Recurrent Neural Networks

Gavin Wenzel, Dylan Lai, and Hao Zhang

Uses of Recurrent Neural Networks (RNN)

Natural Language Processing

Translating between human and computer language

Chatbots

Image Captioning

Sentiment Analysis

Predictive Text

Machine Translation

Speech Recognition

ChatGPT

Autocorrect

Grammarly

Google Translate

Siri

Time Series Analytics

Performing analysis on sequences of data

Weather Forecasting

Market Forecasting

Anomaly Detection

	x		y
auto complete	not interested at	→	this time
translation	how are you?	→	क्या हाल है?
NER	Rudolph Smith bought 1000 shares of tesla Inc. in March 2020	→	Person: Rudolph Smith bought 1000 shares of Company: tesla Inc. in time: March 2020
Sentiment Analysis	Not only the fan was expensive, but it was broken when it arrived.	→	★☆☆☆☆

Prediction Based on Context

Without RNN:

The response might be something bizarre like *"Run Away!"*

With RNN:

Based on the context, the RNN predicts *"Boo who?"*

Friend: *"Knock, knock!"*
You: *"Who's there?"*
Friend: *"Boo."*

Predicting Autocomplete

Typing: *"I want to order pizza with…"*

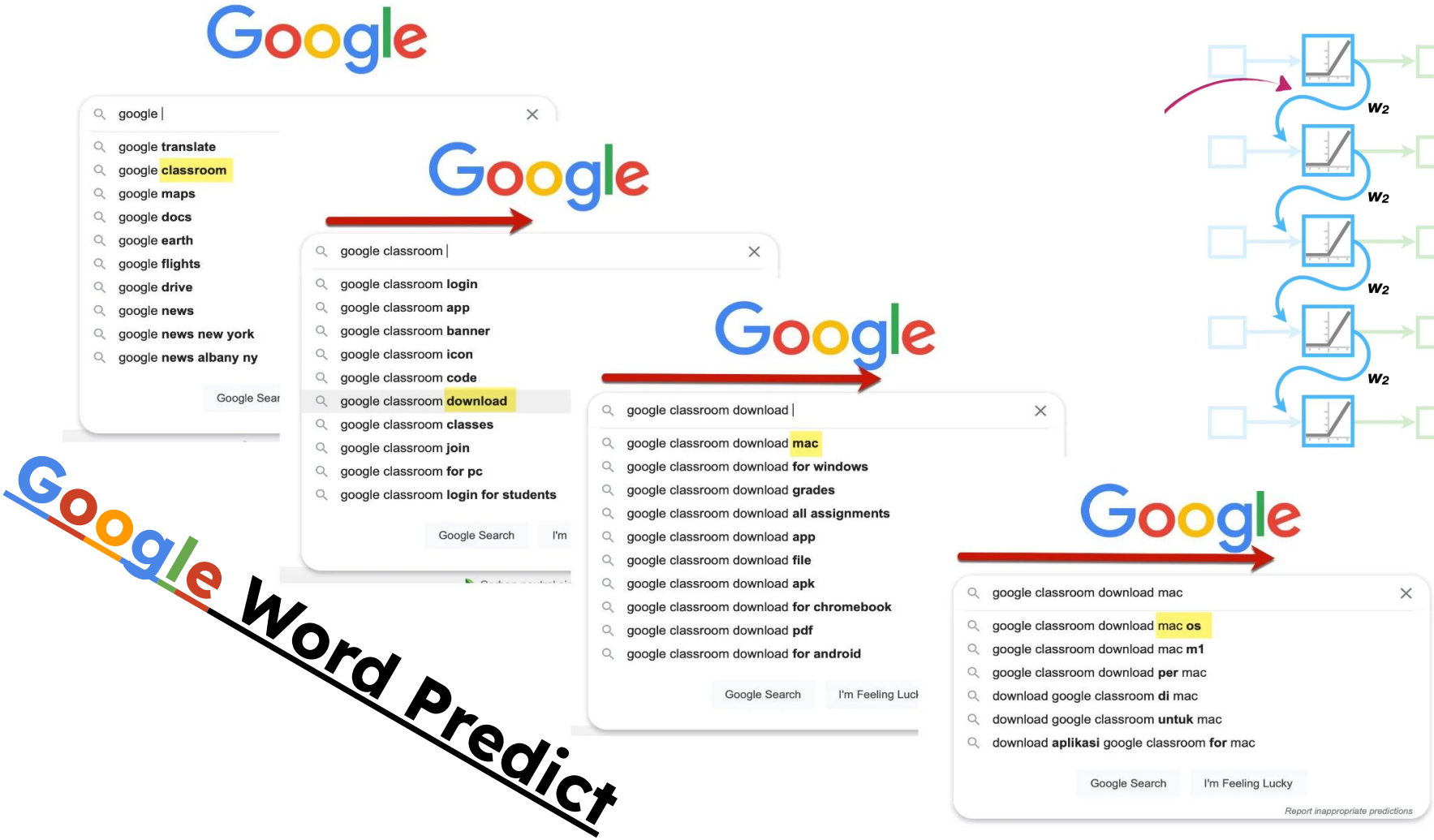
Ideal for tasks where context is important

