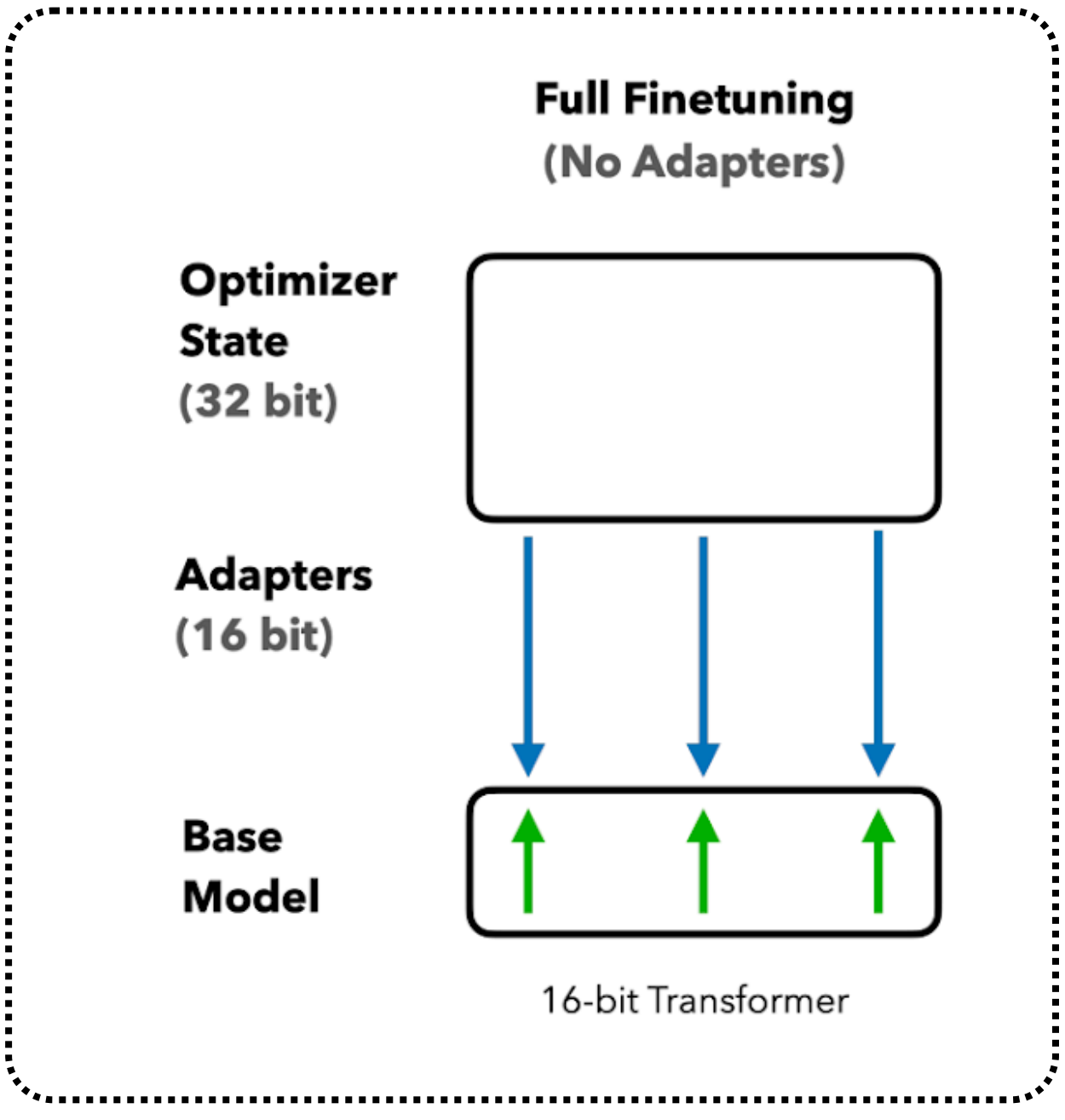
