Building your own IoT Assistant

@ElizaCamber
elizacamber.dev

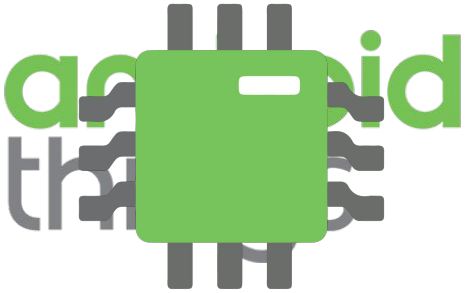
Women Techmakers Netherlands

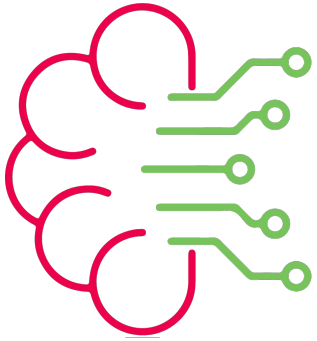
GDG Netherlands

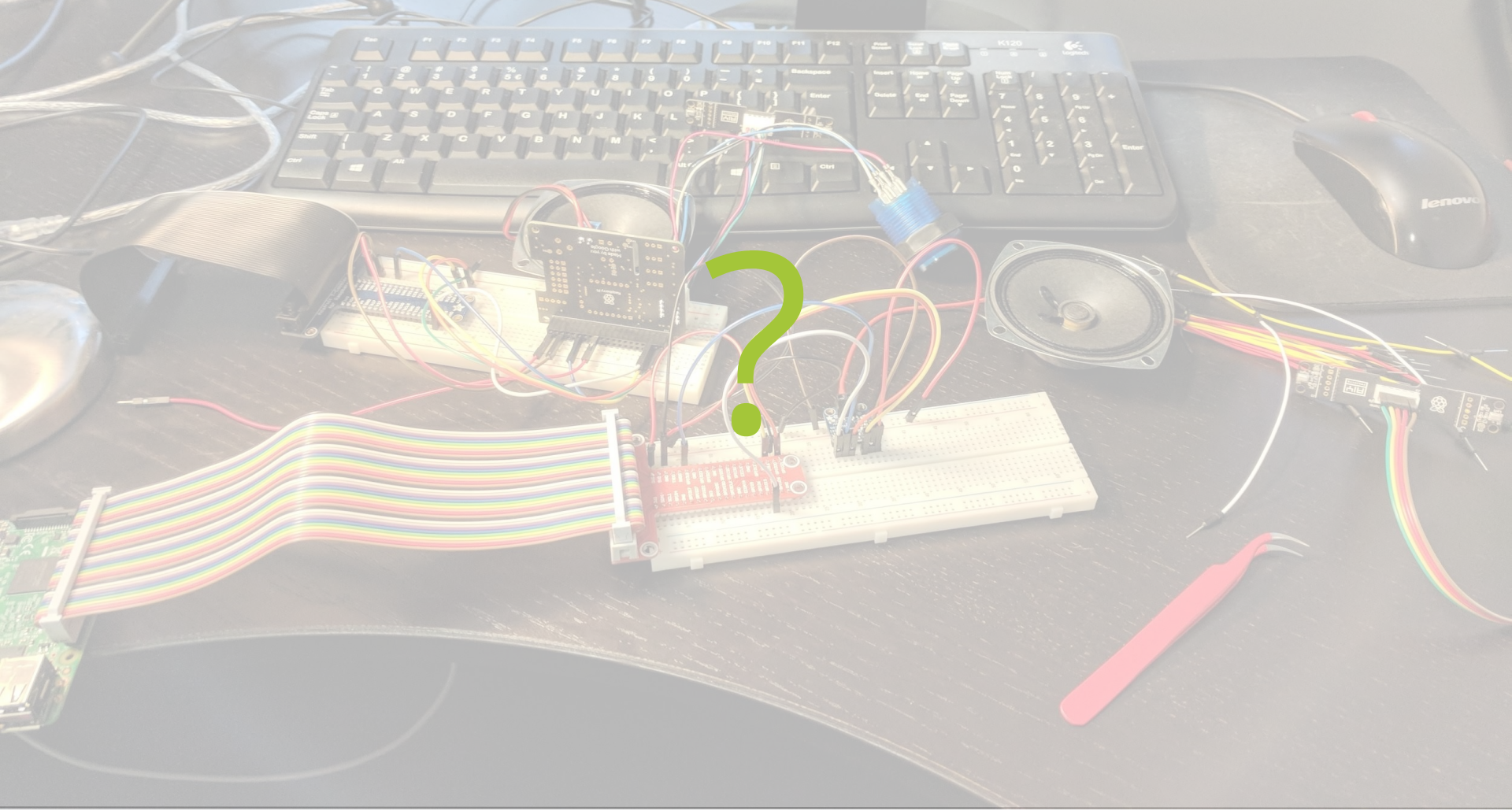
PIXPLICITY

Google Developers Experts


Agenda








TOM and JERRY
CLASSIC


LOONEY TUNES
™ & © Warner Bros. Entertainment Inc. (s18)


ONCE
upon a time...

Android Developers Blog

The latest Android and Google Play news for app and game developers.

Say Hello to Android Things 1.0

07 May 2018

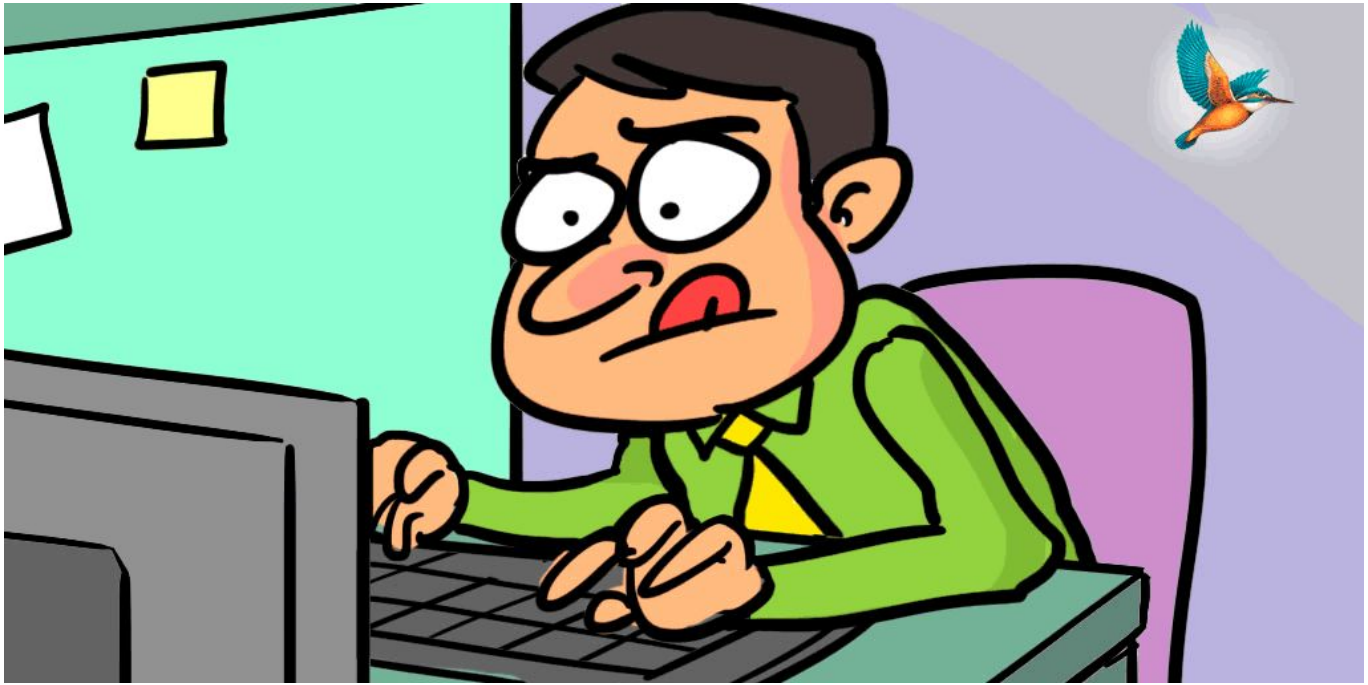
Posted by Dave Smith, Developer Advocate for IoT

