Introduction to Deep Learning

Dr. Yaohua Zang & Vincent Scholz

Professorship for Data-driven Materials Modeling
Technical University Munich

April 23, 2025

Contact Information

- **Professor:** Prof. Koutsourelakis
- **Instructor:** Dr. Yaohua Zang, Vincent Scholz
- **Email:** yaohua.zang@tum.de
- **Office Location:** ED Building, Room 0438
- **Course Website:**
  `https://www.moodle.tum.de/course/view.php?id=107803`

Motivation & Applications

- **Speech Recognition**
  - Siri, Google Assistant
- **Computer Vision**
  - Image classification, object detection
- **Natural Language Processing**
  - ChatGPT, translation services
- **Scientific Applications**
  - Drug discovery, genomics, physics
- **Autonomous Vehicles**
- **Medical Diagnosis**

Figure: AI applications across industries

Foreshadowing (I)

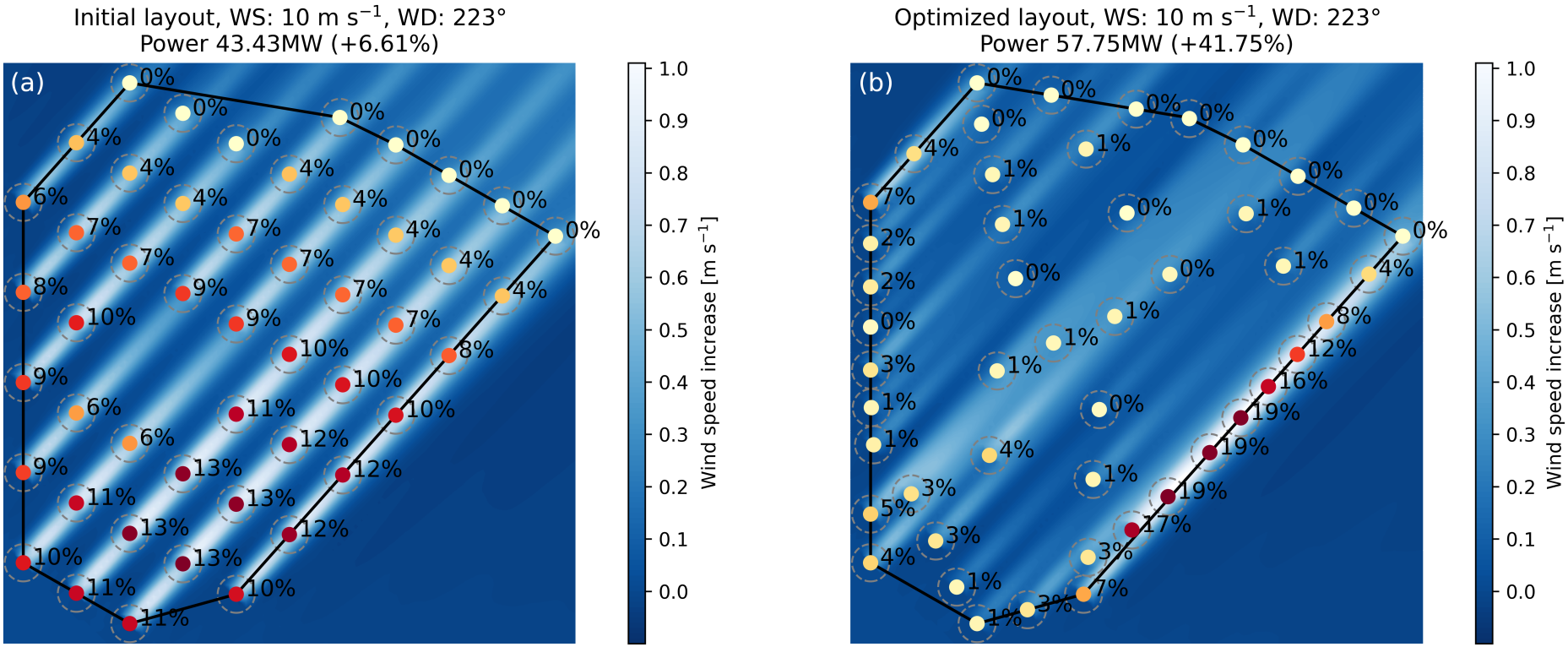


Figure: Example wind farm optimization.

Foreshadowing (II)

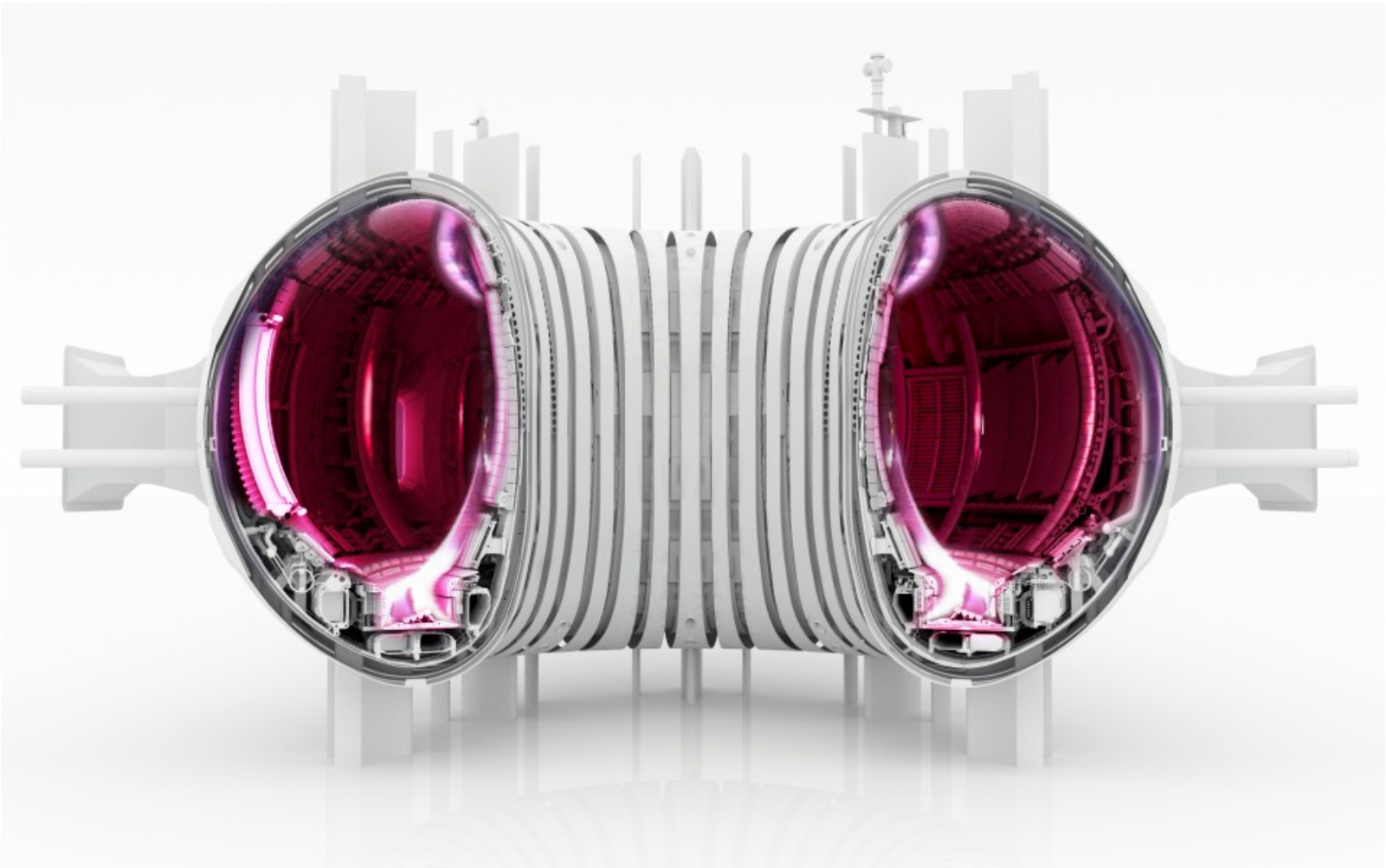


Figure: Example JET Fusion reactor.

