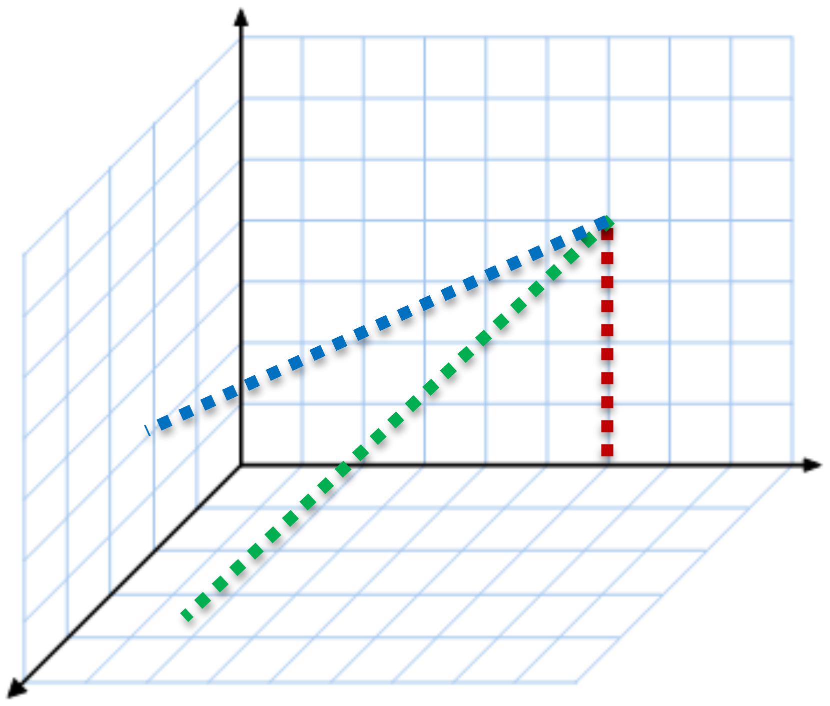
key concepts

what we need to know

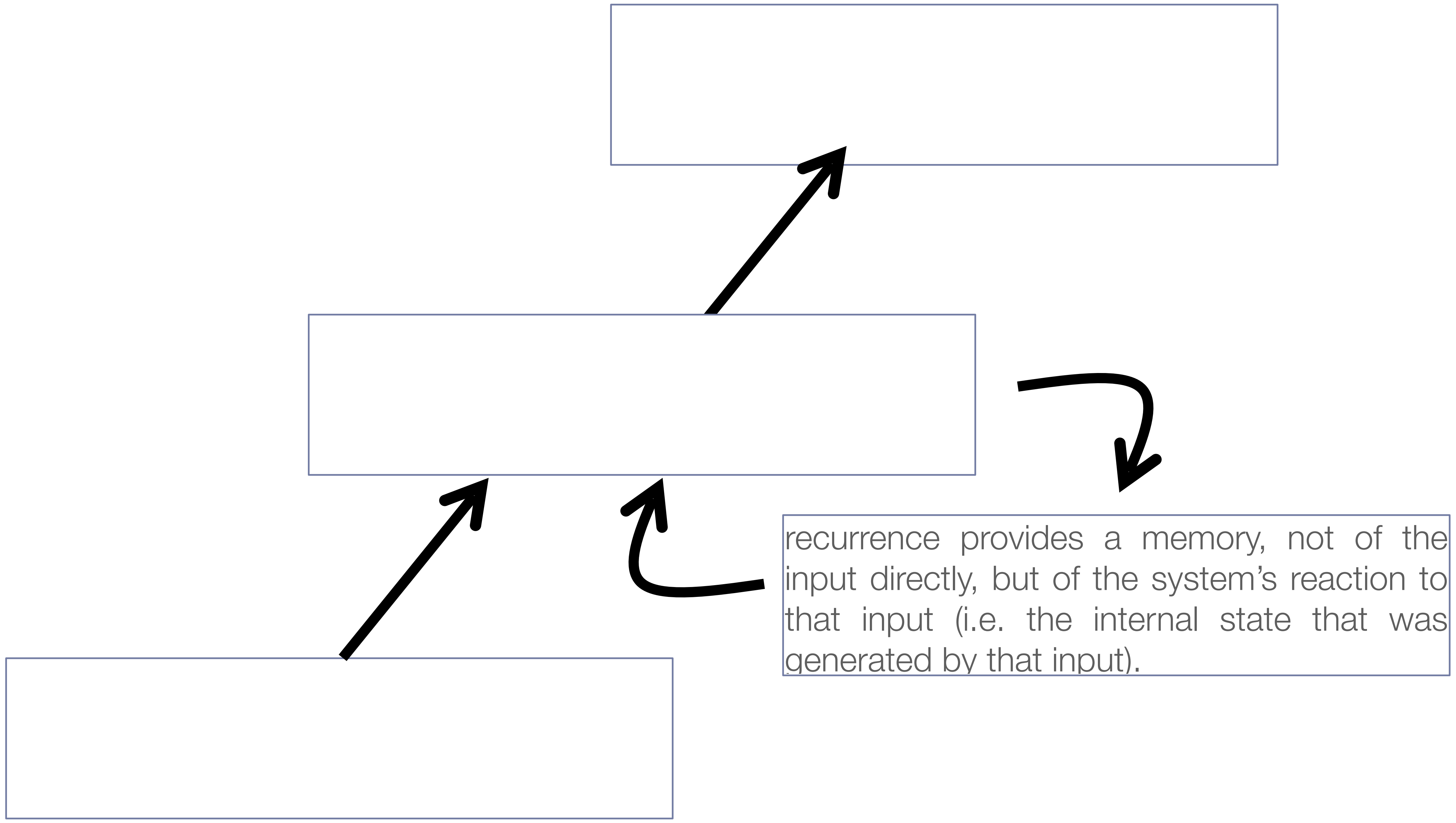
why we can take the leap of faith…

words as vectors

Context (Document)				
	#1	#2	#3	coordinate
CAT	✔			1, 0, 0
DOPAMINE			✔	0, 0, 1
DOG	✔	✔		1, 1, 0
REWARD		✔	✔	0, 1, 1



