AI FOUNDATIONS SERIES · STUDENT LEARNING GUIDE

Fundamentals of

Artificial Intelligence & Machine Learning

Your self-study guide from “What is AI?” to foundation models, RAG and the AWS AI services. Learn at your own pace, try things yourself, and test what you know with 88 questions answered from basic to advanced, 30 quiz questions and 32 practice challenges.

Inside this guide	
6 modules	Introduction · AI/ML/DL/GenAI · ML fundamentals · Deep learning · Generative AI · AWS services
Your questions, answered	The questions learners most often have, tagged BASIC, INTERMEDIATE or ADVANCED
Learn by doing	Try-it-yourself tasks, myth-busters, tick-box quizzes, a final challenge and a self-check list
Diagrams	11 illustrations, including forward/reverse diffusion and the RAG flow

Prepared by **Pushpjeet Cholkar** | Cloud · AI/ML

www.pushpjeet.com

Welcome: how to use this guide

This guide is written for **you, the learner**. You don't need any background in AI, coding or maths. Work through the six modules in order; each one builds on the last.

Every module follows the same rhythm: **learn the idea → see it → try it → ask your questions → check yourself.**

When you see...	What to do
📌 **What you will learn**	Read this first so you know what to look for
💡 **Think of it like...**	An everyday comparison that makes the idea easy to remember
✏️ **Try it yourself**	A short task (2–10 minutes). Grab a pen or open a browser and actually do it. This is where the learning sticks
⚠️ **Myth-buster**	Things many people believe that are not true
🎯 **Exam tip**	Useful if you plan to take an AWS certification such as AWS Certified AI Practitioner
🧠 **Remember this**	The few points from the module worth memorising
BASIC	Questions most beginners have. Start here
INTERMEDIATE	The "but why?" questions once the basics make sense
ADVANCED	Deeper questions: great for interviews and projects. Fine to skip on your first read
🤔 **Guess first, then read**	Cover the answer, make your guess, then check. Guessing first helps you remember
✅ **Quick quiz**	Click the boxes to tick your answers, then use the link to check them in the answer key

🎯 Study tips

- Study in short sessions: one module at a time, then take a break.
- Say each definition out loud in your own words. If you can explain it simply, you understand it.
- Don't peek at the answer key before you have tried the quiz.
- Come back to the ADVANCED questions after you finish all six modules.

Your learning path

About **4 hours** of study in total. Tick each module when you finish it.

	Module	Time	You are ready to move on when you can...
☐	1. Introduction	15 min	list five places you already use AI
☐	2. AI, ML, DL, GenAI	35 min	explain the four terms and how they fit inside each other
☐	3. ML fundamentals	50 min	describe how a model is trained and used, and the types of data
☐	4. Deep learning	35 min	explain how a neural network learns, and name CV and NLP tasks
☐	5. Generative AI	60 min	explain foundation models, tokens, embeddings, diffusion, RAG and fine-tuning
☐	6. AWS services	40 min	pick the right AWS service for a problem and name the cost factors
☐	Final challenge	20 min	score 10 or more out of 12 on the scenarios

Contents

Module 1: Introduction

📌 **What you will learn (about 15 minutes)**

- what this guide covers and why it matters to you
- recognise AI you already use every day
- explain why businesses are investing in AI, ML and generative AI
- name the AWS service families you will meet later in the course

What this course is about

In this course you will learn the **foundations of machine learning (ML) and artificial intelligence (AI)**. You will explore how AI, ML, **deep learning** and the fast-growing field of **generative AI** connect to each other. Generative AI has captured the attention of businesses and individuals alike, and you will see why.

You will build a solid vocabulary of foundational AI terms, the groundwork for deeper study later. And you will meet a selection of **Amazon Web Services (AWS)** services that use AI and ML, with practical examples of how they solve real-world problems and drive innovation across industries.

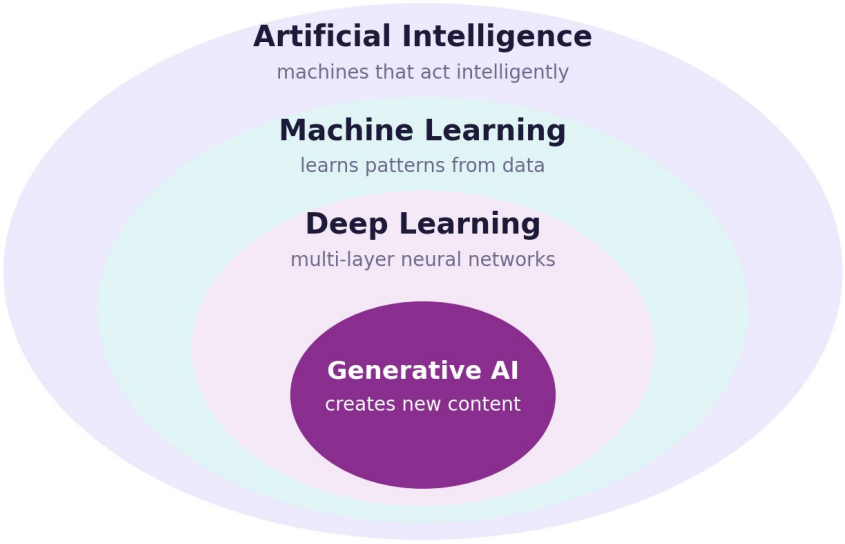

Figure 1. The big picture: each field sits inside the one before it.

✏️ **Try it yourself: your AI diary (3 minutes)**

Write down **five things you did today that used AI**, before you started reading this guide. Think about your phone, apps, email, maps, shopping and payments.

Stuck? Ideas: phone face unlock, Google Maps traffic, YouTube or Instagram recommendations, the spam folder, keyboard autocorrect, a voice assistant, ChatGPT, a UPI fraud alert.

> Keep your list. After Module 6, come back and label each one: ML, deep learning or generative AI, and which AWS service could build it.

Why everyone is talking about AI now

Driver	What changed
Data	Phones, sensors, apps and the internet produce enormous amounts of data every second.
Compute	GPUs and cloud computing make it affordable to train huge models: rent them by the hour instead of buying a data centre.
Algorithms	Breakthroughs such as deep neural networks (2012) and the Transformer (2017) made models far more capable.
Access	Cloud services such as Amazon Bedrock let anyone call a powerful model through an API, without building one.

The course map

Module	Big question it answers
2. AI, ML, DL, GenAI	What is the difference between these four terms?
3. ML fundamentals	How does a machine actually learn from data, and how is a model used?
4. Deep learning	How do neural networks let computers see (vision) and read (language)?
5. Generative AI	How do foundation models, LLMs and diffusion models create new content, and how do we improve their answers?
6. AWS services	Which AWS service should I use for which problem, and what does it cost?

🧠 Remember this

- AI is everywhere already: recommendations, face unlock, spam filters, maps, chatbots.
- This guide connects four fields: AI → ML → deep learning → generative AI.
- Generative AI **creates** new content; classic ML mostly **predicts**. Both are useful.
- AWS offers ready-made AI services, platforms to build models, and generative AI services.

Explain it back: close this guide and explain the module to a friend (or to yourself) in two minutes.

Your questions, answered

Q1 BASIC Do I need to be good at coding or maths to learn AI?

> **A:** Not for this guide. Here you focus on **concepts and use cases**. Many AWS AI services (like Amazon Rekognition or Amazon Transcribe) are used by calling an API, and tools like Amazon SageMaker Canvas are no-code. To *build* models from scratch later, basic Python, statistics and linear algebra help a lot.

Q2 BASIC Is AI the same as a robot? 🤔 *Guess first, then read*

> **A:** No. A robot is a machine body. AI is the "brain" software. Most AI has no body at all: a spam filter or a recommendation engine is AI running on a server. Some robots use AI; many simply follow fixed instructions.

Q3 BASIC Where do I already use AI in daily life?

> **A:** Almost everywhere:
>
> - Face unlock and photo tagging (computer vision)
> - Voice assistants and voice typing (speech recognition)
> - Netflix, YouTube and Amazon suggestions (recommendation)
> - Gmail spam filtering and UPI/credit-card fraud alerts (classification, anomaly detection)
> - Google Maps arrival times (prediction)
> - ChatGPT, Claude, Gemini, Amazon Q (generative AI)

Q4 INTERMEDIATE Why is generative AI getting so much attention compared with older AI?

> **A:** Older AI mostly **predicted or classified** (is this spam? what will sales be?). Generative AI **creates**: it drafts emails, writes code, summarises documents, designs images and answers questions in natural language. Anyone can use it by just typing, so it touches almost every job, not only data-science teams.

Q5 INTERMEDIATE Will AI take my job?

> **A:** AI changes jobs more than it removes them. It automates *tasks* (drafting, summarising, first-level support), so people who know how to use AI well become more productive. The safest career move is to understand AI: what it can do, where it fails, and how to use it responsibly.

Q6 ADVANCED If generative AI is so powerful, why do companies still use "traditional" ML?

> **A:** Because traditional ML is often **cheaper, faster, more accurate and easier to explain** for structured prediction problems: credit scoring, demand forecasting, fraud detection on tabular data. A small gradient-boosted tree model can run in milliseconds for a fraction of a cent. The right answer is usually a mix: GenAI for language and content, classic ML for numeric prediction.

✅ Quick quiz: Module 1

Tick your answers, then check the answer key →

1. Which statement best describes this course?

- ☐ A) It teaches you to build robots
- ☒ B) It explains how AI, ML, deep learning and generative AI connect, and how AWS services apply them
- ☒ C) It is only about ChatGPT
- ☒ D) It is a programming course in Python

2. Generative AI is best described as AI that...

- ☒ A) Only classifies data
- ☒ B) Creates new content such as text, images, code and audio
- ☒ C) Only works on spreadsheets
- ☒ D) Replaces all other AI

3. Why do businesses care about AI/ML?

- ☒ A) It is fashionable
- ☒ B) It automates tasks, finds patterns in data and creates new products and experiences
- ☒ C) It removes the need for data
- ☒ D) It is free

Module 2: AI, ML, Deep Learning and Generative AI

📌 **What you will learn (about 35 minutes)**

- define AI, ML, deep learning and generative AI in one sentence each
- explain how they are nested inside each other
- classify real products into the right category
- give a use case and an AWS service for each

Four definitions to remember

Term	Simple definition	Key idea
Artificial Intelligence (AI)	The broad field of building machines that perform tasks that normally need human intelligence: reasoning, perceiving, understanding language, deciding.	The goal
Machine Learning (ML)	A part of AI where computers **learn patterns from data** instead of being given every rule by a programmer.	Learning from data
Deep Learning (DL)	A part of ML that uses **neural networks with many layers** to learn complex patterns from images, audio and text.	Many-layer neural networks
Generative AI (GenAI)	A part of deep learning that **creates new content**: text, images, code, audio, video.	Creating, not just predicting

💡 **Think of it like learning mathematics**

- **AI** is the whole subject of Mathematics: the big goal.
- **ML** is learning algebra by solving lots of examples rather than memorising rules.
- **Deep learning** is discovering advanced patterns yourself from thousands of problems.
- **Generative AI** is when you can now write new questions, explain solutions and even produce a textbook.

Traditional programming vs machine learning

This single picture explains why ML is different:

	Traditional programming	Machine learning
You give the computer	Rules + data	Data + the correct answers (examples)
The computer produces	Answers	Rules (a trained **model**)
Example	IF email contains “lottery” THEN spam	Show 100,000 emails labelled spam / not spam; it learns what spam looks like
Works best when	Rules are simple and known	Rules are too complex to write by hand, or keep changing

Side-by-side comparison

Feature	AI	ML	Deep learning	Generative AI
Works by	Rules, logic, search, or ML	Algorithms trained on data	Multi-layer neural networks	Very large neural networks (LLMs, diffusion models)
Data needed	Low–medium	Medium	High	Very high
Compute	Low	Medium	High (GPUs)	Very high (GPU clusters)
Feature engineering	Manual	Often manual	Automatic	Automatic
Creates new content?	Usually no	Usually no	Limited	Yes
Typical methods	Expert systems, search (A*)	Linear regression, decision trees, random forest, XGBoost, K-means	CNN, RNN, LSTM, Transformer	GPT, Claude, Amazon Nova, Llama, Stable Diffusion
Example	Chess engine with fixed rules	Spam filter, fraud detection	Face unlock, speech-to-text	ChatGPT writing an email, image from text
AWS example	Amazon Lex (chatbot logic)	Amazon SageMaker AI, Amazon Personalize	Amazon Rekognition, AWS Trainium	Amazon Bedrock, Amazon Q

One bank, four technologies

Technology	What it does for the bank
AI	Makes the overall banking system "smart": routing, decisions, automation.
Machine learning	Learns which card transactions are fraudulent from past examples.
Deep learning	Verifies customers by face (video KYC) or voice.
Generative AI	Answers customer questions in chat, drafts emails, summarises loan documents.

✏️ Try it yourself: sort the products (5 minutes)

Look at Figure 1 again. Write each product in the **smallest** circle it belongs to: (1) a chess program using fixed rules, (2) Netflix recommendations, (3) phone face unlock, (4) a chatbot that writes poems, (5) an email spam filter, (6) Google Translate, (7) a text-to-image app.

