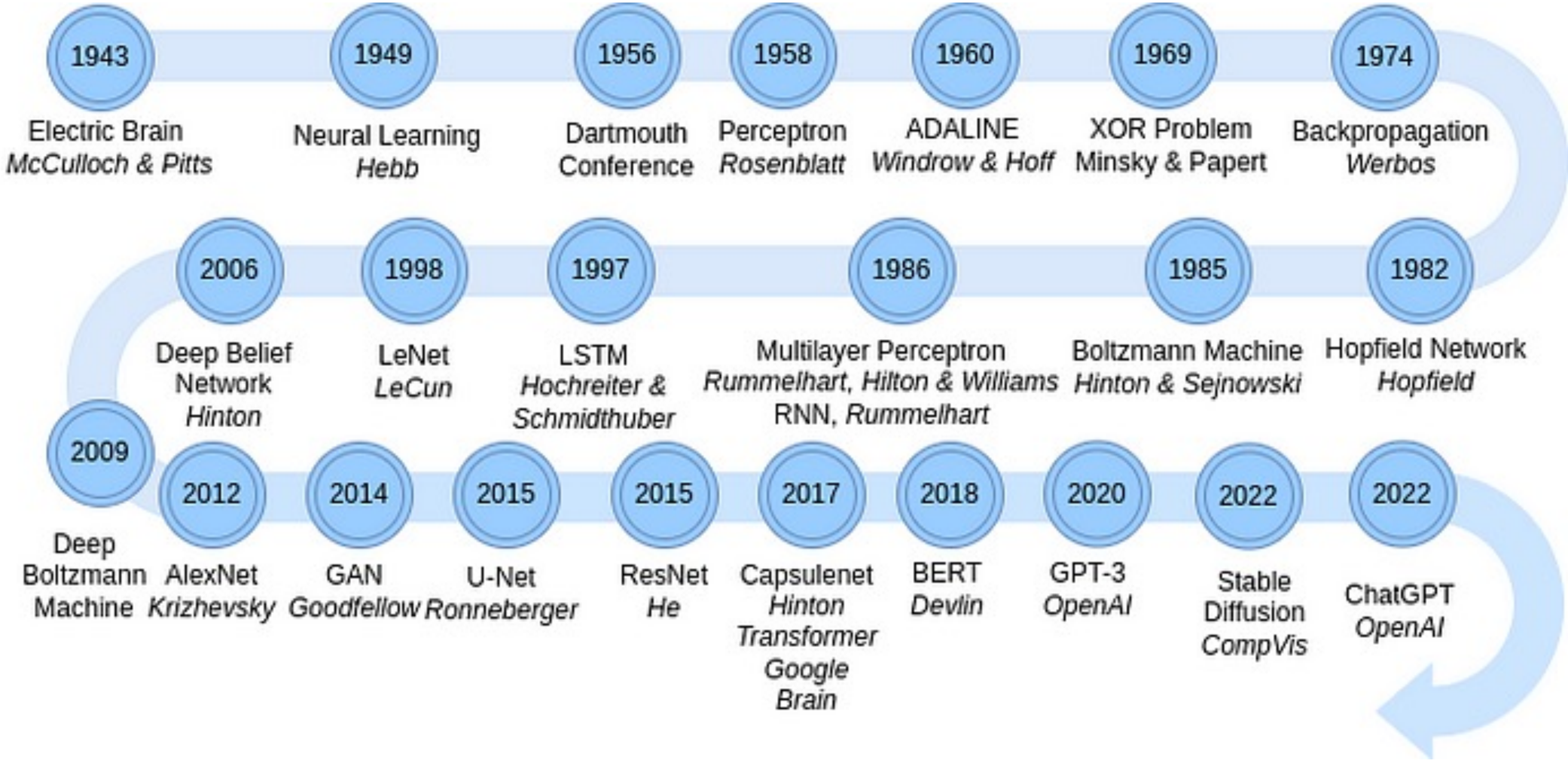
Deep Learning
Recurrent Neural Networks

CS 334: Machine Learning

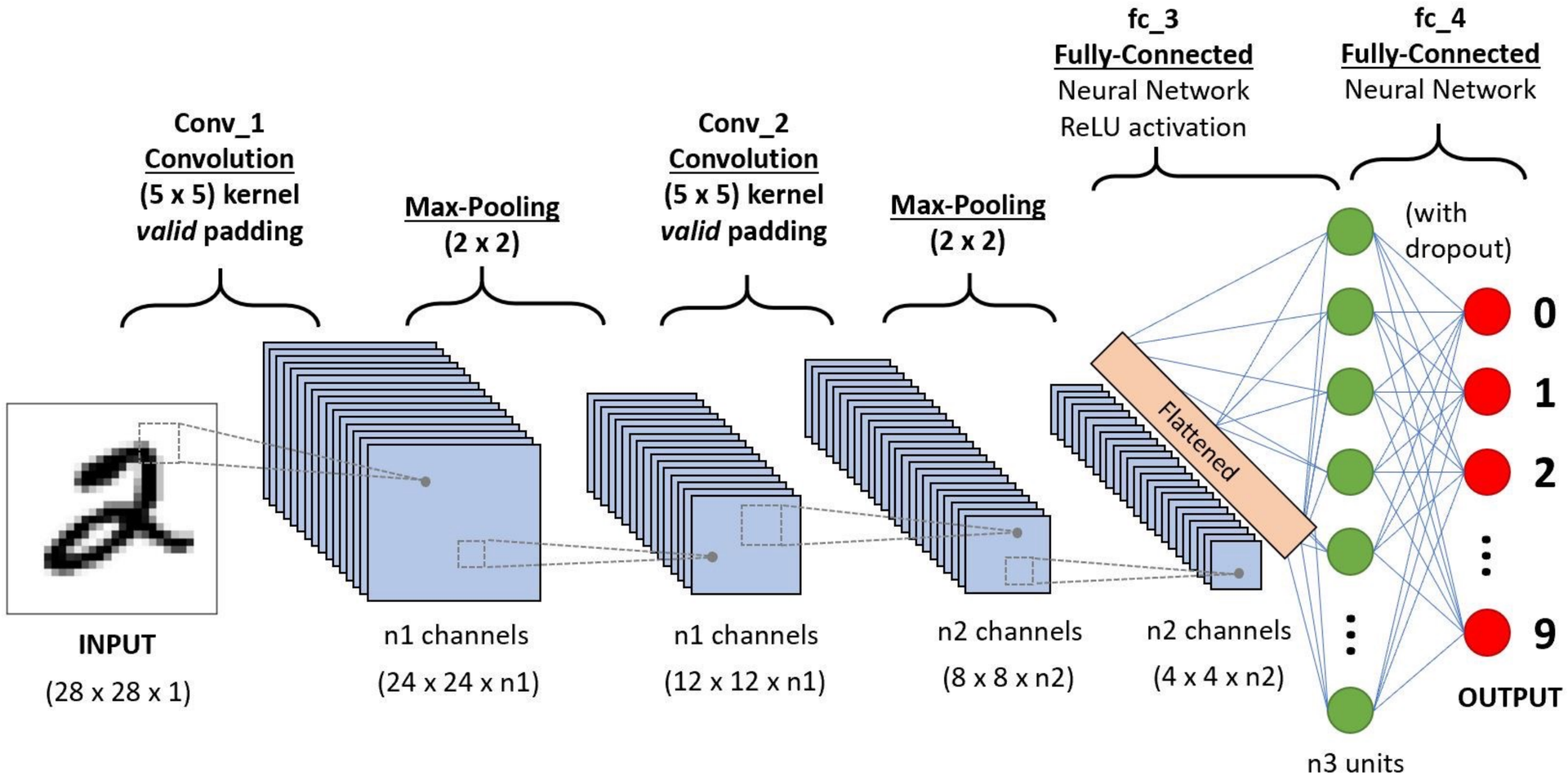
Slides adapted from Jinho Choi, Stuart Russell, Josh Tobin, Andrej Karpathy, John Buillinaria, and Kyunghyun Cho

Deep learning: a brief history



https://pub.towardsai.net/a-brief-history-of-neural-nets-472107bc2c9c

Review: CNN



https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-understand-the-basics/

Should we treat them the same?

News Cryptocurrency News Today June 12 DATE Bitcoin GPE Dogecoin Shiba Inu PERSON and other top coins prices and all latest updates cryptocurrency Latest News ORG Today June 12 DATE Bitcoin GPE and all major top cryptocurrencies were trading in red at 345 pm TIME on Saturday June 12 DATE In line with its recent trends overall global crypto market was down by over 15 per cent on the weekend DATE View in App GPE Bitcoin GPE was down by 6 CARDINAL and was trading at Rs 2728815 DATE after hitting days high of Rs 2900208 Source Reuters ORG Reported By ZeeBiz NORP WebTeam Written By Ravi Kant Kumar PERSON Updated Sat Jun PERSON 12 20210646 pm TIME Patna ORG ZeeBiz WebDesk PERSON RELATED NEWS Cryptocurrency Latest News Today June 14 DATE Bitcoin GPE leads crypto rally up over 12 CARDINAL after ELON MUSK TWEET Check Ethereum Polka ORG Dot Dogecoin Shiba Inu PERSON and other top coins INR ORG price World India ORG updates Bitcoin GPE law is only latest headturner by El Salvadors MILLENNIAL ORG PRESIDENT Chinas cryptocurrency mining crackdown spreads to Yunnan GPE in southwest media Cryptocurrency latest news ALERT Rs

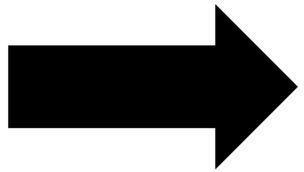
Motivation: Text

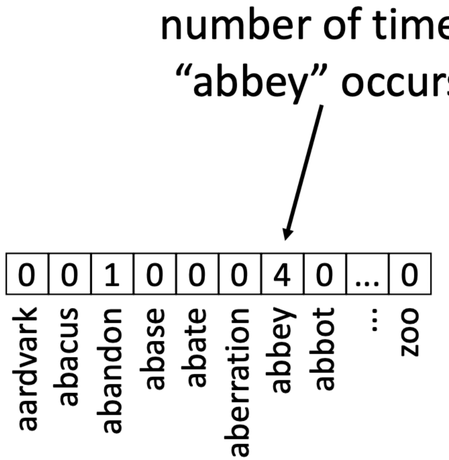
- CNN works well with images
  - Images are numbers (greyscale, RGB)
- What about text?
  - Issue #1: How to convert words into something numerical?
  - Issue #2: How to deal with sequences of words that vary in length?

