Behavioral Cloning Cheatsheet

Paul Heraty • Dec 08, 2016

112 There are about 10 things that I wish I knew before I started this exercise.

Note: These points below are not the 'only' way of solving this problem. Think of them as pointers and feel free to pick and choose as you see fit.

1. How to use Python generators in Keras. This was critical as I was running out of memory on my laptop just trying to read in all the image data. Using generators allows me to only read in what I need at any point in time. Very useful.
2. Use a GPU. This should almost be a prerequisite. It is too frustrating waiting for hours for results on CPU. I must have run training 100 times over the past 3 weeks and it was driving me crazy. Using a GTX980M was around 20x faster in training that a quad-core Haswell CPU.
3. Use an analog joystick. This also should be a prerequisite. I'm not sure if its even possible to train with keyboard input. I think some have managed it, bu for me it's a case of garbage in, garbage out.
4. Use successive refinement of a 'good' model. This really saves time and ensures that you converge on a solution faster. So when you get a model working a little bit, say passing the first corner, then use that model as a starting point for your next training session (kinda like Transfer Learning). Generate some new IMG data, lower the learning rate, and 'fine tune' this model.
5. Use the 50Hz simulator. This generates much smoother driving angle data. Should be the default. You can find a link to download this on the Slack channel. Choose the fastest graphic quality and lowest screen resolution has helped the model to perform better
6. You need at least 40k samples to get a useful initial model. Anything less was not producing anything good for me.
7. Copy the Nvidia pipeline. It works :) And it's not too complex.
8. Re-size the input image. I was able to size the image down by 2, reducing the number of pixels by 4. This really helped speed up model training and did not seem to impact the accuracy.
9. I made use of the left and right camera angles also, where I modified the steering angles slightly in these cases. This helped up the number of test cases, and these help cases where the car is off center and teaches it to steer back to the middle.
10. Around 5 epochs seems to be enough training. Any more does not reduce the mse much if at all.
11. When you're starting out, pick three images from the .csv file, one with negative steering, one with straight, and one with right steering. Train a model with just those three images and see if you can get it to predict them correctly. This will tell you that your model is good and your turn-around time will be very quick. Then you can start adding more training data

Hopefully the above will help you converge quicker. It can be a pretty frustrating project at the beginning, but worth it when you have it working!

Most of these ideas have come from the great folks on the Slack channel, which reminds me:

12. Use the Slack channel. The people on there are awesome!

behavioralcloninglab behavioral-cloning behavior-cloning

Comment

Matthew Zimmer • Dec 09, 2016

Thank you, @Paul Heraty. This is an invaluable post.

I wanted to reiterate the importance of #11 as it helped me tremendously for project 2 when just getting my feet wet. @Vivek Yadav actually mentioned this a while back and it stuck with me.

> every time i fit neural networks, I test the following.
>
> 1- Start with very little data, and over fit. If you cant overfit, your implementation is wrong. This is typically not done, but is actually a very powerful tool for checking your implementation is correct. We know if we have too little data, we will overfit. So its good to actually see if your network does it.

Read @Vivek Yadav's entire answer here:

What could be causing very low accuracy?

Dhruv Parthasarathy [Administrator] • Dec 09, 2016
I would also suggest checking out the following model from comma.ai if you're looking for a place to start. Good luck!

Guy Pavlov • Dec 09, 2016
Which slack channel do you guys use to discuss this project?

Carlos Galvez • Dec 20, 2016
After 1 week trying to figure out what was wrong with my model, I think the following should be added to the cheatsheet:

0. drive.py sends RGB images to the model; cv2.imread() reads images in BGR format!!!!

Fixed the bug and passed the 2 sharp corners at the first attempt lol.

David Lo • Dec 20, 2016
Thx Paul

David A Ventimiglia • Jan 08, 2017
@Guy Pavlov, I encountered people discussing Project 3 in the "p-behavioral-cloning" Slack channel. I assume that's the one.

Wolfgang Steiner • Jan 18, 2017
Could someone provide a link to the 50Hz simulator? I can't find it on the slack channel.

Yufeng Guo • Jan 21, 2017
@Wolfgang Steiner it's pinned in the channel that @David A Ventimiglia mentioned. Lot's of good stuff in the pinned posts, would recommend taking a look.

