Stony Brook University

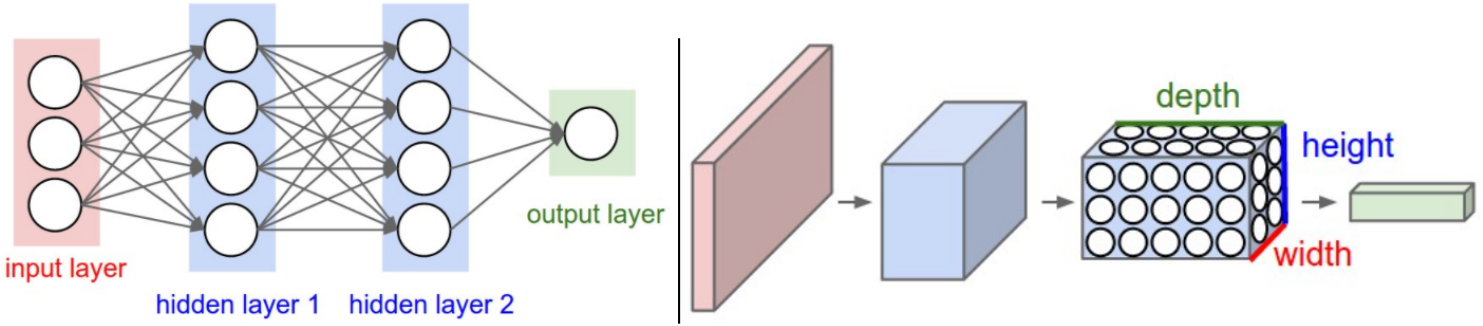
Google Cloud Platform

Yongjun Zhang, Ph.D.
Dept of Sociology and IACS
https://yongjunzhang.com

Today's Agenda

1. Mini Lecture (6:00-6:50PM)
2. Lab Tutorial (7:20-8:20 PM)

CNN Review

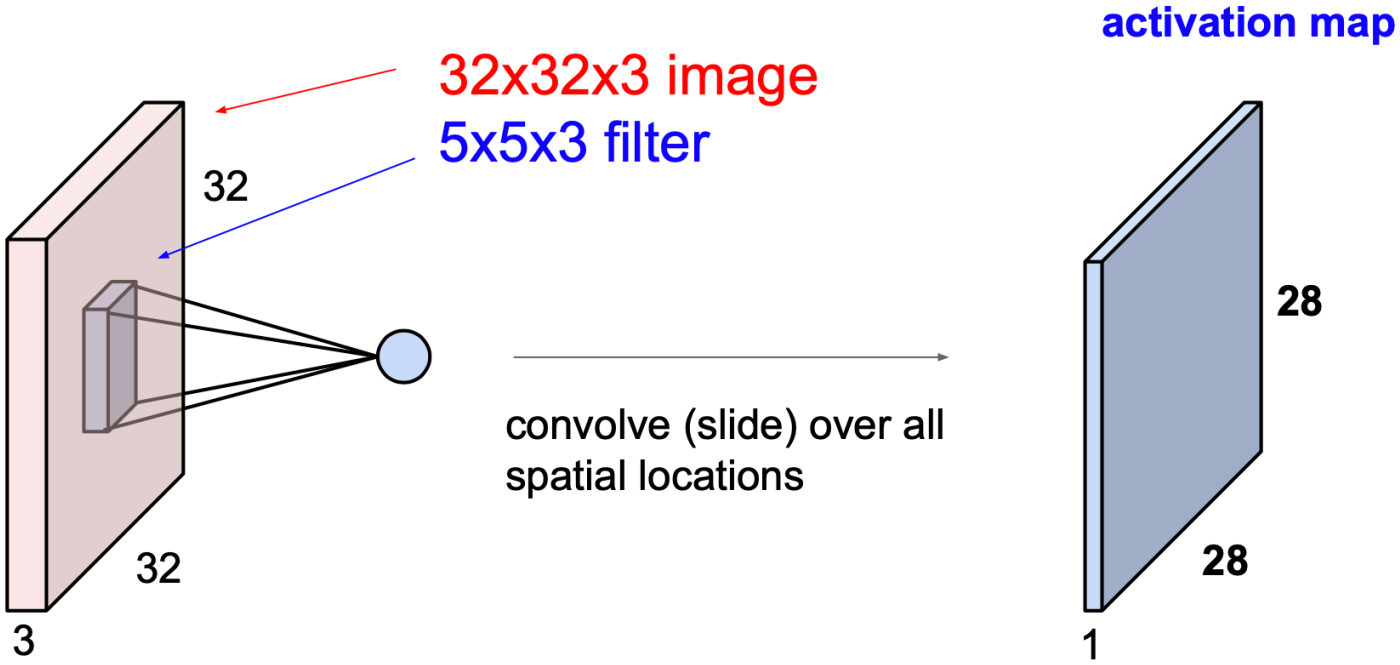


Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

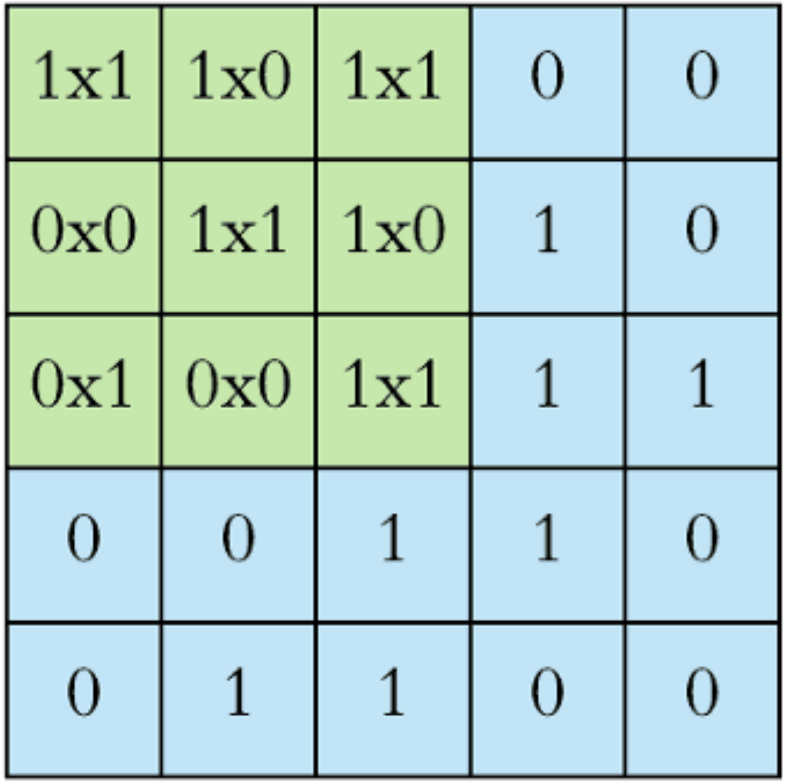
https://cs231n.github.io/

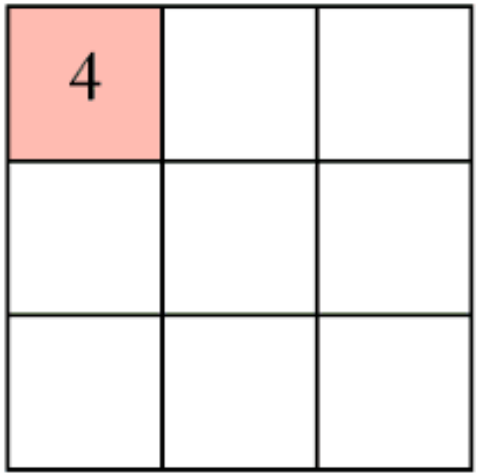
CNN stacks conv layers, pooling layers, and fully connected layers together…

Conv Layers

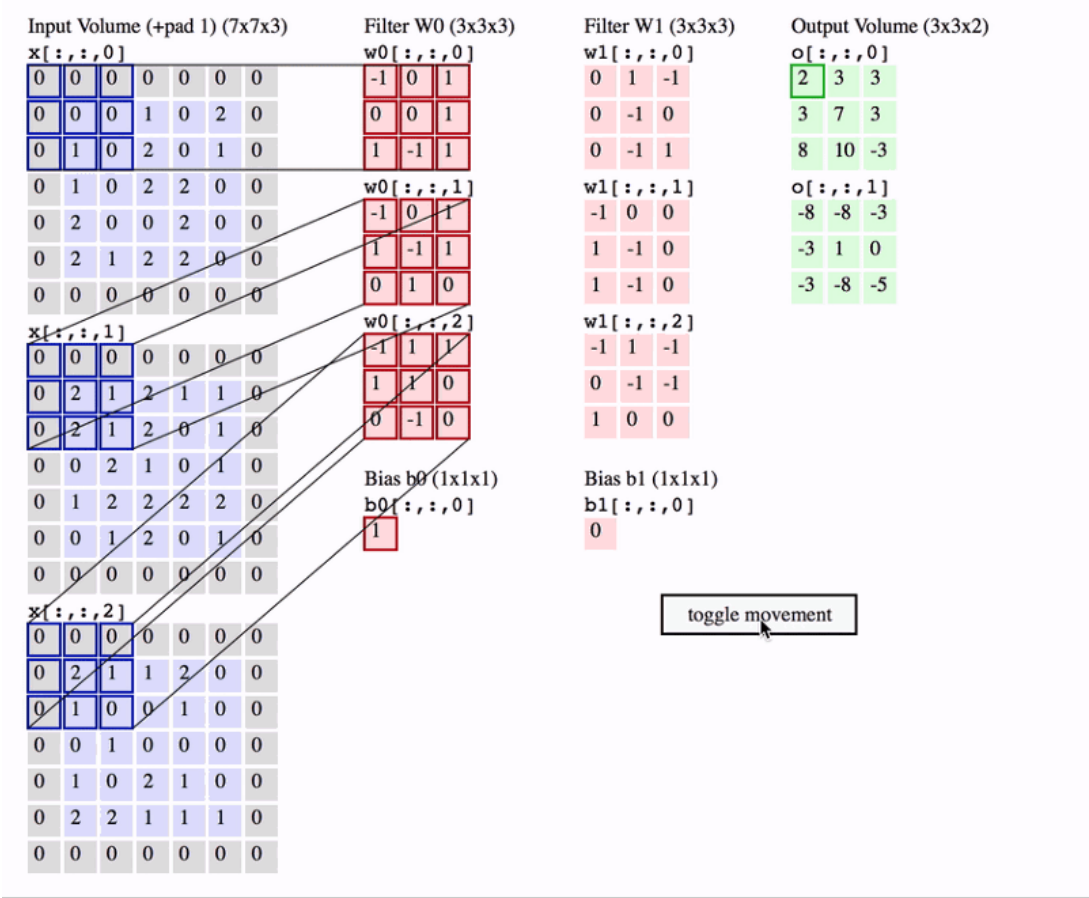


https://cs231n.github.io/





https://cs231n.github.io/

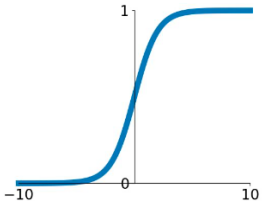


https://cs231n.github.io/

Activation Functions

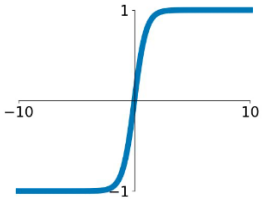
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



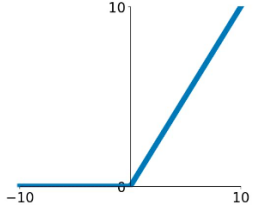
tanh

$$\tanh(x)$$



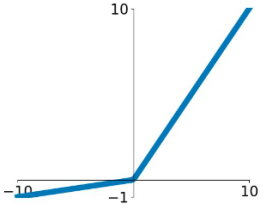
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

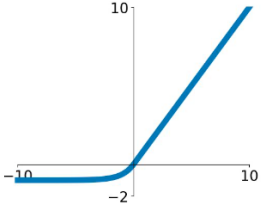


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



https://cs231n.github.io/

Convolution layer: summary

Let's assume input is W_1 x H_1 x C
Conv layer needs 4 hyperparameters:

