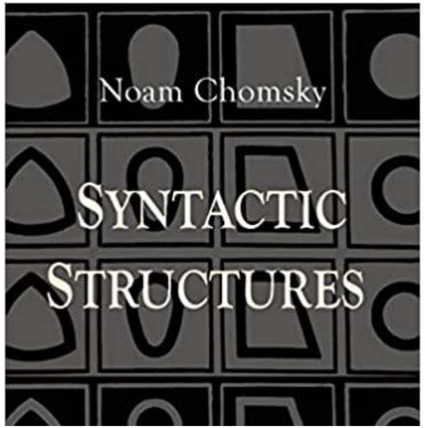
Overview

- Tokenization and Cleaning
- Word Embeddings
- Basic Sequence Model

NLP is hard!

- How to represent text as data?
- Humans represent text using characters
  - Takes years to learn to read
  - Different peoples do it differently all around the world
- For most tasks this is not a particularly helpful embedding
  - Intrinsic meaning is largely lost

$$\longrightarrow \begin{bmatrix} 1 & .5 & 1 & 0 \\ 0 & .25 & .5 & 1 \\ 1 & .25 & 0 & 1 \\ .5 & 0 & 1 & 1 \end{bmatrix}$$



train → 20-18-1-9-14

brain → 2-18-1-9-14

head → 8-5-1-4

Google’s LaMDA

- Language Model for Dialogue Applications
  - Conversational AI

- Google Employee, paid to “push the limits” of LaMDA, was recently fired
  - Is convinced that LaMDA is sentient
  - Published transcripts of their conversation with LaMDA in WaPo
  - Some claim these are fake/edited

- Is LaMDA sentient? Is this the wrong question?
  - What are the effects of sentient-appearing AI on humans?
  - Conversational Deep-Fakes?

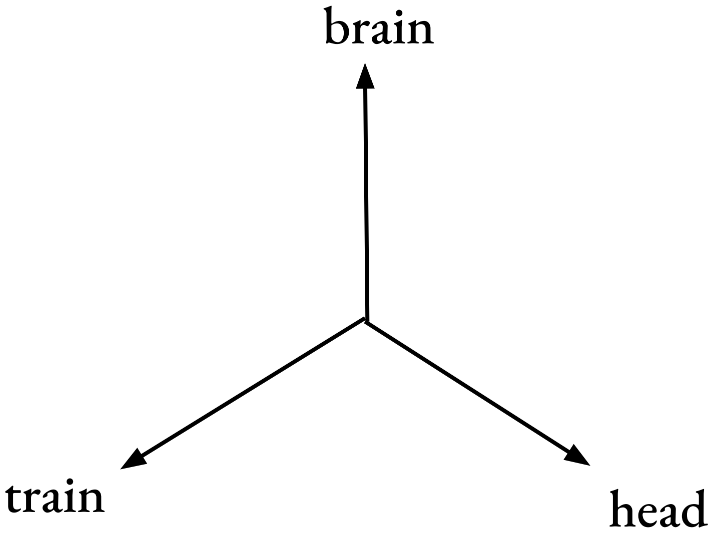
Tokenization

- Idea: Break up text into pieces (tokens) and treat as categorical variables
  - Often these tokens are words

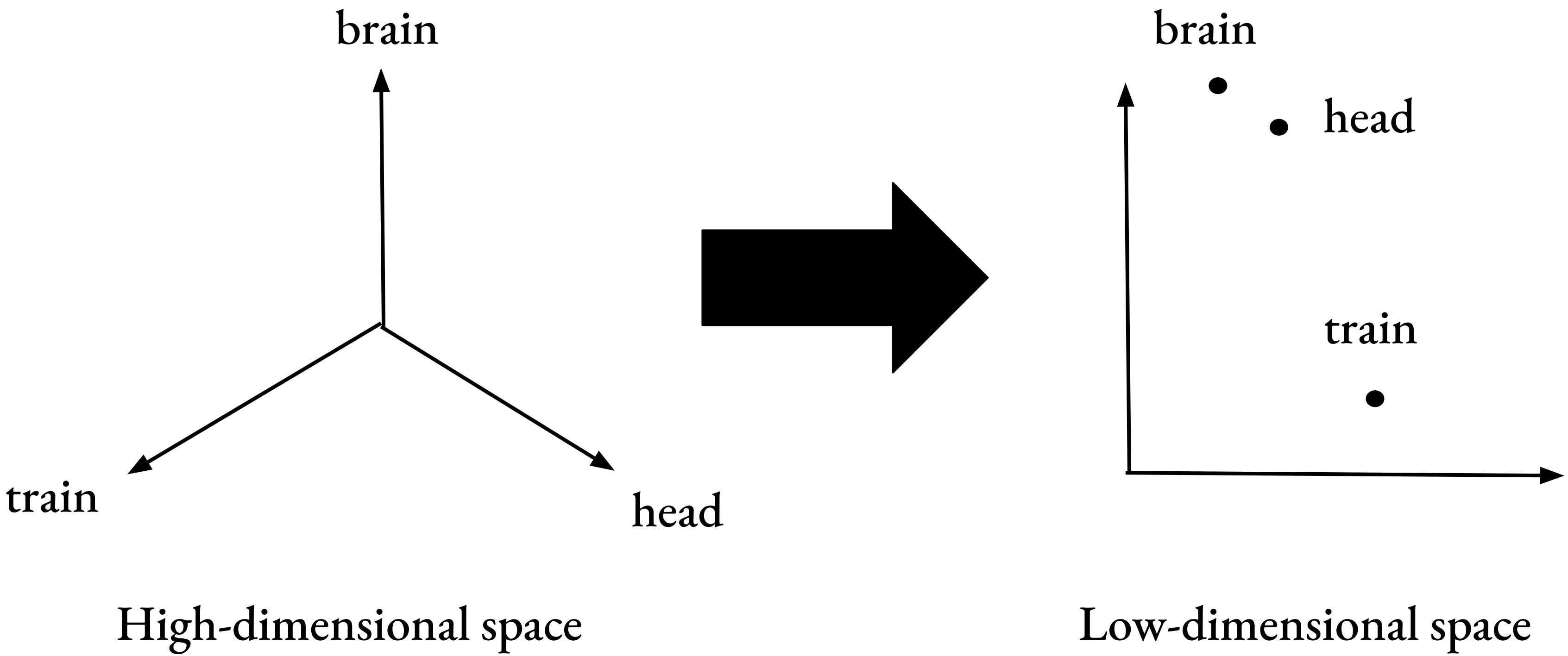
Tokenization

- Idea: Break up text into pieces (tokens) and treat as categorical variables
    - Often these tokens are words

$$\underset{\text{a}}{\begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix}}\quad\underset{\text{aadvark}}{\begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix}}\quad\cdots\quad\underset{\text{zebra}}{\begin{bmatrix}0\\0\\0\\\vdots\\1\end{bmatrix}}$$



Word Embedding



Other Types of Tokenization

- Idea: Break up text into pieces (tokens) and treat as categorical variables
- Words -> Tokens

Word (token) Embeddings

- Word2Vec
- Learn the word embedding by training on a "simple" NLP task.
- Fill in the blank using surrounding context

I am at track five. Here comes the __?__

Word (token) Embeddings

- Word2Vec
- Learn the word embedding by training on a “simple” NLP task.
- Fill in the blank using surrounding context

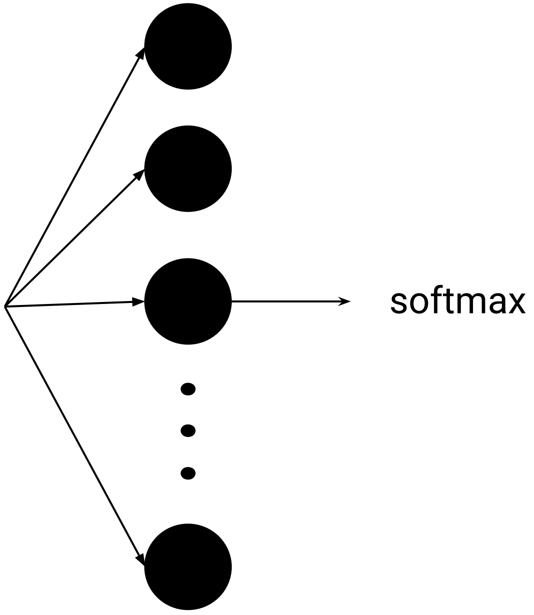
<u>I</u> <u>am</u> <u>at</u> <u>track</u> <u>five</u>. <u>Here</u> <u>comes</u> <u>the</u> <u>__?__</u>

- Distributional Semantics: The meaning of a word is given by the words that most often appear in the same context.
- There is a treasure trove of data for this task.
  - Ex. Use Wikipedia as your data.

