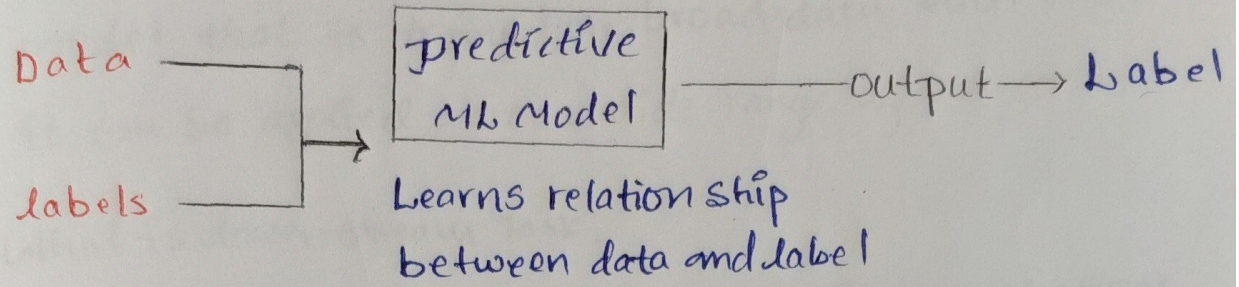
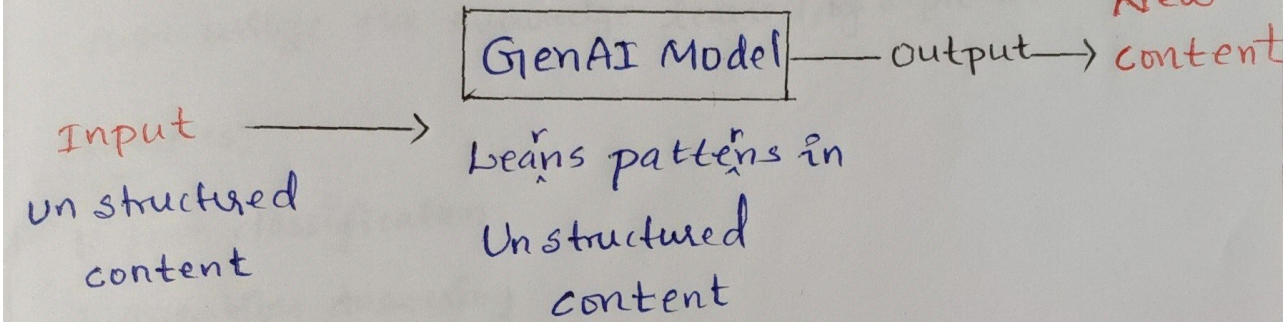
Generative AI

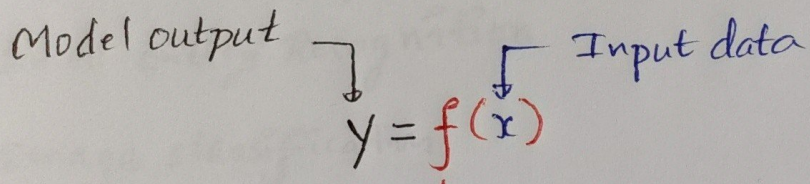
Date:- 30-06-2024

Def" - GenAI is a type of Artificial Intelligence that creates new content based on what it has learned from existing content.

How to Understand GenAI?







Not GenAI

When y is a:

- Number
- Discrete
- Class
- probability

Is GenAI

When y is:

- Natural language
- Image
- Audio
- Video

What is fine-tuning?

Making small adjustments to foundation model in order to achieve the best performance in a down-stream task.

What is foundation model:

Is also known as large AI model, is a ML or DL model that is trained on broad data such that it can be applied across wide range of use-cases.

What is down-stream Task:

Downstream tasks in AI refer to specific applications that utilize the knowledge learned by a pre-trained model

Examples:

1. Text classification
2. Question Answering
3. Named Entity Recognition
4. Image classification
5. object detection
6. Medical Image Analysis
7. Fraud Detection
8. Algorithmic Trading

- If we update all the model parameters (Re-train) during fine tuning then it is called full fine tuning

parameter Efficient Fine-Tuning (PEFT)

Why we need PEFT:-

The major downside of fine tuning is that the new model contains as many parameters as in a original model.