Without RNN:

Autocomplete:: *"…salute and Italy."*

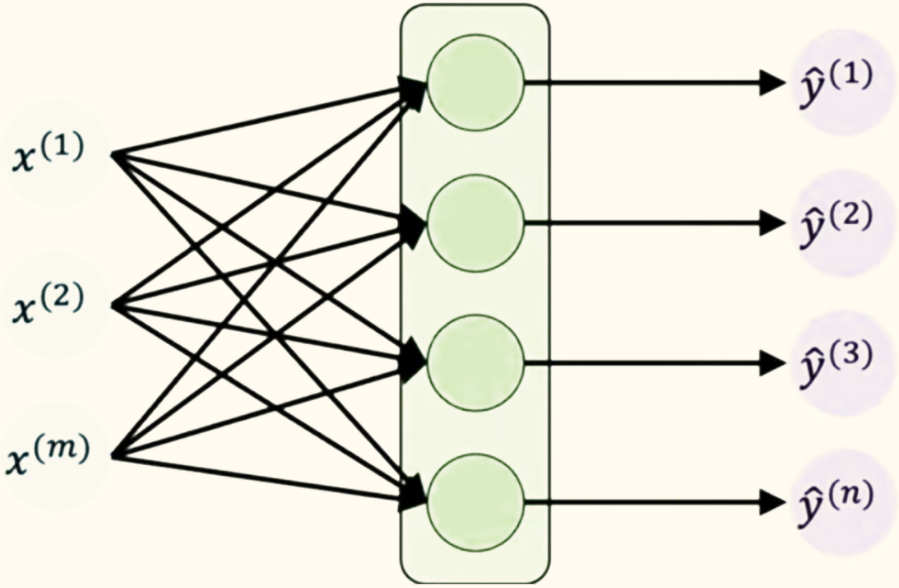
With RNN:

Autocomplete: *"…extra cheese and mushrooms."*

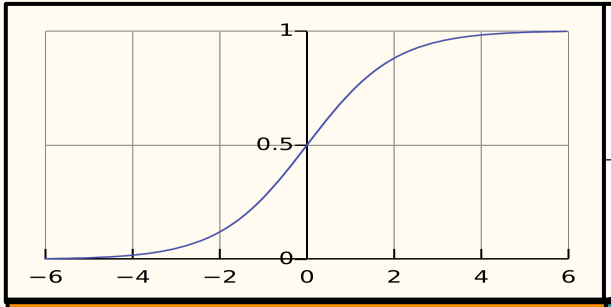

Google
google
google translate
google classroom
google maps
google docs
google earth
google flights
google drive
google news
google news new york
google news albany ny
Google Sear
Google
google classroom
google classroom login
google classroom app
google classroom banner
google classroom icon
google classroom code
google classroom download
google classroom classes
google classroom join
google classroom for pc
google classroom login for students
Google Search
I'm
Google
google classroom download
google classroom download mac
google classroom download for windows
google classroom download grades
google classroom download all assignments
google classroom download app
google classroom download file
google classroom download apk
google classroom download for chromebook
google classroom download pdf
google classroom download for android
Google Search
I'm Feeling Luck
Google
google classroom download mac
google classroom download mac os
google classroom download mac m1
google classroom download per mac
download google classroom di mac
download google classroom untuk mac
download aplikasi google classroom for mac
Google Search
I'm Feeling Lucky
Report inappropriate predictions
Google Word Predict
w2
w2
w2
w2

Neural Networks

- Composed of node layers
- Each node uses a linear regression model of weight*input+bias
- Data passes through layers in a Feed Forward Network and through Activation functions
- Weights are updated using backpropagation
- Rely on training data
- Multiple types of NN including RNN, LTSM, GRU, and transformers



Activation Functions



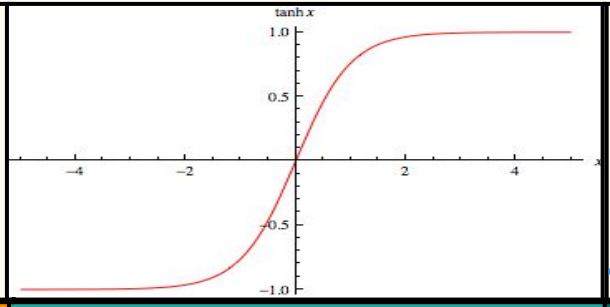
Sigmoid

x is the input

e is the base of natural log

Output between (0,1)

