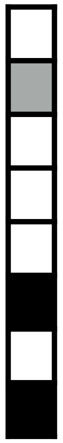
Modeling with Machine Learning: RNN (part 2)

Recall: learning to encode/decode

- Language modeling

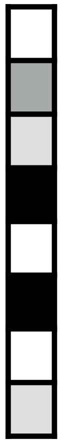
This course has been a



success (?)

- Sentiment classification

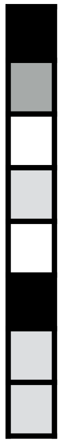
I have seen better lectures



-1

- Machine translation

I have seen better lectures



Olen nähnyt parempia luentoja

encoding **decoding**

Outline (part 2)

- Modeling sequences: language models
  - Markov models
  - as neural networks
  - hidden state, Recurrent Neural Networks (RNNs)
- Example: decoding images into sentences

Markov Models

- Next word in a sentence depends on previous symbols already written (history = one, two, or more words)

The lecture leaves me bumfuzzled

- Similar, next character in a word depends on previous characters already written

bumfuzzled

- We can model such kth order dependences between symbols with Markov Models

Markov Language Models

- Let $w \in V$ denote the set of possible words/symbols that includes
  - an UNK symbol for any unknown word (out of vocabulary)
  - <beg> symbol for specifying the start of a sentence
  - <end> symbol for specifying the end of the sentence

<beg> The lecture leaves me UNK <end>

w_0 w_1 w_2 w_3 w_4 w_5 w_6

- In a first order Markov model (bigram model), the next symbol only depends on the previous one

A first order Markov model

- Each symbol (except <beg>) in the sequence is predicted using the same conditional probability table until an <end> symbol is seen

w_i

		ML	course	is	UNK	<end>
	<beg>	0.7	0.1	0.1	0.1	0.0
	ML	0.1	0.5	0.2	0.1	0.1
w_{i-1}	course	0.0	0.0	0.7	0.1	0.2
	is	0.1	0.3	0.0	0.6	0.0
	UNK	0.1	0.2	0.2	0.3	0.2

Sampling from a Markov model

w_i

w_{i-1}	ML	course	is	UNK	<end>
<beg>	0.7	0.1	0.1	0.1	0.0
ML	0.1	0.5	0.2	0.1	0.1
course	0.0	0.0	0.7	0.1	0.2
is	0.1	0.3	0.0	0.6	0.0
UNK	0.1	0.2	0.2	0.3	0.2

Maximum likelihood estimation

- The goal is to maximize the probability that the model can generate all the observed sentences (corpus S)

$$s \in S, \quad s = \{w_1^s, w_2^s, \ldots, w_{|s|}^s\}$$

Maximum likelihood estimation

- The goal is to maximize the probability that the model can generate all the observed sentences (corpus S)

$$s \in S, \quad s = \{w_1^s, w_2^s, \ldots, w_{|s|}^s\}$$

- The ML estimate is obtained as normalized counts of successive word occurrences (matching statistics)

Feature based Markov Model

- We can also represent the Markov model as a feed-forward neural network (very extendable)

Feature based Markov Model

- We can also represent the Markov model as a feed-forward neural network (very extendable)

Temporal/sequence problems

- Language modeling: what comes next?

This course has been a tremendous …

tremendous $\begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix}$

?

