Unit 1: Simple Neural Networks

9. Recurrent neural networks

9/29/2020

Recurrent networks

1. Recurrent neural networks can discover structure in time

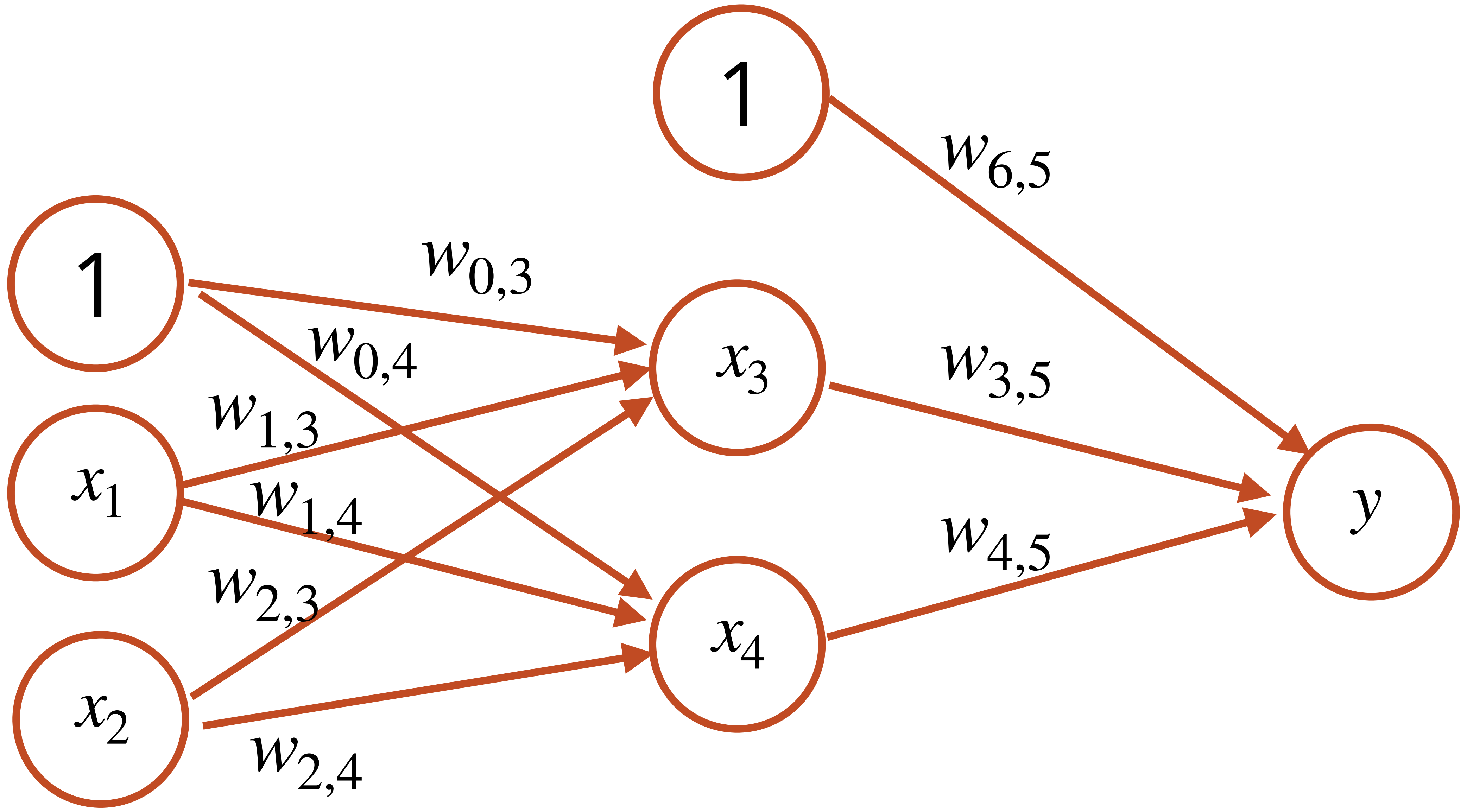
2. Connectionism recap

3. Voting resources

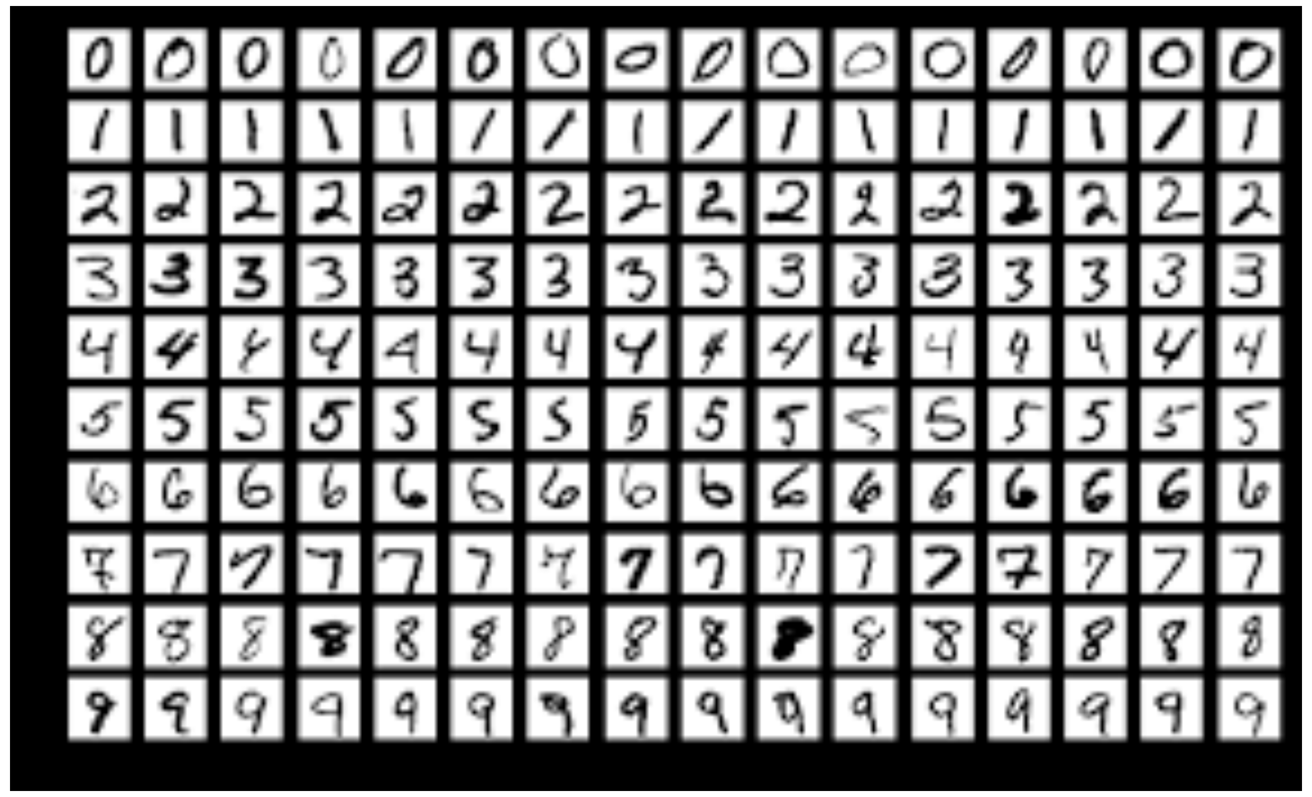
4. Homework 2

What these networks can do

Networks like this one can solve problems where there is structure in co-occurrence



With a little modification, they can also find structure in space (as you'll see in the Homework 2)



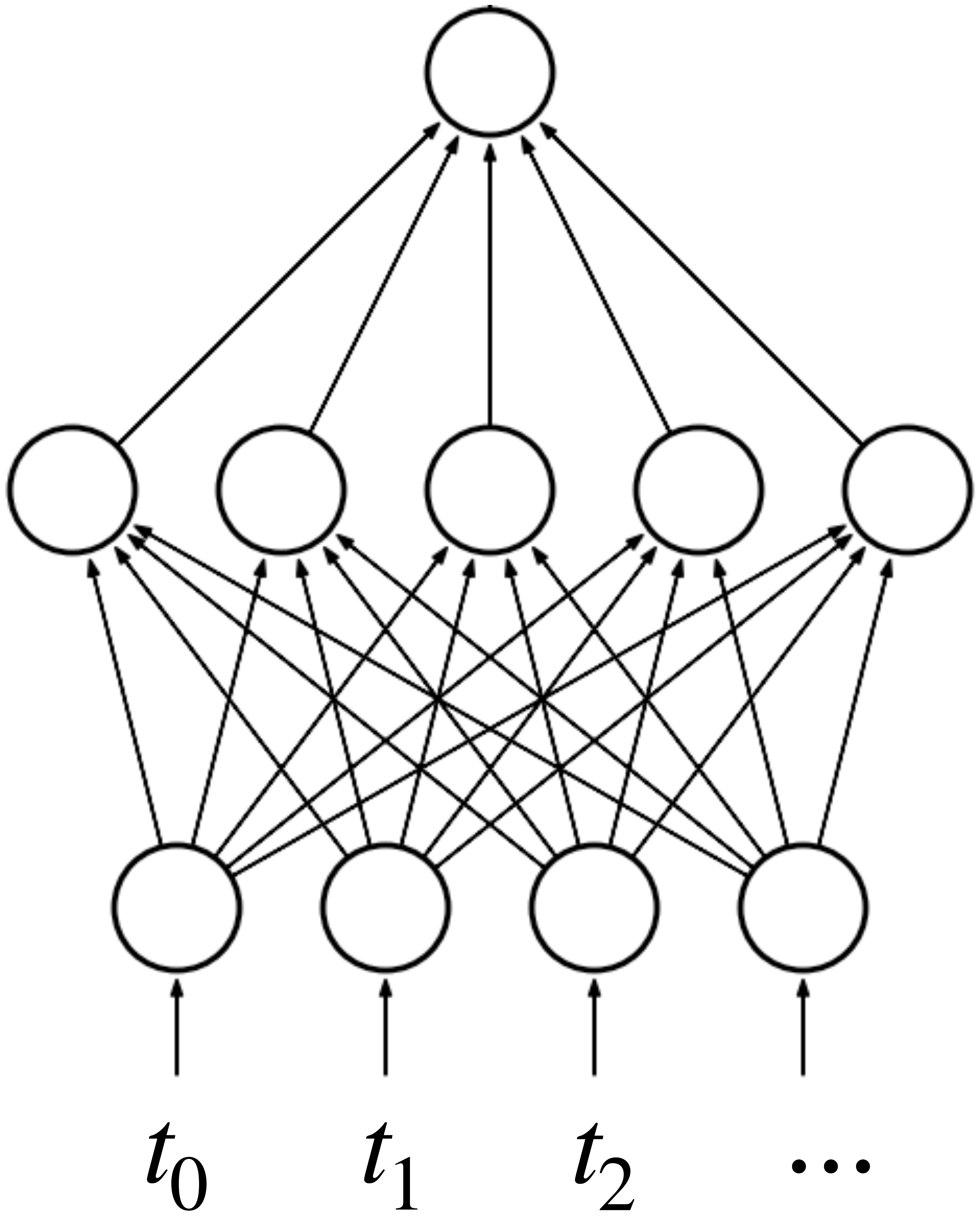
What about structure in time?

Many of the things people learn, and we want machines to learn are about **structure in time**

From our affinity diagram:

- Learning a song
- Learning to knit
- Playing video games
- Learning a dance routine
- Driving
- Cooking
- …

How would we represent structure in a network?

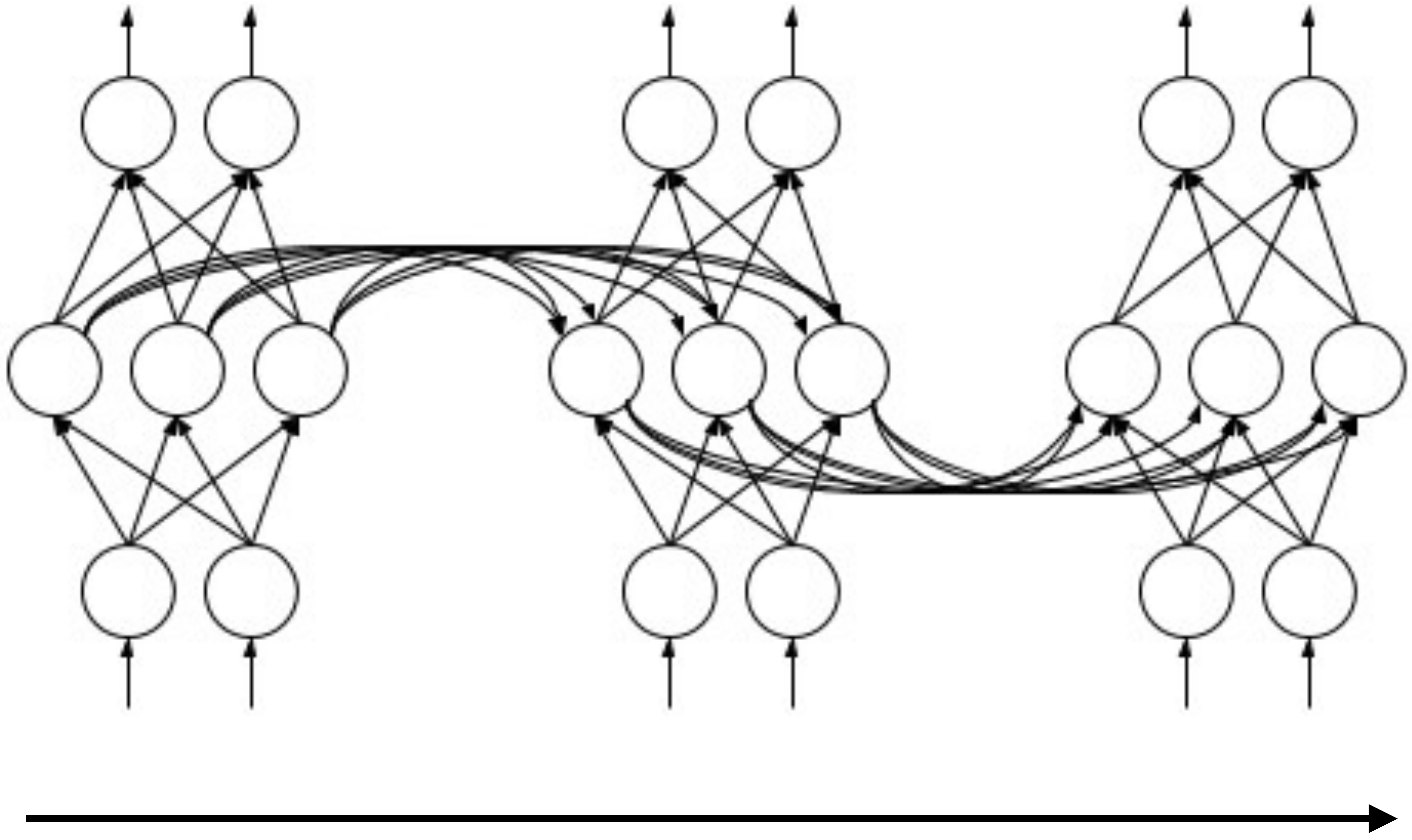


Naive approach: Time as space

Problems:

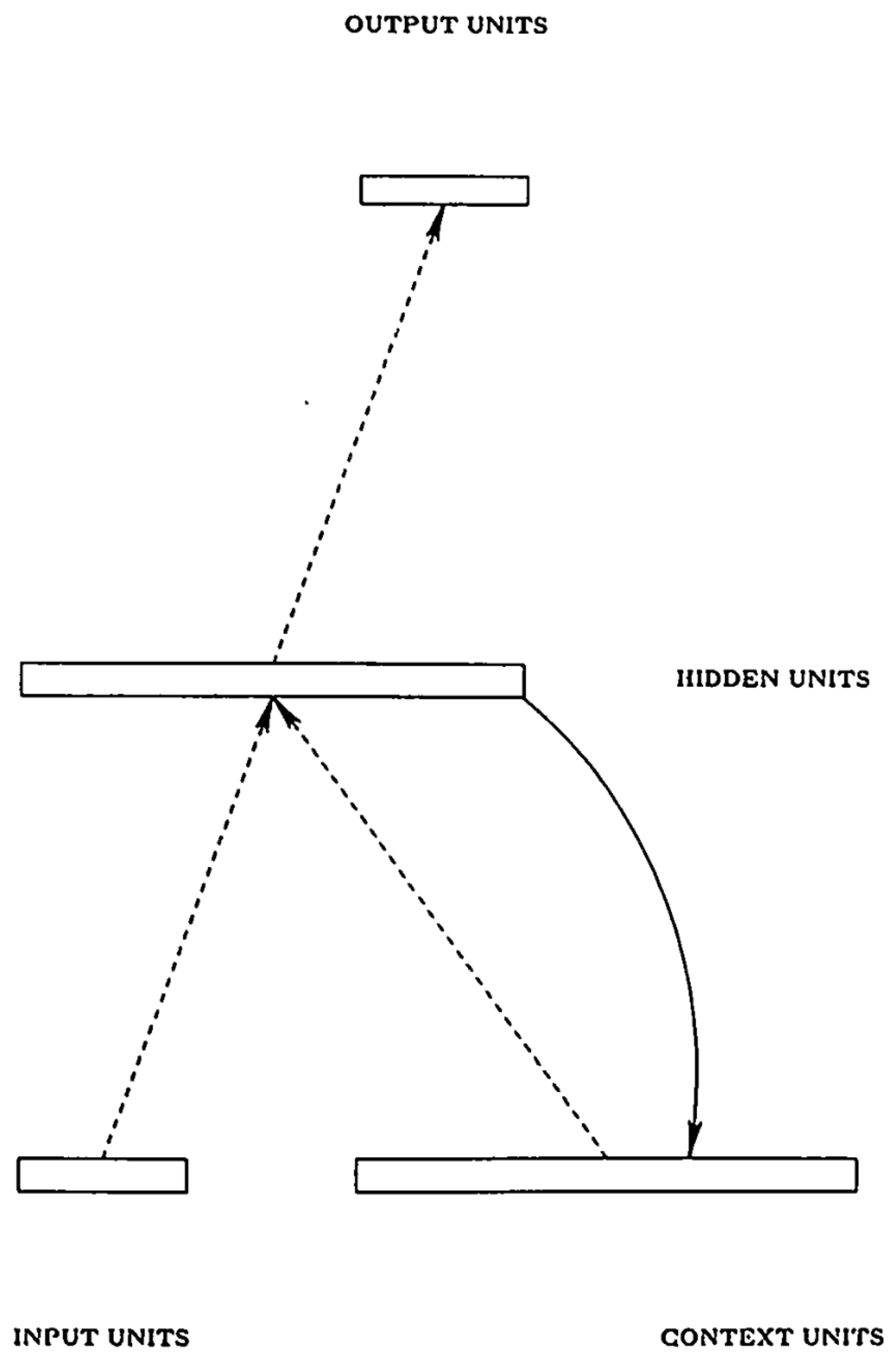
- How big do you make the buffer?
- Two identical patterns translated in time have no natural overlap, e.g. **[0 1 1 0 0 0]** and **[0 0 0 1 1 0]**

Recurrent Neural Networks



Time

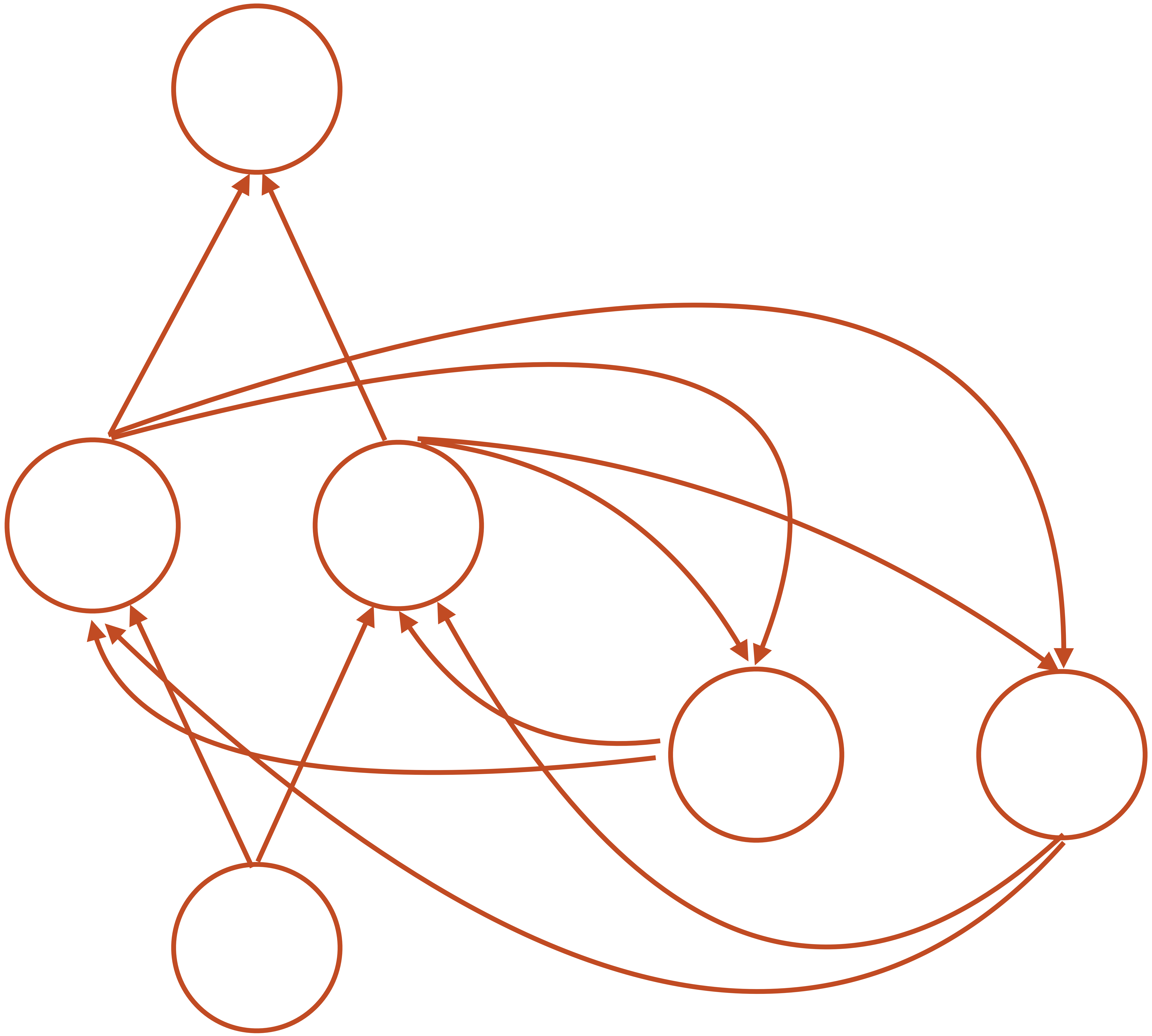
Simple Recurrent Networks (Elman Networks - Elman, 1990)



A set of **context units** that are an exact copy of the hidden layer at $t-1$

The hidden layer at time t gets input from both the **input units** and the **context units**

XOR in an Elman network



Output

0 1 1 0 0 0 1 1 1 1 0 0 0 1 ?

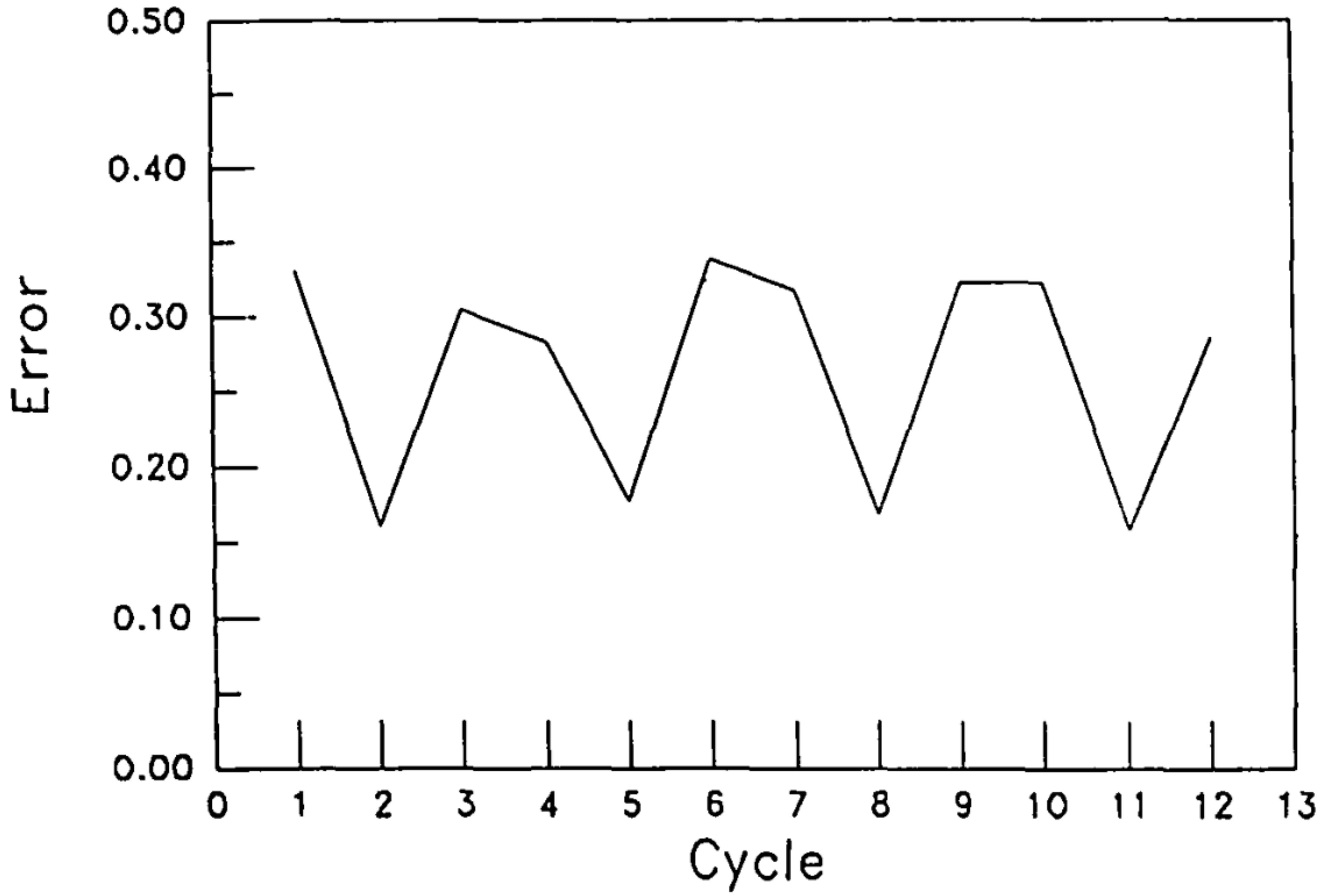
1 0 1 0 0 0 0 1 1 1 1 0 1 0 1...

Input

Goal: Output is XOR of previous 2 inputs

Error after training

What is happening here?



Output

0 1 0 0 0 0 1 1 1 1 0 1 0 1 ?

1 0 1 0 0 0 0 1 1 1 1 0 1 0 1…

Input

How far back does the network remember?

Input

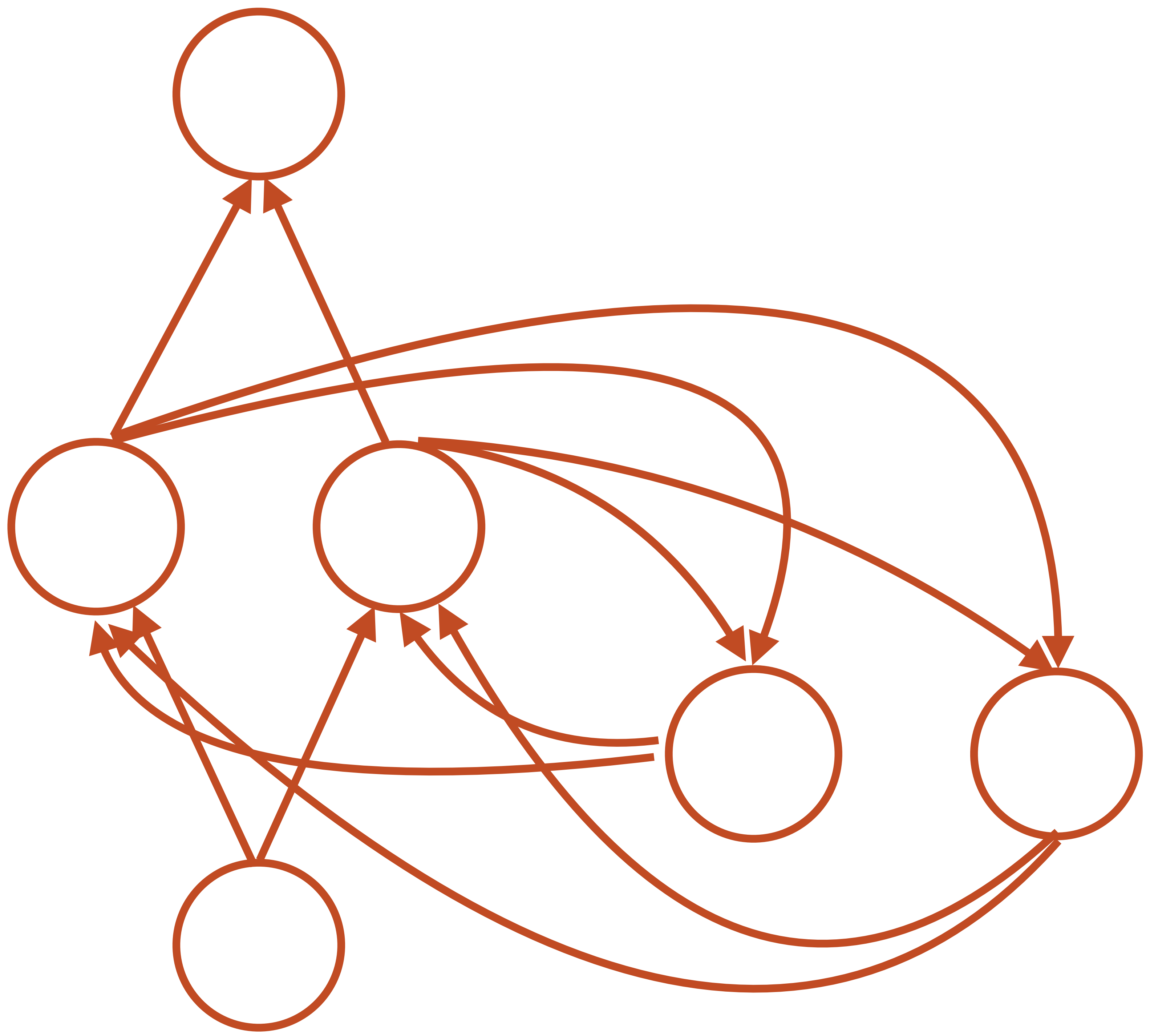
A random concatenation of the words **ba**, **dii**, and **guuu**

badiibaguuubadiguuguudiba ...

