Google Cloud

# Recurrent Neural Networks (RNNs)

# Advanced ML with TensorFlow on GCP

End-to-End Lab on Structured Data ML

Production ML Systems

Image Classification Models

**Sequence Models**

Recommendation Systems

# Agenda

Recurrent Neural Networks (RNNs)

RNNs and the variable-length problem

Optimizing procedure

Review RNN limitations

# RNNs handle variable-length sequences differently

Representing an entire variable-length sequence

Representing a single event given what has come before.

# This is not just reasonable but actually consistent with what you do every day

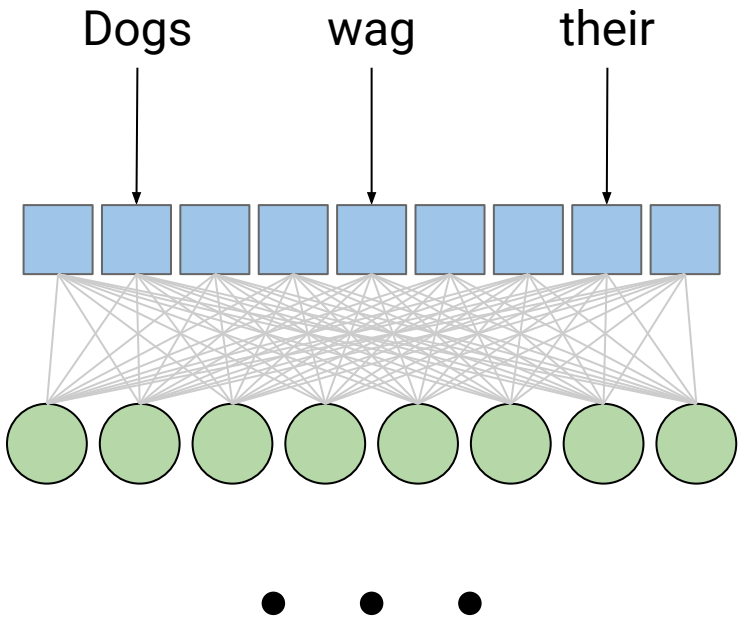
Dogs are my favorite animals, I love how they wag their _____

Humans build up representations over time.

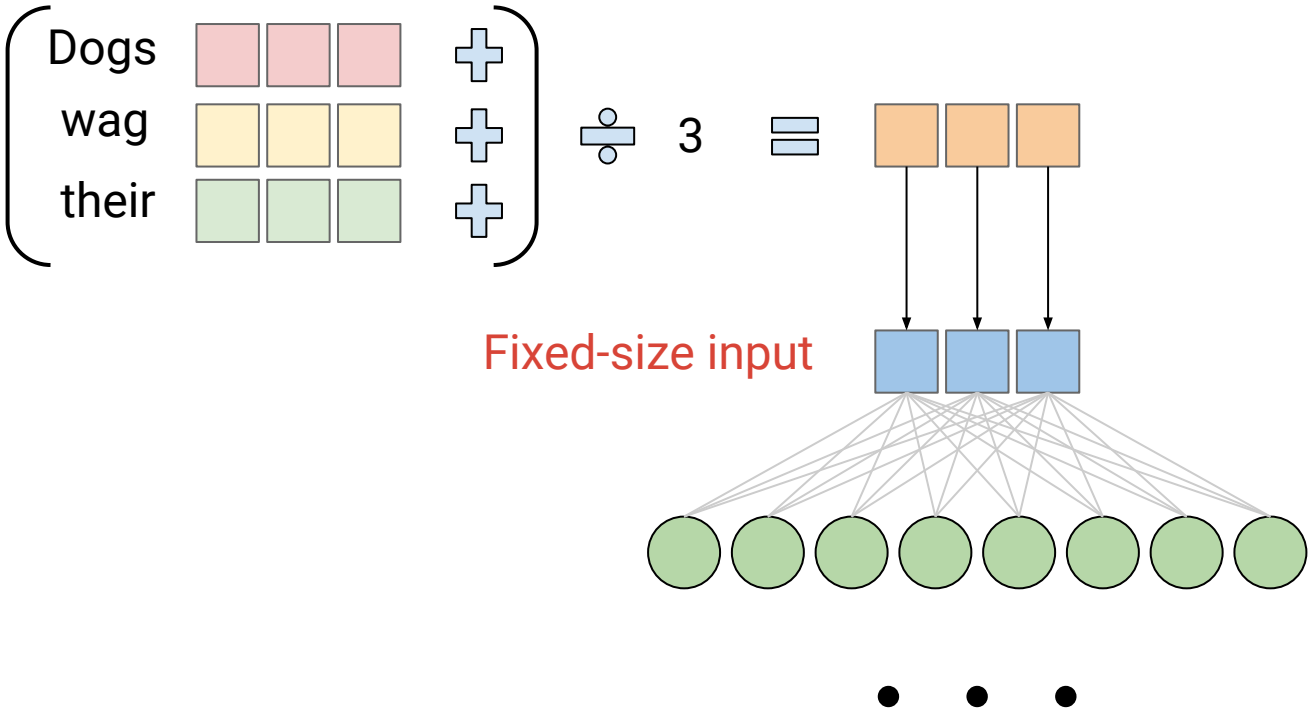
RNNs work differently than other models.