Recommended Literature

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Chollet, F. (2021). *Deep Learning with Python*. Manning Publications.
- Aggarwal, C. C. (2018). *Neural Networks and Deep Learning: A Textbook*. Springer.
- Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). *Dive into Deep Learning*. Cambridge University Press.

Universal Approximation Theorem

Theorem (Universal Approximation Theorem)

A sufficiently large neural network with at least one hidden layer can approximate any continuous function to an arbitrary degree of accuracy

- **Mathematical formulation:**
  - For any continuous function $u : K \to \mathbb{R}$ in a compact $K \subset \mathbb{R}^d$ and $\epsilon > 0$, there exists a MLP $NN(x; \theta)$ such that:

$$\sup_{x \in K} |u(x) - NN(x, \theta)| < \epsilon \tag{1}$$

- Explains why neural networks are universal function approximators

Table of Contents

Theorem (Universal Approximation Theorem)

A sufficiently large neural network with at least one hidden layer can approximate any continuous function to an arbitrary degree of accuracy

Table of Contents

Theorem (Universal Approximation Theorem)

A sufficiently large neural network with at least one hidden layer can approximate any continuous function to an arbitrary degree of accuracy

Artificial Neural Networks (ANNs)

- **Core of deep learning**
- Enable computers to:
  - Recognize complex patterns
  - Solve intricate problems
  - Adapt to dynamic environments
- **Revolutionary applications:**
  - Natural Language Processing
  - Self-Driving Vehicles
  - Medical Diagnosis
  - Automated Decision-Making

Evolution of Neural Networks

- **1940s-1950s:** McCulloch & Pitts - first mathematical model
- **1960s-1970s:** Rosenblatt - Perceptron (linearly separable problems)
- **1980s:** Rumelhart, Hinton, Williams - Backpropagation algorithm
- **1990s:** Applications in image recognition; "AI winter"
- **2000s:** GPU advancements, larger datasets, new architectures
- **2010s-Present:** Deep learning dominates AI
  - CNNs for image recognition
  - RNNs for sequential data
  - Transformers for NLP (ChatGPT, BERT)

Basic Structure of Neural Networks

- **Inspired by human brain**
- Each neuron performs mathematical operations
- Multiple layers of neurons
- **Components:**
  - Input Layer
  - Hidden Layers
  - Output Layer
  - Connections

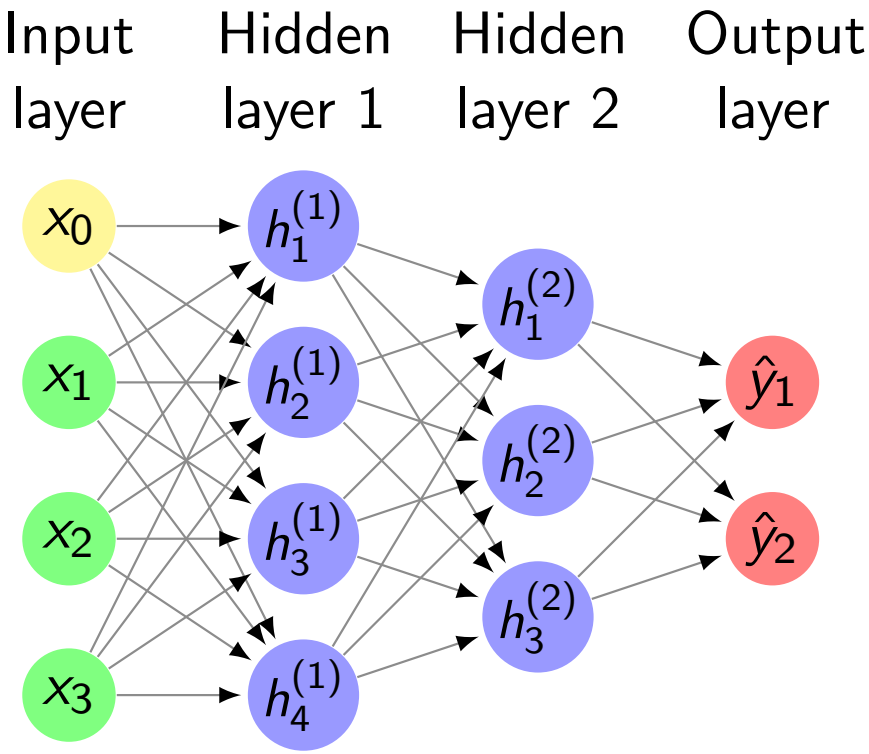


Figure: Neural Network Architecture

Key Components of Neural Networks

- **Neurons:** Fundamental processing units
  - Governed by threshold and activation function
- **Connections:** Links between neurons
  - Transmit information between layers
- **Weights and Biases:** Control signal strength
  - Adjusted during learning process
- **Activation Functions:** Introduce non-linearity
  - Enable modeling of complex patterns

Single-Layer Perceptron (SLP)

- Simplest neural network (Rosenblatt, 1958)
- Solves simple problems (AND, OR, NOR gates)
- **Operation:**
  - Takes inputs $\{1, x_1, \ldots, x_n\}$
  - Applies weights and bias
  - Passes through activation function
- Mathematical representation:
  $y = f(\sum_{i=1}^{n} w_i x_i + b_0 * 1)$

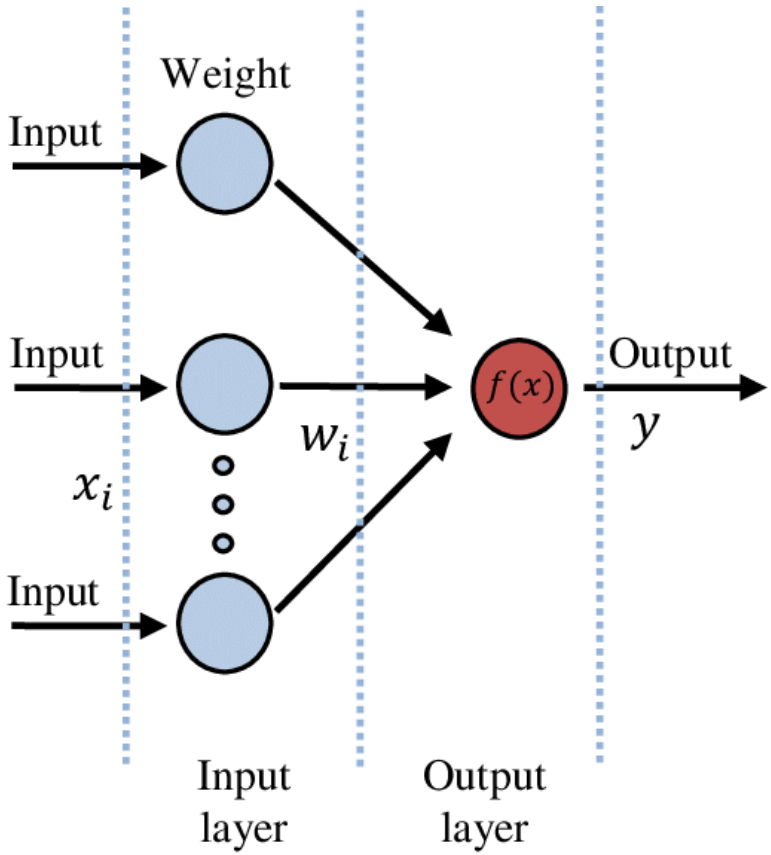


Figure: Single-layer perceptron.

