Working with Sequences

Advanced ML with TensorFlow on GCP

End-to-End Lab on Structured Data ML

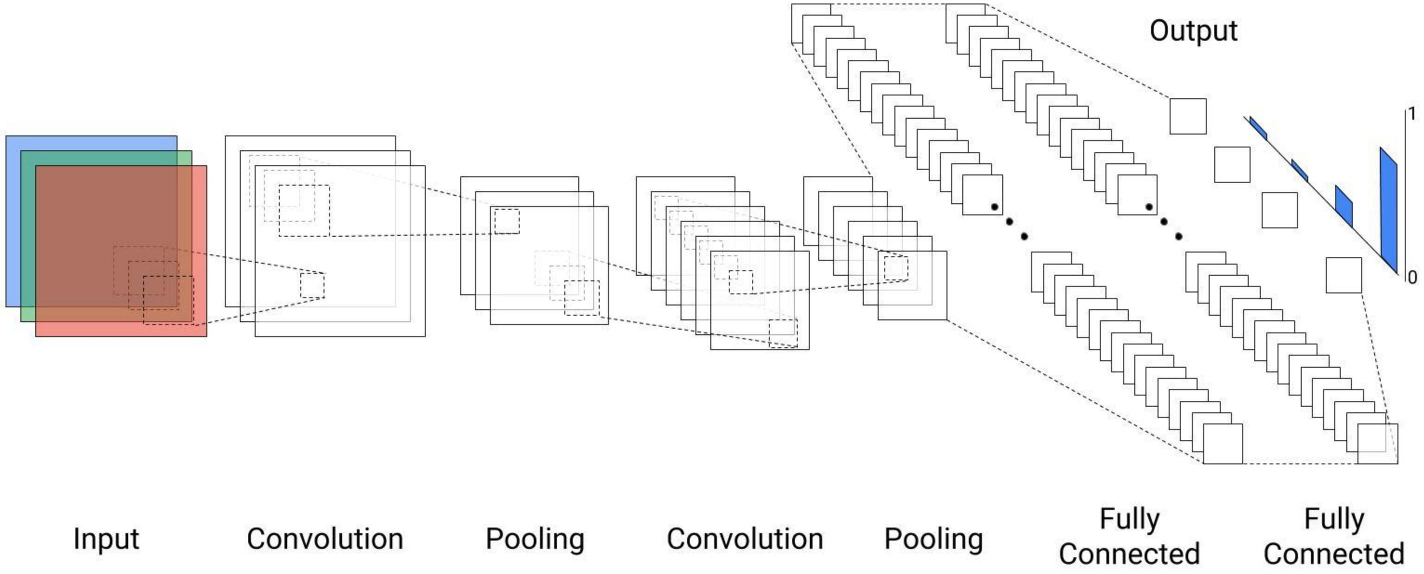
Production ML Systems

Image Classification Models

Sequence Models

Recommendation Systems

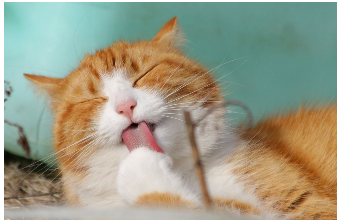
Recall: CNNs as feature extractors

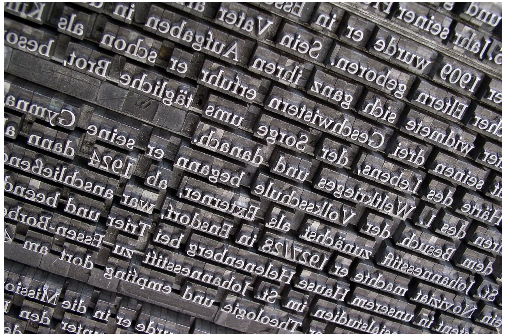


Sequences are another common and important domain

The cat ate the mouse

Le chat a mangé la souris ∅





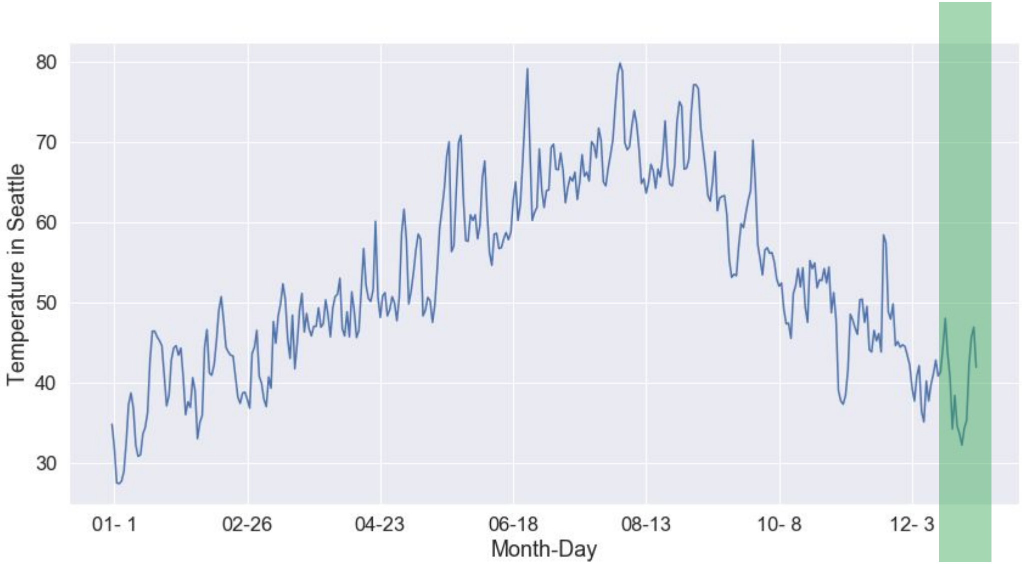
Learn how to...

Define what sequence data is

Prepare sequence data for modeling

Apply classical approaches to sequence modeling

What is a sequence?



Temperature (F)	...	Pressure (mBar)	Order
45	...	1105	1
62	...	976	2

Predicting coin flips

Value	Time
heads	9:01:05am
tails	9:01:18am
heads	9:01:24am
...	...

Is this sequence data?

A. Yes

B. No

Predicting coin flips

Value	Time
heads	9:01:05am
tails	9:01:18am
heads	9:01:24am
...	...

Is this sequence data?

A. Yes

B. No

Coin flips

t1 t2 t3 t4

What about natural language?

Is this sequence data?

A. Yes

B. No

Word	Order
The	1
language	2
spoken	3
in	4
France	5
is	6

What about natural language?

Is this sequence data?

A. Yes

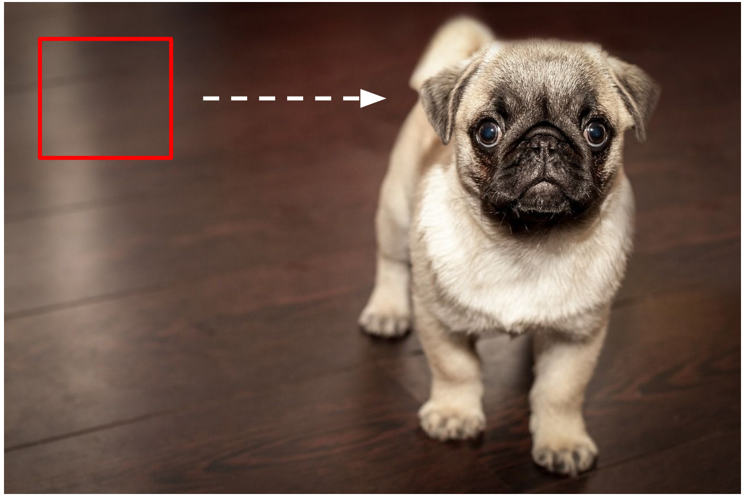
B. No

Word	Order
The	1
language	2
spoken	3
in	4
France	5
is	6

Can we treat image data as a sequence?

