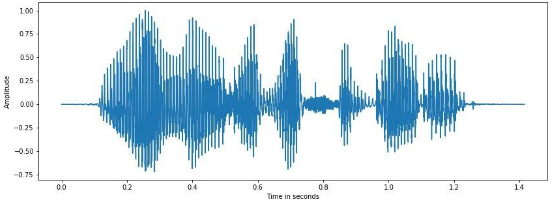
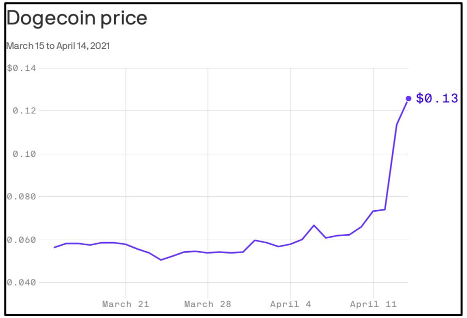
11-785: Lab 7 (Spring 24)
RNN Basics

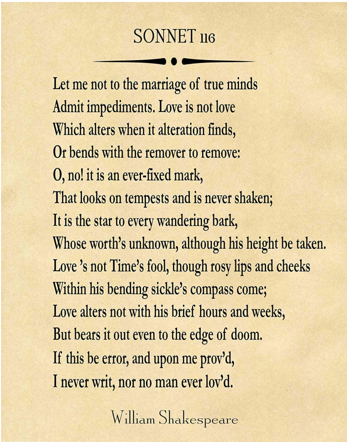
Harshit, Miya

Sequential Data

- Data from which various inputs are dependent
- Examples:
    - Text: *“Hi. How are you doing today?”*
    - Audio/speech
    - Video
    - Any other time series data like stock price, daily temperature, etc.





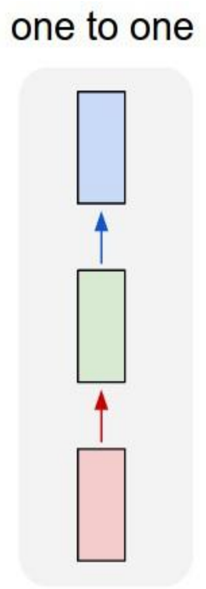

SONNET 116

Let me not to the marriage of true minds
Admit impediments. Love is not love
Which alters when it alteration finds,
Or bends with the remover to remove:
O, no! it is an ever-fixed mark,
That looks on tempests and is never shaken;
It is the star to every wandering bark,
Whose worth's unknown, although his height be taken.
Love 's not Time's fool, though rosy lips and cheeks
Within his bending sickle's compass come;
Love alters not with his brief hours and weeks,
But bears it out even to the edge of doom.
If this be error, and upon me prov'd,
I never writ, nor no man ever lov'd.

William Shakespeare

Reference: [Audio](), [Stock](), [Text](), [Video]()

Data Modeling



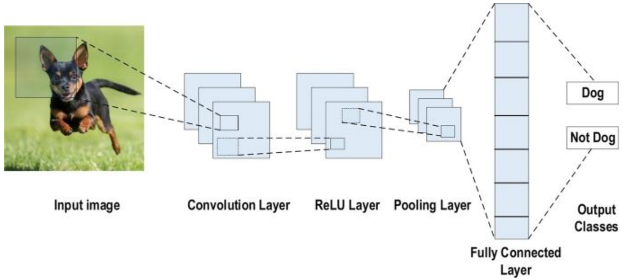


Image Classification (ref)

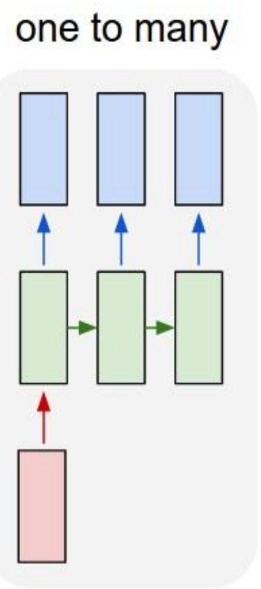
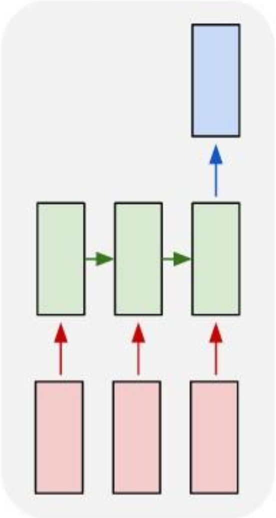


Image Captioning (ref)

(https://i.stack.imgur.com/b4sus.jpg)

Data Modeling

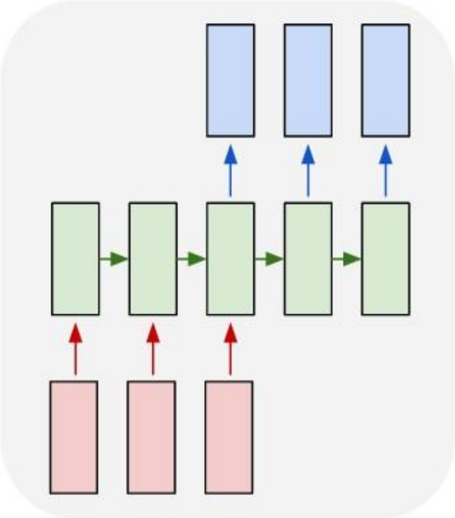
many to one



Sentiment Analysis (Movie Review)

The Batman (2022) is everything a superhero movie should be. **(Positive)**

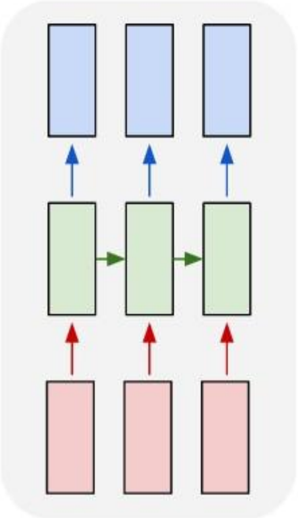
many to many



Machine Translation

“How are you?” -> “எப்படி இருக்கிறீர்கள்?"

many to many

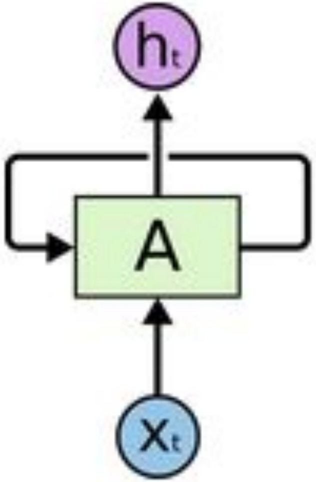


Object Tracking in videos

[Video]

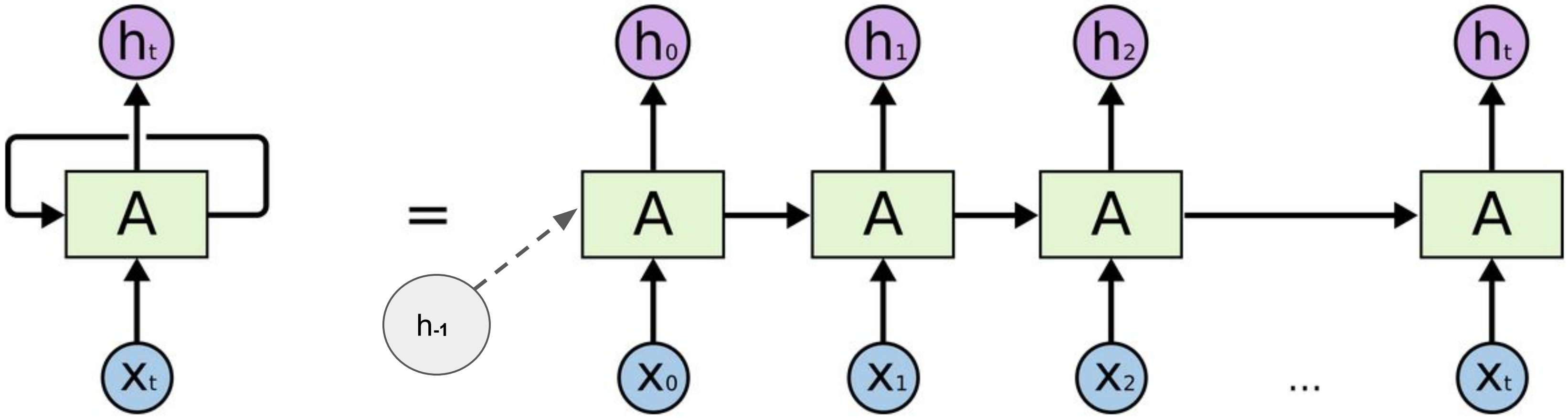
Recurrent Neural Networks

- Looping network
- Parameter sharing across timesteps
- Derivatives aggregated across all time steps
- “Backpropagation through time (BPTT)”



(http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

RNN Unrolled

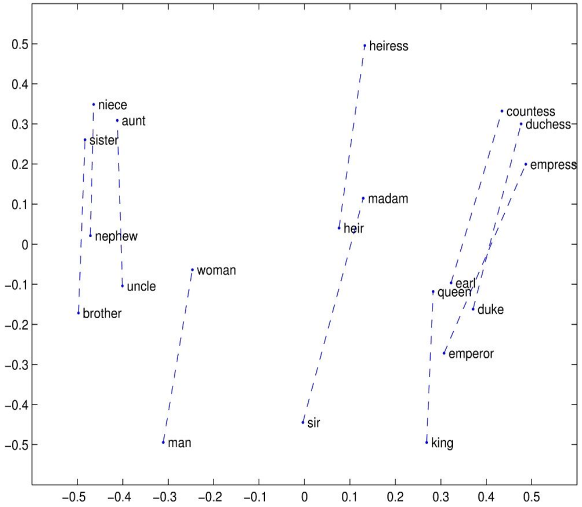


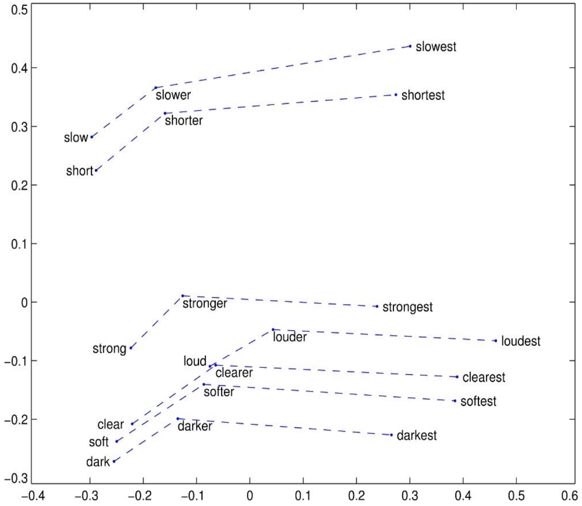
An unrolled recurrent neural network.