- Number of filters **K**
- The filter size **F**
- The stride **S**
- The zero padding **P**

This will produce an output of W_2 x H_2 x K
where:

- $W_2 = (W_1 - F + 2P)/S + 1$
- $H_2 = (H_1 - F + 2P)/S + 1$

Number of parameters: F^2CK and K biases

https://cs231n.github.io/

Stony Brook University

Conv2D layer

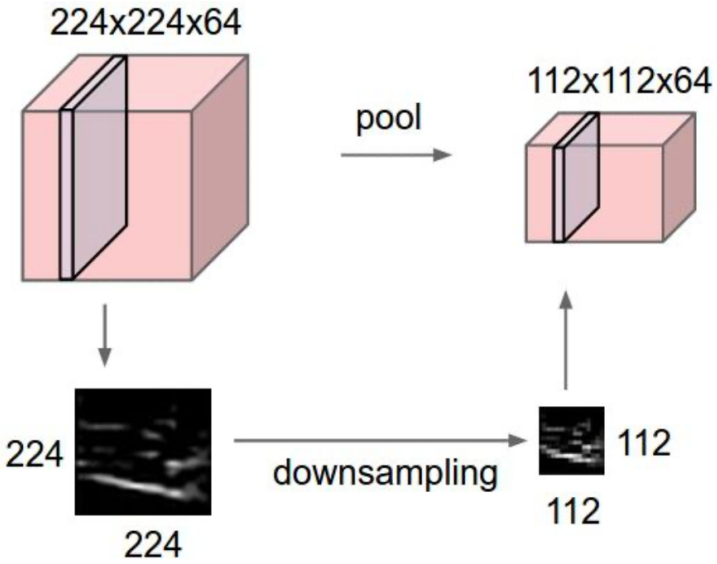
`Conv2D` class

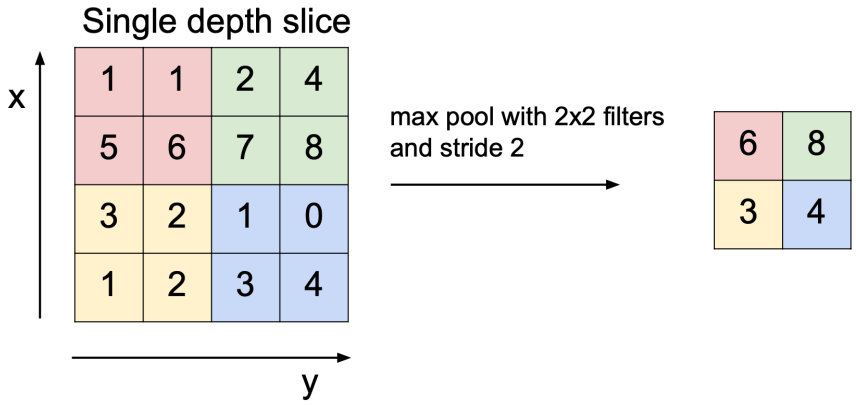
```
tf.keras.layers.Conv2D(
    filters,
    kernel_size,
    strides=(1, 1),
    padding="valid",
    data_format=None,
    dilation_rate=(1, 1),
    groups=1,
    activation=None,
    use_bias=True,
    kernel_initializer="glorot_uniform",
    bias_initializer="zeros",
    kernel_regularizer=None,
    bias_regularizer=None,
    activity_regularizer=None,
    kernel_constraint=None,
    bias_constraint=None,
    **kwargs
)
```

Arguments

- **filters**: Integer, the dimensionality of the output space (i.e. the number of output filters in the convolution).
- **kernel_size**: An integer or tuple/list of 2 integers, specifying the height and width of the 2D convolution window. Can be a single integer to specify the same value for all spatial dimensions.
- **strides**: An integer or tuple/list of 2 integers, specifying the strides of the convolution along the height and width. Can be a single integer to specify the same value for all spatial dimensions. Specifying any stride value != 1 is incompatible with specifying any `dilation_rate` value != 1.
- **padding**: one of `"valid"` or `"same"` (case-insensitive). `"valid"` means no padding. `"same"` results in padding evenly to the left/right or up/down of the input such that output has the same height/width dimension as the input.
- **data_format**: A string, one of `channels_last` (default) or `channels_first`. The ordering of the dimensions in the inputs. `channels_last` corresponds to inputs with shape `(batch_size, height, width, channels)` while `channels_first` corresponds to inputs with shape `(batch_size, channels, height, width)`. It defaults to the `image_data_format` value found in your Keras config file at `~/.keras/keras.json`. If you never set it, then it will be `channels_last`.
- **dilation_rate**: an integer or tuple/list of 2 integers, specifying the dilation rate to use for dilated convolution. Can be a single integer to specify the same value for all spatial dimensions. Currently, specifying any `dilation_rate` value != 1 is incompatible with specifying any stride value != 1.
- **groups**: A positive integer specifying the number of groups in which the input is split along the channel axis. Each group is convolved separately with `filters / groups` filters. The output is the concatenation of all the `groups` results along the channel axis. Input channels and `filters` must both be divisible by `groups`.
- **activation**: Activation function to use. If you don't specify anything, no activation is applied (see `keras.activations`).
- **use_bias**: Boolean, whether the layer uses a bias vector.
- **kernel_initializer**: Initializer for the `kernel` weights matrix (see `keras.initializers`).
- **bias_initializer**: Initializer for the bias vector (see `keras.initializers`).
- **kernel_regularizer**: Regularizer function applied to the `kernel` weights matrix (see `keras.regularizers`).
- **bias_regularizer**: Regularizer function applied to the bias vector (see `keras.regularizers`).
- **activity_regularizer**: Regularizer function applied to the output of the layer (its "activation") (see `keras.regularizers`).
- **kernel_constraint**: Constraint function applied to the kernel matrix (see `keras.constraints`).
- **bias_constraint**: Constraint function applied to the bias vector (see `keras.constraints`).

Pooling Layer





https://cs231n.github.io/

MaxPooling2D layer

`MaxPooling2D` class

```
tf.keras.layers.MaxPooling2D(
    pool_size=(2, 2), strides=None, padding="valid", data_format=None, **kwargs
)
```

Max pooling operation for 2D spatial data.

Downsamples the input representation by taking the maximum value over the window defined by `pool_size` for each dimension along the features axis. The window is shifted by `strides` in each dimension. The resulting output when using "valid" padding option has a shape(number of rows or columns) of: `output_shape = (input_shape - pool_size + 1) / strides)`

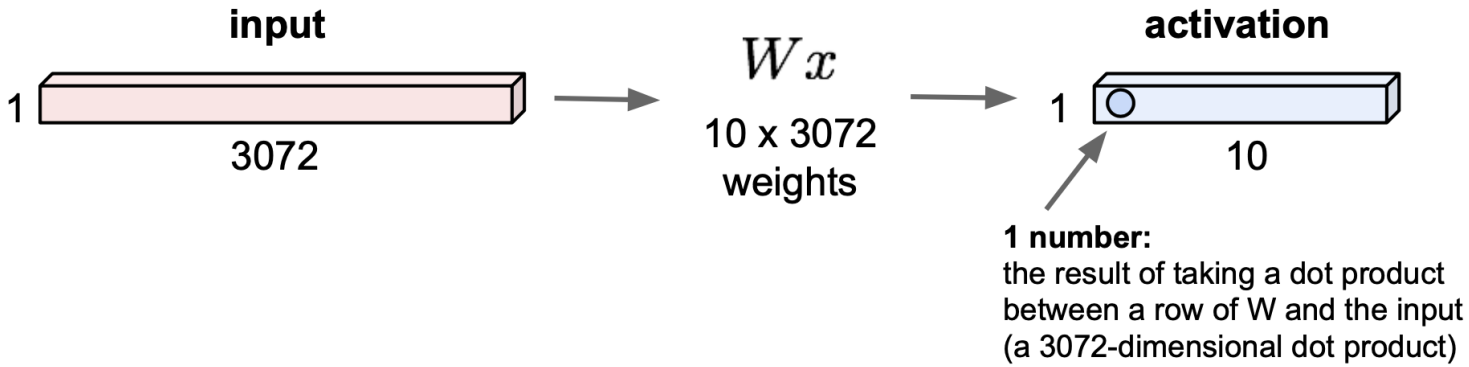
The resulting output shape when using the "same" padding option is: `output_shape = input_shape / strides`

Arguments

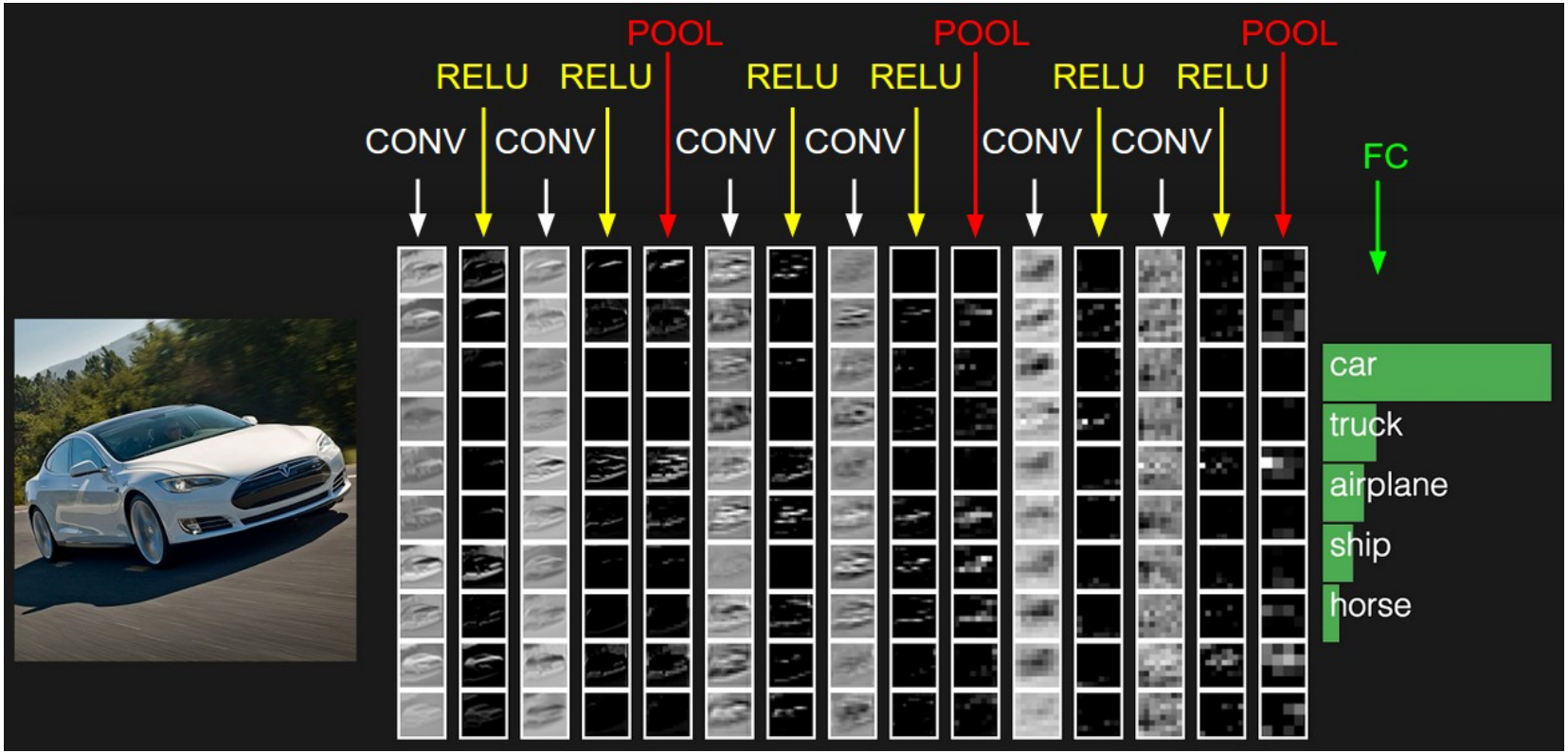
- **pool_size**: integer or tuple of 2 integers, window size over which to take the maximum. `(2, 2)` will take the max value over a 2x2 pooling window. If only one integer is specified, the same window length will be used for both dimensions.
- **strides**: Integer, tuple of 2 integers, or None. Strides values. Specifies how far the pooling window moves for each pooling step. If None, it will default to `pool_size`.
- **padding**: One of `"valid"` or `"same"` (case-insensitive). `"valid"` means no padding. `"same"` results in padding evenly to the left/right or up/down of the input such that output has the same height/width dimension as the input.
- **data_format**: A string, one of `channels_last` (default) or `channels_first`. The ordering of the dimensions in the inputs. `channels_last` corresponds to inputs with shape `(batch, height, width, channels)` while `channels_first` corresponds to inputs with shape `(batch, channels, height, width)`. It defaults to the `image_data_format` value found in your Keras config file at `~/.keras/keras.json`. If you never set it, then it will be "channels_last".