Direct link: https://files.slack.com/files-pri/T2HQV035L-F3A47CT7Z/download/simulator-linux.zip

Yu Shen • Jan 22, 2017
Hi @Paul Heraty, what's the batch size when you use 5 epochs? Thanks,

Yu Shen • Jan 22, 2017
@Paul Heraty, could you elaborate how you resize the image? You text:

"Re-size the input image. I was able to size the image down by 2, reducing the number of pixels by 4. This really helped speed up model training and did not seem to impact the accuracy."

I had difficulty to understand.

Do you mean by reducing the width, and height by 2 pixels, respectively? Then the number of pixels reduced cannot by just 4 pixels. It may be 2*width*2 + 2*(height - 2)*2 = 4*(width + height -2)

Yu Shen • Jan 22, 2017
@Paul Heraty, what the adjustment to the turning angles, for left/right cameras. It seems should be a function of the angle between the orientation angle of the left/right camera, and the central camera. For the left camera, the turning angle should adjust to the right by the angle, and the right camera, the turning angle should be adjusted to the left by the angle. But we don't know the orientation angles of the left, and right. Did you experiment to find some optimal estimations of the angles?

Yu Shen • Jan 22, 2017
Which joystick to recommend? I don't play game at all. Any suggestion would be appreciated.

Vamshi Korivi • Jan 28, 2017
@Paul Heraty. Thanks for this awesome post which helped almost everyone who is starting off their work on P3. Now that it is included in the project guidelines, i would like to add a line to the point#5. As Udacity has released the updated version of simulator, we have the choice of choosing the model resolution and speed. Along with other classmates on slack and I have experienced that choosing the fastest graphic quality and lowest screen resolution has helped the model to perform better. If you could add that line, it will be helpful for those who are yet to start/already working on P3. Thanks once again for your really helpful post.

Paul Heraty • Jan 29, 2017
@Yu Shen let me try to answer some of your questions. Apologies for the delay, I don't visit this post much anymore.

1/ What's the batch size when you use 5 epochs? Not sure that it matters really, I think I used 128. Each epoch should run enough batches to iterate through the entire dataset.

2/ Do you mean by reducing the width, and height by 2 pixels, respectively? No, I mean reduce by a factor of 2, so that if the input was 128x128, you resize to 64x64, thus reducing the image size by a factor of 4.

3/ What the adjustment to the turning angles, for left/right cameras. You should play with this number and figure out what works best. I used a constant rather than a function to calculate a value.

4/ Which joystick to recommend? I ended up buying the only thing my local shop had, which was a Thrustmaster flight simulator joystick. Any analog joystick should work though so long as there are drivers for the O/S you are using.

Paul Heraty • Jan 29, 2017

@
Vamshi Korivi good suggestion, done

Wei-Chung Chen • Jan 29, 2017
Cloud you explain more on the 4 method? how to reuse the better model by using Keras? do you have any information can share? Thanks.

Paul Heraty • Jan 30, 2017
Hi Wei-Chung,

first thing is to learn how to save and restore a model and it's params using Keras. You can save a model and params like this:

```
################################################################

# Save the model and weights

################################################################

model_json = model.to_json()

with open("./model.json", "w") as json_file:

    json.dump(model_json, json_file)

model.save_weights("./model.h5")

print("Saved model to disk")
```

I'll leave restoring to you to read up on.

Once you have a model that works a little, what I did was to successively refine it. Think of this as making minor tweaks to it. By lowering the learning rate, we are lowering the amount of change that a new epoch will make to the model. So get some new driving data for the section of the track where you are having issues, and train/tweak your existing model with this new data using a low learning rate. This should have the effect of tuning your model a little with an emphasis on 'learning' just that new section of track without breaking anything else.

It's not an exact science, but with some effort I was able to get this to work for me.

Joel Yarde • Jan 31, 2017
@Paul Heraty, Thank you for this guide. I initially tried using the keyboard before reading this going out and buying a joystick. I'm also using a GPU for training which makes it possible to iterate quickly. I'm also using transfer learning with a sizeable CNN (VGG16) so most of my work has been in a GPU instance on Amazon. Finally, I based my approach on the NVIDIA paper (as per your suggestion) after initially trying it my way. Iteration 9 (my next attempt) will add their image translation to simulate varying positions in the lane; which I think is key to good generalization. Thus far, I've artificially augmented my samples by 5 times with great results.