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

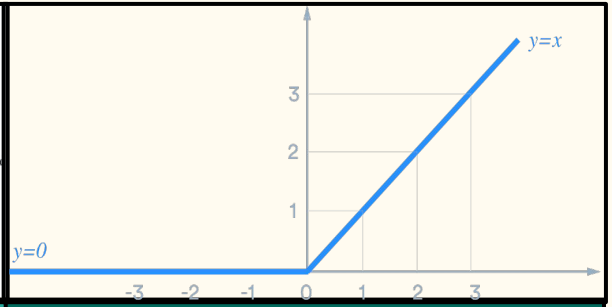


Tanh

Output between
(-1, 1) zero-centered

f(x) = tanh(x)

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^{2x} - 1}{e^{2x} + 1}$$

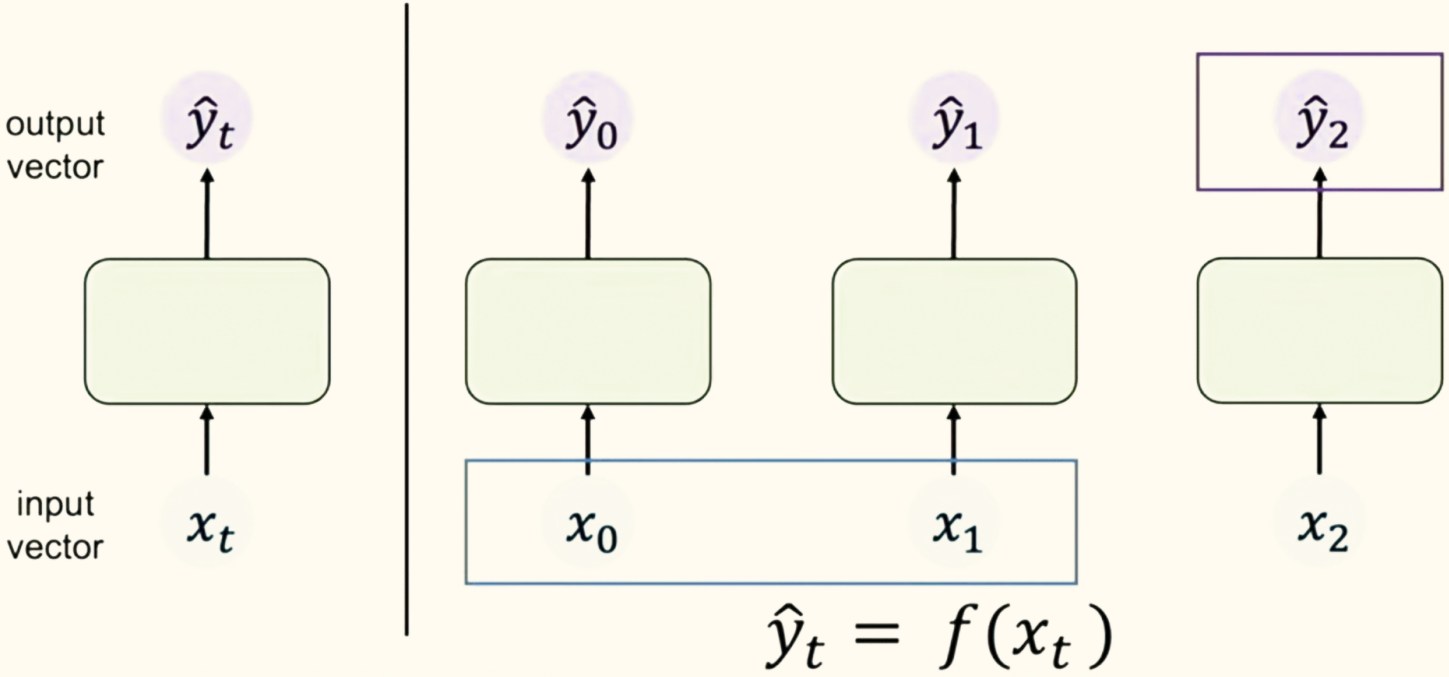


Relu

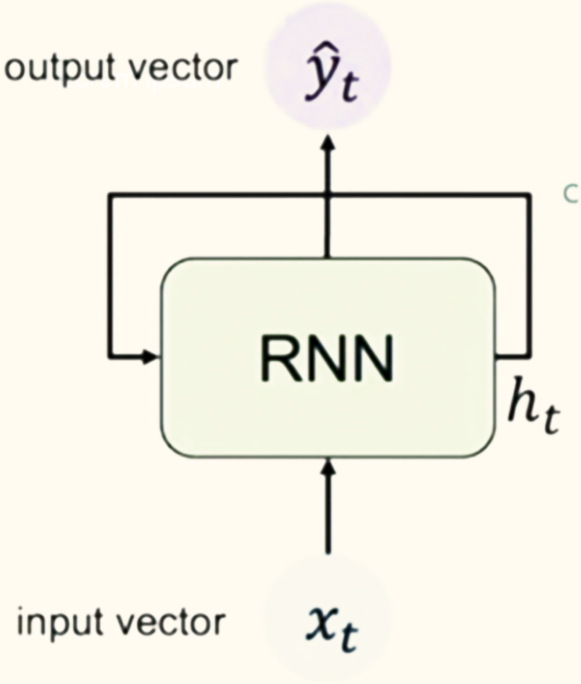
f(x) = max(0,x)

