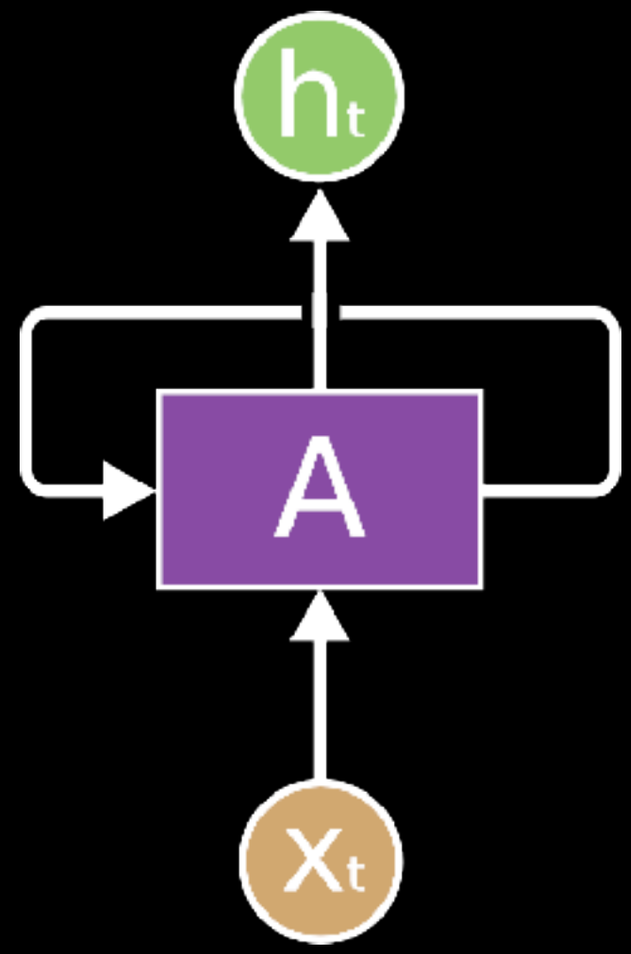
Deep Learning for Computer Vision

Lecture 12: Time Sequence Data, Recurrent Neural Networks (RNNs), Long Short-Term Memories (LSTMs), and Image Captioning

Peter Belhumeur

Computer Science
Columbia University

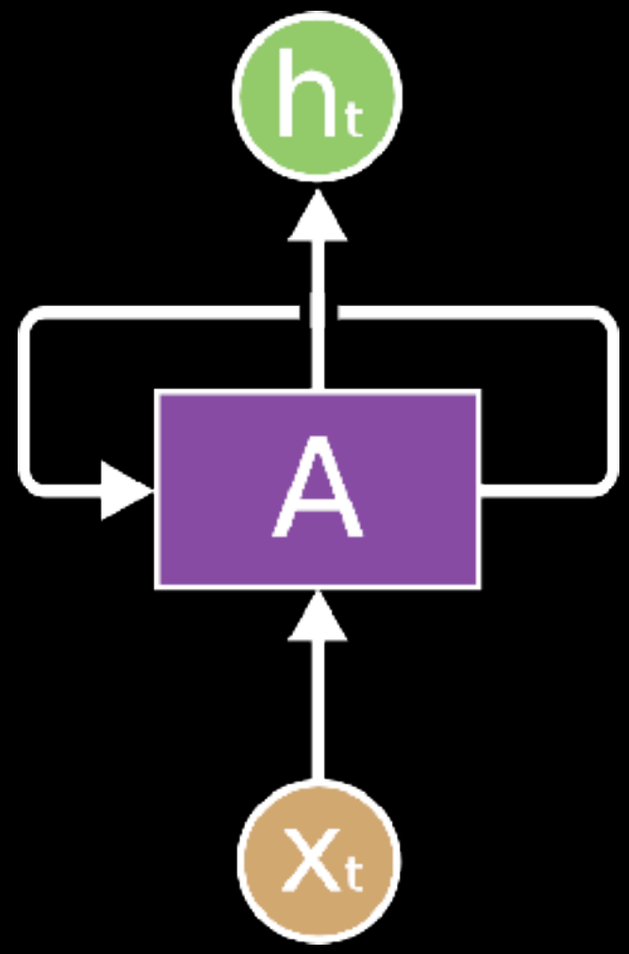
Recurrent Neural Nets (RNNs)



Input and output is a sequence

Diagrams are adapted from [C. Olah, 2015]

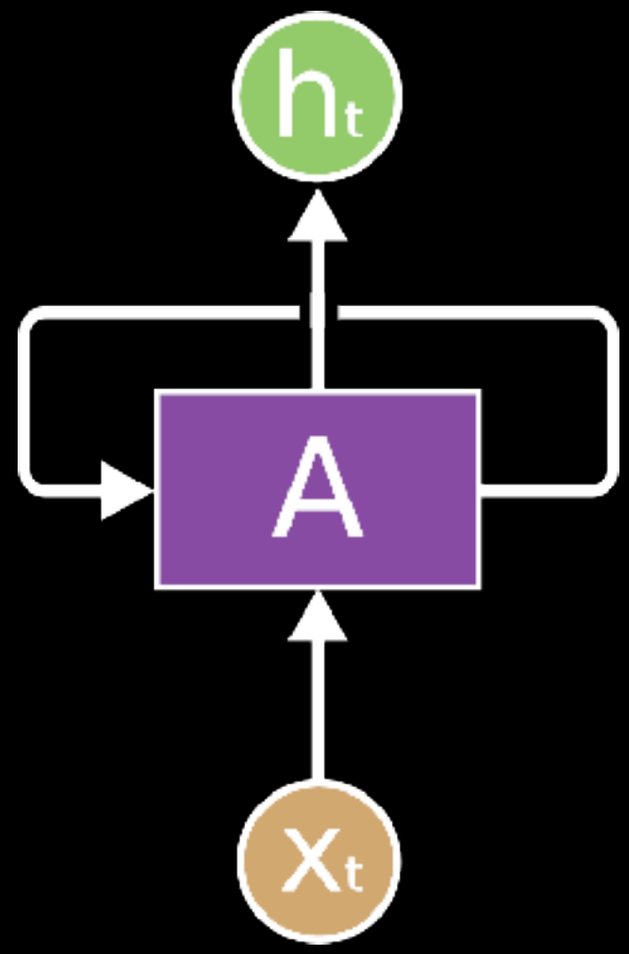
Recurrent Neural Nets (RNNs)



Sequence examples:

1. Frames in a video
2. Words in a sentence
3. Speech Signal
4. A user's sequence of actions
5. Any sequence of ordered measurements

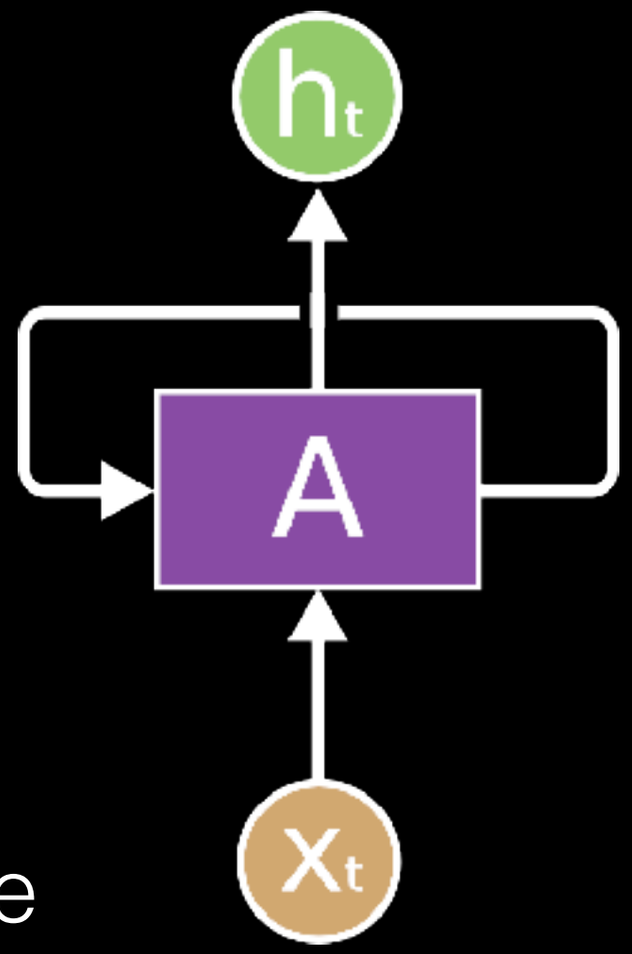
Recurrent Neural Nets (RNNs)



Applications:

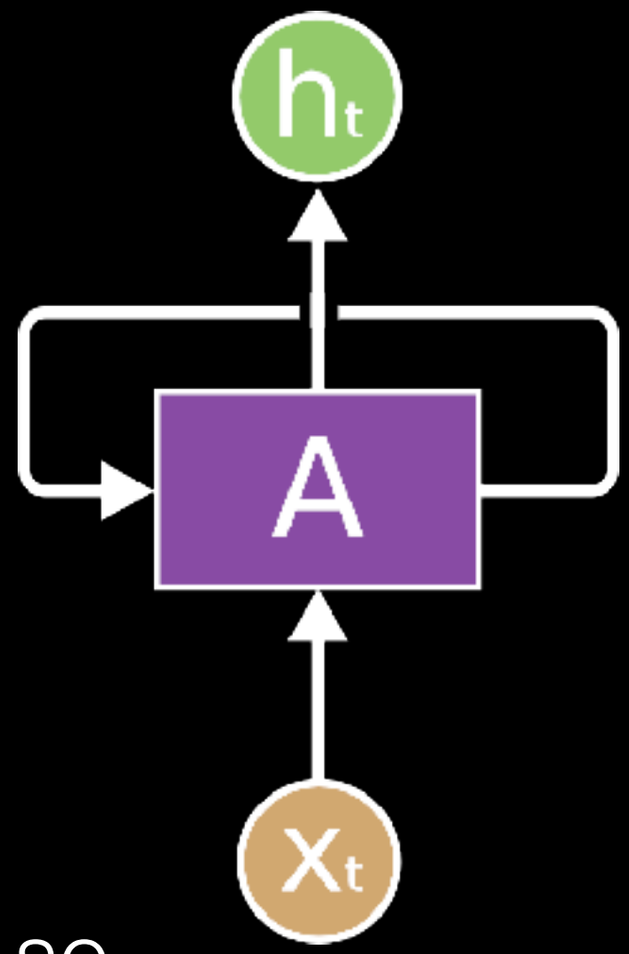
1. Video Understanding
2. Image Captioning
3. Speech Recognition
4. Language Modeling
5. Language Translation
6. …

Recurrent Neural Nets (RNNs)



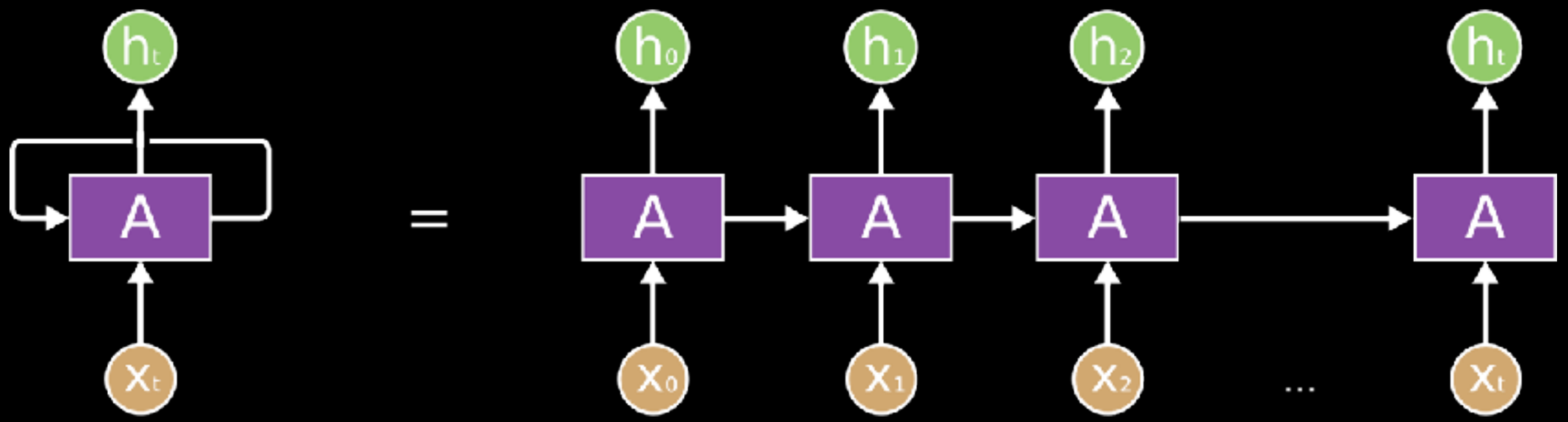
RNNs are repeated instances of the same network with a previous instance passing information to the next instance.