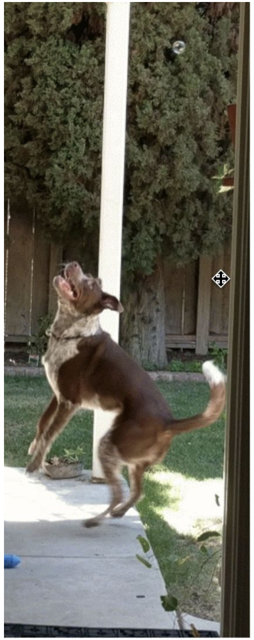
Pixel RGB value	Pixel position
[255,255,0]	[0,0]
[255,0,255]	[0,1]
...	...

Images and traditional sequences are still similar in one respect

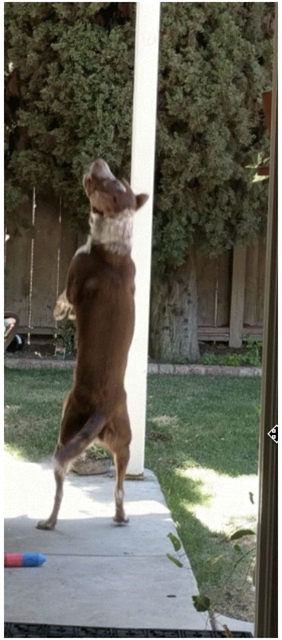


Movies are sequences

t1

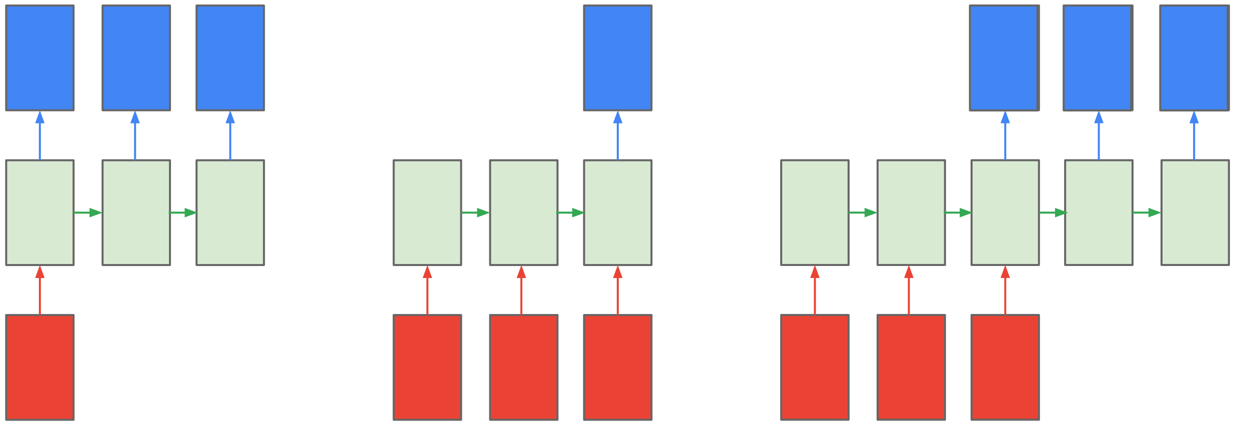


t2



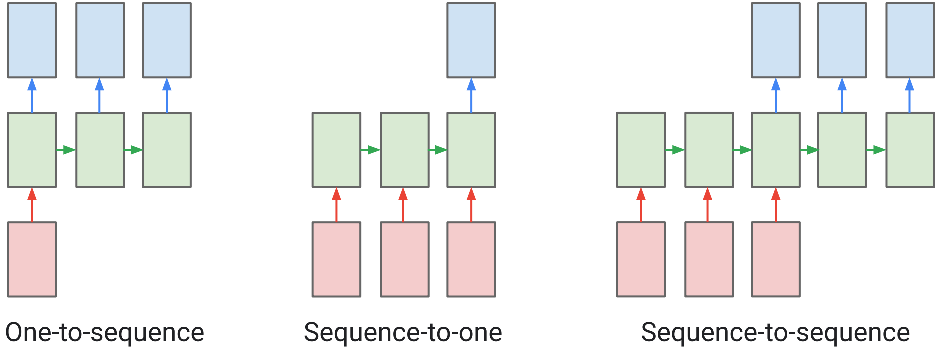
t3

Types of sequence models

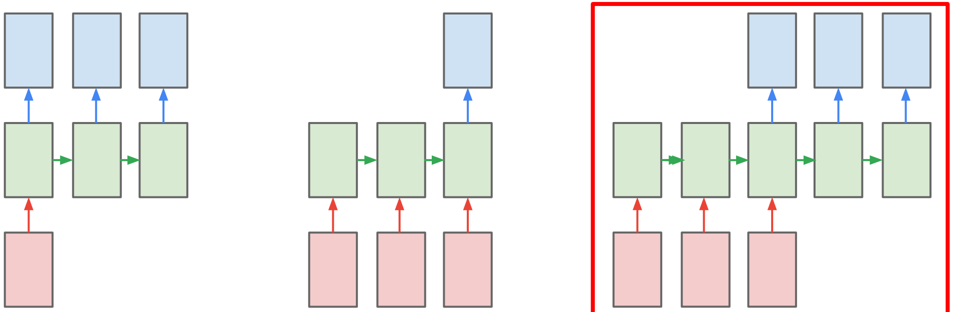


One-to-sequence
Sequence-to-one
Sequence-to-sequence

What sort of problem is translation?



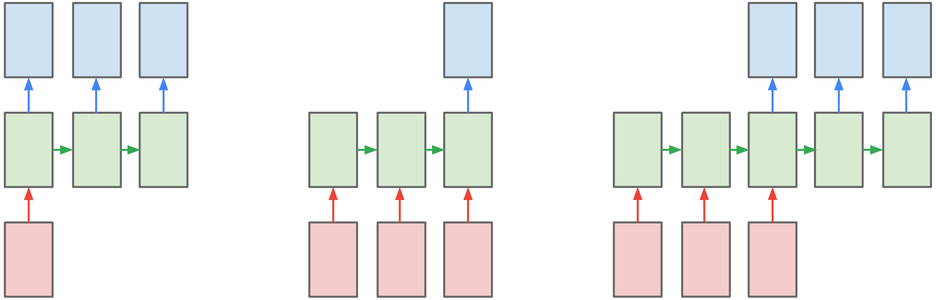
What sort of problem is translation?



One-to-sequence Sequence-to-one Sequence-to-sequence

What sort of problem is image captioning?

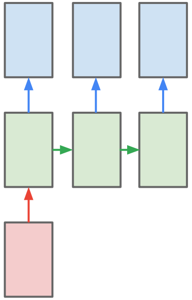
Two hockey players are fighting over the puck.



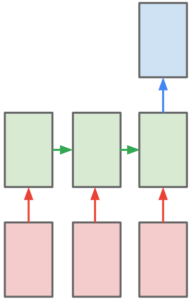
One-to-sequence Sequence-to-one Sequence-to-sequence

What sort of problem is image captioning?

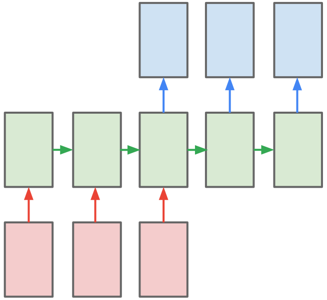
Two hockey players are fighting over the puck.



One-to-sequence

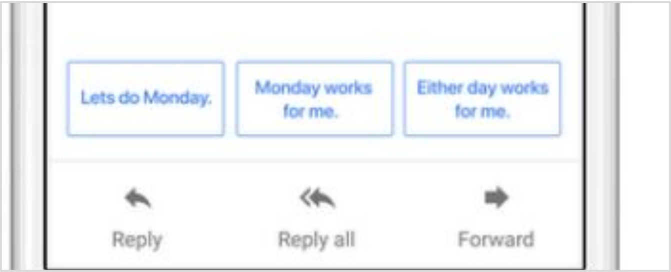


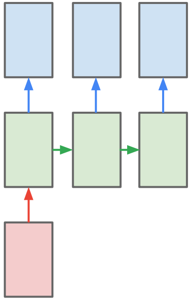
Sequence-to-one



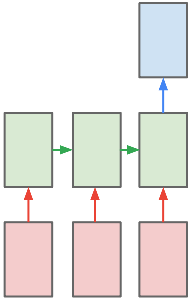
Sequence-to-sequence

What sort of problem is Smart Reply?