- As larger models contains ~~as many~~ trained every few months, this changes from a mere "inconvience" for GPT-2 to a critical deplyment challenge for GPT-3 with 175 billion paramet

rs

Low Rank Adaptation (LORA):

- LoRA is one of the most used fine-tuning technique in the umbrella of PEFT

AREN'T EXISTING SOLUTIONS GOOD ENOUGH?

- There are two prominent strategies when it comes to efficient adaptations:

1. Adding Adapter layers.

2. optimizing some forms of the input layer activations

1. Adapter Layers Introduce Inference Latency:

There are many variants of adapters. Adapter layers are those which we fine tune/train for the down stream task. add to

- There are no direct ways to bypass extra compute in adapter layers. Since we increased no.of parameters by adding the Extra layes Inference latency is increased.

2. Directly optimizing the prompt is Hard:

Observed that prefix tuning is difficult to optimi -ze and that its performance changes non-monotonicall -y in trainable parameters.

prefix tuning: It is nothing but adding a prefix before the prompt.

before.

Eg: prompt: "Generate a smart phone product description".

prompt_prefix: [positive tone] generate a product description for a smartphone".

OUR METHOD: (LoRA)

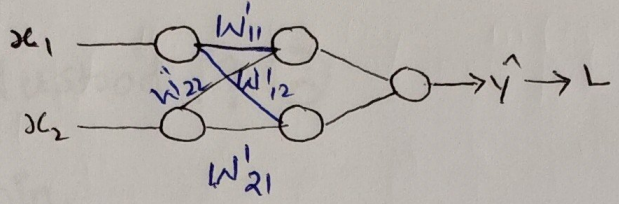
First, what is Rank?

The no of independent rows or columns in a matrix.

- Do, explore the examples ☺

* We know that weights in a specific layer of Neural Network can be written in matrix form.

Example:



$$W' = \begin{bmatrix} W'_{11} & W'_{21} \\ W'_{12} & W_{22} \end{bmatrix}_{2\times 2}$$

* When we are adapting to a specific task, we came to know that the pre-trained language models have a low "Intrisic dimension" (search for research paper, you will get it on intrisic dimension).

Wait! Wait! Wait! what is Intrisic dimension Jayanth?

An objective function's intrinsic dimension measures the minimum number of parameters (weights + biases) needed to reach satisfactory solutions for respective objective

(or)

The Intrinsic dimension represents the lowest dimensional subspace in which one can optimize the original objective function to within a certain level of approximation error.

Not Understood !? ☹

Let me explain,

Let's take GPT 3, which has 175 Billion parameters

Let $\theta^D = [W_1, W_2 \cdots W_{175\ Billion}]$

↑

Consider the subspace of foundational model parameters

Let $\theta^d \subseteq \theta^D$ $[d \leq D]$

Basically think θ^d is the projection of θ^D

Let $\theta^d = [W_1, W_2 \cdots W_{1\ Million}]$

think θ^d is the basis of θ^D

There will be somany basis for θ^D, which one I have to select Jayanth ?

Well,

The authors are saying that:

We have to select θ^d such that it gives 90% of full-fine tuning accuracy.

Then they are calling this θ^d as Intrinsic dimensi

-on

Now,

Taking the inspiration from Intrinsic dimension the LORA authors thought there would be Intrinsic Rank as well for LLM's.

* Author's hypothesized the updates to weights also have a low "Intrinsic rank" during adaptation

Let $W_0 \in \mathbb{R}^{d \times k}$ is the pre trained weight matrix (foundation model) of a layer

Let $\Delta W = B \cdot A$, where
$d \times k \quad d \times r \quad r \times k$

'B' is installized with "zeros" &

'A' is Random "Gaussian Intialization"

Now Instead of training W_0 (frozen) we train A and B matrices for the down-stream task. low-Rank decomposition

Then we Multiply them to get ΔW, since ΔW is same dimension as W_0. We add them
$d \times K$ $d \times K$

$$h = xW_0 + \Delta W x = W_0 x + BAx$$

↑ Input (embedding prompt)

<show figure>

Jayanth, we apply LoRA on a foundation model right but your apply on a layer?

yes, when we apply LoRA on a foundation model like llama. It means we are applying LoRA to each layer of it!

Remember,

Before we add $\Delta W x$ with $W_0 x$ we scale it with $\frac{\alpha}{r}$.