Output

The next letter in the sequence

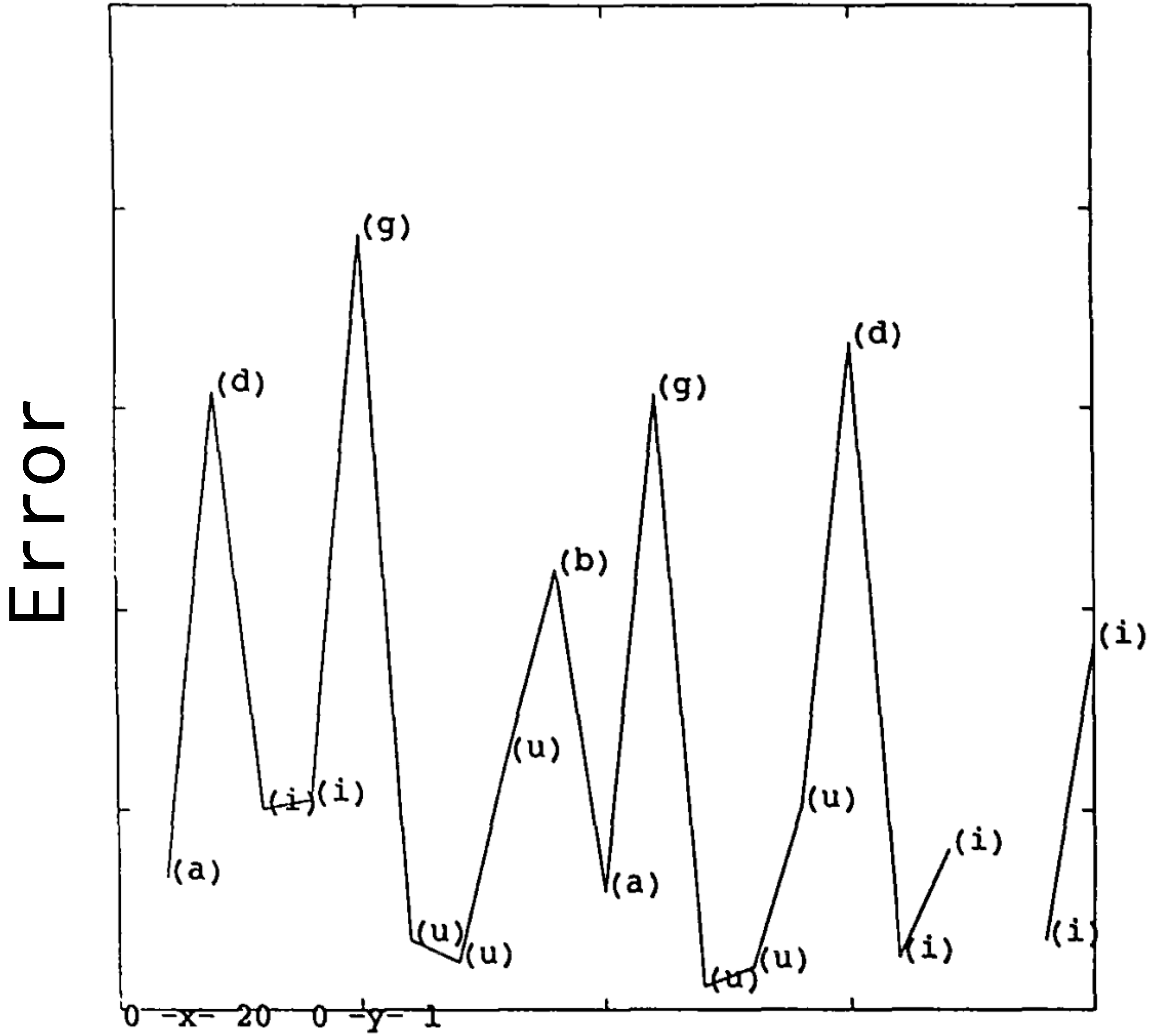
a d i i b a g u u...



Vector Definitions of Alphabet

		Consonant	Vowel	Interrupted	High	Back	Voiced	
b	[	1	0	1	0	0	1	]
d	[	1	0	1	1	0	1	]
g	[	1	0	1	0	1	1	]
a	[	0	1	0	0	1	1	]
i	[	0	1	0	1	0	1	]
u	[	0	1	0	1	1	1	]

What has the network learned?

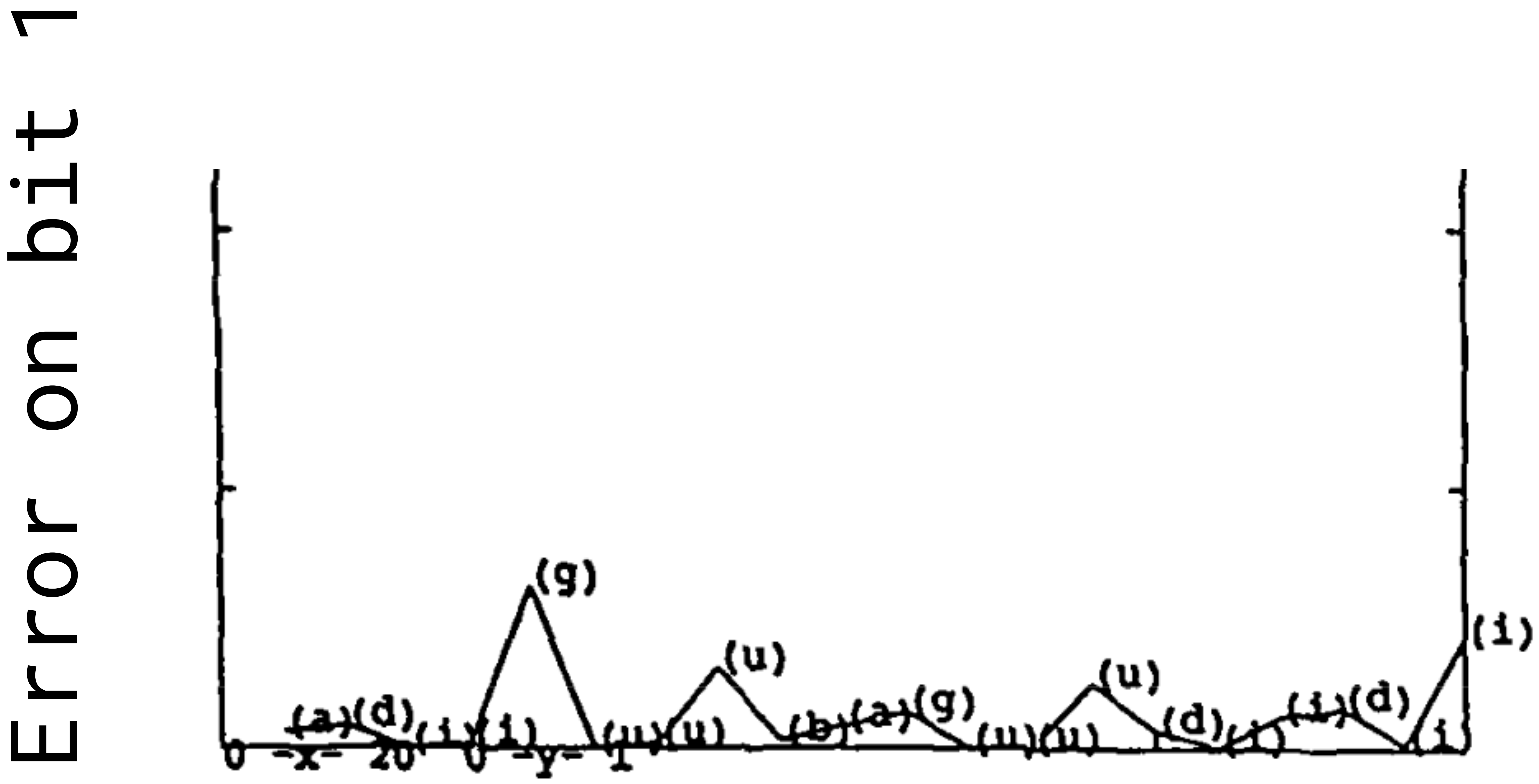


1. Averaging over the bits, it learns **which letters form words**

Vector Definitions of Alphabet

		Consonant	Vowel	Interrupted	High	Back	Voiced	
b	[	1	0	1	0	0	1	]
d	[	1	0	1	1	0	1	]
g	[	1	0	1	0	1	1	]
a	[	0	1	0	0	1	1	]
i	[	0	1	0	1	0	1	]
u	[	0	1	0	1	1	1	]

What has the network learned?

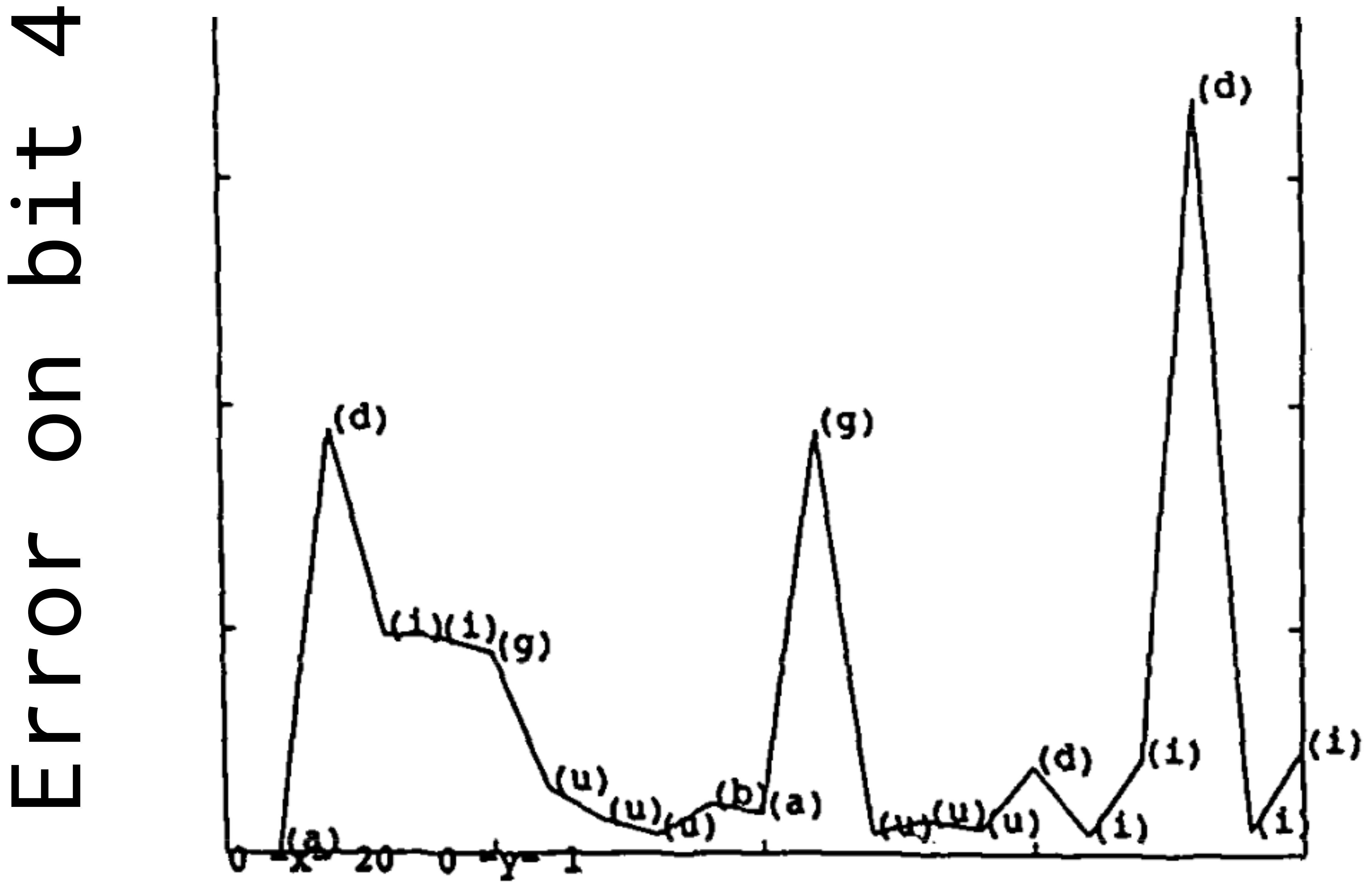


Vector Definitions of Alphabet

	Consonant	Vowel	Interrupted	High	Back	Voiced
b	[1	0	1	0	0	1]
d	[1	0	1	1	0	1]
g	[1	0	1	0	1	1]
a	[0	1	0	0	1	1]
i	[0	1	0	1	0	1]
u	[0	1	0	1	1	1]

2. **Vowels** follow **consonants**

What has the network learned?



Vector Definitions of Alphabet

		Consonant	Vowel	Interrupted	High	Back	Voiced	
b	[	1	0	1	0	0	1	]
d	[	1	0	1	1	0	1	]
g	[	1	0	1	0	1	1	]
a	[	0	1	0	0	1	1	]
i	[	0	1	0	1	0	1	]
u	[	0	1	0	1	1	1	]

2. Because **consonants** differ on the **High** feature, it knows that a consonant is coming but not **which one**

What is a “word”?