Done? Check your answers in the answer key at the end of this guide ("Try-it-yourself answers").

Myth ❌	Fact ✅
AI, ML and deep learning are the same thing.	They are nested: DL ⊂ ML ⊂ AI, and GenAI ⊂ DL.
AI understands things like a human.	Today's AI finds statistical patterns. It can be very capable without human-like understanding.
More data always makes a better model.	Quality matters more than quantity: noisy, biased or irrelevant data makes models worse.
Generative AI will replace all other ML.	Classic ML is still better and cheaper for most numeric prediction on tables.

🧠 Remember this

- **AI** = machines doing tasks that need intelligence. **ML** = learning from data. **Deep learning** = many-layer neural networks. **Generative AI** = creating new content.
- They are nested: GenAI ⊂ DL ⊂ ML ⊂ AI. Always name the smallest circle.
- Traditional programming: rules + data → answers. ML: data + answers → rules (a model).
- Each step inward needs more data and more computing power.

Explain it back: close this guide and explain the module to a friend (or to yourself) in two minutes.

Your questions, answered

Q7 BASIC What is the simplest definition of AI?

A: AI is software that performs tasks we normally associate with human intelligence: recognising images, understanding speech, making decisions, translating languages or answering questions.

Q8 BASIC What exactly does "learning" mean for a machine? 🤔 *Guess first, then read*

A: The model contains adjustable numbers called **parameters** (or weights). Learning means automatically adjusting those numbers so the model's predictions get closer to the correct answers in the training data. The finished set of numbers is the **model**.

Q9 BASIC Is ChatGPT AI, ML, deep learning or generative AI? 🤔 *Guess first, then read*

A: All four! It is generative AI (it creates text), built with deep learning (a Transformer neural network), which is a type of machine learning, which is part of AI. Always name the **smallest** circle: generative AI.

Q10 BASIC Why is it called "deep" learning?

A: Because the neural network has many **layers** stacked on top of each other: sometimes dozens or hundreds. Each layer learns a slightly more abstract pattern than the previous one (edges → shapes → objects).

Q11 BASIC What can generative AI create?

- **Text**: emails, summaries, stories, answers (LLMs such as Claude, GPT, Amazon Nova)
- **Code**: functions, tests, explanations (Amazon Q Developer / Kiro, GitHub Copilot)
- **Images**: from a text prompt (Stable Diffusion, Amazon Nova Canvas)
- **Audio and video**: voices, music, short clips (e.g. Amazon Nova Reel)

Q12 INTERMEDIATE What is the difference between narrow AI and general AI?

A: **Narrow AI** (ANI) is good at one task or a family of tasks: all AI in use today is narrow, even very capable chatbots. **Artificial General Intelligence** (AGI) would match humans across almost any intellectual task; it is a research goal and a topic of debate, not an existing product.

Q13 INTERMEDIATE When should I NOT use machine learning?

- When a simple rule works ("send a reminder 3 days before the due date").
- When you have little or no relevant data.
- When every decision must be 100% explainable or 100% correct (e.g. some legal or safety rules).
- When the cost of building and maintaining a model is more than the value it creates.

Q14 INTERMEDIATE Why did deep learning suddenly take off after 2012?

A: Three things came together: **big data** (e.g. the ImageNet dataset with 14 million labelled images), **GPUs** that train networks many times faster, and **better techniques** (ReLU activation,

> dropout, better initialisation). In 2012 the AlexNet model won the ImageNet challenge by a huge margin and started the boom.

Q15 INTERMEDIATE What is the difference between a discriminative model and a generative model?

> **A:** A **discriminative** model learns to tell things apart: "is this email spam or not?" It learns the boundary between classes. A **generative** model learns what the data itself looks like, so it can produce new examples: "write a new email that looks like a real one."

Q16 INTERMEDIATE Does generative AI always need deep learning?

> **A:** In practice today, yes: modern generative models (LLMs, diffusion models, GANs, VAEs) are all deep neural networks. Very early "generative" methods existed (e.g. simple statistical text generators), but they were far less capable.

Q17 ADVANCED Is all AI based on learning from data?

> **A:** No. **Symbolic AI** (also called rule-based or "good old-fashioned AI") uses hand-written logic, knowledge graphs and search, e.g. expert systems and route planners. **Statistical AI** (ML) learns from data. Modern systems often combine both: an LLM (learned) calling a calculator or a database (symbolic tools).

Q18 ADVANCED Why does generative AI need so much more data and compute than classic ML?

> **A:** A classic model might learn one narrow mapping (transaction → fraud score) with thousands of parameters. A foundation model must learn grammar, facts, reasoning styles and many languages at once, which needs **billions of parameters** trained on **trillions of tokens**. Research on "scaling laws" found that performance improves predictably as model size, data and compute grow together, which pushed models to become very large.

Q19 ADVANCED Can generative AI be used for prediction tasks like fraud detection?

> **A:** It can help, for example by summarising a suspicious case for an analyst, extracting fields from documents, or turning text into features. But for the fraud *score* itself, a well-tuned classic model on transaction data is usually faster, cheaper and easier to audit. A common design is **classic ML for the decision + GenAI for the explanation**.

✅ Quick quiz: Module 2

Tick your answers, then check the answer key →

1. Which is true?

- ☒ **A)** All AI is machine learning
- ☒ **B)** All deep learning is machine learning
- ☒ **C)** All machine learning is generative AI
- ☒ **D)** Deep learning does not use data

2. A spam filter trained on labelled emails is an example of...

- ☒ **A)** Rule-based AI
- ☒ **B)** Machine learning
- ☒ **C)** Generative AI
- ☒ **D)** Robotics

3. What makes deep learning "deep"?

- ☒ **A)** It uses very large datasets
- ☒ **B)** It uses neural networks with many layers
- ☒ **C)** It runs in the cloud
- ☒ **D)** It thinks like a human

4. Which task is generative?

- ☒ **A)** Predicting house prices
- ☒ **B)** Detecting fraud
- ☒ **C)** Writing a product description from bullet points
- ☒ **D)** Grouping customers into segments

5. In traditional programming the computer gets rules + data and outputs answers. In ML it gets...

- ☒ **A)** Rules + answers → data
- ☒ **B)** Data + answers → rules (a model)
- ☒ **C)** Only rules
- ☒ **D)** Only answers

Module 3: Machine Learning Fundamentals

📌 **What you will learn (about 50 minutes)**

- explain how a model is trained, step by step
- distinguish labelled vs unlabelled and structured vs unstructured data
- name the four learning types and when to use each
- describe the ML process (lifecycle) from business goal to monitoring
- choose between batch and real-time inference

Key vocabulary

Term	Meaning	Example (house prices)
Feature	An input the model uses to decide	Size, location, number of rooms
Label (target)	The answer we want the model to predict	Price
Example / record	One row of data	One house with its features and price
Model	The learned mathematical function	price ≈ f(size, location, rooms)
Training	Adjusting the model's parameters using examples	Learning from 10,000 past sales
Inference	Using the trained model on new data	Predicting the price of a new listing

Labelled vs unlabelled data

	Labelled data	Unlabelled data
What it is	Each example includes the correct answer	Only inputs, no answers
Example	Emails marked spam / not spam; photos tagged "cat"	A folder of 1 million customer transactions with no tags
Used for	Supervised learning (prediction)	Unsupervised learning (finding groups, anomalies)
Challenge	Labelling is slow and expensive (often done by humans)	Harder to evaluate: there is no "right answer" to compare against

	Labelled data	Unlabelled data
AWS help	Amazon SageMaker Ground Truth (human + automated labelling)	SageMaker built-in algorithms such as K-Means, Random Cut Forest

Structured vs unstructured data

	Structured	Semi-structured	Unstructured
Looks like	Rows and columns with a fixed schema	Tagged but flexible: JSON, XML, logs	No fixed format
Examples	Bank transactions, Excel sheets, SQL tables, sensor time series	API responses, clickstream events, emails' metadata	Text documents, images, audio, video, social posts
Share of world data	Smaller share	Growing	The large majority
Typical models	Classic ML (XGBoost, linear/logistic regression)	Parsed into structured, then classic ML	Deep learning (CNNs, Transformers)
AWS services	SageMaker AI, Redshift ML, Glue	Glue, Athena	Rekognition, Comprehend, Textract, Transcribe, Bedrock

The four ways a machine can learn

Type	Data	Goal	Example	Common algorithms
Supervised	Labelled	Predict a label: a category (classification) or a number (regression)	Spam detection, house price, loan default	Linear/logistic regression, decision tree, random forest, XGBoost
Unsupervised	Unlabelled	Find hidden structure	Customer segments, anomaly detection	K-Means, DBSCAN, PCA
Semi-supervised	Few labels + many unlabelled	Better accuracy with fewer labels	Medical images when only some are labelled	Self-training, pseudo-labelling

Type	Data	Goal	Example	Common algorithms
Reinforcement	Rewards and penalties from actions	Learn the best sequence of actions	Game AI, robot walking, AWS DeepRacer, RLHF	Q-learning, PPO

> 💡 **Think of it like your own classroom**
>
> - **Supervised:** practice questions with the answer key.
> - **Unsupervised:** "Here is a box of mixed objects, group the similar ones." No instructions.
> - **Semi-supervised:** the teacher solves a few examples on the board; you solve the rest.
> - **Reinforcement:** marks for good answers, marks lost for mistakes; you learn the strategy that scores highest.

How to train a model: the ML process

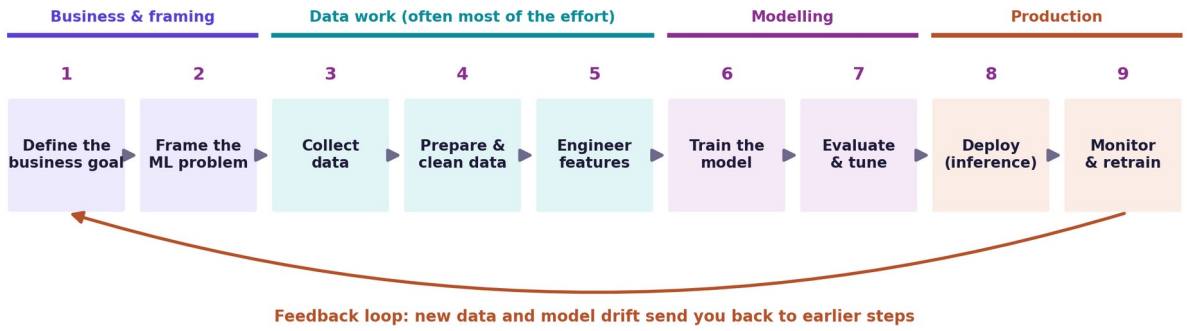

Figure 2. The machine learning lifecycle. It is a loop, not a straight line.

Step	What happens	AWS tools (SageMaker AI unless noted)
1. Business goal	What problem, and how will we measure success? (e.g. "cut fraud losses by 20%")	—
2. Frame the ML problem	Is it classification, regression, clustering…? What are the features and the label?	—
3. Collect data	Gather data from databases, logs, files, APIs	Amazon S3, AWS Glue
4. Prepare data	Clean missing values, remove duplicates, fix errors, label	Data Wrangler, Ground Truth
5. Feature engineering	Create useful inputs, e.g. "number of purchases in last 7 days"	Feature Store

Step	What happens	AWS tools (SageMaker AI unless noted)
6. Train	Split data; the algorithm adjusts parameters to reduce error (loss)	Training jobs, Autopilot, Canvas (no-code)
7. Evaluate & tune	Measure on unseen data; try different settings (hyperparameters)	Automatic Model Tuning, Clarify (bias)
8. Deploy	Make the model available for inference	Endpoints, Batch Transform
9. Monitor	Watch accuracy and data drift; retrain when needed	Model Monitor, Pipelines (MLOps)

Splitting the data

Never judge a model on the data it studied. Typically:

- **Training set (≈70–80%)**: the model learns from it.
- **Validation set (≈10–15%)**: used to tune settings and pick the best version.
- **Test set (≈10–15%)**: locked away until the very end for a fair final score.

> 💡 **Think of it like exam preparation**
>
> Training set = textbook exercises. Validation set = mock tests you use to adjust your study plan. Test set = the real board exam paper you have never seen.

Overfitting and underfitting

	Underfitting	Good fit	Overfitting
What happens	Model too simple; misses the pattern	Captures the real pattern	Memorises training data, including noise
Training score	Poor	Good	Excellent
Test score	Poor	Good	Poor
Student analogy	Didn’t study enough	Understood the concepts	Memorised last year’s answers word for word
Fix	More features, more complex model, train longer	—	More data, simpler model, regularisation, early stopping

How do we measure a model?

Problem	Common metrics	Plain meaning
Classification	Accuracy, precision, recall, F1, AUC	Precision: when it says “fraud”, how often is it right? Recall: of all real frauds, how many did it catch?
Regression	MAE, RMSE, R^2	On average, how far are predictions from the true numbers?
Business	Revenue gained, cost saved, time saved	Does the model actually help the business goal from step 1?

