"Do Computers Dream of Electric Carriages?": Implementing a (potentially) Novel Neural Network Architecture for Preliminary Design of Electric/Hybrid Cars

Andre Nakkab

Abstract

In this project, I performed a comparative analysis of different makes and models of electric/hybrid car in order to train a multi-faceted neural network to suggest the best possible basic design elements for new car designs. By combining a Generative Adversarial Network with an identification network, which I called a Labeler, I was able to target specific labels for generation. Normally, when given an unbalanced dataset, a GAN will settle into producing items of data that are representative of the majority of the dataset. For the car generator, this meant cars of the average rating, i.e. 6's with the occasional 7. Implementing the additional training step of targeting allowed me to train the network to only generate high-quality vehicular designs. In fact, this method would allow me to generate cars of any specific rating just by modifying this targeting step. This may illuminate some aspects of vehicular design that are key to improving the overall quality of the car, but perhaps more importantly, it may have wide-reaching applications in other fields. I will be referring to this architecture as a Targeted GAN, or TGAN.

GitHub repository:

https://github.com/ajn313/Deep-Learning-Final-Project-Fall2021

Intro

Literature Review

Generative design techniques have recently been used for development of specific parts in vehicular design. For example, Airbus utilized AI to design structural components of the cabin partitions in their A320 commercial aircraft.[1]

More specifically to cars, BMW has an entire department dedicated to implementing AI with regards to designing cars with improved performance, and has even acquired companies specializing in machine learning to further this goal.[2]

As might be expected, most research done in this field utilizes generative adversarial networks. Due to the unstable nature of the training process of a generative adversarial network, in which both the generator and discriminator are trained simultaneously, certain considerations must be given to its design. For example, ReLU is the suggested activation function for intermediate steps, but Tanh() should be used as the last step for the generator, and Sigmoid() should be used as the last step for the discriminator. Additionally, an Adam optimizer should be used (Brownlee, 2019).

Similar work to this project has been done in the field of medical imaging, albeit with somewhat inverse methodology. For example, one group of researchers used a DCGAN to generate instances of chest X-ray images for underrepresented illnesses (Venu et al, 2020). Another team did something similar for meteorological imaging (Fang et al., 2018). However, both of these projects relied on the small existing dataset of those specific chest X-rays/weather patterns to train their DCGAN. This method could circumvent that problem. I would have to do a deeper lit review to be absolutely certain, but it does seem like this is a novel methodology to solve problems like this one.

Data

Specification details for various vehicles were obtained through The Car Connection, which also provides expert ratings on overall vehicular quality.[3]
These ratings will be used as labels to be correlated with a range specifications of the car itself. By default, the dataset includes 109 input parameters.

As can be seen from the sample in Figure 1, some of the parameters are undefined for individual vehicles (NaN). The parameters were pruned to disinclude redundant parameters and those likely unrelated to rating or performance, such as gas mileage which does not apply to electric cars, or part order codes which are not actually related to the design of the car.

In Figure 2, we have a sample of the pruned dataset, with the categorical parameters moved to the front for easier encoding. This data was collected automatically using a script written specifically for this project, implemented with the aid of the URLlib and Beautiful Soup packages for Python. It obtains the HTML files from the website, and parses the data to extract the relevantly tagged keys and values, then uses the Python csv library to collate the data into a neat Excel file.
The data was then manually pruned. Additionally, missing

Name	Rating	Gas Mileage	Engine	EPA Class	Style Name	Drivetrain	Passenger Capacity
2021 Kia Niro	6	51 mpg City/46 mpg Hwy	Gas/Electric I-4, 1.6 L	Small Station Wagons	EX Premium FWD	Front Wheel Drive	5
2021 Tesla Model 3	8	NaN	Electric	Midsize Cars	Long Range AWD	All Wheel Drive	5
2020 Tesla Model S	8	NaN	Electric	Large Cars	Long Range AWD *Ltd Avail*	All Wheel Drive	5
2021 Chevrolet Bolt EV	7	NaN	Electric,	Small station wagon	5dr Wgn LT	Front Wheel Drive	5
2021 Ford Mustang Mach-E	9	NaN	Electric	Small Station Wagons	California Route 1 RWD	Rear Wheel Drive	5

