

NYC taxicab drivers

**Can we use reinforcement learning to
maximize daily revenue?**

Agenda

- Introduction
- Overview of the data
- Architectures used for each approach
- Results pre/post training
- Live demo!
- Paths taken by each algorithm

1

Introduction



- Source : <http://www.nycbikemaps.com/wp-content/uploads/2008/04/2008-map-top.jpg>

0 NW	1 N	2 NE
3 W	4 Stay	5 E
6 SW	7 S	8 SE

Imagine you are a taxicab driver in NYC. What is your 'policy' to earn the most money each day?

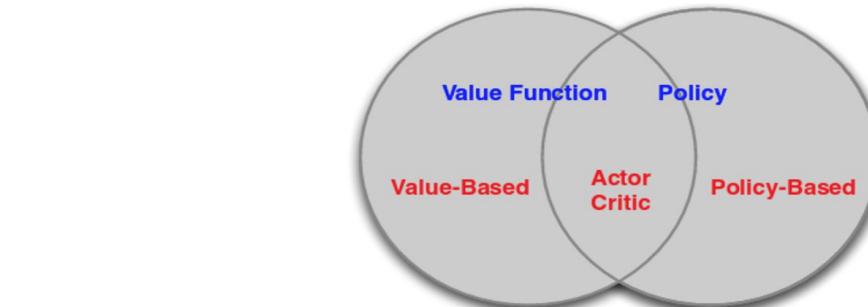
Three major types of reinforcement learning

Value Function

- Deep Q-learning *
- Approximate the long run utility each state to choose action

Policy based

- Policy gradients
- Update action gradients once reward is seen



- Source : <https://yanpanlau.github.io/img/torcs/actor-critic.png>

Hybrid

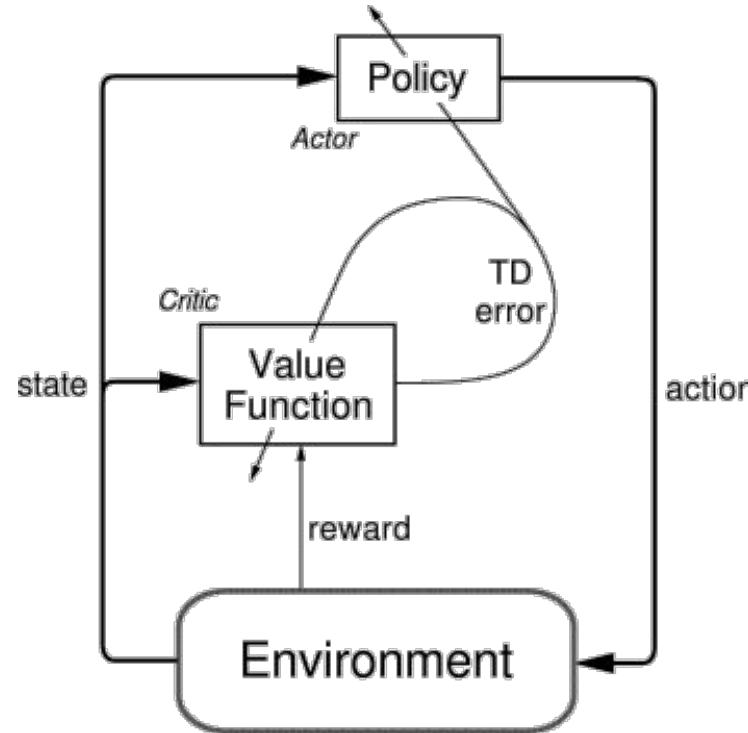
- Actor-Critic *
- Combine Q-learning with 'supervision' from a critic

* Implemented in this project

Pseudocode for Actor Critic

Overview

- Actor makes a move to state_1 based on the state_0
- Critic predicts value of the v_state_0 and value of v_state_1.
- The value of the actor's action is updated using a modified($v_{state_1} - v_{state_0}$).



Overview of each technique used

DQN

- Use a neural network to approximate the Q-function

$$Q_{t+1}(s_t, a_t) = \underbrace{Q_t(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha_t(s_t, a_t)}_{\text{learning rate}} \times \left[\underbrace{R_{t+1} + \gamma \max_a Q_t(s_{t+1}, a)}_{\substack{\text{reward} \\ \text{discount factor} \\ \text{estimate of optimal future value}}} - \underbrace{Q_t(s_t, a_t)}_{\text{old value}} \right]$$

high values = pleasure
high values = pleasant anticipation
low values = pain
low values = fear

▪ Source: <http://reducing-suffering.org/wp-content/uploads/2015/01/Q-learning-with-labels.png>

Actor Critic

- Two networks
- 1) Actor uses Q-function
- 2) Critic looks at temporal difference error of actor's move

$$\delta_k = r_{k+1} + \gamma V_{\theta_k}(x_{k+1}) - V_{\theta_k}(x_k).$$

▪ Source: <http://www.jianshu.com/p/03cf10cbf93a>

Temporal Difference

- Key innovation to allow neural networks to be applied to RL is **experience replay**

2

Overview of the data

NYC Yellow Cab Data

- Looked at data from January 2016 ~ 10 million trips

tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	pickup_longitude	pickup_latitude	geohash_pickup	jan_day	jan_minute	geohash_dropoff	trip_time_minutes
2016-01-01 00:00:00	2016-01-01 00:00:00	2	1.10	-73.990372	40.734695	Zct8X	1	0	dr5rsx	10
2016-01-01 00:00:00	2016-01-01 00:00:00	5	4.90	-73.980782	40.729912	Zct8c	1	0	dr5rtk	30

- Data Cleaning

- ▷ Rounded each trip to nearest 10 minutes (8.6 mph average speed in NYC)
- ▷ Converted latitude and longitude to geohash (~ 3k unique geohashes)

- Final Data Structure

- ▷ {Time: {geohash_dropoff : [(fare_1, time_1, **fare_1/time_1***)]} } *Objective for RL algorithm

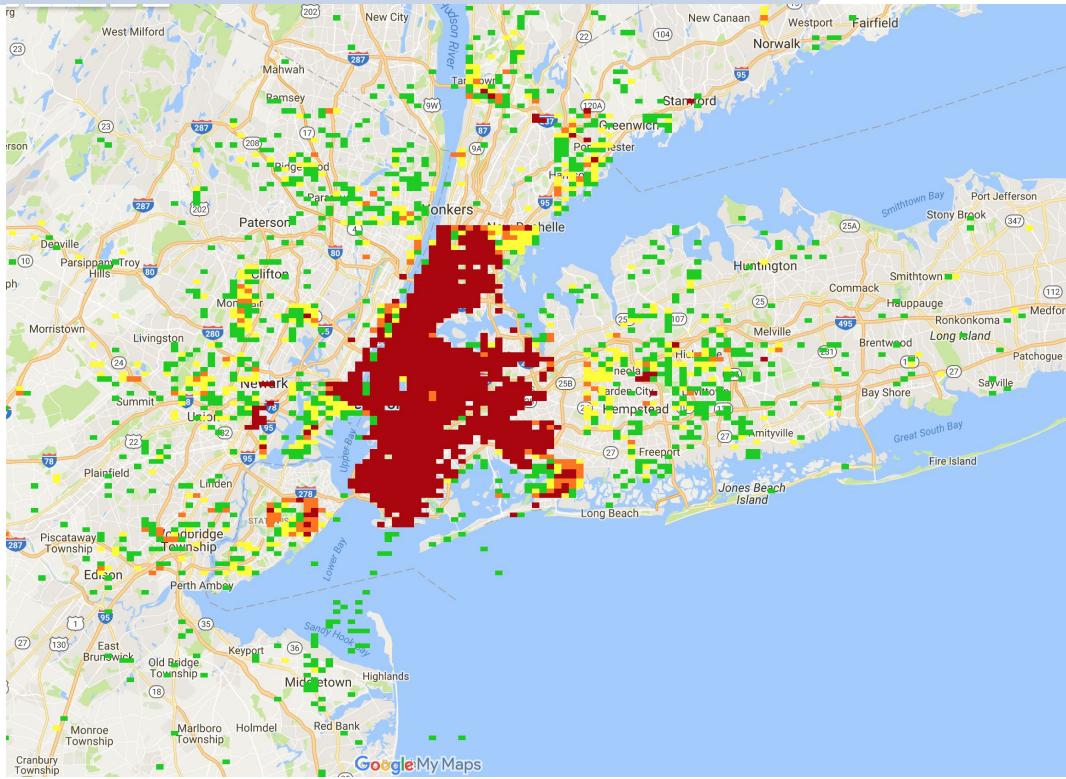
- Input: [input_time (i.e. 110), geohash_index (i.e. 0)]

