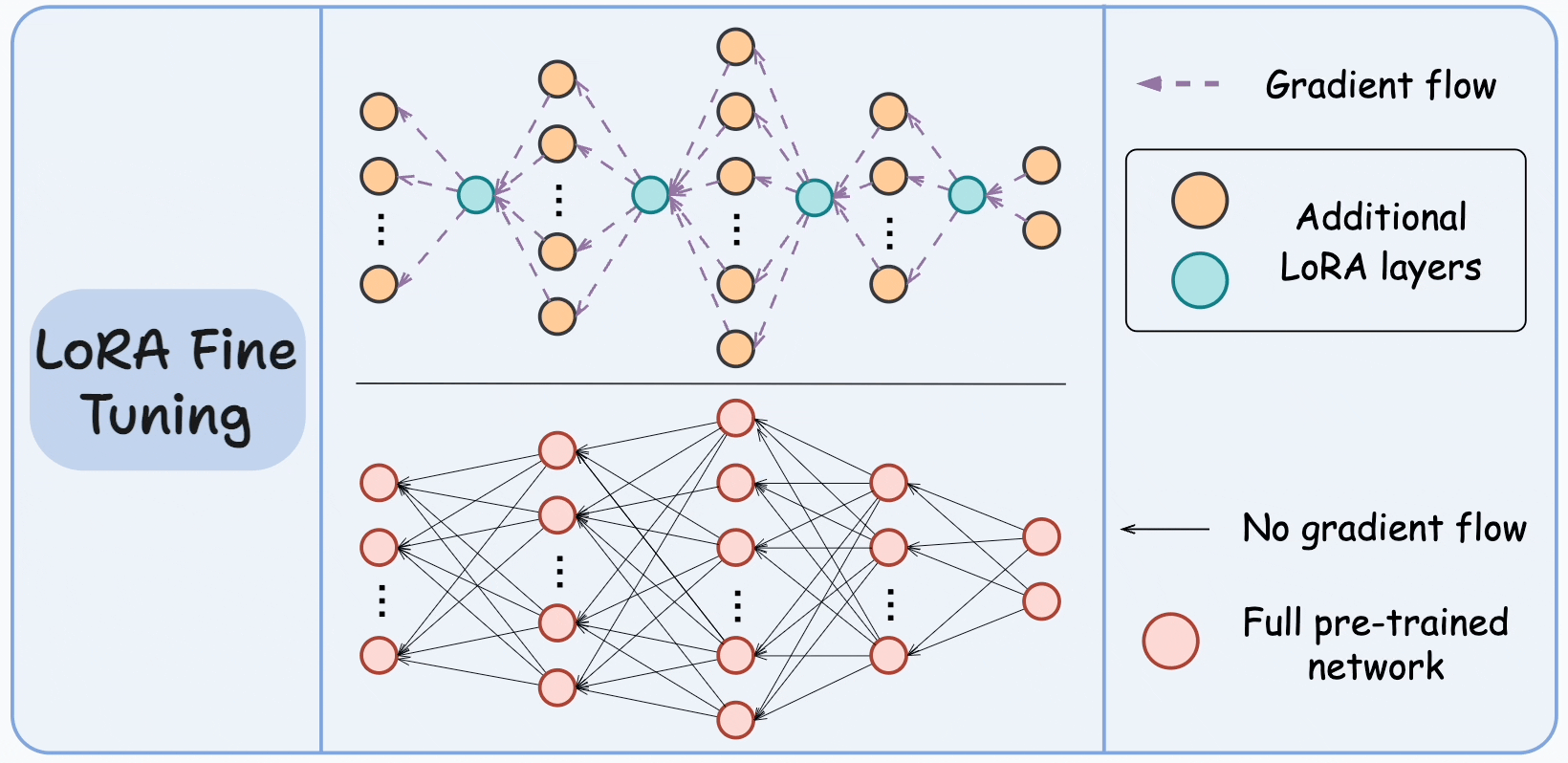
Lecture Notes

# Part 1



Low-Rank Adaptation:  
- The concept of low-rank adaptation is used in this context.  
- The idea is to learn low-rank matrices first, which have fewer parameters, and then combine them to get a matrix of the original size.  
- The low-rank matrices, after learning, are projected to the original dimension of the matrix, and then they are added.  
- There is another variation, alpha by R, where R stands for rank and alpha is another scaling factor.  
- This is based on the principle similar to SVD decomposition.  
  
SVD Decomposition:  
- SVD decomposition is used to compress or reduce the dimensionality of a matrix.  
- A matrix is decomposed into three matrices: U, S, and V, where S is a diagonal matrix.  
- The singular value matrix, which is a diagonal matrix, contains singular values only across the diagonal, and the rest of the entries are zero.  
- By removing some singular values, the dimensionality of the matrix can be reduced, leading to a loss of information but significant dimensionality reduction.  
  
Low-Rank Matrices:  
- The trick used in low-rank matrices is to decompose a large matrix into smaller matrices with fewer parameters.  
- Instead of injecting a full-sized matrix, the lower-rank matrices are used, which are learnt with a smaller number of parameters.  
- These low-rank matrices are then blown up to the original size at the time of inferencing.  
- The memory required for low-rank matrices is significantly less than that of the original matrix.  
- The low-rank matrices capture useful information, and the loss of information is not significant.  
- The final output at the time of inference requires the fine-tuned weights, the original weights, and the low-rank matrices.

# Part 2



Low-Rank Matrices in Neural Networks:  
  
- Concept:  
 - Injection of low-rank matrices into adapter layers for parameter-efficient fine-tuning.  
 - Linear algebra principle of decomposition is used.  
 - Smaller number of parameters are learned, reducing the memory requirement.  
 - During inference, the low-rank matrices are blown up to the original size.  
  