Figure 1: Sample from Unpruned Dataframe

Name	Rating	Engine	EPA Class	Drivetrain	Body Style	Brake Type
2021 Kia Niro	6	Gas/Electric I-4, 1.6 L	Small Station Wagon	Front Wheel Drive	Sport Utility	4-Wheel Disc
2021 Tesla Model 3	8	Electric	Midsize	All Wheel Drive	4dr Car	4-Wheel Disc
2020 Tesla Model S	8	Electric	Large Cars	All Wheel Drive	4dr Car	4-Wheel Disc
2021 Chevrolet Bolt EV	7	Electric	Small Station Wagon	Front Wheel Drive	4dr Car	Partially Regenerative Electro-Hydraulic
2021 Ford Mustang Mach-E	9	Electric	Small Station Wagon	Rear Wheel Drive	Sport Utility	4-Wheel Disc

Figure 2: Sample from Finalized Dataframe

parameters, due to the Car Connection pages being incomplete or errors on the part of the data collection script, were filled in manually using data from manufacturer websites. The data was then normalized and one-hot encoded for use in training the network.

Input parameters selected include things like Engine Type, EPA Class, Transmission Type, Base Curb Weight, etc. The full dataset with a list of parameters will be included with my submission, as well as the pruned version used for the project.

Methods/Model

All of the data was compiled into an easily accessible PyTorch datset/dataloader, with numerous utility functions to aid in the training of the model. The model consists of a generative adversarial network and a Labeler, constructed using the PyTorch neural network library. The GAN uses ReLU activation for intermediary layers in both networks, Sigmoid activation for the final Discriminator layer, and Tanh activation for the final Generator layer. The Labeler uses ReLU activation alone. Each network utilizes an Adam optimizer. Binary Cross Entropy loss was used to train the GAN, as is standard, and Cross Entropy loss was used to train the Labeler.

Hyperparameters selected include a learning rate of 0.0001 for the Labeler, a learning rate of 0.001 for the Generator and the Discriminator. Both the Labeler and Discriminator were designed with 4 hidden layers of sizes 1024, 512, 256, and 128. The Generator was designed with 3 hidden layers, of sizes 512, 256, and 128, and takes as input an array of noise sized as the batch size x 100. A batch size of 5 was used for training, and the torch.randn() function was used to generate noise.

Combining these into the structure of the TGAN, the generator goes the an additional training step of Cross Entropy loss using the predicted labels from the Labeler. Since the Labeler is not trained to identify false inputs, however, this can lead to errors in the values for the one-hot encoded categorical parameters. To circumvent this, rather than averaging the binary cross entropy loss from the Discriminator's predictions and the cross entropy loss from the Labeler's predictions, the losses were weighted at 2/3 for the Discriminator's predictions, and 1/3 for the Labeler's predictions.

The network was trained for 500 epochs, using my NVIDIA RTX 2080 Ti graphics card.

After training, the generated samples were fed through a function to de-encode and de-normalize the output parameters, and printed in a legible fashion.

The rating scale used ranged from 5 to 9, as examples of vehicles rated as 10's or as worse than 5 were incredibly rare and would only serve as outliers.

Results

As a control, the network without the Targeting step was trained to generate cars. As expected, it generated cars with specifications which were representative of the majority of samples in the dataset. These were rated as mostly 6's, with

the occasional 7
For example:

Rating: 6
Engine Type: Electric ; EPA Class: Midsize ; Transmission Type: 1
Drivetrain: Front Wheel Drive ; Body Style: 4dr Car ; Brake Type: 4-Wheel Disc
Front Wheel Material: Aluminum ; Rear Wheel Material: Aluminum
Suspension Type - Front: Strut ; Suspension Type - Rear: Torsion Beam
Passengers: 4 ; Passenger Doors: 5 ; Base Curb Weight: 3352.5176
Front Leg Room(in): 42.420338 ; Front Shoulder room(in): 52.962883
Front Head Room(in): 39.331444 ; Track Width - Front: 62.026535 ; Track Width - Rear 61.654778
Height: 59.594288 ; Length: 181.32388 ; Wheelbase: 112.41529 ; Width Max w/o Mirrors: 72.39361
SAE Net Horsepower: 168.46411 ; Rear Break Ratio: 11.022988 ; Front Break Ratio: 11.776337
Front Wheel Size: 104.36987 ; Rear Wheel Size: 106.62739 ; Basic Miles: 6000
Corrorsion Years: 5 ; Drivetrain Years: 7
Air Bag-Side Body-Rear: Yes ; Air Bag-Side Head-Front: Yes ; Air Bag-Side Head-Rear: Yes
Child Safety Rear Door Locks: Yes ; Daytime Running Lights: Yes ; Rollover Protection Bars: Yes
Fog Lamps: Yes ; Parking Aid: Yes ; Back-Up Camera: Yes

