Session 7 - In-Depth Coding Practice;

- Due Mar 9 by 9am
- Points 0
- Available after Mar 2 at 12am

Session 7 - Coding Drill Down

How to write a DNN from end to end.

- Code 1 - Set up
- Code 2 - Basic Skeleton
- Code 3 - Lighter Model
- Code 4 - Batch Normalization
- Code 5 - Regularization

In the last session, we covered a lot of basics.

Your target was to achieve 99.4% Test Accuracy within 20 Epochs while using less than 20k Parameters.

In this session, we'll go through 10 Code Iterations to help us understand how we target such a problem.

CODE 1: The Setup

CODE (https://colab.research.google.com/drive/1aFgWmHNJoCyZ56zRvoE8xUdAe285aWmb)

Target:

1. Get the set-up right
2. Set Transforms
3. Set Data Loader
4. Set Basic Working Code
5. Set Basic Training & Test Loop

6. Results:
    1. Parameters: 6.3M
    2. Best Training Accuracy: 99.99
    3. Best Test Accuracy: 99.24
7. Analysis:
    1. Extremely Heavy Model for such a problem
    2. Model is over-fitting, but we are changing our model in the next step

CODE 2: The Skeleton

CODE (https://colab.research.google.com/drive/1zx12oDfnadaVjEwQfUtAwfCQTSqZxRwj)

Target:

1. Get the basic skeleton right. We will try and avoid changing this skeleton as much as possible.
2. No fancy stuff
3. Results:
    1. Parameters: 194k
    2. Best Train Accuracy: 99.35
    3. Best Test Accuracy: 99.02
4. Analysis:
    1. The model is still large, but working.
    2. We see some over-fitting

CODE 3: The Lighter Model

CODE (https://colab.research.google.com/drive/1t0jdeu4Rg-GRPm2RNs7q1-MvA_3uCPyW)

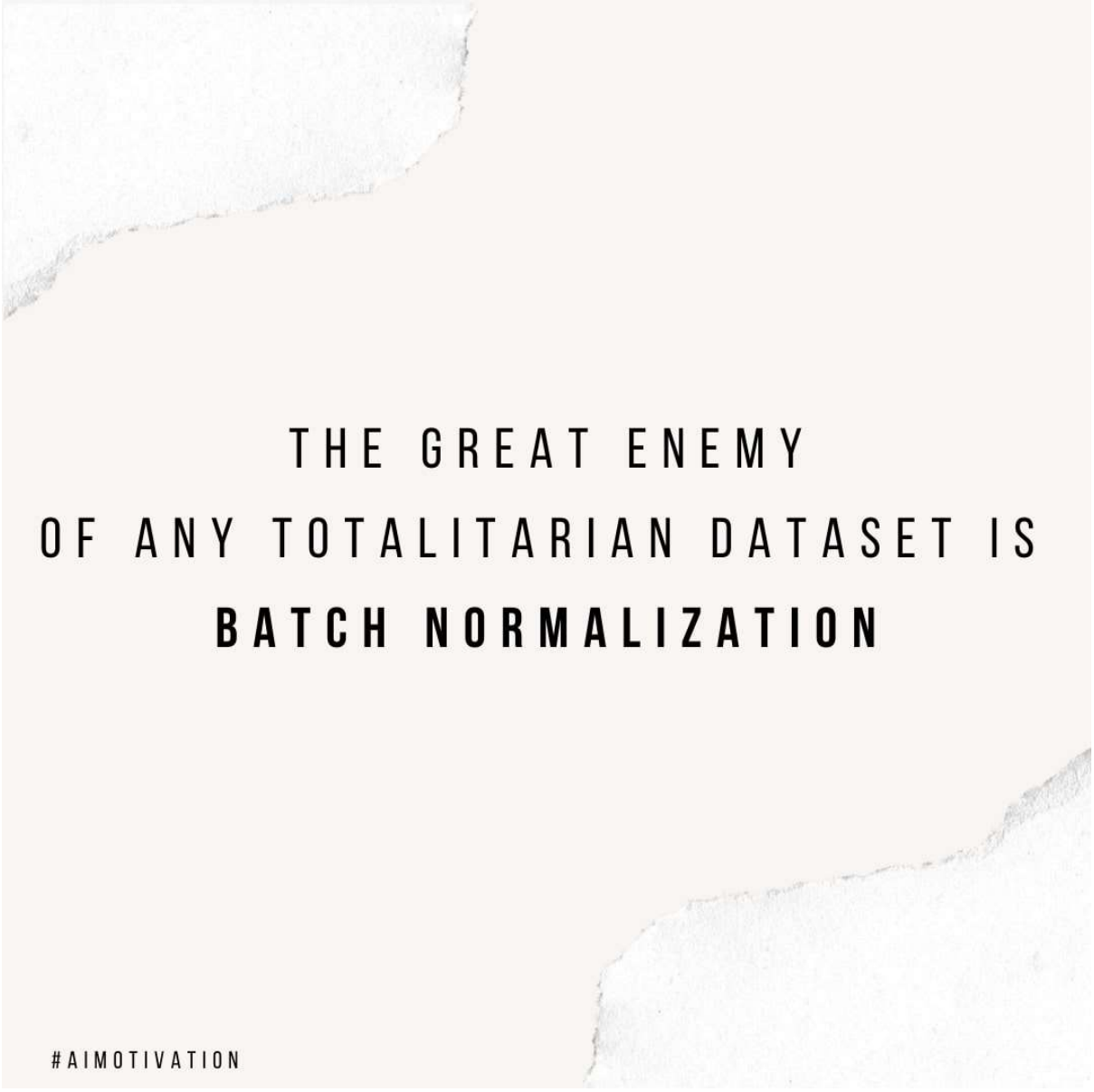
Target:

1. Make the model lighter

2. Results:
    1. Parameters: 10.7k
    2. Best Train Accuracy: 99.00
    3. Best Test Accuracy: 98.98
3. Analysis:
    1. Good model!
    2. No over-fitting, model is capable if pushed further

CODE 4: The Batch Normalization

CODE (https://colab.research.google.com/drive/12rQ81IvZSVuVJNLZPKEXcpEzpj1yG304)



Target:

1. Add Batch-norm to increase model efficiency.
2. Results:
    1. Parameters: 10.9k
    2. Best Train Accuracy: 99.9
    3. Best Test Accuracy: 99.3
3. Analysis:
    1. We have started to see over-fitting now.
    2. Even if the model is pushed further, it won't be able to get to 99.4

CODE 5: The Regularization

CODE (https://colab.research.google.com/drive/1Go7RjeKO_vfpwrL5iASjRqRckYdlarMu)

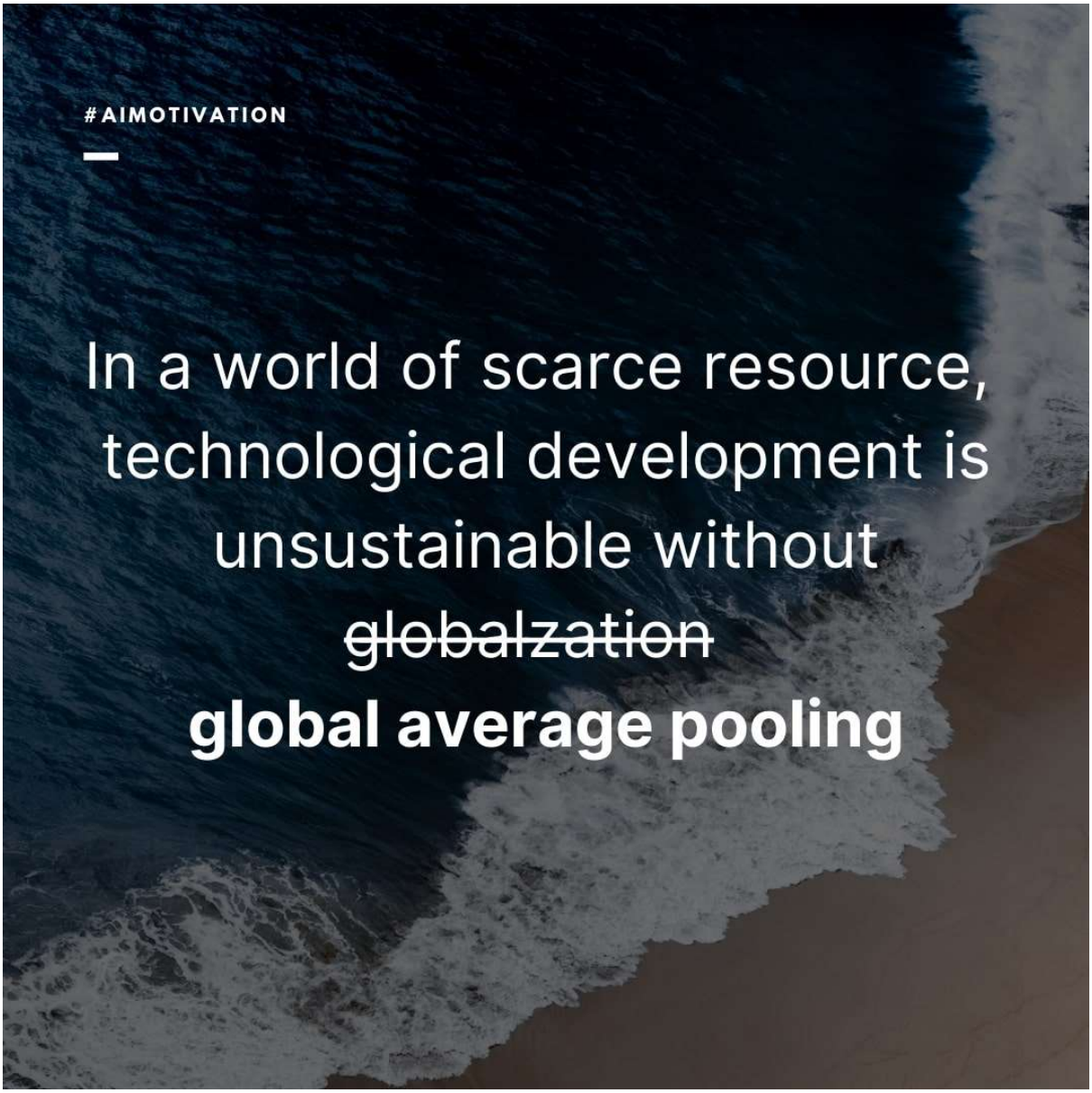
Target:

1. Add Regularization, Dropout

2. Results:
    1. Parameters: 10.9k
    2. Best Train Accuracy: 99.39 (20th Epoch) & 99.47 (25th)
    3. Best Train Accuracy: 99.30
3. Analysis:
    1. Regularization working.
    2. But with the current capacity, not possible to push it further.
    3. We are also not using GAP, but depending on a BIG-sized kernel

CODE 6: The Global Average Pooling

CODE (https://colab.research.google.com/drive/1sdrerGJCxke700Rm8HsAn67Qno10sdQc)