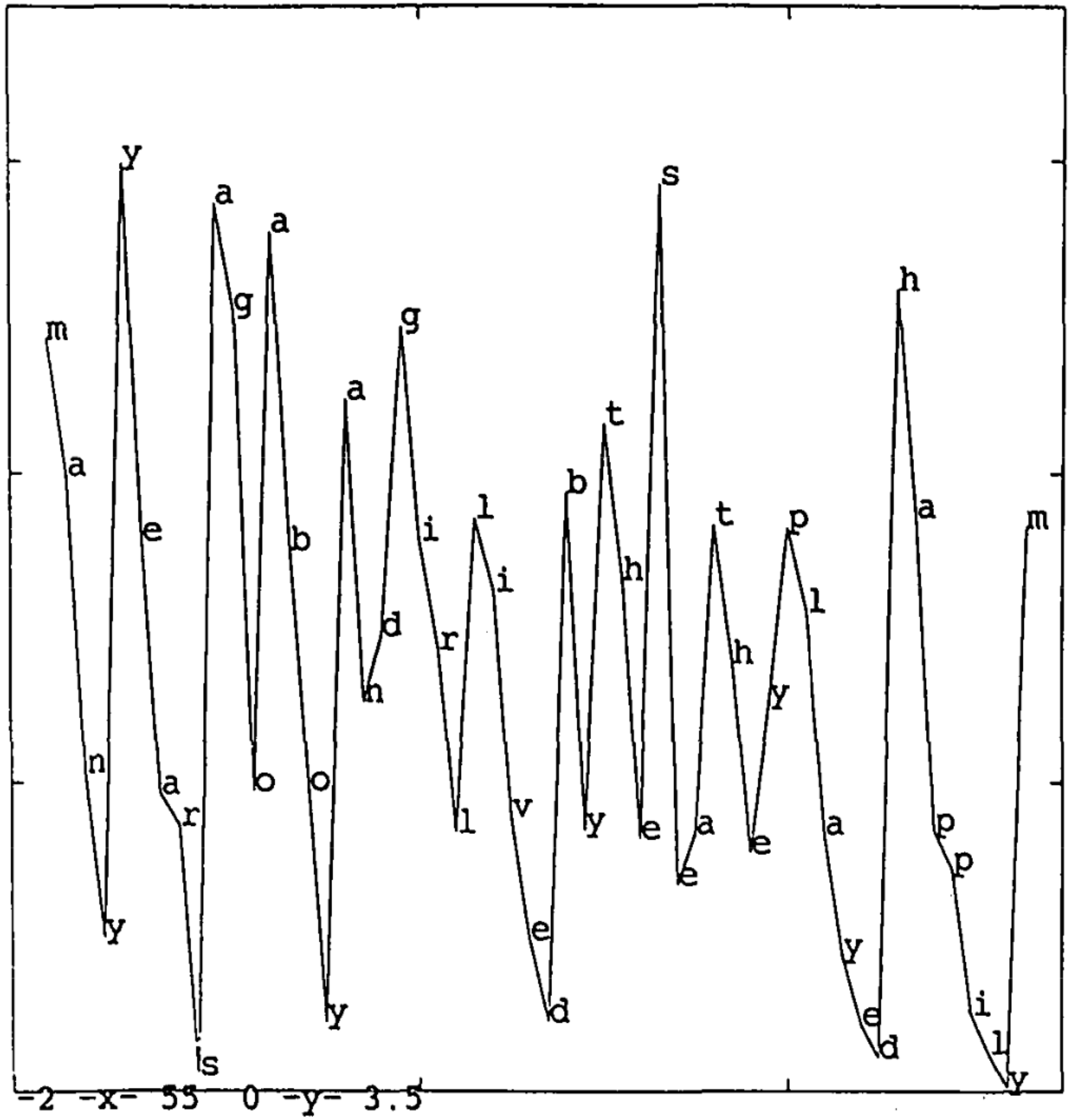
Input

A concatenation of words in English Sentences

manyearsagoaboyandagirl …

A lot of the previous work **assumed** structure in Language (e.g., phonemes, morphemes, words)

But what if “**words**” are just sequences of low prediction error



Syntactic structure through prediction error

Input

A concatenation of triplets

subject - verb - object

womansmashplatecatmovemanbreak

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

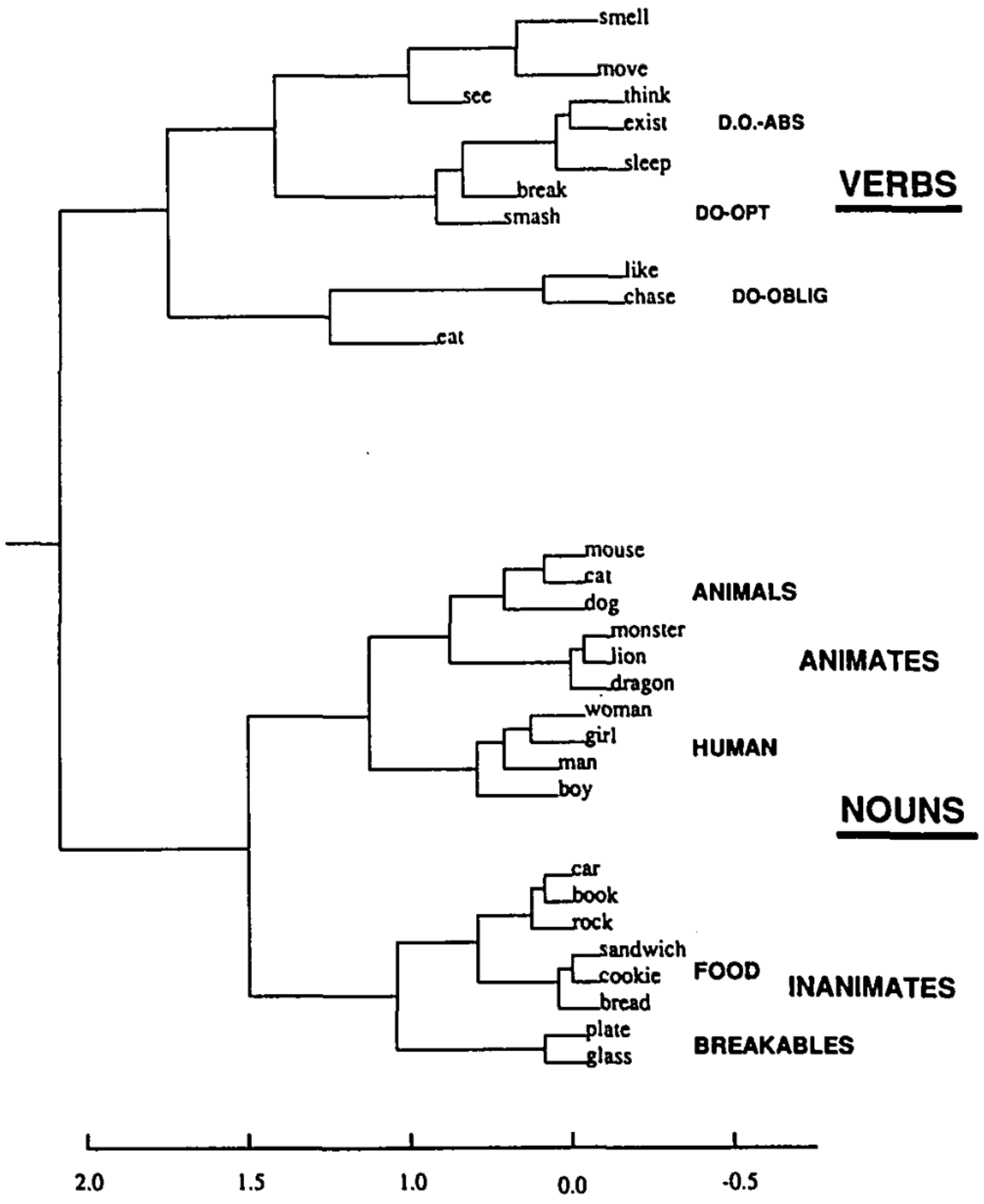
Input	Output
0000000000000000000000000000010 (*woman*)	0000000000000000000000000010000 (*smash*)
0000000000000000000000000010000 (*smash*)	0000000000000000000001000000000 (*plate*)
0000000000000000000001000000000 (*plate*)	0000010000000000000000000000000 (*cat*)
0000010000000000000000000000000 (*cat*)	0000000000000000000100000000000 (*move*)
0000000000000000000100000000000 (*move*)	0000000000000000100000000000000 (*man*)
0000000000000000100000000000000 (*man*)	0001000000000000000000000000000 (*break*)
0001000000000000000000000000000 (*break*)	0000100000000000000000000000000 (*car*)
0000100000000000000000000000000 (*car*)	0100000000000000000000000000000 (*boy*)
0100000000000000000000000000000 (*boy*)	0000000000000000000100000000000 (*move*)
0000000000000000000100000000000 (*move*)	0000000000001000000000000000000 (*girl*)
0000000000001000000000000000000 (*girl*)	0000000000100000000000000000000 (*eat*)
0000000000100000000000000000000 (*eat*)	0010000000000000000000000000000 (*bread*)
0010000000000000000000000000000 (*bread*)	0000000010000000000000000000000 (*dog*)
0000000010000000000000000000000 (*dog*)	0000000000000000000100000000000 (*move*)
0000000000000000000100000000000 (*move*)	0000000000000000001000000000000 (*mouse*)
0000000000000000001000000000000 (*mouse*)	0000000000000000001000000000000 (*mouse*)
0000000000000000001000000000000 (*mouse*)	0000000000000000000100000000000 (*move*)
0000000000000000000100000000000 (*move*)	1000000000000000000000000000000 (*book*)
1000000000000000000000000000000 (*book*)	0000000000000001000000000000000 (*lion*)

Syntactic structure through prediction error

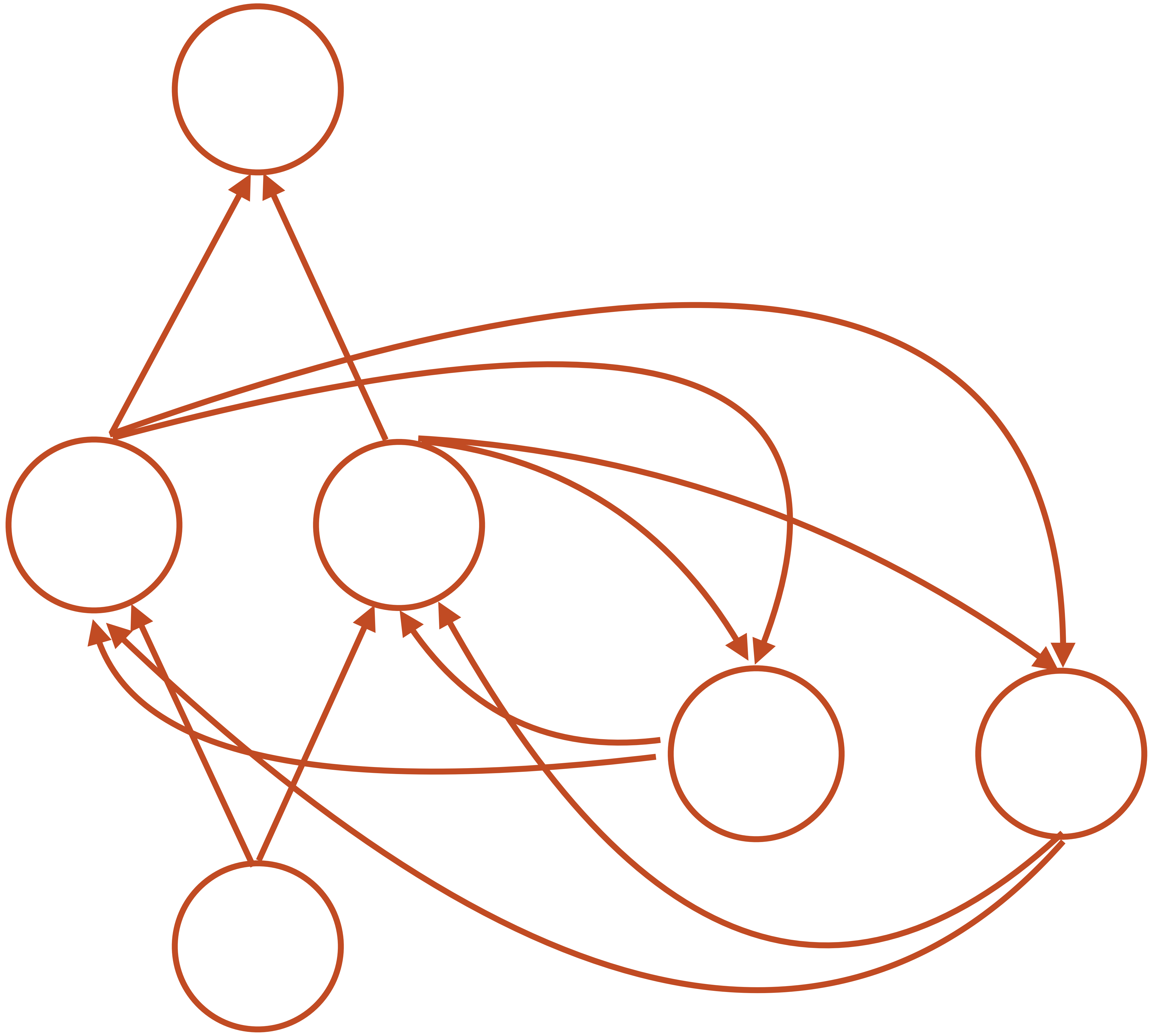
Hierarchically clustering the hidden layer activations for words reveals structure!

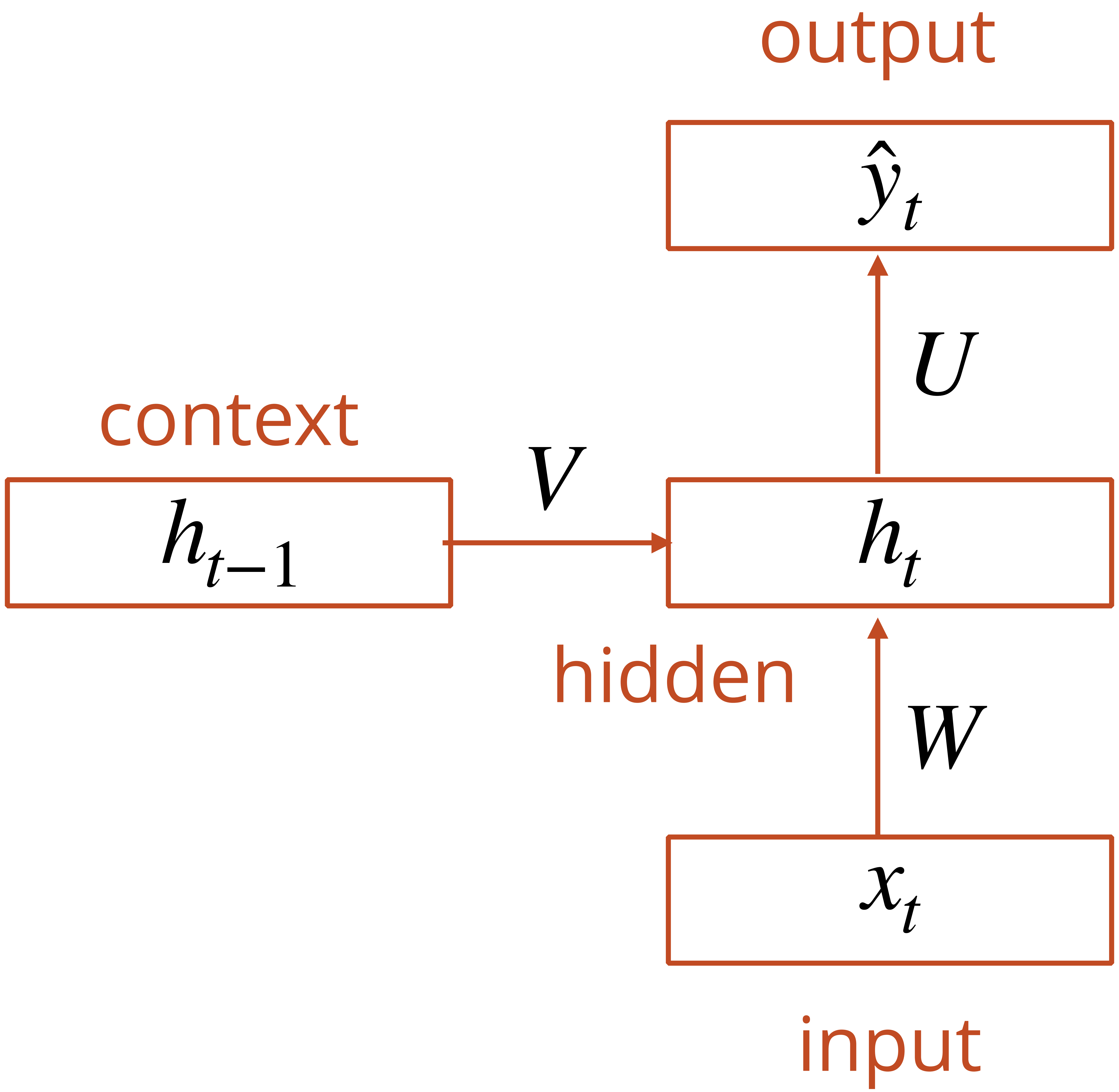
The network learns **syntactic** and **semantic** roles

Why?



