Contents

WEEK 15: FUTURE TRENDS IN LLMS

Multimodal Large Language Models

The evolution of large language models beyond pure text represents one of the most significant trends in AI. Multimodal models can process and generate content across different modalities—text, images, audio, video, and more—opening new possibilities for AI applications that mirror human perception more closely. This shift from language-only models to truly multimodal systems marks a fundamental expansion in how AI systems understand and interact with the world.

The technical foundation of multimodal LLMs involves learning joint representations across modalities. Early approaches often trained separate encoders for different modalities and aligned their outputs in a shared embedding space. Modern architectures like GPT-4V, Gemini, and LLaVA instead train end-to-end models that can process multiple modalities in their native token streams. For instance, an image might be tokenized into visual patches that are interleaved with text tokens, allowing the transformer architecture to attend across both modalities simultaneously.

Vision-language models represent the most mature category of multimodal LLMs. Models like CLIP (Contrastive Language-Image Pre-training) learn to associate images and text through contrastive learning on large datasets of image-caption pairs. The model learns to maximize similarity between matching image-text pairs while minimizing similarity between non-matching pairs. This enables powerful zero-shot capabilities: CLIP can classify images into categories it was never explicitly trained on by comparing image embeddings to text embeddings of category descriptions.

More advanced vision-language models like Flamingo, BLIP-2, and GPT-4V go beyond classification to enable complex visual reasoning and generation. These models can answer detailed questions about images, generate image captions, perform visual reasoning tasks like counting objects or understanding spatial relationships, and even generate images from text descriptions when combined with diffusion models. The key architectural innovation is efficient cross-modal attention mechanisms that allow the language model to attend to relevant parts of an image when generating text.

Audio-language integration represents another frontier. Models like Whisper have demonstrated remarkable speech recognition capabilities by training on massive amounts of multilingual audio data with a simple encoder-decoder architecture. More recently, models are emerging that can understand and generate both speech and text in an integrated manner, enabling natural spoken conversation with AI systems. AudioLM and similar models can even generate realistic speech, music, or other audio from text descriptions.

Video understanding poses additional challenges beyond static images due to temporal dynamics and the massive amount of data in video streams. Video-language models must learn to identify and track objects across frames, understand actions and events unfolding over time, recognize cause-and-effect relationships, and summarize or answer questions about video content. Approaches include extending vision transformers with temporal attention mechanisms, using 3D convolutions to capture spatio-temporal features, and employing hierarchical processing where frames are first processed individually then aggregated temporally.

The implications of multimodal LLMs are profound. They enable more natural human-AI interaction through multiple channels, support accessibility applications like image descriptions for the visually impaired or automatic captioning, power augmented reality applications that understand and respond to the visual environment, facilitate content creation spanning text, images, and video, and enable AI systems that can learn from and operate in the real world rather than just text.

Efficient Training and Fine-Tuning

As language models grow larger, the computational costs and technical challenges of training and adapting them have become increasingly prohibitive. A single training run for a frontier model can cost millions of dollars and consume energy equivalent to the annual usage of hundreds of homes. This has sparked intense research into more efficient training paradigms and fine-tuning methods that make these powerful models more accessible and sustainable.

Parameter-Efficient Fine-Tuning (PEFT) methods aim to adapt large pre-trained models to specific tasks while updating only a small fraction of the parameters. This dramatically reduces memory requirements, training time, and compute costs compared to full fine-tuning. The key insight is that the representations learned during pre-training capture general knowledge, and task-specific adaptation requires only modest adjustments in a much lower-dimensional space.

Low-Rank Adaptation (LoRA) represents a breakthrough in parameter-efficient fine-tuning. The method is based on the hypothesis that the weight updates during fine-tuning have low "intrinsic rank." Instead of updating the full weight matrix $W \in \mathbb{R}^{d \times k}$, LoRA represents the update as a product of two low-rank matrices: $\Delta W = BA$ where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$ with $r \ll \min(d, k)$. The forward pass becomes:

$$h = Wx + BAx$$

where only A and B are trainable. For typical values like $r = 8$ and model dimensions in the thousands, this reduces trainable parameters by orders of magnitude. LoRA adapters can be trained, saved separately, and swapped in and out efficiently, enabling a single base model to serve multiple specialized tasks.