Take #1: Tokenization

- Tokens: One hot encoding of words (smartly dealing w/ case, punctuation, etc)
- Bag-of-Words: Treat each document as an unordered set of words

document x





Limitations of Tokenization

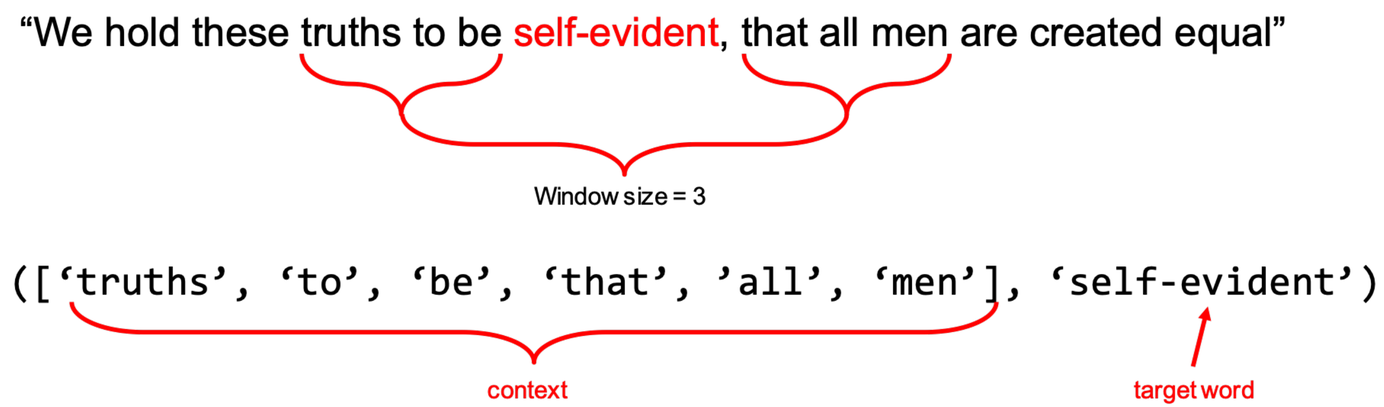
- Loses lots of information: Part of speech, synonym (distinct words w/ similar meanings), polysemy (single word w/ multiple meanings), general context of the word
  - “Took money out of the *bank*”
  - “Got stuck on the river *bank*”
- Increasing vocabulary size is difficult
- Vector length is huge yet sparse information

Take #2: Word Vectors

- Goal: Represent a word by an m-dimensional vector
- “Similar” words are represented by “nearby” vectors in m-dimensional space
- Words in particular domain (economics, healthcare, etc.) would be closer
- Helps w/ synonym and polysemy

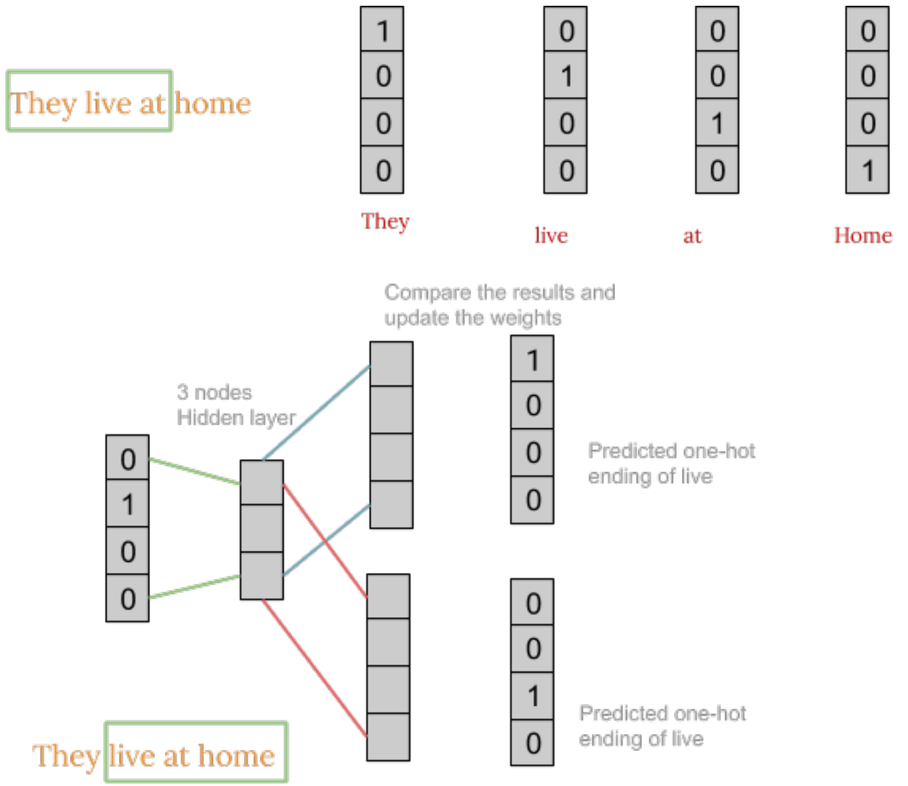
Intuition: Similar Words

- Idea: Similar words occur in similar contexts
- For a given word, look at the “window” around it
- Consider trying to predict a word given the context

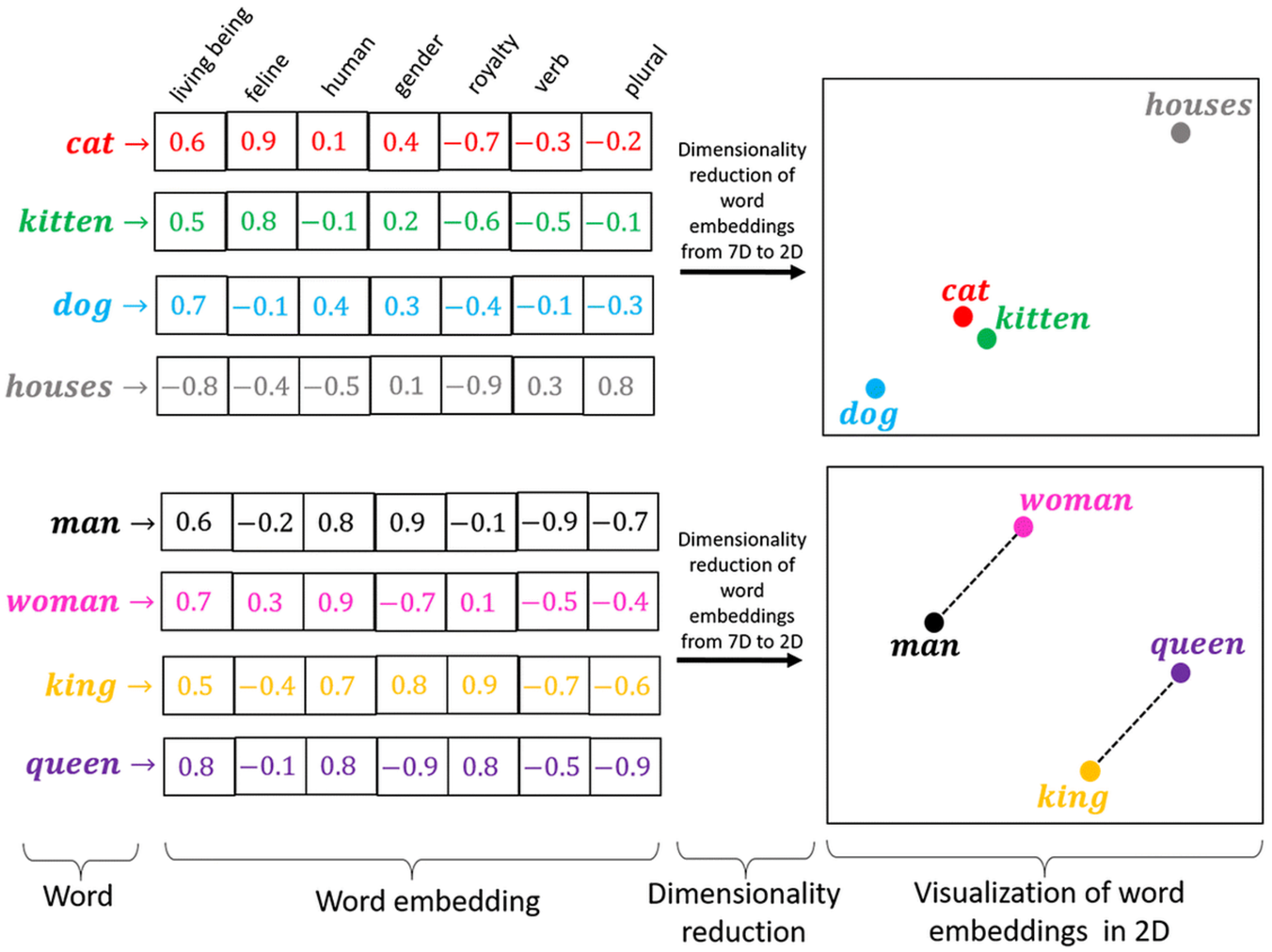


Popular Word Embeddings

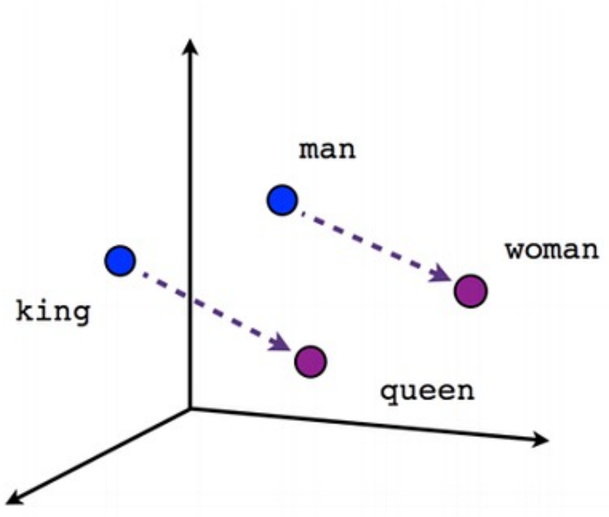
- (Word2Vec) *Distributed representations of Words and Phrases and Their Compositionality* by Mikolov et al.
- *GloVe: Global Vectors for Word Representation* by Pennington et al.