Fully Connected Layer

32x32x3 image -> stretch to 3072 x 1



https://cs231n.github.io/



https://cs231n.github.io/

```
model = keras.Sequential(
    [
        keras.Input(shape=input_shape),
        layers.Conv2D(32, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Flatten(),
        layers.Dropout(0.5),
        layers.Dense(num_classes, activation="softmax"),
    ]
)

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 1600)              0
_________________________________________________________________
dropout (Dropout)            (None, 1600)              0
_________________________________________________________________
dense (Dense)                (None, 10)                16010
=================================================================
Total params: 34,826
Trainable params: 34,826
Non-trainable params: 0
_________________________________________________________________
```

For more details on CNN architecture

https://cs231n.github.io/

https://cs231n.github.io/convolutional-networks/

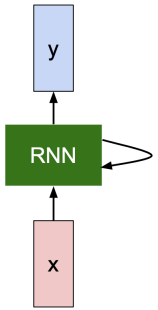
http://cs231n.stanford.edu/slides/2020/lecture_9.pdf

Recurrent Neural Network

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

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

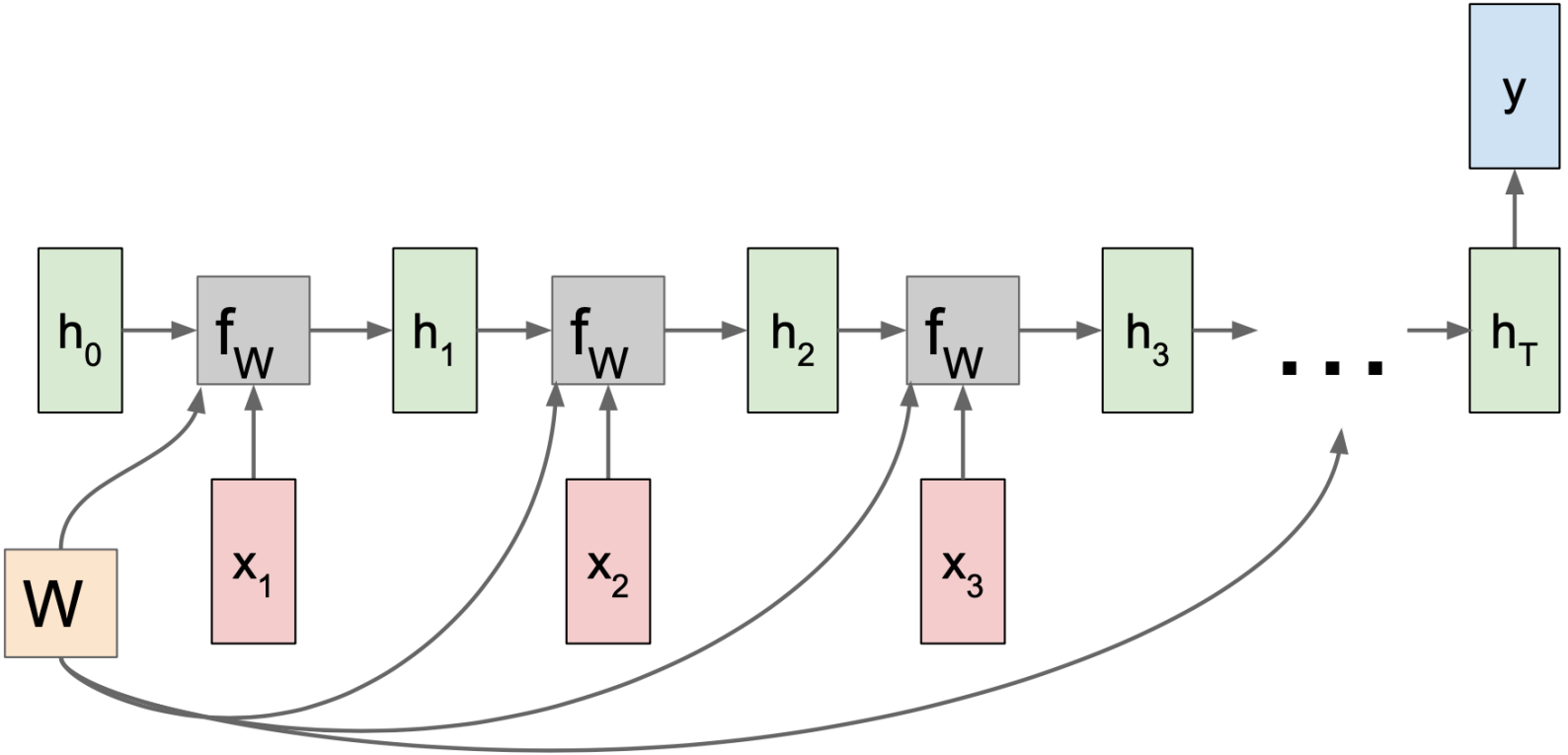
new state — some function with parameters W — old state — input vector at some time step



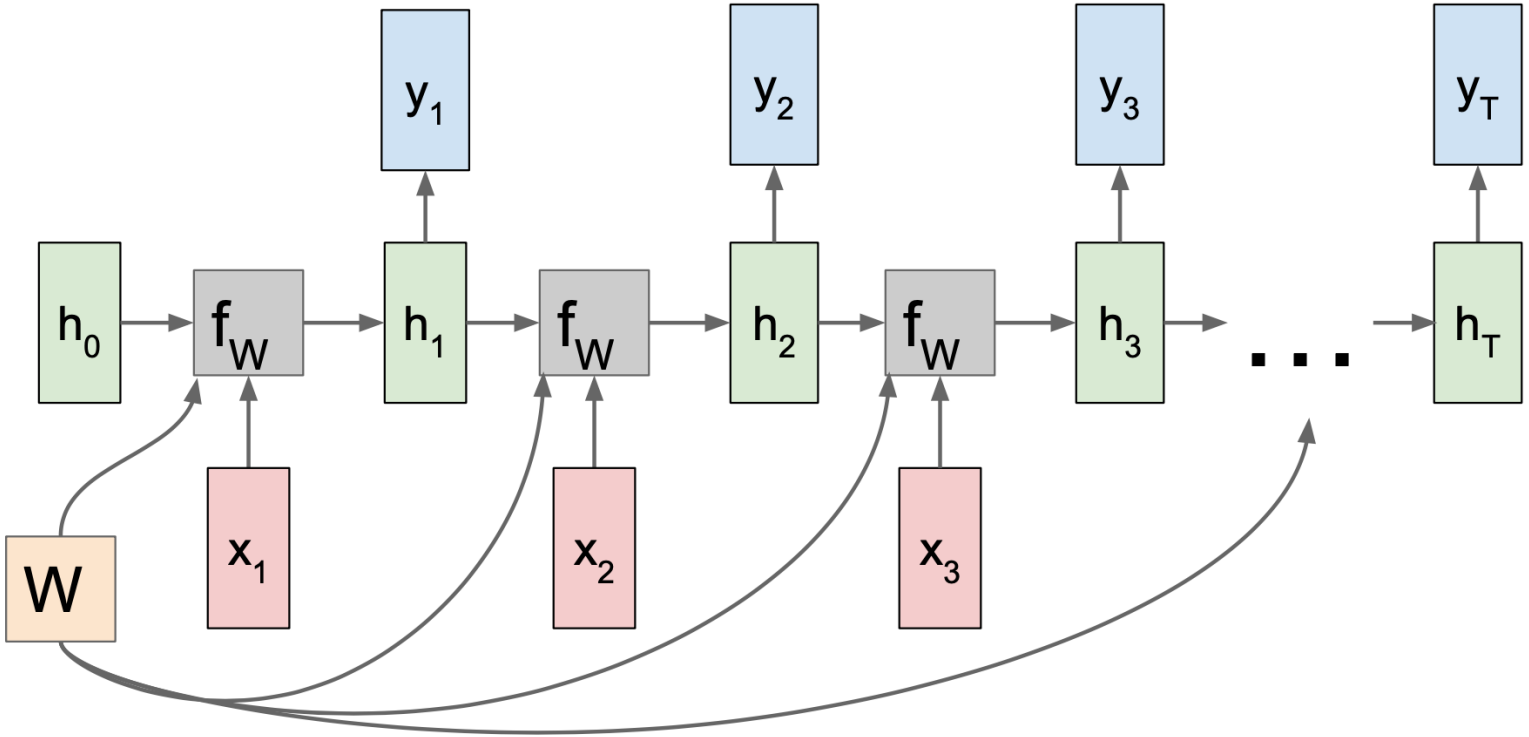
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

https://cs231n.github.io/



https://cs231n.github.io/

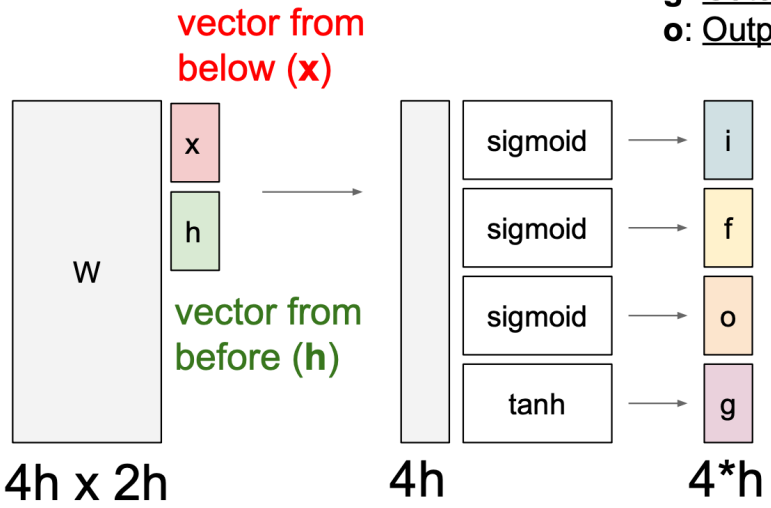


https://cs231n.github.io/

Long Short Term Memory (LSTM)

[Hochreiter et al., 1997]

f: Forget gate, Whether to erase cell
i: Input gate, whether to write to cell
g: Gate gate (?), How much to write to cell
o: Output gate, How much to reveal cell