# Some models require cutting and padding

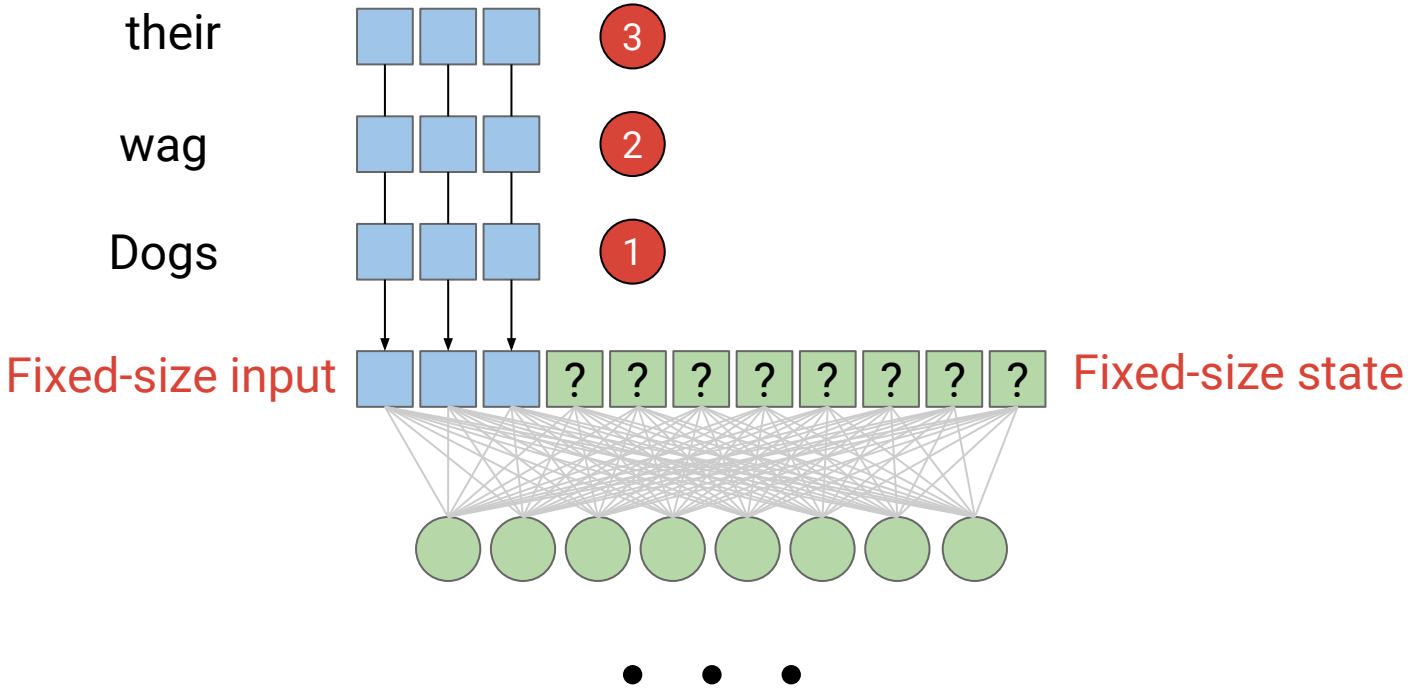
Fixed-size input



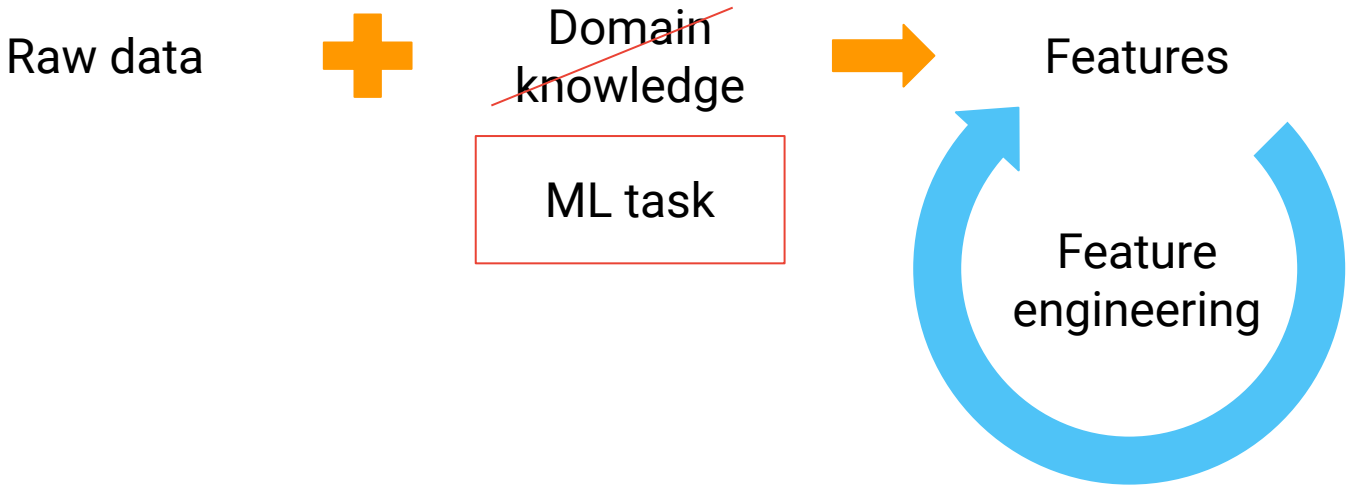
# Other models require bagging



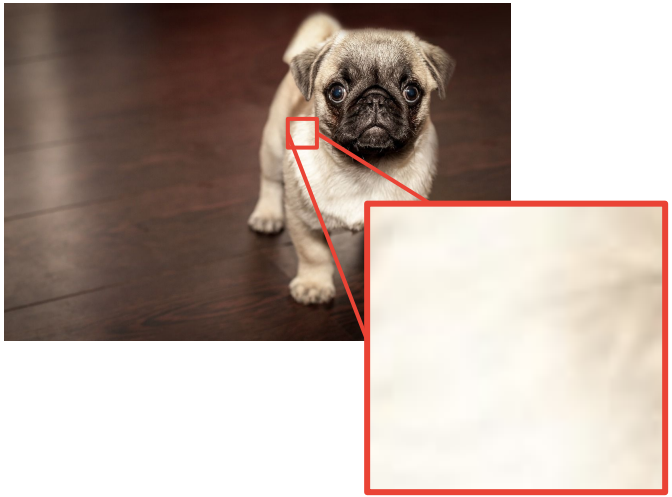
# How might we represent the past in a fixed-size state?