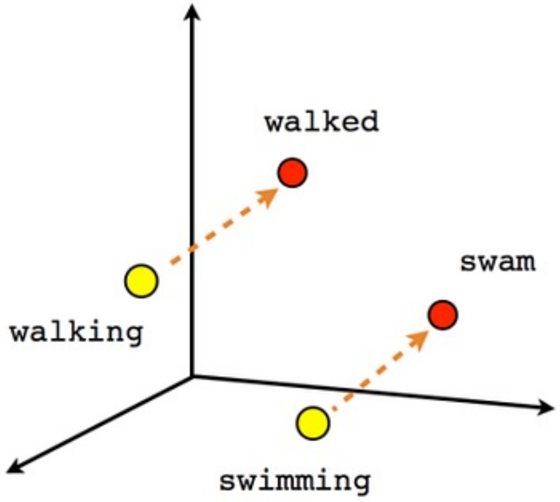
Word Embeddings



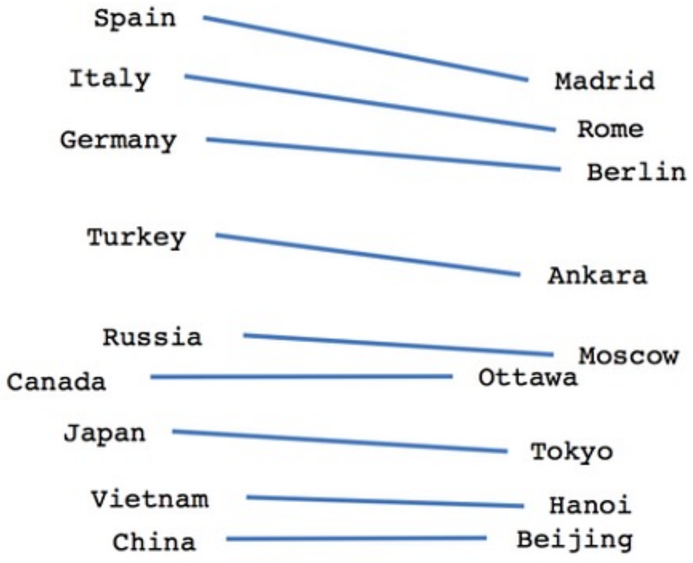
Word Embeddings: Properties



Male-Female



Verb tense



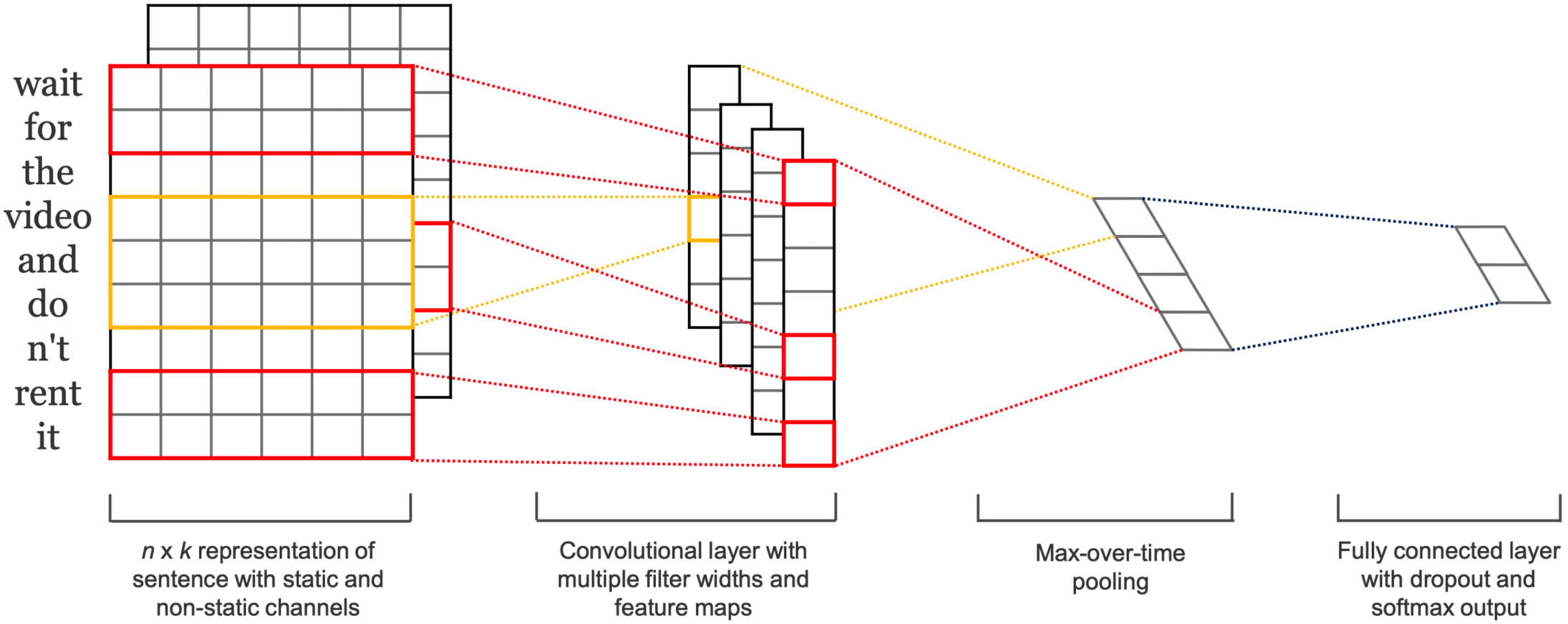
Country-Capital

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

Downside to Word Embeddings

- Loses information about “context”
  - “He ate a tasty apple”
  - “She wrote her essay on an Apple laptop”
- Words should be handled differently depending on the “context”
- Each word should update the “context”

CNN for Text



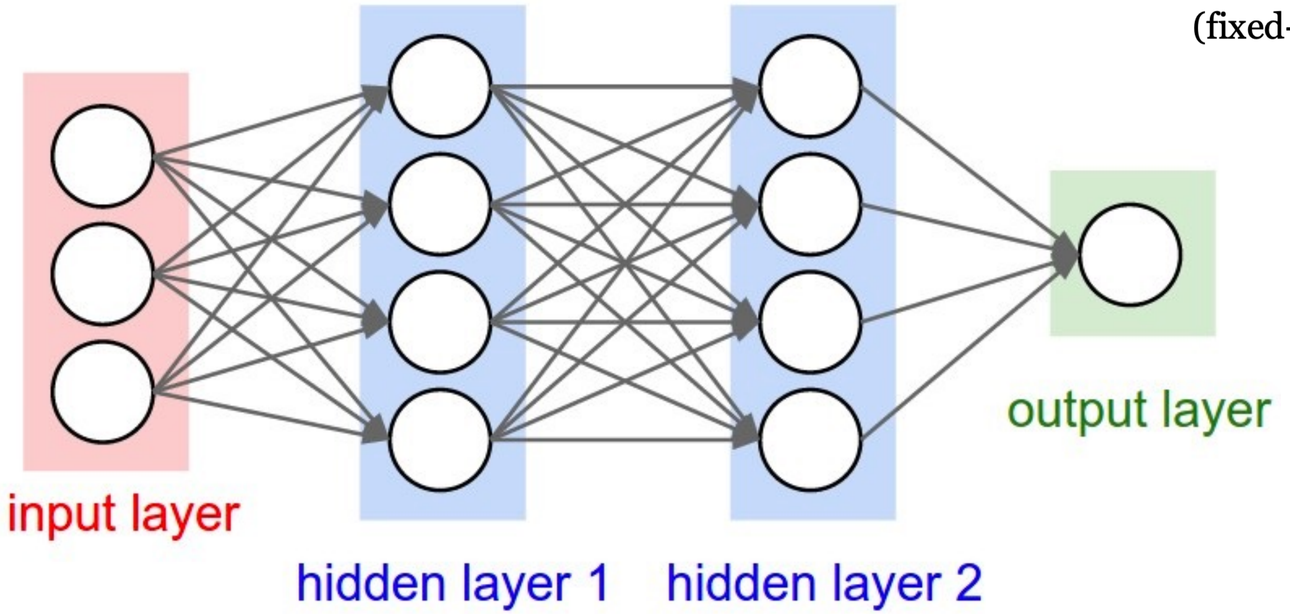
Word embeddings **Window of k words** Source: Y. Kim.(2014). Convolutional Neural Networks for Sentence Classification

Need for Variable Length

Sentence: The dogs are <u>barking.</u>

$$\mathbf{x} = [\mathbf{e}_{\text{the}}, \mathbf{e}_{\text{dogs}}, \mathbf{e}_{\text{are}}] \in \mathbb{R}^{3d}$$

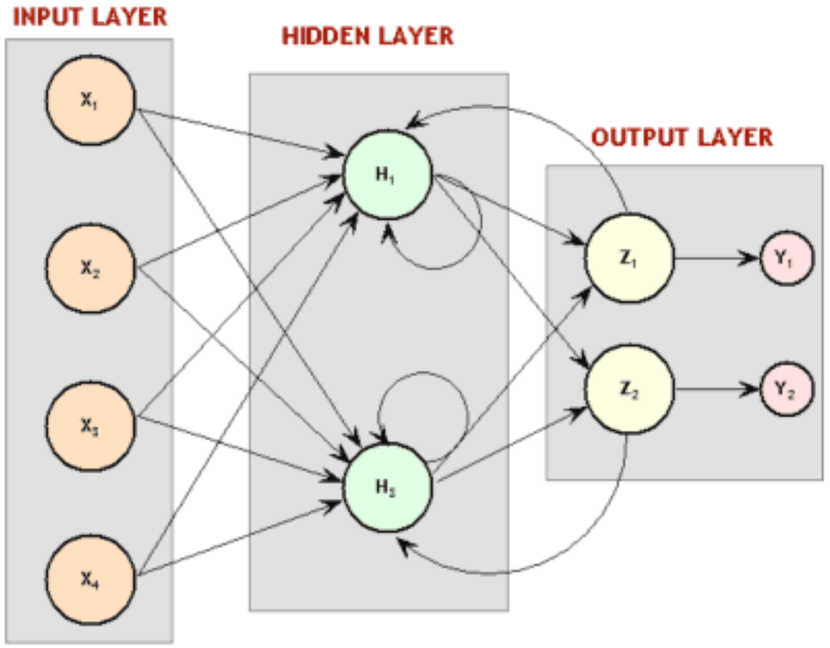
(fixed-window size = 3)



