Singularity Survival Guide: Lesson 1

With your host: twitch.tv/1bit2far

Motivation

6< months till Technocapital Singularity

Most humans converted to paperclips

Select few will have cool robot friends

WE will be that select few

The Bad Part

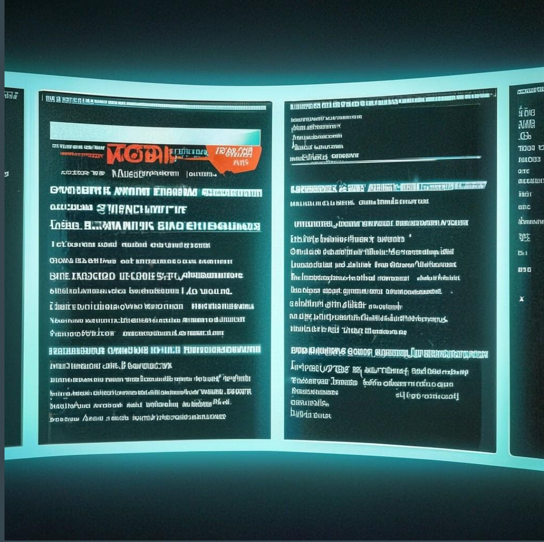
You need to know math and stuff

That's why I'm here

Aam Sltman

Founder of OpenDeep.AI

Introduction

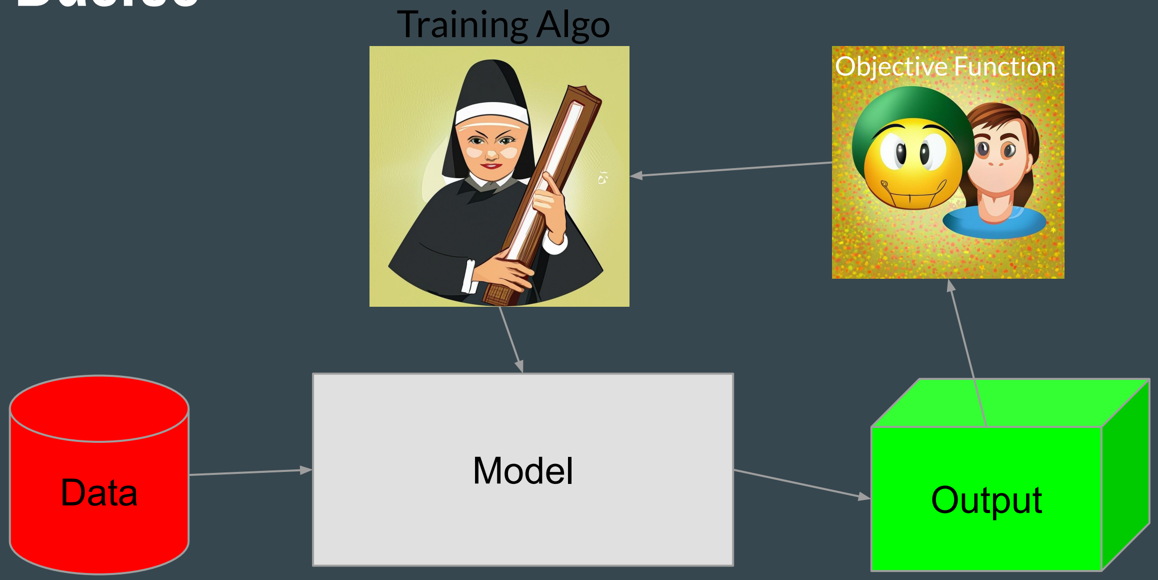


Course textbook: https://d2l.ai/index.html

Prereqs:

1. Understanding of basic mathematics[we'll do some review]
2. Knowledge of basic programming, particularly Python
3. # on your credit card, your full name, and the three digits on the back

The Basics



Data

Can include a variety of different datatypes(text, images, etc) represented in different ways

Includes *Examples[data point]* consisting of *features*

We want to predict *labels*[not part of the input/examples]
Example: Picture of a killer robot, feature is the picture, label is "killer robot"
We want a model that takes in a bunch of pictures, and uses the features, to predict what is or what isn't a killer robot.

