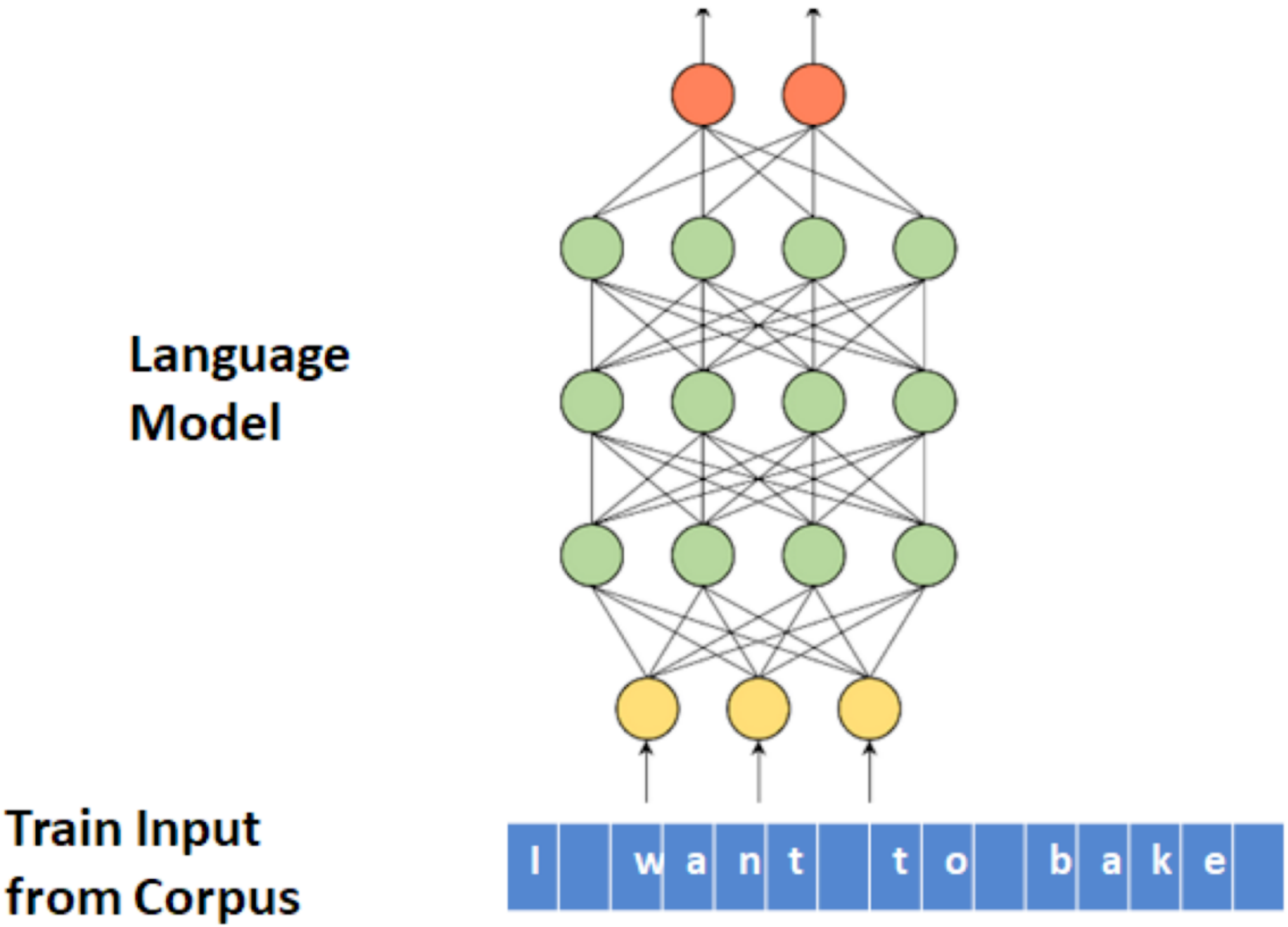
Helping Users with Text Generation

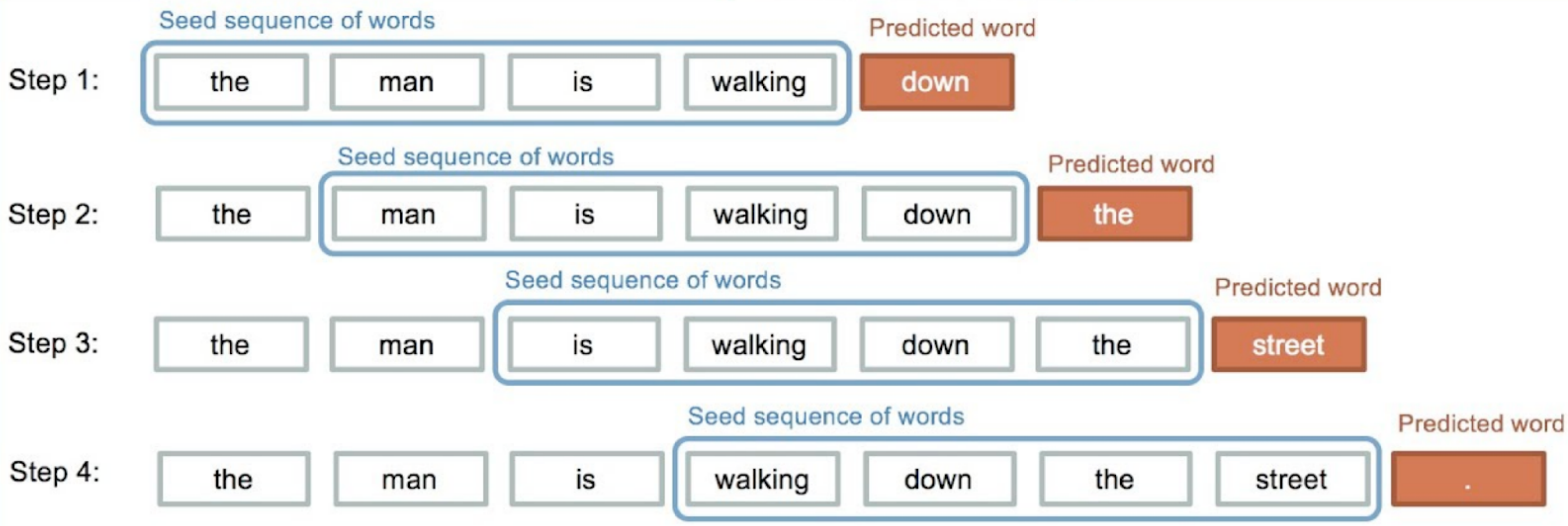
Axel Sirota
AI and Cloud Consultant

@AxelSirota

Probabilities over char set

	a	b	c	d	e	f	g	...	z
0.01	0.02	0.36	0.25	0.02	0.001	0.22	0.001	...	0.06





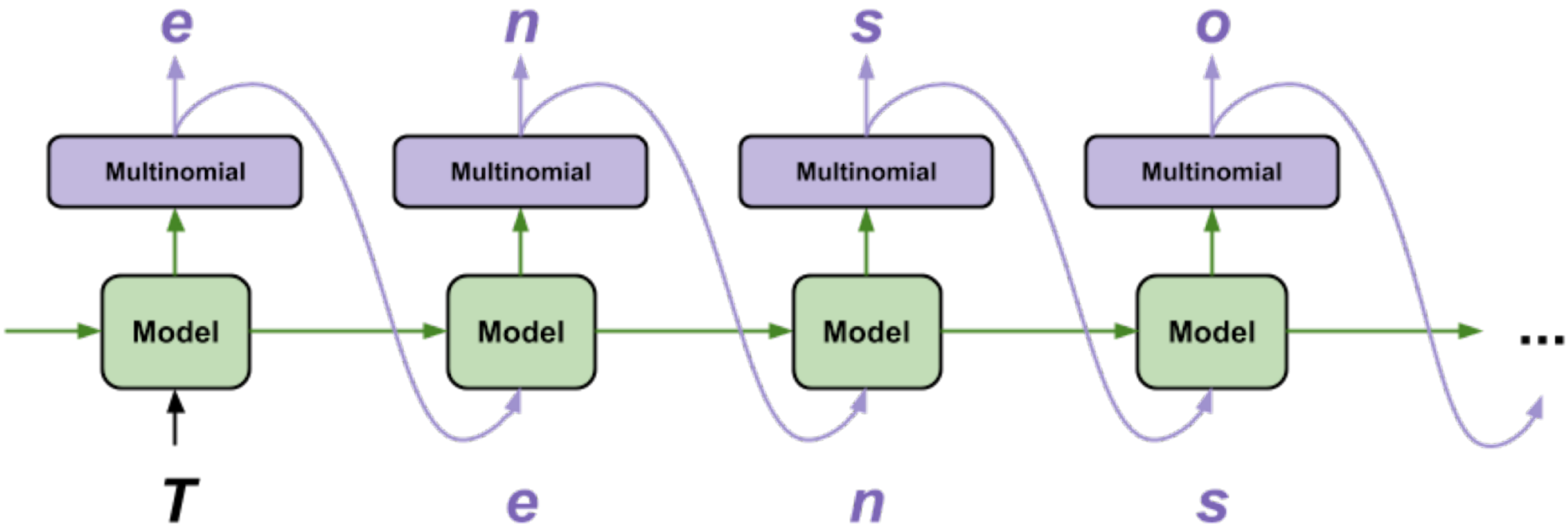
Seed sequence of words
Predicted word
Step 1:
the
man
is
walking
down
Seed sequence of words
Predicted word
Step 2:
the
man
is
walking
down
the
Seed sequence of words
Predicted word
Step 3:
the
man
is
walking
down
the
street
Seed sequence of words
Predicted word
Step 4:
the
man
is
walking
down
the
street
.