Key Points:  
  
- Low-rank matrices are used to adapt layers, one above self-attention, one above feed-forward network in every layer.  
- The original weights are frozen and not touched during training.  
- At inference time, the new input is processed, and the answer is given by performing the operation B into A, scaling the lower weights, adding it to the original weights, and then applying x on the final W to get the inference value.  
- Alpha by R is used to modulate the amount of lower weights used during inference. Typical values for alpha are 16, and rank is 4 to 8, 16 to 32.  
- The rank value determines the number of parameters to be learned during fine-tuning. Increasing the rank means more parameters are learned, bringing it closer to full fine-tuning.  
- The rank value should be decided based on the original size and the amount of knowledge you want to learn.  
- For a 25-size model, a rank of 1 is sufficient. For larger models, ranks of 16, 64, or even 64 can be used.  
- Design decisions that need to be taken while fine-tuning include the choice of rank and the amount of knowledge to be learned.

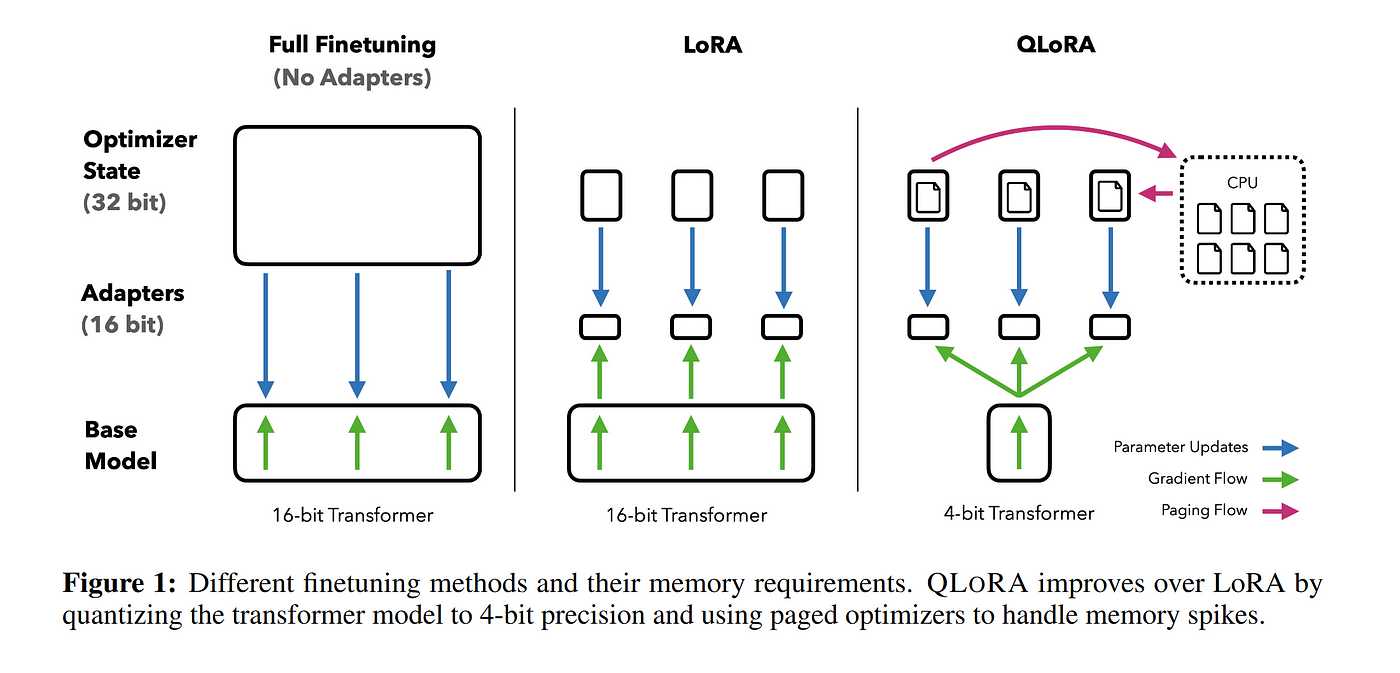
# Part 3

Low-Rank Adaptation in Fine-Tuning  
  
Introduction:  
- The more parameters, the better the performance, but not to the point of losing the original thought.  
- The rank is a design parameter that depends on the original size of the model.  
- For big models, the rank is usually around 16, 64. For smaller models, a rank of 1 is sufficient.  
  
Key Points:  
1. Rank:  
 - The rank of a matrix determines the number of independent pieces of information it contains.  
 - A high rank matrix can overtake the original model, causing catastrophic forgetting.  
 - The rank should be chosen based on the original size of the model and the amount of knowledge to be learned.  
  
2. Alpha and R:  
 - Alpha is a parameter that determines how much of the learned weights should be added to the original weights.  
 - R is a parameter that determines the amount of lower weights to be added to the original weights to prevent catastrophic forgetting.  
 - Both alpha and R play a crucial role in balancing the learning of generic and specific information.  
  
3. Catastrophic Forgetting:  
 - Catastrophic forgetting occurs when a high rank matrix is used for fine-tuning, causing the model to forget its original parametric knowledge and only learn the new fine-tuned weights.  
  
4. Low-Rank Adaptation:  
 - Low-rank adaptation allows learning low-rank matrices and projecting them to high-rank matrices without losing useful information.  
 - This principle is used in weight update rules during inference time, not during training.  
 - The decomposition matrix still carries the full-blown up information.  
  
5. Quantization (Q-Lower):  
 - In Q-Lower, the pre-trained weights are stored as SP 16 or SP 32 in memory.  
 - B and A are always floating point and are not quantized.  
 - During fine-tuning, B and A are learned as SP 16 representations, and then added to the original weights to get the final W.  
  
6. Q-Lower:  
 - In Q-Lower, the goal is to reduce the memory footprint of the model by quantizing the weights.  
 - The details of Q-Lower will be discussed in a later section.

# Part 4

\*\*Pre-trained Weights and Quantization\*\*  
  
- Pre-trained weights are represented as Single Precision (SP) 16 or SP 32.  
- B and A are always floating point and are never quantized.  
- In Q lower, the pre-trained weights are quantized into a 4-bit quantized matrix.  
- The quantized matrix is a mapping and does not contain parametric information.  
  
\*\*Fine-tuning in Q lower\*\*  
  
- B and A are learned again in Q lower.  
- B and A are not quantized and are always SP 32 or SP 16.  
- Real parametric knowledge on the fine-tuned dataset is learned.  
- At inference time, the quantized weight matrix is dequantized.  
- Dequantization is done part by part to reduce memory requirements.  
- Dequantization introduces errors.  
  