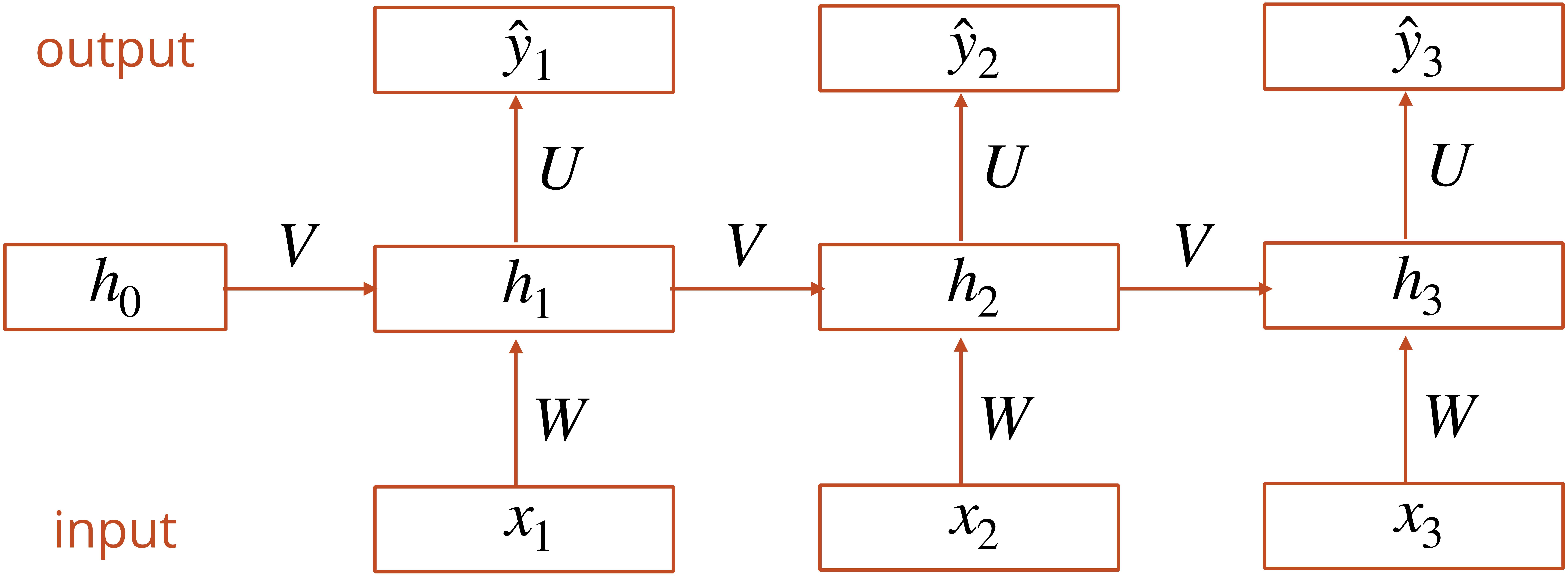
Training a Recurrent Neural Network





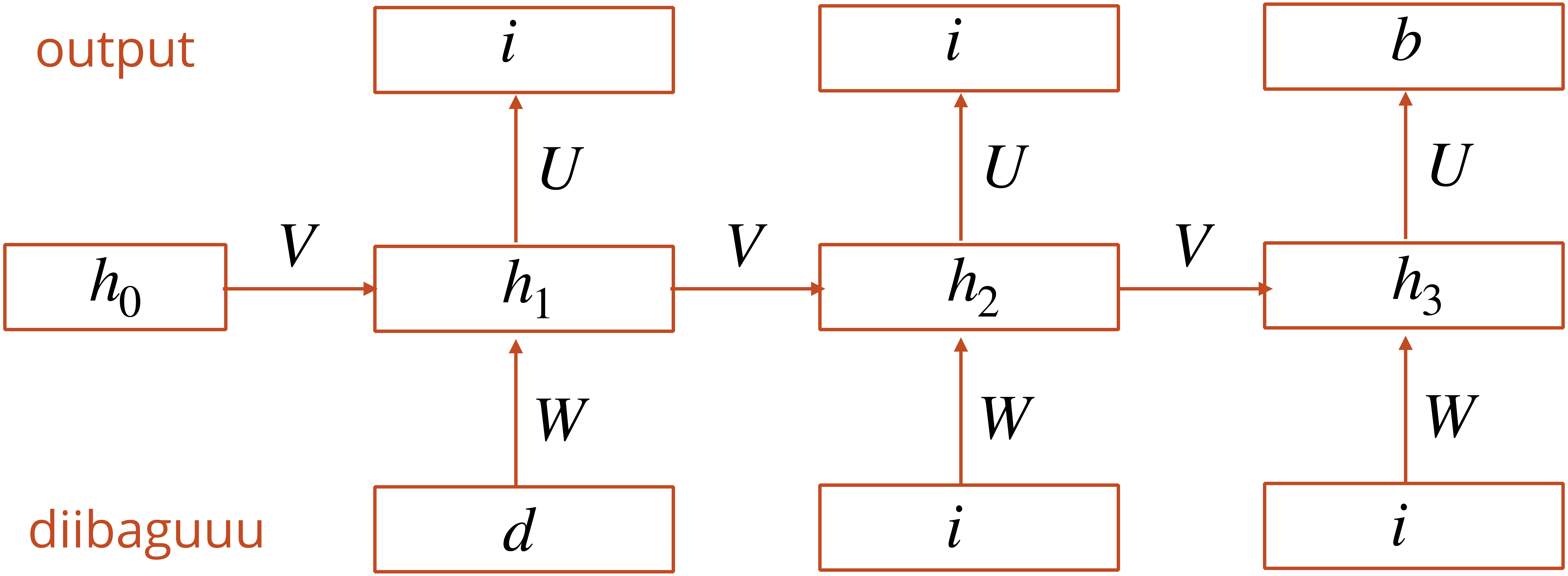
Unrolling a network in time

Global Error: $E = \sum_t E_t$



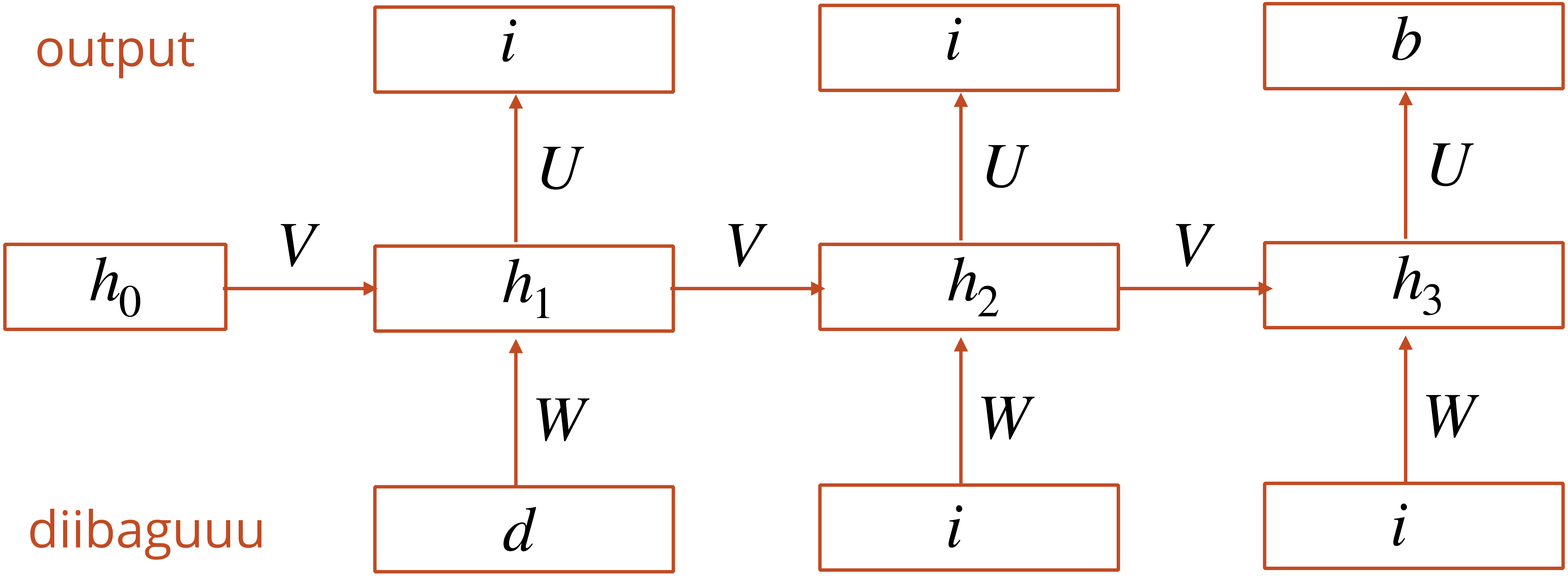
Unrolling a network in time

Global Error: $E = \sum_t E_t$



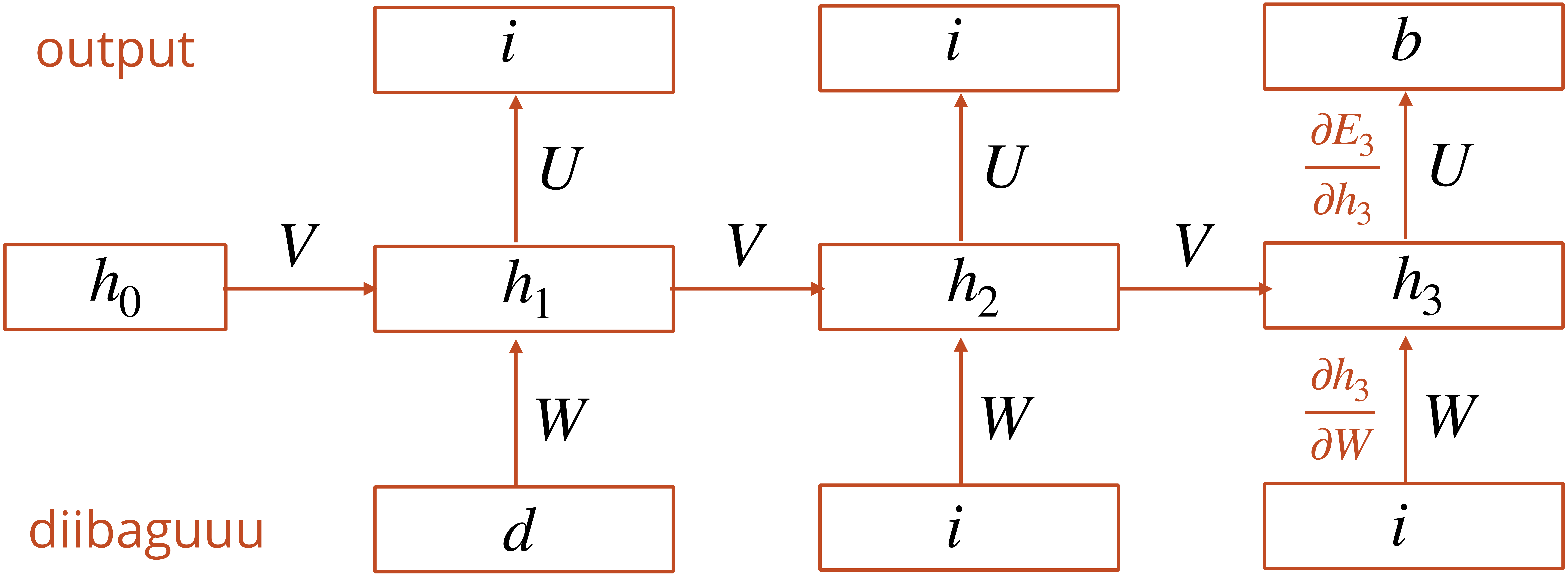
Backpropagation through time

Global Error: $E = \sum_t E_t$ $\quad \dfrac{\partial E}{\partial W} = \sum_t \dfrac{\partial E_t}{\partial W}$



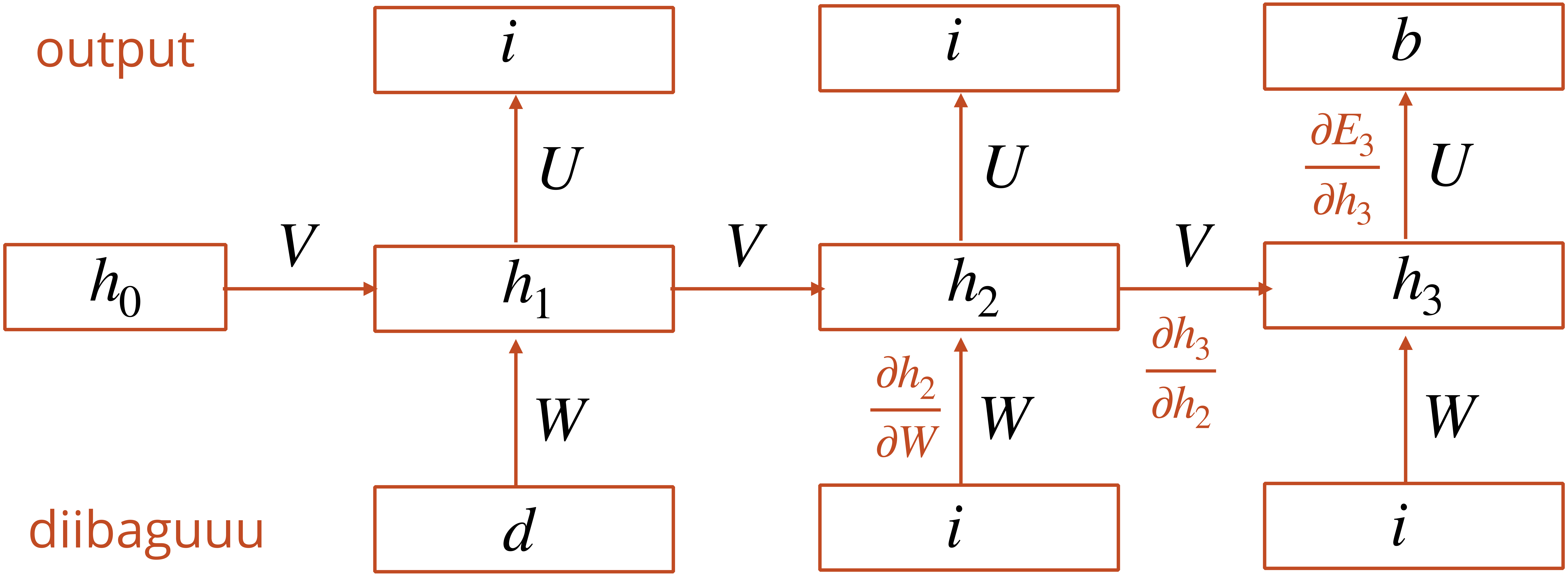
Backpropagation through time

Global Error: $E = \sum_t E_t$ $\qquad \frac{\partial E}{\partial W} = \sum_t \frac{\partial E_t}{\partial W}$



