Compilation of Our Best Articles

Team Vizuara

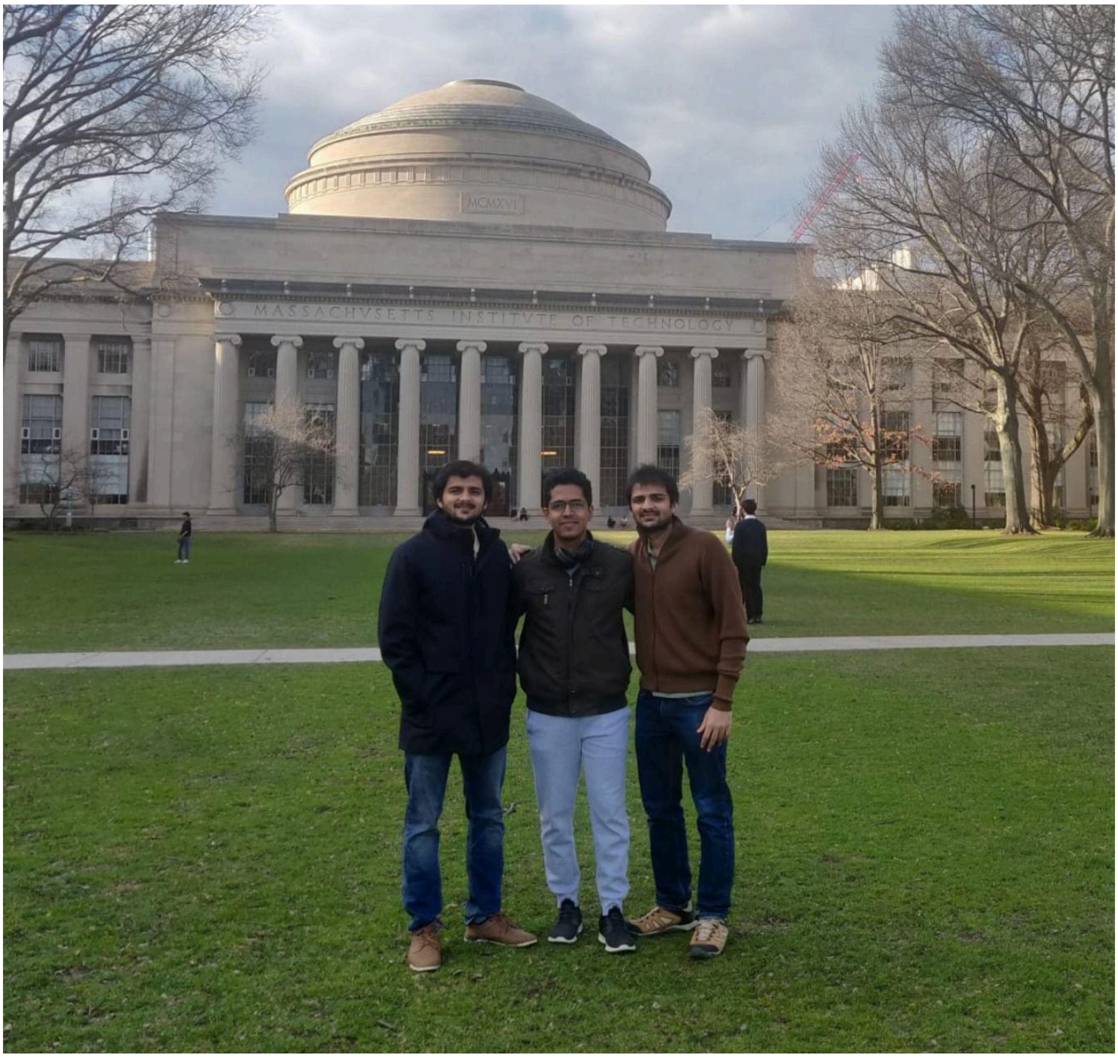


Table of content

Machine Learning

Neural Network to Play Snake Game: Reinforcement Learning

If you give your phone with snake game to a 4-year-old for the first time, he will make random plays at the beginning. You can tell the rules of the snake game to the kid. But still, he may initially play randomly.

After a few games, he will understand what to do and where exactly to take the snake.

You can train a neural network exactly this way to play the snake game or any game using QLearning from Reinforcement Learning.

Define the state of the game at each step as shown below.

1) Collision straight? FALSE
2) Collision right? FALSE
3) Collision left? FALSE
4) Direction West? FALSE
5) Direction East? FALSE
6) Direction North? FALSE
7) Direction South? TRUE
8) Food Westwards? FALSE
9) Food Eastwards? TRUE
10) Food Northwards? FALSE
11) Food Southwards? TRUE

Then modify the state by taking an action: Move forward, left, or right.

There are two kinds of rewards: short-term and long-term.

If the food is touching the wall and if you are close to the food, moving perpendicular to the wall, there is an immediate reward if you move straight toward the food, but you may collide with the wall in the next step.

But if you are parallel to the wall and food is in front of you then immediate reward does not pose imminent danger.

Short-term reward makes the snake eat the food in a given moment. Long-term reward makes the snake maximize the personal best score.

An interesting thing to note is exploration vs exploitation. In the first few games, the snake will make random movements. This will result in many collisions and a lot of negative rewards. Thus, it may learn that it is better to avoid collisions and earn 0 rewards by just moving in circles.

Once the snake plays 75-80 games there is not much exploration. At each step, the snake will take the step proposed by the neural network and will move exactly toward the food. In fact, the snake will play better than you even if you are a pro at the game.

In this lecture I published on Vizuara's YouTube channel, you will learn how to implement this. I also share the GitHub repo with the code. Please feel free to try this yourself: https://lnkd.in/gFY3HMdZ

This approach can be modified to teach AI to play any game. All we need to define is the state at any moment and actions to be taken and a reward/punishment system.

Author: Dr. Sreedath Panat

Differential physics simulations for training neural networks

Ever heard of PhiFlow? It is a differentiable physics simulation toolkit for fluids.

PhiFlow is particularly useful for Physics Informed Neural Networks (PINN). Imagine you are trying to predict how a river flows in different weather conditions. With a regular neural network, you need a huge amount of data to train the model. But with a PINN using PhiFlow, you can teach the model both from the data and from the physical laws that govern water flow, making it much more effective even with less data.

Think of PhiFlow as a way to create virtual experiments on a computer. And you use those experiments and its physics for training a neural network.

Phiflow easily integrates with Tensorflow and PyTorch. In the Neural Network training process, you can define custom loss functions that depend on the outcome of a simulation. For example, in a fluid simulation, the loss function could penalize deviations from expected physical behaviors (e.g. conservation of mass or energy). The neural network is trained to minimize this loss, leading to a model that respects the underlying physics.

If you are interested in PINNs or 2D fluid flows or physics in general, definitely check out their GitHub. Their repo has 1.4k stars: https://lnkd.in/gf3AJ2C5

You can run their fluid simulations super efficiently. For example, I could create this smoke plume simulation in just 1 minute, in about 15 lines of code. Just so cool.

Author: Dr. Sreedath Panat

Regularization and Overfitting

1️⃣Overfitting is when a model fits the training data too well, leading to poor performance on unseen data. This happens because the model becomes too complex and sensitive to the specific examples in the training set.

2️⃣Overfitting can be addressed by reducing model complexity through regularization. Regularization techniques aim to prevent the model from becoming overly complex by penalizing large parameter values or reducing the number of parameters.

3️⃣There are two main ways to achieve regularization: constraining the number of parameters and constraining the size of parameter values.

4️⃣Constraining the number of parameters can involve techniques that promote sparsity, where many parameters are set to zero.

5️⃣Constraining the size of parameter values typically involves adding a penalty term to the cost function, which increases the loss when parameters get too large.

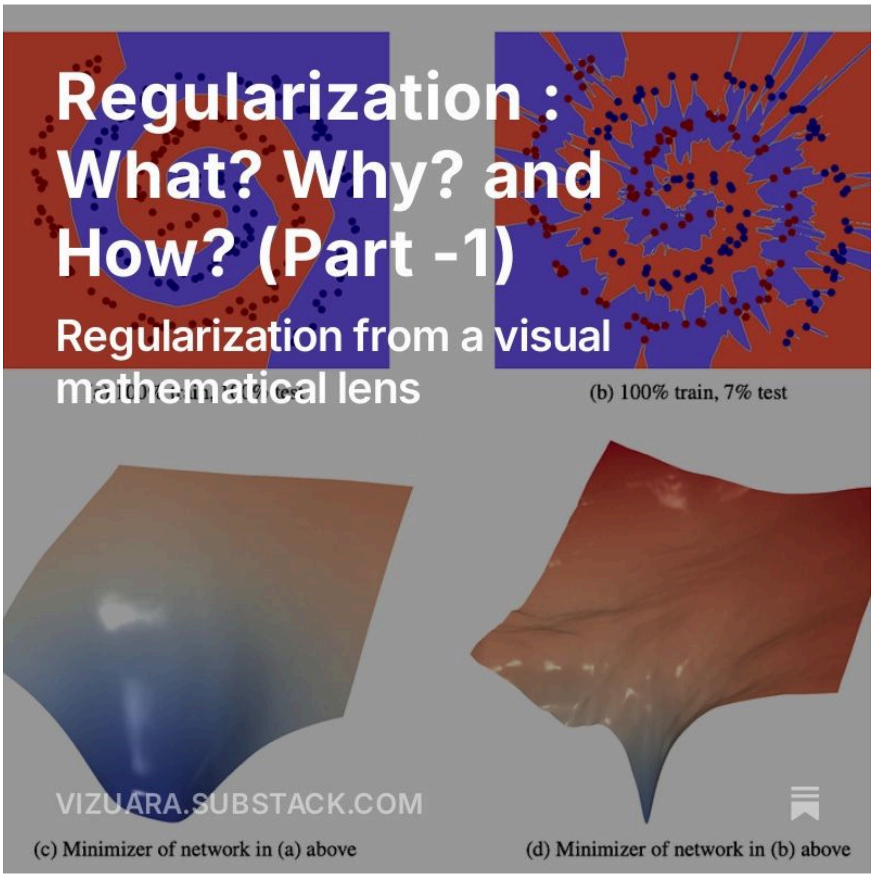
6️⃣Unconstrained optimization can lead to overfitting because the parameters can grow arbitrarily large to fit the training data. This makes the model complex and highly sensitive to small changes in the input.

7️⃣Constrained optimization involves adding penalty terms to the objective function to restrict the parameter values. This helps to prevent overfitting by encouraging the model to find a simpler solution that still fits the data well.

8️⃣The choice of penalty term determines the type of regularization. Common choices include L1 and L2 norms, which penalize the absolute values and squared values of the parameters, respectively.

9️⃣The regularization parameter, often denoted as lambda, controls the balance between fitting the data and keeping the model simple.

🔟 A higher lambda value places more emphasis on regularization, leading to a simpler model but potentially underfitting the data.



Author: Dr. Raj Dandekar

Cheat Sheet to solve any ML problem

1️⃣Understand Your Problem Type

- Start by asking: What are you trying to predict?
- Category/Label (Classification): Examples include spam detection, image recognition, etc.
- Quantity (Regression): Predicting continuous values like price, temperature, or sales.
- Structure or Clusters (Clustering): Grouping similar data points (e.g., customer segmentation).
- Feature Simplification (Dimensionality Reduction): Reducing the complexity of your data for easier analysis.

2️⃣Evaluate Your Dataset:

- Is dataset labeled or unlabeled? (Classification/Regression needs labeled data; Clustering doesn't.)
- How many samples? (<10K, <100K, or more).
- Does it contain text data or numerical features?

3️⃣Classification:
- For text data: Use Naive Bayes.
- For small datasets (<100K samples): Start with Linear SVC or SGD Classifier.
- For larger or complex datasets: Use SVC, Ensemble Classifiers, or KNeighbors.

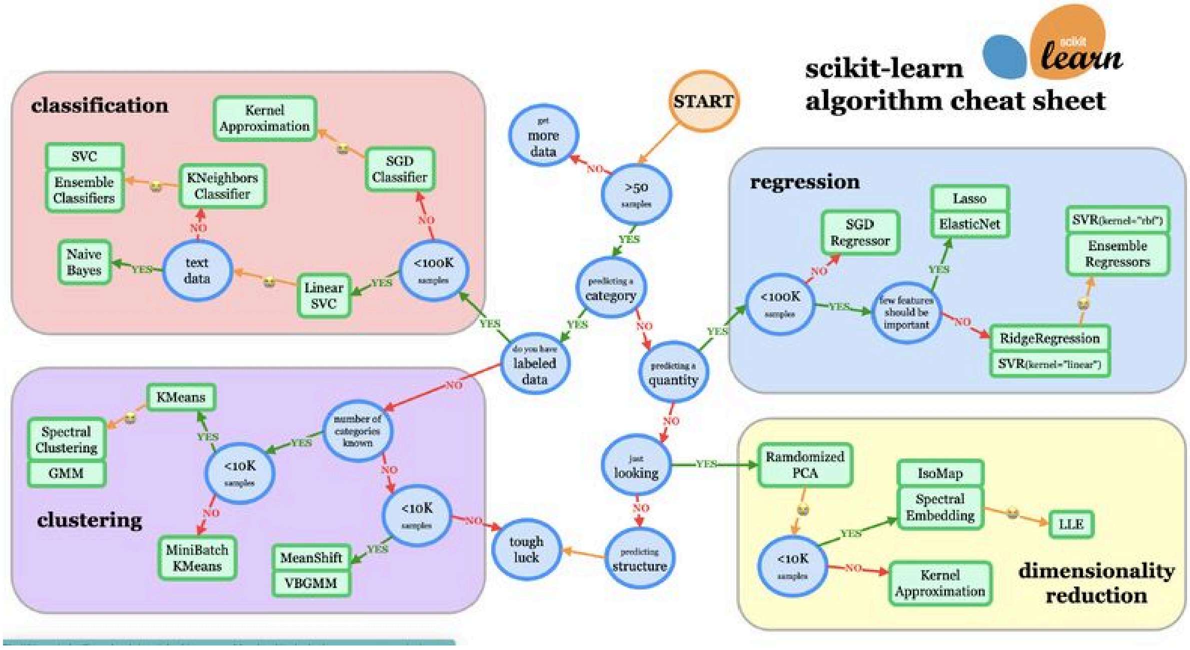
4️⃣Regression:
- Use Lasso or ElasticNet if some features are more important.
- Otherwise, use Ridge Regression or SVR (Linear/RBF Kernel).

5️⃣Clustering:
- Small datasets: Use KMeans if the number of groups is known, or MeanShift if not.
- Large datasets: Try Spectral Clustering or Gaussian Mixture Models (GMM).

6️⃣Dimensionality Reduction:
- Use PCA for linear reduction or IsoMap/LLE for non-linear patterns.

7️⃣Keep on experimenting!
- This cheat sheet provides a starting point, but experiment with hyperparameters, cross-validation, and different algorithms to find the best fit for your specific problem.

Author: Dr. Raj Dandekar


scikit-learn
algorithm cheat sheet
START
classification
regression
clustering
dimensionality reduction
get more data
>50 samples
predicting a category
do you have labeled data
predicting a quantity
just looking
predicting structure
tough luck
<100K samples
text data
Naive Bayes
Linear SVC
KNeighbors Classifier
SVC
Ensemble Classifiers
SGD Classifier
Kernel Approximation
SGD Regressor
Lasso
ElasticNet
few features should be important
RidgeRegression
SVR(kernel="linear")
SVR(kernel="rbf")
Ensemble Regressors
number of categories known
<10K samples
KMeans
Spectral Clustering
GMM
MiniBatch KMeans
MeanShift
VBGMM
Ramdomized PCA
IsoMap
Spectral Embedding
LLE
Kernel Approximation
YES
NO

Anyone can transition to Machine Learning!

I firmly believe that anyone can transition to machine learning, if they have the will power to do so.

One of the biggest mistakes students make at their early stages is the choice of projects.

Choice 1: Toy Kaggle projects

Kaggle is a website which has beginner level ML projects. These projects are good to start doing at the initial stage.

However, adding them on your resume or CV does not add value to your profile.

The reason is that all other students will also have this same point on their CV, since everyone does Kaggle projects.

Kaggle projects are not unique.

Most importantly, these projects can't be converted into impactful research publications.

Choice 2: Research based ML projects

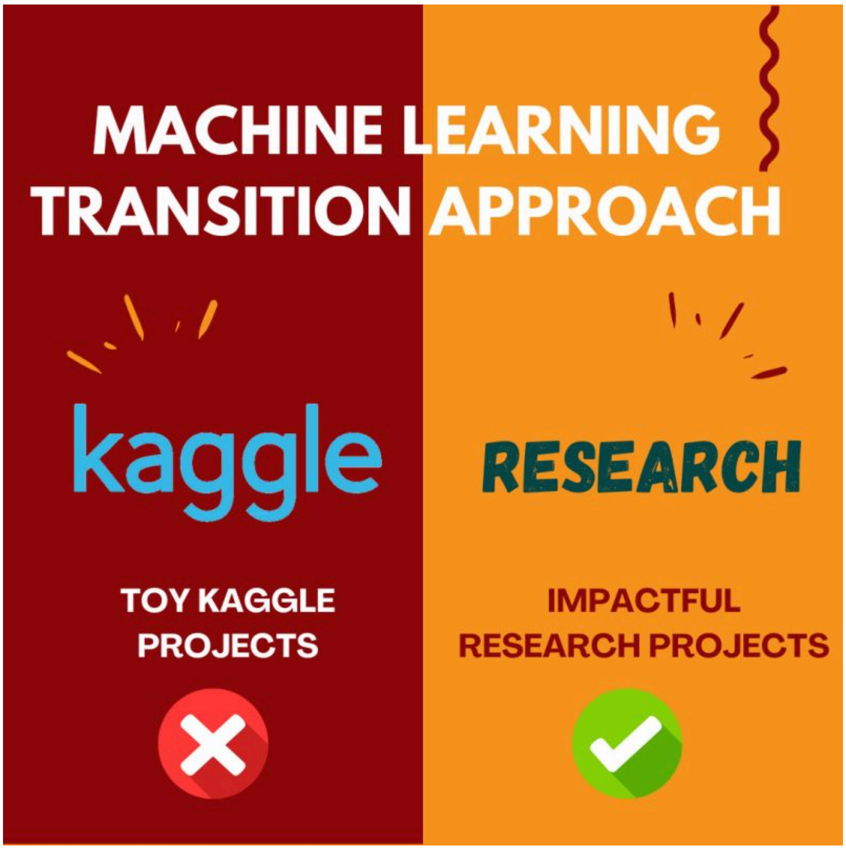
Research based ML projects involve starting with a problem and working on it till it gets converted into a publication: conference, journal or preprint.

Research based projects require patience and take longer than simple Kaggle projects.

However, these projects are unique.

Students who have such projects and eventual publications, are considered to be serious ML engineers.

Author: Dr. Raj Dandekar

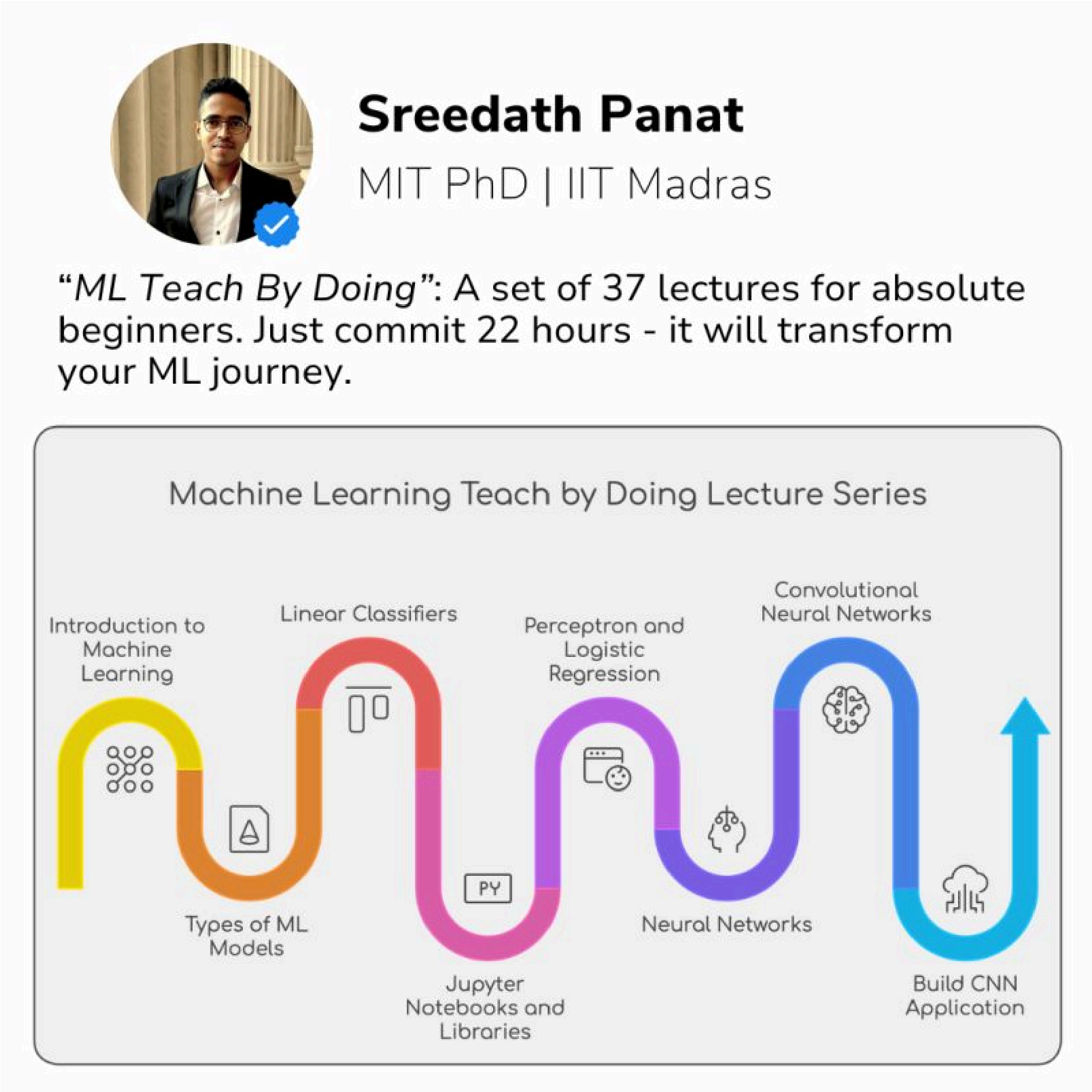

MACHINE LEARNING
TRANSITION APPROACH
kaggle
RESEARCH
TOY KAGGLE
PROJECTS
IMPACTFUL
RESEARCH PROJECTS

Machine Learning Teach by Doing: Lecture series

If you are a beginner and would like to get a deep, foundational understanding here are a set of 37 lectures for you by Dr. Raj Abhijit Dandekar, MIT PhD on Vizuara's YouTube channel. This is by far the best course for beginners I have seen anywhere on the internet.

Introduction
https://lnkd.in/gKr8_EhC
Day 2
https://lnkd.in/gzXbzNeV
Types of ML Models
https://lnkd.in/g5rw8Qbk
The 6 Steps of Any ML Project
https://lnkd.in/gUvM6MRt
Install Python and Run Your First Code
https://lnkd.in/gCnch699
Linear Classifiers Part 1
https://lnkd.in/gDjSqGaC
Linear Classifiers Part 2
https://lnkd.in/gia8Twkd
Jupyter Notebooks
https://lnkd.in/gXnK-FrY
Running the Random Linear Classifier Algorithm in Python
https://lnkd.in/g4XB26UT
The Perceptron Explained
https://lnkd.in/gFepjD7h
Coding the Perceptron
https://lnkd.in/gTfVhKcz
Perceptron Convergence Theorem
https://lnkd.in/gExapstD
Magic of Features
https://lnkd.in/gNjqe7hz
One Hot Encoding
https://lnkd.in/g6izV8z4
Logistic Regression Part 1
https://lnkd.in/gA3YQVvF
Cross Entropy Loss
https://lnkd.in/gwWy_dfk
Gradient Descent
https://lnkd.in/g_PUuKUQ

Logistic Regression from the Ground Up
https://lnkd.in/gh3Dj8Ym
Regularization
https://lnkd.in/gkHeYSRF
Implementing Regularization in Python for Logistic Regression
https://lnkd.in/gjy4N9sV
Linear Regression
https://lnkd.in/geXxr5cC
Ordinary Least Squares Method
https://lnkd.in/gue-ijPH
Ridge Regression Fundamentals and Intuition https://lnkd.in/gcHBvP9Z
Regression for Interviews
https://lnkd.in/g2dFK5WY
Neural Network Architecture
https://lnkd.in/gHqegQsN
Backpropagation
https://lnkd.in/guNdC8i9
Neural Network Activation Functions
https://lnkd.in/gYkduWB3
Momentum in Gradient Descent
https://lnkd.in/gdfJNbrF
Hands-on Neural Network Training
https://lnkd.in/gqfACA_g
Introduction to CNNs
https://lnkd.in/grcZPeqv
Filters in 1D and the Convolution Operation
https://lnkd.in/gxAbd-SX
Filters in 2D, Channels, and Feature Identification
https://lnkd.in/gMT8nP8P
Filtering Layers in CNN
https://lnkd.in/gWXuayfE
What is Max Pooling?
https://lnkd.in/gyeQZrkn
CNN Explained
https://lnkd.in/gCmBzeX3
Backpropagation in CNN
https://lnkd.in/gRQfVqAb
Build Your Brain Tumor Classification CNN
https://lnkd.in/giHeWSf6


Sreedath Panat
MIT PhD | IIT Madras
"ML Teach By Doing": A set of 37 lectures for absolute beginners. Just commit 22 hours - it will transform your ML journey.
Machine Learning Teach by Doing Lecture Series
Introduction to Machine Learning
Types of ML Models
Linear Classifiers
PY
Jupyter Notebooks and Libraries
Perceptron and Logistic Regression
Neural Networks
Convolutional Neural Networks
Build CNN Application

Here is a 5-phase (8-month) roadmap to transition to AI/ML

Sorry there are no shortcuts

Stepping into ML can be difficult, but a clear roadmap can make all the difference. The only resource you need: Vizuara's YouTube playlists.

Phase 1️⃣: Mathematical Foundations (20-25 hours)
Playlist 1: Foundations for ML: https://lnkd.in/gKz-eybU
-Why Begin Here: Grasp the basics- Linear algebra, Probability, Statistics, Calculus, Optimization, Programming fundamentals
-Commitment: 2-3 hours weekly for 8 weeks.

Phase 2️⃣: Machine Learning (60-65 hours)
📌Playlist 1: ML Teach by Doing: https://lnkd.in/gn2dEcE2
-Why It's Important: Practical, project-based learning to understand ML workflows.
-Commitment: 4 hours weekly for 10 weeks.