Recurrent Neural Nets (RNNs)



The weights are shared across time, so the weights of the network are the same for each time instance!

Recurrent Neural Nets (RNNs)

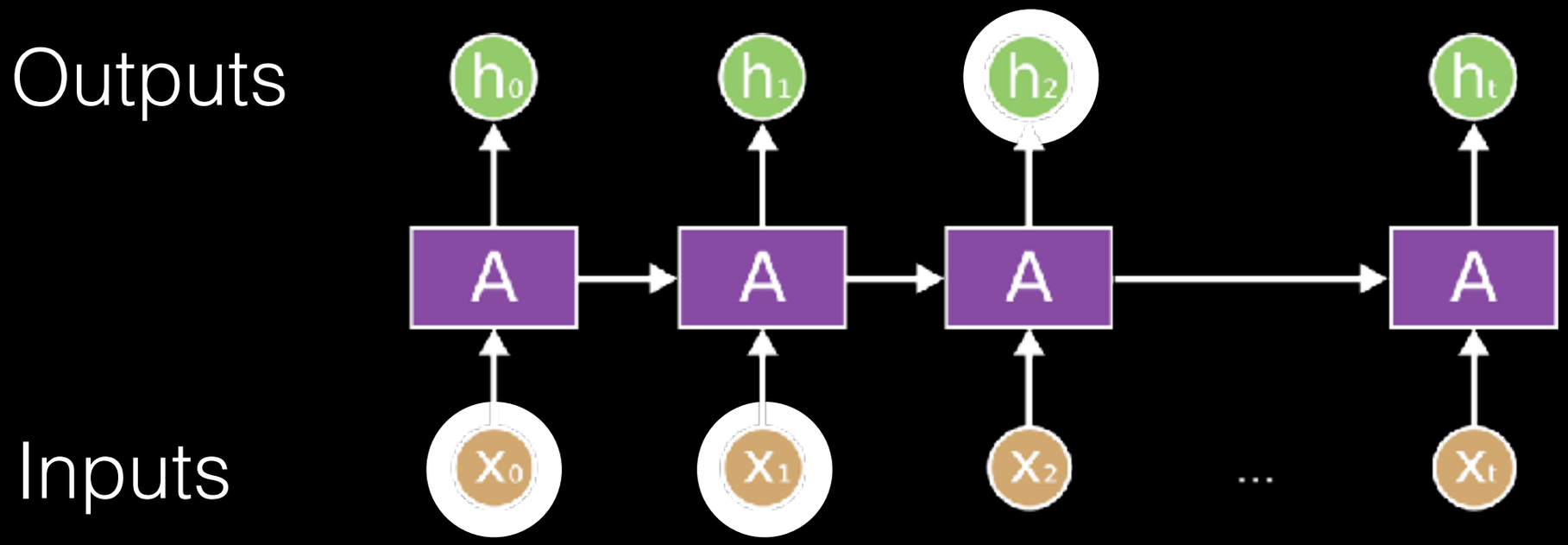


We can “unroll” RNNs to see how this network is well suited for “time” sequence problems.

Short-Term Dependencies

- “The Beatles Abbey …”
- “Show me the …”
- “The Wolf of Wall …”
- “The surfer rode the …”

Short-Term Dependencies

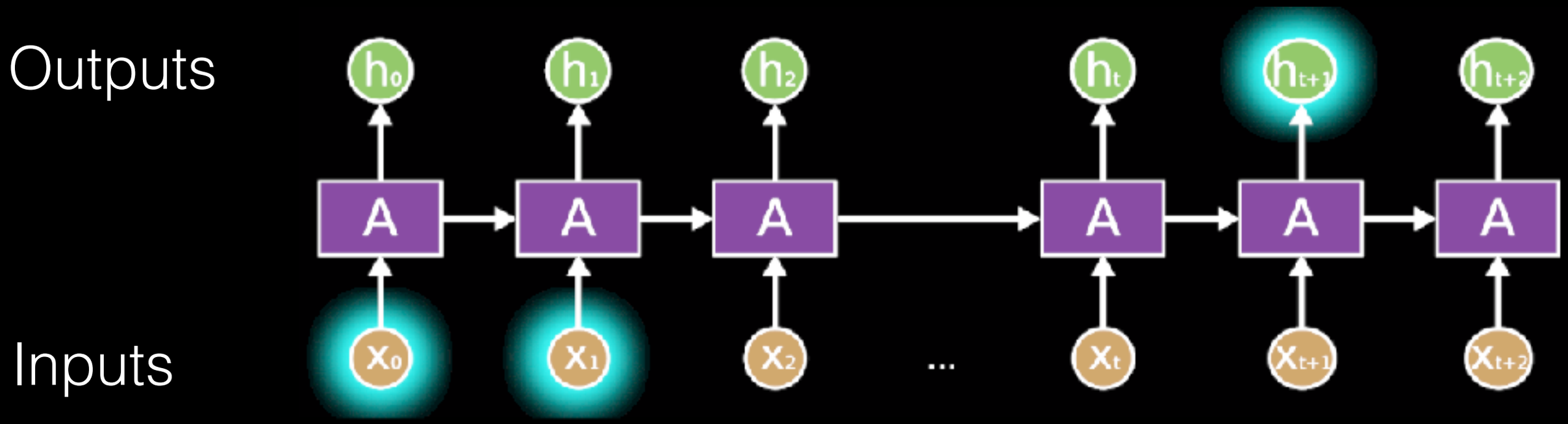


RNNs are good at handling short-term dependencies.

Long-Term Dependencies

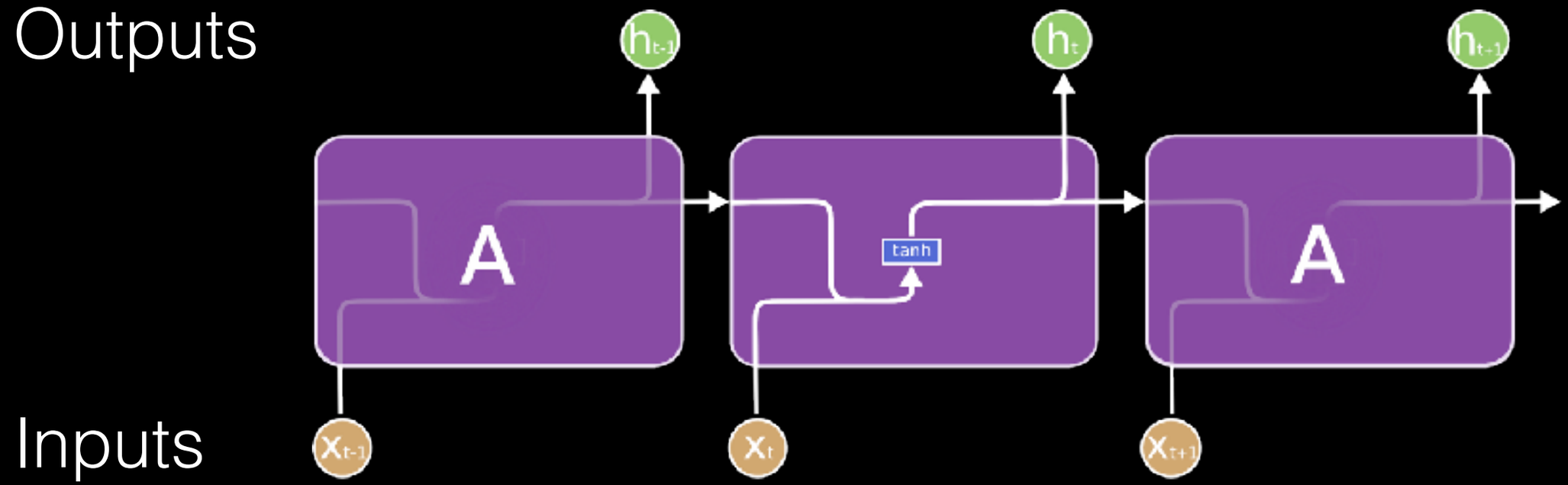
- “He grew up skating on the ice ponds of Canada and, like most kids his age, learned to play … at a very young age.”
- “Tom Brady, one of the greatest athletes of his generation, has now won 5 …”

Long-Term Dependencies



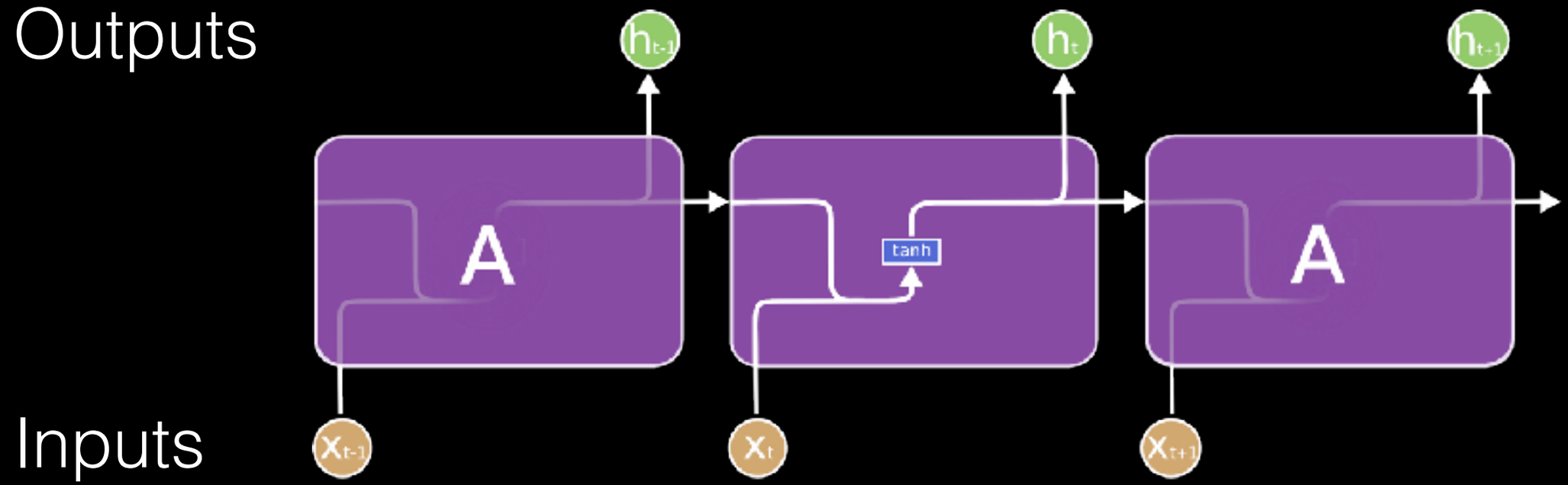
Learning long-term dependencies with standard RNNs are difficult.

Long-Term Dependencies



Learning long-term dependencies with standard RNNs is difficult.