I've gone through 8 iterations of the model architecture and sample structure to-date with successively better results. Hopefully iteration 9 will be the successful one.

I will share my results/success on this thread. Thank you again for sharing. You saved me loads of time.

Raghupathy • Feb 01, 2017
@Joel Yarde so your network architecture is: data->vgg16→your_nvidia_cnn_implemetation correct?

Joel Yarde • Feb 01, 2017
@Raghupathy My Network architecture is **Data→VGG16 CNN→Custom Classifier**. The general approach is based on the NVIDIA paper. I didn't feel like I would learn much by replicating exactly their approach but rather, apply the basic principles they used. Probably would be faster if I replicated their work but my experience has been worth the pain of iterating through various conditions.

Using VGG16 CNN allows me to focus on: data sampling and pre-processing (how the sample size and preprocessing influences the results) and the classifier. And tune those to gradually improve results.

Add your comment...

20 answers


Soon Cheong • Dec 08, 2016

23 This is not a cheat sheet, this is something that should have been in the project description.

Comment

Dhruv Parthasarathy [Administrator] • Dec 09, 2016

10

Hey @Paul Heraty thank you for this guide! I'm going to link to it from the project description. For others running into issues, we are doing the following:

1. Releasing our own training data so that you don't have to buy a joystick etc. if you don't want to (This will happen by Monday). We are currently testing our model on our data to make sure it works.
2. We have a new build of the simulator that only writes to disk at the end instead of continuously to support slower computers. We are testing this now against our models and aim to upload it next week.

One more helpful note:

- When creating my model, I leaned heavily on this code from comma.ai . Good luck!

And one last point - this is a hard project! The problems in the Self-Driving Car space are difficult and take time to get through! It took me a while to get a working solution for this project but when I did it felt amazing. I'm confident all of you can do this - we accepted you because we know you can. But it will take hard work and persistence. Continue to help each other, chat on the #p-behavioral-cloning channel and you'll be amazed at what you can learn in a short amount of time. That being said, the difficulty should be in the concepts and applying them, not our provided software etc. and we will fix that immediately. Good luck!

Comment

Lance Chen • Dec 09, 2016
@Dhruv Parthasarathy [Administrator] Thanks for the extra tip. Could you share where I can find the #p-behavioral-cloning channel? It seems it is not in nd013 Slack

Jou-ching (George) Sung • Dec 09, 2016
The #p-behavioral-cloning channel is on the enrolled students' Slack: https://carnd.slack.com/

Lance Chen • Dec 09, 2016
@Jou-ching (George) Sung I joined it. Thanks for your help!

Shirish Jamthe • Dec 09, 2016
Hi Lance, Can you tell me how to join? I am also only in nd013 not in carnd. thanks

Alex Cui • Dec 09, 2016
@Jou-ching (George) Sung How can I join the carnd slack. It seems I am not in the team.

Jou-ching (George) Sung • Dec 09, 2016
@Shirish Jamthe @Alex Cui Try https://carnd-slack.udacity.com/ and sign up from there. I got the link from the Student Handbook.

Lance Chen • Dec 09, 2016
@Shirish Jamthe just saw your message. George always helps 🙂

Yu Shen • Jan 22, 2017
@Dhruv Parthasarathy [Administrator], how can I tell if I'm using the 50K simulator? As of Jan. 20, the software that I downloaded from the project page, there is no mention of 50K or 10K simulator. In the simulators, there is no explicit mention of 10K or 50K simulator. Maybe, I don't understand the gaming jargon?

Yu Shen • Jan 22, 2017
It seems from Tips for Behavior Cloning, I found answer to my question above. The default simulator provided by Udacity is of 10K.

"

Simulator:

If you have GPU, then you can ask people in slack to share 50Hz simulator. . It produces better angle smoothing. Strictly not for windows and cpu users.Currently what you would download for the link provided by Udacity, uses 10Hz simulator."

Add your comment...

Patrick Kern • Dec 08, 2016

7

@Paul Heraty

I know a lot of students are struggling with this project and it think it's great that you share your experiences in the forum to help them. However I'm a bit concerned since for me your post sound like a checklist to follow or otherwise the project is not doable which just isn't true. There a tons of possible solutions to this challenge and no right or wrong way. I would really like students to stay open minded and come up with unique solutions.

