

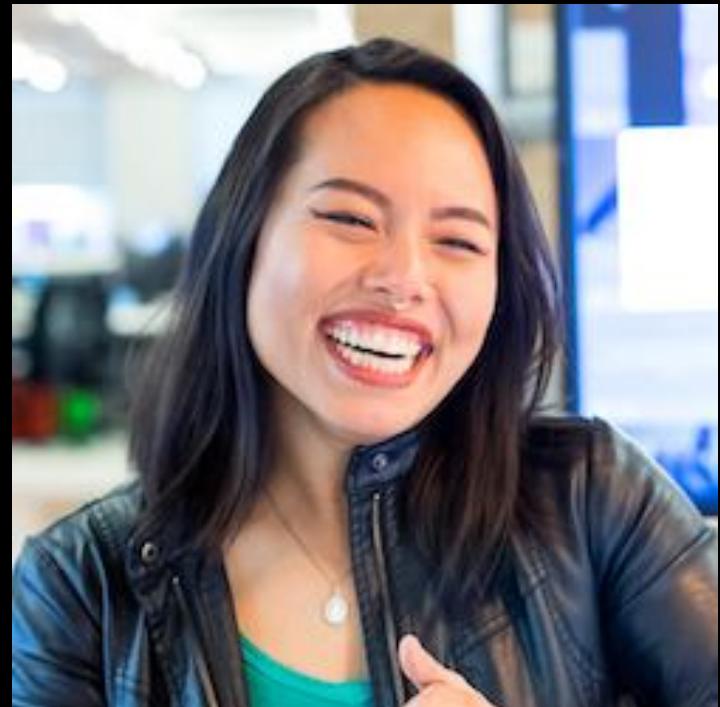
BLACK MIRROR

BUILDING A CHATBOT TO TALK LIKE YOU WITH DEEP LEARNING

ODSC 2017 WORKSHOP

ABOUT JENNIFER VAN

- **Current:**
 - Machine learning engineer
- **Former:**
 - Computational biology researcher
 - National level rugby player



ABOUT DOMENIC PUZIO

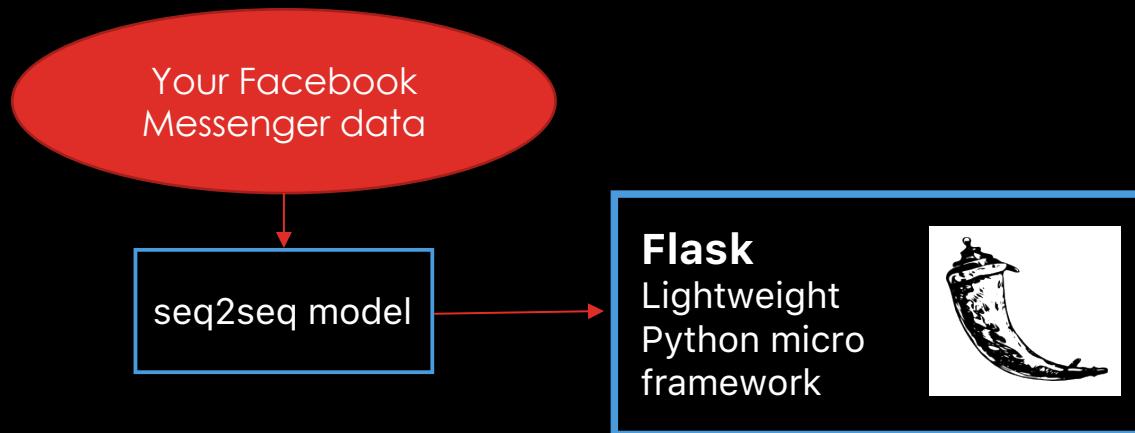
- **Current:**
 - Machine learning engineer
 - Tolkien enthusiast
- **Former:**
 - Cyber security engineer at the Federal Reserve



AGENDA

1. Introduction to neural networks and deep learning
2. Introduction to long-short term memory networks (LSTMs),
word embeddings, and seq2seq
3. Train and deploy a model
Set-up Flask, Express, and Facebook Messenger

WHAT WE WILL BUILD TODAY



WHAT YOU CAN ADD LATER

Your Facebook
Messenger data

seq2seq model

Flask
Lightweight
Python micro
framework



Express
Node.js web app
server framework

**Facebook
messenger
bot**



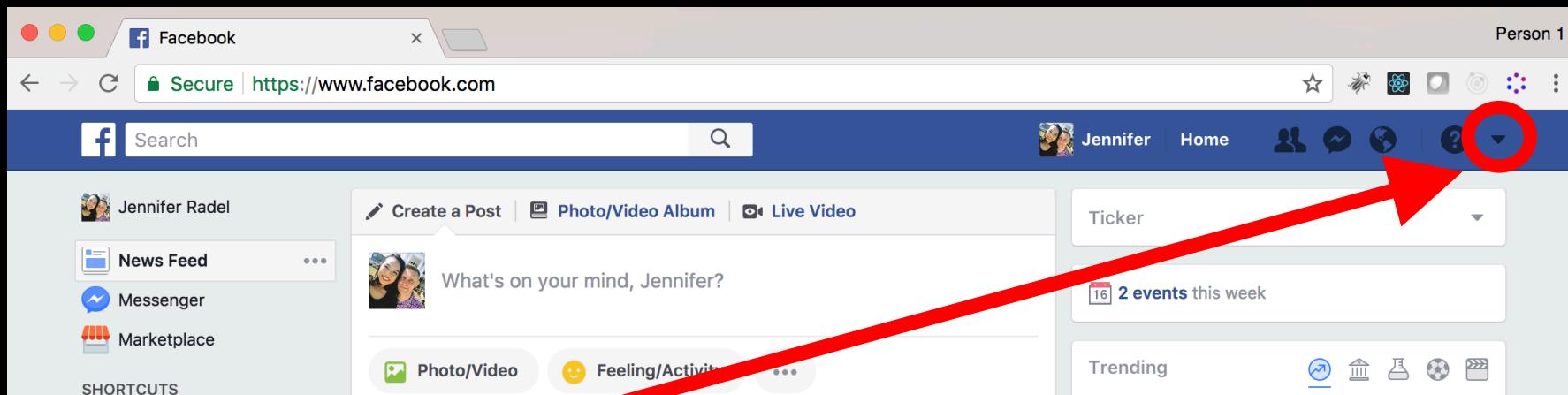
DISCLAIMER

We're going to go through a lot in a short amount of time today! If you're missing a dependency or if WiFi is giving you trouble, it might be best to follow along with the person next to you or with us on the big screen! These steps will be made available to you later on.

BEFORE WE BEGIN...

Let's get some data to play with. We will be using your Facebook messages from Facebook Messenger to train our deep learning model. If you do not have Facebook or you do not use FB Messenger, follow along with a partner nearby.

LOGIN TO FACEBOOK!



1 Click **▼** at the top right of any Facebook page and select **Settings**

1 Click ▾ at the top right of any Facebook page and select **Settings**

A screenshot of a Facebook page for a user named Jennifer Radel. The page features a background image of a cat walking through a starry, nebula-filled space. On the left, there's a sidebar with links for News Feed, Messenger, Marketplace, and various shortcuts like Short Girl Climbing C... and My Neighbor Totoro ... The main navigation bar includes a search bar, a profile picture for Jennifer, and links for Home, News Feed, Messenger, Marketplace, and a three-dot menu. A red arrow points from the top right towards the three-dot menu, which is expanded to show a list of options: Create Page, Manage Page, Create Group, Manage Groups, Create Fundraiser, Create Ad, Advertising on Facebook, Activity Log, Notifications Preferences, **Settings** (which is circled in red), Send Money, Payment History, and Log Out.

2 Click Download a copy of your Facebook data at the bottom of General Account Settings

The screenshot shows the Facebook General Account Settings page. On the left, there's a sidebar with various settings categories like General, Security and Login, Privacy, etc. The main area displays account details such as Name (Jennifer Radel), Username (http://www.facebook.com/mjvan), Contact (Primary: hitimmycosmo@aol.com), Ad account contact (hitimmycosmo@aol.com), Networks (UVA), Temperature (Fahrenheit), and Manage Account (Modify your Legacy Contact settings or deactivate your account). At the bottom, there's a link: "Download a copy of your Facebook data.". A large red arrow points from the top text instruction down to this link, and a red circle highlights it.

General Account Settings

We reorganized a few things. Password is now under **Security and Login**.

Name	Jennifer Radel	Edit
Username	http://www.facebook.com/mjvan	Edit
Contact	Primary: hitimmycosmo@aol.com	Edit
Ad account contact	hitimmycosmo@aol.com	Edit
Networks	UVA	Edit
Temperature	Fahrenheit	Edit
Manage Account	Modify your Legacy Contact settings or deactivate your account.	Edit
Download a copy of your Facebook data.		