# Traditional ML would try to use feature engineering



# RNNs scan their input just like CNNs

# Two key ideas for RNNs

1. RNNs learn a compact hidden state that represents the past.

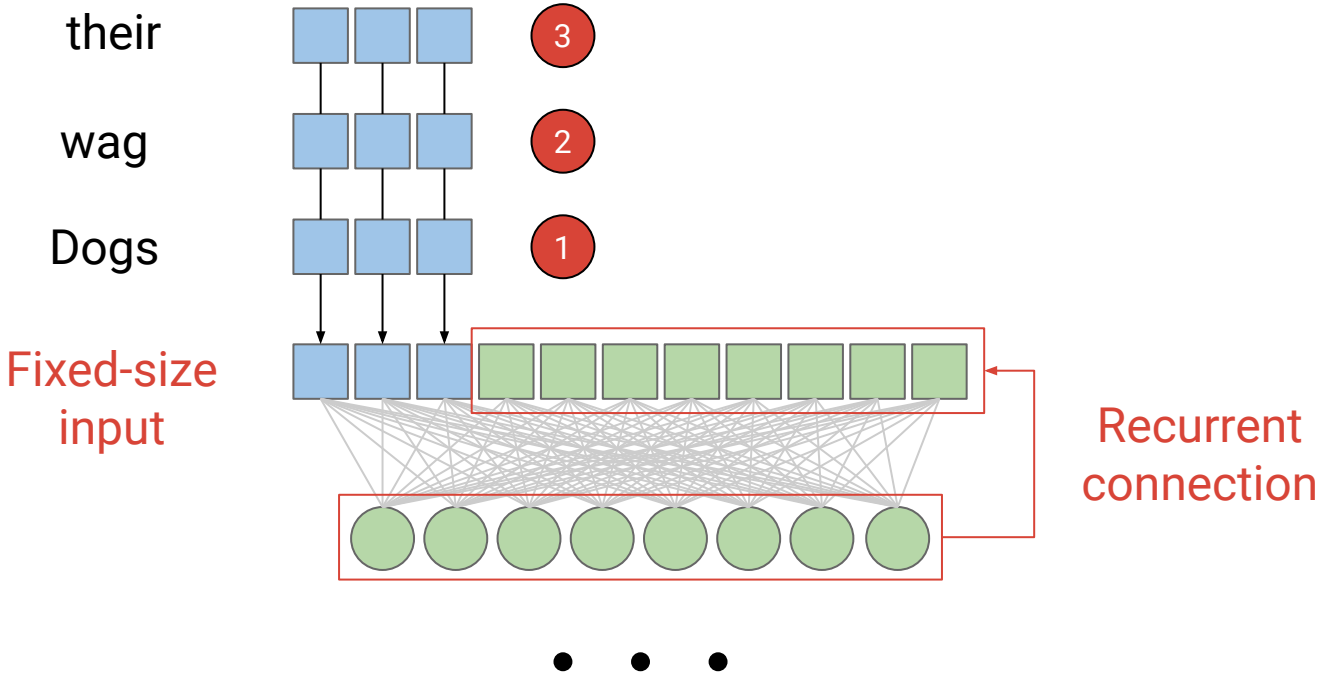
2. The input to an RNN is a concatenation of the original, stateless input and the hidden state.

