Lecture Notes for

Machine Learning in Python

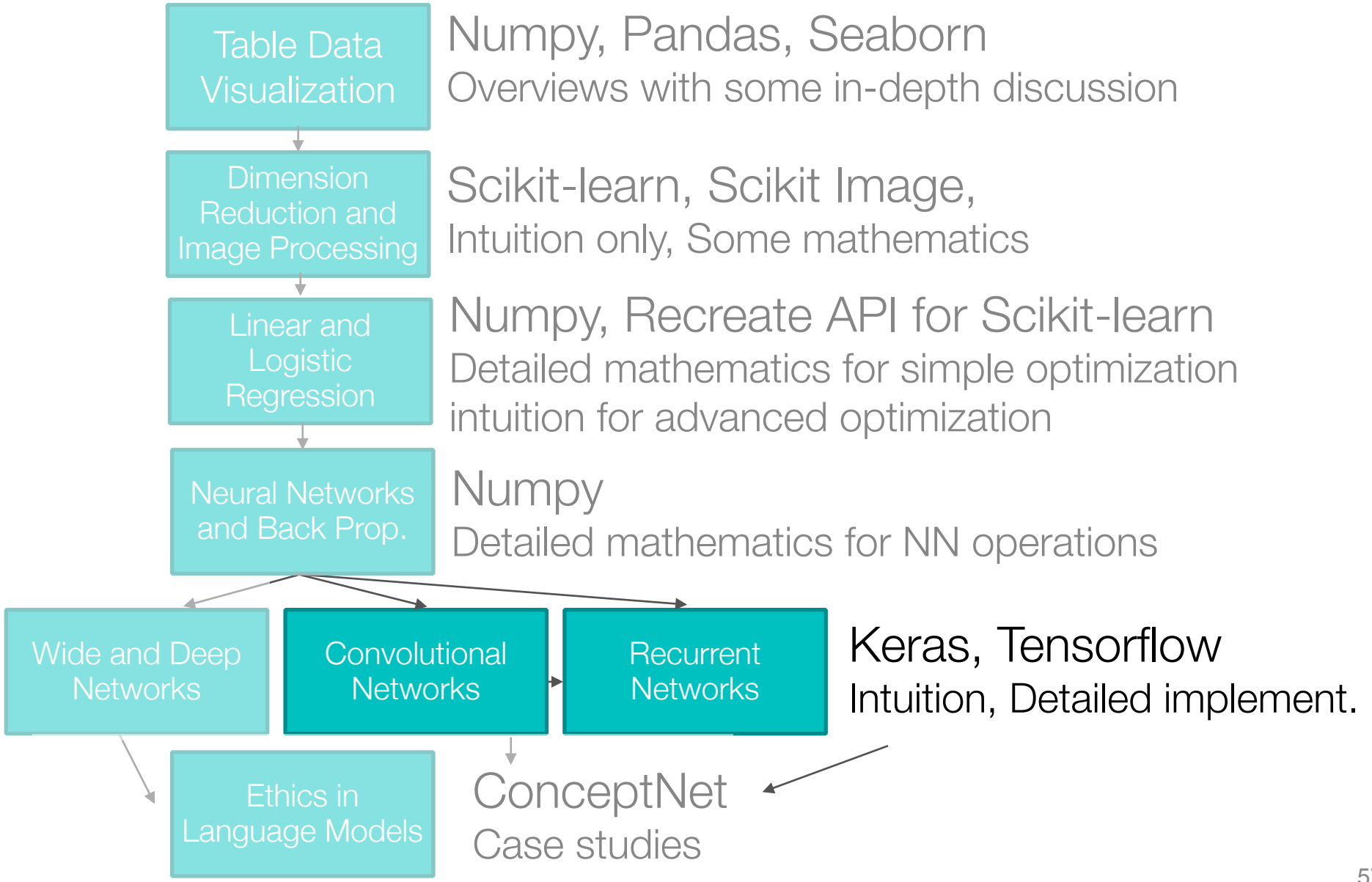
Professor Eric Larson

Sequential Processing with RNNs and CNNs

Lecture Agenda

- Logistics
  - RNNs due **Last Day of Finals**
- Agenda
  - Finish RNN/CNN Demo
  - Town Hall
- Next Time:
  - An Ethical Case Study
  - Retrospective and Evaluations