Word (token) Embeddings

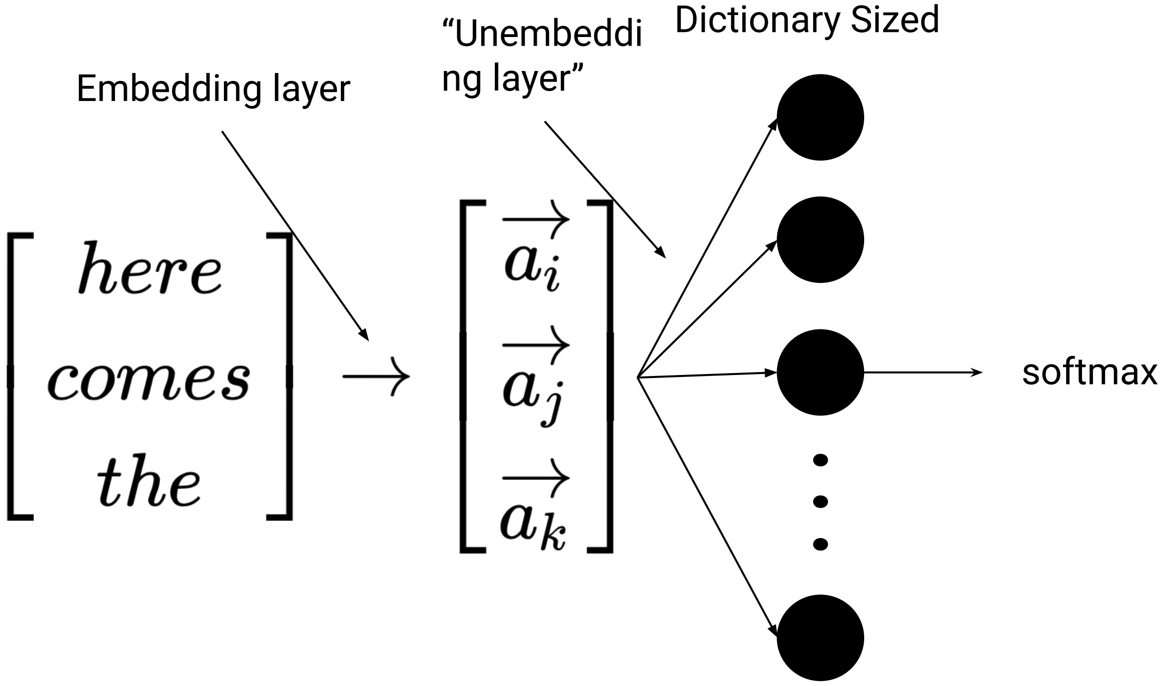
- Word2Vec
- Learn the word embedding by training on a "simple" NLP task.
- Fill in the blank using surrounding context

$$\begin{bmatrix} here \\ comes \\ the \end{bmatrix} \rightarrow \begin{bmatrix} \overrightarrow{a_i} \\ \overrightarrow{a_j} \\ \overrightarrow{a_k} \end{bmatrix}$$



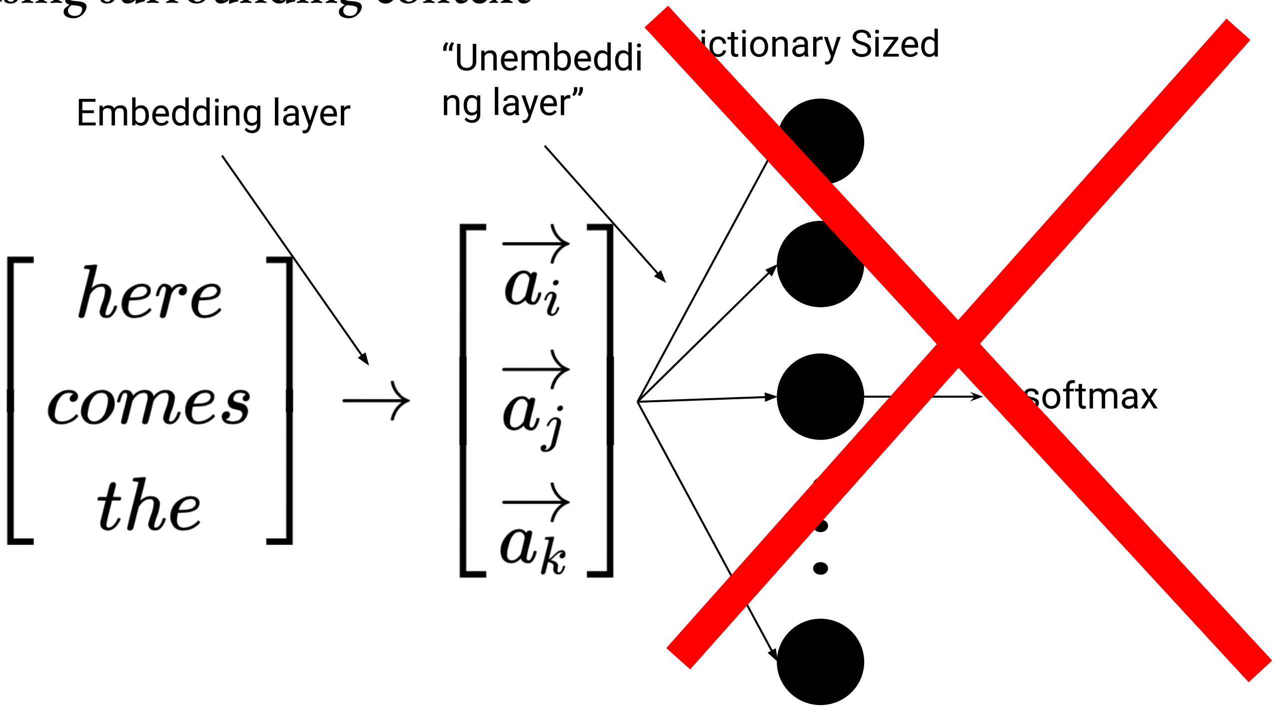
Word (token) Embeddings

- Word2Vec
- Learn the word embedding by training on a "simple" NLP task.
- Fill in the blank using surrounding context



Word (token) Embeddings

- Word2Vec
- Learn the word embedding by training on a “simple” NLP task.
- Fill in the blank using surrounding context



Word (token) Embeddings

- Word2Vec
- GloVe
    - Unsupervised learning using co-occurences of words in your corpus

GloVe: Global Vectors for Word Representation

Jeffrey Pennington, Richard Socher, Christopher D. Manning
Computer Science Department, Stanford University, Stanford, CA 94305
jpennin@stanford.edu, richard@socher.org, manning@stanford.edu

Word (token) Embeddings

- Idea: closeness in feature space <-> similarity in meaning

Word (token) Embeddings

- Idea: closeness in feature space <-> similarity in meaning
- Construct Analogies
    - v(cat) - v(feline) ~ v(dog) - v(canine)

Word (token) Embeddings

- Idea: closeness in feature space <-> similarity in meaning
- Construct Analogies
    - v(cat) - v(feline) ~ v(dog) - v(canine)
- Word embedding only as good as your text!

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings

Tolga Bolukbasi[1], **Kai-Wei Chang**[2], **James Zou**[2], **Venkatesh Saligrama**[1,2], **Adam Kalai**[2]
[1]Boston University, 8 Saint Mary's Street, Boston, MA
[2]Microsoft Research New England, 1 Memorial Drive, Cambridge, MA
tolgab@bu.edu, kw@kwchang.net, jamesyzou@gmail.com, srv@bu.edu, adam.kalai@microsoft.com

Deep Learning and NLP

- Before Deep Learning: Statistics, Handcrafted features for text/words

Deep Learning and NLP

- Before Deep Learning: Statistics, Handcrafted features for text/words
- Now: Use Deep Learning to take advantage of tons of text data

Deep Learning and NLP

- Before Deep Learning: Statistics, Handcrafted features for text/words
- Now: Use Deep Learning to take advantage of tons of text data
- NLP Tasks
    - Sequence Classification (Sentiment analysis)
    - Summarization
    - Question Answering
    - Similarity Detection
    - Translations
    - And more!

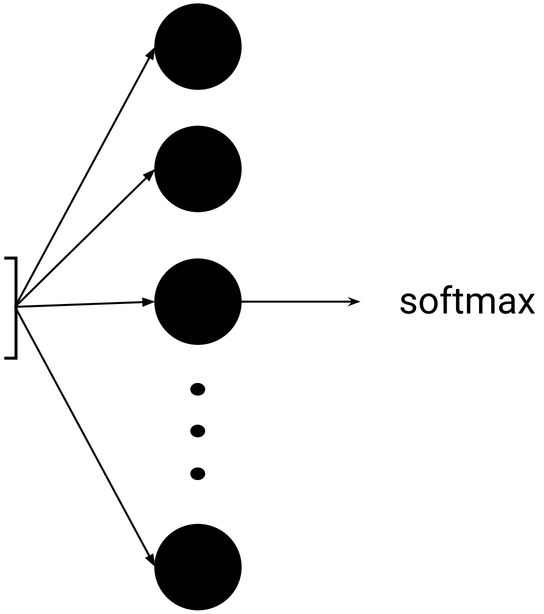
Deep Learning and NLP

- Sequences
  - Variable length
  - Relationships between elements of sequence

Deep Learning and NLP

- Sequences
    - Variable length
    - Relationships between elements of sequence
- Continuous Bag of Words (CBOW)-style Model

$$\begin{bmatrix} here \\ comes \\ the \end{bmatrix} \rightarrow \begin{bmatrix}\overrightarrow{a_i}\\ \overrightarrow{a_j}\\ \overrightarrow{a_k}\end{bmatrix} \rightarrow \left[(\overrightarrow{a_i}+\overrightarrow{a_j}+\overrightarrow{a_k})/\mathbf{3}\right]$$



Deep Learning and NLP

- Sequences
  - Variable length
  - Relationships between elements of sequence
- Continuous Bag of Words (CBOW)-style Model

Take average of features

$$\begin{bmatrix} here \\ comes \\ the \end{bmatrix} \rightarrow \begin{bmatrix} \overrightarrow{a_i} \\ \overrightarrow{a_j} \\ \overrightarrow{a_k} \end{bmatrix} \rightarrow \left[(\overrightarrow{a_i} + \overrightarrow{a_j} + \overrightarrow{a_k})/3 \right]$$

Deep Learning and NLP

- Sequences
    - Variable length (OVERCOME)
    - Relationships between elements of sequence (LOST)
- Continuous Bag of Words (CBOW)-style Model

Take average of features

$$\begin{bmatrix} here \\ comes \\ the \end{bmatrix} \rightarrow \begin{bmatrix} \overrightarrow{a_i} \\ \overrightarrow{a_j} \\ \overrightarrow{a_k} \end{bmatrix} \rightarrow \left[(\overrightarrow{a_i} + \overrightarrow{a_j} + \overrightarrow{a_k})/3 \right]$$

Deep Learning and NLP

- Sequences
  - Variable length
  - Relationships between elements of sequence
- Continuous Bag of Words (CBOW)
- 1D CNN
  - 1-dimensional filter

I am at track five. Here comes the train.