I was able to build a model using the 10hz simulator with joystick input using transfer learning (a model from scratch also worked). For the first track I din't need 40k images and I also didn't do any "successive refinement".

Just training with adam optimizer with default learning rate. The key for me was doing a lot of image argumentation on the fly while training.

I believe @Henrik Tünnermann was able to build a model using the 50hz simulator using keyboard input even without smoothing.

Others were working with smoothing and filter algorithms to improve the quality of keyboard input.

Some things I would like to point out:

- The 50hz suimulator was the default simulator but there were serious performance issues especially for windows users. It was basically not usable. The udacity team is working to improve the simulator and to make it usable for everyone even with out gaming experience.
- The number of epochs to train depends on the model and the data (quantity and quality). Also a low mse doesn't nessesaraly mean your car will drive well. I discarded models which had way lower mse then my final model because they drove horribly.

Best wishes

Patrick

Comment

Soon Cheong • Dec 08, 2016
I think it is safe to say that no one seems to be able to get it working with the "default 10hz simulator and keyboard" that I suspect most people started with. Previous advices by others were valuable but they left too possibilities to experiment with. For example, when some said he needed 50k samples but didn't say whether the simulator was 10hz or 50hz and the former requires 5x time to collect the same amount of data. My impression is that those who have successfully completed this project have access to good computer,GPU and joystick. So I assume those without will struggle to finish it, these are exactly the same group of people that do not have resources and time to experiment with seemingly endless combination of things. Since Udacity is not providing help (I didn't receive any notification that they were working on new simulator and did they realise the project deadline is 3 days away?) I think it is good that Paul shared a setup that was proven to work so now I know I can move on to study the advance lane finding and come back to this problem again after my joystick has arrived.

Paul Heraty • Dec 08, 2016
Fair points Patrick. I will edit the post to say it's not the only way, but it is 'a way'. In general, I'd like to think that there's value in seeing the points. If I were starting out again, I'd love to get some pointers like the above.

Aside: successive refinement and transfer learning are somewhat similar, no? You're taking something that works to a degree, and refining it 🙂

Patrick Kern • Dec 08, 2016
@Paul Heraty. I agree that you points a very helpful. Just want to make sure students don't give up to quickly and settle for a given approach. I think experimenting with different ideas is great way to learn. Of course it takes a lot of time and can some times be frustrating but also very rewarding when ending up with a working solution.

To your side node: Yes it's similar. Difference being that the model in tranfer learning was trained on a different problem and also you starts out with pretrained weights. With successive refinement you start basically from scratch with the problem at hand which might take more time but probably yields better results?!?

Aitor Ruano • Dec 08, 2016
I am not able to find the 50 Hz simulator version in Slack, can someone provide a link to it? 🙂

Patrick Kern • Dec 08, 2016
Hey @Aitor Ruano,

I only found the link to the linux simulator. https://carnd.slack.com/files/paulheraty/F3A47CT7Z/simulator-linux.zip

Aitor Ruano • Dec 08, 2016
Thanks! Exactly what I needed 🙂

Shirish Jamthe • Dec 09, 2016
Patrick,

How can I download the 50hz simulator? I only have access to nd013 not carnd. thx

Patrick Kern • Dec 10, 2016
@Shirish Jamthe https://carnd-slack.udacity.com/

Wei-Chung Chen • Jan 23, 2017
could you explain some of the image augmentation methods a bit? I did not know it before. Thanks.

Add your comment...

Joel Bertomeu • Dec 09, 2016

5 Same, here but still didn't get it working... using Ubuntu I spent several days getting steering angles = -1, 0 and 1 and thinking they were correct until I suspected they were not, and asked... 2 days later I got the response they were not correct so I had to install windows and run it on it... since until now there is no other solution.

Then, the problem with loading images into memory causing a Memory error...

Then, blindly training on my CPU... taking forever...

Then, setting up amazon AWS and after 1 day realizing that using python2 was directly defaulting to use the CPUs and not the GPUs...

Then moving to python3...

Now, everything working but I am using the keyboard and in all fairness the sampling time seems quite low... it might be that the problem is this... I am kind of drained... after many days and many many more hours than 10 hours/week...