\*\*Differences between Lora and Qlora\*\*  
  
- Lora receives the frozen weight in the FP32 or the 16 format.  
- Qlora receives W in the quantized format.  
- Both are fine-tuning techniques.  
  
\*\*Qlora Model Configuration\*\*  
  
- Quantization configuration, load in 4-bit.  
- Alpha set to 32, rank set to 4.  
- Dropout and target modules are used.  
- Only Q projection and V projection are affected for fine-tuning.  
- The PESTModel API from the Hugging Face library ensures that the pre-trained model is frozen.

# Part 5



\*\*Introduction to Qlora\*\*  
  
- Qlora is a method for fine-tuning pre-trained models.  
- It only requires a small amount of memory.  
- The principle is to dequantize step by step and B into A.  
- The configuration, load, and alpha have been set.  
- The rank is 4 and dropout is used.  
- The target modules affected for fine-tuning are Q projection and V projection.  
  
\*\*GetPestModel API\*\*  
  
- This API ensures that the pre-trained model is frozen.  
- It automatically freezes the model or gradients can be set to false explicitly.  
- It sets up a QLOWRUP principle by freezing the model and producing a wrapper around it.  
  
\*\*Inference and Fine-Tuning\*\*  
  
- Inference is the process of running the model to get an output.  
- Fine-tuning means the parameters are updated, resulting in faster inference.  
- Rag is a method of fine-tuning where the parameters are learned at inference time.  
- Fine-tuning is faster during inference but requires more training time.  
  
\*\*Parameters\*\*  
  
- F is 1632, 16 quantization, and the rank.  
- These parameters are used to design the Qlora model and are adjusted through trial and error.  
  
\*\*Challenges\*\*  
  
- Building an LLM (Language Model) requires specific parameters.  
- Fine-tuning and RIG (Reinforcement Learning with Human Feedback) are complex topics.  
- Reinforcement learning uses reward and punishment models, but the details of how q values are created and updated are complex.  
- It requires a deep understanding of calculus and linear algebra.  
- With multiple projects, it is difficult to give proper time to any one project and produce meaningful results.  
  
\*\*Conclusion\*\*  
  
- Qlora is a method for fine-tuning pre-trained models that requires a small amount of memory.  
- It involves dequantizing step by step and B into A.  
- The GetPestModel API ensures that the pre-trained model is frozen.  
- Fine-tuning results in faster inference but requires more training time.  
- Building an LLM and understanding reinforcement learning are complex topics that require a deep understanding of calculus and linear algebra.

# Part 6

\*\*Reinforcement Learning\*\*  
  
- Concept: Reward and punishment model, driven by reinforcement learning.  
- Behind the scenes: Discounted reward, temporal difference, Q learning, value learning, policy learning.  
- Q Value Creation and Update:  
 - Q value creation method and update rules are not discussed in detail.  
 - Deep learning models are used to learn Q values.  
 - Data feeding is not explicitly mentioned.  
  
\*\*Challenges in Education\*\*  
  
- Balancing theory and practice:  
 - Lack of proper balance between theoretical knowledge and practical application.  
 - Theoretical foundations are taught, but the connection to practical applications is not made.  
- Interest and Attention Span:  
 - Lack of interest in the subject makes it difficult to learn and understand.  
 - Short attention spans make it hard to focus on a single topic for an extended period.  
- Project-based Learning:  
 - Having too many projects can lead to shallow understanding and inability to focus on any project properly.  
 - Two to three meaningful, big-sized projects are recommended.  
  
\*\*Education System Reform\*\*  
  
- Question of reforming the education system:  
 - Balancing theory and practice is a significant concern.  
 - There is a need to find a way to balance the two effectively.  
- Expression Learning Required:  
 - The enjoyment and expression that used to happen during teaching are missing.  
 - Students are relying on resources like YouTube and chatGPT, which can lead to a lack of understanding.  
  
\*\*Multiple Resources and Infrastructure\*\*  
  
- Multiple resources are available, but no single gospel truth.  
- The lack of a proper textbook is an issue.  
- Infrastructure and resources are crucial for research and development.  
  
\*\*Productization and Recommended Systems\*\*  
  
- The use of LLM is essential, but the extent of productization is unclear.  
- The number of recommended systems is unknown.  
- There is a need to think of something new and innovative in the field of LLM.  
  
\*\*Student Exposure and Migration\*\*  
  
- Students are not getting the right exposure, leading to a gap between their knowledge and that of students abroad.  
- Students often go abroad to gain the experience they are missing in their home country.

# Part 7

\*\*Introduction\*\*  
- The speaker discusses the gap between the recommended systems and the current state of the field, citing the example of a chat GPT system.  
- They mention the possibility of using LLM (Linear Logic with Modalities) for something new.  
  
\*\*The Gap in Exposure\*\*  
- The speaker points out that the students in their current environment may not be getting the right exposure due to the lack of high-end infrastructure and students compared to other research labs.  
- They mention that students often leave for better opportunities abroad, seeking the experience of cutting-edge technology work.  
  
\*\*Case Study\*\*  
- The speaker shares an instance where a student who was offered 22 lakhs in Citrix left for a lesser salary at Microsoft Research because they were more interested in the work.  
  
\*\*Motivation and Work Interest\*\*  
- The speaker emphasizes that students are not just looking for high salaries but are interested in doing good work, especially those in the top-tier institutions.  
  
\*\*The Importance of Technical Depth\*\*  
- The speaker stresses the need for technical depth, as it is crucial for impressing managers and getting ahead in the tech industry.  
- They mention that having technical depth earns respect and makes one less vulnerable to being fooled or manipulated.  
  
\*\*Examples of Tech Strong Individuals\*\*  
- The speaker provides examples of tech strong individuals like Anantraman, who cannot be touched or approached easily due to their expertise.  
  
\*\*Conclusion\*\*  
- The speaker concludes by mentioning Anantraman's achievements and association with C.C.B. and being an adjunct professor.  
- They also mention their own desire to teach LLM but being unable to do so due to their daughter being in Jenny-I, and how they took the opportunity to teach in the current environment instead.  
- They also mention that LLM has been modified in Jenny-I after the initial version, and they are asking OMA madam to teach all those aspects that were missed out.

