

QuanTA: Efficient High-Rank Fine-Tuning of LLMs with Quantum-Informed Tensor Adaptation

Zhuo Chen Rumen Dangovski Charlotte Loh
Owen Dugan Di Luo Marin Soljacic
presenter: Shen Yuan



中國人民大學
RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院
Gaoling School of Artificial Intelligence

Outline

Motivation: Low Rank is not Always Sufficient

Preliminary: Quantum Circuit

- Quantum state and vector representation

- Quantum circuit and matrix representation

Quantum-informed Tensor Adaptation

- Construction

- Initialization

Experiments

Conclusion

Outline

Motivation: Low Rank is not Always Sufficient

Preliminary: Quantum Circuit

Quantum state and vector representation

Quantum circuit and matrix representation

Quantum-informed Tensor Adaptation

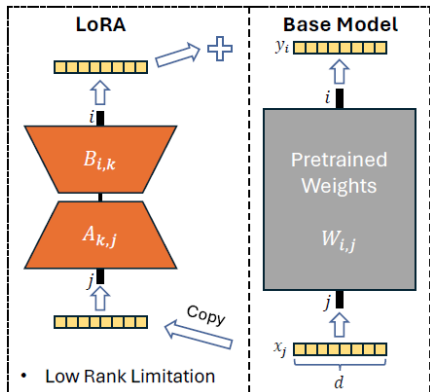
Construction

Initialization

Experiments

Conclusion

LoRA



LoRA operates under the hypothesis that parameter updates during fine-tuning exhibit a low “**intrinsic rank**”.

For a pretrained weight matrix $W_0 \in \mathbb{R}^{d \times k}$, LoRA parameterizes the weight update as $W' = W_0 + \delta W = W_0 + BA$, where $A \in \mathbb{R}^{r \times k}$ and $B \in \mathbb{R}^{d \times r}$ are low-rank matrices.

Motivation: Low Rank is not Always Sufficient

The low-rank hypothesis may **fail for more complex tasks**, especially for those that significantly differ from the pre-training dataset.

Motivation: Low Rank is not Always Sufficient

The low-rank hypothesis may **fail for more complex tasks**, especially for those that significantly differ from the pre-training dataset.

The RTE dataset is simpler, thus more likely to conform to the low-rank hypothesis, whereas the DROP dataset presents a greater challenge.

Model	Accuracy/ F_1 -Score (\uparrow)	
	RTE	DROP
LLaMA2 _{7B} Base	61.0	19.8
LLaMA2 _{7B} LoRA _{$r=64$}	86.0	55.2
LLaMA2 _{7B} LoRA _{$r=128$}	85.8	56.2

Table 1: Performance of base and LoRA fine-tuned LLaMA2-7B on RTE [49] and DROP [50] datasets. We use accuracy and F_1 -score as the metrics for them respectively.

Motivation: Low Rank is not Always Sufficient

To further measure the “intrinsic rank” of weight updates for these datasets, we compare the subspace spanned by the right singular vectors.

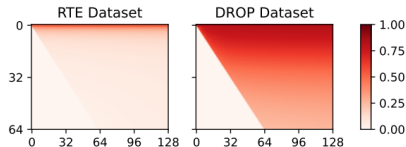


Figure 2: Subspace similarities between two LoRA experiments of different ranks (64 and 128) for two datasets. Each point (i, j) represents the subspace similarity between the first i right singular vectors of the $r = 64$ experiment, and the first j right singular vectors of the $r = 128$ experiment. Only points for $i \leq j$ are plotted. DROP dataset has a significantly high “intrinsic rank” than RTE dataset.

A subspace similarity close to 1 indicates significant overlap, suggesting that the subspace is crucial for fine-tuning, while a similarity close to 0 suggests orthogonality, implying that the vectors represent noise.

Outline

Motivation: Low Rank is not Always Sufficient

Preliminary: Quantum Circuit

Quantum state and vector representation

Quantum circuit and matrix representation

Quantum-informed Tensor Adaptation

Construction

Initialization

Experiments

Conclusion

Quantum state and vector representation

An N -qubit quantum state $|\psi\rangle$ is a 2^N -dimensional complex-valued vector in Hilbert space,

$$|\psi\rangle = \sum_i \psi_i |i\rangle \in \mathbb{C}^{2^N} \quad (1)$$

where ψ_i are the components and $|i\rangle$ are the basis vectors (similar to \mathbf{e}_i in vector notation).