Problems??

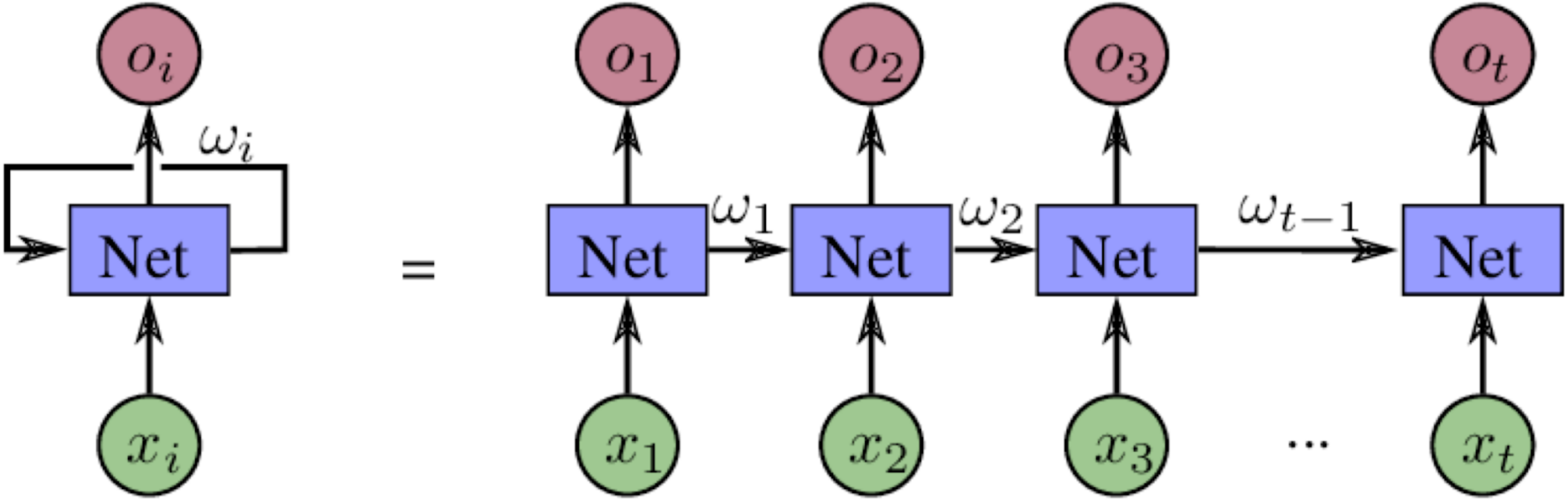
Predicting words needs many parameters

To lose the long-term relationships:

- Local high school dropouts cut in half
- 7-foot doctors sue hospitals
- I am a big metal fan


e
n
s
o
Multinomial
Multinomial
Multinomial
Multinomial
Model
Model
Model
Model
...
T
e
n
s