Always positive

Single Isolated Time Steps



Recurrent Neural Networks



$$h_t = f_W(x_t, h_{t-1})$$

cell state

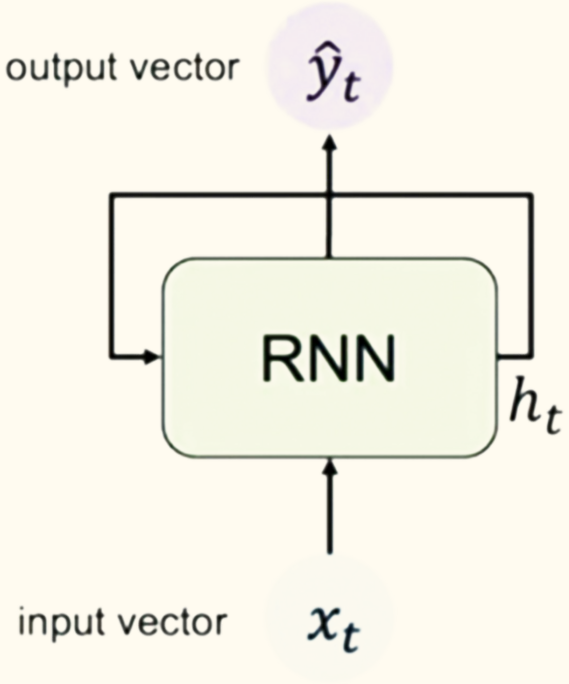
function with weights W

input

old state

RNNS have a state (ht), that is updated at each time step as a sequence is processed