Quantum state and vector representation

An N -qubit quantum state $|\psi\rangle$ is a 2^N -dimensional complex-valued vector in Hilbert space,

$$|\psi\rangle = \sum_i \psi_i |i\rangle \in \mathbb{C}^{2^N} \quad (1)$$

where ψ_i are the components and $|i\rangle$ are the basis vectors (similar to \mathbf{e}_i in vector notation).

We can view the quantum state as a multi-dimensional tensor with different indices labeling different qubits:




$$|\psi\rangle = \psi_{i_1, i_2, \dots, i_N} |i_1, i_2, \dots, i_N\rangle \quad (2)$$

This can be equivalently viewed as reshaping the quantum state from a vector in \mathbb{C}^{2^N} to a tensor in $\mathbb{C}^{2 \times 2 \times \dots \times 2}$.

Quantum circuit and matrix representation

A quantum circuit is a unitary matrix $\mathcal{U} \in \mathbb{U}(2^N) \subset \mathbb{C}^{2^N \times 2^N}$ that transforms one quantum state into another: $|\phi\rangle = \mathcal{U}|\psi\rangle$.

These circuits are constructed from smaller unitary matrices known as quantum “gates,” which operate on one or two qubits.

Operator	Gate(s)	Matrix
Pauli-X (X)	 \oplus	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

Quantum circuit and matrix representation

A one-qubit gate is a unitary matrix $U^{(1)} \in \mathbb{U}(2^1)$, while a two-qubit gate is a unitary matrix $U^{(2)} \in \mathbb{U}(2^2)$.

The one-qubit gate is as follows:

$$U^{(1)}|\psi\rangle = \sum_{j_n} U_{i_n j_n}^{(1)} \psi_{i_1, i_2, \dots, j_n, \dots, i_N} |i_1, i_2, \dots, i_N\rangle \quad (3)$$

The two-qubit gate as follows:

$$U^{(2)}|\psi\rangle = \sum_{j_m, j_n} U_{i_m, i_n; j_m, j_n}^{(2)} \psi_{i_1, i_2, \dots, j_m, \dots, j_n, \dots, i_N} |i_1, i_2, \dots, i_N\rangle \quad (4)$$

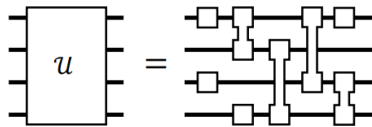


Figure 3: Any unitary matrix can be decomposed into a quantum circuit using one- and two-qubit gates.

torch example

```
>>> a=torch.randn(2,3,4)
>>> u=torch.randn(3,3)
>>> a=torch.permute(a,(0,2,1))
>>> a.shape
torch.Size([2, 4, 3])
>>> output=torch.matmul(a,u)
>>> output.shape
torch.Size([2, 4, 3])
```

Outline

Motivation: Low Rank is not Always Sufficient

Preliminary: Quantum Circuit

Quantum state and vector representation

Quantum circuit and matrix representation

Quantum-informed Tensor Adaptation

Construction

Initialization

Experiments

Conclusion

Construction

In this paper, the authors focus on the case of square weight matrices $W \in \mathbb{R}^{d \times d}$.



By reshaping $x \in \mathbb{R}^d$ to $x \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_N}$, the hidden vector can be interpreted as a quantum state with N “qudits,” with the n th axis corresponding to a qudit with local dimension d_n .

Quantum-informed Tensor Adaptation

Similar to a quantum circuit, QuanTA consists of “gates” (i.e., trainable tensors) that apply to only specific axes.

Since single-axis gates are subsets of two-axis gates, QuanTA uses only two-axis gates.

Let $T^{(\alpha)}$ be a tensor of shape $T^{(\alpha)} \in \mathbb{R}^{d_{m(\alpha)} d_{n(\alpha)} \times d_{m(\alpha)} d_{n(\alpha)}}$ that operates on the $m^{(\alpha)}$ th and $n^{(\alpha)}$ th axes with corresponding dimensions $d_{m(\alpha)}$ and $d_{n(\alpha)}$.

$$(T^{(\alpha)} \mathbf{x})_{i_1, \dots, i_m, \dots, i_n, \dots, i_N} := \sum_{j_m j_n} T_{i_m, i_n; j_m j_n}^{(\alpha)} x_{i_1, \dots, j_m, \dots, j_n, \dots, i_N} \quad (5)$$

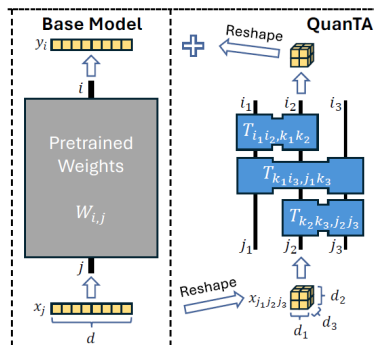
This operation can be viewed as a matrix-vector multiplication with all but the $m^{(\alpha)}$ th and $n^{(\alpha)}$ th axes created as batch dimensions.