- Output: Utility estimate for each of the nine possible moves

- Total possible moves per day:

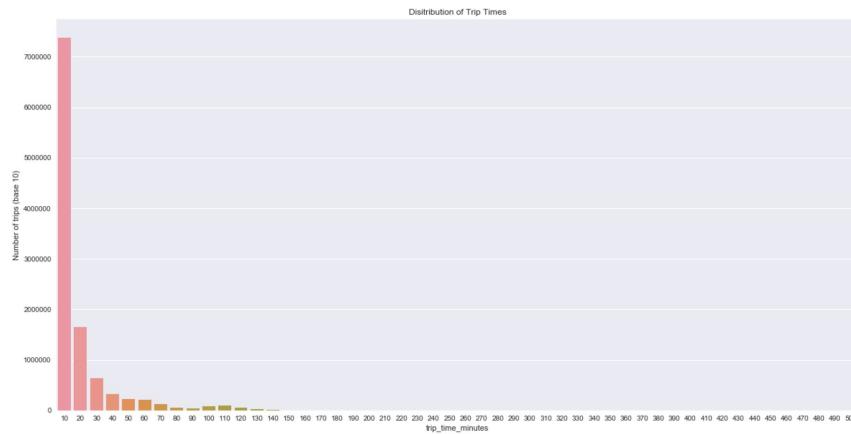
2,318,269,218,083,089,908,234,280,468,459,894,595,445,540,070,540,355,238,386,428,052,606,631,732,586,480,323,016,481,331,7
02,297,098,820,479,338,186,895,528,168,360,391,078,014,797,448

NYC Fares January '16

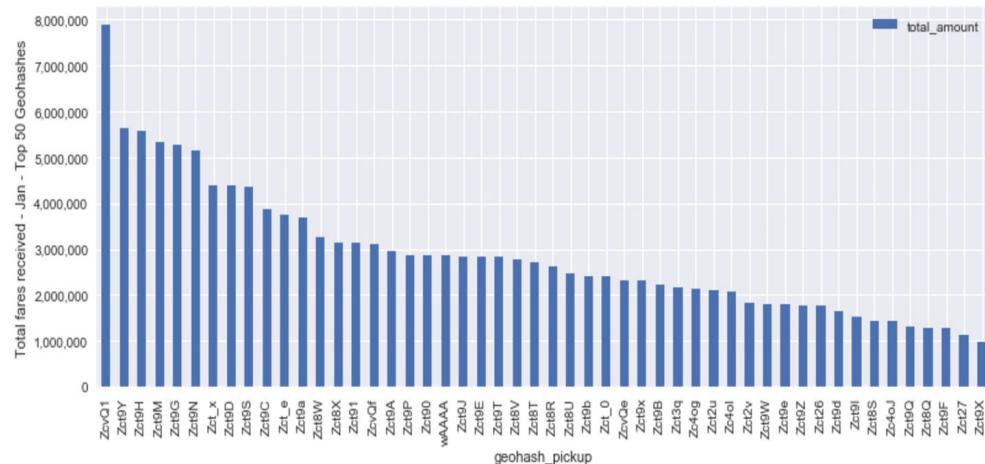


Exploratory Analysis

Trip times



Total fares in top geohashes



- 1) JFK
 - 2) Midtown - MoMa
 - 3) Midtown - Times Square

3

Model Architecture

DQN and Actor Critic

Hyperparameter Selection - Hyperas

- Easy hyperparameter selection
- Can also search over model architectures
- Uses random search

```
model_lstm = Sequential()
model_lstm .add(LSTM({{choice([64, 126, 256, 512, 1024])}}, dropout={{uniform(0, .5)}},
                     batch_input_shape=[1,x_train.shape[1], 2],
                     recurrent_dropout={{uniform(0, .5)}},return_sequences = True))
model_lstm.add(BatchNormalization())
condition = conditional({{choice(['one','two','three', 'four'])}})

if condition == 'one':
    pass
elif condition == 'two':
    model_lstm .add(LSTM({{choice([64, 126, 256, 512, 1024])}}, dropout={{uniform(0, .5)}},
                          recurrent_dropout={{uniform(0, .5)}}))
    model_lstm.add(BatchNormalization())
elif condition == 'three':
    model_lstm .add(LSTM({{choice([64, 126, 256, 512, 1024])}}, dropout={{uniform(0, .5)}},
                          recurrent_dropout={{uniform(0, .5)}}))
    model_lstm.add(BatchNormalization())
    model_lstm.add(Dense({{choice([126, 256, 512, 1024])}}))
    model_lstm.add(BatchNormalization())
    model_lstm.add(Activation({{choice(['relu','tanh','sigmoid'])}}))
elif condition == 'four':
    model_lstm .add(LSTM({{choice([64, 126, 256, 512, 1024])}}, dropout={{uniform(0, .5)}},
                          recurrent_dropout={{uniform(0, .5)}}))
    model_lstm.add(BatchNormalization())
    model_lstm.add(Dense({{choice([126, 256, 512, 1024])}}))
    model_lstm.add(BatchNormalization())
    model_lstm.add(Activation({{choice(['relu','tanh','sigmoid'])}}))
    model_lstm.add(Dense({{choice([126, 256, 512, 1024])}}), activation='relu'))
    model_lstm.add(BatchNormalization())
    model_lstm.add(Activation({{choice(['relu','tanh','sigmoid'])}}))

model_lstm .add(Dense(9, activation='linear',name='dense_output'))
adam = Adam(clipnorm=.5, clipvalue=.5)
model_lstm .compile(loss='mean_squared_error', optimizer=adam)
model_lstm.summary()
```

Architectures used

DQN/AC - MLP

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 100)	300
batch_normalization_1 (Batch Normalization)	(None, 100)	400
activation_1 (Activation)	(None, 100)	0
dropout_1 (Dropout)	(None, 100)	0
dense_2 (Dense)	(None, 500)	50500
batch_normalization_2 (Batch Normalization)	(None, 500)	2000
activation_2 (Activation)	(None, 500)	0
dropout_2 (Dropout)	(None, 500)	0
dense_3 (Dense)	(None, 1000)	501000
batch_normalization_3 (Batch Normalization)	(None, 1000)	4000
activation_3 (Activation)	(None, 1000)	0
dropout_3 (Dropout)	(None, 1000)	0
dense_4 (Dense)	(None, 9)	9009

Total params: 567,209.0
Trainable params: 564,009.0
Non-trainable params: 3,200.0

For actor-critic model, both the actor and the critic used the architecture above with different outputs

DQN - LSTM

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(1, None, 512)	1054720
batch_normalization_10 (Batch Normalization)	(1, None, 512)	2048
lstm_2 (LSTM)	(1, None, 1024)	6295552
batch_normalization_11 (Batch Normalization)	(1, None, 1024)	4096
dense_13 (Dense)	(1, None, 512)	524800
batch_normalization_12 (Batch Normalization)	(1, None, 512)	2048
activation_10 (Activation)	(1, None, 512)	0
dense_output (Dense)	(1, None, 9)	4617

Total params: 7,887,881.0
Trainable params: 7,883,785.0
Non-trainable params: 4,096.0