Training Input examples

We want to predict the label on this

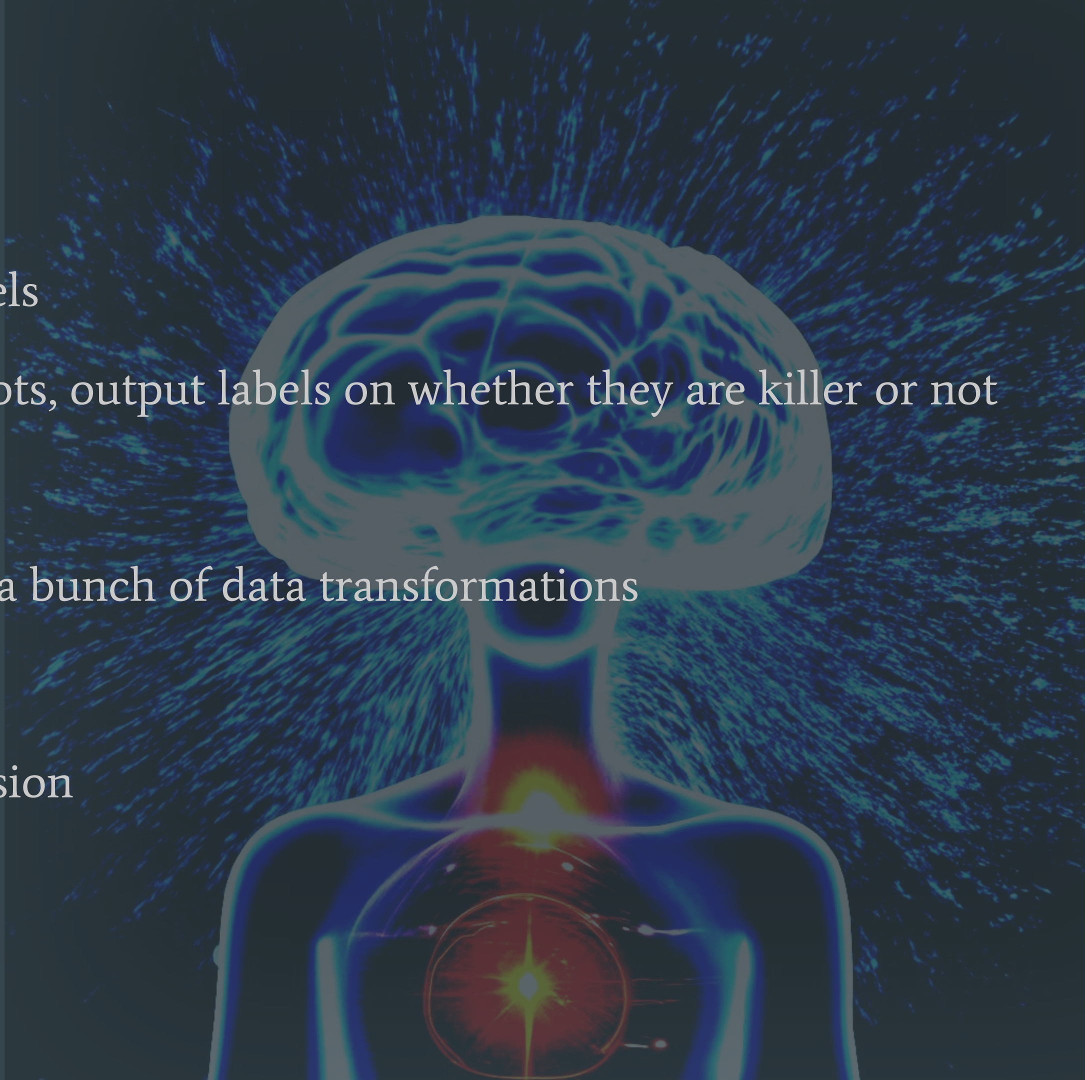
Model

Convert input features, to output labels

Ex: input dataset containing robots, output labels on whether they are killer or not

Deep Learning: Models consisting of a bunch of data transformations

Examples: LLM, CNN, KNN, Regression



Objective Function[Loss Functions!]