Filter size: 3

Deep Learning and NLP

- Sequences
  - Variable length
  - Relationships between elements of sequence
- Continuous Bag of Words (CBOW)
- 1D CNN
  - 1-dimensional filter

$$\begin{bmatrix} f_1 & f_2 & \dots & f_7 \end{bmatrix}$$

I am at track five. Here comes the train.

Filter size: 3

Deep Learning and NLP

- Sequences
  - Variable length
  - Relationships between elements of sequence
- Continuous Bag of Words (CBOW)
- 1D CNN
  - 1-dimensional filter

$$\begin{bmatrix} f_1 & f_2 & \ldots & f_7 \end{bmatrix}$$

I am at track five. Here comes the train.

Filter size: 3

Deep Learning and NLP

- Sequences
  - Variable length
  - Relationships between elements of sequence
- Continuous Bag of Words (CBOW)
- 1D CNN
  - 1-dimensional filter

$$\begin{bmatrix} f_1 & f_2 & \dots & f_7 \end{bmatrix}$$

I am at track five. Here comes the train.

Filter size: 3

Deep Learning and NLP

<u>I</u> <u>am</u> <u>at</u> <u>track</u> <u>five</u>. <u>Here</u> <u>comes</u> <u>the</u> <u>train.</u>

$$\begin{bmatrix} a_I^1 \\ a_I^2 \\ \vdots \\ a_I^{100} \end{bmatrix} \begin{bmatrix} a_{am}^1 \\ a_{am}^2 \\ \vdots \\ a_{am}^{100} \end{bmatrix} \begin{bmatrix} a_{at}^1 \\ a_{at}^2 \\ \vdots \\ a_{at}^{100} \end{bmatrix}$$

100-dim word embedding

Deep Learning and NLP

I am at track five. Here comes the train.

$$\begin{bmatrix} a_I^1 \\ a_I^2 \\ \vdots \\ a_I^{100} \end{bmatrix} \begin{bmatrix} a_{am}^1 \\ a_{am}^2 \\ \vdots \\ a_{am}^{100} \end{bmatrix} \begin{bmatrix} a_{at}^1 \\ a_{at}^2 \\ \vdots \\ a_{at}^{100} \end{bmatrix}$$

100-dim word embedding

“Width 3, 100 channels”

Deep Learning and NLP

I am at track five. Here comes the train.

$$\begin{bmatrix} a_I^1 \\ a_I^2 \\ \vdots \\ a_I^{100} \end{bmatrix} \begin{bmatrix} a_{am}^1 \\ a_{am}^2 \\ \vdots \\ a_{am}^{100} \end{bmatrix} \begin{bmatrix} a_{at}^1 \\ a_{at}^2 \\ \vdots \\ a_{at}^{100} \end{bmatrix} * \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ \vdots & \vdots & \vdots \\ w_{100,1} & w_{100,2} & w_{100,3} \end{bmatrix}$$

100-dim word embedding

Filter size 3, 100 channels

“Width 3, 100 channels”

$$\begin{bmatrix} f_1 & f_2 & \dots & f_7 \end{bmatrix}$$

I am at track five. Here comes the train.

$$\begin{bmatrix} a_{am}^{1} \\ a_{am}^{2} \\ \vdots \\ a_{am}^{100} \end{bmatrix} \begin{bmatrix} a_{at}^{1} \\ a_{at}^{2} \\ \vdots \\ a_{at}^{100} \end{bmatrix} \begin{bmatrix} a_{track}^{1} \\ a_{track}^{2} \\ \vdots \\ a_{track}^{100} \end{bmatrix} * \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ \vdots & \vdots & \vdots \\ w_{100,1} & w_{100,2} & w_{100,3} \end{bmatrix}$$

100-dim word embedding

Filter size 3, 100 channels

“Width 3, 100 channels”

$$\begin{bmatrix} f_1 & f_2 & \dots & f_7 \end{bmatrix}$$

<u>I</u> <u>am</u> <u>at</u> <u>track</u> <u>five</u>. <u>Here</u> <u>comes</u> <u>the</u> <u>train.</u>

Length 9 sequence embedding, 100 channels

Length 7 sequence of features, 50 channels

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{19} \\ \vdots & & & \vdots \\ a_{100,1} & a_{100,2} & \cdots & a_{100,9} \end{bmatrix} \longrightarrow \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{17} \\ \vdots & & & \vdots \\ f_{50,1} & f_{50,2} & \cdots & f_{50,7} \end{bmatrix}$$

50 filters of width 3 with 100 channels

Deep Learning and NLP

- Sequences
  - Variable length
  - Relationships between elements of sequence
- Continuous Bag of Words (CBOW)
- 1D CNN
- Recurrent Neural Network (RNN)
  - Keep track of a hidden state vector of features as you move along a sequence
  - Sequence length agnostic

Deep Learning and NLP

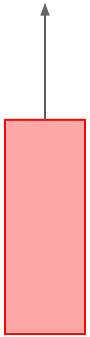
- Sequences
    - Variable length
    - Relationships between elements of sequence
- Continuous Bag of Words (CBOW)
- 1D CNN
- Recurrent Neural Network (RNN)
    - Keep track of a hidden state vector of features as you move along a sequence
    - Sequence length agnostic
- Diagrams shown without bias term (optional)