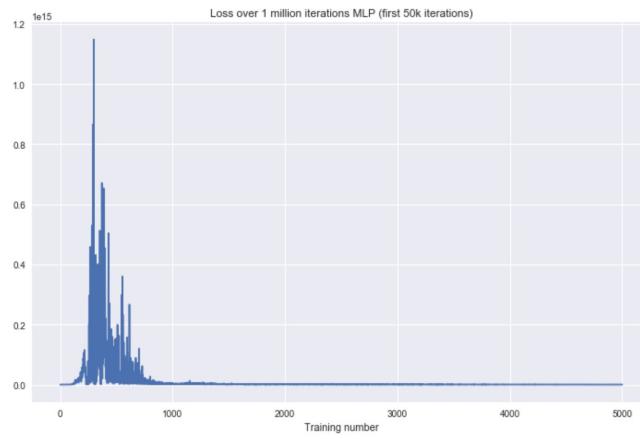
LSTM didn't take in a sequence during training due to experience replay

4

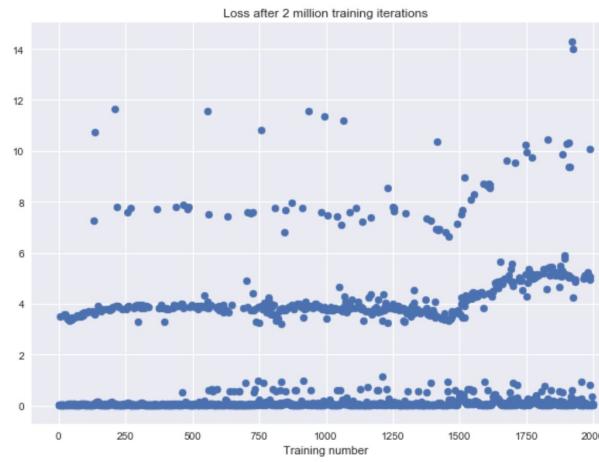
Results

Loss Curves - DQN MLP - 3 days training

Training loss with random weights

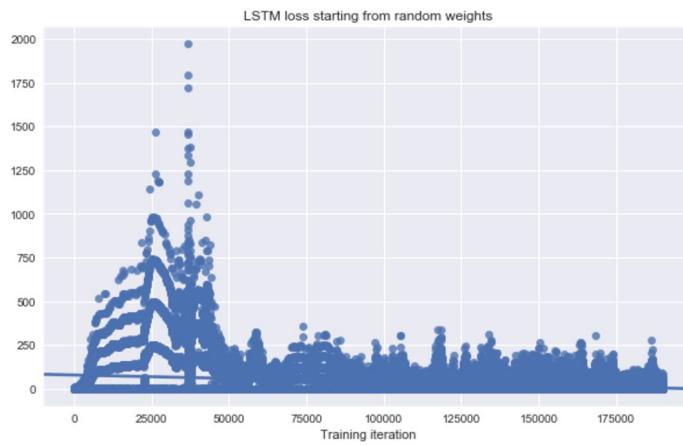


Loss with trained weights

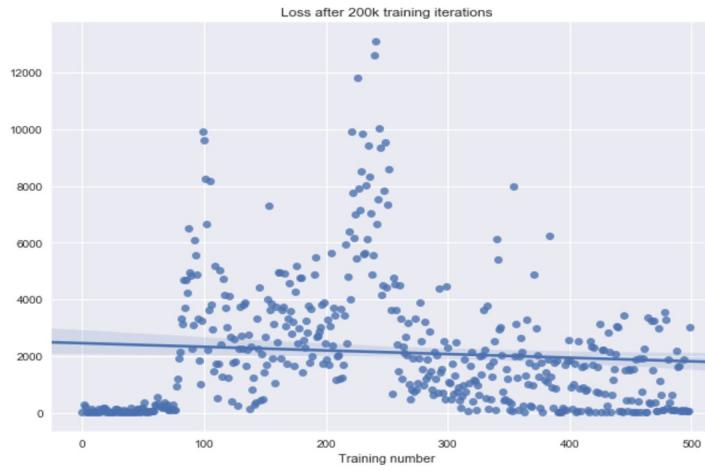


Loss Curves - DQN LSTM - 5 days training

Training loss with random weights

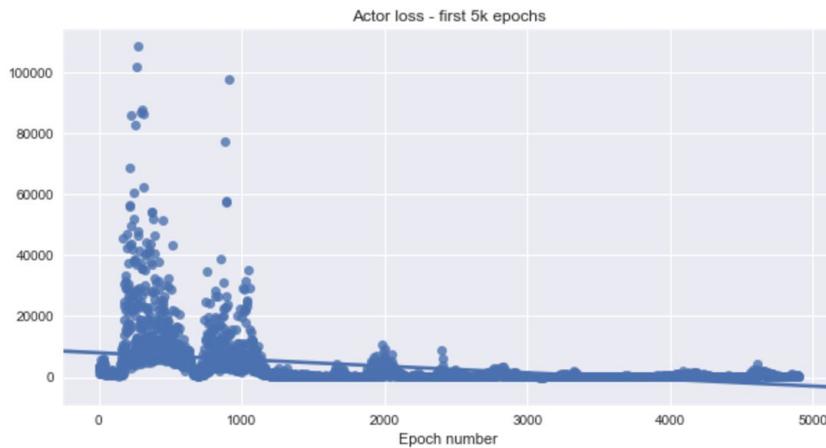


Loss with trained weights

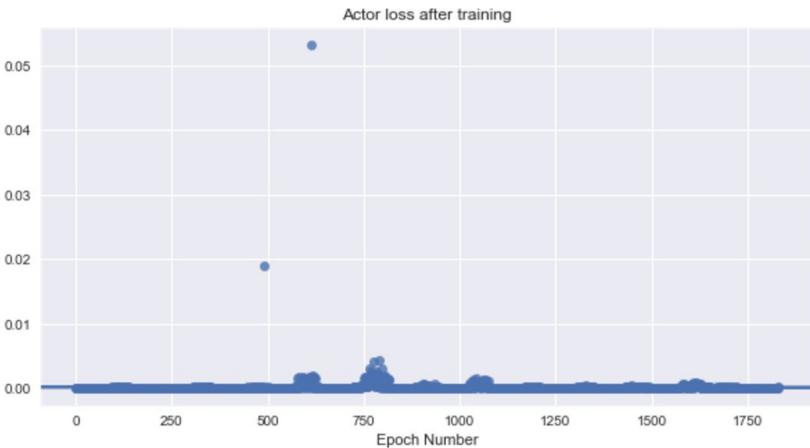


Loss Curves - Actor/Critic - .5 days training

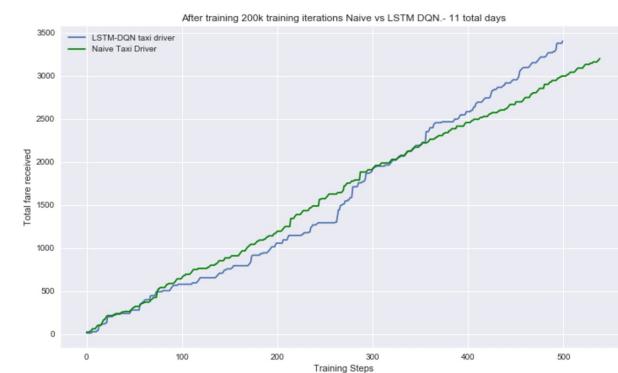
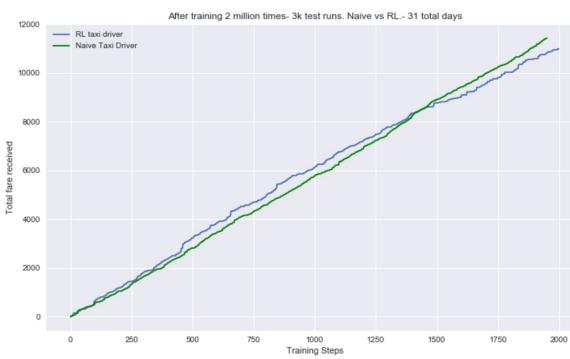
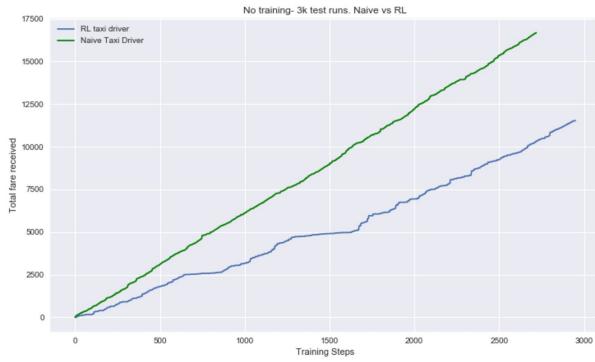
Training loss with random weights