Android Things is Google's managed OS that enables you to build and maintain Internet of Things devices at scale. We provide a robust platform that does the heavy lifting with certified hardware, rich developer APIs, and secure managed software updates using Google's back-end infrastructure, so you can focus on building your product.

After a developer preview with over 100,000 SDK downloads, we're releasing Android Things 1.0 to developers today with long-term support for production devices. Developer feedback and engagement has been critical in our journey towards 1.0, and we are grateful to the over 10,000 developers who have provided us feedback through the issue tracker, at workshop events, and through our Google+ community.

Powerful production hardware

Today, we are announcing support for new System-on-Modules (SoMs) based on the NXP i.MX8M, Qualcomm SDA212, Qualcomm SDA624, and MediaTek MT8516 hardware platforms. These modules are certified for production use with guaranteed long-term support for three years, making it easier to bring prototypes to market. Development hardware and reference designs for these SoMs will be available in the coming months.




LOONEY TUNES


(SPITS)

An Update on Android Things

12 February 2019

Posted by Dave Smith, Developer Advocate for IoT

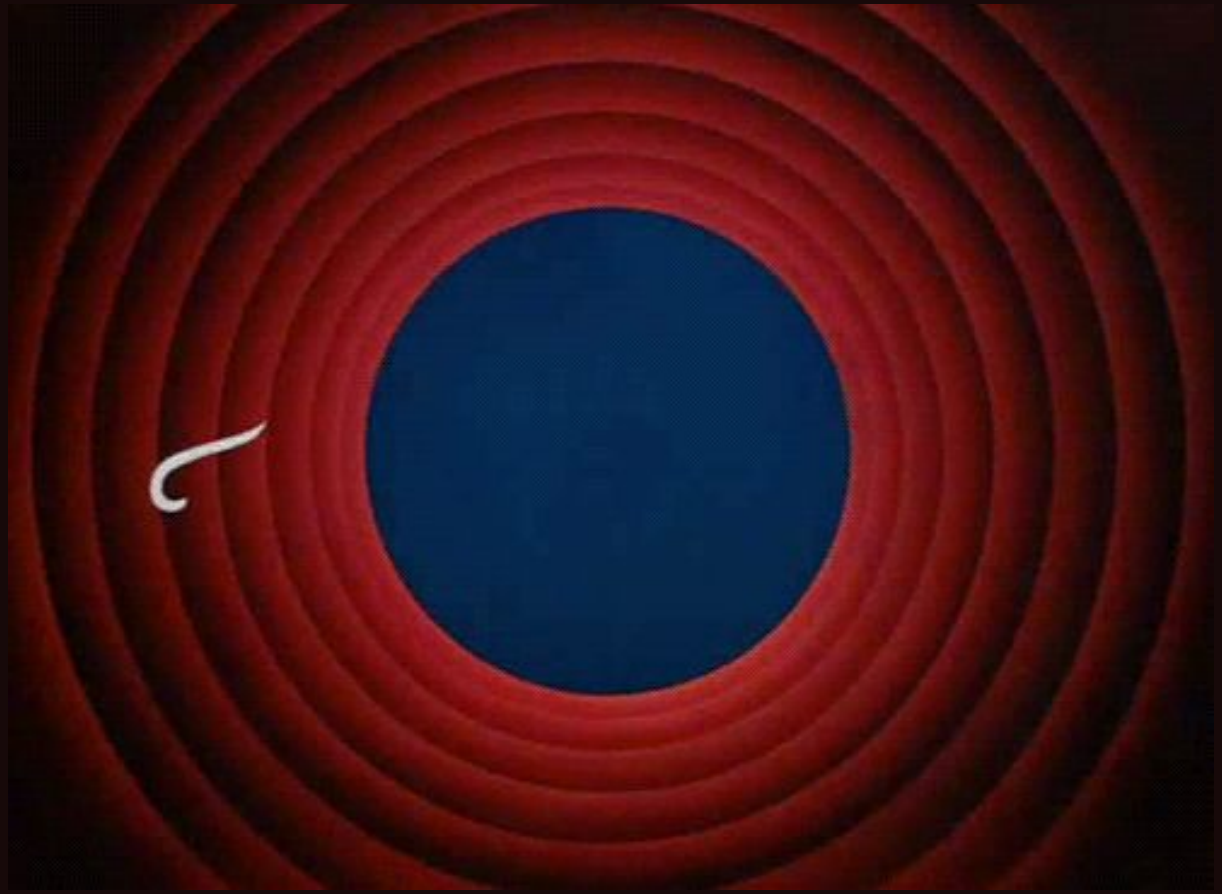
Over the past year, Google has worked closely with partners to create consumer products powered by Android Things with the Google Assistant built-in. Given the successes we have seen with our partners in smart speakers and smart displays, we are refocusing Android Things as a platform for OEM partners to build devices in those categories moving forward. Therefore, support for production System on Modules (SoMs) based on NXP, Qualcomm, and MediaTek hardware will not be made available through the public developer platform at this time.

Android Things continues to be a platform for experimenting with and building smart, connected devices using the Android Things SDK on top of popular hardware like the NXP i.MX7D and Raspberry Pi 3B. System images for these boards will remain available through the Android Things console where developers can create new builds and push app updates for up to 100 devices for non-commercial use.