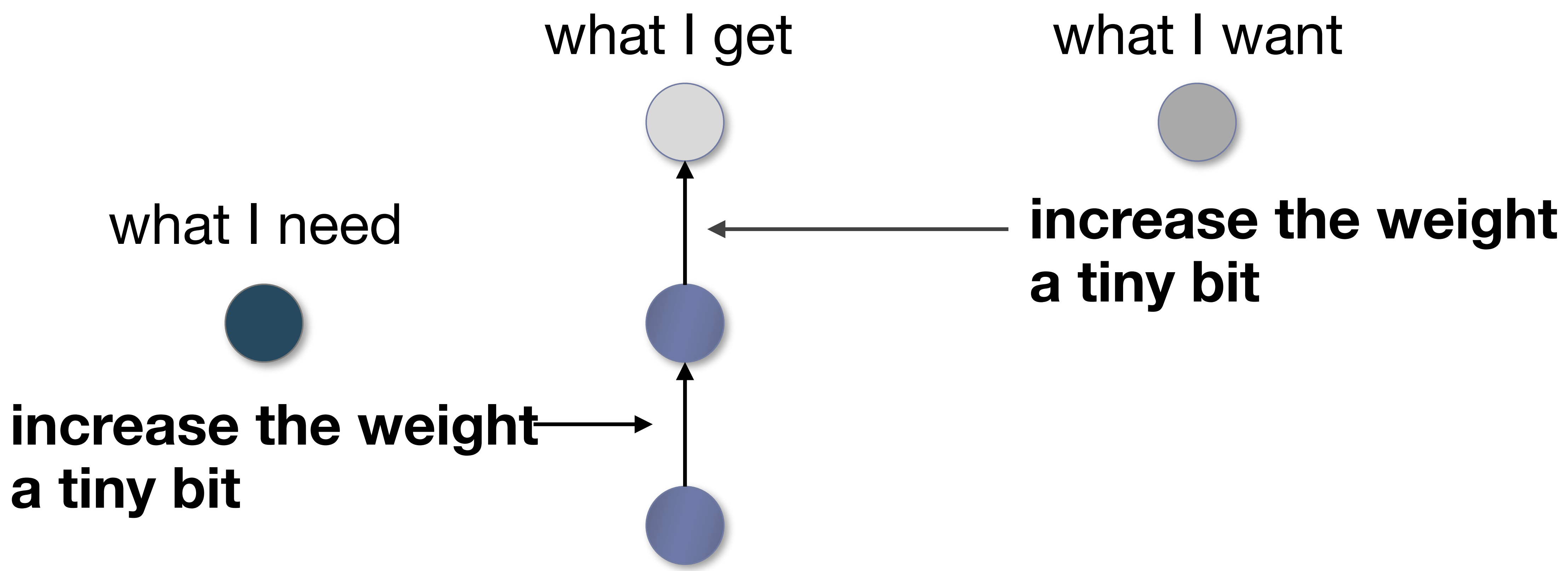
SRNs



recurrence provides a memory, not of the input directly, but of the system's reaction to that input (i.e. the internal state that was generated by that input).

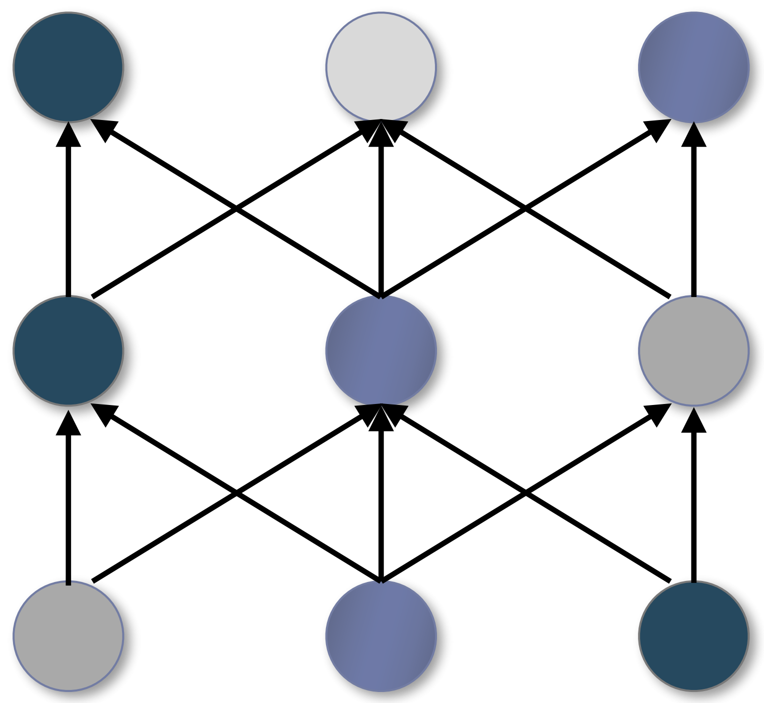