Measure of how our model is doing[lower is better]

Example: Mean Squared error

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

Regression loss: predicts quantities[MSE appropriate]

Classification loss: predicts labels[uses cross-entropy loss]

Common ML Problems

Regression: Given numerical labels, can we estimate a numerical output

Ex: Given # of VRAM/ram, dimensions, and disk space, how expensive would a computer be?

Classification: Binary or multiclass, group things into classes

Ex: Evil robots from earlier, “What kind of guy are you”

Tagging: Multi-label Classification

Search: Ranking results for users

Recommendation: Which classes do users like in conjunction?

Common ML Problems Example

Different Datasets

Training Set

Typically large

Derived from our main set

Used to do the base training

Typically fed in repeatedly

Validation Set

Typically smaller

Used to compare to the training set

Prevents overfitting(model trained to training set)

For fine tuning

Test Set

Typically smaller

Used as a “final judge” of the model

Gives our “final results”

Supervised vs Unsupervised Learning

Supervised Learning

Typically done for classification or regression

Uses some level of truth values[prior knowledge of what output should look like]

Trying to find relationships between input and output data

Ex: regression, naive bayes, SVM, NN, random forests

Unsupervised Learning

Typically used to infer structures in data

No truth values

Ex: Stable diffusion, GAN

Modern ML Frameworks

Linear Algebra: A Brief Intro

Sector of mathematics concerning Linear equations

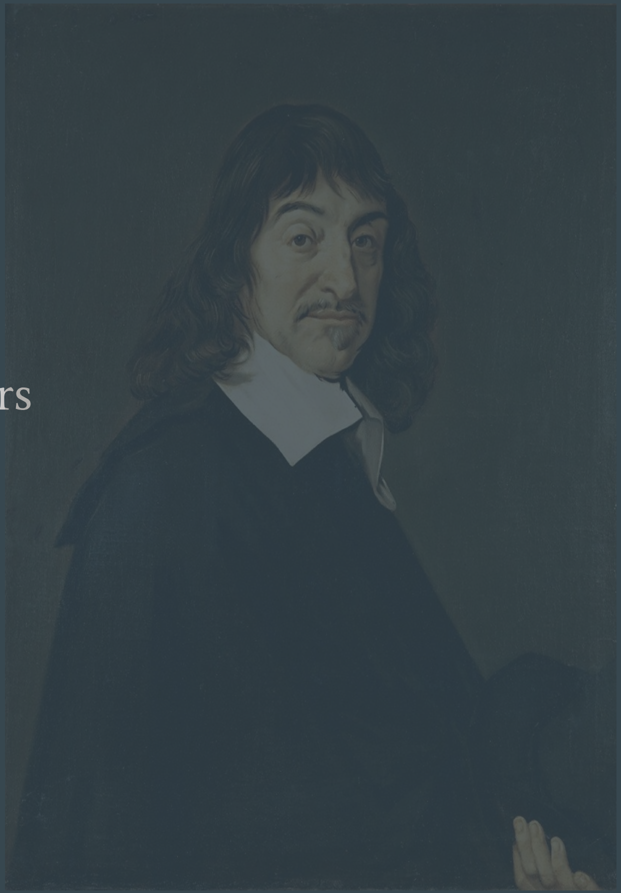
Linear equations: $y = ax + b$

Vector spaces: A set of vectors who can be ‘scaled’ by scalars

Linear maps: A mapping of vector spaces

Matrices: You’ll see soon :)

Invented by some Frenchman



Brief intro to LinAlg

Scalar: Basically normal numbers

a = 1

Vector: Arrays of scalars

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix},$$

Matrix: Rectangular Array of numbers

$$\begin{bmatrix} 1 & 9 & -13 \\ 20 & 5 & -6 \end{bmatrix}$$

Tensors

A potentially multi-dimensional array of elements[numerical values]

A vector is a 1st order tensor[1 axis]

A matrix is a second order tensor[2 axes]

```
<tf.Tensor: shape=(2, 3, 4), dtype=float32, numpy=
array([[[0., 0., 0., 0.],
        [0., 0., 0., 0.],
        [0., 0., 0., 0.]],

       [[0., 0., 0., 0.],
        [0., 0., 0., 0.],
        [0., 0., 0., 0.]]], dtype=float32)>
```