Activation Functions

- **ReLU (Rectified Linear Unit):**
  - $f(x) = \max(0, x)$
  - Mitigates vanishing gradient problem
  - Most popular in modern networks
- **Tanh (Hyperbolic Tangent):**
  - $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
  - Values between -1 and 1
  - Common in hidden layers
- **Sigmoid:**
  - $f(x) = \frac{1}{1+e^{-x}}$
  - Used in binary classification

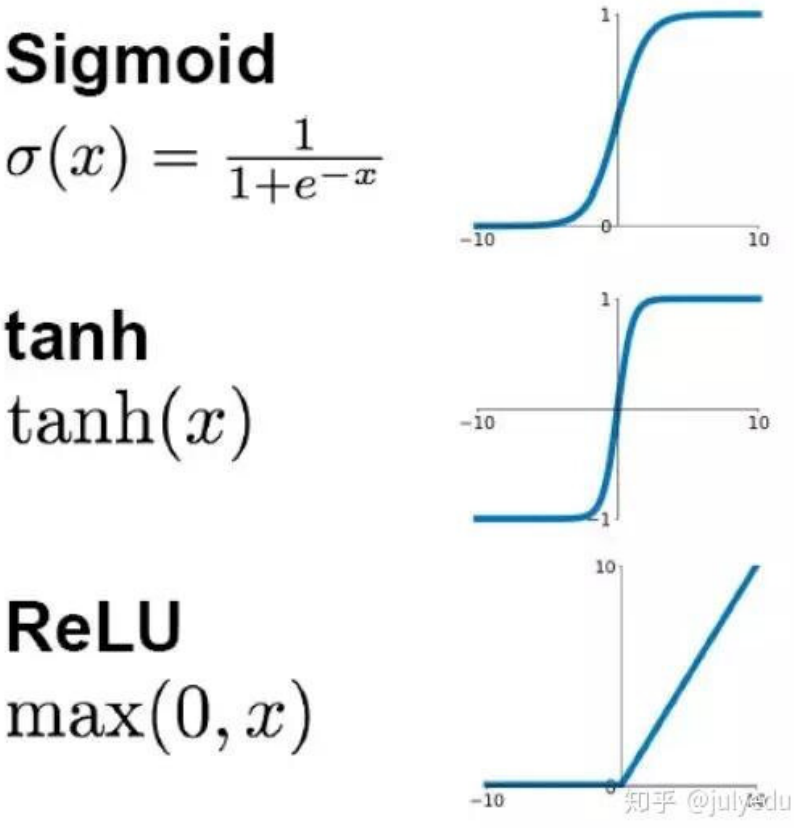


Figure: Some common activation functions.

Multi-Layer Perceptrons (MLP)

- Fully connected dense layers
- Transforms input data across dimensions
- **Components:**
  - Input layer
  - One or more hidden layers
  - Output layer
- Models complex input-output relationships
- Mathematically:
  $h_j^{(l)} = f_l(\sum_{i=1}^{m} w_{i,l} x_i + b_{j,l} * 1)$

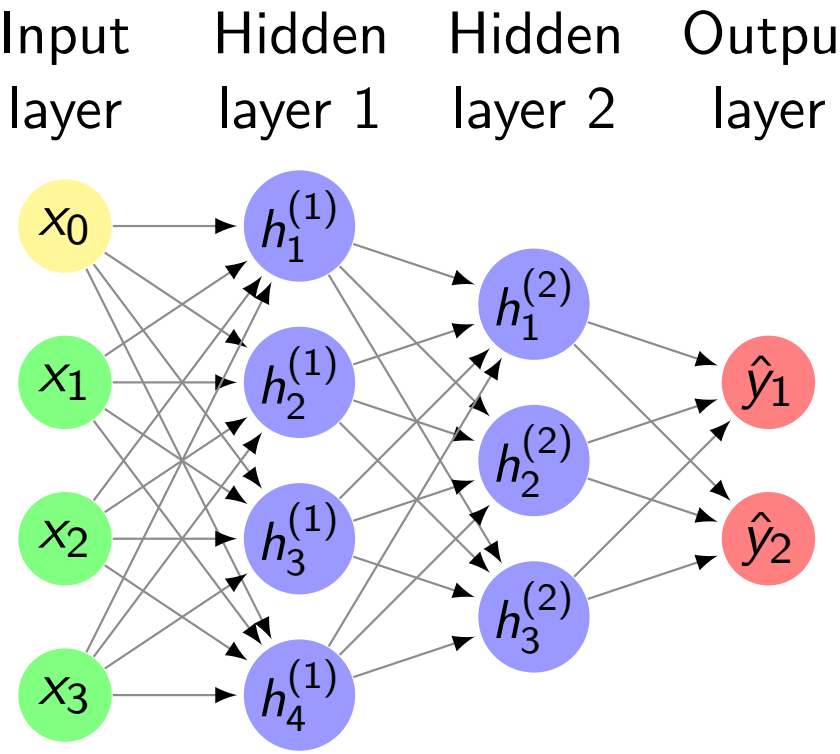


Figure: Multy-layer perceptron

Mathematical Representation of MLP

- Output of j-th neuron in l-th layer:

$$h_j^{(l)} = f_l(\sum_{i=1}^{m} w_{i,l} x_i + b_{j,l} * 1) \tag{2}$$

where:

- $h_j^{(l)}$ is neuron output
- $w_{i,l}$ are connection weights
- $b_{j,l}$ is bias term
- f_l is activation function

- Complete network representation:

$$y = NN(x;\theta) = W^{(l)} f^{(l)} \odot (W^{(l-1)} \odot f^{(l-1)}(\cdots) + b_{l-1}) + b_l \tag{3}$$

where θ represents all parameters

Table of Contents

Theorem (Universal Approximation Theorem)

A sufficiently large neural network with at least one hidden layer can approximate any continuous function to an arbitrary degree of accuracy

What is Deep Learning?

- Subset of Machine Learning
- **Utilizes multiple processing layers**
- Enables computational models to automatically learn features and patterns
- No manual feature engineering required
- Learns hierarchical data representations

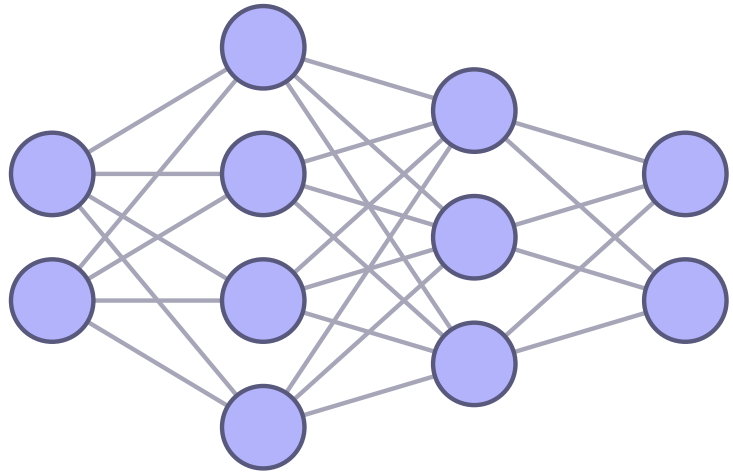
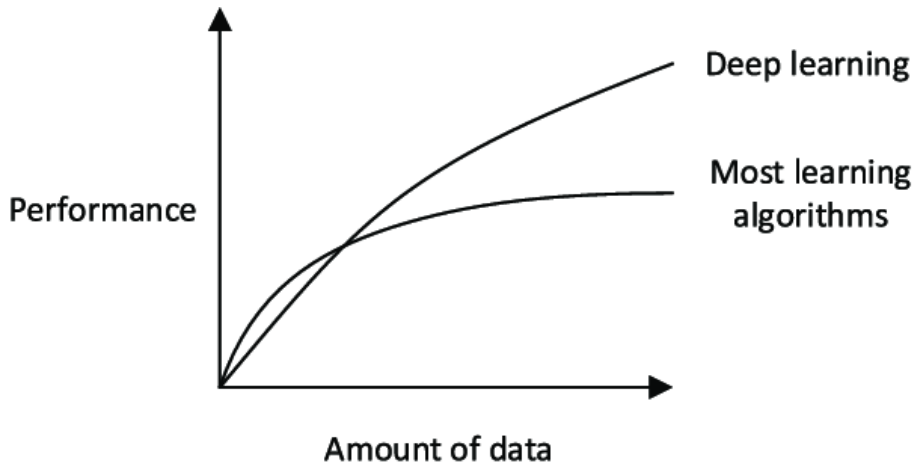