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

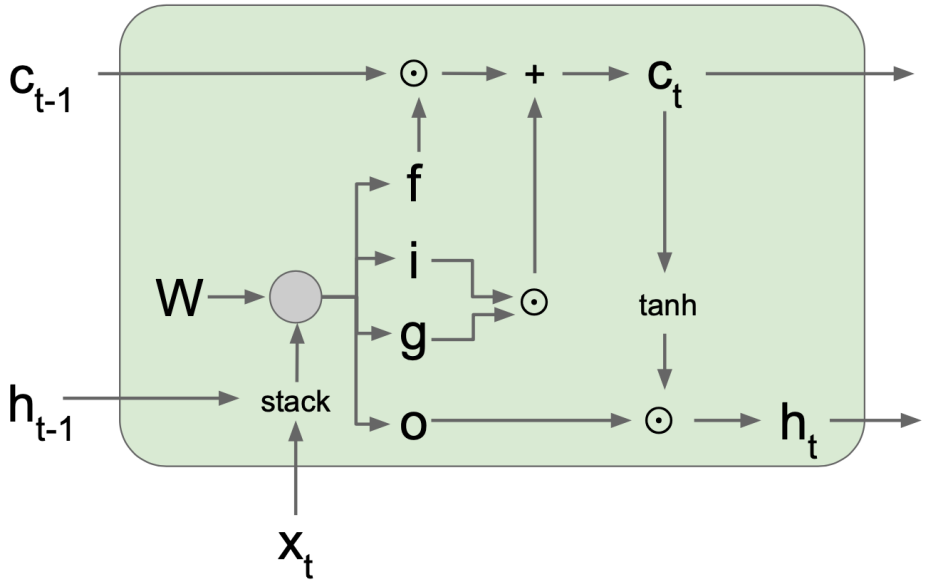
$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

https://cs231n.github.io/

Long Short Term Memory (LSTM)

[Hochreiter et al., 1997]



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

https://cs231n.github.io/

LSTM layer

`LSTM` class

```
tf.keras.layers.LSTM(
    units,
    activation="tanh",
    recurrent_activation="sigmoid",
    use_bias=True,
    kernel_initializer="glorot_uniform",
    recurrent_initializer="orthogonal",
    bias_initializer="zeros",
    unit_forget_bias=True,
    kernel_regularizer=None,
    recurrent_regularizer=None,
    bias_regularizer=None,
    activity_regularizer=None,
    kernel_constraint=None,
    recurrent_constraint=None,
    bias_constraint=None,
    dropout=0.0,
    recurrent_dropout=0.0,
    return_sequences=False,
    return_state=False,
    go_backwards=False,
    stateful=False,
    time_major=False,
    unroll=False,
    **kwargs
)
```

Long Short-Term Memory layer - Hochreiter 1997.

Arguments

- **units**: Positive integer, dimensionality of the output space.
- **activation**: Activation function to use. Default: hyperbolic tangent (`tanh`). If you pass `None`, no activation is applied (ie. "linear" activation: `a(x) = x`).
- **recurrent_activation**: Activation function to use for the recurrent step. Default: sigmoid (`sigmoid`). If you pass `None`, no activation is applied (ie. "linear" activation: `a(x) = x`).
- **use_bias**: Boolean (default `True`), whether the layer uses a bias vector.
- **kernel_initializer**: Initializer for the `kernel` weights matrix, used for the linear transformation of the inputs. Default: `glorot_uniform`.
- **recurrent_initializer**: Initializer for the `recurrent_kernel` weights matrix, used for the linear transformation of the recurrent state. Default: `orthogonal`.
- **bias_initializer**: Initializer for the bias vector. Default: `zeros`.
- **unit_forget_bias**: Boolean (default `True`). If True, add 1 to the bias of the forget gate at initialization. Setting it to true will also force `bias_initializer="zeros"`. This is recommended in Jozefowicz et al..
- **kernel_regularizer**: Regularizer function applied to the `kernel` weights matrix. Default: `None`.
- **recurrent_regularizer**: Regularizer function applied to the `recurrent_kernel` weights matrix. Default: `None`.
- **bias_regularizer**: Regularizer function applied to the bias vector. Default: `None`.
- **activity_regularizer**: Regularizer function applied to the output of the layer (its "activation"). Default: `None`.
- **kernel_constraint**: Constraint function applied to the `kernel` weights matrix. Default: `None`.
- **recurrent_constraint**: Constraint function applied to the `recurrent_kernel` weights matrix. Default: `None`.
- **bias_constraint**: Constraint function applied to the bias vector. Default: `None`.
- **dropout**: Float between 0 and 1. Fraction of the units to drop for the linear transformation of the inputs. Default: 0.
- **recurrent_dropout**: Float between 0 and 1. Fraction of the units to drop for the linear transformation of the recurrent state. Default: 0.
- **return_sequences**: Boolean. Whether to return the last output. in the output sequence, or the full sequence. Default: `False`.
- **return_state**: Boolean. Whether to return the last state in addition to the output. Default: `False`.
- **go_backwards**: Boolean (default `False`). If True, process the input sequence backwards and return the reversed sequence.
- **stateful**: Boolean (default `False`). If True, the last state for each sample at index i in a batch will be used as initial state for the sample of index i in the following batch.
- **time_major**: The shape format of the `inputs` and `outputs` tensors. If True, the inputs and outputs will be in shape `[timesteps, batch, feature]`, whereas in the False case, it will be `[batch, timesteps, feature]`. Using `time_major = True` is a bit more efficient because it avoids transposes at the beginning and end of the RNN calculation. However, most TensorFlow data is batch-major, so by default this function accepts input and emits output in batch-major form.
- **unroll**: Boolean (default `False`). If True, the network will be unrolled, else a symbolic loop will be used. Unrolling can speed-up a RNN, although it tends to be more memory-intensive. Unrolling is only suitable for short sequences.

Bidirectional layer

`Bidirectional` class

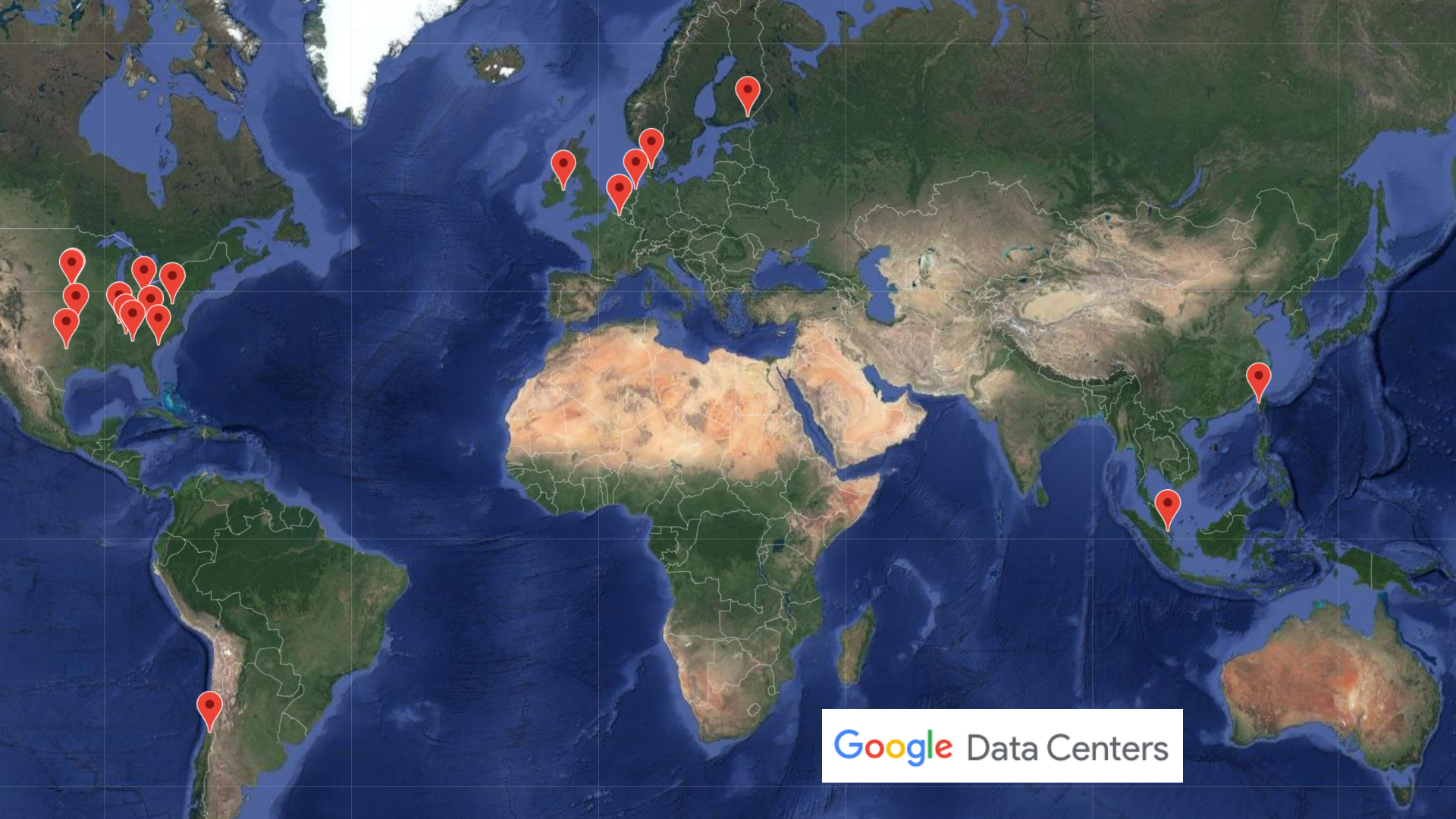
```
tf.keras.layers.Bidirectional(
    layer, merge_mode="concat", weights=None, backward_layer=None, **kwargs
)
```

Bidirectional wrapper for RNNs.

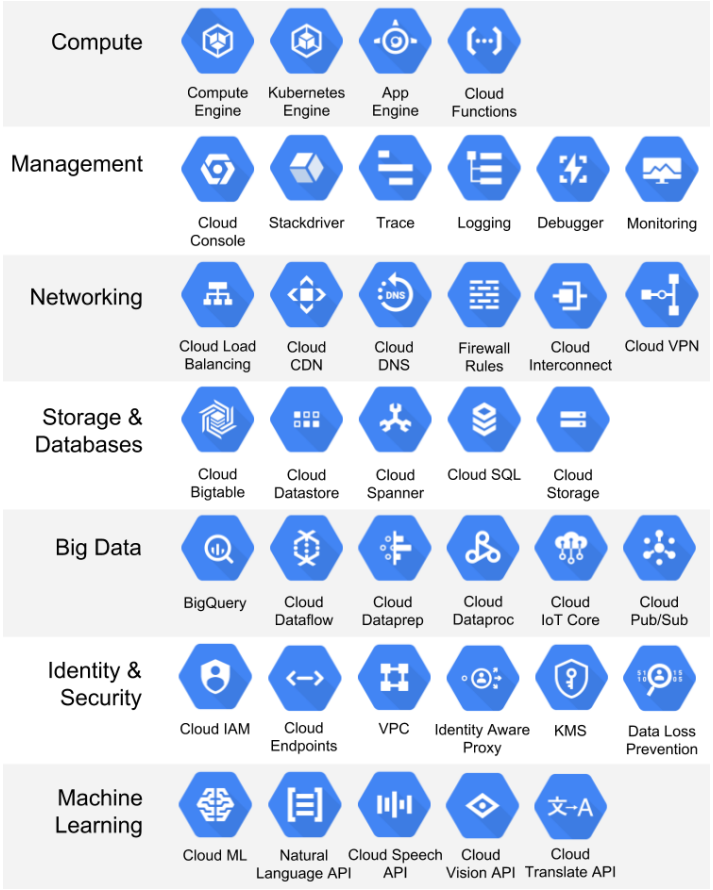
Arguments

- **layer**: `keras.layers.RNN` instance, such as `keras.layers.LSTM` or `keras.layers.GRU`. It could also be a `keras.layers.Layer` instance that meets the following criteria:
    1. Be a sequence-processing layer (accepts 3D+ inputs).
    2. Have a `go_backwards`, `return_sequences` and `return_state` attribute (with the same semantics as for the `RNN` class).
    3. Have an `input_spec` attribute.
    4. Implement serialization via `get_config()` and `from_config()`. Note that the recommended way to create new RNN layers is to write a custom RNN cell and use it with `keras.layers.RNN`, instead of subclassing `keras.layers.Layer` directly.
- **merge_mode**: Mode by which outputs of the forward and backward RNNs will be combined. One of {'sum', 'mul', 'concat', 'ave', None}. If None, the outputs will not be combined, they will be returned as a list. Default value is 'concat'.
- **backward_layer**: Optional `keras.layers.RNN`, or `keras.layers.Layer` instance to be used to handle backwards input processing. If `backward_layer` is not provided, the layer instance passed as the `layer` argument will be used to generate the backward layer automatically. Note that the provided `backward_layer` layer should have properties matching those of the `layer` argument, in particular it should have the same values for `stateful`, `return_states`, `return_sequence`, etc. In addition, `backward_layer` and `layer` should have different `go_backwards` argument values. A `ValueError` will be raised if these requirements are not met.

