ANLP Week 8 / Lecture 2
Recurrent Neural Networks

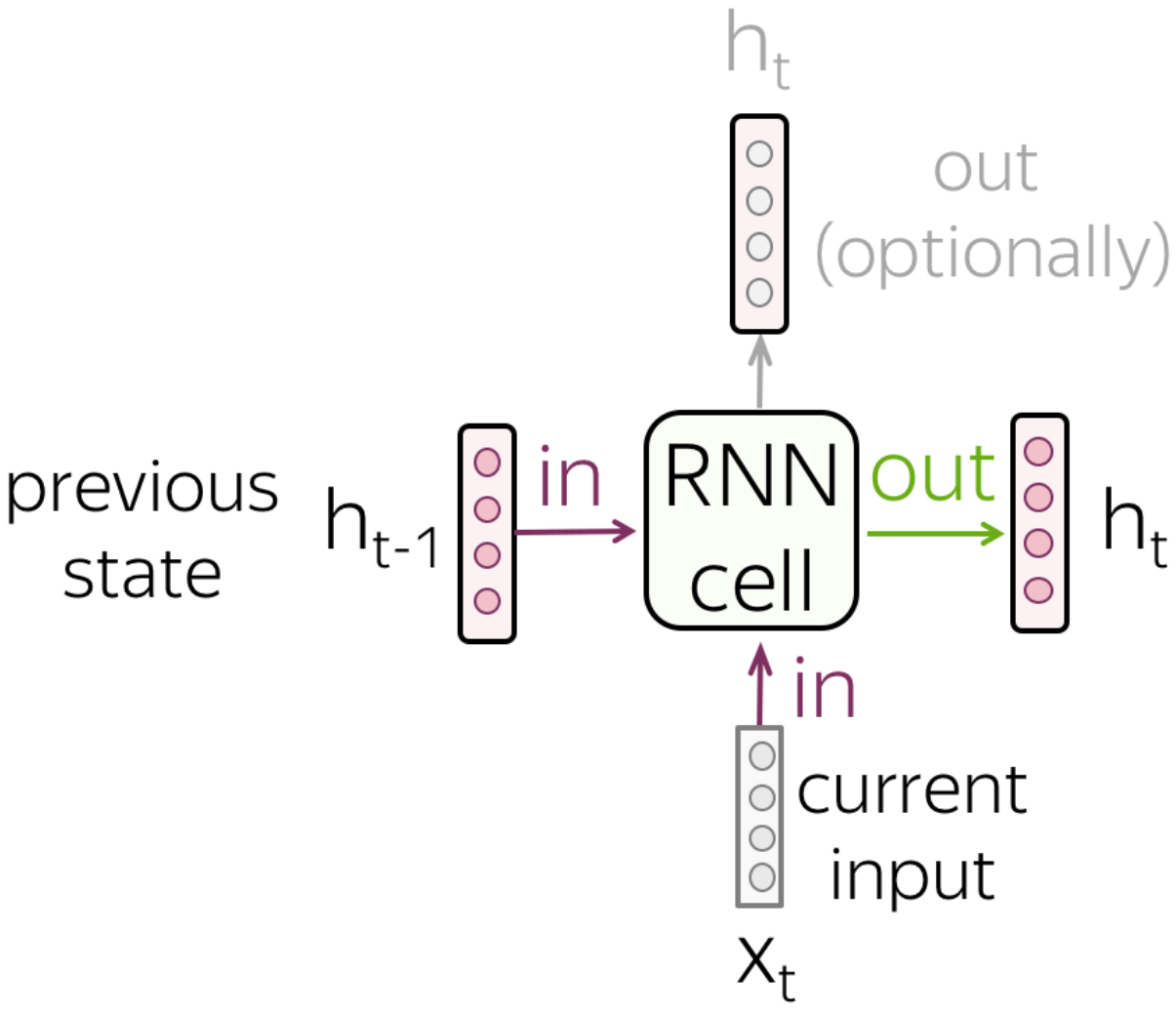
Edoardo Ponti
(with slides from Ivan Titov and Lena Voita)

School of informatics

Recap

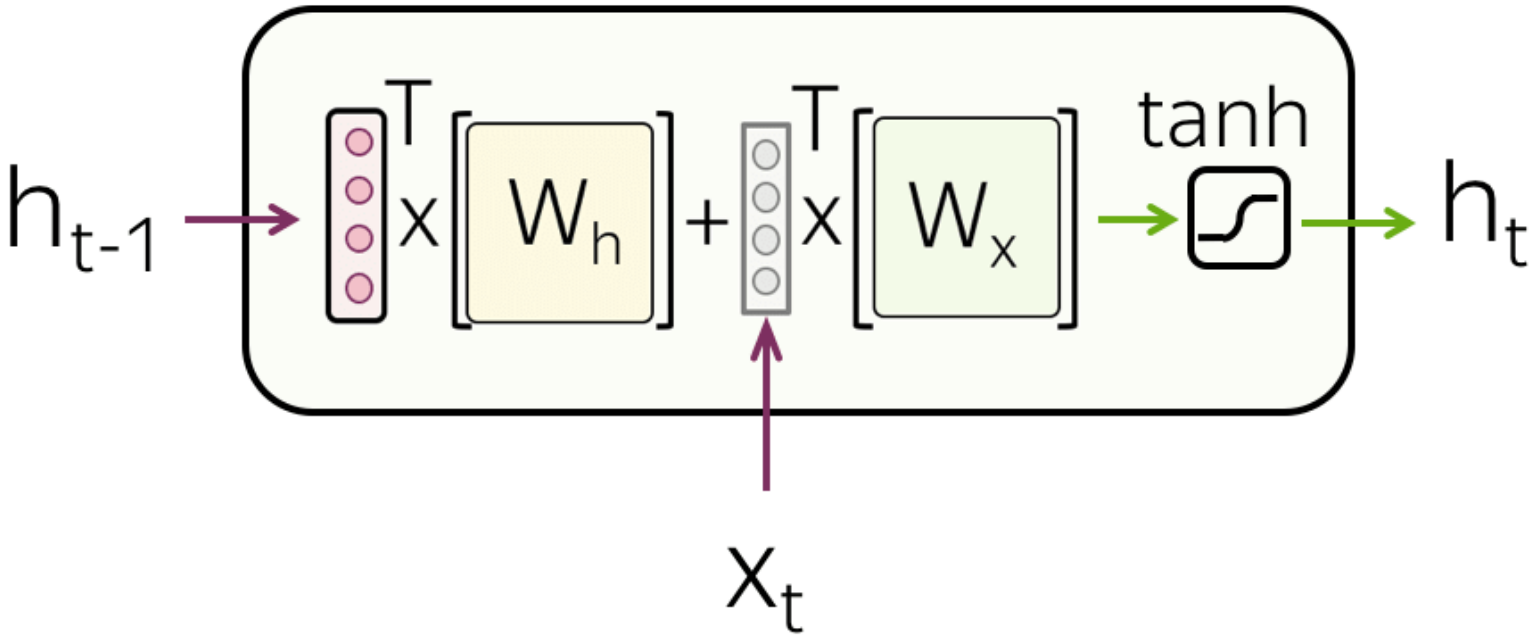
- Neural language models can represent word similarity by learning their representations, overcoming a major limitation of n-gram language models
- Multi-Layer Perceptron LMs are still based on the Markov assumption (fixed-size context)
- How can we develop neural LMs with unbounded context?

