AI PRODUCT MANAGEMENT 2 · DETAILED COURSE

Generative AI and Large Language Models

What generative models are, how LLMs work, and the practical limits that shape product decisions.

Chapter 12 of 26 · 30 topics · Edition 2026 · skillnavora.in

CHAPTER 12

Generative AI and Large Language Models

What generative models are, how LLMs work, and the practical limits that shape product decisions.

Introduction to Generative AI

Models that generate new content from learned patterns, across text, image, audio, video and code.

Generative AI models create new content such as text, images, audio, video and code, based on patterns learned from data. Its outputs are probabilistic, so the same prompt can give different answers.

Generative AI Architecture

The stack from foundation model to prompts, retrieval, tools, guardrails and application layer.

A typical stack includes the foundation model, prompts, retrieval of company data, tools the model can call, guardrails and the application layer that the user sees. Each layer can fail, so each needs monitoring.

Foundation Models

Large models pre-trained on broad data and adapted to many tasks.

Foundation models are large models pre-trained on broad data and adapted to many tasks. Products build on them through prompting, retrieval and fine-tuning rather than training from scratch.

Large Language Models (LLMs)

Transformer models trained to predict the next token, which gives them broad language ability.

LLMs are transformer models trained to predict the next token in text. That simple objective, at scale and with alignment training, gives broad language ability, including summarising and coding.

GPT Architecture

A decoder-only transformer trained to predict the next token, scaled up and then aligned with instructions and feedback.

GPT models are decoder-only transformers trained to predict the next token, then aligned with instructions and human feedback. The name refers to a family of OpenAI models; the architecture idea is shared widely.

Claude Model Ecosystem

Anthropic's model family across capability and cost tiers, with long context, tool use and strong coding and writing; check current model names in its documentation.

Anthropic offers a family of Claude models across capability and cost tiers, with long context windows and strong performance on writing and coding. Model names and prices change, so check the current documentation before choosing.

Gemini Model Ecosystem

Google's multimodal model family for text, image, audio, video and code, offered through the Gemini API and Vertex AI.

Google's Gemini models handle text, images, audio, video and code, and are available through the Gemini API and Vertex AI. Their multimodal inputs suit document and video workflows.

Open-Source LLMs

Models with downloadable weights, such as Llama, Mistral and Qwen, that you can host and tune, with licence terms to check.

Models with downloadable weights, such as Llama, Mistral and Qwen, can be hosted and tuned by the company. Check the licence, because some restrict commercial use or large-scale deployment.

Transformer Architecture

Stacked attention and feed-forward layers with embeddings and positional information.

A transformer stacks attention and feed-forward layers, with embeddings and positional information. It processes all tokens in parallel, which makes training at scale practical.

Tokens and Tokenisation

Text is split into sub-word tokens; cost, limits and speed are measured in tokens.

Text is split into sub-word tokens. Cost, context limits and speed are measured in tokens, so a 1,000-word document is roughly 1,300 to 1,500 tokens in English, and more in some other languages.

Context Windows

The maximum tokens a model can consider at once, covering prompt and response.

The context window is the maximum number of tokens the model can consider at once, covering the prompt and the response. Long documents may need retrieval rather than being pasted in full.

Embeddings

Numeric vectors that capture meaning so similar items sit close together.

Embeddings are numeric vectors that capture meaning, so similar texts sit close together. They power semantic search, recommendations and clustering.

Pre-Training

Initial large-scale training on broad data to learn language and world knowledge.

Pre-training trains the model on very large general data to learn language and world knowledge. It is expensive and done by model providers, not by most product teams.

Instruction Tuning

Training on instruction-response pairs so the model follows directions.

Instruction tuning trains the model on instruction and response pairs so it follows directions. It turns a text predictor into an assistant.

Fine-Tuning

Further training on task or domain data to change behaviour, style or accuracy.

Fine-tuning trains a model further on task or domain data to change behaviour, style or accuracy on a narrow task. It is often unnecessary when good prompting and retrieval work.

RLHF

Reinforcement learning from human feedback: human preferences train a reward signal to align outputs.

Reinforcement learning from human feedback trains a reward model on human preferences and optimises the model against it. It makes outputs more helpful and safer, but it can also make them overly cautious.

Reasoning Models

Models that spend extra compute thinking through steps before answering, improving hard tasks at higher cost and latency.

Reasoning models spend extra computation working through steps before answering. They improve on hard maths, planning and coding, but cost more and respond more slowly.

