Module 16: Neural Networks and Deep Learning: Part Two: Advanced Concepts

Learning outcomes:

1. Analyse the key concepts of deep learning and neural networks.
2. Analyse the five building blocks of deep learning.
3. Select the most suitable coding libraries and packages for addressing a deep learning problem.
4. Construct neural networks with decision boundaries.
5. Assess the feasibility of using deep learning to solve specific business challenges.

Deep learning and neural networks

Deep learning is a subset of machine learning that uses multi-layered neural networks to model complex, non-linear patterns. Neural networks are inspired by biological neurons, but they rely on mathematical operations such as weighted sums, activations and backpropagation.

Key breakthroughs, such as **ImageNet classification**, demonstrated deep learning's potential, pushing it into widespread use across industries.

The building blocks of deep learning

Neural networks are constructed from a few core components:

Building block	Function	Notes
Inputs	Accept data features for the model	Size depends on the data set
Weights and biases	Parameters adjusted during training	Define how signals are transformed
Activation functions	Introduce non-linearity	Examples: ReLU, sigmoid, tanh
Layers	Organise transformations (hidden, output)	Depth increases model capacity
Loss function	Measures error in predictions	Guides weight updates

Example: ImageNet and benchmarks

- **ImageNet (2012, AlexNet)** marked a breakthrough with deep convolutional networks dramatically outperforming traditional approaches.
- **Human benchmarks** remain an important comparison, as models aim to match or exceed human-level performance in specific tasks.

Frameworks for implementation

- **PyTorch**: flexible, Pythonic, widely used in research. Strong GPU acceleration and debugging support
- **TensorFlow**: scales well to production environments with tools such as TensorFlow Serving.

Tip: PyTorch is often chosen for experimentation, while TensorFlow is often chosen for deployment.

Training neural networks

Training involves adjusting weights to minimise loss:

- **Forward pass:** computes predictions from inputs
- **Backward pass (backpropagation):** computes gradients using the chain rule
- **Optimisers (e.g. SGD, Adam):** update weights based on gradients

Trade-offs to manage

- Complexity vs computation cost
- Generalisability vs overfitting
- Interpretability vs accuracy

Industry applications of deep learning

Deep learning powers applications across fields:

Industry	Use case	Example
Healthcare	Medical imaging analysis	Detecting tumours from scans
Finance	Fraud detection	Monitoring transaction patterns
Retail	Recommendation engines	Personalised product suggestions
Transportation	Autonomous vehicles	Vision-based navigation