3 Click Start My Archive

Download Your Information

Secure | https://www.facebook.com/dyi?x=AdkrO...Nk8hEe8NYv

Jennifer Home

Download Your Information

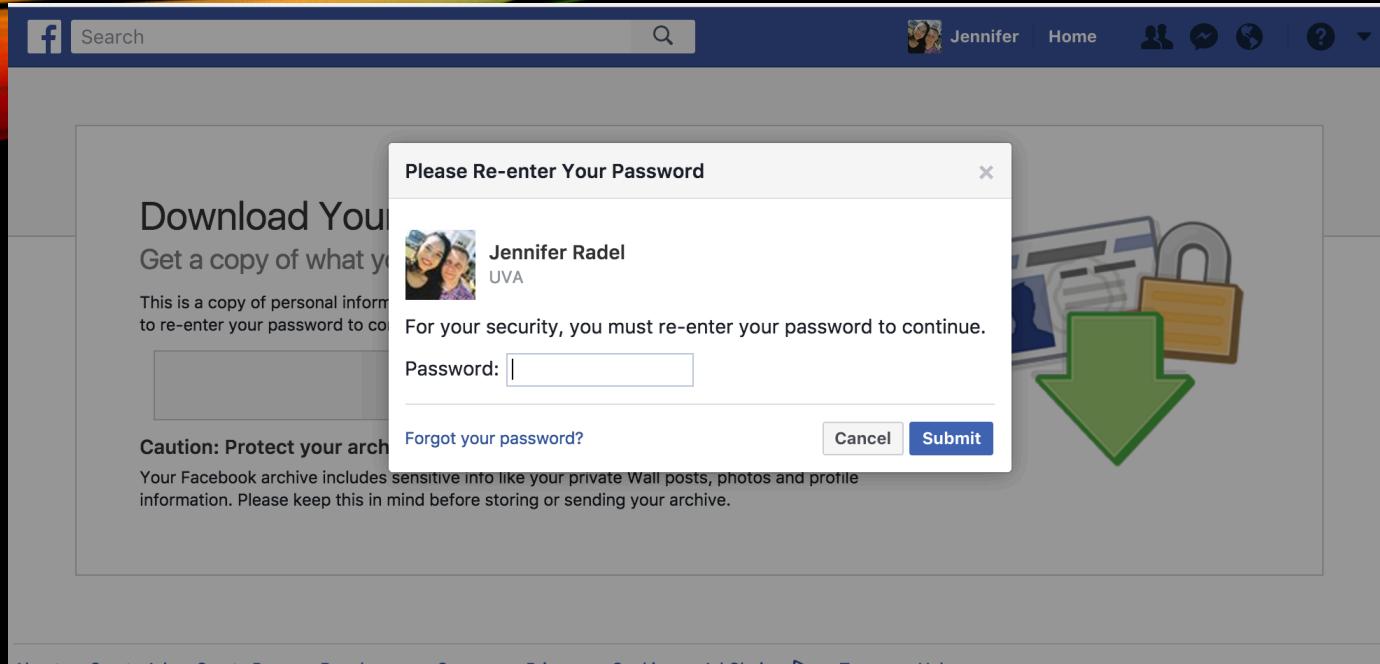
Get a copy of what you've shared on Facebook.

This is a copy of personal information you've shared on Facebook. To protect your info, we'll ask you to re-enter your password to confirm that this is your account.

Download Archive

Caution: Protect your archive

Your Facebook archive includes sensitive info like your private Wall posts, photos and profile information. Please keep this in mind before storing or sending your archive.

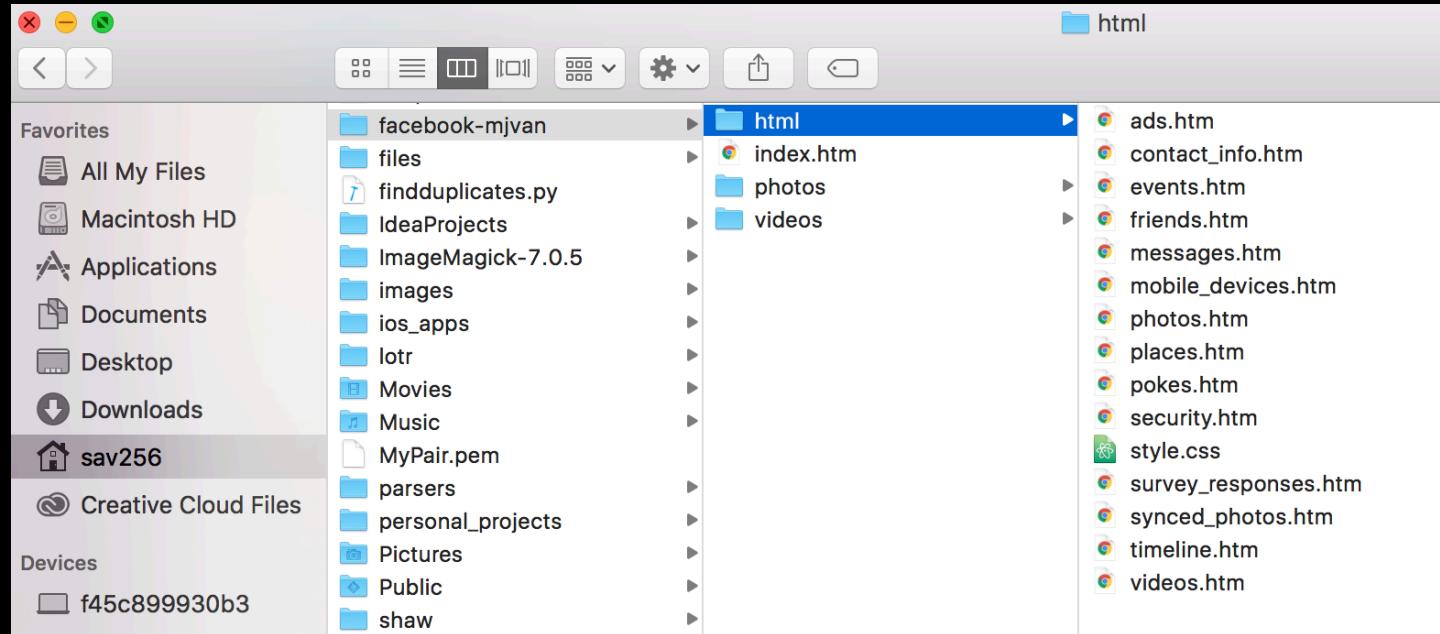


Type in your password and click 'Submit.' An archive will be sent to the email account linked to your Facebook account in 5-20 minutes. If this e-mail address is your personal e-mail that you cannot access through proxy, then log onto your personal email account and forward the email to your work email. In the meantime, let's talk about machine learning.

CLONE GITHUB REPO

```
$ git clone  
https://github.com/sugargreenbean/deep-  
learning-workshop
```

COPY YOUR HTML AND MESSAGES.HTM FILE TO YOUR GIT REPO



CREATE DATA SET

```
$ cd fb-bot/deepest-learning/preprocessing/html  
$ rm dummy-messages.htm //remove dummy file  
$ cp <<your FB folder>>/html/messages.htm fb-bot/deepest-  
learning/preprocessing/html  
$ rm fb-bot/deepest-learning/preprocessing/messages/dummy-messages.txt  
//remove dummy file  
$ cp -r <</your FB folder>>/messages fb-bot/deepest-  
learning/preprocessing/messages  
$ cd fb-bot/deepest-learning/preprocessing/html/  
$ fbcap -z CDT=-0500,CST=-0600,EDT=-0400,EST=-0400,PST=-0700,PDT=-  
0700,CET=+0100 messages.htm > fbMessages.txt  
$ python createDataset.py (Type in your full name as it appears on Facebook. If  
you have multiple names, just choose one...)
```

CREATE DATA SET

Sanity Check: How you know you did it correctly

createDataset.py should create:

- conversationDictionary.npy, numpy object with pairs in form of (FRIENDS_MESSAGE, YOUR RESPONSE)
- conversationData.txt, large text file of the dictionary data

Expected output on Terminal from python createDataset.py

```
f45c899930b3:preprocessing sav256$ python createDataset.py
Enter your full name as it appears on Facebook: Jennifer Radel
Total len of dictionary 6766
Saving conversation data dictionary
```

CREATE WORD EMBEDDINGS

```
$ cp conversationData.txt .../wordembeddings/  
$ cp conversationDictionary.npy .../wordembeddings/  
$ cd .../wordembeddings  
$ python Word2Vec.py
```

CREATE WORD EMBEDDINGS

Sanity Check: How you know you did it correctly

Word2Vec.py should show "Current loss is:..." and generate the following:

- Word2VecXTrain.npy, training matrices
- Word2VecYTrain.npy, training matrices
- wordList.txt, all unique words in corpus
- embeddingMatrix.npy, numpy matrix with all generated word vectors

```
f45c899930b3:wordembeddings sav256$ python Word2Vec.py
Finished parsing and cleaning dataset
Finished 0/163309 total words

Finished 100000/163309 total words
Finished creating training matrices
2017-11-03 02:31:36.814958: I tensorflow/core/platform/cpu_feature_guard.cc:137] Your CPU supports instructions that this TensorFlow binary
was not compiled to use: SSE4.1 SSE4.2 AVX AVX2 FMA
Current loss is: 257.353
Current loss is: 0.950654
Current loss is: 2.79727
Current loss is: 2.74299
Current loss is: 2.4493
Current loss is: 1.36529
Current loss is: 1.79511
Current loss is: 1.76605
Current loss is: 2.26754
Current loss is: 4.67828
Saving the word embedding matrix
```

TRAIN YOUR SEQ2SEQ MODEL

```
$ cp conversationData.txt .../seq2seq  
$ cp conversationDictionary.npy .../seq2seq  
$ cp embeddingMatrix.npy .../seq2seq  
$ cp wordList.txt .../seq2seq  
$ cp Word2VecXTrain.npy .../seq2seq  
$ cp Word2VecYTrain.npy .../seq2seq  
$ cd .../seq2seq  
$ python Seq2Seq.py
```

Sanity Check: How you know you did it correctly

Seq2Seq.py should show “Current loss: ... at iteration ...” and show a phrase with an array. These phrases are hard-coded to validate against your model, and what’s in the bracket is what your model would respond with. In the beginning, your array is empty because your model doesn’t know what to respond with. As well, Seq2Seq.py should create:

- Seq2SeqXTrain.npy, training matrices that Seq2Seq will use
- Seq2SeqYRain.npy
- models folder contains checkpoints

Section 1

Intro to Neural Networks & Deep Learning



What is a neuron?

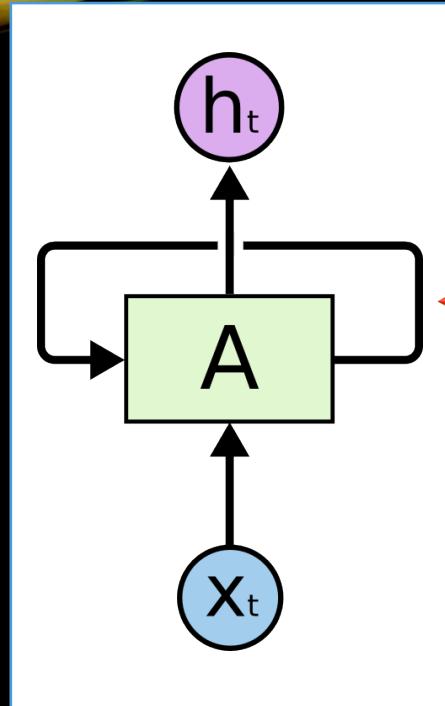


*What is a neural
network?*



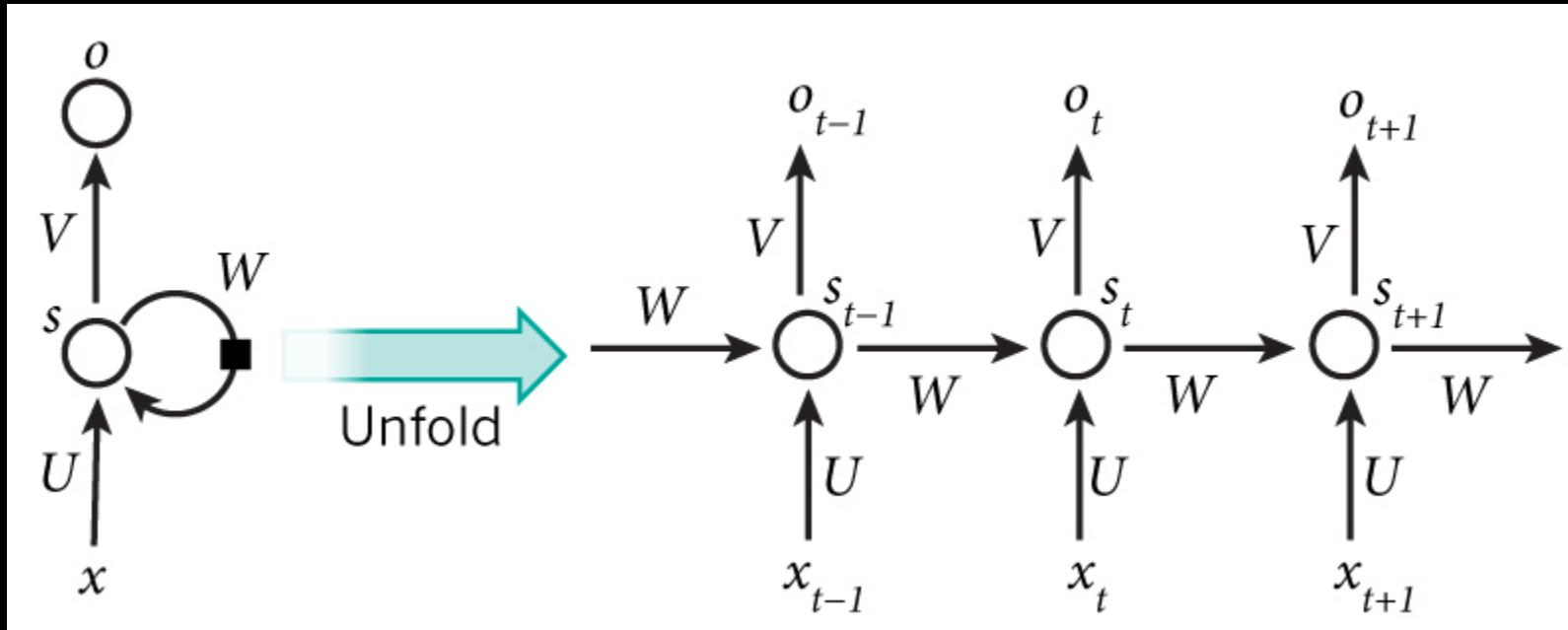
*What is a
recurrent neural
network (RNN)?*

Key takeaway:
Loops provide memory

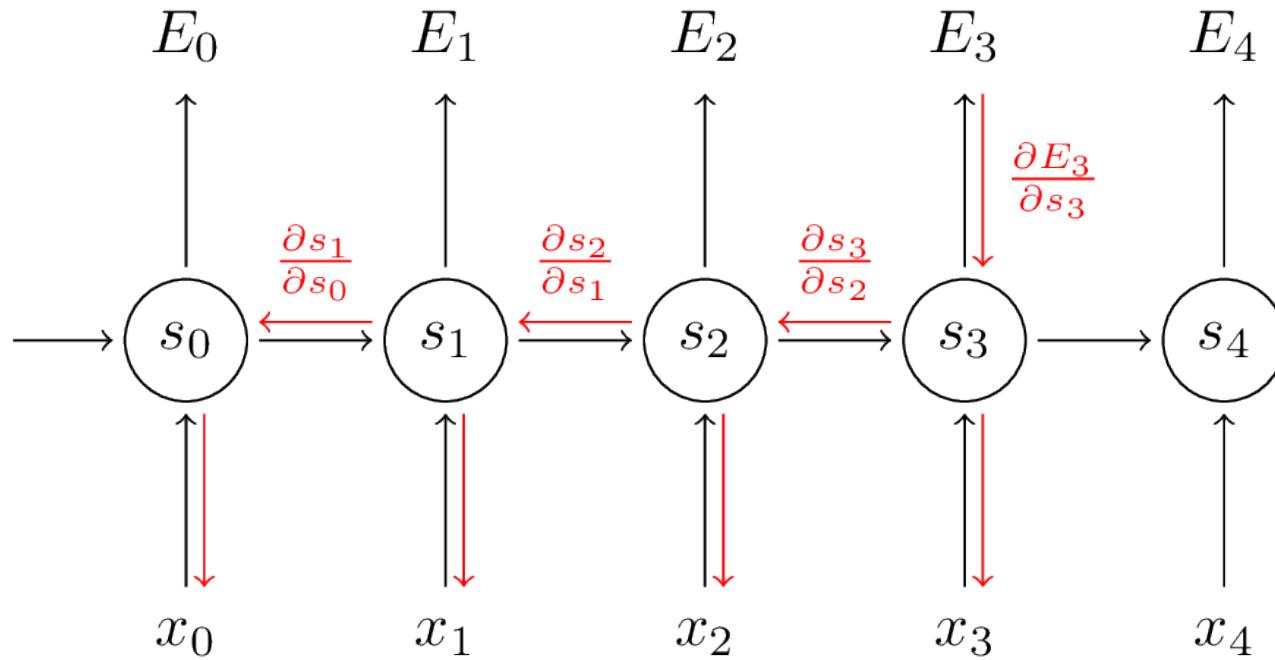


RECURRENT NEURAL NETWORK (RNN)