📌Playlist 2: Decision Trees from Scratch: https://lnkd.in/g3cmj2BR
-Why It's Useful: Master decision tree algorithms are the backbone of many ML models.
-Commitment: 4 hours weekly for 5 weeks.

Phase 3️⃣: Deep Learning (35-40 hours)
📌Playlist 1: Neural Networks from Scratch: https://lnkd.in/gj8kHe2T
-Why It Matters: Understand the mechanics of neural networks through implementation.
-Commitment: 5 hours weekly for 8 weeks.

Phase 4️⃣: Advanced topics: Graph Neural Networks (40-45 hours)
📌Playlist 1: Graph Neural Networks - Theory, Applications and Research:
https://lnkd.in/g3RCPS8e
-Why Learn This: Graph-based ML is becoming increasingly relevant in fields like social networks and biology.
-Commitment: 3 hours weekly for 8 weeks.

📌Playlist 2: ML Project-Based Course: Explainable AI: https://lnkd.in/gNEx3ghr
Why XAI?: Build ML projects with a focus on interpretability
-Commitment: 3 hours weekly for 5 weeks.
-Outcome: Publish your first research paper using XAI techniques.

Phase 5️⃣: Generative AI, Transformers, and LLMs (100-110 hours)
📌Playlist 1: GenAI for Beginners (8 hours): https://lnkd.in/gUgXxVzh
📌Playlist 2: LLMs from scratch (40-45 hours): https://lnkd.in/gjcyfCcE
📌Playlist 3: Hands-on LLMs (40-45 hours): https://lnkd.in/gJQ7ryE4
📌Playlist 4: Transformers (15 hours): https://lnkd.in/g_3Qdu6d
-Why These Topics?: Learning about LLMs, transformers, and generative AI will make you future-ready.
-Commitment: 5 hours weekly for 20 weeks.

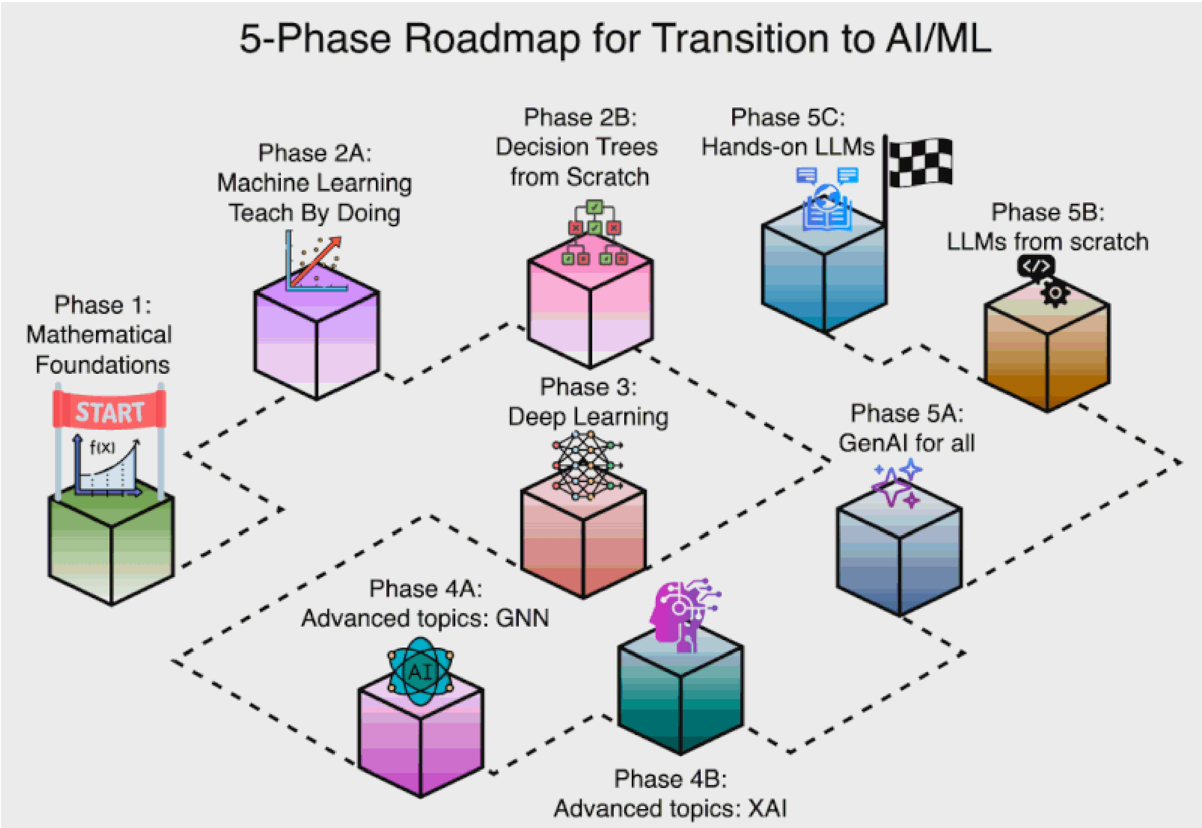
🔸 Optional [140 hours]
📌Introduction to Machine Learning in Julia [40 hours]: https://lnkd.in/g8A3DtQW
📌Zero to Hero in Data Science [40 hours]: https://lnkd.in/gNEgx2Cz
📌Hands-on PINN [20 hours]: https://lnkd.in/gta5hgHZ
📌ML in Hindi [40 hours]: https://lnkd.in/giD88GzZ

***
✅Total Duration: 275 hours + optional 140 hours
✅Timeline: 6-8 months, balancing learning with practical application.
✅Outcome: Build foundational ML knowledge, gain practical skills, and stay ahead with advanced topics.

***
If you are willing to spend time, this roadmap will help you get there.

Follow Vizuara’s YouTube channel for structured and beginner-friendly playlists: https://lnkd.in/g455AJVw

Your ML journey begins now—start building your expertise today.


5-Phase Roadmap for Transition to AI/ML
Phase 1: Mathematical Foundations
START
f(x)
Phase 2A: Machine Learning Teach By Doing
Phase 2B: Decision Trees from Scratch
Phase 3: Deep Learning
Phase 4A: Advanced topics: GNN
AI
Phase 4B: Advanced topics: XAI
Phase 5A: GenAI for all
Phase 5B: LLMs from scratch
Phase 5C: Hands-on LLMs

Where exactly is linear algebra, probability, statistics, calculus and optimization used in ML?

1. Linear Algebra
Where?
-Input, hidden & output layers.
-Forward propagation and weight updates.
How?
-Representing input data, weights, and biases as matrices.
-Efficiently handling batch inputs.

2. Probability
Where?
-Activation functions (sigmoid, softmax etc).
-Dropout.
How?
-Softmax: Converts raw logits into probabilities for classification tasks.
-Dropout: Randomly drops units with a probability to prevent overfitting.

3. Statistics
Where?
-Data preprocessing and normalization.
-Evaluation metrics (e.g., mean squared error, accuracy).
-Weight initialization and regularization.
How?
-Standardizing input features (mean = 0, variance = 1) to improve training stability.
-Analyzing performance using metrics like cross-entropy loss.

4. Calculus
Where?
-Backpropagation and gradient computation.
-Derivatives of activation functions.
How?
-Calculating gradients of loss with respect to weights and biases using the chain rule.
-Derivatives of activation functions (e.g., sigmoid, ReLU) are crucial for backpropagation.

5. Optimization
Where?
-Weight updates during training.
-Loss minimization.
How?
-Algorithms like Gradient Descent, Adam, RMSProp, etc., to minimize the loss function.

-Finding optimal weights and biases to improve model predictions.

In this age of capsule machine learning, only deep, foundational knowledge will help you stand out.

If you are serious about transitioning to ML, don't start with toy Kaggle projects like millions. Start by building a strong foundation.

On Vizuara's YouTube channel, I have released an entire course on "**Foundations for Machine Learning**".

Here are the lectures:
1) Course introduction: https://lnkd.in/gDdj2Wim

2) Linear Algebra: Vector transformation, span, and basis: https://lnkd.in/gBbrSnMd

3) Linear transformation as a matrix multiplication: https://lnkd.in/gpJTNauq

4) Product of two matrices as composite transformation: https://lnkd.in/gWb3RWsD

5) 3D linear transformation: https://lnkd.in/g94VzbeX

6) A simple physical intuition for determinants: https://lnkd.in/g_5JnYbD

7) Transformation with non-square matrices (2D to 3D): https://lnkd.in/gXS3Ggmg

8) Matrix inverses and their physical meaning in transformations: https://lnkd.in/g6pHWj-Q

9) Relation between dot product and transformations: https://lnkd.in/geEPwAbi

10) Simple intuition of eigenvalues and eigenvectors: https://lnkd.in/gg6dVPVM

11) Probability and statistics an introduction: https://lnkd.in/g-Ze6UY5

12) Introduction to conditional probability: https://lnkd.in/gJyr4yFs

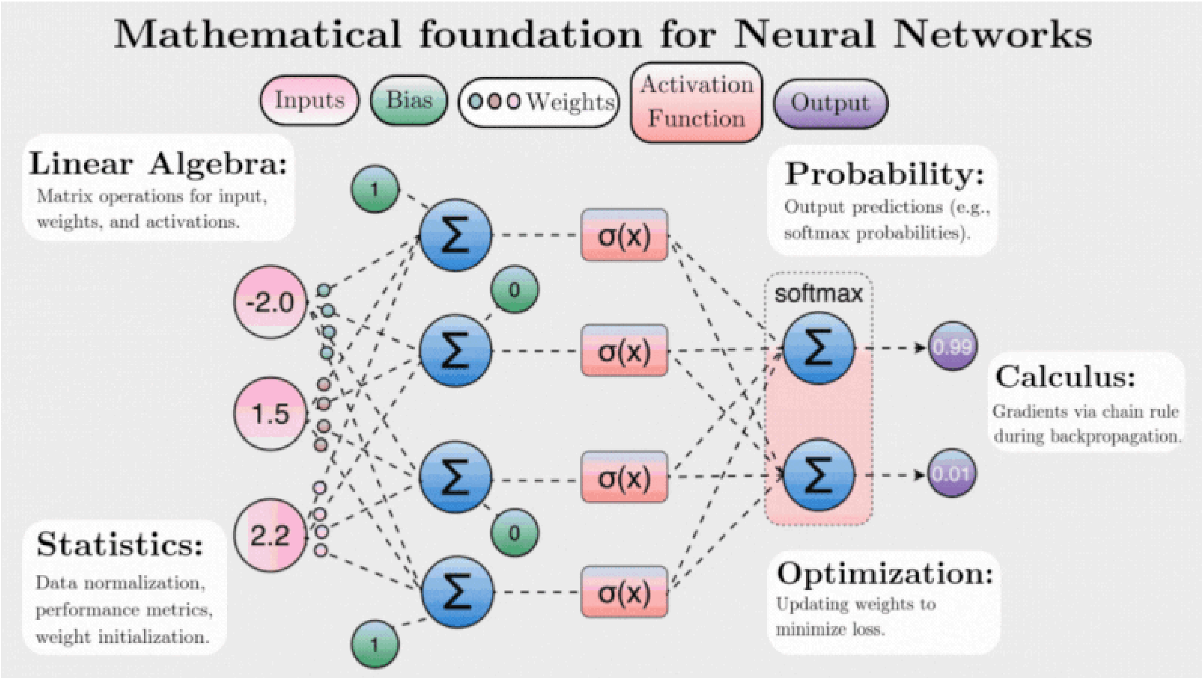
13) Intuition and basics of Bayes theorem: https://lnkd.in/gvt4vTQT

14) Probability distributions: https://lnkd.in/gpS5cuPu

15) Hypothesis testing: https://lnkd.in/gUUJj8QG

16) Introduction to Naive-Bayes classification: https://lnkd.in/gXk_qygU

Author: Dr. Sreedath Panat



Build strong foundations to transition to ML

Linear Algebra, Probability, Statistics, Calculus, and Optimization. These are the foundations behind building machine learning models.

In this age of capsule machine learning, **only deep**, **foundational knowledge will help you stand out**.

If you are serious about transitioning to ML, don't start with toy Kaggle projects like millions. Start by building a strong foundation.

On Vizuara's YouTube channel, I have released an entire course on "**Foundations for Machine Learning**" where we cover the above topics along with programming fundamentals. If you are a beginner, you can start with this. You will be already ahead of 99.9% of the people.

Here are the lectures I have published so far. More will be published on this playlist: https://lnkd.in/g2K97xHW

1) Course introduction: https://lnkd.in/gDdj2Wim

2) Linear Algebra: Vector transformation, span, and basis: https://lnkd.in/gBbrSnMd

3) Linear transformation as a matrix multiplication: https://lnkd.in/gpJTNauq

4) Product of two matrices as composite transformation: https://lnkd.in/gWb3RWsD

5) 3D linear transformation: https://lnkd.in/g94VzbeX

6) A simple physical intuition for determinants: https://lnkd.in/g_5JnYbD

7) Transformation with non-square matrices (2D to 3D): https://lnkd.in/gXS3Ggmg

8) Matrix inverses and their physical meaning in transformations: https://lnkd.in/g6pHWj-Q

9) Relation between dot product and transformations: https://lnkd.in/geEPwAbi

10) Simple intuition of eigenvalues and eigenvectors: https://lnkd.in/gg6dVPVM

11) Probability and statistics an introduction: https://lnkd.in/g-Ze6UY5

12) Introduction to conditional probability: https://lnkd.in/gJyr4yFs

13) Intuition and basics of Bayes theorem: https://lnkd.in/gvt4vTQT

14) Probability distributions: https://lnkd.in/gpS5cuPu

15) Hypothesis testing: https://lnkd.in/gUUJj8QG

Once you build a strong foundation, go ahead and start building ML models yourself.

These are 3 world-class playlists from MIT PhD Dr. Raj Abhijit Dandekar that you must follow.

1) ML teach by doing [37 lectures]: https://lnkd.in/gn2dEcE2

2) Building Neural Networks from scratch [34 lectures]: https://lnkd.in/gj8kHe2T

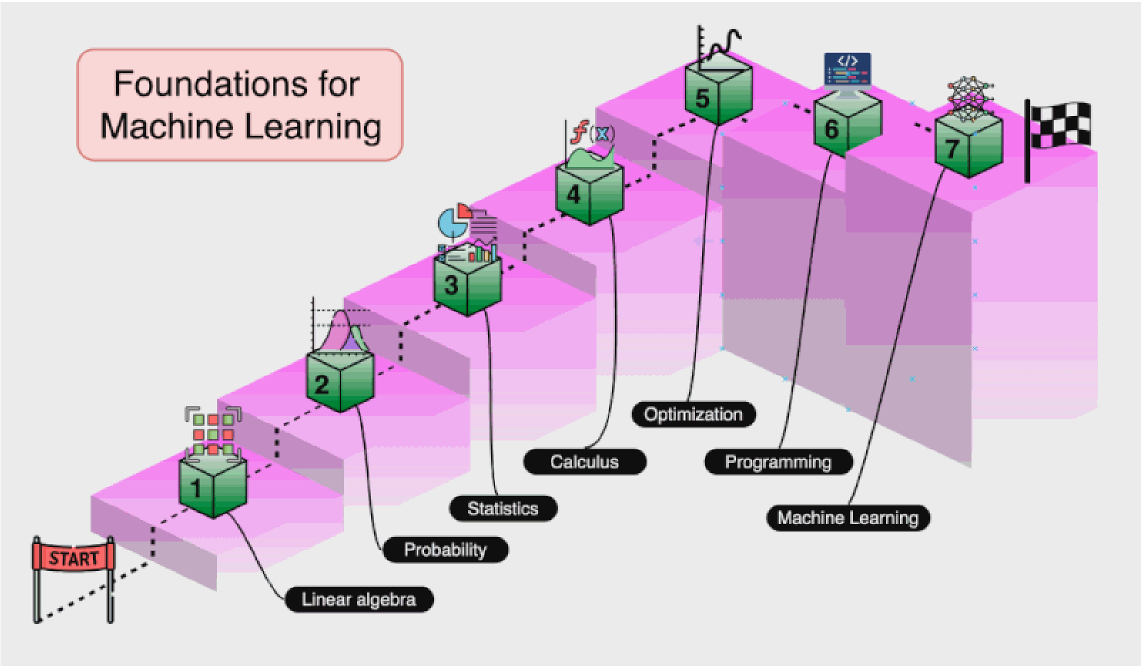
P.S: Want to learn about machine learning in depth without all the fluff?

Join our live bootcamp starting from January 2025: https://vizuara.ai/spit/

The instructors will be PhDs from top universities like MIT and ML industry professionals.

50+ students have already registered and we are closing registrations very soon!

Author: Dr. Sreedath Panat



An intro to ML concepts

Machine learning is one of the most fascinating and transformative fields of our time. If you have ever wondered how machines learn to identify patterns, make predictions, or cluster data without explicit programming, this lecture is for you.

As part of the "Introduction to Machine Learning in Julia" series, I am thrilled to share the latest lecture, "Introduction to ML Concepts", now available on Vizuara's YouTube channel.

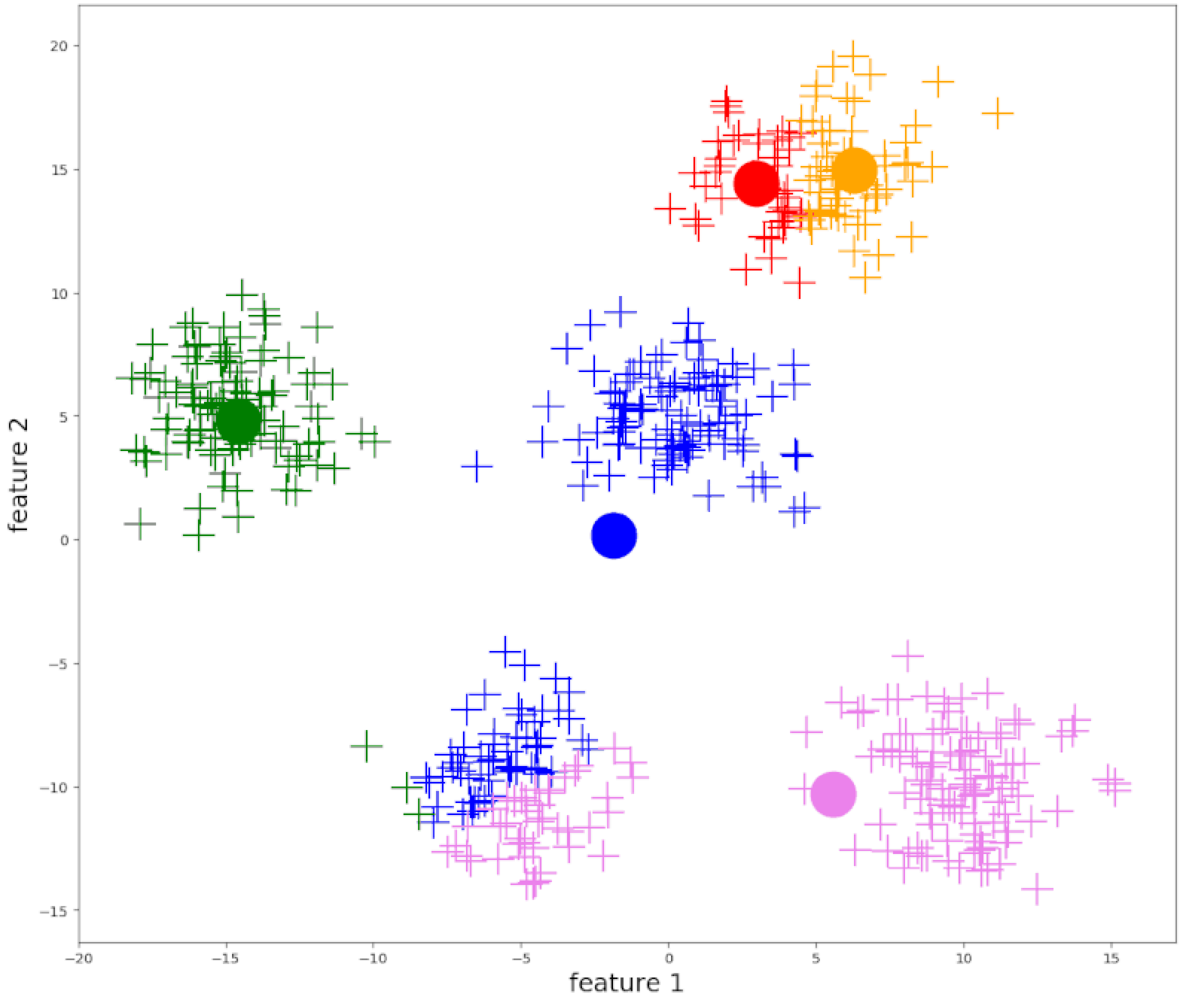
In this lecture, I dive deep into:

1) Supervised Learning: How models learn from labeled data to make predictions.
Example: Predicting house prices using features like size, location, and number of bedrooms.
2) Unsupervised Learning: How models identify patterns in unlabeled data.
Example: Clustering customer behavior or detecting anomalies in transactions.
3) Key Model Evaluation Metrics: Accuracy, precision, recall, F1 score, and how they help evaluate a model's performance.
4) Hands-on Implementation in Julia: A practical demonstration of linear regression (supervised learning) and k-means clustering (unsupervised learning), showing the power of ML concepts in action.

This lecture is carefully crafted to help beginners build a strong foundation in machine learning, with examples explained step-by-step. If you are curious about how ML concepts work under the hood or want to see them implemented in the Julia programming language, I encourage you to watch this session.

Link to the full lecture: https://lnkd.in/gfFByGgK

Author: Dr. Sreedath Panat



feature 1
feature 2
20
15
10
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
15

Deep Learning

Let us hand-craft a Generative Neural Network from scratch in 60 minutes!

Imagine you have a neural network (NN1) that initially outputs a random 32x32 noise image. This image is then fed into a second neural network (NN2), which predicts the probability (p) that the input image is of a cat.

Initially, since the image is just noise (and not a cat), p will be low, say 0.001. We define loss functions for both NN1 and NN2. If p is close to zero, NN1's loss is low, and if p is close to 1, NN2's loss is low. Essentially, if p is close to 1, NN1 has generated an image that looks very much like a cat to NN2.

To achieve this, we make NN1 and NN2 adversaries: NN1 tries to make p close to 1, while NN2 tries to make p close to 0. Training both networks against each other leads to NN1 becoming a good generator of realistic 'fake' cat images.

This process is the basic principle of Generative Adversarial Networks (GANs). NN1 is the Generator, and NN2 is the Discriminator. They are adversaries, like in deepfake technology where GANs can create videos of one person speaking like another.

The best way to learn GAN is to build one from scratch. I have created a 60-minute lecture that teaches you the basic principles of GAN and teaches you how to build a very, very simple GAN from scratch: https://lnkd.in/gFjWdHx6

We will hand-calculate the weights of the neural networks and see what kind of output we get.

The lecture will cover the following

1) What is GAN?
2) Defining a simple problem statement where an image is only 2x2 pixels
3) Defining the Generator that randomly creates a 2x2 image and discriminator that says whether this image is real or fake
4) Hand-calculate the updated weights for Generator and Discriminator

Try implementing this GAN by hand. I am sure you will enjoy it.

This lecture is part of the Explainable AI (XAI) series.

Author: Dr. Sreedath Panat

ADAM vs ADOPT

Why read machine learning research papers?

Because you can discover a gem like this:

"ADOPT: Modified Adam Can Converge with Any β2 with the Optimal Rate"

This paper was accepted at NeurIPS 2024 and has been published on ArXiV here:
https://lnkd.in/gQxFrCed

What's so great about this paper?

It shows that a new optimizer called ADOPT converges better than the ADAM optimizer in both theory and practice.

The authors show that ADOPT achieves better results compared to ADAM and its variants across a wide range of tasks, including image classification, generative modeling, natural language processing, and deep reinforcement learning.

You can start using ADOPT today by just replacing one line in your Python code, as shown in the image.

Author: Dr. Raj Dandekar

```python
from adopt import ADOPT


# optimizer = Adam(model.parameters(), lr=1e-3)

optimizer = ADOPT(model.parameters(), lr=1e-3)
```

The essence of deep learning

The core of deep learning is optimization.

The core of optimization is gradient descent.

The core of gradient descent is:

- Writing the loss function as matrices
- Taking derivatives of those matrices with respect to parameters.

If you want to really understand machine learning, linear algebra and vector calculus are the most important tools.

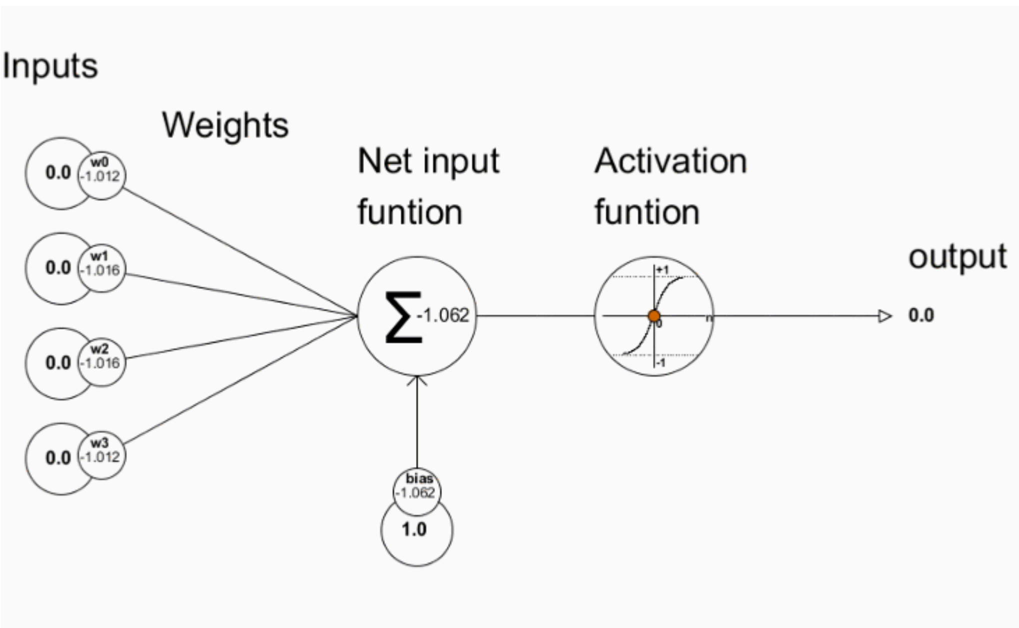
That linear algebra lecture where you were taught to express summations in terms of matrices will help you master the closed form ordinary least squares solution.

That vector calculus lecture where you learnt how to take derivatives of one matrix with respect to another matrix, will help you master backpropagation and neural networks.

Knowledge of matrices and vector calculus will also help you understand tensors and master PyTorch, an essential machine learning library of today.