Long-Term Dependencies

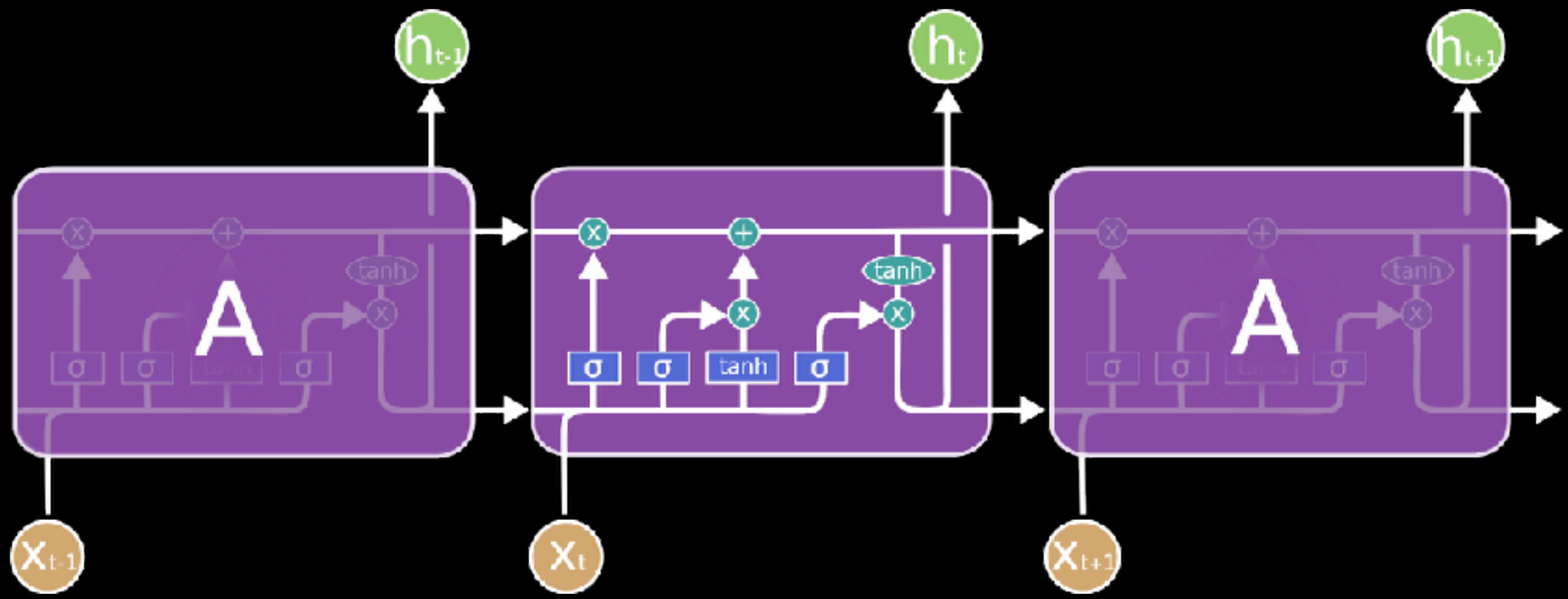


A single layer gates what to pass forward to the next time instance in standard RNN.

Long Short Term Memory (LSTM) Networks

LSTM Networks [Hochreiter and Schmidhuber, 1997]

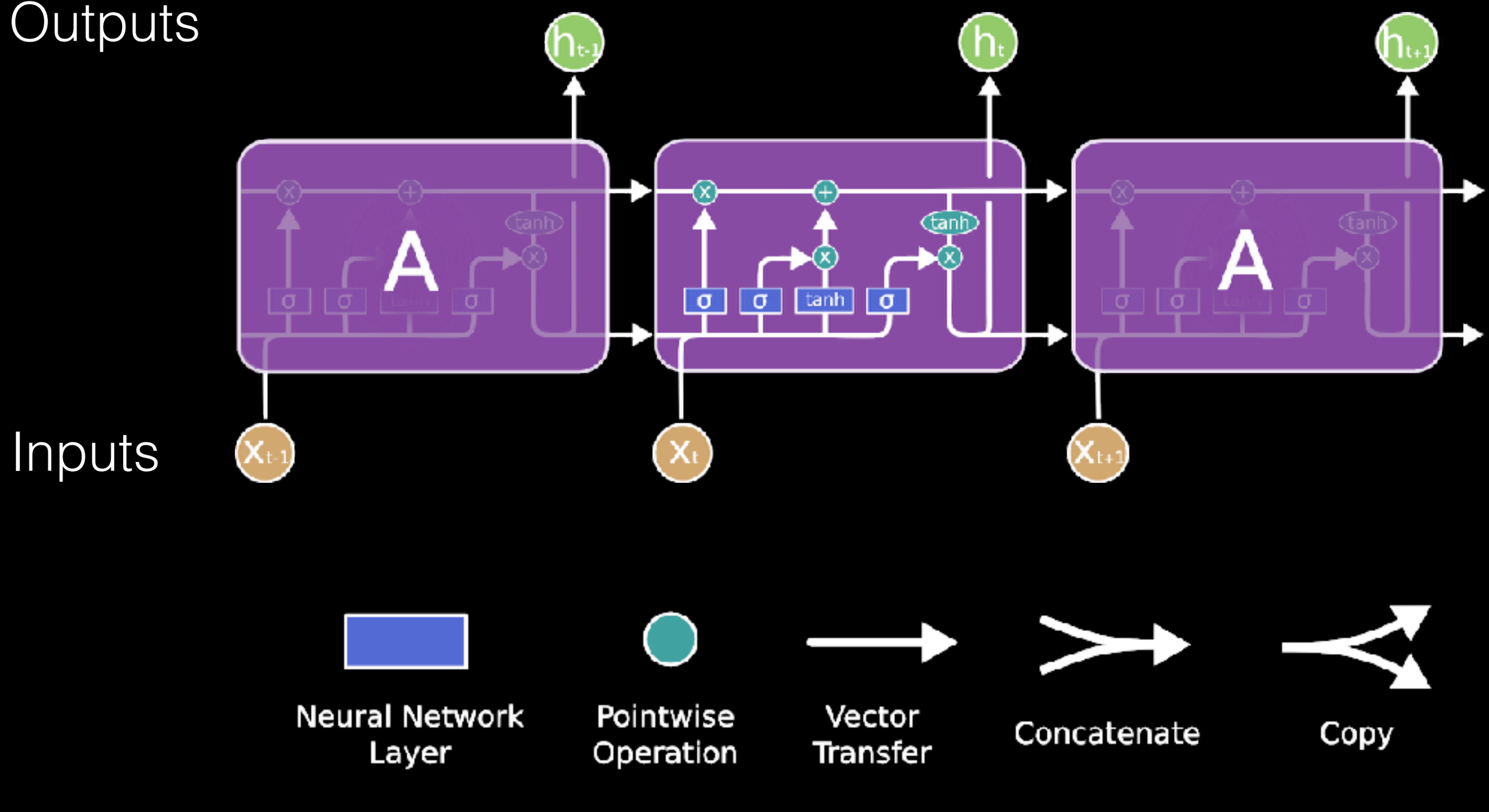
Outputs



Inputs

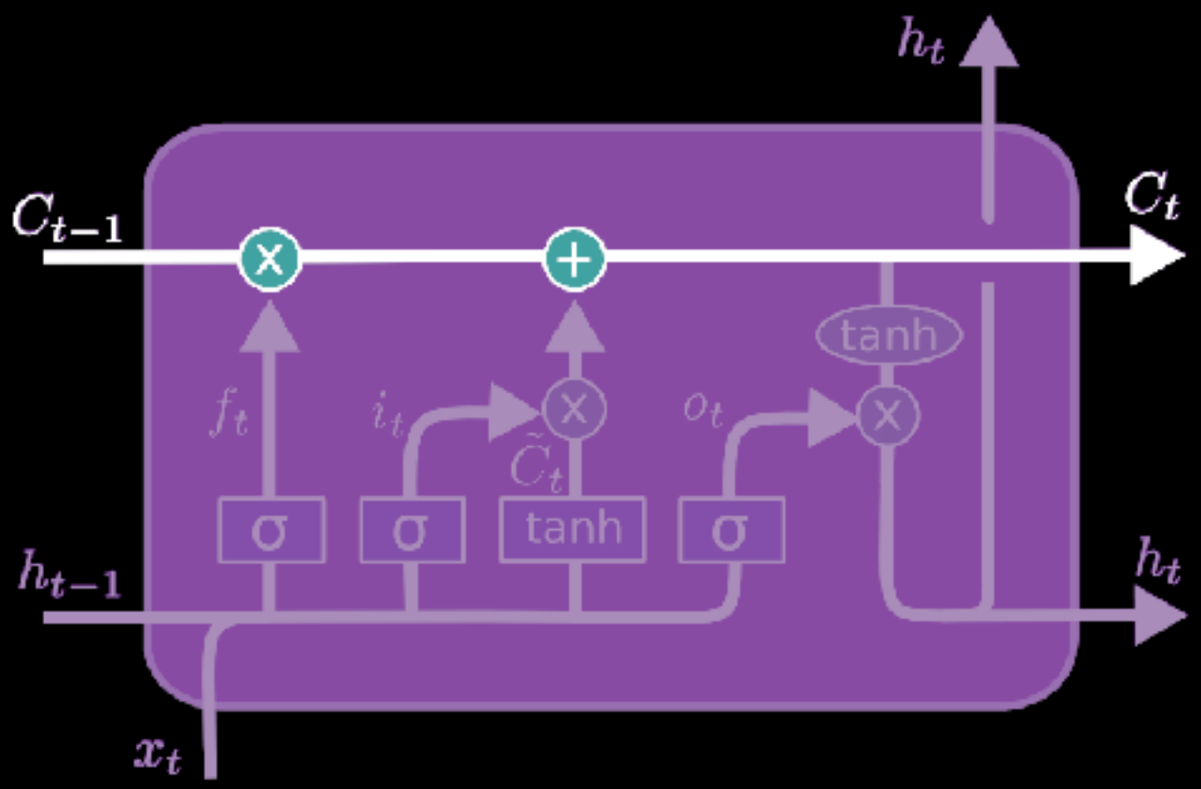
Four layers gate what to pass forward to next time instance in LSTM networks.

LSTM Networks



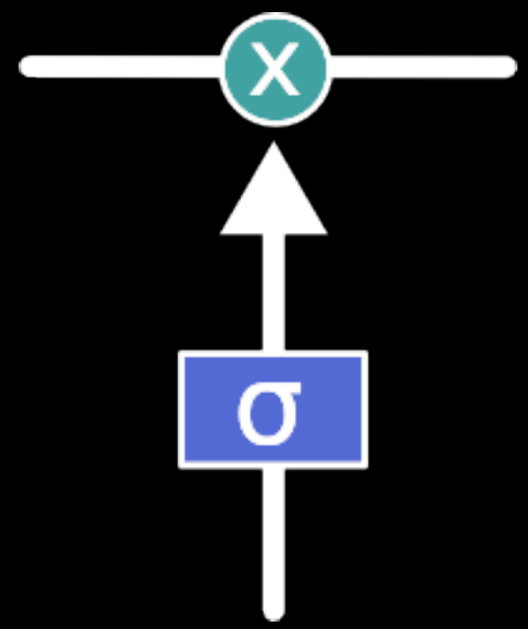
Let’s look at the different components that make up an LSTM.

Cell State



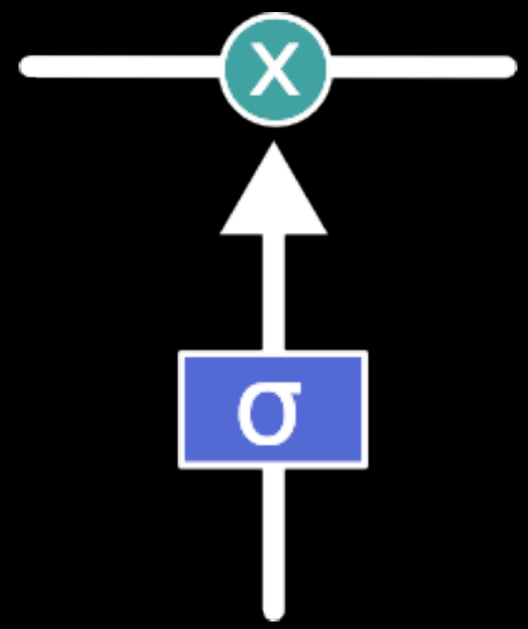
The cell state is directly carried forward from one time instance to the next — but with a number of modifications.