Input: The dogs in the neighborhood are _____

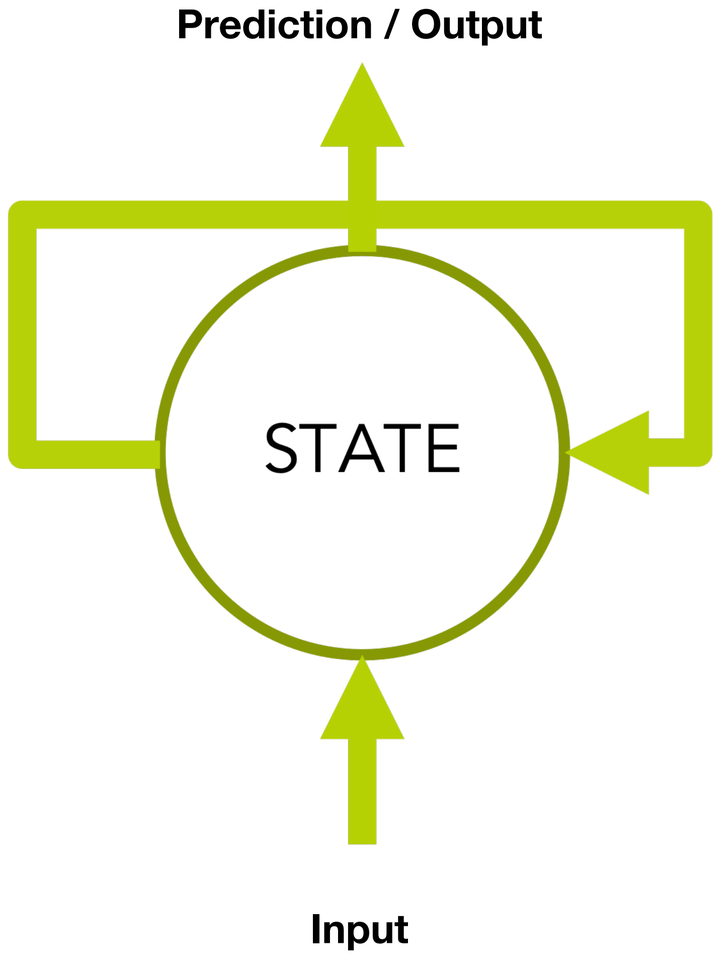
Recurrent Neural Networks (RNN)

- Family of neural networks for processing sequential data
- Output of the layer can connect back to the neuron itself or a layer before it
- Share same weights across several time steps
- Proposed by Rumelhart in 1986



Intuition: “Recurrence”

- Input words one by one
- Network outputs two things
  - Prediction: What would be the prediction if the sequence ended with that word
  - State: Summary of everything that happened in the past
- Response to a word depends on words that preceded it



RNN: Recurrence

We can process a sequence of vectors **x** by applying a recurrence formula at every time step:

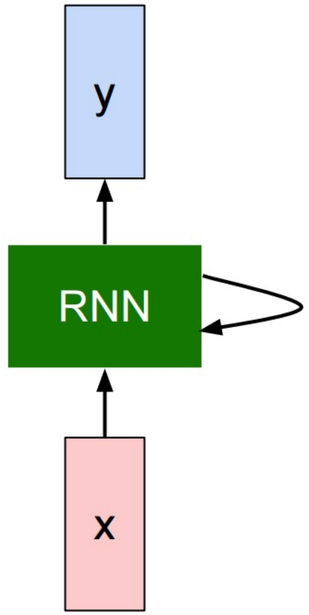
$$h_t = f_W(h_{t-1}, x_t)$$

new state

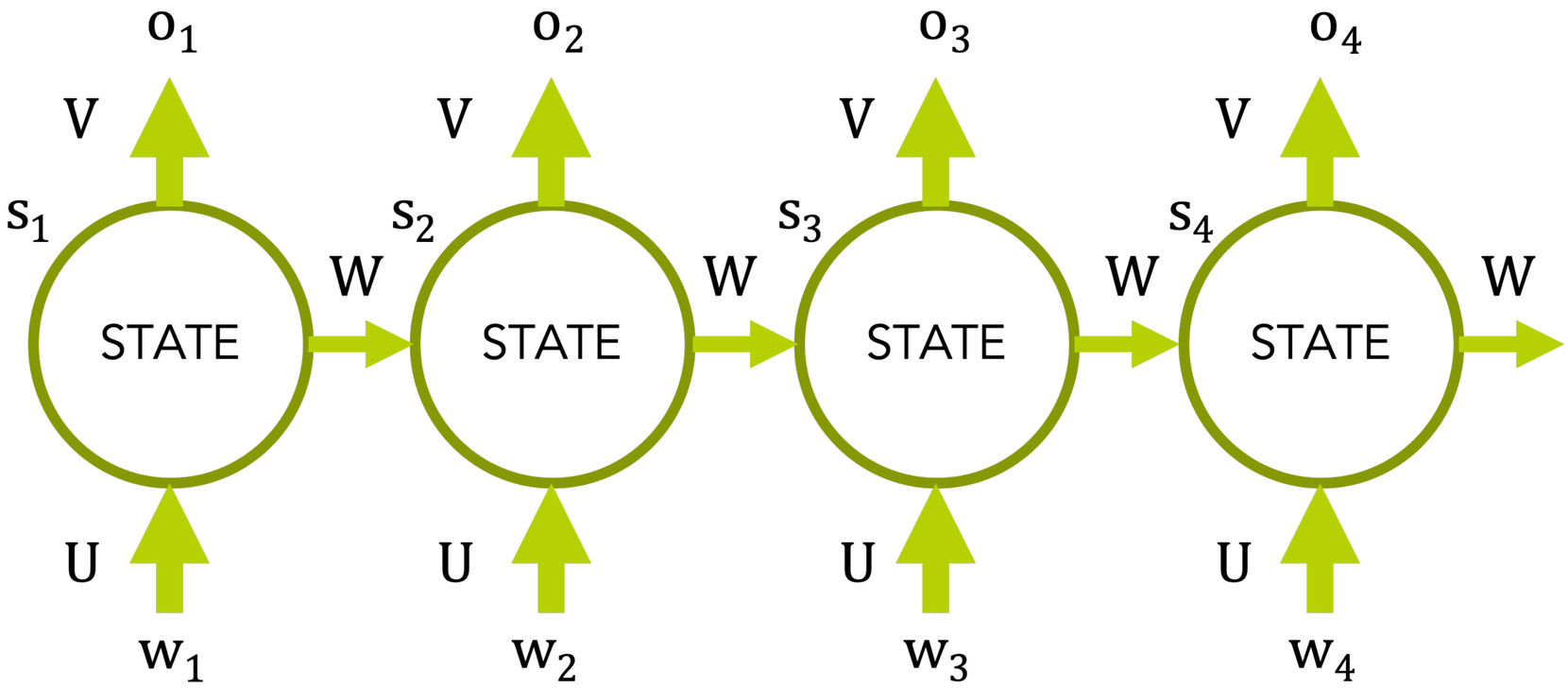
some function with parameters W

old state

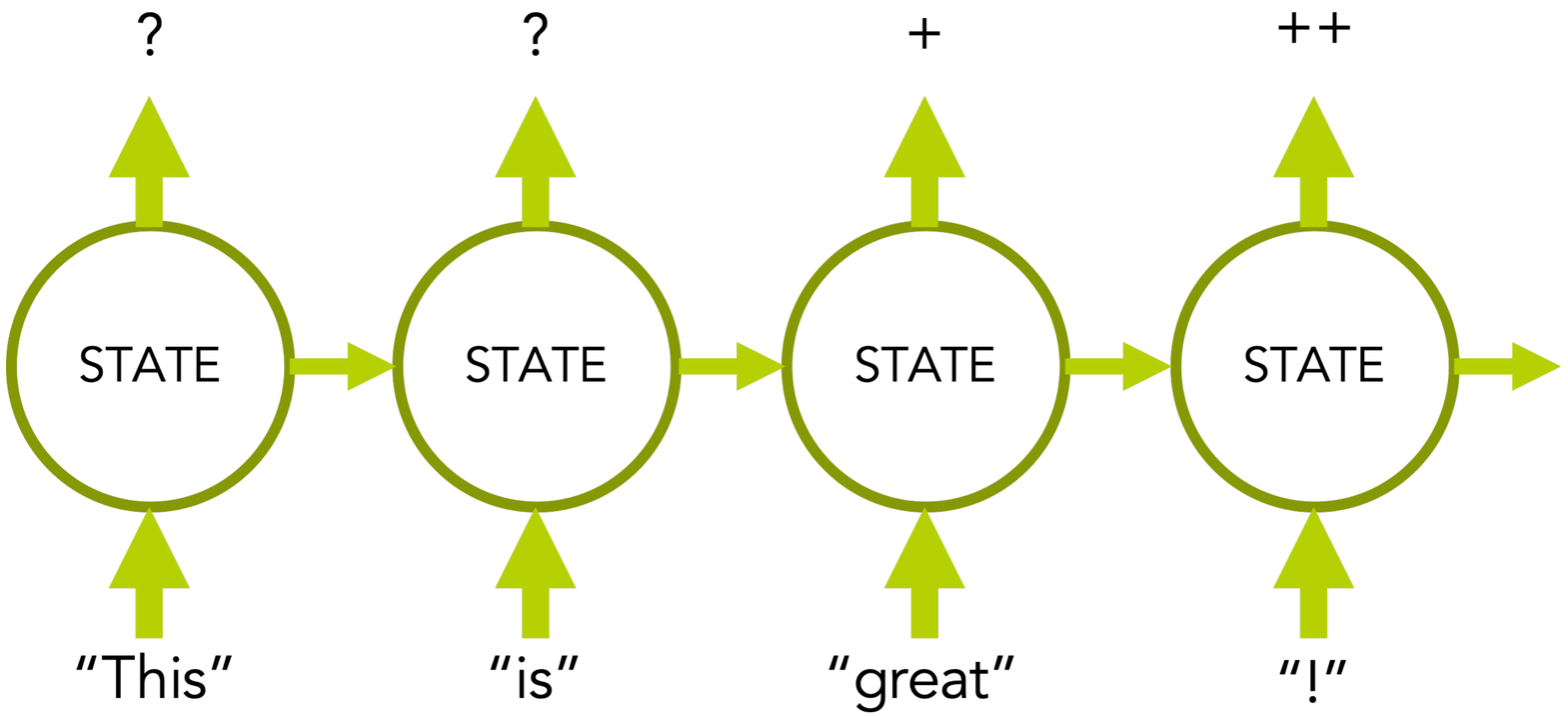
input vector at some time step



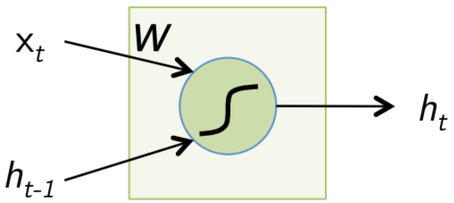
“Unrolling” the RNN



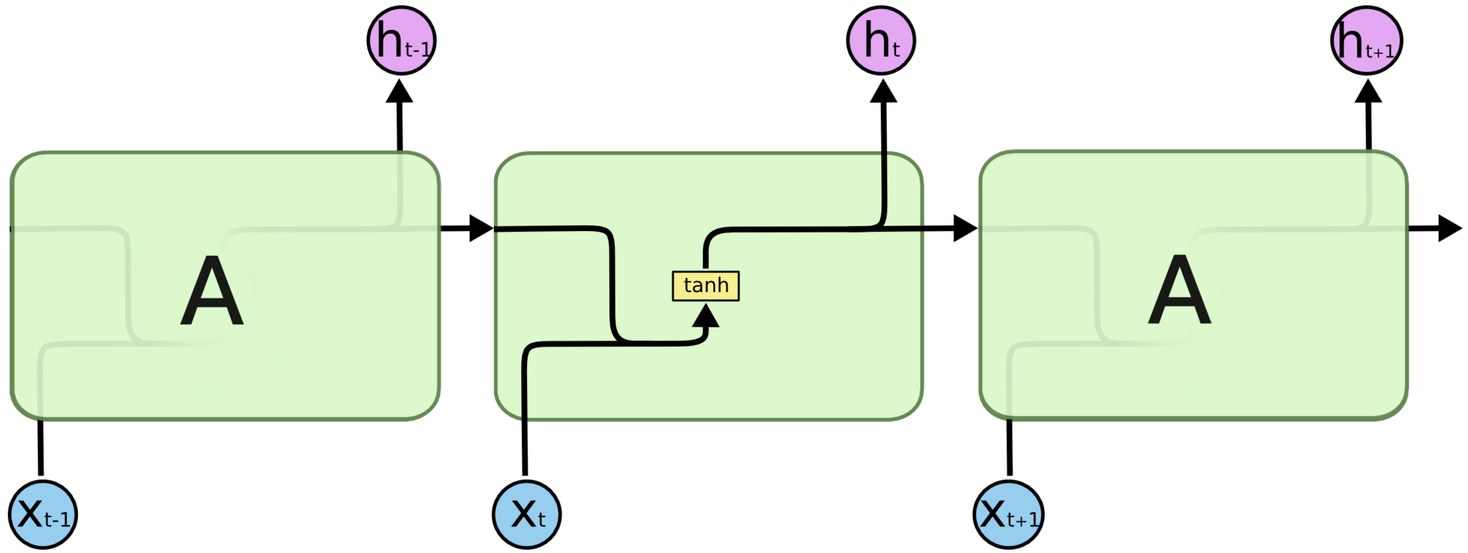
Example: “Unrolling” the RNN



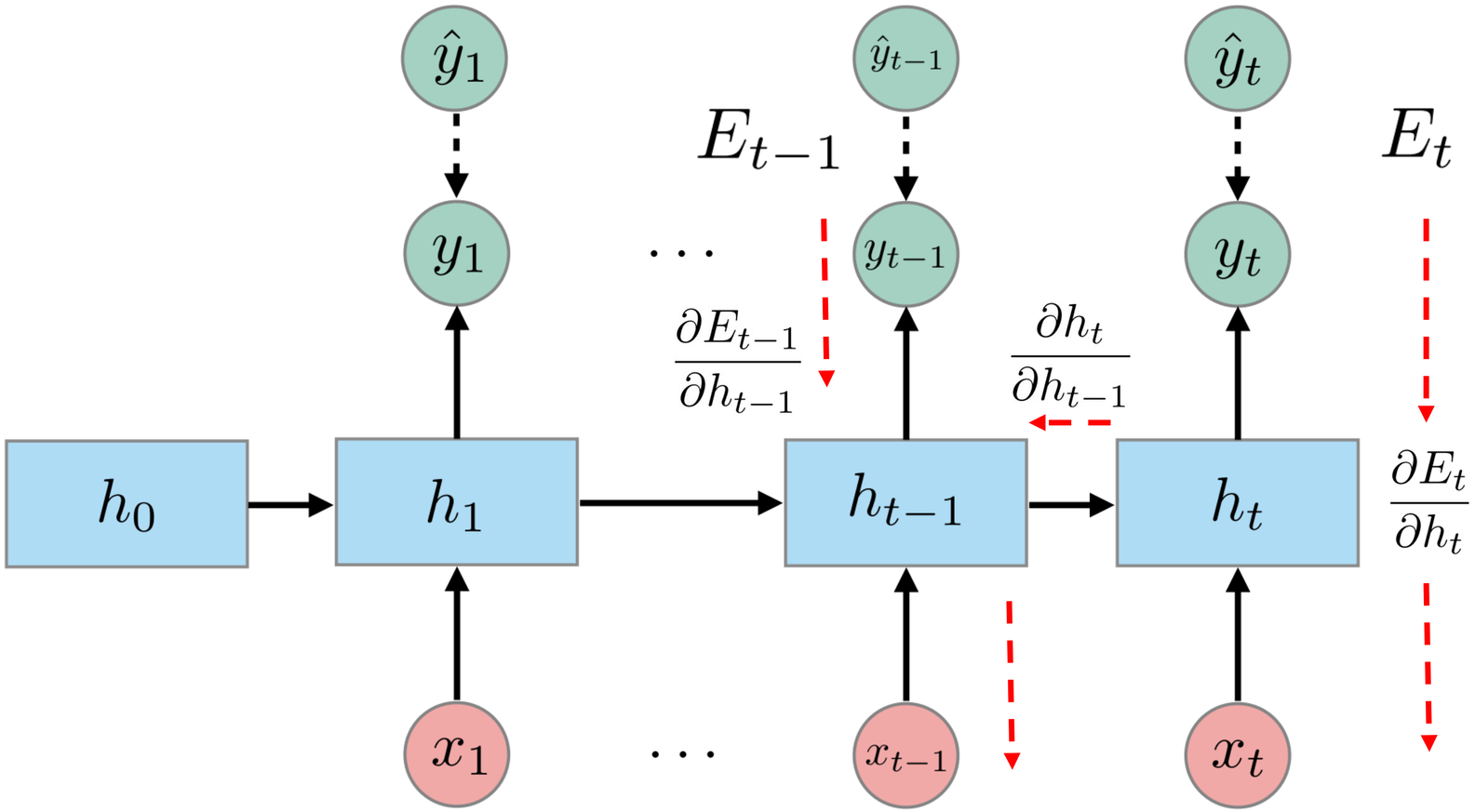
Vanilla RNN



$$h_t = \tanh W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix}$$

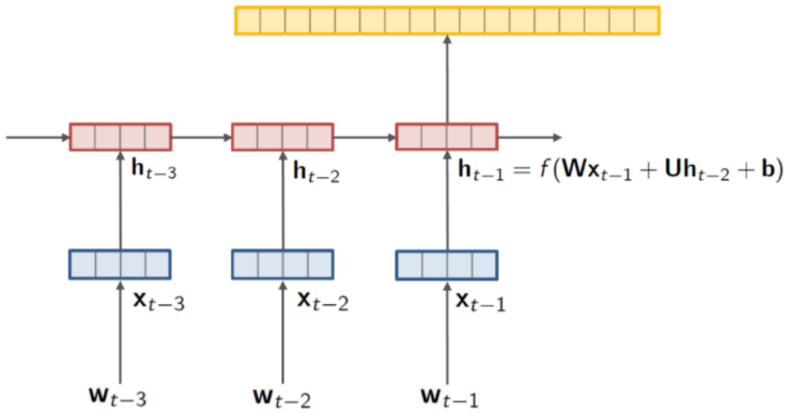


Backpropogation through time

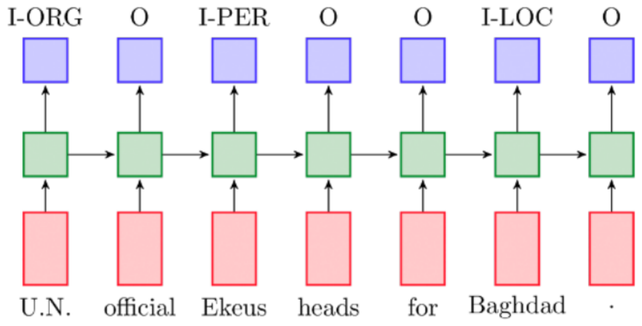


RNN Applications

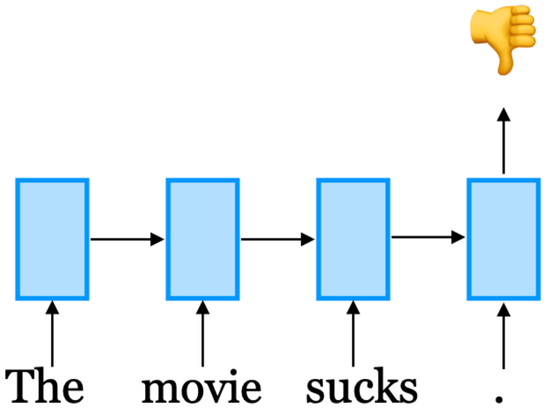
Language modeling



Sequence tagging



Text classification

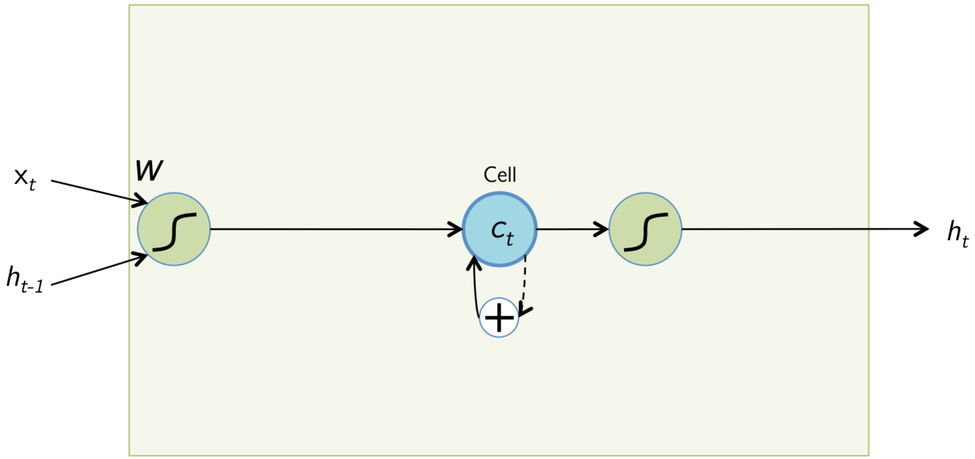


Long-Term Dependency Problems

- Appeal of RNN is to connect previous information to present task
