OxML 2025 Practical Tutorial:
Deep Learning and Representation Learning

Dr. Hao Liang
hao.liang@kcl.ac.uk
King's College London

Outline

- **Part I: Representation Learning Basics**
- **Part II: Deep Learning with MNIST and CIFAR-10**

Link and Slides

- Link for Part I
- Link for Part II

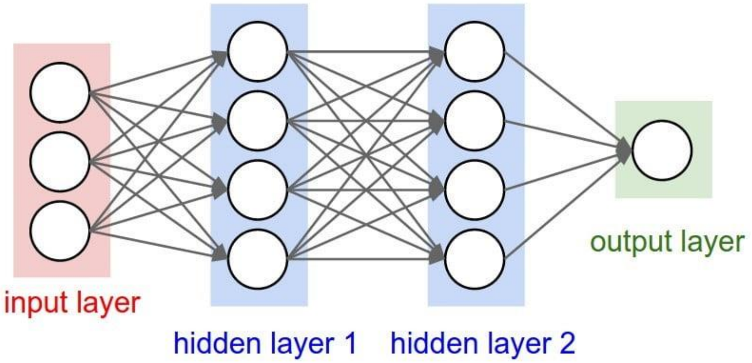
Objectives for today

- Built a tiny neural-network framework
- Trained a PyTorch model on MNIST digits
- Extended to a convolutional network on CIFAR-10 images

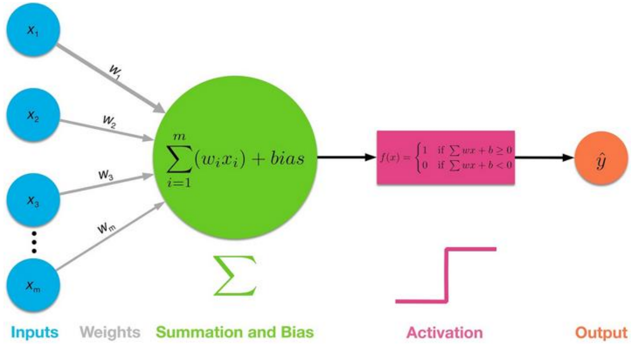
Part I: Representation Learning Basics

Neural Network Basics

- **Structure**
  - Input Layer: Receives raw data.
  - Hidden Layers: Perform transformations.
  - Output Layer: Produces predictions.



- **Components/Parameters**
  - **Weights:** Parameters learned during training.
  - **Biases:** Allow shifting of activation functions.



- **Forward Propagation:** Data flows forward, passing through layers, transforming inputs into outputs.

Activation Functions

- **Why Activation Functions?**
  - Introduce non-linearities to capture complex patterns.
- **Common Choices**
  - **Sigmoid**
  - **Tanh**
  - **ReLU (Rectified Linear Unit)**

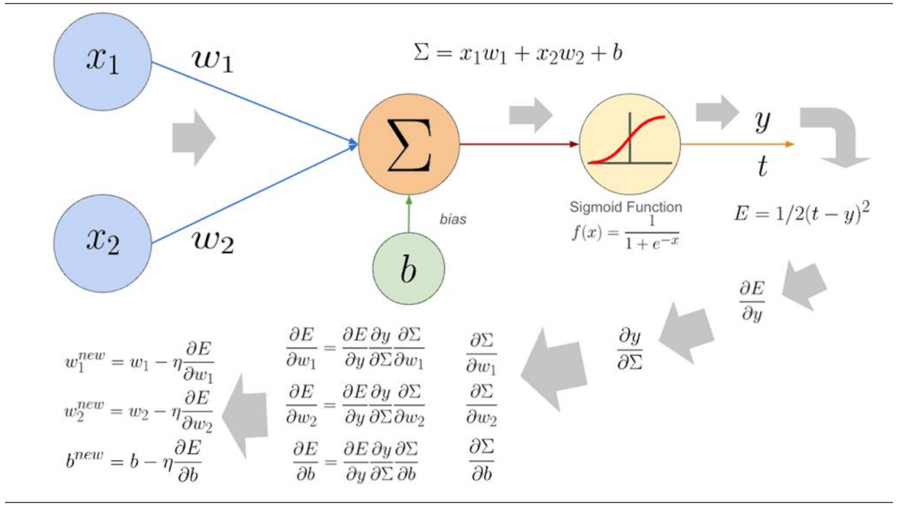
Loss Functions

- **Role:** Quantify discrepancy between predictions and actual labels.

- **Types**
  - **Mean Squared Error (MSE)**
  $$\mathbf{MSE}(y, y') = (y - y')^2$$
  - **Cross-Entropy Loss**
  $$\mathbf{CE}(y, y') = -y\log(y') - (1 - y)\log(1 - y')$$

Backpropagation & Gradient Descent

- **Backpropagation (Day 1)**
  - Employs chain rule to compute gradients efficiently.
  - Gradients propagate backward from output to input layers.
- **Gradient Descent Process (Day 1)**
  - Gradients inform parameter updates to reduce loss.
  - Optimization strategies: adjusting learning rate, parameter initialization.

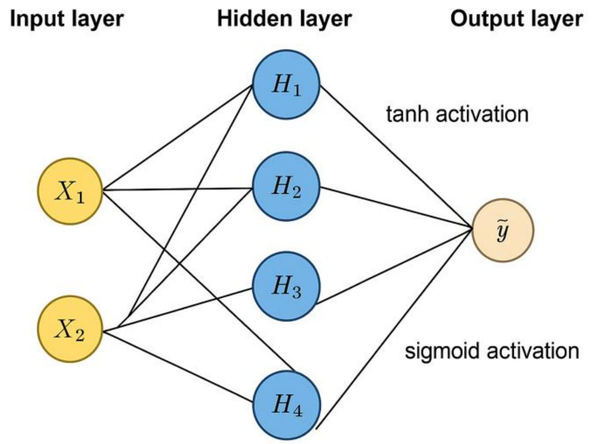


Coding Practical

- **Hands-on Exercise: Solve XOR binary classification problem step-by-step.**

x_1	x_2	$y = x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

$$y = x_1 \oplus x_2$$



- **Visualization of Training Loss over Epochs**
- **PyTorch Comparison**
  - Simplified code structure.
  - Built-in automatic differentiation.

Part II: Deep Learning with MNIST and CIFAR-10