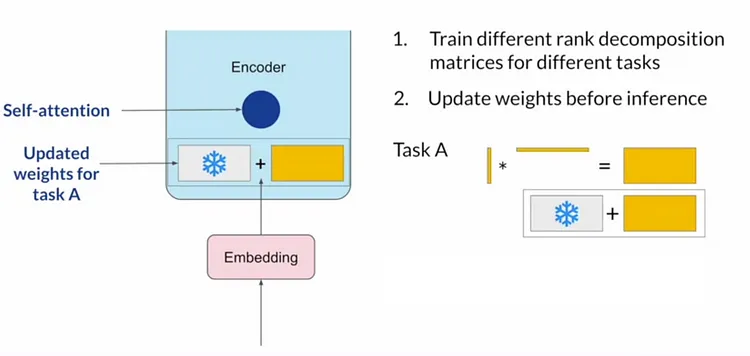
# Part 8

\*\*Background of the Speaker\*\*  
- The speaker is a professor who was supposed to teach a course but couldn't due to a conflict of interest as his daughter is enrolled in the same course.  
- He has worked as a director in Nadobi, in several big companies, and ran his own startup.  
- He is associated with C.C.B and is an adjunct professor.  
  
\*\*Association with C.C.B and Jenny-I\*\*  
- The speaker designed the Jenny-I course for C.S., but due to a conflict of interest, he couldn't teach it.  
- LLM (Lightweight Learning Machine) was modified by the speaker as it is lighter compared to the one proposed in Jenny-I.  
- The speaker suggested another professor to teach the parts he missed out, such as parent coding, BM25, etc.  
- The speaker will return to C.S. next year to teach.  
  
\*\*Teaching Experience\*\*  
- Last year, the speaker was not able to teach ML, TDL, GenNIA, etc., in C.S. due to a conflict of interest.  
- He taught data structures last year and this year he is not able to teach TDL.  
- In the TDL class last year, all students were 9 pointers, an elective batch.  
- The speaker found it challenging to teach in that class as the students were not familiar with the concepts.  
  
\*\*Interview Experience\*\*  
- The speaker shared an experience where a student who was in Maxwell TDL class and was doing an internship in the center didn't get selected.  
- The student had a personal issue, and the speaker advised him to improve his behavior and confidence.  
- The student followed the advice and got selected in the next job.  
  
\*\*Counseling Experience\*\*  
- The speaker shared an experience where he had to counsel a student who was not getting placed due to family issues and low confidence.  
- The student's mother was putting too much pressure on him, and the speaker had to intervene and counsel the mother first.  
- The student made a promise to attend interviews and followed through, eventually getting placed.

# Part 9

Mentorship Interventions and Counseling:  
  
\* Mentor intervened in a student's life who was struggling with confidence due to his single parent upbringing and his mother's nightly prayers and physical fights.  
\* Conducted counseling sessions with the mother and the student.  
\* Promised the student to attend at least one placement event and kept his promise.  
\* The student eventually got a job and is now doing well.  
\* Mentor also helped another student who was struggling with job placements and boosted his confidence.  
\* The mentor and the student have maintained a friendship and the student has invited the mentor to his marriage.  
  
Quantization and Decontization:  
  
\* The mentor explained that the models they train are based on neural networks and the models learning is in the form of weights.  
\* The weight matrix is a huge matrix representing the parameters or the weights of the model.  
\* The mentor mentioned examples of large models like GPT with 176 billion parameters, LAMMA 3.2 with 7 billion, 8 billion, or 13 billion parameters.  
\* These large matrices represent the parametric knowledge of the model.

# Part 10



\*\*Neural Networks and Weight Matrices\*\*  
  
- Models trained are based on neural networks, resulting in weight matrices.  
- Weight matrices represent the parametric knowledge of the model.  
- The size of the weight matrix corresponds to the number of parameters in the model.  
- For example, GPT is 176 billion parameters, LAMMA 3.2 is 7 billion, 8 billion, 13 billion.  
  
\*\*Weight Matrix Composition\*\*  
  
- The weight matrix consists of all the layers and heads in the transformer.  
- The 176 billion parameters in GPT are not just for one layer or one head.  
  
\*\*Weight Matrix Format\*\*  
  
- The values of the weight matrix are represented in floating point format.  
- The floating point representation is used for large numbers to avoid precision errors.  
- The internal representation of floating point numbers is binary.  
- The floating point 32-bit representation consists of:  
 - 1 bit for the sign bit  
 - 8 bits for the exponent  
 - 23 bits for the mantissa  
  
\*\*Memory Requirements\*\*  
  
- The memory required to store the weights is approximately 800 GB for GPT.  
- The memory required to store the gradients is also approximately 800 GB.  
- The total memory required for weights and gradients is easily crossing 1600 GB.  
  
\*\*Training Large Language Models\*\*  
  
- Large language models require a significant amount of computational resources.  
- Open AI, Google, and Microsoft are the only companies that can afford to train large language models.  
- They use tensor parallelism, clusterization, and many trials to train the models.  
- The investment required to train a large language model is huge, even for big companies.  
  
\*\*Alternative: Floating Point 16-bit Representation\*\*  
  
- To reduce the memory footprint, a 16-bit floating point representation can be used.  
- This reduces the memory footprint to approximately 400 GB.  
- This compromise allows smaller companies and academic institutions to work with large language models.

# Part 11



\*\*Large Language Models and Memory Requirements\*\*  
  
- GPT models require 800 GB of memory for learnt weights during training, and 1600 GB for training data.  
- This makes it impossible for small startups, medium-sized companies, and academic institutions to replicate.  
- Only a small number of large companies can afford to invest in building such models.  
  
\*\*16-bit Floating Point Precision\*\*  
  
- To reduce memory footprint, 16-bit floating point precision can be used instead of 32-bit.  
- This reduces the number of bits used to store the floating point number:  
 - 1 bit for the sign  
 - 5 bits for the exponent  
 - 10 bits for the mantissa  
- The range of numbers that can be represented is lower, but the performance regression is not significant.  
  