Recurrent Neural Networks: RNN cell

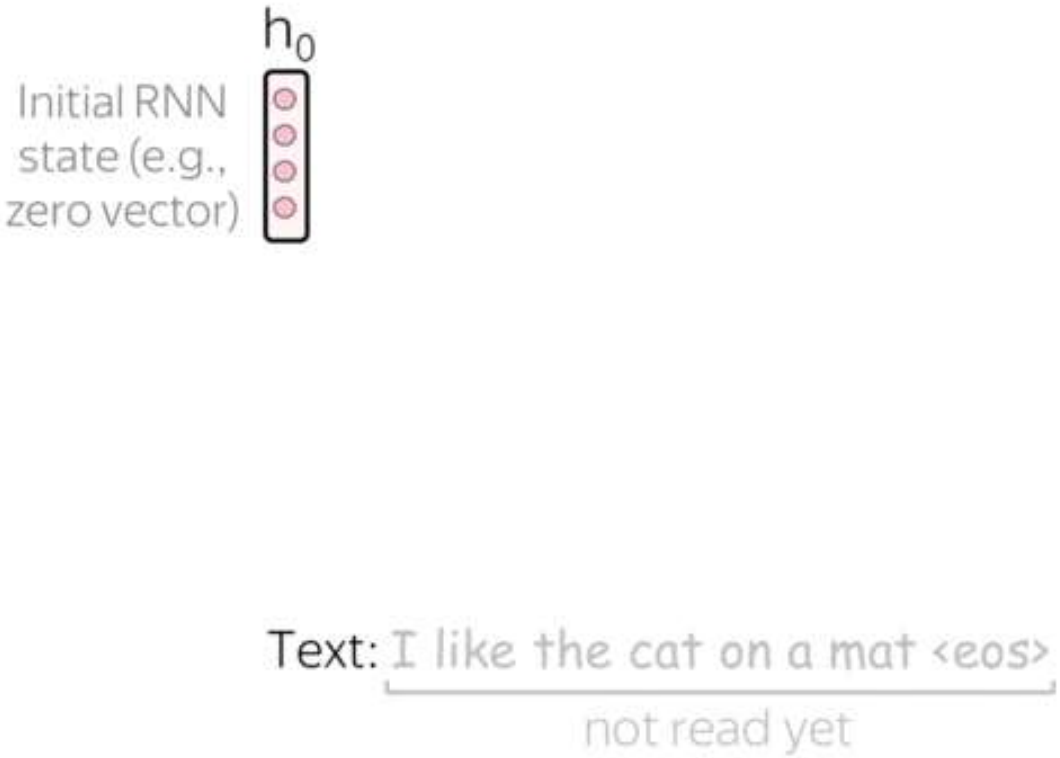


Vanilla RNN

$$h_t = \tanh(h_{t-1}W_h + x_t W_x)$$



RNN reads a sequence of tokens (video)



Language modelling

Recall, the language model assigns the probability to a sequence of words $y_1, y_2, \ldots, y_n$g on the chain rule:

$$P(y_1, y_2, \ldots, y_n) = P(y_1) \cdot P(y_2|y_1) \cdot P(y_3|y_1, y_2) \cdot \cdots \cdot P(y_n|y_1, \ldots, y_{n-1})$$

How do we compute $P(y_t|y_{<t})$

N-gram models and MLP language models made independence assumptions, basically breaking the sequence into smaller subsequences for estimation

Samples from n-gram models: any issues?

```
hahn , director of the christian " love and
compassion " was designed as a result of any form ,
in the transaction is active in the stuva grill .
_eos_
```

```
pupils from eastern europe , africa , saudi arabia
' s church , yearn for such an open structure of
tables several times on monday 14 september 2003 ,
his flesh when i was curious to know and also to
find what they are constructed with a speeding
arrow . _eos_
```

Samples from n-gram models: any issues?

```
hahn , director of the christian " love and
compassion " was designed as a result of any form ,
in the transaction is active in the stuva grill .
_eos_
```

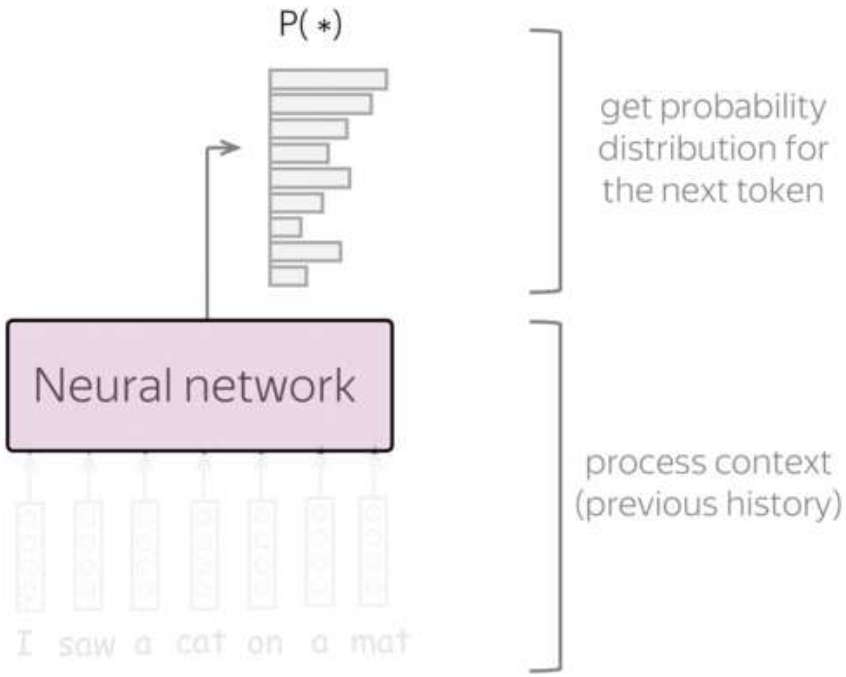
```
pupils from eastern europe , africa , saudi arabia
' s church , yearn for such an open structure of
tables several times on monday 14 september 2003 ,
his flesh when i was curious to know and also to
find what they are constructed with a speeding
arrow . _eos_
```

N-gram models clearly struggle with capturing longer context.
RNNs get rid of explicit independence assumptions

Recap for neural language models (video)

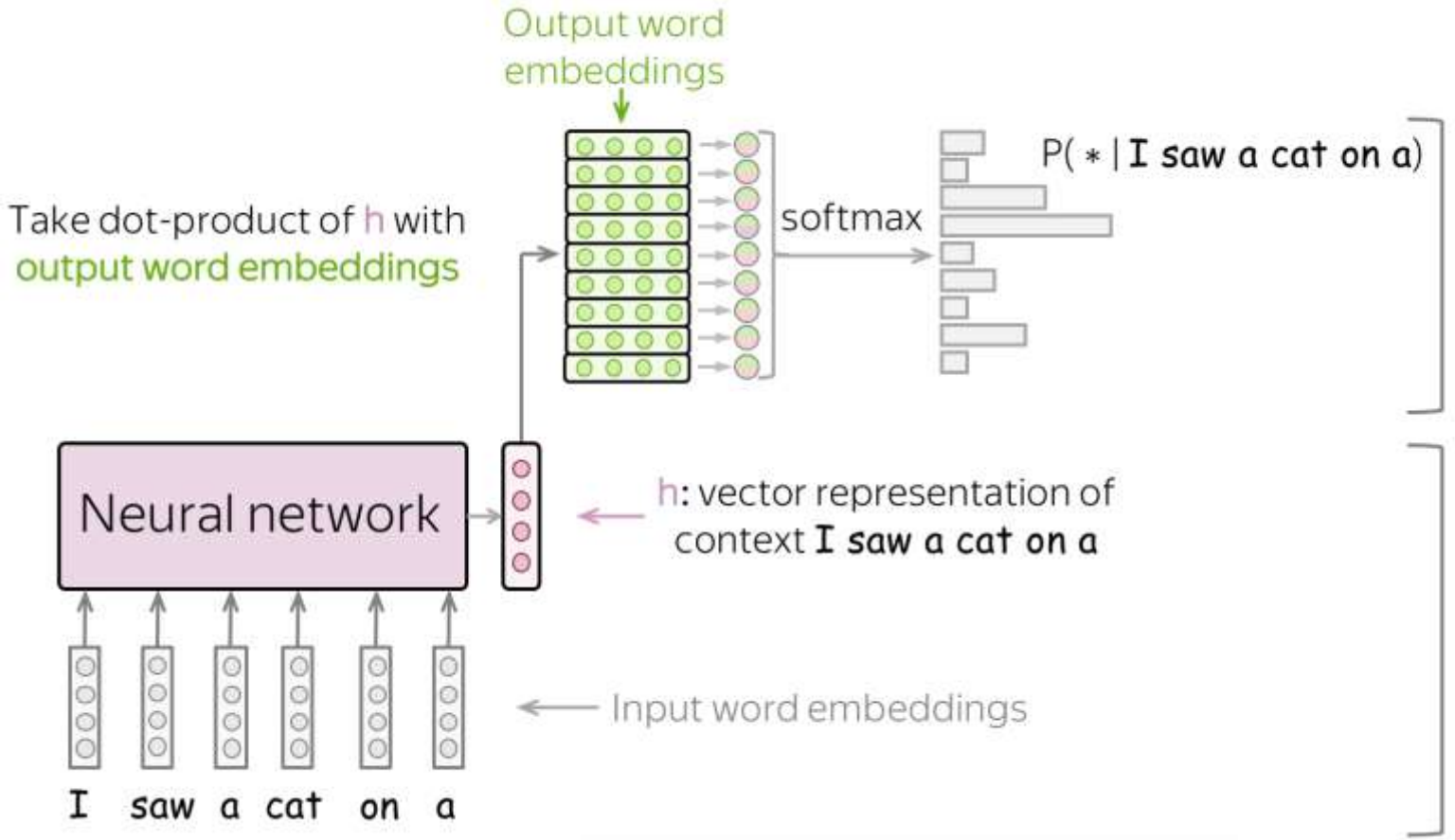
Neural language models has to:

1. *Produce a representation of the prefix*
2. *Generate a probability distribution over the next token*



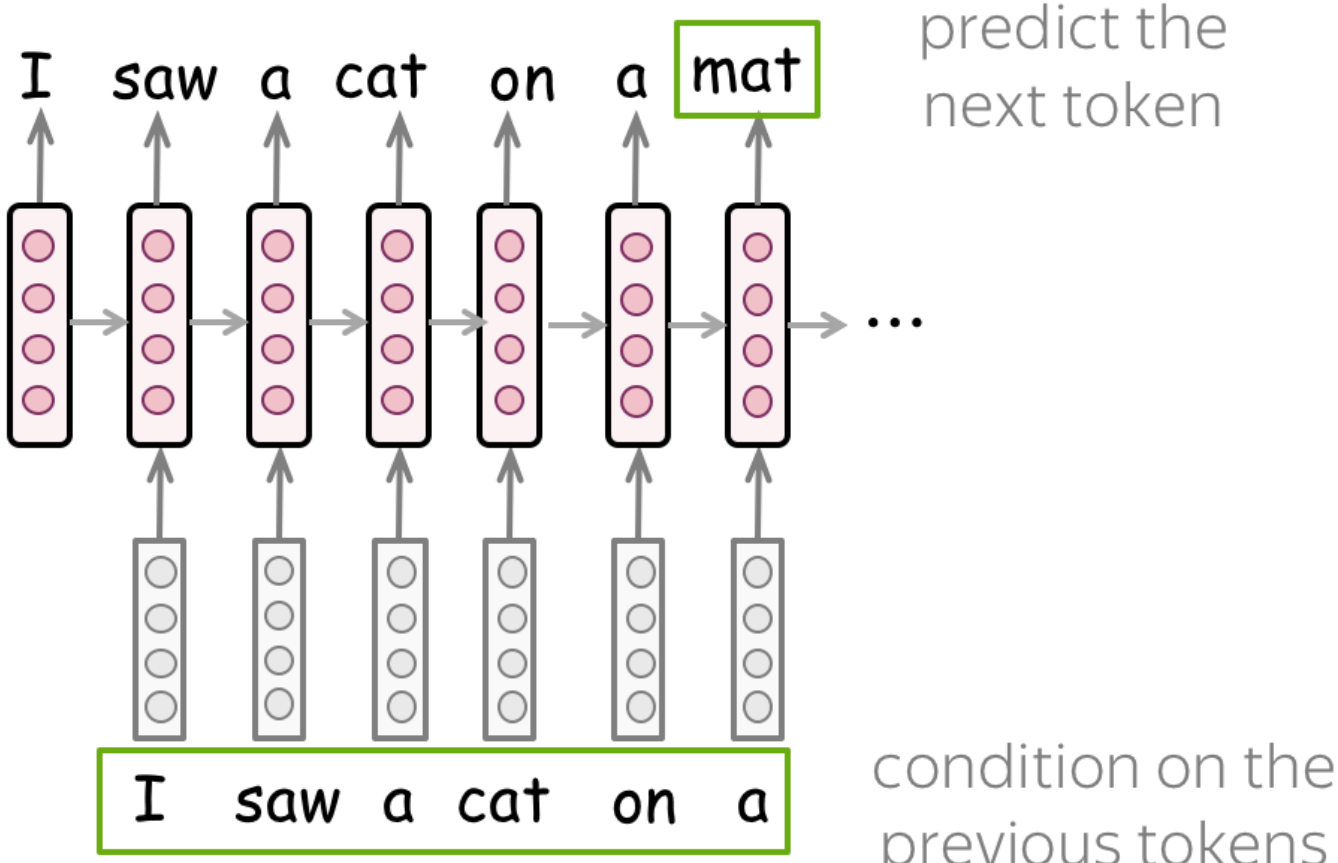
Predicting a word, given a prefix, is just a classification problem!

High-level intuition for an RNN language model

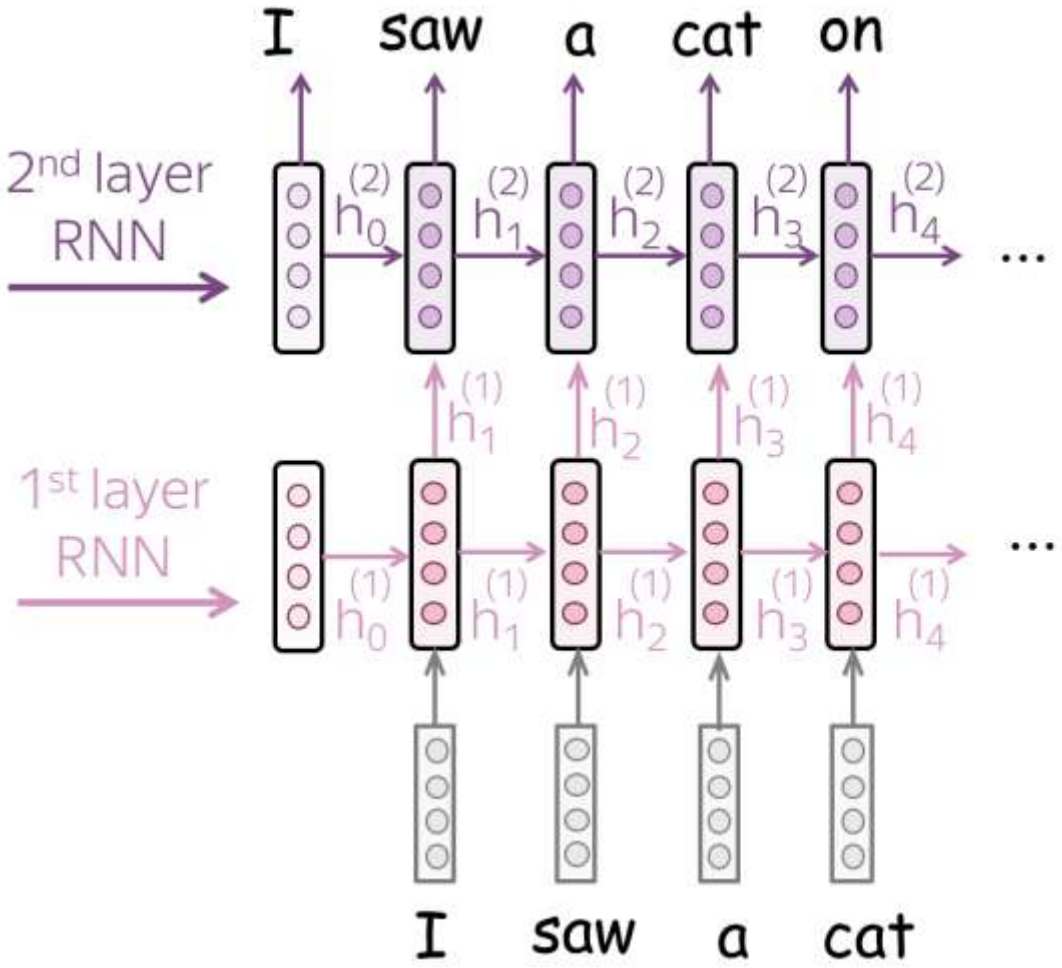


$$p(y_t|y_{<t}) = \frac{exp(h_t^T e_{y_t})}{\sum_{w \in V} exp(h_t^T e_w)}$$

RNN language model



Multi-layer RNN language model



Training the language model

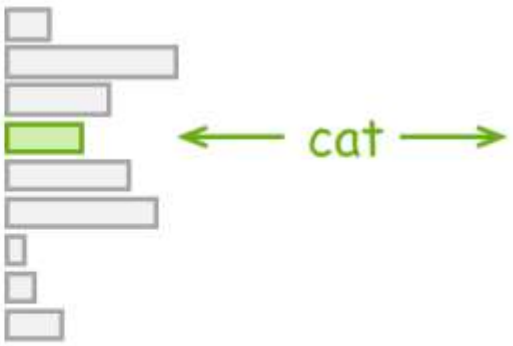
Training is similar to what we saw for logistic regression classifiers!

$$Loss = -\log(p(y_t|y_{<t}))$$

we want the model to predict this

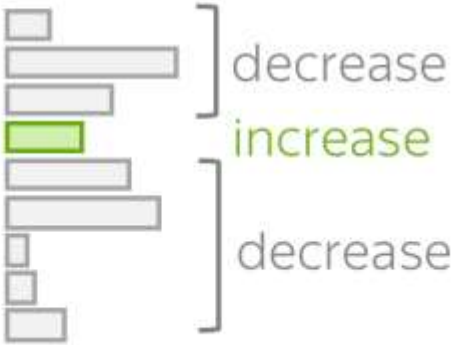
Training example: I saw a cat on a mat <eos>

Model prediction: p(* | I saw a)



Target

Loss = -log (p(cat)) → min



Training for one sentence with RNN LM (video)

Training (practical considerations)

- Sentences may be very long, so it becomes inefficient / ineffective (due to vanishing and exploding gradients) to backpropagate all the way to the start. So we need to **truncate** backpropagation through time (BPTT) in chunks.