(http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Slight Detour - Text Vectors

- One hot encoding
  - "Never gonna give you up" {N=5}
    One Hot Encoding: Never = [1, 0, 0, 0, 0]



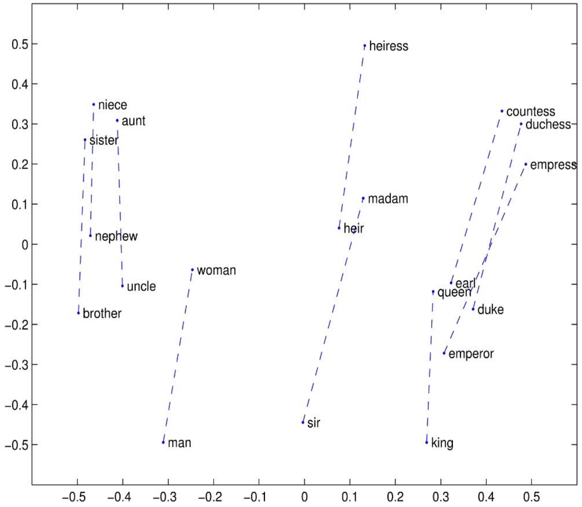


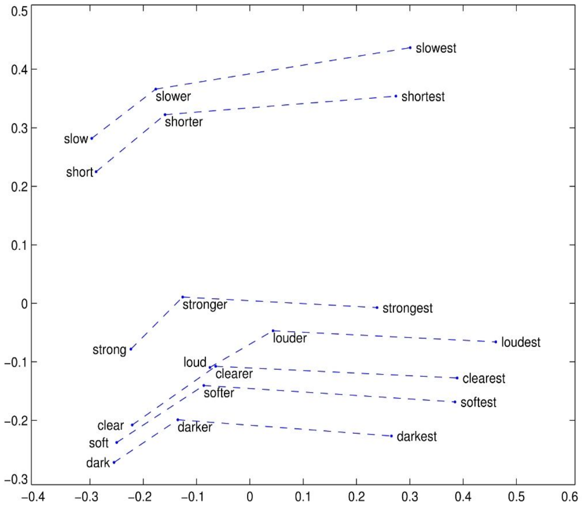
https://nlp.stanford.edu/projects/glove/

Slight Detour - Text Vectors

- One hot encoding
  - “Never gonna give you up” {N=5}
    One Hot Encoding: Never = [1, 0, 0, 0, 0]

- Input/Post-processing: Word embedding
  - Efficient use of space (denser)
  - Can represent relationships





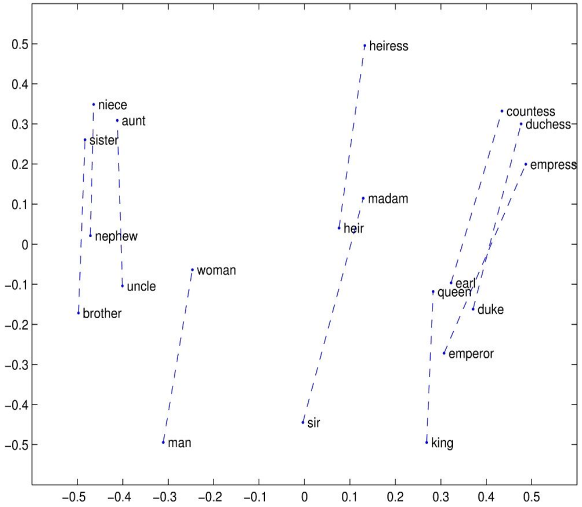
https://nlp.stanford.edu/projects/glove/

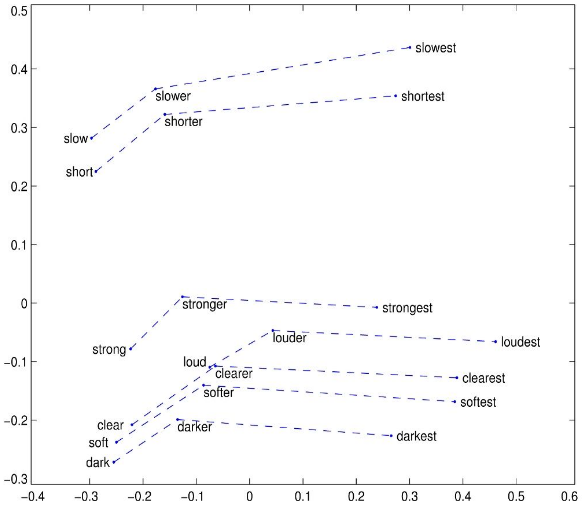
Slight Detour - Text Vectors

- One hot encoding
  - “Never gonna give you up” {N=5}
    One Hot Encoding: Never = [1, 0, 0, 0, 0]

- Input/Post-processing: Word embedding
  - Efficient use of space (denser)
  - Can represent relationships

- Output: Probability Distribution
  - “Never gonna give you **up**” {N=5}





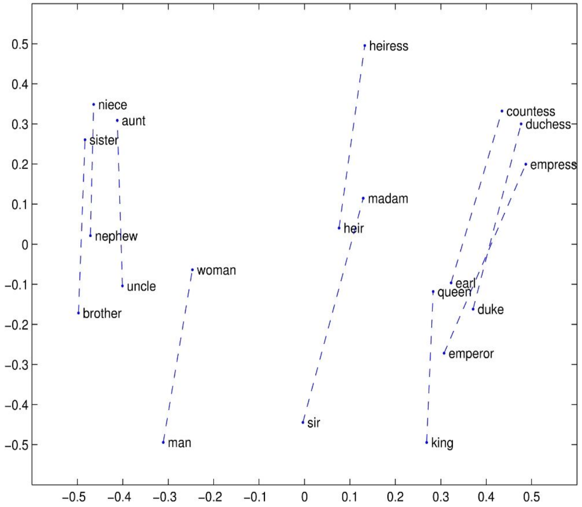
https://nlp.stanford.edu/projects/glove/

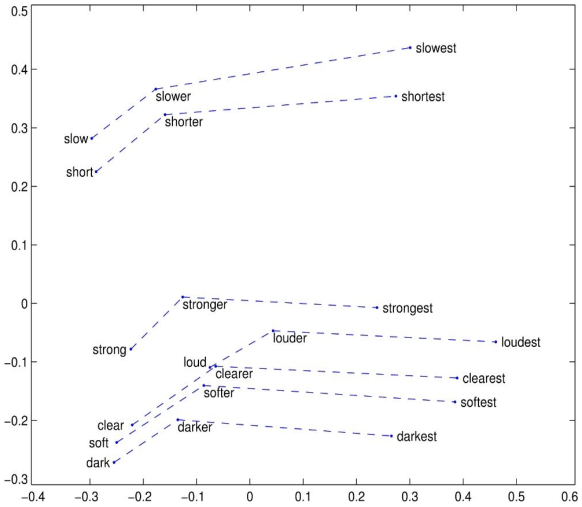
Slight Detour - Text Vectors

- One hot encoding
  - “Never gonna give you up” {N=5}
    One Hot Encoding: Never = [1, 0, 0, 0, 0]

- Input/Post-processing: Word embedding
  - Efficient use of space (denser)
  - Can represent relationships

- Output: Probability Distribution
  - “Never gonna give you **up**” {N=5}
    [Never, gonna, give, you up]
    P(w)=[0.01, 0.03, 0.04, 0.05, 0.87]





https://nlp.stanford.edu/projects/glove/

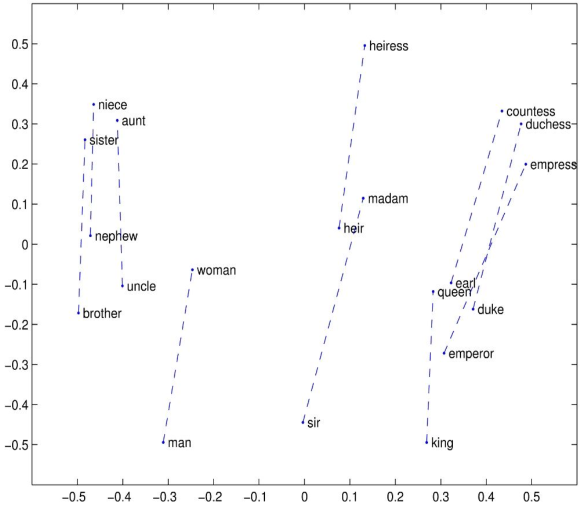
Slight Detour - Text Vectors

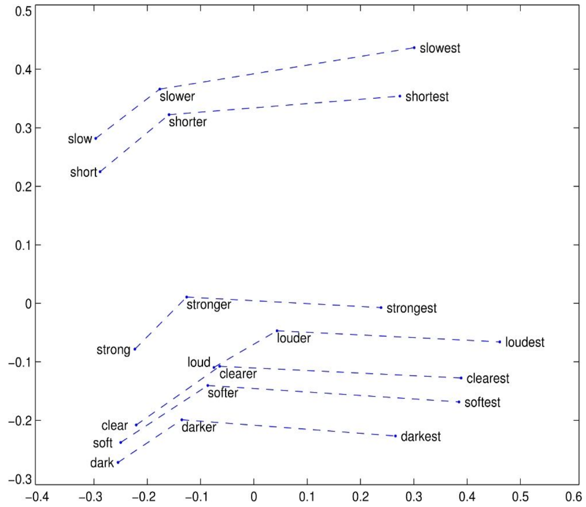
- One hot encoding
  - “Never gonna give you up” {N=5}
    One Hot Encoding: Never = [1, 0, 0, 0, 0]

- Input/Post-processing: Word embedding
  - Efficient use of space (denser)
  - Can represent relationships

- Output: Probability Distribution
  - “Never gonna give you **up**” {N=5}
    [Never, gonna, give, you up]
    P(w)=[0.01, 0.03, 0.04, 0.05, 0.87]

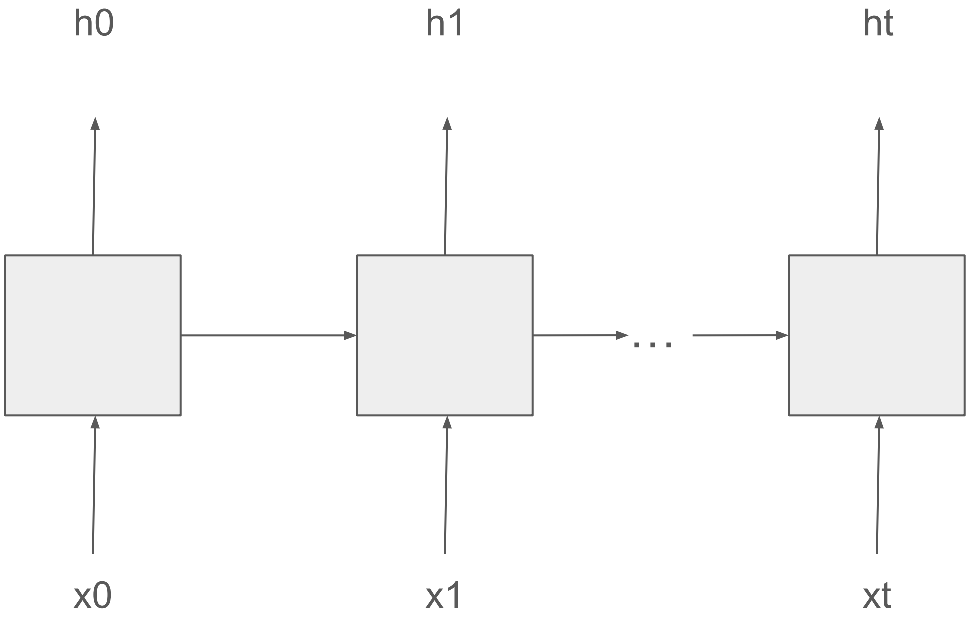
    “Never gonna give you up. Never gonna let you **down**” {N=8}
    [Never, gonna, give, you, up, let, down]
    P(w)=[0.01, 0.01, 0.01, 0.03, 0.44, 0.03, 0.03, 0.44]