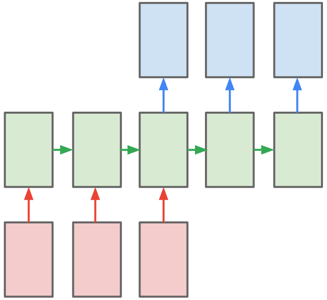




One-to-sequence

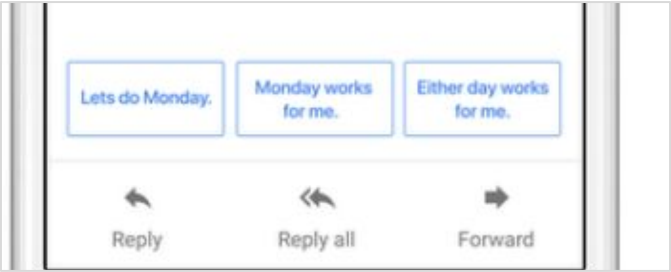


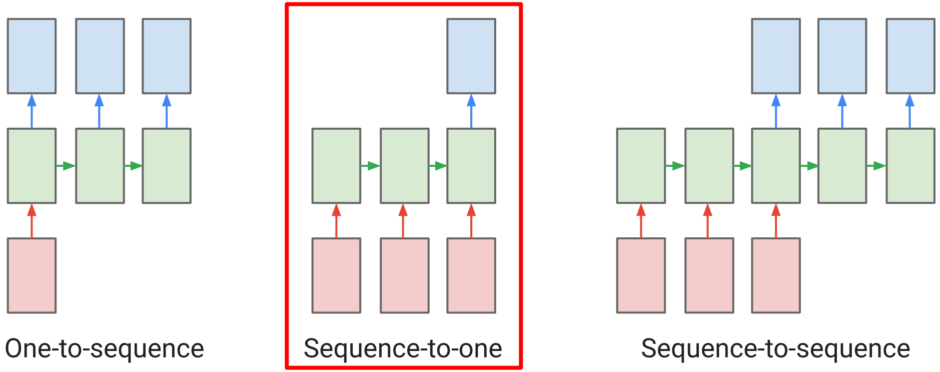
Sequence-to-one



Sequence-to-sequence

What sort of problem is Smart Reply?



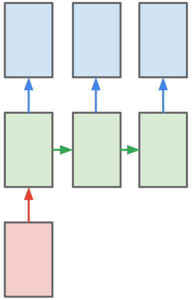


What about "Predict the next X" models?

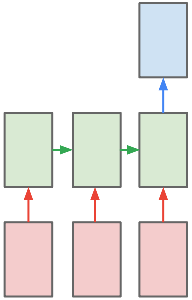
Recommended videos to watch:

Migrating your data warehouse to Google BigQuery: Lessons
Google Cloud Platform
5.6K views

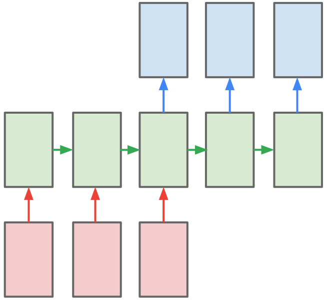
Adding Machine Learning to your applications
Google Cloud Platform
21K views



One-to-sequence



Sequence-to-one



Sequence-to-sequence

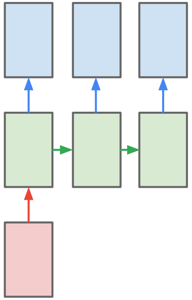
What about “Predict the next X” models?

Recommended videos to watch:

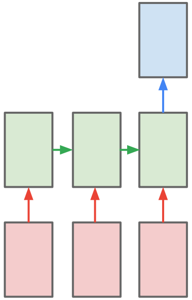
Migrating your data warehouse to Google BigQuery: Lessons
Google Cloud Platform
5.6K views

Adding Machine Learning to your applications
Google Cloud Platform
21K views

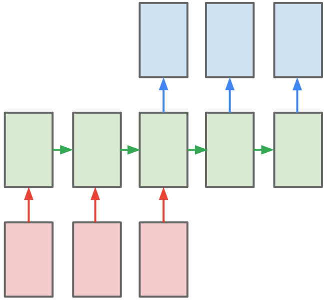
Many possible models!