# How RNNs create powerful representations of the past

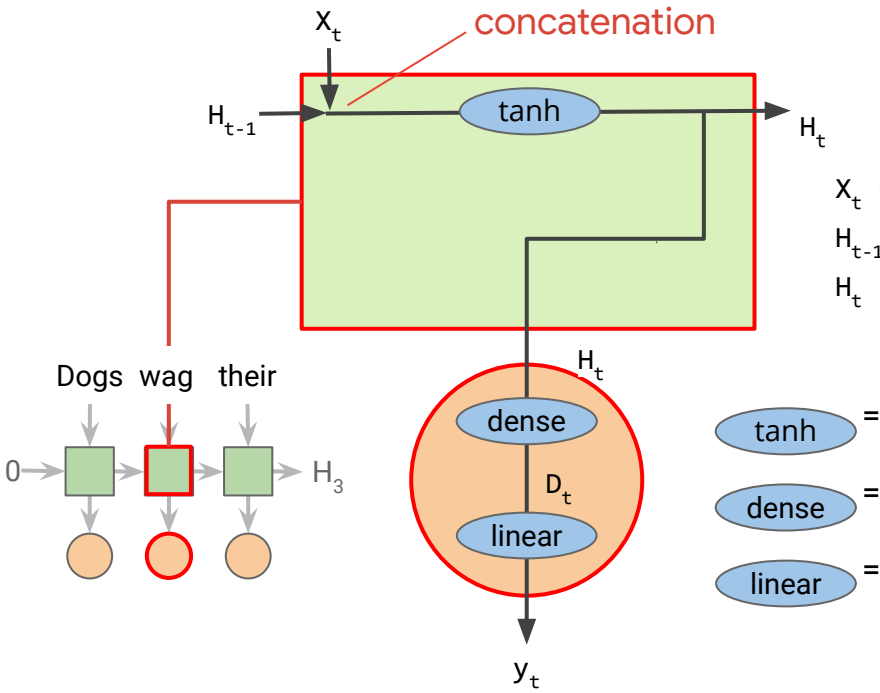
1 Recurrent connection

2 Clever optimization

# The recurrent connection



# The zoomed-out view of RNNs




$X_t$ = [3., 7., .8]

$H_{t-1}$ = [-.5, .2, -.1]

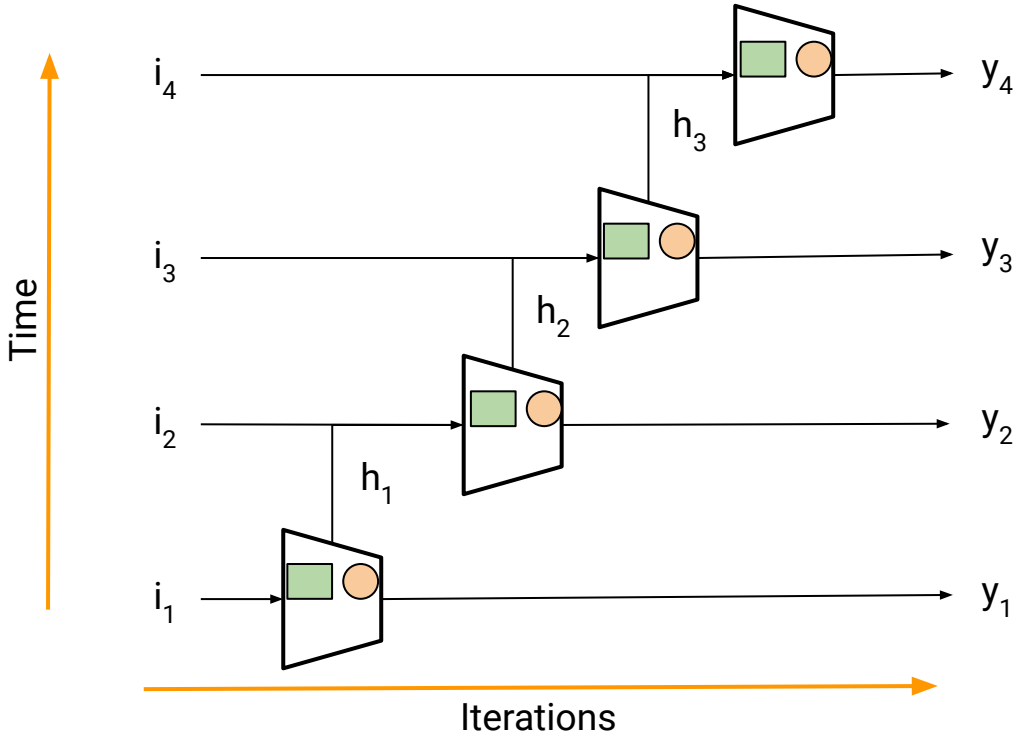
$H_t$ = tanh([3., 7., .8, -.5, .2, -.1].$W_H$ + $b_t$)

tanh = tanh([$X_t$|$H_{t-1}$].$W_H$ + $b_H$)