https://nlp.stanford.edu/projects/glove/

RNN examples

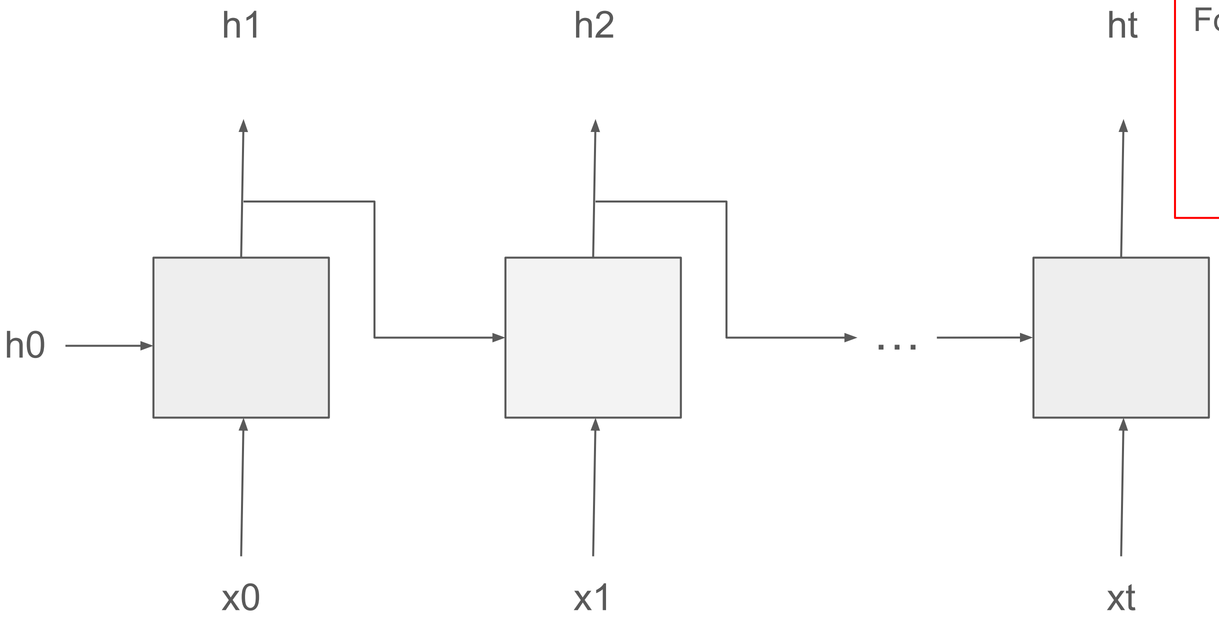


For the next images:

x0 → h0: transcription

x0 → h1: prediction/generation

RNN examples

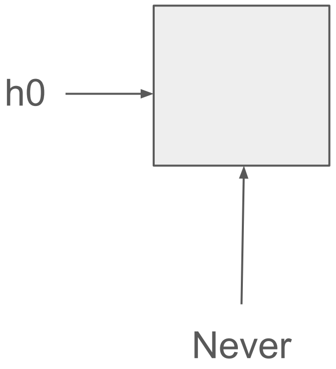


For the next images:

x0 → h0: transcription

x0 → h1: prediction/generation

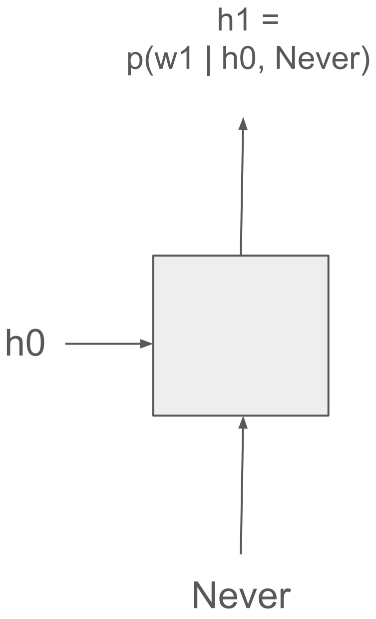
RNN example: prediction



Never gonna give you ___

Never gonna give you up

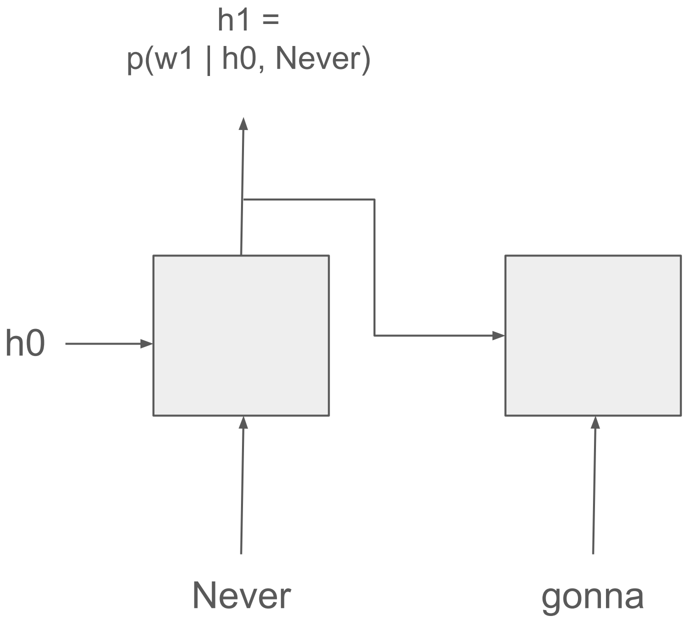
RNN example: prediction



Never gonna give you ___

Never gonna give you up

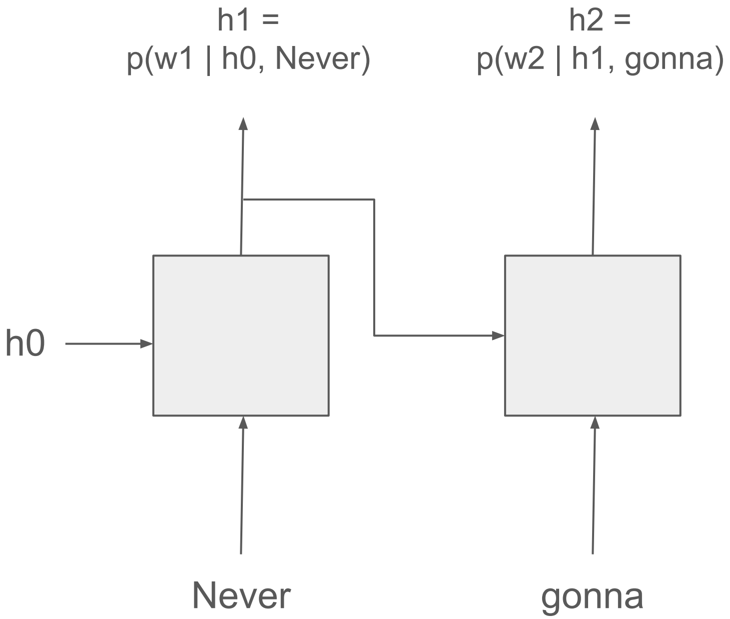
RNN example: prediction



Never gonna give you ___

Never gonna give you up

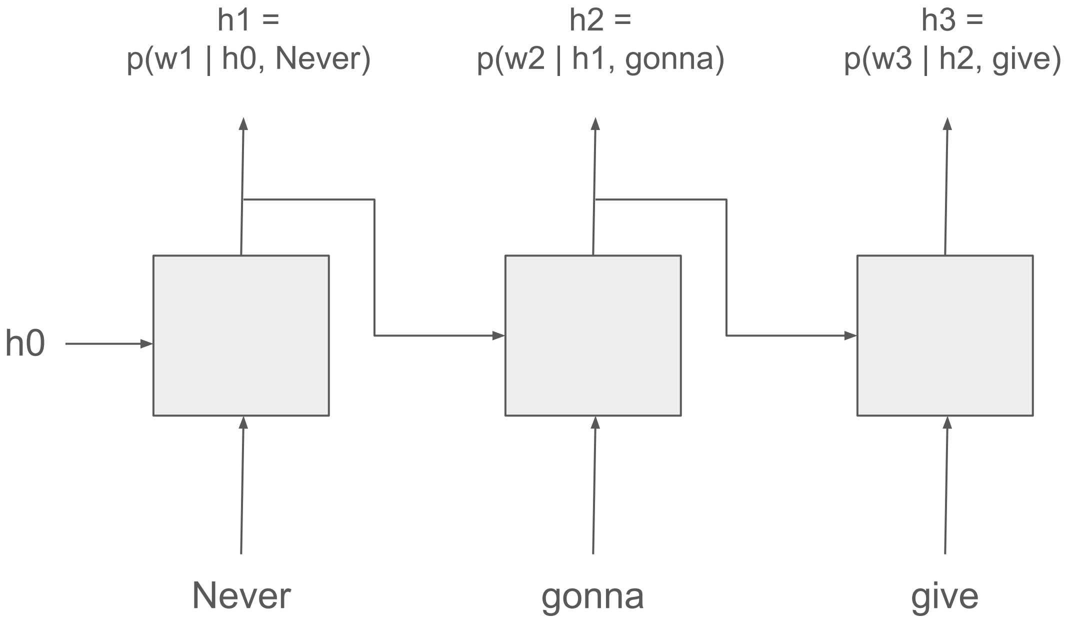
RNN example: prediction



Never gonna give you ___

Never gonna give you up

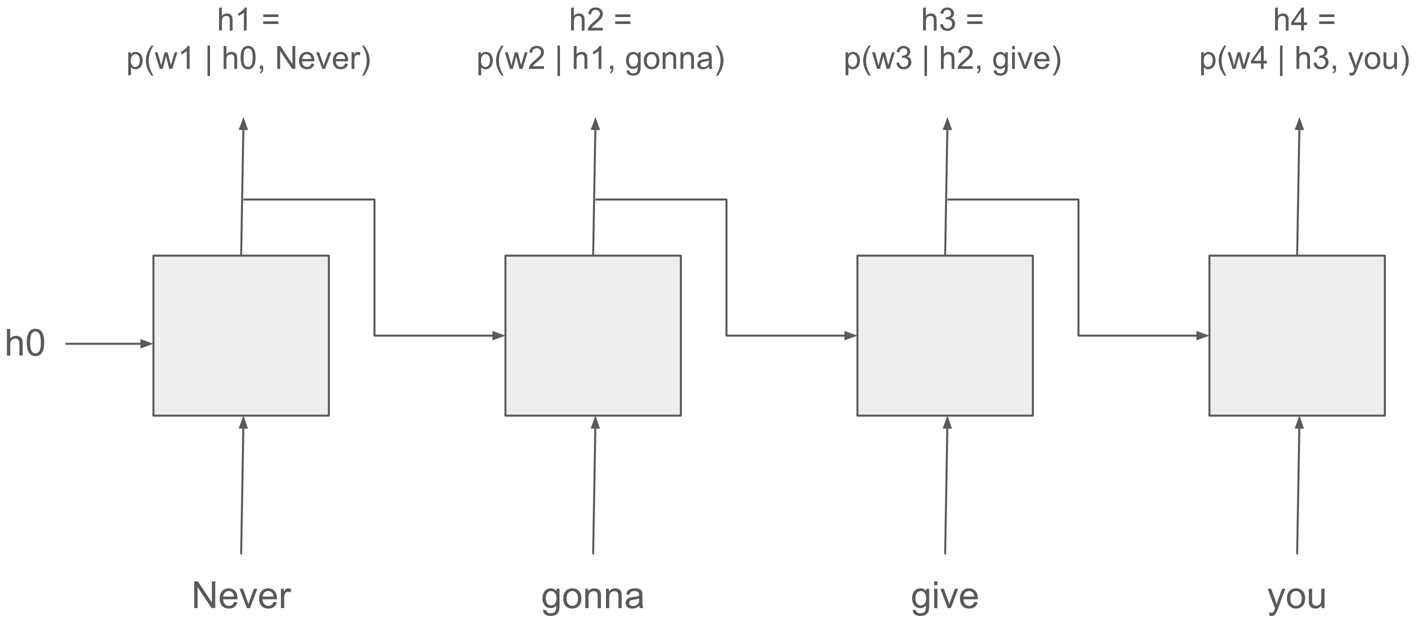
RNN example: prediction



Never gonna give you ___

Never gonna give you up

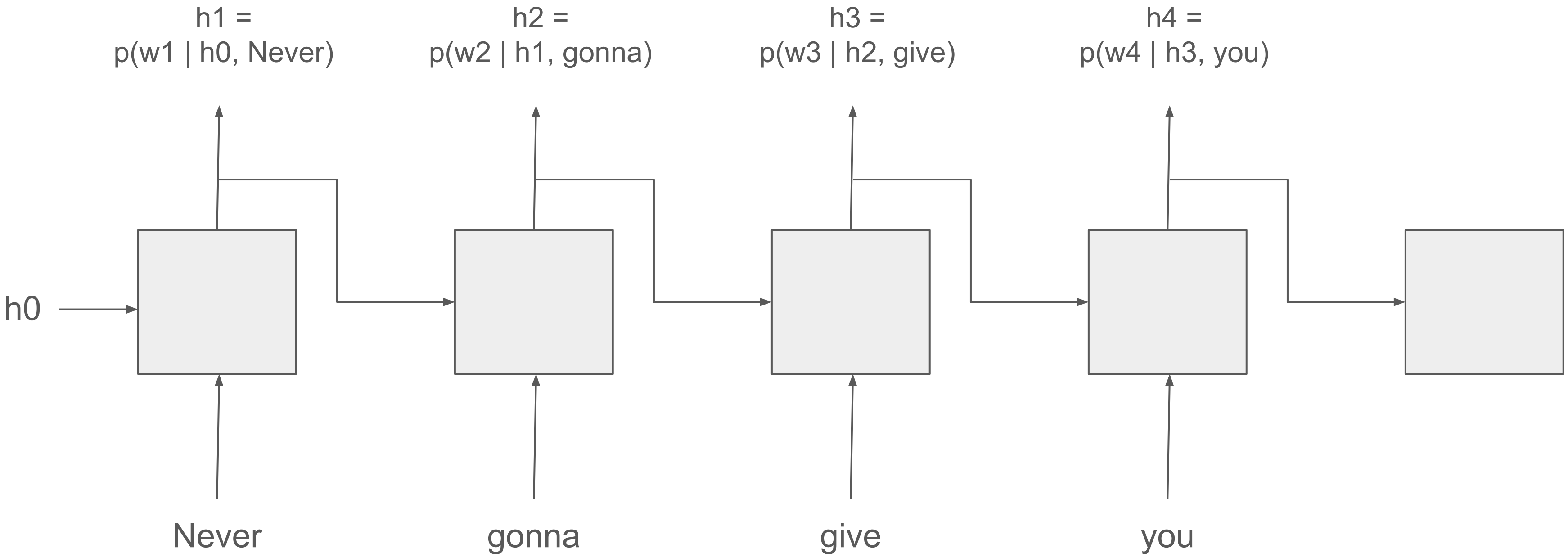
RNN example: prediction



Never gonna give you ___

Never gonna give you up

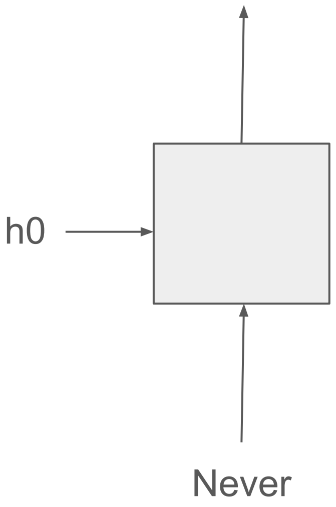
RNN example: prediction



Never gonna give you ___