QLoRA (Quantized LoRA) extends LoRA to enable fine-tuning of even larger models on consumer hardware. The key innovations include quantizing the pre-trained model weights to 4-bit precision using specialized NormalFloat data types optimized for neural network weights, keeping LoRA adapter parameters in higher precision for accuracy, and employing memory-efficient techniques like paged optimizers and gradient checkpointing. This makes it possible to fine-tune models with up to 65B parameters on a single consumer GPU.

Prefix Tuning and Prompt Tuning take a different approach by prepending learnable "soft prompts" (continuous vectors) to the input. Unlike discrete text prompts, these vectors are optimized via backpropagation. The model parameters remain frozen while only the soft prompts are updated. Prefix tuning adds trainable tokens to both the input and every layer of the transformer, while prompt tuning adds them only to the input. These methods often match or exceed the performance of full fine-tuning while updating fewer than 0.1% of parameters.

Adapter Layers insert small trainable modules between the frozen layers of a pre-trained model. A typical adapter consists of a down-projection to a low-dimensional bottleneck, a non-linearity, and an up-projection back to the original dimension, often with a residual connection. Multiple adapters can be composed or selected based on the task, and mixing adapters trained on different tasks can sometimes yield better performance than training on combined data.

Mixed Precision Training uses lower-precision arithmetic (like 16-bit floats instead of 32-bit) to reduce memory usage and increase training speed, while maintaining numerical stability through techniques like loss scaling and mixed-precision optimizer states. BFloat16 (Brain Floating Point) has become particularly popular as it maintains the range of FP32 while using only 16 bits, making it more robust to precision issues than FP16.

Mixture of Experts (MoE) architectures increase model capacity without proportionally increasing computation by having different "expert" sub-networks that specialize in different types of inputs. A gating network determines which experts to activate for each input, typically routing to only a small subset (e.g., 2 out of 64 experts). This allows models to scale to trillions of parameters while keeping computational cost manageable. Challenges include load balancing across experts, training instability, and increased infrastructure complexity.

Flash Attention and Memory-Efficient Transformers address the quadratic memory complexity of standard attention mechanisms. Flash Attention reorganizes attention computation to minimize memory reads/writes and avoid materializing the full attention matrix, achieving 2-4x speedups and enabling much longer context lengths. Techniques like ring attention enable distributing attention computation across multiple devices, supporting contexts of millions of tokens.

Alignment and RLHF

Ensuring that large language models behave in ways aligned with human values and intentions has become a critical challenge. Raw pre-trained models, while knowledgeable, often generate outputs that are unhelpful, harmful, or disagree with human preferences in various ways. Alignment research focuses on techniques to make models more helpful, harmless, and honest.

Reinforcement Learning from Human Feedback (RLHF) has emerged as the dominant paradigm for alignment. The process involves several stages: First, the model is pre-trained on a large text corpus using standard language modeling objectives. Second, human labelers rank or rate model outputs for various prompts based on quality criteria like helpfulness, truthfulness, and harmlessness. Third, these human preferences are used to train a reward model that predicts human judgments. Fourth, the language model is fine-tuned using reinforcement learning (typically Proximal Policy Optimization or PPO) to maximize the reward model's score while maintaining similarity to the original model through KL divergence penalties.