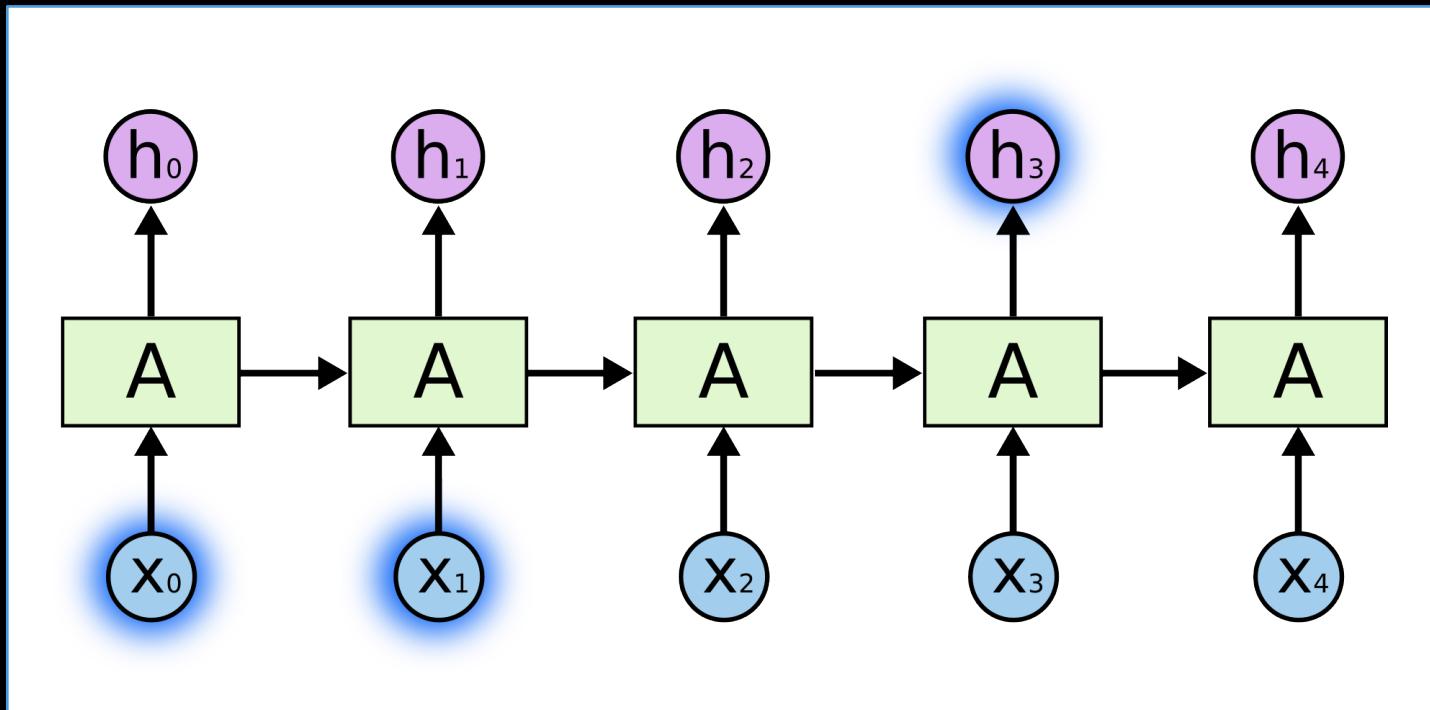
RNN'S IN DEPTH



BACKPROPAGATION THROUGH TIME

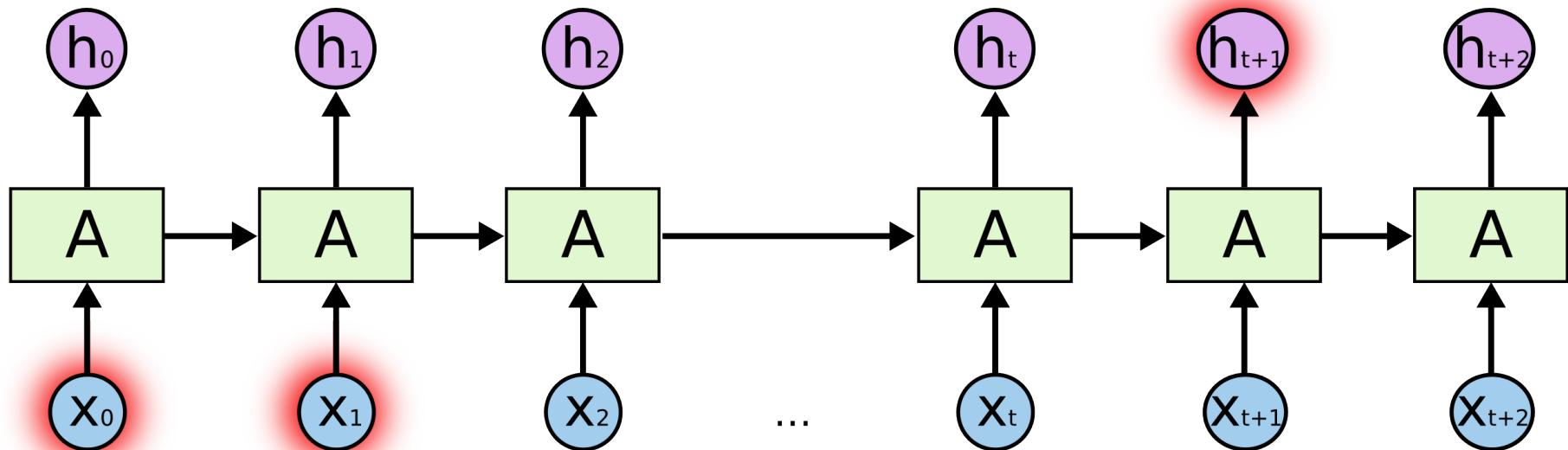


LIMITATIONS OF RNNS



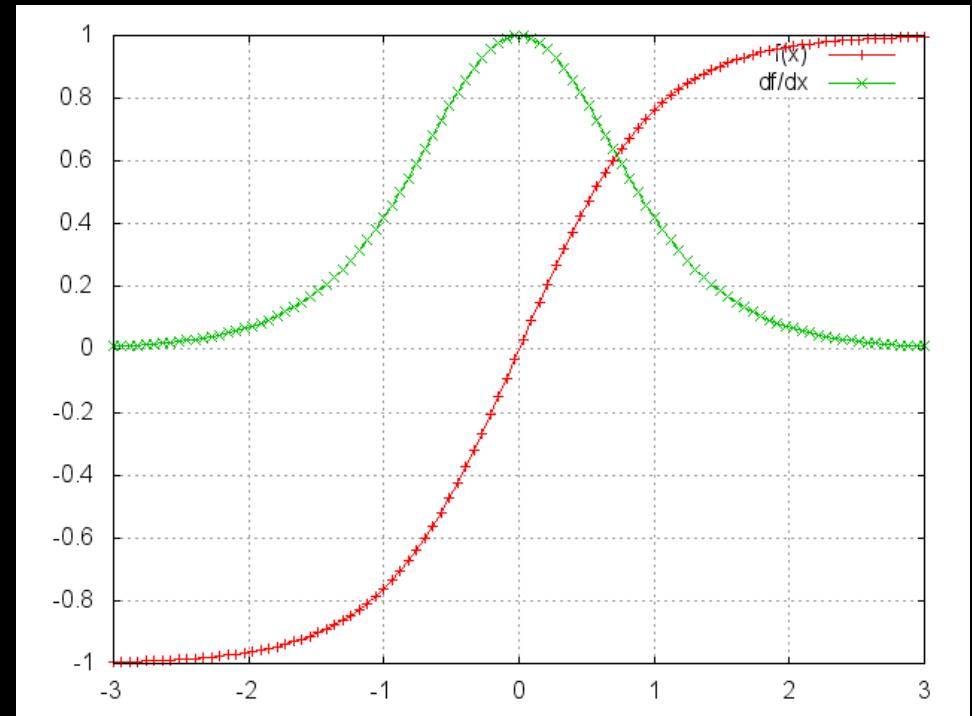
Source: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

LONG-TERM DEPENDENCIES



VANISHING GRADIENT

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \boxed{\frac{\partial s_3}{\partial s_k}} \frac{\partial s_k}{\partial W}$$