Loss with trained weights



DQN - MLP RL Fare vs. Random Fare



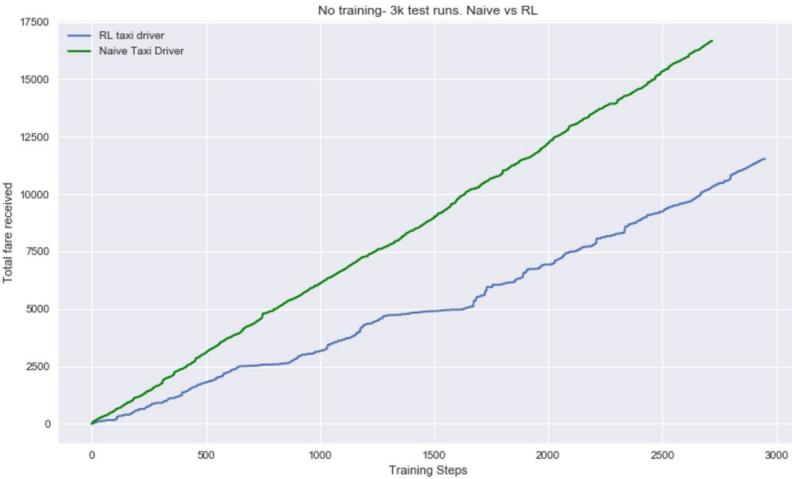
No training

2 million epochs (3 days)
DQN / MLP

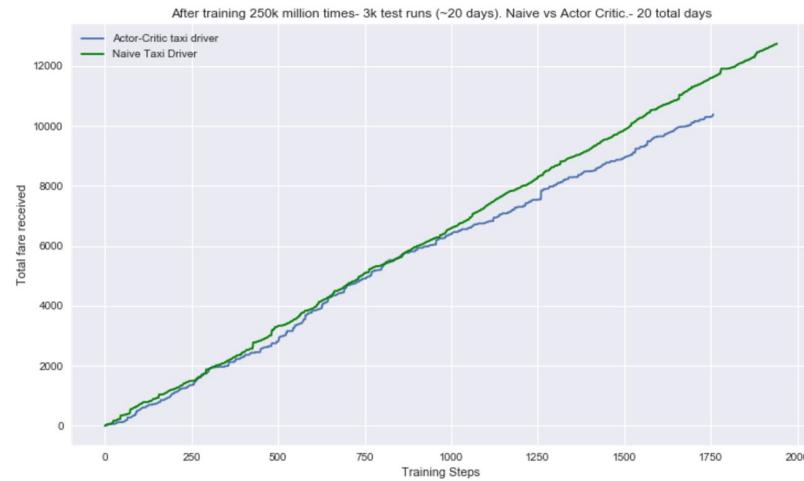
Average fare increase of 6-18% depending on start time and location