LSTM Gates



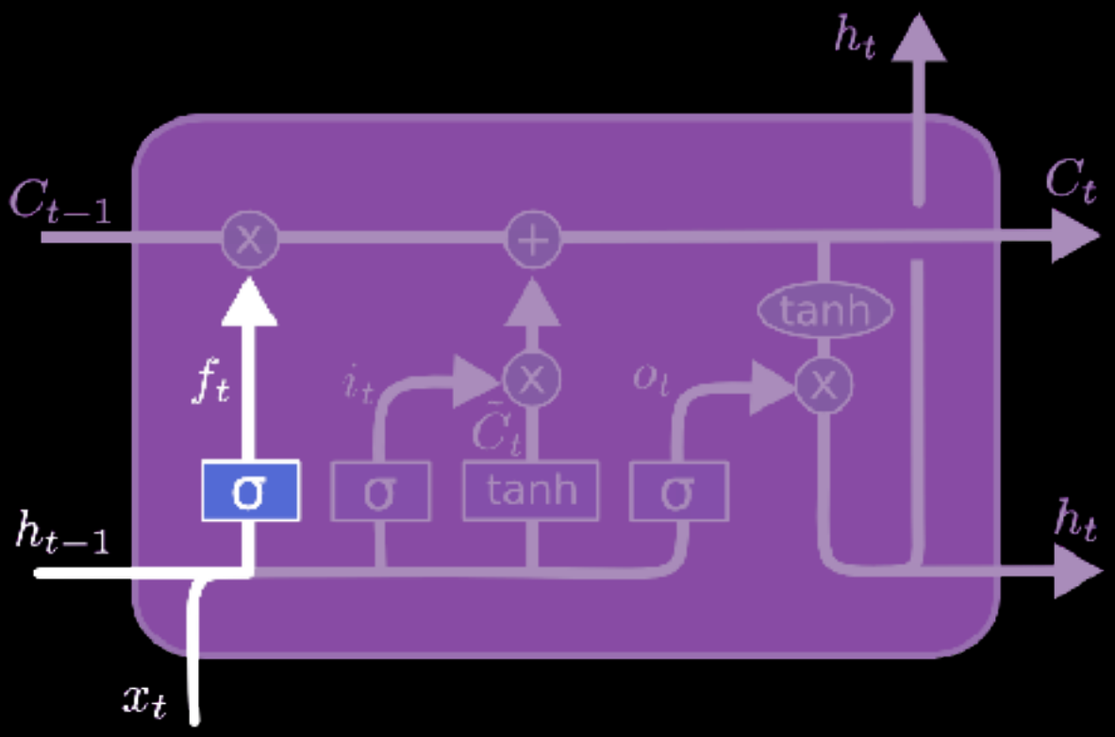
This is an LSTM “gate” which makes an element-wise decision on what to allow through. It is a fully connected layer followed by a sigmoid with outputs between 0 and 1.

LSTM Gates



Together with the “x” operation, this layer is a gate-keeper for the channel.

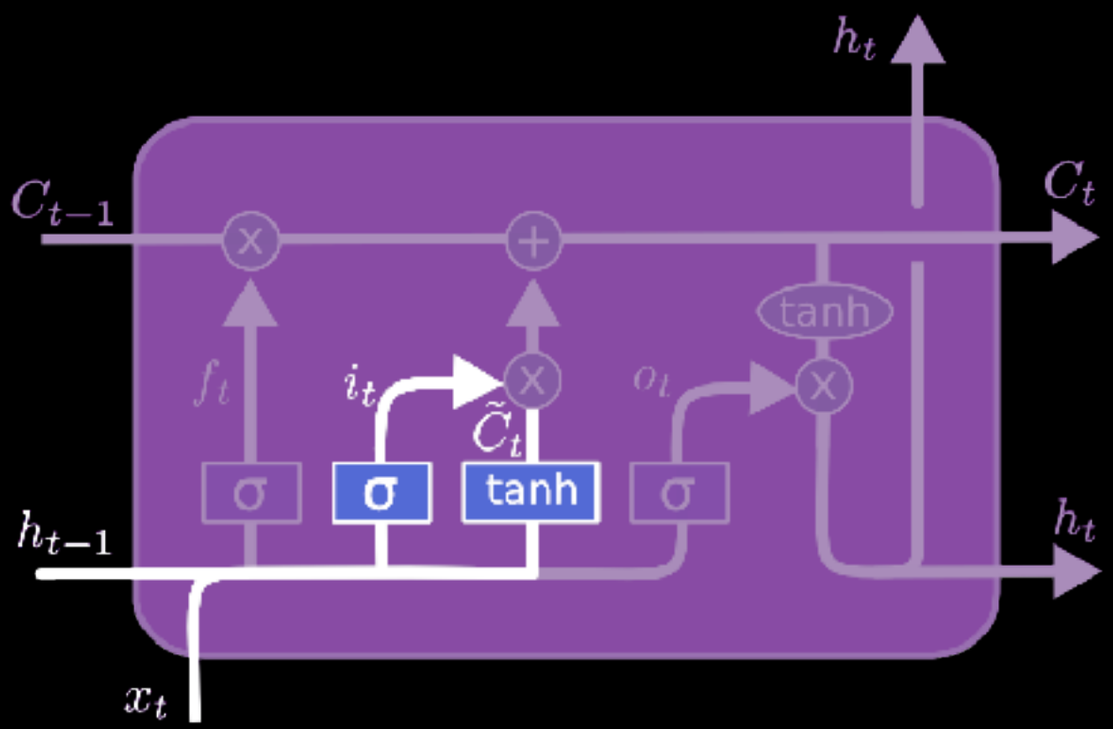
Forget Gate



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

The forget gate decides what elements of the previous cell state should be forgotten.

Input Gate

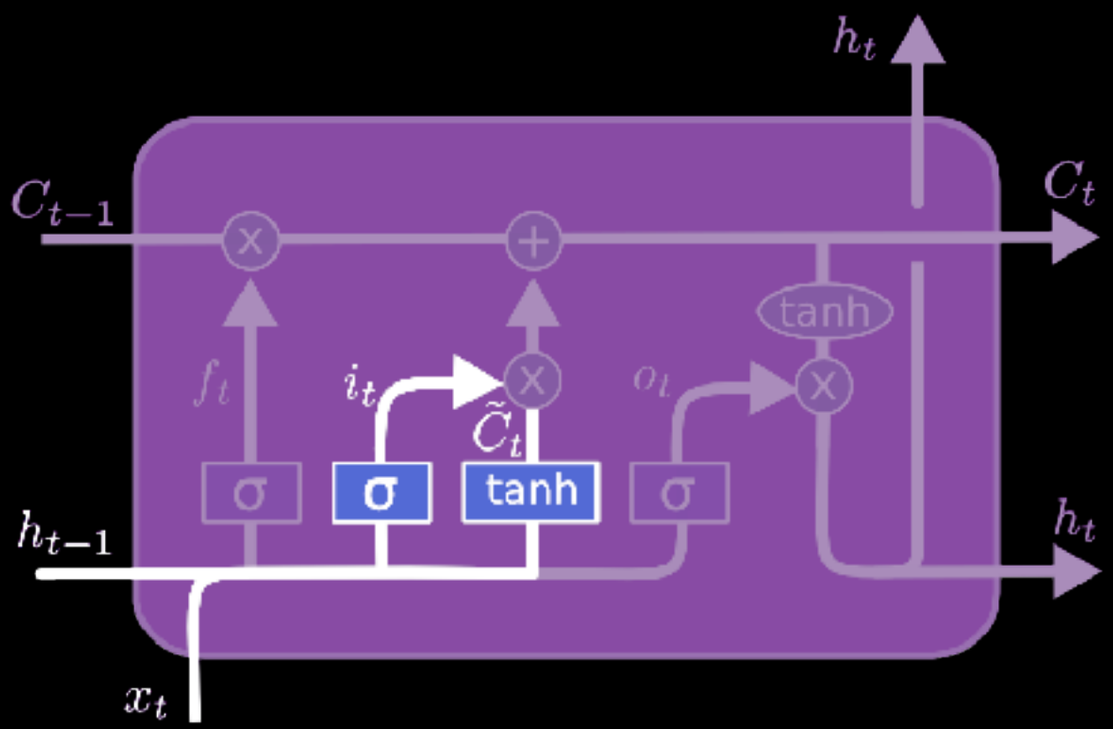


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

The input gate decides what new information should be added to the cell state.

Input Gate

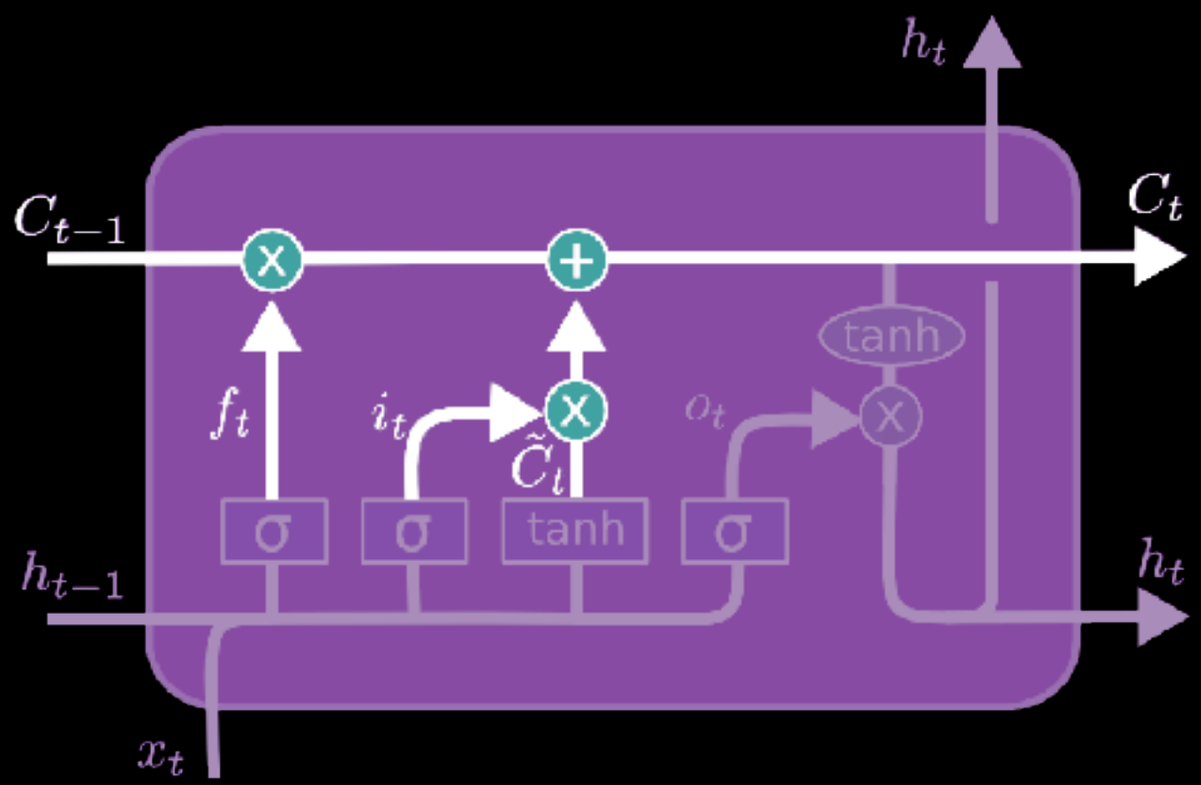


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

This new information is linearly transformed and then squashed by the ***tanh*** function which forces the output to lie between -1 and 1.