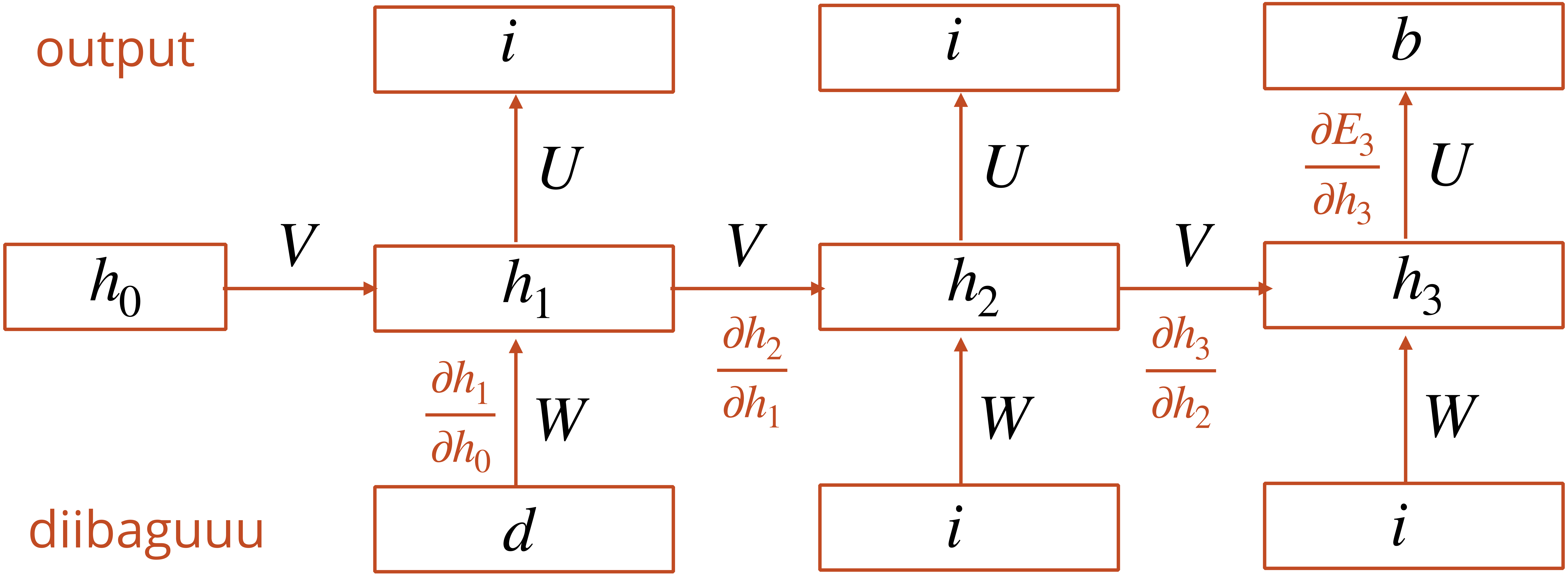
Backpropagation through time

Global Error: $E = \sum_t E_t$ $\qquad \frac{\partial E}{\partial W} = \sum_t \frac{\partial E_t}{\partial W}$

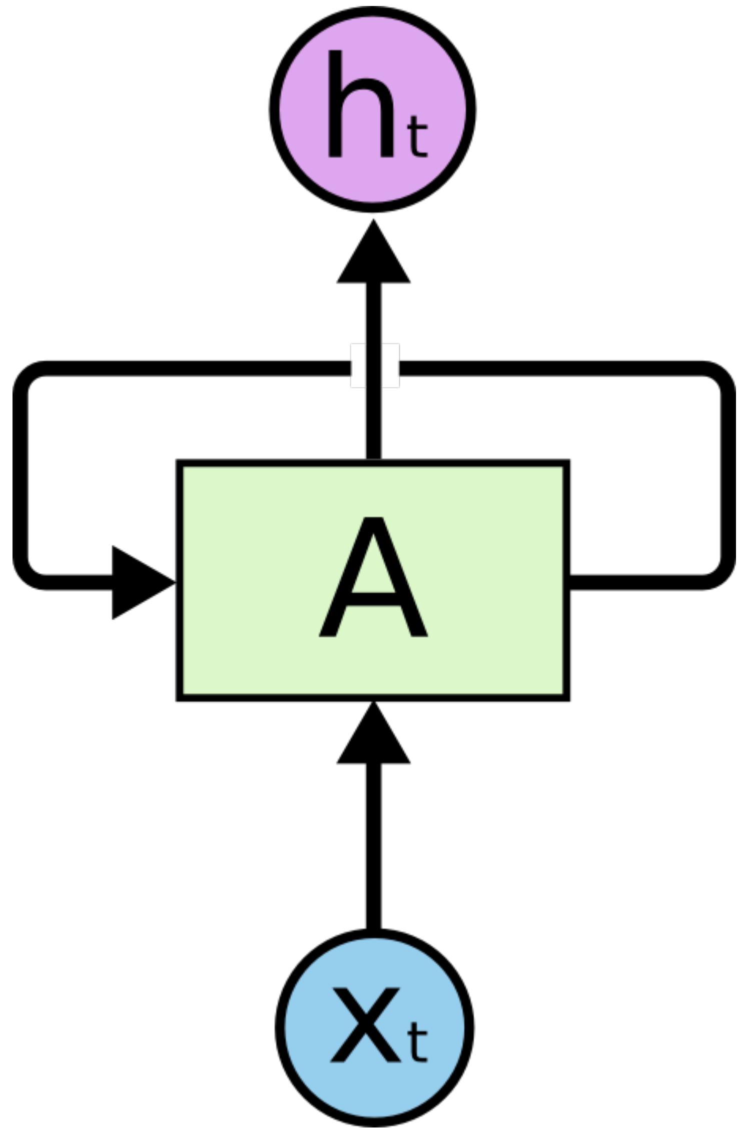


Backpropagation through time

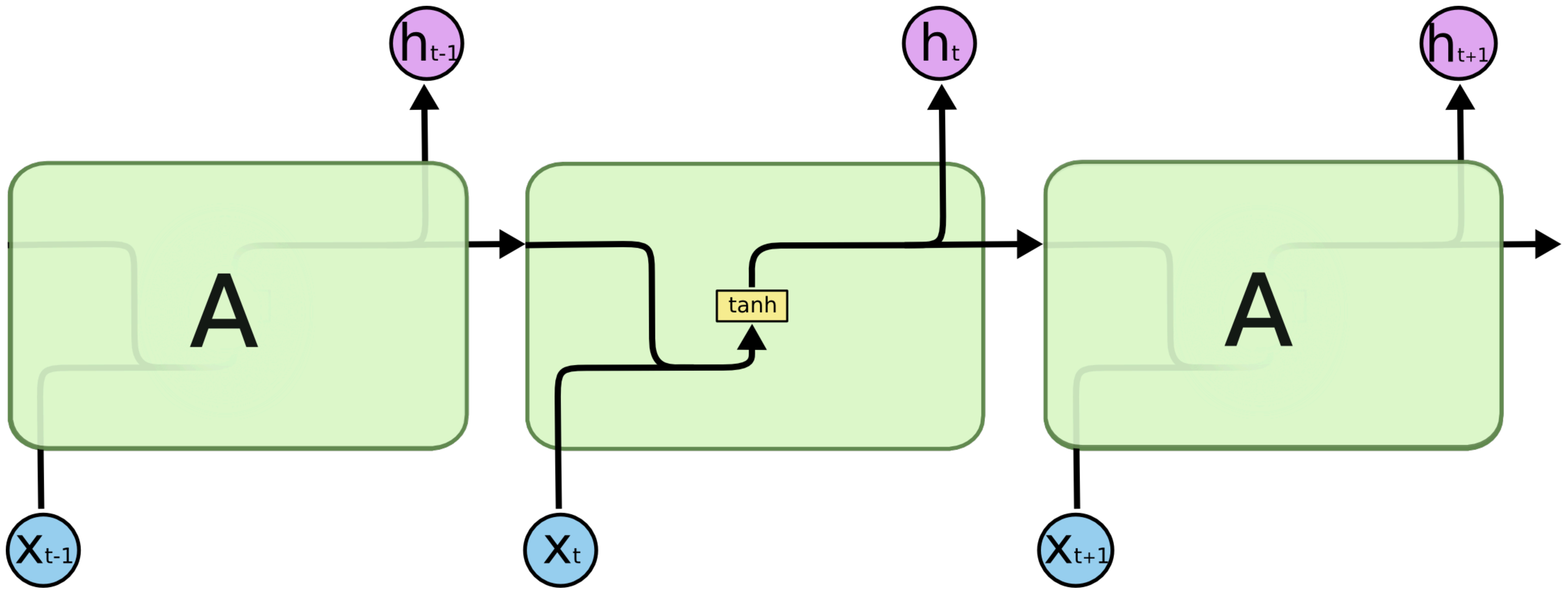
Global Error: $E = \sum_t E_t$ $\qquad \frac{\partial E}{\partial W} = \sum_t \frac{\partial E_t}{\partial W}$



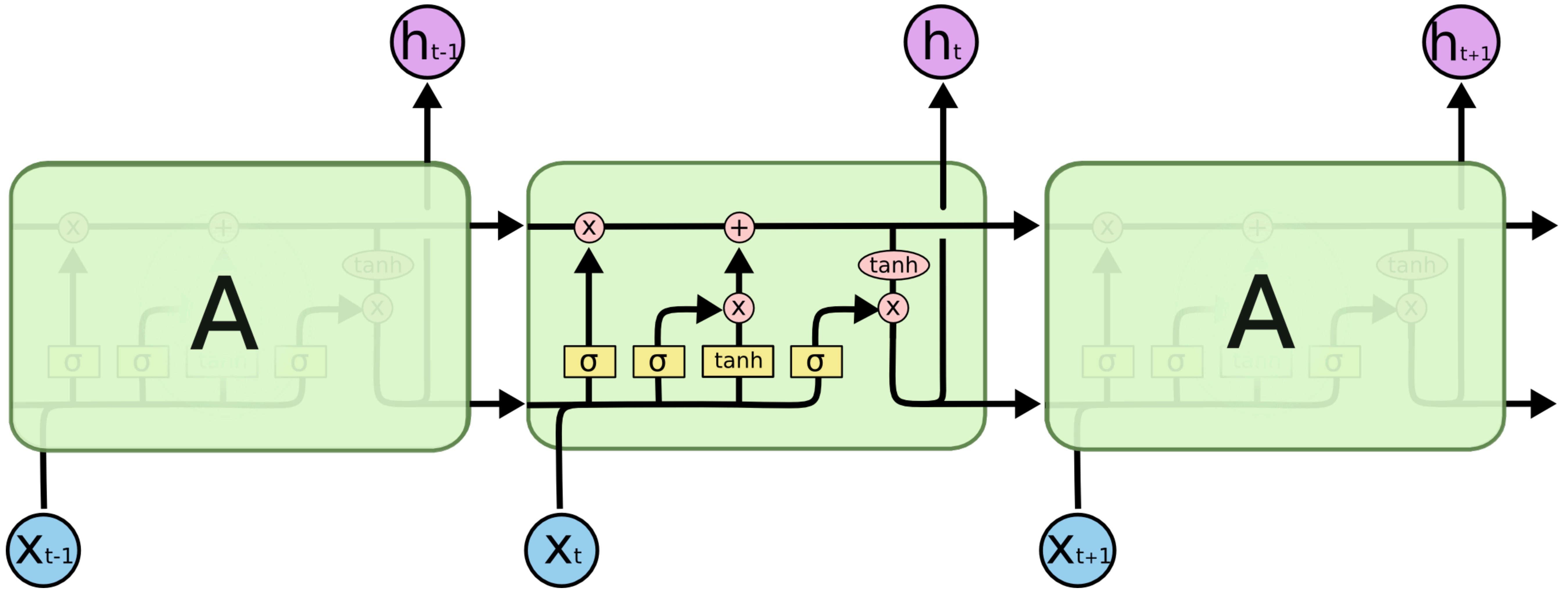
Modern language models extend this idea



Elman Network

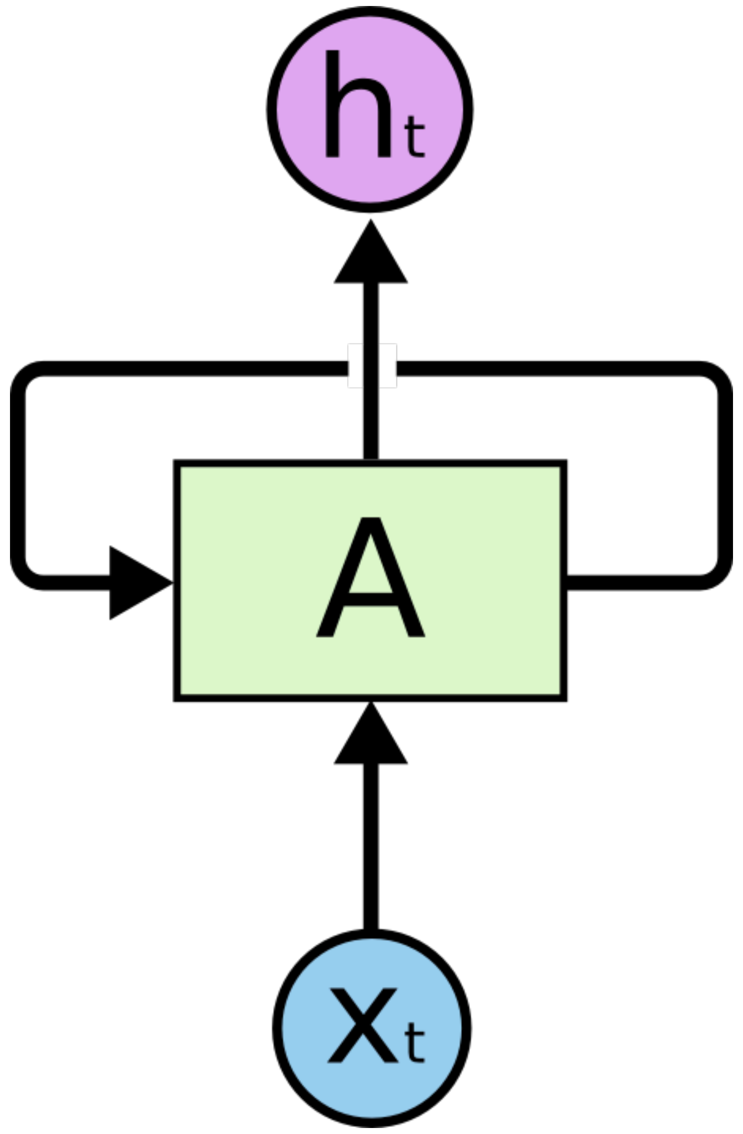


Long Short Term Memory (LSTM)