a $\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

$x^{(t)}$ $y^{(t)}$

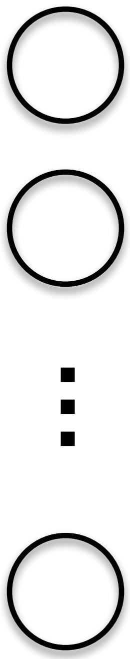
Temporal/sequence problems

- A trigram language model

tremendous $\begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix}$

a $\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

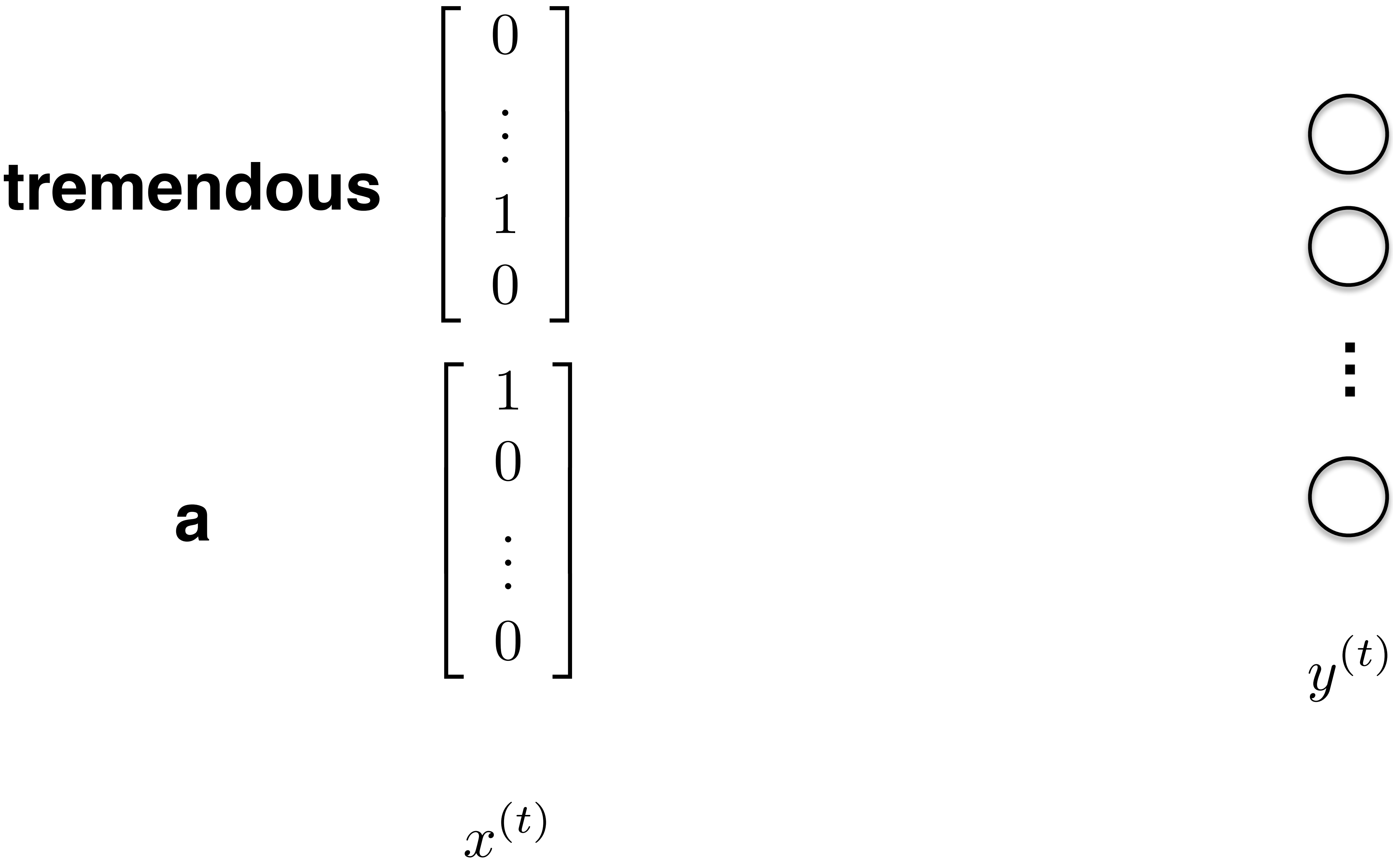
$x^{(t)}$



$y^{(t)}$

Temporal/sequence problems

- A trigram language model



tremendous

$$\begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix}$$

a

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$x^{(t)}$

$y^{(t)}$

RNNs for sequences

- Language modeling: what comes next?

This course has been a tremendous | …

tremendous $\begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix}$ **?**

RNNs for sequences

- Language modeling: what comes next?

This course has been a tremendous |...

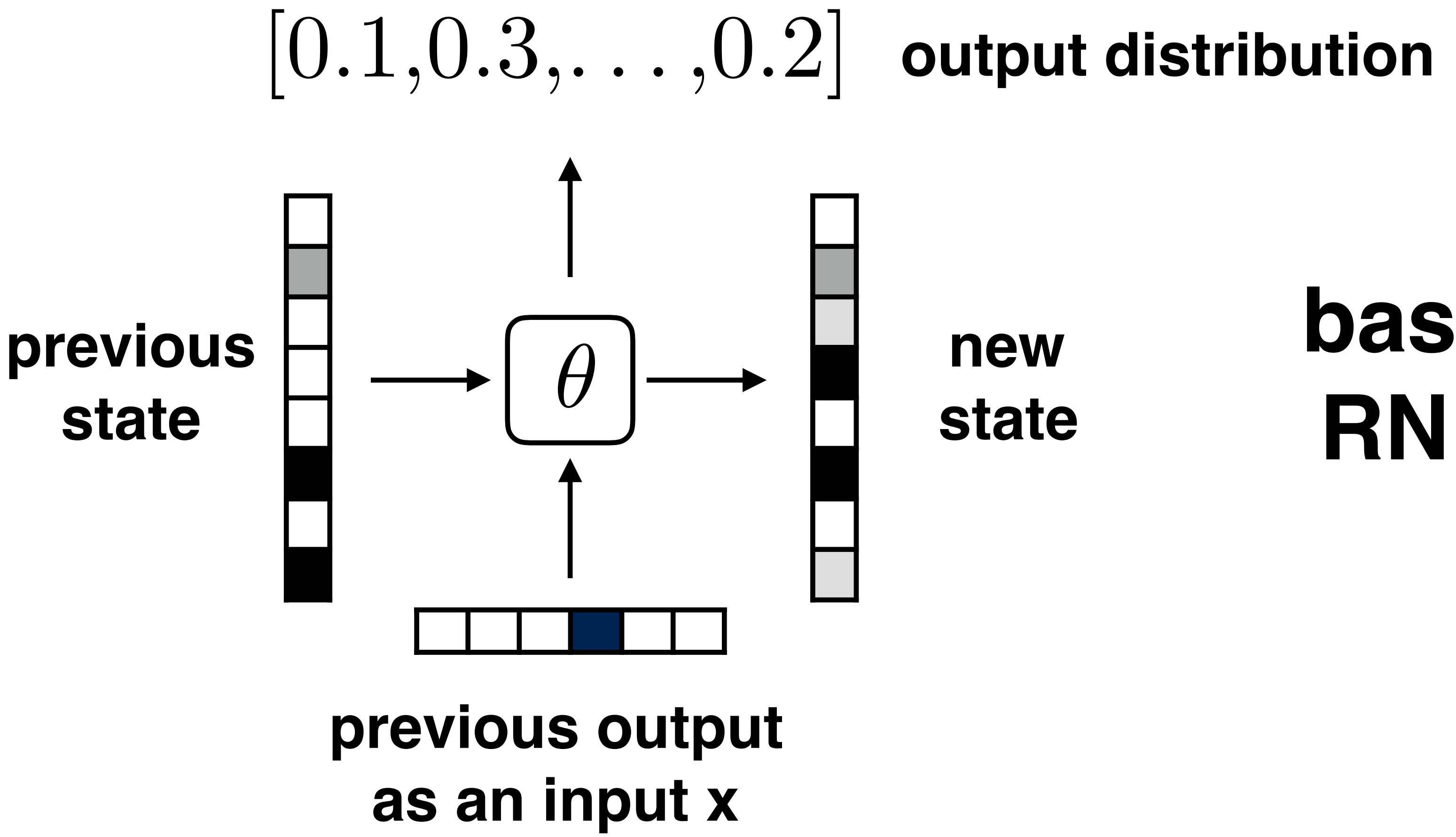
tremendous $\begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix}$ **?**