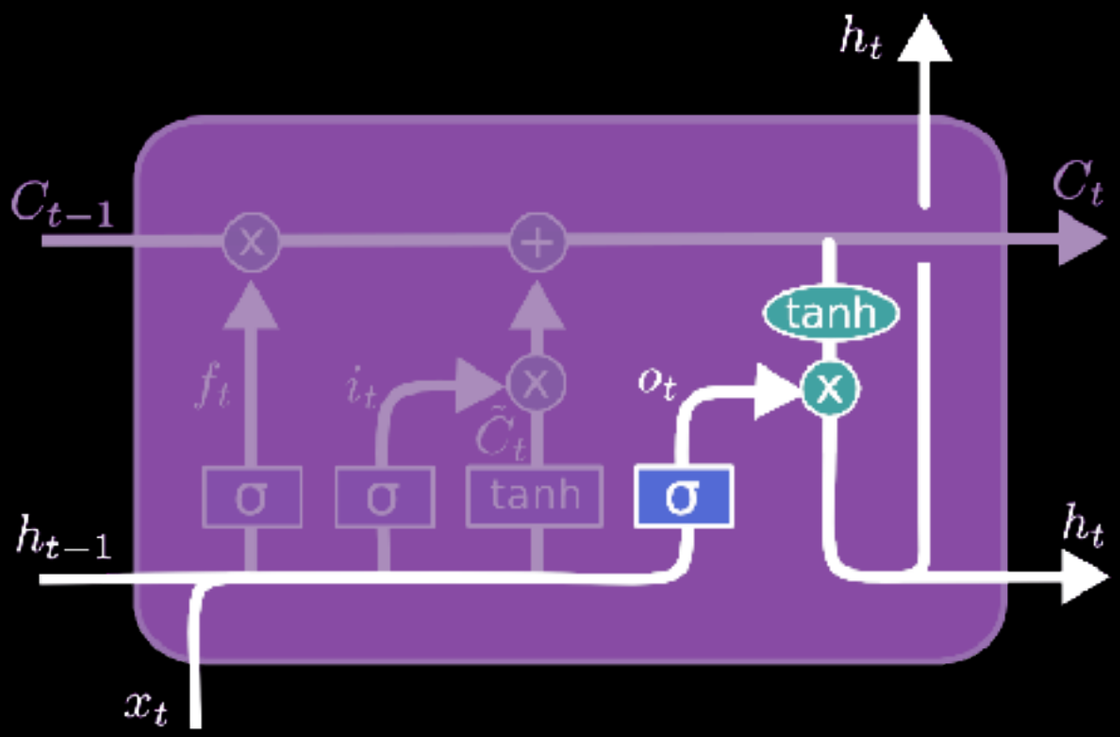
Cell State is updated



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Outputs from the forget gate and the input gate combine to update the current cell state.

Output Gate



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

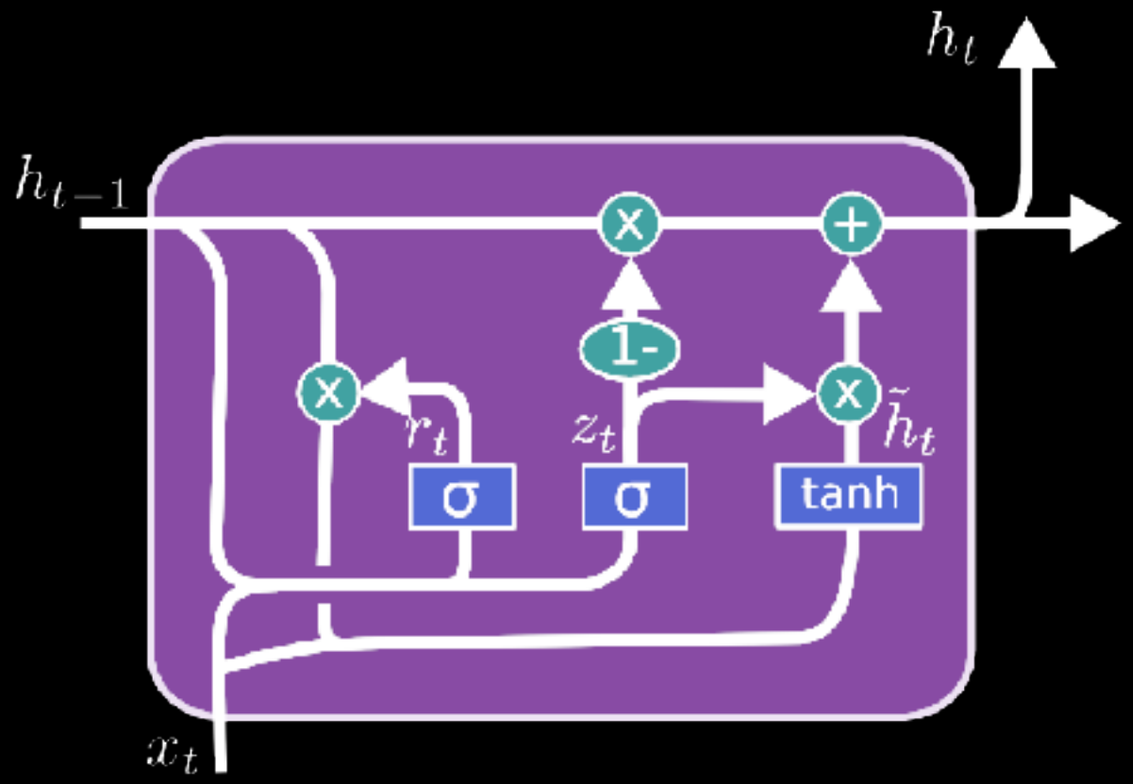
$$h_t = o_t * \tanh\left(C_t\right)$$

The output gate decides what information from the cell state should be output by the LSTM.

Naturally, there are lots of variations on this architecture…

Gated Recurrent Unit (GRU)

[Cho, et al., 2014]



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

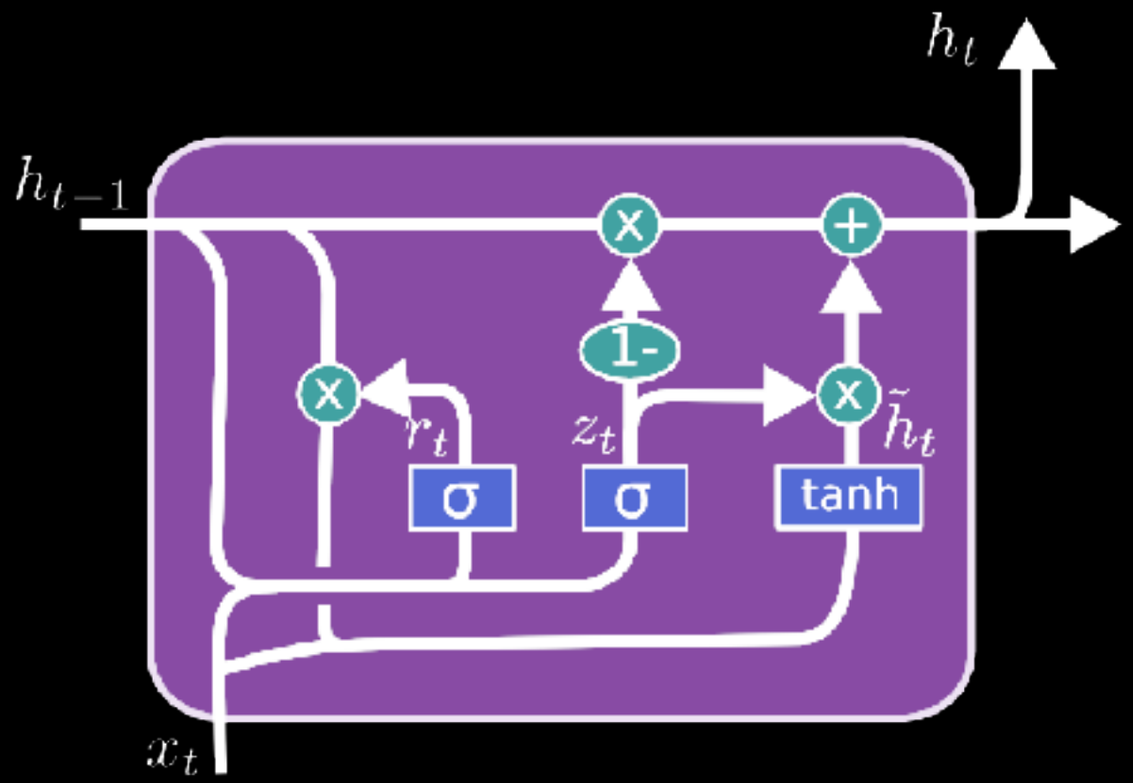
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Simpler variant of LSTM having one fewer layers so less parameters.

GRU



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

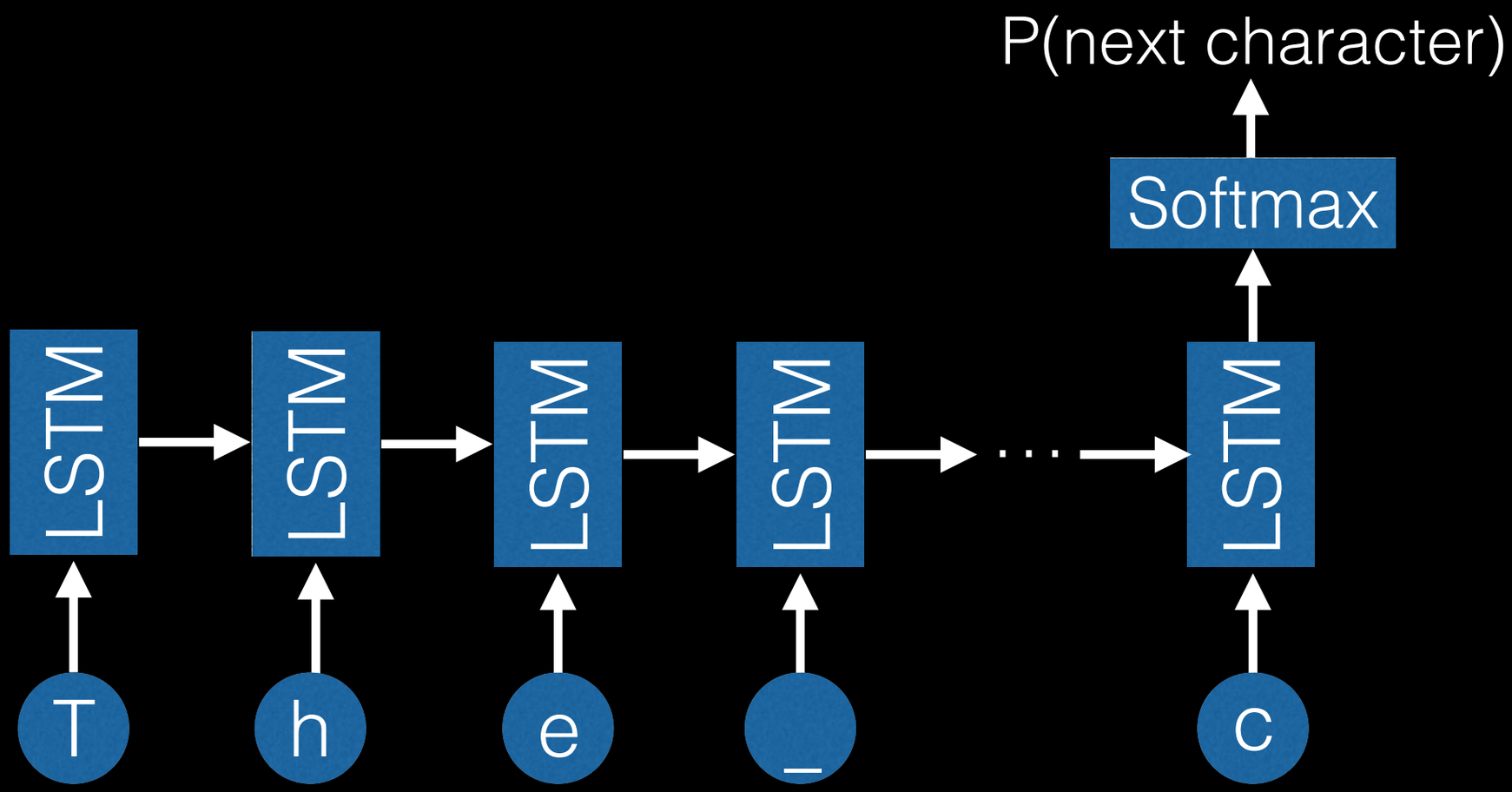
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Note that the cell state and output have been merged into one.

Character/Word Prediction



Given a sequence of characters we can learn to predict the next character with LSTM.