- During inference, we feed as input for time step t+1 the token generated at t. During training, instead, we feed the gold-standard token for t+1, which is known as **teacher forcing.**

- We will see **backpropagation** (efficient gradient estimation for parameter updating) in more detail during the next lecture.

RNNs vs MLP vs n-gram models

n-gram language model

- relies on a short prefix, explicit independence assumption
- smoothing is necessary
- treats words as atomic symbols, cannot model their semantic similarity

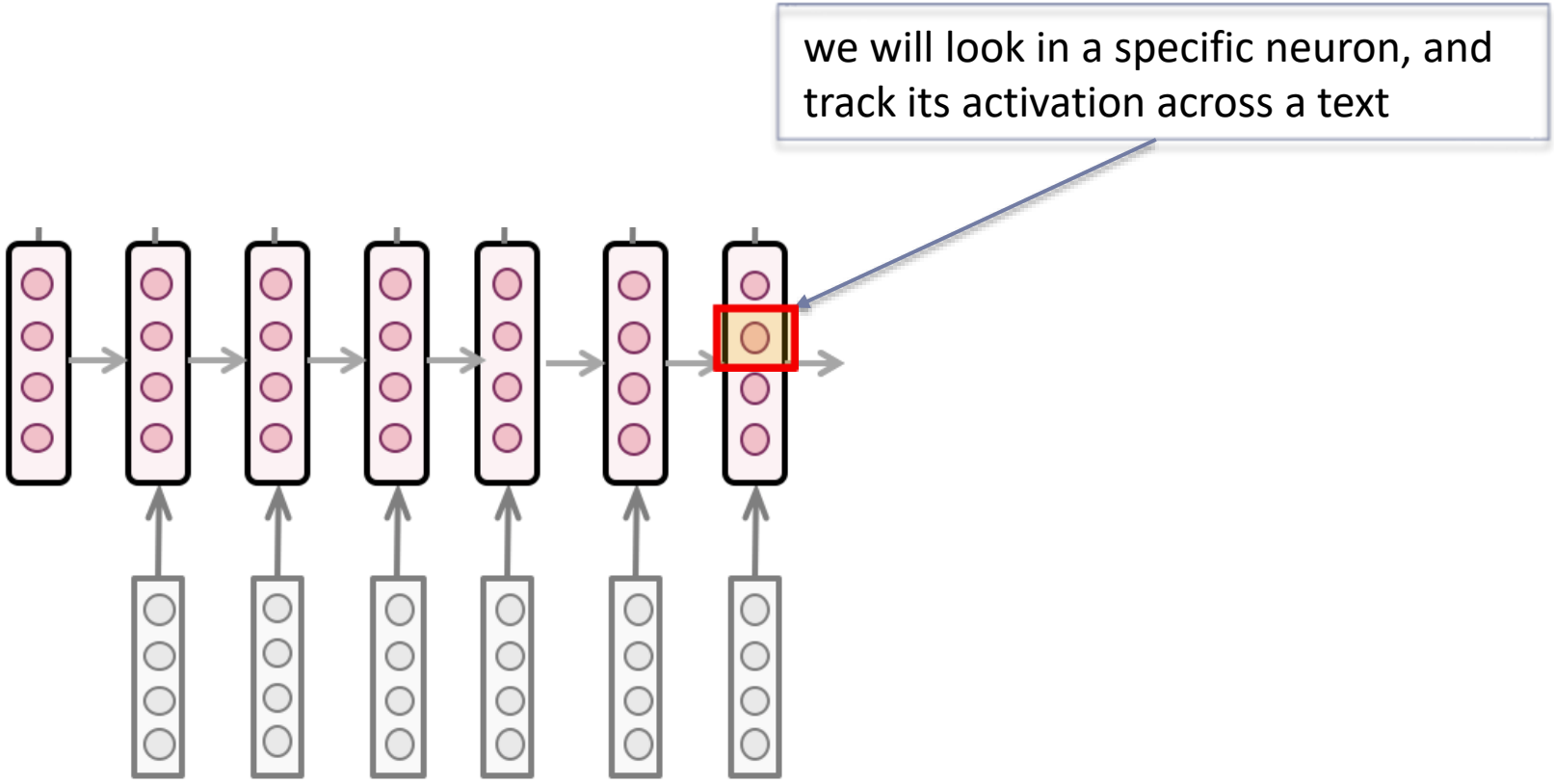
MLP language model

- models word similarity with dense representations
- stills makes independence assumption

RNN language model

- ‘compresses’ the past into a state, no independence assumptions
- all the information is carried through hidden states (hard to carry it across long distances)

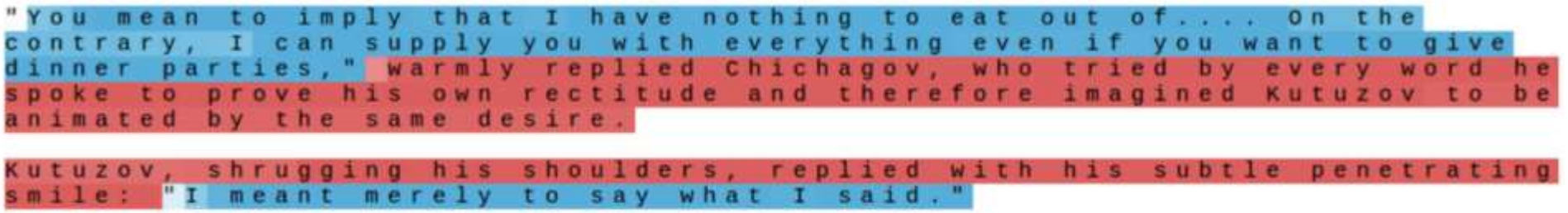
What does an RNN capture in its state?



It is a character-level LM, i.e. models a sequence of characters (rather than words)
Trained on Tolstoy's War and Peace and the source code of Linux Kernel (in C)

Karpathy et el., 2015
https://arxiv.org/abs/1506.02078

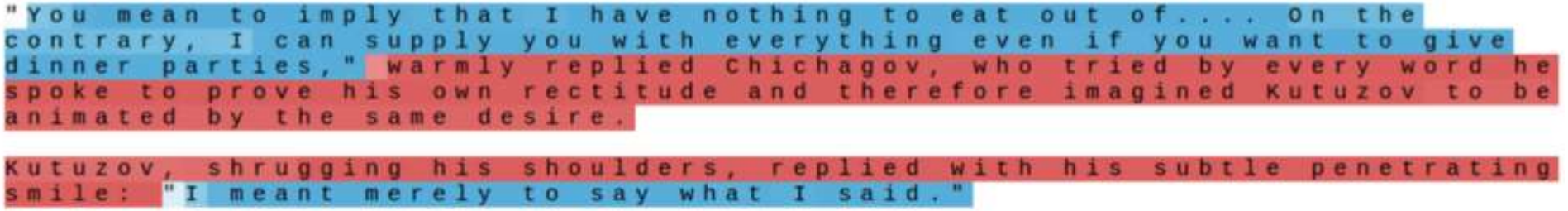
What does an RNN capture in its state?



Any hypothesis what this neuron is doing?

Karpathy et el., 2015
https://arxiv.org/abs/1506.02078

What does an RNN capture in its state?



Any hypothesis what this neuron is doing?

It activates within the quotes (“ ... ”)

Karpathy et el., 2015
https://arxiv.org/abs/1506.02078

What an RNN does capture in its state?