\*\*Quantization\*\*  
  
- Quantization is the process of extracting certain discrete values from a continuous signal.  
- Instead of representing a signal with a high degree of precision, only certain quantization steps are taken.  
- This can help reduce the memory footprint and computational requirements of a model.

# Part 12

Quantization in Signal Theory:  
- Quantization is the process of selecting only certain discrete values from a continuous signal.  
- Instead of sending a continuous value, a discrete value is sent, which is close to the continuous value.  
- This process reduces the variations and drastically reduces the number of bits required for representation.  
- Quantization is a reversible process, but there will be some amount of error.  
- In last language models, quantization is used to compress the range of weights.  
  
4-bit Quantization:  
- 4-bit quantization reduces the range of minus infinity to plus infinity to 16 levels.  
- Each level represents an integer number, and how many levels have taken 16 levels is called as 4-bit quantization.  
- To represent these numbers, 4 bits are required.  
  
Quantization of Weights:  
- In the given weight matrix, the weights are in floating point format.  
- A new weight matrix called as W dash is created, which is a quantized weight matrix.  
- The given weight matrix is converted into a quantized weight matrix, which is integer in nature.  
- The quantized weight matrix is used for further processing.

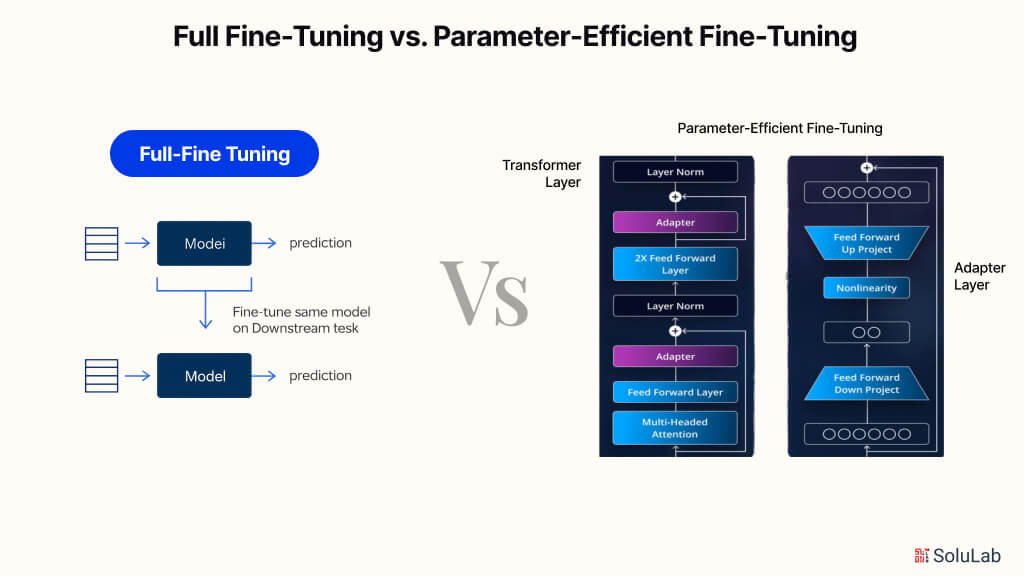
# Part 13

Quantization of Weights:  
- The range of weights is from negative infinity to positive infinity, but for practical purposes, only the range from -1 to 1 is used.  
- The number of quantization levels is 16, which is 4-bit quantization.  
- A weight matrix (W) is converted into a quantized weight matrix (W\_dash) by converting floating-point numbers into integer representations.  
- The quantization step is the process of converting floating-point weights into an integer representation.  
- The number of different integers depends on the number of quantization levels.  
- The quantized weight matrix reduces the memory requirement from 800 GB to 80 GB for 176 billion parameters.  
- The quantized weights are temporarily stored and de-quantized on the fly for further downstream tasks.  
- The quantized weights are used for temporary storage, but for further processing, floating-point numbers (FP16 or FP32) are used.  
  
Impact of Weight Representation on Memory:  
- FP32: Floating-point 32-bit representation.  
- FP16: Floating-point 16-bit representation.  
- 4-bit quantization model: Reduces the memory requirement significantly compared to FP32 or FP16.  
- Storing all learned weights in FP32 or FP16 requires a lot of memory, while storing them in a 4-bit quantization model saves a lot of memory.  
- The GPT model can be stored in a system with an ATGB memory or an A-100 GPU if the weights are stored in a 4-bit quantization model.

# Part 14

Fine Tuning:  
- A process of updating a pre-trained model with new data to adapt it to a specific task.  
- Involves updating a portion of the parameters, not the entire model.  
- Full fine tuning: updating all 176 billion parameters, not typically done due to infrastructure limitations.  
  
Parameter Efficient Fine Tuning (Pept):  
- A method that only updates a portion of the parameters, not the entire model.  
- Freezes the original 176 billion parameters and trains a new set.  
  
LORA:  
- One method of parameter efficient fine tuning.  
- Trains a new set of parameters, called "tuning parameters," that are a linear combination of the original parameters and new parameters.  
- The new parameters are learned during the fine-tuning process.  
- The original parameters are frozen and not updated.  
  
In LORA, the original matrix is multiplied by a learned matrix to produce the final output. The learned matrix is updated during the fine-tuning process. The original matrix is frozen and not updated.

# Part 15



Fine-tuning a large pre-trained model:  
- Full fine-tuning: Updating all 176 billion parameters, not typically done.  
- Parameter efficient fine-tuning (Pepped): Updating only a portion of the parameters.  
  
Lora:  
- Lora is a method for parameter efficient fine-tuning.  
- It adds Lora adapter layers above self-attention and feed-forward layers in the transformer model.  
- The adapter layers consist of two divisions: B layer (5x1) and A layer (1x5).  
- During training, only B and A are updated, while the original weights (W) remain frozen.  
- After training, the low-rank matrices B and A are ready for use.  
- B into A is used to project the 10 parameters learned during fine-tuning to the original 25 parameters.  
- Lora allows for learning a small number of parameters (10 in this example) while still projecting the results to the original number of parameters (25).