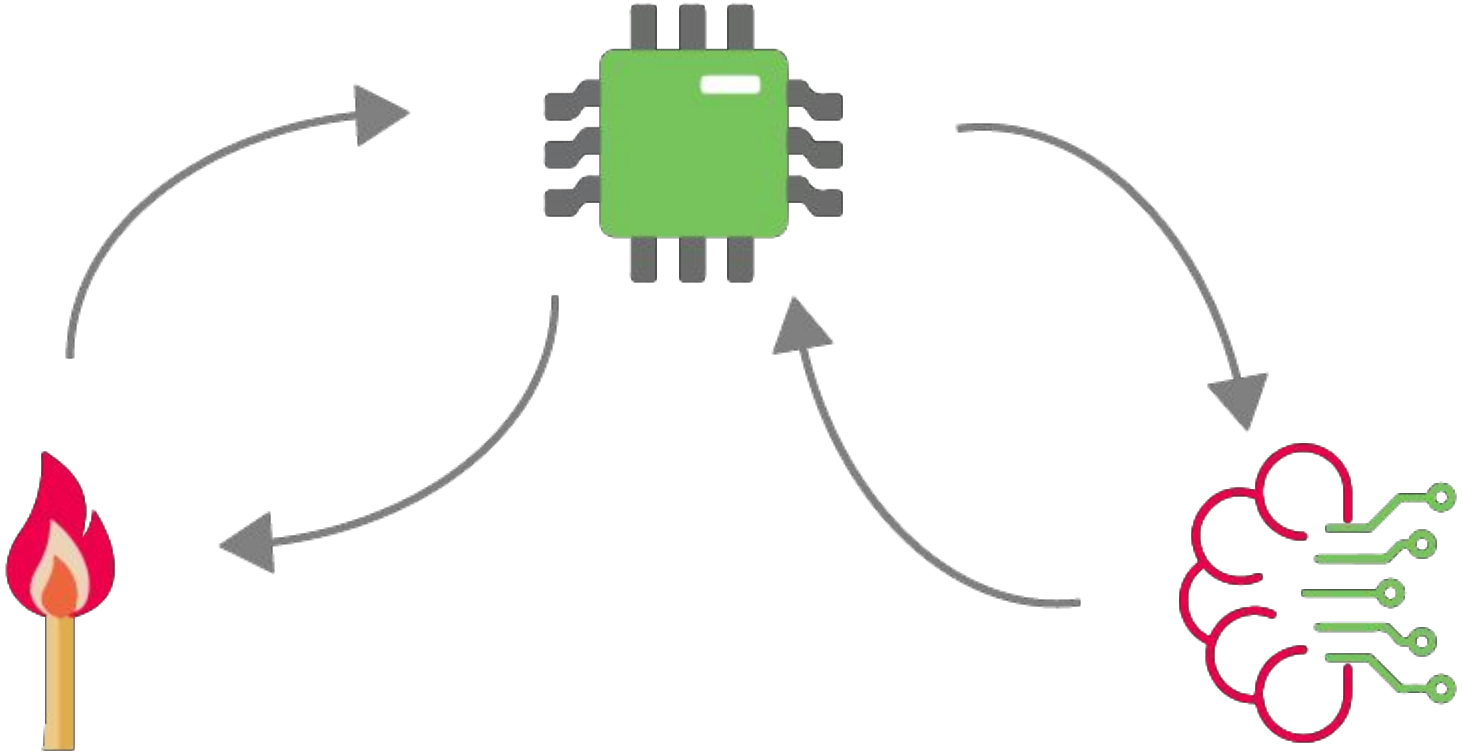
We remain dedicated to providing a managed platform for IoT devices, including turnkey hardware solutions. For developers looking to commercialize IoT products in 2019, check out Cloud IoT Core for secure device connectivity at scale and the upcoming Cloud IoT Edge runtime for a suite of managed edge computing services. For on-device machine learning

9 months == 3 years

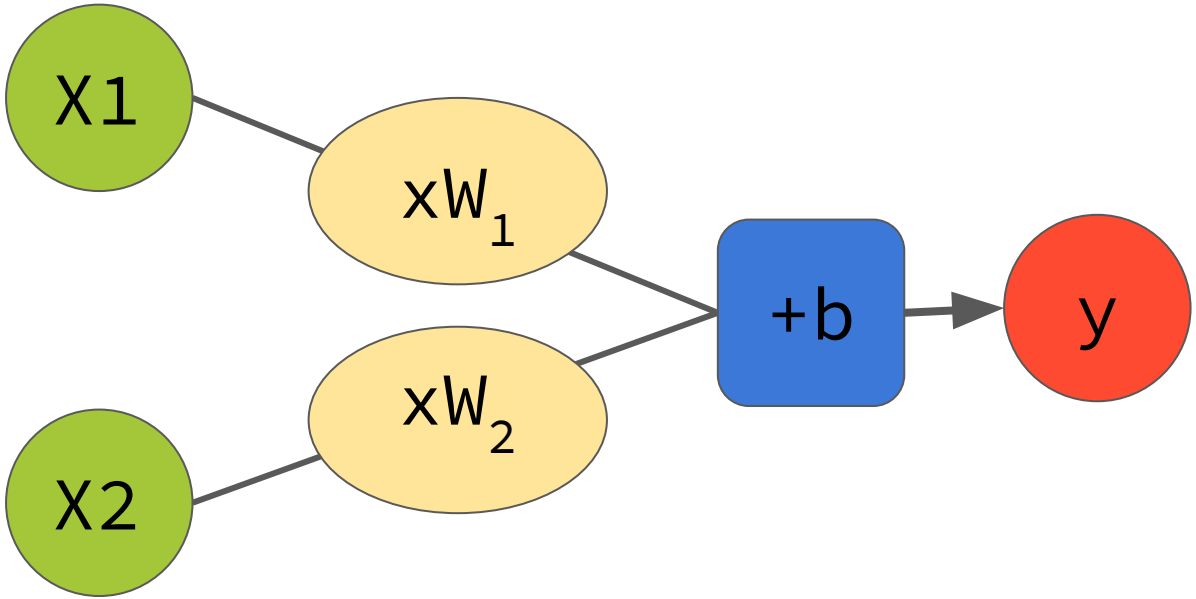

LOONEY TUNES
™ & © Warner Bros. Entertainment Inc. (s18)




FLOW



Neural network



TensorFlow

Covering...

- Basics
- Preparation
  - Importing and cleaning data
  - Feature selection and engineering
  - Value representation
- Tensorflow model
  - parameters
  - Regularization
  - (logistic) regression and classification
- Convolutional neural networks for audio & image classification

TensorFlow basics

Tensors

- Scalar → 0-d array → 0th-order tensor → 5
- Vector → 1-d array → 1st-order tensor → [2,5]
- Matrix → 2-d array → 2nd- order tensor → [[2,5][3,6]]
- n-Tensor → n-d array → nth-order tensor

Tensor shape → used to characterize the size and number of dimensions of a tensor. It's expressed as a list, where the length indicates the rank of the tensor

Tensor rank vs shape

image_a

(colored image)

- Rank = number of dimensions

  image_A rank → [height, width, depth] →

  rank 3

- Shape = detailed number of components

  in each dimension

  image_A shape → [600,400,3]

Tensor rank vs shape

```
a1=tf.constant(
[
     [
          [1,2,3,4,5],
          [1,2,3,4,5],
          [1,2,3,4,5],
          [1,2,3,4,5]
     ]
])
```

