

## Tutorial 2

# Learning the English Past Tense

1. Download the archive `PastTense.zip` to your Tlearn folder and unzip. it.
2. In this simulation, the network's task will be to map a set of verb stems with their past tense forms. There are four types of verbs: Regulars (with three *-ed* allomorphs, see lecture slides), arbitrary irregulars, no-change verbs and vowel-change verbs (no blends).

The verbs belong to an artificial language which conforms to the phonotactics of English. All of these “verbs” are three phonemes long, each phoneme being represented by a six-bit vector based on phonological features. Past tense suffixes follow a different coding scheme using 2 bits each (see table on page 2).

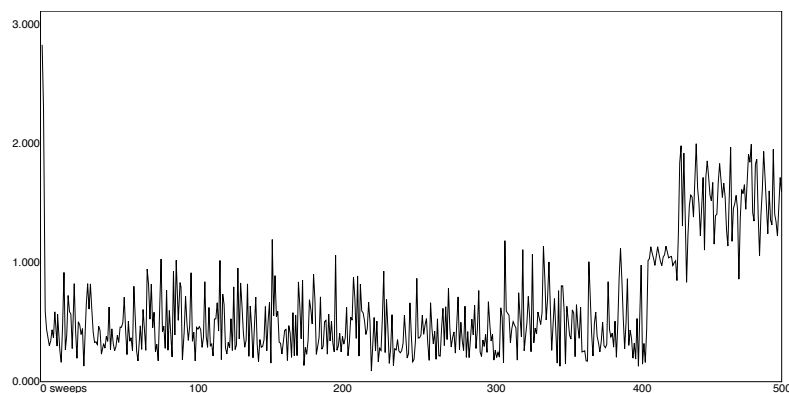
Examine the files `stems` and `past`s in the subdirectory `Verbs`. They contain a list of 500 verbs and their past tense: Two arbitrary verbs, 410 regulars, 20 no change, and 68 vowel change verbs, in that order.

3. These files have been converted to Tlearn vector input for you. The result is in `Network1`, in the files `pt.data` and `pt.teach`. The network architecture file is also provided, which contains 18 input nodes (3×6bits), 20 hidden units and 20 output nodes (18+2 bits for the suffix).

Open the project and train the network for 50 epochs, learning rate 0.3, momentum 0.75, randomly.

4. Test the network on the training data, sending the output to a file to save it for further analysis, e.g. `output_activations`. Check the box to calculate the error and open the error display.

It should contain four clearly identifiable sections corresponding to the four types of verbs in the simulation. Label the graph below to show which section corresponds to which type of verb:



	Phoneme	Ascii	Example	Cons./Vowel	Rounding/Voicing	Place	Closure/Manner	
Vowels (+ Semivowels)	/i:/	i	eat	1	0	0	1	1
	/u:/	u	boot	1	1	1	0	1
	/ɪ/	ɪ	bit	1	1	0	1	1
	/ʊ/	U	book	1	0	1	0	1
	/ɛɪ/	e	bait	1	0	0	1	0
	/ɛ/	E	bet	1	1	0	1	0
	/ʌ/	^	but	1	0	0	0	0
	/oʊ/	o	boat	1	0	1	0	0
	/ɔ:/	O	bought	1	1	1	0	0
	/æ/	a	bat	1	0	0	1	0
	/aɪ/	Y	bite	1	0	0	0	0
	/aʊ/	W	cow	1	1	0	0	0
	/w/	w		1	1	1	1	1
	/j/	j	yes	1	0	0	0	1
Consonants	/b/	b		0	1	1	0	0
	/p/	p		0	0	1	0	0
	/m/	m		0	1	1	0	1
	/f/	f		0	0	1	0	1
	/v/	v		0	1	1	0	1
	/θ/	T	thin	0	0	0	0	1
	/ð/	D	this	0	1	0	0	1
	/d/	d		0	0	0	0	0
	/t/	t		0	1	0	0	0
	/n/	n		0	1	0	0	1
	/l/	l		0	0	1	0	1
	/z/	z		0	1	0	1	1
	/s/	s		0	0	0	1	1
	/k/	k		0	0	0	1	0
	/g/	g		0	1	0	1	0
/ŋ/	N	sing	0	1	0	1	1	
/r/	r		0	0	0	0	1	
/h/	h		0	0	0	1	1	
Suffixes	/d/	*	blowed	0	1			
	/t/	+	wished	1	0			
	/d/	~	pitted	1	1			
	-	-	hit	0	0			

5. Open the file `output_activations`. As you can imagine, examining the output activations directly would be very tedious (20 real values for each of 500 verbs). However, we can convert those activations back to type of phonological representation we used to represent the original materials before converting to Tlearn vectors.

First, you must specify how vectors will be translated back to phonemes by using the **Output Translation** utility in the **Special** menu. Select the file prepared for this in the archive and choose “Euclidean distance” so that the closest phoneme corresponding to the real values in the to the output activations is chosen. Then test the network again, this time specifying that you want to use output translation, but leaving out the computation of the error. The result can be found in `Output1` (`output_translated`).

- a) How many training patterns does the network get right overall (number and percentage)?
  - b) How many regular verbs does it get right?
  - c) How many irregular verbs of each type?
  - d) Classify and then quantify the errors the network makes (overregularizations, irregularizations, blends, suffix errors, other), and report this evaluation, including any files to support your conclusions.
6. Generalization: What about new verbs on which the network has not been trained? The file `test` contains 211 new verbs. The result of testing the network on this file is contained in `test_translated`.
    - a) Analyze the original data, and make a prediction for each verb in the new data (what is the “correct” past tense form? Note there may be several possible answers for some verbs.)
    - b) How does the network perform on the new data given your predictions? Report your results with the accompanying data.
  7. As you noticed, although the network performed acceptably well on regular verbs, irregulars were not learned well. In order to improve the result, token frequency of the irregulars can be increased. This is justified by the high frequency of irregular verbs in child (directed) speech. In the archive, the files `stems2` and `pasts2` contain the same verbs, but now the arbitrary verbs occur 20 times and all other irregulars 10 times. Train the network again. How does it improve compared to your previous results?