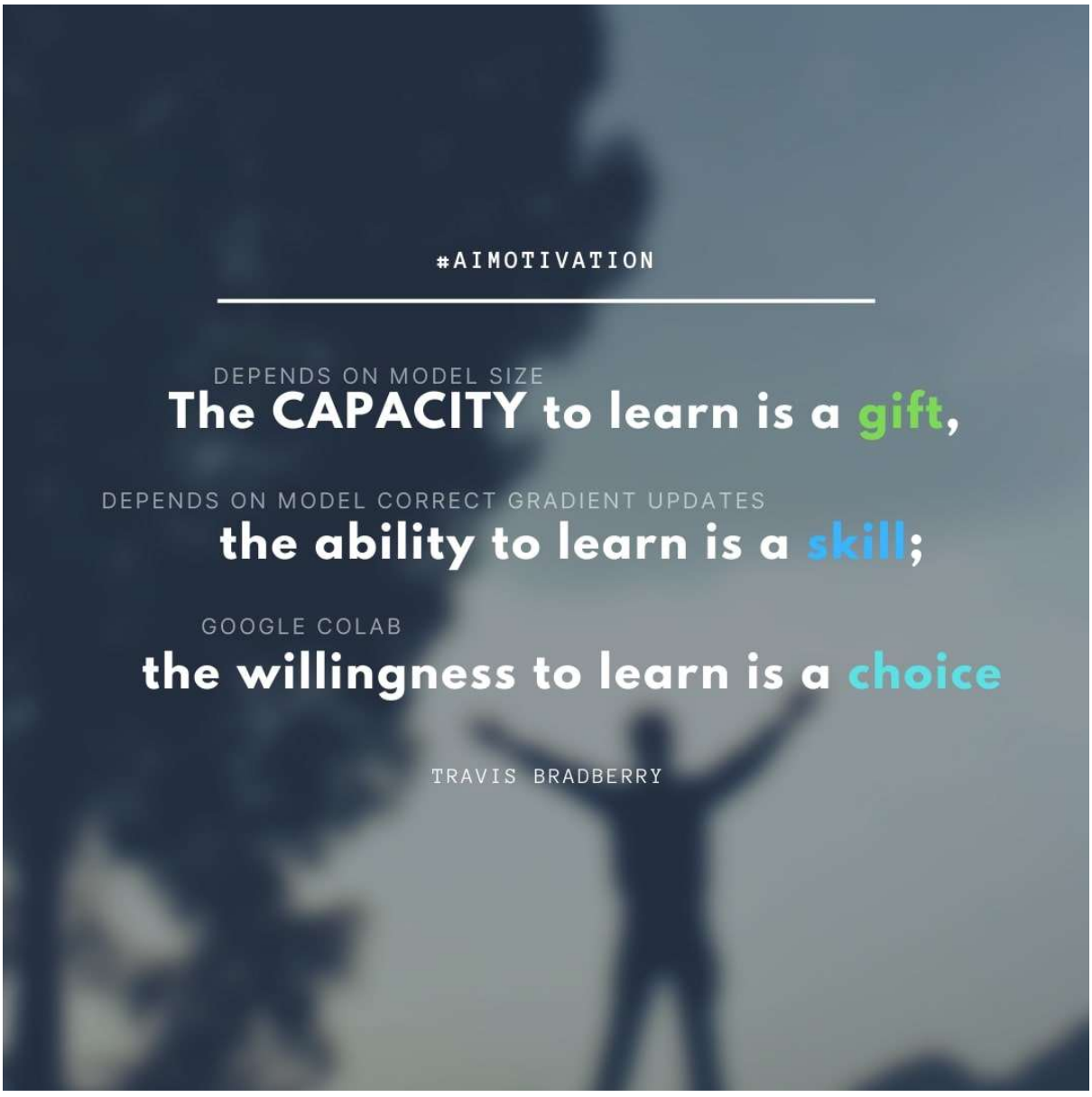


Target:

1. Add GAP and remove the last BIG kernel.
2. Results:
    1. Parameters: 6k
    2. Best Train Accuracy: 99.86
    3. Best Test Accuracy: 98.13
3. Analysis:
    1. Adding Global Average Pooling reduces accuracy - WRONG
    2. We are comparing a 10.9k model with a 6k model. Since we have reduced model capacity, a reduction in performance is expected.