adjusting the weights

(back propagating the error)


what I get
what I want
what I need
increase the weight a tiny bit
increase the weight a tiny bit

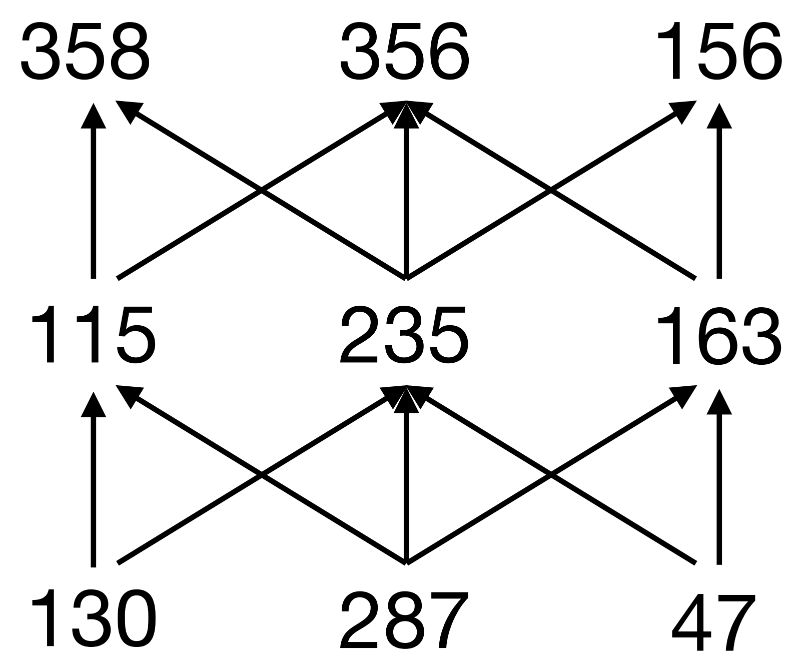
adjusting the weights

(back propagating the error)