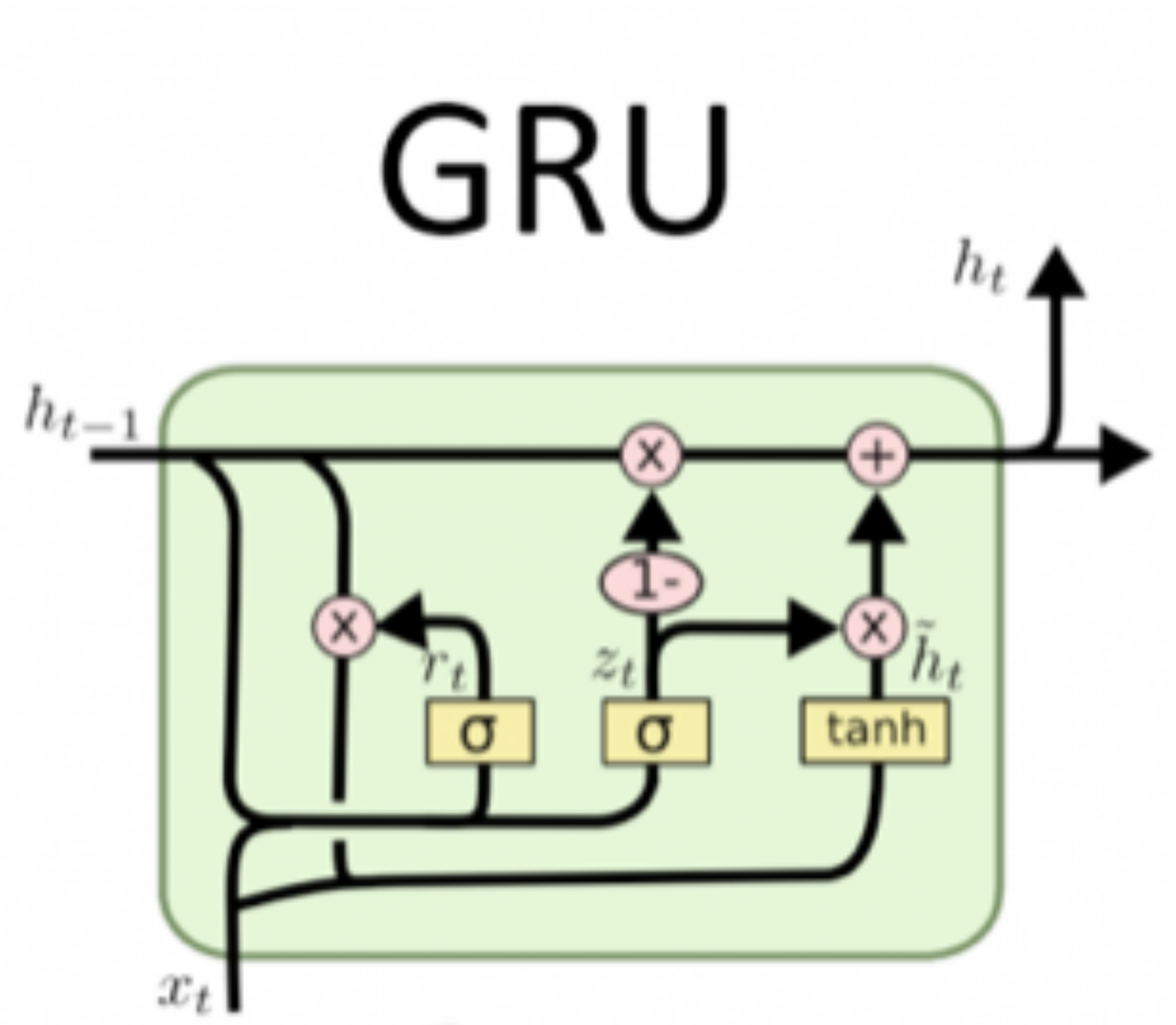
Introducing RNN and LSTM



o_i
ω_i
Net
x_i
=
o_1
o_2
o_3
o_t
ω_1
ω_2
ω_{t-1}
Net
Net
Net
Net
x_1
x_2
x_3
...
x_t

As the network grows, the gradient of the first steps are closer to zero, which makes the whole gradient of the network closer to zero due to the chain rule