*Each epoch corresponds to moving from geohash A to geohash B

Actor-Critic vs Random fare



No training



350k epochs (.5 days) training

5

Live Demo

6

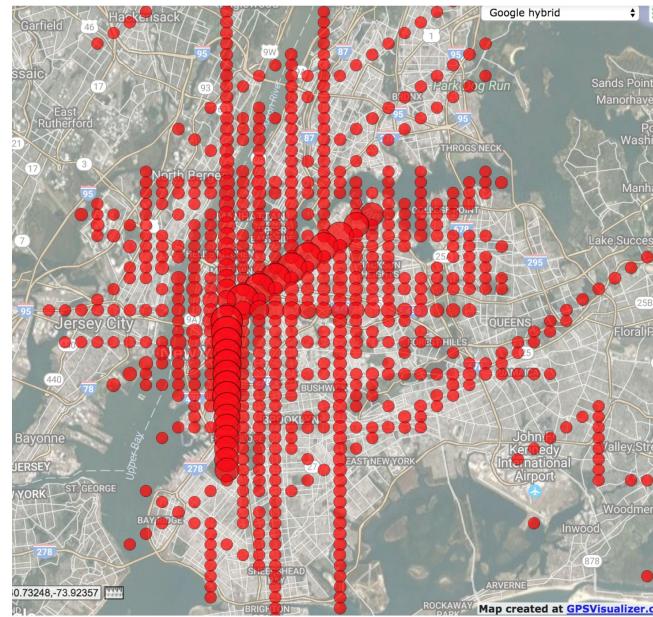
Paths learned by each algorithm

DQN - Paths Learned

Random Approach

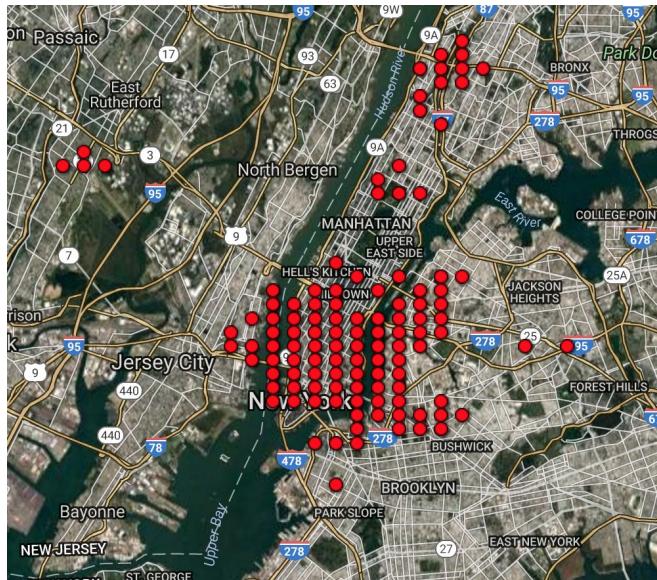


DQN - MLP

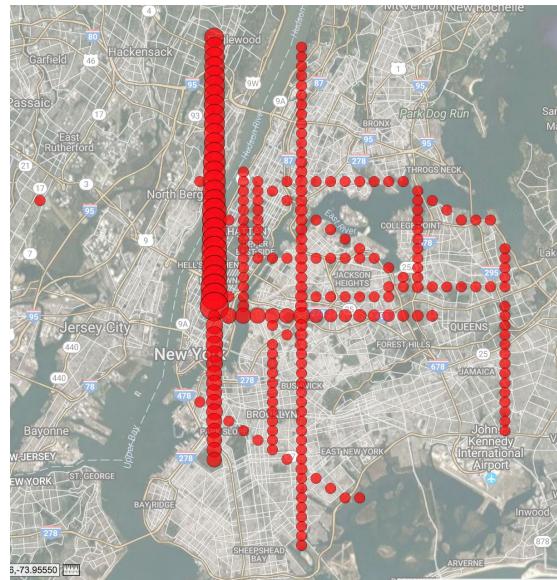


DQN - Paths Learned

Random

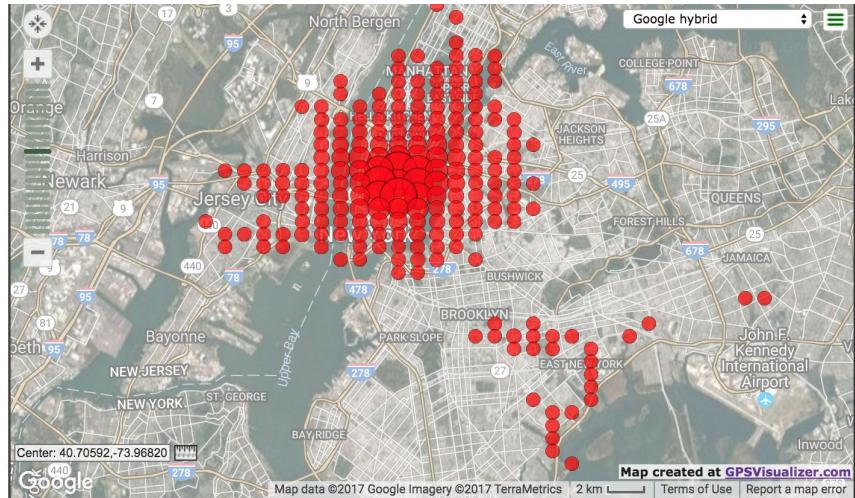


LSTM

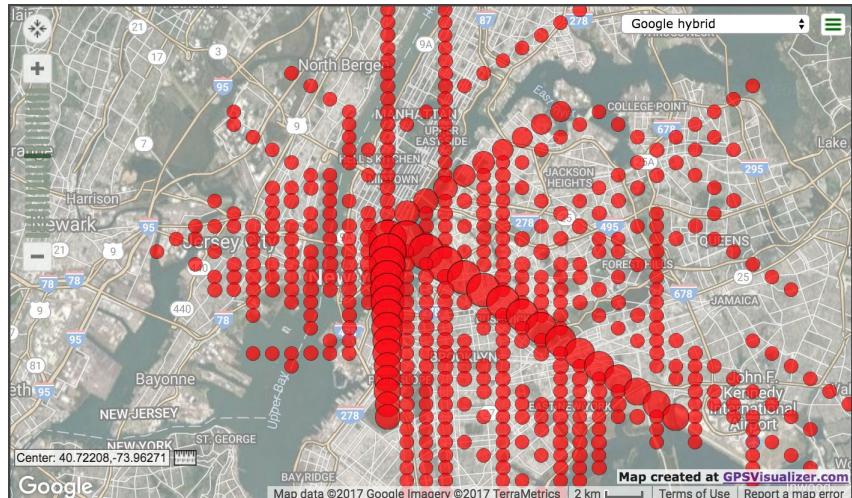


Actor-Critic Paths Learned

Random



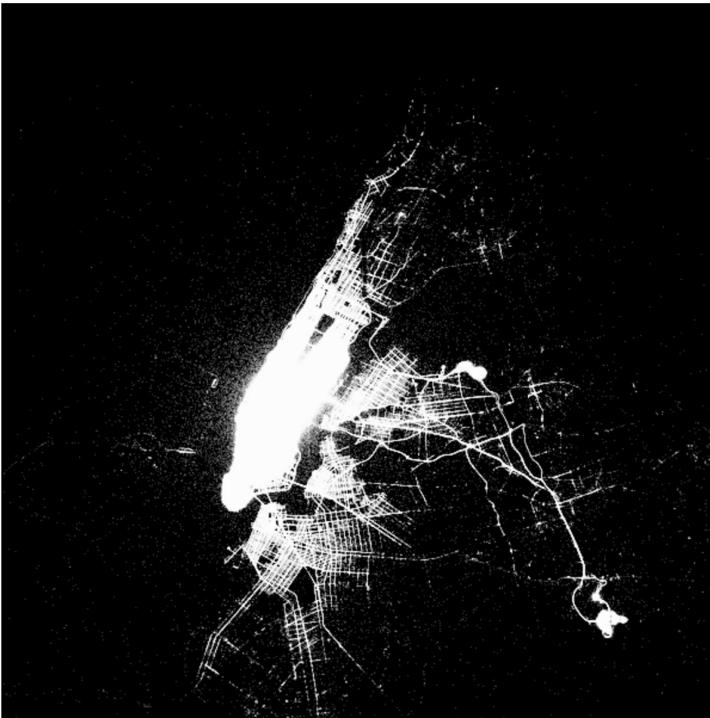
Actor-Critic



Thank you!

Moral: You always need more training epochs...

NYC Pickups 09-15



Source: <http://minimaxir.com/2015/11/nyc-ggplot2-howto/>

“ Quotations are commonly printed as a means of inspiration and to invoke philosophical thoughts from the reader.



THIS IS A SLIDE TITLE

- Here you have a list of items
- And some text
- But remember not to overload your slides with content

Your audience will listen to you or read the content, but won't do both.



INSTRUCTIONS FOR USE

Open this document in Google Slides (if you are at slidescarnival.com use the button below this presentation)

You have to be signed in to your Google account

EDIT IN GOOGLE SLIDES

Go to the *File* menu and select *Make a copy*.

You will get a copy of this document on your Google Drive and will be able to edit, add or delete slides.

EDIT IN POWERPOINT®

Go to the *File* menu and select *Download as Microsoft PowerPoint*. You will get a .pptx file that you can edit in PowerPoint.

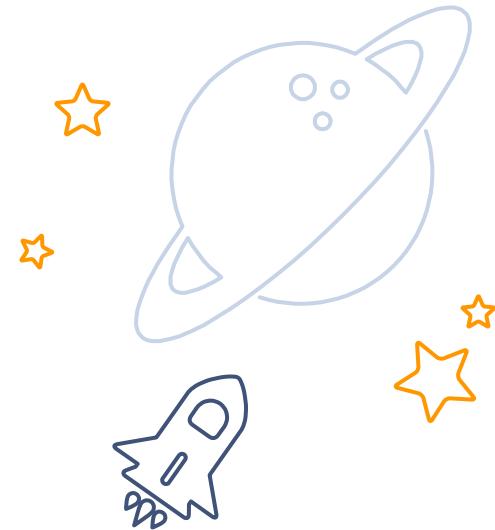
Remember to download and install the fonts used in this presentation (you'll find the links to the font files needed in the Presentation design slide)

More info on how to use this template at www.slidescarnival.com/help-use-presentation-template

This template is free to use under [Creative Commons Attribution license](#). You can keep the Credits slide or mention SlidesCarnival and other resources used in a slide footer.

BIG CONCEPT

Bring the attention of your audience over a key concept using icons or illustrations





YOU CAN ALSO SPLIT YOUR CONTENT

White

Is the color of milk and fresh snow, the color produced by the combination of all the colors of the visible spectrum.

Black

Is the color of coal, ebony, and of outer space. It is the darkest color, the result of the absence of or complete absorption of light.



IN TWO OR THREE COLUMNS

Yellow

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.

Blue

Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.