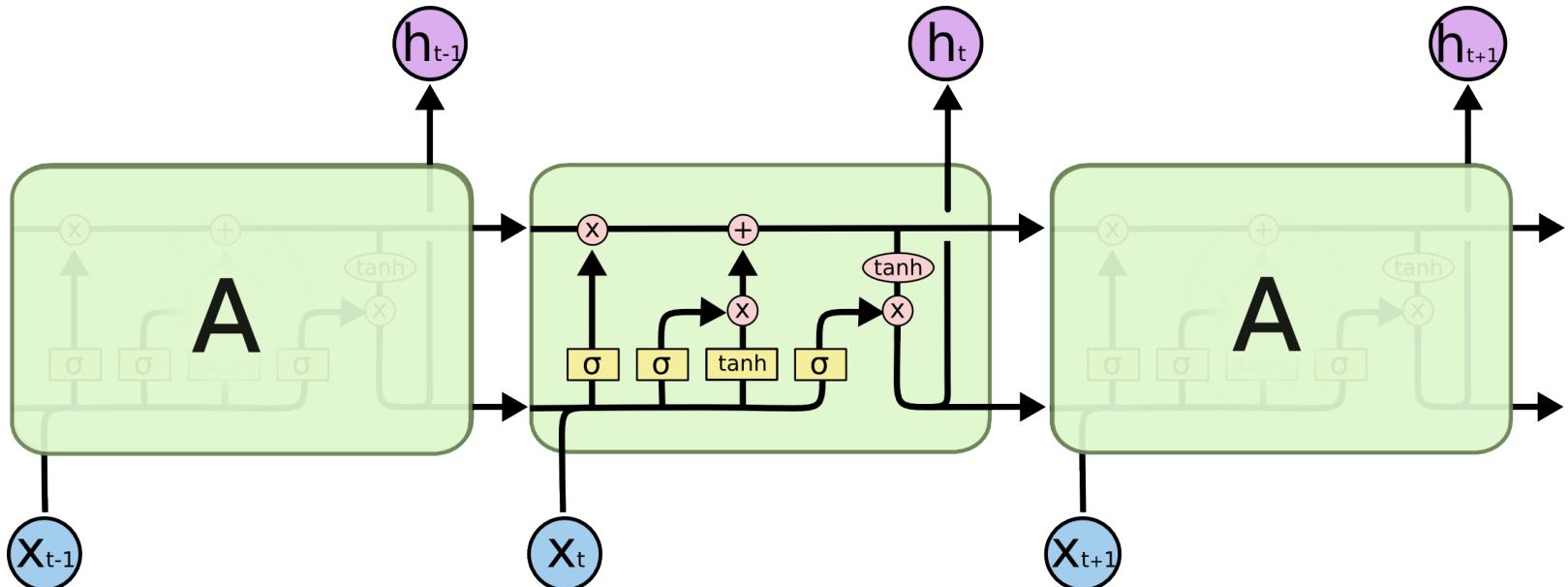
Section 2

Long Short-Term Memory Networks, Word Embeddings, and Sequence2Sequence

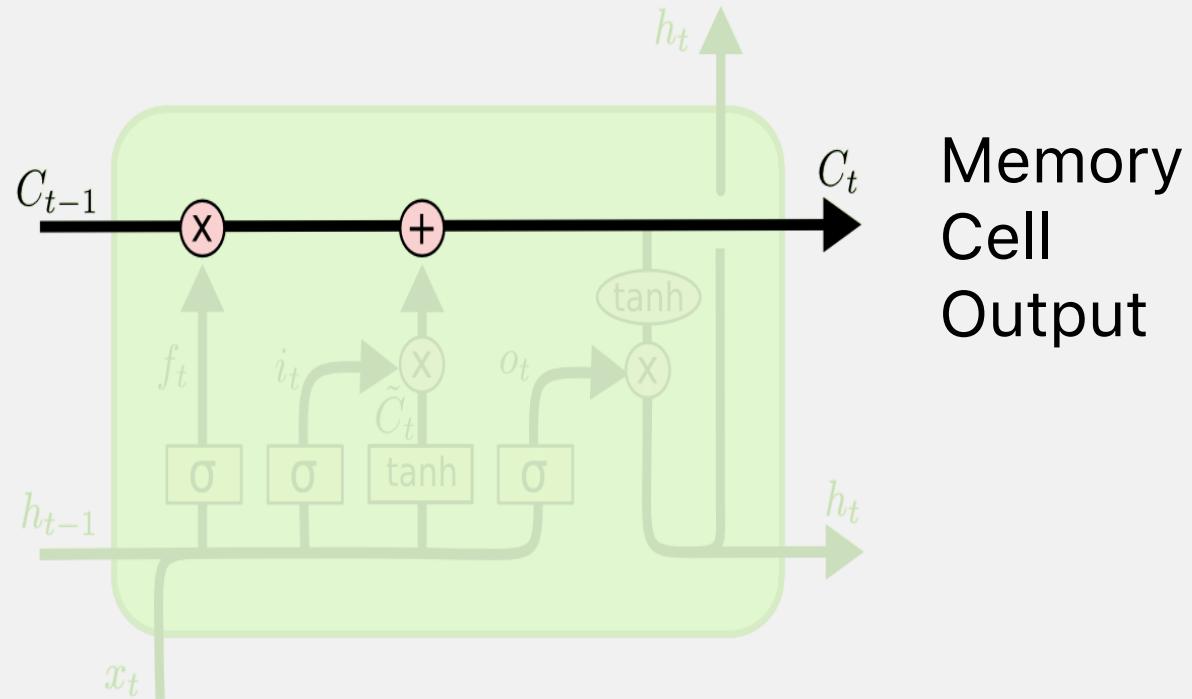


*What is a long short-term memory NN
(LSTM)?*

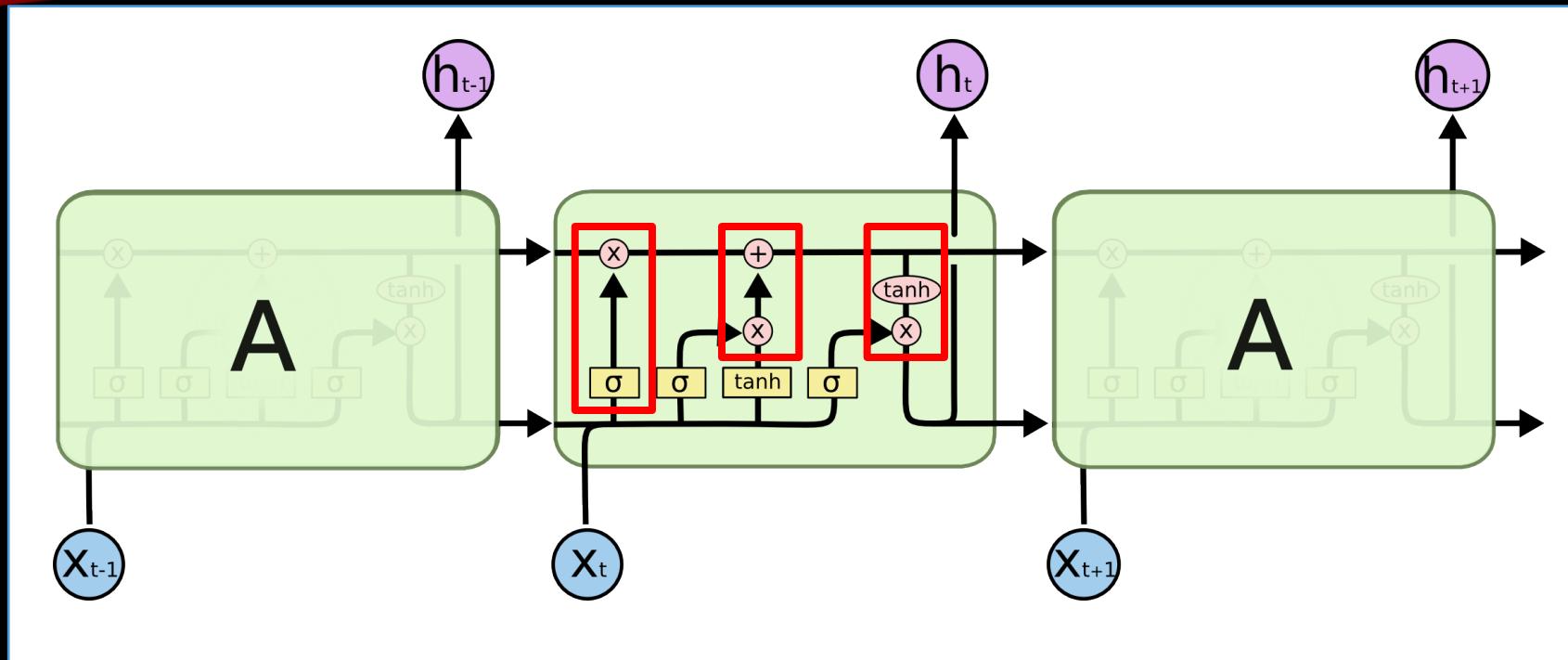
SOLUTION: LONG SHORT-TERM MEMORY



Memory
Cell
Input



MEMORY CELL GATES





*What are word
embeddings?*

I WANT TO USE MATH ON WORDS!

