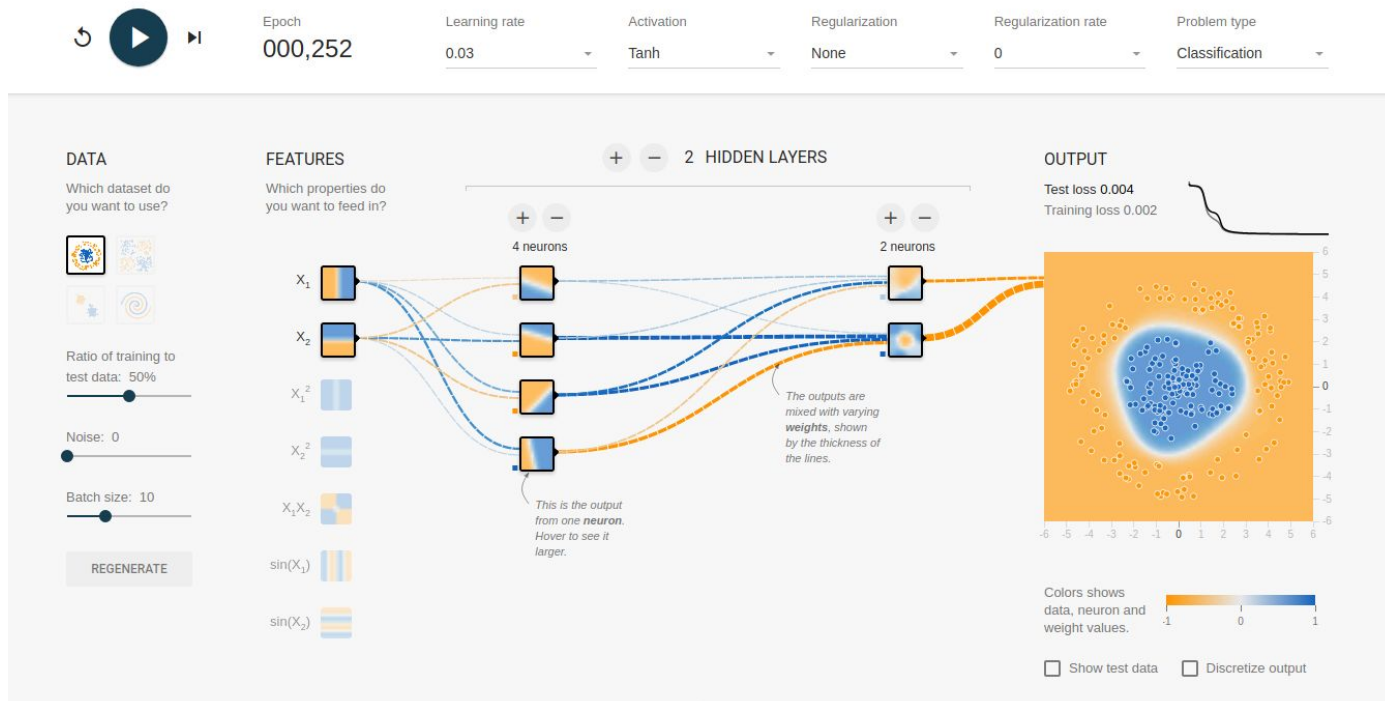


# Tensorflow JS

Deep learning getting easier!

By Husein Zolkepli

# First time I saw, playground tensorflow



# First time I saw, playground tensorflow (cont)

I found it was really cool, really well to deliver and explain what is Neural Network just on top of any browser.

Huge educational purpose! I used most of the time to explain roughly what is neural network to students and beginners.