Multimodal AI

Models that handle several input or output types, such as text, images and audio.

Multimodal models accept or produce several types of data, such as text, images and audio. A product can let users photograph a receipt and ask a question about it.

Text Generation

Producing written content, summaries, answers, code and structured data.

Text generation produces summaries, drafts, answers, code and structured data. Quality depends on the prompt, the context and the evaluation applied to the output.

Image Generation

Creating images from prompts using diffusion or similar models.

Image models create pictures from prompts, often using diffusion methods. Commercial use depends on the provider's licence and on the rights of training data, so check terms.

Audio Generation

Generating speech, music or sound effects.

Audio models generate speech, music and sound effects. Voice cloning raises consent and misuse risks that the product must control.

Video Generation

Creating video clips from text or images; quality, length and cost are still constraints.

Video models create clips from text or images. Quality, length and cost remain constraints, so products usually limit duration and review outputs before publishing.

Speech-to-Text

Transcribing audio into text; accuracy depends on accents, noise and vocabulary.

Speech recognition converts audio to text. Accuracy depends on accents, background noise and domain vocabulary, so test with real recordings from your users.

Text-to-Speech

Converting text into natural-sounding voice with controllable style.

Text-to-speech generates natural voice from text, with control over style and pace. Disclose when users hear AI-generated voices.

Structured Outputs

Constraining model output to a schema such as JSON so software can use it reliably.

Structured outputs constrain the model to produce data in a schema such as JSON, so software can use the result reliably. Validate the output in code as well.

Function Calling

The model returns a structured request to call a named function, which your code runs.

The model returns a structured request to call a named function with arguments, and the application runs it. This lets models query databases or book appointments safely.

Model Selection and Benchmarking

Choosing a model by testing it on your own task data, not only public leaderboards.

Choose a model by testing it on your own tasks, measuring quality, latency and cost. Public leaderboards are a hint, not proof of fit for your use case.

Model Latency

Time to first token and total time; shaped by model size, output length and load.

Latency is shaped by model size, output length and load. Time to first token affects perceived speed, so streaming responses improves user experience.

Model Cost and Token Economics

Price per input and output token, multiplied by volume; shorter prompts, caching and smaller models cut cost.

Cost equals tokens processed multiplied by price per token. Shorter prompts, caching, smaller models for easy tasks and batching reduce cost. Track cost per successful task, not per call.

LLM Limitations and Hallucinations

Models can state false things confidently, lack current knowledge, and vary between runs; design for verification.

Models can state false information confidently, lack current knowledge and vary between runs. Design for verification: show sources, check high-stakes outputs and allow users to correct errors.

Summary of terms

Term	Meaning
Introduction to Generative AI	Models that generate new content from learned patterns, across text, image, audio, video and code.
Generative AI Architecture	The stack from foundation model to prompts, retrieval, tools, guardrails and application layer.
Foundation Models	Large models pre-trained on broad data and adapted to many tasks.
Large Language Models (LLMs)	Transformer models trained to predict the next token, which gives them broad language ability.
GPT Architecture	A decoder-only transformer trained to predict the next token, scaled up and then aligned with instructions and feedback.
Claude Model Ecosystem	Anthropic's model family across capability and cost tiers, with long context, tool use and strong coding and writing; check current model names in its documentation.
Gemini Model Ecosystem	Google's multimodal model family for text, image, audio, video and code, offered through the Gemini API and Vertex AI.
Open-Source LLMs	Models with downloadable weights, such as Llama, Mistral and Qwen, that you can host and tune, with licence terms to check.
Transformer Architecture	Stacked attention and feed-forward layers with embeddings and positional information.
Tokens and Tokenisation	Text is split into sub-word tokens; cost, limits and speed are measured in tokens.
Context Windows	The maximum tokens a model can consider at once, covering prompt and response.
Embeddings	Numeric vectors that capture meaning so similar items sit close together.
Pre-Training	Initial large-scale training on broad data to learn language and world knowledge.
Instruction Tuning	Training on instruction-response pairs so the model follows directions.
Fine-Tuning	Further training on task or domain data to change behaviour, style or accuracy.
RLHF	Reinforcement learning from human feedback: human preferences train a reward signal to align outputs.