As we can see, this is very much a believable set of specifications for a vehicle, and is a close match with other electric cars on the market. However, herein lies the problem with the standard GAN that the TGAN addresses. This model, statistically speaking, will never generate a 9. With the addition of the targeting step, I was able to train the generator to generate not only believable cars, but high quality cars. Functionally, this means adding a cross entropy loss function between the Labeler's predicted labels of the generated data, and a vector of the target label. In this case, that is target fours, as the set of labels is [5,6,7,8,9], with the index of rating 9 being 4. One example:

Rating: 9
Engine Type: Electric ; EPA Class: Large Cars ; Transmission Type: 1
Drivetrain: All Wheel Drive ; Body Style: 4dr Car ; Brake Type: 4-Wheel Disc
Front Wheel Material: Aluminum ; Rear Wheel Material: Aluminum
Suspension Type - Front: Double Wishbone ; Suspension Type - Rear: Multi-Link
Passengers: 4 ; Passenger Doors: 5 ; Base Curb Weight: 4553.1777
Front Leg Room(in): 42.292812 ; Front Shoulder room(in): 56.16861
Front Head Room(in): 38.358616 ; Track Width - Front: 62.906033 ; Track Width - Rear 65.458824
Height: 54.523083 ; Length: 194.00633 ; Wheelbase: 108.93307 ; Width Max w/o Mirrors: 75.45264
SAE Net Horsepower: 311.98157 ; Rear Break Ratio: 13.249602 ; Front Break Ratio: 14.006966
Front Wheel Size: 145.28474 ; Rear Wheel Size: 143.22247 ; Basic Miles: 4500
Corrorsion Years: 4 ; Drivetrain Years: 10
Air Bag-Side Body-Rear: Yes ; Air Bag-Side Head-Front: Yes ; Air Bag-Side Head-Rear: Yes
Child Safety Rear Door Locks: Yes ; Daytime Running Lights: Yes ; Rollover Protection Bars: Yes
Fog Lamps: Yes ; Parking Aid: Yes ; Back-Up Camera: Yes

The specifications of this car resemble a high-end car, like a Tesla Model S or a BMW 7-Series, but with additional safety features, and tweaked dimensions/performance specifications. Looking at a larger number of samples, we can begin to see patterns here, of which dimensions, features, and performance specs are most important to vehicular quality. I believe further investigation into these specifications is warranted in this situation, however that is outside the scope of this project.
With regards to the loss functions at play, the dual-training method had an interesting effect on the Generator's loss over time, as can be seen in Figure 3. The differing loss functions seem to continuously pull the generator in different directions, which would explain the wildly variable loss values.
We also see that the Discriminator's loss over time very closely resembles that of the Labeler, which stands to reason, as they have similar designs and perform similar tasks.
Additionally, it should be noted that the Labeler's final accuracy was around 94%, giving us a high degree of confidence in the generated results.
As I hoped, the finalized Generator for the TGAN produces nothing but cars rated as 9's, as can be seen from the following examples:

Car 1
Rating: 9
Engine Type: Intercooled Turbo Premium Unleaded I-6, 3.0 L ; EPA Class: Large Cars ; Transmission Type: 8
Drivetrain: Rear Wheel Drive ; Body Style: 4dr Car ; Brake Type: 4-Wheel Disc
Front Wheel Material: Aluminum ; Rear Wheel Material: Aluminum
Suspension Type - Front: Double Wishbone ; Suspension Type - Rear: Multi-Link
Passengers: 4 ; Passenger Doors: 5 ; Base Curb Weight: 4962.7393
Front Leg Room(in): 43.0801 ; Front Shoulder room(in): 57.4098
Front Head Room(in): 40.855194 ; Track Width - Front: 63.27289 ; Track Width - Rear 64.59242
Height: 56.075737 ; Length: 204.92033 ; Wheelbase: 130.74065 ; Width Max w/o Mirrors: 76.68552
SAE Net Horsepower: 290.15787 ; Rear Break Ratio: 14.632092 ; Front Break Ratio: 15.222887
Front Wheel Size: 147.75504 ; Rear Wheel Size: 137.16666 ; Basic Miles: 49496.105
Corrorsion Years: 4.445423 ; Drivetrain Years: 11.724068
Air Bag-Side Body-Rear: Yes ; Air Bag-Side Head-Front: Yes ; Air Bag-Side Head-Rear: Yes
Child Safety Rear Door Locks: Yes ; Daytime Running Lights: Yes ; Rollover Protection Bars: Yes
Fog Lamps: Yes ; Parking Aid: Yes ; Back-Up Camera: Yes

Car 2
Rating: 9
Engine Type: Intercooled Turbo Premium Unleaded I-6, 3.0 L ; EPA Class: Large Cars ; Transmission Type: 8
Drivetrain: Rear Wheel Drive ; Body Style: 4dr Car ; Brake Type: 4-Wheel Disc
Front Wheel Material: Aluminum ; Rear Wheel Material: Aluminum
Suspension Type - Front: Double Wishbone ; Suspension Type - Rear: Multi-Link
Passengers: 4 ; Passenger Doors: 5 ; Base Curb Weight: 4965.914
Front Leg Room(in): 43.091736 ; Front Shoulder room(in): 57.41596
Front Head Room(in): 40.874043 ; Track Width - Front: 63.28329 ; Track Width - Rear 64.5764
Height: 56.109024 ; Length: 205.05852 ; Wheelbase: 130.98735 ; Width Max w/o Mirrors: 76.710655
SAE Net Horsepower: 289.53622 ; Rear Break Ratio: 14.6458845 ; Front Break Ratio: 15.253819
Front Wheel Size: 147.94154 ; Rear Wheel Size: 137.01448 ; Basic Miles: 49562.316
Corrorsion Years: 4.4531126 ; Drivetrain Years: 11.735649
Air Bag-Side Body-Rear: Yes ; Air Bag-Side Head-Front: Yes ; Air Bag-Side Head-Rear: Yes
Child Safety Rear Door Locks: Yes ; Daytime Running Lights: Yes ; Rollover Protection Bars: Yes
Fog Lamps: Yes ; Parking Aid: Yes ; Back-Up Camera: Yes

Car 3
Rating: 9
Engine Type: Electric ; EPA Class: Large Cars ; Transmission Type: 1
Drivetrain: All Wheel Drive ; Body Style: 4dr Car ; Brake Type: 4-Wheel Disc
Front Wheel Material: Aluminum ; Rear Wheel Material: Aluminum
Suspension Type - Front: Double Wishbone ; Suspension Type - Rear: Multi-Link
Passengers: 4 ; Passenger Doors: 5 ; Base Curb Weight: 4553.1777
Front Leg Room(in): 42.292812 ; Front Shoulder room(in): 56.16861
Front Head Room(in): 38.358616 ; Track Width - Front: 62.906033 ; Track Width - Rear 65.458824
Height: 54.523083 ; Length: 194.00633 ; Wheelbase: 108.93307 ; Width Max w/o Mirrors: 75.45264
SAE Net Horsepower: 311.98157 ; Rear Break Ratio: 13.249602 ; Front Break Ratio: 14.006966
Front Wheel Size: 145.28474 ; Rear Wheel Size: 143.22247 ; Basic Miles: 44627.223
Corrorsion Years: 3.727656 ; Drivetrain Years: 9.978254
Air Bag-Side Body-Rear: Yes ; Air Bag-Side Head-Front: Yes ; Air Bag-Side Head-Rear: Yes
Child Safety Rear Door Locks: Yes ; Daytime Running Lights: Yes ; Rollover Protection Bars: Yes
Fog Lamps: Yes ; Parking Aid: Yes ; Back-Up Camera: Yes