$$s_t = \tanh(W^{s,s} s_{t-1} + W^{s,x} x_t)$$ state

$$p_t = \mathrm{softmax}(W^o s_t)$$ output distribution

Decoding, RNNs

- Our RNN now also produces an output (e.g., a word) as well as update its state



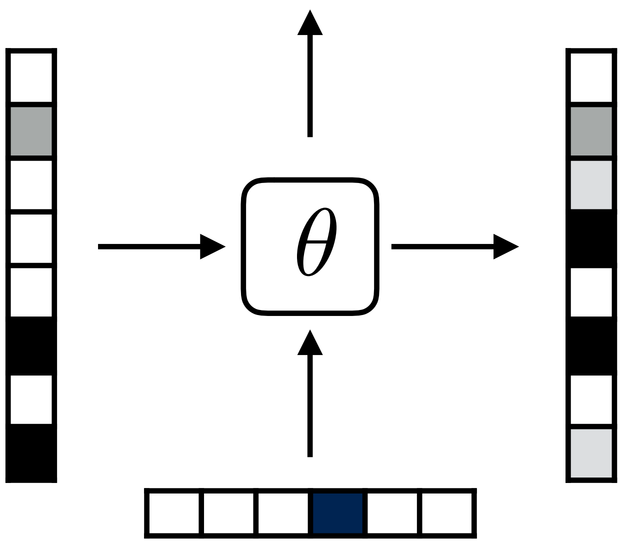
basic RNN

$$s_t = \tanh(W^{s,s} s_{t-1} + W^{s,x} x_t) \quad \text{state}$$

→ $$p_t = \text{softmax}(W^o s_t) \quad \text{output distribution}$$

Decoding, LSTM

$[0.1, 0.3, \ldots, 0.2]$ **output distribution**



previous state

new state

LSTM

previous output as an input x

$$f_t = \text{sigmoid}(W^{f,h} h_{t-1} + W^{f,x} x_t)$$ forget gate

$$i_t = \text{sigmoid}(W^{i,h} h_{t-1} + W^{i,x} x_t)$$ input gate

$$o_t = \text{sigmoid}(W^{o,h} h_{t-1} + W^{o,x} x_t)$$ output gate

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W^{c,h} h_{t-1} + W^{c,x} x_t)$$ memory cell

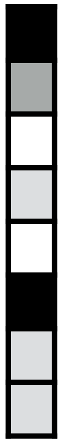
$$h_t = o_t \odot \tanh(c_t)$$ visible state

→ $$p_t = \text{softmax}(W^o h_t)$$ output distribution

Decoding (into a sentence)