Class Overview, by topic



Last Time

- LSTM prototype

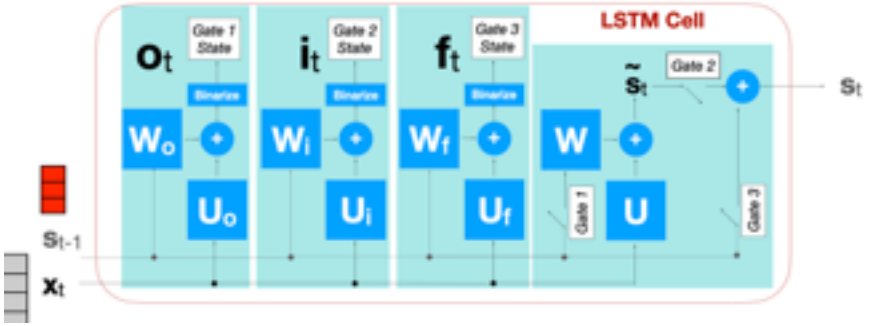
Selectivity controls (**gates, 0 or 1**)

$$o_t = \sigma(W_o s_{t-1} + U_o x_t + b_o)$$
$$i_t = \sigma(W_i s_{t-1} + U_i x_t + b_i)$$
$$f_t = \sigma(W_f s_{t-1} + U_f x_t + b_f)$$

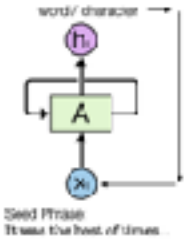
selectively remember past / with influence

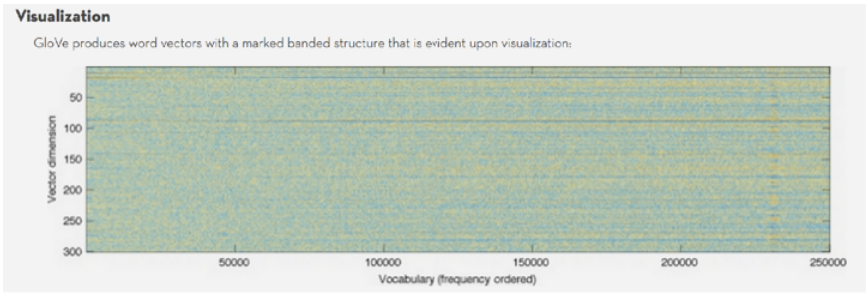
$$\tilde{s}_t = \phi(W(o_t \odot s_{t-1}) + Ux_t + b)$$
$$s_t = f_t \odot s_{t-1} + i_t \odot \tilde{s}_t$$

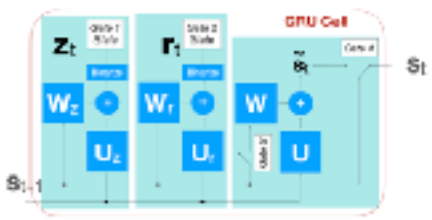
selectively remember past / with past weighted influence



- Highly "sophisticated" steps:
  - train an RNN to generate the **next** word/character from the **current** word/character
  - train on a corpus of text
    - Shakespeare
    - Movie Scripts
    - Whatever!
  - seed with random word, feed output words as input to next node
  - rinse, repeat







$$r_t = \sigma(W_r s_{t-1} + U_r x_t + b_r)$$
$$z_t = \sigma(W_z s_{t-1} + U_z x_t + b_z)$$

$$\tilde{s}_t = \phi(W(r_t \odot s_{t-1}) + Ux_t + b)$$
$$s_t = z_t \odot s_{t-1} + (1 - z_t) \odot \tilde{s}_t$$

To back propagate, we need sensitivity of state vector, w.r.t previous state

Product Rule

Product Rule

$$\partial s_t / \partial s_{t-1} = (\partial z_t \times s_{t-1}) + (\partial s_{t-1} \times z_t) + \partial \tilde{s}_t - (\partial z_t \times \tilde{s}_t) - (\partial \tilde{s}_t \times z_t)$$