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
        siginfo_t *info)
{
  int sig = next_signal(pending, mask);
  if (sig) {
    if (current->notifier) {
      if (sigismember(current->notifier_mask, sig)) {
        if (!(current->notifier)(current->notifier_data)) {
          clear_thread_flag(TIF_SIGPENDING);
          return 0;
        }
      }
    }
    collect_signal(sig, pending, info);
  }
  return sig;
}
```

Any hypothesis what this neuron is doing?

Karpathy et el., 2015
https://arxiv.org/abs/1506.02078

What does an RNN capture in its state?

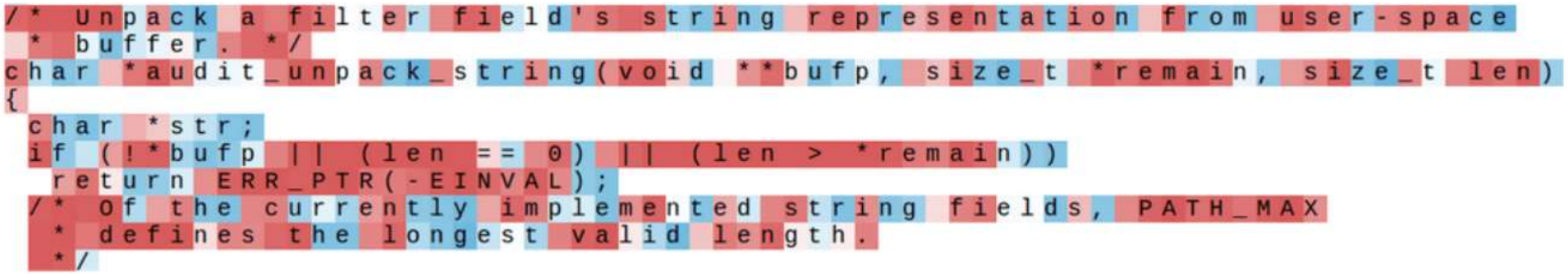
```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
        siginfo_t *info)
{
  int sig = next_signal(pending, mask);
  if (sig) {
    if (current->notifier) {
      if (sigismember(current->notifier_mask, sig)) {
        if (!(current->notifier)(current->notifier_data)) {
          clear_thread_flag(TIF_SIGPENDING);
          return 0;
        }
      }
    }
    collect_signal(sig, pending, info);
  }
  return sig;
}
```

Any hypothesis what this neuron is doing?

Activates within an if statement

Karpathy et el., 2015
https://arxiv.org/abs/1506.02078

What does an RNN capture in its state?



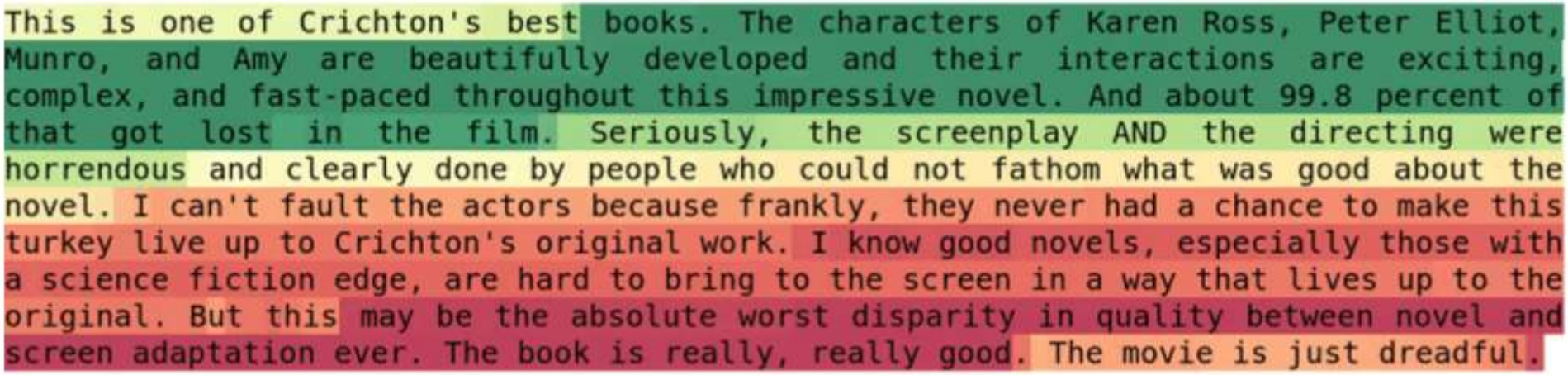
```
/* Unpack a filter field's string representation from user-space
 * buffer. */