Never gonna give you up

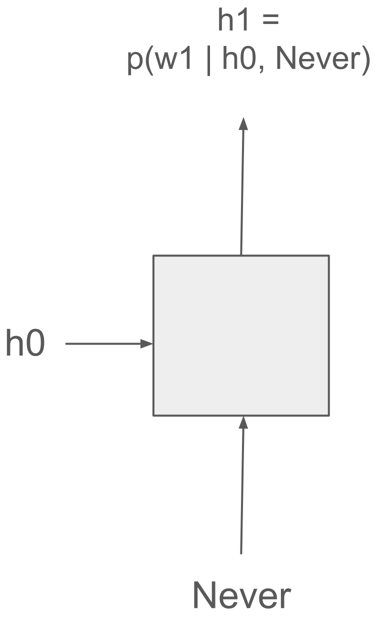
RNN example: generation



Never gonna ___ ___ ___

Never gonna give you up

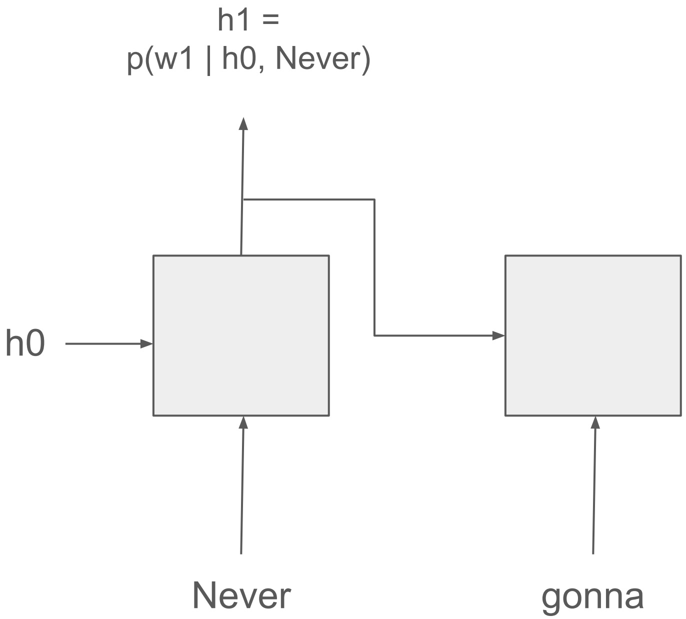
RNN example: generation



Never gonna ___ ___ ___

Never gonna give you up

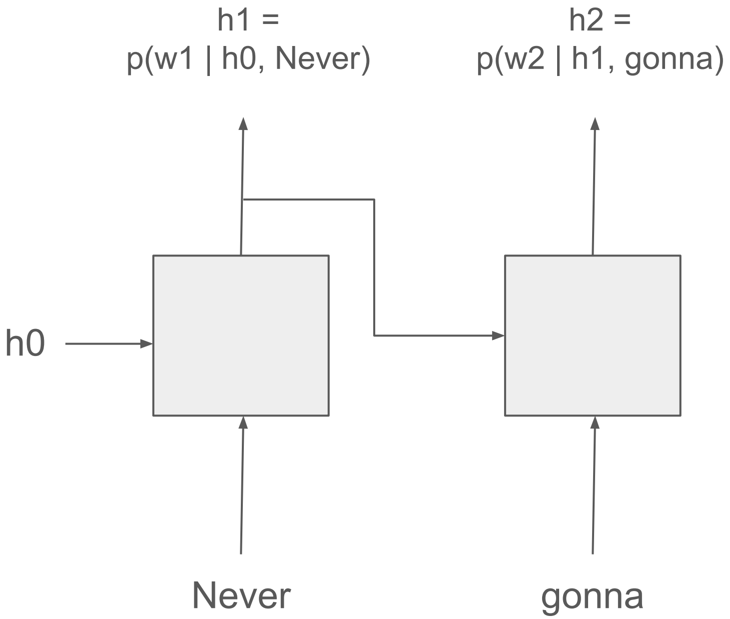
RNN example: generation



Never gonna ___ ___ ___

Never gonna give you up

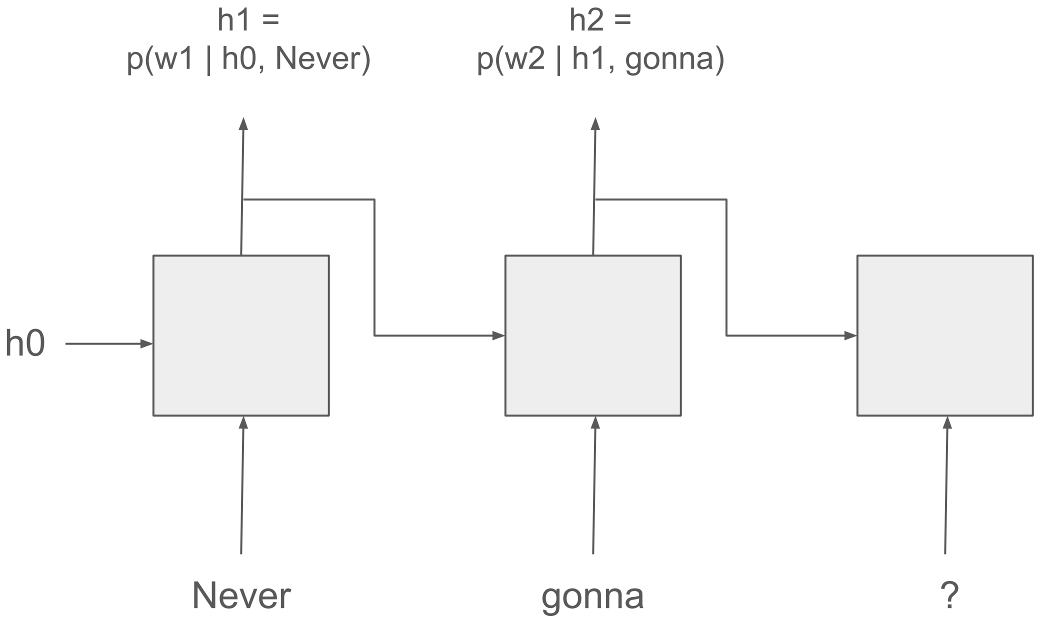
RNN example: generation



Never gonna ___ ___ ___

Never gonna give you up

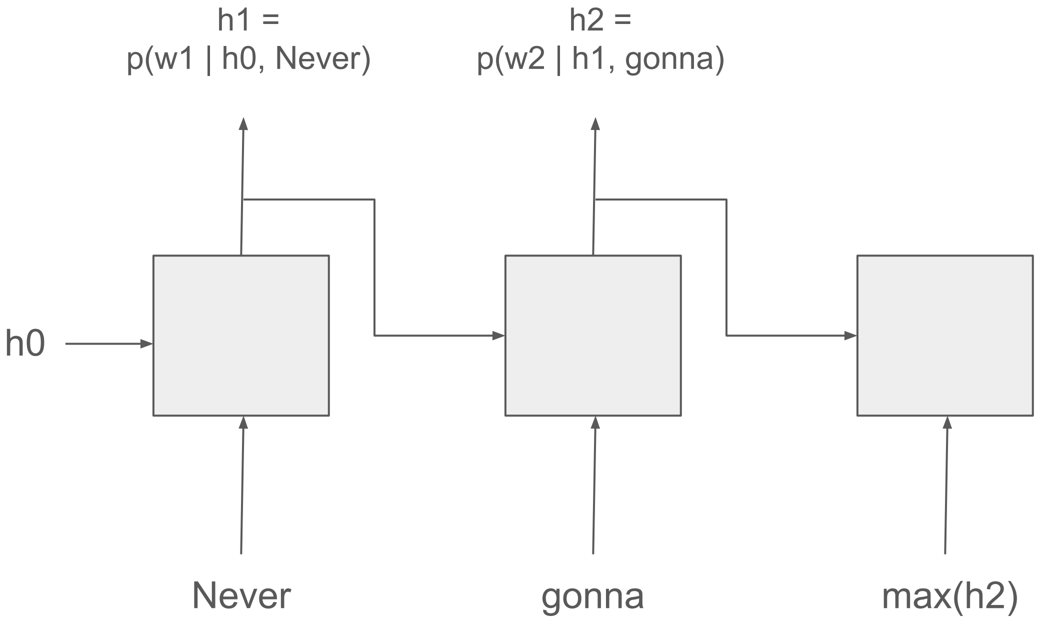
RNN example: generation



Never gonna ___ ___ ___

Never gonna give you up

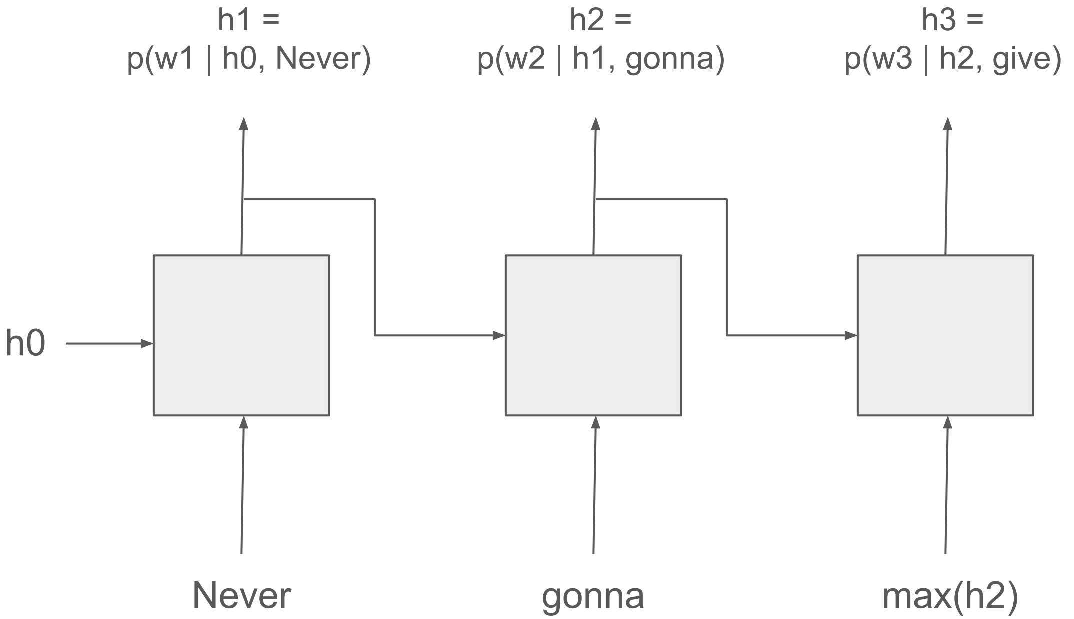
RNN example: generation



Never gonna ___ ___ ___

Never gonna give you up

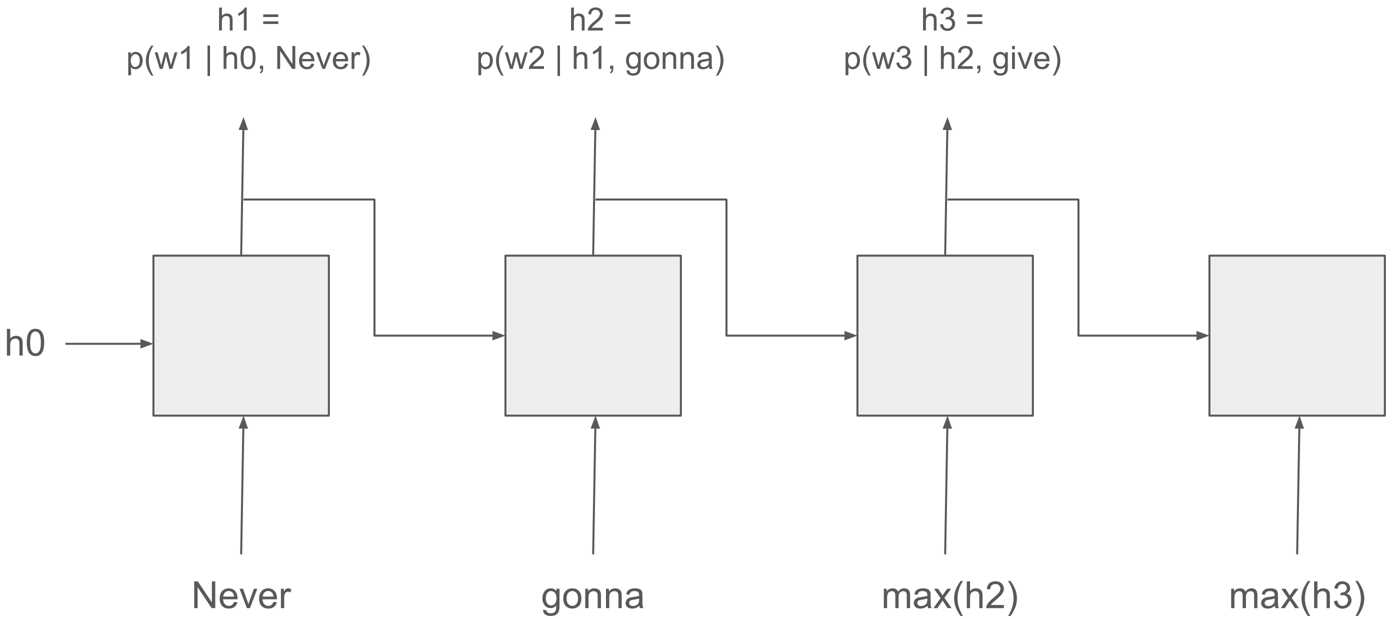
RNN example: generation



Never gonna ___ ___ ___

Never gonna give you up