Deep Learning and NLP

- Vanilla RNN

Input sequence $(x_1, x_2, \ldots, x_N)$

$$\overrightarrow{a_i} = embedding(x_i)$$



$\overrightarrow{a_i}$

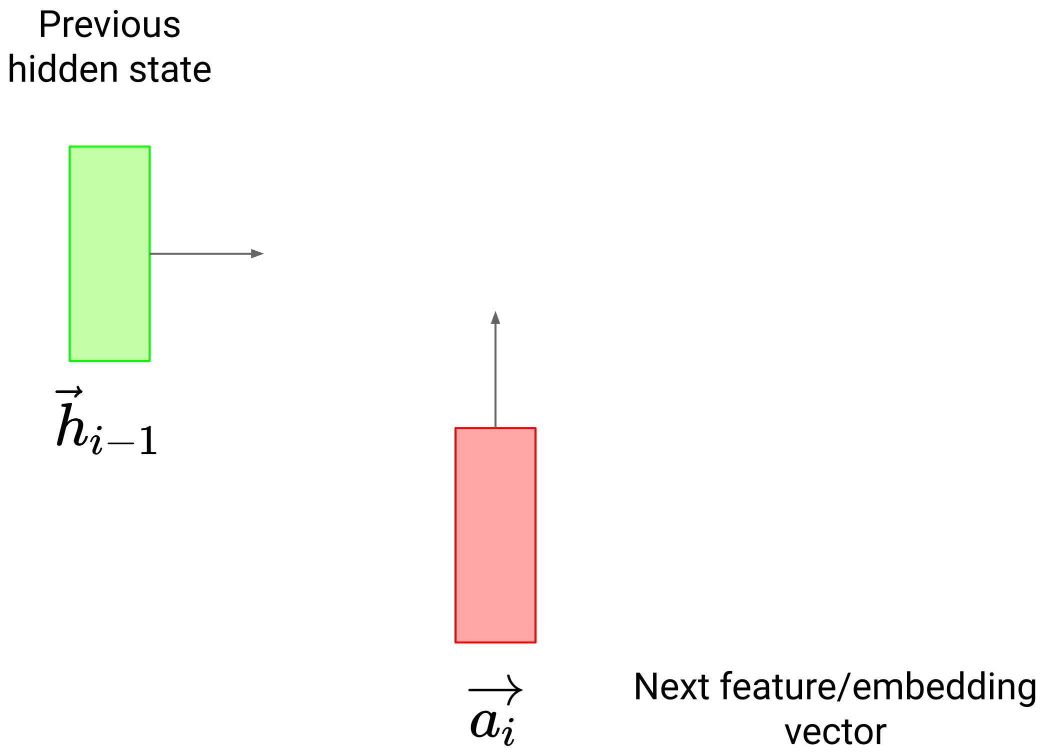
Next feature/embedding vector

Deep Learning and NLP

- Vanilla RNN

Input sequence $(x_1, x_2, \ldots, x_N)$

$$\overrightarrow{a_i} = embedding(x_i)$$

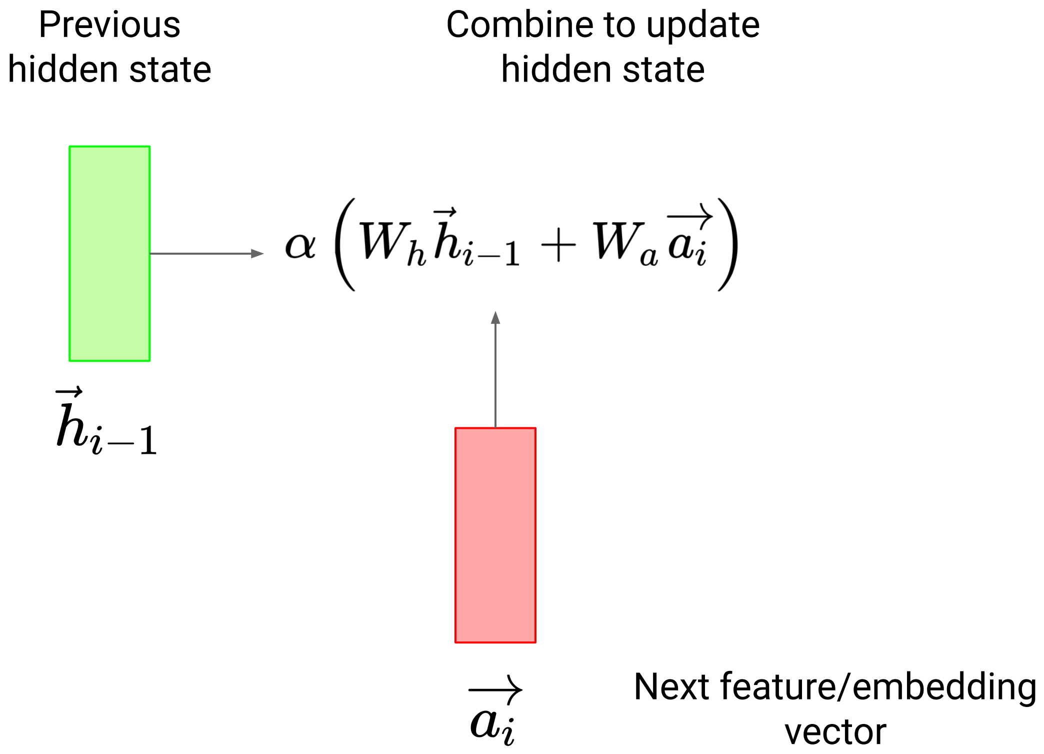


Deep Learning and NLP

- Vanilla RNN

Input sequence $(x_1, x_2, \ldots, x_N)$

$$\overrightarrow{a_i} = embedding(x_i)$$

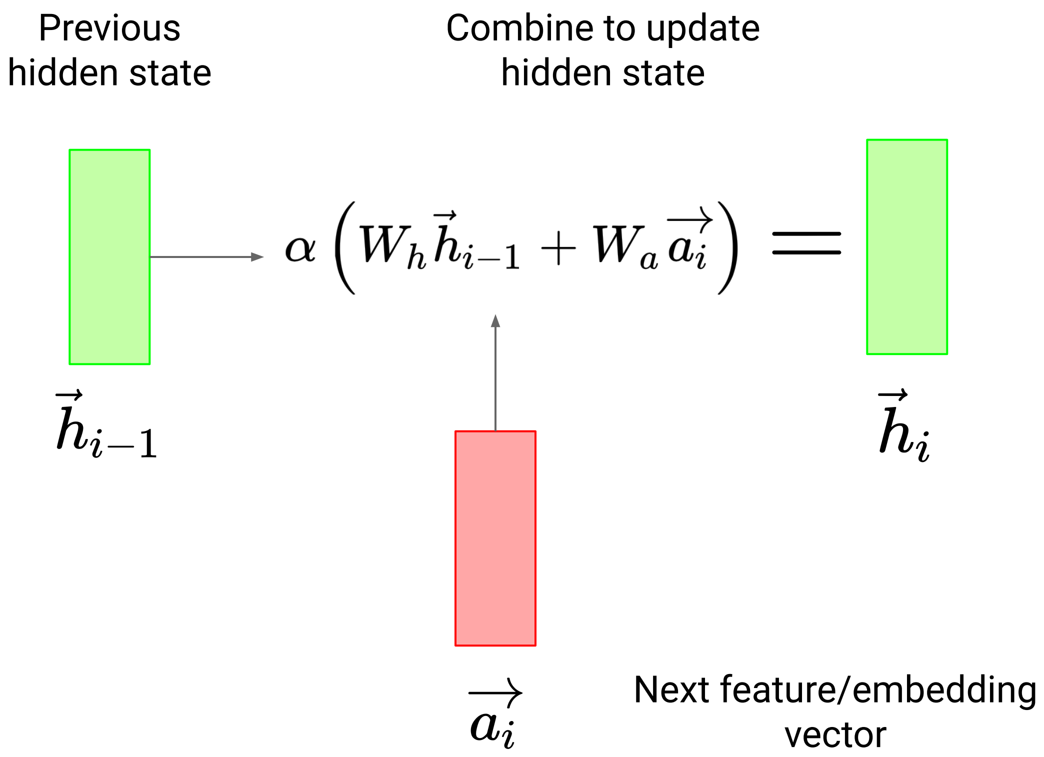


Deep Learning and NLP

- Vanilla RNN

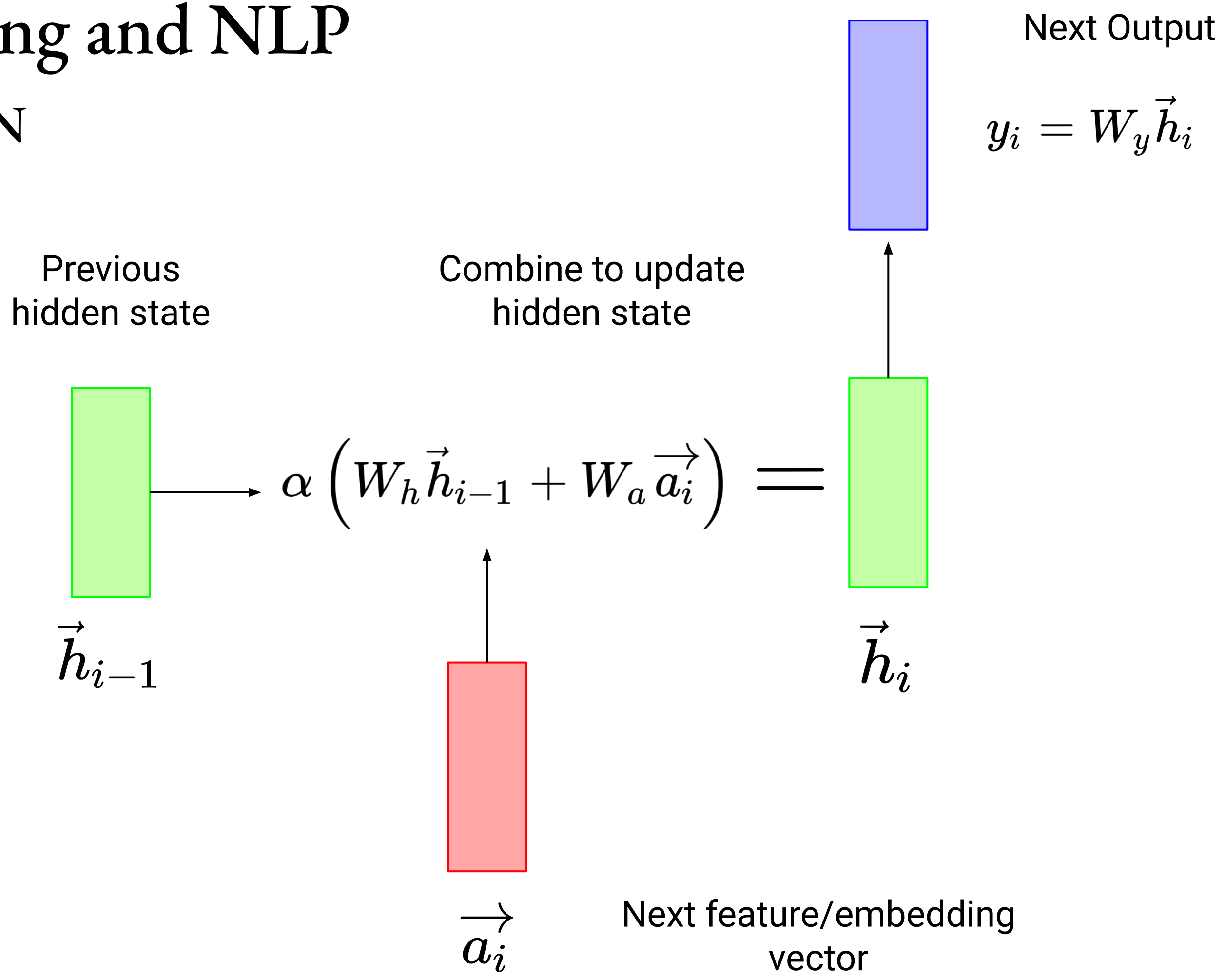
Input sequence $(x_1, x_2, \ldots, x_N)$