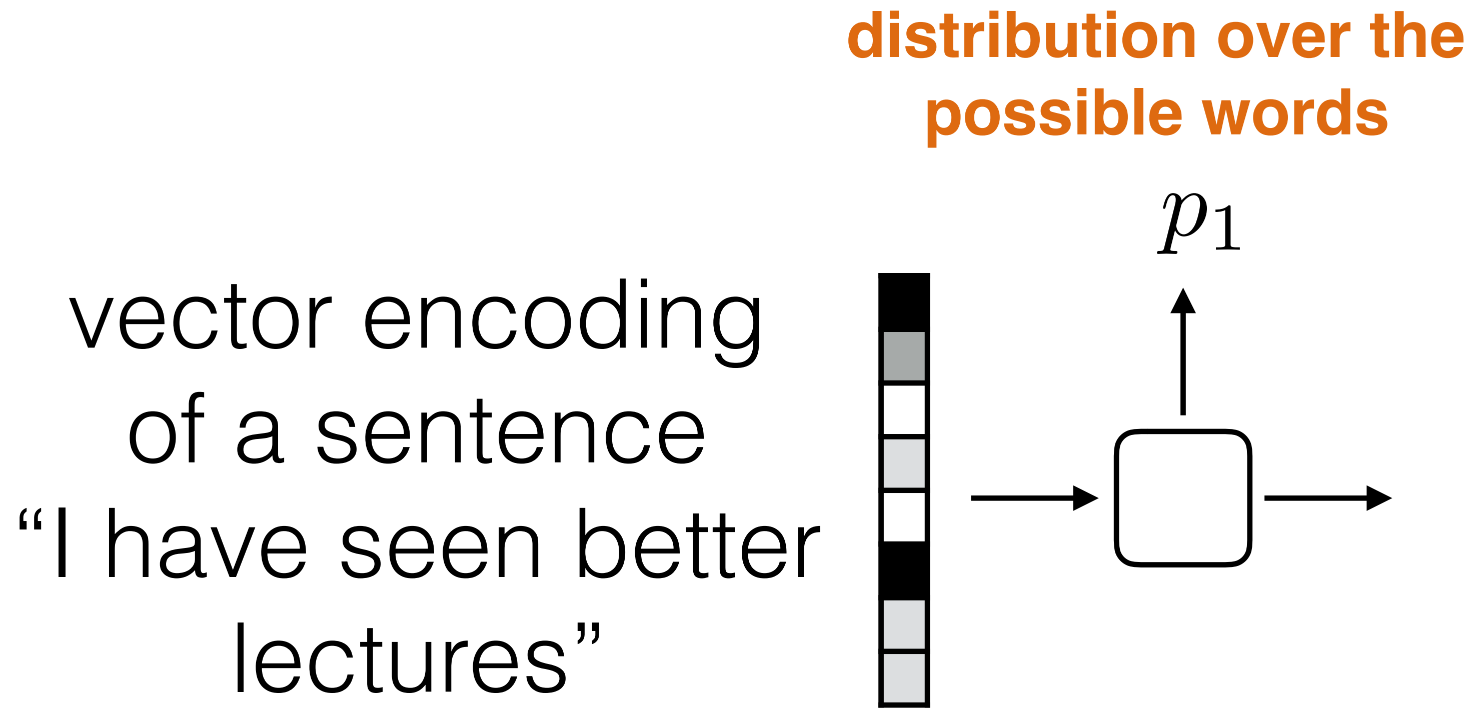
- Our RNN now needs to also produce an output (e.g., a word) as well as update its state

vector encoding
of a sentence
“I have seen better
lectures”



Decoding (into a sentence)

- Our RNN now needs to also produce an output (e.g., a word) as well as update its state

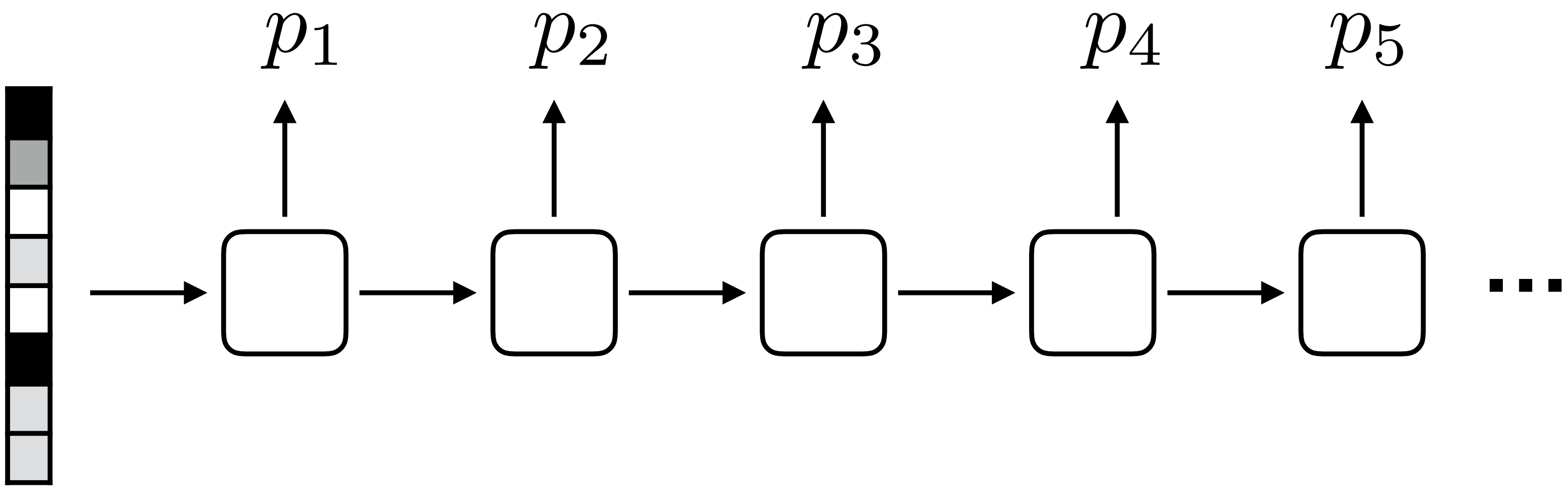

distribution over the possible words
p_1
vector encoding of a sentence "I have seen better lectures"

Decoding (into a sentence)

- Our RNN now needs to also produce an output (e.g., a word) as well as update its state