Seriously I like challenges and I like a lot learning but there are way too many things to learn here in one go, the ones who got it working first time right either you have some experience on this or you are geniuses...

Comment

Paul Heraty • Dec 09, 2016
Stick with it Joel, and use the Slack channel for help. Sounds like you've already made a lot of progress!

Dhruv Parthasarathy [Administrator] • Dec 09, 2016
@Joel Bertomeu we are going to upload our own training data for you to work from by Monday. Hopefully that will help. This project is difficult but I've done it on my own and I'm confident you can too! Also, feel free to wait till Monday for the data.

Prakhar Sharma • Dec 10, 2016
@Dhruv Parthasarathy [Administrator] submission deadline for the project remains Monday?

Joel Bertomeu • Dec 10, 2016
Good question...

Jon Hauris • Jan 05, 2017
Where is this uploaded data located? Cant find it.

David A Ventimiglia • Jan 07, 2017
@Dhruv Parthasarathy [Administrator] where were the training data uploaded?

JR Heard • Jan 07, 2017
@Dhruv Parthasarathy [Administrator] I would also like the training dataset 🙂

Add your comment...

Gil Carmel • Dec 09, 2016

4 Great post and great thread, everyone! Getting this project working reminds me of when I first learned to program, and I felt like a magician.

Comment

John Chen • Dec 10, 2016

3 If you want to do refined tuning, you now have an additional option besides P3: Want to use Agile and Deliberate Practice Approaches to Train your Model?. @Thomas Antony has re-imagine the live trainer and put together a keyboard version! It is located here: https://github.com/thomasantony/sdc-live-trainer. Thanks @Thomas Antony!

Comment

Shirish Jamthe • Dec 09, 2016

3 The keyboard controls are pretty bad so I decided to write a Computer Vision based lane detection and steering control first. Just by using my code from project1. Each run is going for 4-6 laps and produces 4000-6000 images.

I have uploaded the code to github https://github.com/sjamthe/Self-Driving-Car-ND-Predict-Steering-Angle-with-CV

With that I am training my CNN model. The only drawback of this method is that I only get center image from the server. It will be good it the sent left and right images too.

Comment

Henry X • Dec 09, 2016
Intuitive

Add your comment...

 Stewart DeSoto • Dec 09, 2016

2

Just to emphasize the point that the 11 rules are helpful, but not necessary. I was able to get a model that has the car successfully driving around the course indefinitely with only 6k samples, using the slow 10Hz keyboard input, and training on a Macbook with 8GB RAM, without using Python generators, and with just 5 training epochs. But it took a heck of a lot of work.

Comment

Henry X • Dec 09, 2016
What hack lot of work? Is it patch the sampling at points of failure?

Stewart DeSoto • Dec 10, 2016
No, just playing around with architectures, regularization approaches, training optimizers, various training sets. The number of combinations rapidly approaches infinity.

I tried to avoid patch sampling at failure points.

Yu Shen • Jan 22, 2017
What's your batch size with 5 training epochs with which that you successfully trained? Is it the default 128 images per epoch? Thanks.

Add your comment...

 Arnaldo Gunzi • Feb 02, 2017

1

Reading this post and other tips from this forum, I could complete this very hard project.

Here follows a very detailed post about things I learned and reflections on this project:

https://medium.com/@arnaldogunzi/teaching-a-car-to-drive-himself-e9a2966571c5#.d06wvx7hy

Comment

Jon Hauris • Feb 02, 2017
@Arnaldo Gunzi Great write up Arnaldo. Thanks for the effort and the very interesting work!

1. Looking at your stuff it looks like you only use the Udacity provided data. And then form that you really only use the center image. Is that correct?

2. Then it seems like the only pre-processing that you do is to take 8000 random samples of the provided data, crop it, resize it, take random sample of half of these and flip them, and then normalize. Is that correct in what you are doing? Have I missed anything?

3. I am doing something very similar except I am included the left and right images and adjusting the steering angle offset for them.

4. I am doing the Nvidia model but not including pooling. It was not in their paper. Do you think that is important?

5. Could you please explain a bit more about your first Convolution2D(1,1,1, ...) do? The Nvidia model has the 3 channel data going straight into the (24,5,5) kernel which outputs a 24 channel map. What are you doing with the (1,1,1) kernel which outputs a 1 channel map? Sorry, I don't understand that.