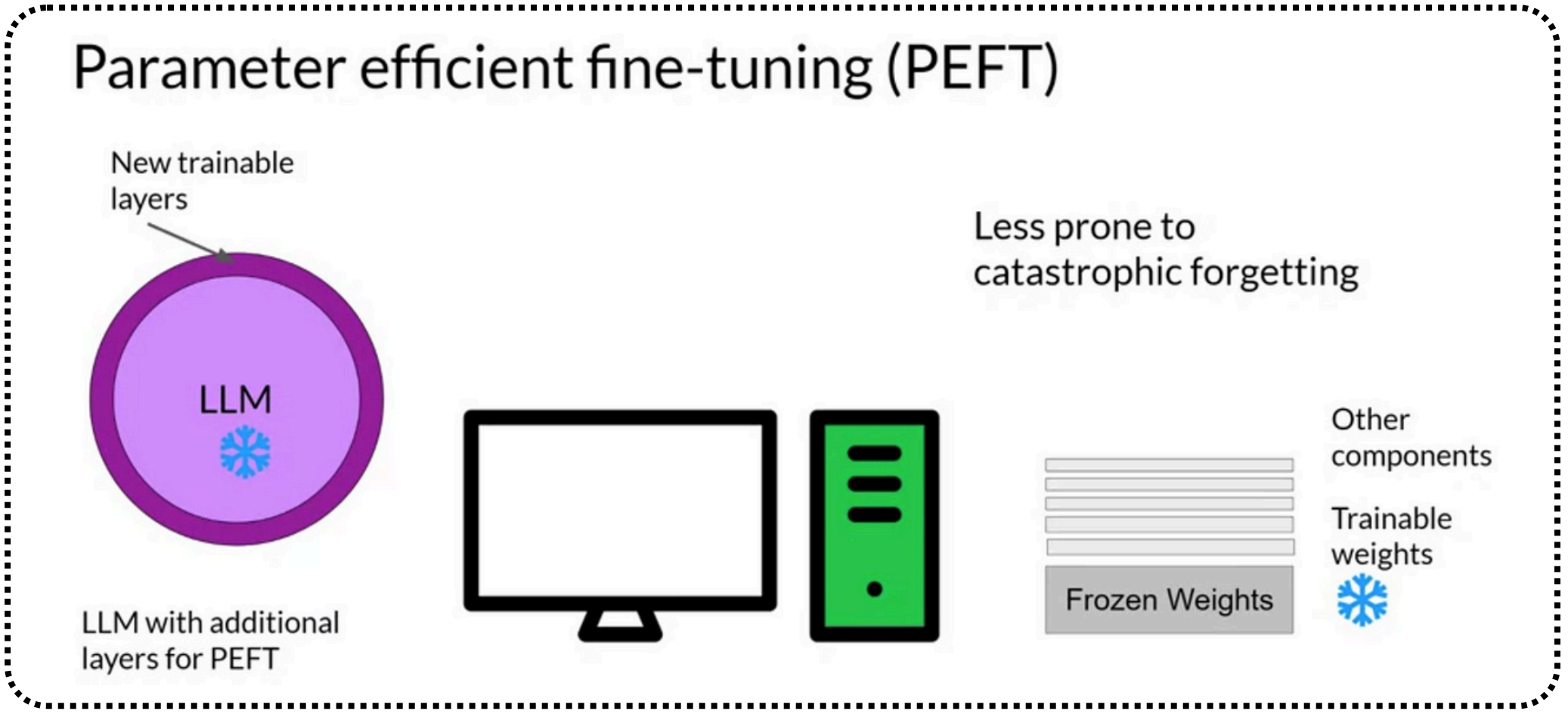
Mastering LLMs