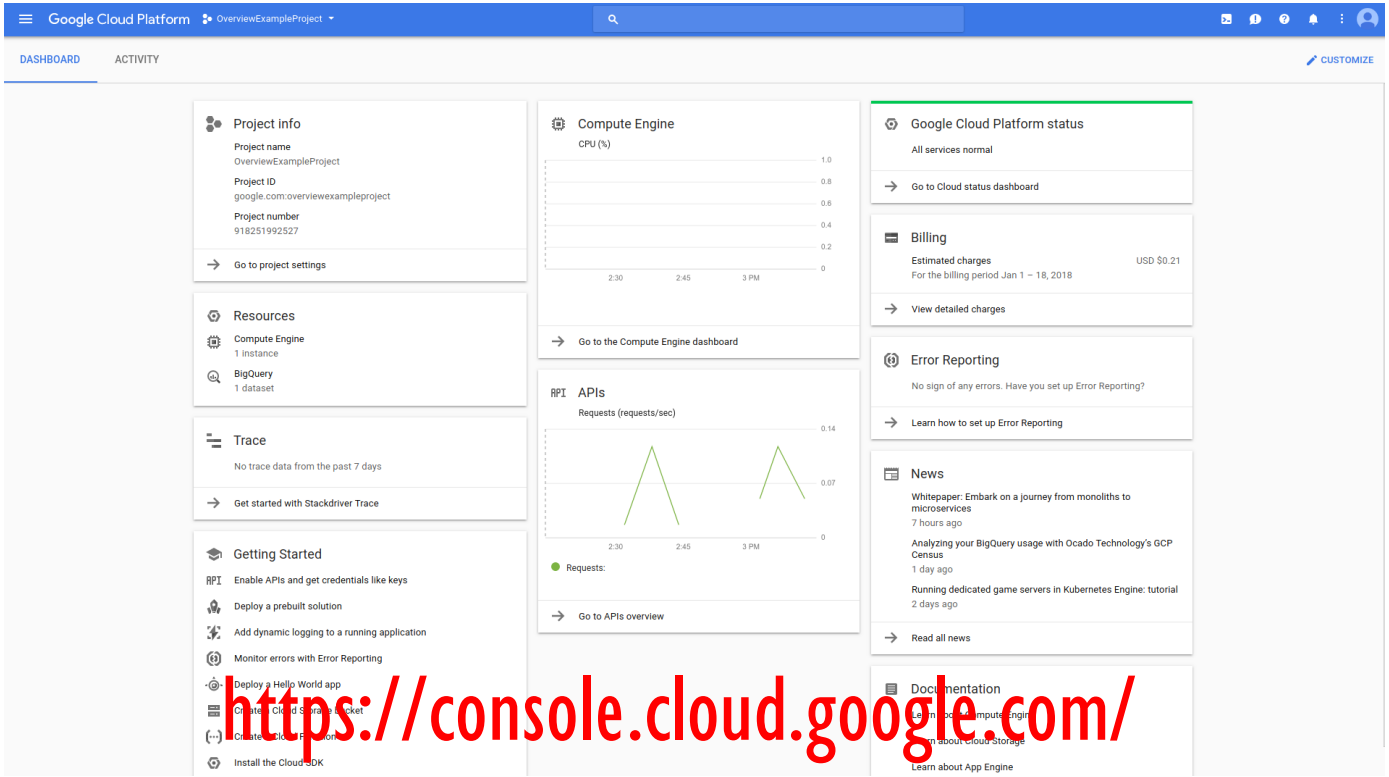
Google Cloud Platform


Google Data Centers

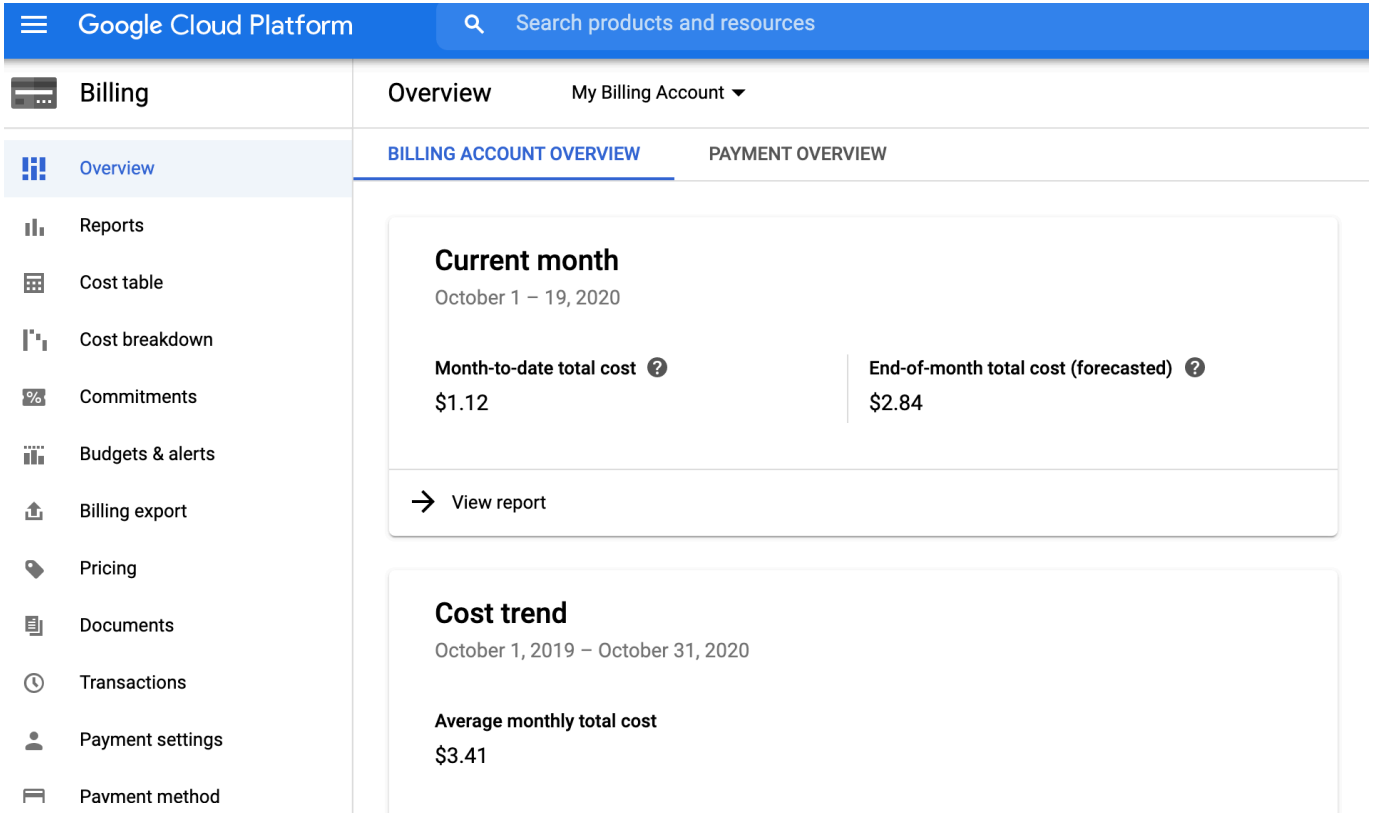
Google Cloud



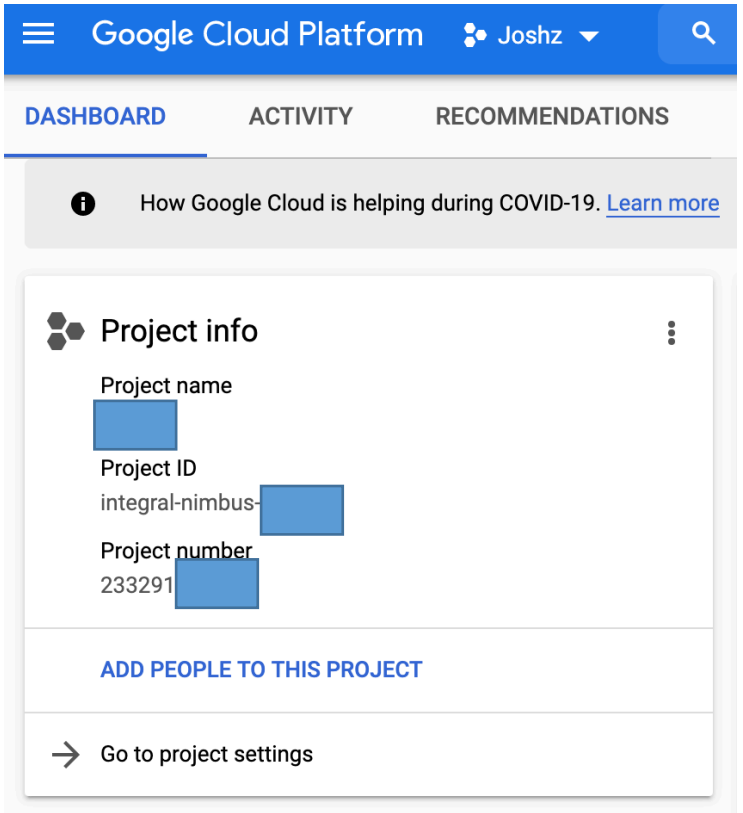
After you signed in GCP, you should see the Console

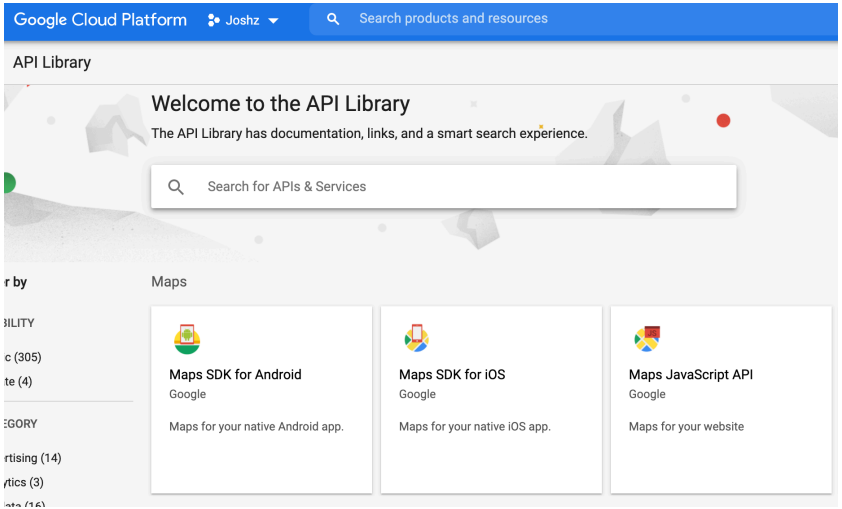


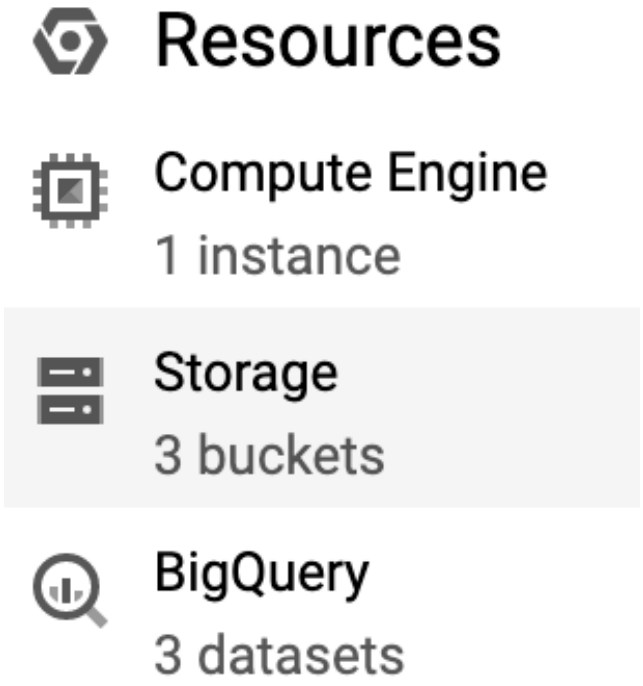
https://console.cloud.google.com/


Google Cloud Platform
Search products and resources
Billing
Overview
Reports
Cost table
Cost breakdown
Commitments
Budgets & alerts
Billing export
Pricing
Documents
Transactions
Payment settings
Payment method
Overview
My Billing Account
BILLING ACCOUNT OVERVIEW
PAYMENT OVERVIEW
Current month
October 1 – 19, 2020
Month-to-date total cost
$1.12
End-of-month total cost (forecasted)
$2.84
View report
Cost trend
October 1, 2019 – October 31, 2020
Average monthly total cost
$3.41

The Google Cloud Console provides a web-based, graphical user interface that you can use to manage your Google Cloud projects and resources. When you use the Cloud Console, you create a new project, or choose an existing project, and use the resources that you create in the context of that project. You can create multiple projects, so you can use projects to separate your work in whatever way makes sense for you. For example, you might start a new project if you want to make sure only certain team members can access the resources in that project, while all team members can continue to access resources in another project.