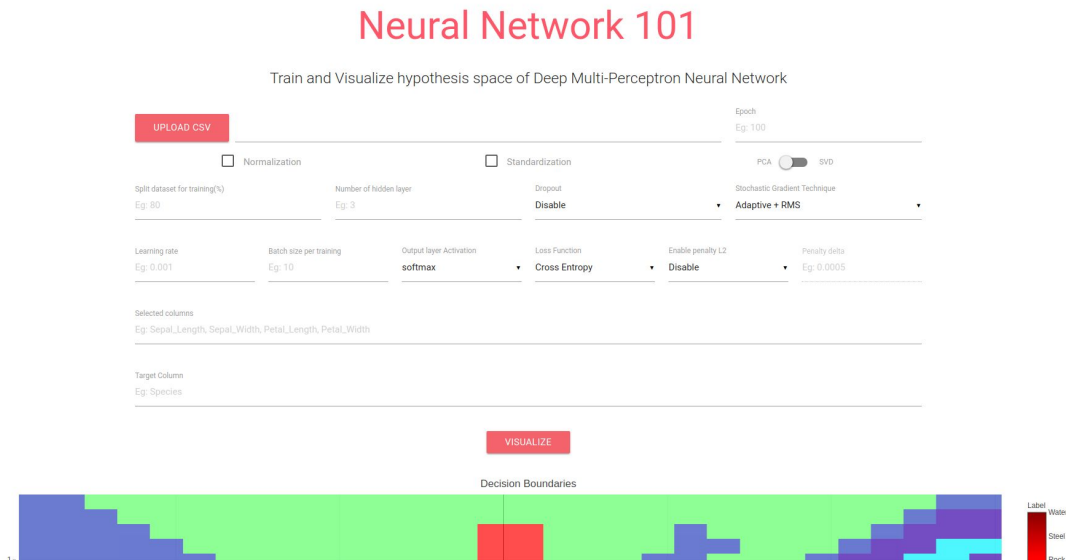
Does we know, Artificial Neural Network is one of the application of Form 4 Additional Mathematics? Keep derivatives our function.

Now, math is so cool!

# So, I try to duplicate the same thing, but with custom dataset

Yes, we can upload custom dataset, but I had to code Python Backend, POST the data to the backend, and private streaming every training iteration using Socket-IO.

<http://www.huseinhouse.com/neuralnet/>



# Not enough with that, I also do for another architecture of Neural Network

Same thing like previous website, but this specifically to forecast any timestamp based data using custom Recurrent Neural Network.

<http://www.huseinhouse.com/recurrentnet/>

## Recurrent Network 101

Train and Visualize prediction graph of Recurrent Neural Network

UPLOAD CSV

Number of hidden layer  
Eg: 3

Layer size  
Eg: 128

Split dataset for training(%)  
Eg: 80

Gate Cell  
LSTM

Dropout  
Disable

Dropout rate  
Eg: 0.5

Learning rate  
Eg: 0.001

Timestamp per training  
Eg: 10

Enable penalty L2  
Disable

Penalty delta  
Eg: 0.0005

Activation function  
tanh

SGD Technique  
Adaptive Momentum (ADAM)

Loss Function  
Root Mean-Square Error

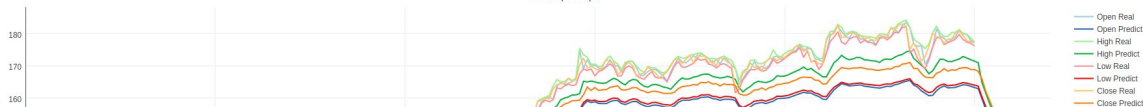
Epoch  
Eg: 30

Selected columns  
Eg: Open, High, Close, Volume

Future timestamp count  
Eg: 100

TRAIN

Overlap Graph



# I was realised,

To respawn a backend, it is very expensive especially for Recurrent Neural Network.

It requires a live connection to listen to broker Socket IO, private room.

Not everyone can spawn the server, not everyone able to understand to control a server.

Even both are open-sourced, <https://github.com/huseinzol05/Neural-Network-101>, <https://github.com/huseinzol05/Recurrent-Network-101>, but people doesn't like about backend thingy.

# Why playground tensorflow is very successful?

- No drivers or initial setup

Like I said, not everyone know how to do backend. I don't think everyone know what is SSH, even.