The reward model typically takes a prompt and a completion as input and outputs a scalar reward. It's usually initialized from the pre-trained language model and fine-tuned on the human preference data formulated as pairwise comparisons: given outputs y_1 and y_2 for prompt x, if humans prefer y_1, the training loss encourages $r(x, y_1) > r(x, y_2)$. The Bradley-Terry model provides a probabilistic framework for this:

$$P(y_1 \succ y_2|x) = \frac{\exp(r(x, y_1))}{\exp(r(x, y_1)) + \exp(r(x, y_2))}$$

The RL fine-tuning phase uses the reward model to provide learning signal. The policy (the language model generating text) is updated to maximize expected reward:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim D, y\sim\pi_\theta(\text{ů}|x)}[r(x, y)] - \beta \cdot D_{KL}[\pi_\theta||\pi_{ref}]$$

where π_{ref} is the reference model (typically the supervised fine-tuned model before RL), and β controls how much the policy can deviate. This KL penalty prevents the model from exploiting the reward model by generating unusual outputs that happen to score highly.

Direct Preference Optimization (DPO) is a simpler alternative to RLHF that eliminates the separate reward model and RL training. DPO directly optimizes the policy on preference data using a closed-form objective derived from the RLHF objective. This makes training more stable and efficient while often achieving comparable results. The key insight is that the optimal policy under the RLHF objective has a closed form in terms of the reward function, which can be inverted to directly optimize preferences.

Constitutional AI aims to make the alignment process more scalable and principle-driven. Instead of relying solely on human feedback, the approach uses a set of principles or "constitution" (rules about desired behavior) to generate self-critiques and revisions. The model critiques its own outputs for violations of principles, revises them accordingly, and these improved outputs are used for supervised fine-tuning. This self-improvement process can be iterated and combined with RLHF on the revised outputs.

Alignment Challenges remain numerous: reward hacking where models exploit flaws in the reward model rather than behaving genuinely well, distributional shift between training and deployment, specification gaming where models satisfy the letter but not spirit of objectives, value alignment complexity as human values are nuanced, contextual, and sometimes contradictory, and scalability of human feedback as models become more capable.

Interpretability and Control are increasingly important for alignment. Mechanistic interpretability aims to understand what computations models perform internally, potentially enabling detection and correction of problematic behaviors. Techniques include probing for specific concepts in activation spaces, identifying and intervening on circuits responsible for particular capabilities, and using sparse autoencoders to find interpretable features in activations.

Emerging Capabilities and Scaling Laws

As models grow larger and training progresses, they exhibit emergent abilities—capabilities not present in smaller models that suddenly appear at certain scales. Understanding these phenomena and the scaling laws that govern model behavior is crucial for predicting future developments and allocating research resources effectively.

Scaling Laws describe how model performance improves with scale, typically measured in terms of parameters N, dataset size D, and compute C. The seminal work by Kaplan et al. (2020) showed that loss follows power-law relationships with each axis:

$$L(N) \propto N^{-\alpha}$$

$$L(D) \propto D^{-\beta}$$

$$L(C) \propto C^{-\gamma}$$

These scaling laws enable predicting model performance before training, optimal allocation of compute budget between model size and training length, and estimation of returns on investment in larger models or datasets. The Chinchilla scaling laws refined these, showing that previous models were undertrained—optimal compute allocation requires increasing dataset size proportionally with parameters, not focusing solely on parameter count.

Emergent Abilities are capabilities that appear suddenly at certain scales rather than gradually improving. Examples include multi-step arithmetic, answering questions requiring world knowledge, translating between languages not seen during training, and certain forms of logical reasoning. The emergence is often sharp: performance may be near random up to a certain scale then jump dramatically. This unpredictability makes it challenging to anticipate what capabilities future models will have.

In-Context Learning exemplifies emergent behavior. Smaller models struggle to learn from examples provided in the prompt, while sufficiently large models can perform few-shot learning effectively, adapting to new tasks without parameter updates based solely on examples in the context. This ability unlocks enormous flexibility, allowing a single model to perform countless tasks by just changing the prompt.

Chain-of-Thought Reasoning is another emergent capability where models generate intermediate reasoning steps before answering, dramatically improving performance on complex reasoning tasks. This ability appears only in sufficiently large models and can be elicited through prompting techniques like "Let's think step by step." The mechanism isn't fully understood but demonstrates that models can perform sequential, compositional reasoning when given the opportunity to generate intermediate steps.