One-to-sequence



Sequence-to-one

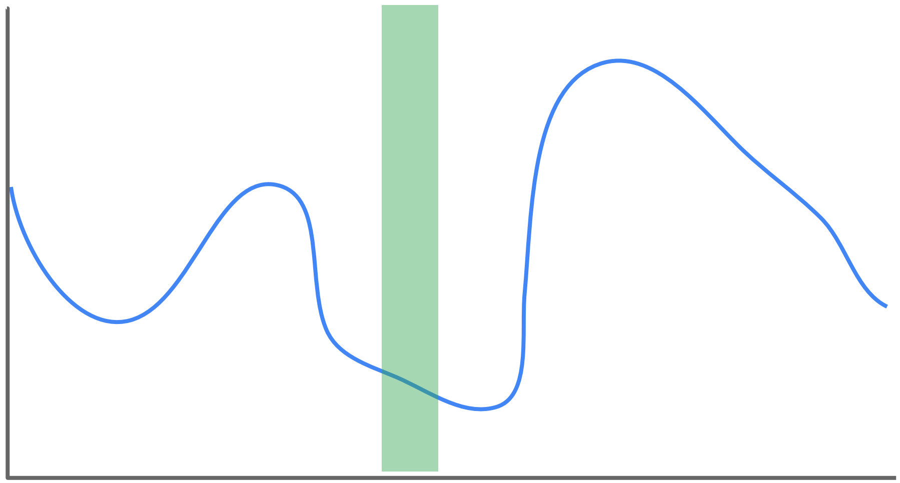


Sequence-to-sequence

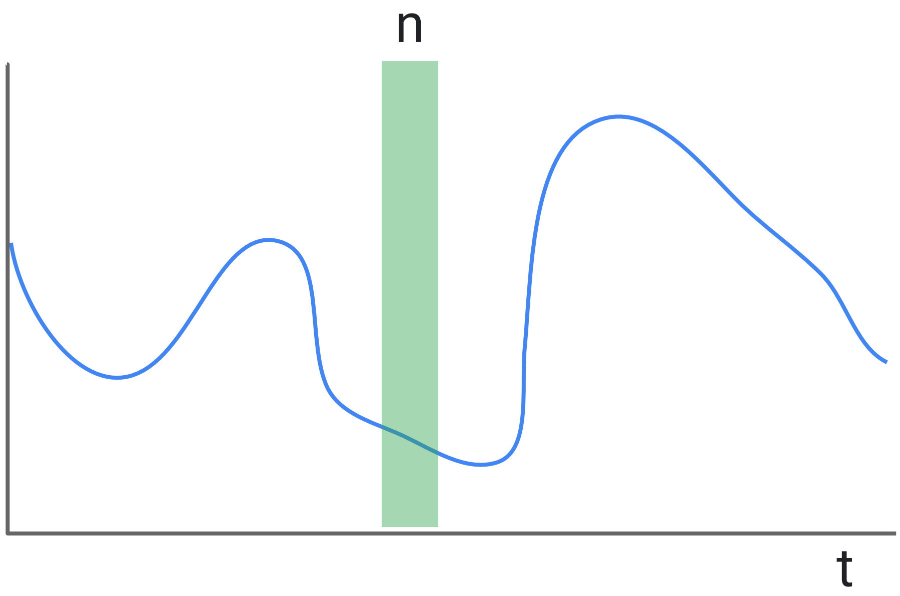
Observing events as a sequence

Temperature (F)	...	Pressure (mBar)	Ordering Feature
45	...	1105	1
62	...	976	2

Observing events as a sequence

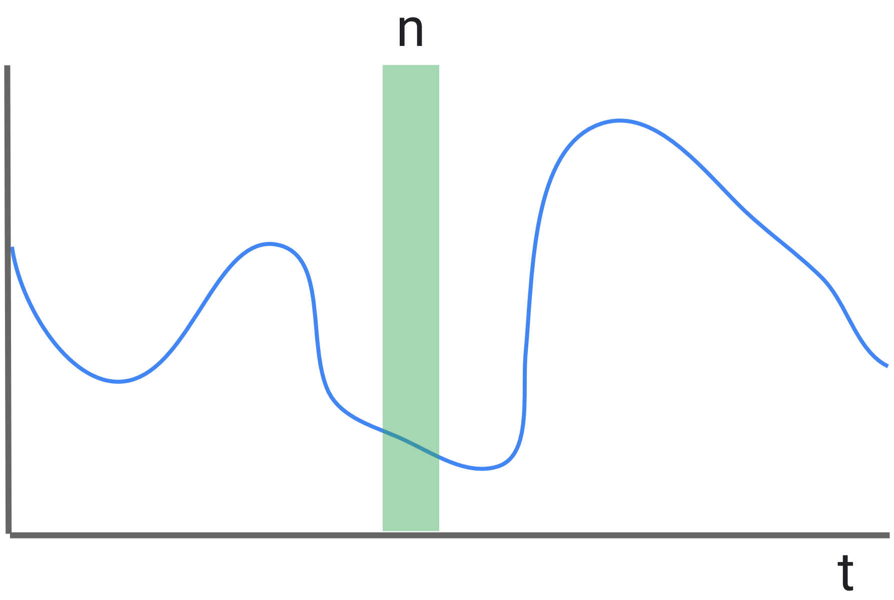


Quiz: How many rows will there be in our training set?



A. t
B. $\max(0, t - n + 1)$
C. $t - n$
D. $\max(0, t + n)$

Quiz: How many rows will there be in our training set?

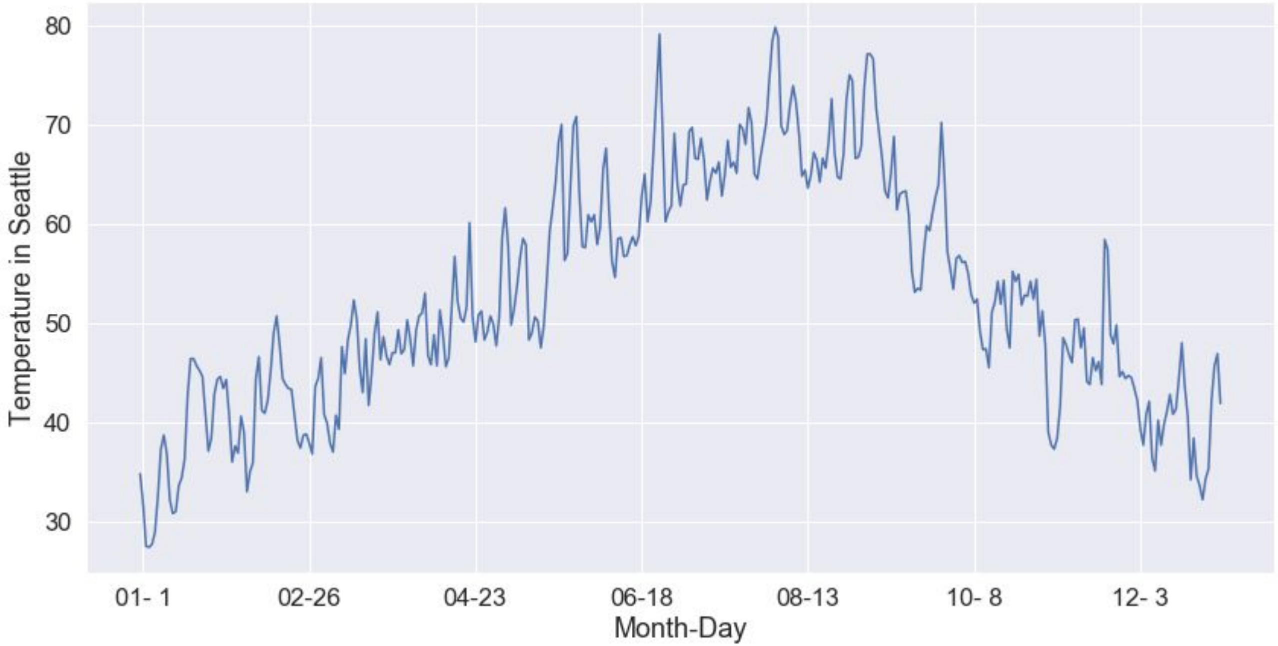


A. t
B. max(0, t - n + 1)
C. t - n
D. max(0, t + n)

Quiz: How many rows will there be in our training set?

Price 10 minutes prior	...	Price 1 minute prior	Price
t_0	...	t_9	t_{10}
t_1	...	t_{10}	t_{11}
t_2	...	t_{11}	t_{12}
...	...	...	...
$t_{seqlen-10}$	...	$t_{seqlen-1}$	t_{seqlen}

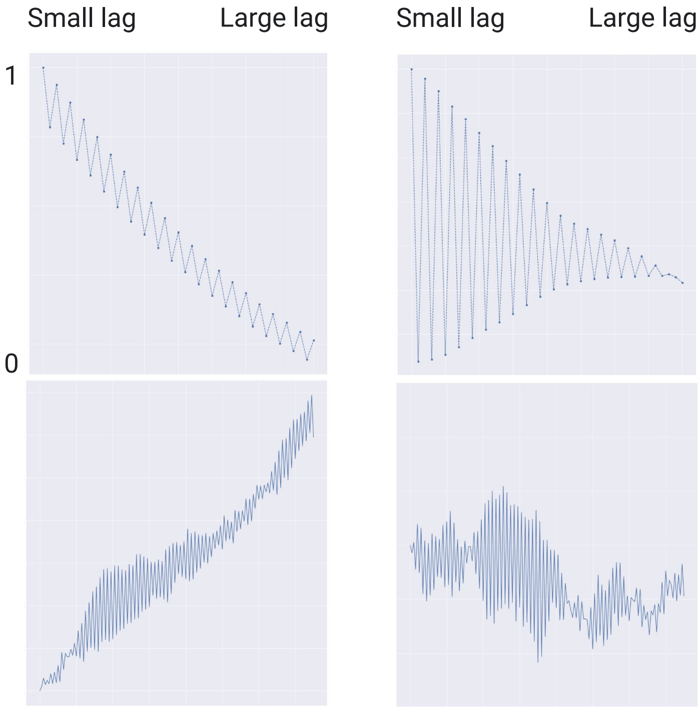
What’s a good size for our sliding window?



Autocorrelation graphs can reveal dependencies

Correlation coefficient versus Time lag

Raw sequence data versus Time



Quiz: Sometimes dependencies are known

Assuming that:

1. You knew the data were periodic.
2. Rotation speed and water pressure were constant.
3. The sprinkler completed 1 rotation/minute.

What would be a good choice for sliding window length?

A. 1 minute

B. 30 seconds

C. 2 minutes

D. As long as possible

Quiz: Sometimes dependencies are known

Assuming that:

1. You knew the data were periodic.
2. Rotation speed and water pressure were constant.
3. The sprinkler completed 1 rotation/minute.

What would be a good choice for sliding window length?