# Part 16

Low-Rank Matrix Adaptation:  
  
- Used for internal weight updates during gradient backpropagation  
- Only B and A are updated, W is not touched  
- B into A results in a 25x25 matrix, despite learning only 10 parameters  
- W + B A or alpha by R into B A are used for weight updates  
- Similar to SVD decomposition for dimensionality reduction  
- SVD decomposition compresses a matrix by removing singular values close to zero  
- Low-rank matrix adaptation is a concept used in low-rank adaptation, where a low-rank matrix is learned and then combined to form a new matrix  
- Learning low-rank matrices still captures useful information with minimal loss  
- Equivalent to learning a 5x5 matrix, but learning only 10 parameters and projecting it to 25  
- Trick used in low-rank matrices: instead of injecting a 5x5 full blown up matrix, the lower rank matrix is decomposed into 5x1 and 1x5, and then combined to form a 25x25 matrix.

# Part 17

Low-Rank Matrix Approximation:  
- A technique used to represent a large matrix with a smaller matrix, called a low-rank matrix, while minimizing the loss of information.  
- The low-rank matrix is a combination of two smaller matrices, typically 5x1 and 1x5, which are learned instead of the full-sized matrix.  
- This technique is useful for parameter-efficient fine-tuning, as it requires learning fewer parameters.  
- The low-rank matrix is equivalent to learning a 5x5 matrix, but with a smaller memory footprint.  
- At inference time, the low-rank matrix is expanded to the original size and combined with the original weights.  
  
Decomposition of a Matrix:  
- A low-rank matrix is a decomposition of a larger matrix into smaller matrices, representing the rank of the matrix.  
- The rank of the matrix is the number of non-zero singular values.  
- In the case of a 5x5 matrix, the decomposition results in 5x1 and 1x5 matrices, each representing a row or column of the original matrix.  
  
Fine-Tuning with Low-Rank Matrix:  
- Low-rank matrix is used to adapt the layers in a neural network during fine-tuning.  
- The original weights are frozen, and only the low-rank matrices are learned.  
- The original weights are not touched during backpropagation, but only the low-rank matrices are updated.  
- After fine-tuning, the learned low-rank matrices (B and A) and the original weights (W) are available for inference.  
  
Inference with Low-Rank Matrix:  
- During inference, a new input (X) is given, and the answer is to be obtained.  
- At inference time, the low-rank matrix is scaled by a factor (alpha) and added to the original weights (W).  
- The final answer is obtained by applying the scaled and combined weights to the input (X).  
  
Alpha and Rank:  
- Alpha is a scaling factor used to modulate the amount of low-rank weight used in inference.  
- Typical values for alpha are 16, and the rank is 4 to 8, 16 to 32.  
- The rank value determines the number of low-rank matrices to be injected into the lower case.  
  
Behavioral Issue:  
- A student who was not getting placed in interviews due to low confidence and fear of rejection.  
- The student's behavior was affected by the pressure of not getting placed, leading to a lack of attendance in interviews.  
- The student's story was shared by a professor who was heading the IAS department at that time.

# Part 18

\*\*Introduction\*\*  
  
- The speaker discusses the concept of injecting a rank to matrices to lower their behavior in certain situations.  
  
\*\*Behavioral Issues in Matrices\*\*  
  
- The speaker explains that matrices can exhibit behavioral issues, such as rejection in the last row, which can be indicative of low confidence.  
  
\*\*Case Study: Student's Confidence and Placement Issues\*\*  
  
- The speaker shares a personal anecdote about a student who struggled with placement due to home environment pressures and low confidence.  
- The student's mother required counseling to reduce the pressure on the student.  
- The student eventually got placed in Adobe after intervention and counseling.  
  
\*\*Mentorship and Counseling\*\*  
  
- The speaker emphasizes the importance of mentorship and counseling in boosting confidence, especially during distressing situations.  
- Mentorship can be as simple as a pep talk and does not require grand interventions.  
  
\*\*Friendship and Alumni Connections\*\*  
  
- The speaker shares a story about another student, Manisha, who struggled with placement but eventually succeeded and maintained a friendship with the speaker.  
- The speaker also mentions a collaboration opportunity with another alumnus.  
  
\*\*Formula for Quantization and Decontization\*\*  
  
- The speaker transitions to discussing a formula used for quantization and decontization and lower updates.  
- The speaker presents a problem for the audience to solve to better understand the concept.

# Part 19

\*\*Friendship and Collaboration\*\*  
  
- A student (from Surat, North Gujarat) invited the speaker to his marriage, and they have maintained a friendship and plan for a joint collaboration.  
- The student graduated in 2014 and promised to return to serve the institution, which he has done for 10 years.  
  
\*\*Quantization and Decontization\*\*  
  
- The speaker presents a problem for quantization and decontization, asking the audience to find the error in quantization.  
- The problem is to find the minimum and maximum values of X from a given set, given that B is 4 bits.  
- The speaker explains that the quantization of every value is X, and provides an example of quantization using a formula.  
- The formula for quantization is given as: X = (delta \* X) + 6 (minimum).  
- The speaker asks the audience to solve the problem later, and mentions that there will be a lab experimentation on Lora on a later date.  
  
\*\*Dequantization\*\*  
  
- The speaker mentions that after dequantization, the audience will have to find the error.  
- The speaker explains that dequantization depends on the number of billion parameters, and may have a little extra overhead.  
- The speaker provides an example of dequantization, but does not elaborate further.  
  