Rank → number of open square braces

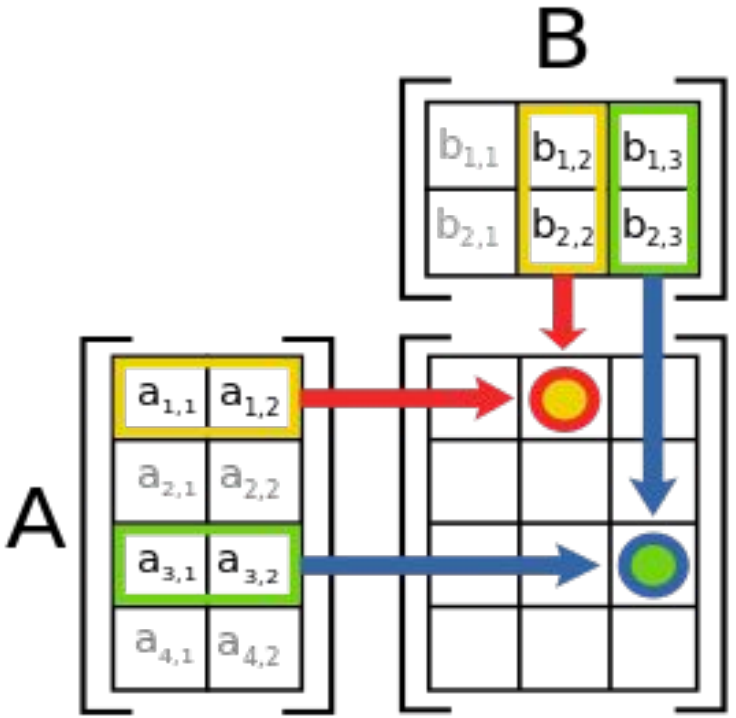
Shape → format (x,y,z)

Where

x → Number of commas in the first brace plus 1 (so 1)
y → rows (so 4)
z → columns (so 5)

Basic Tensor manipulation

- Broadcasting
- Multiplication
- Reshaping → Eg: (8,2) → (2,2,4)



TensorFlow basics

Model types :

- Regression → output is a number
- Classification → output is a category

Feature and feature columns

Feature/labels = a measurable property or characteristic

Types of data

Categorical - Numerical - bucketized

1. Importing data

`tf.data API`

- tf.data.Dataset
- tf.data.Iterator

2. Cleaning data

- **Scale feature values** → convert floating-point from their natural range into a standard range
- **Handling extreme outliers** → clipping
- **Binning** → map nonlinearities by grouping into bins (bucketizing)
- **Scrubbing** → omitted values, duplicate examples, bad labels, bad feature values

Feature engineering

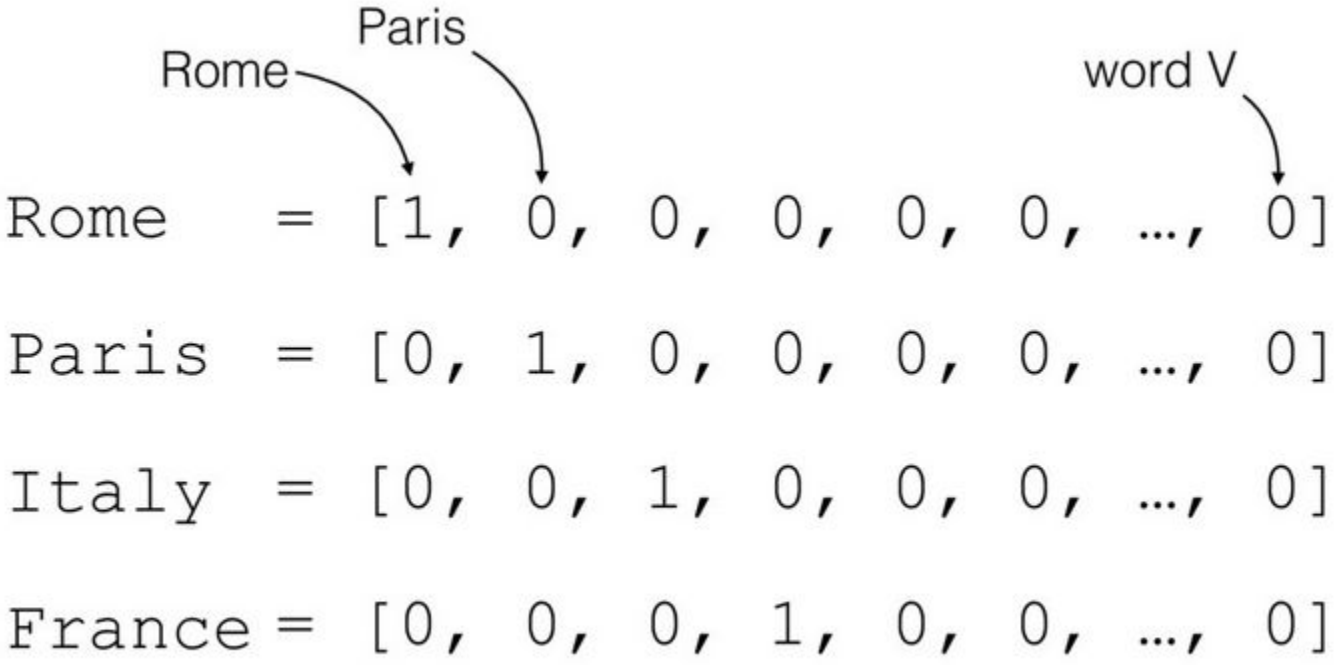
Transform raw data to feature vector

- **Real-valued features** → copy directly

  Num_wheels: 2 ----------> num_wheels_feat = [2.0]

- **String features** → one-hot encoding or multi-hot encoding handling

Feature engineering example



Tensorflow parameters

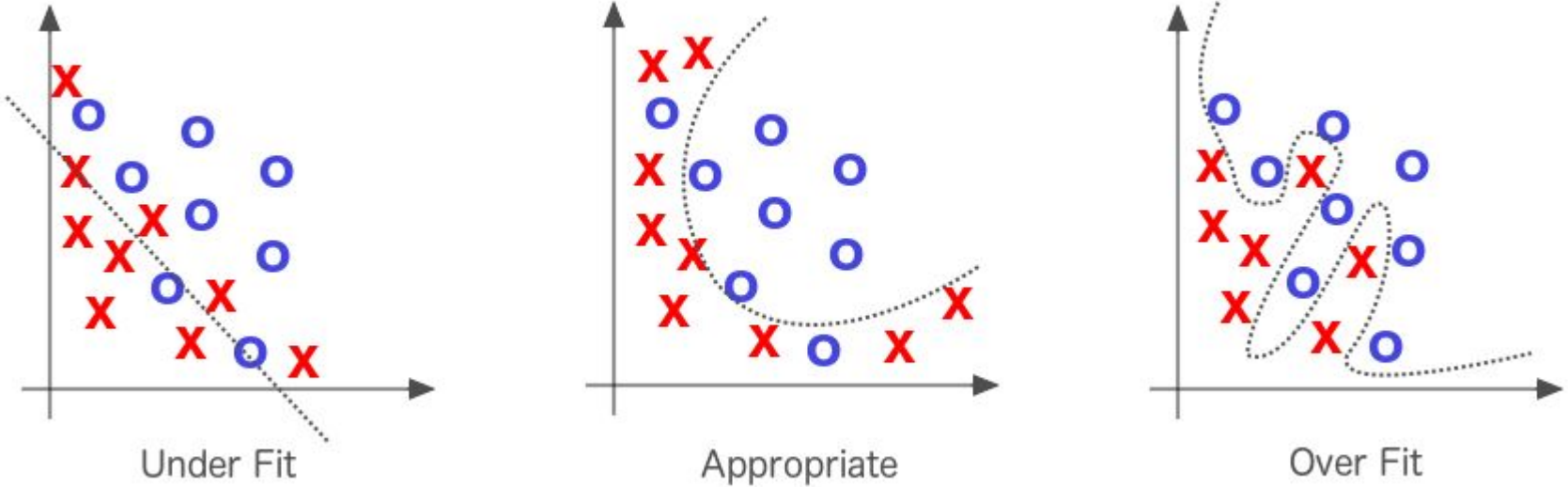
Batch size → the number of examples (chosen at random) for a single step