likely vanish

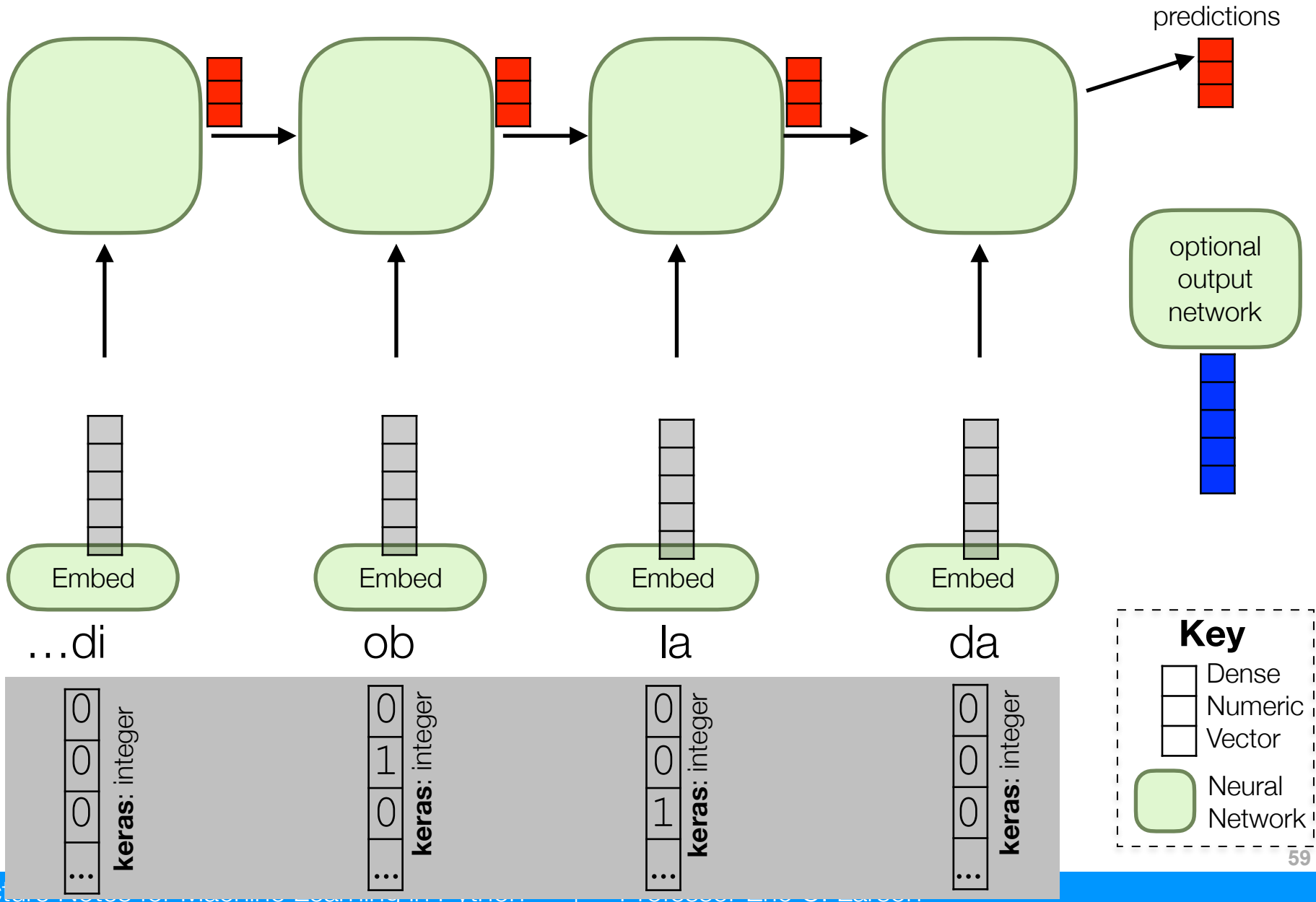
could vanish, depending on ϕ

likely vanish

could vanish, depending on ϕ

hard to vanish unless $z_t = 0$

Review: General recurrent flow (many to one)



Final Demo - Part A

Many to one:

- Simple RNNs
- GRUs
- LSTMs

More examples:

https://github.com/tensorflow/tensorflow/tree/r0.11/tensorflow/examples/skflow
http://r2rt.com/recurrent-neural-networks-in-tensorflow-i.html
http://machinelearningmastery.com/sequence-classification-lstm-recurrent-neural-networks-python-keras/