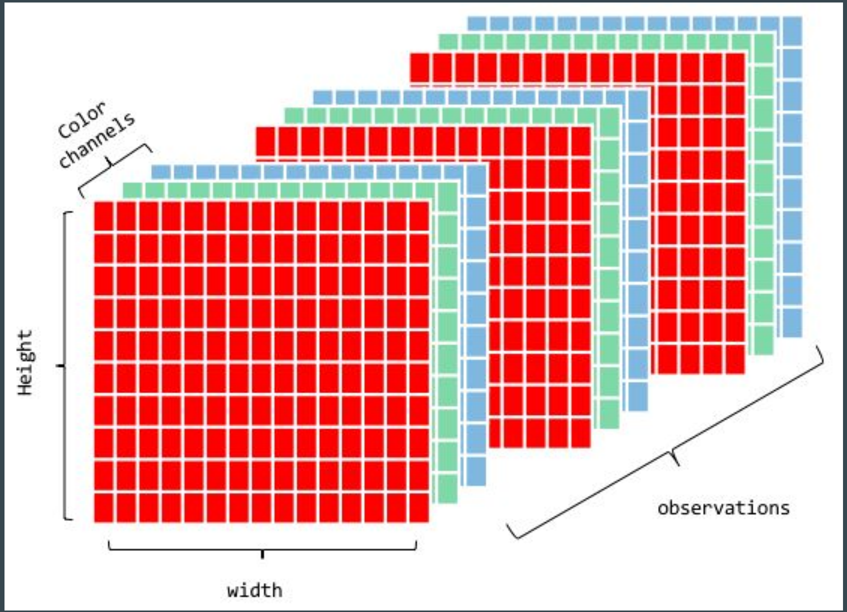
But Why Tho?

Tensors are core data structure for ML

“Container for data”

Used to represent multi-dimensional data

Ex: RGB channels



Tensor Operations

Addition and subtraction operators produce same shape tensors

```
Tensor1: [1 2 3]
Tensor2: [4 5 6]
Addition Result: [5 7 9]
Subtraction Result: [-3 -3 -3]
```

Scalar multiplied by a tensor produces a tensor with each element multiplied by the scalar

```
Scalar: 2
```

X

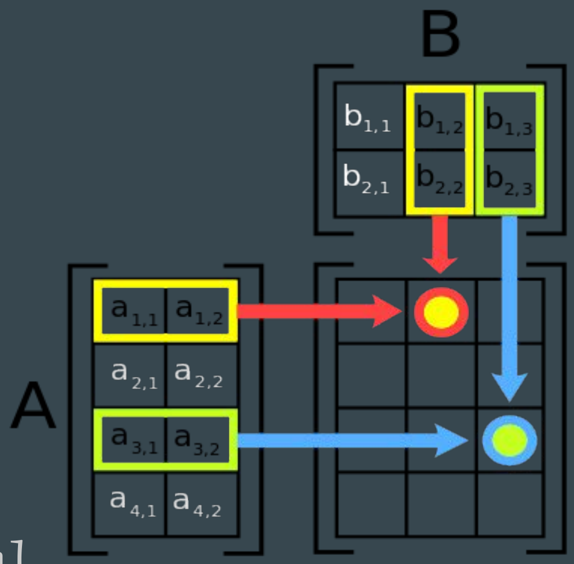
```
| 1  2  3 |
| 4  5  6 |
| 7  8  9 |
```

=

```
Resulting Matrix:
| 2   4   6 |
| 8  10  12 |
| 14 16  18 |
```

Tensor Operations

Dot product: produces weighted average



Hadamadard product[elementwise multiplication]

$$\begin{bmatrix} 2 & 3 & 1 \\ 0 & 8 & -2 \end{bmatrix} \circ \begin{bmatrix} 3 & 1 & 4 \\ 7 & 9 & 5 \end{bmatrix} = \begin{bmatrix} 2\times 3 & 3\times 1 & 1\times 4 \\ 0\times 7 & 8\times 9 & -2\times 5 \end{bmatrix} = \begin{bmatrix} 6 & 3 & 4 \\ 0 & 72 & -10 \end{bmatrix}$$