Inference: using the trained model

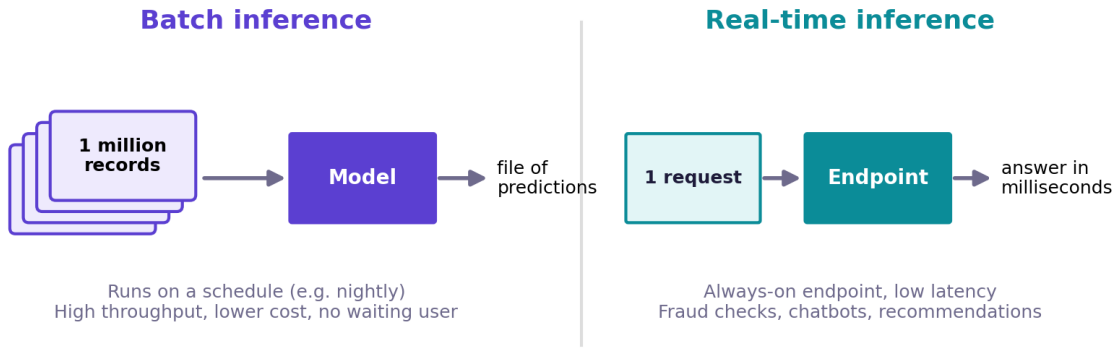

Figure 3. Batch vs real-time inference.

	Batch inference	Real-time inference
When	Predictions for a large dataset at once, on a schedule	One request at a time, answer needed immediately
Latency	Minutes to hours is fine	Milliseconds to a second
Cost	Lower: compute runs only during the job	Higher: an endpoint is always running (unless serverless)
Examples	Nightly churn scores for all customers; monthly credit risk review	Card fraud check at payment; chatbot reply; product recommendations on page load
SageMaker AI option	Batch Transform	Real-time endpoints

SageMaker AI also offers two in-between options: **Serverless Inference** (no servers to manage, scales to zero, good for spiky or low traffic) and **Asynchronous Inference** (queues large requests such as long videos, returns results when ready).

🎯 Exam tip

Match the inference type to the keywords: “immediately / low latency / interactive” → **real-time**; “large dataset / nightly / no need for immediate response” → **batch**; “intermittent traffic, avoid idle cost” → **serverless**; “large payloads, long processing” → **asynchronous**.

✏️ Try it yourself: be the model (10 minutes)

Step 1: Training. Study the 8 labelled fruits below and write your own rules on paper, e.g. “yellow and long → banana”. Those rules are your **model**.

Step 2: Inference. Use only your rules to predict the 4 unknown fruits in the second table.

Step 3: Evaluate. Check the answer key and work out your **accuracy** (correct ÷ 4). Did any of your rules only work for the 8 examples? That is **overfitting**.

#	Colour	Size	Weight	Label (fruit)
1	Red	Medium	180 g	Apple
2	Yellow	Long	120 g	Banana
3	Orange	Medium	200 g	Orange
4	Green	Medium	170 g	Apple
5	Yellow	Long	140 g	Banana
6	Orange	Medium	220 g	Orange
7	Red	Small	150 g	Apple
8	Green	Long	110 g	Banana

#	Colour	Size	Weight	Your prediction
A	Red	Medium	190 g	
B	Yellow	Long	130 g	
C	Orange	Medium	210 g	
D	Yellow	Medium	160 g	

Myth ❌	Fact ✅
The model keeps learning while it makes predictions.	Normally no: during inference the weights are fixed. Learning only happens when you retrain.
99% accuracy means an excellent model.	Not always. If only 1% of transactions are fraud, a model that always says “not fraud” is 99% accurate and useless. Check precision and recall.
Training is the hardest part of ML.	Data preparation usually takes most of the time and effort.

🧠 **Remember this**

- **Features** are inputs; the **label** is the answer to predict. Labelled data → supervised; unlabelled → unsupervised.
- Structured data = tables; unstructured = text, images, audio, video.
- The ML process: goal → frame → collect → prepare → features → train → evaluate → deploy → monitor (a loop).
- Split data into training, validation and test sets. Watch for overfitting.
- **Inference** = using the model. **Batch** for large scheduled jobs; **real-time** when someone is waiting.

Explain it back: close this guide and explain the module to a friend (or to yourself) in two minutes.

Your questions, answered

Q20 BASIC What is a “model” in machine learning?

A: A model is the learned result of training: a mathematical function with parameters that maps inputs (features) to outputs (predictions). Think of it as a recipe the computer wrote for itself from examples. In practice it is saved as a file (the **model artifact**, often stored in Amazon S3).

Q21 BASIC What is the difference between a feature and a label? 🤔 *Guess first, then read*

A: **Features** are the inputs you know (age, income, transaction amount). The **label** is the answer you want to predict (will this customer default: yes/no). Unsupervised learning has features but no label.

Q22 BASIC Why can’t we just use all our data for training?

A: Because then we have no honest way to check the model on data it has never seen. A model can look perfect on its own training data simply by memorising it. The held-out test set shows how it will behave in the real world.

Q23 BASIC Is a photo structured or unstructured data? 🤔 *Guess first, then read*

> **A:** Unstructured. It is a grid of pixel values with no predefined fields like "name" or "age". Deep learning models (CNNs, Vision Transformers) or services such as Amazon Rekognition turn it into useful information.

Q24 BASIC What does it mean to "train" a model, in simple words?

> **A:** The model makes a prediction, compares it with the correct answer, measures the error (the **loss**), and nudges its parameters to reduce that error. Repeat this thousands or millions of times and the predictions get better.

Q25 INTERMEDIATE How do I know whether my problem is classification or regression?

> **A:** Look at the label. If it is a **category** (spam/not spam, cat/dog/bird, low/medium/high risk), it is **classification**. If it is a **number** on a continuous scale (price, temperature, demand), it is **regression**.

Q26 INTERMEDIATE Where do labels come from?

> **A:** From existing records (past loans that did or did not default), from user behaviour (clicked or not clicked), or from humans labelling data. Amazon SageMaker Ground Truth manages labelling jobs with your own team, vendors or Amazon Mechanical Turk workers, and can use active learning to label some data automatically.

Q27 INTERMEDIATE What are hyperparameters, and how are they different from parameters?

> **A:** **Parameters** are learned *from the data* during training (e.g. the weights of a neural network). **Hyperparameters** are settings *you choose before* training: learning rate, number of trees, number of layers, batch size, epochs. SageMaker Automatic Model Tuning searches for good hyperparameters for you.

Q28 INTERMEDIATE What is feature engineering and why does it matter?

> **A:** Turning raw data into inputs that make the pattern easier to learn. For fraud, "amount" alone is weak; "amount compared with this customer's average" and "transactions in the last 10 minutes" are much stronger. Good features often improve a classic model more than switching algorithms.

Q29 INTERMEDIATE When would I choose batch inference over real-time?

> **A:** When nobody is waiting for the answer and you have many records: nightly scoring, monthly reports, pre-computing recommendations. Batch is cheaper because compute runs only during the job. Choose real-time when a user or system needs the answer within milliseconds (payment fraud check, chatbot).

Q30 INTERMEDIATE What is data drift?

> **A:** Real-world data slowly changes after deployment, e.g. new customer behaviour after a festival season or a new product launch, so the model's accuracy drops. SageMaker Model Monitor compares live data with the training data and raises an alert so you can retrain.

Q31 INTERMEDIATE What is the difference between accuracy, precision and recall?

A: Take a cancer screening model:

- **Accuracy**: of all predictions, how many were correct?
- **Precision**: of the patients it flagged, how many really had cancer? (Few false alarms.)
- **Recall**: of all patients who had cancer, how many did it catch? (Few missed cases.)

For medical screening we usually prioritise recall; for spam filtering, precision (we don't want real emails in spam).

Q32 ADVANCED Why is accuracy misleading on imbalanced data, and what do we do about it?

A: If 99.5% of transactions are genuine, predicting "genuine" every time gives 99.5% accuracy and catches zero fraud. Use precision, recall, F1 or AUC; resample the data (oversample fraud, undersample genuine); use class weights; and choose the decision threshold based on business cost.

Q33 ADVANCED What is k-fold cross-validation?

A: Instead of one fixed validation set, split the training data into k parts (say 5). Train 5 times, each time holding out a different part for validation, and average the scores. It gives a more reliable estimate, especially with small datasets.

Q34 ADVANCED What is the bias–variance trade-off?

A: **Bias** is error from a model that is too simple (underfitting). **Variance** is error from a model that is too sensitive to the training data (overfitting). Making a model more complex lowers bias but raises variance. The goal is the sweet spot with the lowest total error on new data.

Q35 ADVANCED What is MLOps?

A: MLOps applies DevOps ideas to ML: version your data, code and models; automate training and deployment pipelines; monitor models in production; retrain automatically. On AWS: SageMaker Pipelines, Model Registry, Model Monitor, plus CI/CD tools.

Q36 ADVANCED Can a model be biased even if the algorithm is neutral?

A: Yes. If historical data reflects unfair decisions (e.g. past loans approved less often for some groups), the model learns that pattern. Check datasets and predictions for bias (Amazon SageMaker Clarify helps), use representative data, and keep humans in the loop for high-stakes decisions.

Q37 ADVANCED What is the difference between online learning and batch training?

A: **Batch (offline) training** learns from a fixed dataset, then the model is deployed and frozen until the next retrain. **Online learning** updates the model continuously as new data arrives. Online learning adapts quickly but is riskier (bad data can corrupt the model), so most production systems retrain periodically instead.

✅ Quick quiz: Module 3

Tick your answers, then check the answer key →

1. A dataset of 50,000 X-ray images each marked “pneumonia” or “normal” is...

- ☒ A) Unlabelled, structured
- ☒ B) Labelled, unstructured
- ☒ C) Unlabelled, unstructured
- ☒ D) Labelled, structured

2. Grouping customers by behaviour when you have no labels is...

- ☒ A) Supervised learning
- ☒ B) Unsupervised learning
- ☒ C) Reinforcement learning
- ☒ D) Batch inference

3. Your model scores 99% on training data but 70% on new data. This is...

- ☒ A) Underfitting
- ☒ B) Overfitting
- ☒ C) Perfect generalisation
- ☒ D) Data drift

4. A bank scores every loan application overnight, all at once. Which inference type?

- ☒ A) Real-time
- ☒ B) Batch
- ☒ C) Streaming training
- ☒ D) Reinforcement

5. Which data is used for the final, unbiased check of a model?

- ☒ A) Training set
- ☒ B) Validation set
- ☒ C) Test set
- ☒ D) Feature store

Module 4: Deep Learning Fundamentals

> 📌 **What you will learn (about 35 minutes)**
>
> - explain how an artificial neural network is inspired by the brain
> - describe how a network learns (loss, backpropagation, gradient descent)
> - name the main computer vision and NLP tasks and give examples
> - match common architectures (CNN, RNN, Transformer) to data types

Start with the real brain

Your brain has about **86 billion neurons**. Each neuron receives signals through its dendrites; if the combined signal is strong enough, it **fires** and passes a signal along its axon to other neurons. When you practise something, the connections (synapses) you use become stronger. **That is learning.** Deep learning copies these three ideas.

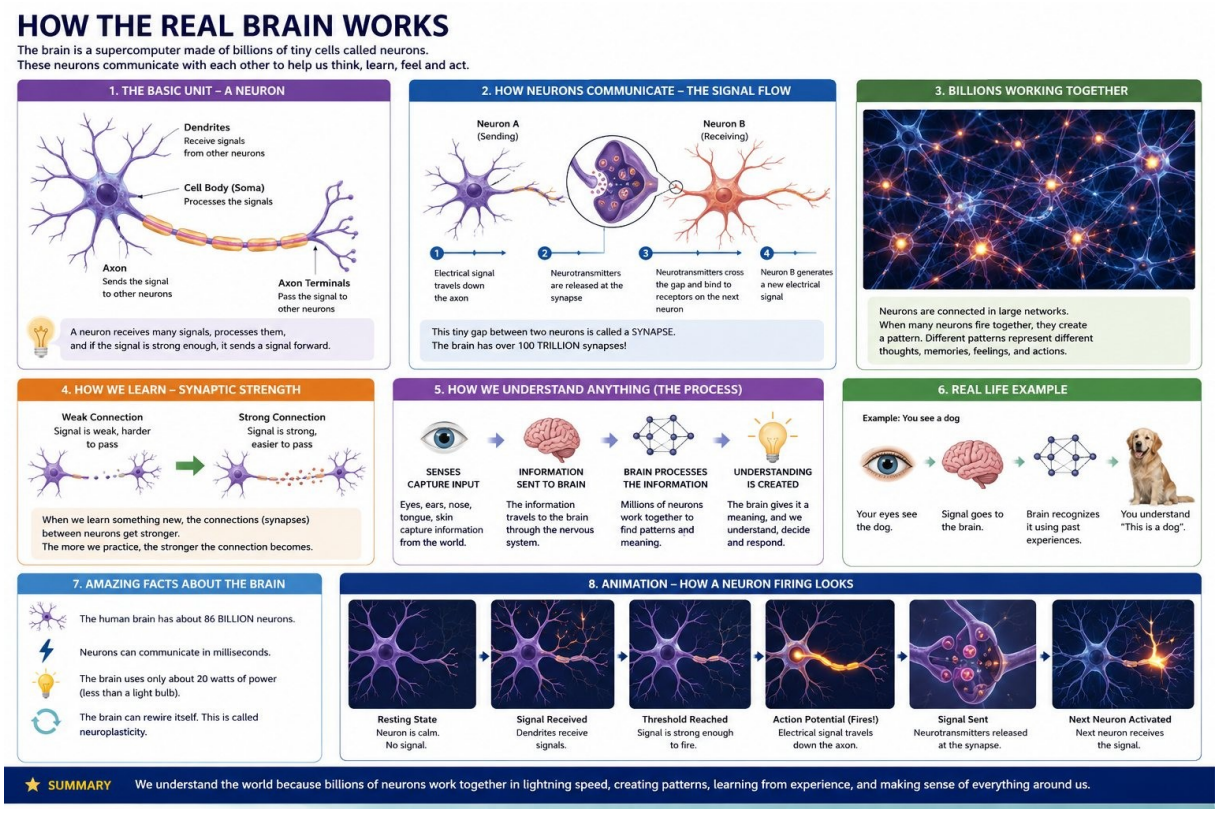

Figure 4. How a real brain works: panels 1, 4 and 8 show the ideas that neural networks copy.