Google Cloud Platform
Joshz
Search products and resources
API Library
Welcome to the API Library
The API Library has documentation, links, and a smart search experience.
Search for APIs & Services
Maps
Maps SDK for Android
Google
Maps for your native Android app.
Maps SDK for iOS
Google
Maps for your native iOS app.
Maps JavaScript API
Google
Maps for your website

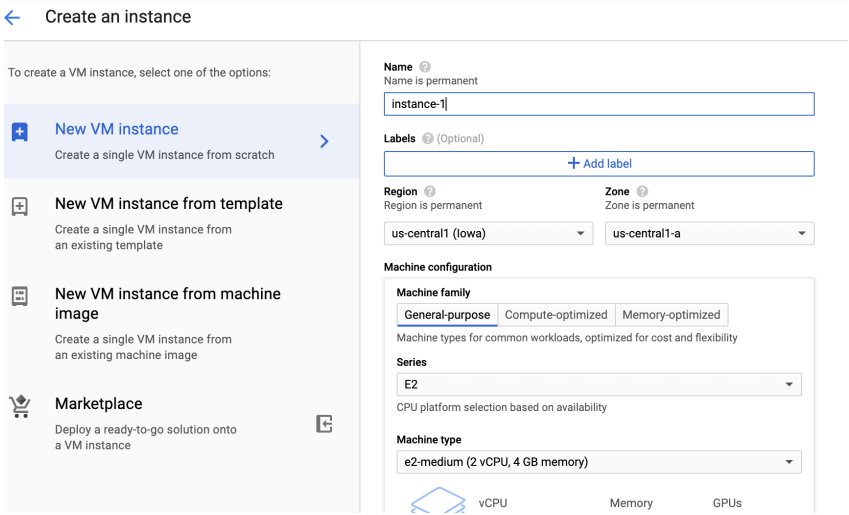

Resources
Compute Engine
1 instance
Storage
3 buckets
BigQuery
3 datasets

If you don't have Seawulf HPC account, you can use GC computing engine or just use colab.research.goolge.com for free computing resources

In case you need heavy computing resources, you can create a CE for your task.

Login to your GCP
Click Computer Engine
Create an VM
Always stop your instance after you finish your task

You can use GC SDK tools to access your VM



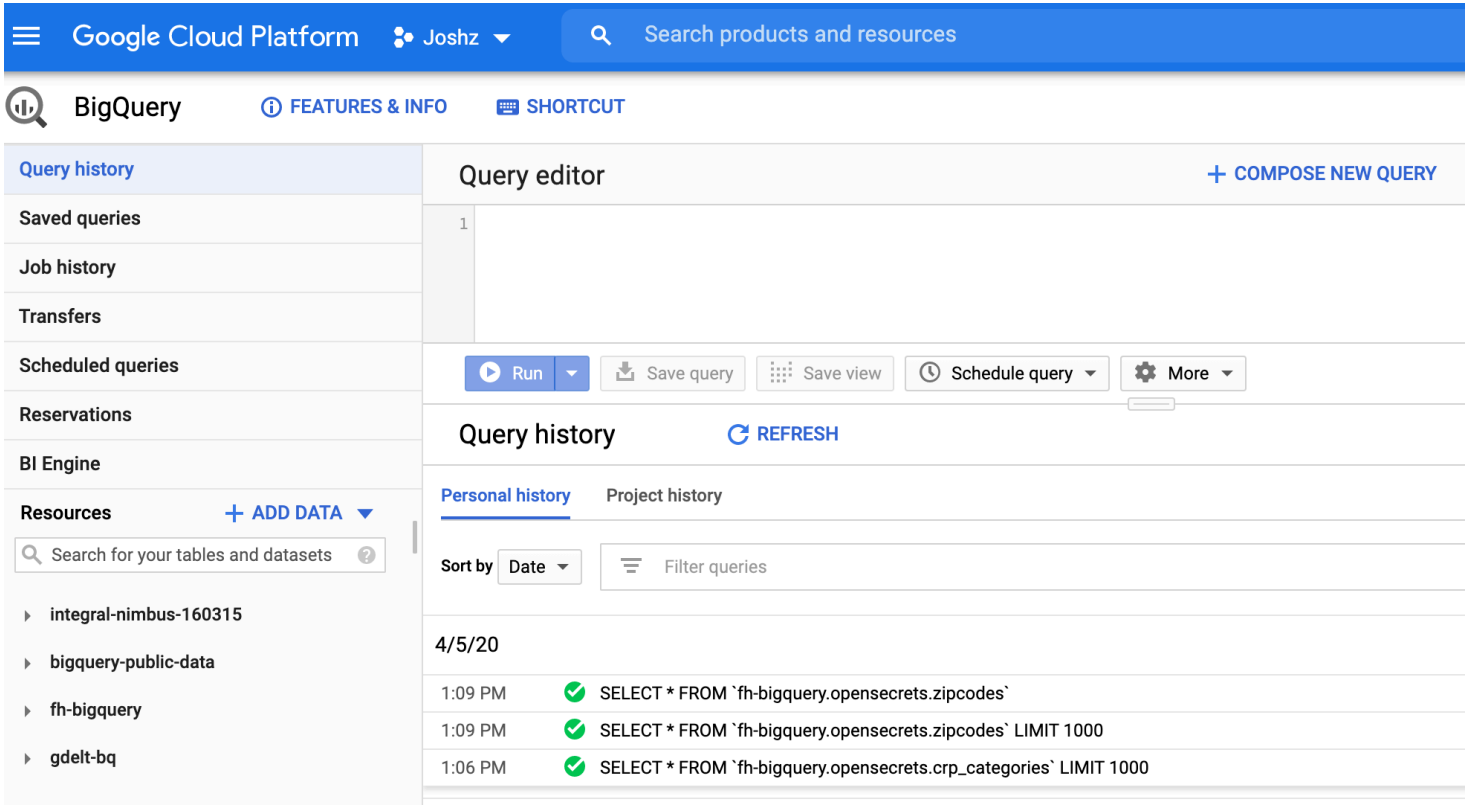
GCLOUD

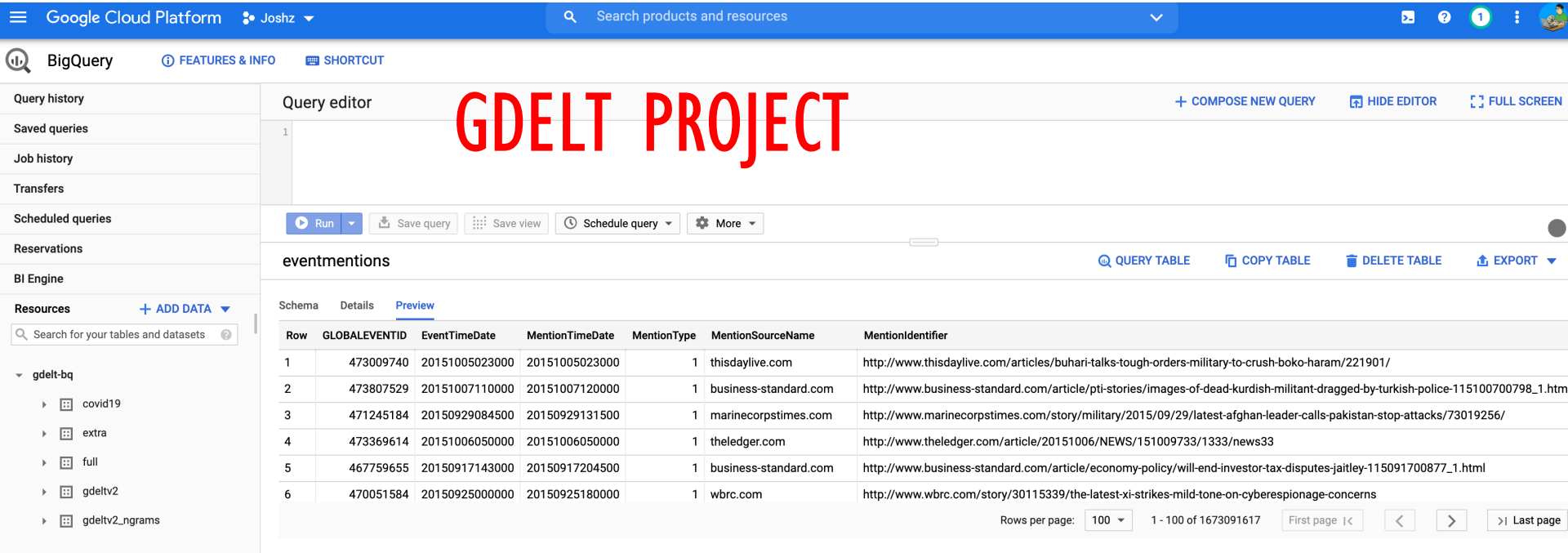
https://cloud.google.com/sdk/gcloud

```
Last login: Sun Oct 18 15:47:25 on ttys000
joshzyj@Yongjuns-MacBook-Pro ~ % gcloud auth login
Your browser has been opened to visit:

    https://accounts.google.com/o/oauth2/auth?client_id=32555940559.apps.googleusercontent
com&redirect_uri=http%3A%2F%2Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googleapi
.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+http
%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fau
h%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&code_challenge=VfROk
fbEYiEGWwVpiYGzKYKnpgw1OVNyxGWfCgIilc&code_challenge_method=S256&access_type=offline&respo
se_type=code&prompt=select_account



You are now logged in as [joshzyj@gmail.com].
Your current project is [integral-nimbus-160315].  You can change this setting by running:
  $ gcloud config set project PROJECT_ID


Updates are available for some Cloud SDK components.  To install them,
please run:
  $ gcloud components update



To take a quick anonymous survey, run:
  $ gcloud survey

joshzyj@Yongjuns-MacBook-Pro ~ %
```

```
$ gcloud config set account YourGMAIL@gmail.com
$ gcloud config set project  YOUR GC PROJECT ID
```

Big Query


Google Cloud Platform
Joshz
Search products and resources
BigQuery
FEATURES & INFO
SHORTCUT
Query history
Saved queries
Job history
Transfers
Scheduled queries
Reservations
BI Engine
Resources
ADD DATA
Search for your tables and datasets
integral-nimbus-160315
bigquery-public-data
fh-bigquery
gdelt-bq
Query editor
COMPOSE NEW QUERY
1
Run
Save query
Save view
Schedule query
More
Query history
REFRESH
Personal history
Project history
Sort by
Date
Filter queries
4/5/20
1:09 PM
SELECT * FROM `fh-bigquery.opensecrets.zipcodes`
1:09 PM
SELECT * FROM `fh-bigquery.opensecrets.zipcodes` LIMIT 1000
1:06 PM
SELECT * FROM `fh-bigquery.opensecrets.crp_categories` LIMIT 1000