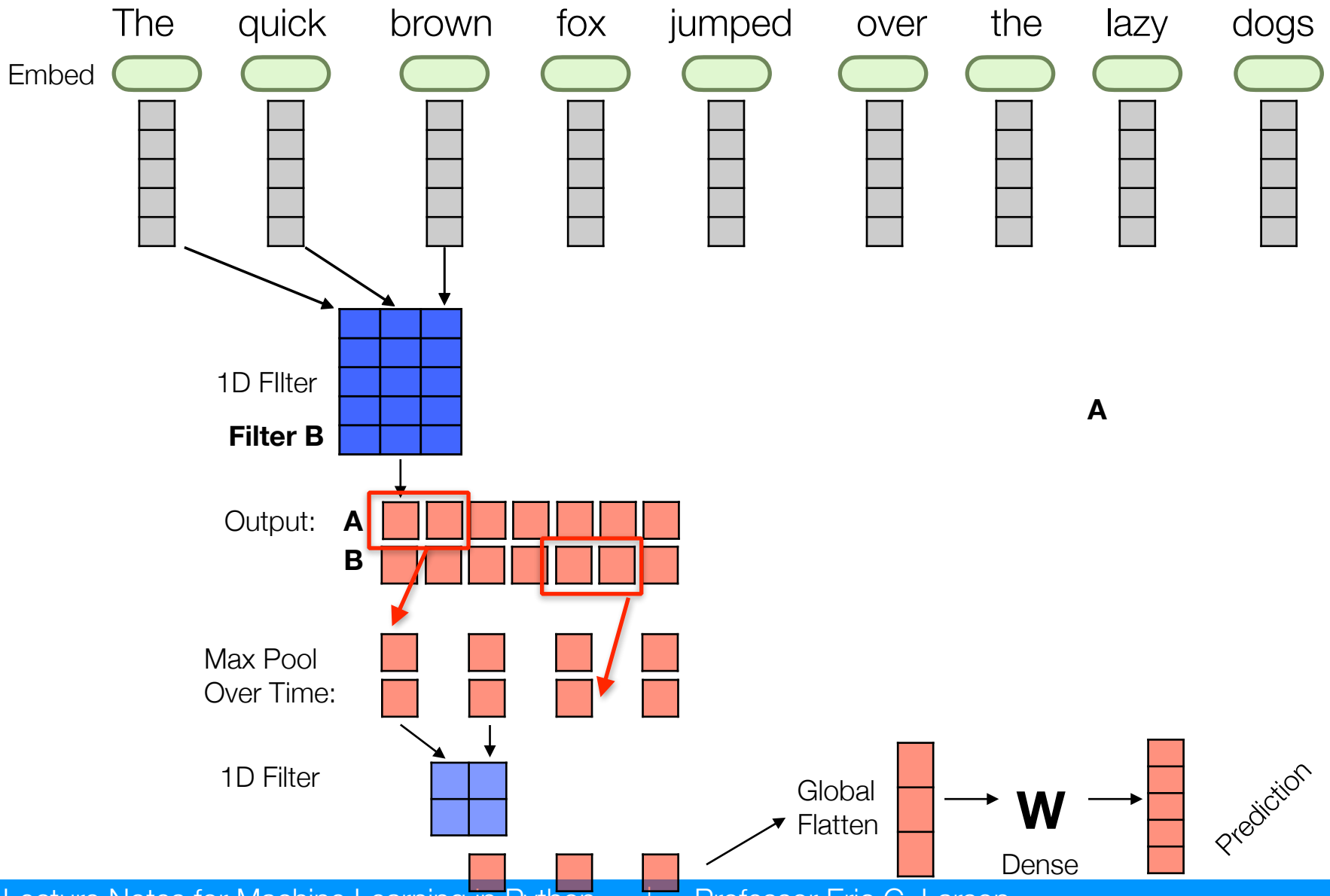
Seq2Seq:

https://github.com/tensorflow/tensorflow/blob/r0.11/tensorflow/examples/skflow/neural_translation_word.py