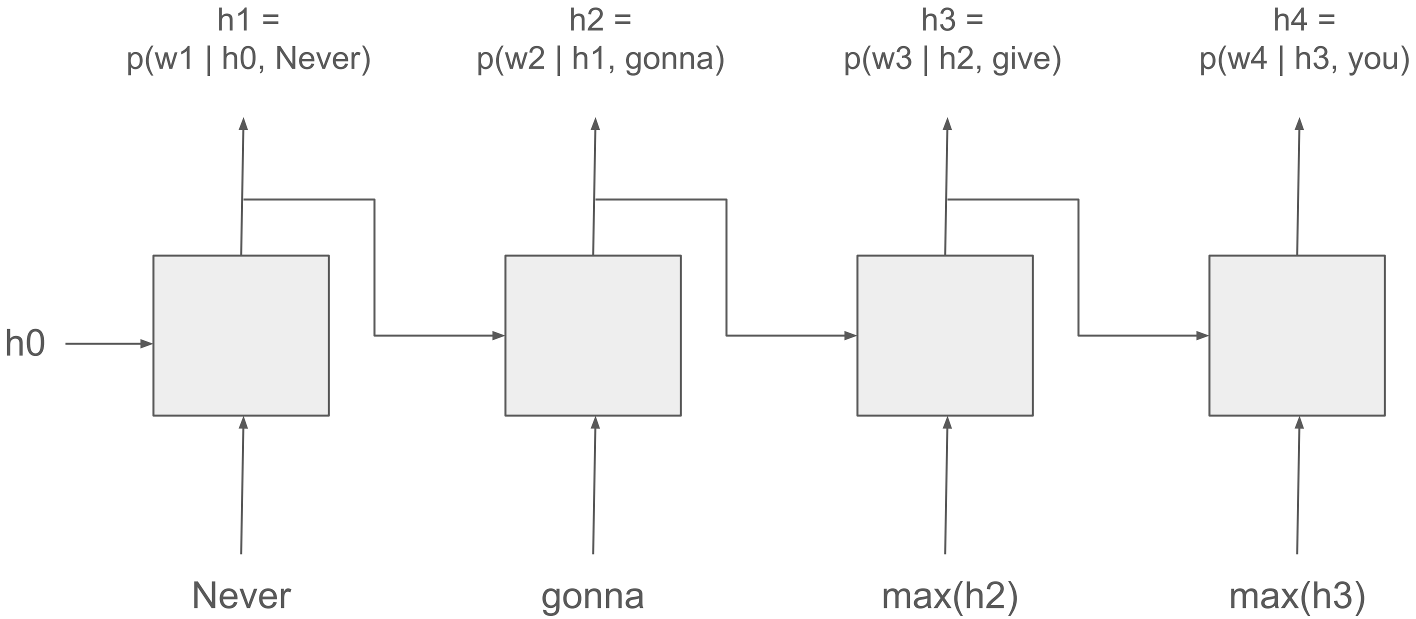
RNN example: generation



Never gonna ___ ___ ___

Never gonna give you up

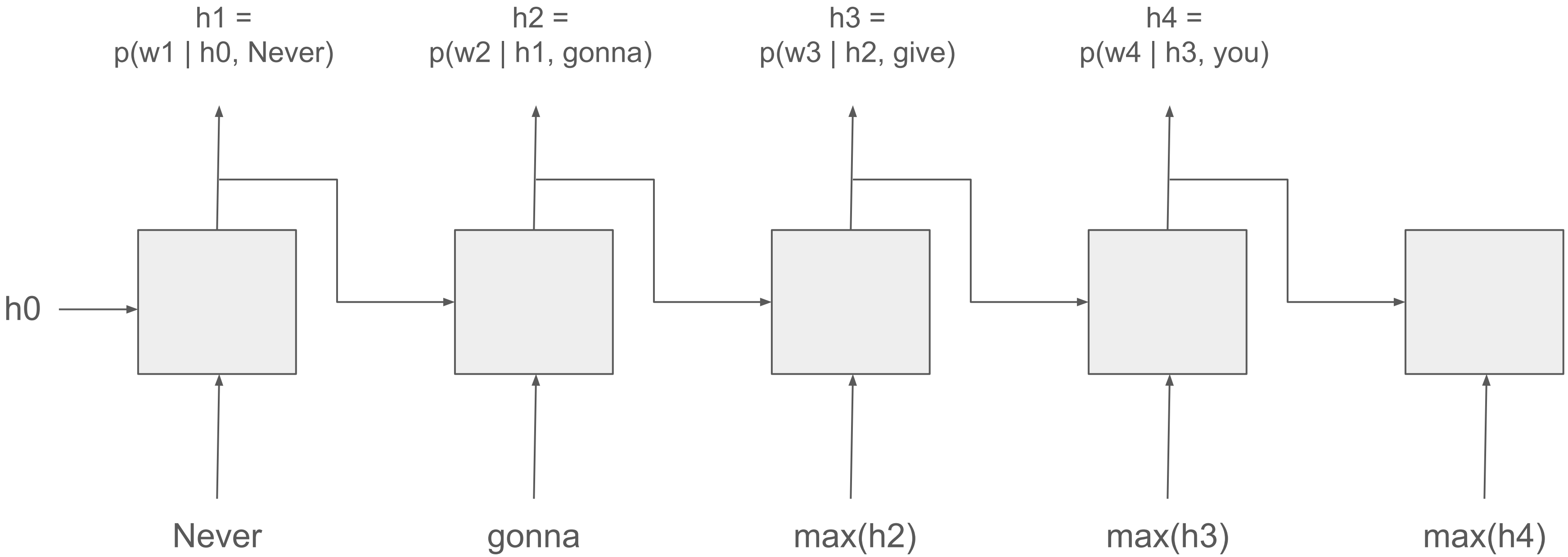
RNN example: generation



Never gonna ___ ___ ___

Never gonna give you up

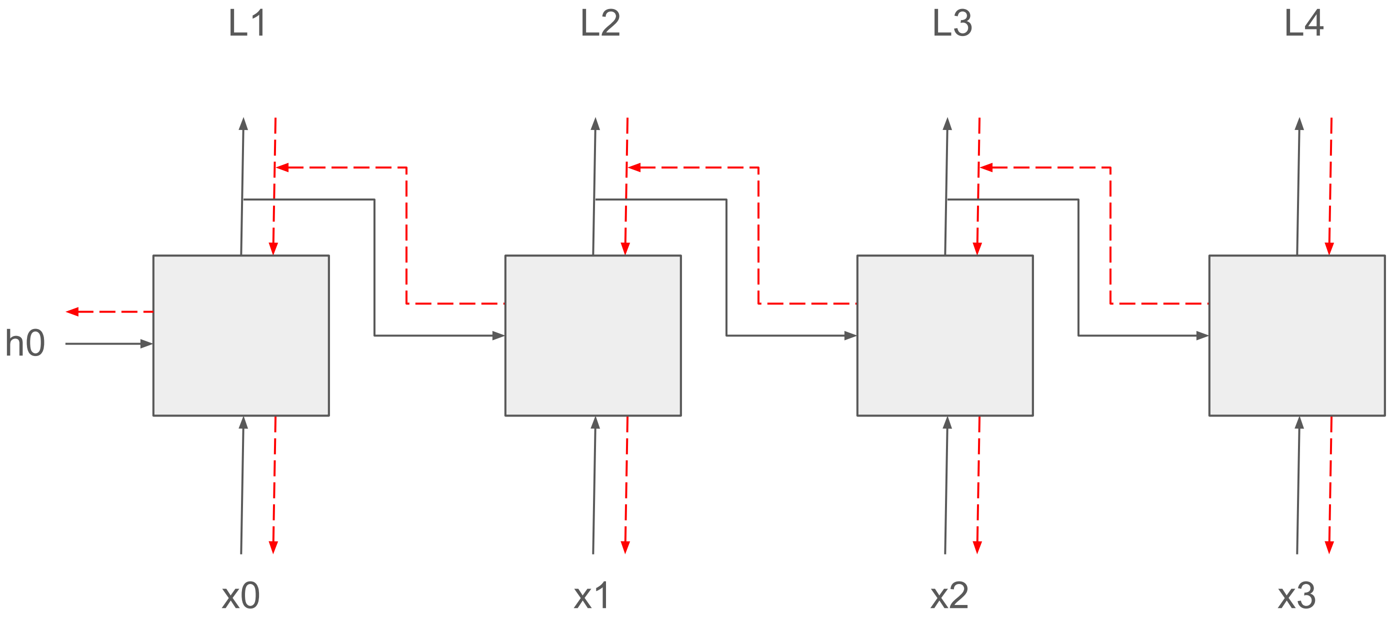
RNN example: generation



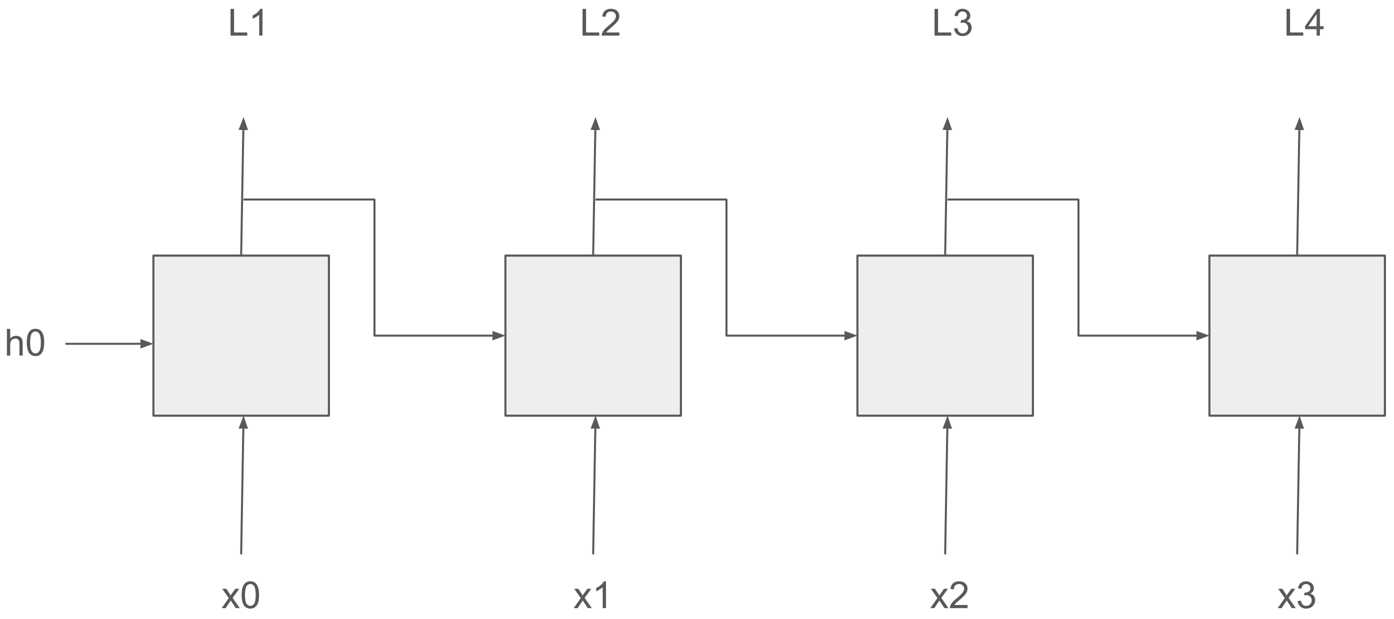
Never gonna ___ ___ ___

Never gonna give you up

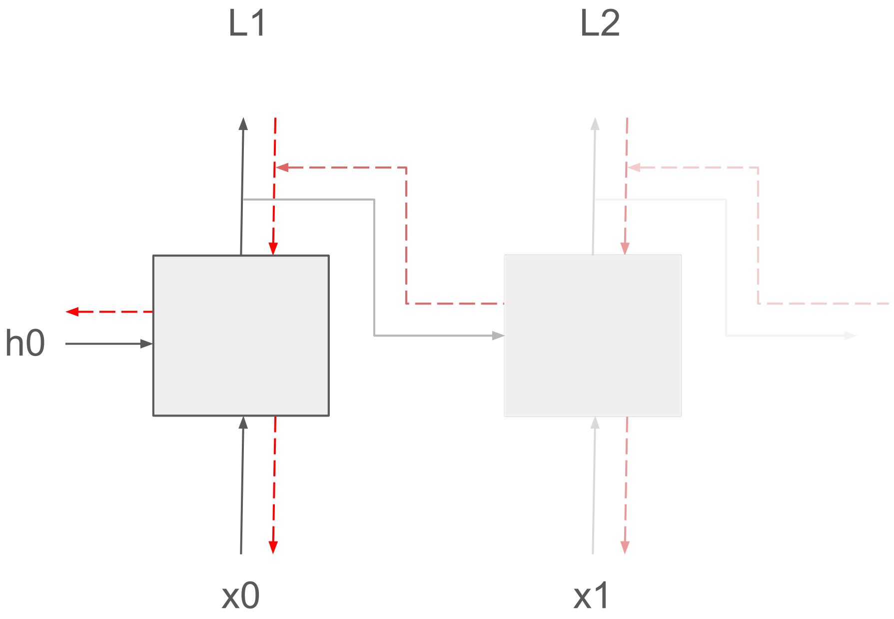
RNN backprop



RNN

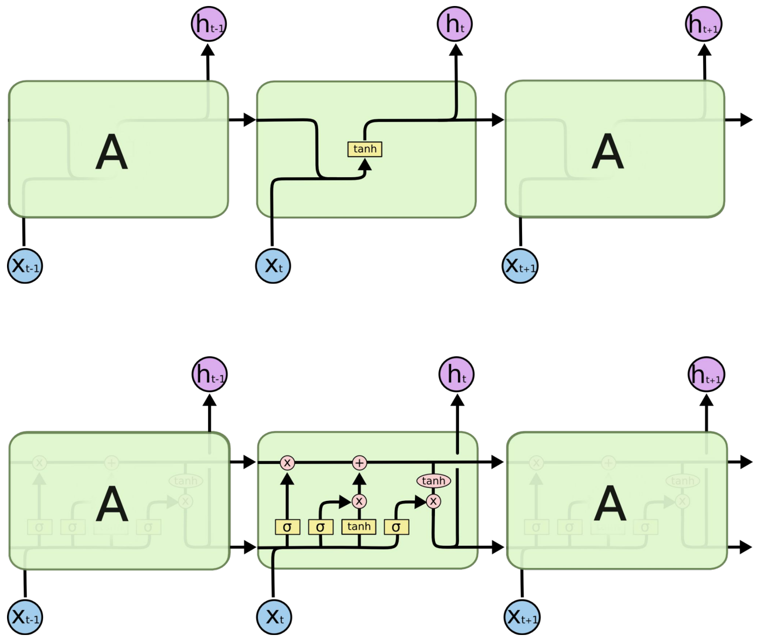


RNN Problems → Architectural Solutions



- After many iterations
  - Short Term Memory
  - Vanishing Gradients

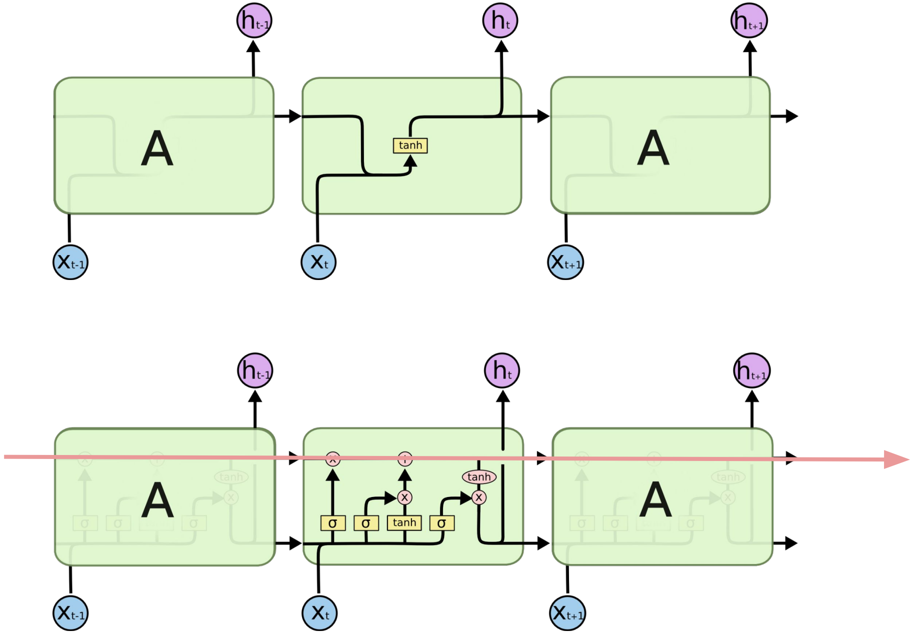
RNN Problems → Architectural Solutions



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

- After many iterations
  - Short Term Memory
  - Vanishing Gradients
  - **LSTMs and GRUs combat these issues**