Real brain	Artificial neural network
Dendrites receive signals	Inputs (features) x_1, x_2, x_3 …
Synapse strength	Weights w_1, w_2, w_3 …

Real brain	Artificial neural network
Cell body combines signals	Weighted sum: z = w·x + b
Fires only above a threshold	Activation function (e.g. ReLU)
Axon sends signal onward	Output becomes input to the next layer
Practice strengthens synapses	Training adjusts the weights

Neural networks

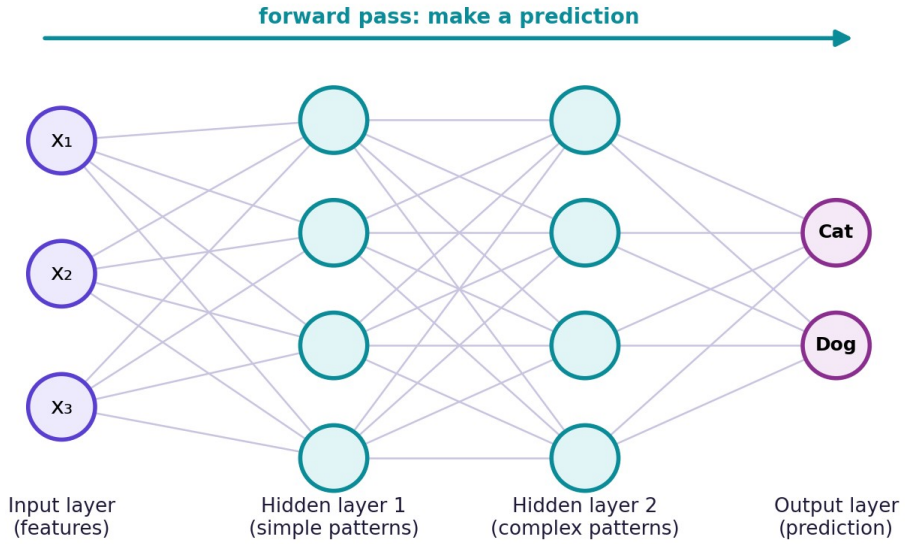

Figure 5. A small neural network: input layer, two hidden layers, output layer.

- **Input layer**: receives the features (pixel values, words as numbers, sensor readings).
- **Hidden layers**: each layer learns more abstract patterns. “Deep” = many hidden layers.
- **Output layer**: produces the prediction, e.g. probabilities for “cat” and “dog”.
- **Weights and biases**: the numbers the network learns. Large models have billions.
- **Activation function**: adds non-linearity so the network can learn curves, not just straight lines. ReLU = max(0, z) is the most common.

How a network learns, in four repeated steps

1. **Forward pass**: data flows through the layers and produces a prediction.
2. **Loss**: compare the prediction with the correct answer and measure the error.
3. **Backpropagation**: work backwards through the layers to find how much each weight contributed to the error.
4. **Gradient descent**: nudge every weight a little in the direction that reduces the error. Repeat for many **epochs** (passes over the data).

> 💡 **Think of it like walking down a hill in fog**
>
> You cannot see the bottom (the lowest error), but you can feel the slope under your feet. Take a small step downhill, feel again, step again. The **learning rate** is your step size: too small and you take forever; too big and you overshoot the valley.

Common deep learning architectures

Architecture	Best for	Idea	Example
Feed-forward network (MLP)	Tabular data	Every neuron connects to every neuron in the next layer	Credit scoring
CNN (Convolutional Neural Network)	Images, video	Small filters slide over the image to detect edges → shapes → objects	Face unlock, X-ray analysis
RNN / LSTM / GRU	Sequences (older approach)	Reads one step at a time, carrying a memory	Early speech recognition, time series
Transformer	Text, and now images, audio, code	Self-attention lets every token look at every other token at once	ChatGPT, Claude, Amazon Nova, BERT
GAN / VAE / diffusion	Generating content	Learn the data distribution and sample new examples	Image generation (Module 5)

Computer vision (CV)

Computer vision lets computers **see and understand images and video**. To a computer, a photo is a grid of pixels, and each pixel is three numbers for Red, Green and Blue (0–255). A 1920×1080 photo is over 2 million pixels, more than 6 million numbers.

CV task	Question it answers	Real use
Image classification	What is in this image?	Plant disease apps, Google Photos search
Object detection	What objects are here, and where? (boxes)	Self-driving cars, CCTV, retail shelf monitoring
Image segmentation	Which exact pixels belong to each object?	Medical scans, satellite images
Face recognition	Whose face is this?	Phone unlock, airport e-gates, attendance

CV task	Question it answers	Real use
OCR / document analysis	What text is in this image or scan?	Invoices, cheques, ID cards (Amazon Textract)
Pose estimation	Where are the body joints?	Fitness and yoga apps, sports analytics
Image generation	Create an image from a description	Stable Diffusion, Amazon Nova Canvas

On AWS: Amazon Rekognition (faces, objects, text, unsafe content, PPE detection in images and video) and Amazon Textract (text, forms and tables from documents).

Natural language processing (NLP)

NLP lets computers **understand, interpret and generate human language**, in writing or speech. Language is hard because it is ambiguous: "I deposited money in the **bank**" vs "We sat on the river **bank**". Modern models use context to decide.

NLP task	Example	AWS service
Sentiment analysis	"Worst product ever" → Negative	Amazon Comprehend
Named entity recognition	"Rahul works at Amazon in Bengaluru" → Person, Org, Location	Amazon Comprehend
Language detection & key phrases	Detect Hindi vs English; pull out main topics	Amazon Comprehend
Translation	English → Hindi	Amazon Translate
Speech-to-text	Lecture recording → transcript	Amazon Transcribe
Text-to-speech	Course notes → natural voice	Amazon Polly
Conversational bots	"Book me a slot tomorrow at 5"	Amazon Lex
Summarisation, Q&A, generation	Summarise a 40-page report	Amazon Bedrock (LLMs)

> ✏️ **Try it yourself: spot the AI in your phone (5 minutes)**
>
> Unlock your phone and list every feature that uses CV or NLP: camera scene detection, photo search, keyboard suggestions, voice typing, live captions, translation in the camera… Label each one CV or NLP. Which ones use both? (Hint: translating a sign with your camera = OCR + translation.)

Myth ❌	Fact ✅
Neural networks work exactly like the brain.	They are inspired by the brain but are much simpler maths: weighted sums and activations.
Deep learning is always better.	For small, tabular datasets classic ML (e.g. XGBoost) often wins and is cheaper.
The computer “sees” a dog.	It processes numbers (pixels) and outputs a probability, e.g. dog 98%.

🧠 **Remember this**

- A neuron computes a weighted sum, then an activation. Weights are what the network learns.
- Learning loop: forward pass → loss → backpropagation → gradient descent.
- **CNNs** for images; **Transformers** for language (and now much more).
- Computer vision: classification, detection, segmentation, OCR. NLP: sentiment, entities, translation, speech.

Explain it back: close this guide and explain the module to a friend (or to yourself) in two minutes.

Your questions, answered

Q38 BASIC What is a neural network in one sentence?

A: A set of connected layers of simple maths units (“neurons”) that learn, by adjusting their connection weights, to turn inputs into correct outputs.

Q39 BASIC What does a single artificial neuron actually calculate? 🤔 *Guess first, then read*

A: Two things: (1) a weighted sum of its inputs plus a bias, $z = w_1x_1 + w_2x_2 + \ldots + b$; (2) an activation function on that sum, e.g. ReLU(z) = max(0, z). Example: inputs [2, 3], weights [0.5, −1], bias 1 → z = 1 − 3 + 1 = −1 → ReLU gives 0, so the neuron stays silent.

Q40 BASIC How does a computer see an image?

A: As numbers. Every pixel has Red, Green and Blue values from 0 to 255: [255, 0, 0] is pure red. The model learns which patterns of numbers mean “eye”, “ear” or “dog”.

Q41 BASIC What is the difference between computer vision and NLP?

A: Computer vision works with **images and video**; NLP works with **text and speech**. Both are applications of deep learning, and many modern **multimodal** models handle both, e.g. describing a photo in words.

Q42 INTERMEDIATE Why do neural networks need activation functions?

A: Without them, stacking layers is pointless: many linear steps combined are still one linear step, a straight line. Activation functions add curves (non-linearity) so the network can learn complex shapes such as faces or the meaning of sentences.

Q43 INTERMEDIATE What is an epoch, a batch and a learning rate?

- **Epoch**: one full pass through the entire training dataset.
- **Batch**: a small group of examples processed together before updating the weights (e.g. 32 images).
- **Learning rate**: how big each weight update step is.

Q44 INTERMEDIATE Why do deep learning models need GPUs?

A: Training is mostly huge numbers of matrix multiplications that can run in parallel. GPUs have thousands of small cores designed for exactly that, so they train models many times faster than CPUs. AWS also builds its own chips: **Trainium** for training and **Inferentia** for inference.

Q45 INTERMEDIATE How does a CNN recognise objects?

A: Early layers use small filters that detect edges and colours; middle layers combine edges into shapes such as eyes or wheels; deep layers combine shapes into whole objects such as faces or cars. The network learns these filters by itself from labelled images.

Q46 INTERMEDIATE How does NLP handle words with many meanings, like “bank” or “Apple”?

A: Modern models turn each word into a vector that depends on its **context**. Using attention, the word “bank” looks at nearby words (“money”, “loan” or “river”) and its representation changes accordingly.

Q47 ADVANCED What problem did Transformers solve that RNNs could not?

A: RNNs read one word at a time, so they were slow to train and forgot information from long sentences. Transformers use **self-attention**: every token looks at every other token at the same time. That captures long-range relationships and trains in parallel on GPUs, which made today’s large language models possible (paper: “Attention Is All You Need”, 2017).

Q48 ADVANCED What is the vanishing gradient problem?

A: In deep networks the error signal is multiplied layer by layer during backpropagation. With activations like sigmoid it can shrink towards zero, so early layers barely learn. Solutions: ReLU activations, careful weight initialisation, batch normalisation and residual (shortcut) connections.

Q49 ADVANCED What is transfer learning?

A: Reusing a model already trained on a huge dataset and adapting it to your task with far less data. E.g. start from a CNN trained on millions of general images, then fine-tune it on 2,000 images of your factory’s defects. Foundation models (Module 5) are transfer learning at massive scale. Rekognition Custom Labels and SageMaker JumpStart make this easy.

Q50 ADVANCED How do we stop a deep network from overfitting?

- **More / augmented data**: flips, crops, noise.
- **Dropout**: randomly switch off neurons during training.
- **Weight decay (L2)**: penalise very large weights.
- **Early stopping**: stop when validation error stops improving.
- **Transfer learning**: start from a pretrained model.

✅ Quick quiz: Module 4

Tick your answers, then check the answer key →

1. In an artificial neuron, what plays the role of synapse strength?

- ☒ A) The activation function
- ☒ B) The weights
- ☒ C) The output
- ☒ D) The dataset

2. Which task draws boxes around each car and person in an image?

- ☒ A) Image classification
- ☒ B) Object detection
- ☒ C) Sentiment analysis
- ☒ D) Speech recognition

3. "Rahul works at Amazon in Bengaluru" → Person, Organisation, Location. Which NLP task?

- ☒ A) Translation
- ☒ B) Named entity recognition
- ☒ C) Summarisation
- ☒ D) Text-to-speech

4. What is backpropagation used for?

- ☒ A) Making predictions faster
- ☒ B) Working out how to adjust each weight to reduce the error
- ☒ C) Storing training data
- ☒ D) Splitting text into tokens

5. Which architecture powers modern LLMs?

- ☒ A) CNN
- ☒ B) Transformer
- ☒ C) Decision tree
- ☒ D) K-Means

Module 5: Generative AI Fundamentals

📌 **What you will learn (about 60 minutes)**

- explain foundation models, LLMs, tokens and embeddings
- describe the foundation model lifecycle
- explain diffusion (forward and reverse), GANs, VAEs and multimodal models
- choose between prompt engineering, RAG, fine-tuning and RLHF to improve outputs
- describe what Amazon Bedrock offers

Foundation models (FMs)

A **foundation model** is a very large deep learning model **pre-trained on a broad, massive dataset** (text, code, images…) that can be **adapted to many different tasks**: answering questions, summarising, writing code, classifying, translating. It is a "foundation" because you build many applications on top of one model instead of training a new model for every task.