A. 1 minute

B. 30 seconds

C. 2 minutes

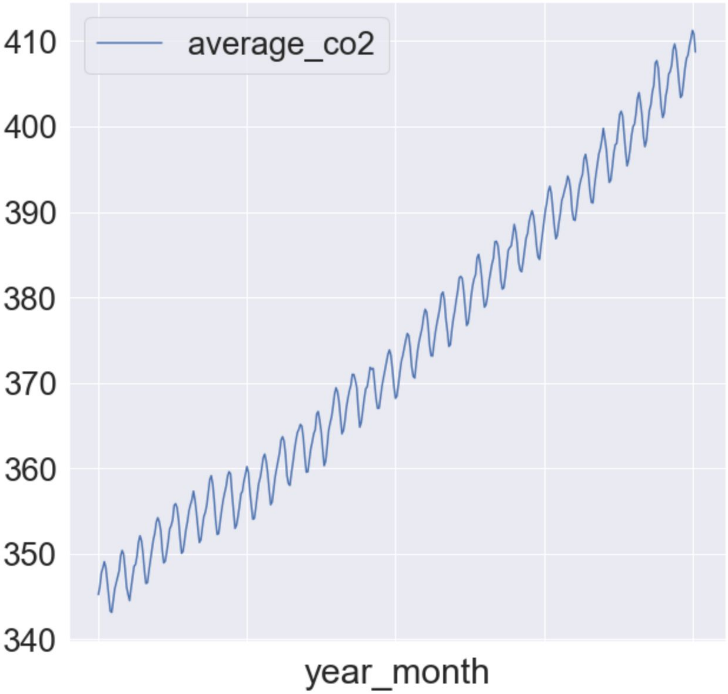
D. As long as possible

First try a linear model

Generating synthetic sequence data

```
def create_time_series():
  freq = (np.random.random()*0.5) + 0.1  # 0.1 to 0.6
  ampl = np.random.random() + 0.5  # 0.5 to 1.5
  # -0.3 to +0.3 uniformly distributed
  noise = [np.random.random()*0.3 for i in xrange(SEQ_LEN)]
  x = np.sin(np.arange(0,SEQ_LEN) * freq) * ampl + noise
  return x
```

Real-world sequences are more complicated



Each file consists of a comma-delimited string

```
def to_csv(filename, N):
  with open(filename, 'w') as ofp:
    for lineno in xrange(0, N):
      seq = create_time_series()
      line = ",".join(map(str, seq))
      ofp.write(line + '\n')
```

```
train.csv
0.13,0.52,0.76,0.71,0.63,0.29,0.15,-0.23,-0.34,-0.38
0.27,0.32,0.63,0.63,0.8,0.75,0.63,0.23,0.18,-0.23
0.3,0.49,0.87,1.08,1.17,1.25,1.46,1.52,1.65,1.46
```

Creating an input function

```
def read_dataset(filename, mode, batch_size=512):
    def _input_fn():
        def decode_csv(row):
            # row is a string tensor containing the contents of one row
            features = tf.decode_csv(row, record_defaults=DEFAULTS)
            # string tensor -> list of 50 rank 0 float tensors
            label = features.pop()  # remove last feature and use as label
            features = tf.stack(features)  # list of rank 0 tensors -> single rank 1 tensor
            return {TIMESERIES_COL: features}, label

        # Create list of file names that match "glob" pattern (i.e. data_file_*.csv)
        dataset = tf.data.Dataset.list_files(filename)
        # Read in data from files
        dataset = dataset.flat_map(tf.data.TextLineDataset)
        # Parse text lines as comma-separated values (CSV)
        dataset = dataset.map(decode_csv)
        …
    return _input_fn
```

Compare model performance

```
eval_metric_ops = {
      "RMSE": rmse,
      "RMSE_same_as_last": same_as_last_benchmark(features, labels),
  }

# RMSE when predicting same as last value
def same_as_last_benchmark(features, labels):
    predictions = features[TIMESERIES_COL][:,-1] # last value in input sequence
    return tf.metrics.root_mean_squared_error(labels, predictions)
```

Defining a linear model for time series prediction

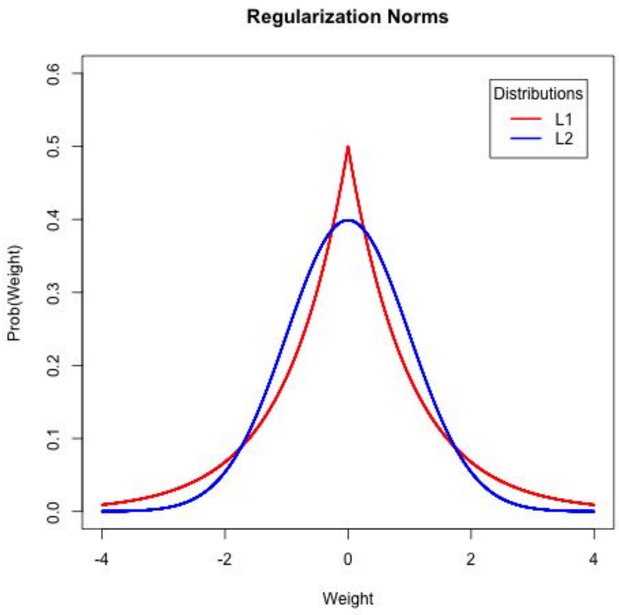
```
def linear_model(features, mode, params):
   X = features[TIMESERIES_COL]
   predictions = tf.layers.dense(X, 1, activation=None)
   return predictions
```

Lab

Using linear models for sequences

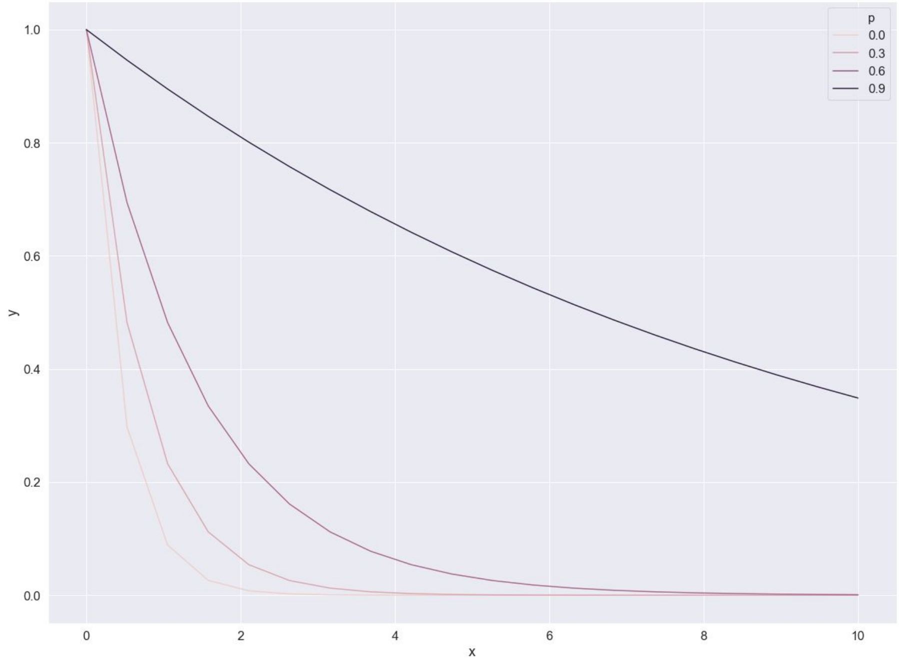
In this lab you will be creating a linear model by completing the TODO steps in the model.py in your lab notebook.

Sometimes, it’s important that models capture aspects of the real world



Regularization can constrain for sparsity.