Grokking describes the phenomenon where models suddenly generalize perfectly after prolonged training despite already achieving low training loss. The model first memorizes the training data, then after continued training on the same data, abruptly learns the underlying pattern or algorithm. This challenges conventional understanding of overfitting and suggests models may continue improving with training even after apparent convergence.

Transfer and Multi-Task Learning improves with scale in interesting ways. Larger models transfer knowledge more effectively across tasks and handle multitask learning better. They can leverage shared structure and common knowledge across diverse tasks, sometimes showing "positive transfer" where training on one task improves performance on related tasks more than the compute spent would predict.

Inverse Scaling phenomena are cases where larger models perform worse on certain tasks. These often involve spurious correlations or biases that larger models pick up more strongly, reliance on memorization rather than reasoning, or failure modes that are less obvious to models trained to predict next tokens. Understanding these helps identify limitations and design better training procedures.

Future Predictions based on scaling laws and trends suggest continued performance improvements with scale, though at diminishing returns. Key questions include whether scaling alone will achieve artificial general intelligence, what fundamental capabilities may remain beyond pure scaling, and whether we're approaching practical limits in terms of data availability, compute costs, or environmental impact. Alternative paradigms like test-time compute scaling (giving models more "thinking time" for harder problems) may become increasingly important.

Specialized Architectures and Innovations

While the transformer architecture has dominated recent LLM development, active research explores modifications and alternatives that may offer advantages for specific applications or overcome current limitations.

State Space Models (SSMs) like S4 (Structured State Space Sequence Model) and Mamba offer an alternative to attention mechanisms. These models are inspired by classical state space models from control theory and process sequences through recurrent state updates: $h_t = Ah_{t-1} + Bx_t$ and $y_t = Ch_t + Dx_t$. The key innovation of modern SSMs is using structured matrices (like diagonal plus low-rank) for A that enable efficient computation. Mamba specifically uses selective state spaces where the model can choose which information to retain based on input, combining the efficiency of linear recurrence with content-based reasoning.

SSMs have several advantages: they scale linearly rather than quadratically with sequence length in both time and memory, they can be computed efficiently in parallel during training via convolution formulations, and they may be better suited than transformers for certain modalities like audio or continuous control. However, they haven't yet matched transformers' performance across the breadth of language tasks, particularly those requiring long-range reasoning.

Mixture of Depths architectures dynamically allocate compute across the sequence. Rather than processing every token with equal computational depth, the model learns which tokens need more processing and routes them through more layers. This can dramatically improve efficiency by avoiding wasted computation on simple tokens while focusing resources on challenging portions of the input.

Sparse Transformers modify the attention mechanism to attend over only a subset of positions rather than all positions, reducing complexity from $O(n^2)$ to $O(n\sqrt{n})$ or $O(n \log n)$ depending on the sparsity pattern. Patterns include local attention (attending only to nearby tokens), strided attention (attending at fixed intervals), and learned sparsity where the model learns which positions to attend to. While these enable processing longer sequences, they risk missing important long-range dependencies.

Retrieval-Augmented Architectures integrate retrieval directly into the model architecture rather than as a separate pipeline. RETRO (Retrieval-Enhanced Transformer) chunks the input, retrieves relevant documents for each chunk, and integrates retrieved information through cross-attention in intermediate layers. This allows the model to leverage a vast external memory while keeping parameters manageable.

Tool-Using Models integrate the ability to call external tools directly into the architecture. Models like Toolformer learn when and how to call APIs, search engines, calculators, or other tools by training on data augmented with tool calls. The model generates special tokens indicating tool calls, the tool is executed, and results are incorporated back into the generation. This extends model capabilities beyond what can be captured in parameters alone.