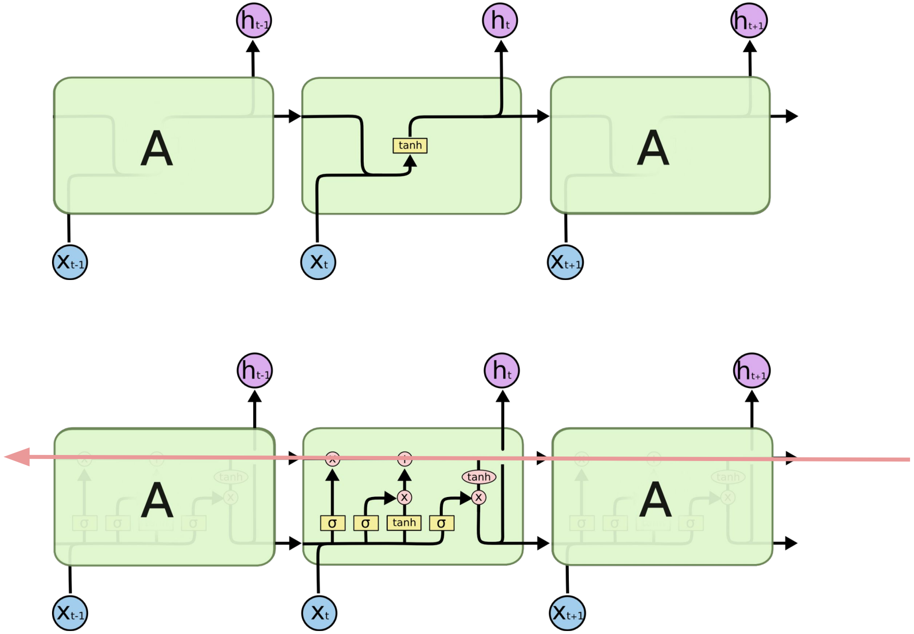
RNN Problems → Architectural Solutions



- After many iterations
  - Short Term Memory
  - Vanishing Gradients
  - **LSTMs and GRUs combat these issues**

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

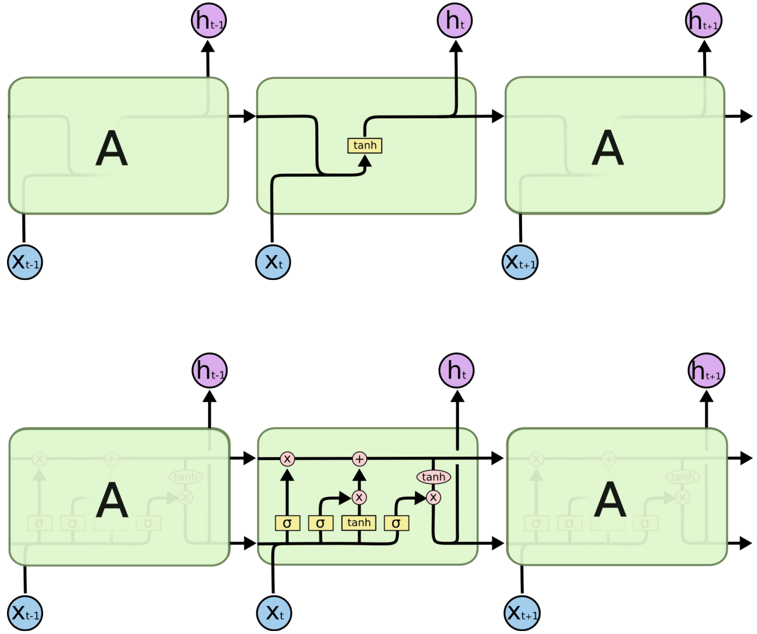
RNN Problems → Architectural Solutions



- After many iterations
  - Short Term Memory
  - Vanishing Gradients
  - **LSTMs and GRUs combat these issues**

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

RNN Training Issues → Noise and Cold Starts

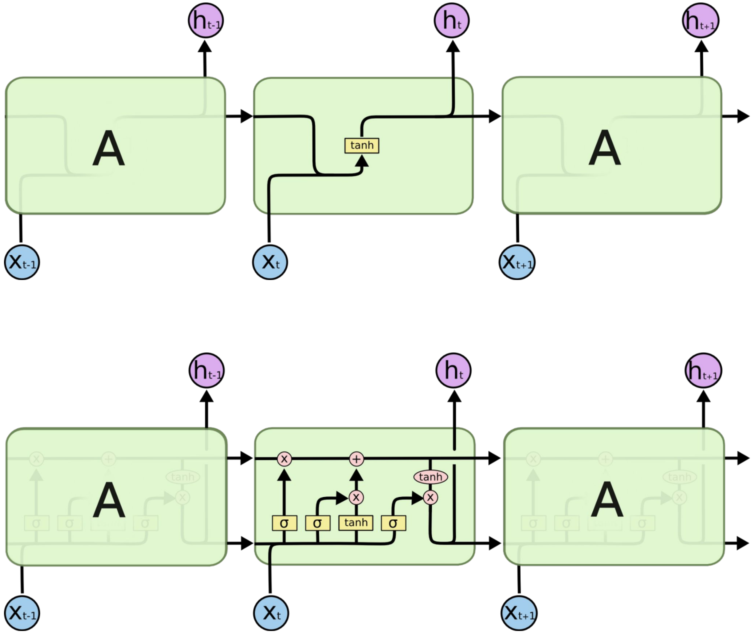


- After many iterations
  - Short Term Memory
  - Vanishing Gradients
  - **LSTMs and GRUs combat these issues**

- Early training for tasks like generation

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

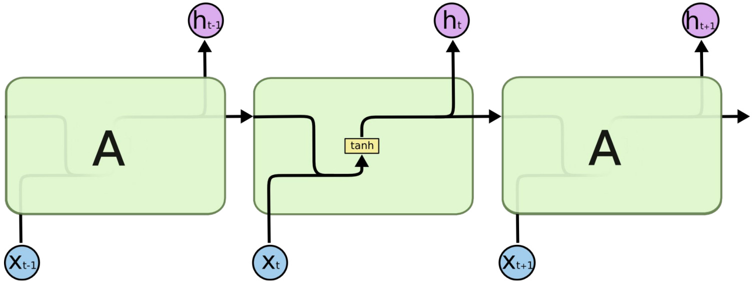
RNN Training Issues → Noise and Cold Starts

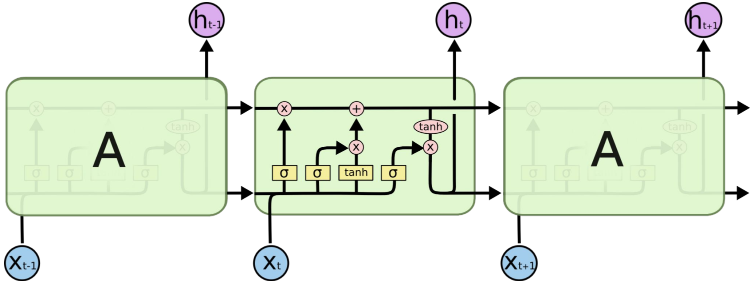


https://colah.github.io/posts/2015-08-Understanding-LSTMs/

- After many iterations
  - Short Term Memory
  - Vanishing Gradients
  - **LSTMs and GRUs combat these issues**
- Early training for tasks like generation
  - Lack of exploration - noise

RNN Training Issues → Noise and Cold Starts



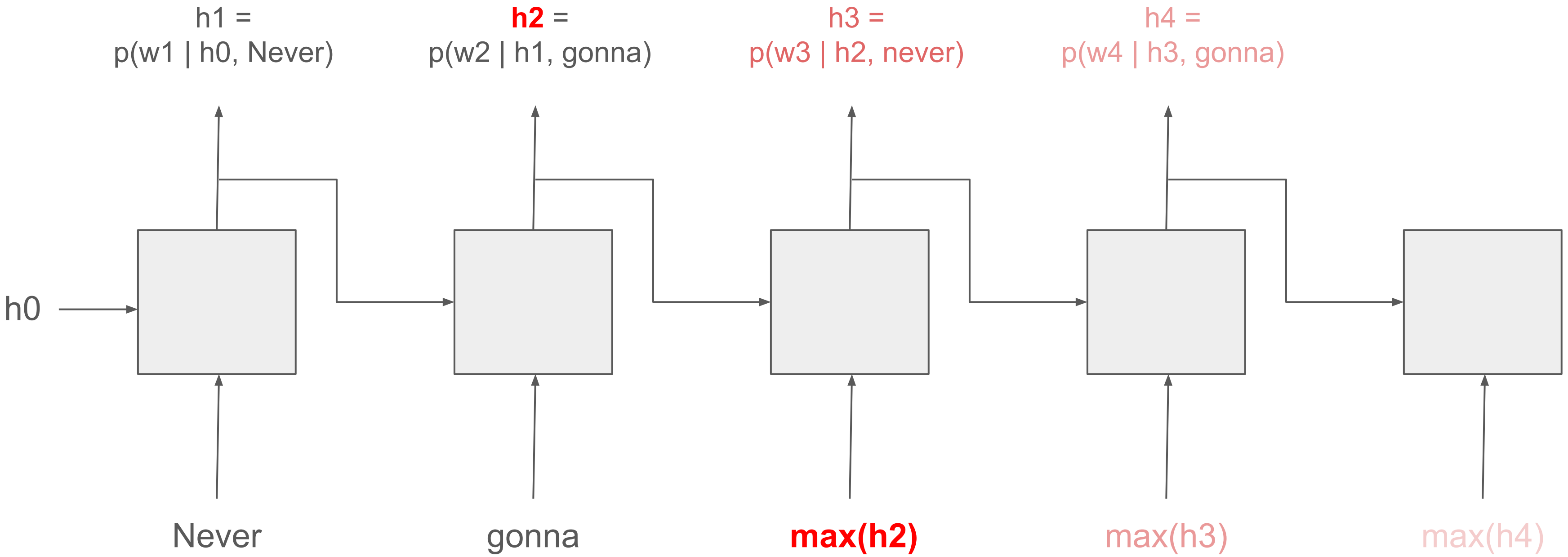


https://colah.github.io/posts/2015-08-Understanding-LSTMs/

- After many iterations
  - Short Term Memory
  - Vanishing Gradients
  - **LSTMs and GRUs combat these issues**

- Early training for tasks like generation
  - Lack of exploration - noise
  - Cold start - teacher forcing