- cross-platform

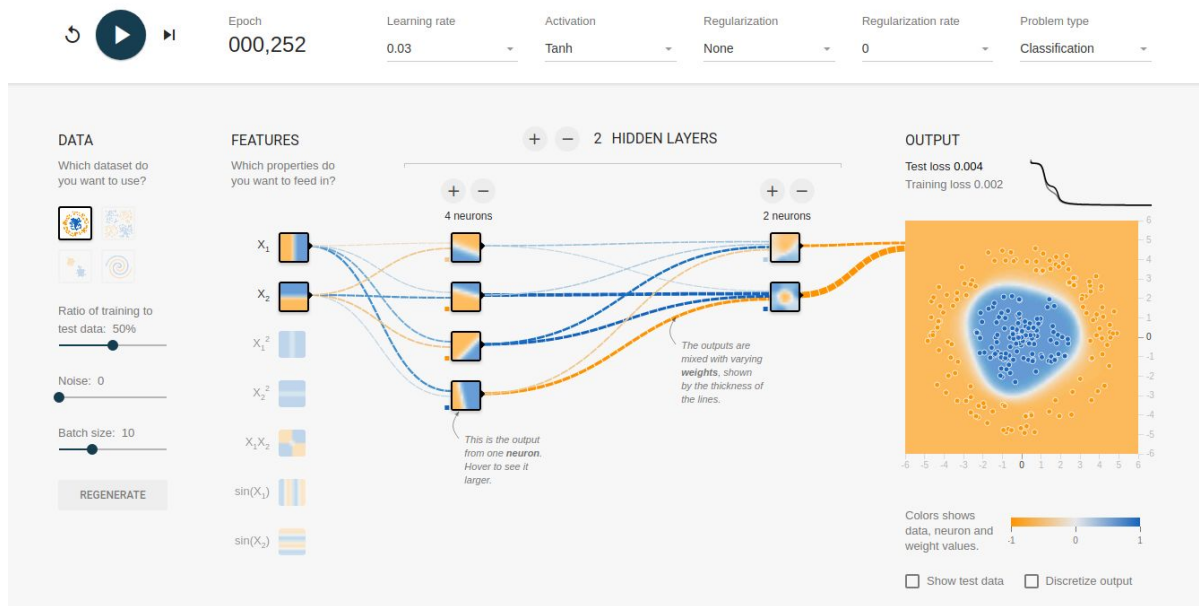
We just need a modern browser, and done!

Share the link, and everyone can play it.

# Why playground tensorflow is very successful? (cont)

- Interactive

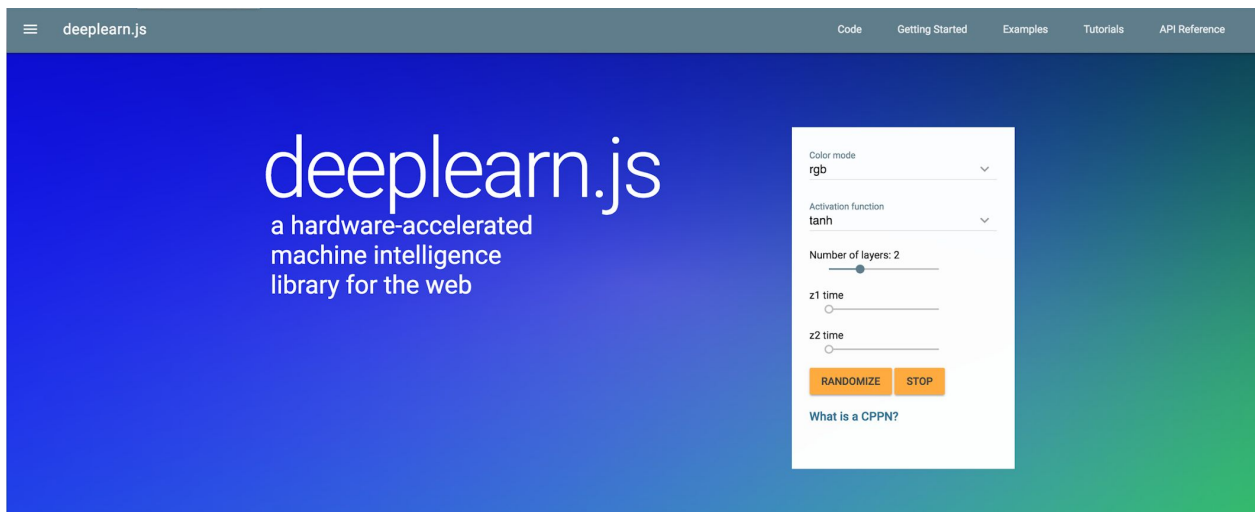
With browser, we can do a lot of thing related to user experiences. There are buttons everywhere, slides and etc. so it is very interactive.





# deeplearn.js

- Released August 2017
- GPU-accelerated via WebGL
- Allows to control the engine using pure interface



# A new door for front-end

After deeplearn.JS released, deep learning able to train distributedly using WebGL on any browser.

From here we know, developing AI especially Deep Learning is getting easier and easier.

It requires hybrid of frontend knowledge and some linear algebra understanding to develop using deeplearn.JS

A bonus if we understood Keras or Tensorflow syntax.