6. Finally can you please elaborate on the following line in your model:

```
p[i]=float(row[2])
```

Thanks for any help and clarification you can give. Jon

Arnaldo Gunzi • Feb 02, 2017
Hi @Jon Hauris.

1. I used the three cameras.

4. I used the pooling, because without it the memory requirement was greater.

5.The model was good, till the curve after the bridge. My guess was that there was no lane there, and the model got confused. I was trying to use grayscale, channel S of HSV and so on, to make the model distinguish asphalt and unpaved road. But, instead of me choosing the right channel, I asked Keras to do it. The input of this layer has 3 channels, the output, 1 channel, keeping the same dimension.

But the main mistake I was doing was not it. The main mistake was that I didn't noticed opencv imread function gives BGR, and the drive.py reads in RGB.

6. I did a preprocessing to match images and angles in an auxiliar file. The line you quoted is the steering angle. But it is a particular solution, not important to understand the whole picture of the problem.

Dushyant Wadivkar • Feb 15, 2017
Hi @Arnaldo Gunzi, great write up. It is preventing me from given up.

I am now making it as far as the curve after the bridge and then veering straight. I have tried patch training and it makes it worse. I am a python newbie and would appreciate if you could tell me how you fixed the RGB to BGR fix.

Thanks,

Dushyant

Arnaldo Gunzi • Feb 15, 2017
Hi @Dushyant Wadivkar.

If the image is read by opencv imread() function, it will load as BGR. If you're using other function, perhaps it is reading in RGB, you'll have to read the documentation of the function.

If it is BGR, just use opencv conversion function.

imgRGB = cv2.cvtColor(imgBGR, cv2.COLOR_BGR2RGB)

Add your comment...

Thomas W • Jan 24, 2017

1

I want to start the 3rd project behavioral cloning, but I had to shudder after reading all of these threads... Before I start, I wanted to ask you for some advise for starting:

- I have Windows 10 and Ubuntu 14 on my laptop. After reading the input of Joel Bertomeu, would you recommend me to start with windows?
- I have a rather weak laptop without GPU. Can I use Amazon AWS for this project? (is it possible to open and play the simulator with SSH?)

Probably, before I start I will take some time first and read a lot of "lessons learned" from all of you 🙂

Joel Bertomeu Comment

Matt Harrington • Jan 29, 2017
@Thomas W: You can train on AWS or Azure and then copy your model & weights to your PC for use with the simulator. I use Windows with Anaconda, and it works just fine.

Add your comment...

Jon Hauris • Jan 05, 2017

1

@Paul Heraty. In your #11 what is meant by negative, straight, and right steering?

Comment

Paul Heraty • Jan 09, 2017
Hi Jon. I mean steering angles inthe .csv file, where a negative values means steer left, straight is 0.0 and means no turn/angle in the steering wheel, and a positive value means steer right. These should have associated images depicting left turns, straights and right turns.

Tim Slator • Jan 18, 2017
Is the suggestion to run just those three images repeatedly, or should they be used as the basis for generating augmented images?

Yufeng Guo • Jan 21, 2017

I *think* he means a dataset of literally 3 images. The implementation might look more like some sort of generator that wraps it, maybe a little wobble or something if you're feeling fancy?

There was a more detailed comment (on this very page I think) on this point.

Paul Heraty • Jan 22, 2017
Yes, I mean train with a dataset of just 3 images. What you're trying to prove is that your model can determine a steering angle from an image, and if you run prediction on the same 3 images, you should get predictions that match the original labels.

Then once you have this working, you have high confidence that your network architecture is appropriate. You can then train with a larger dataset.

Add your comment...


Ralph Tigoumo • Feb 06, 2017

0

Comment


Ralph Tigoumo • Feb 06, 2017

0

Comment


Dennis Chew • Jan 30, 2017

0

An advice i could give is to try out your model at low speeds, 10mph, 15 mph, 20 mph and slowly stepping it up. Do not be too upset if it is unable to reach top speed as hardware matters in this project.

The same model on a 2015 Macbook Air (2.2GHz i7) reached top speed of only approx 18 mph but was able to achieve top speed of 30 mph on i7 4.2GHz/GTX 1070.