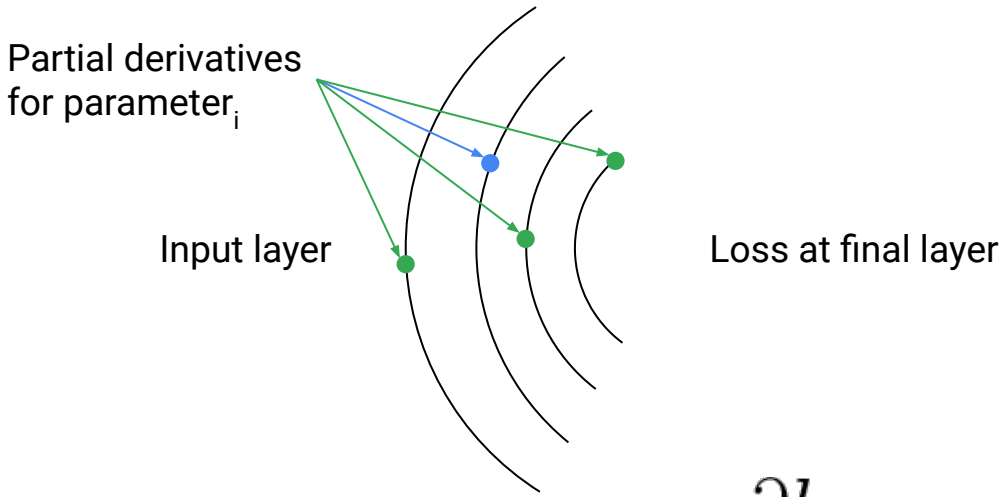
dense = relu($H_t$.$W_d$ + $b_d$)