# Examples of deeplearn.JS

Real Time style transfer for images on any browser.

We can import from Python models and load on deeplearn.JS

There are a lot more, text prediction, image classification.



Original photo

Reference photo

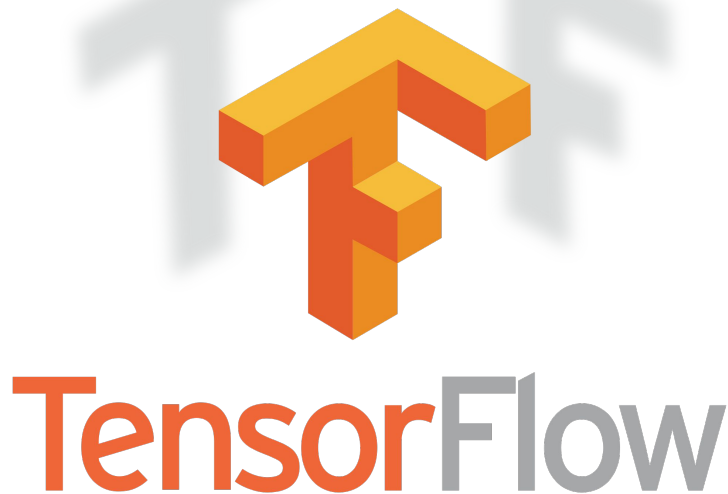
Result

# After that,

Google released Tensorflow.JS on top of deeplearn.JS.

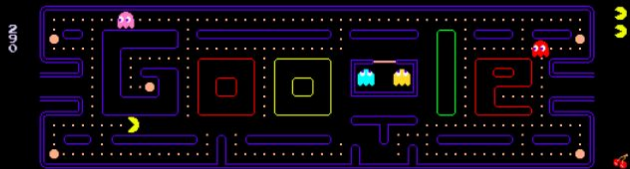
3 important features from this library:

1. Alter models directly on the browser
2. Import pre-trained models from any languages to browser
3. Re-train pre-trained models from any languages on a browser



# Examples of Tensorflow JS

Turn your **Web Camera** into a controller using a **Neural Network**.



LOSS: 0.00011



PLAY

Learning rate

0.0001

Batch size

0.4

Epochs

20

Hidden units

100



Click to add the  
current camera  
view as an example  
for that control



3 examples



2 examples



1 examples



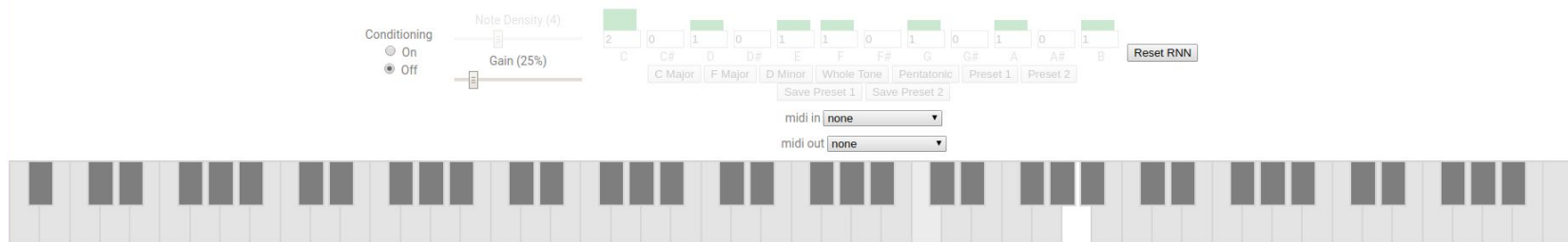
2 examples

PAC-MAN™ © BANDAI NAMCO Entertainment Inc.

We can play Pac-Man, control using our faces!