One-Hot
Encoding

$V \times 1$
sparse
vector

0
0
...
0
0
1
0

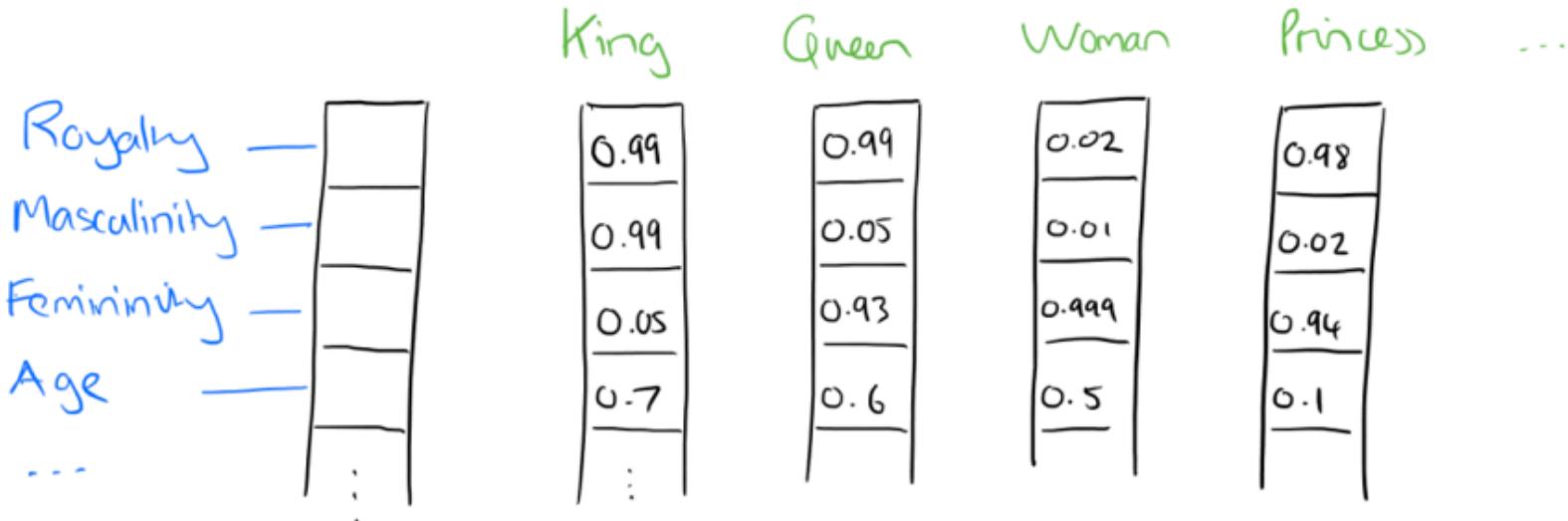
VS.

Dense
Embedding

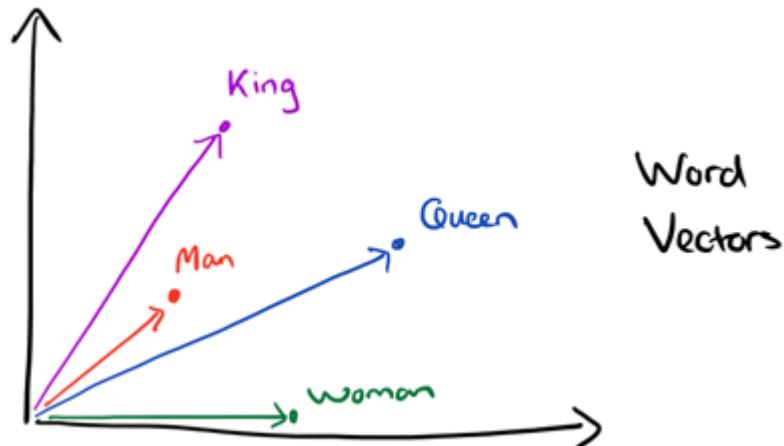
$D \times 1$
dense
vector

1.24
-
0.24
...
2.51

DENSE EMBEDDING (WORD2VEC)

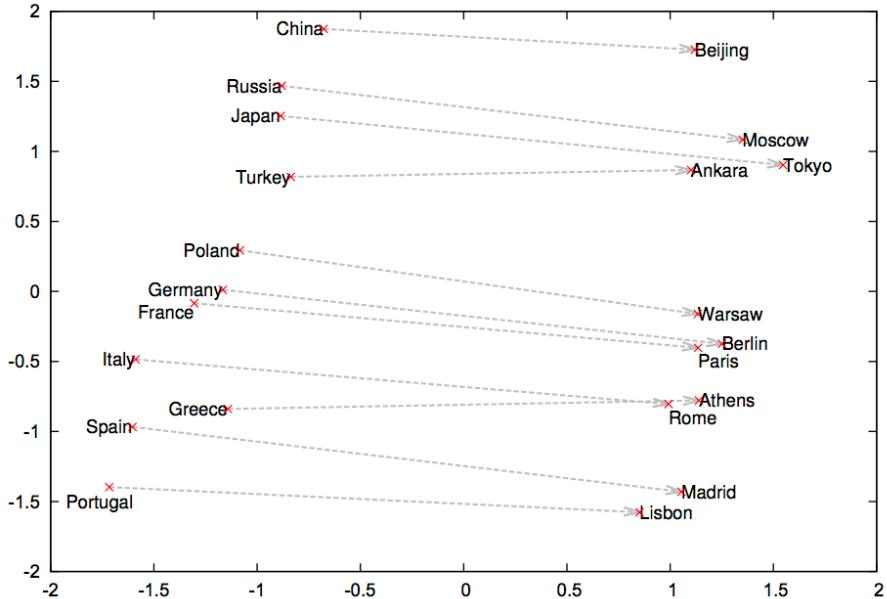


DENSE EMBEDDING (WORD2VEC)



DENSE EMBEDDING (WORD2VEC)

Country and Capital Vectors Projected by PCA

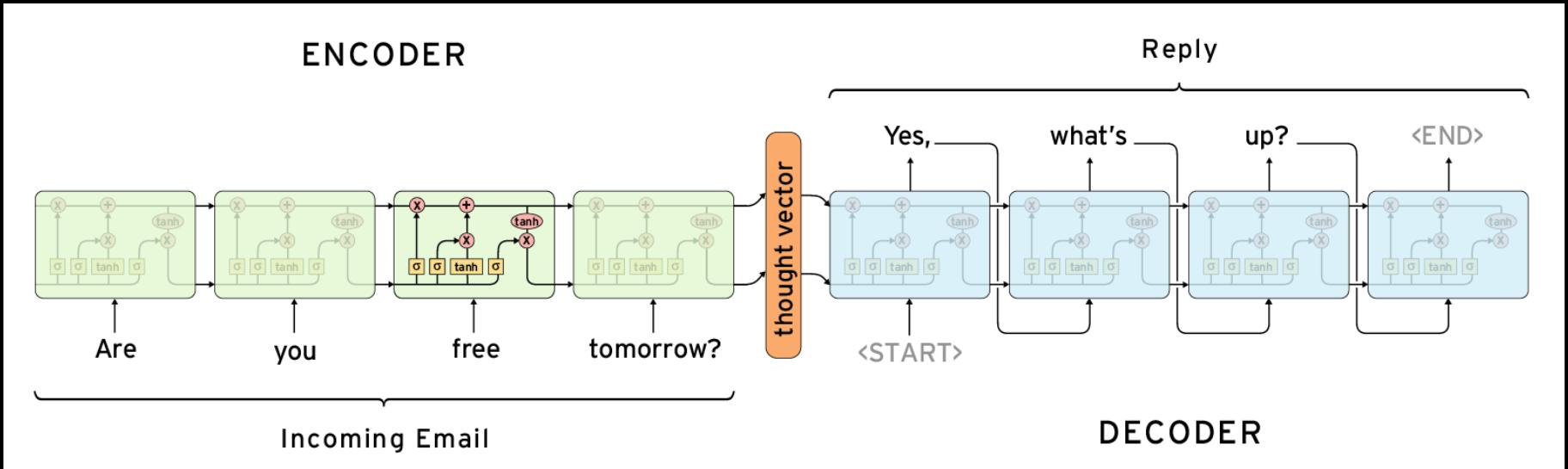


Newspapers			
New York	New York Times	Baltimore	Baltimore Sun
San Jose	San Jose Mercury News	Cincinnati	Cincinnati Enquirer
NHL Teams			
Boston	Boston Bruins	Montreal	Montreal Canadiens
Phoenix	Phoenix Coyotes	Nashville	Nashville Predators
NBA Teams			
Detroit	Detroit Pistons	Toronto	Toronto Raptors
Oakland	Golden State Warriors	Memphis	Memphis Grizzlies
Airlines			
Austria	Austrian Airlines	Spain	Spanair
Belgium	Brussels Airlines	Greece	Aegean Airlines
Company executives			
Steve Ballmer	Microsoft	Larry Page	Google
Samuel J. Palmisano	IBM	Werner Vogels	Amazon