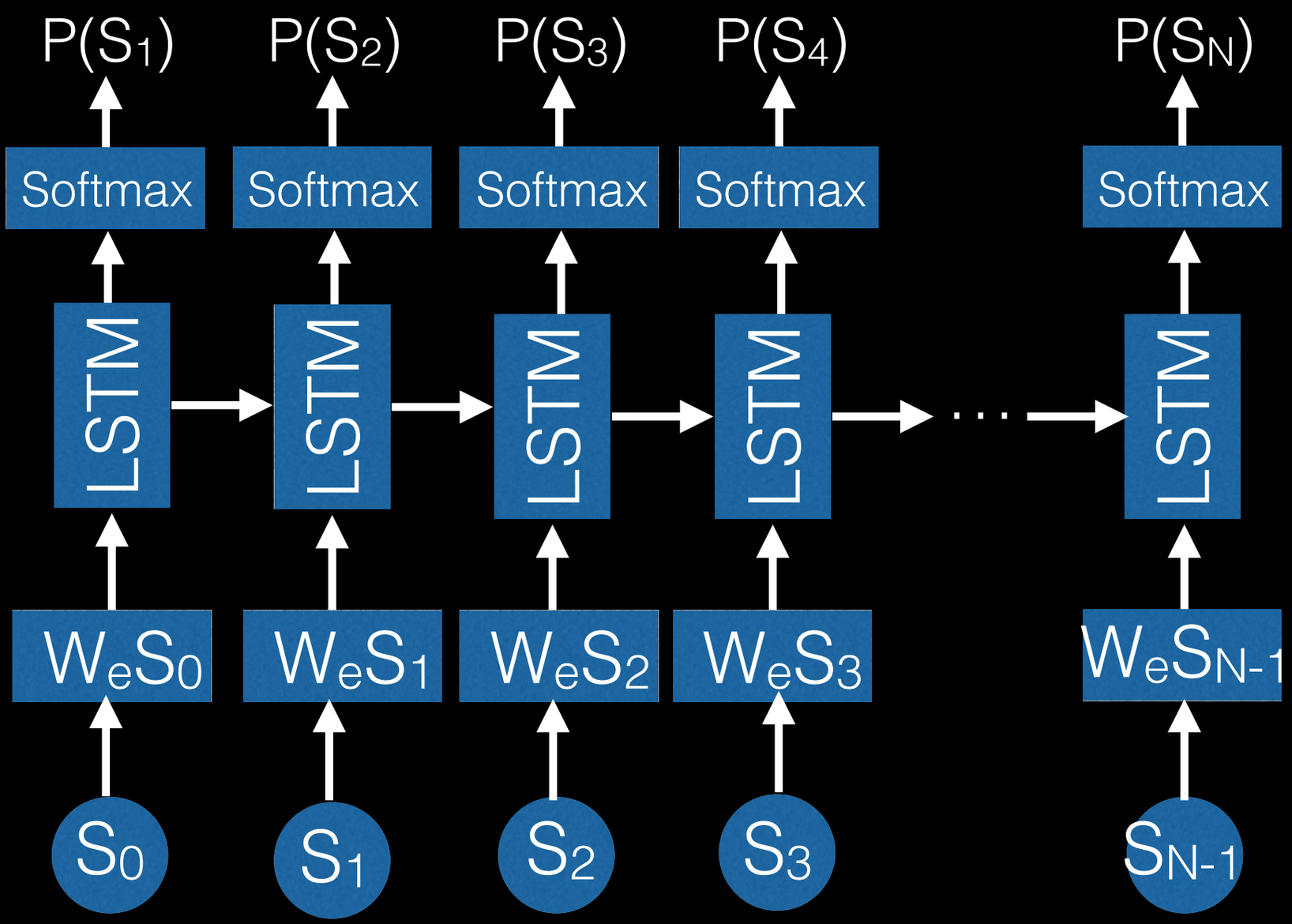
Beyond Good and Evil, F. Nietzsche

THE RELIGIOUS MOOD

The human soul and its limits, the range of man's inner experiences hitherto attained, the heights, depths, and distances of these experiences, the entire history of the soul UP TO THE PRESENT TIME, and its still unexhausted possibilities: this is the preordained hunting-domain for a born psychologist and lover of a "big hunt". But how often must he say despairingly to himself: "A single individual! alas, only a single individual! and this great forest, this virgin forest!" So he would like to have some hundreds of hunting assistants, and fine trained hounds, that he could send into the history of the human soul, to drive HIS game together. In vain: again and again he experiences, profoundly and bitterly, how difficult it is to find assistants and dogs for all the things that directly excite his curiosity. The evil of sending scholars into new and dangerous hunting-domains, where courage, sagacity, and subtlety in every sense are required, is that they are no longer serviceable just when the "BIG hunt," and also the great danger commences,—it is precisely then that they lose their keen eye and nose. In order, for instance, to divine and determine what sort of history the problem of KNOWLEDGE AND CONSCIENCE has hitherto had in the souls of homines religiosi, a person would perhaps himself have to possess as profound, as bruised, as immense an experience as the intellectual conscience of Pascal; and then he would still require that wide-spread heaven of clear, wicked spirituality, which, from above, would be able to oversee, arrange, and effectively formulize this mass of dangerous and painful experiences.—But who could do me this service! And who would have time to wait for such servants!—they evidently appear too rarely, they are so improbable at all times! Eventually one must do everything ONESELF in order to know something; which means that one has MUCH to do!—But a curiosity like mine is once for all the most agreeable of vices—pardon me! I mean to say that the love of truth has its reward in heaven, and already upon earth.

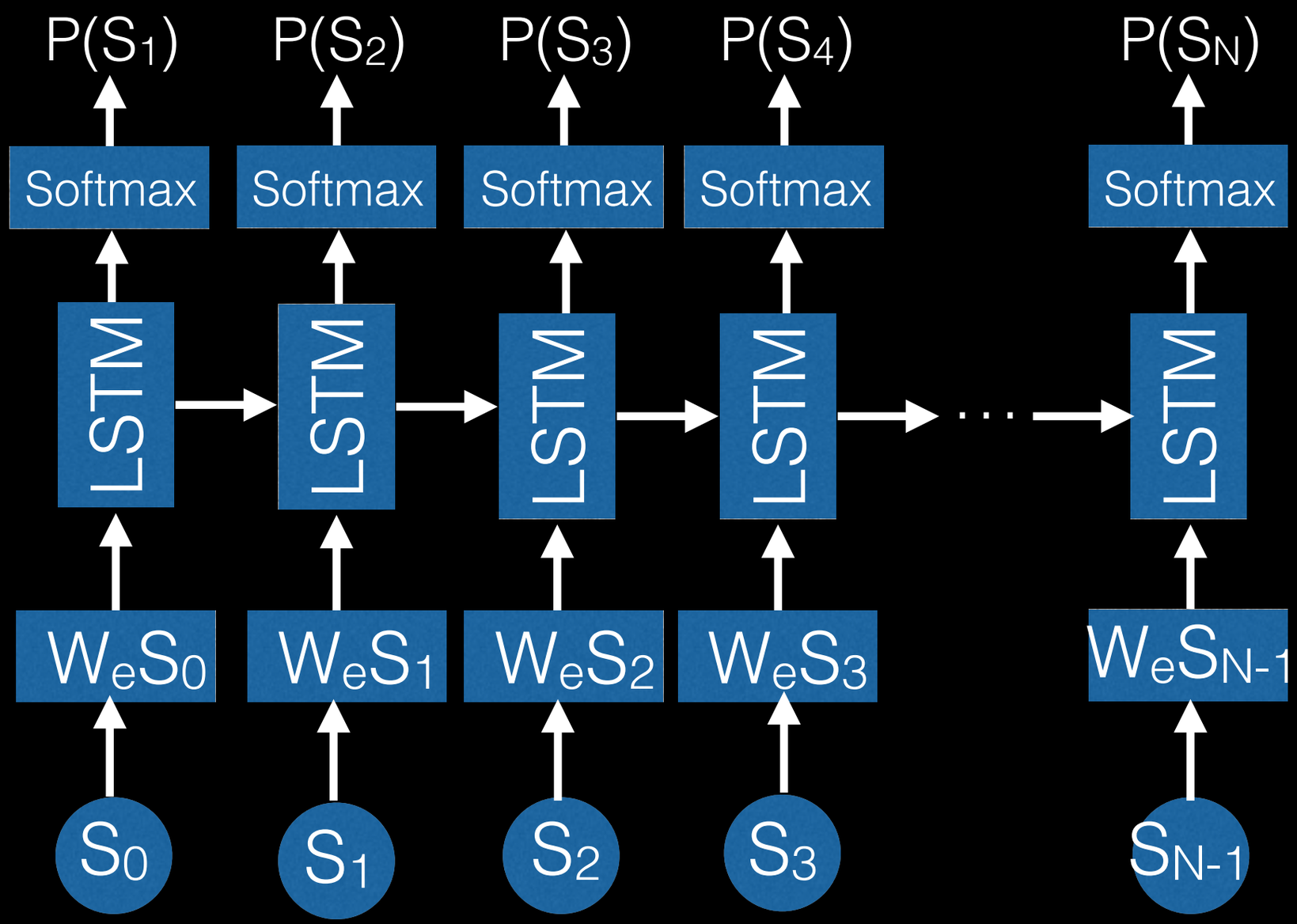
Now let’s try to learn sentences…

Word/Sentence Prediction



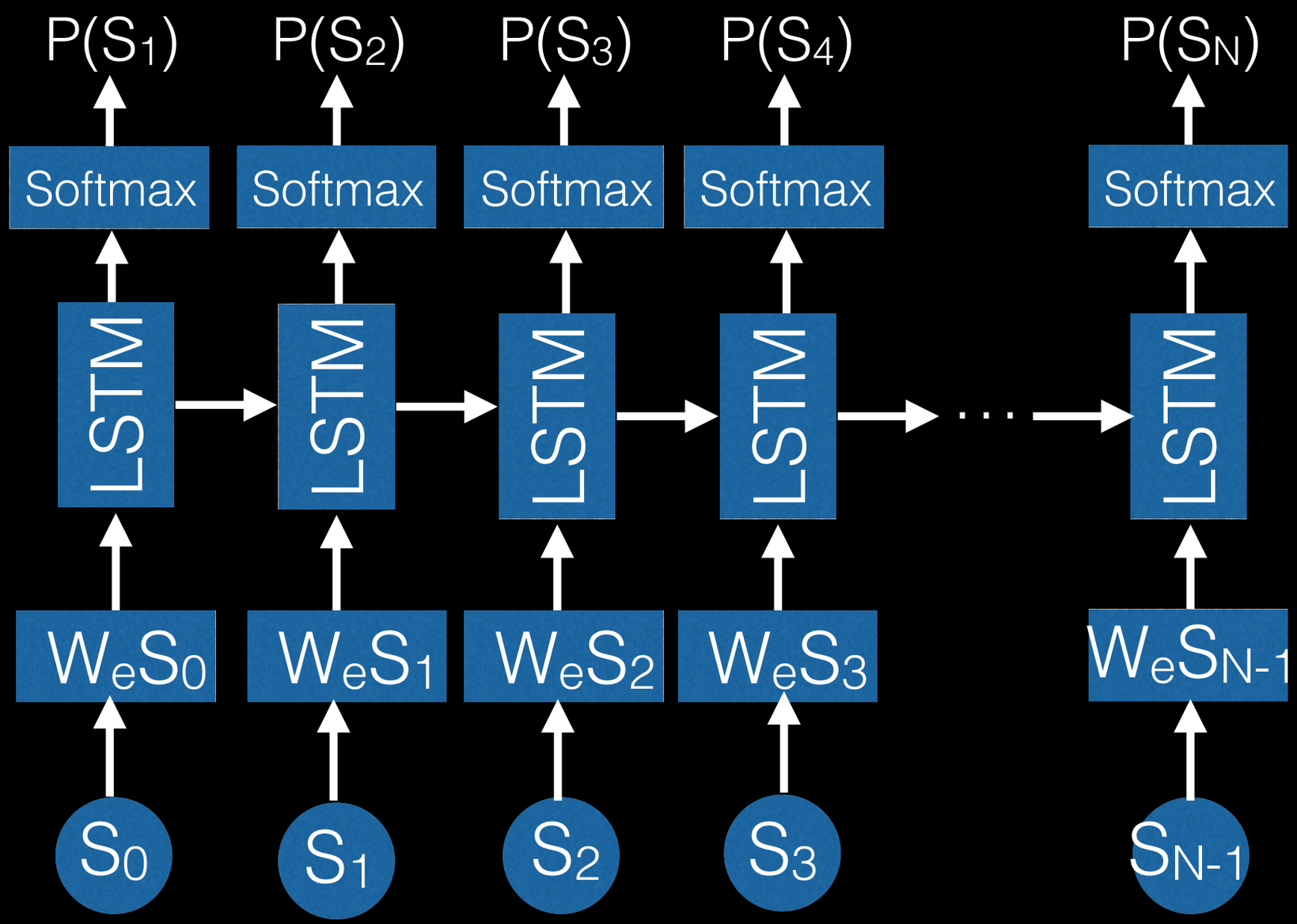
S_i are one-hot vectors for words in our dictionary.