Figure: Deep Neural Network Architecture

Deep Learning vs. Machine Learning

Feature	**Machine Learning**	**Deep Learning**
Feature Extraction	Manual feature engineering	Automatic feature learning
Scalability	Struggles with large datasets	Excels with large datasets
Computation	Less computing power	Requires GPUs/high-performance
Performance (Complex Tasks)	Limited	Superior for image/speech/text



Advantages of Deep Learning

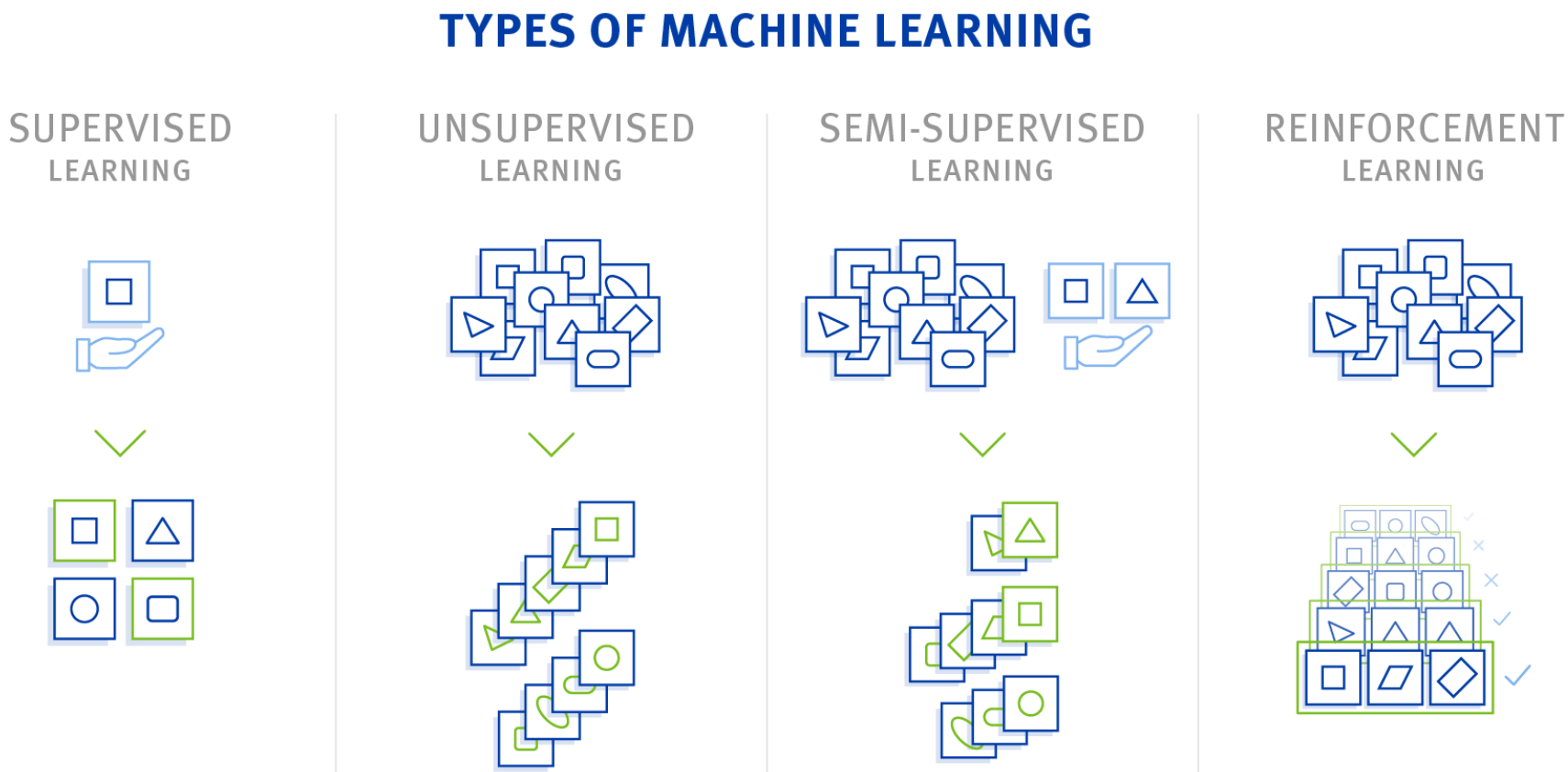
- **Handles large-scale data efficiently**
  - Performance improves with more data
- **Learns directly from raw data**
  - Eliminates manual feature engineering
- **Excels in complex tasks**
  - Image recognition
  - Speech processing
  - Autonomous systems
- **Better generalization**
- **Higher accuracy** with sufficient data

Table of Contents

Theorem (Universal Approximation Theorem)

A sufficiently large neural network with at least one hidden layer can approximate any continuous function to an arbitrary degree of accuracy

Overview of Learning Paradigms



Supervised Learning

- Model learns from **labeled input-output pairs**
- Predicts outputs based on inputs
- Compares predictions with known outputs
- Adjusts parameters to **minimize errors**
- **Applications:**
  - Image classification
  - Speech-to-text conversion
  - Medical diagnosis

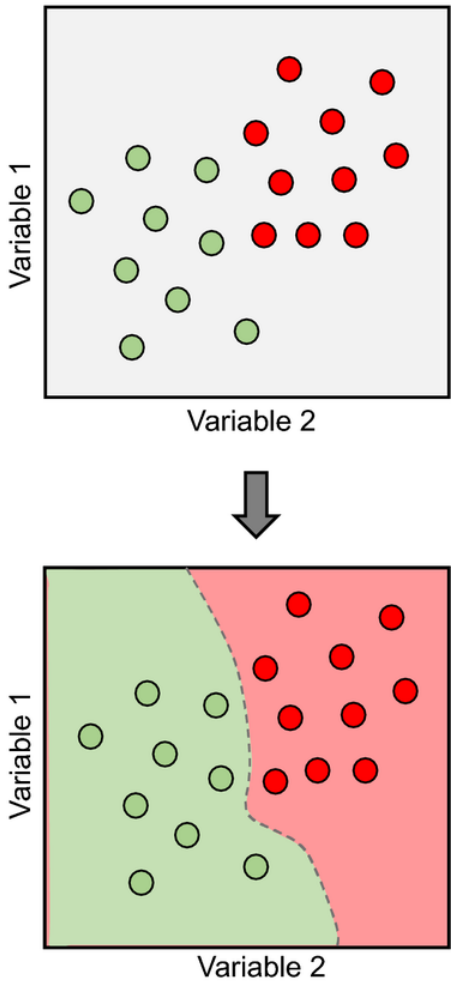


Figure: Supervised Learning

Example Supervised learning

- Plasma fusion reactor JET in Culham (UK)
- 30-second experiment generates 50 GB of data
- reconstruct the 2D plasma profile takes 1 hour per times step $\Delta t = 0.005s$
- total simulation time would be 250 days
- instead: use Convolutional Neural Network (CNN) to learn from reconstructions that have been computed at JET for all the experimental campaigns between 2011 and 2016
- multiple orders of magnitude speed up (250 days to less than 1 sec)

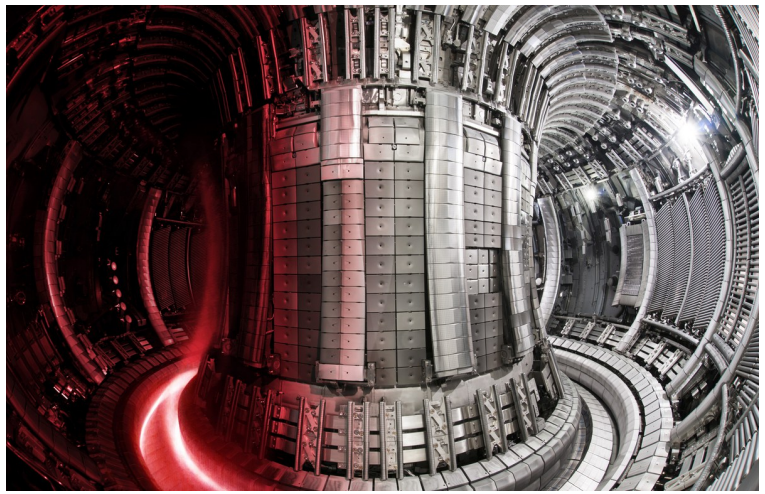


Figure: Nuclear fusion reactor JET.

