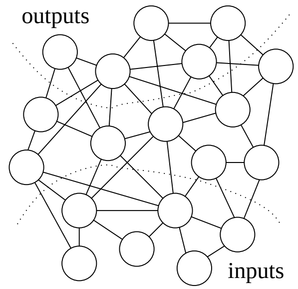
Temporal processing: Recurrent networks

Learning in networks with unrestricted connectivity:
Back-propagation through time

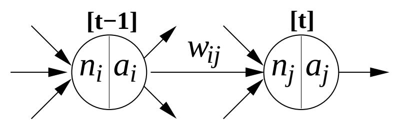
- Repeatedly update unit activations synchronously (first n_j, then a_j)
- Store entire activation history of each unit
- Attribute error to sending activations computed earlier in time



$$n_j^{[t]} = \sum_i a_i^{[t-1]} w_{ij}$$

$$a_j^{[t]} = \frac{1}{1+\exp\left(-n_j^{[t]}\right)}$$

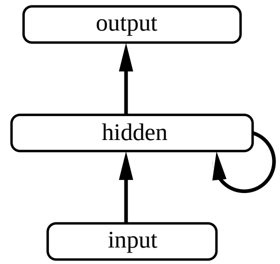
$$E^{[t]} = \frac{1}{2}\sum_j \left(t_j^{[t]} - a_j^{[t]}\right)^2$$



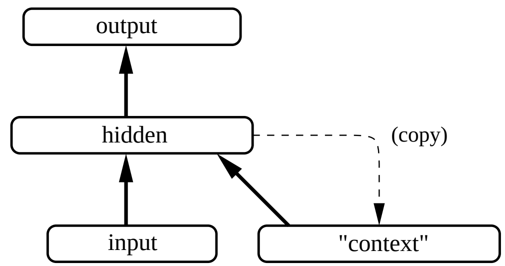
$$n_i^{[t-1]} \longrightarrow a_i^{[t-1]} \xrightarrow{w_{ij}} n_j^{[t]} \longrightarrow a_j^{[t]} \xrightarrow{t_j^{[t]}} E$$

Temporal processing: Simple recurrent networks

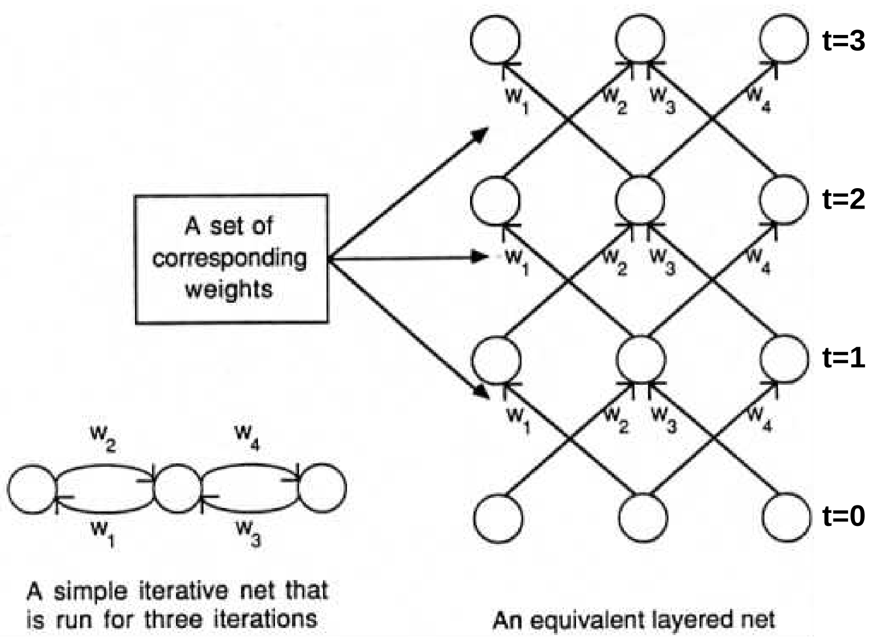
- Fully recurrent networks are computationally intensive to simulate
  - Must update unit activities multiple times per input



- "Simple" recurrent networks (Elman, 1990) adapt feedforward networks to learn temporal tasks
  - Computationally efficient but functionally limited compared to fully recurrent networks