Constrain the weights for better performance



Autoregressive ARMA models use moving averages

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-1} + \varepsilon_t$$

Defining a DNN model for time series prediction

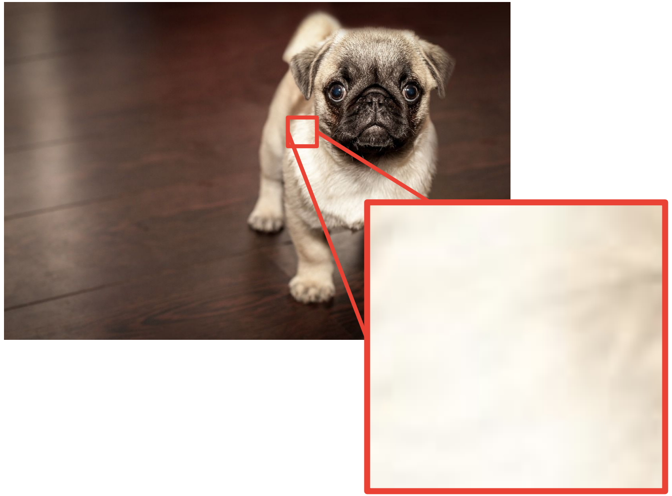
```
def dnn_model(features, mode, params):
 X = features[TIMESERIES_COL]
 h1 = tf.layers.dense(X, 10, activation=tf.nn.relu)
 h2 = tf.layers.dense(h1, 3, activation=tf.nn.relu)
 predictions = tf.layers.dense(h2, 1, activation=None)
 return predictions
```

Lab

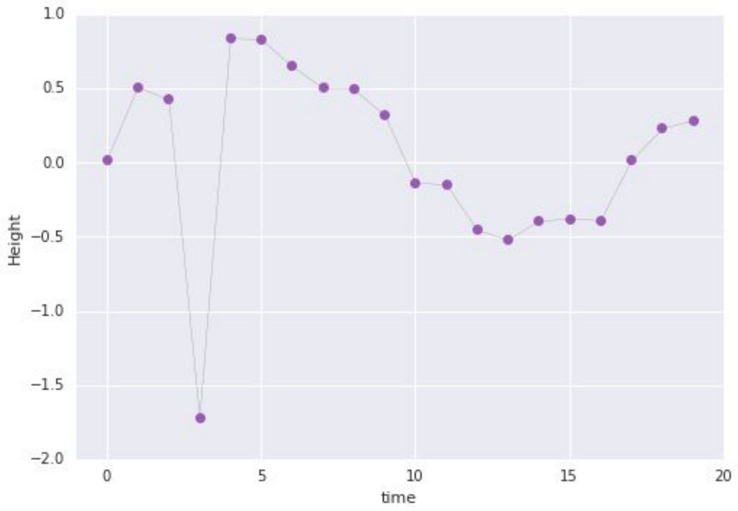
Using DNNs for sequences

In this lab you will be using TensorFlow Hub to create re-usable embeddings.

Locality is important for images and sequences



Quiz: Which convolution filters would be highly active at t = 4?



A. `[.5, .5, .5, .5]`

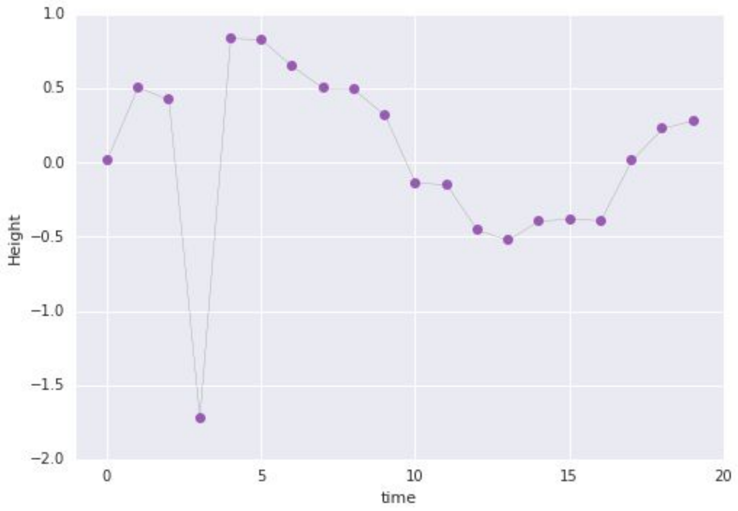
B. `[.33, -.67, .33]`

C. `[1.2, 1.0, .75, 1.0]`

D. `[1.2, 1.0, -.75, 1.0]`

Choose all that could apply.

Quiz: Which convolution filters would be highly active at t = 4?



A. [.5, .5, .5, .5]

B. [.33, -.67, .33]

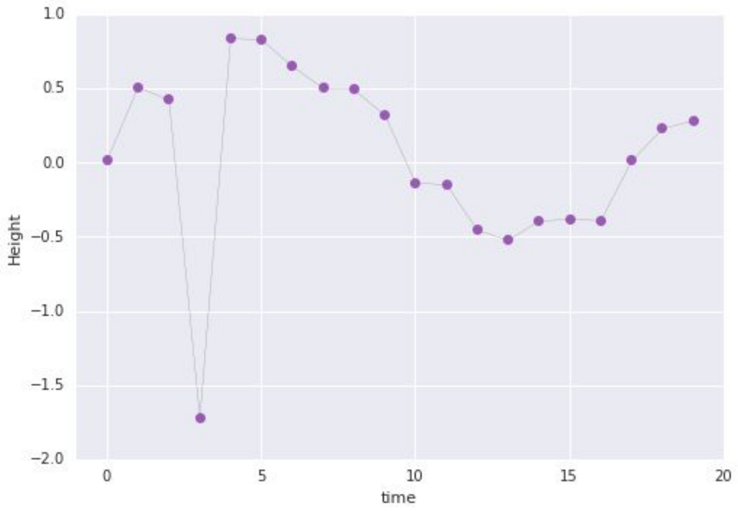
C. [1.2, 1.0, .75, 1.0]

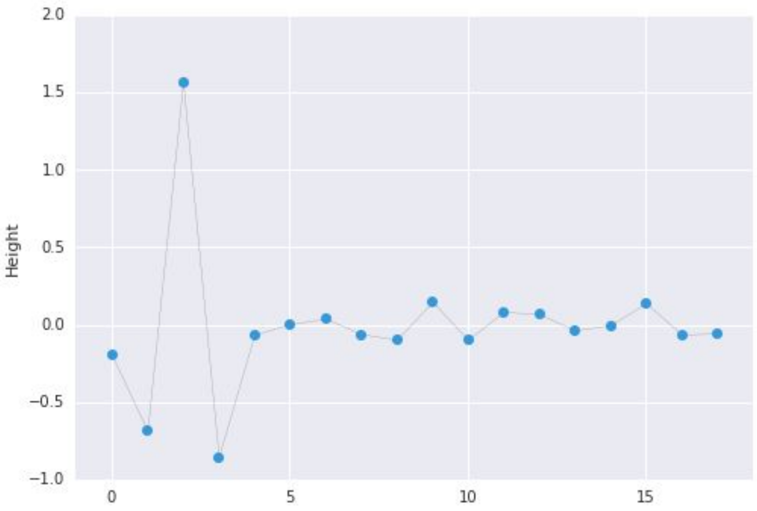
D. [1.2, 1.0, -.75, 1.0]

Choose all that could apply.

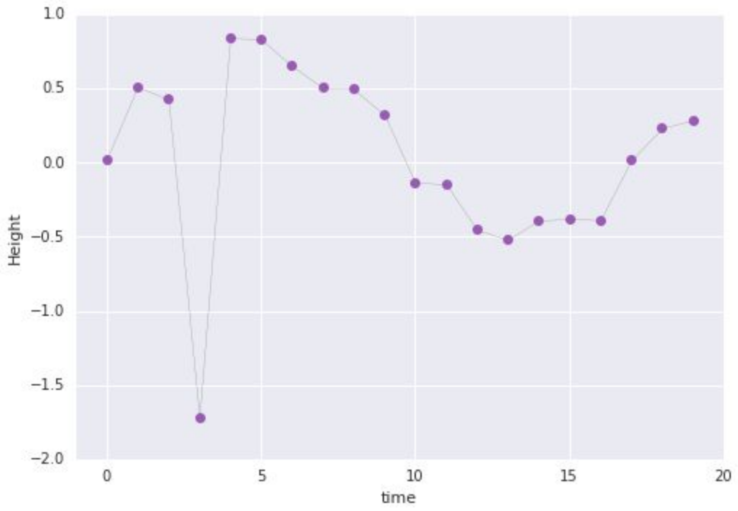
Convolutional filters are pattern matchers

After convolving `[.33, -.67, .33]`





Quiz: Of those that were active, which is more specific?