Updating the Hidden State



Output Vector

$$\hat{y}_t = \boldsymbol{W}_{\boldsymbol{hy}}^{\boldsymbol{T}} h_t$$

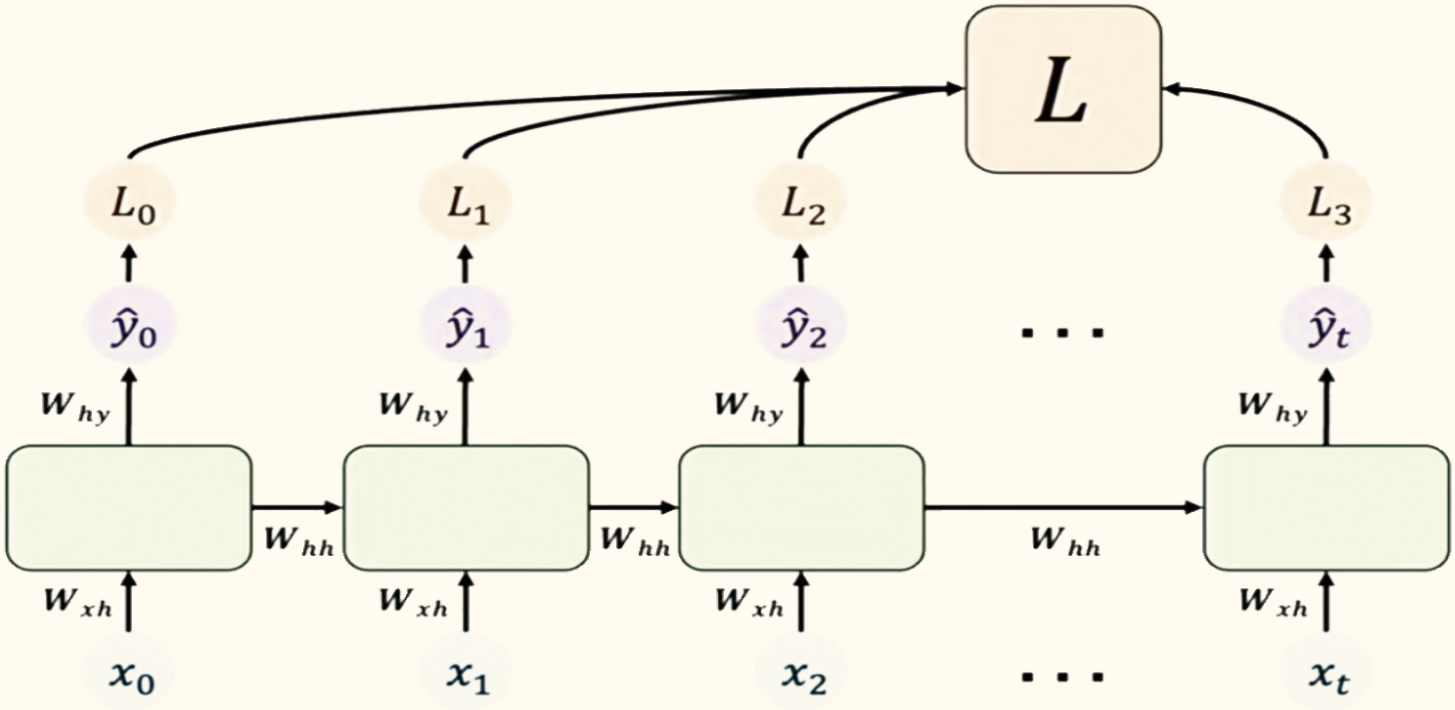
Update Hidden State

$$h_t = \tanh(\boldsymbol{W}_{\boldsymbol{hh}}^{\boldsymbol{T}} h_{t-1} + \boldsymbol{W}_{\boldsymbol{xh}}^{\boldsymbol{T}} x_t)$$

Input Vector

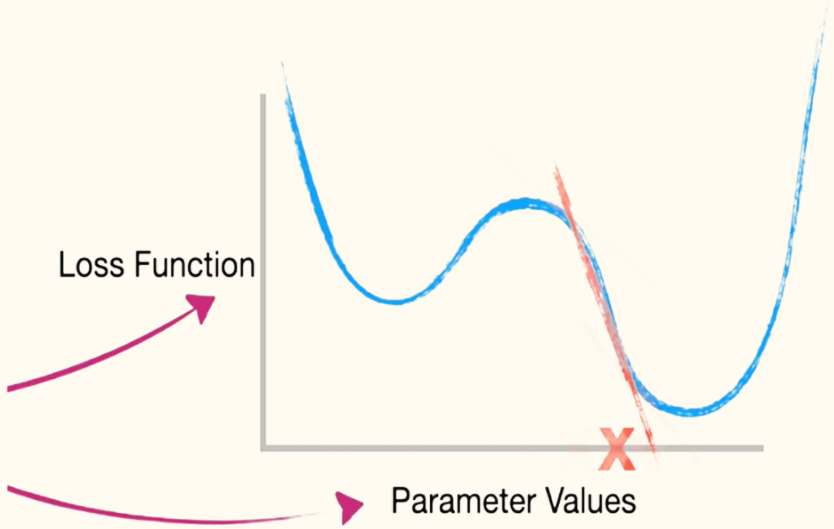
$$x_t$$

Updating the Model



Gradient Descent

- During Back Propagation we first find the gradients, or derivatives of the loss with respect to each parameter
- We then plug the gradients into a Gradient Descent Algorithm to find parameter values that minimize a loss function such as MSE



Predicting Stock Price

Without RNN:

Prediction using Average/Median/Feedforward Neural Networks : $12

$10, $11, $12, $13, $14, $?

Traditional methods process data in one directional, from input to output, without retaining information from previous inputs. This makes them only suitable for tasks with inputs that are independent of each other

With RNN:

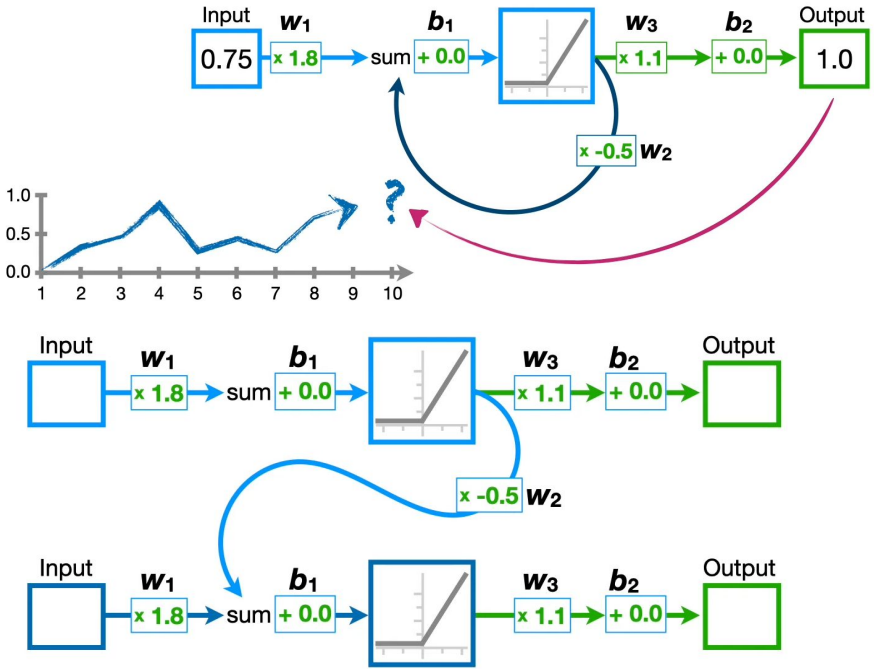
Predicts Next In Sequence: $15

RNNs preserve and pass on the relevant information of previous inputs, allowing for sequential pattern recognition, and enabling it to learn temporal dependencies