Anyone can load Pytorch packages. But to debug, you need to know the details. You need to know how the dimensions work and how the tensors multiply with each other.

Author: Dr. Raj Dandekar



Physics Informed Neural Networks

Physics Informed Neural Networks (PINN) is that branch of Scientific Machine Learning which has led to a lot of innovative research by mixing physics with neural networks.

It has transformed many fields of science and engineering from battery modeling to epidemiology to fluid mechanics!

Why has PINN become so popular?

Because it mixes the power of neural networks with the knowledge of science.

Traditional neural networks are a black box. PINNs is like shining a torch on this black box.

It's an amazing technique which can help you merge your domain knowledge with machine learning.

With PINN, you can do the following projects:

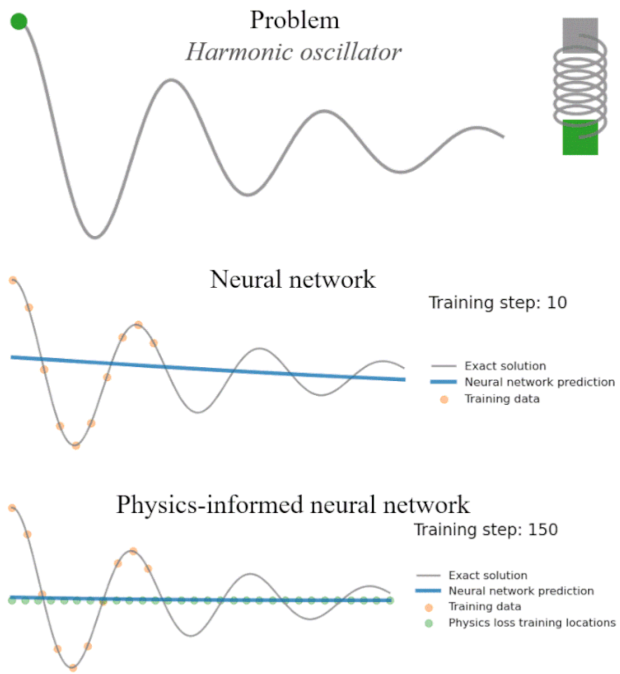
(1) Integrate neural networks with computational fluid mechanics

(2) Use machine learning to model black hole physics

(3) Use machine learning to model chemical reaction systems

(4) Integrate machine learning and finance

Author: Dr. Raj Dandekar



Importance of foundational knowledge

Blindly using powerful Python packages like Tensorflow, PyTorch and Scikit-learn really leads to a poor understanding of core ML concepts.
(1) Tensorflow:

```
model.fit(x_train, y_train, epochs=5)
```

In one line, you can fit a neural network between the X and y data.
What about understanding of key concepts like backpropagation?
What about understanding dropout, batch normalisation and regularisers to prevent overfitting?
What about understanding the difference between RMSProp and ADAM?

(2) Scikit learn

```
clf_dt = DecisionTreeClassifier()
clf_dt = clf_dt.fit(X_train, y_train)
```

In 2 lines, you can run a decision tree on the heart disease dataset.
What about understanding Gini impurity vs Entropy?
What about understanding pruning?
What about the effect of having very few samples in a leaf - is it good or not?

Python libraries have certainly made our lives easier.
As you use these libraries or even before you use these libraries, try understanding everything from scratch.

At least once, write down the equations for the Ordinary Least Squares (OLS) solution.

At least once, write down the neural network forward pass on a notebook.

At least once, prune a decision tree yourself on a toy project.

If you spend time on understanding these fundamentals, you will become a much stronger machine learning engineers.

Author: Dr. Raj Dandekar

Large Language Models

Transformer oversimplified for all

Poloclub has released a beautiful transformer visualization. They use GPT-2 to explain how transformers (the heart of LLMs) work.

This is one of the best visualizations of transformers I have come across among a few others. Please check out: https://lnkd.in/guWi_af7

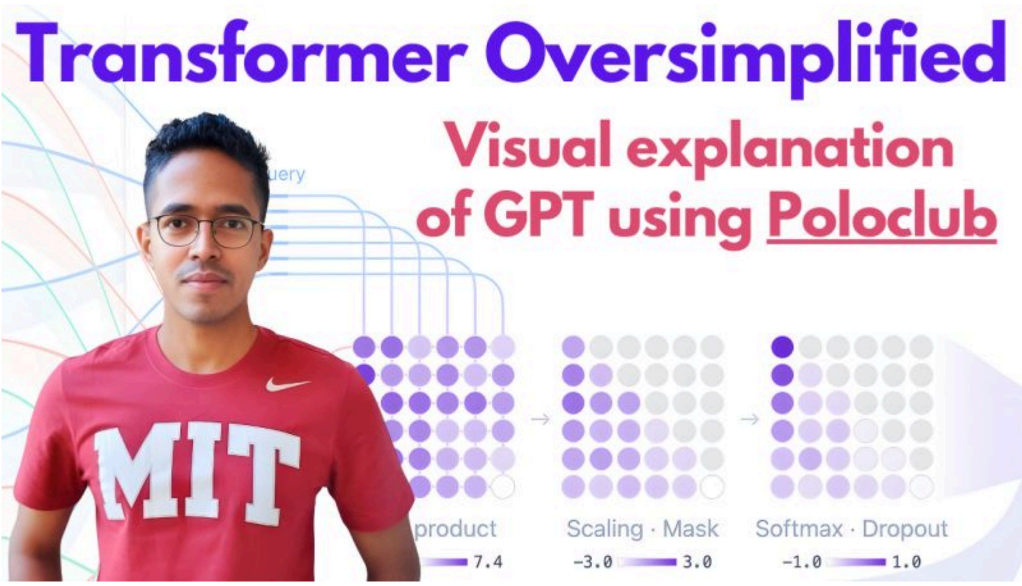
This visualization inspired me to create a lecture on how transformers work. I have released the lecture on Vizuara's YouTube channel. Here is the link to the lecture: https://lnkd.in/gFWE7XbJ

I title this lecture "Transformer Oversimplified" because I have eliminated the difficult aspects of transformers and made them as beginner-friendly as possible to the extent of oversimplification.

Check out this lecture. I am sure you will enjoy it and understand a lot about transformers, LLMs, and GPT: https://lnkd.in/gFWE7XbJ

If you are interested in learning about LLMs, we are releasing a lecture series on LLMs from scratch soon on our channel. The first lecture will be out in 2 days. Stay tuned.

Author: Dr. Sreedath Panat



Tokenization is a great opportunity for LLM Research

If you are new to the field of Large Language Models (LLMs) and want to start LLM research, tokenization might be one of the best fields to do so.

Tokenization, Pre-training and Inference are all important aspects of LLMs.

However, the amount of research done in pre-training and inference is much higher than tokenization.

Tokenization remains the most neglected aspects of LLM research, compared to the other aspects.

This presents an incredible opportunity to do novel research in this area.

Author: Dr. Raj Dandekar

Byte Pair Encoding

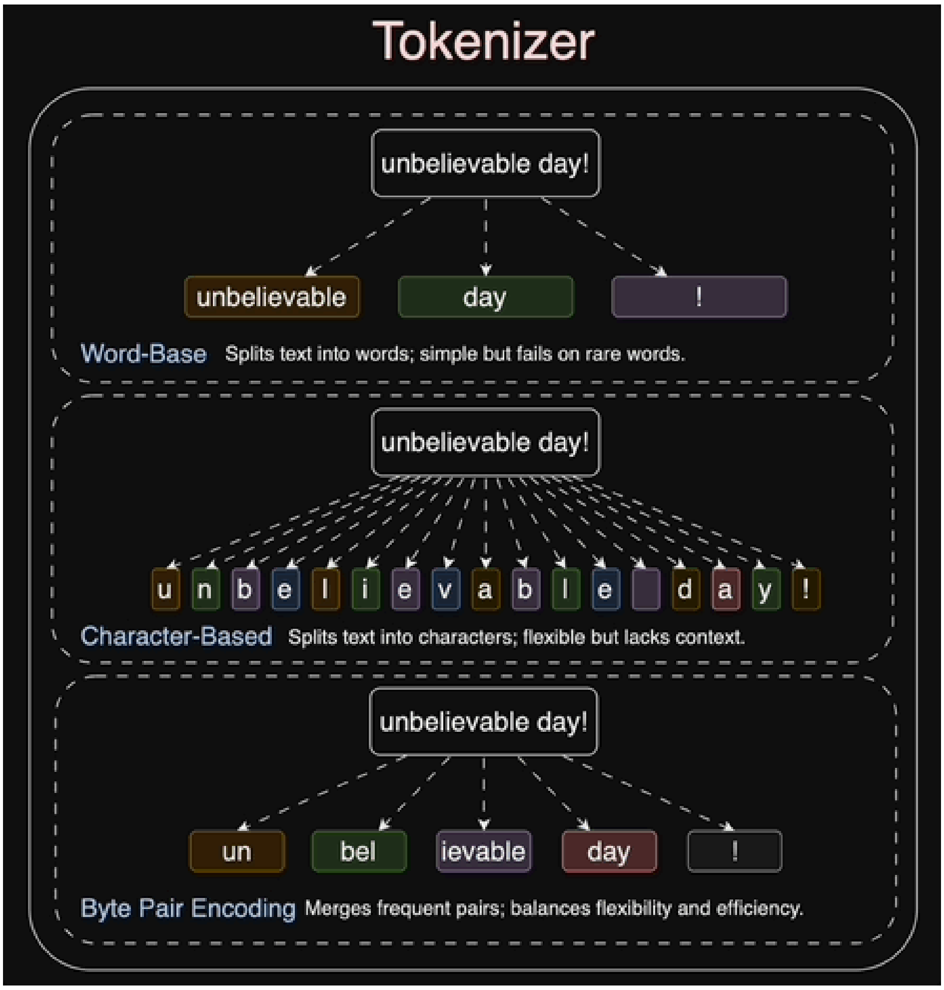
1 Character level tokenization is not very efficient.

2 Byte Pair Encoder (BPE) uses subwords as tokens instead of characters.

3 Instead of treating every character or word as an individual unit, BPE operates on subword units, which helps solves 2 things:

- It prevents the number of tokens needed to represent the text from increasing too much, which is the main issue with character tokenization.
- We retain linguistic patterns: such as common prefixes, suffixes, and word stems.

4 BPE iteratively merges the most frequent pair of characters or subwords in a corpus until a predefined vocabulary size is reached.



Author: Dr. Raj Dandekar

ModernBERT

ModernBERT : A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Fine-tuning and Inference🪩🆕

Remember BERT👴? The original representation model we loved and adored for almost all text-level tasks (be it for classification or inherently as part of any retrieval system). BERT is an encoder-only architecture, it is fast, efficient and convenient to use, but, with advent of much larger decoder only architecture trained on more data with all the advancements both in terms of compute and quality, BERT slowly took a back seat, but, here it is again.

The authors Benjamin Warner, Antoine Chaffin, Benjamin Clavié and others present modernBERT, which is a drop in replacement of previous BERT architecture with all the latest tech/advancements we love in decoder-only architectures.

Specifically for retrieval systems this is huge, because Decoder architecture are slow and being in AR setup can't attend to full sequence of text, basically for any token at T it only attends to (T-k-1, T-1);k>=0 tokens, hence it's uni-directional/restrictive. Whereas bidirectional architecture (like BeRT) not only attends to the entire sequence (more info aware, hence representative) but is also much efficient.

Methodology🧐

1. Architectural improvements
 - Now it uses RoPE instead of older fixed positional embeddings.
 - for representation stabilization an additional normalization layer is applied.
 - Older MLP layers are removed by GeGLU layers(Gated Linear Units with Gelu attention), read about it here : https://lnkd.in/gXnvRQAD

2. Computational/efficiency inducing changes
 - Customized for flash-attention-2 and hence uses it.
 - Alternating attention : Instead of applying attention over entire sequence length for each each layer, they follow the recent trend of alternating attention (Global/full attention every three layers, sliding-window/local attention with 128 tokens window every other layer)
 - Unpadding and sequence padding : This is something new actually, we all know that for any predictions encoder only models have to pad the current sequence to a fixed length, but, this waste time and compute (both during training and Inference), to avoid that, authors followed a simple method
 - Remove all padding from each sequence and concatenate them together to form one long sequence.
 - The sequence is divided into mini-batches of size 1.
 - The token are the stitched in sequence by using attention masks.
 - Attention is performed for each sequence by using attention masks.
 - the method works when better with flash-attention-2.

Also, ModernBERT is trained on much larger data (2 Trillion tokens); much larger than other encoder models.

	Model	Params	Short BS	Short Fixed	Short Variable	Long BS	Long Fixed	Long Variable
Base	BERT	110M	1096	**180.4**	90.2	–	–	–
	RoBERTa	125M	664	179.9	89.9	–	–	–
	DeBERTaV3	183M	236	70.2	35.1	–	–	–
	NomicBERT	137M	588	117.1	58.5	36	46.1	23.1
	GTE-en-MLM	137M	640	123.7	61.8	38	46.8	23.4
	GTE-en-MLM$_{xformers}$	137M	640	122.5	128.6	38	47.5	67.3
	ModernBERT	149M	**1604**	148.1	**147.3**	**98**	**123.7**	**133.8**
Large	BERT	330M	**792**	**54.4**	27.2	–	–	–
	RoBERTa	355M	460	42.0	21.0	–	–	–
	DeBERTaV3	434M	134	24.6	12.3	–	–	–
	GTE-en-MLM	435M	472	38.7	19.3	28	16.2	8.1
	GTE-en-MLM$_{xformers}$	435M	472	38.5	40.4	28	16.5	22.8
	ModernBERT	395M	770	52.3	**52.9**	**48**	**46.8**	**49.8**

Table 2: Memory (max batch size, *BS*) and Inference (in thousands of tokens per second) efficiency results on an NVIDIA RTX 4090, averaged over 10 runs. Dashes indicate unsupported configurations.

Author: Siddhant Rai

COCONUT - A new way to train LLMs

COCONUT 🥥: Training Large Language Models to Reason in a Continuous Latent Space🌐

Hallucinations and non-grounded answers in reasoning tasks had been one of the central pain point in research behind LLM as reasoning mechanisms.💭

Methods like CoT (Chain of Thought) tend to solve this issue by specifically grounding the outputs by making a requirement of sequential step-by-step generation of steps. Abstractly, it serves like a guidance/conditioning over a graph (discrete in observation) or a manifold (very similar to how we do text-based conditioning in diffusion models). The purpose it serves is essentially transitioning/moving the input vector space close to accurate output space. But, the way it is done in itself is dependent/function over Language model itself. Hence, 2 major issues pop-up

1. Compute wastage : most of the generated tokens are wasteful.
2. As the language is a discrete representation of our thought, which could be linked to LLMs as tokens and embedding space respectively, with every sampling operation we do, we are inherently loosing info/making representation sparse (which is again wasteful),

This is handled by recent work from Shibo Hao, Sainbayar Sukhbaatar and others at Meta and UC San Diego. Authors introduced COCONUT or simply continuous chain of thought.

Methodology 🧐

0. Applied on Pre-trained model.
1. Contrary to CoT which uses teacher forcing for next token prediction (discrete) even in reasoning step, the authors replaced it with simpler yet continuous conditioning; here for step t, model instead of using reasoning token from t-1, uses embedding from previous step along with question embeddings.
2. For any task, for first c steps, model performs continuous operation by
 - Attaching a <bot> (beginning of tokens) for model know about start of reasoning
 - Then it is attached with [thoughts] tokens which serve as markers and then enclosed by <eot> (end of token)
3. Given above setup, during training, where data contains reasoning steps and solution/response. Each reasoning step is iteratively replaced by continuous [thoughts] token. The loss is then computed over the remaining tokens.
4. This basically serves as information distillation from reasoning tokens to continuous thought tokens as the process is end-to-end differentiable
5. For switching from (thoughts mode) to (language mode) during inference, they employ two methods
 - simple binary classifier over latent thoughts
 - fixed length of [thoughts] tokens (preferred)

Outcome🎯

1. Sweet spot between number of tokens and accuracy.
2. Significantly outperforms non-CoT methods, but, almost on par with CoT (Coconut is more efficient as the number of tokens are 1/3rd)
3. Able to complex reasoning operations as the conditioning steps behaves as combo of BFS+DFS (sort of K-hops) over nodes.

Author: Siddhant Rai

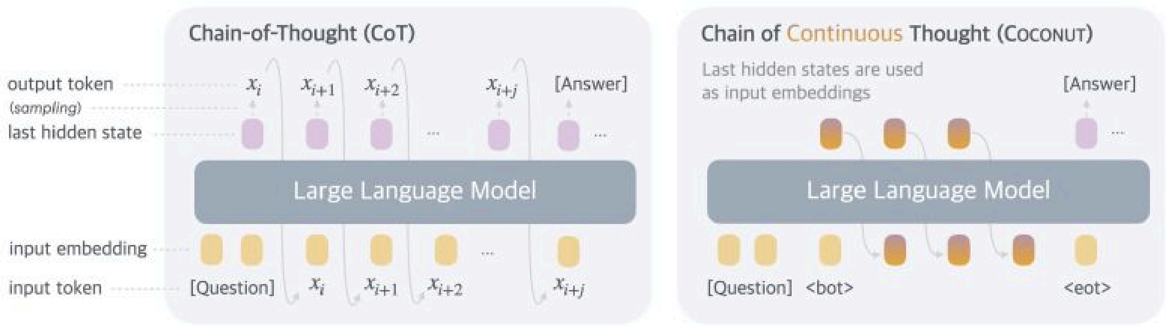


Figure 1 A comparison of Chain of Continuous Thought (Coconut) with Chain-of-Thought (CoT). In CoT, the model generates the reasoning process as a word token sequence (e.g., $[x_i, x_{i+1}, ..., x_{i+j}]$ in the figure). Coconut regards the last hidden state as a representation of the reasoning state (termed "continuous thought"), and directly uses it as the next input embedding. This allows the LLM to reason in an unrestricted latent space instead of a language space.

Why character level tokenization fails?

Tokenization converts text into numerical data for LLMs.

Using the Unicode standard, each character is mapped to an integer, so a sentence can be broken down into individual characters which become the tokens.

For example, "Today" is represented by the token IDs 84, 111, 100, 97, 121.

However, there are 3 major reasons why character level tokenization does not work:

1️⃣Ballooning effect: Every character, including spaces and punctuation, becomes a token.

This creates a huge number of tokens, even for short texts.

For example, "Today, I want to start my day with a cup of coffee" becomes 50 tokens.

This increases computational costs and reduces how much text a model can process.

2️⃣Context window limitations: An LLM's context window, or the amount of text it can process at once, is quickly filled by character-level tokenization.

This can cut off the model’s view of the text and impact its ability to understand the meaning.

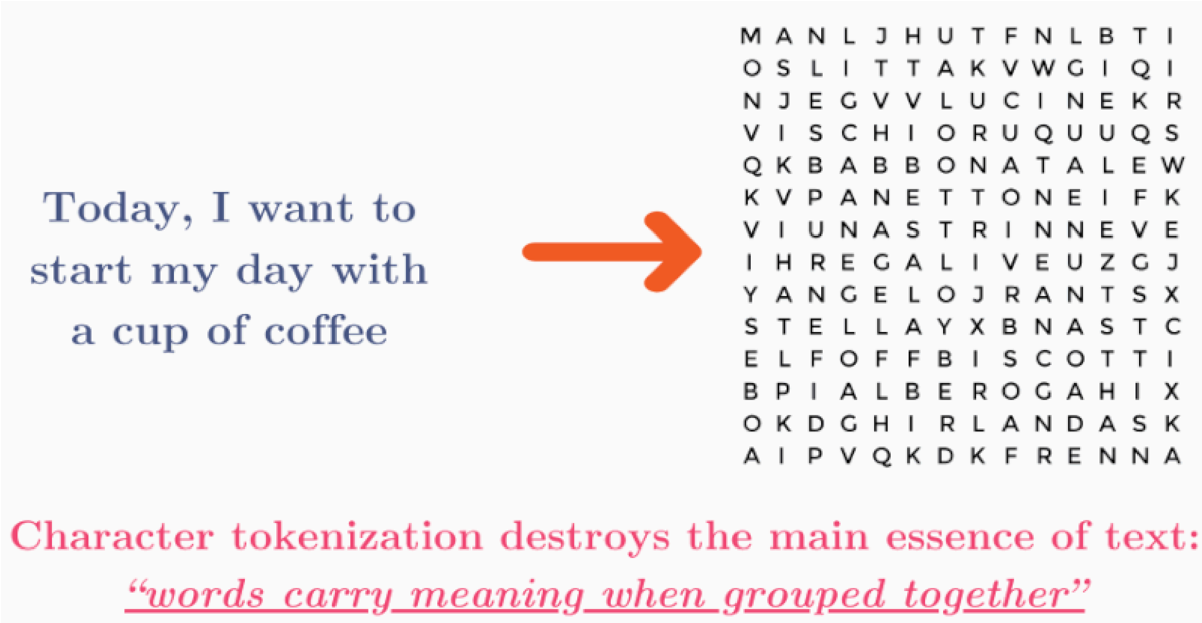
For example, if a context window is 32 tokens, the model can only see the beginning of the sentence: "Today, I want to start".

3️⃣Loss of Meaningful Units: Human language uses words, parts of words, and common phrases.

Unicode treats each character as a meaningless token, destroying the meaning that words carry.

Models learn more effectively with units that carry linguistic meaning.

That's why we have to turn to sub-word based tokenization algorithms like "Byte Pair Encoding".



Author: Dr. Raj Dandekar

The necessary (and neglected) evil of Large Language Models: Tokenisation

1️⃣Why we should pay more attention to tokenization? Because it's more important than you might think.

2️⃣My earlier understanding of LLMs was that we have a huge corpus of data on which we pre-train gigantic neural network architectures for the next token prediction task.

3️⃣Just as important as pre-training is one more piece of the puzzle: tokenization.

4️⃣First the tokenizer is trained on a dataset.

5️⃣Every single sentence, words and characters are broken down into tokens and token IDs.

6️⃣The token IDs are then passed to the Large Language Model (LLM).

7️⃣Then the Large Language Model is trained on a dataset which might be completely different from the dataset on which the tokenizer is trained.

After the tokenization step, **the LLM only sees the tokens**. **It never deals directly with text**.

8️⃣When we pre-train LLMs, the LLM is solely relying on the tokenized text.

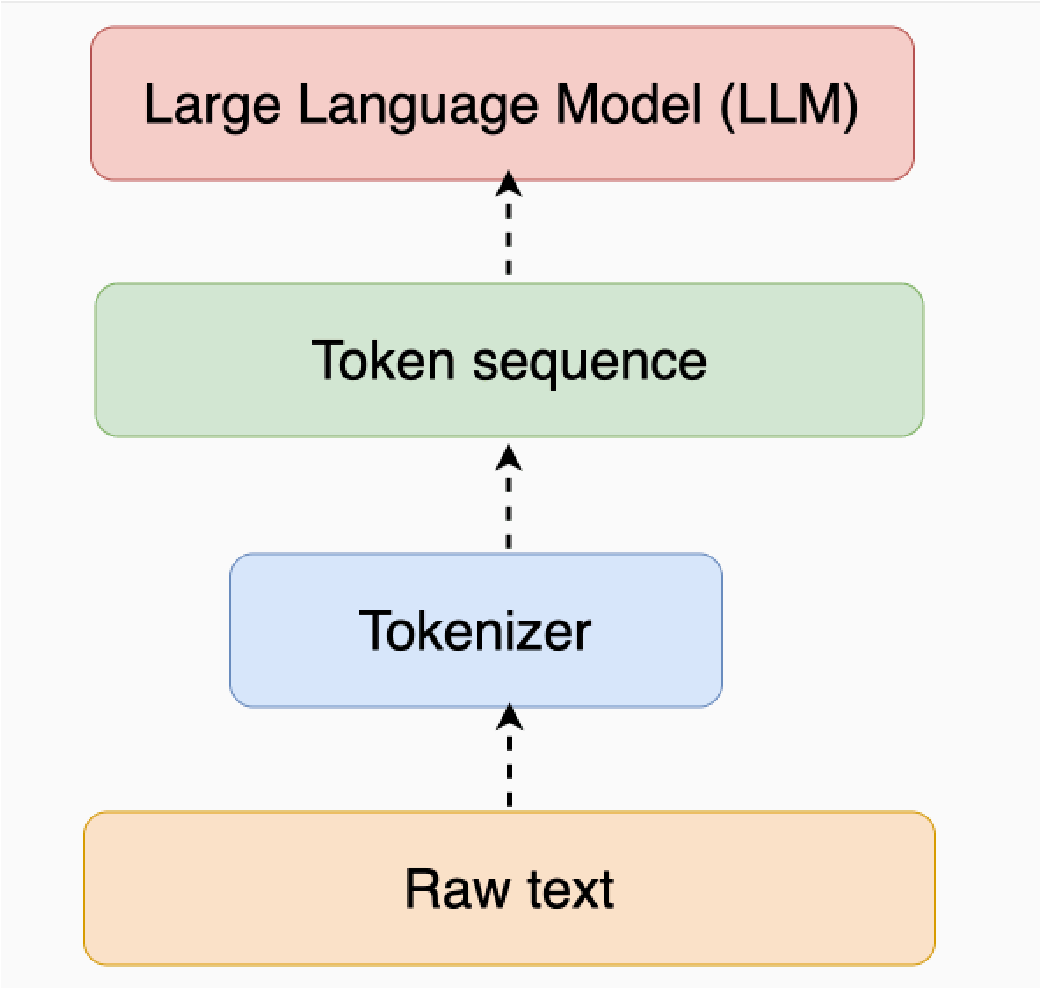
9️⃣If we mess up the tokenization, no matter how hard we try to pre-train, our LLM will always lead to a poor performance.

🔟 Here's a good way to think about tokenization and LLMs:

Next time you think of LLMs, think about them as a relay race team of 2 athletes.

The first athlete is tokenization and the second athlete is LLM pre-training.

No matter how good the second athlete (the LLM pre-training) is, we won’t win the race unless the first athlete (the tokenizer) is equally good.



Author: Dr. Raj Dandekar

Sebastian Raschka's "Build a LLM from scratch" book

Building LLM from scratch" by Sebastian Raschka, PhD.

4 months ago, I stumbled across this book during my search.