*What is a sequence
to sequence
learning?*

SEQ2SEQ



Section 3

Train & deploy a model

RECAP: TRAINING A SEQ2SEQ MODEL

1. Parse words out and create a dictionary
2. Create word vectors
3. Train seq2seq model and save it

DEPLOYMENT: A CLOSER LOOK

1. Deploy Flask server to serve your model's predictions
2. *Create a Facebook messenger bot*
Do this on your own at a later time
3. *Deploy Express app with Heroku*
Do this on your own at a later time

DEPLOY EXPRESS APP IN HEROKU

```
$ cp fb-bot/express-server ../  
$ cd ../  
$ npm init (ctrl+c)
```

```
f45c899930b3:express-server sav256$ npm init  
This utility will walk you through creating a package.json file.  
It only covers the most common items, and tries to guess sensible defaults.  
  
See `npm help json` for definitive documentation on these fields  
and exactly what they do.  
  
Use `npm install <pkg>` afterwards to install a package and  
save it as a dependency in the package.json file.  
  
Press ^C at any time to quit.  
package name: (fb-bot) |
```

DEPLOY EXPRESS APP IN HEROKU

```
$ npm install express request body-parser –save  
$ git init  
$ git add .  
$ git commit --m "hello world"  
$ heroku create  
$ git push heroku master
```

CONNECT EXPRESS TO FACEBOOK

Token Generation

Page token is required to start using the APIs. This page token will have all messenger permissions even if your app is not approved to use them yet, though in this case you will be able to message only app admins. You can also generate page tokens for the pages you don't own using Facebook Login.

Page	Page Access Token
Odsc-Bot-Test ▾	EAAQLRT9PGZBYBAHYVB5M9AzF1WSfUwmlwSVQJyslJl30Vm92pXQA9c1g5UZADg3X3oVI2bwSWwiiR1KlpZAjHXYdokZC3ZBmzJjEc2b2mhETaUZBhecJQaTqZAyfsfsCc8PsvUXvndKJ3QE1pWmxX8Wz5idTLIyA9Q8Fv2IM0Pl6AZDZD

Create a new page

```
f45c899930b3:express-server sav256$ heroku config:add PAGE_ACCESS_TOKEN=EAAQLRT9PGZBYBAHYVB5M9AzF1WSfUwmlwSVQJyslJl30Vm92pXQA9c1g5UZADg3X3oVI2bwSWwiiR1KlpZAjHXYdokZC3ZBmzJjEc2b2mhETaUZBhecJQaTqZAyfsfsCc8PsvUXvndKJ3QE1pWmxX8Wz5idTLIyA9Q8Fv2IM0Pl6AZDZD
  ▶ heroku-cli: This CLI is deprecated. Please reinstall from https://cli.heroku.com
Setting PAGE_ACCESS_TOKEN and restarting ⚡ young-cliffs-96893... done, v3
PAGE_ACCESS_TOKEN: EAAQLRT9PGZBYBAHYVB5M9AzF1WSfUwmlwSVQJyslJl30Vm92pXQA9c1g5UZADg3X3oVI2bwSWwiiR1KlpZAjHXYdokZC3ZBmzJjEc2b2mhETaUZBhecJQaTqZAyfsfsCc8PsvUXvndKJ3QE1pWmxX8Wz5idTLIyA9Q8Fv2IM0Pl6AZDZD
```

```
$ heroku config:add PAGE_ACCESS_TOKEN=<your page token here>
```

CONNECT EXPRESS TO FACEBOOK

```
$ curl -X POST  
https://graph.facebook.com/v2.6/me/subscribed_apps?access_token=<PAGE_ACCE  
SS_TOKEN>
```

```
// Trigger Facebook app to send messages
```



Congrats! Facebook and Heroku can now
speak to each other

ADD YOUR MODEL TO FLASK SERVER

```
$ cp -r fb-bot/deepest-learning/seq2seq/models  
/fb-bot/flask-server/models
```

//copy folder from seq2seq training to flask server

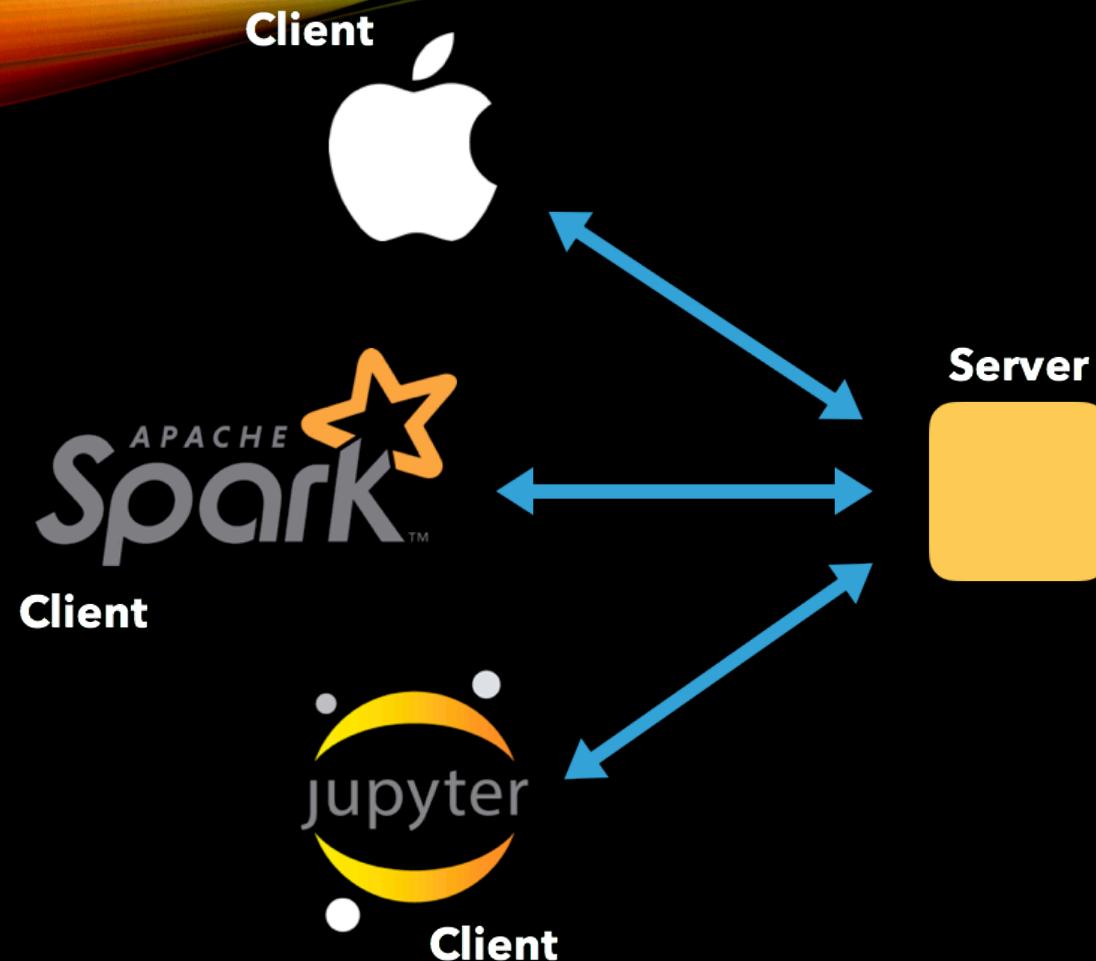
```
$ cp fb-bot/deepest-  
learning/wordembeddings/wordList.txt fb-  
bot/flask-server/data
```