$$\overrightarrow{a_i} = embedding(x_i)$$



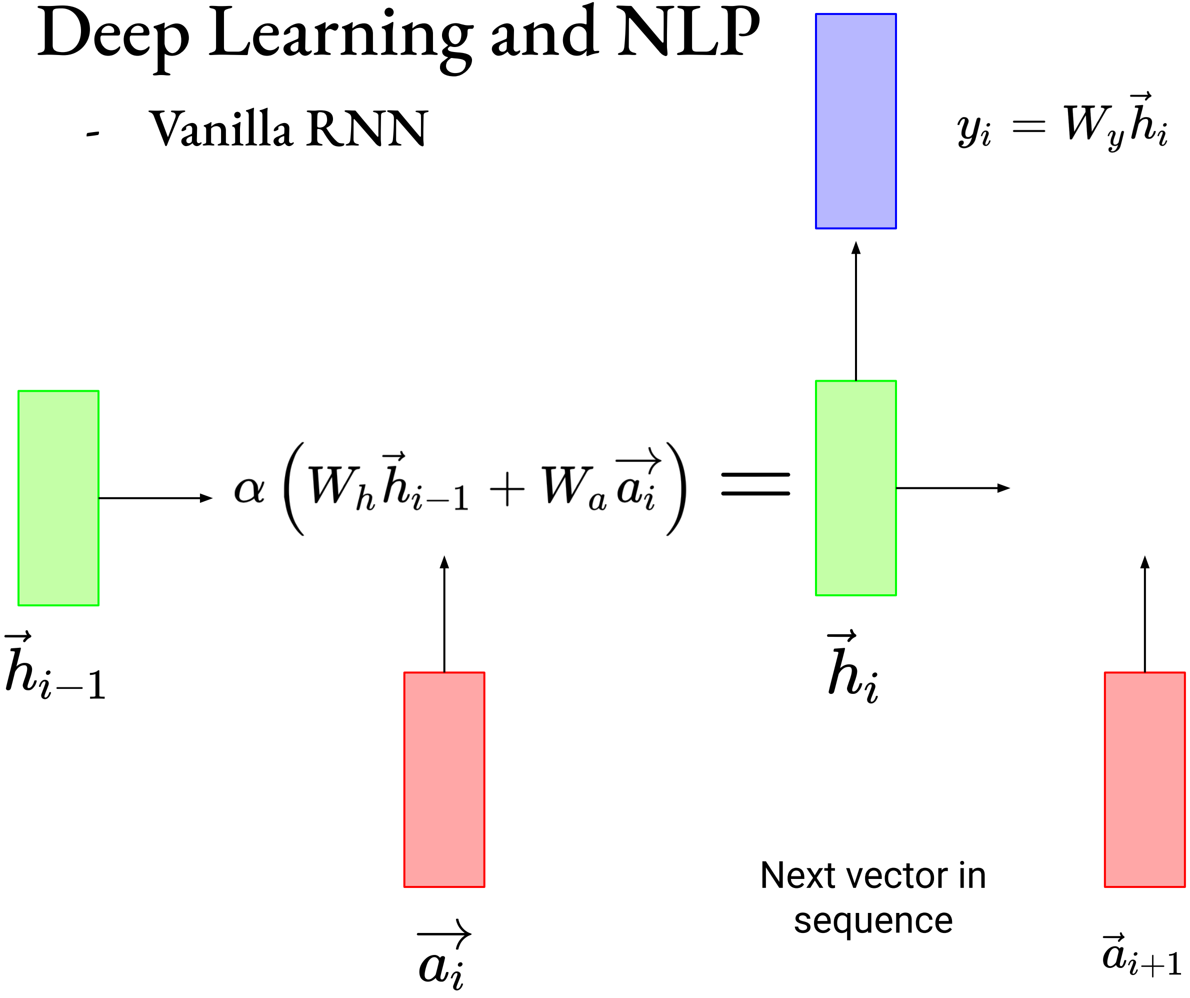
Deep Learning and NLP

- Vanilla RNN



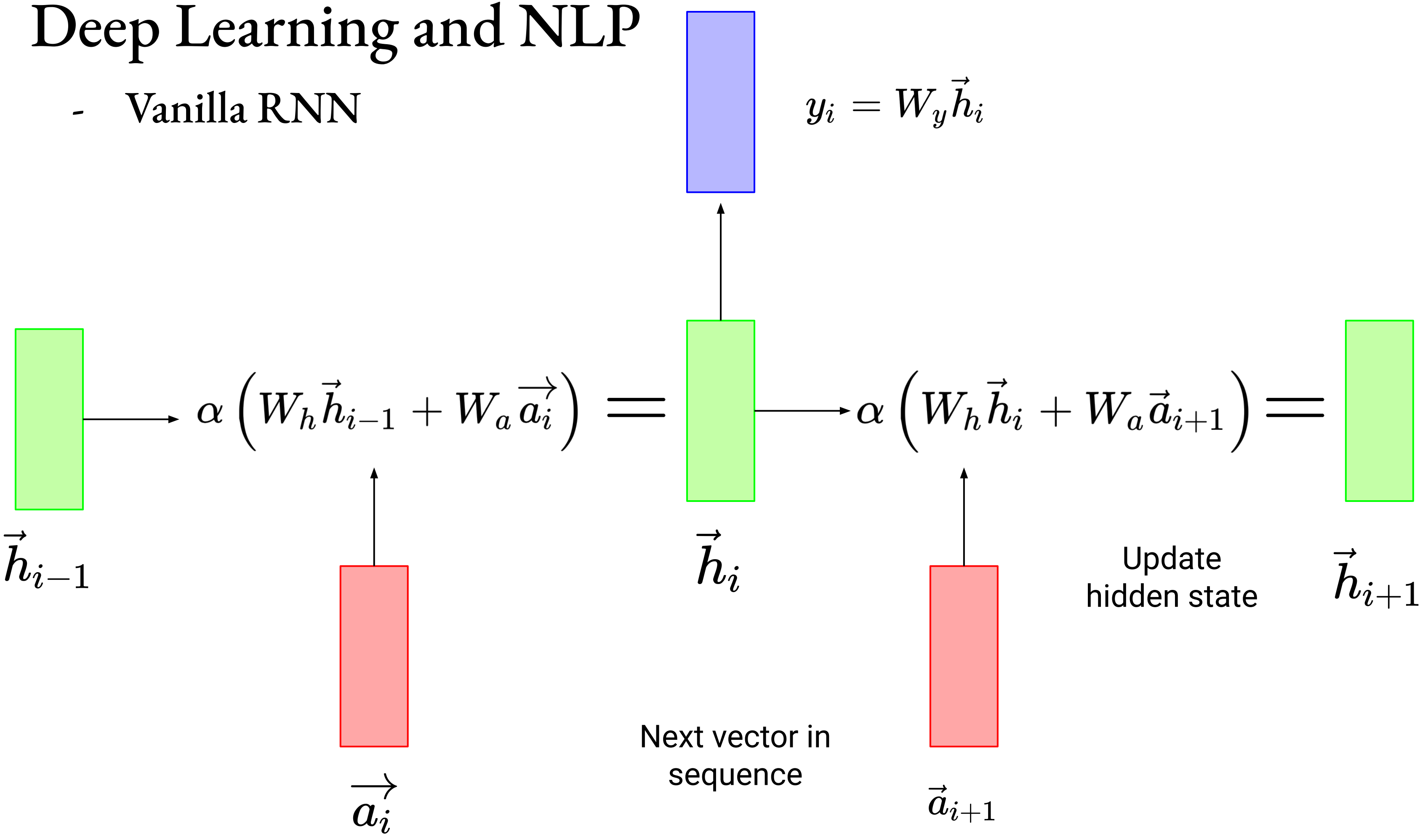
Deep Learning and NLP

- Vanilla RNN



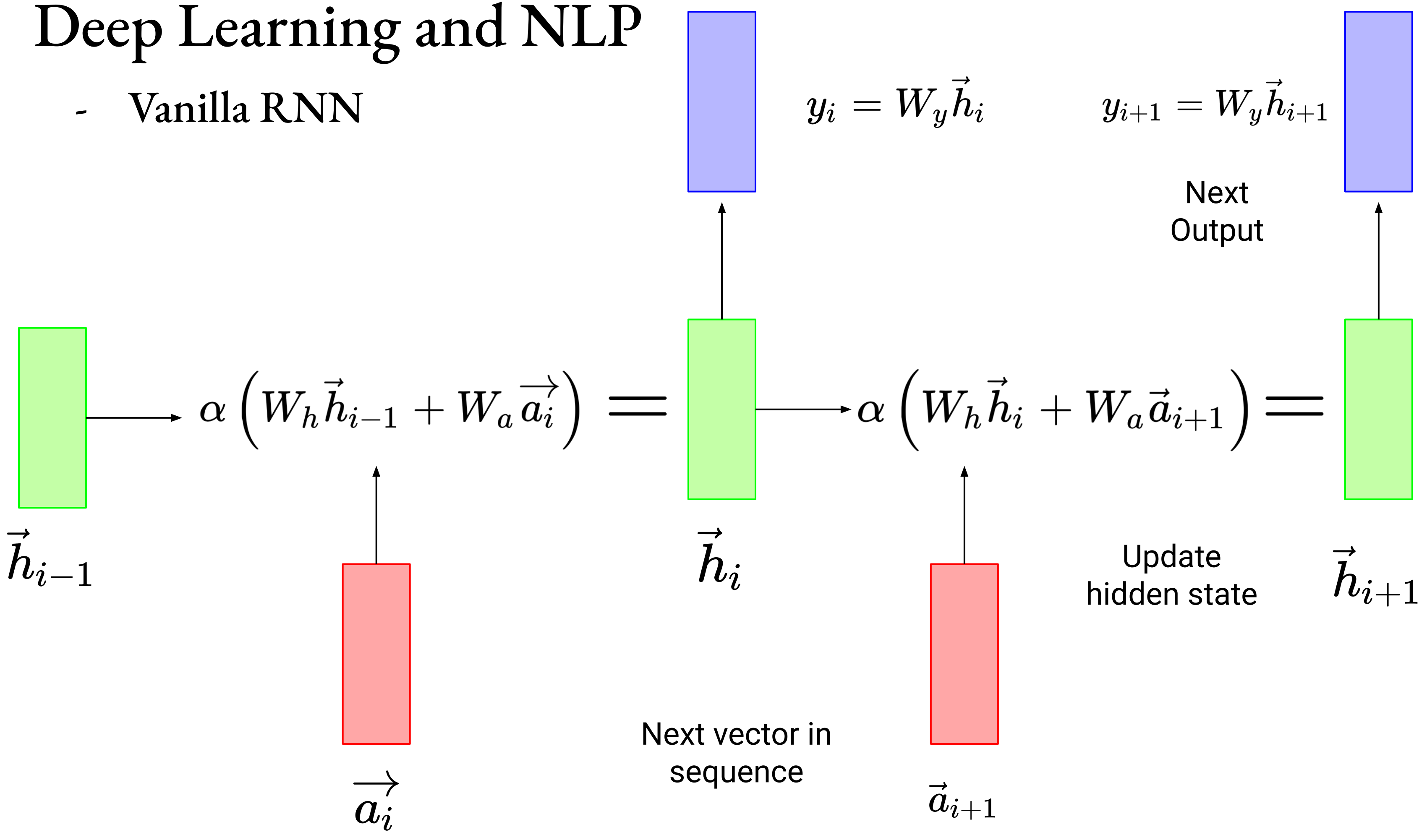