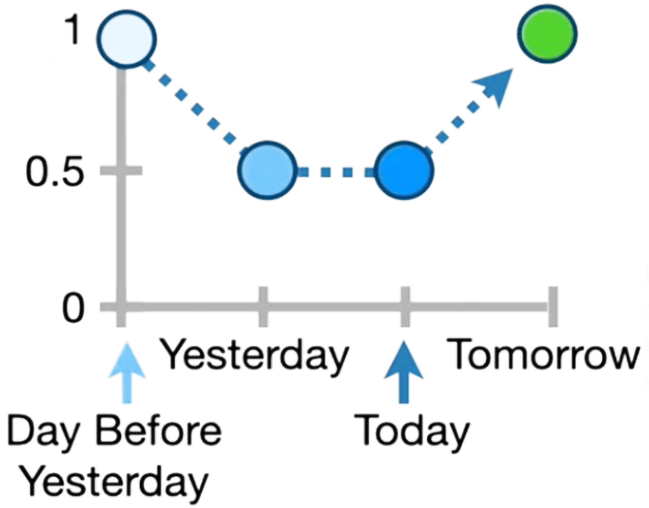
Recurrent “Unrolling”

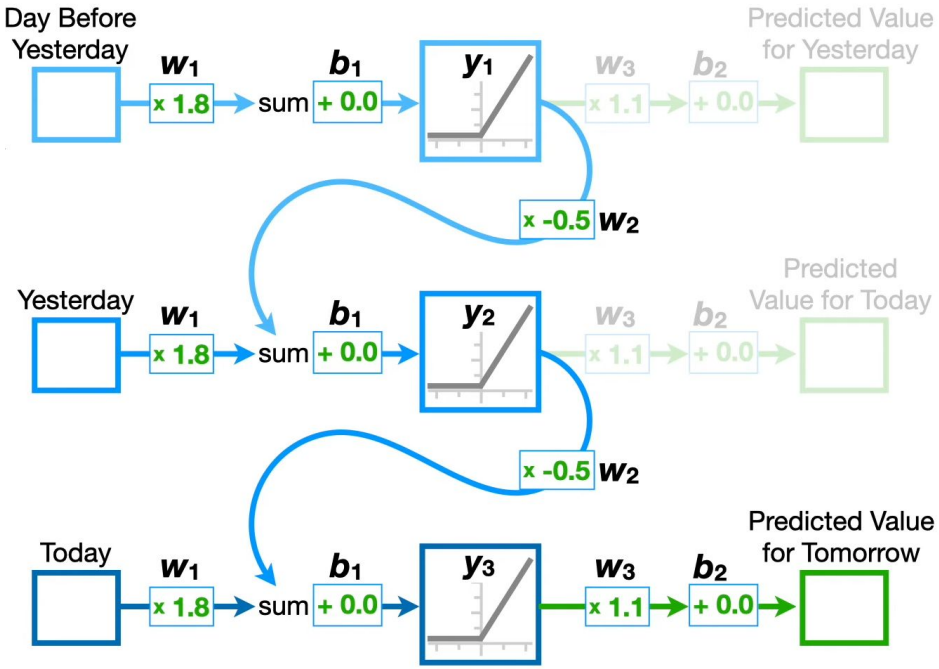
- Feedback Loop
- Time Sequence
- Hidden state, “memory state”, serves to retain essential information from previous inputs, and incorporates that info into the next iteration, which influences future outcomes

- Note: Each unrolled neural network is actually still the same one -> all weights and biases stay the same