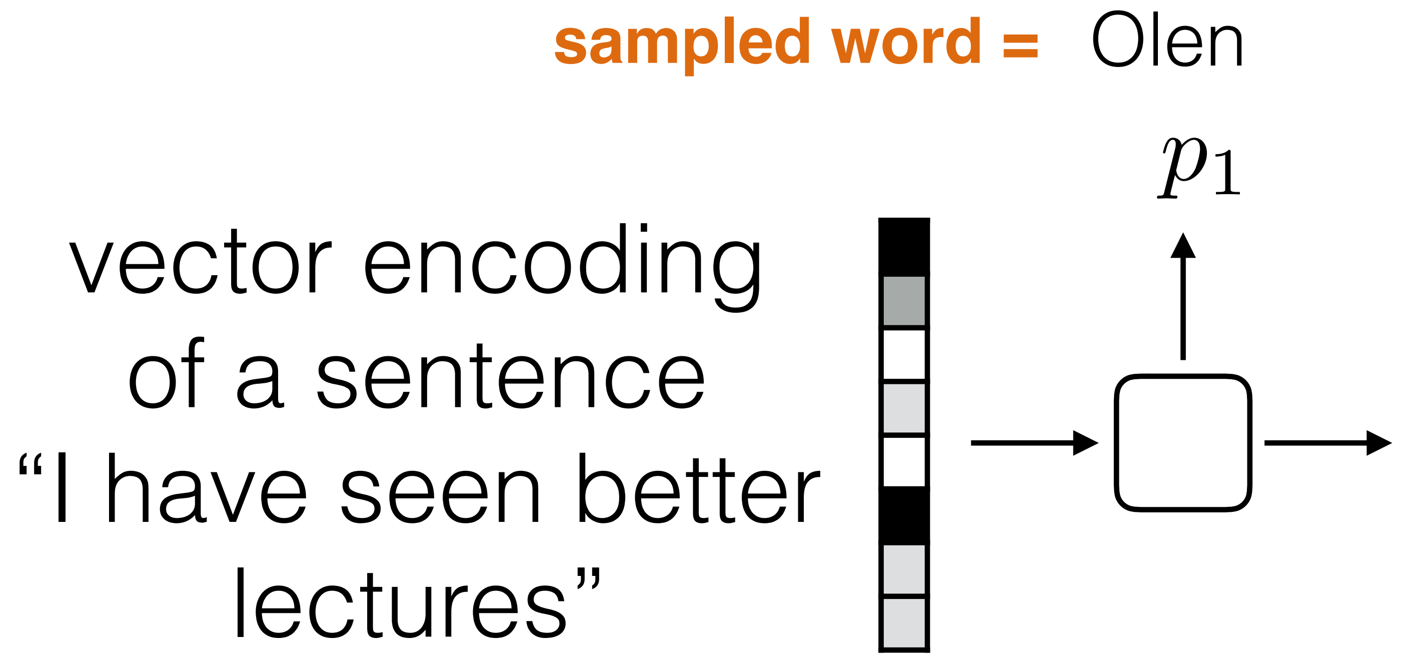
distribution over the possible words

vector encoding of a sentence “I have seen better lectures”



Decoding (into a sentence)

- Our RNN now needs to also produce an output (e.g., a word) as well as update its state



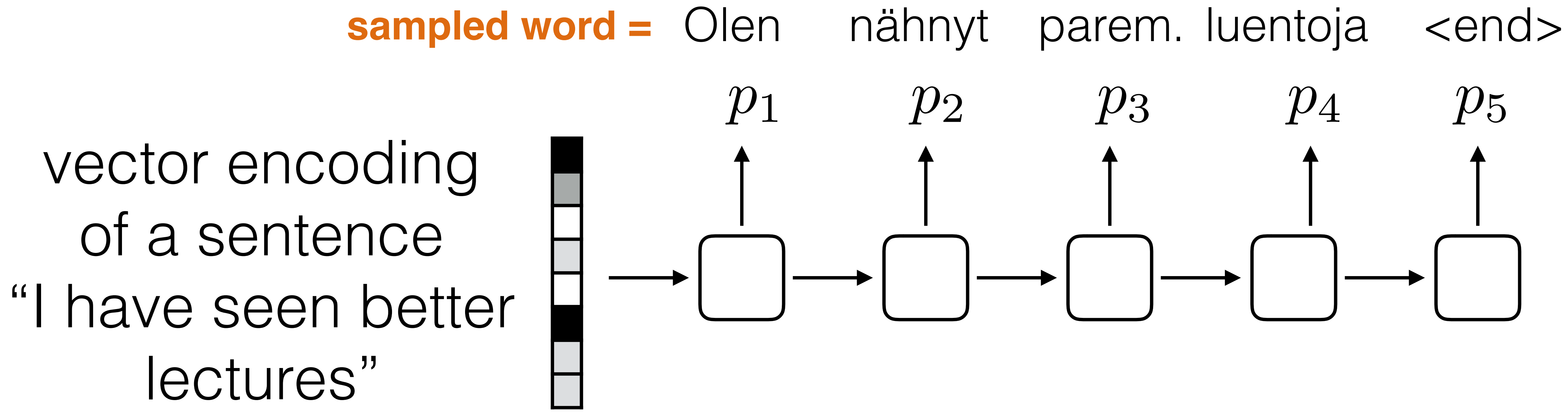
sampled word = Olen

p_1

vector encoding of a sentence "I have seen better lectures"