adjusting the weights

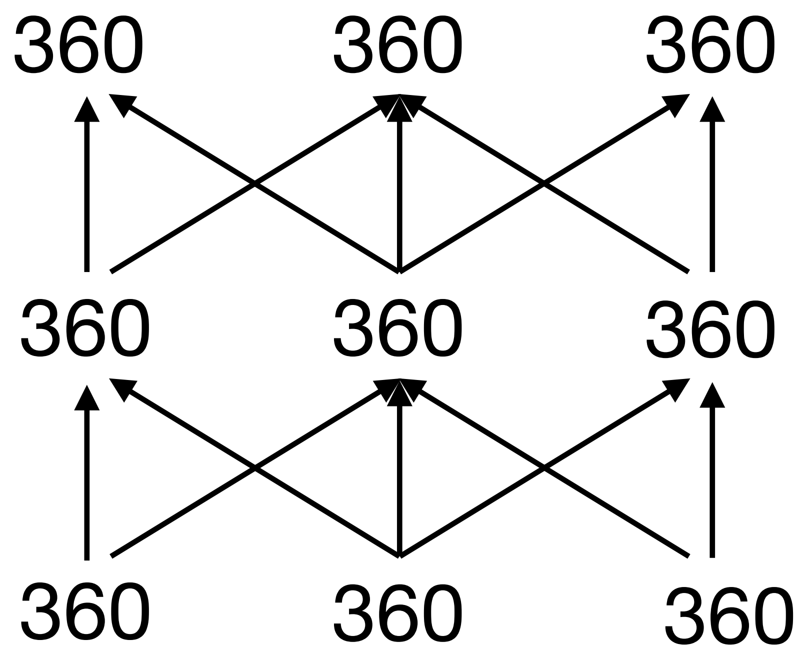
(back propagating the error)



Let’s suppose these are the initial activations based on random weights

adjusting the weights

(back propagating the error)



And this is what we want them to be

adjusting the weights

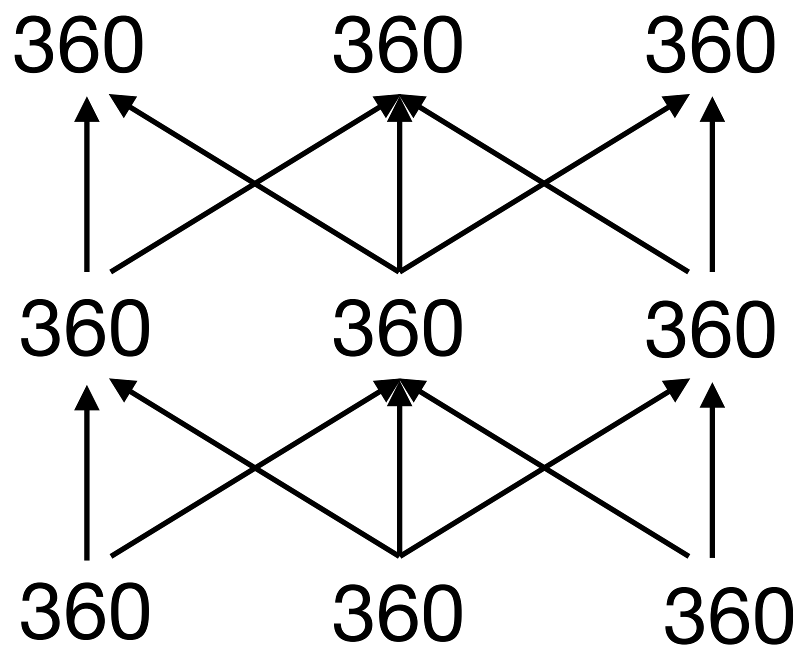
(back propagating the error)

$$\begin{tikzcd}
353 & 162 & 141 \\
307 \arrow[u] \arrow[ur] & 7 \arrow[ul] \arrow[u] \arrow[ur] & 294 \arrow[ul] \arrow[u] \\
31 \arrow[u] \arrow[ur] & 120 \arrow[ul] \arrow[u] \arrow[ur] & 286 \arrow[ul] \arrow[u]
\end{tikzcd}$$