Google Cloud Platform
Joshz
Search products and resources
BigQuery
FEATURES & INFO
SHORTCUT
Query history
Saved queries
Job history
Transfers
Scheduled queries
Reservations
BI Engine
Resources
ADD DATA
Search for your tables and datasets
gdelt-bq
covid19
extra
full
gdeltv2
gdeltv2_ngrams
Query editor
GDELT PROJECT
COMPOSE NEW QUERY
HIDE EDITOR
FULL SCREEN
Run
Save query
Save view
Schedule query
More
eventmentions
QUERY TABLE
COPY TABLE
DELETE TABLE
EXPORT
Schema
Details
Preview
Row GLOBALEVENTID EventTimeDate MentionTimeDate MentionType MentionSourceName MentionIdentifier
1 473009740 20151005023000 20151005023000 1 thisdaylive.com http://www.thisdaylive.com/articles/buhari-talks-tough-orders-military-to-crush-boko-haram/221901/
2 473807529 20151007110000 20151007120000 1 business-standard.com http://www.business-standard.com/article/pti-stories/images-of-dead-kurdish-militant-dragged-by-turkish-police-115100700798_1.htm
3 471245184 20150929084500 20150929131500 1 marinecorpstimes.com http://www.marinecorpstimes.com/story/military/2015/09/29/latest-afghan-leader-calls-pakistan-stop-attacks/73019256/
4 473369614 20151006050000 20151006050000 1 theledger.com http://www.theledger.com/article/20151006/NEWS/151009733/1333/news33
5 467759655 20150917143000 20150917204500 1 business-standard.com http://www.business-standard.com/article/economy-policy/will-end-investor-tax-disputes-jaitley-115091700877_1.html
6 470051584 20150925000000 20150925180000 1 wbrc.com http://www.wbrc.com/story/30115339/the-latest-xi-strikes-mild-tone-on-cyberespionage-concerns
Rows per page: 100
1 - 100 of 1673091617
First page
Last page

You can use bigquery via graphical console

Use regular SQL queries

```
SELECT
    GLOBALEVENTID,
    EventCode,
    EventBaseCode,
    Actor1CountryCode,
    Actor2CountryCode,
    MonthYear,
    Year,
    AvgTone,
    GoldsteinScale,
    NumMentions,
    NumArticles
FROM
    `gdelt-bq.gdeltv2.events_partitioned`
WHERE
    Actor1CountryCode IS NOT NULL
    AND YEAR<=2020
    AND YEAR>=2014
    AND EventRootCode="14"
```

You can also do this remotely

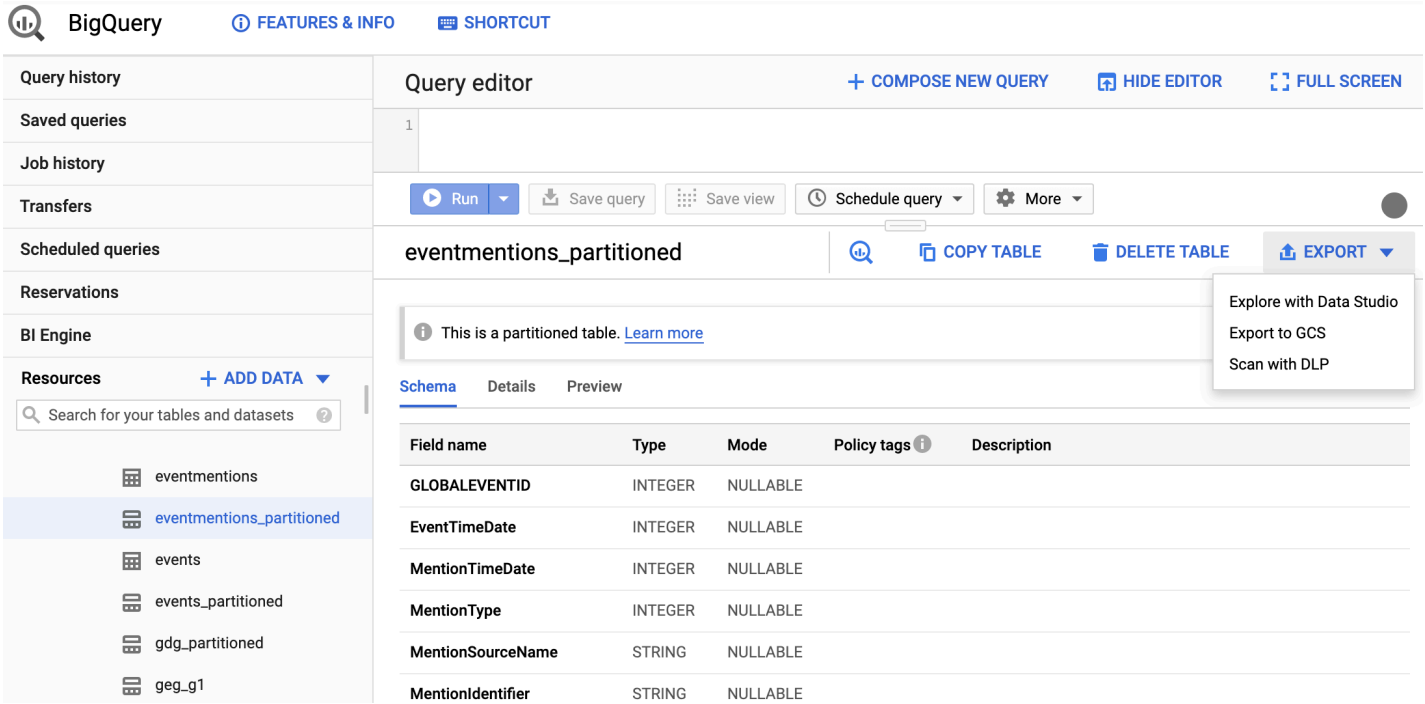
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on 2020-10-14

@author: Yongjun Zhang

Use Google BigQuery to extract protest events from GDELT Project

"""

# Before running the following code, you need to set up a service account and download credentials
# Following the tutorial to set up your service account:
# https://cloud.google.com/bigquery/docs/quickstarts/quickstart-client-libraries
# once you download your json file, open your terminal and run this:
# export GOOGLE_APPLICATION_CREDENTIALS="/Your path to/client_secrets.json"

import os
# replace the path to your PATH
os.chdir("/users/joshzyj/google drive/gdelt/")

# You should install google.cloud sdk first
# You need to install google.cloud.bigquery first
# pip install google.cloud.bigquery

from google.cloud import bigquery

client = bigquery.Client()
```

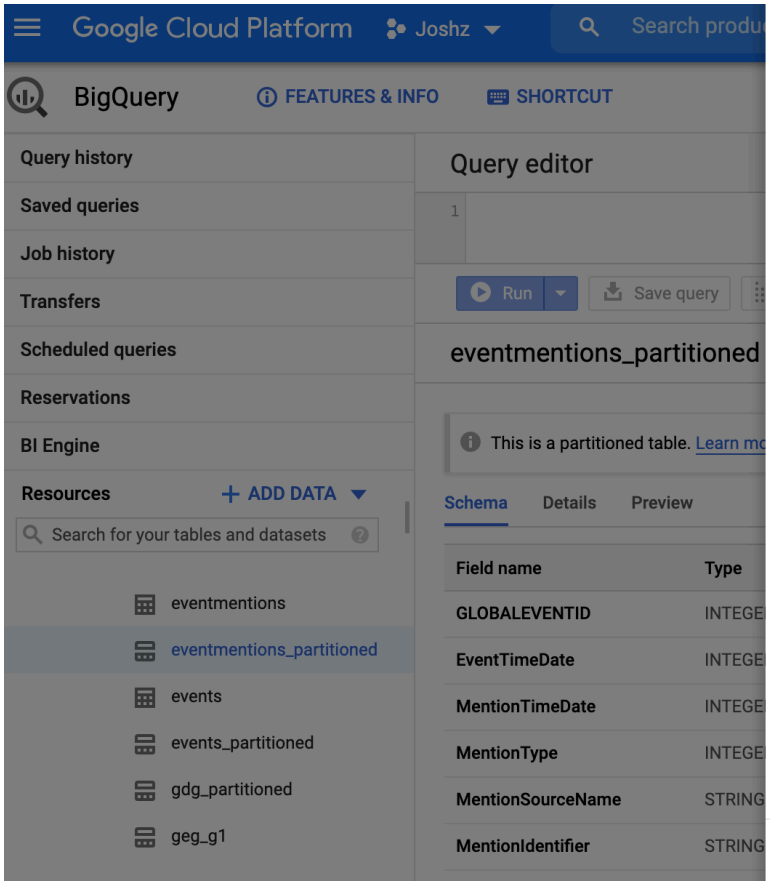
```
#####################################
# The following codes query events_partitioned table and save a protest table
# Note that we have not yet saved a table into local.
# We remotely access bigquery and save a table into bigquery

# Here is the webpage for gdelt v2
# https://blog.gdeltproject.org/gdelt-2-0-our-global-world-in-realtime/
# It provdes some explanations about the database
# Here is the codebook:
# http://data.gdeltproject.org/documentation/GDELT-Event_Codebook-V2.0.pdf
# Here is the codebook for GKG:
# http://data.gdeltprojec.org/documentation/GDELT-Global_Knowledge_Graph_Codebook-V2.1.pdf

# Define the query syntax.
# Note Year is the event year.
# Event root code 14 indicates protests
```

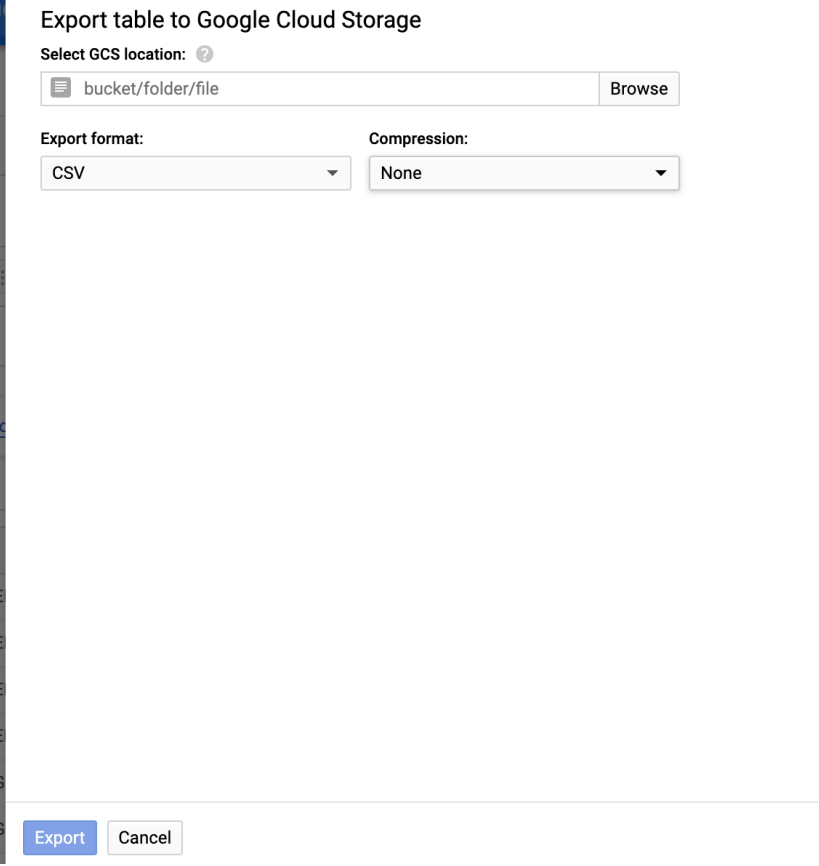
```
# Here is the webpage for gdelt v2
# https://blog.gdeltproject.org/gdelt-2-0-our-global-world-in-realtime/
# It provdes some explanations about the database
# Here is the codebook:
# http://data.gdeltproject.org/documentation/GDELT-Event_Codebook-V2.0.pdf
# Here is the codebook for GKG:
# http://data.gdeltprojec.org/documentation/GDELT-Global_Knowledge_Graph_Codebook-V2.1.pdf