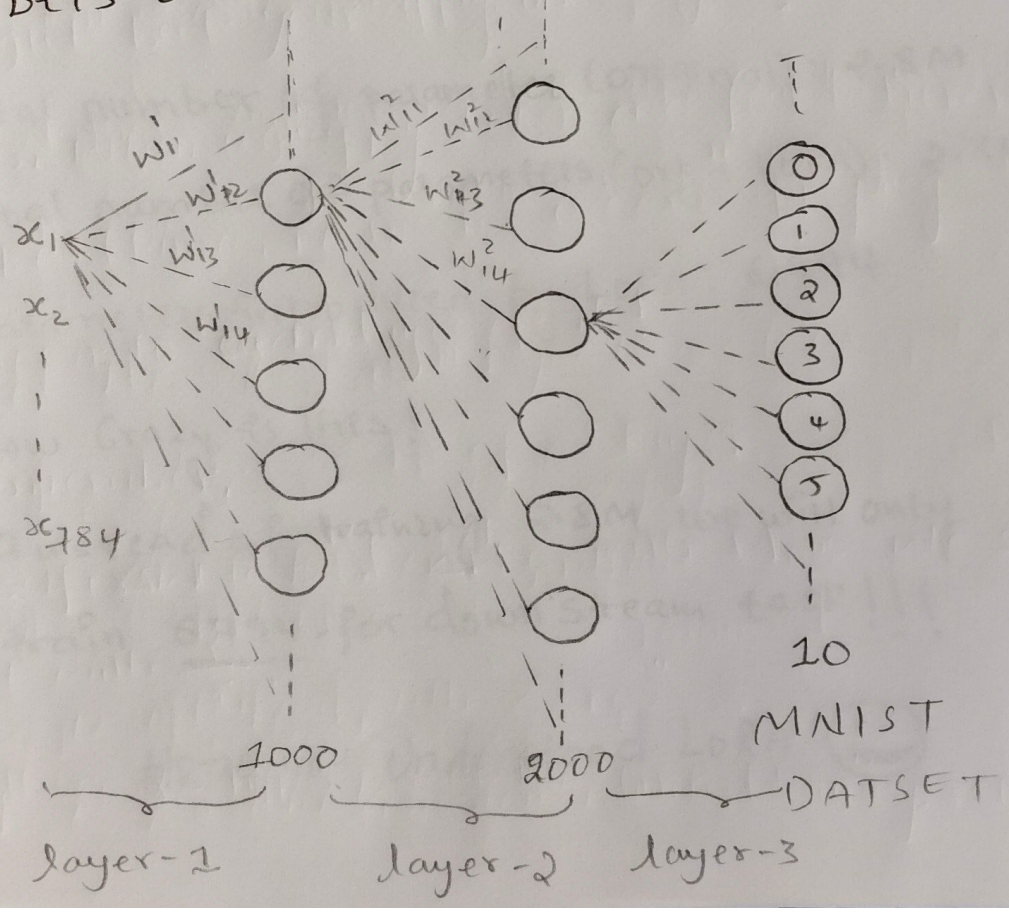
where, α - hyper parameter like learning rate

Why we need to scale $\Delta W x$?

- Controlling the influence of LoRA updates:
  - A higher α/r means the lora updates will have a stronger effect.
  - lower $\frac{\alpha}{r}$ ⇒ keeping the model closer to the original pre-trained weights.
- Stabilizing the Training process.

  When using higher LoRA ranks. Without this scaling, the LoRA updates could overwhelm the original model weights and lead to unstable traning.

Let's take an Example:



Output ↑ Input ↓

Layer 1 : W_0 : [1000, 784] + B_0 : [1000]

layer 2 : W_0 : [2000, 1000] + B_0 : [2000]

Layer 3 : W_0 : [10,00, 2000] + B_0 : [10]

Let's Apply LoRA to Layer 1 :

Layer 1 : W_0 : [1000, 784] + B_0 : [1000] +

lora_A : [1, 784] + loraB : [1000, 1]

Now,

We freeze W_0 & B_0 and train only LoRA-A & LoRA-B Matrices for Layer-1

W'll for other layers

Total number of parameter (original) : 2.8M

Total number of parameters (org + LoRA) : 2.813M

parameters introduced by LoRA : 6,794

How Crazy is this!

Instead of training 2.8M we will only train 6794 for down stream task !!!

Hope you understood LoRA