Unsupervised Learning

- Learns from **unlabeled data**
- No explicit output targets
- Identifies **structures, clusters, relationships**
- **Applications:**
  - Customer segmentation
  - Anomaly detection
  - Dimensionality reduction

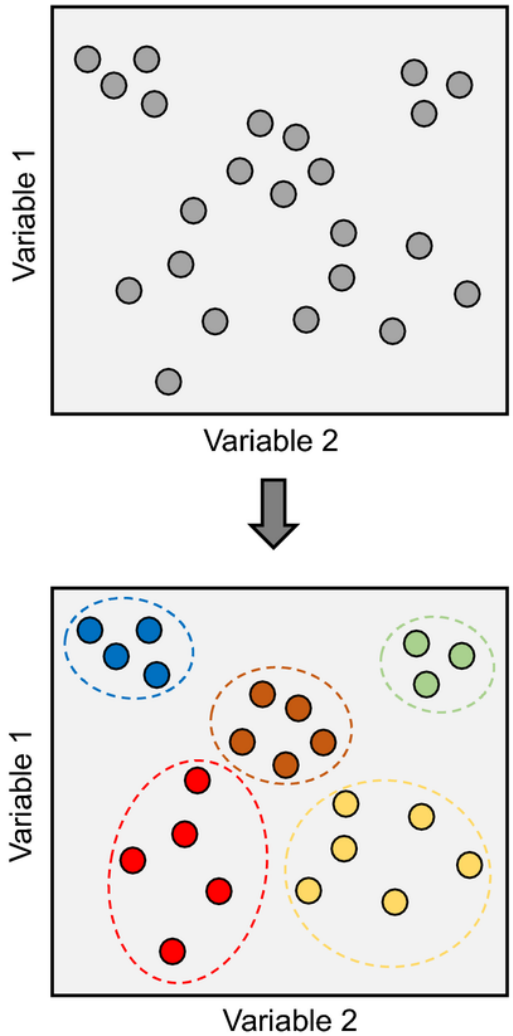


Figure: Unsupervised Learning

Example: Physics-Informed Neural Network

- No data available; only the input
- learns by a loss function that integrates the problem PDE

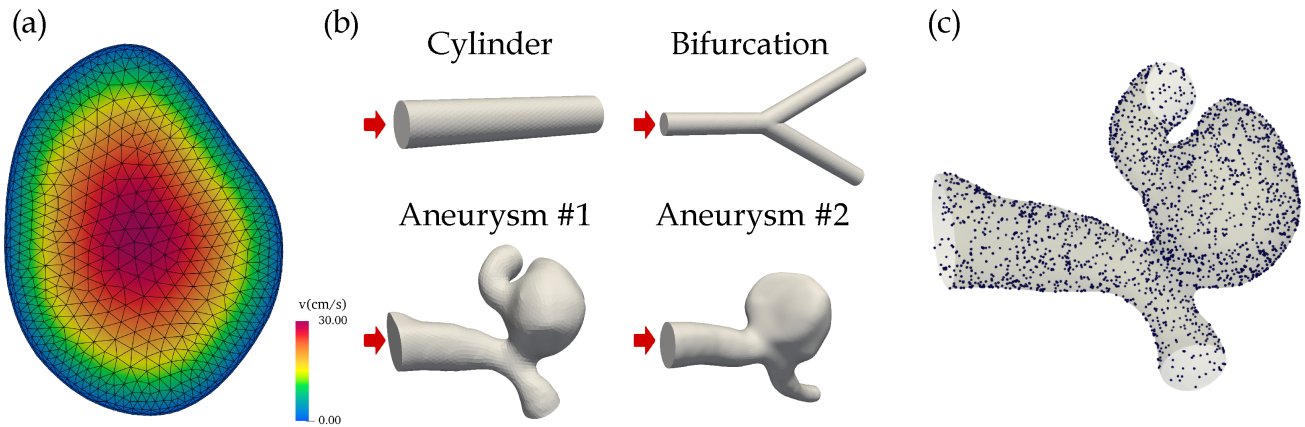
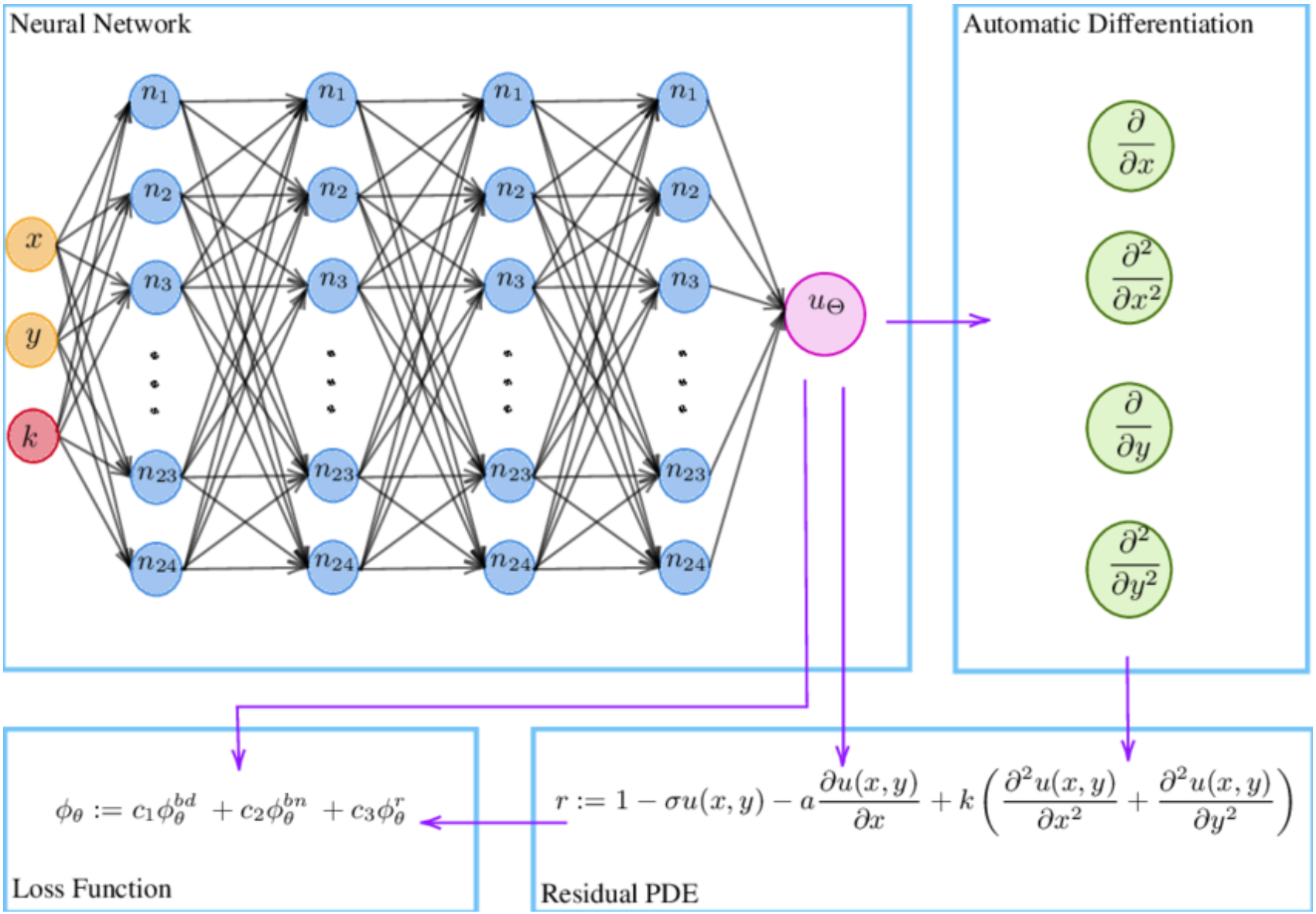