# Also Recurrent Neural Network available

## Performance RNN

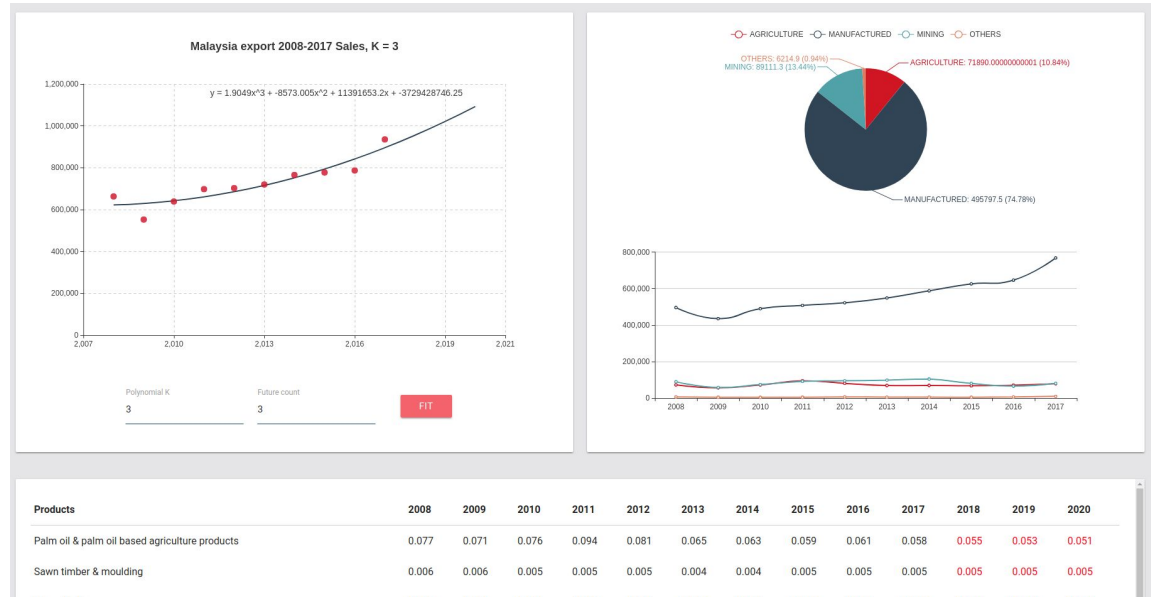


[Performance RNN](#) was trained in TensorFlow on MIDI from piano performances. It was then ported to run in the browser using only Javascript in the [TensorFlow.js](#) environment. Piano samples are from [Salamander Grand Piano](#).

You can play a digital keyboard with an AI, live!