\*\*Miscellaneous Points\*\*  
  
- The speaker mentions that the student who invited him to his marriage was the best outgoing student of his batch.  
- The speaker mentions that he has written to several alumni to assist in his capsule teams, and that they have all agreed to help.  
- The speaker mentions that the teams are not showing interest, but that there are many alumni ready to help.  
- The speaker mentions that there will be a hackathon, and that he will announce it later.  
- The speaker mentions that the student who invited him to his marriage had asked for a notebook, and that the speaker should ask the attendant to clean it up.  
- The speaker mentions that the audience's thought process of disrespecting people will have an impact on placement.  
- The speaker mentions that the audience should calculate later, and that they can call it anything.  
- The speaker mentions that after dequantization, the audience will have some value, and that they should find the error in this value.  
- The speaker mentions that the dequantization error is not the same as the quantization error.  
- The speaker mentions that the dequantization error is not the same as the quantized value.  
- The speaker mentions that the dequantization error is not the same as the original value.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization and rounding.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, and finding the error.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, finding the error, and writing it as 0.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, finding the error, writing it as 0, and writing it as 0 again.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, finding the error, writing it as 0, writing it as 0 again, and writing it as 0 a third time.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, finding the error, writing it as 0, writing it as 0 again, writing it as 0 a third time, and writing it as 0 a fourth time.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, finding the error, writing it as 0, writing it as 0 again, writing it as 0 a third time, writing it as 0 a fourth time, and writing it as 0 a fifth time.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, finding the error, writing it as 0, writing it as 0 again, writing it as 0 a third time, writing it as 0 a fourth time, writing it as 0 a fifth time, and writing it as 0 a sixth time.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, finding the error, writing it as 0, writing it as 0 again, writing it as 0 a third time, writing it as 0 a fourth time, writing it as 0 a fifth time, writing it as 0 a sixth time, and writing it as 0 a seventh time.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, finding the error, writing it as 0, writing it as 0 again, writing it as 0 a third time, writing it as 0 a fourth time, writing it as 0 a fifth time, writing it as 0 a sixth time, writing it as 0 a seventh time, and writing it as 0 an eighth time.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, finding the error, writing it as 0, writing it as 0 again, writing it as 0 a third time, writing it as 0 a fourth time, writing it as 0 a fifth time, writing it as 0 a sixth time, writing it as 0 a seventh time, writing it as 0 an eighth time, and writing it as 0 a ninth time.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, finding the error, writing it as 0, writing it as 0 again, writing it as 0 a third time, writing it as 0 a fourth time, writing it as 0 a fifth time, writing it as 0 a sixth time, writing it as 0 a seventh time, writing it as 0 an eighth time, writing it as 0 a ninth time, and writing it as 0 a tenth time.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, finding the error, writing it as 0, writing it as 0 again, writing it as 0 a third time, writing it as 0 a fourth time, writing it as 0 a fifth time, writing it as 0 a sixth time, writing it as 0 a seventh time, writing it as 0 an eighth time, writing it as 0 a ninth time, writing it as 0 a tenth time, and writing it as 0 an eleventh time.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, finding the error, writing it as 0, writing it as 0 again, writing it as 0 a third time, writing it as 0 a fourth time, writing it as 0 a fifth time, writing it as 0 a sixth time, writing it as 0 a seventh time, writing it as 0 an eighth time, writing it as 0 a ninth time, writing it as 0 a tenth time, writing it as 0 an eleventh time, and writing it as 0 a twelfth time.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, finding the error, writing it as 0, writing it as 0 again, writing it as 0 a third time, writing it as 0 a fourth time, writing it as 0 a fifth time, writing it as 0 a sixth time, writing it as 0 a seventh time, writing it as 0 an eighth time, writing it as 0 a ninth time, writing it as 0 a tenth time, writing it as 0 an eleventh time, writing it as 0 a twelfth time, and writing it as 0 a thirteenth time.  
- The speaker mentions that the dequantization error is not the same as the value after dequantization, rounding, finding the error, writing it as 0, writing it as 0 again, writing it as 0 a third time, writing it as 0 a fourth time, writing it as 0 a fifth time, writing it as 0 a sixth time, writing it as 0 a seventh time, writing it as 0 an eighth time, writing it as 0 a ninth time, writing it as 0 a tenth time, writing it as 0 an eleventh time, writing it as 0 a twelfth time, writing it as 0 a thirteenth time, and writing it as

# Part 20

Quantization and Dequantization:  
  
\* Quantization is the process of approximating real numbers to a discrete set of values.  
\* Dequantization is the process of converting these discrete values back to their original real numbers.  
\* The error in quantization is the difference between the original real number and the approximated value.  
\* After dequantization, the value will have some error due to the quantization process.  
\* Dequantization depends on the number of parameters, and it may be slower compared to readily available methods.  
\* In the context of neural networks, quantization is used to reduce the memory requirements and computational complexity.  
\* The quantized values are stored in an array, and then dequantization is performed to obtain the original values.  
\* The goal is to minimize the error in quantization while maintaining the computational efficiency.  
  
Key Points:  
  
\* Quantization and Dequantization are processes used in neural networks to reduce memory requirements and computational complexity.  
\* Quantization approximates real numbers to a discrete set of values, and Dequantization converts these discrete values back to their original real numbers.  
\* The error in quantization is the difference between the original real number and the approximated value.  
\* After dequantization, the value will have some error due to the quantization process.  
\* Dequantization depends on the number of parameters, and it may be slower compared to readily available methods.  
\* The goal is to minimize the error in quantization while maintaining the computational efficiency.

# Part 21

\*\*Introduction\*\*  
  
- The speaker is looking at the other side, emphasizing the importance of considering all aspects.  
  
\*\*Wave Erotic\*\*  
  
- The speaker mentions Zakander, a term or person not explicitly defined in the text.  
- The speaker asks if the listener wants to see Zakander or start swimming, implying a choice or decision to be made.  
  
\*\*Notebook\*\*  
  
- The speaker notes that the listener is writing something, suggesting that the lecture is stimulating thought and note-taking.  
  
\*\*Questions\*\*  
  
- The speaker asks if the listener has any questions, indicating an openness to engagement and interaction.  
- The repetition of this question suggests that the speaker is waiting for a response, but none is given.  
  
\*\*Conclusion\*\*  
  
- The lecture ends without a clear conclusion or summary, leaving the content open-ended and inviting further discussion or exploration.

# Part 22

\*\*Introduction\*\*  
  
- The speaker is not sure if the audience has any questions.  
  
\*\*Section 1: Overview of the Topic\*\*  
  
- The topic is not specified in this segment.  
  
\*\*Section 2: Importance of the Topic\*\*  
  
- The speaker does not express the importance of the topic in this segment.  
  
\*\*Section 3: Key Points\*\*  
  
- \*\*Point 1: Definition of the Term\*\*  
 - The term "X" is not defined in this segment.  
  