Red

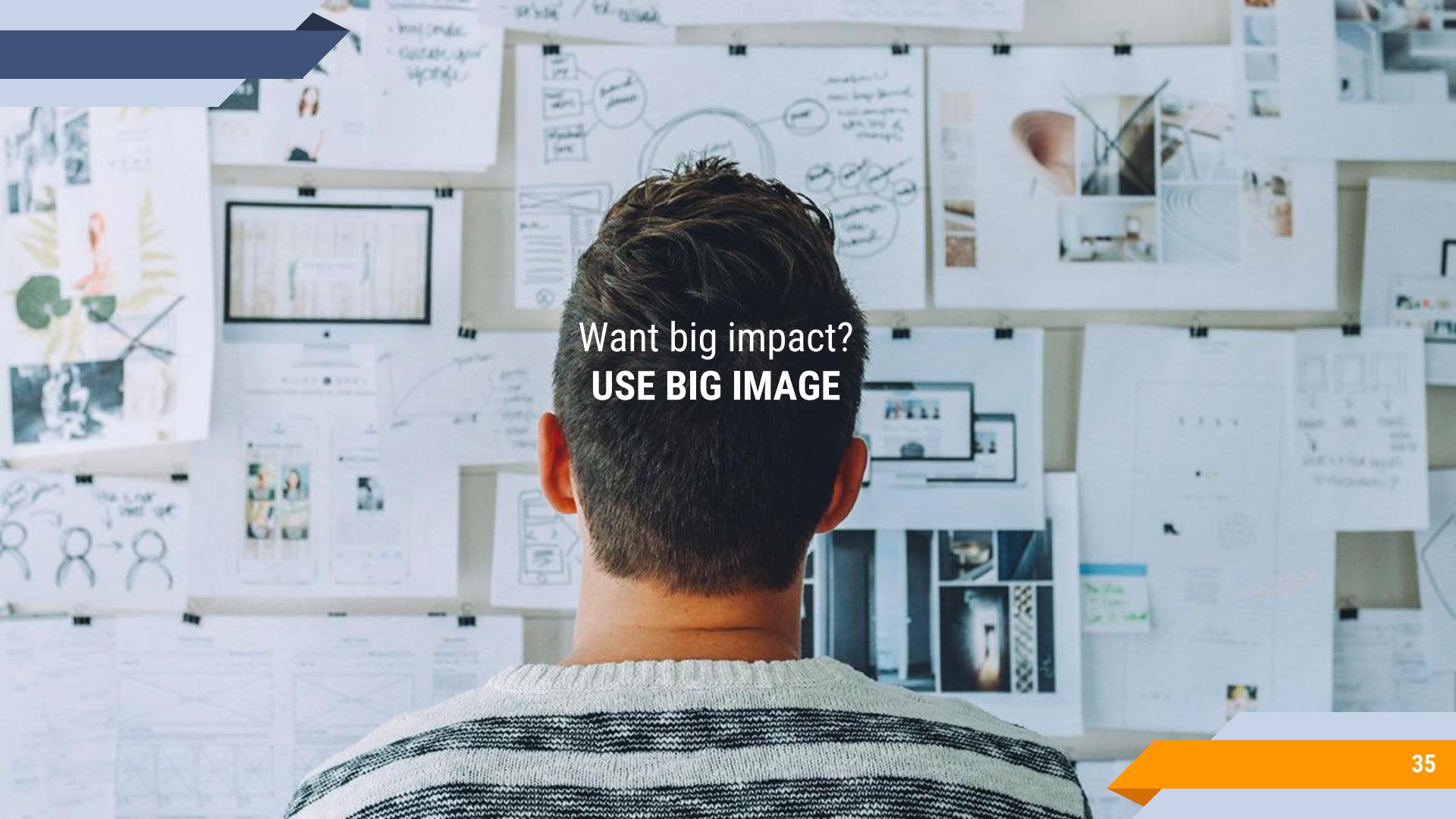
Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.



A PICTURE IS WORTH A THOUSAND WORDS

A complex idea can be conveyed with just a single still image, namely making it possible to absorb large amounts of data quickly.



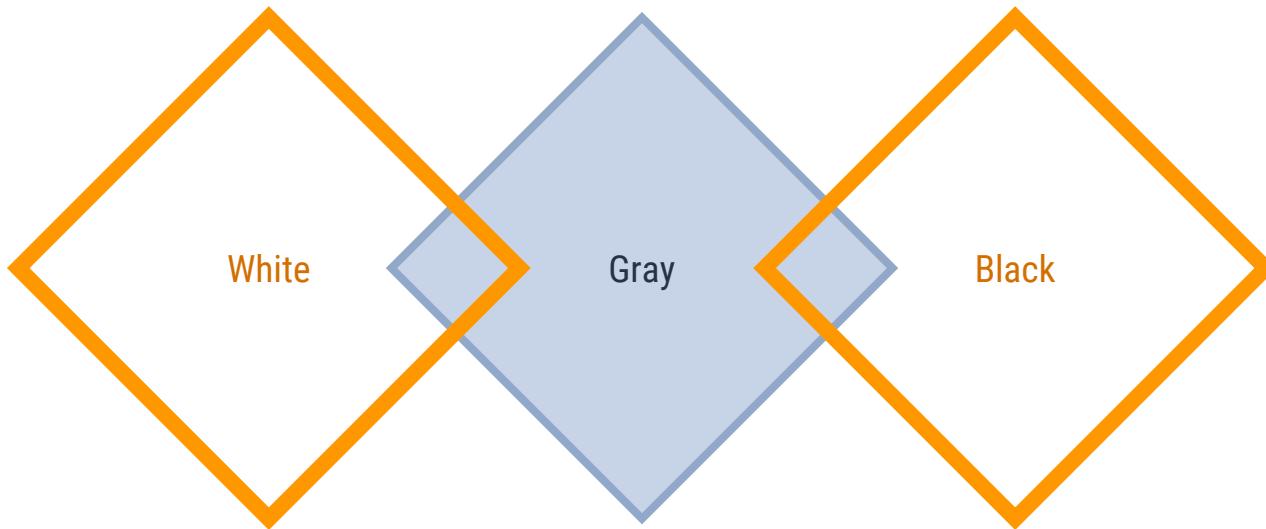


Want big impact?
USE BIG IMAGE

A photograph of a man from behind, looking at a wall covered in various design sketches, prototypes, and photographs. The wall is a creative workspace, with a large blue arrow pointing towards the top left corner.