Sentence Prediction



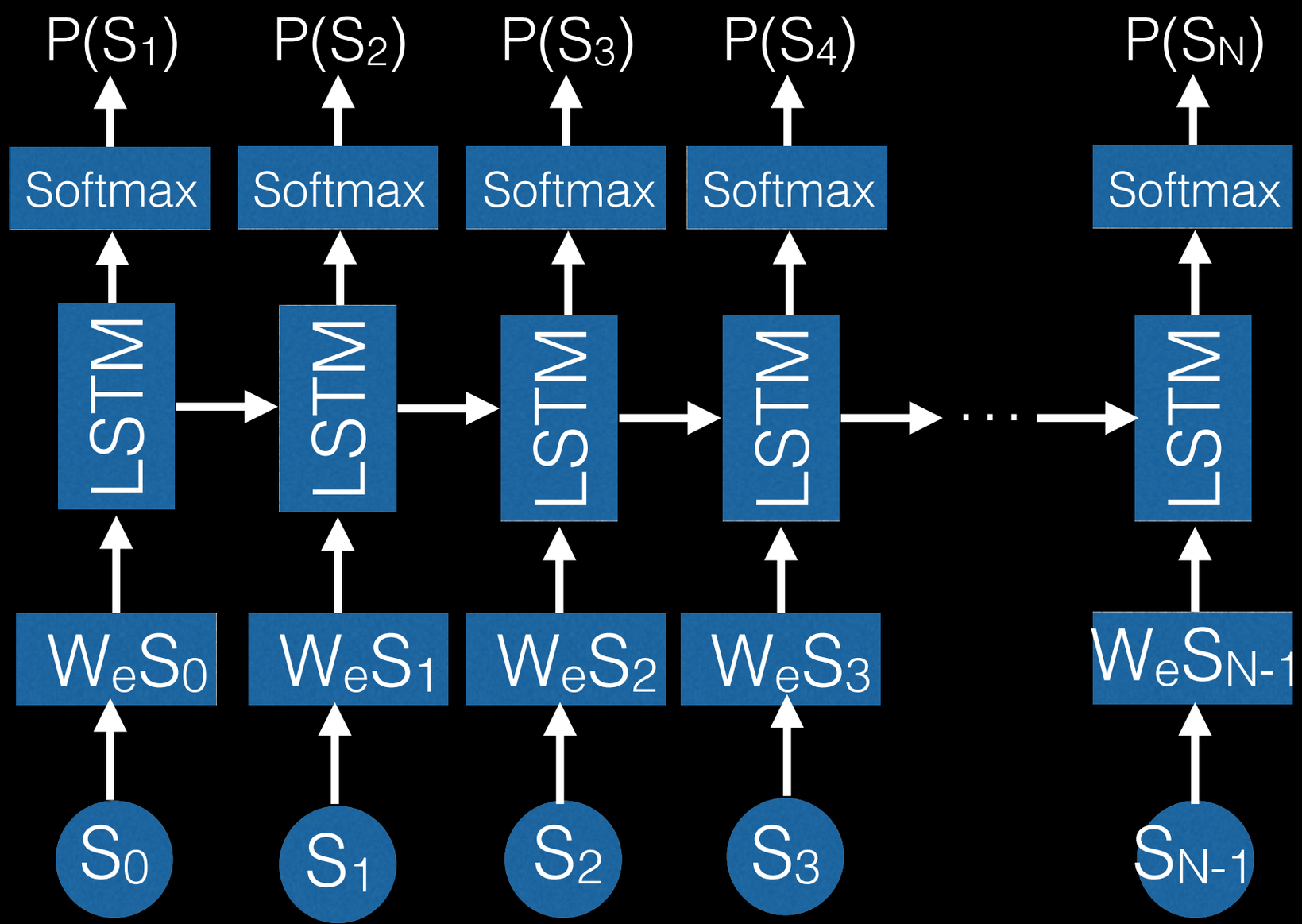
W_e is learned word embedding, a matrix that maps into a “word space.”

Sentence Prediction



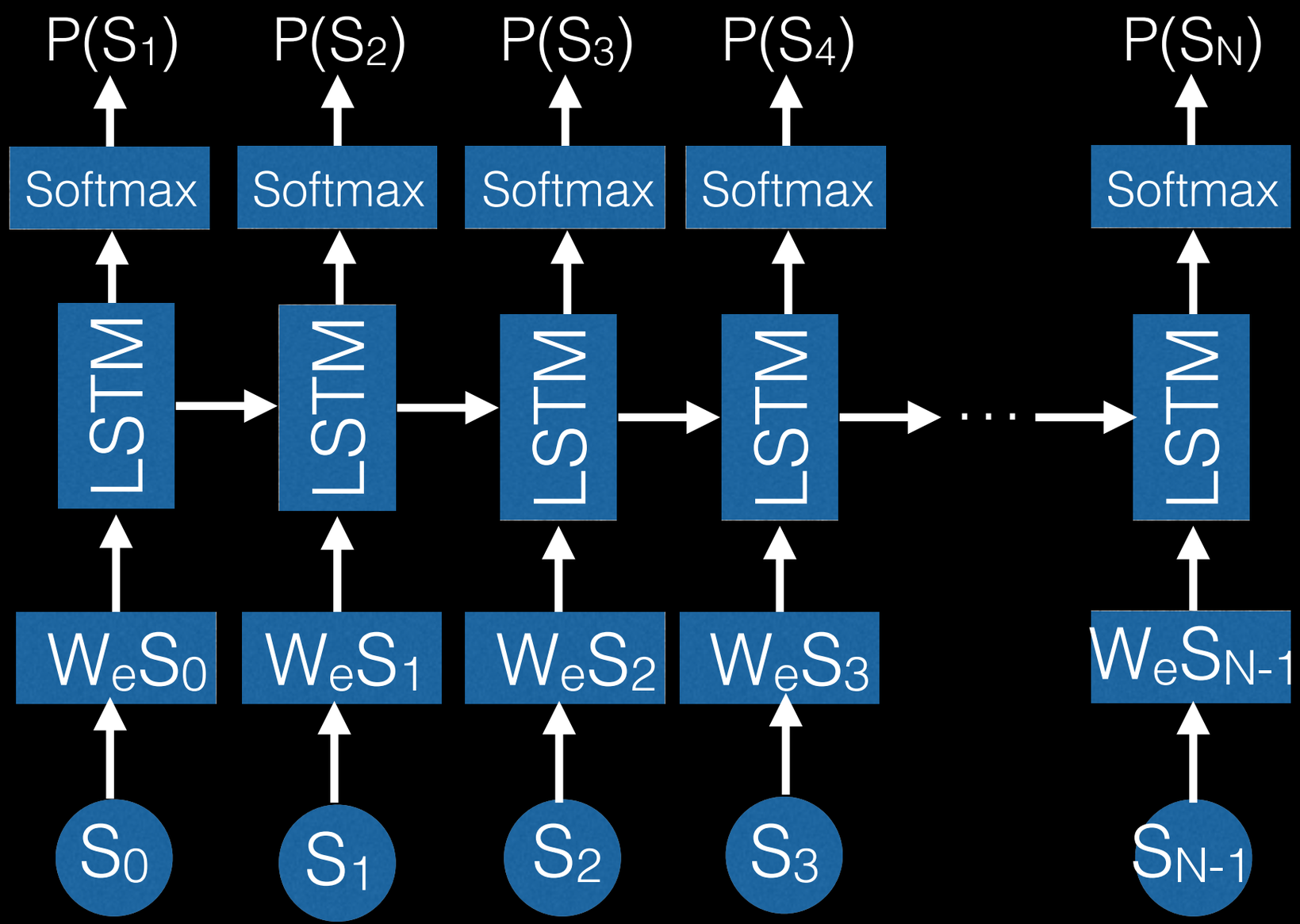
We could feed this network sentences and…

Sentence Prediction



… and learn to predict next words by …

Sentence Prediction

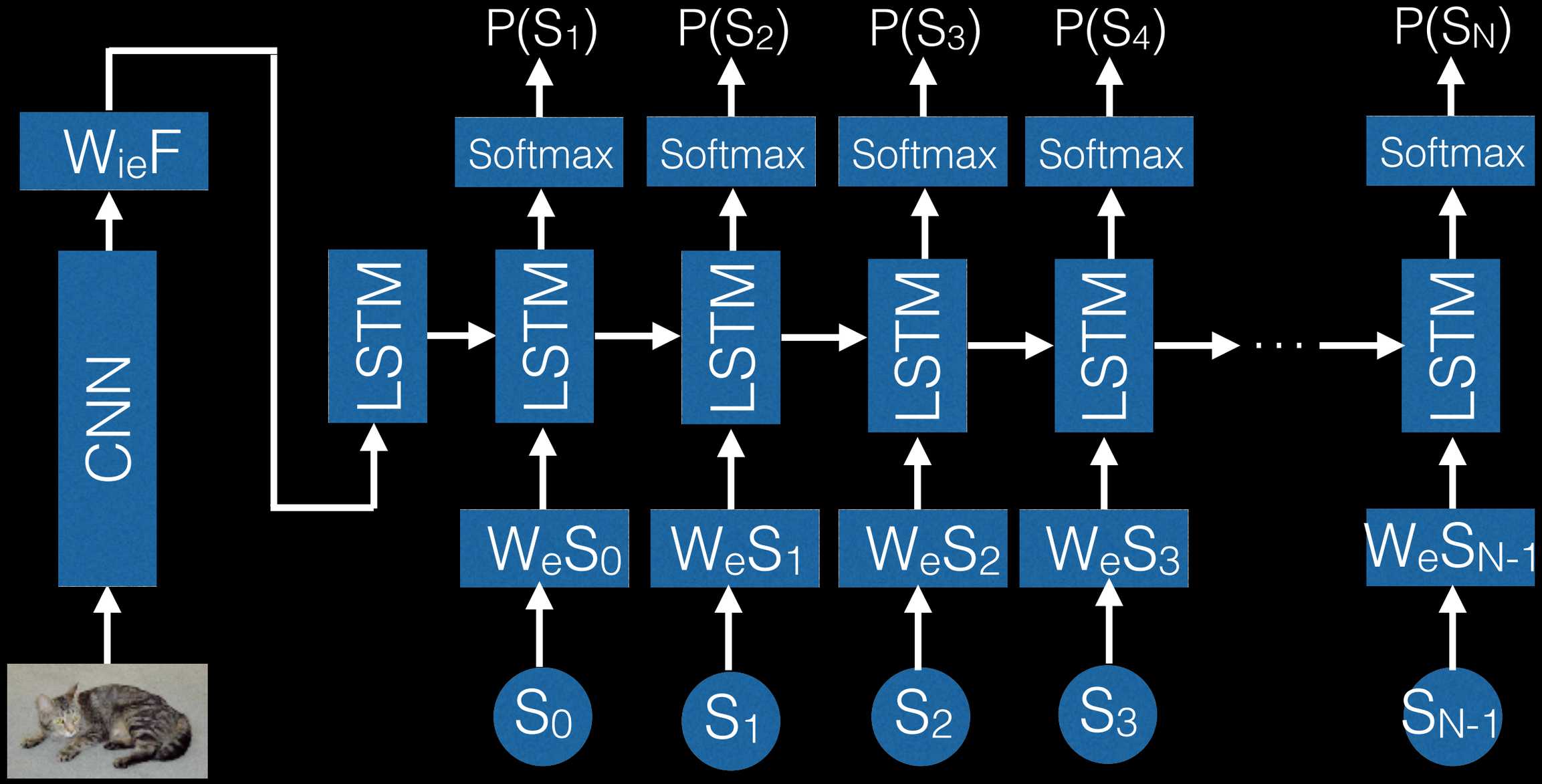


Minimize the loss = $-\log p(S) = \sum_{t=0}^{N} \log p(S_t|S_0, ..., S_{t-1})$

What about learning phrases or sentences that described images?!