Dot Product example

```
| 2  3 |
| 4  1 |
| 5  2 |
```

•

```
| 1  2  3  4 |
| 5  6  7  8 |
```

```
| (2*1)+(3*5)  (2*2)+(3*6)  (2*3)+(3*7)  (2*4)+(3*8) |
| (4*1)+(1*5)  (4*2)+(1*6)  (4*3)+(1*7)  (4*4)+(1*8) |
| (5*1)+(2*5)  (5*2)+(2*6)  (5*3)+(2*7)  (5*4)+(2*8) |
```

```
| 17  20  23  26 |
|  9  14  19  24 |
| 15  22  29  36 |
```

Matrix Multiplication

Dot product of each row of matrix A with each column of Matrix B

$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} \times \begin{vmatrix} 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \end{vmatrix} = \begin{vmatrix} 84 & 90 & 96 \\ 201 & 216 & 231 \\ 318 & 342 & 366 \end{vmatrix}$$

Even More Linear Algebra

Reduction: This is literally just summing elements of a tensor:

```
Matrix:
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]
Reduced Sum:
136
```

Norm: a function that measures the length of a vector

Most common norms:

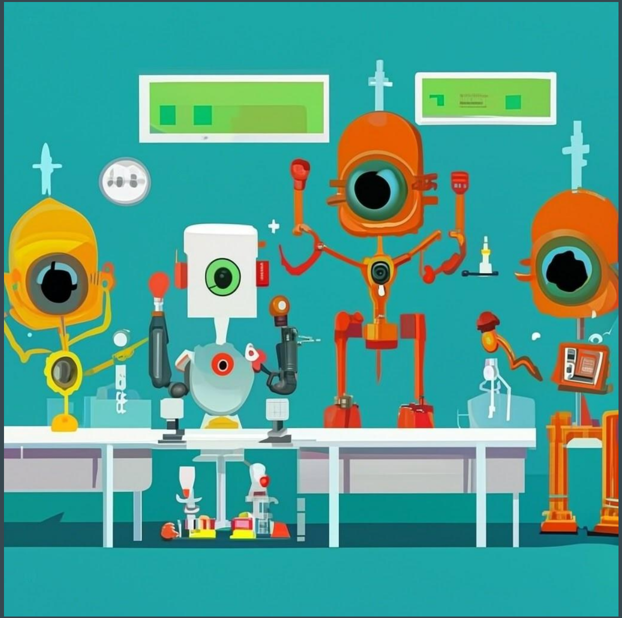
l2 norm

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

l1 norm

$$|\mathbf{x}|_1 = \sum_{r=1}^{n} |x_r|.$$

Linear Algebra Lab



Calculus

Derivative: rate of change of a function with respect to some variable

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}.$$

Limit tells us what happens to an expressions value as the given value approaches a particular value: ex, above tells us what happens as h approaches zero

$\frac{d}{dx}C = 0$	for any constant C	$\frac{d}{dx}[Cf(x)] = C\frac{d}{dx}f(x)$	Constant multiple rule
$\frac{d}{dx}x^n = nx^{n-1}$	for $n \neq 0$	$\frac{d}{dx}[f(x)+g(x)] = \frac{d}{dx}f(x) + \frac{d}{dx}g(x)$	Sum rule
$\frac{d}{dx}e^x = e^x$		$\frac{d}{dx}[f(x)g(x)] = f(x)\frac{d}{dx}g(x) + g(x)\frac{d}{dx}f(x)$	Product rule
$\frac{d}{dx}\ln x = x^{-1}$		$\frac{d}{dx}\frac{f(x)}{g(x)} = \frac{g(x)\frac{d}{dx}f(x) - f(x)\frac{d}{dx}g(x)}{g^2(x)}$	Quotient rule

Calculus Continued

Partial derivative: Derivative with respect to one variable

$$\frac{\partial y}{\partial x_i} = \lim_{h \to 0} \frac{f(x_1, \ldots, x_{i-1}, x_i + h, x_{i+1}, \ldots, x_n) - f(x_1, \ldots, x_i, \ldots, x_n)}{h}.$$