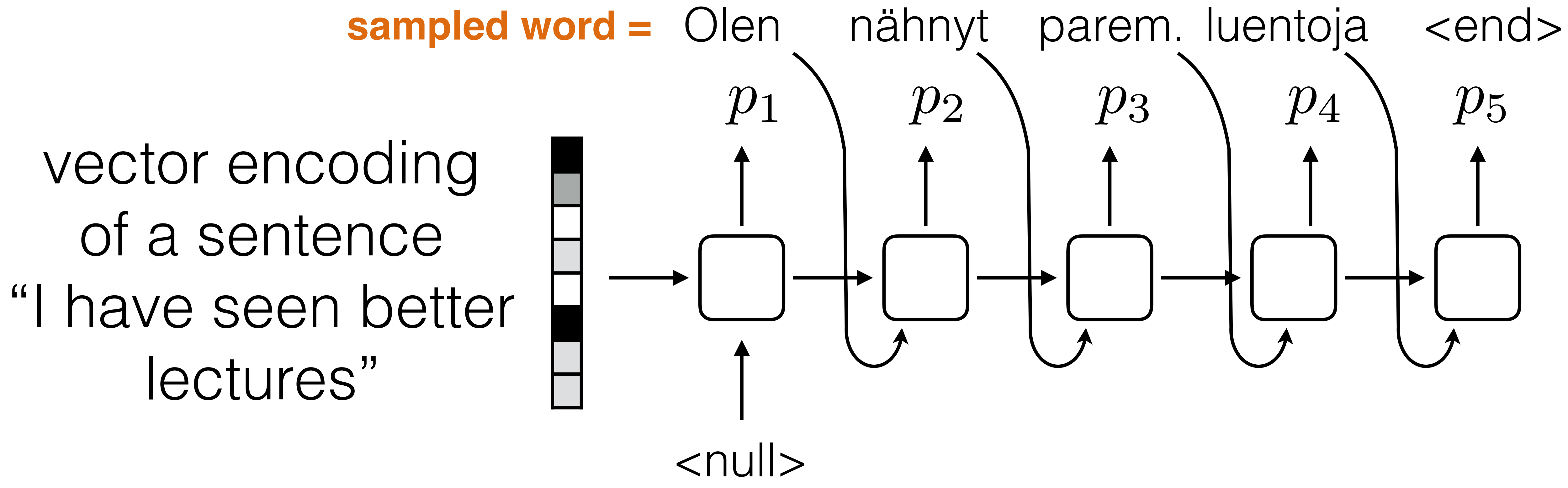
Decoding (into a sentence)

- Our RNN now needs to also produce an output (e.g., a word) as well as update its state