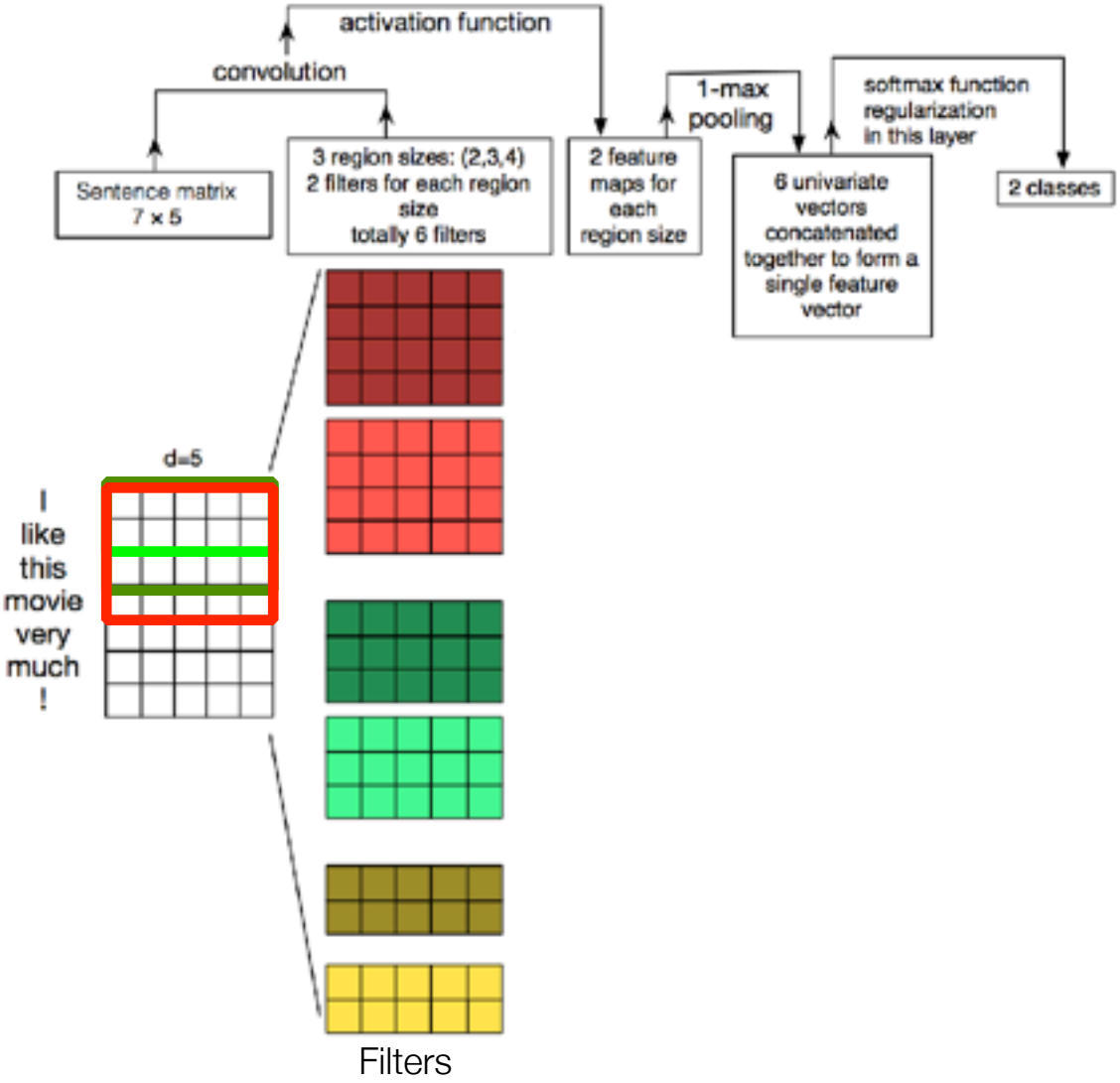
CNNs for Sequences

CNNs and RNNs

is an RNN similar to a CNN?



CNNs with Multiple Region Sizes



http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

Demo - Part B

Back to the CNN

More examples:

http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

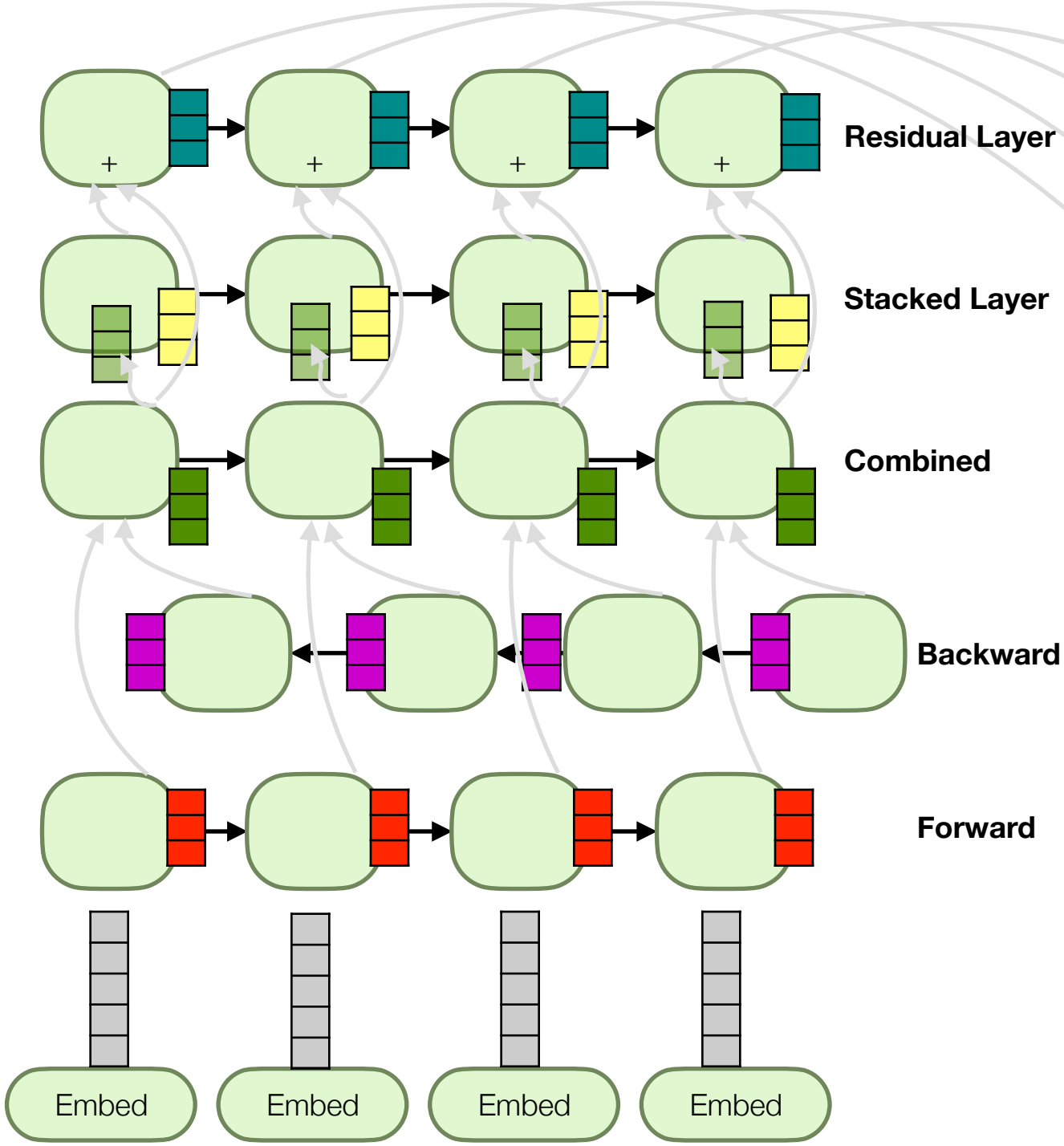
Seq2Seq:

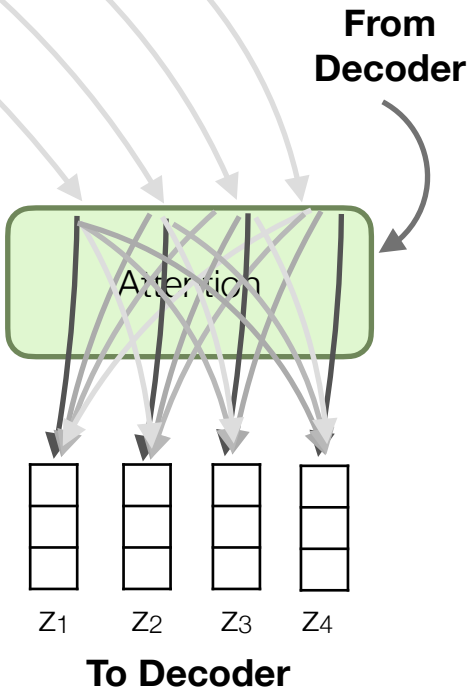
https://github.com/tensorflow/tensorflow/blob/r0.11/tensorflow/examples/skflow/neural_translation_word.py

RNN Complexity

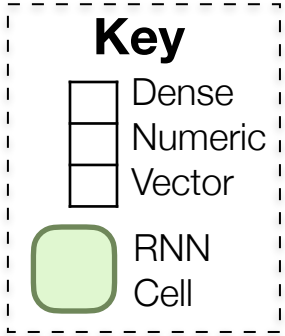
Everything we have studied from CNNs applies to RNNs as well

More Complex RNNs





Encoder LSTMs



Yann LeCun @ylecun · 6h

The *actual* Fashion MNIST.

Town Hall