Analytics Vidhya

Day 29: Full Fine-Tuning vs. PEFT

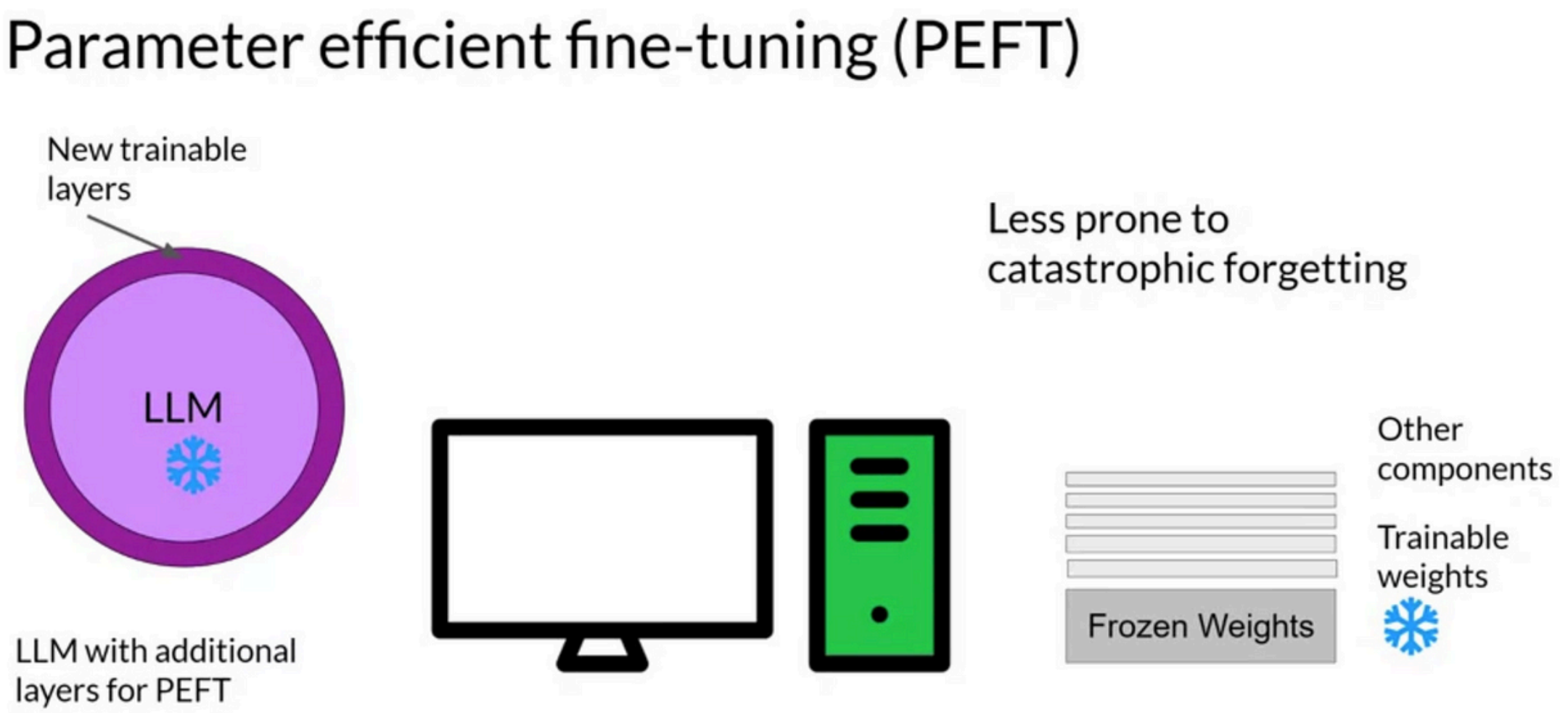


Fine-tuning large-scale models is essential for adapting them to specific tasks, but traditional full fine-tuning can be computationally expensive. Parameter-Efficient Fine-Tuning (PEFT) methods provide an alternative by updating only a small subset of model parameters. This article compares full fine-tuning and PEFT across various dimensions, helping practitioners choose the right approach.



Why PEFT?

- Fine-tuning massive models like GPT and BERT can be computationally expensive and storage-intensive. PEFT methods tackle this challenge by introducing small, trainable parameters that modify specific layers of the model while keeping the majority of weights frozen. This dramatically reduces computational overhead and enables efficient adaptation to new tasks.



What is Full Fine-Tuning?

Full fine-tuning involves updating all model parameters based on a new dataset and task. This is commonly used in adapting pre-trained models like GPT, BERT, and T5 to domain-specific applications.

Advantages

- Maximum Task Adaptation: Fully adapts the model to new tasks without constraint.
- No Architectural Modifications: Maintains the model's original structure.
- Better Performance on Large Datasets: Ideal when extensive labeled data is available.

Disadvantages

- High Computational Cost: Requires significant GPU/TPU resources and storage.
- Catastrophic Forgetting: May lose knowledge from pre-training.
- Inefficient for Multi-Task Learning: Requires retraining for each new task.

What is PEFT?

Parameter-Efficient Fine-Tuning (PEFT) optimizes adaptation by modifying a smaller set of parameters while keeping the majority of the model frozen. Common PEFT techniques include:

- **LoRA (Low-Rank Adaptation)** – Introduces trainable low-rank matrices.
- **IA3 (Infused Adapter-Attention)** – Modifies activations using learned scaling factors.
- **Adapters** – Introduces small task-specific layers between transformer layers.
- **Prefix Tuning** – Optimizes continuous prompts instead of modifying weights.

Advantages

- **Lower Computational Requirements**: Reduces memory and compute costs.
- **Preserves Pre-trained Knowledge**: Minimizes catastrophic forgetting.
- **Efficient for Multi-Task Learning**: Enables modular adaptation to multiple tasks.
- **Faster Training & Deployment**: Less overhead compared to full fine-tuning.

Disadvantages

- **Limited Adaptability**: May not fully capture domain-specific knowledge.
- **Increased Inference Complexity**: Some PEFT methods require additional computation during inference.
- **Task-Specific Hyperparameter Tuning**: Requires careful selection of adaptation layers and learning rates.

Full Fine-Tuning vs. PEFT

Aspect	Full Fine-Tuning	PEFT
Trainable Parameters	Entire Model	Small Subset (e.g., adapters, LoRA)
Computational Cost	High	Low to Moderate
Memory Usage	High	Low
Training Time	Long	Short
Performance	Best (with sufficient data)	Good (for most tasks)
Multi-Task Adaptability	Poor	Excellent
Pretrained Knowledge Retention	May be lost	Preserved

When to use Full Fine-Tuning

Use Full Fine-Tuning If:

- You have abundant computational resources.
- You need full adaptation to a domain-specific task.
- You are working with a small or specialized model.
- You require the highest performance for a single large dataset.

When to use PEFT?

Use PEFT If:

- You have limited computational resources.
- You need to fine-tune large models efficiently.
- You are working on multi-task learning or transfer learning.
- You want to retain knowledge from the pre-trained model while adapting to new tasks.

Analytics Vidhya

Stay Tuned for **Day 30** of

Mastering LLMs