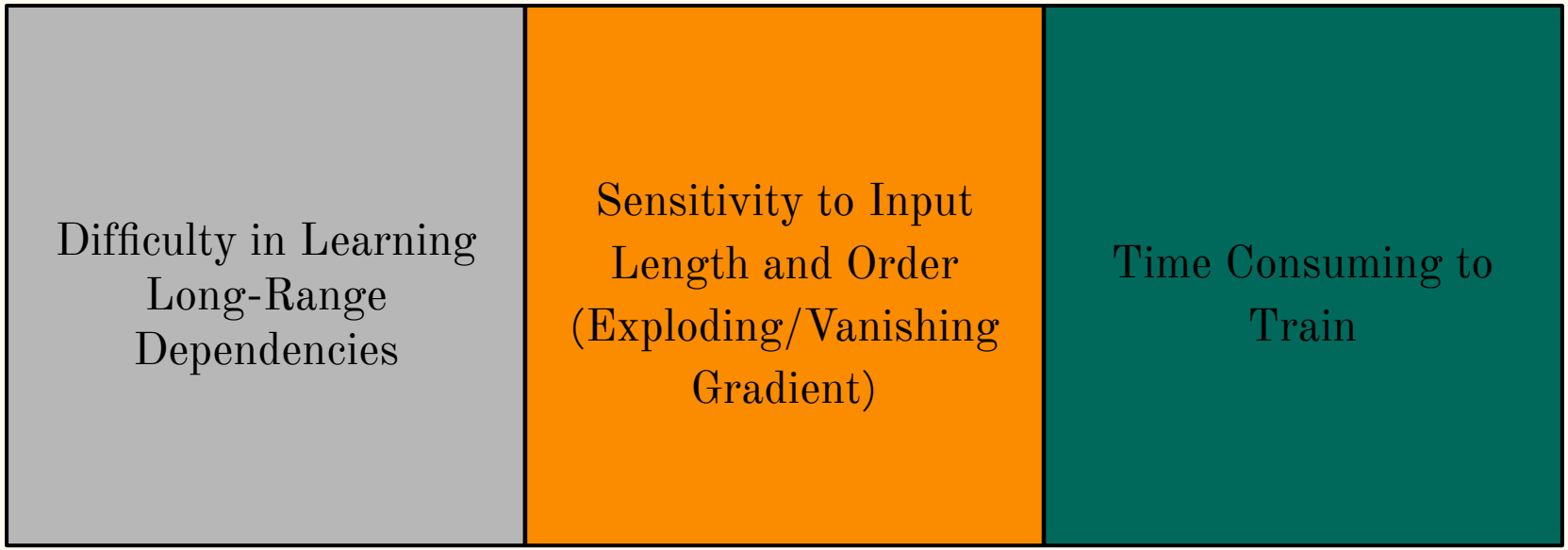


Adding More Inputs





Challenges of RNNs



Exploding Gradient Problem

- Too many inputs
- $W > 1$
- Output increases exponentially as number of sequences increase
- Memory output approaches an extremely large number after many time steps, which makes updating the weights very difficult or even impossible with back propagation

$$\text{Input}_1 \times 2 \times 2 \times 2 \times 2$$

$$= \text{Input}_1 \times 2^4 = \text{Input}_1 \times 16$$

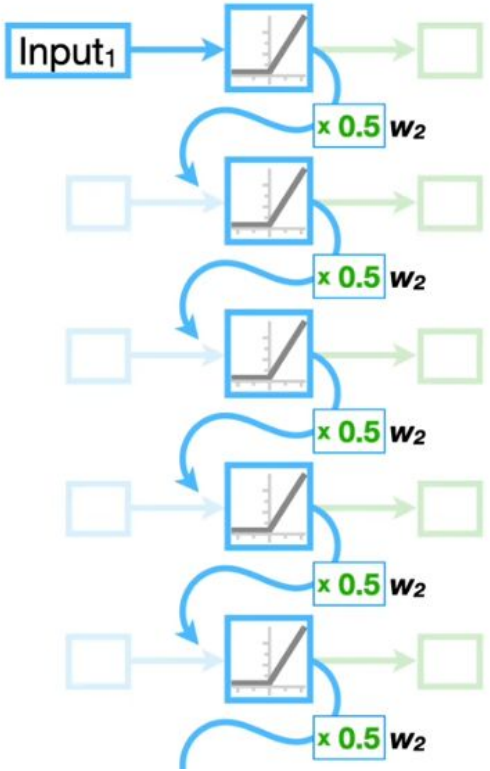
$$= \text{Input}_1 \times w_2^{\text{Num. Unroll}}$$

Vanishing Gradient Problem

- Too many inputs
- $W < 1$
- Output decreases exponentially as number of sequences increase
- Memory output approaches 0 after many time steps which makes it irrelevant to the final output, so past data loses influence on future prediction