- Gap between relevant information and point of needing it can be large
  - Example: I grew up in France … I speak fluent ___
- Long-range dependencies are difficult to learn because of vanishing gradient or exploding gradient problem (depending on the activation function)

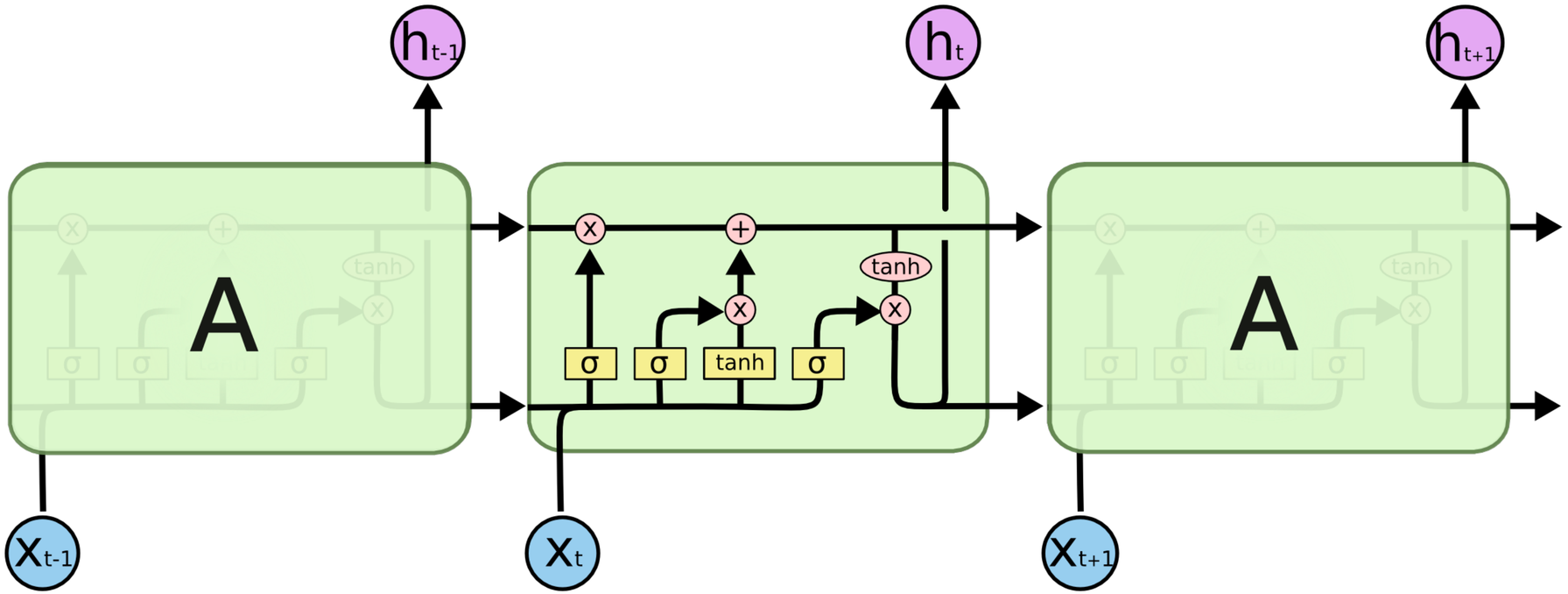
Long Short-Term Memory Units (LSTM)

- Invented by Hochreiter and Schmidhuber in 1997
- Intuition: Make “remembering” easy
- Introduce a more complicated update mechanism for changing of the internal state
- Store long-term information and requires active choice to forget this information



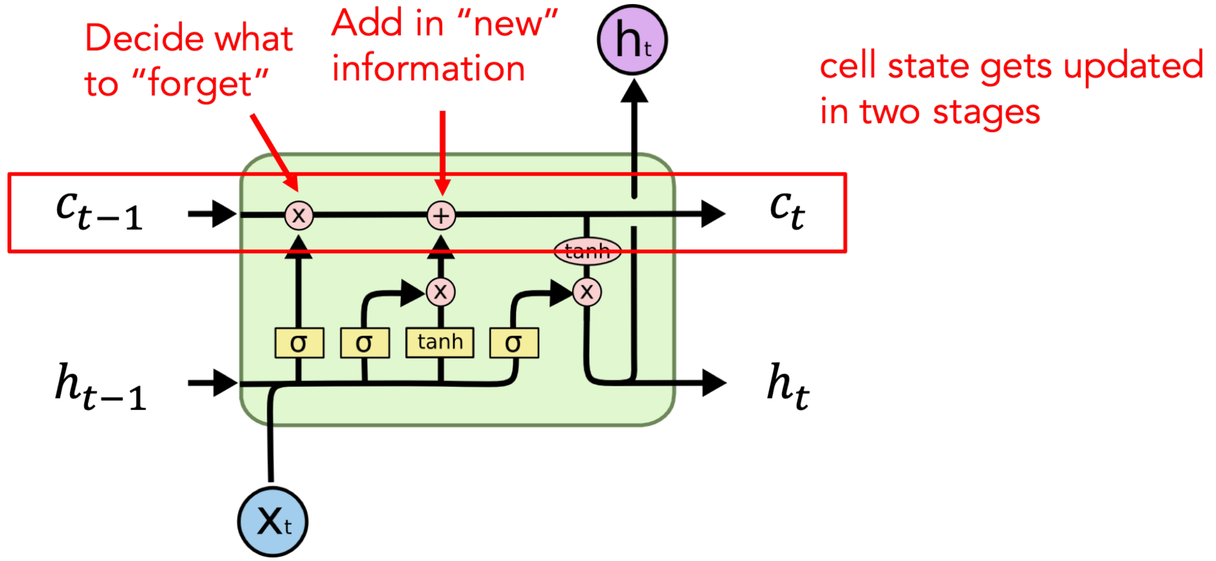
$$c_t = c_{t-1} + \tanh W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} \qquad h_t = \tanh c_t$$

LSTM



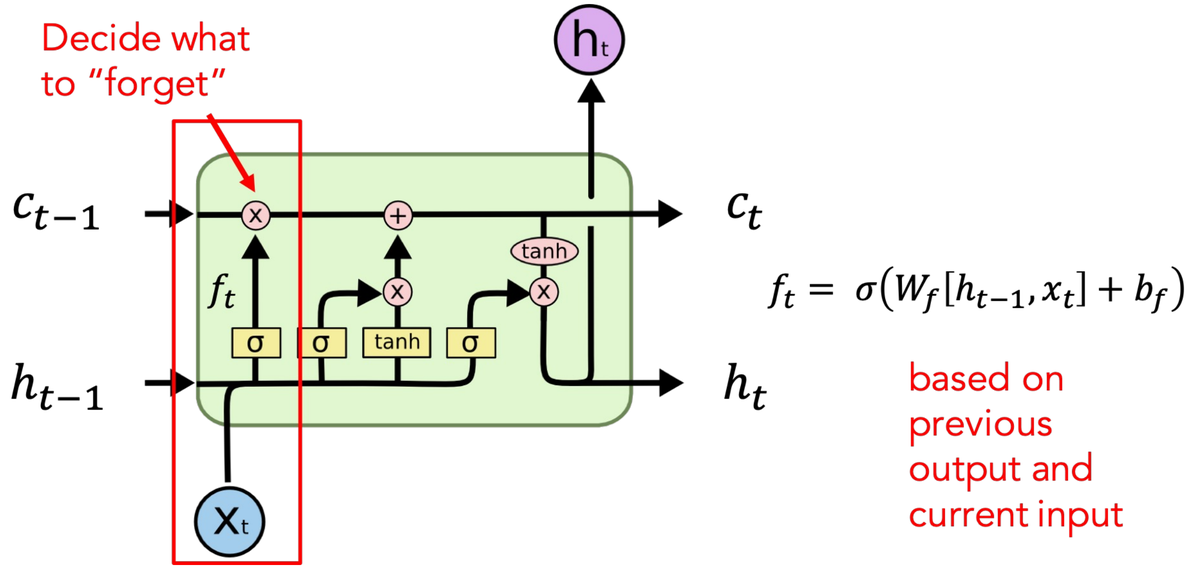
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM: Cell State