char *audit_unpack_string(void **bufp, size_t *remain, size_t len)
{
  char *str;
  if (!*bufp || (len == 0) || (len > *remain))
    return ERR_PTR(-EINVAL);
  /* Of the currently implemented string fields, PATH_MAX
   * defines the longest valid length.
   */
```

Many neurons are not so easily interpretable

Karpathy et el., 2015
https://arxiv.org/abs/1506.02078

Sentiment neuron

This is from a much bigger LSTM model trained by OpenAI on Amazon reviews



They could also switch the sentiment at generation time to change the sentiment of an utterance (call *interventions*)

https://openai.com/research/unsupervised-sentiment-neuron

Do RNNs learn syntax?

Remember, we observed that n-grams are not able to capture syntactic agreement due to their Markov assumption (limited window)

Sam/Dogs sleeps/sleep soundly
Sam, who is my cousin, sleeps soundly
Dogs often stay at my house and sleep soundly
Sam, the man with red hair who is my cousin, sleeps soundly

Can neural networks accomplish this?

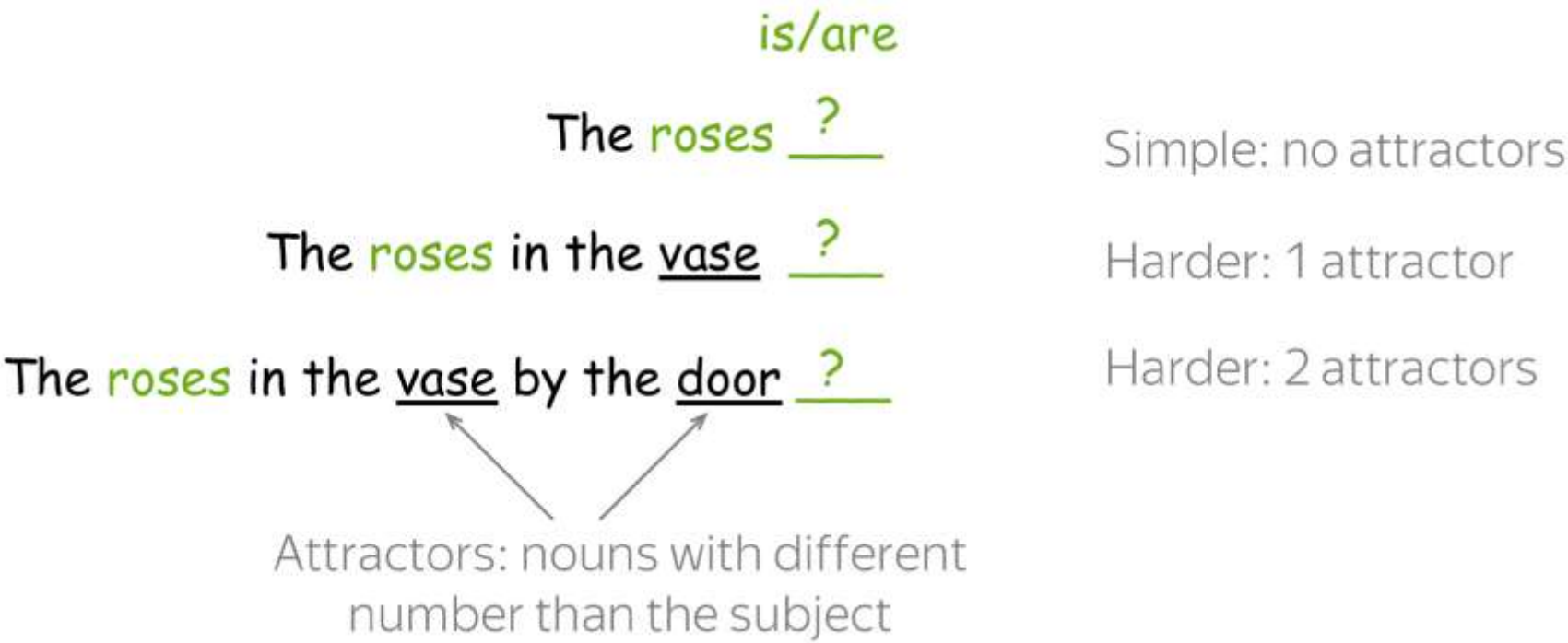
The roses in the vase by the door __?__ Competing answers: is, are

P(The roses in the vase by the door are)

P(The roses in the vase by the door is)

Is the correct answer ranked higher?
P(...are) > P(...is)?

Contrastive evaluation

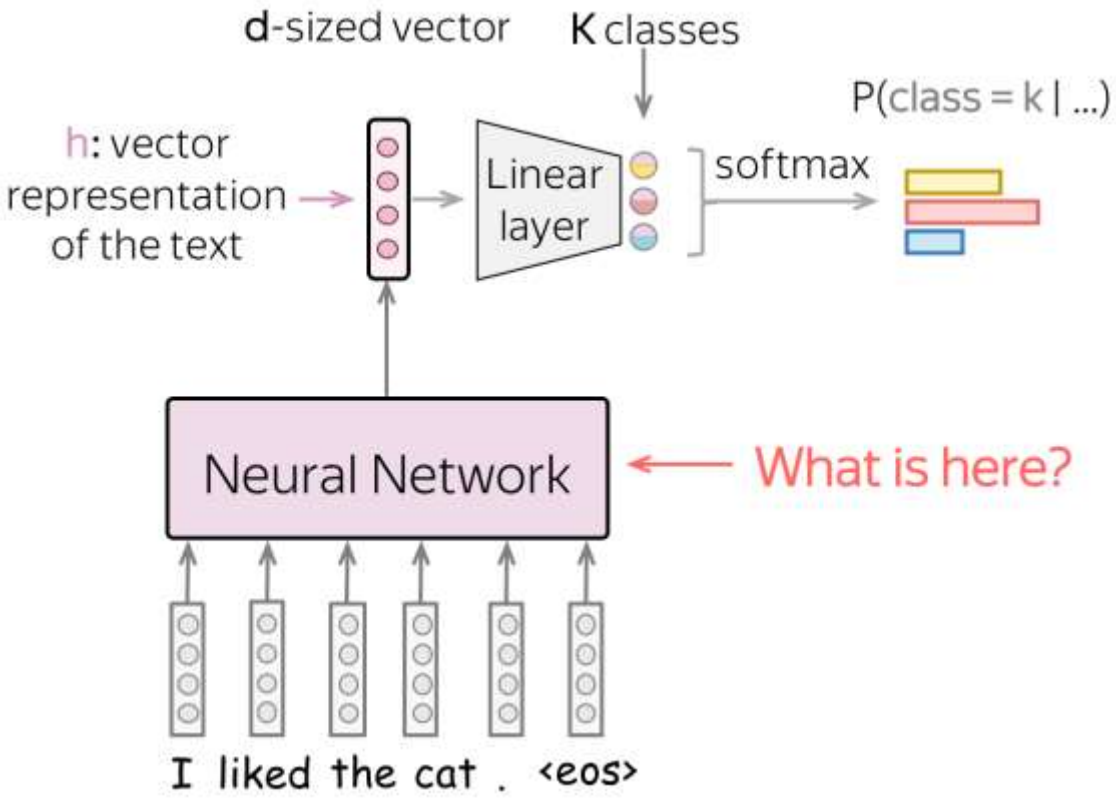


Short summary:

- need to be careful to prevent the model from relying on non-syntactic 'shortcuts'
- LSTMs models trained for language modeling were not as strong in that evaluation (but more powerful models will be)

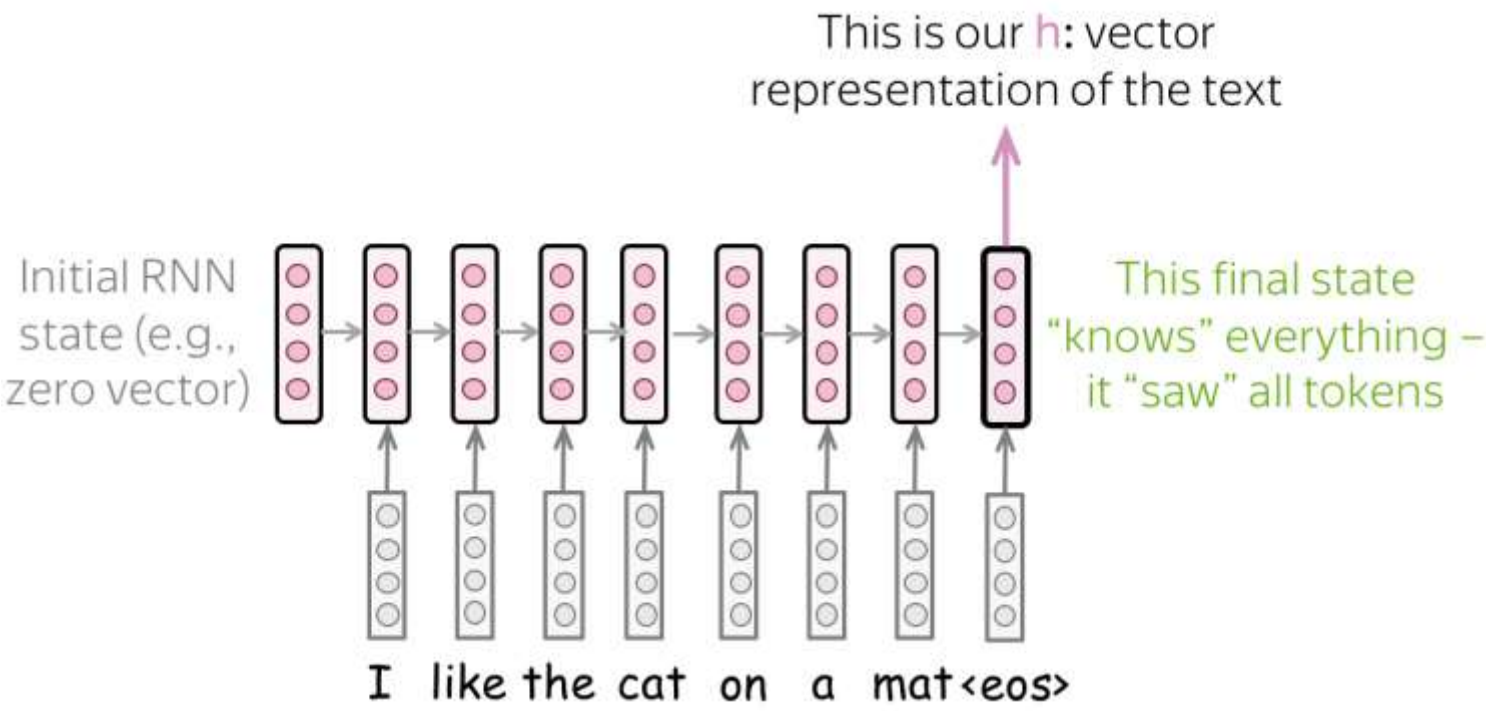
Linzen, Dupoux and Goldberg, 2016
https://aclanthology.org/Q16-1037.pdf

NNs for text classification

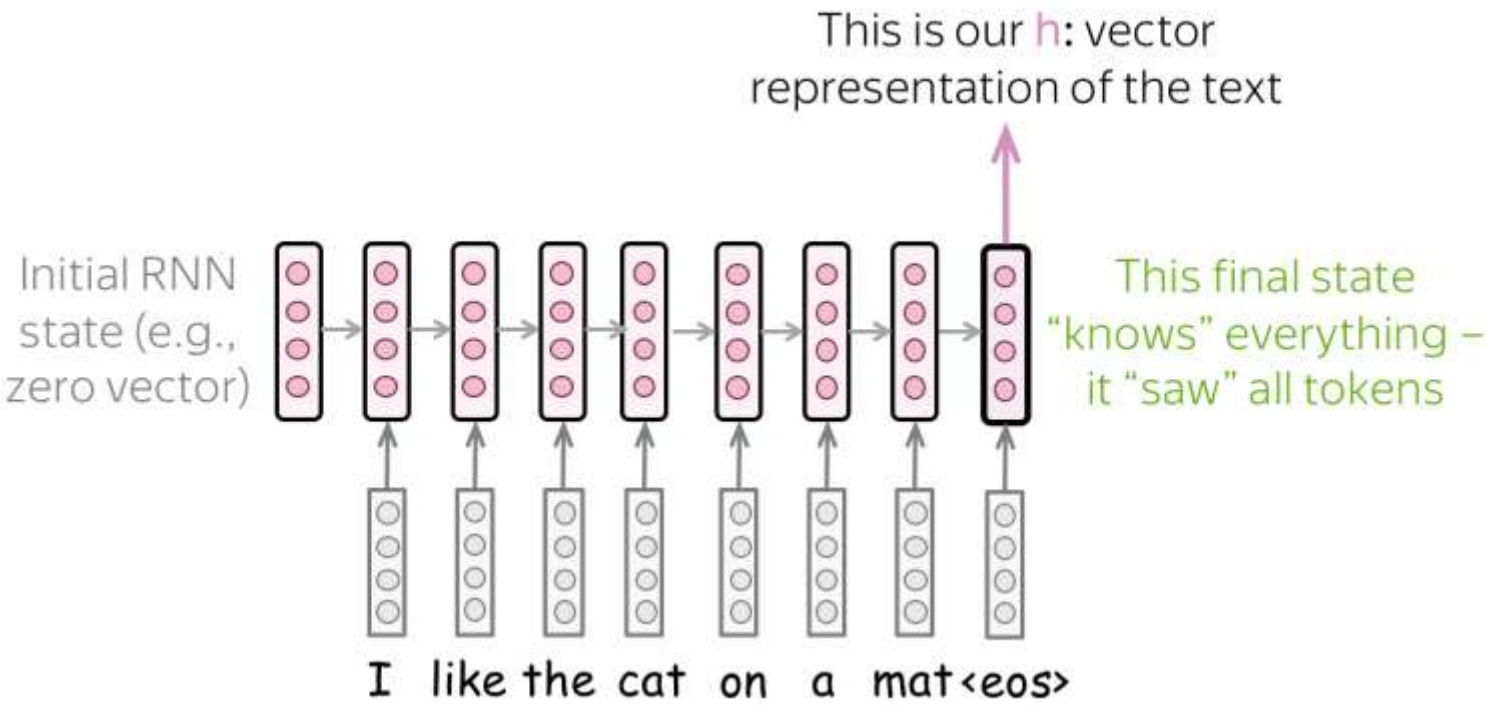


We will finish up with classification today by introducing how RNNs can represent entire sentences (instead of needing feature engineering)

Text representation with RNN



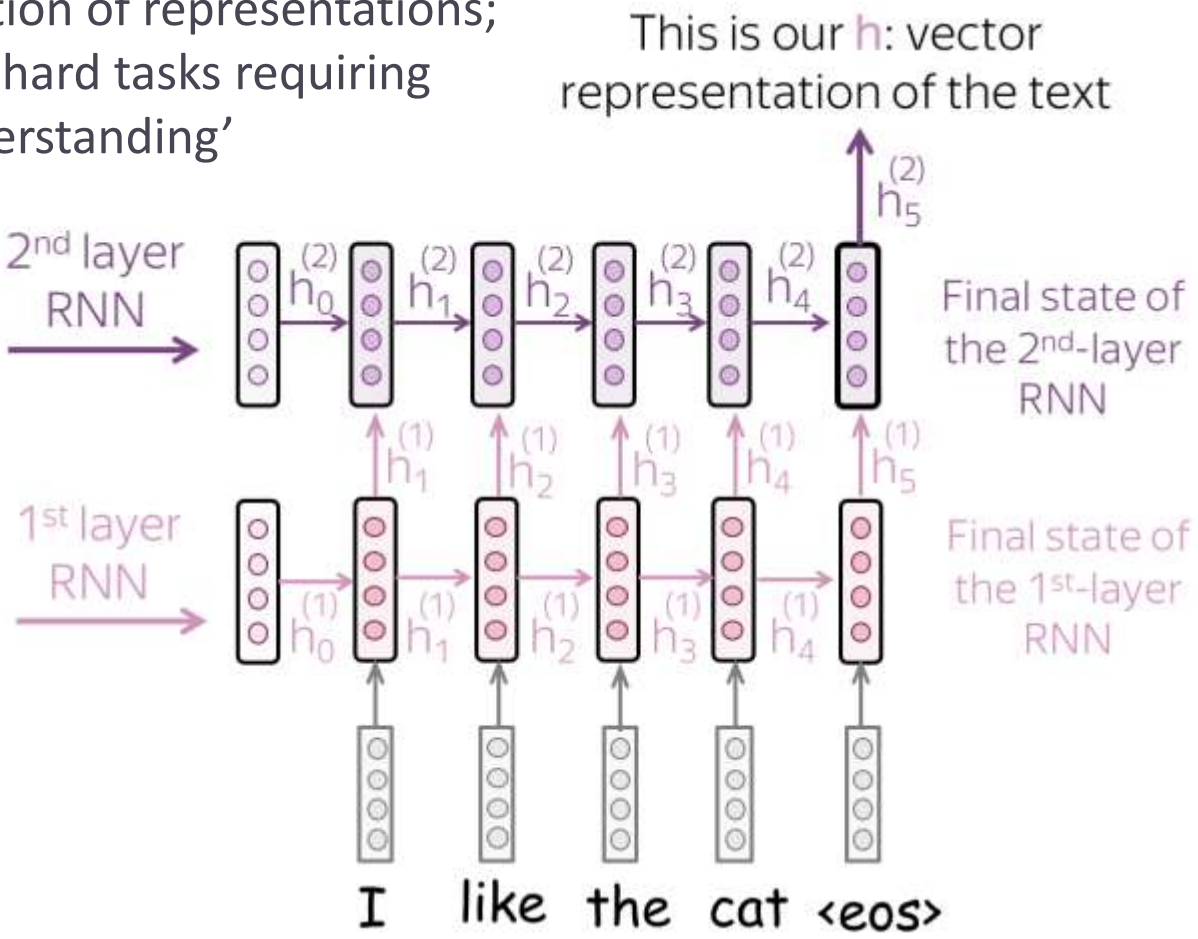
Text representation with RNN



The architecture may not be expressive enough, how can we make the model more powerful?

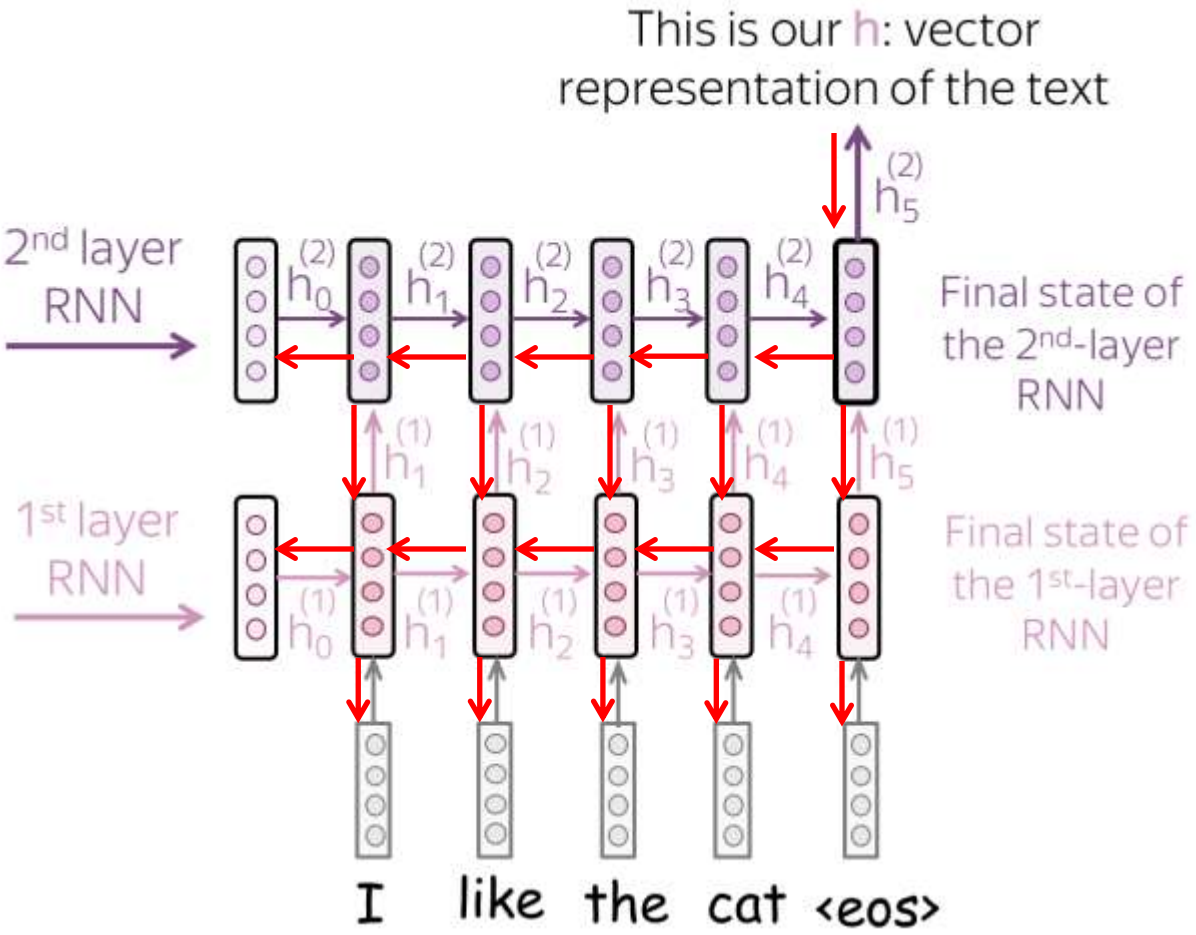
Text representation with multi-layer RNN

Better at capturing different levels of abstraction of representations; crucial for hard tasks requiring 'deep understanding'



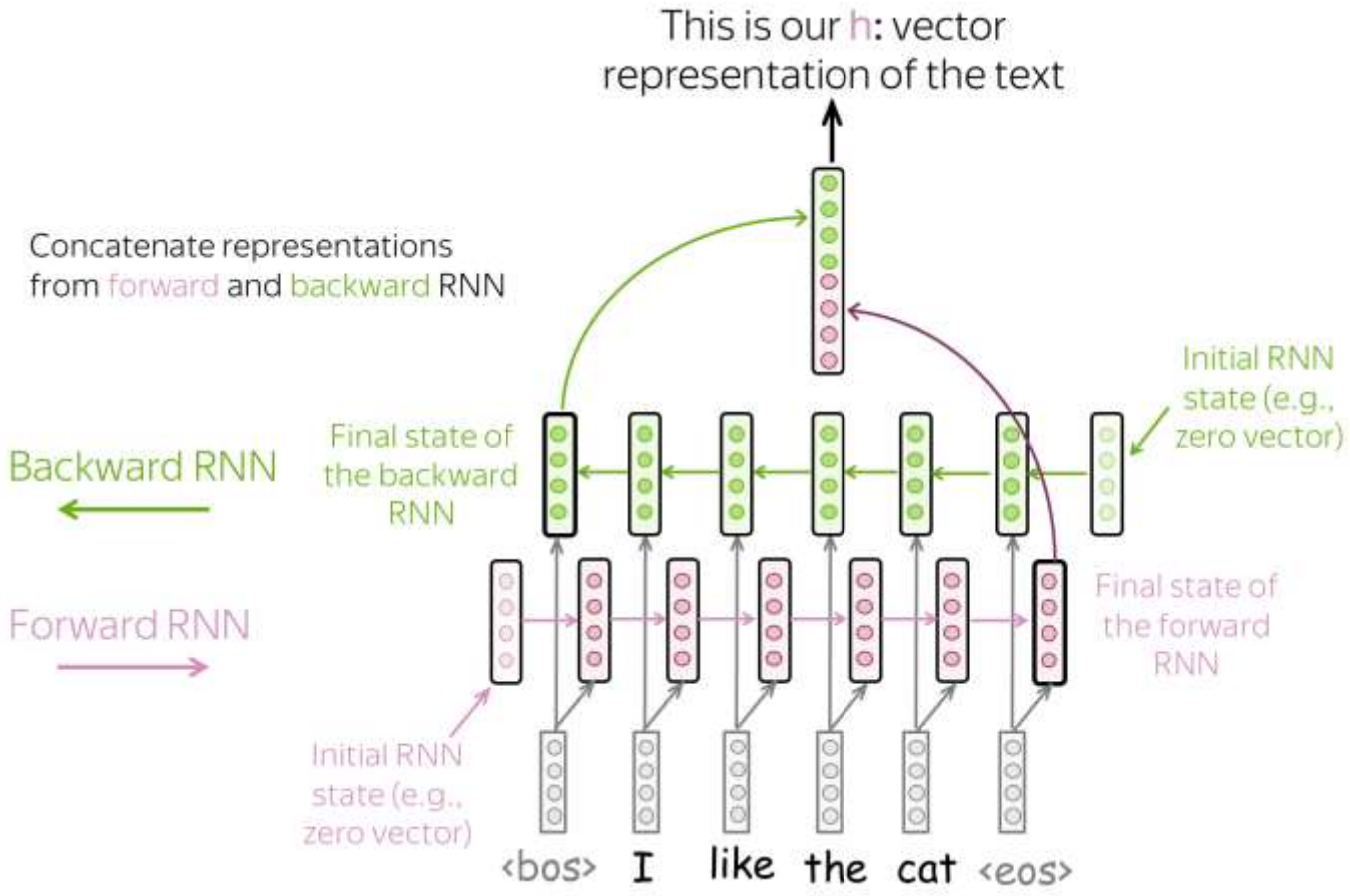
Is there a problem with passing only the last state as *h*?

Text representation with multi-layer RNN



Models learn by backpropagation, it takes many steps to propagate to the very beginning of the sentence; the model will not learn to reliably encode early parts of the sentence, how can we address it?

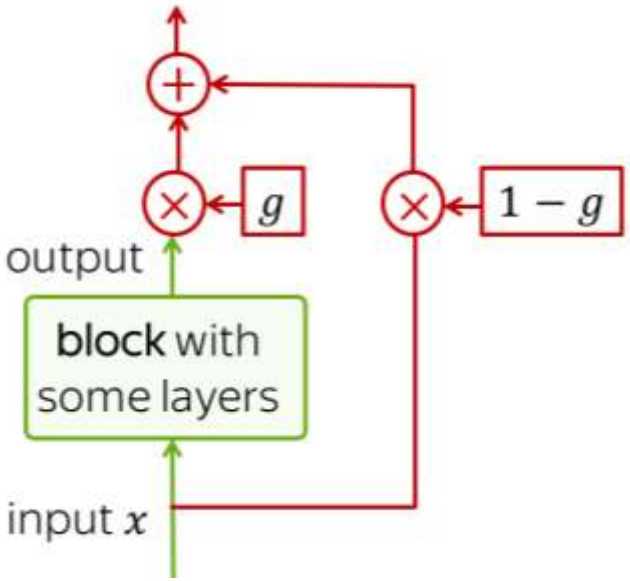