Deep Learning and NLP

- Vanilla RNN



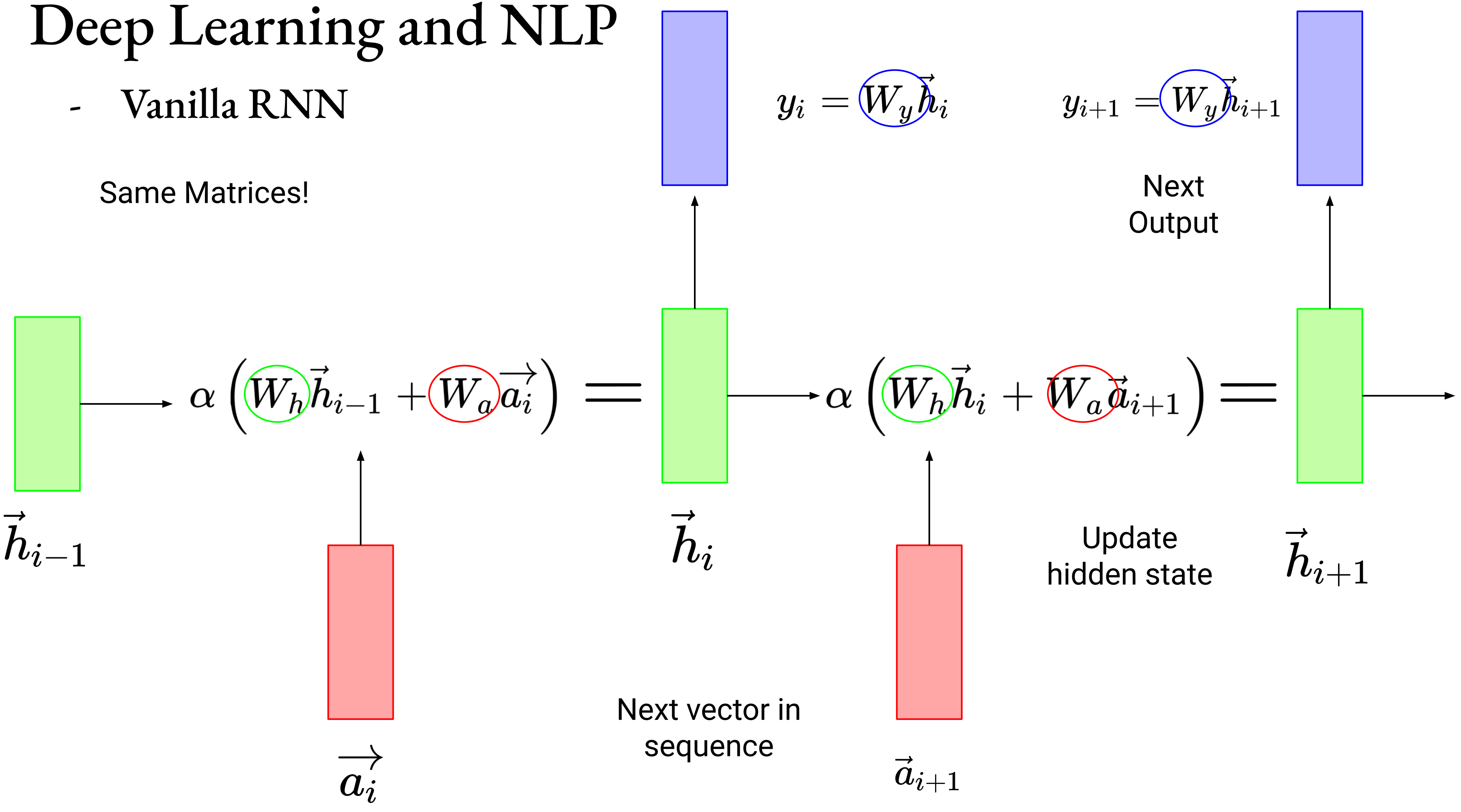
Deep Learning and NLP

- Vanilla RNN



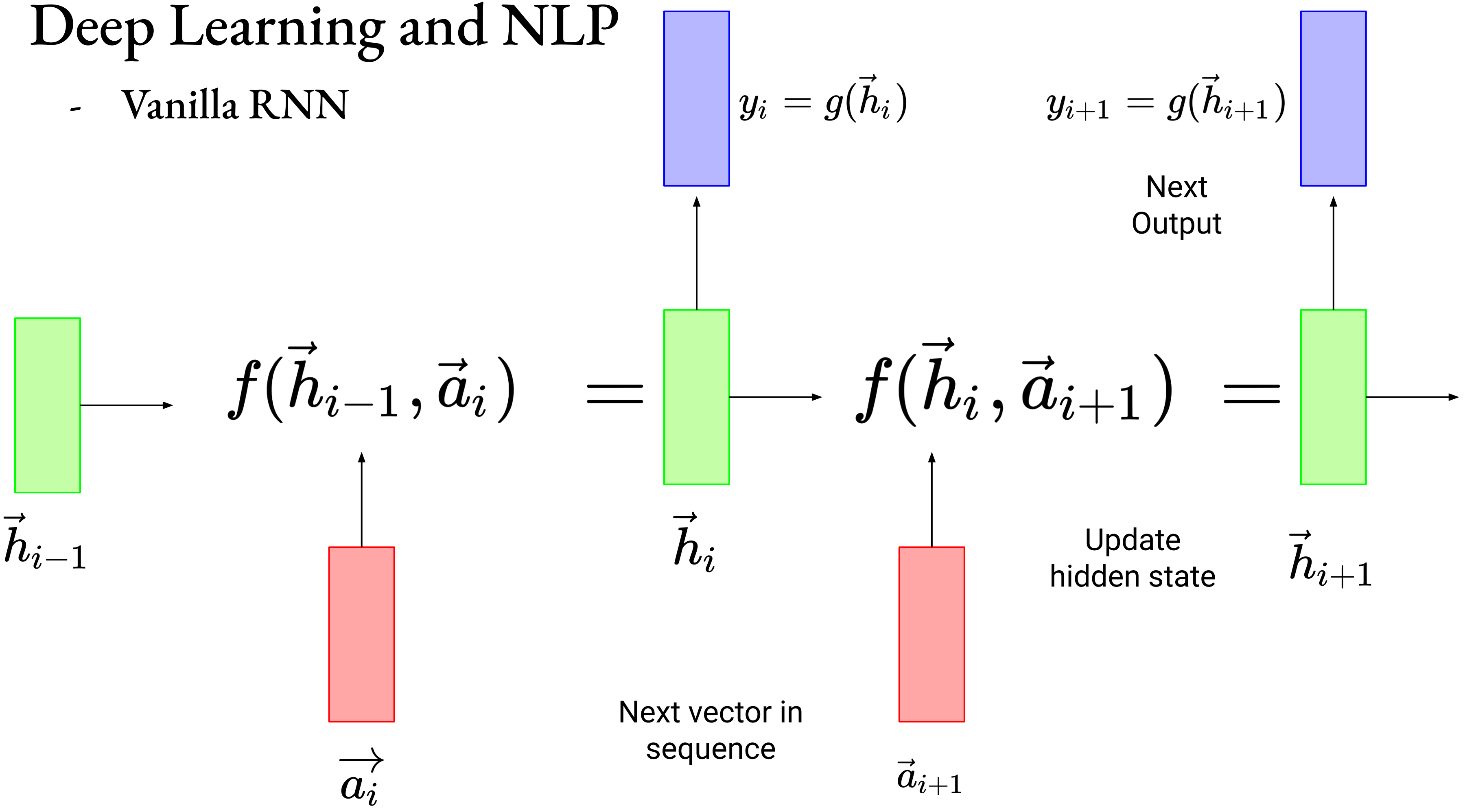
Deep Learning and NLP

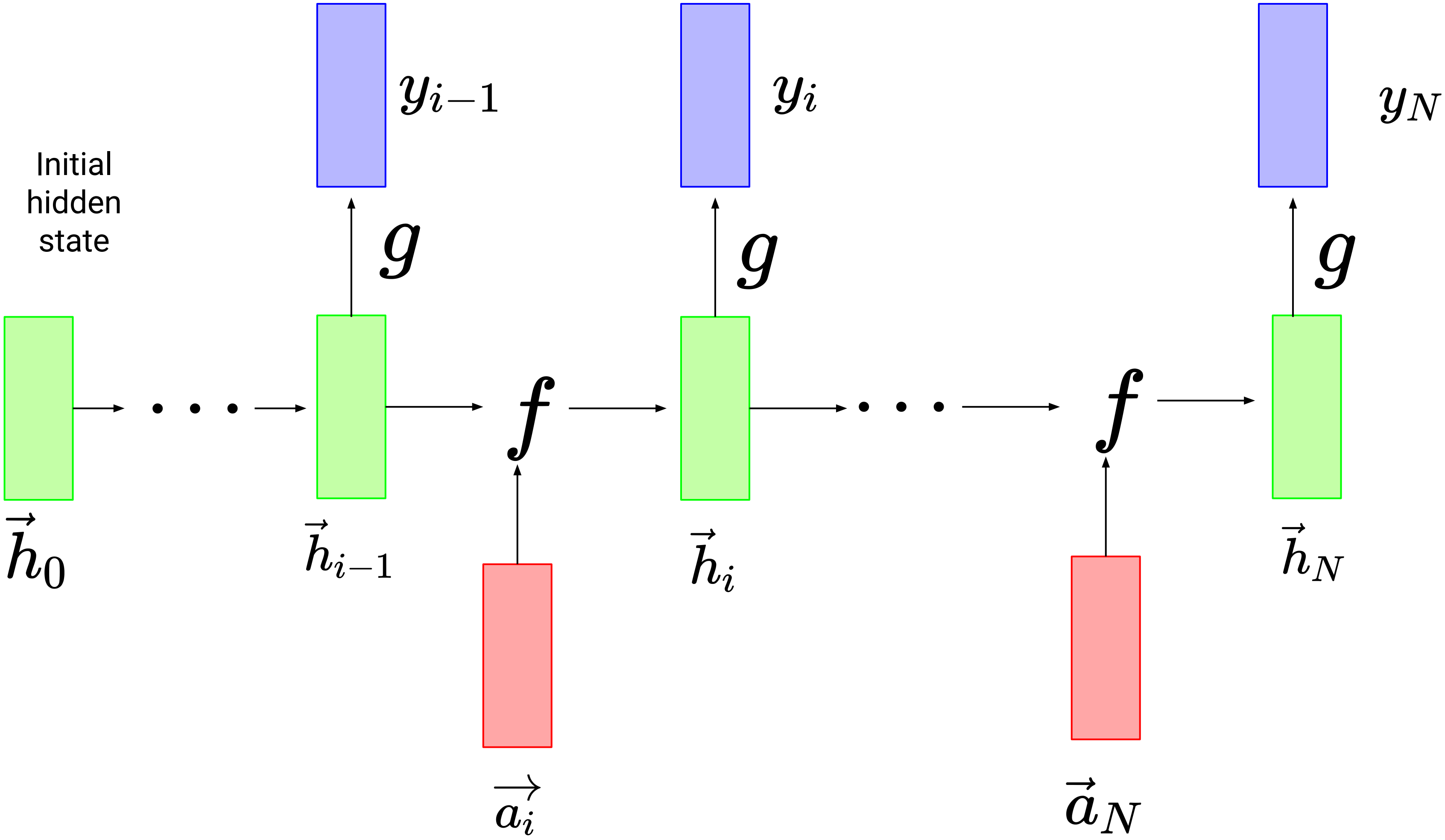
- Vanilla RNN