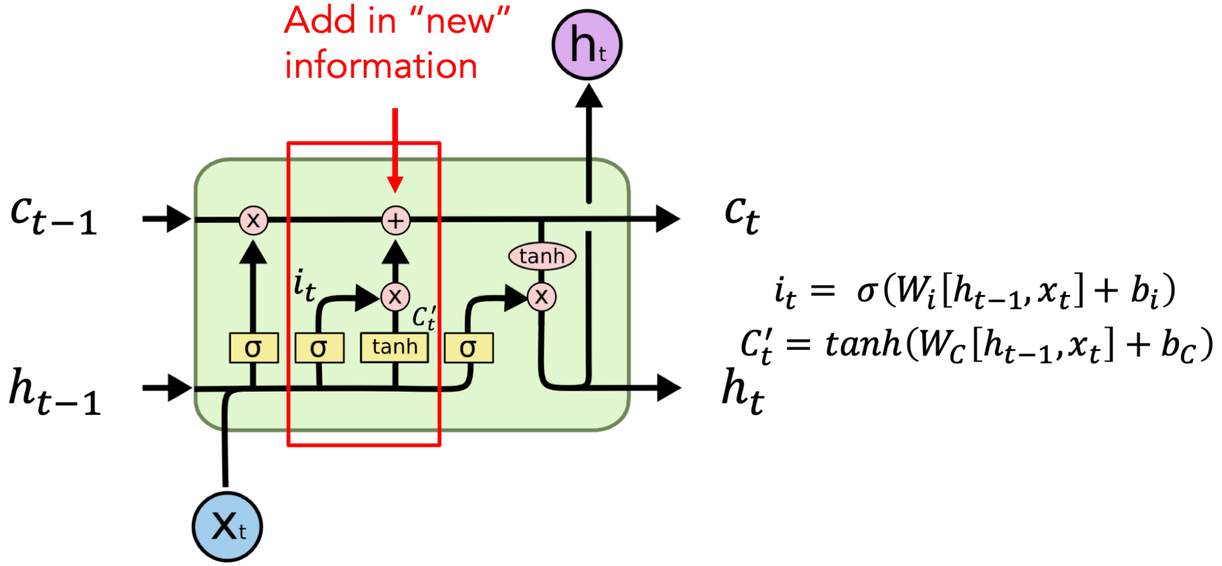
- Key idea: cell state (memory cell) and runs through the entire chain
- Easy for information to just flow along unchanged
- Add/remove information via gates

LSTM: Forget Gate Layer



- Looks at the current value x_t and the previous state (h_{t-1}) and outputs a number between 0 and 1 for each number in cell state
  - 1 = completely keep, 0 = completely forget

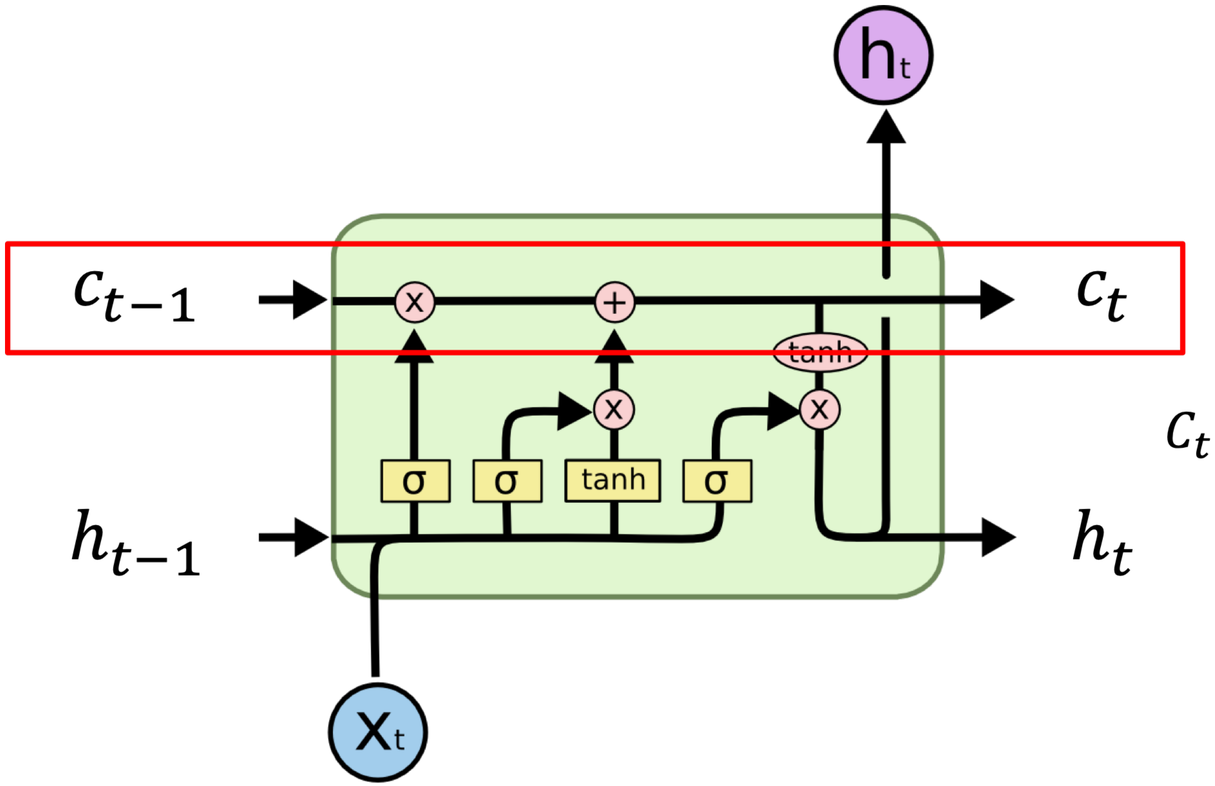
LSTM: Input Layer



- Input gate layer (sigmoid layer) decides which values to update (how much to write)

- Tanh layer creates a new vector of candidates to be added to the state

LSTM: Update Layer



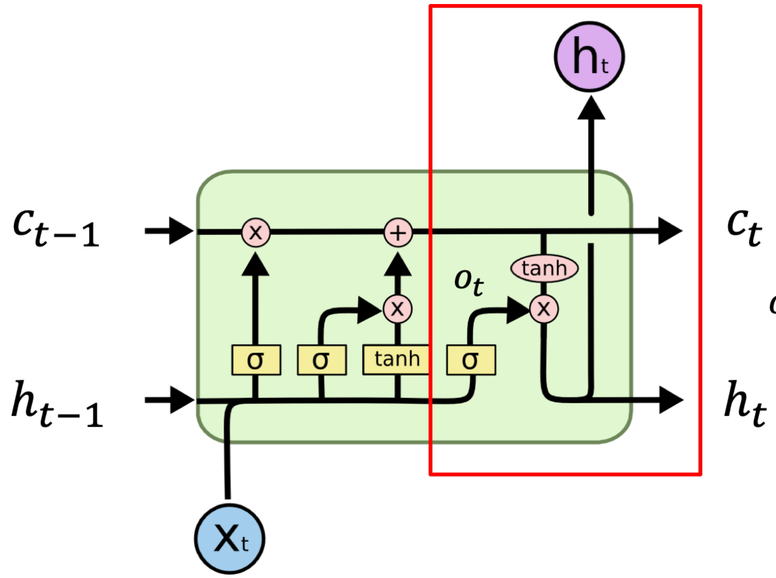
Note: '$*$' represents element-wise multiplication

$$C_t = f_i * C_{t-1} + i_t * C'_t$$

forget the old (or not)

add the new (or not)

LSTM: Output Layer



Final stage computes the output

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

- Output based on cell state (filtered version)
- Sigmoid layer determines which parts of cell states to output
- Tanh pushes values between -1 and 1

Experiment: Shakespearean Writing

- Download all works of Shakespeare into single file
- Train 3-layer RNN with 512 hidden nodes on each layer
- Create samples for both speaker's names and the contents

```
VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.
```


http://karpathy.github.io/2015/05/21/rnn-effectiveness/

RNN Notes

- Most common version of LSTM but there are many different “flavors”
  - Gated Recurrent Unit (GRU): slightly simpler
  - Depth-Gated RNN
- LSTMs have considerably more parameters than plain RNNS
- Most of the previous big performance improvements in NLP came from LSTMs, not plain RNN

PyTorch: RNN

```
import torch.nn as nn

class RNN(nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        super(RNN, self).__init__()

        self.hidden_size = hidden_size

        self.i2h = nn.Linear(input_size + hidden_size, hidden_size)
        self.h2o = nn.Linear(hidden_size, output_size)
        self.softmax = nn.LogSoftmax(dim=1)

    def forward(self, input, hidden):
        combined = torch.cat((input, hidden), 1)
        hidden = self.i2h(combined)
        output = self.h2o(hidden)
        output = self.softmax(output)
        return output, hidden

    def initHidden(self):
        return torch.zeros(1, self.hidden_size)

n_hidden = 128
rnn = RNN(n_letters, n_hidden, n_categories)
```

https://pytorch.org/tutorials/intermediate/char_rnn_classification_tutorial.html

New Winter Dawns

Failure of backpropogation and ascent of SVMs, random forests led to a slump in the early 2000s

Deep Learning: The Dark Ages

- Hinton & Bengio hatched plan to “rebrand” neural networks with deep learning
- Resurgence with “A fast learning algorithm for deep belief nets” [Hinton et al., 2006]
  - Clever way to initialize neural networks rather than randomly
- Followed by “Greedy layer-wise training of deep networks” [Bengio et al., 2007]