Car 4
Rating: 9
Engine Type: Electric ; EPA Class: Large Cars ; Transmission Type: 1
Drivetrain: All Wheel Drive ; Body Style: 4dr Car ; Brake Type: 4-Wheel Disc
Front Wheel Material: Aluminum ; Rear Wheel Material: Aluminum
Suspension Type - Front: Double Wishbone ; Suspension Type - Rear: Multi-Link
Passengers: 4 ; Passenger Doors: 5 ; Base Curb Weight: 4792.99
Front Leg Room(in): 42.887028 ; Front Shoulder room(in): 57.50068
Front Head Room(in): 37.757545 ; Track Width - Front: 62.694294 ; Track Width - Rear 65.82798
Height: 56.254658 ; Length: 188.08661 ; Wheelbase: 113.247955 ; Width Max w/o Mirrors: 76.93451
SAE Net Horsepower: 338.0023 ; Rear Break Ratio: 13.713912 ; Front Break Ratio: 13.974908
Front Wheel Size: 145.92473 ; Rear Wheel Size: 151.22794 ; Basic Miles: 46054.105
Corrorsion Years: 3.713604 ; Drivetrain Years: 9.857504
Air Bag-Side Body-Rear: Yes ; Air Bag-Side Head-Front: Yes ; Air Bag-Side Head-Rear: Yes
Child Safety Rear Door Locks: Yes ; Daytime Running Lights: Yes ; Rollover Protection Bars: Yes
Fog Lamps: Yes ; Parking Aid: Yes ; Back-Up Camera: Yes

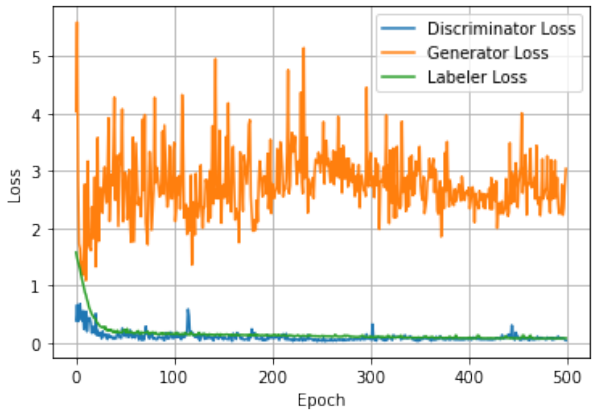

Figure 3: Loss graph for TGAN Car Generator

Car 5
Rating: 9
Engine Type: Electric ; EPA Class: Large Cars ; Transmission Type: 1
Drivetrain: Rear Wheel Drive ; Body Style: 4dr Car ; Brake Type: 4-Wheel Disc
Front Wheel Material: Aluminum ; Rear Wheel Material: Aluminum
Suspension Type - Front: Double Wishbone ; Suspension Type - Rear: Multi-Link
Passengers: 4 ; Passenger Doors: 5 ; Base Curb Weight: 4575.3843
Front Leg Room(in): 42.572758 ; Front Shoulder room(in): 56.884174
Front Head Room(in): 38.60179 ; Track Width - Front: 65.6322 ; Track Width - Rear 66.7841
Height: 56.944805 ; Length: 211.95184 ; Wheelbase: 111.08761 ; Width Max w/o Mirrors: 76.88259
SAE Net Horsepower: 263.8007 ; Rear Break Ratio: 13.961429 ; Front Break Ratio: 17.551115
Front Wheel Size: 158.79376 ; Rear Wheel Size: 138.44577 ; Basic Miles: 49257.164
Corrorsion Years: 3.8867793 ; Drivetrain Years: 11.283527
Air Bag-Side Body-Rear: Yes ; Air Bag-Side Head-Front: Yes ; Air Bag-Side Head-Rear: Yes
Child Safety Rear Door Locks: Yes ; Daytime Running Lights: Yes ; Rollover Protection Bars: Yes
Fog Lamps: Yes ; Parking Aid: Yes ; Back-Up Camera: Yes

In the future, I would like to develop a method to visualize these generated cars, perhaps using something like Unreal Engine. This would likely require a much more purpose-built and detailed dataset. So, for the sake of something that can be easily visualized, I have included a proof of concept for my TGAN architecture using the Fashion MNIST dataset.