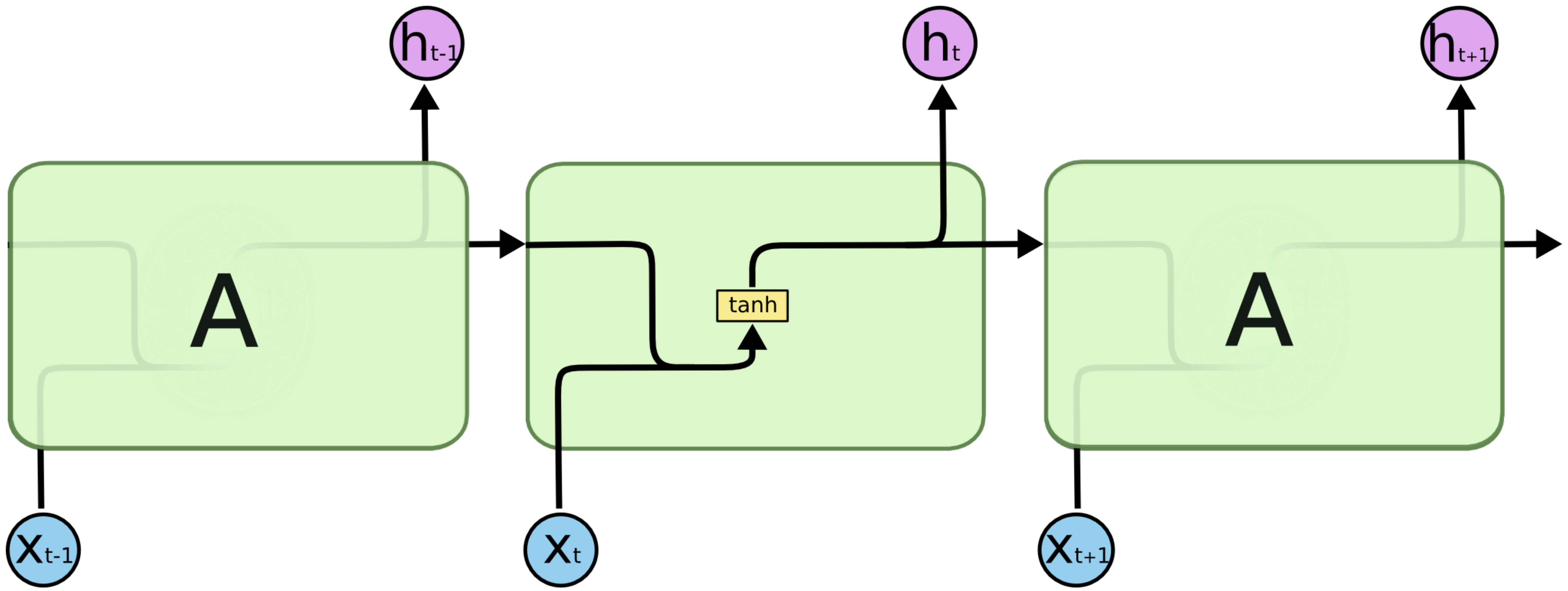


http://colah.github.io/posts/2015-08-Understanding-LSTMs/

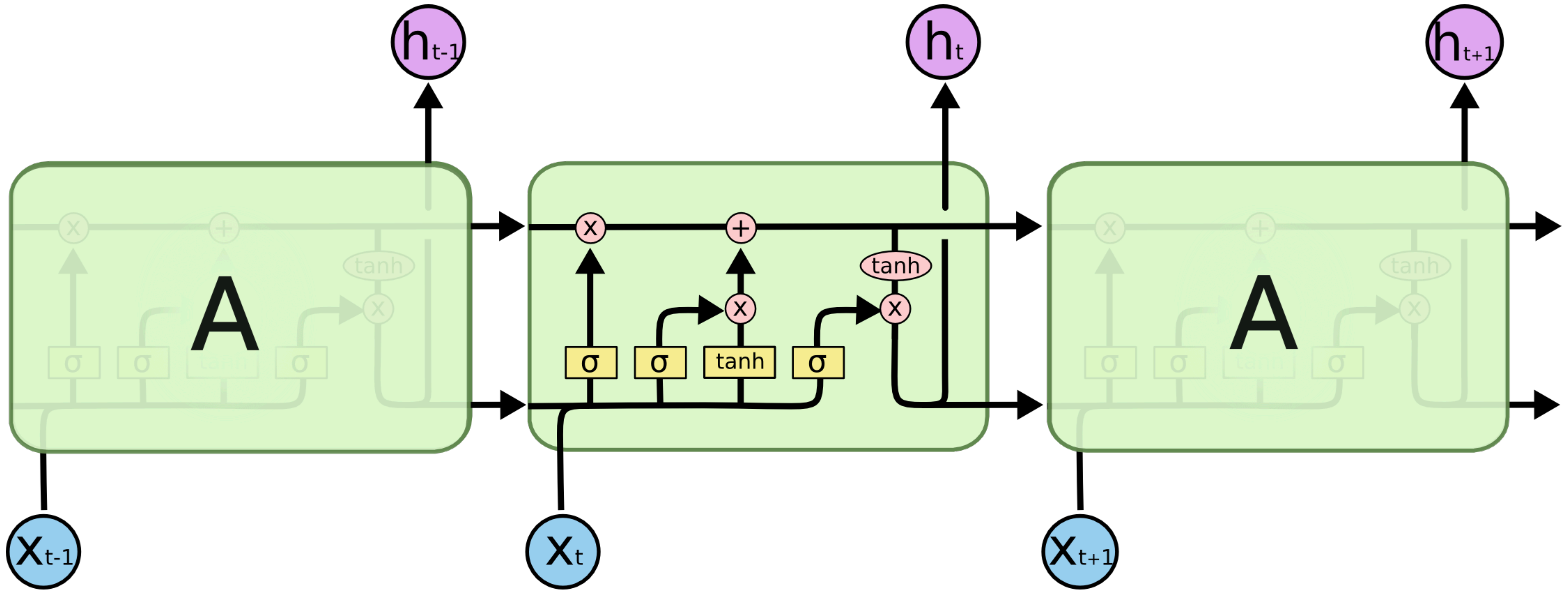
Modern language models extend this idea



Elman Network



Long Short Term Memory (LSTM)



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Connectionism as a framework for learning

1. Complex behavior can arise from the interaction of simple individual units (emergence)

2. Rule-like behavior need not be controlled by rule-like representations (sub-symbolic computation)

3. You don’t need to know the structure of a domain priori to model learning in that domain. (Domain-general)
   1. Although it can help with deciding network architecture

4. The same kind of learning model can work for explicit supervision and prediction

Limits to connectionism

1. Very data-hungry

2. Very computation-intensive

4. No built-in distinctions between correlation and causation

5. Hard to learn abstractions (or approximate them)

6. Models are large and hard to understand

VOTING RESOURCES

Thanks to Anna Fisher

VOTER REGISTRATION

Register! https://turbovote.org/

Check your voter registration!

https://www.vote.org/am-i-registered-to-vote/

CAN YOU VOTE IN PA?

College students can register to vote in Pennsylvania after living here for 30 days.

Students can give either an on-campus or an off-campus Pennsylvania address.

Students can use a school photo ID as verification.

MAKE A PLAN

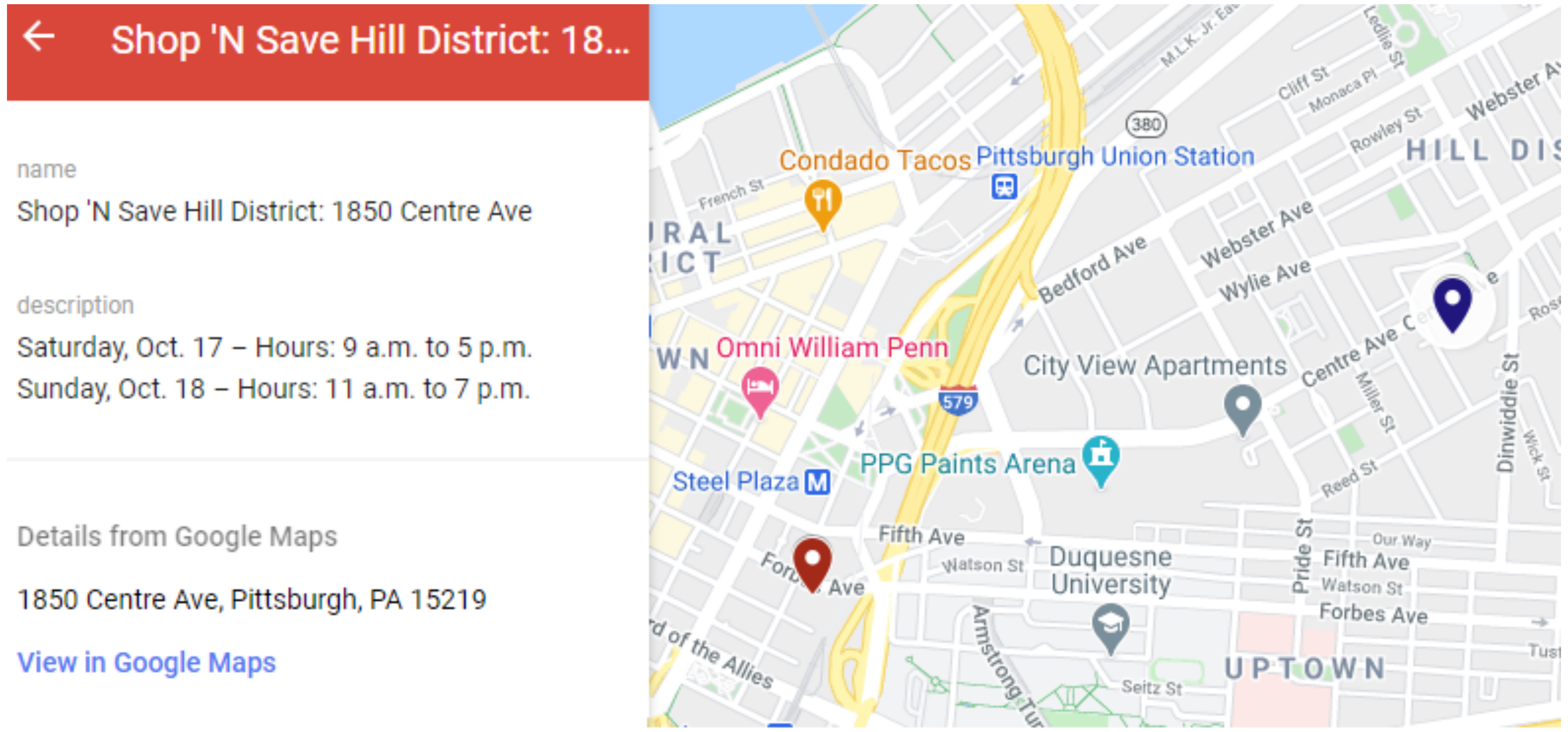
Make a voting plan ahead of time. Ask me for help if you have questions.

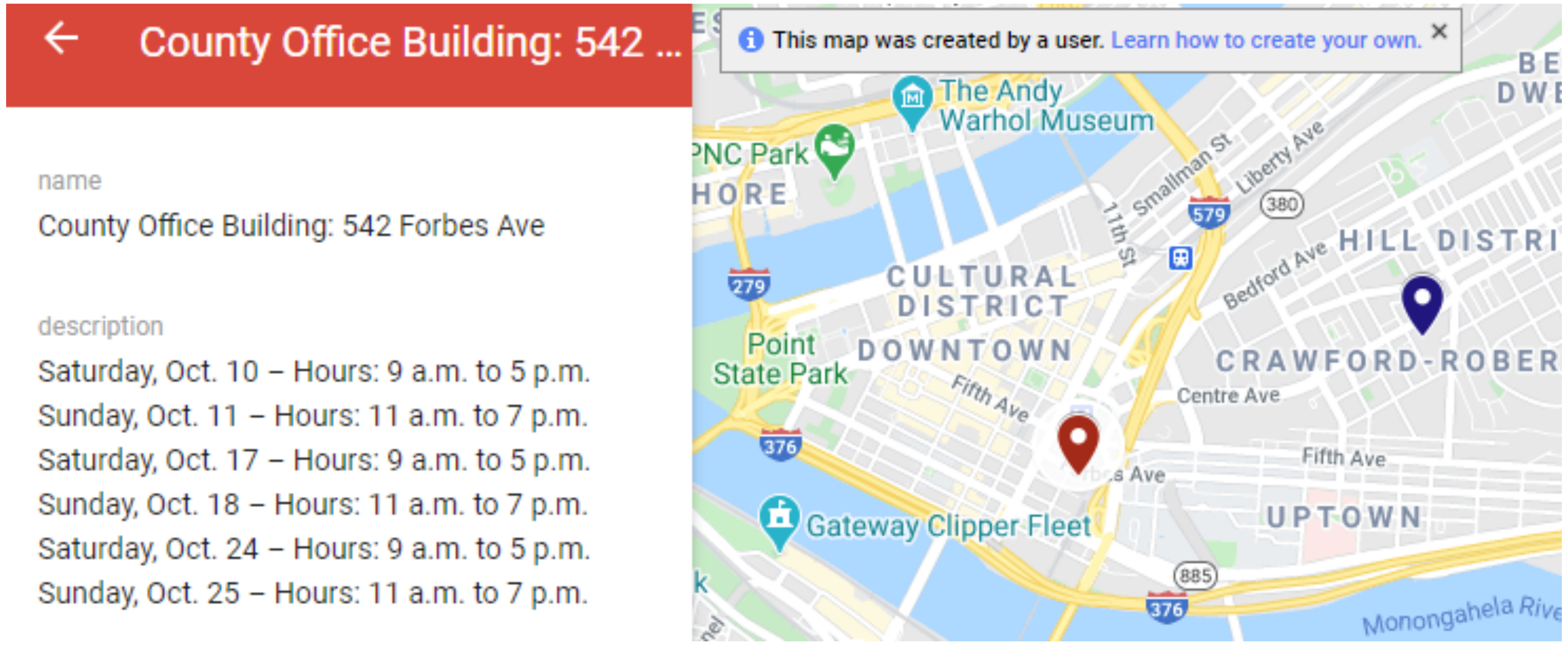
If you are voting by mail, make sure to request your ballot asap! https://www.vote.org/absentee-ballot/

NEW THIS YEAR IN PA: EARLY VOTING!

If you have NOT requested a mail ballot, you can vote early at these satellite locations!