GRU Diagram

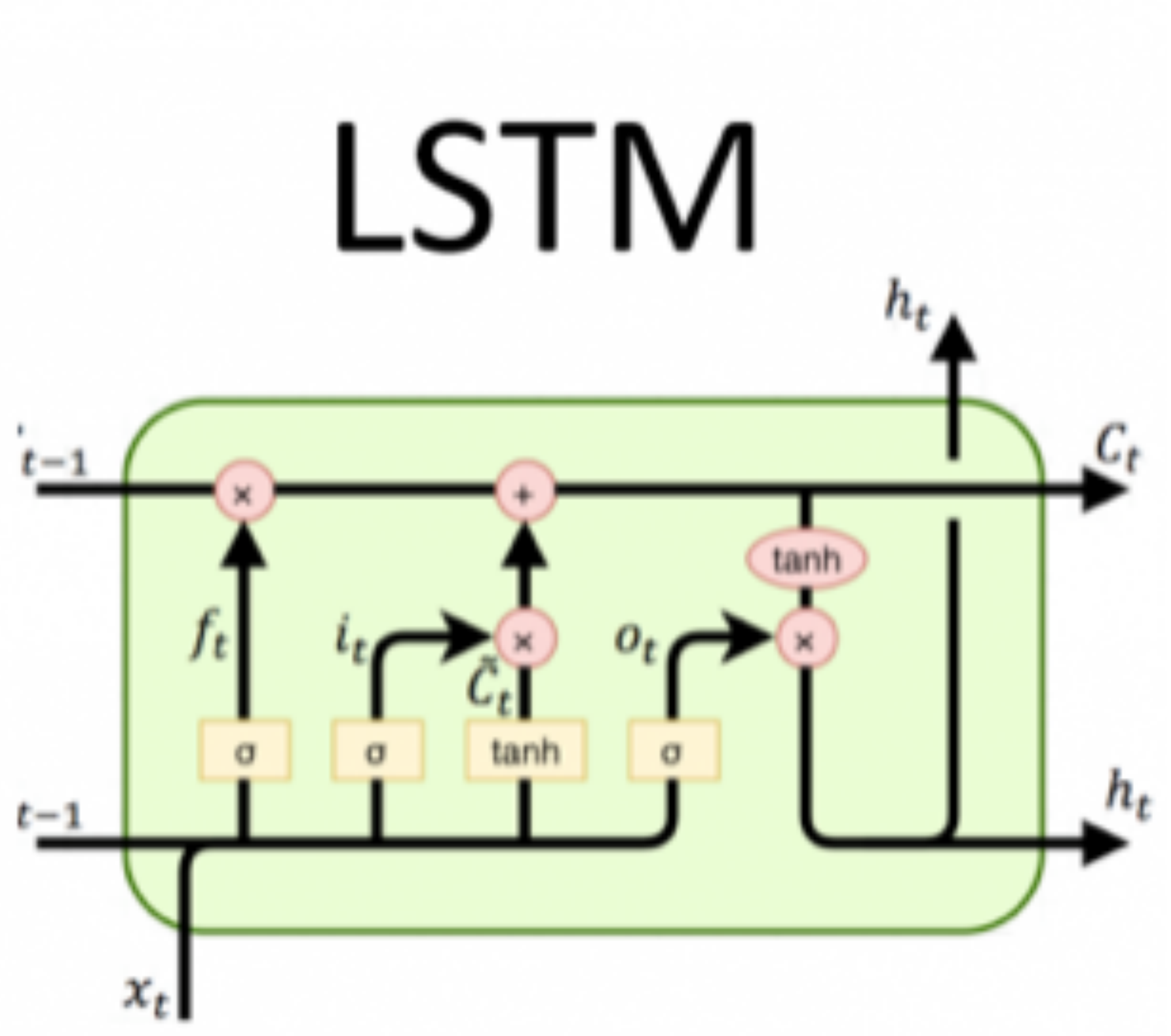


h_{t-1} represents memory in the previous cell

x_t represents input at time t

h_t represents memory in time t

LSTM Diagram



This hidden vector h from always

The output vector as always

New memory vector C to capture long-term relationships

Training a character-based text generation model

Takeaways

Recurrent networks solve the issue of fixed size window sizes for variable length text

The two most common ones are GRUs and LSTMs

Both pass a hidden state vector weighting the input vs previous outputs; but LSTMs also have a proper memory state.

Keys

- Ensure to understand how the secret vector and memory passes through the units in each scenario
- Check the OneStep model and make sure you understand each step
- Do the homework with LSTMs

Where to Go Next?

Google: Sequence Models for Time Series and Natural Language Processing on Google Cloud

“AI and Machine Learning for Coders: A Programmer's Guide to Artificial Intelligence” by Laurence Moroney

“Deep Learning”, by Ian Goodfellow, Yoshua Bengio and Aaron Courville