So this is the difference between “what we get” and “what we want”

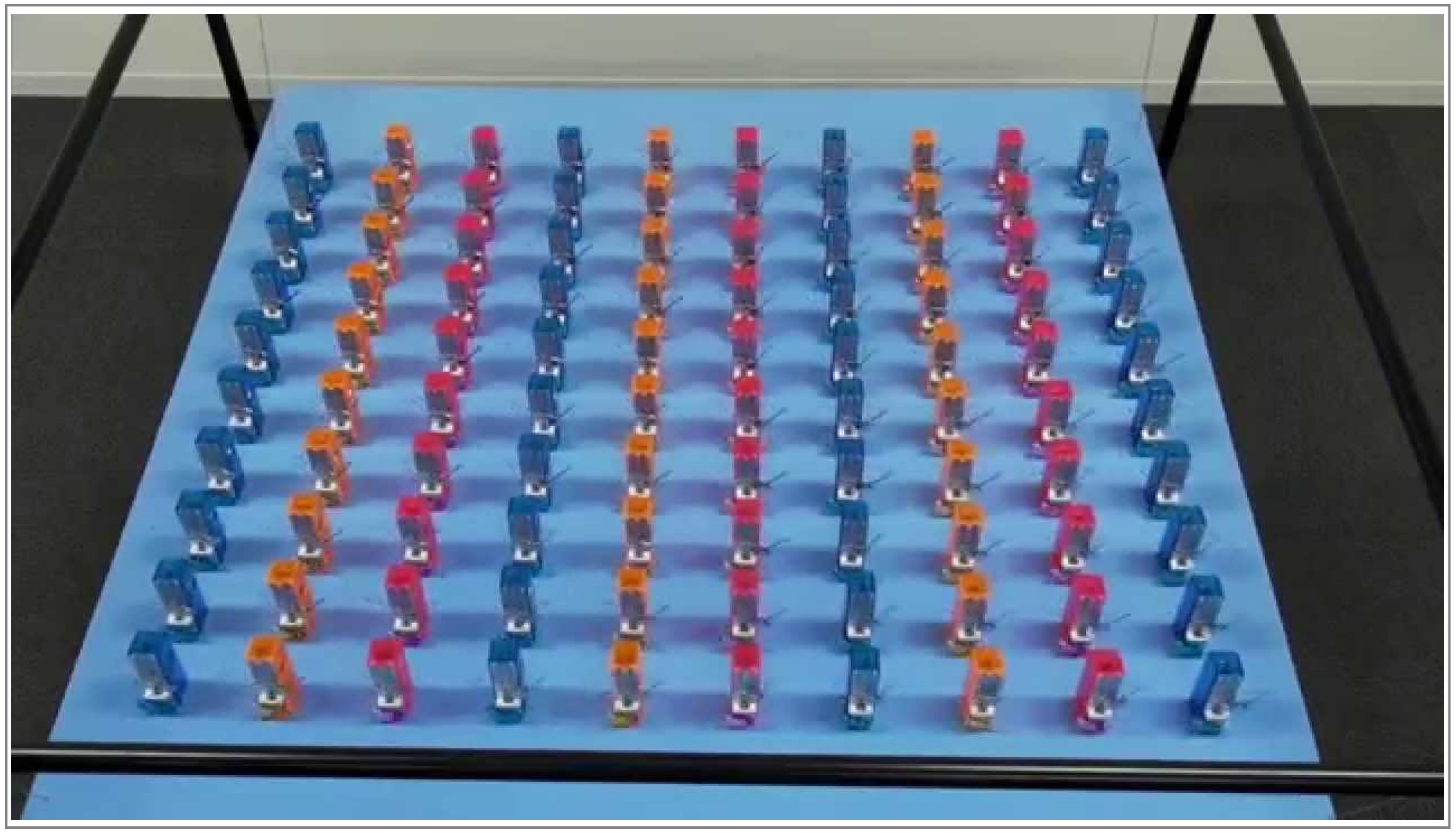
adjusting the weights

(back propagating the error)



So how do we change the weights to achieve this desired result? We make many tiny tiny changes to the weights (using backdrop) socthat each one gradually converges on the “correct” weight that gives the desired output.