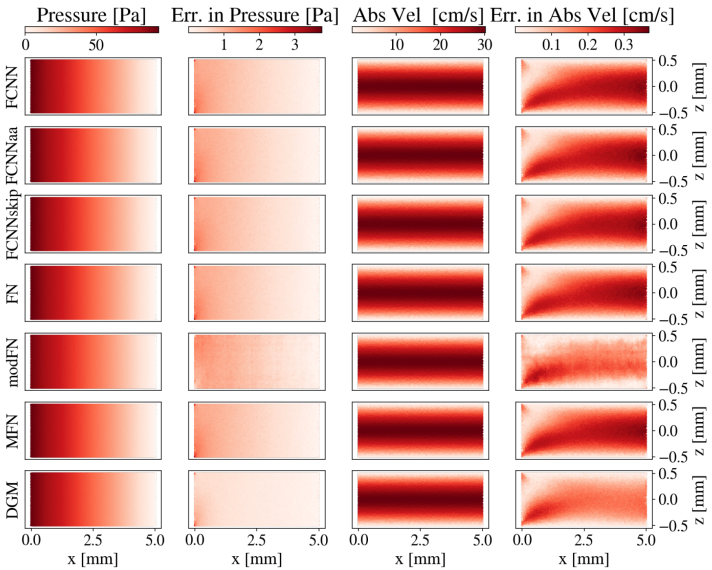
Figure: Semi-Supervised Learning

Example: Physics-Informed Neural Network

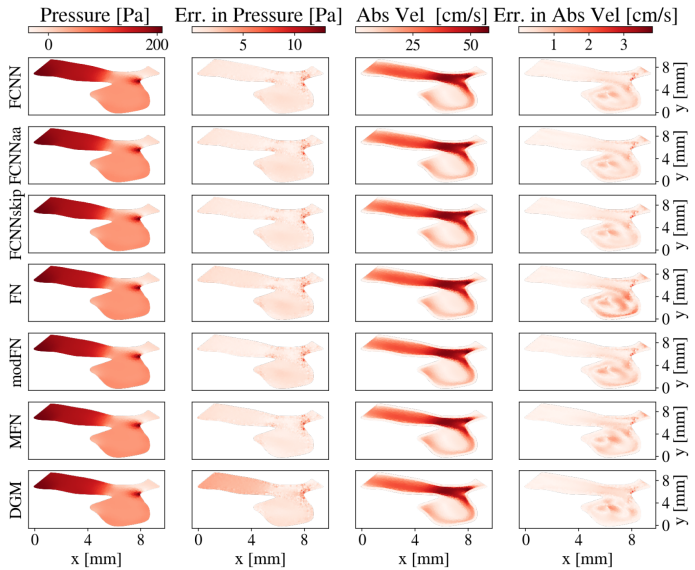


Example: Physics-Informed Neural Network

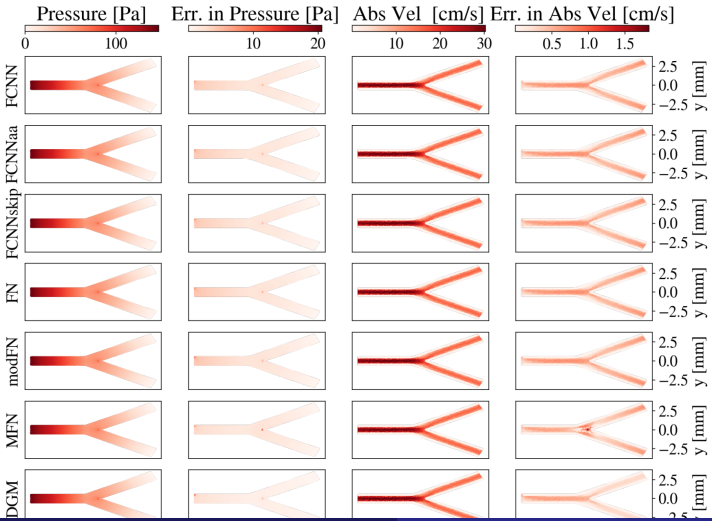
Cylinder



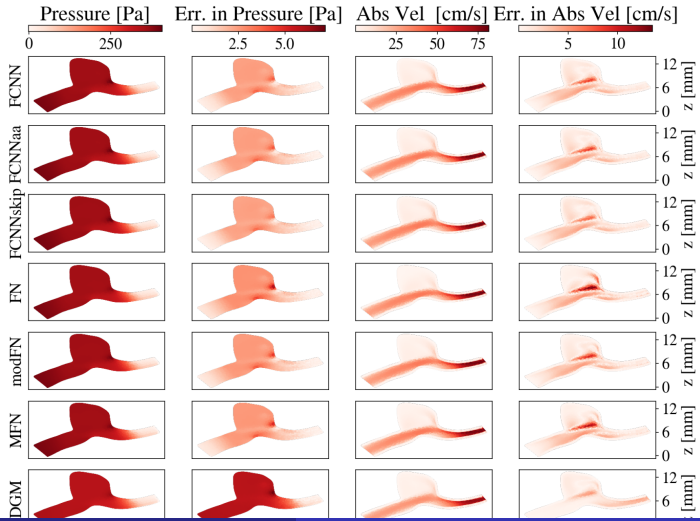
Aneurysm #1



Bifurcation



Aneurysm #2



Semi-Supervised Learning

- **Hybrid approach**
- Combines labeled and unlabeled data
- Useful when labeling is expensive
- Learns efficiently with **small amount of labeled data**
- **Applications:**
  - Speech recognition
  - Medical image analysis
  - Web page classification

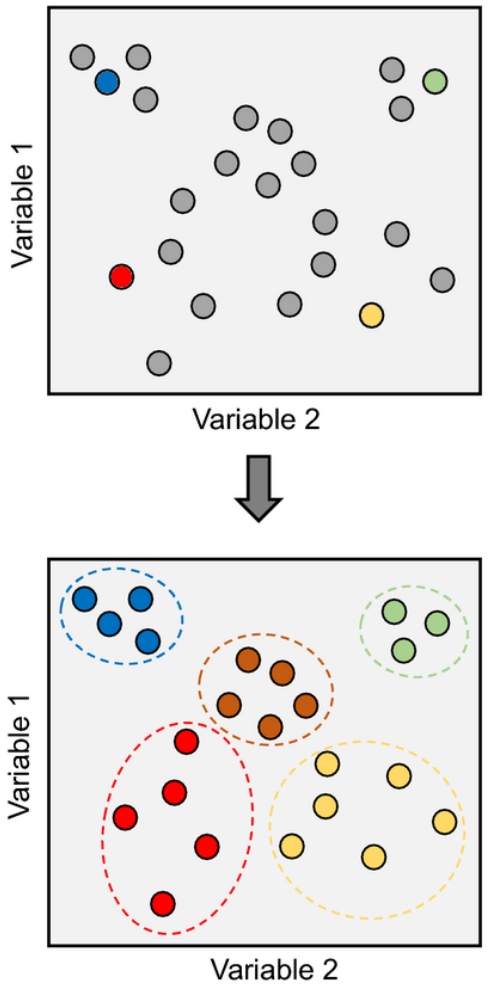


Figure: Semi-Supervised Learning

Example: Wind Farm Optimization

- Wind farm optimization takes a lot of simulations
- solution is to use a few high-resolution (HR, expensive) and many low-resolution solutions (LR, cheap)