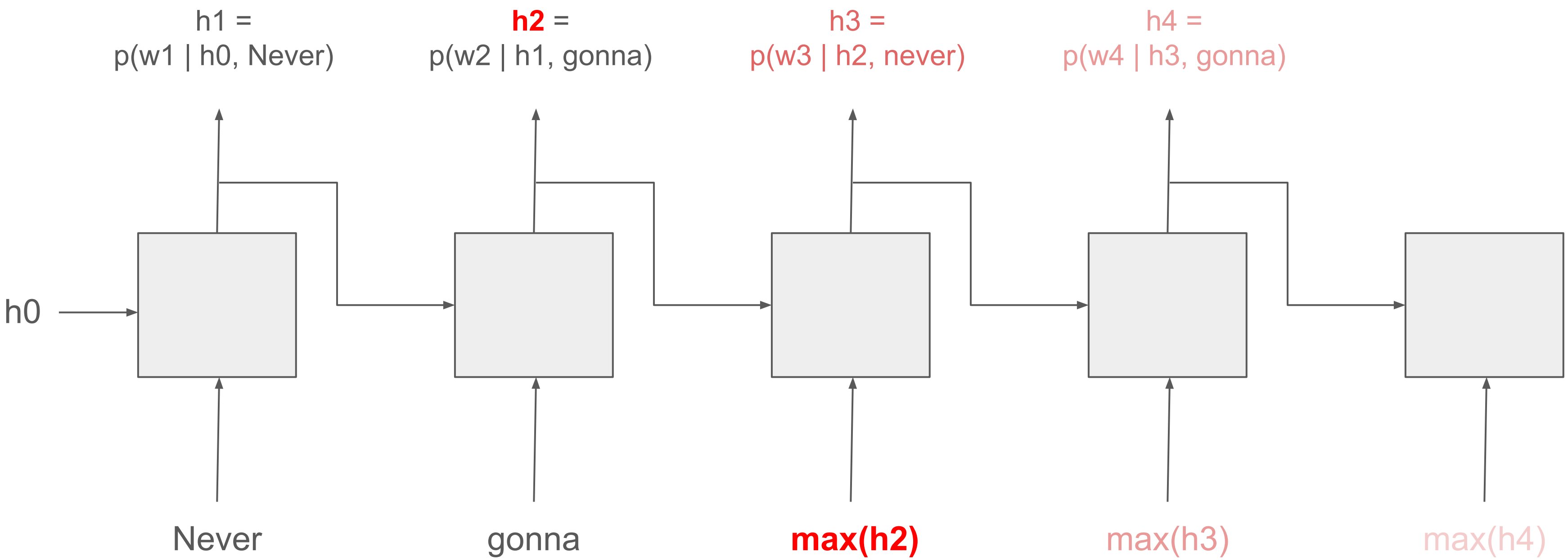
RNN Training Issues → Noise and Cold Starts



Never gonna never gonna ___

Never gonna give you up

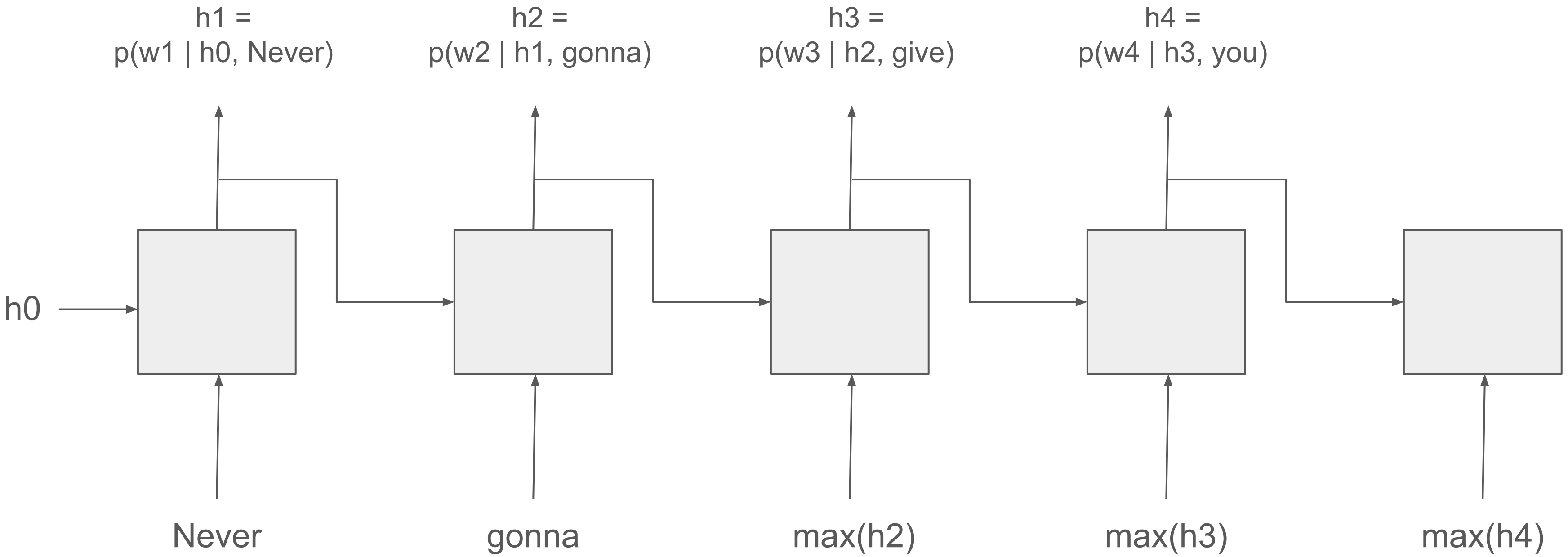
RNN Training Issues → Noise and Cold Starts



Never gonna never gonna ___

Never gonna give you up

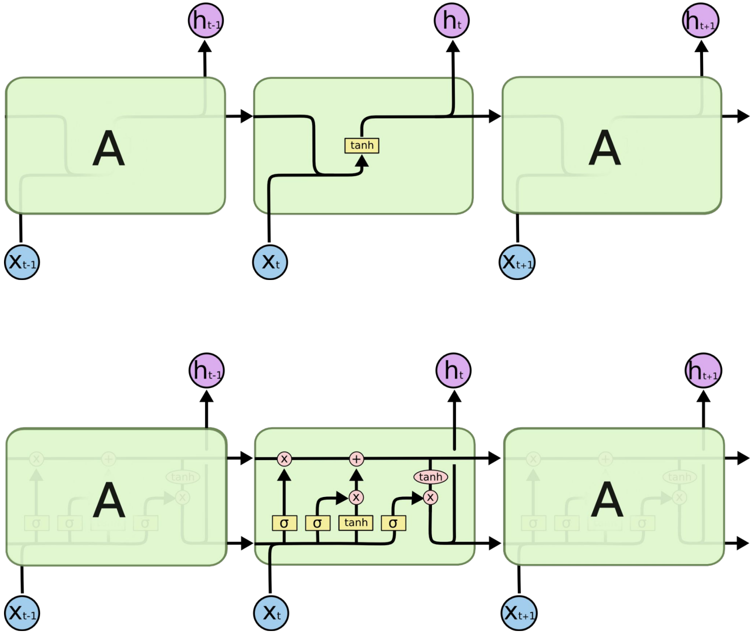
RNN Training Issues → Noise and Cold Starts



Never gonna ___ ___ ___

Never gonna give you up .

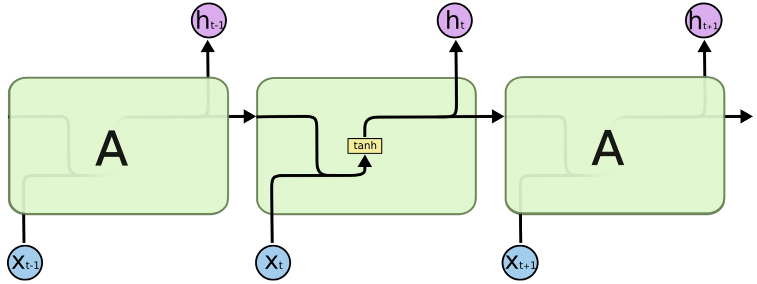
RNN Training Issues → Noise and Cold Starts

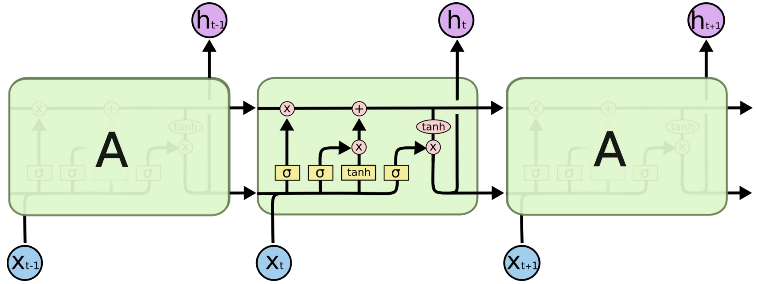


https://colah.github.io/posts/2015-08-Understanding-LSTMs/

- After many iterations
  - Short Term Memory
  - Vanishing Gradients
  - **LSTMs and GRUs combat these issues**
- Early training for tasks like generation
  - Lack of exploration - noise
  - Cold start - teacher forcing

RNN Dependency Issues → Attention





https://colah.github.io/posts/2015-08-Understanding-LSTMs/

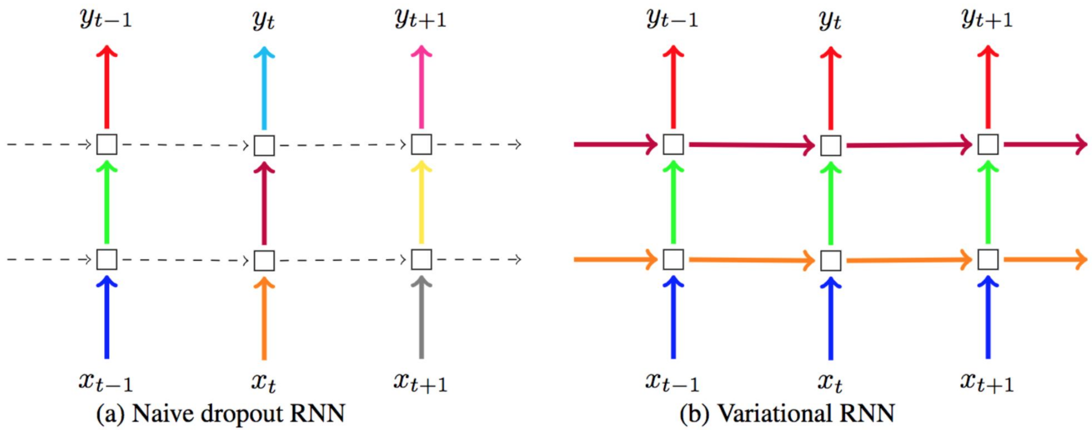
- After many iterations
  - Short Term Memory
  - Vanishing Gradients
  - **LSTMs and GRUs combat these issues**

- Early training for tasks like generation
  - Lack of exploration - noise
  - Cold start - teacher forcing

- Long-term dependencies may be reduced or lost
  - Attention (later lectures)

Dropout in sequence models

1. Different mask on each timestep (naive, available in PyTorch LSTM)
2. Same mask on each timestep for input/output connections (locked dropout)
3. Variational dropout - same mask on each time step for input/output and recurrent connections



Gal, Yarin, and Zoubin Ghahramani. "A theoretically grounded application of dropout in recurrent neural networks." *Advances in neural information processing systems* 29 (2016).