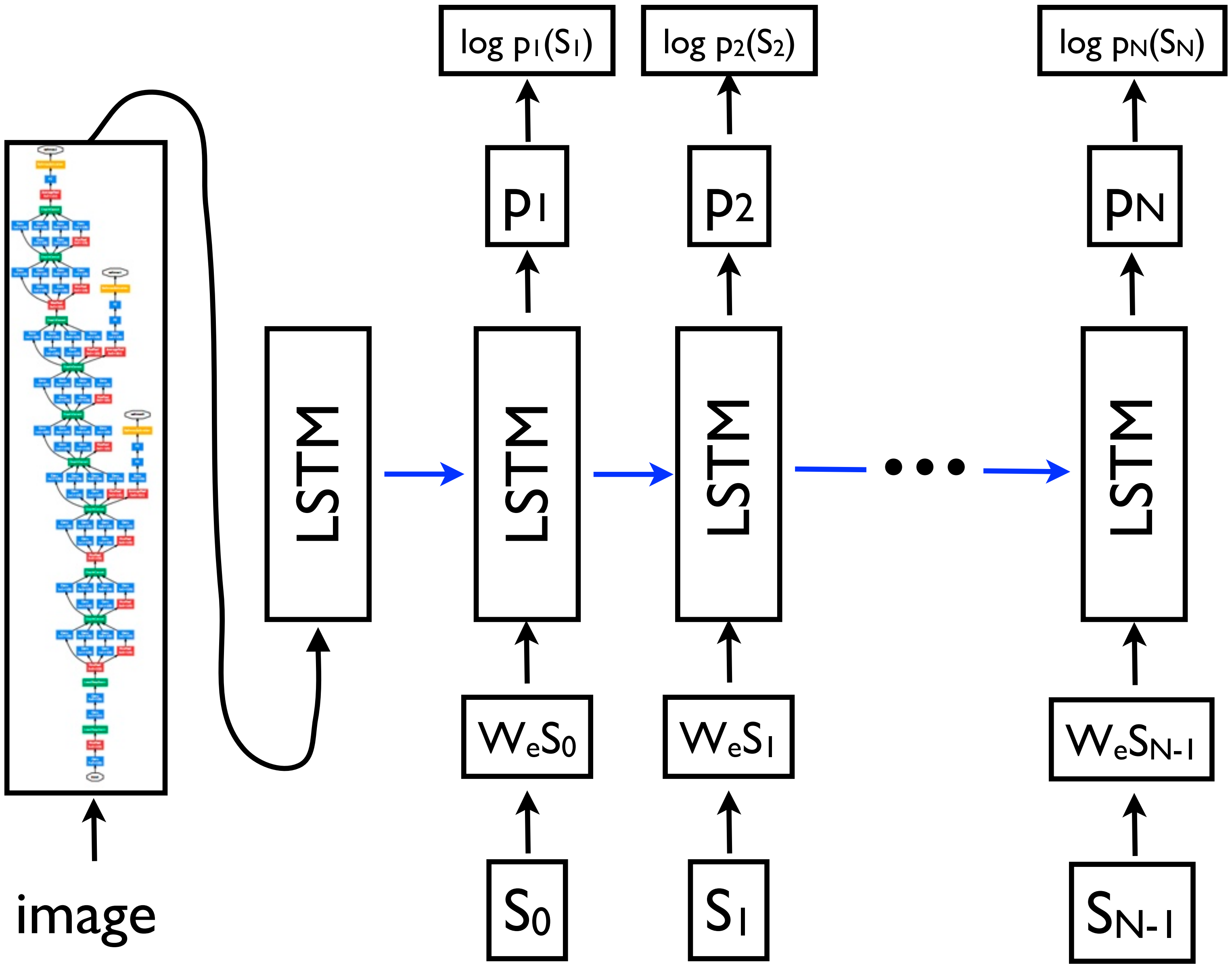


Decoding (into a sentence)

- Our RNN now needs to also produce an output (e.g., a word) as well as update its state
- The output is fed in as an input (to gauge what's left)

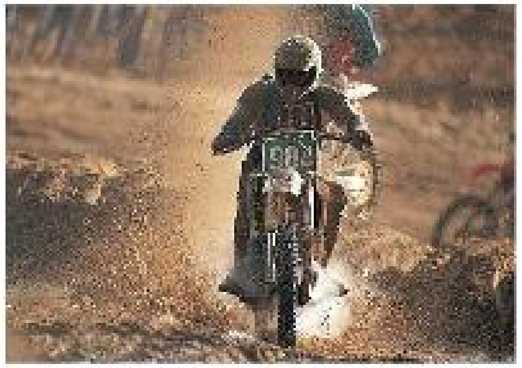


Mapping images to text

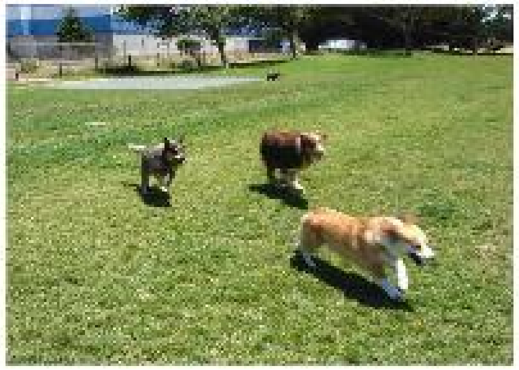


Examples

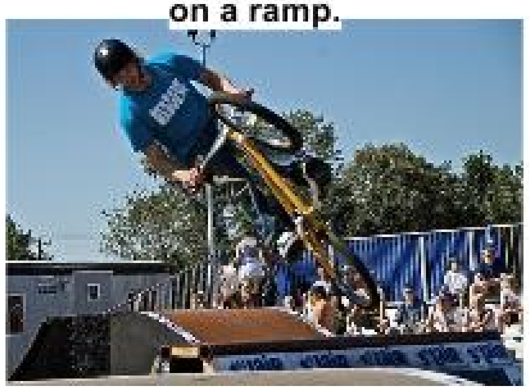
A person riding a motorcycle on a dirt road.



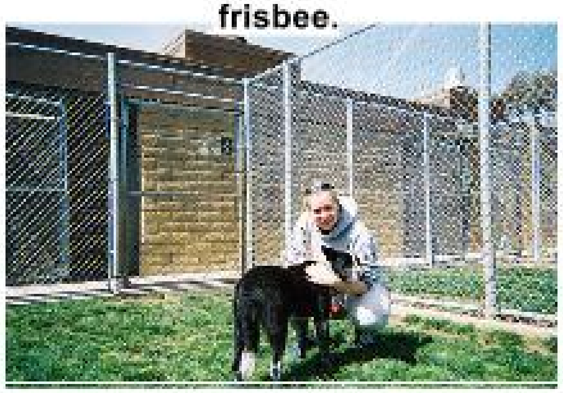
Two dogs play in the grass.



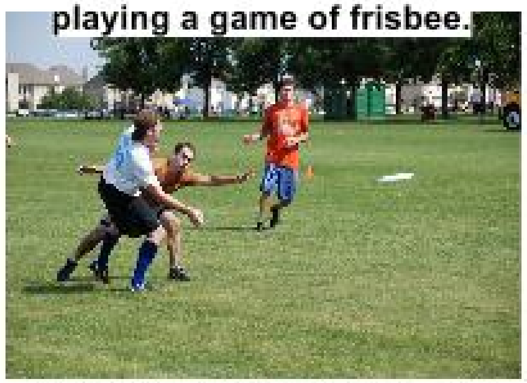
A skateboarder does a trick on a ramp.