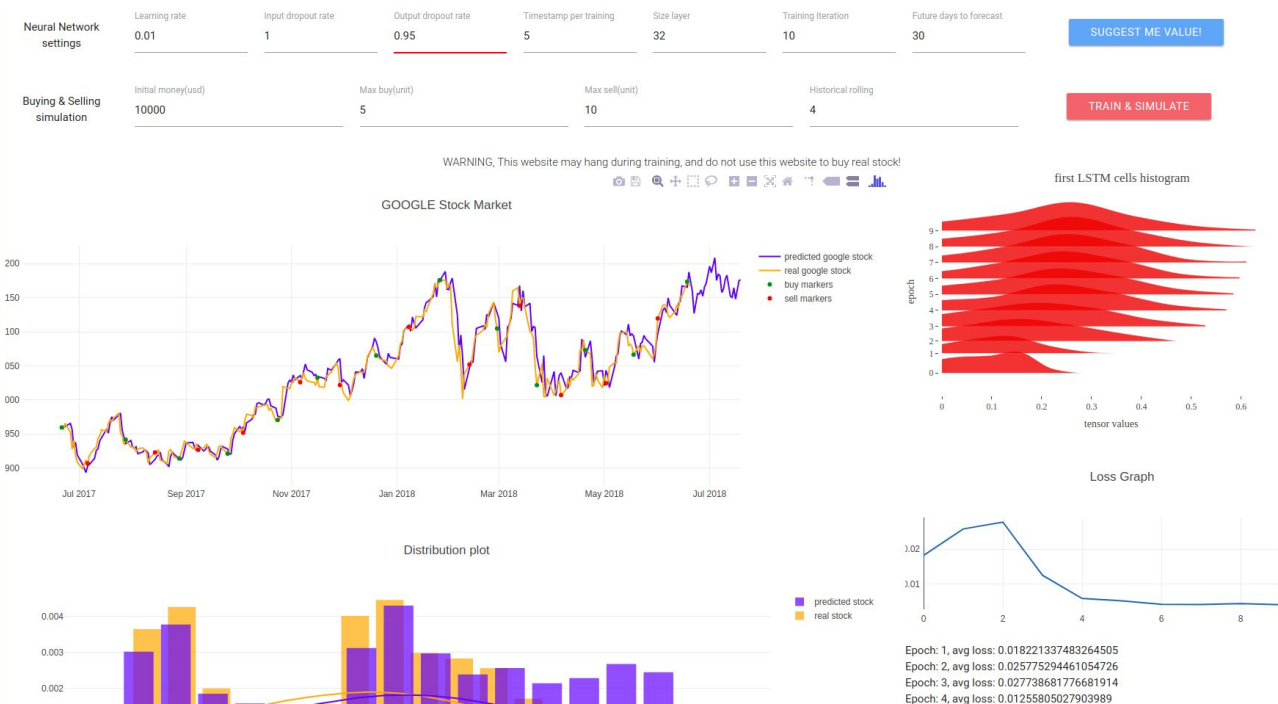
# We want to solve some real problem

Forecast our export data by sectors and countries. it also included neural network that able to forecast future years and every sector percentages for every future years.



# We want to solve some real problem (cont)

Forecasting stock market with investment simulation.

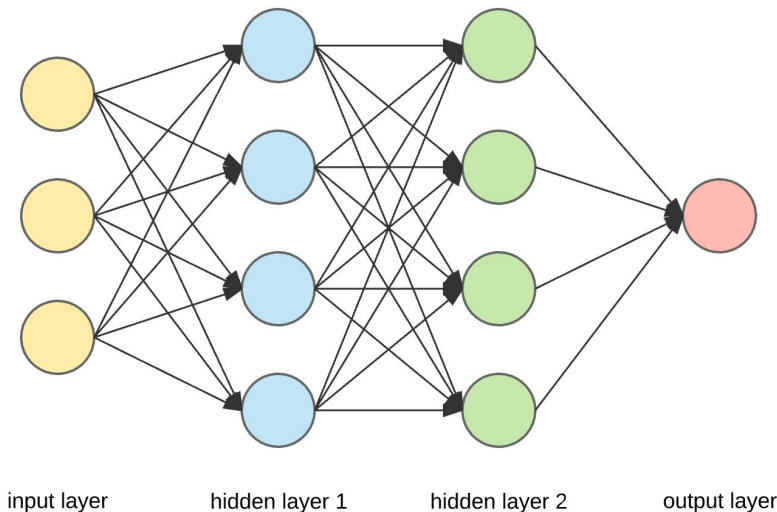




# Sometime,

I just want to deploy a shallow network for some non-linear forecasting engine or simple classification engine.

To code any backend for that, irritating me sometime. Plus you need to code an API for that.

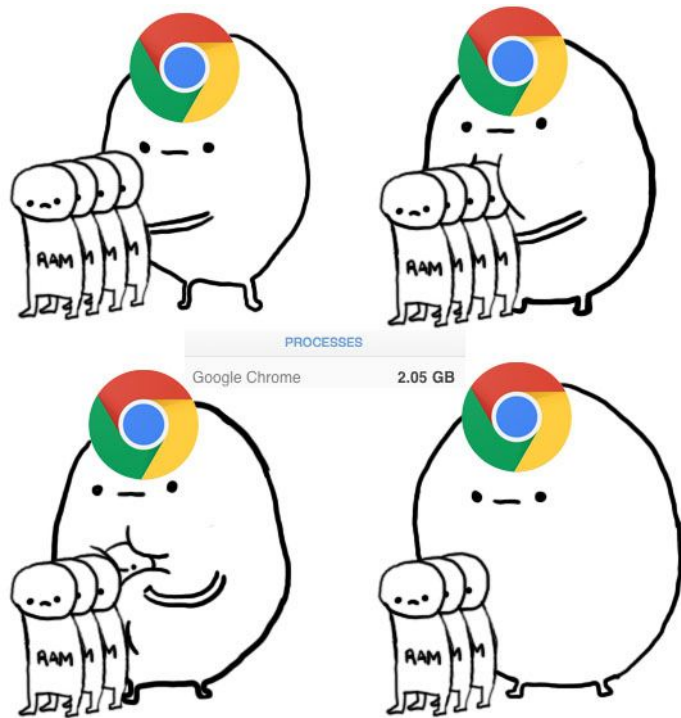


{ api }

# But we must remember

Tensorflow.JS calculated everything on top of a browser, it really depends on how our OS treated a browser.

It may consumed all of our machine resources if we are care about CPU and RAM usage. Browser acted as a virtual environment.



# I really suggest you

Not to train any deep learning models using Tensorflow.JS. Better use Python or any backend languages.

Tensorflow.JS need to cross a lot of interface level to contact with real system and GPU.