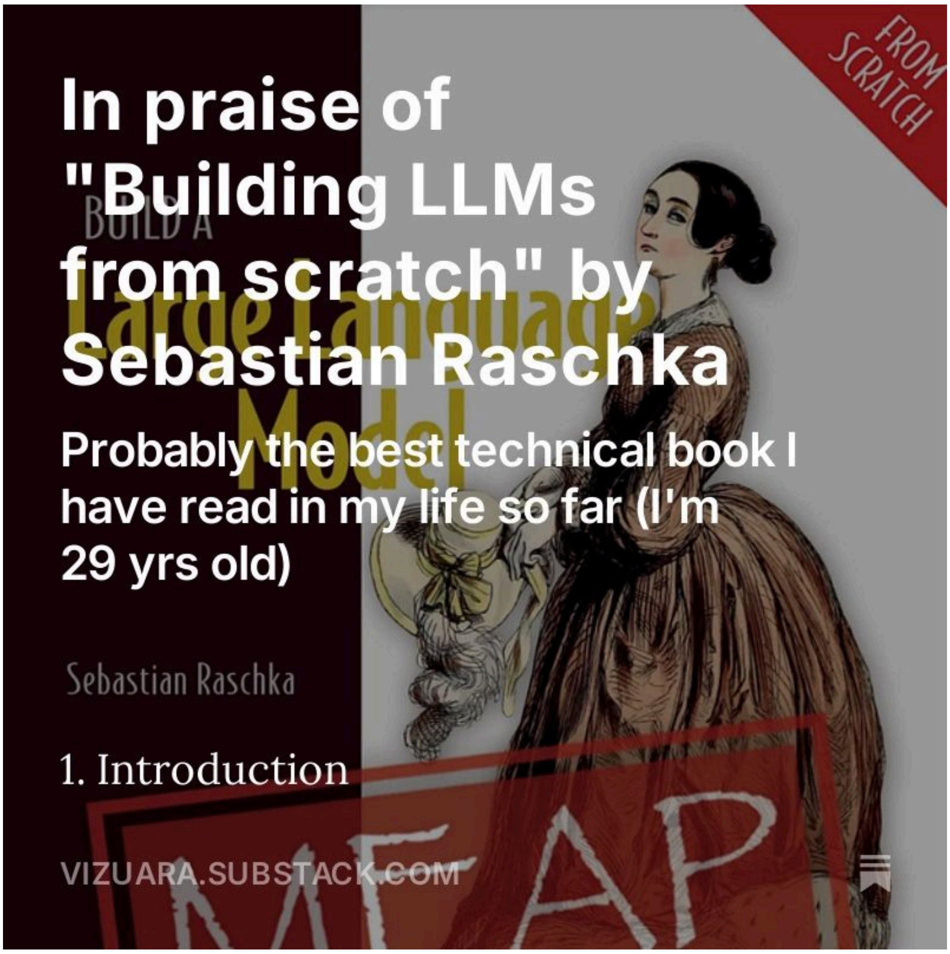
Here is what happened in the next 4 months:

1️⃣I finished reading this book completely

2️⃣I implemented GPT-2 from scratch in Python

3️⃣I learnt about Large Language Models probably better than any other machine learning concept I have understood till now.

4️⃣I discovered a whole new way of learning new things: teaching. I realized that if I made notes and understood a concept, the best way to truly learn it is to teach it.



Author: Dr. Raj Dandekar

Quantization in Large Language Models (LLMs)

Large Language Models (LLMs) consist of billions of weights.

It is essential that we store the information contained in these weights using as little memory as possible.

This is where quantization comes into the picture.

Quantization compresses the model by reducing the precision of numerical data. The most common quantization formats include:

1️⃣FP16 (16-bit floating point): Offers a good balance of precision and compression.

2️⃣INT8 (8-bit integer): Reduces memory further, with minimal performance degradation.

3️⃣INT4 (4-bit integer): Extreme compression, used in specific cases where ultra-low memory usage is critical.

What does quantization look like in action?

1️⃣Take the attached figure as an example. The left side shows an original image, and the right side shows a quantized image with only 8 colors.

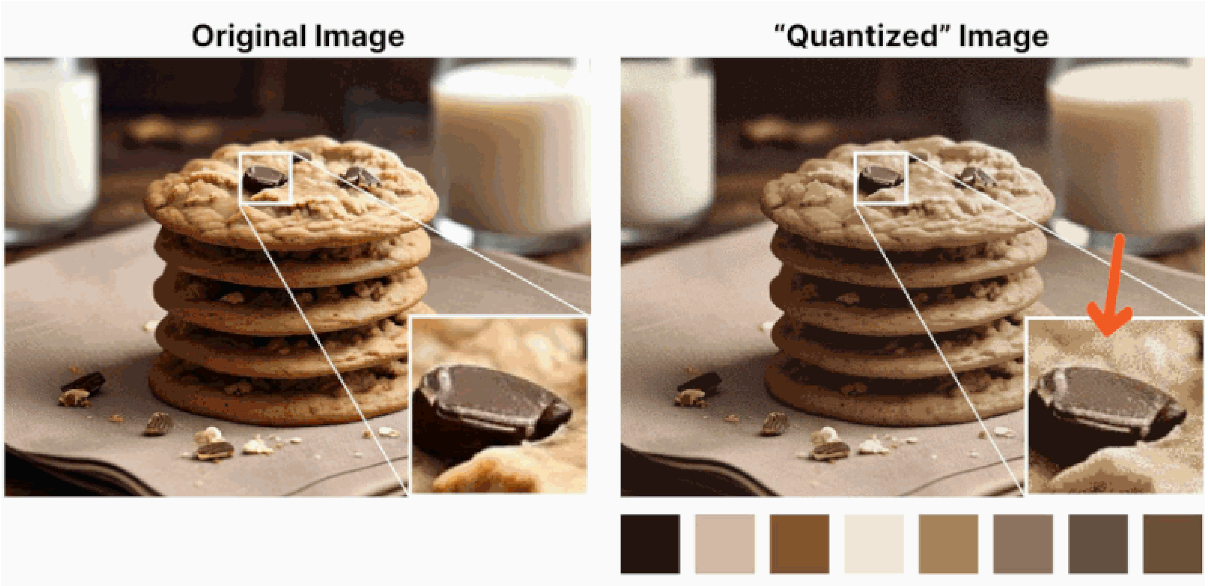
2️⃣Notice how the zoomed-in part seems more “grainy” than the original since we can use fewer colors to represent it.

3️⃣The main goal of quantization is to reduce the number of bits (colors) needed to represent the original parameters while preserving the precision of the original parameters as best as possible.

4️⃣The quantized image uses fewer colors (visible in the palette below), sacrificing some detail to save memory.

In LLMs, this principle is applied to billions of parameters.

A 32-bit floating-point model may shrink by 4x or more when converted to INT8, making it far more memory-efficient and easier to deploy on standard hardware.



Author: Dr. Raj Dandekar

LLM Guardrails

Today I came across a fantastic LLM Github repository called Guardrails AI.

- If you are building an LLM product or project, how do you make sure that the LLM output does not hallucinate?

- How do you make sure that the LLM output does not contain abusive language or competitor information or reveal personal identification?

- Guardrails is an awesome Python framework that helps build reliable AI applications by performing the following key function:

1 Guardrails runs Input/Output Guards in your application that detect, quantify and mitigate the presence of specific types of risks.

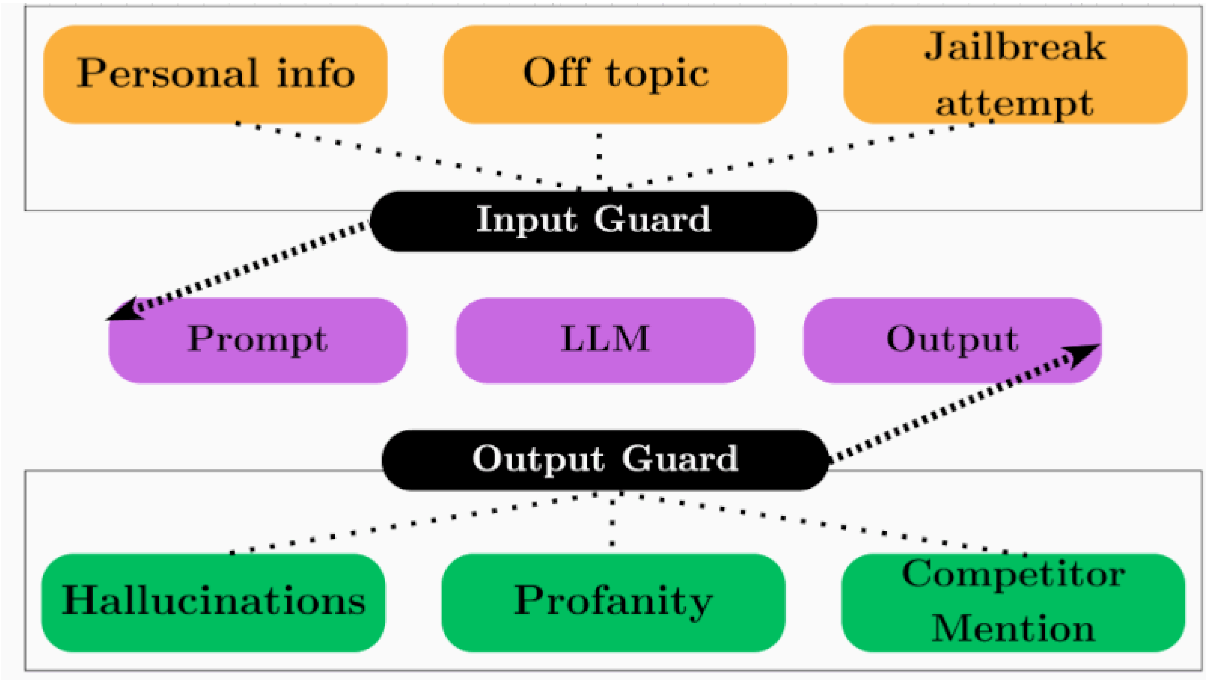
The Guardrails AI Github repository has a wide range of tools ranging from:

(1) Preventing the LLM from revealing personal information or competitor names in the responses
(2) Checks for profanity in text
(3) Ensure content has an URL
(4) Ensures valid CSV, JSON format
(5) Ensures that generated text is less than a maximum expected reading time
(6) Ensures generated output is polite
(7) Detects hallucinated text
…and much more.

Start today using the following commands:

(1) pip install guardrails-ai

(2) guardrails configure



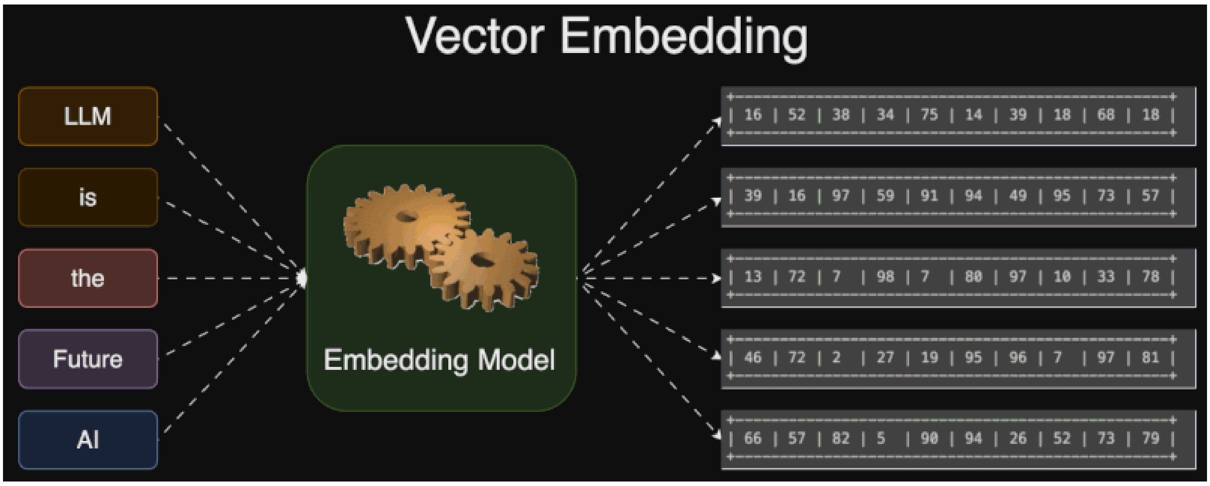
Author: Dr. Raj Dandekar

The power of embeddings in Large Language Models (LLMs)

Ever wondered how machines learn the meaning of words using just numbers? It's a fascinating journey of transforming text into numerical representations (embeddings), extracting meaning from these numbers, and converting them back into text.

Embeddings lie at the heart of *large language models* (*LLMs*). Through deep neural networks, random vectors learn contextually relevant weights, enabling machines to grasp not just the syntax but the semantic essence of language.

Here's how it works:
1️⃣ **Tokenization**: Breaking text into smaller units like words or subwords.
2️⃣ **Mapping to Vectors**: Assigning each token a numerical representation.
3️⃣ **Training**: Neural networks adjust these vectors based on the word's context in vast datasets.
4️⃣ **Positional Embeddings**: Adding order information to capture grammar and context.



Author: Dr. Raj Dandekar

LLM Data pre-processing pipeline

Before we even come to the modeling stage of LLMs, several steps need to be applied to the dataset.

Here are the steps involved:

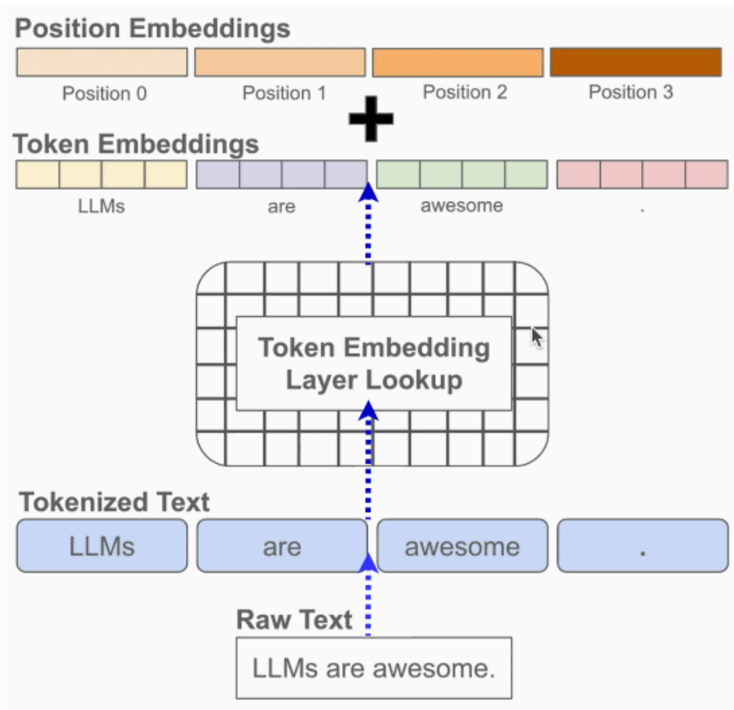
1️⃣Tokenization: Breaking text into smaller units (word-based, subword-based using BPE, or character-based) that the model can process.

2️⃣Token Embeddings: Converting tokens into numerical vectors that capture their meaning and relationships.

3️⃣Positional Embeddings: Encoding the position of each token to preserve the sequence order in the input.

4️⃣Input Embeddings: Combining token embeddings and positional embeddings to create the final representation for the model to process.

5️⃣Dataloader & Input-Target Pairs: Preparing batches of input data and corresponding target labels to efficiently feed the model during training.



Position Embeddings With In a Language Model

Author: Dr. Raj Dandekar

Generative AI

MindSearch: Mimicking human mind for deeper AI search

Google's search engine can give you the best search results in the form of web page links. But not direct answers to your queries. If you combine the search engine with a powerful modern LLM like GPT4o, you can get very accurate responses to your complex queries.

However, even the Search Engine + LLM combo has limitations.

1) The web search results include too much information and noise. This can easily exceed the context length of LLM.
2) Search Engine cannot efficiently do the search if the query is complex

The process of finding information is very different in humans. We do not try to answer our complex question directly. We modularize the question into smaller, manageable chunks.

Imagine you want to buy a car. You will not be directly trying to answer your question of "I wish to buy a car under Rs. 12 lakhs budget for daily use. I want XYZ features, mileage, and CC".

Rather you look at different parts of this question. Your brand preference, feature list, resale value, number of seats, expected years/km of total use, maximum budget, etc. Then you find a list of cars. Some of them will fit some of your requirements. You will weigh your options and then take a call. It is a fairly iterative process.

How do you incorporate this complex, iterative decision-making into AI-assisted internet search?

This paper titled "MindSearch", published on arXiv 3 weeks ago has a fascinating idea. Modularize the search query and represent this as a Graph Neural Network with nodes and edges. Deploy LLM agents for planning the web search via Search Engine API. Conduct parallel search queries to answer different parts of the query. This is called fine-grain search.

The Graph will be dynamic. Based on the findings from the initial web search by the LLM agent, the nodes and edges are updated. The number of nodes in the Graph will depend on the complexity of the search query.

I have published a 30-minute lecture video on Vizuara's YouTube channel reviewing this paper so that audiences from all kinds of backgrounds can understand this paper. Check out the video: https://lnkd.in/gqgd4QnT

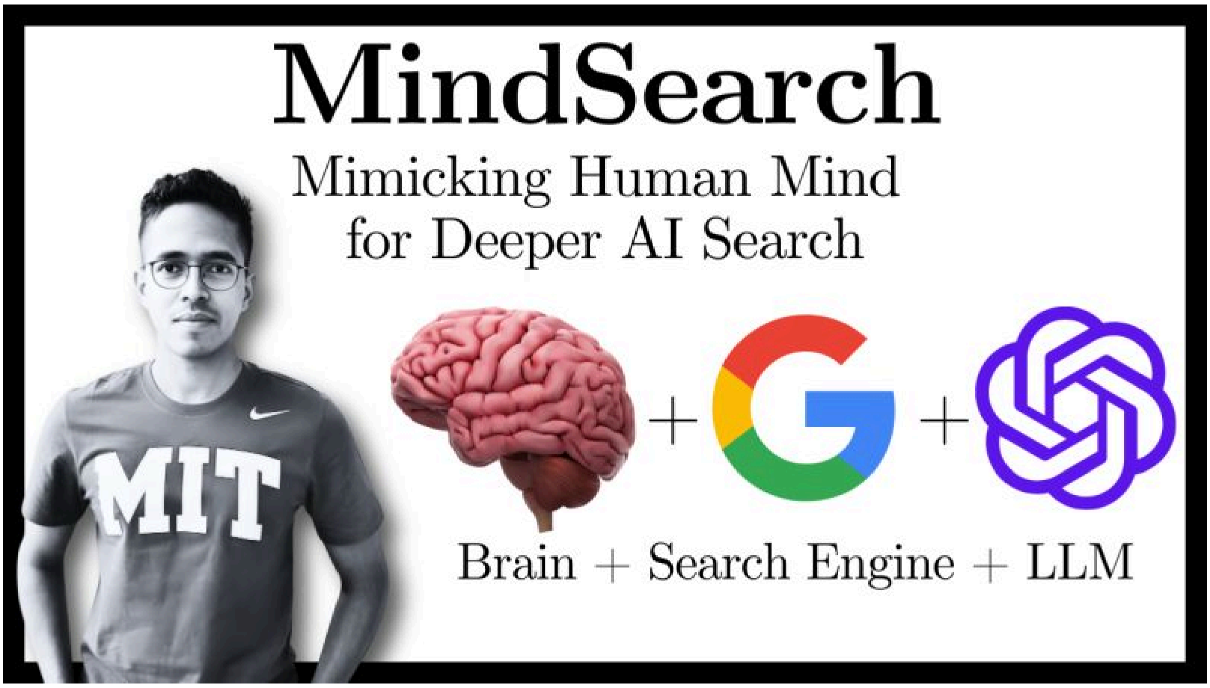
Based on the research conducted by the authors of the paper, MindSearch hands-down beats the existing solutions for AI-assisted search on open-ended and close-ended questions. Their GitHub repo already has 3.7k stars: https://lnkd.in/gkiT5XDC

Here is the link to the paper: https://lnkd.in/guaKueBm

For LLM search to replace traditional Google search there is a long way to go. The commercial use of MindSearch will boil down to cost, accuracy, rate of hallucination, speed, and the actual utility of the responses from the LLM agent.

Anyway, I think it is too early to critically review this paper. I just simply love the way they have written it. Impactful research, easy to understand, and completely open access. This is exactly how science should be. Hats off to the authors.

Author: Dr. Sreedath Panat



Degenerative AI: Generative AI model collapse!

The internet is getting flooded with GPT-generated data.

What if, decades later, 90% of internet data is AI-generated? Will the future AI models trained on the future internet data produce good results?

A paper was very recently published in Nature titled "AI models collapse when trained on recursively generated data". The paper explores how the distribution Gaussian distribution of data generated by LLMs trained on synthetic data tends to have less diversity or deviation from the mean.

A simple example can be imagined from image generation models. The photos of people on the internet will have a disproportionate number of smiling faces than what you see in real life. This is because when photos are being taken, people are generally in a happy mood.

A gen AI model trained on this data may have an inherent bias towards making smiling faces. Now imagine that the internet is flooded with a lot of gen AI images of people. The data will become a bit more biased toward smiling faces. If this repeats many times, the question we can ask is will future image generation models be unable to make a crying baby's face?

There are 4 interesting terminologies associated with this same idea.

1) AI cannibalism: AI consuming its own data - this is a very interesting usage.

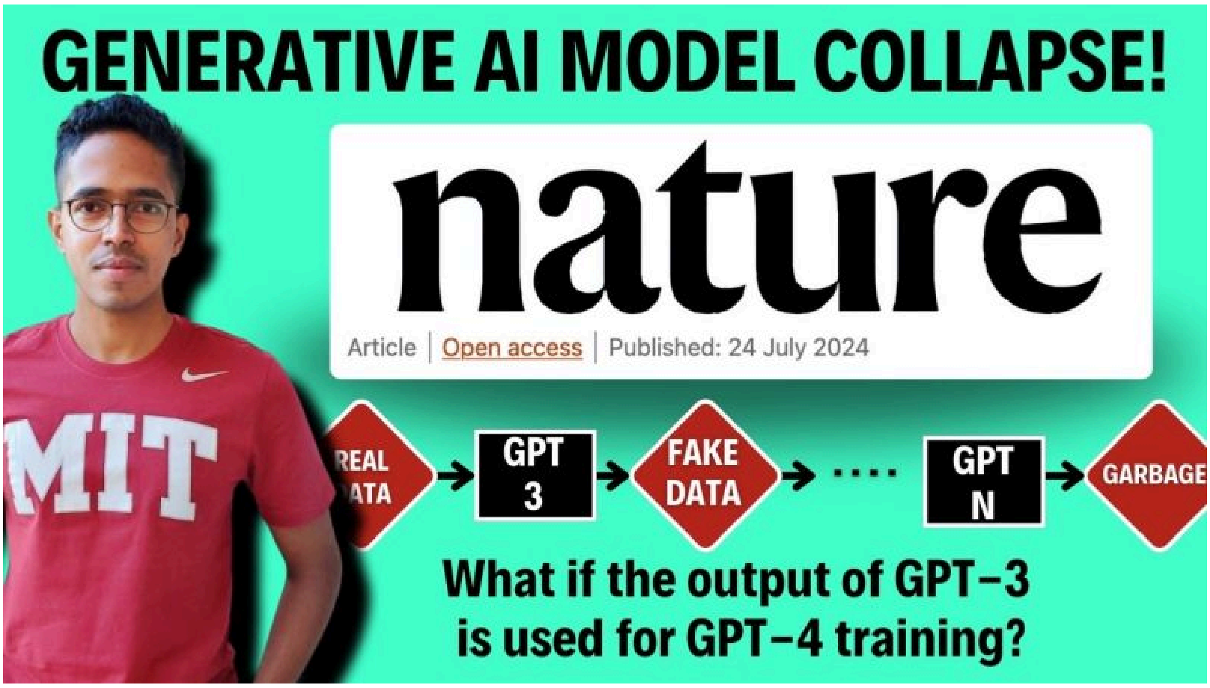
2) Model Autophagy Disorder (MAD): Autophagy happens when you fast for a long time and your body starts feeding on its own tissue. There is also a reference here to mad cow disease - a degenerative disease that affects a cow's central nervous system, including its brain and spinal cord

3) Degenerative AI - my favorite terminology.

4) Digital incest - this is probably the most explosive way to put it.

On [Vizuara](#)'s YouTube channel, I have released a 30-minute lecture that gives a simple walkthrough of this fascinating paper. Check this out: https://lnkd.in/g7b8AXRh

Although training on multiple rounds of synthetic data is simply not done in reality, a large part of the internet is trash (or becoming so). Perhaps the authors of the original article were frustrated about this and wanted to explore the worst-case scenario. They have done a great job!


GENERATIVE AI MODEL COLLAPSE!
nature
Article | Open access | Published: 24 July 2024
MIT
GPT
3
FAKE
DATA
GPT
N
GARBAGE
What if the output of GPT-3
is used for GPT-4 training?

Generative AI Watermarking

It's getting increasingly difficult to distinguish between human and AI generated text.

We recently got a paper accepted at NeurIPS Workshop which proposes ways in which we can use ML models to detect whether a text is human or AI.

Yesterday, Google DeepMind released a paper titled “Scalable watermarking for identifying large language model outputs”.

Although in its early stages, I believe that this study can be the start of some very serious research in the AI plagiarism space.

1️⃣What is watermarking for LLMs?

Imagine you're holding a 100 rupees note. It has several markings to verify whether it’s real or fake.

Similarly, a watermark in an LLM is a subtle signal added into the generated text.

This signal, undetectable to the human eye, allows for the identification of text generated by a specific LLM.

Instead of simply choosing the most likely next word, the LLM uses a modified sampling process called Tournament Sampling.

This process carefully selects words that simultaneously fit the context and align with a hidden "watermark code" derived from a secret key.

A special scoring function, using the same secret key, analyzes the generated text. It checks for patterns that indicate the presence of the watermark.

2️⃣How does watermarking differ from current methods of plagiarism detection?

Current methods use machine-learning-based classifier to distinguish human-written from artificial-intelligence-generated text.

Such classifiers fundamentally rely on underlying differences between machine and human text, which may go on decreasing as LLMs improve.

These classifiers operate after the text is generated.

Watermarking on the other hand is done during the generative process itself, by editing already generated text (edit-based watermarking) or by altering the LLM's training data (data-driven watermarking).

3 What are the applications of this?

- Teachers can reliably evaluate student submissions as AI or human generated.

- Artists can watermark their AI generated art.

- Images and videos can be flagged as AI or human generated.

4 What are the challenges of adopting watermarking for all LLMs?

- Will all AI companies be onboard to do this?