USE CHARTS TO EXPLAIN YOUR IDEAS

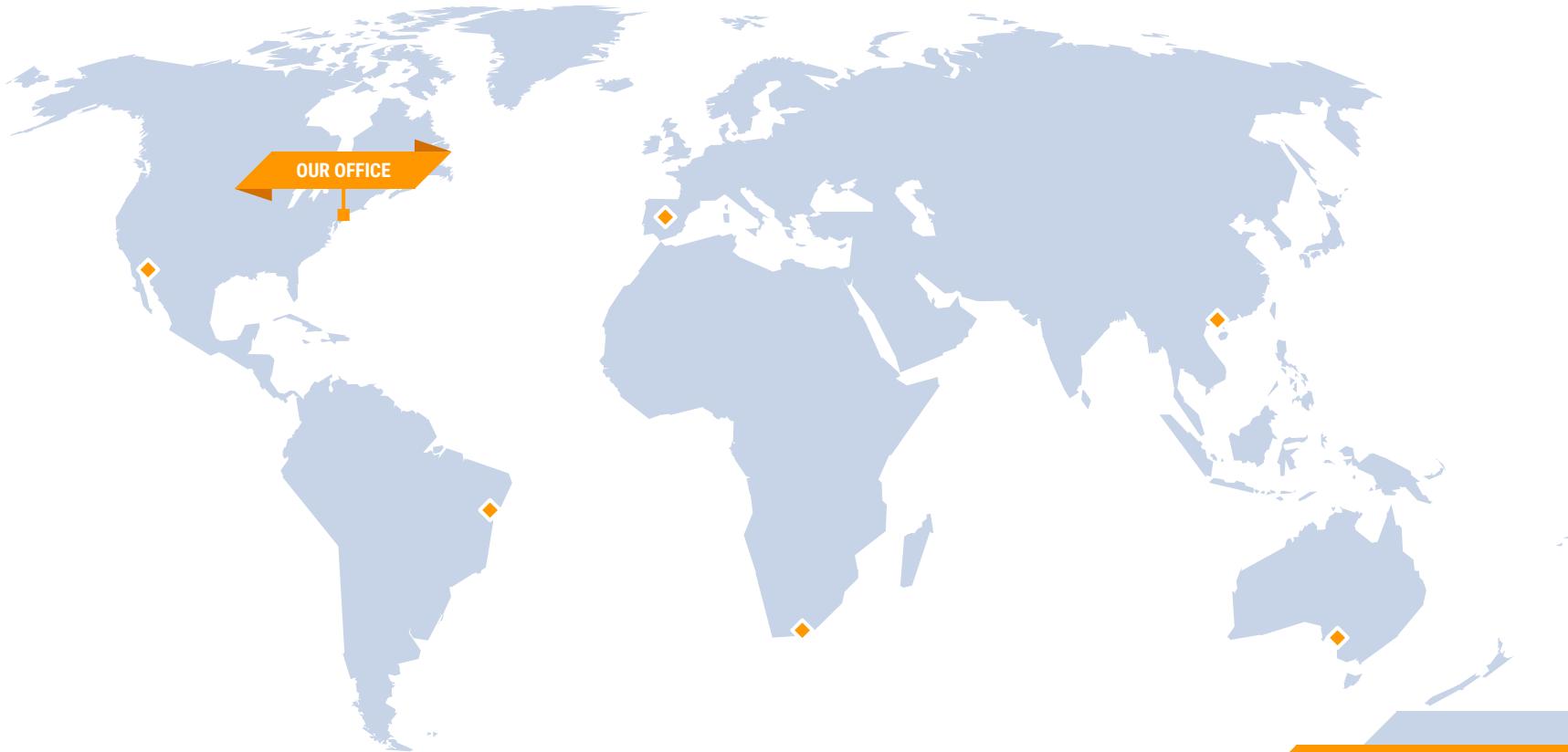




AND TABLES TO COMPARE DATA

	A	B	C
Yellow	10	20	7
Blue	30	15	10
Orange	5	24	16

MAPS





89,526,124

Whoa! That's a big number, aren't you proud?



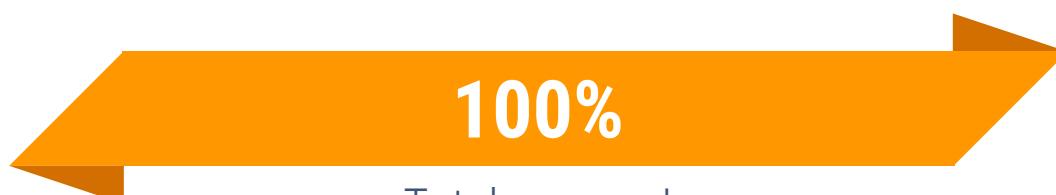
89,526,124\$

That's a lot of money



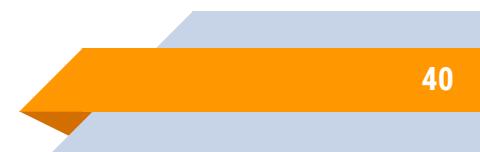
185,244 users

And a lot of users



100%

Total success!





OUR PROCESS IS EASY

first

second

last



LET'S REVIEW SOME CONCEPTS

Yellow

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.

Yellow

Is the color of gold, butter and ripe lemons. In the spectrum of visible light, yellow is found between green and orange.

Blue

Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.

Blue

Is the colour of the clear sky and the deep sea. It is located between violet and green on the optical spectrum.

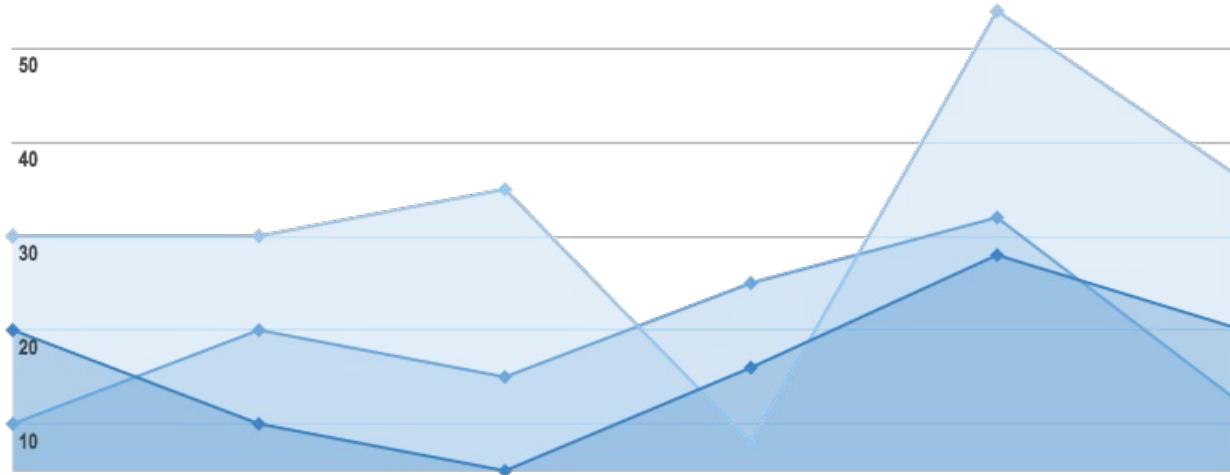
Red

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.

Red

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.

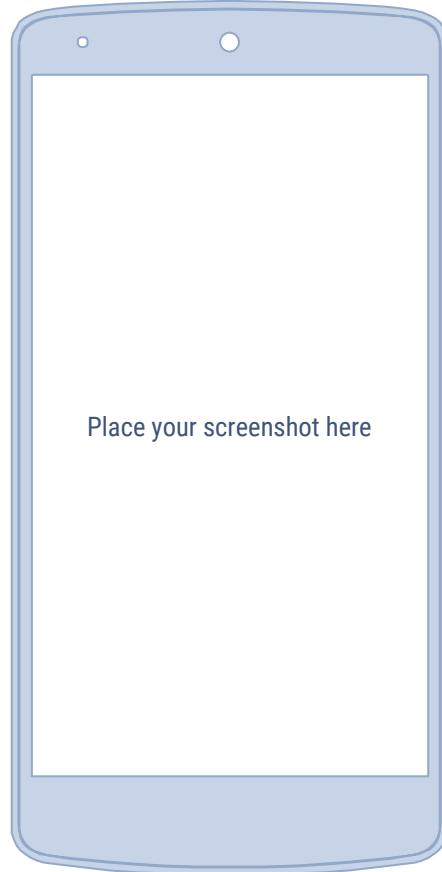
GRAPH TITLE



You can insert graphs from [Google Sheets](#)

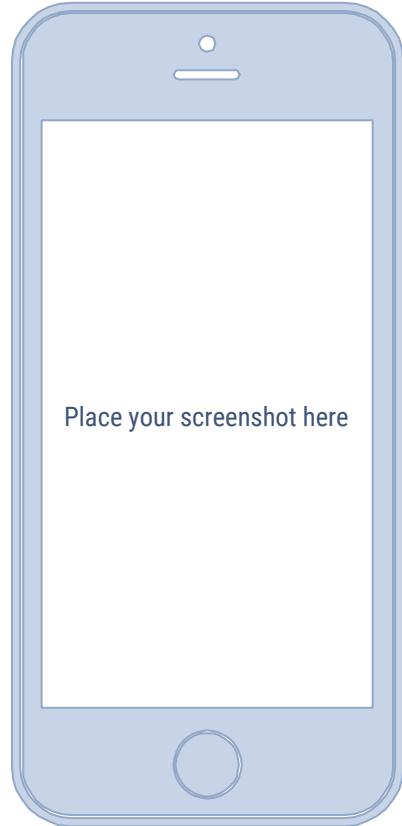
ANDROID PROJECT

Show and explain your web, app or software projects using these gadget templates.