Neurosymbolic Approaches combine neural networks with symbolic reasoning systems. These hybird architectures might use neural networks for perception and pattern recognition while using symbolic methods for logical reasoning, planning, or knowledge representation. Examples include neural theorem provers, models that generate and execute code to solve problems, and systems that maintain explicit symbolic knowledge graphs alongside learned representations.

Societal Impact and Responsible AI

The rapid advancement of LLMs raises profound questions about their societal impact and the responsibility of those developing and deploying them. Understanding and addressing these concerns is essential for ensuring these technologies benefit humanity.

Misinformation and Deepfakes: LLMs can generate convincing but false content at scale, potentially accelerating the spread of misinformation. Text generation can create fake news articles or social media posts, while multimodal models enable deepfake images, audio, and video. Mitigation approaches include watermarking generated content using cryptographic signatures or subtle patterns detectable by specialized models, developing robust detection methods for synthetic content, implementing usage policies and monitoring for platforms that provide LLM access, and user education about AI-generated content. However, the arms race between generation and detection continues, with no perfect solution currently available.

Bias and Fairness: LLMs learn from internet text and other data that contains societal biases regarding race, gender, religion, and other attributes. These biases can manifest in model outputs through stereotypical associations, unequal performance across demographic groups, and perpetuation or amplification of harmful stereotypes. Addressing bias requires diverse teams in AI development, careful curation and auditing of training data, bias measurement and mitigation techniques, and ongoing monitoring of deployed systems. However, completely eliminating bias while maintaining model performance remains an open challenge.

Privacy Concerns: LLMs may memorize and reproduce training data, potentially exposing sensitive information. Concerns include generation of personal information from training data (names, addresses, etc.), potential for extracting proprietary or confidential information, and privacy implications of queries revealing user information. Approaches to privacy preservation include differential privacy during training, which adds calibrated noise to prevent memorization, data filtering to remove personal information before training, and query filtering and output monitoring in deployed systems. The European Union's GDPR "right to be forgotten" creates additional challenges for models that have memorized personal data.

Environmental Impact: Training large language models consumes enormous amounts of energy and has measurable carbon footprints. A single training run for a large model can emit as much CO2 as several transatlantic flights. Sustainability considerations include measuring and reporting environmental impact of training runs, developing more efficient training methods and architectures, using renewable energy for compute infrastructure, and balancing model capability improvements against environmental costs. The AI community increasingly recognizes responsibility to develop sustainable practices.

Economic Disruption: LLMs may automate cognitive work previously requiring human expertise, raising questions about technological unemployment in fields like content writing, customer service, and programming, economic inequality as those with AI access gain productivity advantages, and need for education and retraining as job requirements change. While technology has historically created new jobs while displacing others, the pace and breadth of AI-driven change may be unprecedented.

Governance and Regulation: As LLMs become more capable and widely deployed, questions of governance become urgent. Key issues include what safety requirements should apply to AI systems, how to ensure accountability when AI systems cause harm, what transparency obligations should exist (model cards, training data disclosure), international coordination on AI regulation, and balancing innovation encouragement with risk mitigation. Different jurisdictions are taking varied approaches, from the EU's relatively comprehensive AI Act to more sector-specific regulations elsewhere.

Beneficial Applications: Alongside risks, LLMs offer enormous potential benefits including democratizing access to information and expertise through conversational AI, accelerating scientific research through literature analysis and hypothesis generation, improving education through personalized tutoring and learning assistance, enhancing healthcare through medical information synthesis and decision support, and expanding accessibility for people with disabilities through improved assistive technologies. Realizing these benefits while managing risks requires thoughtful development, deployment, and governance practices that prioritize human welfare.

Learning Objectives

- Understand emerging trends in multimodal AI and cross-modal learning
- Master efficient training and fine-tuning techniques like LoRA and QLoRA
- Comprehend alignment challenges and RLHF methodology
- Analyze scaling laws and emergent capabilities in large models
- Evaluate societal impacts and responsible AI considerations