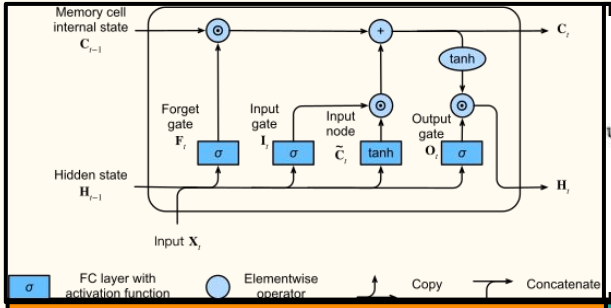
$$\text{Input}_1 \times 0.5^{50}$$

$$\text{Input}_1 \times w_2^{\text{Num. Unroll}}$$

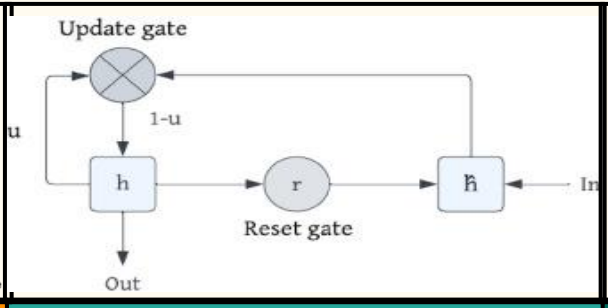


SOLUTION TO THE ISSUES

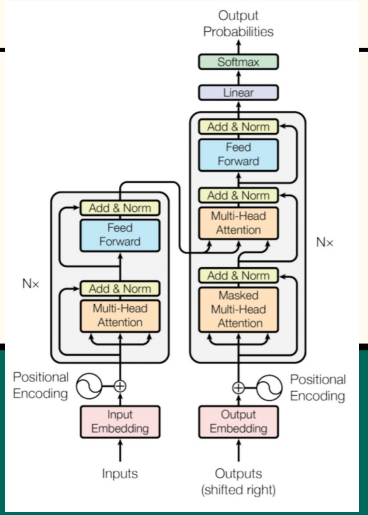
Advanced RNN Variants



Long Short-term Memory Networks



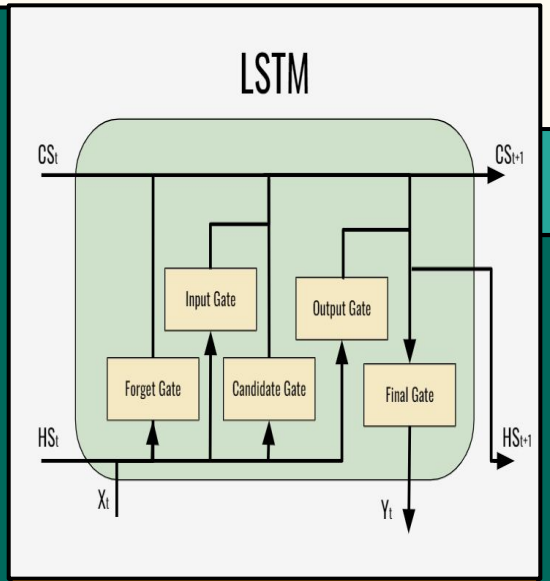
Gated Recurrent Units



Transformers

Long Short Term Memory(LSTM) / Gated Recurrent Unit(GRU)

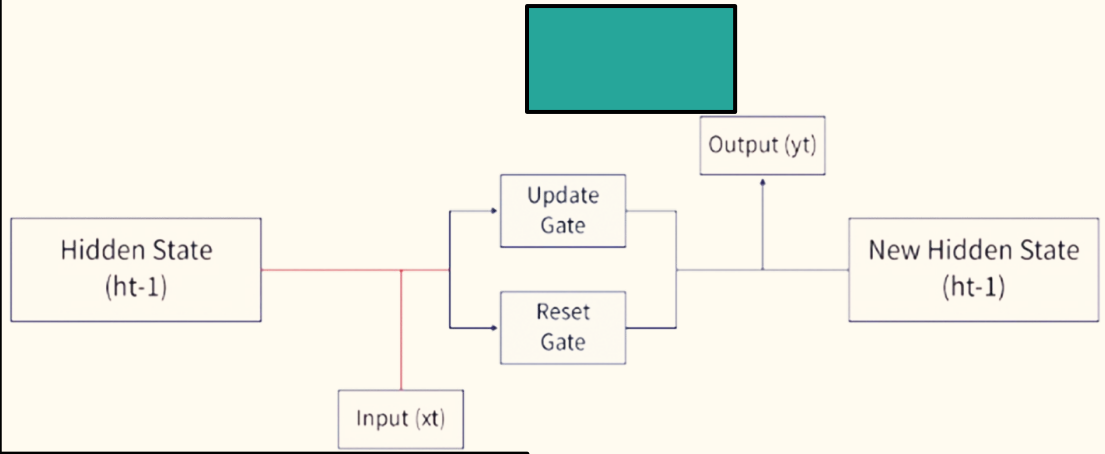
Allows the Neural Network to selectively keep and discard information using gates to control memory flow. This selective memory allows the Neural Network to reduce exploding gradients and prioritize vanishing gradients. This also allows for better understanding of long range dependencies.



For LSTM, the gates are an input gate to control what information is added, a forget gate to control what information is deleted, and an output gate to control what information is moved forward.

Gated Recurrent Unit(GRU)

For GRU, the gates are a reset gate to control what information is added and deleted, and an update gate to control what information is moved forward. In other words



Transformers(BERT)

- Processes inputs simultaneously
- Uses a self-attention node
- Captures relationships with other inputs
- Keeps embedded position so order is still maintained

- Faster than previous RNN variants
- Views the entire set of inputs at once instead of sequentially

Comparisons to Other Machine Learning Methods

Convolution Neural Networks

Similarities:

- Use hidden layers to create a black box for computation

Differences:

- Works better for images because of the use of convolutions (a filter used in convolutional networks)
- Worse at sequential information

Reinforcement Learning

Similarities:

- Retains information about past observations

Differences:

- Memory works by modifying weights in response to the environment
-

Q&A