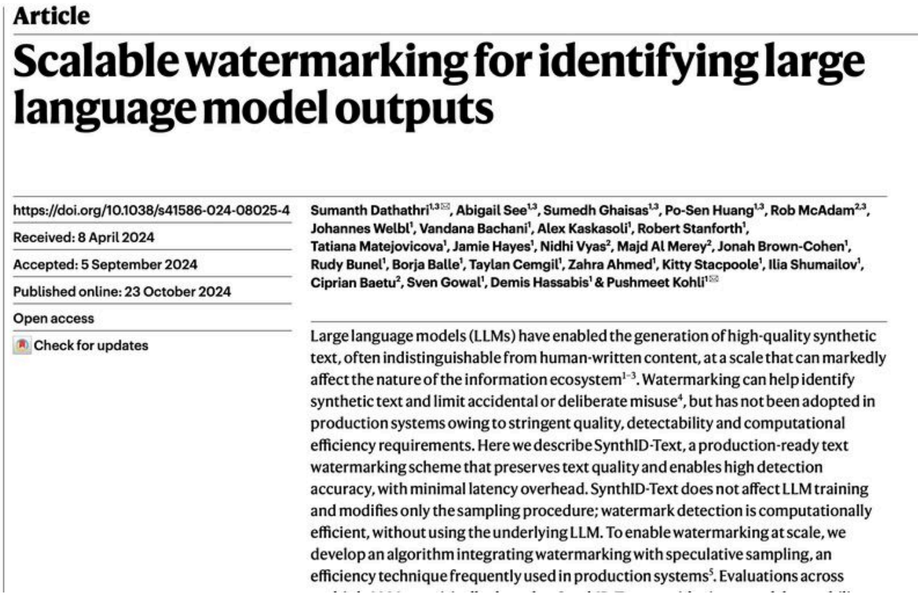
- How many AI companies will add watermarking to their text generation process?

- What about OpenSource LLMs? Since anyone in the world is free to modify these models, how can we incorporate watermarking here?

-What about humans modifying the generated text and bypassing the watermark?

Consider all this, I do believe that this work is in its early stages. However, it does open the door to a much needed research effort on developing AI plagiarism tools.

Author: Dr. Raj Dandekar



Article

Scalable watermarking for identifying large language model outputs

https://doi.org/10.1038/s41586-024-08025-4

Received: 8 April 2024

Accepted: 5 September 2024

Published online: 23 October 2024

Open access

Check for updates

Sumanth Dathathri[1,3], Abigail See[1,3], Sumedh Ghaisas[1,3], Po-Sen Huang[1,3], Rob McAdam[2,3], Johannes Welbl[1], Vandana Bachani[1], Alex Kaskasoli[1], Robert Stanforth[1], Tatiana Matejovicova[1], Jamie Hayes[1], Nidhi Vyas[2], Majd Al Merey[2], Jonah Brown-Cohen[1], Rudy Bunel[1], Borja Balle[1], Taylan Cemgil[1], Zahra Ahmed[1], Kitty Stacpoole[1], Ilia Shumailov[1], Ciprian Baetu[2], Sven Gowal[1], Demis Hassabis[1] & Pushmeet Kohli[1]

Large language models (LLMs) have enabled the generation of high-quality synthetic text, often indistinguishable from human-written content, at a scale that can markedly affect the nature of the information ecosystem[1–3]. Watermarking can help identify synthetic text and limit accidental or deliberate misuse[4], but has not been adopted in production systems owing to stringent quality, detectability and computational efficiency requirements. Here we describe SynthID-Text, a production-ready text watermarking scheme that preserves text quality and enables high detection accuracy, with minimal latency overhead. SynthID-Text does not affect LLM training and modifies only the sampling procedure; watermark detection is computationally efficient, without using the underlying LLM. To enable watermarking at scale, we develop an algorithm integrating watermarking with speculative sampling, an efficiency technique frequently used in production systems[5]. Evaluations across

Generative AI based Operating System

Instead of Windows or Mac as your operating system (OS), what if you had a Large Language Model.

What if all the below tasks are done by an LLM without you manually doing anything:

1️⃣"Go to my Documents folder and find all PDF files. Move the ones larger than 50 MB to a new folder called 'Large PDFs'.

2️⃣"In my Pictures folder, rename all files that contain the word 'Screenshot' to 'Screenshot_[Date]_[Time]' based on their creation date.

3️⃣"Monitor my system's CPU usage. If any program is consuming more than 70% of the CPU for more than 10 minutes, notify me and automatically close non-essential applications.

Anthropic has announced Computer Use (Public Beta).

Link: https://lnkd.in/gvdZ2_hK

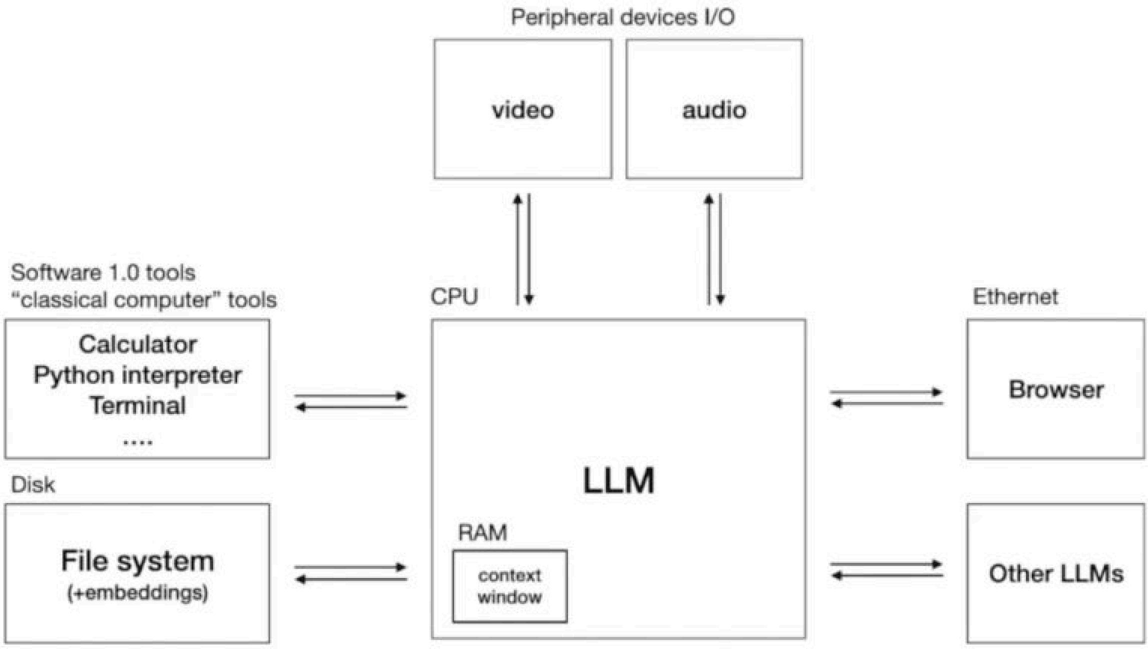
It allows Claude to control your computer screen based on a prompt and take actions on your behalf.

AI models can now perform actions directly on your computer, revolutionizing workflows in coding, debugging, and beyond.

The potential of agentic coding with automated debugging, customer support, and education is set to skyrocket, thanks to new capabilities in direct screen control and real-time interaction.

Excited for this advancement and how it might fundamentally change the way we interact with computers.

Author: Dr. Raj Dandekar


Peripheral devices I/O
video
audio
Software 1.0 tools
“classical computer” tools
Calculator
Python interpreter
Terminal
....
CPU
Ethernet
Browser
LLM
Disk
RAM
File system
(+embeddings)
context
window
Other LLMs

MS/PhD Applications

Keep the SOP genuine and simple. Focus on communication, not sophistication

For a successful MS/PhD application, you need a captivating, coherent story. The story should not be convoluted. It should be straightforward and must appear genuine. You should upsell yourself, not by using complex words, but by using simple logic.

If you don't have a great story now, you have to craft one. This story should manifest beautifully in your SOP, Resume, LORs, and portfolio website.

Author: Dr. Sreedath Panat

Statement of Purpose (SOP)

Using complex jargon and sophisticated language. Quoting Albert Einstein or Elon Musk and mentioning how they inspired to do research.

Being genuine. Plain introduction of who you are, what you are doing, and what you wish to do in the future.

This is not a family

If you notice that your potential PhD advisor or future manager excessively uses the word "family", be very cautious. "Family" usage might be a way to cover their toxic workplace culture.

Companies and research groups are not families. Families do not "fire" their members or "evaluate" their performance. They stick together at tough times and truly love and care for each other.

A better analogy would be that of a sports team where each player has a different role and the goal is to make the team win. Team members' performances are constantly evaluated and they can get fired unlike family.

Author: Dr. Sreedath Panat

Be a good banana seller

I received many messages when I defended my PhD at MIT and posted about it. Most of them were about how to get into US universities, including MIT for grad school.

Here is my view on the application process.

University application is about selling your profile. You may not have total control over GPA, you have absolute control over SOP and resume to pitch a compelling story. However, students usually drastically undersell their profiles.

When I applied to MIT and other top US universities, my profile was good. But what helped me stand out from 95% of the applicants from around the world was the compelling storytelling technique I used.

Think of your profile as a banana. Your profile has strengths and weaknesses. Like a normal banana with delicious parts and some overripe/stale portions :(. However, if you were a banana seller, how would you sell? When there are so many top applicants with great GPAs and research profiles, you need to be like a great banana seller who can convince the customer of the banana's value regardless of how it looks.

Always ask: “Am I underselling myself here?”. For example, if you have done 2 course projects and worked incredibly hard on one of them, it doesn't make sense to mention both in the same “course projects” section on the resume. You have to distinguish the values and sell them appropriately. Most students fail to recognize this. Very frequently I have seen strong candidates apply for PhD positions, burying their great publications deep in their resume and mentioning irrelevant (for the position) software skills at the very top. If you were a banana seller, would you bring the first attention of your customer to the overripe dots of the banana you are trying to sell?

When you pitch your profile, be extremely passionate. Most of the SOP's I have seen have paragraphs with abrupt starting, resume-esque points in the body, and abrupt ending. Then the admission committee won't be able to see your project's value. If you are particularly proud of a certain work, you must have some emotions to convey apart from numbers. Say something as simple as- "This was the project where I really felt proud of myself" in the SOP.

Finally, be ready to spend a lot of mental energy to identify underselling. Once you make it a habit to identify your poor salesmanship, you will slowly emerge to be a compelling storyteller, like a successful banana seller who can sell any kind of banana. Especially if you are applying to top universities like MIT, they only want to buy great bananas :).

Author: Dr. Sreedath Panat

Do not embark

Try not to use "embark", "delve", "leverage", and "tapestry" in your SOP. ChatGPT has overused these words and the mere presence of these words in your essay might create a negative bias.

Author: Dr. Sreedath Panat

Research

HULLMI: Our NeurIPS workshop paper

Today is a very happy day for us. Our first paper as part of Vizuara AI Labs Research titled "HULLMI: HUMAN VS. LLM IDENTIFICATION WITH EXPLAINABILITY" is published on arXiv: https://lnkd.in/dXfxbczi

When many teachers from our partner schools expressed a pressing need for an AI plagiarism checker, we decided to build an online tool for this (similar to or better than Quillbot AI detector). But while making this product we realized that this is not an easy problem statement. It is a full-scale research problem statement.

So we decided to work on this paper, spearheaded by PRATHAMESH JOSHI and Sahil Pocker as part of Vizuara Summer of Code (VSoC).

We explored how traditional ML methods compare with advanced NLP models like T5-sentinel and RoBERTa-Sentinel. The results are very interesting.

If you take a random AI vs human dataset from Kaggle or HuggingFace, you will most likely get 100% test accuracy as we did in our initial experiments. But then when you test with a custom dataset generated by ChatGPT, the results will be very poor. The model might classify ChatGPT generated text as human-written with 99% confidence.

There are 3 important aspects we realized early on to focus on if we need any meaningful results.

1. Quality of dataset
2. The model
3. and hyperparameters

Nothing fancy.

With T5, we obtain a custom test-data accuracy of 98% with a true positive rate of 99.6%. We have published the code in this GitHub repo: https://lnkd.in/dwa3Vqe2

The models work fantastic for actual AI-generated text detection.

For Raj, Rajat and me, it has been an absolute pleasure to work with PRATHAMESH and Sahil on this paper. Your work has been very inspiring.

We are continuing to work on submitting an advanced version of this paper to NeurIPS next year.

If you are interested in working AI/ML research projects with us, feel free to reach out.

Author: Dr. Sreedath Panat

HuLLMI: Human vs. LLM Identification with Explainability

Prathamesh Dinesh Joshi
Indian Institute of Information
Technology Design and Manufacturing,
Kurnool, India 518008
121cs0072@iiitk.ac.in

Sahil Pocker
Vizuara AI Labs
Bangalore, India
sahilpocker98@vizuara.com

Raj Abhijit Dandekar
Vizuara AI Labs
Bangalore, India
raj@vizuara.com

Rajat Dandekar
Vizuara AI Labs
Bangalore, India
rajatdandekar@vizuara.com

Sreedath Panat
Vizuara AI Labs
Bangalore, India
sreedath@vizuara.com

September 10, 2024

How much do PhD students overwork?

PhD programs are known for their overwork culture, as they are for being severely underpaid. Extreme overwork is a reality for a lot of PhD students and tenure-track professors. I am writing this article to share my views and experiences from my 5-years at MIT.

1. There are students who are super motivated to work. They "live" research. However, on average, the number of hours you work is by far determined by the lab you join. Some labs have it extreme where students are forced to work 90+ hours every week but are paid for only 20 hours per week.

2. Your lab determines whether you will work on the weekends or if you will be present in the lab at 3 am. I have seen this happen mostly with students working with tenure-track professors whose career depends on the research output of students.

3. There are labs that are infamous for making their students seek mental health help. The boundary between toxic lab culture and overworking lab culture is very thin.

4. I have heard several horrible stories. One in which a visiting student was made to work extremely long hours and was made to cry at the end of 2 months for not producing publish-worthy results; another in which a PhD student worked 18-hour work days continuously and temporarily lost their vision due to lack of sleep; and another in which the professor made her student work on a paper manuscript on New Year's Eve at 12 am, while the rest of the university was celebrating.

5. It is very hard to recognize that you are being overworked because your lab mates will be doing the same. If you have poor sleep or wake up in the middle of the night thinking about work regularly, it is very likely that you are in such an environment.

6. There are a lot of business owners who work incredibly long hours. There are farmers who wake up at 5 am and take care of their own farmland till 7 pm or beyond. Similarly, there are a lot of researchers who love working long hours. But if you are forced to work long hours due to an overworking culture in the lab, it is against the very spirit of the scientific discovery process.

I am writing this article to inform the students to think before committing to particular labs for their PhD. Please try to identify the work culture before you commit, for the sake of your mental and physical health. Know your rights. There are plenty of ways to reach out to the current students in any lab.

Author: Dr. Sreedath Panat

Research is hugely underrated

Here’s my machine learning publication history in undergraduate and PhD:

Undergraduate at IIT Madras:

2013 (1st year): 0 publications
2014 (2nd year): 0 publications
2015 (3rd year): 0 publications
2016 (4th year): 0 publications

PhD at MIT:

2017 (1st year): 0 publications
2018 (2nd year): 0 publications
2019 (3rd year): 0 publications
2020-2021 (4th, 5th year): 4 publications. 500+ citations until today

Those publications in my final year of PhD got me ML job offers, I defended my PhD thesis, got featured in MIT News and collaborated with governmental agencies.

Research is like the growth of the Chinese Bamboo tree. You take a little seed, water it and fertilise it for years. Nothing seems to happen.

During the 5th year, the bamboo tree grows 90 feet in 6 weeks!

Research takes a very long time to show results. The first 3 years of both my undergraduate and PhD had no results.

If you are patient, the results will come and the results will be very, very impactful.

Having respectable publications led to the biggest career transitions for me.

It has been the highest return on investment of my career. The investment of countless hours spent in the lab with no results, lab transitions, reviewer rejections and years of working on a problem in isolation.

Research is underrated. The impact it can have on your career is not just underrated, it’s neglected.

It’s time to take research seriously.

Author: Dr. Raj Dandekar

You don’t need a million dollars

You don't need a million dollars in funding to get started on AI/ML research. You don't need professors or a university ecosystem. All you need is 3 people who have a research mindset who share a similar vision as you, and who can motivate you when your research idea does not work out. You will need motivation to write your results and rewrite them. Our latest 2 papers from Vizuara AI Labs published on arXiv and submitted to NeurIPS workshop so far cost us ~$20 in LLM API, ~$60 in GPU, ~$50 in other software subscriptions, and some good quality human hours over the last few months. This makes me so happy because this new system we have built opens a new door for me to stay as part of academia without having to become a professor. It also helps me stay on the cutting edge in some areas of AI/ML research that I really like. Also, I get to work with students from any part of the world. I am very much enjoying doing research with some great minds while building a startup.

Author: Dr. Sreedath Panat

Tools that immensely helped me during my research at MIT

The most prolific researchers I have seen were masters of software. Here I share with you the most important tools that helped me during my PhD at MIT and beyond.

1. Mendeley: I use Mendeley Desktop to manage references, its Chrome Extension to quickly add papers I find to Mendeley Desktop, and its MS Word extension to add citations to my Word docs.

2. Overleaf: The best online LaTeX editor. Various journals have their LaTeX template on Overleaf. MIT and other universities have their thesis templates too. PDFs compiled on Overleaf look elegant and classy.

3. MS Excel: The best tool for plotting graphs that aren't too complex. There is so much customization available. If you master Excel, you can create publish-quality plots very quickly.

4. Blender: A free 3D modeling software. I used to create hyperrealistic renders of my experimental setup to be included as figures in presentations, papers, and thesis. Beautiful 3D figures elevate the visual appeal of your work, something that many researchers ignore.

5. Adobe Illustrator: The best tool for creating 2D schematics/sketches. I think all researchers should master Illustrator. I have seen hundreds of papers with poor figures. Crafting poor figures creates a bad impression of your work.

6. Canva: The best tool for creating presentations for storytelling. I used to use MS PPT initially and later switched to Canva. I am never going back to PPT or Google Slides. I envy the founders of Canva for finding value in a space that was completely dominated by MS PPT and Adobe.

7. Litmaps: A brilliant tool for quickly finding lit-review papers in your domain. It generates a visual flowchart of all papers at a glance, reducing the time taken to find papers by ~50%.

8. Grammarly: The biggest no-no in scientific writing is spelling and grammar mistakes. Install Grammarly plugin for Chrome. It helps you while compiling online documents and sending emails.

9. Adobe Stock Images: If you are presenting your work, you need to show high-quality images. Adobe Stock is a reliable source for copyright-cleared images. Beautiful images improve the quality of your presentation by 2X.

10. GitHub Desktop: Best for collaborating on code. It's essential for computational/ML projects, as it

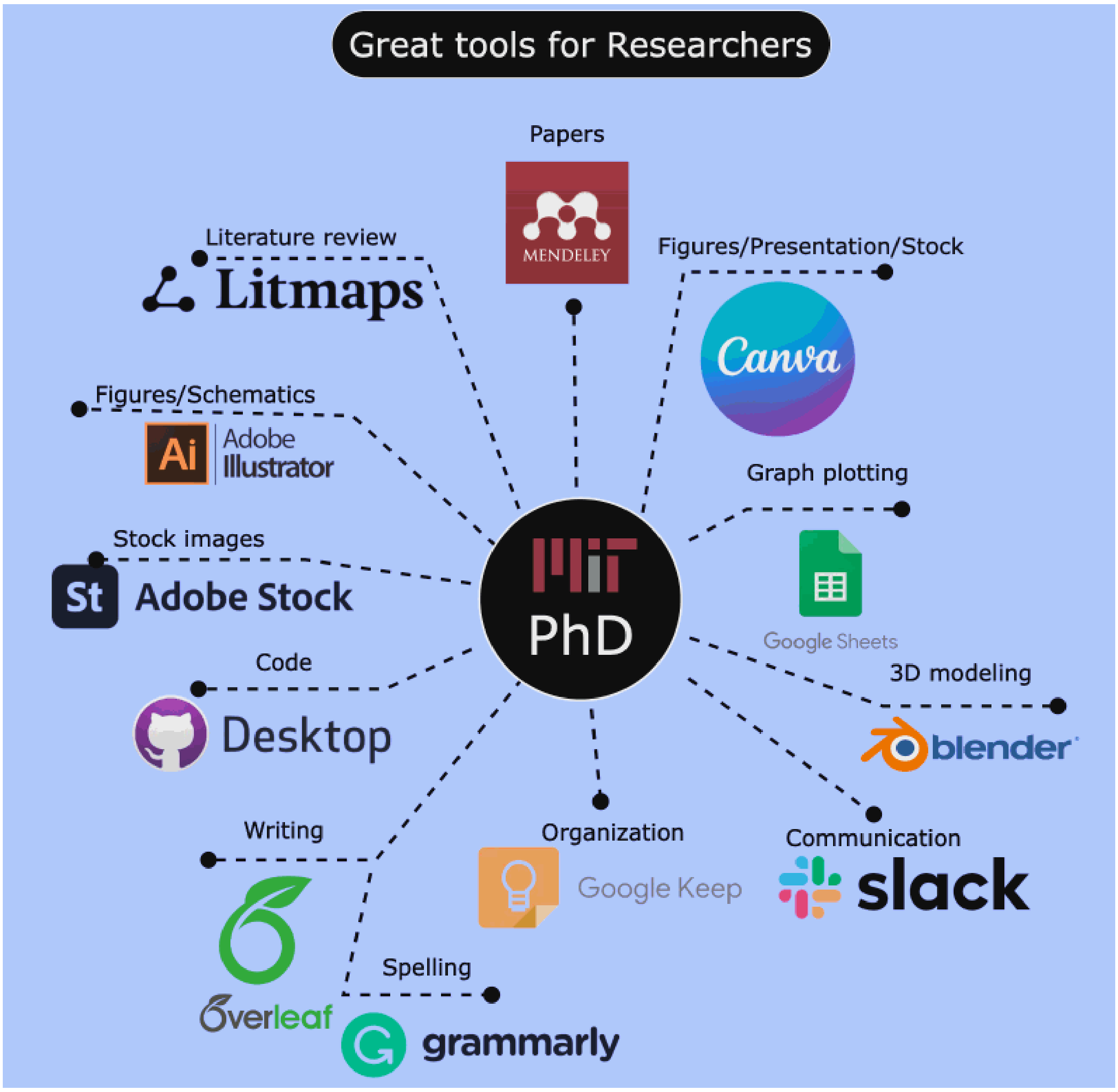
allows for code review, version control, and collaboration. I find it very convenient to use Github Desktop instead of terminal. Highly recommended.

11. Google Keep: A tool that I extensively use for to-do lists. I really like how we can strike through the finished tasks. It gives immense satisfaction.

12. Slack: The best tool for collaborating and communicating in teams, creating channels for various topics, and sharing files. You can also integrate various other software with Slack to make it the one place to work.

These tools have significantly enhanced my productivity and organization throughout my PhD, and I highly recommend them to anyone doing research.

Author: Dr. Sreedath Panat



Evaluating Cultural Awareness of LLMS

I am very happy to announce that our latest paper from Vizuara AI Labs on "Evaluating Cultural Awareness of LLMS for Yoruba, Malayalam, and English" has been published on arXiv: https://lnkd.in/gdjnHKiA

We wanted to answer a simple question: "If LLMs are predominantly trained in English, does it mean that LLMs capture the cultural nuances of English-speaking people better than those of those who speak regional languages?"

We wanted to quantify, how much LLMs are aware of the cultural nuances.

Personally, this was my first research project involving a detailed survey of a large number of people.

To quantify cultural awareness, there is a famous method based on Hofstede's cultural dimensions. By asking the respondents to answer a series of binary questions, we can quantify the masculinity index, power imbalance, long-term orientation etc of a society.

We conducted our survey among close to 100 people each from Malayalam-speaking and Yoruba-speaking communities. The survey was conducted to quantify Hofstede's cultural dimensions. Then the same survey was conducted over different LLM agents. Then we created a similarity score.

Our results indicate that indeed the LLM vs human cultural awareness score is better for English compared to Malayalam or Yoruba. Please check the paper for details.

Raj, Rajat, and I want to thank Fiifi Dawson, Zainab Mosunmola Hassan, and Sahil Pocker for all the hard work. It has been a pleasure working on this project with all of you. Also, congrats Fiifi for presenting our work at the recent Pycon Africa conference.

If anyone is interested in publishing follow-up papers based on this research, here are a few areas for further improvement

1) Increase the number of survey respondents: 1000+ people
2) Increase the spectrum of human survey questions from a few dozen to 400+
3) Conduct the study for more regional languages: dozen+
4) Increase the number of LLM agents over which the survey is conducted

Author: Dr. Sreedath Panat

Evaluating Cultural Awareness of LLMs for Yoruba, Malayalam, and English

Fiifi Dawson[1], Zainab Mosunmola[1], Sahil Pocker[1], Raj Abhijit Dandekar[1], Rajat Dandekar[1], and Sreedath Panat[1]

[1]Vizuara AI Labs

October 4, 2024

Abstract

Although LLMs have been extremely effective in a large number of complex tasks, their understanding and functionality for regional languages and cultures are not well studied. In this paper, we explore the ability of various LLMs to comprehend the cultural aspects of two regional languages: Malayalam (state of Kerala, India) and Yoruba (West Africa). Using Hofstede's six cultural dimensions: Power Distance (PDI), Individualism (IDV), Motivation towards Achievement and Success (MAS), Uncertainty Avoidance (UAV), Long Term Orientation (LTO), and Indulgence (IVR), we quantify the cultural awareness of LLM-based responses. We demonstrate that although LLMs show a high cultural similarity for English, they fail to capture the cultural nuances across these 6 metrics for Malayalam and Yoruba. We also highlight the need for large-scale regional language LLM training with culturally enriched datasets. This will have huge implications for enhancing the user experience of chat-based LLMs and also improving the validity of large-scale LLM agent-based market research.

1 Introduction

Large Language Models (LLMs) have emerged as powerful tools in Natural Language Processing (NLP), demonstrating remarkable capabilities in tasks ranging from text generation to complex reasoning, including language translation and summarization [1, 2]. These models, trained on vast amounts of textual data, have revolutionized language-based AI applications [2]. However, a critical issue has come to light: the majority of LLMs are primarily trained in English language data, introducing several limitations and biases [3].

The current state of LLM development heavily favors English, with most large-scale datasets and training procedures focusing on English text by 30-60%[4]. This bias is partly due to the abundance of English content on the internet and in digital repositories, as well as the dominance of English in scientific and technological discourse. While some efforts have been made to incorporate other languages, the extent and effectiveness of these inclusions remain limited [3].

Due to the training data bias, LLMs may have a significantly reduced ability to respond effectively in other languages. This limitation manifests in various ways, including reduced accuracy in translation, poor understanding of cultural nuances, and inability to generate coherent text in non-English languages [5]. For instance, in sentiment analysis, LLMs can exhibit biases favoring dominant cultural groups, leading to inaccurate interpretations in languages like Italian, Chinese, and Spanish [6].

Moreover, these models often generate text containing social biases related to gender, age, sexual orientation, ethnicity, religion, and culture, highlighting the need to mitigate such biases [7]. LLMs also struggle with accuracy and fluency in non-English languages due to insufficient high-quality training data [8] and structural differences between languages [9], resulting in less coherent and contextually inappropriate responses [10, 11].