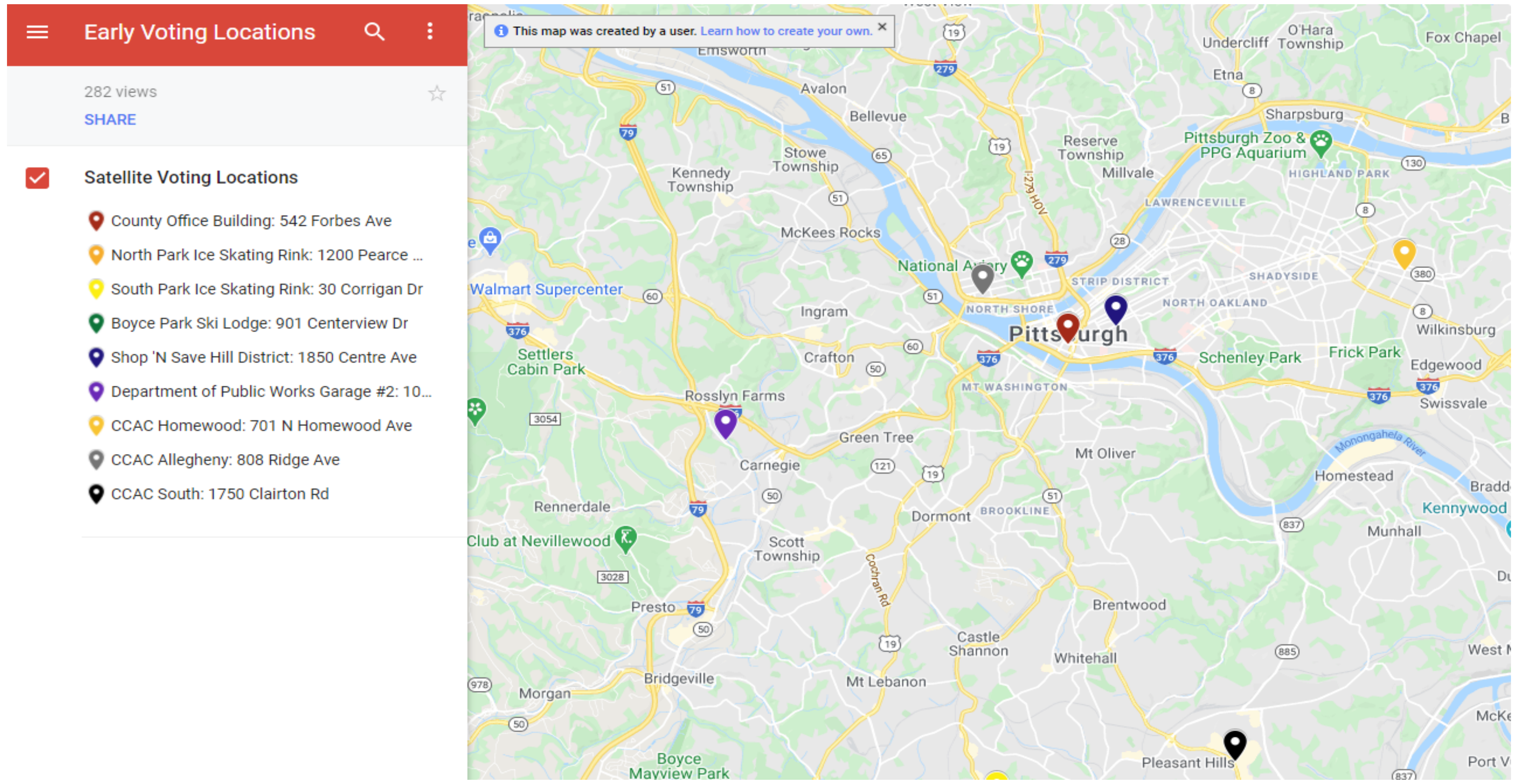
If you HAVE requested your mail ballot, you can drop it off at these locations!





NEW THIS YEAR IN PA: EARLY VOTING!

You can see all of the satellite locations on this great map! https://tinyurl.com/yymhqsph



<u>Saturday, October 10 – Hours: 9 AM to 5 PM</u>

1. County Office Building
2. North Park Ice Rink
3. South Park Ice Rink
4. DPW Garage # 2 (Carnegie)
5. CCAC Homewood

<u>Sunday, October 11 – Hours: 11 AM to 7 PM</u>

1. County Office Building
2. North Park Ice Rink
3. South Park Ice Rink
4. DPW Garage # 2 (Carnegie)
5. CCAC Homewood

<u>Saturday, October 17 – Hours: 9 AM to 5 PM</u>

1. County Office Building
2. South Park Ice Rink
3. CCAC South
4. Boyce Park Ski Lodge
5. Shop 'n Save Hill District

<u>Sunday, October 18 – Hours: 11 AM to 7 PM</u>

1. County Office Building
2. South Park Ice Rink
3. CCAC South
4. Boyce Park Ski Lodge
5. Shop 'n Save Hill District

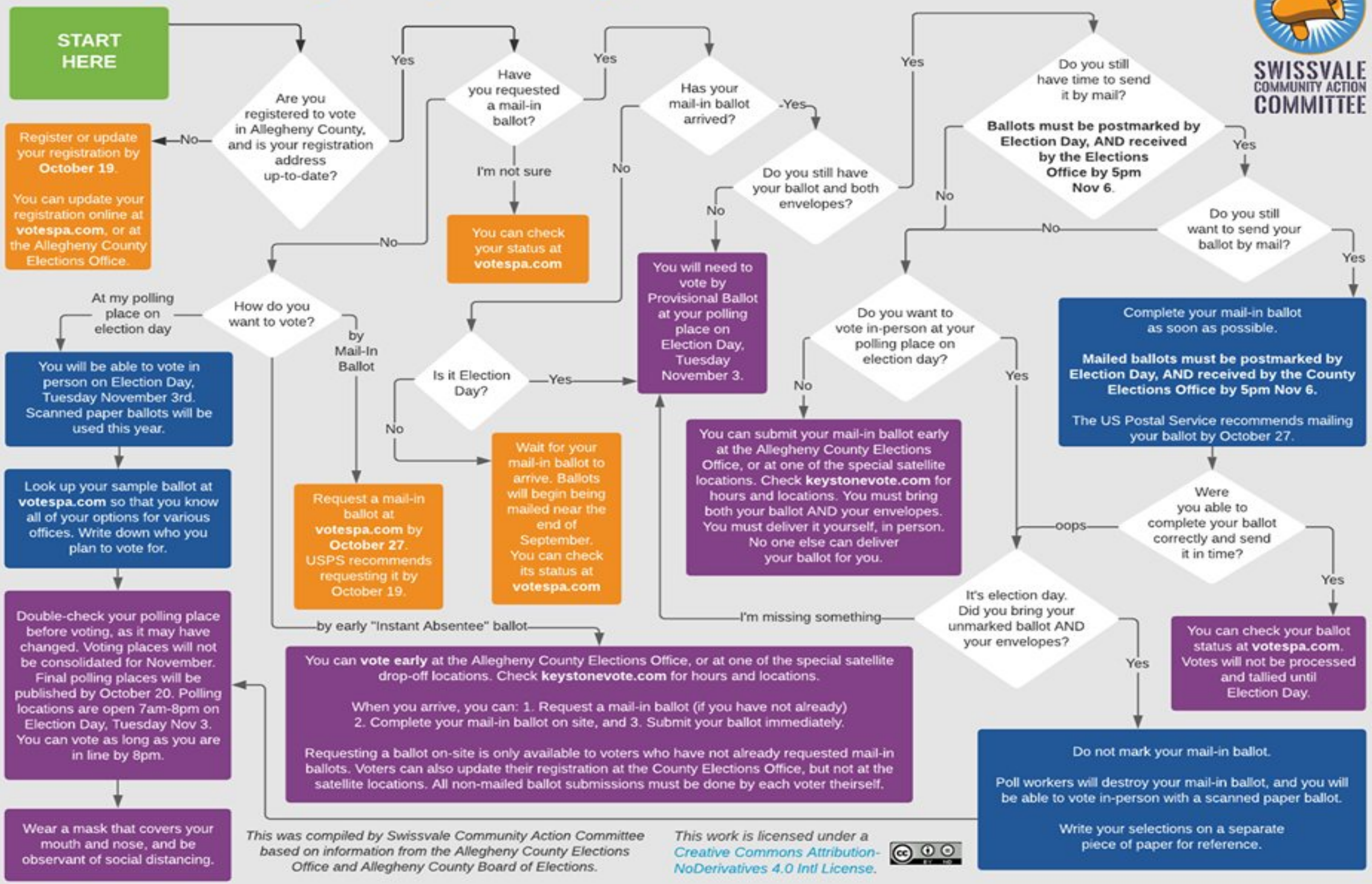
<u>Saturday, October 24 – Hours: 9 AM to 5 PM</u>

1. County Office Building
2. Boyce Park Ski Lodge
3. CCAC South
4. CCAC Allegheny
5. North Park Ice Rink

<u>Sunday, October 25 – Hours: 11 AM to 7 PM</u>

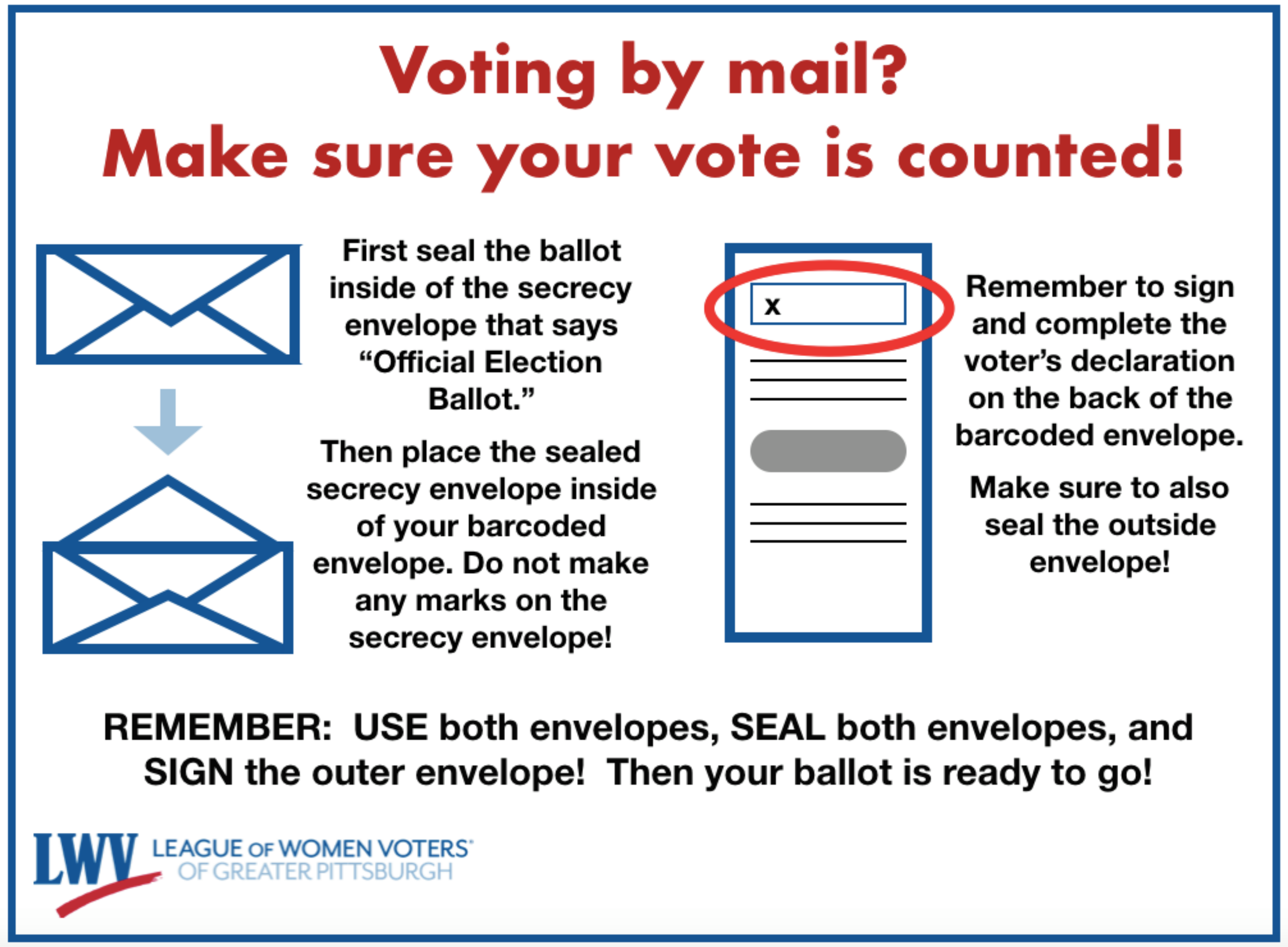
1. County Office Building
2. Boyce Park Ski Lodge
3. CCAC South
4. CCAC Allegheny
5. North Park Ice Rink

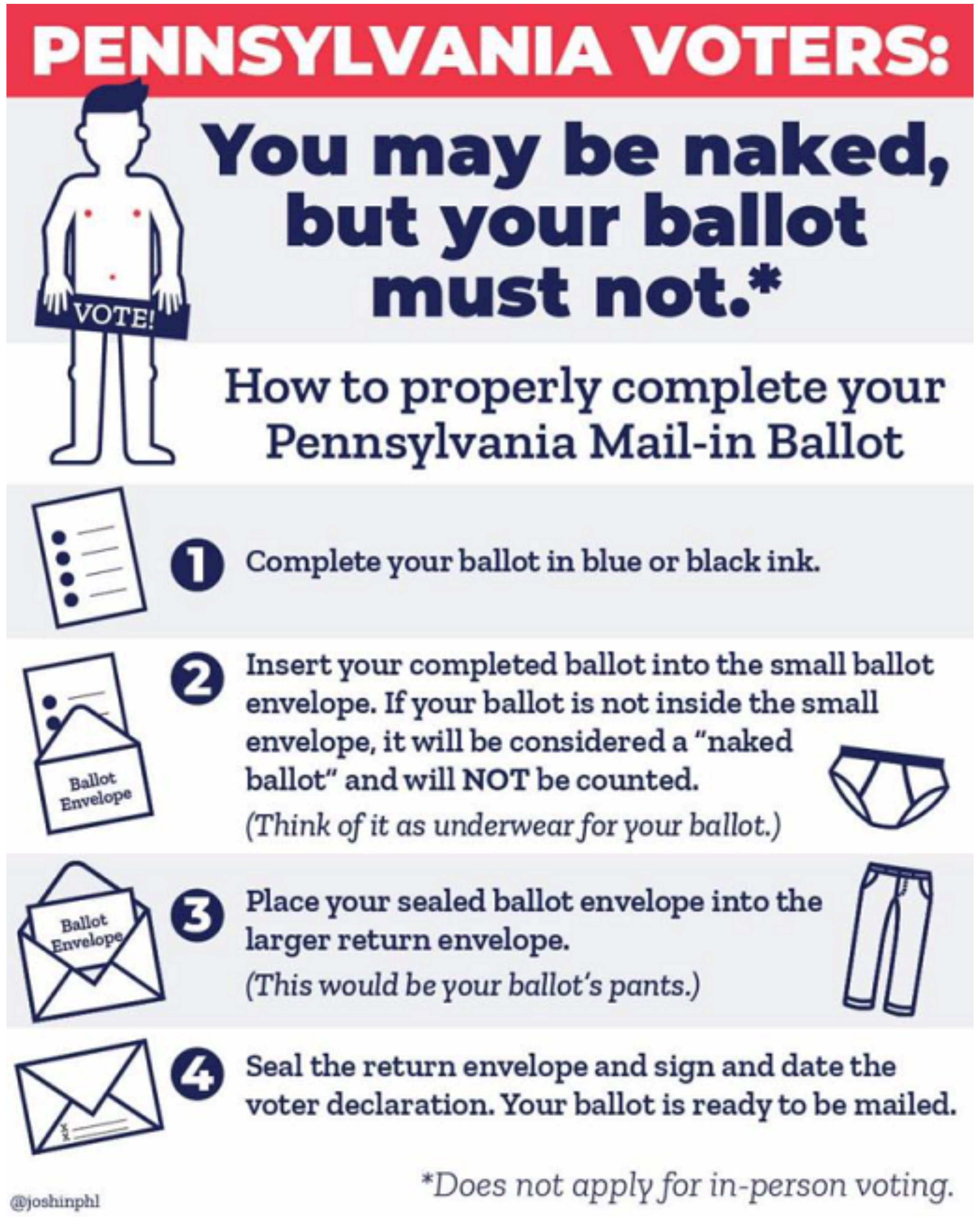
Voting in Allegheny County, November 2020



VOTING BY MAIL IN PA!

Watch this 1-min non-partisan video on how to complete your mail ballot correctly! https://www.youtube.com/watch?v=krhGbx7fA4o&t=0s





QUESTIONS?

Get your questions about voting in PA answered here: https://tinyurl.com/yyyzuvda

Need Immediate Support?

Call our Voter Assistance Hotline:

833-PAVOTES (833-728-6837)

Recurrent networks

1. Recurrent neural networks can discover structure in time

2. Connectionism recap

3. Voting resources

4. Homework 2