Deep Learning and NLP

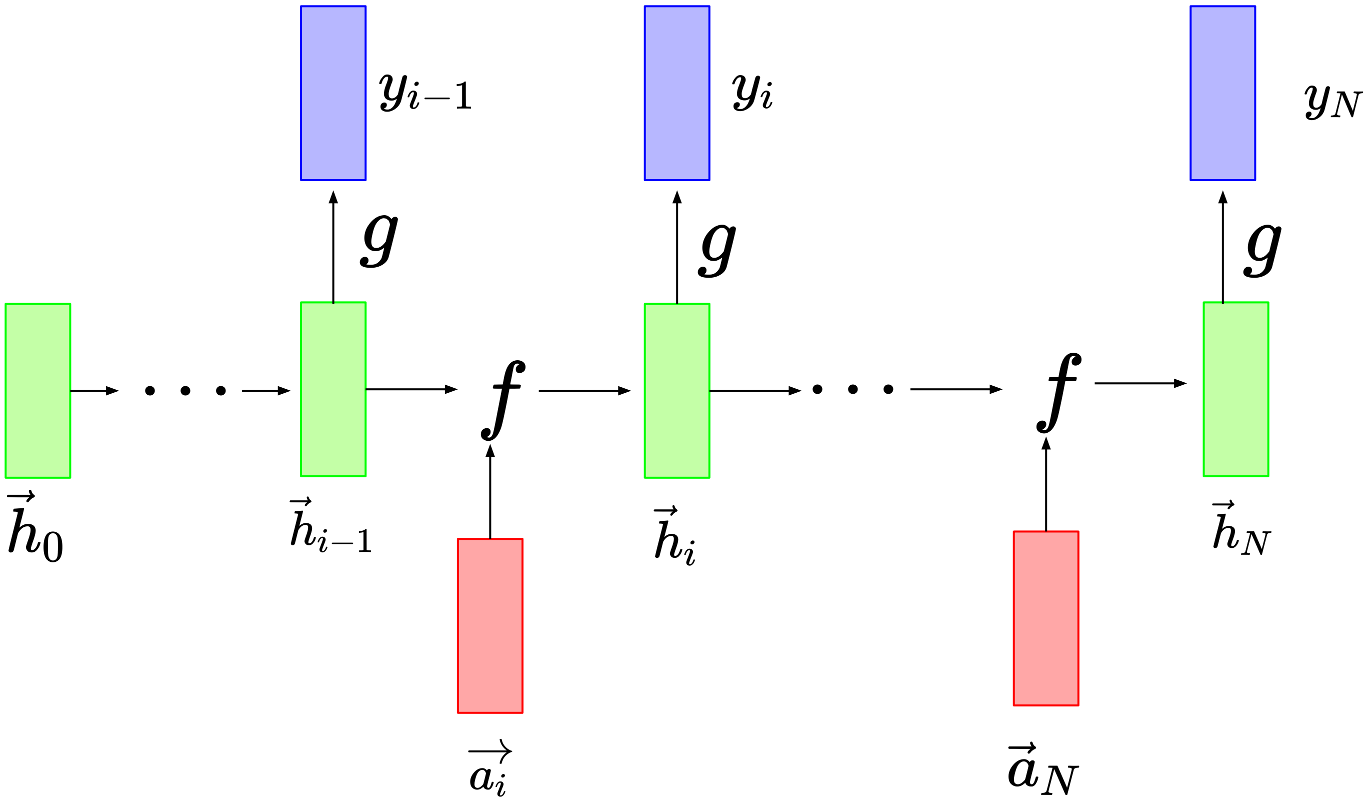
- Vanilla RNN



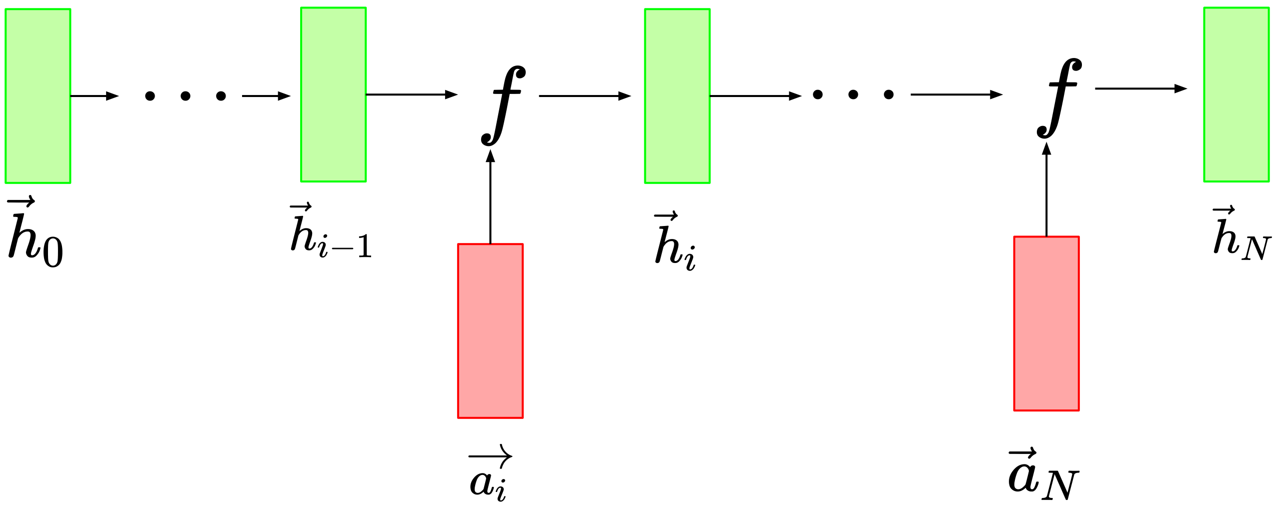


Initial
hidden
state
y_{i-1}
y_i
y_N
g
g
g
f
f
$\vec{h}_0$
$\vec{h}_{i-1}$
$\vec{h}_i$
$\vec{h}_N$
$\overrightarrow{a_i}$
$\vec{a}_N$