Steps → total number of training iterations. →

```
total number of trained examples = batch size x steps
```

Periods → controls reporting output

Underfit - Overfit



Overfitting prevention

- Cross-validation
- Train with more data
- Remove features
- Early stopping
- Regularization
- Ensembling

Regularization → Penalize complex models

λ= regularization rate

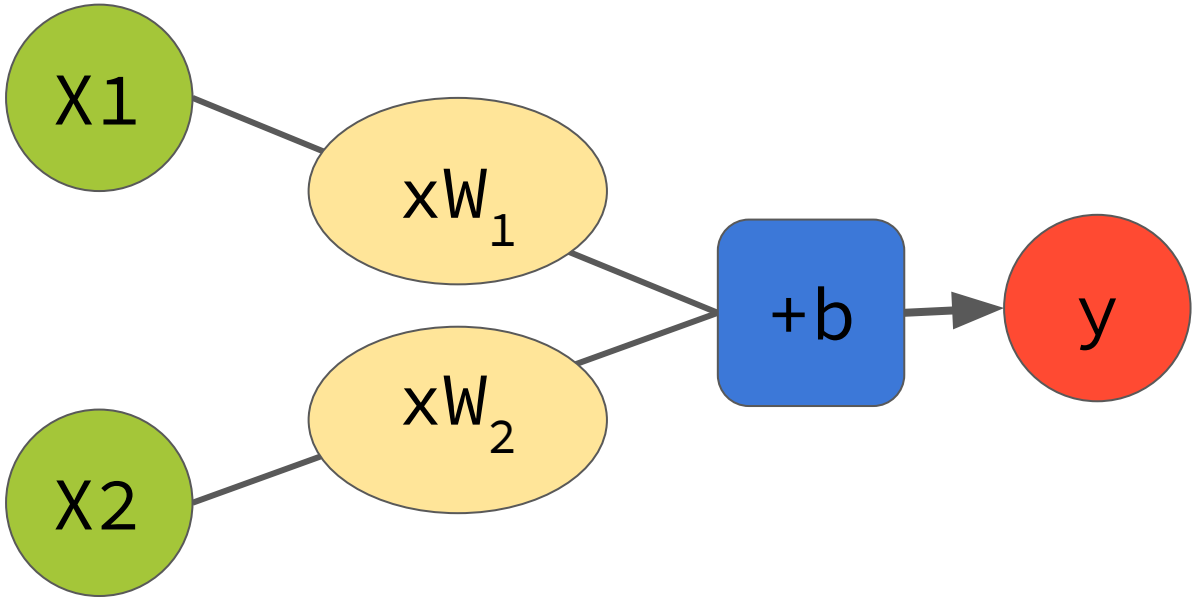
λ too low → risk of underfitting → won't learn enough

λ too high → risk of overfitting → won't be able to generalize

L_1 → penalizes $weight^2$

L_2 → penalizes $|weight|$

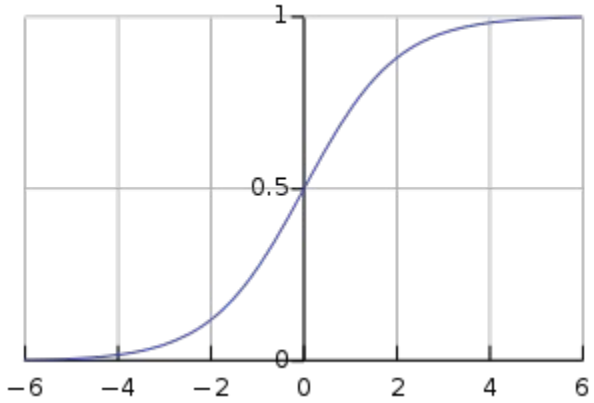
Neural network



(Logistic) Regression

Instead of predicting exactly 0 or 1, logistic regression creates a probability, a value between 0 and 1.

Sigmoid function → maps linear values to probabilities



Classification

Confusion matrix

True positives (TP)	False positives (FP)
False negatives (FN)	True negatives (TN)

Example:

TP = 1 FP = 1 → Accuracy = (TP+TN)/ (TP+TN+FP+FN) = 91/100 = 91%

FN = 8 TN = 90 → Precision = TP/(TP+FP) = ½ = 50%

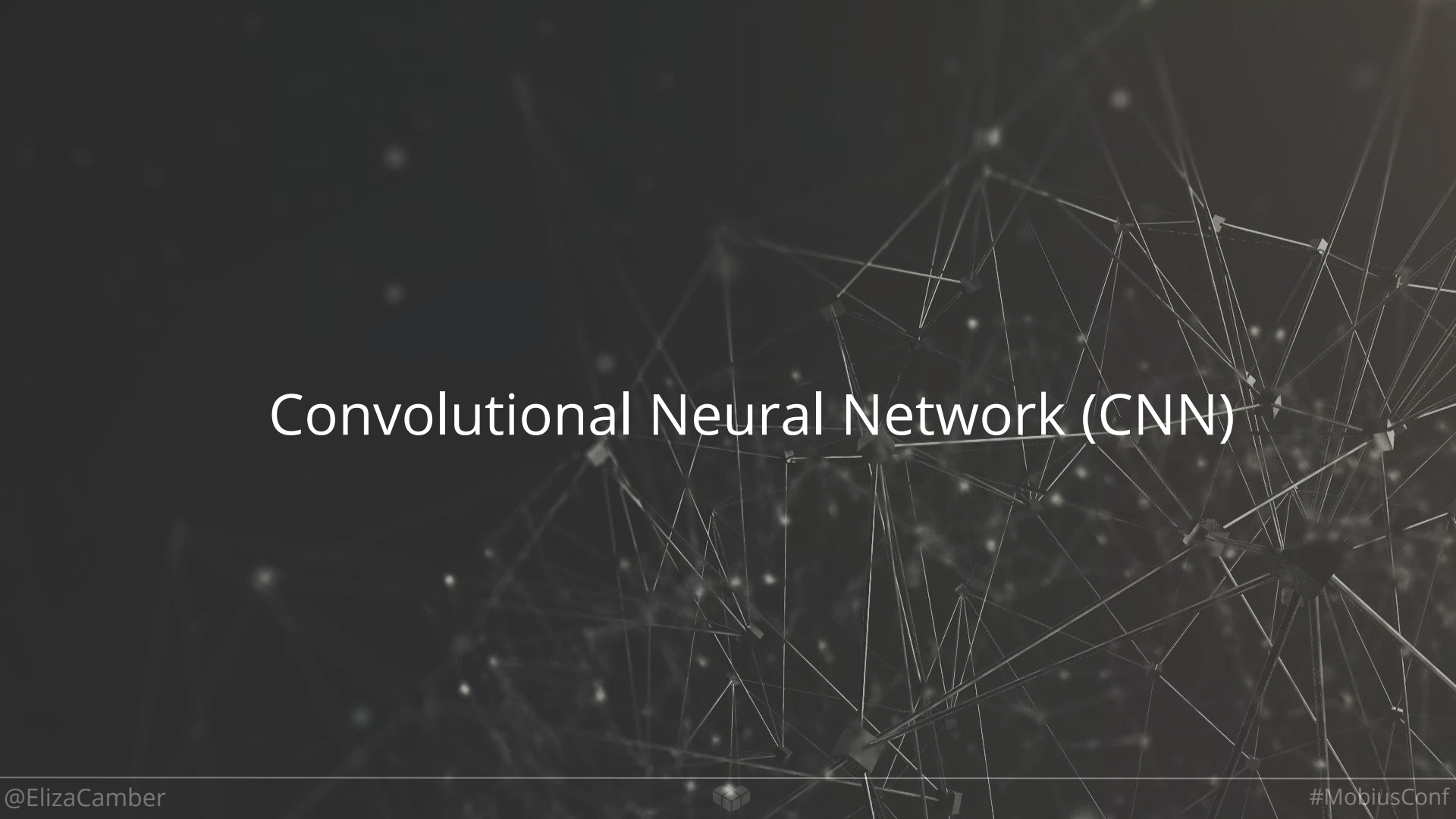
→ Recall = TP/ (TP+FN) = 1/9 = 11%

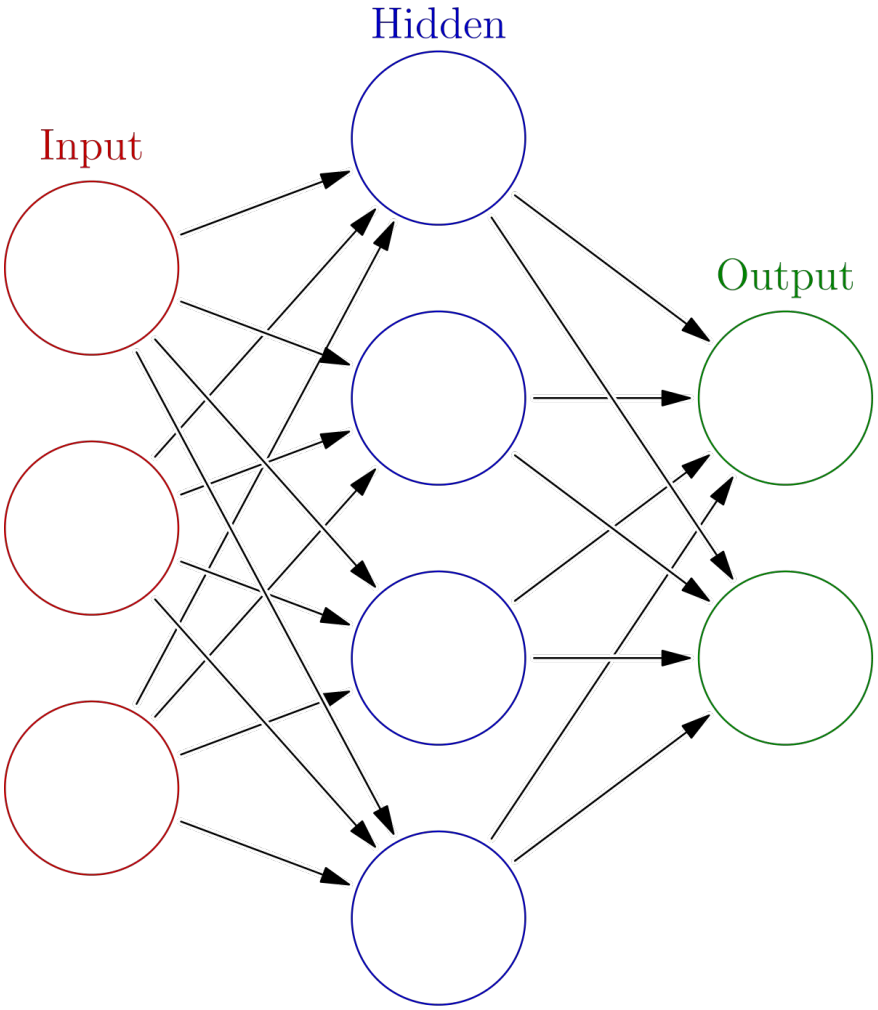
False Positive vs False Negative

Model tuning (reducing error)

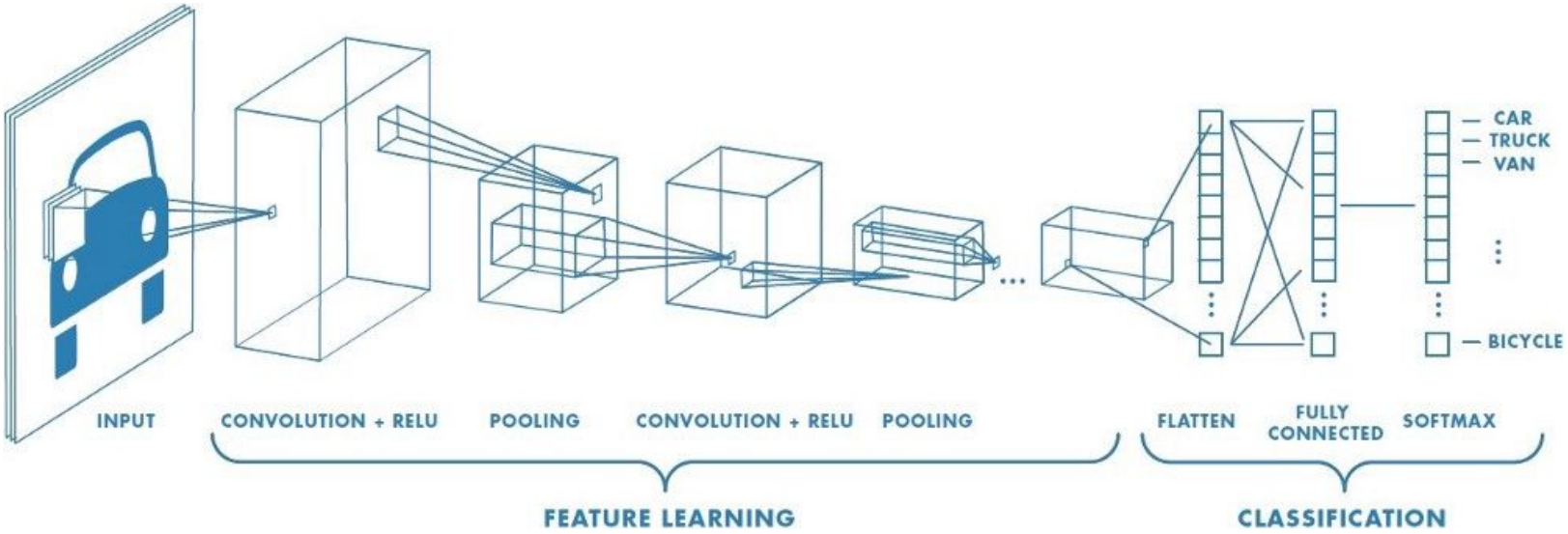
- Training error should steadily decrease, steeply at first and should eventually plateau as training converges
- If the error hasn't converged, try running it for longer
- If the training error reduces too slowly, increase the learning rate
- If the training error varies wildly, decrease the learning rate
- Very small batch sizes can also cause instability. First try large values and decrease until you see degradation
- Clip outliers