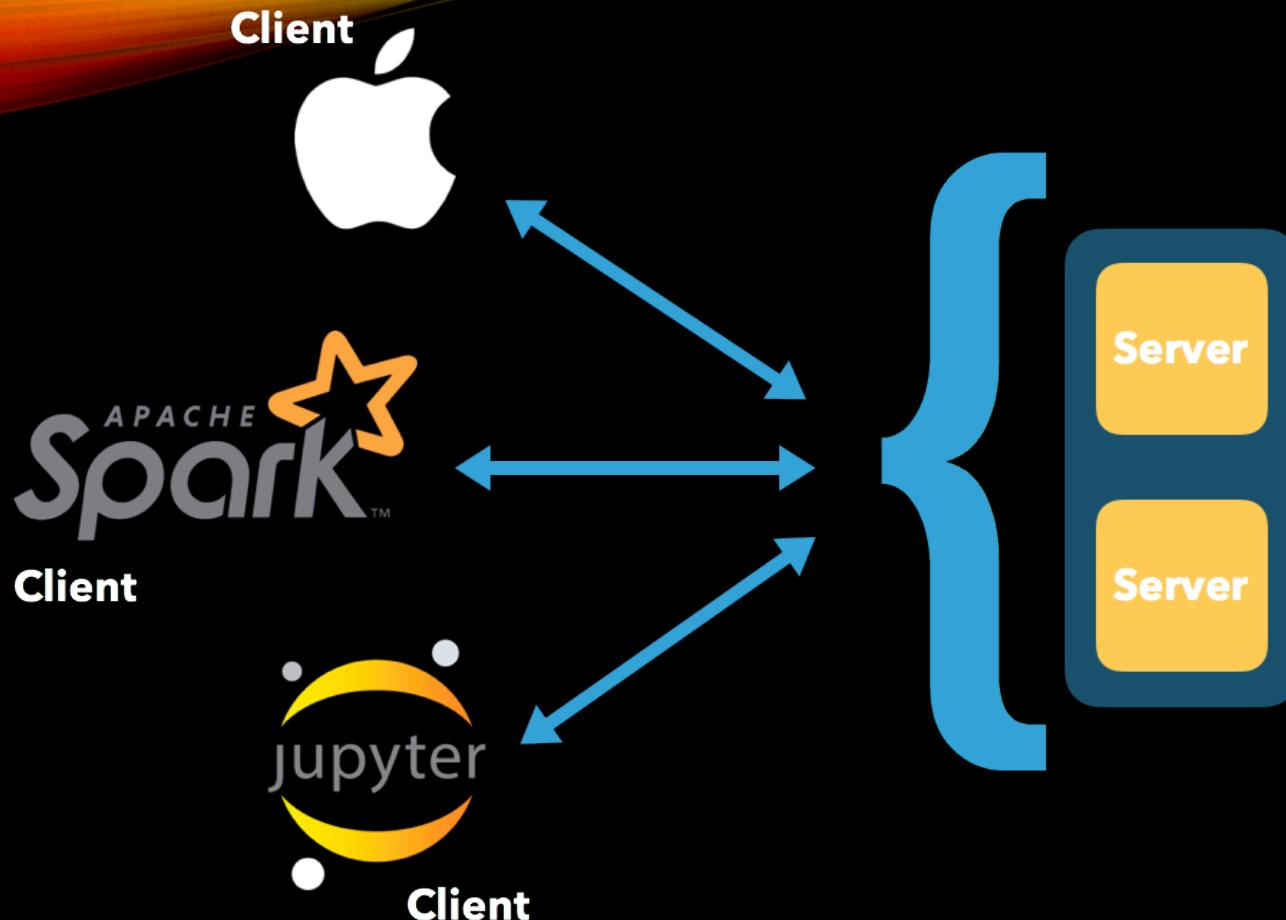
RUN FLASK SERVER

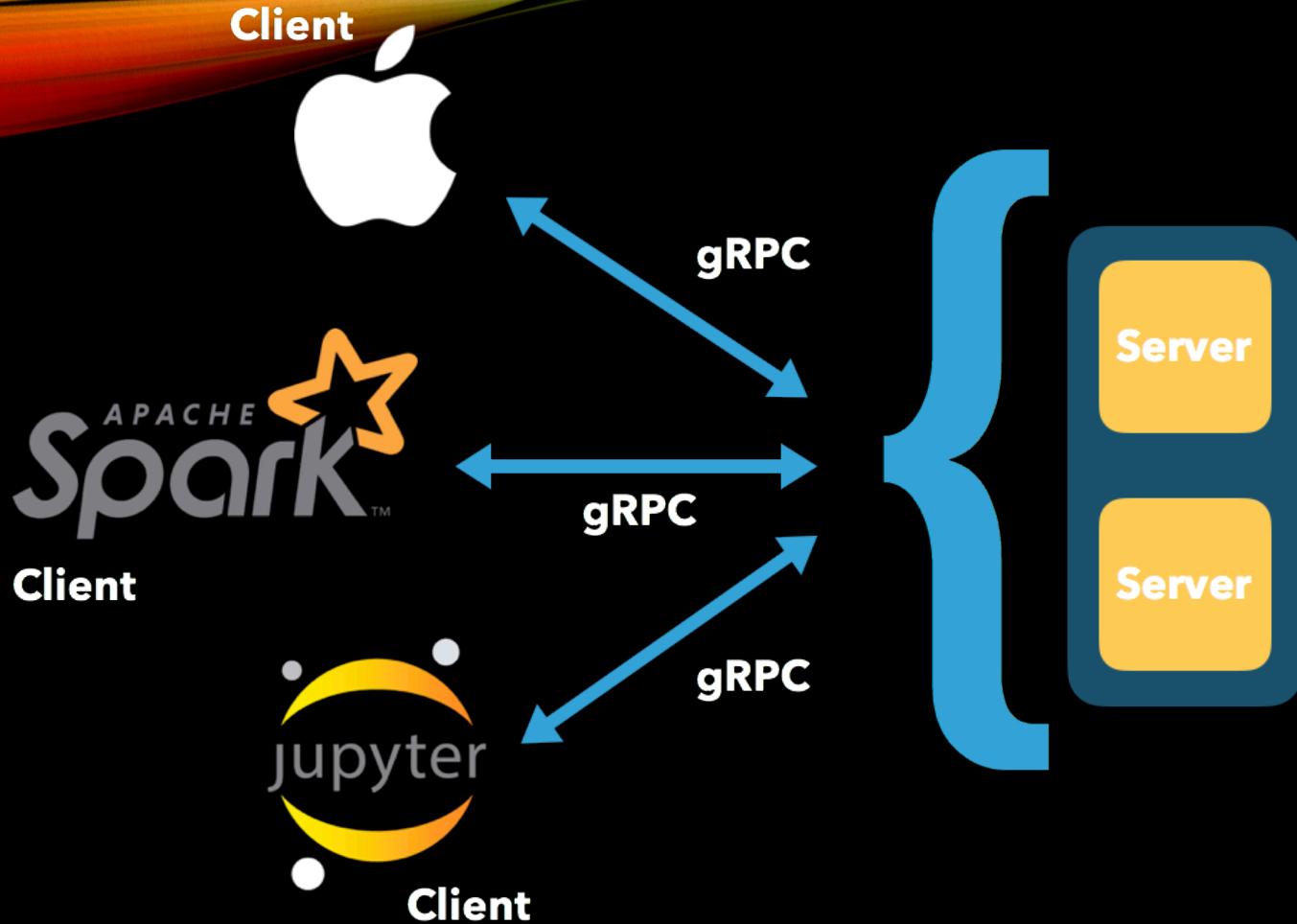
```
$ cd fb-bot/flask-server  
$ python app.py  
//Wait 30 seconds  
//Go to ip-address in browser
```

DEPLOYMENT ++

- A solo Flask server is perfect for a few users
- But let's say we want to scale up...







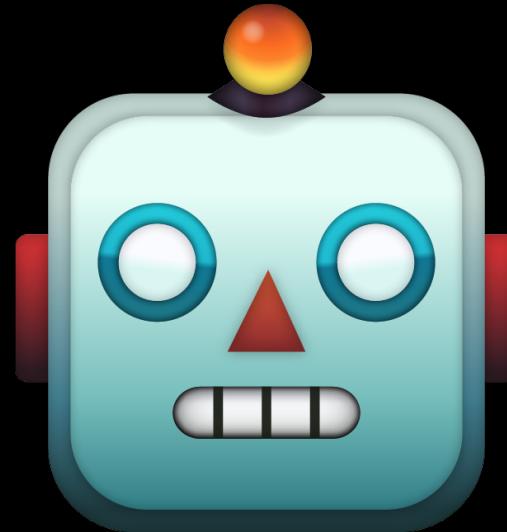
GRPC VS. FLASK

- Increases speed significantly
- Makes scaling easy
- Forces strict type checks
- Adds complexity

SUMMARY

What did we learn today?

- Brief intro to neural networks
- RNNs, LSTMs, and word embeddings
- Sequence to sequence modeling
- Model deployments



FOLLOW-UP

- Increase training data set for seq2seq
- Increase training iterations for seq2seq
- Further resources:
 - Elements of Statistical Learning:
http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf
 - Intro to ML: <https://www.coursera.org/learn/machine-learning/home/info>
 - Deep Learning book: https://www.deeplearningbook.org/front_matter.pdf
 - Keep up with current research: <https://arxiv.org/>, <http://www.arxiv-sanity.com/>