25 pain points as a PhD student

Wrapping up your PhD is the greatest joy. I got to experience this after 5 years of blood and sweat at MIT. However, no PhD is ever easy. Researchers go through struggles of all kinds.

Before joining for my PhD, I had a rosy picture in my mind. I thought I would sip some hot coffee and come up with theorems and experimental ideas that no on has ever come up with. Boy, I could never be so wrong.

Many of my friends also experienced something similar. PhD experience really humbles you. If you think PhD is just about science, check out this list of pain points from experiences of my own and other students.

1. Conflicts with advisor
2. Not knowing if your RA will last till the end of your PhD or if you'll have to find TAship
3. Lost interest in your project
4. Getting paid much lower than your industry peers
5. Confused about whether you should go for a post-doc to buy yourself some time or whether the post-doc itself is a waste of time
6. Your lab is too small, so the advisor is micromanaging you
7. You are the only one working on a project that the lab has no expertise on
8. You find the solution you are proposing stupid
9. You cannot identify any gap in the literature
10. You have some good results. But someone has already published it.
11. You are writing a grant proposal. You know that what your advisor wants you to propose is not scientifically feasible. So you are worried that if the grant is approved you will be the one who will work on the project.
12. Your friends or lab mates have already published their work
13. You don't know when your advisor will allow you to defend the thesis
14. Your research has no relevance to the industry. So you are worried about jobs.
15. There is a major loophole in your paper, and you are worried the reviewers will find it
16. You are getting older, and you don't know if/when/how you will find a life partner
17. You feel like you should have just done masters and joined the industry
18. You feel like you have no specific skills from PhD. So you can apply for only consulting jobs.
19. You wish to become a professor, but you do not have sufficient papers
20. You want to do an internship during the PhD to convert that into a full-time offer, but you don't know how to convince your advisor
21. You have a meeting with your advisor tomorrow, and you haven't done anything since the last meeting
22. You have course assignment deadlines. But your advisor is pushing you to finish some experiments.
23. Your friend is defending his/her thesis and is moving out

24. You are stressed with poor sleep and are facing mental health troubles
25. Some of your lab mates are in the 7th year of PhD and you worry if you will also take that long

Author: Dr. Sreedath Panat

Nobel Prize for Machine Learning. What a time to be alive!

Scientists John Hopfield (Princeton University) and Geoffrey Hinton (University of Toronto) won the 2024 Nobel Prize in Physics for discoveries and inventions that enable machine learning within artificial neural networks.

The official press release from the Nobel committee says:-

"This year's two Nobel Laureates in Physics have used tools from physics to develop methods that are the foundation of today's powerful machine learning. John Hopfield created an associative memory that can store and reconstruct images and other types of patterns in data. Geoffrey Hinton invented a method that can autonomously find properties in data, and so perform tasks such as identifying specific elements in pictures.
.....

When we talk about artificial intelligence, we often mean machine learning using artificial neural networks. This technology was originally inspired by the structure of the brain. In an artificial neural network, the brain's neurons are represented by nodes that have different values. These nodes influence each other through connections that can be likened to synapses and which can be made stronger or weaker. The network is trained, for example by developing stronger connections between nodes with simultaneously high values. This year's laureates have conducted important work with artificial neural networks from the 1980s onward."
Author: Dr. Sreedath Panat

Why Nobel Prize in Chemistry 2024 go to AI?

Half of the prize goes to Demis Hassabis (CEO of DeepMind) and John Jumper for using AI to predict protein structures, while David Baker receives the other half for creating entirely new proteins.

Proteins are built from 20 amino acids that fold into specific 3D structures. For decades, predicting these structures was a major challenge.

Hassabis and Jumper's AI model, AlphaFold2, solved this problem, predicting structures for nearly all known proteins with unprecedented accuracy.

Baker's work focuses on protein design. His software, Rosetta, can create proteins with entirely new structures and functions. This opens possibilities for developing new materials, pharmaceuticals, and greener chemical processes.

The journey to these breakthroughs involved many steps. Early protein structure determination used X-ray crystallography. It was slow and painful.

Christian Anfinsen showed that a protein's structure is determined by its amino acid sequence, leading to the prediction problem. The CASP competition was established to encourage progress in this field. Every other year, researchers from around the globe were given access to sequences of amino acids in proteins whose structures had just been determined. However, the structures were kept secret from the participants. The challenge was to predict the protein structures based on the known amino acid sequences.

Hassabis, a chess master and AI pioneer entered CASP in 2018 with AlphaFold. Jumper's expertise in protein simulation helped refine the model. In 2020, AlphaFold2 achieved near-experimental accuracy in structure prediction.

Baker's work began with studying protein folding. He developed Rosetta for structure prediction, then reversed the process to design new proteins. His team created Top7, the first protein with a completely novel structure.

These discoveries have immense potential. AlphaFold2 has predicted the structure of virtually all the 200 million proteins that researchers have so far discovered when mapping Earth’s organisms.

Baker's protein design work continues to produce incredible creations with diverse applications.

The ability to visualize and create proteins opens up new frontiers in understanding life, developing treatments, and creating new materials. This Nobel Prize recognizes work that is truly for the greatest benefit of humankind.

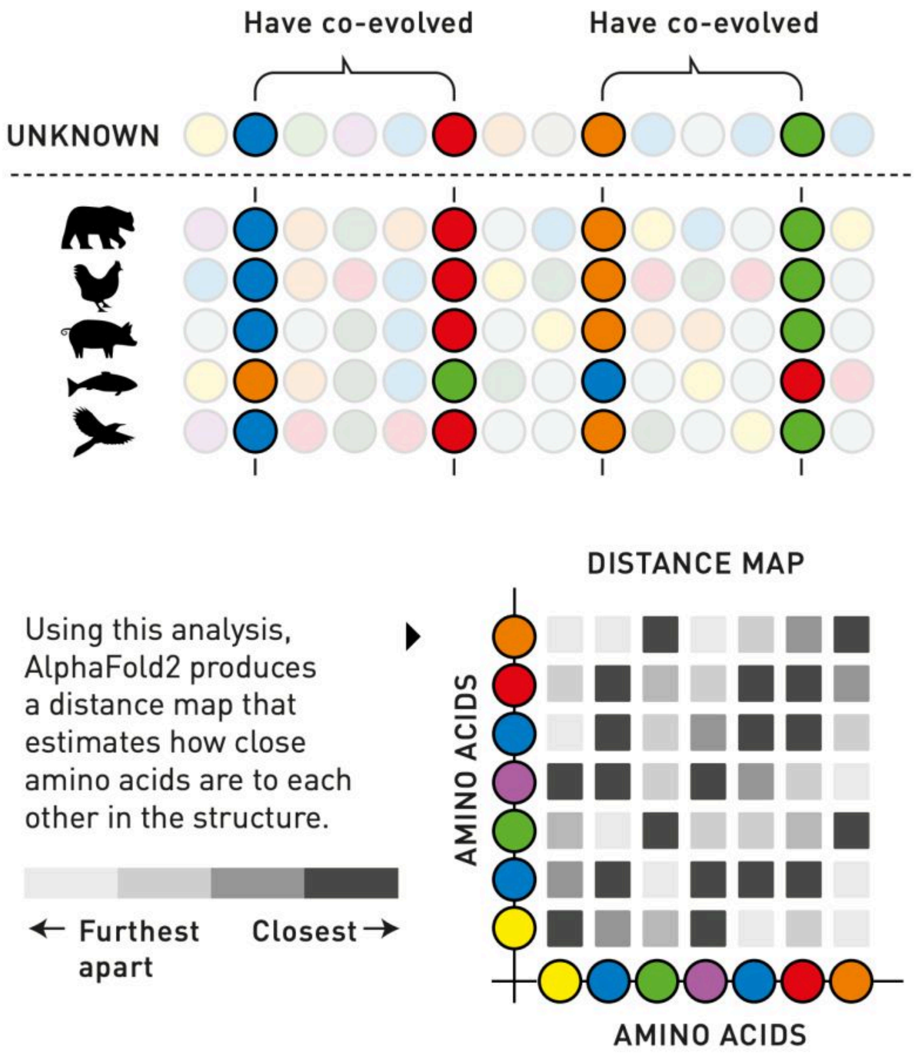
Are you wondering why proteins are such a big deal?

Many drugs work by interacting with specific proteins. Knowing a protein's structure helps us design drugs that can bind to specific sites on the protein, for targeted treatments for diseases.

Many diseases result from protein misfolding or mutations that alter protein structure. Understanding these changes can help in developing treatments or preventive measures for conditions like Alzheimer's, Parkinson's, and even cancer.

So yes proteins are a very big deal.

Author: Dr. Sreedath Panat



Physics Nobel Prize 2024 for AI. But why?

Why Nobel Prize in Physics 2024 goes to AI?

Scientists John Hopfield (Princeton University) and Geoffrey Hinton (University of Toronto) won the 2024 Nobel Prize in Physics for discoveries and inventions that enable machine learning within artificial neural networks.

The official press release from the Nobel committee says:-

"This year's two Nobel Laureates in Physics have used tools from physics to develop methods that are the foundation of today's powerful machine learning. John Hopfield created an associative memory that can store and reconstruct images and other types of patterns in data. Geoffrey Hinton invented a method that can autonomously find properties in data, and perform tasks such as identifying specific elements in pictures.
.....
When we talk about artificial intelligence, we often mean machine learning using artificial neural networks. This technology was originally inspired by the structure of the brain. In an artificial neural network, the brain's neurons are represented by nodes that have different values. These nodes influence each other through connections that can be likened to synapses and which can be made stronger or weaker. The network is trained, for example by developing stronger connections between nodes with simultaneously high values. This year's laureates have conducted important work with artificial neural networks from the 1980s onward."

Incredible!

****

So Why Nobel Prize in Chemistry 2024 goes to AI?

Half of the prize goes to Demis Hassabis (CEO of DeepMind) and John Jumper for using AI to predict protein structures, while David Baker receives the other half for creating entirely new proteins.

Proteins are built from 20 amino acids that fold into specific 3D structures. For decades, predicting these structures was a major challenge.

Hassabis and Jumper's AI model, AlphaFold2, solved this problem, predicting structures for nearly all known proteins with unprecedented accuracy.

Baker's work focuses on protein design. His software, Rosetta, can create proteins with entirely new structures and functions. This opens possibilities for developing new materials, pharmaceuticals, and greener chemical processes.

These discoveries have immense potential. AlphaFold2 has predicted the structure of virtually all the 200 million proteins that researchers have so far discovered when mapping Earth's organisms.

Baker's protein design work continues to produce incredible creations with diverse applications.

The ability to visualize and create proteins opens up new frontiers in understanding life, developing treatments, and creating new materials. This Nobel Prize recognizes work that is truly for the greatest benefit of humankind.

Here is a video published by me on Vizuara's YouTube channel explaining this year's Physics and Chemistry Nobel Prize. Check this out: https://lnkd.in/guUFvBmJ

Author: Dr. Sreedath Panat

We cracked NeurIPS!

Ever since graduating from MIT I had a nagging thought: "How can I be part of cutting-edge science while running a startup? If I do not know what is happening in the latest research, does it matter if I have a PhD?"

At Vizuara, we started a research wing this year to work on cutting-edge AI/ML problems.

The goal of our research wing is simple: only focus on the highest quality research and aim for top-tier publications.

Last week, 2 research papers from the Vizuara AI research wing were accepted at NeurIPS 2024 Workshops. NeurIPS is the most prestigious ML conference in the world.

(1) The first paper "A Comparative Study of Neural ODE and Universal ODE Models in Solving Chandrasekhar's White Dwarf Equation" is accepted at NeurIPS 2024 FM4Science Workshop. This paper is in the field of Scientific ML.

(2) The second paper "HuLLMI: Human vs LLM Identification with Explainability" is accepted at NeurIPS 2024 SFLLM Workshop. This paper is in the field of Large Language models.

Congratulations to the lead authors of these papers: SciML Bootcamp student Raymundo Vazquez Martinez and Vizuara Summer of AI student PRATHAMESH JOSHI and also Sahil Pocker from our team. They have all done incredible work and I am proud of them.

From Vizuara, there will be 2-3 researchers in Vancouver, Canada this year to present our work at NeurIPS.

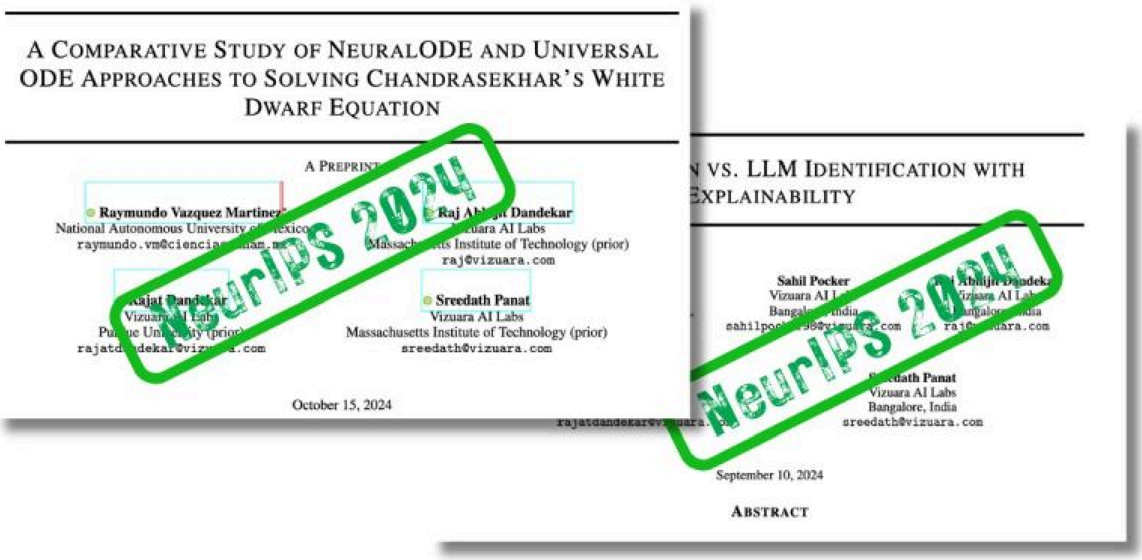
As an AI research organization, we look forward to meeting with other researchers attending the conference.

For me, Raj Abhijit Dandekar and Rajat Dandekar this is the biggest achievement since we started Vizuara, more than any other startup growth metric.

It took 5 years to reach a stage where Vizuara AI is at the forefront of impactful machine learning research.

I am totally convinced that you don't require a university ecosystem or a million dollars in funding to perform cutting-edge, high-impact AI/ML research. You just need 3 people, a vision, and solid human hours.

Author: Dr. Sreedath Panat


A COMPARATIVE STUDY OF NEURALODE AND UNIVERSAL ODE APPROACHES TO SOLVING CHANDRASEKHAR'S WHITE DWARF EQUATION

A PREPRINT

NeurIPS 2024

Raymundo Vazquez Martinez
National Autonomous University of Mexico

Raj Abhijit Dandekar
Vizuara AI Labs
Massachusetts Institute of Technology (prior)
raj@vizuara.com

Rajat Dandekar
Vizuara AI Labs
Purdue University (prior)

Sreedath Panat
Vizuara AI Labs
Massachusetts Institute of Technology (prior)
sreedath@vizuara.com

October 15, 2024

VS. LLM IDENTIFICATION WITH EXPLAINABILITY

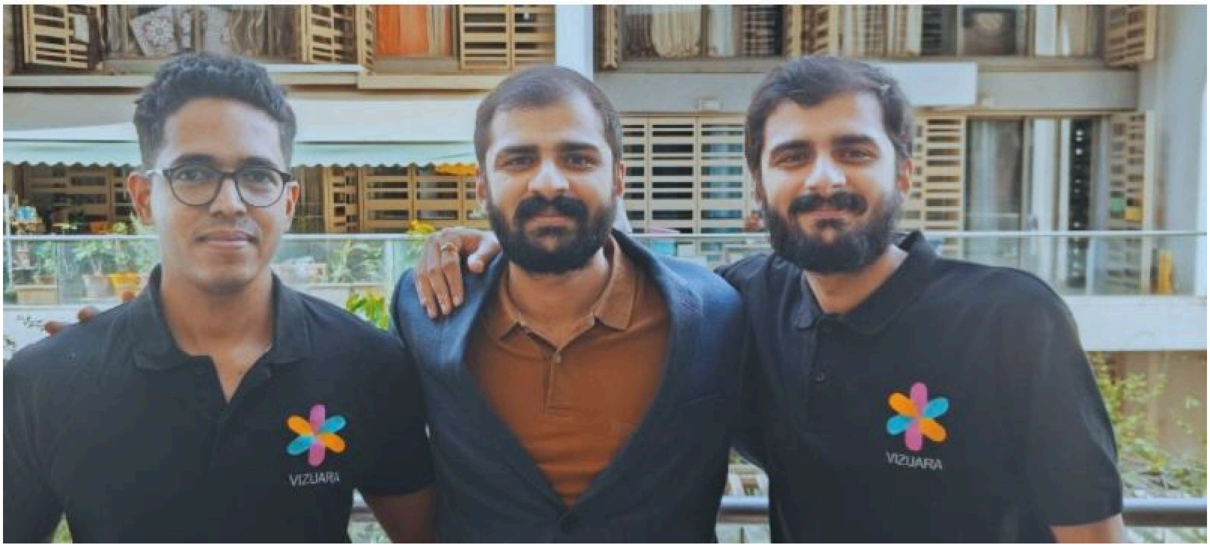
NeurIPS 2024

Sahil Pocker
Vizuara AI Labs
Bangalore, India

Sreedath Panat
Vizuara AI Labs
Bangalore, India
sreedath@vizuara.com

September 10, 2024

ABSTRACT



Cracking NeurIPS with zero fundion

We cracked NeurIPS (#1 ML conference) outside of academia, with $0 in funding, twice this year. How?

It is possible to conduct research without professors and a university ecosystem.

You don't need a million dollars in funding to work on fundamental and applied AI research.

You don't need a big lab.

You can do AI/ML research as a student or as an industry professional.

You don't need a PhD or fancy degrees.

All you need is a group of motivated people who can make things happen.

Here is a video I published on Vizuara's YouTube channel discussing how we cracked NeurIPS this year. Check this out: https://lnkd.in/gNtGgjzV

Author: Dr. Sreedath Panat

10 things I won’t repeat from my MIT PhD

My 5 years at MIT were irreplaceable. Yet, there are a few things I would do differently if I were to time travel and restart my PhD.

1) Boundaries:

Set your boundaries early with peers and advisors. Let others understand where you draw your line, respectfully. This will save you a lot of time and struggle. People will not take you for granted.

2) Mental and physical health

Give physical and mental health the #1 priority. There is no need to prioritize work unless there is an urgent deadline. PhD is a marathon, not a sprint. Work out, eat good food, sleep well, meditate, and socialize.

3) Comparison

Do not compare with your peers. PhD is not a game where if you publish more papers you win or if you don’t, you lose. Enjoy your research and persist. If your peers are publishing, be happy for them. Your time will also come.

4) Communication

Learn to communicate very effectively. Learn to write, make good figures, killer slides, and deliver a great talk. If you cannot communicate your work, your research is as good as non-existent. Science cannot communicate for itself. Scientists need to do the talking.

5) Actions over thoughts

I have wasted too much time thinking about imaginary experiments. I have wasted even more time pondering over what to write in my manuscript. If you want to do an experiment, go to the lab and do it. If you want to write a paper, then write a paper. Actions are real. Thoughts aren’t.

6) Asking for help

I suck at asking for help. I overestimate what I can do alone and underestimate what I can accomplish with the help of someone. Some of my peers were good at asking for help, and I envied them.

7) Assumptions

Once I was experimenting to measure the charge of dust particles under high voltage. I had a cardboard plate covering my setup. We know that cardboard is an insulator, right? No. This assumption cost me 3 months. It took me a long time to realize that cardboard has finite conductivity under high voltage. If you want to do original research, list down all assumptions you are making and question them.

8) Start writing

You won't achieve perfection in your project. Start writing your paper as soon as you have a good story. Don’t be disappointed if you find that someone else has published something similar. You always can find a unique perspective to pitch your work.

9) Be humble

Realize very early that you don’t know much. You might be a class topper. But it doesn’t matter. When you are researching a new topic, your textbook knowledge and analytical skills do not matter. You will inevitably fail.

10) After PhD

Plan what you want to do after PhD well before graduating. All of your actions during your PhD will depend on this. If you want to enter academia, attend conferences, network, and publish quality papers. If you want to enter the industry, do internships and go to career fairs early on. If you want to start up, try out accelerators and incubators.

Author: Dr. Sreedath Panat

Education

GPA doesn’t matter only on movies

I so much wish GPA did not matter too much in life because often GPA doesn’t equate with someone's knowledge or their ability to get things done.

However, the reality is different. Many companies look at your GPA while hiring you as a fresher. If you want to go for higher studies, GPA is one of the most important factors universities consider.

If you are a 1st or 2nd-year student in college and if seniors are advising you that your GPA does not matter, that simply isn’t true. GPA will affect your chances of internships, jobs, and higher studies. So do yourself a favor by taking GPA seriously.

That being said, I personally don’t care about someone’s GPA if we are hiring in our startup. I don’t know the GPA or the college of graduation of most of my team members. I mean, if someone has graduated with a B.Sc. in physics, but they are applying for a full-stack developer role in our startup, why does it matter if they know Noether’s theorem? Only their technical and communication skills matter.

Author: Dr. Sreedath Panat

Don’t let societal tags define you

Any engineer from any college can do the following:

1️⃣Can build a neural network from scratch
2️⃣Can make a full stack website
3️⃣Can build a large language model from scratch
4️⃣Can start Youtube channel to share knowledge
5️⃣Can deploy LLM applications
6️⃣Can learn to solve Leetcode DSA problems

If you don’t like your branch of study, that’s fine.

If you don't like your university, that's fine.

You made a decision to choose a branch and university after 12th standard. That should not define what you learn.

No need to box yourself based on your branch or your university,.

Everything is available for you to learn and master. The content for most of the above is available for free on Youtube.

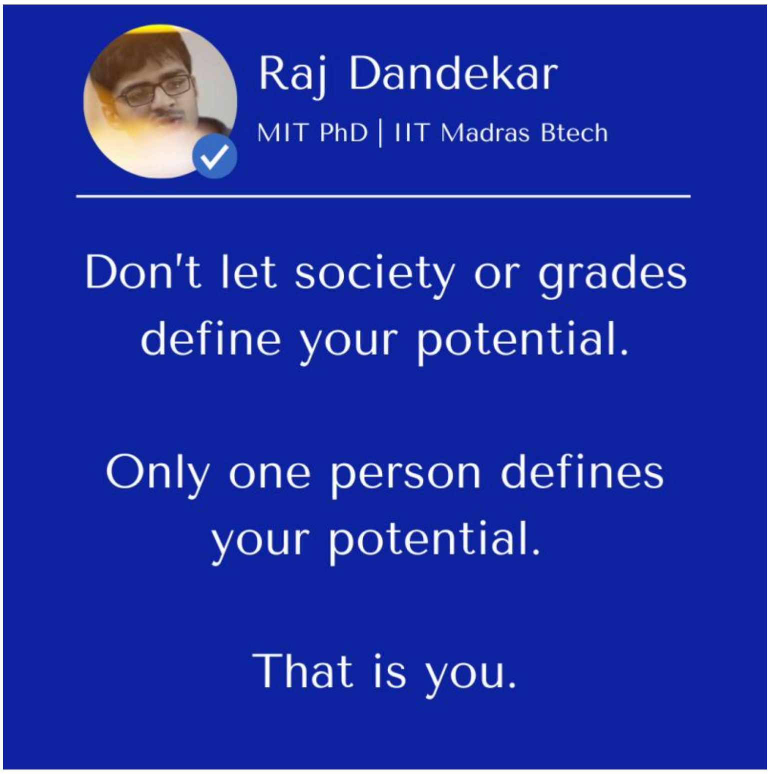
There is no excuse for additional learning: not your branch, not your college.

Have a hunger to learn and show up every single day!

Nothing defines your potential. Not society. Not your university. Not your grades. Not your professors. Not your parents.

Only one person can decide your potential.

That is you.



Author: Dr. Raj Dandekar

IIT vs MIT

Why doesn't India have universities like MIT, Stanford and Harvard which are in the world's top 10?

I have been admitted to both MIT and Stanford. Did my PhD for 5 years at MIT, and have also visited Stanford. I am also a graduate from IIT Madras, so you can trust me when I write this answer.

(1) Students: At MIT and Stanford, all students I interacted with were extremely passionate about what they were studying. They loved what they did so much that you could see the enthusiasm on their face and in their work. They didn't come there because they were forced, it was their choice.

In Indian universities, you just don't see the fire in most of the students. It's missing.They just drag themselves half heartedly to finish their course as if they were forced to do it. If the fire is missing in the students that make the university, how can the university be great?

(2) Professors: Indian professors have a secure job. In universities like MIT and Stanford, professors have to work really hard to keep their job and get tenure. Their research work and teaching has to be really great for them to keep their job.

In India however, how many times have you heard a professor was fired due to bad teaching or low quality research? Doesn't this job security itself lead to mediocrity? Then how do you expect our universities to be great?

(3) Diversity: MIT and Stanford attract talent from all over the world. This really makes the peer group you have unparalleled. You can have a business consultant, a solar panel expert, a rocket enthusiast and a traveller in the same room(My own experience).

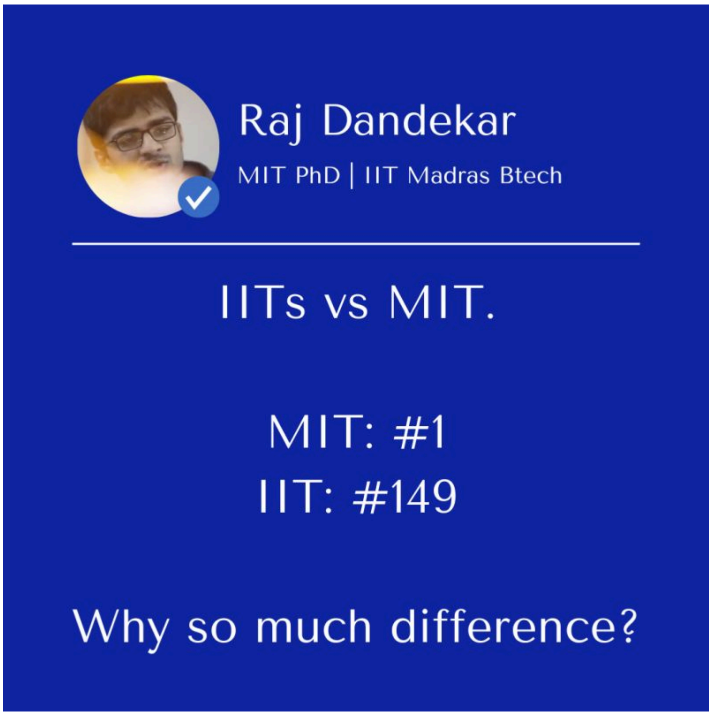
In Indian universities, students don't really know what their passion is. This might be due to the broken engineering entrance examination system and the pressure it puts on students. They find out their passion in those 4 years of engineering. Mostly they are not interested in their field of choice and go on to do different things. If the peer group is not great, then how can the universities be great?