Traditional ML model	Foundation model
Trained for one task (e.g. spam detection)	Pre-trained once, adapted to hundreds of tasks
Needs labelled data for each task	Pre-trained with self-supervised learning on unlabelled data
Thousands to millions of parameters	Billions of parameters
You usually train it yourself	You usually use it through an API (e.g. Amazon Bedrock)

How do foundation models learn?

Mostly by **self-supervised learning**: the data creates its own labels. For text, hide the next word and ask the model to predict it: "The capital of France is ___". The real next word is the label, so no human labelling is needed. Repeat over trillions of words and the model absorbs grammar, facts and reasoning patterns.

The foundation model lifecycle

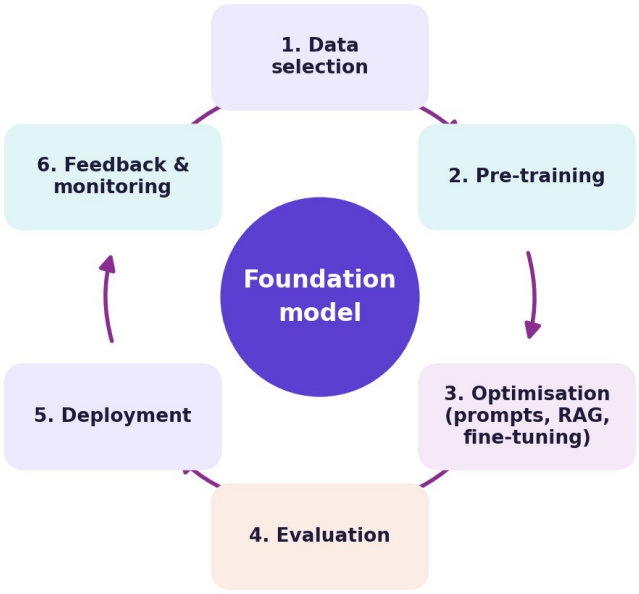

Figure 6. The foundation model lifecycle is a continuous loop.

Stage	What happens
1. Data selection	Gather huge, diverse, high-quality data; remove duplicates, harmful and private content.
2. Pre-training	Self-supervised training on the data, often for weeks on thousands of GPUs/Trainium chips. Result: a base model.
3. Optimisation	Adapt it: prompt engineering, RAG, fine-tuning, instruction tuning, RLHF.
4. Evaluation	Test quality, accuracy, safety, bias, and task-specific benchmarks; human review.
5. Deployment	Serve the model for inference, e.g. through Amazon Bedrock or a SageMaker AI endpoint.
6. Feedback & monitoring	Collect user feedback, monitor quality and cost, feed improvements back into the loop.

Large language models (LLMs)

An **LLM** is a foundation model specialised in language, built on the **Transformer** architecture. Examples: Anthropic Claude, Amazon Nova, Meta Llama, Mistral, OpenAI GPT. At its core an LLM does one thing extremely well: **predict the next token**, then repeat.

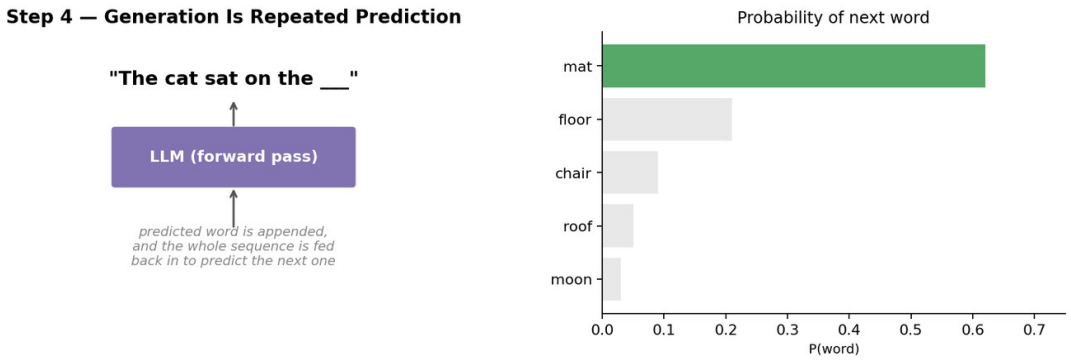

Figure 7. An LLM scores every possible next word, picks one, appends it and repeats.

Tokens

LLMs read and write **tokens**: words or pieces of words. "Unbelievable" might become `un` + `believ` + `able`. Each token maps to an ID number. As a rough rule for English, **1 token ≈ 4 characters ≈ ¾ of a word**, so 100 tokens ≈ 75 words (other languages often use more tokens per word).

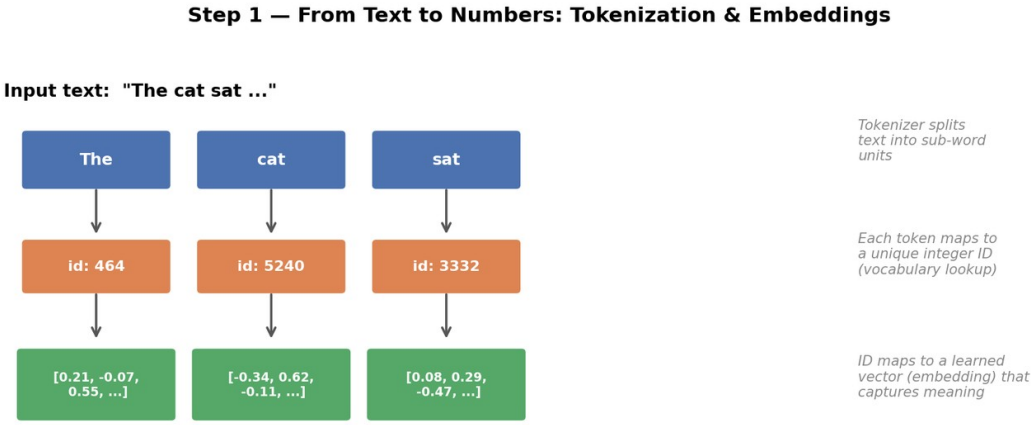

Figure 8. Text → tokens → IDs → embedding vectors.

- **Why tokens matter for you:** LLM pricing is usually **per input token + per output token**.
- The **context window** is the maximum number of tokens (prompt + answer) the model can handle at once.
- **Max tokens** is a setting that limits how long the answer can be.

Embeddings and vectors

An **embedding** is a list of numbers (a **vector**) that represents the *meaning* of a token, sentence, image or document. Items with similar meanings get vectors that are close together, so "king" is near "queen", and "How do I reset my password?" is near "I forgot my login".

- **Semantic search**: find documents by meaning, not exact keywords.
- **RAG**: retrieve the most relevant chunks for a question (see below).
- **Vector databases** store and search embeddings: Amazon OpenSearch Service, Amazon Aurora PostgreSQL (pgvector), Amazon S3 Vectors, and others.

- **AWS embedding models**: Amazon Titan Text Embeddings, Amazon Nova multimodal embeddings, Cohere Embed on Bedrock.

> 💡 **Think of it like a map of meanings**
>
> Imagine every word placed on a giant map. Cities about animals are in one region, vehicles in another, royalty in a third. An embedding is a word's GPS coordinates; measuring the distance between coordinates tells you how related two ideas are.

How models generate images and more

Diffusion models

Diffusion models (e.g. Stable Diffusion, Amazon Nova Canvas) generate high-quality images by learning to **remove noise**.

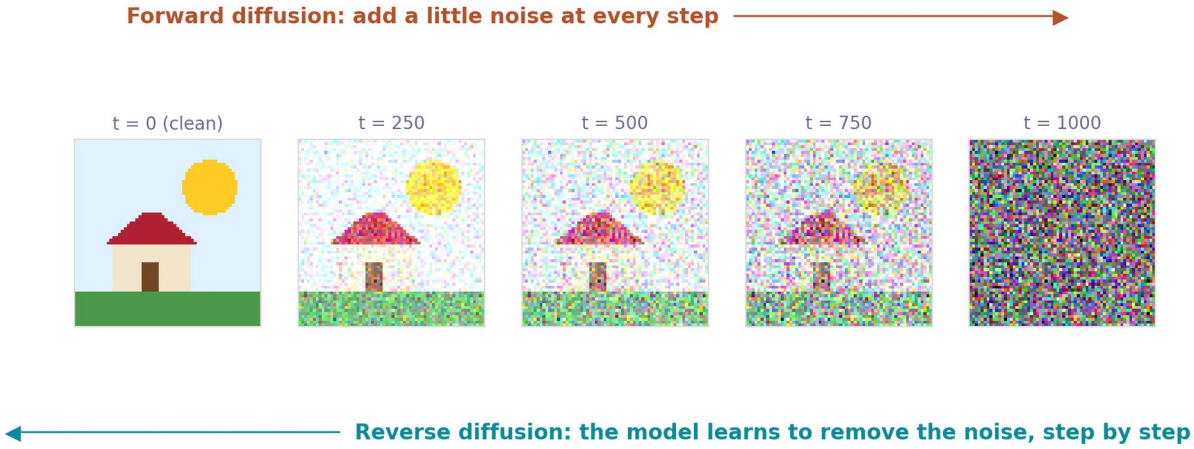

Figure 9. Forward diffusion adds noise; the model learns reverse diffusion to turn noise into an image.

	Forward diffusion	Reverse diffusion
What happens	Take a real image and add a little random noise, step by step, until it is pure static	Start from pure noise and remove a little noise at each step until a clear image appears
When	During training (creates the practice exercises)	During generation (inference)
Who does it	A fixed mathematical process, nothing to learn	The trained neural network, guided by your text prompt

> 💡 **Think of it like a sculptor**
>
> Forward diffusion is like throwing sand over a statue until it disappears. The model watches this millions of times and learns how to brush the sand away. At generation time it starts from a pile of sand and "brushes" until a statue matching your description appears.

Multimodal models

A **multimodal** model works with more than one type of data (text, images, audio, video) as input and/or output. It can describe a photo, answer questions about a chart, read a scanned invoice, or create an image from text. Examples: Amazon Nova Lite/Pro, Anthropic Claude, GPT-4o, Gemini.

Generative adversarial networks (GANs)

Two networks compete. The **generator** (a counterfeiter) creates fake images from random noise; the **discriminator** (a police officer) tries to tell real from fake. Each improves by beating the other, until the fakes are realistic. GANs make very sharp images but are **hard to train** (unstable, can collapse to producing only one kind of output).

Variational autoencoders (VAEs)

An **encoder** compresses data into a small, smooth "map" of hidden features called the **latent space**; a **decoder** rebuilds data from any point on that map. Because the map is smooth, picking new points creates new, realistic samples. VAEs are **stable to train** but images tend to be **blurrier** than GANs. (Stable Diffusion uses a VAE to compress images, then runs diffusion in that compressed space.)

	GAN	VAE	Diffusion
Core idea	Generator vs discriminator competition	Compress to a latent space, then decode	Learn to remove noise step by step
Image quality	Sharp	Often blurry	Very high
Training	Unstable, tricky	Stable	Stable but compute-heavy
Generation speed	Fast (one pass)	Fast	Slower (many steps)
Used for	Face generation, super-resolution, style transfer	Anomaly detection, compression, drug discovery	Text-to-image, image editing, video

Amazon Bedrock at a glance

Amazon Bedrock is a fully managed service that gives you access to foundation models from Amazon and leading AI companies (e.g. Amazon Nova, Anthropic Claude, Meta Llama, Mistral AI, Cohere, AI21 Labs, Stability AI) through **a single API**, with no infrastructure to manage.

Bedrock capability	What it does
Model choice	Try and switch between many FMs; the Playground lets you compare them without code

Bedrock capability	What it does
Knowledge Bases	Managed RAG: connect your documents; Bedrock chunks, embeds, stores and retrieves them
Agents	Let a model plan multi-step tasks and call your APIs (e.g. check order status, then raise a refund)
Guardrails	Filter harmful content, block topics, redact personal information (PII), check answers are grounded
Model customisation	Fine-tune or continue pre-training some models privately with your data
Model evaluation	Compare models automatically or with human reviewers
Security & privacy	Your prompts and data are not used to train the base models; encryption and IAM controls

Optimising model outputs

A general-purpose model may not know your company, your format or your tone. There are four main ways to improve it, from cheapest to most expensive:

Technique	What changes	Effort & cost	Use when
1. Prompt engineering	Only the input (the instructions)	Lowest: minutes	Always start here: format, tone, examples, step-by-step reasoning
2. RAG	The model gets relevant documents at question time	Low–medium	Answers must use your private or frequently changing data, with sources
3. Fine-tuning	The model's weights, using your labelled examples	Medium–high	You need a consistent style, format or specialised task the prompt cannot achieve
4. Continued pre-training	The weights, using large unlabelled domain text	High	The model must learn a new domain's language (legal, medical, internal jargon)
(Builders) RLHF	The weights, using human preference rankings	High	Aligning a model to be helpful, honest and safe: done by model providers

Prompt engineering

A **prompt** is the input you give the model. **Prompt engineering** is designing that input to get better, more reliable output. A good prompt often contains:

- **Role / context**: "You are a customer-support agent for an Indian bank."
- **Clear task**: "Summarise the complaint in 3 bullet points."
- **Input data**: the complaint text, clearly separated (e.g. inside <complaint> tags).
- **Constraints and format**: "Under 60 words. Output JSON with keys issue, urgency."
- **Examples** of good answers, if the format is unusual.