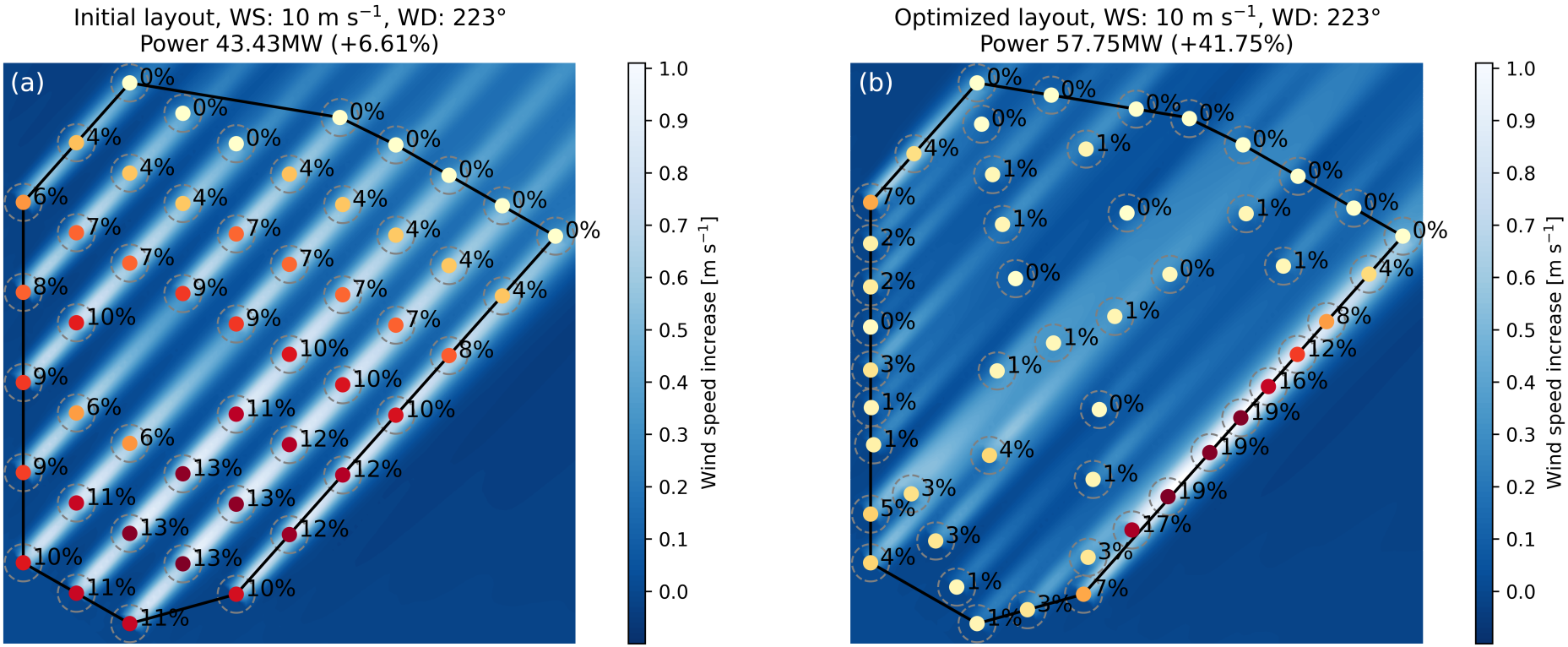


Figure: Example wind farm optimization

Example: Wind Farm Optimization

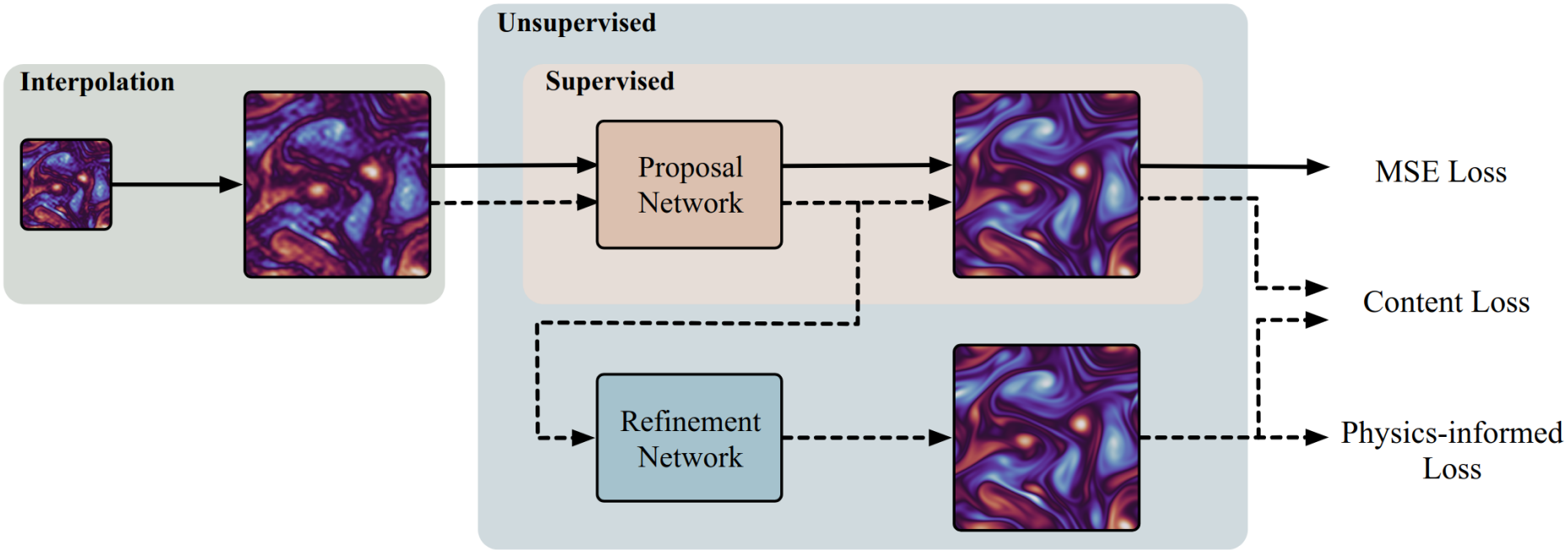


Figure: Example wind farm optimization

Reinforcement Learning

- Learning through **trial and error**
- Agent interacts with environment
- Takes actions, receives feedback
- Learns to **maximize long-term rewards**
- **Applications:**
  - Autonomous robots
  - Game AI (AlphaGo, chess)
  - Trading strategies