"Unfolding" a recurrent network into a feedforward one



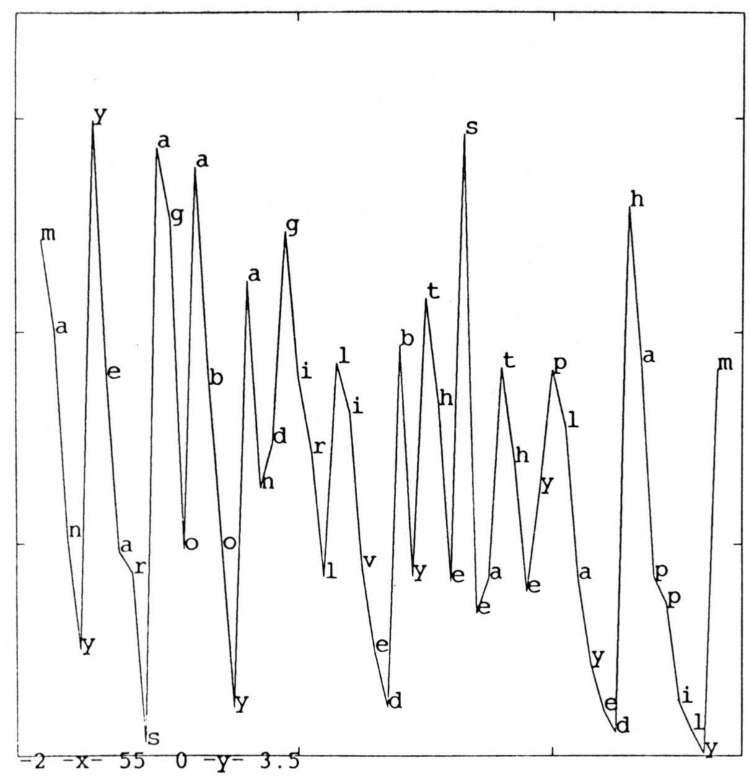


TABLE 3
Categories of Lexical Items Used in Sentence Simulation

Category	Examples
NOUN-HUM	man, woman
NOUN-ANIM	cat, mouse
NOUN-INANIM	book, rock
NOUN-AGRESS	dragon, monster
NOUN-FRAG	glass, plate
NOUN-FOOD	cookie, break
VERB-INTRAN	think, sleep
VERB-TRAN	see, chase
VERB-AGPAT	move, break
VERB-PERCEPT	smell, see
VERB-DESTROY	break, smash
VERB-EAT	eat

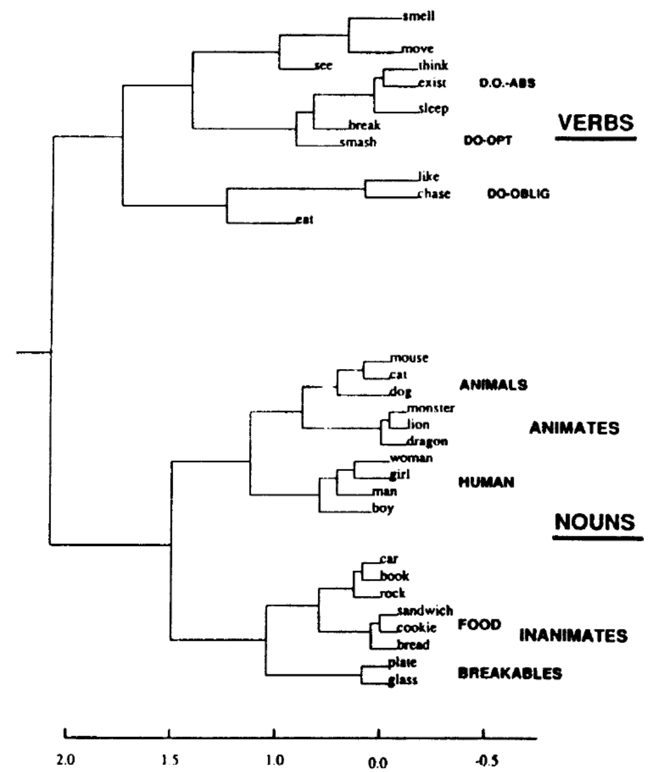


TABLE 4
Templates for Sentence Generator

WORD 1	WORD 2	WORD 3
NOUN-HUM	VERB-EAT	NOUN-FOOD
NOUN-HUM	VERB-PERCEPT	NOUN-INANIM
NOUN-HUM	VERB-DESTROY	NOUN-FRAG
NOUN-HUM	VERB-INTRAN	
NOUN-HUM	VERB-TRAN	NOUN-HUM
NOUN-HUM	VERB-AGPAT	NOUN-INANIM
NOUN-HUM	VERB-AGPAT	
NOUN-ANIM	VERB-EAT	NOUN-FOOD
NOUN-ANIM	VERB-TRAN	NOUN-ANIM
NOUN-ANIM	VERB-AGPAT	NOUN-INANIM
NOUN-ANIM	VERB-AGPAT	
NOUN-INANIM	VERB-AGPAT	
NOUN-AGRESS	VERB-DESTROY	NOUN-FRAG
NOUN-AGRESS	VERB-EAT	NOUN-HUM
NOUN-AGRESS	VERB-EAT	NOUN-ANIM
NOUN-AGRESS	VERB-EAT	NOUN-FOOD