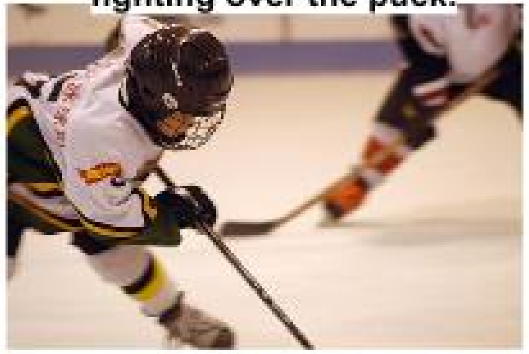
A dog is jumping to catch a frisbee.



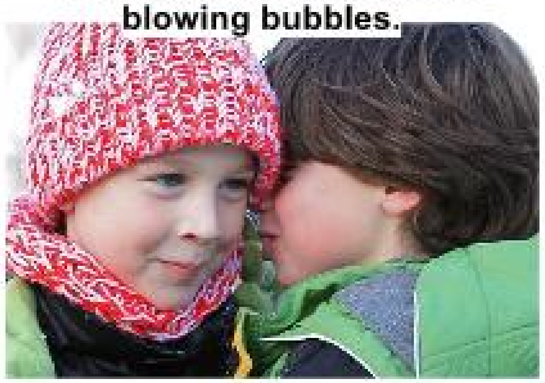
A group of young people playing a game of frisbee.



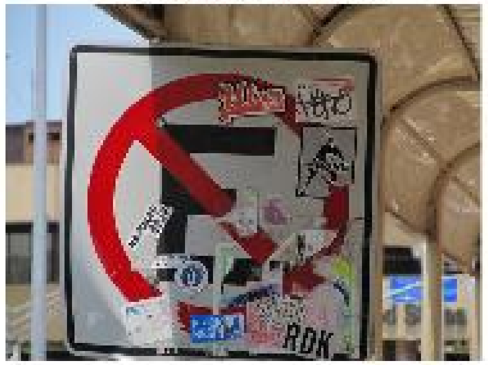
Two hockey players are fighting over the puck.



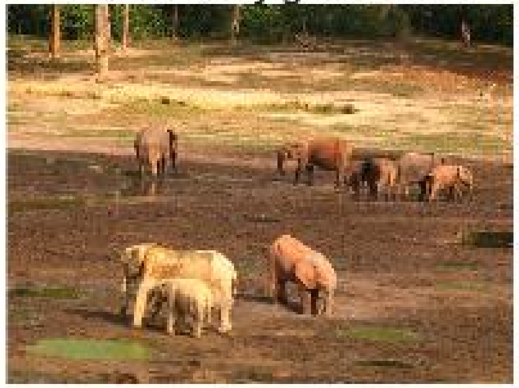
A little girl in a pink hat is blowing bubbles.



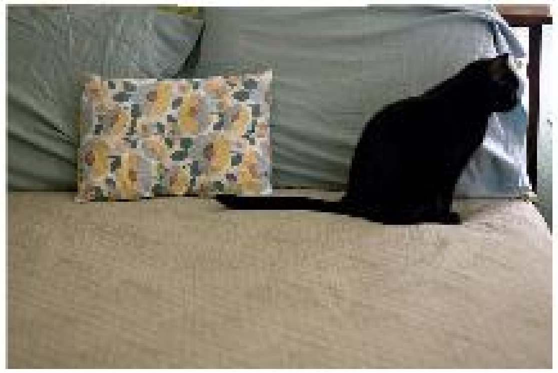
A refrigerator filled with lots of food and drinks.



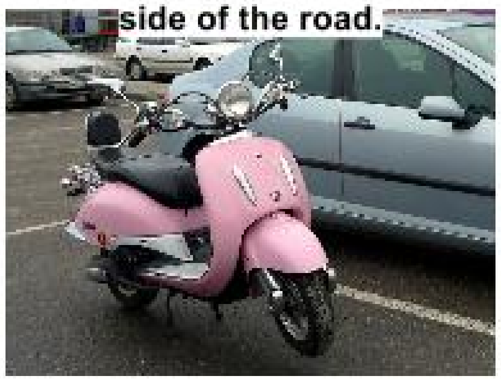
A herd of elephants walking across a dry grass field.



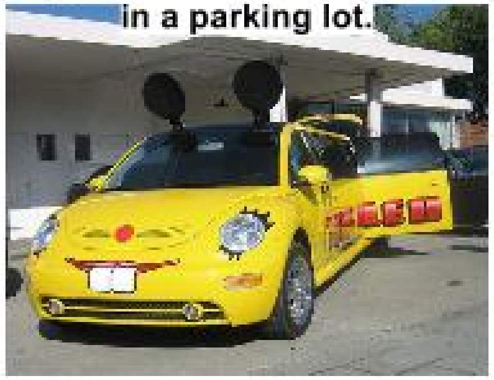
A close up of a cat laying on a couch.



A red motorcycle parked on the side of the road.



A yellow school bus parked in a parking lot.



🟩 Describes without errors | 🟧 Describes with minor errors | 🟨 Somewhat related to the image | 🟥 Unrelated to the image

Key things

- Markov models for sequences
  - how to formulate, estimate, sample sequences from
- RNNs for generating (decoding) sequences
  - relation to Markov models
  - evolving hidden state
  - sampling from
- Decoding vectors into sequences