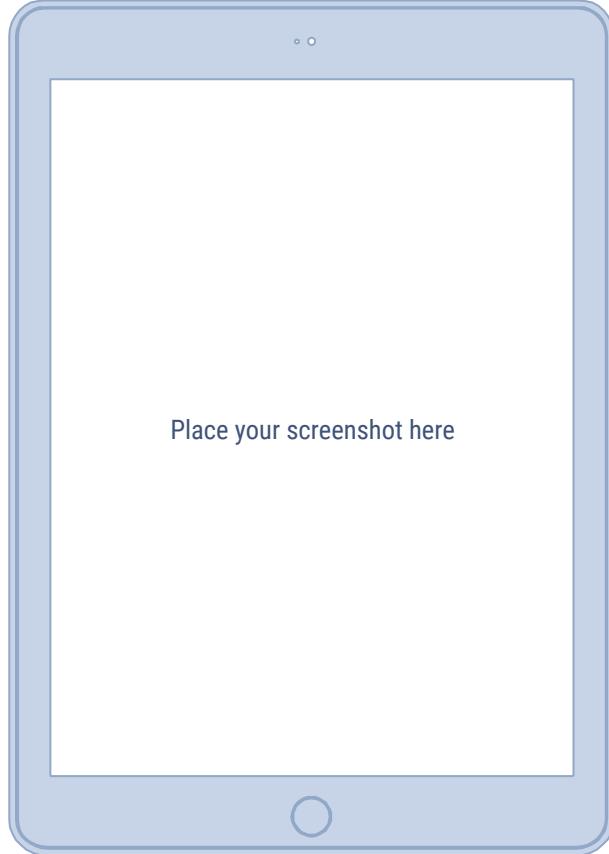
iPhone PROJECT

Show and explain your web, app or software projects using these gadget templates.



TABLET PROJECT

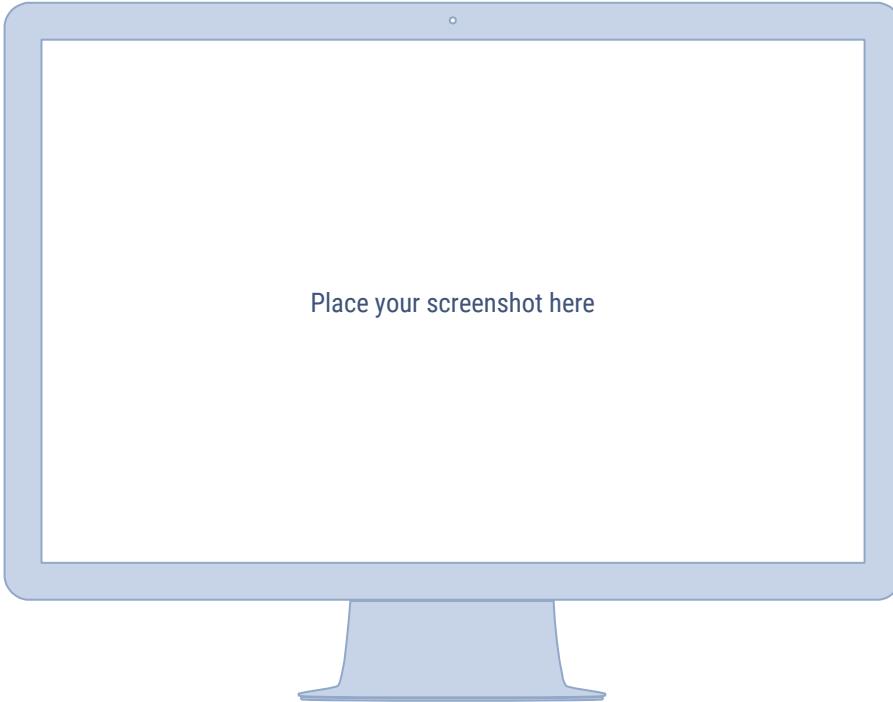
Show and explain your web, app or software projects using these gadget templates.

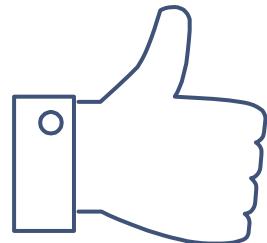


DESKTOP PROJECT

Show and explain your web, app or software projects using these gadget templates.

Place your screenshot here





THANKS!

Any questions?
You can find me at
@username & user@mail.me



CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by [SlidesCarnival](#)
- Photographs by [Startup Stock Photos](#)



PRESENTATION DESIGN

This presentation uses the following typographies and colors:

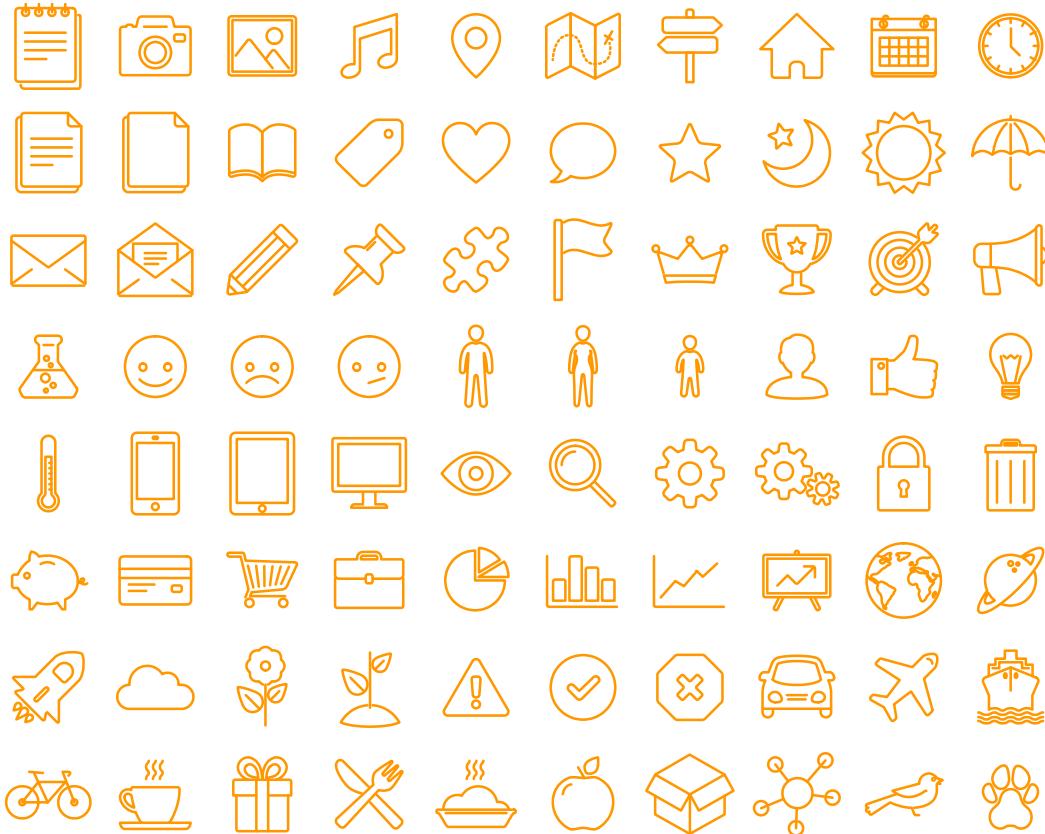
- Titles: Roboto Condensed
- Body copy: Roboto Condensed

You can download the fonts on this page:

<https://material.io/guidelines/resources/roboto-noto-fonts.html>

Navy #3f5378 · Dark navy #263248 · Yellow #ff9800

You don't need to keep this slide in your presentation. It's only here to serve you as a design guide if you need to create new slides or download the fonts to edit the presentation in PowerPoint®



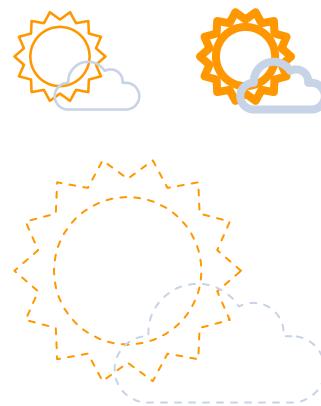
SlidesCarnival icons are **editable shapes**.

This means that you can:

- Resize them without losing quality.
- Change line color, width and style.

Isn't that nice? :)

Examples:





Now you can use any emoji as an icon!

And of course it resizes without losing quality and you can change the color.

How? Follow Google instructions

<https://twitter.com/googledocs/status/730087240156643328>