Additional Example of the TGAN Architecture, specifically the TDCGAN

The Targeted DCGAN, or TDCGAN, follows a similar logic to the above architecture, instead aimed at targeted image generation. The structure is as follows:
The Labeler is a standard image recognition network, taking in an input of size 28*28, with 3 hidden linear layers of sizes 256, 128, and 64. It utilizes ReLU activation.
The Discriminator is a Convolutional Neural Network, with two hidden layers of sizes 64 and 128, with intermediary LeakyReLU(0.3) activation functions, and Dropout2d(0.3) layers. It then has a linear layer of size 128*7*7, and a final Sigmoid activation function.
The Generator is also a CNN, with an initial linear layer with takes in a noise tensor sized as the batch size x 100, which then has a 1-dimensional Batch Normalization, BatchNorm1d (256*7*7). It then has 3 hidden layers, of sizes 256, 128, and 64, with a final Tanh activation function. It has intermediary LeakyReLU(0.3) activation functions, and 2-dimensional Batch Normalizations of the appropriate sizes.
The learning rate used was 0.0001 for all networks, the batch size was 64 images at a time, and all three networks utilized Adam optimization.
The Labeler was trained using cross entropy loss, with a final accuracy of 96%. The Discriminator was trained using binary cross entropy between real images and fake generated images, and the Generator was trained using an average of the binary cross entropy loss with the predicted real/fake labels of the Discriminator, and the cross entropy loss with the predicted categorical labels of the Labeler. The networks were trained for 50 epochs. The target was label 9, ankle boots.
For reference, the labels of the Fashion MNIST dataset are as follows:
0: T-shirt/top
1: Trouser
2: Pullover
3: Dress
4: Coat
5: Sandal
6: Shirt
7: Sneaker
8: Bag
9: Ankle boot
As a control, I used this architecture to train an DCGAN with a Labeler to label generated images. In Figure 4, we can see that it is quite accurate, and the generated images are very convincing.
Then, applying the TDCGAN methodology, I added the additional training step of targeting specifically ankle boots for generation. This was achieved by an additional cross entropy loss between the Labeler's predicted labels of the generated data, and a vector of target nines. As can be seen in Figure 5, as hoped, the final generator produces nothing but ankle boots. This was a very exciting moment for me, to be completely honest.

