetic labels
  - Unclear how the authors reduced it to 41 phoneme labels

Input Rescaling

- Reservoir output is dependent on absolute value of inputs
  - Maximize input impact by computing scaling factors:

$$Z_i = \alpha_i * (U_i - \overline{U}_i)$$

$$Y_i = \beta_i * Z_i$$

  - αs average feature impact (make norm 1)
  - βs free parameter

	Energy features			Cepstral features		
group name	$\log(\mathbf{E})$	$\Delta \log(\mathbf{E})$	$\Delta\Delta \log(\mathbf{E})$	$c_{1\ldots12}$	$\Delta c_{1\ldots12}$	$\Delta\Delta c_{1\ldots12}$
norm factor α	0.27	1.77	4.97	0.10	0.61	1.75
scale factor β	1.75	1.25	1.00	1.25	0.50	0.25

- βs depend on classification on "validation set" that is not described in paper or references
- αs computation poorly described. I had to resort to "forensics"

	log(E)	Δlog(E)	ΔΔlog(E)	c1...c12	Δc1...c12	ΔΔc1...c12
Norm factor α	0.34	2.03	5.16	0.08	0.37	0.90

Reservoir implementation OGER tool

- Python library, easy to install (Linux)
  - built upon the Modular toolkit for Data Processing
- but ... documentation is superficial
  - The format for reservoir data, described only by "input-label pairs", is unclear
  - Work backward from example scripts
- So far, I have been unsuccessful at building a working reservoir classifier

Conclusions

- A lot more work than anticipated
- Many small gaps in the paper
- To be continued in my research...

References

- [1] Phoneme Recognition with Large Hierarchical Reservoirs
  by: Fabian Triefenbach, Azarakhsh Jalalvand, Jean-pierre Martens, Benjamin Schrauwen
  In Neural Information Processing Systems (NIPS2010)
- [2] http://www.scholarpedia.org/article/Echo_state_network
- [3] http://www.amarsi-project.eu
  AMARSi is a EU-funded research project in the Seventh Framework Programme. The project is a large scale integration project hosted in the category Information and Communication Technologies (ICT), unit E5: Cognitive Systems, Interaction and Robotics.
- [4] http://www.humanbrainproject.eu
  The goal of the Human Brain Project is to build a completely new ICT infrastructure for future neuroscience, future medicine and future computing that will catalyse a global collaborative effort to understand the human brain and its diseases and ultimately to emulate its computational capabilities.
- [5] Temporal Recurrent Networks
  see http://organic.elis.ugent.be/flavors and Peter F. Dominey's work.