linear = $D_t$.$W_y$ + $b_y$

# Forward propagation in RNNs



# Normally, we have 1 partial derivative per parameter, but RNNs have more than 1

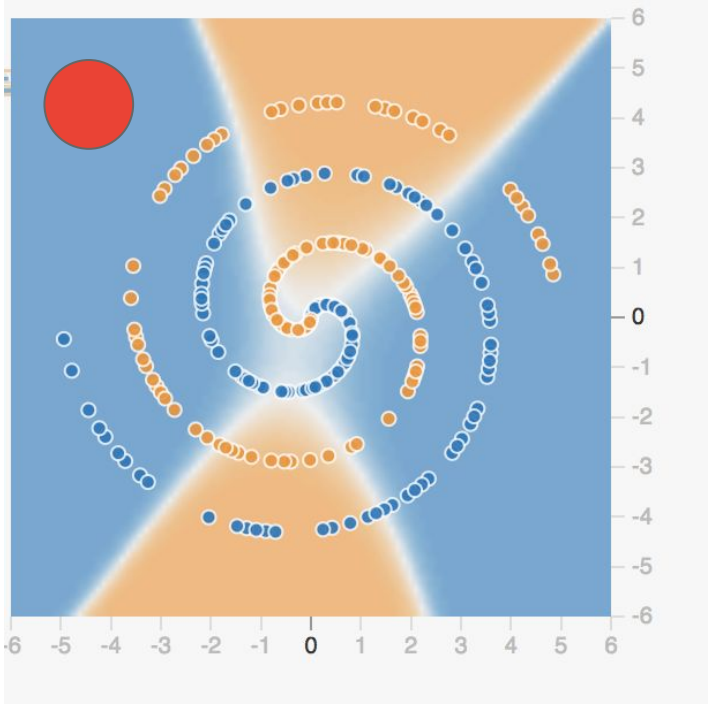



$$p^i_{t+1} = p^i_t + \alpha \times \frac{\partial loss}{\partial p^i}$$

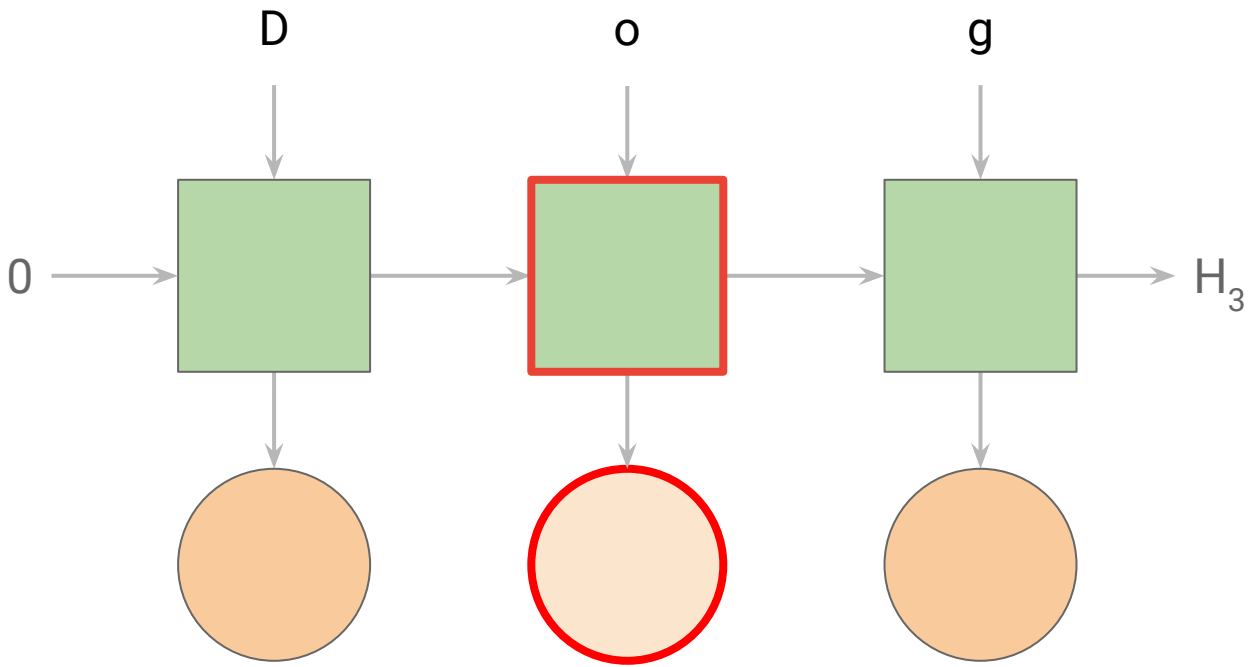
# Backpropagation through time

$$p_{t+1}^{i} = p_{t}^{i} + \alpha \times \frac{\sum_{x=1}^{seq\_len} \frac{\partial loss}{\partial p_{t}^{i,x}}}{seq\_len}$$

# Random inputs often don’t reveal much



# Character RNNs operate on individual characters



# Rules within our first domain: Shakespeare

1. Language-related rules,
e.g. subject - verb agreement.

2. Drama-related rules,
e.g. All plays have titles.

# Some Shakespeare-generating pseudocode

```
generate_title()
for act in acts:
    for scene in act:
        generate_scene(act, scene)
```

# Generating a scene

```
def generate_scene(act, scene):
    num_lines = rand(1,100)
    for line in range(num_lines):
        speak(random.choice(scene.charac
ters))
```

# Bringing our characters to life!

```
def speak(character):
     start_word = random.choice(self.vocabulary)
     print(markov_model.generate(start_word, 10))
```

# Control structures could fit inside list

```
Fixed size state = [...,my_counter = 2,...]