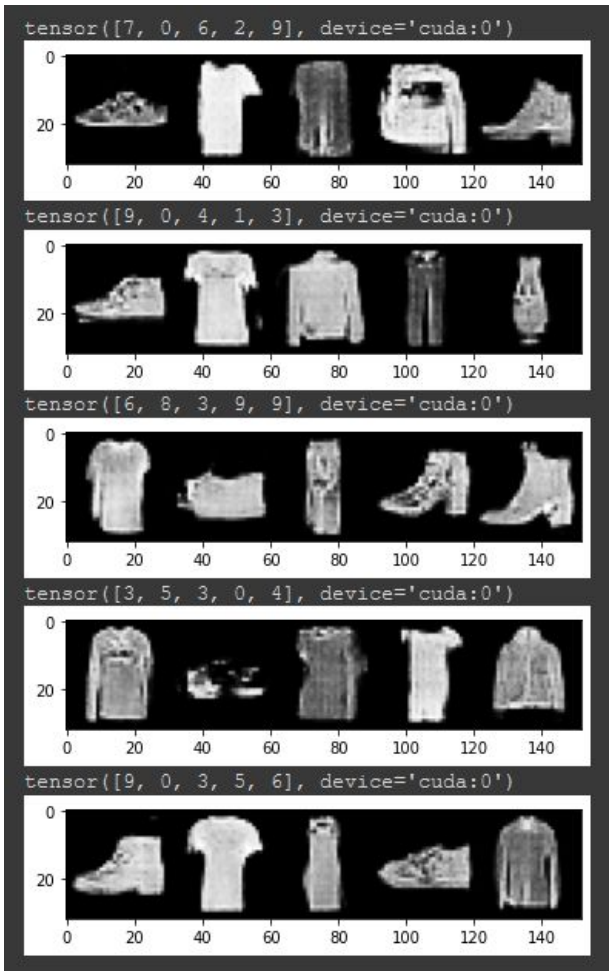

Figure 4: Labels for DCGAN Generated Images

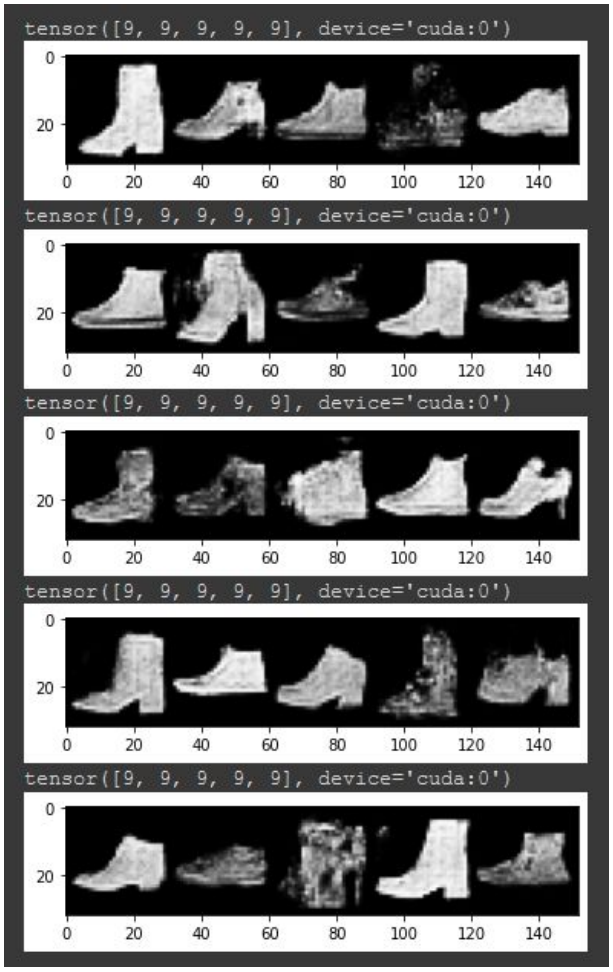

Figure 5: Ankle Boots Only, Generated by TDCGAN

Conclusion

In a field as young as data generation, there is a considerable amount of unexplored potential. As far as I can tell, no one has utilized quite this architecture for quite this problem before. This is a really compelling realization for me, of course, but it also begs the question what other innovations are possible when we are face with new problems to solve and new obstacles to overcome.

I am very proud of the proofs of concept I was able to put together in such a short time here, but I can't help but imagine what could be accomplished with more time and resources. For example, if we used something like ImageNet as the labeler in a TDCGAN structure, we could generate images of anything within the tens of thousands of categories available there, using considerably larger unlabeled image sets than ImageNet normally uses, which would be prohibitive to manually label.

Or, going back to the chest X-ray or meteorology questions I cited earlier, we could generate more genuine-looking instances of images that would be in the extreme minority in

those fields, for training even more accurate image identification networks.
I hope to better flesh out the TGAN concept going forward, experimenting with which designs give the best efficiency and accuracy. I would very much like to develop this idea into a useful part of the metaphorical deep learning tool belt.

References:

1. https://www.bbc.com/future/article/20181129-the-ai-transforming-the-way-aircraft-are-built
2. https://www.forbes.com/sites/bernardmarr/2017/08/01/how-bmw-uses-artificial-intelligence-and-big-data-to-design-and-build-cars-of-tomorrow/?sh=37113d612b91
3. https://www.thecarconnection.com/
4. Brownlee, J. *Tips for Training Stable Generative Adversarial Networks*, 2019. https://machinelearningmastery.com/how-to-train-stable-generative-adversarial-networks/
5. Venu, S., Ravula, S. *Evaluation of Deep Convolutional Generative Adversarial Networks for Data Augmentation of Chest X-ray Images*, 2020. https://www.mdpi.com/1999-5903/13/1/8/pdf
6. Fang, W., Sheng, V. *A method for improving CNN-based image recognition using DCGAN*, 2018. https://www.researchgate.net/publication/328425461$_{A}method_{f}or_{i}mproving_{C}NN-based_{i}mage_{r}ecognition_{u}sing_{D}CGAN$