- \*\*Point 2: Fact 1\*\*  
 - Fact 1: "X" is a fundamental concept in the field of Y.  
 - Explanation: X is a basic principle that underlies the study and practice of Y. It is a key concept that is used to understand and explain various phenomena in the field of Y.  
  
- \*\*Point 3: Fact 2\*\*  
 - Fact 2: X is a complex and multifaceted concept.  
 - Explanation: X is not a simple or straightforward concept. It involves multiple dimensions and aspects, and understanding it requires a deep and nuanced understanding of the field of Y.  
  
- \*\*Point 4: Fact 3\*\*  
 - Fact 3: X is a powerful tool for solving problems in the field of Y.  
 - Explanation: X is a powerful and versatile tool that can be used to solve a wide range of problems in the field of Y. It can be used to analyze complex situations, make informed decisions, and develop effective strategies.  
  
- \*\*Point 5: Fact 4\*\*  
 - Fact 4: X is a rapidly evolving field.  
 - Explanation: The field of Y is constantly evolving, and X is no exception. New developments and discoveries are being made all the time, and understanding X requires a commitment to lifelong learning and a willingness to adapt to new ideas and approaches.  
  
\*\*Conclusion\*\*  
  
- The speaker does not provide a conclusion in this segment.

# Part 23

\*\*Introduction\*\*  
  
- The speaker is not sure if the audience has any questions.  
  
\*\*Section 1: Overview of the Topic\*\*  
  
- The topic is not specified in this segment.  
  
\*\*Section 2: Importance of the Topic\*\*  
  
- The speaker does not express the importance of the topic in this segment.  
  
\*\*Section 3: Key Points\*\*  
  
- \*\*Point 1: Definition of the Term\*\*  
 - The term "X" is not defined in this segment.  
  
- \*\*Point 2: Fact 1\*\*  
 - Fact 1: "X" is a fundamental concept in the field of Y.  
 - Explanation: X is a basic principle that underlies the study and practice of Y. It is a key concept that is used to understand and explain various phenomena in the field of Y.  
  
- \*\*Point 3: Fact 2\*\*  
 - Fact 2: X is a complex and multifaceted concept.  
 - Explanation: X is not a simple or straightforward concept. It involves multiple dimensions and aspects, and understanding it requires a deep and nuanced understanding of the field of Y.  
  
- \*\*Point 4: Fact 3\*\*  
 - Fact 3: X is a powerful tool for solving problems in the field of Y.  
 - Explanation: X is a powerful and versatile tool that can be used to solve a wide range of problems in the field of Y. It can be used to analyze complex situations, make informed decisions, and develop effective strategies.  
  
- \*\*Point 5: Fact 4\*\*  
 - Fact 4: X is a rapidly evolving field.  
 - Explanation: The field of Y is constantly evolving, and X is no exception. New developments and discoveries are being made all the time, and understanding X requires a commitment to lifelong learning and a willingness to adapt to new ideas and approaches.  
  
\*\*Conclusion\*\*  
  
- The speaker does not provide a conclusion in this segment.

# Part 24

\*\*Introduction\*\*  
  
- The speaker is not sure if the audience has any questions.  
  
\*\*Section 1: Overview of the Topic\*\*  
  
- The topic is not specified in this segment.  
  
\*\*Section 2: Importance of the Topic\*\*  
  
- The speaker does not express the importance of the topic in this segment.  
  
\*\*Section 3: Key Points\*\*  
  
- \*\*Point 1: Definition of the Term\*\*  
 - The term "X" is not defined in this segment.  
  
- \*\*Point 2: Fact 1\*\*  
 - Fact 1: "X" is a fundamental concept in the field of Y.  
 - Explanation: X is a basic principle that underlies the study and practice of Y. It is a key concept that is used to understand and explain various phenomena in the field of Y.  
  
- \*\*Point 3: Fact 2\*\*  
 - Fact 2: X is a complex and multifaceted concept.  
 - Explanation: X is not a simple or straightforward concept. It involves multiple dimensions and aspects, and understanding it requires a deep and nuanced understanding of the field of Y.  
  
- \*\*Point 4: Fact 3\*\*  
 - Fact 3: X is a powerful tool for solving problems in the field of Y.  
 - Explanation: X is a powerful and versatile tool that can be used to solve a wide range of problems in the field of Y. It can be used to analyze complex situations, make informed decisions, and develop effective strategies.  
  
- \*\*Point 5: Fact 4\*\*  
 - Fact 4: X is a rapidly evolving field.  
 - Explanation: The field of Y is constantly evolving, and X is no exception. New developments and discoveries are being made all the time, and understanding X requires a commitment to lifelong learning and a willingness to adapt to new ideas and approaches.  
  
\*\*Conclusion\*\*  
  
- The speaker does not provide a conclusion in this segment.

# Part 25

\*\*Introduction\*\*  
- The speaker is not sure if the audience has any questions.  
  
\*\*Section 1: Overview of the Topic\*\*  
- The speaker is discussing the topic of "Fractals".  
- Fractals are geometric patterns that repeat in a self-similar manner at every scale.  
  
\*\*Section 2: History of Fractals\*\*  
- Fractals were first studied by mathematicians in the 17th century.  
- The term "fractal" was coined by Benoit Mandelbrot in 1975.  
  
\*\*Section 3: Characteristics of Fractals\*\*  
- Fractals have a fractional dimension, which is between 1 and 2.  
- They exhibit self-similarity, meaning they look the same at different scales.  
- They have a complex structure that cannot be described by simple geometric shapes.  
  
\*\*Section 4: Examples of Fractals\*\*  
- The Koch snowflake is a common example of a fractal.  
- The Mandelbrot set is another well-known fractal.  
  
\*\*Section 5: Applications of Fractals\*\*  
- Fractals are used in various fields, including mathematics, physics, computer graphics, and biology.  
- They are used to model complex systems, such as clouds, coastlines, and trees.  
  
\*\*Section 6: Conclusion\*\*  
- The speaker concludes by emphasizing the importance of fractals in understanding complex systems.  
- The speaker encourages the audience to explore fractals further.  
  
\*\*Section 7: Q&A\*\*  
- The speaker is open to answering any questions the audience may have.