# Define the query syntax.
# Note Year is the event year.
# Event root code 14 indicates protests
QUERY1 = ("""
    SELECT
        GLOBALEVENTID,
        EventCode,
        EventBaseCode,
        Actor1CountryCode,
        Actor2CountryCode,
        MonthYear,
        Year,
        AvgTone,
        GoldsteinScale,
        NumMentions,
        NumArticles
    FROM
        `gdelt-bq.gdeltv2.events_partitioned`
    WHERE
        Actor1CountryCode IS NOT NULL
        AND YEAR<=2020
        AND YEAR>=2014
        AND EventRootCode="14"
    """
    )
```

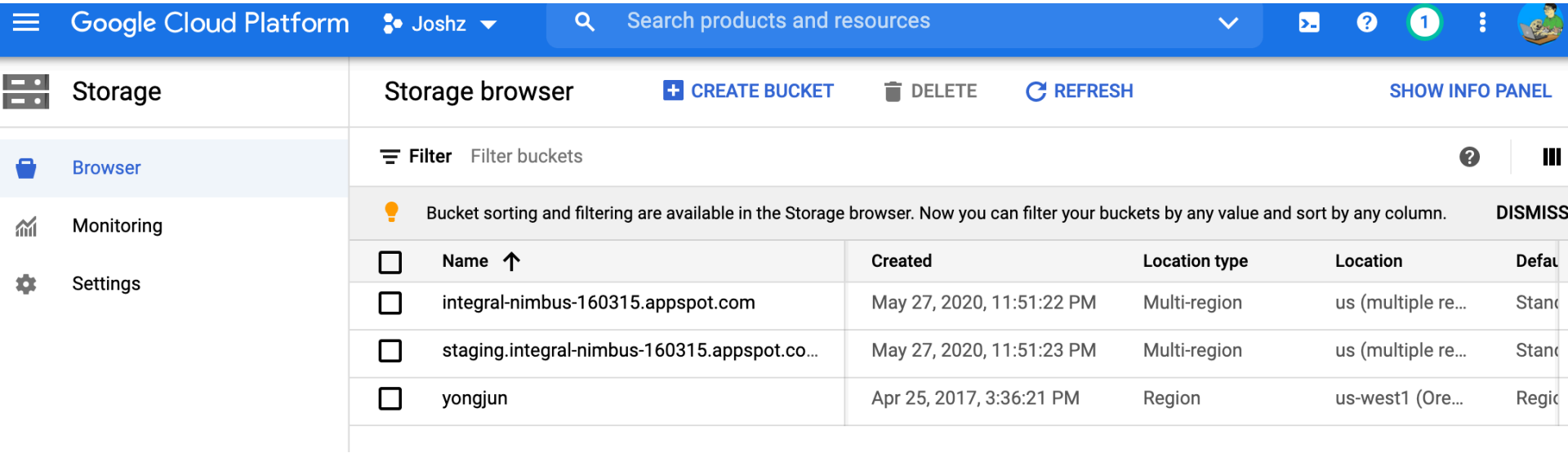
```
# TODO(developer): Set table_id to the ID of the destination table.
# PLS replace the "integral-nimbus-160315" with "Your own project id"
# also create a database "gdelt" to store our tables...
table_id_1 = "integral-nimbus-160315.gdelt.protest"
job_config_1 = bigquery.QueryJobConfig(destination=table_id_1)
# Perform the query
query_job_1 = client.query(QUERY1,job_config=job_config_1)  # API request
query_job_1.result()  # Waits for query to finish
print("Query results loaded to the table {}".format(table_id_1))
# So now we have a table sitting in our own big query. if you use your own browser to log into your google cloud, you should be able to
    see a table named protest under gdelt project...
```

Google Storage




Google Cloud Platform
Joshz
Search produ
BigQuery
FEATURES & INFO
SHORTCUT
Query history
Saved queries
Job history
Transfers
Scheduled queries
Reservations
BI Engine
Resources
ADD DATA
Search for your tables and datasets
eventmentions
eventmentions_partitioned
events
events_partitioned
gdg_partitioned
geg_g1
Query editor
1
Run
Save query
eventmentions_partitioned
This is a partitioned table. Learn mo
Schema
Details
Preview
Field name
Type
GLOBALEVENTID
INTEGE
EventTimeDate
INTEGE
MentionTimeDate
INTEGE
MentionType
INTEGE
MentionSourceName
STRING
MentionIdentifier
STRING


Export table to Google Cloud Storage
Select GCS location:
bucket/folder/file
Browse
Export format:
CSV
Compression:
None
Export
Cancel


Google Cloud Platform
Joshz
Search products and resources
Storage
Browser
Monitoring
Settings
Storage browser
CREATE BUCKET
DELETE
REFRESH
SHOW INFO PANEL
Filter
Filter buckets
Bucket sorting and filtering are available in the Storage browser. Now you can filter your buckets by any value and sort by any column.
DISMISS
Name
Created
Location type
Location
integral-nimbus-160315.appspot.com
May 27, 2020, 11:51:22 PM
Multi-region
us (multiple re...
staging.integral-nimbus-160315.appspot.co...
May 27, 2020, 11:51:23 PM
Multi-region
us (multiple re...
yongjun
Apr 25, 2017, 3:36:21 PM
Region
us-west1 (Ore...

CLOUD Vision API

https://cloud.google.com/vision/docs/libraries#client-libraries-install-python

Setting up authentication

To run the client library, you must first set up authentication by creating a service account and setting an environment variable. Complete the following steps to set up authentication. For other ways to authenticate, see the GCP authentication documentation.

Cloud Console | Command line

1. In the Cloud Console, go to the **Create service account key** page.

   Go to the Create Service Account Key page

2. From the **Service account** list, select **New service account**.
3. In the **Service account name** field, enter a name.
4. Don't select a value from the **Role** list. No role is required to access this service.
5. Click **Create**. A note appears, warning that this service account has no role.
6. Click **Create without role**. A JSON file that contains your key downloads to your computer.

```
import io
import os

# Imports the Google Cloud client library
from google.cloud import vision

# Instantiates a client
client = vision.ImageAnnotatorClient()

# The name of the image file to annotate
file_name = os.path.abspath('resources/wakeupcat.jpg')

# Loads the image into memory
with io.open(file_name, 'rb') as image_file:
    content = image_file.read()

image = vision.Image(content=content)

# Performs label detection on the image file
response = client.label_detection(image=image)
labels = response.label_annotations

print('Labels:')
for label in labels:
    print(label.description)
```

Google Maps Static API

Before you use Maps Statics API, you need to check here to get api key:
https://developers.google.com/maps/documentation/maps-static/get-api-key

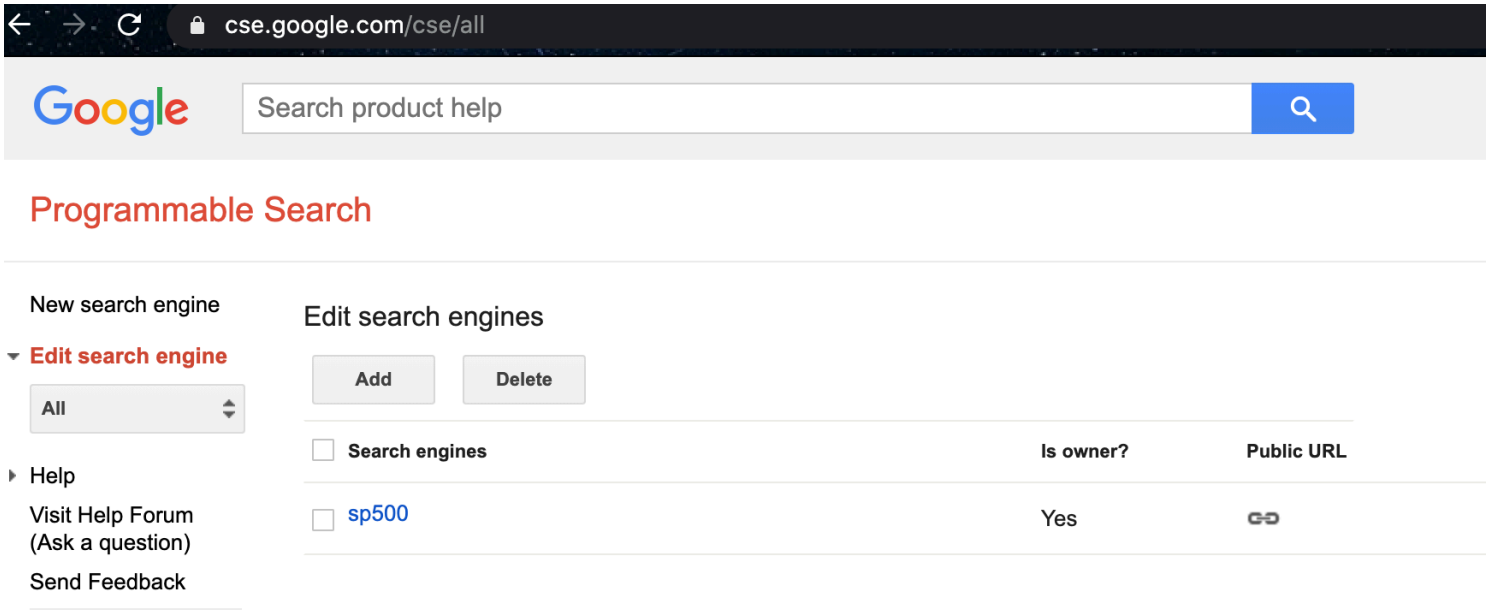
Add the API key to your request

You must include an API key with every Maps Static API request. In the following example, replace `YOUR_API_KEY` with your API key.

```
https://maps.googleapis.com/maps/api/staticmap?center=40.714%2c%20-
73.998&zoom=12&size=400x400&key=YOUR_API_KEY
```

```python
# Python program to get a google map
# image of specified location using
# Google Static Maps API
import requests
import os,re
import numpy as np
import pandas as pd

def google_sat_map(zoom,px,center,fig,save_path):
  # importing required modules
  import requests
  # Enter your api key here
  api_key = "YOUR API KEY"
  # url variable store url
  url = "https://maps.googleapis.com/maps/api/staticmap?"
  # return response object
  r = requests.get(url + "center=" + str(center) + "&scale=4&zoom=" +
                   str(zoom) + "&size="+str(px)+"x"+str(px)+"&maptype=satellite&key=" +
                            api_key)
  #save image
  f = open(str(save_path)+str(fig)+'.png', 'wb')
  f.write(r.content)
  f.close()

def get_cbg_data(file_path):
  cbg=pd.read_csv(file_path+"cbg.csv.gz",compression='gzip')
  cbg_center=cbg["center"]
  cbg_id=cbg["origin_census_block_group"]
  return cbg_center,cbg_id

def get_google_img_for_cbg(file_path,save_path):
  # get centers and ids
  cbg_centers,cbg_ids=get_cbg_data(file_path)
  # loop via centers and ids
  for i in range(0,cbg_centers.size):
    print(i,cbg_ids[i])
    google_sat_map(16,400,cbg_centers[i],cbg_ids[i],save_path)

file_path="/content/drive/My Drive/COVID19/temp/"
save_path="/content/drive/My Drive/COVID19/cbg-img/"
get_google_img_for_cbg(file_path,save_path)
```

Google Custom Search Engine API



```
def google_csa(gvkey,domain,start):
    print(gvkey,domain)
    url='https://www.googleapis.com/customsearch/v1?key=YOUR-KEY&YOUR-PROGRAM-SEARCH-
ID&q=site:'+str(domain)+'+covid19+OR+covid+OR+coronavirus+OR+corona'+'&num=10&start='+str(start)
    r = requests.get(url)
    rjson=json.loads(r.content)
    return rjson
```

Thank you!

Yongjun Zhang, Ph.D
Assistant Professor
Dept of Sociology and
Institute for Advanced Computational Science
Stony Brook University
Yongjun.Zhang@stonybrook.edu
https:yongjunzhang.com