```

# Sets are harder to fit into a finite list

```
Fixed size state = [..., elt_1, elt_2, ...]
```

Remembering which characters have died already so we don’t inadvertently add ghosts to our play!

# A sample ML-generated play

```
        TITUS ANDRONICUS

ACT I

SCENE III An ante-chamber. The COUNT's palace.

 [Enter CLEOMENES, with the Lord SAY]

Chamberlain     Let me see your worshing in my hands.

LUCETTA         I am a sign of me, and sorrow sounds
it.
```

+

ML

# Some sample TensorFlow code

```
# Copyright 2015 The TensorFlow Authors. All Rights Reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in [0.1, 2.0, 3.0]]

def __init__(self, expected):
  return np.array([[0, 0, 0], [0, 0, 0]])
  self.assertAllEqual(tf.placeholder(tf.float32, shape=(3, 3)),
     (shape, prior.pack(), tf.float32))
  for keys in tensor_list:
    return np.array([[0, 0, 0]]).astype(np.float32)

  # Check that we have both scalar tensor for being invalid to a vector of 1 indicating
  # the total loss of the same shape as the shape of the tensor.
  sharded_weights = [[0.0, 1.0]]
  # Create the string op to apply gradient terms that also batch.
  # The original any operation as a code when we should alw infer to the session case.

[Title: Some sample TensorFlow code]
```