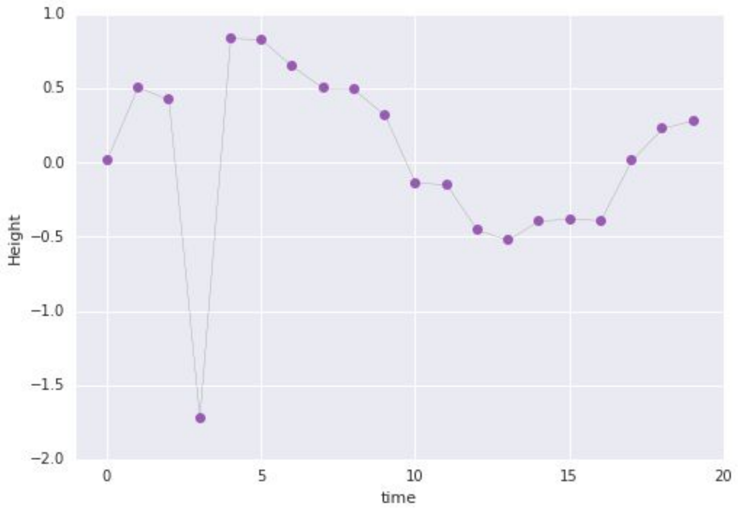
A. [.5, .5, .5, .5]

B. [.33, -.67, .33]

C. [1.2, 1.0, .75, 1.0]

D. [1.2, 1.0, -.75, 1.0]

Quiz: Of those that were active, which is more specific?



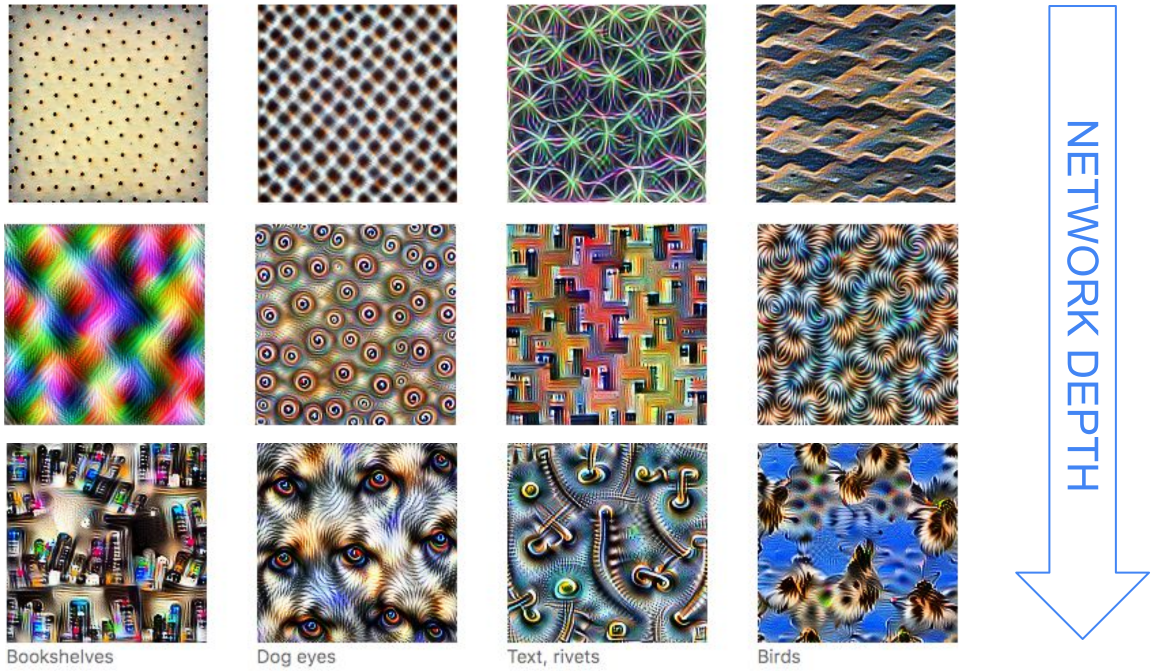
A. [.5, .5, .5, .5]

B. [.33, -.67, .33]

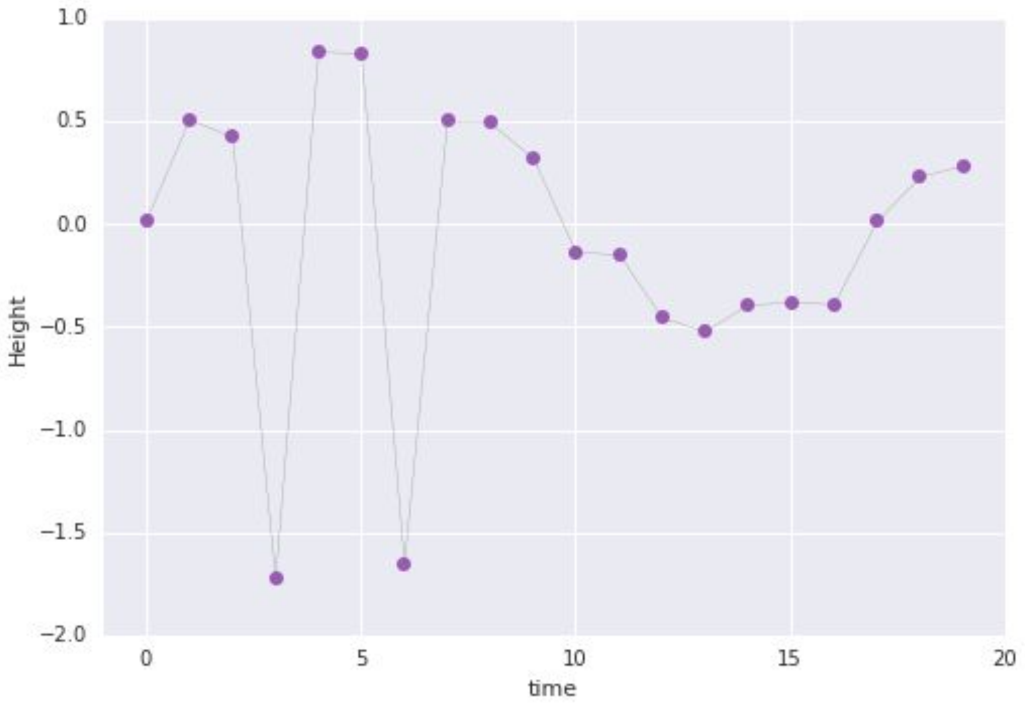
C. [1.2, 1.0, .75, 1.0]

D. [1.2, 1.0, -.75, 1.0]

CNNs learn a hierarchy of features



Detecting two quick drops



Steps in applying a convolution

1. Flatten the input sequence.
2. Use conv1d to apply a number of filters to the sequence.
3. Use max_pooling1d to add some spatial invariance and downscaling.
4. Flatten the resulting output into a sequence.
5. Send it through a fully connected layer with the appropriate output node.