Reasoning Models	Models that spend extra compute thinking through steps before answering, improving hard tasks at higher cost and latency.
Multimodal AI	Models that handle several input or output types, such as text, images and audio.
Text Generation	Producing written content, summaries, answers, code and structured data.
Image Generation	Creating images from prompts using diffusion or similar models.
Audio Generation	Generating speech, music or sound effects.
Video Generation	Creating video clips from text or images; quality, length and cost are still constraints.
Speech-to-Text	Transcribing audio into text; accuracy depends on accents, noise and vocabulary.
Text-to-Speech	Converting text into natural-sounding voice with controllable style.
Structured Outputs	Constraining model output to a schema such as JSON so software can use it reliably.
Function Calling	The model returns a structured request to call a named function, which your code runs.
Model Selection and Benchmarking	Choosing a model by testing it on your own task data, not only public leaderboards.
Model Latency	Time to first token and total time; shaped by model size, output length and load.
Model Cost and Token Economics	Price per input and output token, multiplied by volume; shorter prompts, caching and smaller models cut cost.
LLM Limitations and Hallucinations	Models can state false things confidently, lack current knowledge, and vary between runs; design for verification.

Practice exercise: Compare two models on twenty of your own tasks, noting quality, latency and cost per task.

Self-check

1. Explain Introduction to Generative AI in two sentences to a non-technical stakeholder, then say what you would measure for it.
2. Explain RLHF in two sentences to a non-technical stakeholder, then say what you would measure for it.
3. Explain LLM Limitations and Hallucinations in two sentences to a non-technical stakeholder, then say what you would measure for it.

Related SkillNavora material

- Machine Learning Products for PMs : ML concepts, metrics and lifecycle.
- AI Tutor : Ask questions on any topic.
- Study Library : Every lesson, class and PDF by programme.

© SkillNavora · skillnavora.in · Facts and tools change; check official sites before relying on a figure.

ORIGINAL CONTENT

Original AI Product Management content

The AI Product Management programme already on SkillNavora, included unchanged so the full course is in one place.

Quick links Study Library (all material) Learning sites for PM AI builder tools (how-to) Webhooks & Playwright MCP Webhook URLs and keys (start here) AI Skills Academy AI Skills Map Context engineering (PDF) AI skills & templates (PDF) Orchestration (PDF) PRD library (120 ideas) AI PRD Writer AI PM Toolkit AI Evals (PDF) Browser automation (PDF) AI second brain (PDF) PM handbook Top 100 PM questions AI product sense Q&A (86) PM practice test Practice Lab Tool tutorials AI Agents & MCP for PMs

PRD documents

PRD library: template, framing guide and 120 PRD ideas

Learn how top PMs frame a PRD, download the free template, and turn any of 120 ideas across 15 industries into a full PRD with AI.

Open the PRD library ✨ AI PRD Writer PRD template (Word) AI Evals for PMs (PDF)

Context window

Context window: what a model can see at once

The context window is the most tokens a model can consider in one request. It covers your prompt, any documents you paste, the conversation so far and the reply. Anything outside it is invisible to the model. For PMs: a longer window fits more material but costs more and can slow responses; put the most important facts near the instructions; use retrieval (RAG) rather than pasting whole libraries; and test answers on long inputs, where quality can drop.

Context engineering course Context windows in the AI PM 2 course

Tokens and context in the Grokbot course

AI-First Product Management

Resources for Product Managers

The PM Master Handbook, AI and product tools with their official websites, interview practice, a frameworks cheat sheet, a PRD template and free courses.

Open the PM handbook library: all 68 PDFs ↗ PM Glossary Handbook Tokens, Pricing and Tokenization Vibe Coding addendum

PM Master Handbook by SkillNavora
23 modules and 620+ pages: 12 Product Core modules, an 11-class AI Tools Lab, frameworks, case studies, projects and interview prep.

Open the interactive handbook ↗ Complete handbook PDF (620+ pages, 23 MB) ↗

Product Core modules

1Fundamentals of Product Management*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
2Product Strategy and Strategic Thinking*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
3Problem Discovery and Market Research*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
4Problem Framing and Prioritization*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
5UX and Design Collaboration*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
6Effective Communication and Stakeholder Management*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
7Product Analytics and Metrics*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
8Product Launch, Growth, and Monetization*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
9Tech 101 and System Design for PMs*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
10LLMs, RAGs, and AI Foundations*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
11Vibe Coding for Product Managers*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides
12Agentic AI and AI Systems*Handbook chapter, detailed PDF, handbook and class slides*PDF Detailed PDF Handbook Class Slides