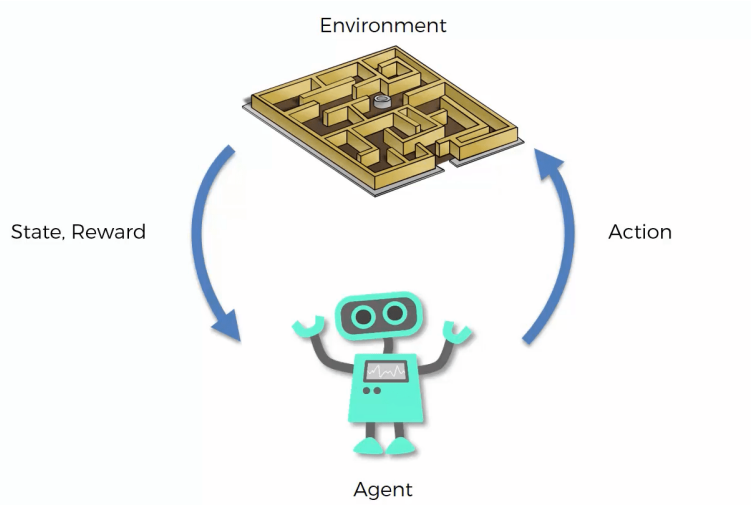
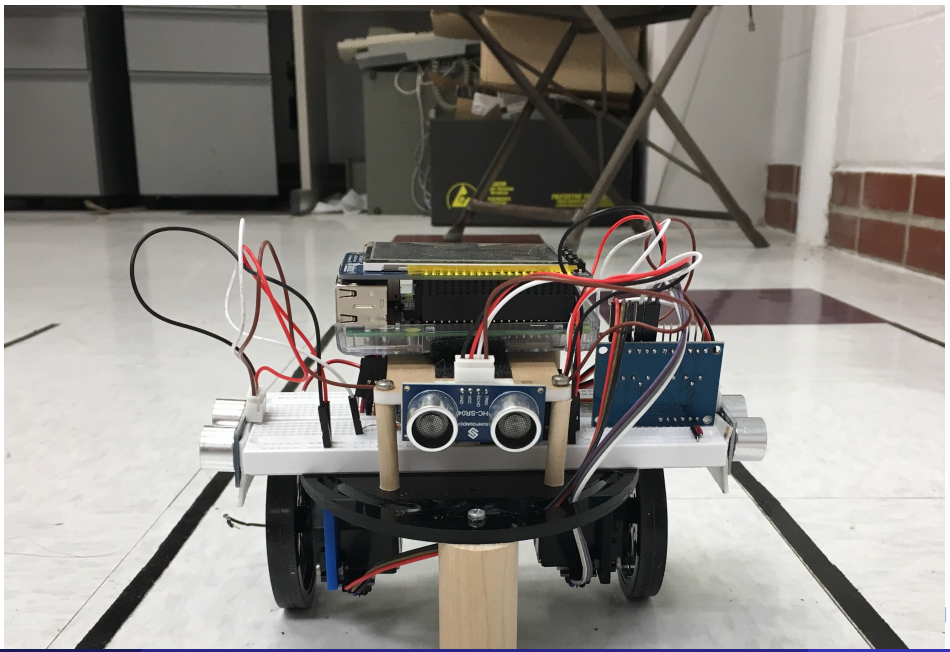


Figure: Reinforcement Learning

Frame Title

- autonomous robots (e.g., in a factory to drive parts)
- task: holding the lane from optical input by changing wheel movement
- full simulation would include solving the balance of linear momentum
- get a reward for holding the lane, get a penalty for leaving it
- training via interaction with the environment
- start training in the virtual environment
- move to real-life experiments



Types of Neural Networks

- **Feedforward Neural Networks (FNNs)**
  - Simplest type, one-directional information flow
  - Applications: Basic classification, simple regression
- **Multilayer Perceptron (MLP)**
  - At least one hidden layer, nonlinear activation
  - Applications: Digit recognition, stock prediction
- **Convolutional Neural Networks (CNNs)**
  - Image processing, spatial feature extraction
  - Applications: Image classification, medical imaging
- **Recurrent Neural Networks (RNNs)**
  - Sequential data, feedback loops
  - Applications: NLP, time series forecasting

Convolutional Neural Networks

- Specialized for **image processing**
- **Key components:**
  - Convolutional layers
  - Pooling layers
  - Fully connected layers
- Automatic feature extraction
- Spatial hierarchy learning

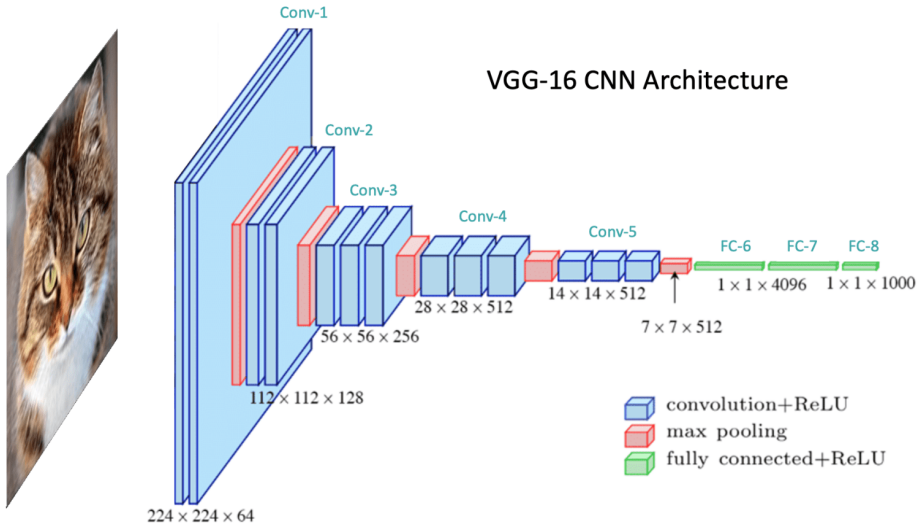


Figure: CNN Architecture

Recurrent Neural Networks

- Designed for **sequential data**
- Contains **feedback loops**
- Information persists over time
- Memory of past inputs
- Variants:
  - LSTM (Long Short-Term Memory)
  - GRU (Gated Recurrent Unit)

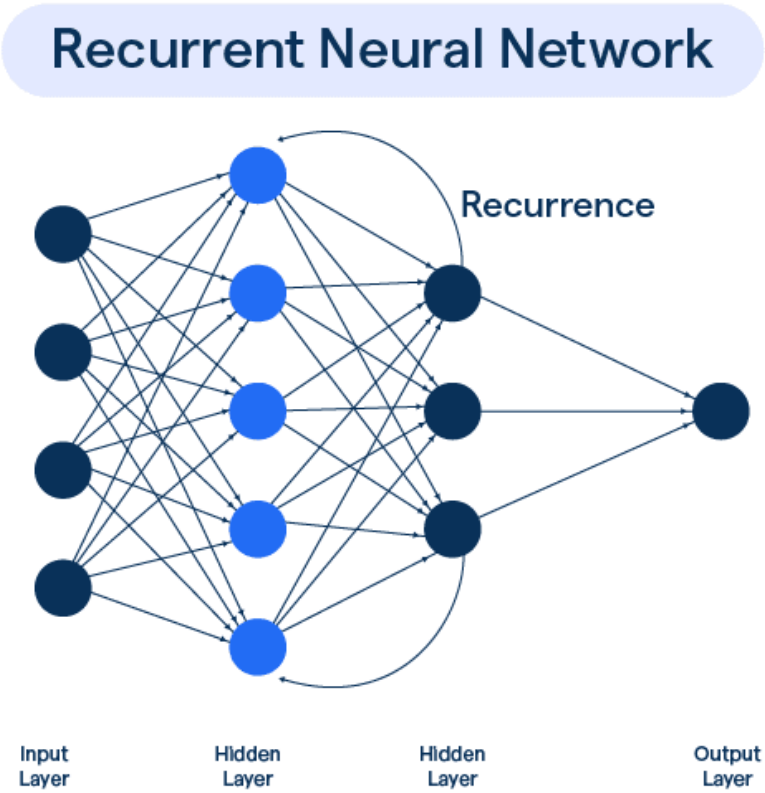


Figure: RNN Architecture

Discussion & Future Directions

- **Current challenges:**
  - Interpretability and explainability
  - Data efficiency and transfer learning
  - Energy consumption and environmental impact
- **Emerging trends:**
  - Foundation models (large pre-trained models)
  - Self-supervised learning
  - Multimodal learning
  - Neuromorphic computing
- **Research opportunities:**
  - Ethical AI and fairness
  - Robust and secure deep learning
  - Domain adaptation techniques

Summary

- **Deep learning** is a subset of machine learning enabling automatic feature learning
- **Four learning paradigms:** supervised, unsupervised, semi-supervised, reinforcement
- **Neural networks** are the foundation of deep learning
- **Key network types:** FNN, MLP, CNN, RNN
- **Activation functions** introduce non-linearity
- **Universal Approximation Theorem** explains theoretical power
- Deep learning continues to transform numerous industries

References I