(4) Funding: If a laser is broken in an Indian lab, it takes 2–3 months to repair it. Till then, the student has to wait. Two months wasted. If a laser is broken in an MIT/Stanford lab, they have 3 more of the same type ready. No time is wasted!

If the government doesn't increase the funding to match that of the best universities in the world, how can you expect the universities to be great?

(5) Improvements: To all the points above, there are of course exceptions. There are some top notch professors and research labs in India too. But don't you think some systemic changes are needed:

- Branch selection not based on rank, but based on choice.
- Tenure track system for professors.
- Increased international collaborations to increase student diversity on campus.
- Increased private/public funding to universities.



Author: Dr. Raj Dandekar

Overrated PORs

In Indian colleges, Positions of Responsibilities (PORs) are overrated. GPA and undergraduate research are underrated.

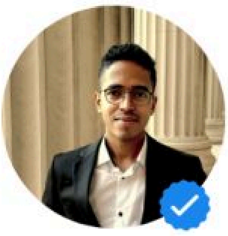
Author: Dr. Sreedath Panat

Vizuara's Goldmine

Vizuara's YouTube channel has become a goldmine for anyone who wants to master advanced AI/ML concepts and more. Here are 21 top playlists from our channel. Pick your topic and start learning.

1. Building LLMs from scratch: https://lnkd.in/gjcyfCcE
2. Building Neural Networks from Scratch: https://lnkd.in/gj8kHe2T
3. Machine Learning: Teach by Doing: https://lnkd.in/gn2dEcE2
4. Foundations for Machine Learning: https://lnkd.in/gKz-eybU
5. ML project based course: Explainable AI (XAI): https://lnkd.in/gNEx3ghr
6. Graph Neural Networks - Theory, Applications and Research: https://lnkd.in/g3RCPS8e
7. Building Decision Trees from Scratch: https://lnkd.in/g3cmj2BR
8. Hands-on Large Language Models: https://lnkd.in/gJQ7ryE4
9. Generative AI: From Fundamentals to Deployment: https://lnkd.in/gUgXxVzh
10. Data Structures and Algorithms in Python: https://lnkd.in/gy6S8Tiq
11. Introduction to Machine Learning in Julia: https://lnkd.in/g8A3DtQW
12. Transformers Series: https://lnkd.in/g_3Qdu6d
13. SQL MasterClass Series: https://lnkd.in/gg3Vb3A3
14. AI Researcher Bootcamp: https://lnkd.in/gRbpfBPA
15. Generative AI for Absolute Beginners: https://lnkd.in/g9xfzgPM
16. Zero To Hero in Data Science: https://lnkd.in/gNEgx2Cz
17. Git Github Masterclass: https://lnkd.in/ggrPe-r5
18. R MasterClass: https://lnkd.in/g2tF3A2t
19. Machine Learning in Hindi: https://lnkd.in/giD88GzZ
20. Hands on Physics Informed Neural Networks: https://lnkd.in/gta5hgHZ
21. Product Management: https://lnkd.in/gEubP8kp

Author: Dr. Sreedath Panat



Sreedath Panat

MIT PhD | IIT Madras

Top playlists on Vizuara's YouTube channel

1. Building LLMs from scratch
2. Building Neural Networks from Scratch
3. Machine Learning: Teach by Doing
4. Foundations for Machine Learning
5. ML project based course: Explainable AI (XAI)
6. Graph Neural Networks - Theory, Applications and Research
7. Building Decision Trees from Scratch
8. Hands-on Large Language Models
9. Generative AI: From Fundamentals to Deployment
10. Data Structures and Algorithms in Python
11. Introduction to Machine Learning in Julia
12. Transformers Series
13. SQL MasterClass Series
14. AI Researcher Bootcamp
15. Generative AI for Absolute Beginners
16. Zero To Hero in Data Science:
17. Git Github Masterclass
18. R MasterClass
19. Machine Learning in Hindi
20. Hands on Physics Informed Neural Networks
21. Product Management

youtube.com/@vizuara

Other technical topics

When it comes to data analysis, you have 2 options: Python and R.

In many data analysis interviews, it is common for interviewers to ask questions about the R language.

I learned R much later than Python. It's a beautiful language and leads to some pretty awesome data visualizations!

At Vizuara, Abhijeet Singh has recorded the most detailed lectures on R that I have come across.

It's a playlist of 500 minutes worth of content: extremely detailed and hugely informative.

Here is a list of lectures covered by Abhijeet:

(1) Why you should learn R: https://lnkd.in/gDJC2DsZ

(2) R installation and setup: https://lnkd.in/gn3Muhyu

(3) R Maths notation and vectors: https://lnkd.in/gEFuHJS9

(4) R Lists and DataFrames: https://lnkd.in/gctNdg62

(5) R Loops and Conditional Statements: https://lnkd.in/gGvrWkY5

(6) R Visualisation and why it's so awesome: https://lnkd.in/gqGRJbTw

(7) R Types of graphs: https://lnkd.in/gJq6k4aC

(8) R bi-variate graphs: https://lnkd.in/gUDtQuP2

(9) R multi-variate graphs: https://lnkd.in/gP9tyxPR

(10) R Maps: https://lnkd.in/g9-BeSt6

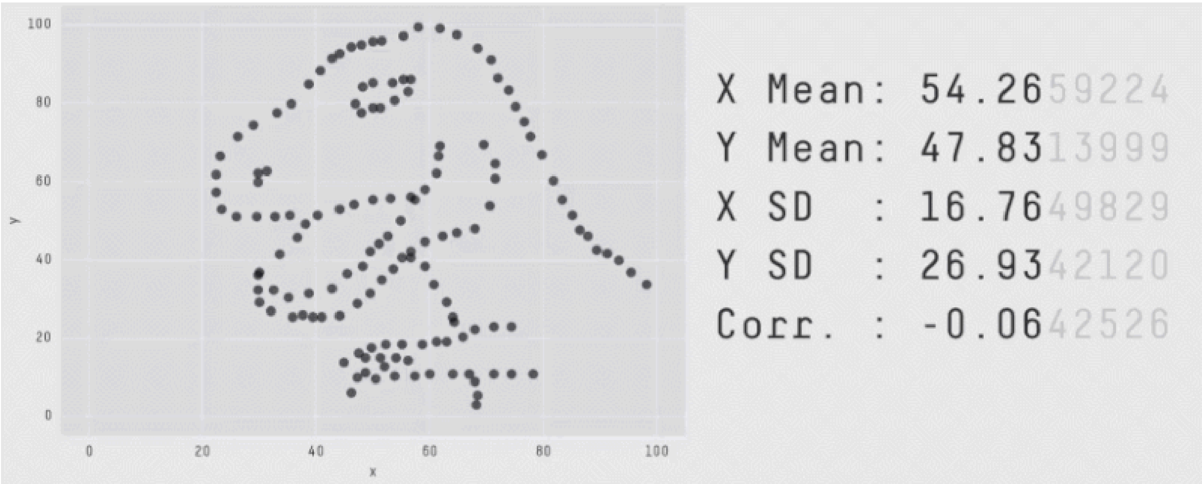
(11) R TimeSeries graphs: https://lnkd.in/gTZUnvpk

[Abhijeet]() has spent a lot of time and effort in making these lectures. He shows everything on a whiteboard and then shows it through Python code.

Nothing is assumed. Everything is spelled out.

Hope you learn and enjoy!

Author: Dr. Sreedath Panat



Bayes' Theorem: A Powerful Tool for Decision-Making

Bayes' Theorem is a cornerstone of probability theory, helping us update beliefs when new evidence is introduced. It is not just about the math—it is about intuition. Let me illustrate this with a real-world example.

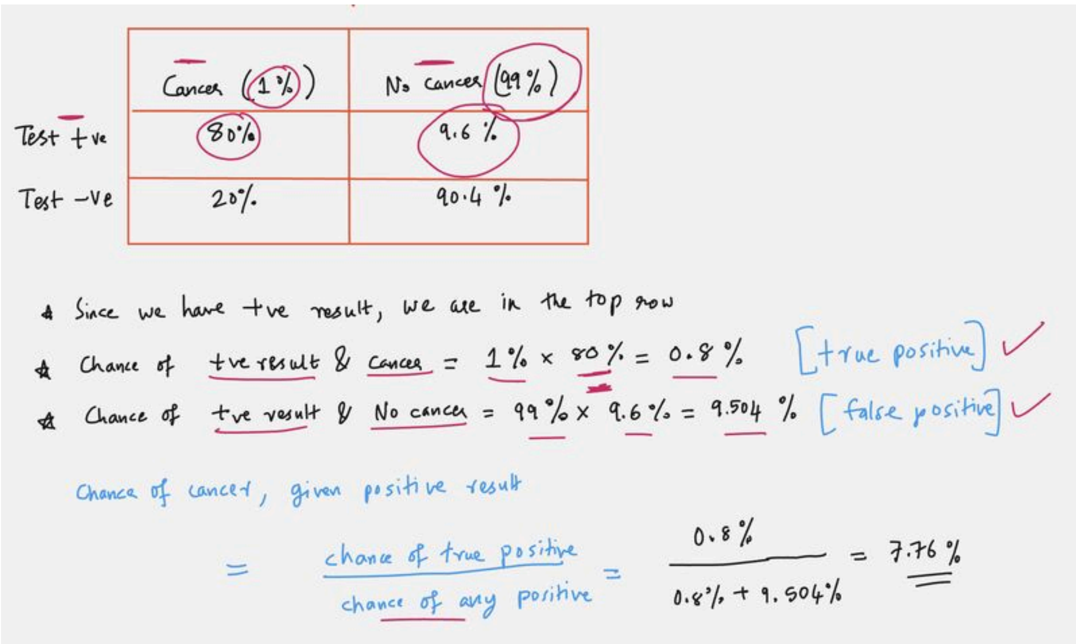
Imagine a mammogram test for breast cancer:
-1% of women in a population have breast cancer.
-The test detects cancer correctly 80% of the time.
-The test shows false positives 9.6% of the time.

If a woman gets a positive result, what is the probability she actually has cancer? While the test is 80% accurate, the probability of having cancer given a positive test is only 7.76%! Why? Because false positives (from the 99% who do not have cancer) heavily skew the results.

This is where Bayes' Theorem shines. It accounts for prior probabilities (how common cancer is) and adjusts for the test's imperfections. The equation transforms complex situations into actionable insights.

Beyond healthcare, Bayes' Theorem powers spam filters, recommendation systems, and machine learning models, shaping smarter decisions in uncertain scenarios.

Learn more from my lecture on Vizuara's YouTube channel: https://lnkd.in/g3wjcjvb



What Makes GPUs so Powerful? Matrix Multiplication!

GPUs have become one of the most essential (and pricey) pieces of hardware today. In my latest video on Vizuara's YouTube channel, I explore why GPUs are so powerful, what they do differently, and how they went from boosting gaming graphics to transforming AI: https://lnkd.in/gpgTJCRR

Behind GPUs' power lies their ability to perform massive matrix multiplications quickly and in parallel. Matrix multiplication is at the heart of 3D graphics rendering and AI model training.

In gaming, each 3D object is broken down into vertices and triangles, and every time the game's scene refreshes, the GPU has to rapidly recalculate positions, textures, and lighting using matrix math. A high-quality game renders millions of vertices and triangles. Without a GPU, gaming as we know it simply would not be possible.

Back in 2008, I remember trying to run GTA San Andreas on an old PC with no graphics card. I had to lower the resolution just to make the game playable. Meanwhile, friends with dedicated graphics cards were enjoying seamless high-res gameplay. That was my first real exposure to what GPUs could do.

At the time, NVIDIA was still mostly focused on enhancing gaming visuals. But the same matrix multiplication operations that GPUs used for graphics ended up being perfect for AI once deep learning emerged.

In fact, as AI researchers began training larger neural networks, they found that the same type of repetitive matrix math used for 3D scenes also applied to neural network computations. NVIDIA was able to transition smoothly from focusing on gaming to being at the forefront of AI hardware. When I first heard about NVIDIA's growing role in AI, I was unsure if it would really take off. Now, with their stock soaring and AI demand at an all-time high, it is clear that they found gold in an unexpected market.

In today's AI world, companies like Google and OpenAI are chasing advancements in AI like miners in a gold rush. NVIDIA, however, supplies the "shovels" – the GPUs that make this AI revolution possible.

These GPUs, priced from $10,000 to $40,000 for high-end models, perform the matrix multiplications and parallel computations needed for training AI models at scale. With only so many chips available and the demand rising, NVIDIA has quickly become one of the most valuable companies in tech.

For anyone curious about why GPUs are so vital to both gaming and AI, my video breaks it down. I cover the specific matrix operations and transformations GPUs handle, explaining why these devices are worth every penny for both gaming enthusiasts and AI researchers.

The journey of GPUs from niche gaming hardware to AI powerhouse is an inspiring story of innovation, adaptation, and surprising new applications. If you are interested in the deeper mechanics behind these breakthroughs, check out the full breakdown in my video: https://lnkd.in/gpgTJCRR

Author: Sreedath Panat

NotebookLM by Google is the 2nd best thing after ChatGPT

I have been testing Google's NotebookLM for the past few weeks, and I need to share this because it's genuinely changed how I work with my documents and research.

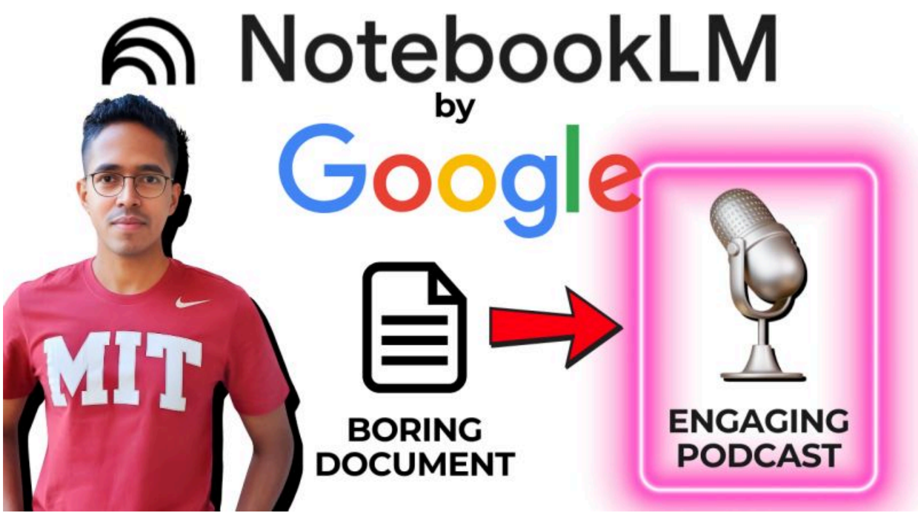
Check out this 10-minute video from Vizuara's YouTube channel: https://lnkd.in/gJwSsTtU

For those who haven't tried it yet - NotebookLM is Google's free AI tool that helps you analyze and draw insights from your own documents. What's really caught my attention is their latest update that lets you turn any document into a podcast-style summary. I tested it with some meeting notes yesterday, and the quality is surprisingly good.

A few things that make this stand out:

-You can upload practically anything (docs, PDFs, YouTube videos)
-It handles up to 50 sources per notebook with 500,000 words each
-Every insight comes with precise citations back to your source material
-It's now available globally, not just in the US

What I find most valuable is how it helps surface connections in your existing content. I've dumped years of meeting notes, reading highlights, and project documents into it, and it's fascinating to see patterns and insights I'd completely forgotten about.
You can check it out at https://lnkd.in/gt8wXqFk



Entrepreneurship

The single biggest decision to become an entrepreneur

The single biggest decision that you have to make if you want to start a company is to decide how seriously you want to do it.

You may wish to do it with some safety net. Like trying to work on some product ideas while managing a full-time job in another company.

When Raj, Rajat, and I decided to start Vizuara while doing our PhD, we thought at least it would become a resume point if nothing came out of our idea. But if we continued with that mindset, our idea would have inevitably become just another resume point.

Our thoughts and actions became very real from the moment we decided to not work on Vizuara as a side project. There is a huge difference between working on a startup with a plan B already in place v/s making your plan A work out.

When your idea starts affecting your income, quality of life, level of respect you get from others, where you live, what you eat, how you travel, etc., it becomes very real. Once you have a real idea, your startup dream is much more likely to happen.

Author: Dr. Sreedath Panat

Do not reject ideas

One of the best ways to test the validity of your startup idea is to build in public. You can quickly take your half-baked products to the market for feedback. You don't have to make a cascade of assumptions, build, and then take your product to market after 2 years of development only to realize that your assumptions were baseless.

The flip side is that depending on what kind of market you interact with, you may get the wrong signals. Your potential customers may show great interest in the beta version of your product. But when it is time to sign a deal, they won't be interested. Also, you will get a lot of unsolicited feedback from your non-customers.

If you have a good system that helps you identify when to take feedback and when to ignore it, you can make your idea public. If you are insecure or cannot withstand criticism, you are better off building in stealth.

Author: Dr. Sreedath Panat

Funding envy

I used to envy heavily funded startups for the wrong reasons. My envy was more about the glamour of fundraising than about what they could actually do with that money to become a sustainable business.

But after I truly realized that glamour wouldn't pay the bills, I think about profitability whenever I hear of funding. Once the Series-A funding hits the bank, and you hire 50 people, what exactly are you going to do to make your business a successful one?

If you really wish to convert your startup into a long-lasting brand, the metrics that matter are customer satisfaction, company culture, and profitability. There are many traditional businesses that no one has ever heard of, but are so profitable and can sustain for another generation if they just continue doing what they are doing. But it takes a lot of patience and time to build such companies. These days I envy such businesses a lot more because I see that their end-game is much longer, and they are playing by fundamental principles of business.

Author: Dr. Sreedath Panat

Funding does not equal success

I have closely seen funded startups struggling to pivot because the founders cannot convince the board. Not all board members are entrepreneurs themselves and do not know first-hand what it takes to build a company. Many board members and advisors are armchair analysts.

One of the many reasons why I prefer bootstrapping is agility. There is no need for pitch decks or convincing the board. You have full autonomy and you can take your startup wherever you want. You also don't have the luxury of money. So every decision you make is much more significant for the business.

Author: Dr. Sreedath Panat

Early v/s later days of entrepreneurship

Early days: Excitement, shortsighted- thinking only about the product, thinking about a billion dollars while having 0 revenue

Later days: Frustration, thoughts of pivot, thinking of million dollars while having very little revenue

With maturity: Humbled, no bullshit, thinking from the customer's POV and revenue projections grounded in reality

Author: Dr. Sreedath Panat

Customers over pitch decks

First-time founders focus on fancy pitch decks to impress investors and spend too much time on branding and technical details of their products. They make assumptions on top of assumptions without testing them in the market. They try to come up with cool products and complex features that the market won't care about.

As founders become experienced, they focus on customers first. They spend a minimum amount of time and money on the initial version of the product/features. They focus on testing their idea in the market rather than building on top of assumptions. They become better self-criticizers.

Author: Dr. Sreedath Panat

Bootstrapping is not for weak people

It needs great patience and perseverance to startup bootstrapped.

In the early stages, you will take a hit on your personal bank balance. It takes time to build any kind of revenue, let alone profit.

But if you are okay with the uncertainty of not having a fixed income, and if you are okay with being in survival mode, you can try bootstrapping.

Bootstrapping makes you feel in total control and if it works out, you have earned every piece of it.

Author: Dr. Sreedath Panat

Spend time with customers, not investors

The best way to know in the early days of your startup if you are going in the right direction or not is based on the proportion of time you spend with your potential customers.

-If you spend most of the time on product development, you might be making a lot of wrong assumptions about your customers' needs

- If you spend most of the time on branding, pitch decks, and your story, you may raise money from investors but not from customers

- If you spend a good amount of time with your potential customers, even if you may feel like you are wasting your time in the short run, you will ultimately save a lot of time and money

Author: Dr. Sreedath Panat

Force-fitting of the story

We have been part of a few startup accelerators in 2021-22 at MIT and elsewhere. It was a great experience to see how other entrepreneurs operate. The advantages are obvious. The problem I noticed was that there was always a need to force fit our idea into a certain narrative. The funding amount was directly proportional to how good the narrative was, rather than how good the actual product market fit (PMF) was, which no one can estimate at a very early stage.

The more time we spent in accelerators, the more we tried to force-fit our story/product to impress the mentors or potential investors rather than customers. The feelings of customers about our idea/product were nowhere in question.

There was a clear storytelling template startups followed. This template was very convenient for investors who have no time to see what the startups are doing in detail. But it did not help us to find a good product-market fit.

What really helped was spending as much time away from pitch decks and advisors as possible and instead spending more time with customers. Customers don't care about your pitch deck or story. They care only about your product or service. It is not advisors/mentors that help you identify PMF. Your customers are your best advisors when it comes to PMF.

Author: Dr. Sreedath Panat

Be mindful of the moment

In the early days of our startup, we had discussions about handling future traffic on our website if we were to scale. We were deeply looking at which tech stack to follow to minimize cost and latency before even having 1 dollar in revenue.

Looking back I can see that this is something many first-time founders fall into. Focusing too much on the tech and trying to solve all problems at the same time without focusing on the immediate issues.

Startups have different problems at different points of time. If you have 0 customers, focus on getting your first customer instead of building something scalable. If you have 10 customers, see what you can do to scale to 100, not to 1000 or 10,000.

If you don't know how to code to build your website, then use no-code platforms. Don't be frugal when taking useful subscriptions that save you time. I remember spending a week coding to figure out how to make a video-call feature using WebRTC work from scratch to save money. Looking back I realize that I should have simply used Zoom or Twilio API instead of reinventing the wheel.

Identify what is most valuable to do in each moment and execute it very well. This is what it means to run a startup.

Author: Dr. Sreedath Panat

The most difficult thing to do in entrepreneurship

The most difficult thing to do in entrepreneurship is to find the right people to work with.

The number of factors that have to align for a group of people to work in cohesion on a singular, long-term vision is uncountable and depends on your luck.

But there is one factor that depends fully on you: your personality. You can mould yourself into the person whom others find good to work with, and you may attract some great people to work with you.

Author: Dr. Sreedath Panat

Build in public

One of the best ways to test the validity of your startup idea is to build in public. You can quickly take your half-baked products to the market for feedback. You don't have to make a cascade of assumptions, build, and then take your product to market after 2 years of development only to realize that your assumptions were baseless.

The flip side is that depending on what kind of market you interact with, you may get the wrong signals. Your potential customers may show great interest in the beta version of your product. But when it is time to sign a deal, they won't be interested. Also, you will get a lot of unsolicited feedback from your non-customers.

If you have a good system that helps you identify when to take feedback and when to ignore it, you can make your idea public. If you are insecure or cannot withstand criticism, you are better off building in stealth.

Author: Dr. Sreedath Panat

TAM? Don’t care

Thinking about the Total Addressable Market (TAM) would be the worst thing you can do while starting up. If someone advises me to not try something because TAM is too small I won't listen.

Showing two bubbles: one with a 10 billion dollar TAM and another with a 100 million dollar beachhead market has become too common in startup pitch decks. There is always a need to say that your TAM is in billions of dollars. Otherwise, if you are looking for funding, you are doomed.

This overemphasis on TAM prevents many founders from looking for small opportunities right in front of them. You can start small, and grow slowly and profitably. You can build a long-lasting brand. None of this has anything to do with thinking about TAM on day 1.

Author: Dr. Sreedath Panat

Where are the right people?

The most difficult thing to do in entrepreneurship is to find the right people to work with.

The number of factors that have to align for a group of people to work in cohesion on a singular, long-term vision is uncountable and depends on your luck.

But there is one factor that depends fully on you: your personality. You can mould yourself into the person whom others find good to work with, and you may attract some great people to work with you.

Author: Dr. Sreedath Panat

Philosophy

Leaving USA for India!

At the end of 2022, I defended my PhD thesis at MIT. It was one of the most satisfying days of my life that happened after 5 years of blood and sweat.

Nearing the end of my PhD, I was constantly thinking about what to do next. There were several lucrative options.

I had already worked on a few groundbreaking projects with an unbelievable team. We made truly fundamental advances while developing technologies for making photovoltaics, oil-water separation, carbon dioxide capture, and agricultural sprays more efficient and sustainable. I helped raise Rs. 5 crores in pitch competitions for one of these projects to spin off a company. My work on photovoltaics was also getting great attention from investors. On top of these, I was getting messages from recruiters to interview for jobs that paid up to 1.5 crores per year. Any of these options would have been the best start for my American dream. However, something did not feel right.

I used to take regular walks to a Belgian waffle place near Harvard with my friends. We used to discuss various ideas about returning to India and starting our own ventures. One day, I realized that I was only talking and not taking any action, just like lakhs of Indians who come to the US and always talk about returning one day. That is when I decided to book a one-way ticket to India.

Now I am back in India, following my burning desire. I left the US so soon after my defense that I had only 2 days to pack all of my stuff. There are a few reasons why I took this step, which is considered bold by my friends and family.

1) I want to experience India's rapid growth in the next 30 years, which will fortunately coincide with my career years. I believe India's GDP can potentially grow 10-15x in this time, and I want to be at the forefront of this growth.