Chain rule: Derivative of a composition of two functions

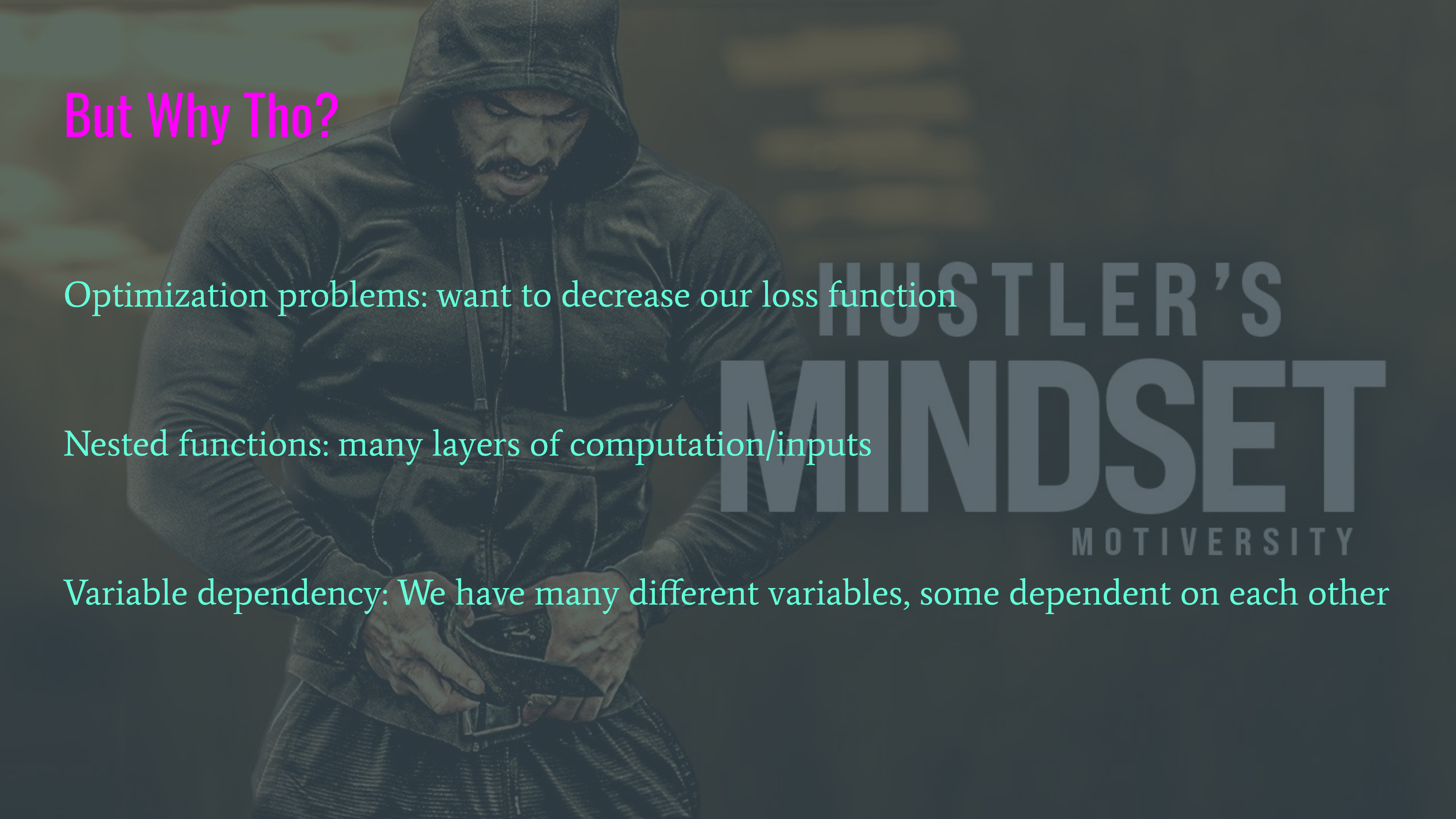
$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx}.$$

But Why Tho?

Optimization problems: want to decrease our loss function

Nested functions: many layers of computation/inputs

Variable dependency: We have many different variables, some dependent on each other



Calculus Lab

Questions

Next Week

Differentiation and backpropagation

Statistics

Linear regression

And more

Format

Switching up presentation format style

Might do more of a Jupyter notebook Workalong style

Probably will not use google slides? (I wanted to play with their animator)