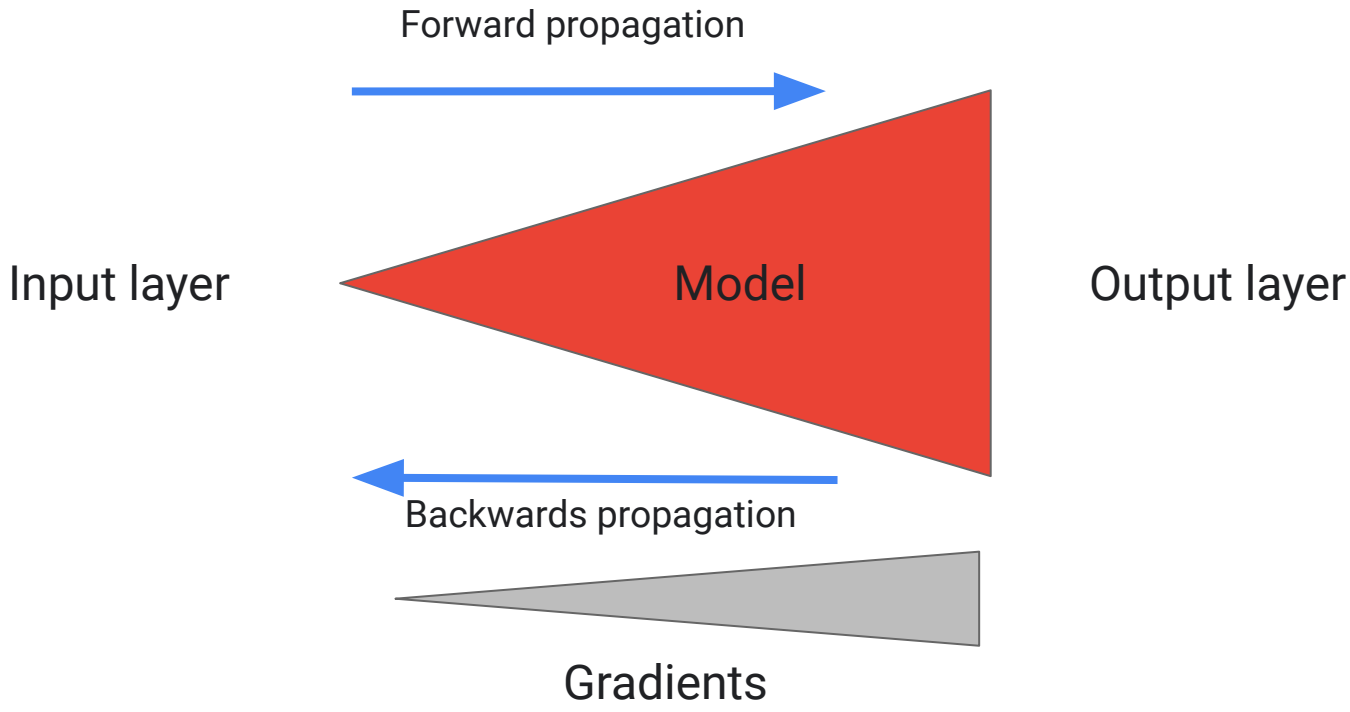
# RNN caveats

# RNN caveats

*"Michel C. was born in Paris, France. He is married and has three children. He received a M.S. in neurosciences from the University Pierre & Marie Curie and the Ecole Normale Supérieure in 1987, and and then spent most of his career in Switzerland, at the Ecole Polytechnique de Lausanne. He specialized in child and adolescent psychiatry and his first field of research was severe mood disorders in adolescent, topic of his PhD in neurosciences (2002). His mother tongue is ? ? ? ?"*

Short context

# Vanishing gradient strikes again



Our usual techniques are not enough.

cloud.google.com