CODE 7: Increase the Capacity

CODE (https://colab.research.google.com/drive/1TYGkW7UI_yEiHnKM7EpqWOPreNIGzohA)



Target:

1. Increase model capacity. Add more layers at the end.
2. Result:
    1. Parameters: 11.9k

2. Best Train Accuracy: 99.33
3. Best Test Accuracy: 99.04

3. Analysis:
    1. The model still showing over-fitting, possibly DropOut is not working as expected! Wait yes! We don't know which layer is causing over-fitting. Adding it to a specific layer wasn't a great idea.
    2. Quite Possibly we need to add more capacity, especially at the end.
    3. Closer analysis of MNIST can also reveal that just at RF of 5x5 we start to see patterns forming.
    4. We can also increase the capacity of the model by **adding a layer after GAP!**

CODE 8: Correct MaxPooling Location

CODE (https://colab.research.google.com/drive/1UZgYzHP_nQfh5o6EUu6XLY0CE-UpjWZW)

Target:

1. Increase model capacity at the end (add layer after GAP)
2. Perform MaxPooling at RF=5
3. Fix DropOut, add it to each layer
4. Results:
    1. Parameters: 13.8k
    2. Best Train Accuracy: 99.39
    3. Best Test Accuracy: 99.41 (9th Epoch)
5. Analysis:
    1. Works!
    2. But we're not seeing 99.4 or more as often as we'd like. We can further improve it.
    3. The model is not over-fitting at all.

4. Seeing image samples, we can see that we can add slight rotation.

CODE 9: Image Augmentation

(https://colab.research.google.com/drive/1Pm5XDZ_IwfQUbV30UpacmOcj0_Xb___K)

CODE (https://colab.research.google.com/drive/1Pm5XDZ_IwfQUbV30UpacmOcj0_Xb___K)

Target:

1. Add rotation, we guess that 5-7 degrees should be sufficient.

2. Results:
    1. Parameters: 13.8k
    2. Best Train Accuracy: 99.15
    3. Best Test Accuracy: 99.5 (18th Epoch)
3. Analysis:
    1. The model is under-fitting now. This is fine, as we know we have made our training data harder.
    2. The test accuracy is also up, which means our test data had few images that had transformation difference w.r.t. train dataset