An example

Let's consider the case of $N = 3$.

$\{T^{(\alpha)}\}$ consists of three tensors, each applied to two axes.

$$(\mathcal{T}x)_{i_1, i_2, i_3} = \sum_{k_1, k_2} T_{i_1, i_2; k_1, k_2}^{(1)} \sum_{j_1, k_3} T_{k_1, i_3; j_1, k_3}^{(2)} \sum_{j_2, j_3} T_{k_2, k_3; j_2, j_3}^{(3)} x_{j_1, j_2, j_3} \quad (6)$$



```
torch.einsum("...abc,efbc,diaf,ghde->...ghi", x, T_3, T_2, T_1)
```

Initialization

At initialization, the adapted model should be the same as the base model. That is, with an untrained adapter, calling forward should be an identity transform.

To address this issue, we use another set of tensors $\{S^{(\alpha)}\}$ (with the corresponding QuanTA operator S) that are initialized to the same value as $\{T^{(\alpha)}\}$ but remain frozen throughout fine-tuning. The adapted layer is defined as:

$$y = W_{\theta}x := W_0x + \mathcal{T}_{\theta}x - Sx \quad (7)$$

Outline

Motivation: Low Rank is not Always Sufficient

Preliminary: Quantum Circuit

Quantum state and vector representation

Quantum circuit and matrix representation

Quantum-informed Tensor Adaptation

Construction

Initialization

Experiments

Conclusion

Experiments

Model	PEFT Method	# Params (%)	F_1 Score (\uparrow)
LLaMA2 _{7B}	FT	100%	59.4
	Series	0.747%	58.8
	Parallel	0.747%	59.0
	LoRA _{$r=8$}	0.062%	54.0
	LoRA _{$r=32$}	0.249%	54.8
	LoRA _{$r=128$}	0.996%	56.2
	QuanTA₁₆₋₈₋₈₋₄ (Ours)	0.041%	59.5
	QuanTA₁₆₋₁₆₋₁₆ (Ours)	0.261%	59.6
LLaMA2 _{13B}	LoRA _{$r=8$}	0.050%	61.0
	QuanTA₁₆₋₈₋₈₋₅ (Ours)	0.029%	69.0
LLaMA2 _{70B}	LoRA _{$r=8$}	0.024%	74.3
	QuanTA₁₆₋₈₋₈₋₈ (Ours)	0.014%	79.4

Table 2: Benchmark of various fine-tuning methods on the DROP dataset using LLaMA2 7-70 billion parameter models as the base model. In each case, we report the average of F_1 score over 2-4 experiments with different random seeds.

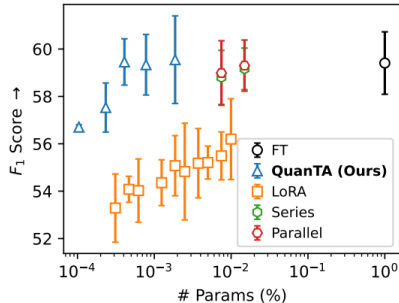


Figure 4: Benchmark of different fine-tuning methods on the DROP dataset as a function of training parameters using LLaMA2 7 billion parameter model as the base model.