Text representation with bidirectional RNN



Stacking many layers

Unfortunately, when stacking a lot of layers, you can have a problem with propagating gradients from top to bottom through a deep network.

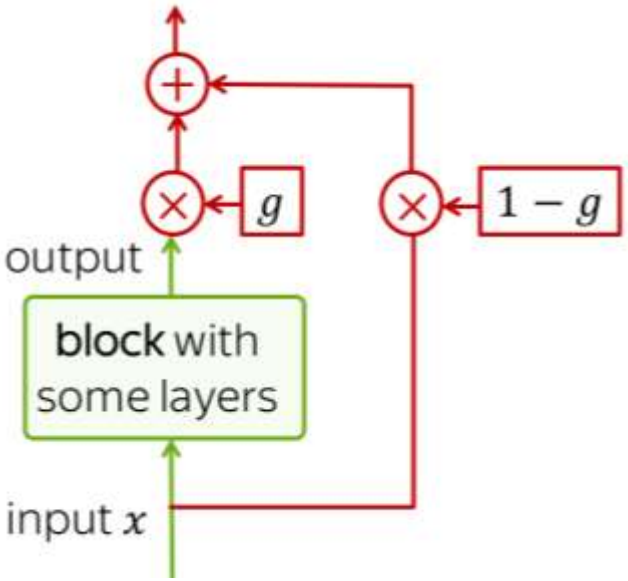
To mitigate, this we use **Residual (aka Highway) connections**:



Stacking many layers

Unfortunately, when stacking a lot of layers, you can have a problem with propagating gradients from top to bottom through a deep network.

To mitigate, this we use **Residual (aka Highway) connections**:



Highway connection: gated sum of a block's input and output

$$g = \sigma(Wx + b)$$

Gate: because of σ, its values are in (0, 1)

Logistic sigmoid:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Summary on classification

- Naïve Bayes

  very fast to train, robust, makes overly strong assumptions

- Logistic regression

  still easy to train, requires strong features, fewer assumptions

- Recurrent Neural Networks

  does not rely on feature engineering, but the information is carried within the sentence through a vector

Summary of the lecture

- Recurrent neural networks (RNNs) can capture long-distance dependences in text by modelling arbitrary-length prefixes
- Some of the neurons in RNNs may be interpretable
- RNNs allow for developing text classifiers by representing a sentence in its entirety.