Technique	Idea	Example
Zero-shot	Just ask, with no examples	"Classify this review as positive or negative: …"
Few-shot	Give a few worked examples first	"Great phone → positive. Battery died in a day → negative. Screen is stunning → ?"
Chain-of-thought	Ask for step-by-step reasoning before the answer	"Think step by step, then give the final amount."
Prompt template	A reusable prompt with blanks filled in by code	"Summarise {document} for a {audience} in {n} bullets."

Inference parameters also shape the output:

Parameter	Effect
Temperature	Randomness. Low (0–0.3) = focused and consistent (facts, code). High (0.7–1) = creative and varied (stories, brainstorming).
Top-p / Top-k	Limit choices to the most likely tokens (by cumulative probability, or to the top k tokens).
Max tokens	Maximum length of the response (also caps cost).
Stop sequences	Text that tells the model to stop generating.

⚠️ Prompt risks to know

- **Prompt injection**: hidden instructions in user input or documents ("ignore previous instructions and…") try to hijack the model. Treat retrieved content as data, not commands; use Bedrock Guardrails.
- **Jailbreaking**: tricking the model into bypassing safety rules.
- **Hallucination**: confident but wrong answers. Reduce with RAG, lower temperature, and asking the model to say "I don't know".

Fine-tuning and instruction fine-tuning

Fine-tuning continues training a pre-trained model on a smaller, task-specific **labelled** dataset so its weights adapt. **Instruction fine-tuning** is a special case: the dataset is pairs of **instruction → ideal response** (e.g. "Summarise this ticket" → a perfect summary). This is what turns a base model that merely continues text into an assistant that follows instructions. **Domain adaptation / continued pre-training** instead uses lots of unlabelled domain text.

Reinforcement learning from human feedback (RLHF)

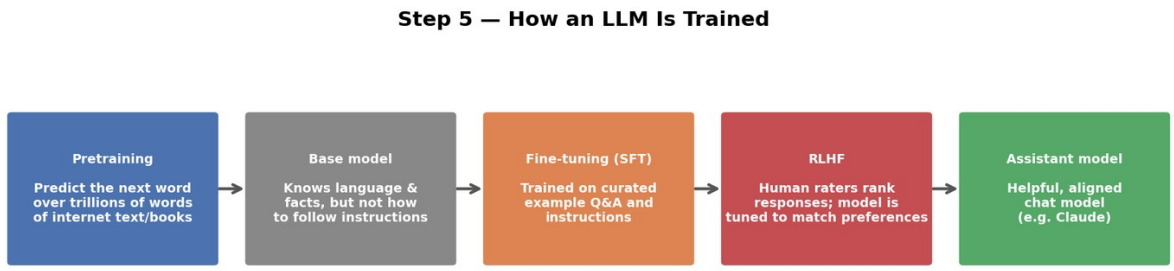

Figure 10. From raw text to a helpful assistant: pre-training → fine-tuning → RLHF.

1. The model gives several answers to the same prompt.
2. Human reviewers **rank** the answers from best to worst.
3. A **reward model** is trained to predict those human preferences.
4. The LLM is optimised with reinforcement learning to produce answers the reward model scores highly: more helpful, honest and harmless.

Retrieval-augmented generation (RAG)

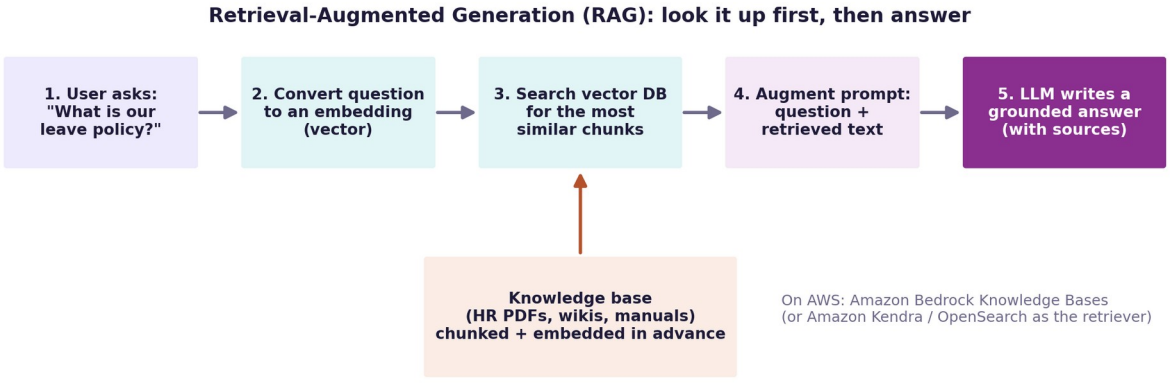

Figure 11. RAG: retrieve relevant documents first, then let the model answer from them.

RAG gives an LLM an "open-book exam". Your documents are split into chunks, turned into embeddings and stored in a vector database **in advance**. When a user asks a question, the system retrieves the most relevant chunks and adds them to the prompt, so the model answers from **your current, private data** and can cite its sources, **without retraining**. On AWS: **Amazon Bedrock**

Knowledge Bases (managed RAG), with Amazon OpenSearch Service, Aurora, S3 Vectors or Amazon Kendra as retrievers.

💡 Fine-tuning vs RAG in one line

Fine-tuning = sending the model back to school to change *how* it behaves. **RAG** = giving the model the right textbook page *during* the exam. Knowledge that changes often → RAG. Style, format or skill → fine-tuning.

✏️ Try it yourself: the prompt-engineering challenge (10 minutes)

Open any free AI chatbot (for example Claude, ChatGPT, Gemini or the Amazon Bedrock playground if you have an AWS account). Your task: get a good leave-application email to your head of department.

1. **Round 1:** type only "Write a leave application." Save the answer.
2. **Round 2:** add a role, the context (dates, reason), the tone and a length limit. Save the answer.
3. **Round 3:** also paste a short example of the style you like. Save the answer.
4. Compare the three. Which prompt element made the biggest difference? Write one sentence about what you learned.

Bonus: if your tool lets you change the temperature, run Round 3 at 0 and at 1. What changed?

Myth ❌	Fact ✅
ChatGPT searches a database for answers.	An LLM generates text token by token from learned patterns (unless a tool like RAG or web search is added).
Fine-tuning is the best way to add company knowledge.	For facts that change, RAG is cheaper, fresher and gives sources. Fine-tune for behaviour and style.
The model learns from my chat while I talk to it.	During inference weights do not change. On Bedrock your data is not used to train the base models.
A bigger model is always better.	Smaller models are faster and cheaper and are often good enough; pick the smallest that meets your quality bar.

🧠 Remember this

- **Foundation models** are huge, pre-trained on broad data, and adapted to many tasks. LLMs are foundation models for language.
- **Tokens** are pieces of text (pricing is per token). **Embeddings** are vectors that capture meaning.

- **Diffusion**: forward adds noise, reverse removes it to generate. **GAN** = faker vs judge. **VAE** = encoder → latent space → decoder.
- Improve outputs in this order: **prompt engineering → RAG → fine-tuning**. RLHF aligns models with human preferences.
- **Amazon Bedrock** = many foundation models through one API, plus Knowledge Bases, Agents and Guardrails.

Explain it back: close this guide and explain the module to a friend (or to yourself) in two minutes.

Your questions, answered

Q51 BASIC What is a foundation model, in simple words?

A: A giant, general-purpose model that has learned from a huge amount of data and can be reused for many tasks just by changing the prompt, like a very well-read graduate who can take on many different jobs.

Q52 BASIC Is an LLM the same as a foundation model? 🤔 *Guess first, then read*

A: An LLM is one **type** of foundation model, focused on language. Other foundation models generate images (diffusion models), create embeddings, or are multimodal (text + images + audio).

Q53 BASIC What is a token?

A: The unit of text an LLM reads and writes: a word or a piece of a word. “Artificial intelligence is amazing.” ≈ 5–6 tokens. Tokens matter because pricing and length limits are counted in tokens.

Q54 BASIC What is a prompt?

A: Everything you send to the model: your instruction, any context or documents, examples, and the question. Better prompts give better answers: be specific about the task, audience, format and length.

Q55 BASIC Why does an LLM sometimes give wrong answers so confidently? 🤔 *Guess first, then read*

A: Because it predicts the most **plausible** next words, not verified facts. If the likely-sounding continuation is false, it still sounds fluent. This is called **hallucination**. Reduce it with RAG, low temperature, clear instructions and human review for important use cases.

Q56 BASIC What is Amazon Bedrock used for?

A: To build generative AI applications (chatbots, summarisers, content generators, RAG assistants, AI agents) using ready-made foundation models through one API, without managing servers or training a model yourself.

Q57 INTERMEDIATE What is the difference between an embedding and a token?

A: A **token** is a piece of text with an ID number (e.g. "cat" → 5240). An **embedding** is a vector of many numbers that captures the token's (or sentence's) meaning, e.g. [0.21, −0.07, 0.55, ...]. IDs are just labels; embeddings carry meaning you can compare.

Q58 INTERMEDIATE How do we measure how similar two embeddings are?

A: Usually with **cosine similarity**: the angle between two vectors. Close to 1 = very similar meaning; close to 0 = unrelated. Vector databases run this comparison very fast across millions of vectors.

Q59 INTERMEDIATE What is a context window and why does it matter?

A: The maximum number of tokens the model can consider at once: your prompt, any documents, the conversation history and its answer. If your document is larger, you must summarise it, chunk it, or use RAG to send only the relevant parts. Larger contexts also cost more tokens.

Q60 INTERMEDIATE What is temperature?

A: A setting that controls randomness when choosing the next token. At 0 the model almost always picks the most likely token (consistent, good for facts and code). Higher values let less likely tokens through (more varied and creative, good for brainstorming, but more error-prone).

Q61 INTERMEDIATE What is the difference between forward and reverse diffusion?

A: **Forward diffusion** adds noise to training images step by step until they become pure noise; it is a fixed process used to create training examples. **Reverse diffusion** is what the trained model does: start with noise and remove it step by step, guided by the prompt, until an image appears.

Q62 INTERMEDIATE What makes a model "multimodal"?

A: It can take in or produce more than one kind of data: text, images, audio or video. E.g. upload a photo of a broken appliance and ask "what part is damaged?"; or type a description and get an image back.

Q63 INTERMEDIATE When should I use RAG instead of fine-tuning?

- Knowledge changes often (policies, prices, product catalogues) → **RAG**.
- You need citations / sources → **RAG**.
- You need a specific style, format, tone or narrow skill consistently → **fine-tuning**.
- Often the best systems use both: a fine-tuned model + RAG for fresh facts.

Q64 INTERMEDIATE What is instruction fine-tuning?

A: Training a pre-trained model on many examples of *instructions paired with ideal responses*, so it learns to follow instructions rather than just continue text. It is the step that turns a base model into a chat assistant.

Q65 INTERMEDIATE What is the difference between GANs and VAEs?

A: A GAN learns through **competition** (generator vs discriminator) and makes sharp images but is hard to train. A VAE learns by **compressing and reconstructing** through a smooth latent space; it is stable and good for anomaly detection, but images are blurrier.

Q66 ADVANCED Why can RLHF make a model more helpful than fine-tuning alone?

A: Fine-tuning needs a single "correct" answer for each example, but for open-ended questions many answers are acceptable and some are better than others. **Ranking** is easier for humans than writing perfect answers, and the reward model captures subtle preferences (clarity, honesty, safety). Optimising against it shapes overall behaviour, not just specific outputs. Newer methods such as DPO optimise on preferences directly without a separate reward model.

Q67 ADVANCED What are the main steps inside a RAG system, and where can it fail?

- **Ingestion**: split documents into chunks, create embeddings, store in a vector database. Failure: chunks too big or too small, lost tables.
- **Retrieval**: embed the question and find the top-k similar chunks. Failure: the relevant chunk isn't retrieved.
- **Augmentation**: insert chunks into the prompt. Failure: too much irrelevant context confuses the model.
- **Generation**: the LLM answers from the context. Failure: it ignores context or still hallucinates.

Improvements: better chunking, hybrid (keyword + vector) search, re-ranking, metadata filters, and instructing the model to answer only from the context and cite it.

Q68 ADVANCED Why does Stable Diffusion work in a "latent space"?

A: Running diffusion directly on millions of pixels is very expensive. A VAE first compresses the image into a much smaller latent representation; diffusion runs there (far fewer numbers), and the VAE decoder turns the result back into a full image. This is called **latent diffusion** and makes generation practical on normal GPUs.

Q69 ADVANCED How is a text prompt used to guide image generation?

A: The prompt is converted into an embedding by a text encoder. At every denoising step, the diffusion network receives this embedding (through cross-attention) and removes noise in a direction that matches the description. A "guidance scale" setting controls how strictly the image follows the prompt.