Comment


Hristo Vrigazov • Jan 29, 2017

0

One thing I want to emphasize is that the quality of the data really matters. When I was starting this project 4 weeks ago, I used the simulator with the keyboard because there was nothing other available and then wasted a lot of time trying to make the model work. Then wasted a lot of time in trying to smooth it out, etc. Then Udacity released sample data, but even on that my model was not able to learn very well, even after applying augmentation. I then recorded my own data using the mouse simulator, which was still kind of crappy and wasted more time there.

So 4 weeks later and after exhausting all my AWS GPU credit, I am sadly back to the drawing board. I have a model which can make a few laps, but is really not reliable enough, can't make it every time. I have tried Nvidia's paper model and Comma.ai's, as well as some designed by me. I also tried VGG16, but it does not fit into memory of the g2 instance. Not sure how people have used it, perhaps they have their own GPUs?

One thing that improved the learning of my model was to lower the batch size to 16, not sure why this happens. Also, resizing the input of the image to something small, 80x60 for example also helped.

Another note: I regularized every layer using l2, and this prevented my model from constantly outputting the same angle, but made it more uncertain.

I won't be able to make this project when its due, but I can still pass if I make the project till the end of the term, right? Because now I also have finals in university and it seems almost impossible to finish the project tomorrow.

Comment

Nico Lindsey • Jan 29, 2017
I train locally on a 970 ti. It's well worth it. I find it to be a lot faster than the aws gpu's too.

Hristo Vrigazov • Jan 31, 2017
Well good for you, but I don't have 970 ti.

Add your comment...

Waldemar Gessler • Jan 25, 2017

0

Comment

Waldemar Gessler • Jan 25, 2017

0

Comment

Dmitry Yanchenko • Jan 24, 2017

0

How have you approached generalisation from track1 to track2? It's quite easy to fit the track1 with any 3+ conv layers architecture, but quite difficult to generalise to track2, training only on track1 data. How has anybody approached this problem?

Comment

Xuanzhe Wang • Jan 22, 2017

0

@Dhruv Parthasarathy [Administrator] why there is a space after each "," in the csv generated by the game and also in the data provided by Udacity?

Though I can parse it, but it will waste some time. Is this a bug of the game?

Comment

Frank Wilson • Jan 23, 2017
Personally I load the data with pandas and it works fine. It also makes it quite easy to filter (such as modifying file paths) and adding and using column names in the code.

Matthew Zimmer • Jan 23, 2017
Great question! In my honest and humble opinion as a senior engineer who has nearly "seen it all" is that this is intentionally left there; it's an Easter egg of sorts. I smiled as soon as I saw it as I thought it was quite clever. Of course, it may not be intentional, but I truly hope it is never cleaned up in the source file because it teaches students how to "clean dirty data"; an invaluable lesson to learn for everyone looking to form a career out of the education provided in this Nanodegree.

David Silver • Jan 23, 2017
It's not an intentional bug. We're not quite that devious.

I will check with the simulator engineers to see if we can remove that space. It hasn't caused me any problems in parsing so I hadn't really noticed it. The Python `csv` library seems to parse it just fine.

Nonetheless, if it's an easy fix we're happy to make it. Thanks for the heads-up.

Matthew Zimmer • Jan 23, 2017
Nice. Thanks, David. I still think it's a cool bug.

Xuanzhe Wang • Jan 23, 2017
@David Silver, maybe it's the version I use or something. I use pandas too. It left a space in the result dataframe for each path str.

Though It's an easy fix, but I just want to make sure if it's a bug or an "evil" intention.

Also, can someone vote this thread up? I don't think this post is invaluable.

Add your comment...

---

fera0013 • Jan 10, 2017

0

Thanks a lot for your getting started, it really helps to avoid frustrating pitfalls.

You mention

"So when you get a model working a little bit, say passing the first corner, then use that model as a starting point for your next training session (kinda like Transfer Learning). "

Why not ACTUALLY using Transfer Learning, instead of doing something "kinda like Transfer Learning" ? Did you try that approach, or is there some reason that transfer learning wouldn't work in this case?

Comment

Paul Heraty • Jan 22, 2017

No reason why you couldn't actually use transfer learning. This was just the way I approached the problem. Hence the opening "Note: These points below are not the 'only' way of solving this problem. Think of them as pointers and feel free to pick and choose as you see fit" 🙂

Add your comment...

Your answer

Paragraph B I U

Save Cancel