How the heck can it learn what you want it to learn?



Those numbers in the example above were actually phase angles (how in or out of phase the waves were to the desired 360, which would mean they were perfectly in phase with some target sinusoid). The red numbers were equivalent to the different phase angles of each metronome in the above movie. Small vibrations travel from each metronome to each other (interacting with others along the way), that cause slight perturbations to the phase of each metronome, resulting in slight changes in phase such that eventually, they all converge on being in phase. Essentially, a natural version of what one wants backdrop to achieve, sort of…!

word embeddings

- generate a semantic space
- enable analogical reasoning (to an extent)
- useful for
  - translation
  - text summarizing
  - question answering (more of that later)
  - sentiment analysis
  - generating sentences within a part of the semantic space
- pre-trained…

why RNNs don’t scale well

This is a story about an alligator

it was walking through the jungle

when suddenly an elephant

alligator

RNNs - and what GRUs do

GRUs (and LSTMs) amplify the context when it’s useful to do that, and forget the context when it’s useful to do *that*.

but amplification, both positively and negatively, is just a weighting.

the weighting is learned (just through backdrop, or near enough…)

Transformers (principles, not practice)

- recurrence enables memory for ***context***.
  - cheat - instead of this input use this input:

the	*the*
boy	*the boy*
ate	*the boy ate*

- recurrence enables positional information
  - cheat - add a positional vector (akin to serial position) to each input (word) vector. And now you can process in parallel…

Attention

- context is important because words mean (subtly) different things in different contexts.
  - *catch fish by the river **bank***
  - *put your money in the **bank***

- (1) the more “bank” resembles “fish” or “river” the more you shift it in their direction within semantic space. Similarly for “bank” and “money”. The less similar, the less you shift it.

Attention

- context is important because words mean (subtly) different things in different contexts.
  - *catch fish by the river* ***bank***
  - *put your money in the* ***bank***

- (2) do this for all words in the sentence…<u>and</u> in the context. Do it in parallel! Instead of vector-by-vector, use matrices (one row per input word/token - multiply the input matrix by a learned weight matrix). Dot product (matrix multiplication) = similarity!

Attention

- context is important because words mean (subtly) different things in different contexts.
  - *catch fish by the river **bank***
  - *put your money in the **bank***

- (3) some words are more influential than others, and some words need more context than others, so multiply the similarity score by a weight that is learned through tons of training, and use multiple **attention** layers (*multi-head attention*) that will learn to attend to different cues.

Attention

- context is important because words mean (subtly) different things in different contexts.
  - *catch fish by the river **bank***
  - *put your money in the **bank***

- to summarize **attention**:

  Add the vectors in the context to the current vector in proportion to their similarity. The output is a “weighted blend” of the vectors in the context. Different “attention layers” simply learn different proportions, to up- or down-weight different influences.

Transformers (principles, not practice)

- There are different kinds of transformers. They all use some combination of “encoders” (generating word embeddings) and/or “decoders” (predicting the next word).
- Size matters - 1 million “tokens” in the context = 700,000 words
  - predictions reflects entire future trajectories through semantic space, not just a single word.

BERT (bidirectional encoder representations from transformers)

- Trained on only 3.3 billion words..! (ChatGPT trained on 300+ billion)
- Trained simultaneously on two tasks
    - takes an entire sequence, masks 15% of those words, and then has to guess them.
        - it doesn't backdrop from all the words in the sentence, only the masked words.
    - has to classify (Y/N) two-sentence sequences in respect of whether the second sentence plausibly follows the first
- It uses attention.
- it's just an encoder architecture (i.e. generates the word embeddings, rather than predicting an unfolding sentence)
- fine-tuning is used to train it to e.g. classify sentiments, or answer questions (it's given a text, a question about the text, and has to learn to mark where in the text the beginning and end of the answer is)