Convolutional Neural Network (CNN)




Hidden
Input
Output

Convolutional neural networks



Applying the convolution filter

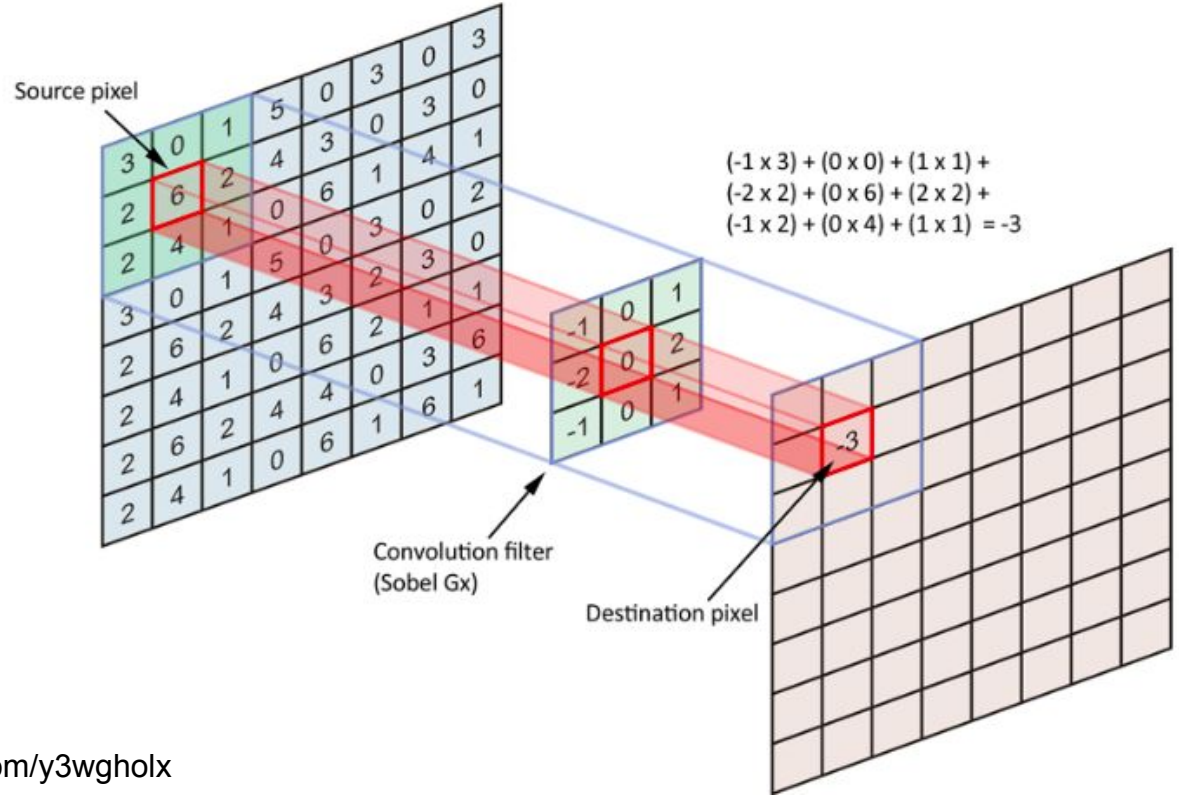
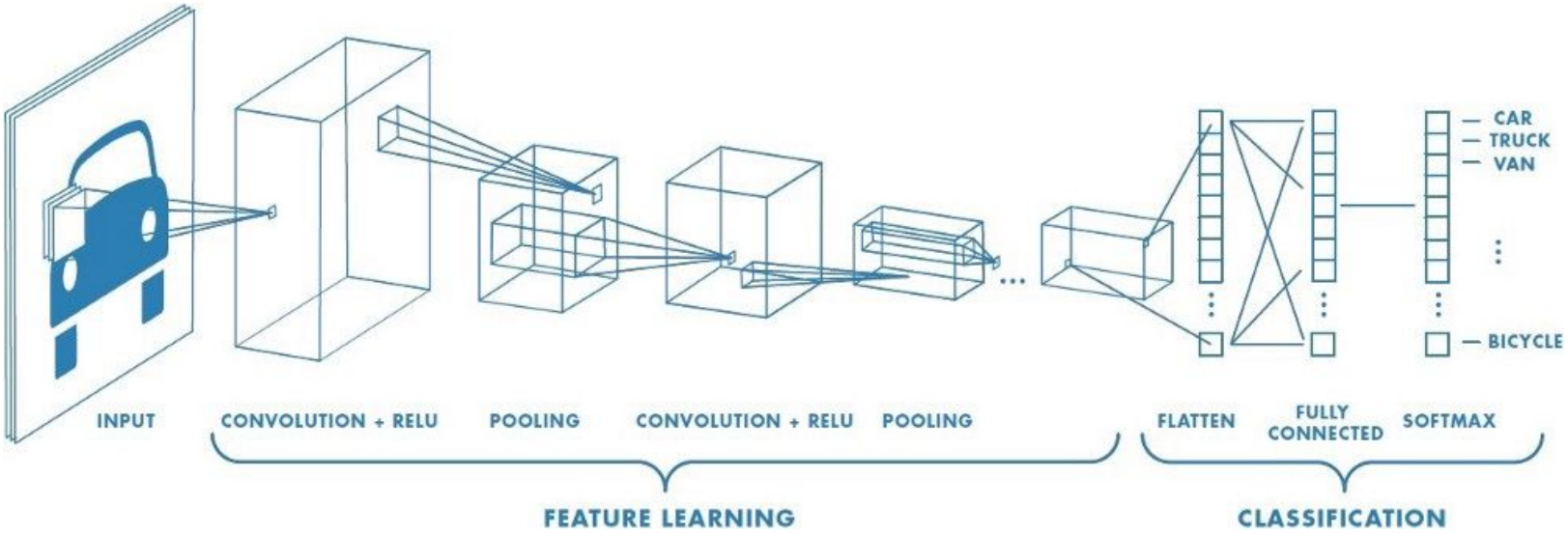