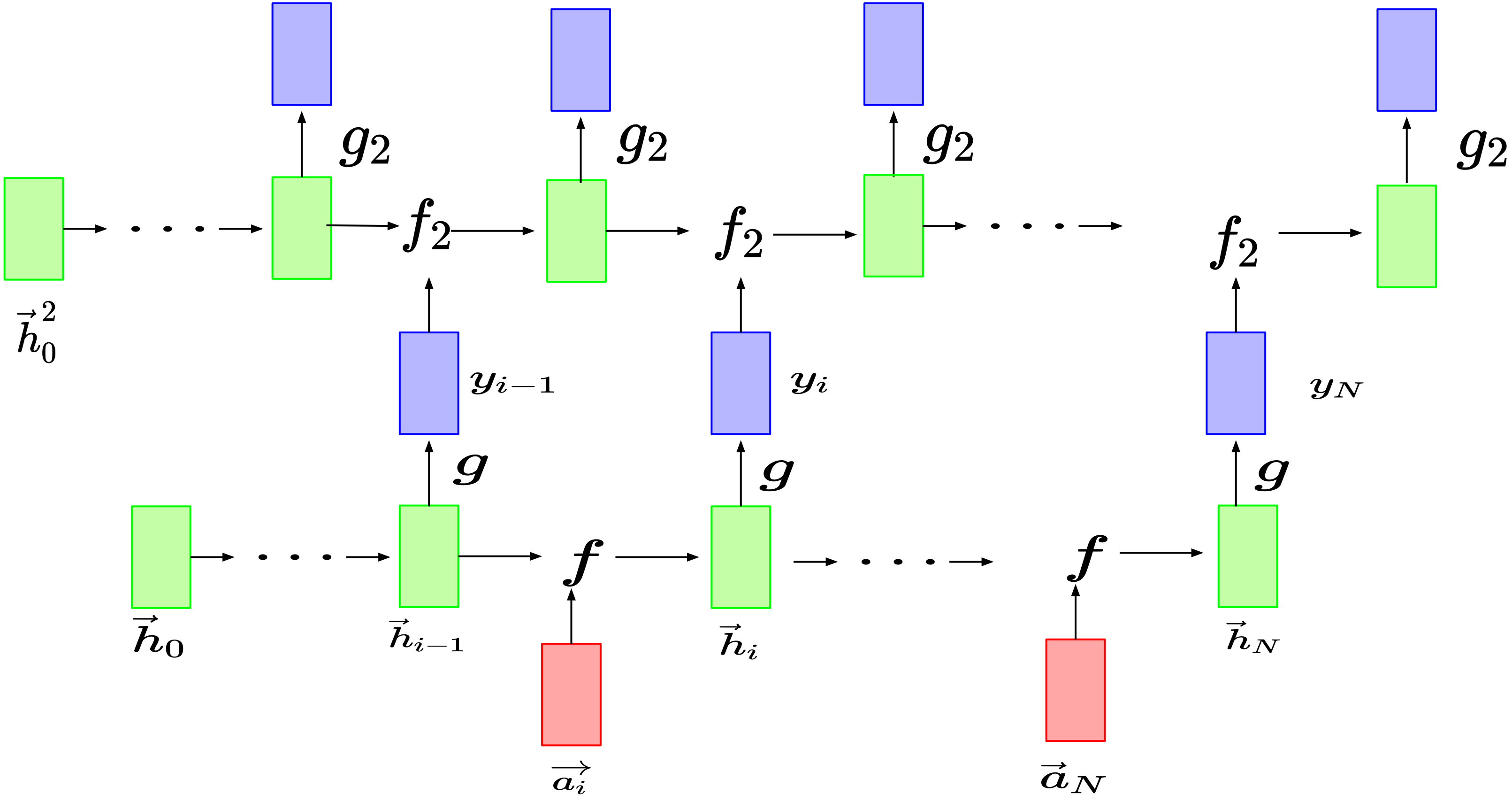
- Can either train on output sequence or discard



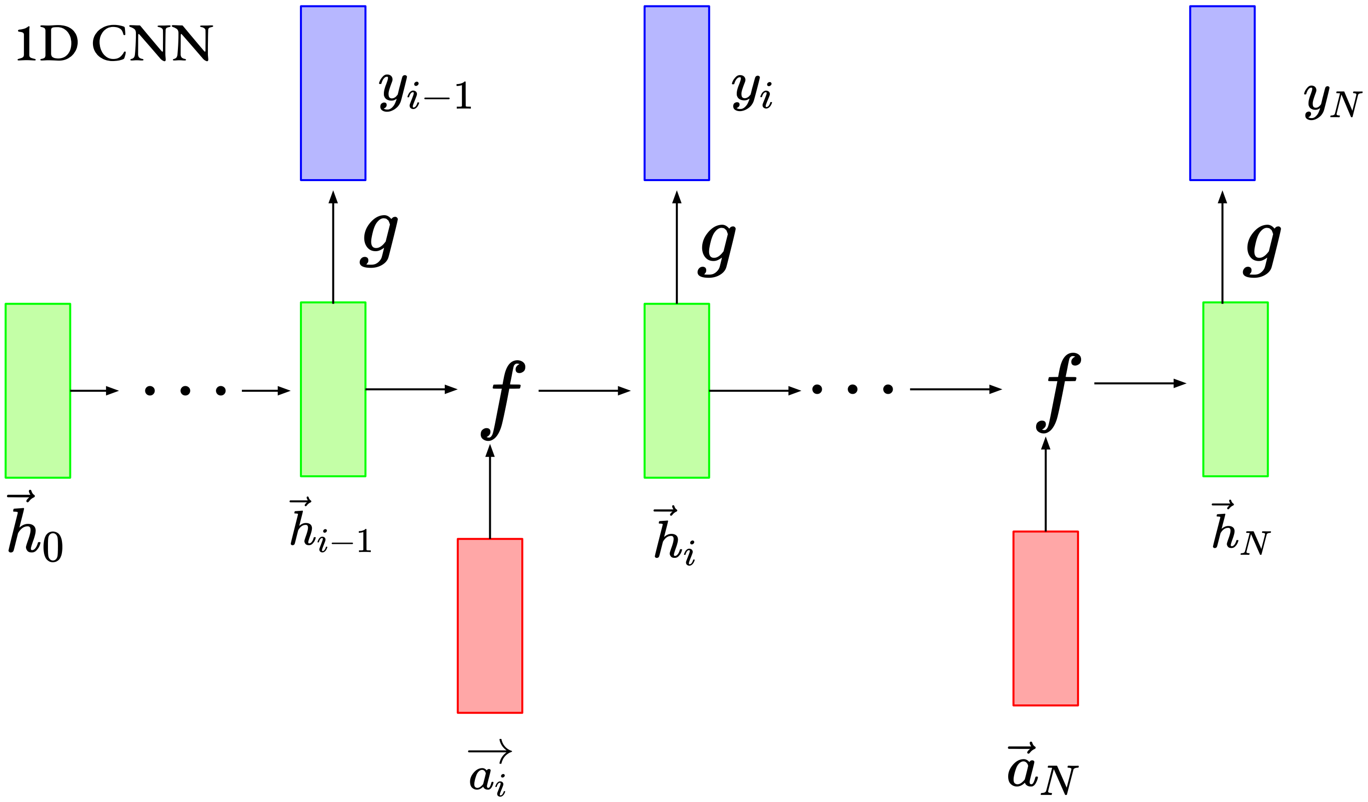
- Can either train on output sequence or discard



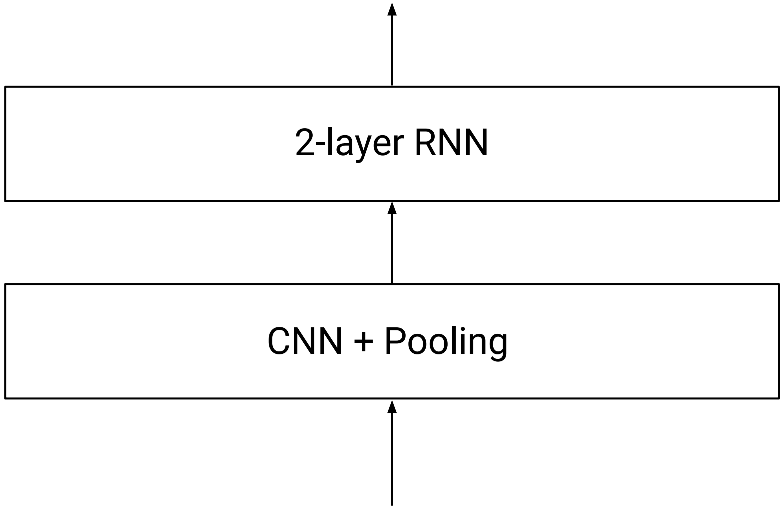
- Can either train on output sequence or discard
- Stack RNNs



- Can either train on output sequence or discard
- Stack RNNs
- Input can be from 1D CNN



- Can either train on output sequence or discard
- Stack RNNs
- Input can be from 1D CNN



$$(x_1, x_2, \ldots, x_N)$$

Twin Neural Networks: HW2

- Use RNNs in PyTorch to determine whether questions are redundant
- Use a Twin Neural Network design
  - Create representation using same parameters of the two inputs
  - Compare representations to determine similarity