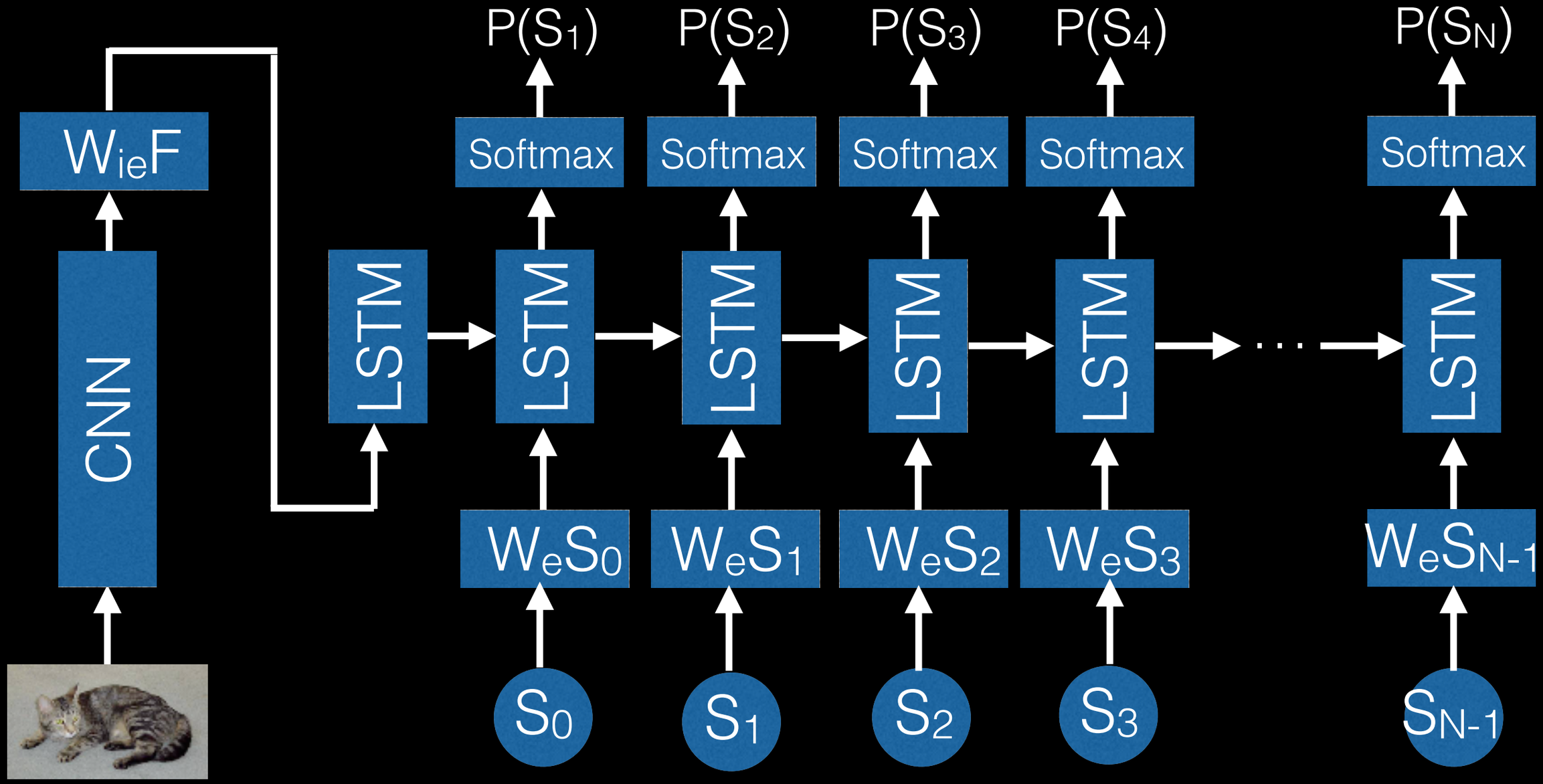
Let’s say we had a dataset of captioned images, how could we use this architecture to do this?

Image Captioning



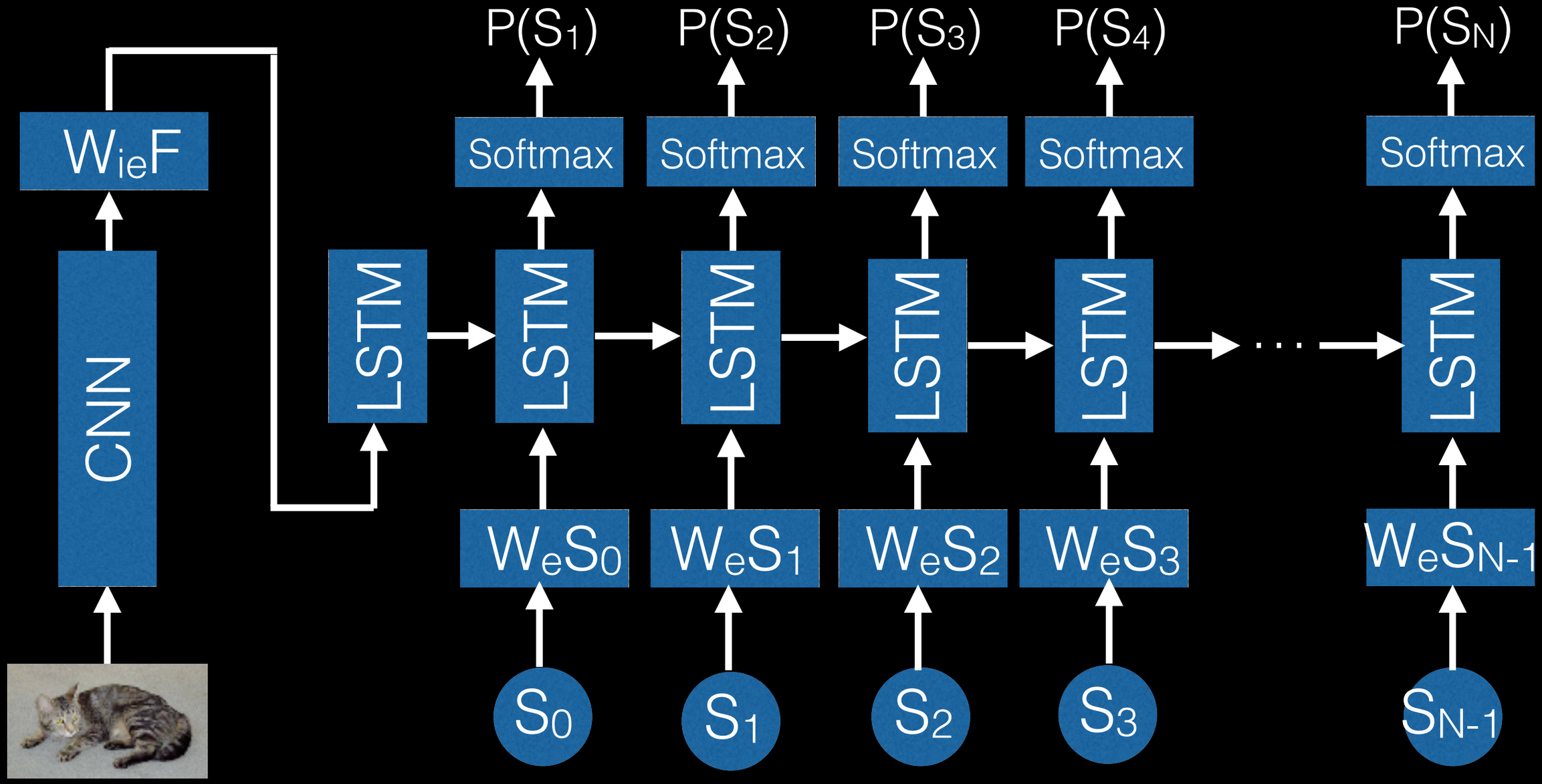
W_{ie} is an image embedding and F are features of the CNN.

Image Captioning



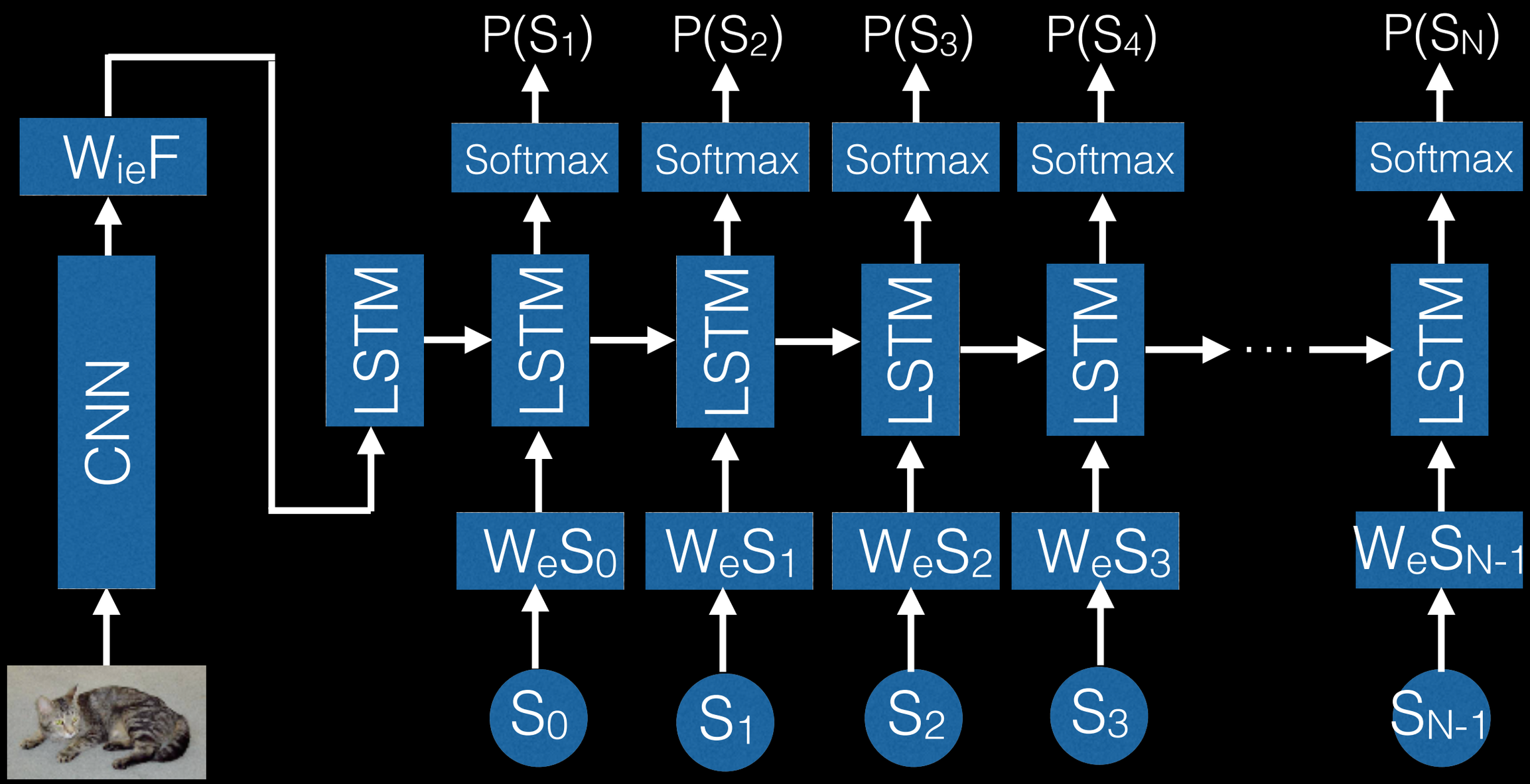
CAPTION: “A striped cat is lying on the ground.”

Image Captioning



S_0 = "->" S_1 = "A" S_2 = "striped" S_3 = "cat" S_{N-1} = "<-"

Image Captioning



Minimize the loss = $-\log p(S|I) = \sum_{t=0}^{N} \log p(S_t|I, S_0, ..., S_{t-1})$

What are the results like?

https://arxiv.org/pdf/1609.06647.pdf

https://research.googleblog.com/2016/09/show-and-tell-image-captioning-open.html