Image credits: tinyurl.com/y3wgholx

Convolutional neural networks



Max pooling

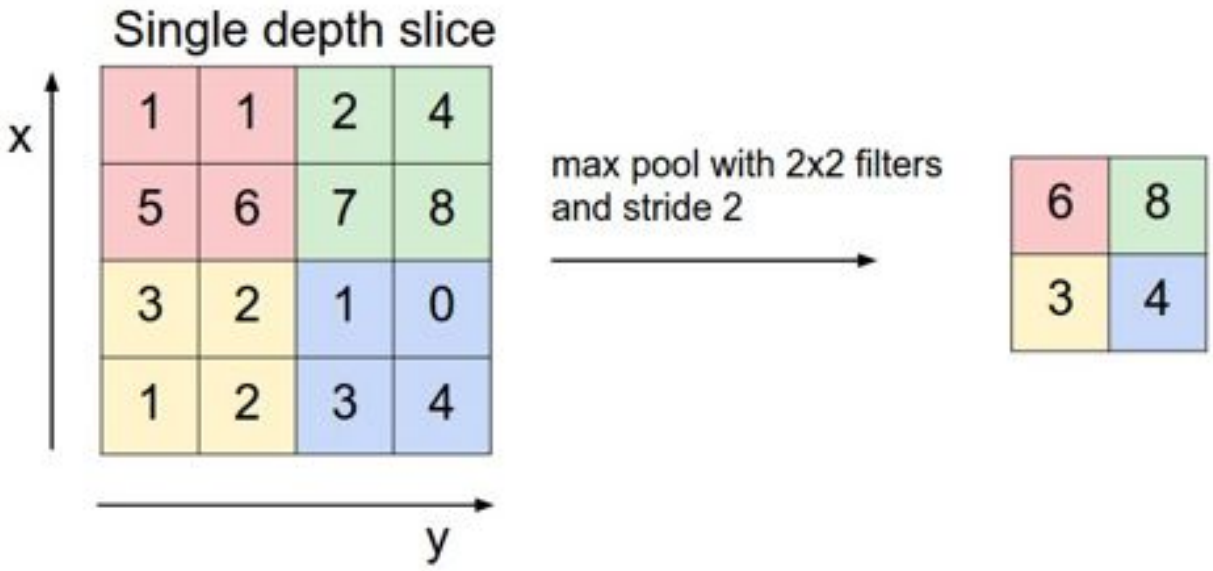


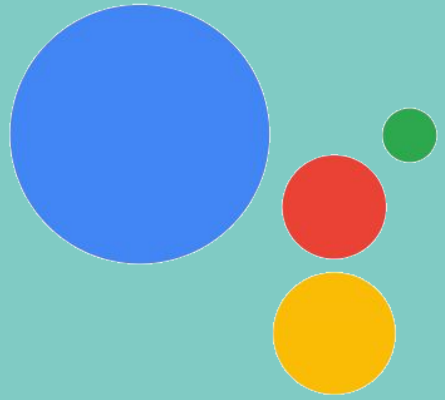
Image credits: cs231n.github.io/convolutional-networks

NLU

→ Intro to DF

→ How intents and entities work

→ Watson vs DialogFlow



Welcome to your project, **Smart-Home!**

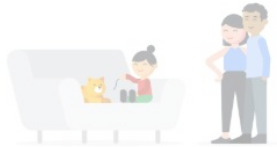
Get started on building by choosing a development experience

Games & fun

Play trivia games
Tell jokes
Get my fortune

Home control

Control lighting
Control appliances & televisions
Control home security

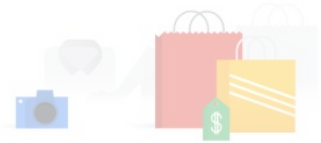


Kids & family

Tell a story
Play trivia
Take a quiz

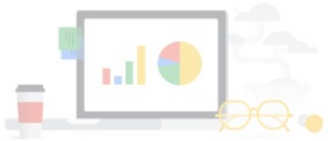
Food & drink

Find food recipes
Order food
Get nutrition facts



Shopping

Manage shopping lists



Business & finance

Check financial markets

Health & fitness

Get info about health & medicine

Social & communication

Make phone calls

Create Smart Home action

Smart Home actions allow your users to control IoT (Internet of Things) devices through the Google Assistant. Building Smart Home apps lets you Sync, Query and Execute actions through your existing cloud infrastructure.

Visit the documentation to get started. Once you're ready, just add a fulfillment URL below.

Add fulfillment URL

https://

Related Github Samples

Smart home example

DONE

actions-on-google / smart-home-nodejs

Watch 61

Code | Issues 87 | Pull requests 3 | Projects 0 | Wiki | Insights

A sample of the Smart Home device control APIs in Actions on Google

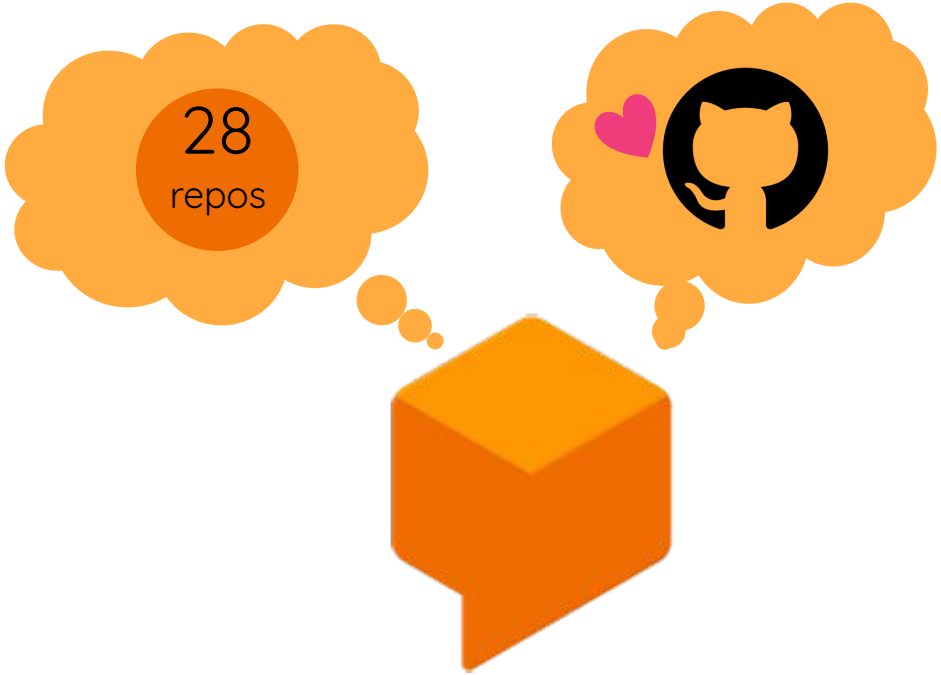
actions-on-google / smart-home-dashboard

Watch 12

Code | Issues 9 | Pull requests 0 | Projects 0 | Wiki | Insights

Report State Dashboard

This is a dashboard that developers can use to help them debug their smart home actions. It will obtain data from the HomeGraph and present it to you, allowing you to verify the data that is stored in it.


28
repos

intent

“I want walking directions to MobiusConf”

entity

Demo


REWIND

THANK YOU!

Questions?

@ElizaCamber
elizacamber.dev