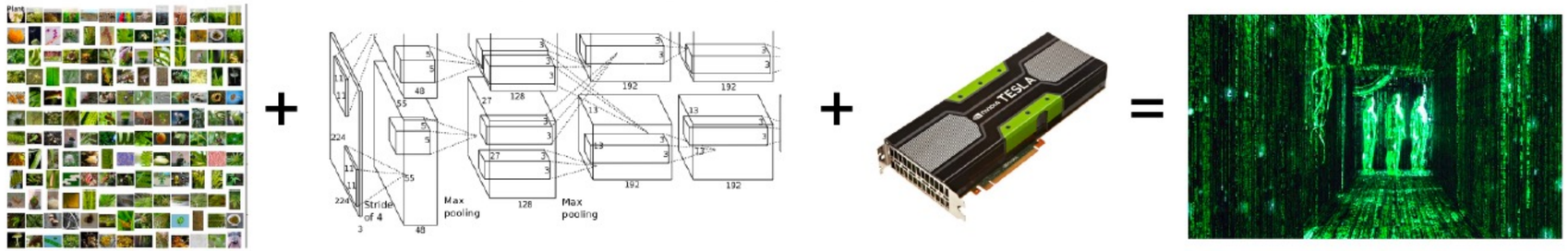
Deep Learning Rises Again

- Labeled datasets were thousands of times too small
  - Unsupervised pre-training could help mitigate bad initialization
- Computers were millions of times too slow
- Weights were initialized in a stupid way
- Used wrong type of non-linearity

http://www.andreykurenkov.com/writing/a-brief-history-of-neural-nets-and-deep-learning-part-4/

Deep Learning Rises Again

The Deep Learning "Computer Vision Recipe"



Big Data: ImageNet | Deep Convolutional Neural Network | Backprop on GPU | Learned Weights

Deep learning = lots of training data + parallel computation + scalable, smart algorithms

http://www.andreykurenkov.com/writing/a-brief-history-of-neural-nets-and-deep-learning-part-4/

nature International weekly journal of science

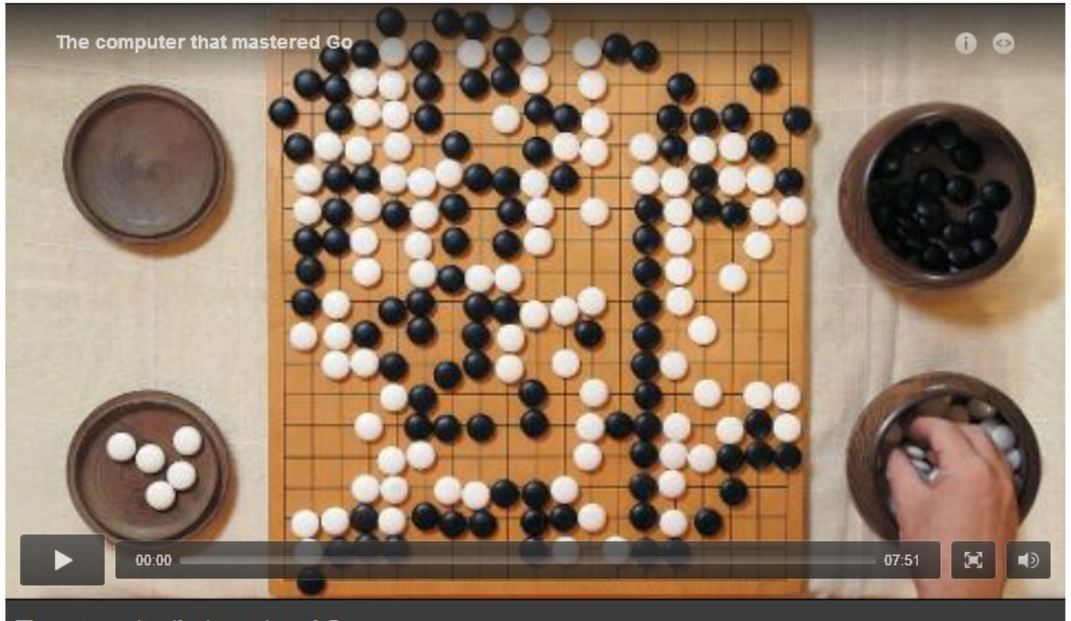
NATURE | NEWS

Google AI algorithm masters ancient game of Go

Deep-learning software defeats human professional for first time.

Elizabeth Gibney

27 January 2016



The computer that mastered Go

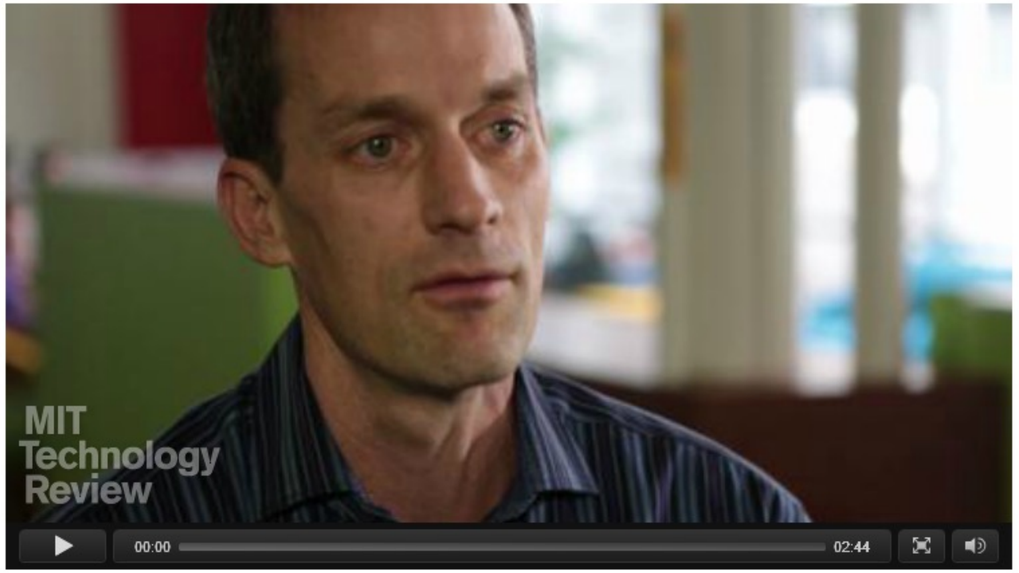
Nature Video

Deep Learning: MIT Technology Review - 10 Breakthrough Technologies 2013

Deep Learning

With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.

When Ray Kurzweil met with Google CEO Larry Page last July, he wasn't looking for a job. A respected inventor who's become a machine-intelligence futurist, Kurzweil wanted to discuss his upcoming book *How to Create a Mind*. He told Page, who had read an early draft, that he wanted to start a company to develop his ideas about how to build a truly intelligent computer: one that could understand language and then make inferences and decisions on its own.



2018 ACM Turing Awards

Fathers of the Deep Learning Revolution Receive ACM A.M. Turing Award

Bengio, Hinton and LeCun Ushered in Major Breakthroughs in Artificial Intelligence

Deep Learning Resources

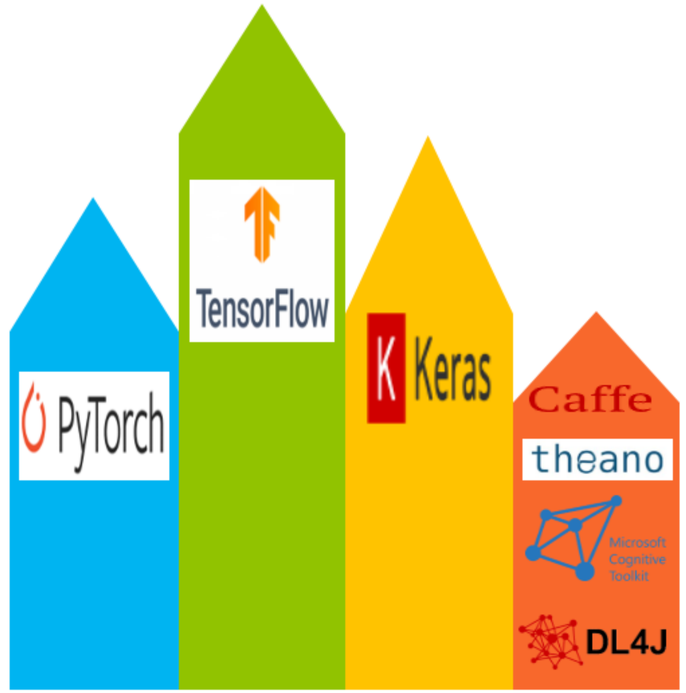
- Website with variety of resources and pointers at deeplearning.net
- Deep Learning Tutorial by Stanford (http://ufldl.stanford.edu/tutorial/)
- Neural Networks and Deep Learning online book (http://neuralnetworksanddeeplearning.com/)
- Deep Learning book by Goodfellow, Bengio, and Courville (http://www.deeplearningbook.org/)

Deep Learning Resources

- Andrej Karpathy's Blog on Neural Networks (http://karpathy.github.io/)
- Colah's Blog on Neural Networks (https://colah.github.io/)
- Lilian Weng's Blog (https://lilianweng.github.io/)

Deep Learning Toolkits

- TensorFlow (by Google)
- PyTorch (by MetaAI)
- Keras
- Caffe, theano, etc.



For a reasonable comparison of the frameworks, see
https://developer.ibm.com/articles/compare-deep-learning-frameworks/

Machine Learning

what society thinks I do

what my friends think I do

what my parents think I do

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^{l} \alpha_i$$

$$\alpha_i \geq 0, \forall i$$

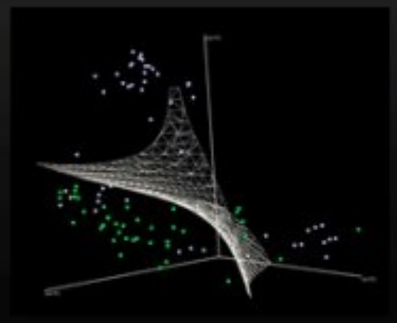
$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i, \sum_{i=1}^{l} \alpha_i y_i = 0$$

$$\nabla \hat{g}(\theta_t) = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(x_i, y_i; \theta_t) + \nabla r(\theta_t).$$

$$\theta_{t+1} = \theta_t - \eta_t \nabla \ell(x_{i(t)}, y_{i(t)}; \theta_t) - \eta_t \cdot \nabla r(\theta_t)$$

$$\mathbb{E}_{i(t)}[\ell(x_{i(t)}, y_{i(t)}; \theta_t)] = \frac{1}{n} \sum_i \ell(x_i, y_i; \theta_t).$$

what other programmers think I do



what I think I do

```
>>> from sklearn import svm
```

what I really do