AI Tools Lab

13AI Tools Landscape and Your Starter StackPDF ↗ 14Claude for Daily PM WorkPDF ↗ 15Claude Code: Basic to AdvancedPDF ↗ 16Lovable: Basic to AdvancedPDF ↗ 17Replit and Other AI App BuildersPDF ↗ 18Figma and AI Design ToolsPDF ↗ 19Notion, PRD Tools and AI DocumentationPDF ↗ 20Automation with Zapier, Make and n8nPDF ↗ 21AI Chatbots for BusinessPDF ↗ 22Voice Agents for BusinessesPDF ↗ 23Learning Paths, Courses and CommunitiesPDF ↗

Business AI Stack (chapters 24 to 31)

24The business AI stack: tools and where to learn themPDF ↗ 25Next.js: customer-facing dashboardsPDF ↗ 26Zoho CRM: managing customers and leadsPDF ↗ 27WhatsApp Business Platform: customer messagingPDF ↗ 28Deployment with Vercel and AWSPDF ↗ 29Jira: tracking product and development workPDF ↗ 30Meta Business Suite, social media marketing and Google AdsPDF ↗ 31A one-page website that generates leadsPDF ↗

Complete AI tool guides (15)

1n8n: workflows, Gmail and an AI reply agentPDF ↗ 2Make: scenarios, routers and lead routingPDF ↗ 3Zapier: invoices from Gmail to Drive and a sheetPDF ↗ 4Webhooks: how apps talk in real timePDF ↗ 5Replit: one app for the web and a phonePDF ↗ 6Lovable: a job application trackerPDF ↗ 7Figma: components, auto layout and handoffPDF ↗ 8Claude Code: an AI coding agent in your projectPDF ↗ 9MCP: connecting AI assistants to your toolsPDF ↗ 10LLMs: how they work and how to call themPDF ↗ 11RAG: answers from your own documentsPDF ↗ 12AI agents: a job-search agent with approvalsPDF

↗ 13Agentic AI: planning, guardrails and teamsPDF ↗ 14Generative AI: uses, limits and a weekly reportPDF ↗ 15Other AI tools: choosing one per jobPDF ↗

Complete AI tool guides (one PDF, 30 pages) ↗

Libraries

★Framework Library*29 frameworks with worked examples and templates*PDF ↗ ★Case Study Library*12 product case studies*PDF ↗ ★Practical Projects and Portfolio*5 end-to-end portfolio projects*PDF ↗ ★Interview Handbook*100+ PM interview questions with model answers*PDF ↗ ★AI Tools Lab*Every AI tools class with step-by-step labs*PDF ↗

PM interview practice (free, on the original sites)

Hand-picked public question banks and guides from other PM communities. Each link opens the original page in a new tab.

PM frameworks cheat sheet

Framework	Use it to	How it works
RICE	Prioritise a backlog	Reach × Impact × Confidence ÷ Effort. Score each idea, then sanity-check the ranking.
Kano	Decide which features delight	Classify features as Must-be, Performance, Attractive, Indifferent or Reverse using a short survey.
MoSCoW	Scope a release	Must have, Should have, Could have, Won't have (this time).
Jobs-to-be-done	Understand motivation	When [situation], I want to [motivation], so I can [outcome].
AARRR (Pirate metrics)	Map the growth funnel	Acquisition, Activation, Retention, Referral, Revenue.
HEART	Measure user experience	Happiness, Engagement, Adoption, Retention, Task success, each with goals, signals and metrics.
North Star metric	Align the team	One metric that captures the value users get, supported by input metrics the team can move.
Five whys	Find the root cause	Ask 'why?' repeatedly until you reach a cause you can act on.
CIRCLES	Answer product design questions	Comprehend, Identify users, Report needs, Cut through prioritisation, List solutions, Evaluate trade-offs, Summarise.

PRD template

PRD: [Feature name] Owner: [Name] · Status: Draft · Last updated: [Date] 1. Problem Who has the problem, what it is, and the evidence (data, interviews, tickets). 2. Goal and success metrics Primary metric (with baseline and target), secondary metrics, guardrails. 3. Users and use cases Target segment, top 3 user stories: "As a [user], I want [action] so that [benefit]." 4. Solution What we'll build (link designs), what's out of scope, open questions. 5. AI behaviour (if applicable) Inputs, model, prompt/approach, what it must never do, human review, evaluation set and pass criteria, cost per request. 6. Launch plan Experiment design, rollout stages, support and comms, rollback criteria. 7. Risks and dependencies Technical, legal/privacy, operational; owners and mitigations.

Free courses