Applying convolutions in TensorFlow

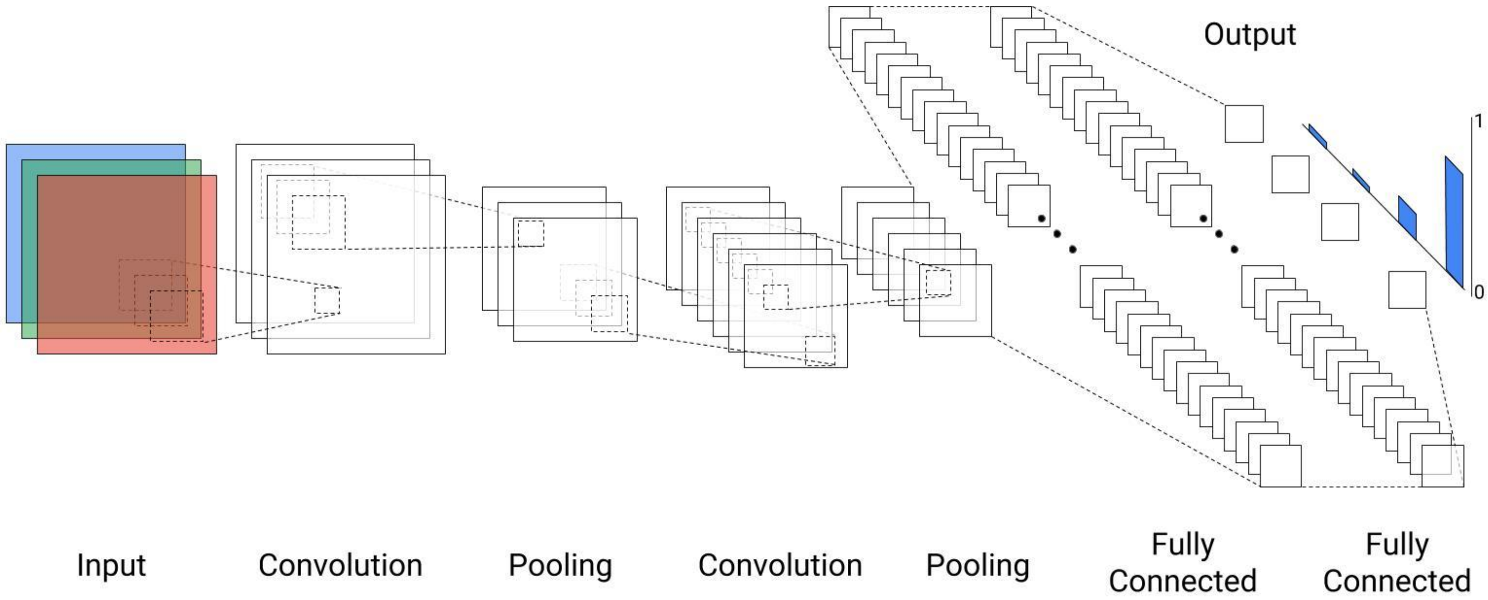
```
def cnn_model(features, mode, params):
  X = tf.reshape(features[TIMESERIES_COL], [-1, N_INPUTS, 1]) # ?x10x1
  c1 = tf.layers.max_pooling1d(
         tf.layers.conv1d(X, filters=N_INPUTS//2,
                          kernel_size=3, strides=1, # ?x10x5
                          padding='same', activation=tf.nn.relu),
         pool_size=2, strides=2
       ) # ?x5x5
  outlen = (N_INPUTS//2) * (N_INPUTS//2)
  c1flat = tf.reshape(c1, [-1, outlen])
  h1 = tf.layers.dense(c1flat, 3, activation=tf.nn.relu)
  return tf.layers.dense(h1, 1, activation=None)
```

Lab

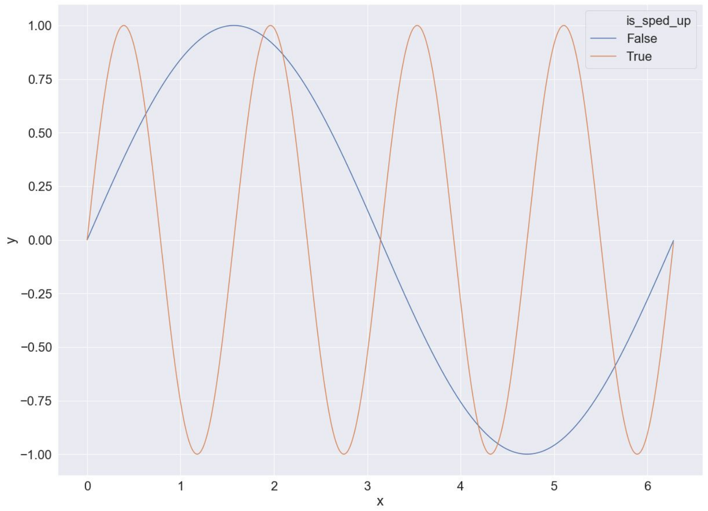
Using CNNs for sequences

In this lab you will complete TODOs in model.py for a CNN model.

Why didn’t our CNN do much better than our DNN?



The limitations of CNNs



Handling variable length inputs and outputs

Handling variable length inputs and outputs

1 Cutting and padding

2 Bagging

Example: Predicting oil prices

Hour	Price
1	2
2	4
3	6
4	8
5	10
...	...

4 Hours Prior	3 Hours Prior	2 Hours Prior	1 Hour Prior	Label
2	4	6	8	10
...	...	...	...	...

$$w = \left[\frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}\right]$$

How do we handle a shorter input?

4 Hours Prior	3 Hours Prior	2 Hours Prior	1 Hour Prior
3	6	9	12

Padding

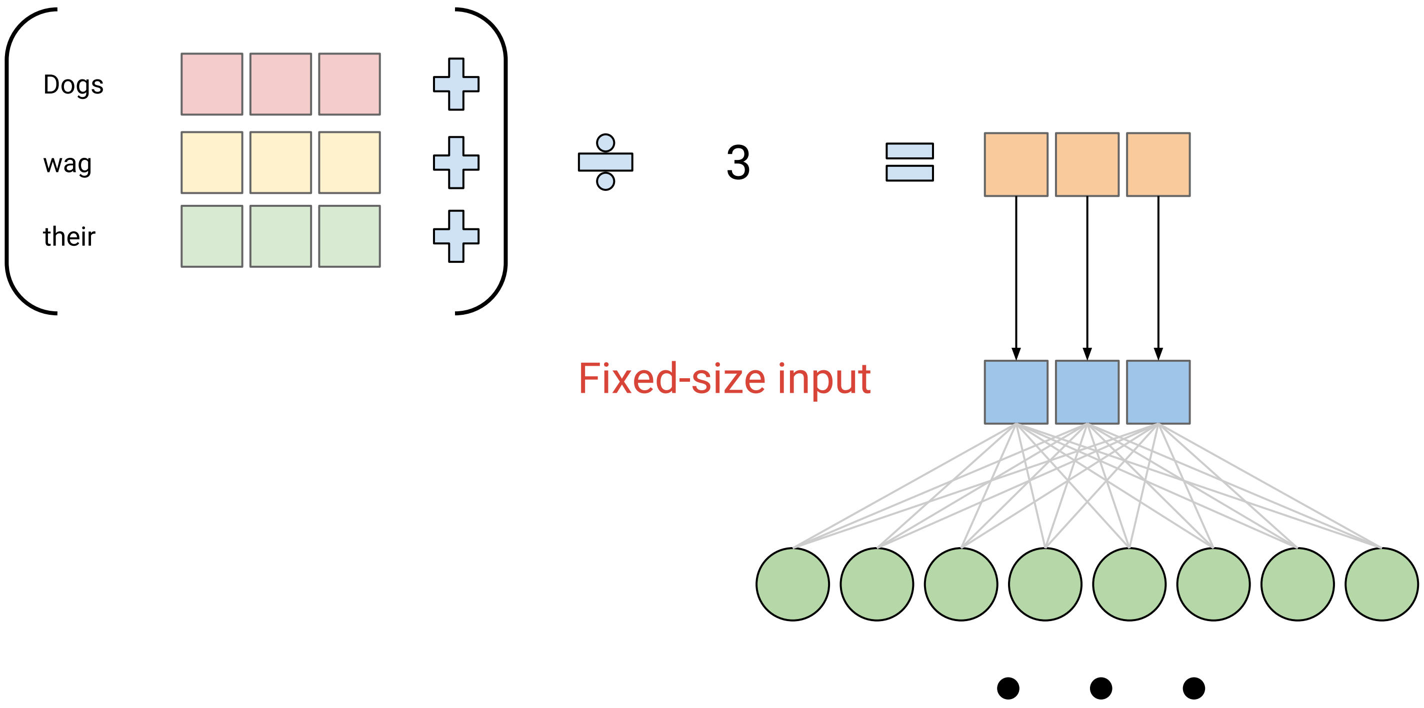
Pad	3 Hours Prior	2 Hours Prior	1 Hour Prior
0	6	9	12

$$w = \left[\cancel{\frac{1}{6}}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}\right]$$

3 Hours Prior	2 Hours Prior	1 Hour Prior	Pad
6	9	12	0

$$w = \left[\frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \cancel{\frac{2}{3}}\right]$$

Bagging is an alternative to padding



Order can be very important...

The
cat
sat
on
the
mat

=

The
mat
sat
on
the
cat

??

cloud.google.com