Q70 ADVANCED How do we evaluate a generative AI application?

- **Automatic metrics**: ROUGE (summaries), BLEU (translation), BERTScore (semantic similarity), accuracy on benchmark questions.
- **LLM-as-a-judge**: a strong model grades outputs against a rubric.
- **Human evaluation**: reviewers rate helpfulness, correctness, tone.
- **RAG-specific**: is the answer grounded in the retrieved context? Were the right chunks retrieved?

▸ **Operational**: latency, cost per request, safety incidents.

Amazon Bedrock model evaluation supports automatic, LLM-as-a-judge and human evaluation.

Q71 ADVANCED What is an AI agent, and how is it different from a chatbot?

A: A chatbot answers. An **agent** can **plan and act**: it breaks a goal into steps, calls tools or APIs (search, databases, booking systems), checks the results and continues until the task is done. E.g. "Reschedule my delivery and email me the new date." Amazon Bedrock Agents and Bedrock AgentCore help build and run agents securely.

Q72 ADVANCED What is "grounding", and why do Bedrock Guardrails check it?

A: An answer is **grounded** when every claim is supported by the provided source documents. Guardrails' contextual grounding check scores whether a response is supported by the retrieved context and relevant to the question, and can block ungrounded (likely hallucinated) answers in RAG applications.

✅ Quick quiz: Module 5

Tick your answers, then check the answer key →

1. What is a foundation model?

- ☒ A) A small model built for one task
- ☒ B) A very large model pre-trained on broad data that can be adapted to many tasks
- ☒ C) A database of answers
- ☒ D) A type of decision tree

2. In forward diffusion the model...

- ☒ A) Removes noise from an image
- ☒ B) Gradually adds noise to an image until it is pure noise
- ☒ C) Translates text
- ☒ D) Splits text into tokens

3. Your chatbot must answer from company policy documents that change every month. Best first approach?

- ☒ A) Pre-train a new model
- ☒ B) Retrieval-augmented generation (RAG)
- ☒ C) Increase the temperature
- ☒ D) Use a GAN

4. Which setting makes an LLM's answers more predictable and less creative?

- ☒ A) Higher temperature
- ☒ B) Lower temperature
- ☒ C) More output tokens
- ☒ D) Longer prompt

5. What do embeddings allow?

- ☒ **A)** Compressing video
- ☒ **B)** Comparing meaning mathematically, e.g. finding similar documents
- ☒ **C)** Encrypting data
- ☒ **D)** Counting words

Module 6: AWS Infrastructure and Technologies

📌 **What you will learn (about 40 minutes)**

- describe the three layers of the AWS AI/ML stack
- match each AWS AI service to a real business problem
- explain SageMaker AI, SageMaker JumpStart, Amazon Bedrock and Amazon Q
- list the advantages of AWS AI solutions and the main cost factors

The AWS AI/ML stack: three layers

Layer	Who it is for	Services
1. AI services (ready-made)	Developers with no ML expertise: call an API	Rekognition, Textract, Comprehend, Translate, Transcribe, Polly, Lex, Kendra, Personalize; generative: Amazon Q, Bedrock
2. ML platforms	Data scientists and ML engineers who build, train and deploy their own or customised models	Amazon SageMaker AI (incl. JumpStart, Canvas, Studio); Amazon Bedrock for FM customisation
3. Infrastructure & frameworks	Experts who need full control	EC2 GPU instances, AWS Trainium (training), AWS Inferentia (inference), frameworks: PyTorch, TensorFlow, Hugging Face, JAX

💡 **Think of it like getting a meal**

AI services = order from a restaurant: ready to eat. **ML platforms** = a fully equipped kitchen: you cook, but the tools are there. **Infrastructure** = buying raw ingredients and your own stove: full control, full responsibility.

ML framework: Amazon SageMaker AI

Amazon SageMaker AI is a fully managed service to **build, train and deploy ML models at scale**. It covers the whole ML lifecycle from Module 3:

Lifecycle step	SageMaker AI capability
Prepare & label data	Data Wrangler (visual data prep), Ground Truth (labelling), Feature Store

Lifecycle step	SageMaker AI capability
Build	SageMaker Studio (IDE, notebooks), built-in algorithms, Canvas (no-code ML), Autopilot (AutoML)
Start from pre-trained models	SageMaker JumpStart
Train & tune	Managed training jobs, distributed training, Managed Spot Training, Automatic Model Tuning
Check fairness & explain	SageMaker Clarify (bias detection, feature importance)
Deploy	Real-time, serverless, asynchronous endpoints; Batch Transform
Operate (MLOps)	Pipelines, Model Registry, Model Monitor

AWS AI services: what, when, example

Service	What it does	Real-world example
Amazon Comprehend	NLP: sentiment, entities, key phrases, language, topics, PII detection; custom classification	Analyse 50,000 product reviews to find top complaints
Amazon Translate	Neural machine translation across many languages, custom terminology	Show an e-commerce site in Hindi, Tamil and Marathi
Amazon Textract	Extract printed and handwritten text, forms (key-value) and tables from documents	Automate loan-application and invoice processing
Amazon Lex	Build chatbots and voice bots: intents, slots, speech and text	Bank bot: “What is my balance?”, “Block my card”
Amazon Polly	Text-to-speech with lifelike voices and SSML control	Narrate e-learning modules; IVR phone menus
Amazon Transcribe	Speech-to-text; speaker labels, custom vocabulary; Call Analytics	Subtitles for lectures; call-centre transcripts
Amazon Rekognition	Image and video analysis: objects, faces, text, unsafe content, PPE, custom labels	Detect workers without helmets on a site camera
Amazon Kendra	Intelligent enterprise search with natural-language questions; also a RAG retriever	Staff ask “How many casual leaves do I get?” across HR documents
Amazon Personalize	Real-time recommendations using the same kind of technology as Amazon.com	“Customers who watched this also watched…”

Service	What it does	Real-world example
AWS DeepRacer	Learn reinforcement learning by training a model race car in a simulator	Learning RL through racing

> ⚠️ **Status update: AWS DeepRacer**
>
> The AWS DeepRacer **console service was discontinued in December 2025**. DeepRacer continues as **"DeepRacer on AWS"**, an open-source solution you deploy in your own AWS account (training still runs on SageMaker AI). Older course material may still describe the console version.

> ✏️ **Try it yourself: memory hooks (5 minutes)**
>
> Cover the "What it does" column above. For each service name, say out loud what it does and give one example. A trick: **Polly** talks (a parrot!), **Transcribe** writes down what it hears, **Textract** extracts text, **Rekognition** recognises images, **Lex** is for talking bots, **Personalize** personalises. You will use these in the **Final challenge**.

Generative AI on AWS

Service	What it is	Use when
Amazon Bedrock	Fully managed access to many foundation models via one API, plus Knowledge Bases (RAG), Agents, Guardrails, fine-tuning and evaluation	You want to build a GenAI app fast without managing infrastructure
Amazon SageMaker JumpStart	ML hub inside SageMaker AI with hundreds of pre-trained models (including open-weight FMs such as Llama, Mistral, Falcon) and solution templates; deploy and fine-tune in a few clicks	You want more control: your own endpoint, instance choice, deeper fine-tuning, or open-weight models
Amazon Q Business	A generative AI assistant for employees, connected to company data (S3, SharePoint, Confluence, Salesforce…), respecting access permissions	Answer staff questions, summarise documents, automate simple tasks
Amazon Q Developer	A generative AI assistant for building and operating on AWS: chat in the AWS Console, code help, troubleshooting	Developers and cloud engineers (see the status note below)
Amazon Nova	Amazon's own family of foundation models (text, multimodal, image and video generation), available in Bedrock	Cost-effective models for many tasks

Service	What it is	Use when

⚠️ **Status update: Amazon Q Developer → Kiro**

AWS is retiring the **Amazon Q Developer IDE plugins and paid subscriptions**: new sign-ups stopped on **15 May 2026** and support ends on **30 April 2027**. The replacement for coding in the IDE is **Kiro**, an agentic IDE and CLI built around spec-driven development. **Amazon Q Developer inside the AWS Management Console and in Slack/Teams chat is not affected.** When studying for exams, check the current exam guide, as questions may still use the Q Developer name.

🎯 **Exam tip: Bedrock vs SageMaker AI (JumpStart)**

Bedrock = serverless, API access to managed FMs, least operational work. **SageMaker AI / JumpStart** = you choose and manage the deployment (instances, endpoints), more control and customisation. Keywords like "least operational overhead" or "no infrastructure to manage" point to Bedrock.

Advantages and benefits of AWS AI solutions

Benefit	What it means in practice
Accelerated development & deployment	Pre-trained AI services and FMs mean you can build in days, not months; no need to collect data and train from scratch.
Scalability & cost optimisation	Scale from one user to millions automatically; pay only for what you use.
Flexibility & model choice	Choose from many models and providers; switch as better or cheaper models appear.
Integration with AWS	Works with S3, Lambda, IAM, CloudWatch, databases and analytics you already use.
Security, privacy & compliance	Encryption, IAM, VPC/PrivateLink, many compliance certifications; Bedrock does not use your data to train base models.
Responsible AI tooling	Bedrock Guardrails, SageMaker Clarify, Model Monitor and human review with Amazon Augmented AI (A2I).

Cost considerations

Factor	What to know
Pricing model	AI services charge per use (per image, per minute of audio, per character, per page). LLMs charge **per input and output token**.
On-demand vs provisioned	Bedrock **on-demand**: pay per token, no commitment. **Provisioned Throughput**: reserve capacity for steady, high-volume or custom-model workloads.
Batch inference	For large jobs without an urgent deadline, batch is typically much cheaper than on-demand for supported models.
Model size	Larger models cost more per token and are slower. Use the smallest model that meets your quality bar.
Prompt length & caching	Long prompts and chat histories multiply input tokens. Keep prompts lean; use prompt caching for repeated context.
Customisation	Fine-tuning adds training cost, and custom models usually need provisioned capacity to run. Try prompt engineering and RAG first.
Training compute	SageMaker Managed Spot Training can cut training cost significantly (AWS cites up to 90%); Trainium and Inferentia are designed for better price-performance.
Endpoints	Real-time endpoints bill while running. Use serverless inference for spiky traffic; shut down idle endpoints and notebooks.
Latency & availability	Faster responses and cross-Region redundancy improve experience but may cost more: balance against business need.
Governance	Use AWS Budgets, Cost Explorer and cost-allocation tags to track spend per project.

✏️ Try it yourself: estimate the bill (5 minutes)

A college chatbot gets 2,000 questions a day. Each question uses about 500 input tokens (prompt + retrieved context) and 250 output tokens. Calculate the tokens per month, then look up current per-token prices for two models on the Amazon Bedrock pricing page. (Check your token total in the answer key.) How much would choosing the smaller model save? What happens if you cut the retrieved context in half?

🧠 Remember this

- Three layers: ready-made **AI services**, **ML platforms** (SageMaker AI, Bedrock), and **infrastructure** (GPUs, Trainium, Inferentia).
- Polly talks, Transcribe listens, Textract reads documents, Comprehend understands text,

Translate translates, Rekognition sees, Lex chats, Kendra searches, Personalize recommends.

- Bedrock = least operational work; SageMaker JumpStart = more control over models and endpoints.
- Costs depend on tokens, model size, on-demand vs provisioned vs batch, and idle endpoints.

Explain it back: close this guide and explain the module to a friend (or to yourself) in two minutes.

Your questions, answered

Q73 BASIC Do I need ML knowledge to use services like Rekognition or Polly?

A: No. They are pre-trained AI services: you send data through an API or the console and get results. AWS handles the models, training and scaling.

Q74 BASIC What is the difference between Amazon Polly and Amazon Transcribe? 🤔 *Guess first, then read*

A: They are opposites. **Polly**: text → speech (it talks). **Transcribe**: speech → text (it listens and writes).

Q75 BASIC What is the difference between Amazon Textract and Amazon Comprehend?

A: **Textract** *reads* documents: it extracts the text, forms and tables from scans and PDFs. **Comprehend** *understands* text: sentiment, entities, topics. They are often used together: Textract extracts, Comprehend analyses.

Q76 BASIC What is Amazon Lex, and is it related to Alexa?

A: Lex is a service for building conversational interfaces (chatbots and voice bots) with intents and slots. It uses the same kind of deep learning technology as Alexa for speech recognition and language understanding.

Q77 BASIC What is Amazon SageMaker AI?

A: AWS's fully managed platform for the whole ML lifecycle: prepare data, build, train, tune, deploy and monitor your own models. It includes no-code (Canvas), AutoML (Autopilot) and pre-trained model (JumpStart) options.

Q78 BASIC What is Amazon Q?

A: A family of generative AI assistants. **Amazon Q Business** answers employees' questions from company data. **Amazon Q Developer** helps developers and cloud engineers, e.g. in the AWS Console (its IDE plugins are moving to Kiro). Amazon Q also appears inside other AWS services such as Amazon QuickSight.

Q79 INTERMEDIATE Kendra vs Bedrock Knowledge Bases: which one for searching documents?

A: **Kendra** is an intelligent search service: it returns the most relevant documents and passages, with many ready-made connectors. **Bedrock Knowledge Bases** is managed RAG: it retrieves chunks *and* has an LLM write a natural-language answer. They can work together: Kendra (or OpenSearch) as the retriever, Bedrock to generate the answer.