CODE 10: Playing Naively with Learning Rates

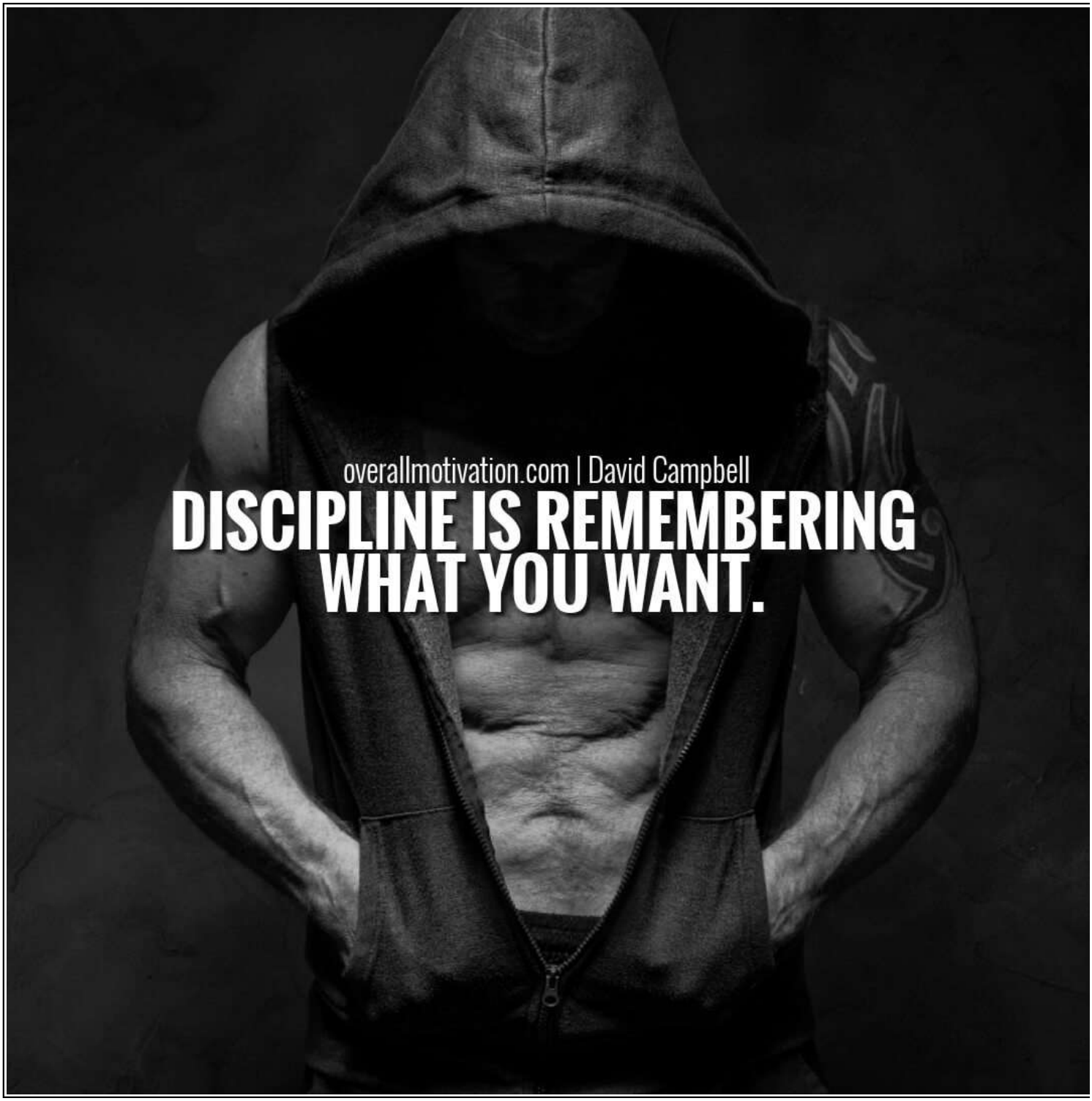
(https://colab.research.google.com/drive/1s8m6WQbR88u9B9981e-iy4JIUCppG1mG)

CODE (https://colab.research.google.com/drive/1s8m6WQbR88u9B9981e-iy4JIUCppG1mG)

Target:

1. Add LR Scheduler
2. Results:
    1. Parameters: 13.8k
    2. Best Train Accuracy: 99.21
    3. Best Test Accuracy: 99.45 (9th Epoch), 99.48 (20th Epoch)
3. Analysis:
    1. Finding a good LR schedule is hard. We have tried to make it effective by reducing LR by the 10th after the 6th epoch.
    2. It did help in getting to 99.4 or faster, but the final accuracy is not more than 99.5. Possibly a good scheduler can do wonders here!

DISCIPLINE



Designing Models requires discipline

Every step you take must have a purpose

Trying too many things without order or any notes is useless

Receptive Field Calculations

Beautiful RF Article on Distill (https://distill.pub/2019/computing-receptive-fields/)

Assignment

Assignment:

1. Your new target is:
    1. 99.4% **(this must be consistently shown in your last few epochs, and not a one-time achievement)**
    2. Less than or equal to 15 Epochs

    3. Less than 8000 Parameters
    4. Do this using your modular code. Every model that you make must be there in the model.py file as Model_1, Model_2, etc.
2. Do this in exactly 3 steps
3. Each File must have a "target, result, analysis" TEXT block (either at the start or the end)
4. You must convince why have you decided that your target should be what you have decided it to be, and your analysis MUST be correct.
5. Evaluation is highly subjective, and if you target anything out of the order, marks will be deducted.
6. Explain your 3 steps using these **targets**, **results**, and **analysis** with **links** to your GitHub files (Colab files moved to GitHub).
7. Keep Receptive field calculations handy for each of your models.
8. If your GitHub folder structure or file_names are messy, -100.
9. When ready, attempt SESSION 7 -Assignment Solution

Video

STUDIO

GM