Experiments

Model	PEFT Method	# Params (%)	Accuracy (†)								
			BoolQ	PIQA	SIQA	HellaS.	WinoG.	ARC-e	ARC-c	OBQA	Avg.
GPT-3 _{175B} *	–	–	60.5	81.0	–	78.9	70.2	68.8	51.4	57.6	–
PaLM _{540B} *	–	–	88.0	82.3	–	83.4	81.1	76.6	53.0	53.4	–
ChatGPT*	–	–	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
LLaMA _{7B}	FT	100%	71.3	82.1	78.6	90.2	79.0	82.9	67.2	76.8	78.5
	Prefix*	0.11%	64.3	76.8	73.9	42.1	72.1	72.9	54.0	60.6	64.6
	Series*	0.99%	63.0	79.2	76.3	67.9	75.7	74.5	57.1	72.4	70.8
	Parallel*	3.54%	67.9	76.4	78.8	69.8	78.9	73.7	57.3	75.2	72.3
	LoRA*	0.83%	68.9	80.7	77.4	78.1	78.8	77.8	61.3	74.8	74.7
	DoRA†	0.43%	70.0	82.6	79.7	83.2	80.6	80.6	65.4	77.6	77.5
	DoRA†	0.84%	69.7	83.4	78.6	87.2	81.0	81.9	66.2	79.2	78.4
	QuanTA (Ours)	0.041%	71.6	83.0	79.7	91.8	81.8	84.0	68.3	82.1	80.3
LLaMA _{13B}	Prefix*	0.03%	65.3	75.4	72.1	55.2	68.6	79.5	62.9	68.0	68.4
	Series*	0.80%	71.8	83.0	79.2	88.1	82.4	82.5	67.3	81.8	79.5
	Parallel*	2.89%	72.5	84.8	79.8	92.1	84.7	84.2	71.2	82.4	81.5
	LoRA*	0.67%	72.1	83.5	80.5	90.5	83.7	82.8	68.3	82.4	80.5
	DoRA†	0.35%	72.5	85.3	79.9	90.1	82.9	82.7	69.7	83.6	80.8
	DoRA†	0.68%	72.4	84.9	81.5	92.4	84.2	84.2	69.6	82.8	81.5
	QuanTA (Ours)	0.029%	73.2	85.4	82.1	93.4	85.1	87.8	73.3	84.4	83.1
LLaMA _{27B}	FT	100%	72.9	83.0	79.8	92.4	83.0	86.6	72.0	80.1	81.2
	LoRA†	0.83%	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0	77.6
	DoRA†	0.43%	72.0	83.1	79.9	89.1	83.0	84.5	71.0	81.2	80.5
	DoRA†	0.84%	71.8	83.7	76.0	89.1	82.6	83.7	68.2	82.4	79.7
	QuanTA (Ours)	0.041%	72.4	83.8	79.7	92.5	83.9	85.3	72.5	82.6	81.6
LLaMA _{213B}	LoRA	0.67%	73.3	85.6	80.8	91.6	85.5	84.2	73.7	83.3	82.3
	QuanTA (Ours)	0.029%	75.8	86.9	81.2	94.4	87.0	89.6	77.9	85.2	84.8
LLaMA _{38B}	LoRA†	0.70%	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
	DoRA†	0.35%	74.5	88.8	80.3	95.5	84.7	90.1	79.1	87.2	85.0
	DoRA†	0.71%	74.6	89.3	79.9	95.5	85.6	90.5	80.4	85.8	85.2
	QuanTA (Ours)	0.035%	74.3	88.1	81.8	95.1	87.3	91.1	81.7	87.2	85.8

Table 3: Benchmark on various commonsense reasoning tasks. All results of models and PEFT methods labeled with “*” are from [54], and results with “†” are from [20].

Experiments

Model	PEFT Method	# Params (%)	Accuracy (\uparrow)				
			AQuA	GSM8K	MAWPS	SVAMP	Avg. W/O AQuA
GPT-3.5 _{175B} *	–	–	38.9	56.4	87.4	69.6	71.1
LLaMA2 _{7B}	FT	100%	19.3	65.2	92.0	80.7	79.3
	LoRA	0.83%	17.5	65.7	91.2	80.8	79.6
	QuanTA (Ours)	0.19%	16.7	67.0	94.3	80.3	80.5
LLaMA2 _{13B}	LoRA	0.67%	16.7	72.3	90.8	84.3	82.5
	QuanTA (Ours)	0.13%	18.9	72.4	94.5	84.8	83.9

Table 4: Benchmark on various arithmetic reasoning tasks. GPT-3.5 (labeled with “*”) results are taken from [54].

Outline

Motivation: Low Rank is not Always Sufficient

Preliminary: Quantum Circuit

- Quantum state and vector representation

- Quantum circuit and matrix representation

Quantum-informed Tensor Adaptation

- Construction

- Initialization

Experiments

Conclusion

Conclusion

- ▶ QuanTA is a very powerful and easy-to-implement adapter with no inference overhead.
- ▶ Although the paper discusses the necessity of high rank, QuanTA adapter does not guarantee high rank.