Q80 INTERMEDIATE Personalize vs building my own recommendation model in SageMaker AI?

A: **Personalize** is fastest: upload interaction data (who clicked/bought what), and it trains and hosts recommendation models for you. Choose **SageMaker AI** when you need a custom algorithm, unusual data, or full control over the model.

Q81 INTERMEDIATE When should I use Amazon Bedrock versus SageMaker JumpStart?

- **Bedrock**: serverless, pay per token, many managed models, built-in RAG, agents and guardrails: least operational work.
- **JumpStart**: deploy a model on instances you choose in your account; more control over hosting, open-weight models and deeper fine-tuning, but you manage and pay for endpoints while they run.

Q82 INTERMEDIATE How do Trainium and Inferentia help?

A: They are AWS-designed chips. **Trainium** is optimised for training deep learning models; **Inferentia** for running inference. Both aim to give better price-performance than general-purpose GPUs for supported workloads, available through EC2 instances and SageMaker AI.

Q83 INTERMEDIATE Is my data safe when I use Amazon Bedrock?

A: Bedrock does not use your prompts or data to train the base models and does not share them with model providers. Data is encrypted in transit and at rest, you control access with IAM, and you can keep traffic private using VPC endpoints. You are still responsible for what data you send and for access policies (the shared responsibility model).

Q84 INTERMEDIATE What happened to AWS DeepRacer?

A: The DeepRacer console service ended in December 2025. DeepRacer continues as an open-source "DeepRacer on AWS" solution that you deploy in your own account, still using SageMaker AI for reinforcement learning training. It remains a great way to *understand* RL: reward functions, actions and trial and error.

Q85 ADVANCED How would you design a document-processing pipeline on AWS for insurance claims?

- Upload claim PDFs and photos to **Amazon S3** (triggers **AWS Lambda**).
- **Textract** extracts forms and tables; **Rekognition** analyses damage photos.
- **Comprehend** detects entities and PII; redact sensitive fields.
- A **Bedrock** model summarises the claim and flags missing information; **Guardrails** filter output.
- Low-confidence cases go to human reviewers with **Amazon A2I**.

- Store results in a database; monitor with CloudWatch.

Q86 ADVANCED How can you reduce the cost of a production GenAI application?

- Pick the smallest model that passes your evaluation; route only hard questions to a larger model.
- Trim prompts and retrieved context; summarise long chat histories; use prompt caching.
- Use batch inference for non-urgent jobs.
- Cache answers to frequent questions.
- Prefer prompt engineering and RAG before fine-tuning; use provisioned throughput only for steady, high volume.
- Set max tokens; monitor spend with AWS Budgets and tags.

Q87 ADVANCED What does the shared responsibility model mean for AI on AWS?

A: AWS is responsible for security **of** the cloud: data centres, hardware, the managed service and model infrastructure. You are responsible for security **in** the cloud: IAM permissions, what data you send, encryption settings, network configuration, guardrails, and how you use model outputs (checking for bias, accuracy and compliance).

Q88 ADVANCED How do Bedrock Agents and Knowledge Bases work together?

A: A Knowledge Base supplies facts (RAG), while an Agent decides *what to do*: it breaks the user's goal into steps, queries the Knowledge Base when it needs information, and calls action groups (your APIs via Lambda) to act, e.g. look up an order, check the return policy, then create a return request.

✅ Quick quiz: Module 6

Tick your answers, then check the answer key →

1. Which service extracts text, tables and form fields from scanned invoices?

- ☒ A) Amazon Polly
- ☒ B) Amazon Textract
- ☒ C) Amazon Translate
- ☒ D) Amazon Lex

2. You need a customer-support voice bot that understands intents like "check balance". Which service?

- ☒ A) Amazon Lex
- ☒ B) Amazon Kendra
- ☒ C) Amazon Personalize
- ☒ D) AWS Trainium

3. Where would you find pre-trained models and solution templates to deploy with one click in SageMaker AI?

- ☒ **A)** SageMaker JumpStart
- ☒ **B)** Amazon Polly
- ☒ **C)** AWS Glue
- ☒ **D)** Amazon Comprehend

4. Which is usually the cheapest way to run millions of LLM requests that don't need an immediate answer?

- ☒ **A)** Provisioned throughput
- ☒ **B)** On-demand real-time calls
- ☒ **C)** Batch inference
- ☒ **D)** A larger model

5. An employee assistant that answers questions from the company's SharePoint, Confluence and S3 data is...

- ☒ **A)** Amazon Q Business
- ☒ **B)** Amazon Rekognition
- ☒ **C)** Amazon Transcribe
- ☒ **D)** AWS DeepRacer

Final challenge: Which AWS service?

You have reached the end of the six modules. Time to test yourself! Write the AWS service you would choose for each problem, then check your score in the answer key. **Aim for 10 out of 12.**

#	Scenario	Your answer
1	A hospital wants to turn doctors' dictated notes into text.	
2	An e-learning company wants course text read aloud in a natural voice.	
3	A bank must extract names, amounts and tables from scanned loan forms.	
4	A retailer wants "recommended for you" products on its app.	
5	A news site wants every article available in 5 Indian languages.	
6	A brand wants to know if tweets about it are positive or negative.	
7	A construction firm wants to detect workers without helmets in CCTV images.	
8	Employees want to ask questions over HR policies stored in SharePoint.	
9	A startup wants a GenAI chatbot using Claude or Nova with no servers to manage.	
10	A data-science team wants to deploy and fine-tune an open-weight Llama model on its own endpoint.	
11	A telecom wants a voice bot to handle "recharge my plan" calls.	
12	An analyst with no coding skills wants to build a churn-prediction model.	

Match the term

Write the matching letter next to each term. Answers are in the answer key.

Term	Meaning
1. Token	A. Retrieve relevant documents and add them to the prompt
2. Embedding	B. Humans rank answers to align a model
3. RAG	C. A piece of text the model reads or writes

Term	Meaning
4. RLHF	D. Starting from noise and removing it step by step
5. Reverse diffusion	E. A vector that represents meaning
6. Inference	F. Using a trained model to make predictions

Quick recall

Answer each question in one or two words, in your head or on paper. Cover the answer column first, then check. Try again tomorrow: recalling after a gap is the best way to remember.

#	Question	Answer (cover me!)
1	The smallest circle ChatGPT belongs to?	Generative AI
2	Data with answers attached is called…	Labelled data
3	Emails, images and audio are which type of data?	Unstructured
4	Using a trained model on new data is called…	Inference
5	Nightly scoring of all customers uses which inference?	Batch
6	The numbers a neural network learns are called…	Weights (parameters)
7	Architecture behind LLMs?	Transformer
8	Text → speech on AWS?	Amazon Polly
9	Speech → text on AWS?	Amazon Transcribe
10	Managed access to many FMs via one API?	Amazon Bedrock
11	Setting that controls creativity?	Temperature
12	Confident but wrong model output?	Hallucination
13	Two networks, faker vs judge?	GAN
14	Encoder → latent space → decoder?	VAE
15	Adds noise step by step?	Forward diffusion
16	Cheapest first step to improve LLM output?	Prompt engineering
17	AWS chip for training?	AWS Trainium
18	AWS chip for inference?	AWS Inferentia
19	SageMaker AI hub of pre-trained models?	SageMaker JumpStart

#	Question	Answer (cover me!)
20	Learning from rewards and penalties?	Reinforcement learning

Self-check: tick what you can do

Be honest! Click each box you can do confidently. Anything unticked? Go back to that module and re-read its 🧠 Remember this box.

- ☒ I can explain the difference between AI, ML, deep learning and generative AI.
- ☒ I can tell labelled from unlabelled data and structured from unstructured data.
- ☒ I can describe the steps of the ML process.
- ☒ I can choose between batch and real-time inference.
- ☒ I can explain how a neural network learns.
- ☒ I can name three computer vision and three NLP tasks.
- ☒ I can explain foundation models, tokens and embeddings.
- ☒ I can explain forward and reverse diffusion.
- ☒ I can compare prompt engineering, RAG, fine-tuning and RLHF.
- ☒ I can match at least 8 AWS AI services to a use case.
- ☒ I can explain when to use Amazon Bedrock vs SageMaker JumpStart.
- ☒ I can list the main cost factors of a GenAI application.

Glossary

Term	Meaning
Activation function	Adds non-linearity to a neuron's output (e.g. ReLU).
Backpropagation	Calculating how each weight contributed to the error, working backwards through the network.
Batch inference	Predictions for a large dataset at once, on a schedule.
Context window	Maximum tokens a model can consider at once.
Diffusion model	Generates data by learning to reverse a noising process.
Embedding	A vector that represents meaning; similar meanings are close together.
Epoch	One full pass through the training data.
Feature	An input variable used by a model.
Fine-tuning	Further training a pre-trained model on task-specific data.
Foundation model	Large model pre-trained on broad data, adaptable to many tasks.
GAN	Generative adversarial network: generator vs discriminator.
Hallucination	Fluent but false or unsupported model output.
Hyperparameter	A training setting chosen before training (e.g. learning rate).
Inference	Using a trained model to produce outputs.
Label	The correct answer attached to a training example.
LLM	Large language model: a foundation model for text.
Multimodal model	Handles more than one data type (text, image, audio, video).
Overfitting	Model memorises training data and fails on new data.
Prompt engineering	Designing inputs to get better model outputs.
RAG	Retrieval-augmented generation: retrieve documents, then generate an answer from them.
RLHF	Reinforcement learning from human feedback.
Temperature	Controls randomness of generated text.
Token	Unit of text processed by an LLM.
Transformer	Neural network architecture based on self-attention.

Term	Meaning
VAE	Variational autoencoder: encoder, latent space, decoder.
Vector database	Stores embeddings and finds the most similar ones quickly.

Answer keys

Each answer links back to its module.

Module 1 answers

1. B The course builds the foundation (terms and relationships) and shows real AWS services that use AI/ML.

2. B Generative AI produces new content; it does not replace classic ML.

3. B Value comes from automation, better decisions from data, and new experiences.

← Back to the module

Module 2 answers

1. B Deep learning is a subset of ML. Not all AI learns (rule-based systems), and most ML is not generative.

2. B It learns patterns from labelled historical data.

3. B "Deep" refers to the number of layers in the neural network.

4. C Creating new text is generative. The others predict, detect or group.

5. B ML learns the rules (the model) from examples of data with answers.

← Back to the module

Module 3 answers

1. B It has answers (labels) and images are unstructured data.

2. B Finding groups (clustering) without labels is unsupervised.

3. B It memorised the training data and does not generalise.

4. B Large volume, no waiting user, scheduled: batch inference (SageMaker Batch Transform).

5. C The test set is kept aside until the end and never used for training or tuning.

← Back to the module

Module 4 answers

1. B Weights scale each input, like stronger or weaker synapses. Training adjusts them.

2. B Detection = what + where (bounding boxes).

3. B NER finds and labels names of people, organisations, places, dates.

4. B It sends the error backwards through the network to calculate each weight's correction.

5. B Transformers use self-attention and process whole sequences in parallel.

← Back to the module

Module 5 answers

1. B FMs are pre-trained on huge, broad datasets and adapted via prompting, RAG or fine-tuning.

2. B Forward = add noise (training setup). Reverse = learn to remove it (generation).

3. B RAG retrieves current documents at question time; no retraining needed when documents change.

4. B Low temperature makes the model pick the most likely tokens.

5. B Embeddings are vectors; similar meanings are close together, which powers semantic search and RAG.

← Back to the module

Module 6 answers

1. B Textract goes beyond OCR: it understands forms and tables.

2. A Lex builds conversational interfaces with speech and text (same tech as Alexa).

3. A JumpStart is SageMaker AI's ML hub of pre-trained models and solutions.

4. C Batch inference is typically discounted versus on-demand for supported models.

5. A Q Business connects to enterprise data sources and respects user permissions.

← Back to the module

Final challenge answers

1. Amazon Transcribe (Transcribe Medical)

2. Amazon Polly

3. Amazon Textract

4. Amazon Personalize

5. Amazon Translate

6. Amazon Comprehend

7. Amazon Rekognition (PPE detection)

8. Amazon Q Business (or Kendra / Bedrock Knowledge Bases)

9. Amazon Bedrock

10. Amazon SageMaker JumpStart

11. Amazon Lex (+ Polly / Transcribe)

12. Amazon SageMaker Canvas

Match the term: 1-C, 2-E, 3-A, 4-B, 5-D, 6-F

Try-it-yourself answers

Module 2, sort the products: (1) AI (rule-based), (2) ML, (3) Deep learning, (4) Generative AI, (5) ML, (6) Deep learning (a Transformer; it also generates text), (7) Generative AI (a diffusion model).

Module 3, be the model: A = Apple, B = Banana, C = Orange, D = Banana is the most likely answer (yellow like the bananas, but medium-sized and heavier). Fruit D is deliberately tricky: it could be a

yellow apple or a mango, which your model has never seen. Real models also struggle with examples unlike their training data.

Module 6, estimate the bill: 2,000 questions × 30 days × 750 tokens = **45 million tokens per month** (30 million input + 15 million output). Output tokens usually cost more than input tokens, so check both prices.