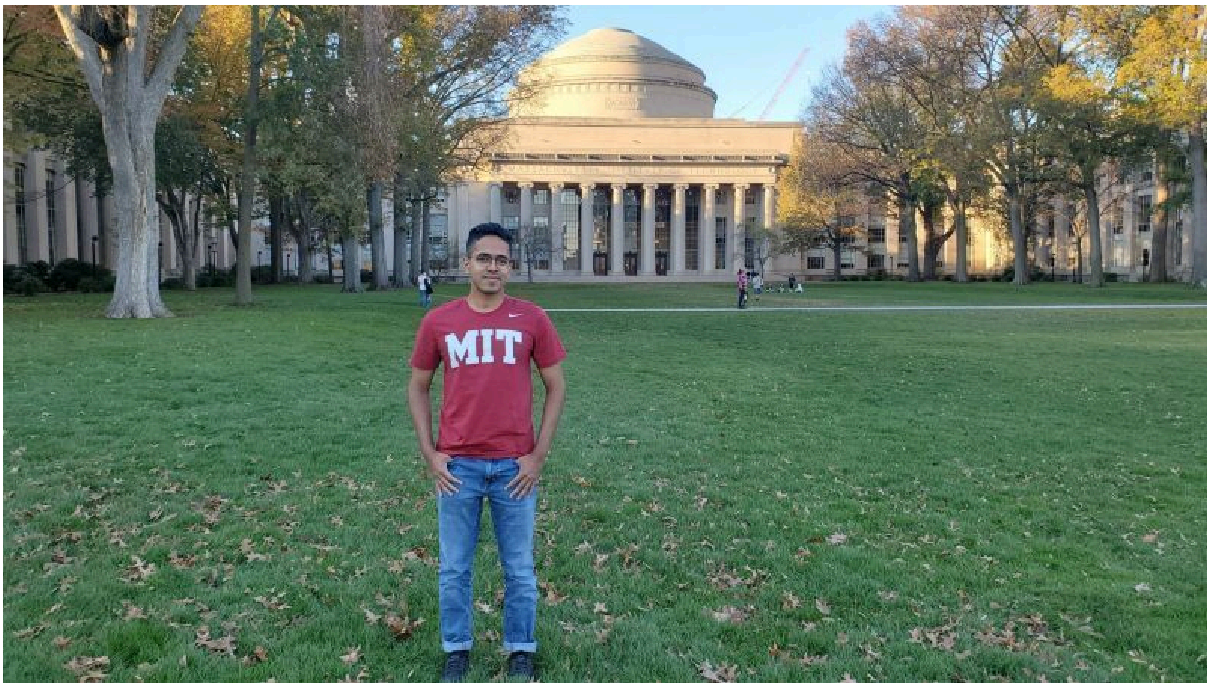
2) I wish to create 50,000 jobs in India by the time I am 60. I want to build a global brand that exports high-quality products & services from India- a very ambitious dream. But this dream gives my efforts in India immense meaning and purpose.

3) I have already co-founded 2 startups in India with my friends Raj Abhijit Dandekar and Rajat Dandekar who convinced each other to leave USA.

4) I want to build a community of entrepreneurs in India who want to address socio-economic and technological challenges here. I am constantly expanding my network. If you are a co-founder, feel free to connect. I am currently based in Bangalore. I also travel frequently to Kerala, Delhi, and Pune. If you are around and want to meet up, feel free to get in touch!

Finally, I relate to India more. I have a great sense of belonging here, which I never felt in the USA, even though I was surrounded by some great people. I also realized that I want to spend more time with my family and friends. I don't regret coming back for a single day!

Author: Dr. Sreedath Panat



My first-day v/s last day at MIT

5 years of research at MIT transformed me in ways I could not have imagined. I am writing this piece to reflect on the changes I experienced from my first day to my last day at MIT.

First day: Wanted to become a professor and work in academia
Last day: Left the idea of academia. Wanted to become an entrepreneur.

First day: Wanted to stay in the US to experience the American dream
Last day: Wanted to return to India to build my Indian dream

First day: Felt intimidated to see super smart students from China, Korea, the US etc.
Last day: Did not feel intimidated by analytically smart people anymore. Realized smartness has a spectrum.

First day: Mindset- Wanted to compete with others and win
Last day: Mindset- Wanted to create my own game and play that

First day: Poor at communication. Placed all importance on analytical skills.
Last day: Good at communication. Much better at analytical stuff.

First day: Wanted to work alone, just focusing on what I did
Last day: Wanted to work with a team of people on a mission

First day: I thought I could out-compete anyone and achieve anything
Last day: Realized it is stupid to think of life as a competition and I can't achieve everything

First day: Thought science in academia was “pure”
Last day: Realized that all shortcomings of people reflect in the process of conducting science. Nothing is “pure” when humans are involved.

First day: Big-time stage fright and under-confidence
Last day: Absolutely no stage fright and supreme level of confidence

First day: Felt very intimidated when my friend said he wanted to build a 100-million dollar business. Thought I would never dream that big.
Last day: Dreaming much bigger things more realistically

First day: Thought academia is only about solving research problems without thinking about anything else
Last day: Realized academia (and everything else humans do) involves understanding and playing the

game of the system

First day: Was judgmental about people based on what they did
Last day: Stopped judging people based on their work. Realized everyone is just trying to live.

First day: Respected people with brands and tags.
Last day: No inherent respect toward brands and tags. Respected people for being good human beings.

I don't intend to attribute all of these changes to MIT. But a good fraction of my change is attributed to the people and my experiences there. Rest was just the process of normal growing up. Either way, I feel like a different person.

Author: Dr. Sreedath Panat

Reverse cultural shock after moving from the US to India

When I returned to India to build Vizuara with Raj and Rajat, my cultural view was heavily based on how people interacted in the US.

When I scheduled meetings with school principals and others to discuss business, often people would not show up on time. People sounded rude. Some people called to discuss business late during the weekend hours.

Initially, I used to get offended because I was not used to any of this during my 5 years in the US. I was in the cocoon of MIT. I was thinking- "do I really want to work with some of these people if they are not being courteous and respectful of my boundaries?".

With experience, I realized that I cannot change the way people think or talk. They are used to a certain way of doing things and that is not going to change.

I should not measure the people in India using a scale that I brought home from the US. I realized a need to recalibrate my cultural metrics.

If the owner of a chain of schools expects to be called sir/madam because that is how things work in India, I have to adapt to that even if I may not like it or believe in it.

It is actually uncomfortable for me to call someone sir/madam just because they are my customers. But it is also weird for many of them to be not addressed as sir/madam. If I value my feelings more than the requirements of the business, I can choose not to do business with them. If I value their business, I should keep my feelings aside.

Ultimately business is about figuring out how people think, and what they want and giving them that in exchange for value.

Now I truly understand the meaning of "think globally and act locally". Yes, I can dream of building a global company. But as far as business goes, I should act according to what works in the local region.

Author: Dr. Sreedath Panat

Don't take right decisions. Take decisions and then make them right.

I have been in Delhi for the past 2 days and the pollution levels are crazy.

The air quality index is 400+ and comes under "hazardous".

After getting down from the plane, I heard a discussion between 2 Indian professionals which went like:

"Why do people even stay here? Boston is 10 times cleaner than this. Better to be in the USA than stay here"

After this, I was thinking:

"What drives people like me back home?"

I found 5 major reasons:

1️⃣A bit crazy:

- I had defended a PhD at MIT: the #1 engineering college in the world.
- I got job offers of 1 crore+ from leading US companies.
- Many people thought "Why is he coming back? Is he crazy?
- You need a bit of crazy to go against the conventional wisdom and return back.

2️⃣Passion for building something India specific:

- If you want to build an AI education company revolving around the Indian education system, you cannot do it sitting in the USA.
- If you want to build healthcare AI assistants in low resource rural Indian villages, you cannot do it sitting it in the USA.
- If you want to become a professor at one of the IITs and give back, you cannot do it sitting it in the USA.

3️⃣Not tied down by commitments:

- One of the major reasons people stay back in the USA and never come back is if they start a family.
- If you start a family, rent a house or put your kids to school in the USA - there is too much sunk cost. It becomes very difficult to come back.
- I moved back at a stage when I was not tied down by any commitments. It makes the decision much easier.

4 Taking a bet on India’s future:

- I am bullish on India’s growth story and I want to be part of it.
- I truly believe that I can contribute hugely to India’s growth story if I am back in India.

5 Family:

- If you want to be close to family and that is one of the top priorities, you will surely come back to India.

I came back home due to majority of the above reasons.

Then what about the issues? What about pollution?

The answer to this is “niches”.

If you achieve what you want in India, you will always find niches which work for you.

Don’t like pollution in Delhi -> move to Pune or move to Chennai.

Don’t like unclean cities -> move to Indore or Visakhapatnam.

Don’t like overpopulation and staying in crowded areas -> work-from-home in Kerala or Darjeeling.

If you truly want to make something work, India offers enough “niches” you can take advantage of.

P.S: In this photo, you can see me at Hauz Khas Village, Delhi. One of the most beautiful parts of Delhi.

Author: Dr. Raj Dandekar

Building relationships with people

One of the best ways to build relationships with people is to not reject their thoughts.

People can think. But they may fail to articulate their thoughts when they communicate ideas with you. Maybe they are tired. Maybe they are not getting the right choice of words at that moment. Maybe they have not fully formed their thought yet, but they are just thinking aloud.

If I know someone personally and if I know that they can think logically, I usually try not to dismiss their idea even if I immediately don't see the merit in it. Instead, I assume that either I am not getting what they are saying or they are not articulating it well enough.

I try to spend more time with them discussing what they are thinking. This helps in two ways. One, even if you may ultimately disagree they will feel respected because you were not outright dismissive. Two, your discussion can help them convey better. And you may end up agreeing with their idea.

Author: Dr. Sreedath Panat

Talking about our work in public

A simple routine that exponentially increased the opportunities I receive is talking about my work in public.

There was a time when I was too shy to talk about what I did. I had an extreme fear of judgment. I have overcome this fear to a great extent by conscious effort.

If you wish to receive opportunities, all you need to do is to show the world how your brain works. Talk about, write about, or share your work. Let people see what you do and let them judge your work.

If you have a great thought process and good intentions, people will recognize that and will want to work with you.

A few people will definitely critique your work, irrespective of what you do. If you can learn to ignore that you open a whole new space of opportunities.

Author: Dr. Sreedath Panat

Smartness is overrated!

During my school days and undergrad at IIT Madras, I believed that smartness was the way to success. I envied super smart folks who could solve complex problems in their head. I wished I could grasp hard concepts like some of the really smart students in the class.

As years passed, I have come to the firm realization that it does not matter how smart you are. The only thing that matters is consistent hard work. The world does not reward smart people. The world rewards people who get useful things done. In most cases I have seen, the only skill you need to do things is perseverance.

Many people I know who come from very humble backgrounds and are not analytical smart have gone on to do great things in life. So if you find yourself surrounded by people you believe are smarter than you, don’t worry. Work hard and stay focused. Time will reward you.

Author: Dr. Sreedath Panat

Don’t be shy

A lot of folks wish to build an online presence and promote themselves but are afraid/shy to do so. This is mainly because they worry about how people will perceive them, especially colleagues.

If you think people may perceive you as…

1) Arrogant

2) Boastful

3) Stupid

4) Wrong

5) Or silly

…based on what you present online, then here is a thought process that may help you.

[1] No one is going to promote you. You have to do it for yourself. Your work cannot speak for itself. You have to do the talking.

[2] Nobody cares.

You can think of your online presence in a few different stages.

Stage 1: Personal accomplishments:

This is the bare minimum you can do. If you get a job/award or get selected for a fellowship/conference, share it with people. No one will judge you for sharing your accomplishments.

Stage 2: Neutral topics:

These are things you can share with people, where they have no chance to disagree with you or judge you. Examples: if you found a very useful GitHub repo or research paper, share it. Or if there is a good YouTube video explaining a relevant topic, you can share it.

Stage 3: Mild opinions:

This is when you enter an area where there is a chance some people may disagree with you. For example, if you talk about the challenges of implementing self-driving cars in India, most people will agree with you, but some may challenge your thoughts.

Stage 4: Strong opinions:

This is explosive content. Strong opinions always invite strong criticism. If you are entering this area, you have to be comfortable with a lot of people disagreeing with you. But the good thing is that your strong opinions also attract people who strongly agree with you. They will really love to follow you because they understand how your brain works.

The moment you are comfortable sharing strong opinions, you have broken the wall to build your personal brand. At this point, you will find it very comfortable to share your thoughts and show how much you are proud of your accomplishments. This is where you truly promote yourself.

When people disagree with you, they are mostly respectful. And you should be okay with that. But some people may make personal statements/hate speech/toxic comments. You can either ignore them or block them.

Ultimately, nobody deeply cares. But if you build a strong personal brand, even if you are working in a job, it will help you in the long run. Think of a decade timeline here.

Feel free to share your thoughts.

Author: Dr. Sreedath Panat

A tale from Chinese folklore

Once upon a time, there was a young man who saw an elderly man carrying a heavy load on his back.

The load was his horse.

Curious, the young man asked, "Why do you carry your horse instead of riding it?"

The old man replied, "When this horse was a baby, I carried it around because it was much lighter then. But as the horse grew, I continued to carry it out of habit, and now, even though it is fully grown, I still find myself carrying it."

There are many ways in which we can interpret this story.

One interpretation that I like is the power of habit.

It is impossible to carry a full-grown horse in one day.

But if you can carry a baby horse yesterday, you can carry it tomorrow. If you can carry it tomorrow, you can carry it the day after tomorrow. If you repeat this every day, although the baby horse will grow a bit every day to become a full-grown horse, you will be able to carry it.

More than the sensibility of the story, I am obsessed with the idea that we can condition ourselves to do the seemingly impossible, with the power of habit.

1) If you don’t know how to start a business, start slowly on the side. Make a small product/service and iterate every day.

2) If you don’t know Machine Learning because you are from mechanical engineering, join a good course and show up every day.

3) If you are not good at coding, code a few lines every day. Start will “Hello world” and improve every day.

4) If you are not good at writing, try writing for 30 minutes every day.

5) If you are not good at math, pick up your books and try math every day.

6) If you don’t know how to sell something, pick up your phone and cold call your customers

We can pretty much learn anything if we show up every day.

Author: Dr. Sreedath Panat

Sense of responsibility

The number one trait that I have seen in successful people is an extreme sense of responsibility. They do not complain or cry about how their life is not going as planned. They do not live in the glory of their past success. They do not apologize too much. If things work out, then they plan the next steps. If things do not work out they find ways to fix it. People who take ownership make your life easy. People who don't take ownership make everything hard.

Author: Dr. Sreedath Panat

Networking is not about exchanging visiting cards

My amateur idea of networking was exchanging contacts-either phone numbers or visiting cards after the first conversation.

In the last few years, I have attended dozens of conferences, networking events, coffee meet-ups, and countless one-to-one Zoom calls.

Most of my interactions were touch and go. We will never see or speak with that person again. This is a bad way of networking.

Later I realized that if I have to increase the quality of networking, there has to be some kind of action item involved. 99% of my meetings without a defined agenda have led to nothing.

On the other hand, if I make it a point to discuss how we can mutually benefit from working together or simply being in touch with each other, and then also discuss what exactly to do next, there is a likelihood of a much more meaningful and long-term connection.

I have taken 100s of visiting cards from people and have given away 500+ visiting cards of mine at big events. Nothing has ever happened beyond that. For me visiting card exchange has become a formality that has no meaning.

These days, instead of exchanging visiting cards, I explicitly talk about in what capacity we can collaborate and how can we both benefit. I explicitly discuss what exactly we can do as the next steps. This approach has led to a stronger and more useful professional network.

Author: Dr. Sreedath Panat

Boring routine will make you outcompete 99%

From my experience the mantra to get ahead of most people is very simple: consistency.

But consistency is hard. It is also very boring.

You will be doing the same thing day after day and week after week. Fewer changes and more repetition.

You may or may not beat the top 1% of the folks by this approach. But you will beat 99%.

Your 9 am will look the same on weekdays or even 7 days a week. You may be eating the same food every day. Your days will not be dynamic. You will probably not be having an impromptu trip. But you are definitely getting ahead.

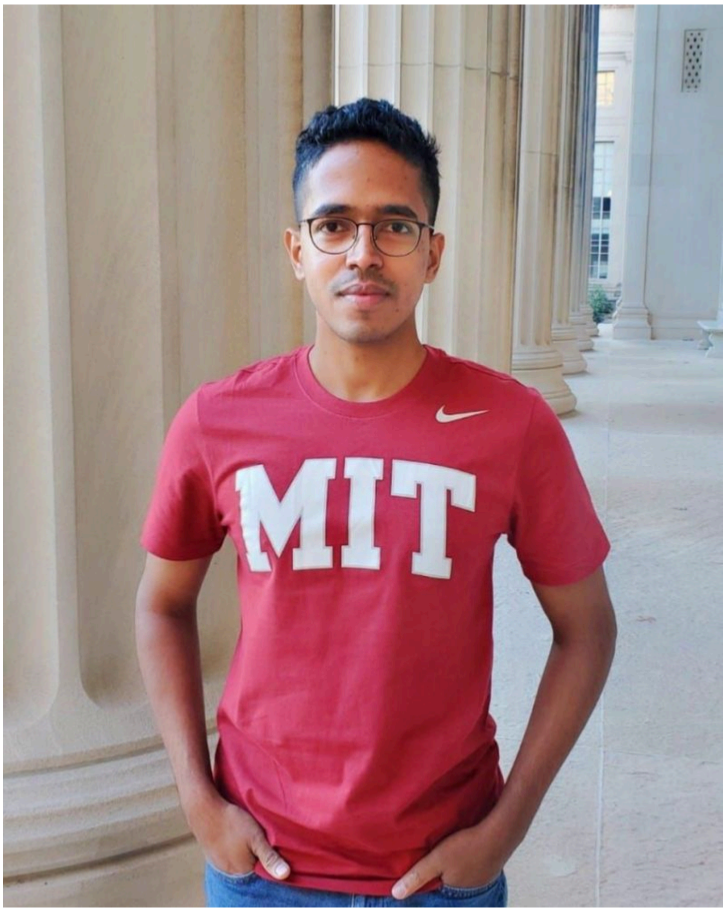
Is the price worth it in the end? It depends on whom you ask.

To some, the boring routine is liberating. It makes you efficient at what you do and leaves a lot of mental space.

When you have a boring routine, you do not depend on your feelings. You get stuff done whether you feel like it or not.

When you do what you need to do, you get ahead of those who do what they feel like doing.

Author: Dr. Sreedath Panat



Physical stamina is the key

The single biggest ability that helps people to get more done is not intelligence, curiosity, or willingness to work hard. It is physical stamina.

When I was 15 years old, I could sit for very long hours with great focus, especially during the preparation of competitive exams like JEE. It was very easy to work 15-16 hours a day. Now I can't imagine that. Not because I have more responsibilities, which I do have. But more because of my reduced physical stamina.

I don't think my mind has become any less curious. In fact, I think my curiosity is at its peak now and my willingness to work hard is also at its peak. But I simply cannot work for as long as I used to because my body refuses.

I work out 4-5 days a week. I sleep 7-8 hours a day and meditate 30-45 minutes a day. Yet on a few days, I feel too physically tired to do work.

My eyes sometimes hurt due to the screen time. Sometimes headache; or body soreness from sitting.

I have seen some of my colleagues who can manage to work incredibly long hours and stay focused. I totally envy them.

Somehow, they have enough physical energy to sit through long meetings.

If I have 3-4 meetings in a day, my physical energy goes down the drain (especially if they are virtual). If I have a 5th meeting, I am as good as a potato.

I totally believe that hours of work do not correlate with productivity. But sometimes my per-hour productivity is so high that I simply wish I could work for more hours to get more done.

I believe physical activities will pay long-term dividends in keeping our physical stamina higher than average people.

I have noticed that when I sleep well, meditate in the morning, do a bit of yoga, my body can generally keep up with the increased physical demands of the work.

Author: Dr. Sreedath Panat

The number of people who are sincerely rooting for you

The number of people who are sincerely rooting for you to succeed can be counted on your hands

Author: Dr. Sreedath Panat

Meditation, cooking...

Meditation, cooking, investing, carpentry, AI, coding, nutrition, basic mechanical & electrical repairing, sales. These are a few skills I wish were taught hands-on in schools.

Author: Dr. Sreedath Panat

Human-to-human communication

I think human-to-human communication is the hardest thing in the world, more than building any technology.

Beautiful and effective communication can make seemingly impossible things happen.

Poor communication can even result in wars.

Author: Dr. Sreedath Panat

The ability to ask

The ability to ask is the most underrated skill. We have no idea of the magnitude of opportunities we missed out on in life because of our inhibition to ask.

Author: Dr. Sreedath Panat

Perception of time

One of the things I don't like about growing up is the perception of time. I feel like time is going much faster as life goes ahead. The 10 years from 1st grade to 10th grade in school felt like an entire lifetime. Whereas the past 10 years just feel like yesterday.

Author: Dr. Sreedath Panat

I do not enjoy 80% of my work

I truly enjoy only 20% of my work and lack the motivation to do the rest because they are boring tasks. But boring tasks are often unavoidable.

This was true during my JEE preparation, undergrad at IITM, PhD at MIT, and while running startups now.

I have realized that motivation does not work for me. I am demotivated to do a major chunk of my daily work. Yet I manage to get the uninteresting, unscalable, and boring stuff done.

The only thing that works for me is discipline.

I am happy when I am disciplined.

So I fit boring and difficult things into my daily routine. If those things are not done, I deviate from my routine. This deviation makes me unhappy.

Motivation from YouTube videos, quotes, and inspirational figures are all short-lived. I feel fired up for 5 minutes. That’s it. It simply does not work beyond that.

I want to be good at getting boring stuff done fast and efficiently. For that, I have to make it part of my life. That is why I personally like being disciplined more than being motivated.

Motivation is a feeling. Discipline is a habit.

Feelings are short-lived. Habits can continue for a lifetime.

Author: Dr. Sreedath Panat

Feelings over facts

I identify the right thing to do mostly based on feelings. It is not based on numbers or metrics. I often know internally if I am doing the right thing and moving in a good direction, even if numbers do not show that. And sometimes even when the numbers are good, I won't continue to do something because I do not feel like it.

It is very interesting how the mind works. Even with a great amount of data, rationality, and the knowledge that we should not rely on feelings, ultimately many decisions boil down to feelings.

Author: Dr. Sreedath Panat

Execute them

People who execute simple actions consistently get far ahead of people who always think of big ideas, but never execute

Author: Dr. Sreedath Panat

Don't be a crowd

The moment you stop acting and thinking like a crowd, and the moment you start taking ownership, you will experience a noticeable change in your life.

Author: Dr. Sreedath Panat

Do not dismiss ideas

People who are outright dismissive of ideas can sound smart because they may be good at armchair analysis.

But people who can envision how to make an idea work-out despite challenges will manage to get stuff done and will win eventually.

Author: Dr. Sreedath Panat

Shortcuts do not work in life

The more experiences I gain, the more I truly realize that shortcuts do not work in life. They may seem to work in the short term, but eventually, only good work done with good intentions succeeds.

Author: Dr. Sreedath Panat

Meditation is the best habit...

Meditation is the best habit I have cultivated in the last few months. Spending 30-45 minutes awake, sitting in one place, without moving, and trying to be mindful is wonderful for the mind.

Author: Dr. Sreedath Panat

Perfectionism is usually a strength. But..

Perfectionism is usually a strength. But it can also become a weakness that slows you down. Prioritization is a greater strength on many occasions.

Author: Dr. Sreedath Panat

Dreaming big and expecting small: The philosophy that got me into MIT

To achieve my goals, I nurture a pessimistic view that may not work for most people. Whenever I dream of achieving something, I highly entertain the possibility of that dream not becoming a reality in my mind.

I have talked to many friends about this approach and many of them truly dislike this because it demotivates them. I completely get that.

However, surprisingly, this philosophy works very well for me. Somehow, I am still very motivated to take action to make the dream that I truly believe is unrealistic, into a reality.

In 2017, during university applications, I gave myself a 20% chance of getting into MIT for a PhD. Because of this, I was totally okay (very honestly) with not getting admitted into MIT.

I knew that I had a good profile and I would get into some decent university. I was prepared to make any outcome workout for me, including getting admitted into my least preferred university.

Because of this approach, I was really stress-free during applications, and it had a huge positive impact I believe. I knew many of my friends who were super stressed about the applications because of their high expectations. My low expectations kept me very calm.

-I was not too worried about getting my SOP reviewed by 15 seniors.

-I was not too worried about what my professors would write in their LOR.

-I was not too worried if my GRE and TOEFL scores would reach the university.

I was too carefree to the point where I submitted my MIT application a few hours before the deadline. I reminded one of my Professors, who had forgotten to upload the LOR for MIT two weeks after the deadline.

I am just writing this to share my approach that worked for me, but not for everyone. It is probably okay if you are pessimistic. Things might still work out for you.

Author: Dr. Sreedath Panat

Returning to India right after your PhD is not a smart move.

You must work here in the USA for a few years, build your bank balance, get a green card, and then you can plan your move to India. There is a time and place to take risks". This was told to me by a very close person.

It is difficult to analyze such advice, especially if it is coming from a person you respect. If you hear this same advice from 4 or 5 people, you may not make any kind of bold decision. This is happening to lakhs of Indians currently living in the US.

I am glad I did not take this advice. I left the USA 2 days after defending my PhD at MIT.

These days my mechanism of evaluating unsolicited advice is simple.

If it is generic advice, I will ignore it.
If it is personal context-specific advice I will analyze it and may or may not take the advice.

At the end of the day, I will treat unsolicited advice as a distraction that I have to be mindful about. I won't let it influence me.

Author: Dr. Sreedath Panat

The single biggest skill that will make a huge impact...

The single biggest skill that will make a huge impact in your life is your ability to storytell. Irrespective of your profession.

Everyone who is at the top of their game are exceptional story tellers.

The good thing about storytelling is that it can be learned. Be conscious about how you portray your work, iterate, and make your story better.

Author: Dr. Sreedath Panat

Collaboration > Competition

In the long run, being collaborative with your peers is much more valuable than being competitive

Author: Dr. Sreedath Panat

I do not follow my passion

Always following our passion would be a beautiful thing in life. But what if what you are passionate about and what you are good at are two different things?

What if you are passionate about cricket but instead you are good at studies?

What if music is your passion, but you are rather good at writing code?

This was true for me most of my life.

- I have always been a studious student. I studied very hard and had good grades. However, I was not always passionate about what I was studying. Many of the subjects I learned were for the sake of marks. I was more afraid of scoring badly in exams than I was passionate about those subjects.

- In PhD, some of the projects I worked on were not based on passion towards that research topic, but it was more based on opportunity. I could identify a clear gap in the literature and a great, innovative way to address that problem. So I decided to work on that. Was that problem statement my true passion? I don't think so. Was I good at solving it? Yes. However, some projects were truly my passion. That is what kept me going in my research.

- Even while running a startup, most of the decisions we make are based on business opportunities rather than just passion. It would be great to work on things we love for 8 hours a day. But in reality, maybe I get to do what I truly love 20% of the time and the rest of the work is something that I have to get done with or without passion.

Most of the people who work in jobs do that not because the work excites them, but because it helps them pay their bills. And I think that is totally okay. I truly respect the people who work hard to provide for their families.

I have contemplated this question a lot. What am I truly passionate about? The only thing I could identify was that I am truly passionate about bettering my life. All the actions that I take are attempts towards this. The only thing I want to ensure is that those actions should not hurt anyone and if possible provide happiness to others.

Author: Dr. Sreedath Panat

Unsubscribing news channels

One small change that significantly improved my quality of the day is actively unsubscribing news channels. Most of the news that used to show up on my feed had a negative impact on my mood. It was difficult to unsubscribe news due to fear of missing out. What if I don't get to hear about an important thing/event? But now I realize that I am not missing out on anything that is useful for me. In the last 5-6 years I haven't read any newspaper and I haven't actively followed any news channels. Yet I don't feel dumb or socially retarded. Somehow I get to know all the major news and my feed and day are better.

Author: Dr. Sreedath Panat