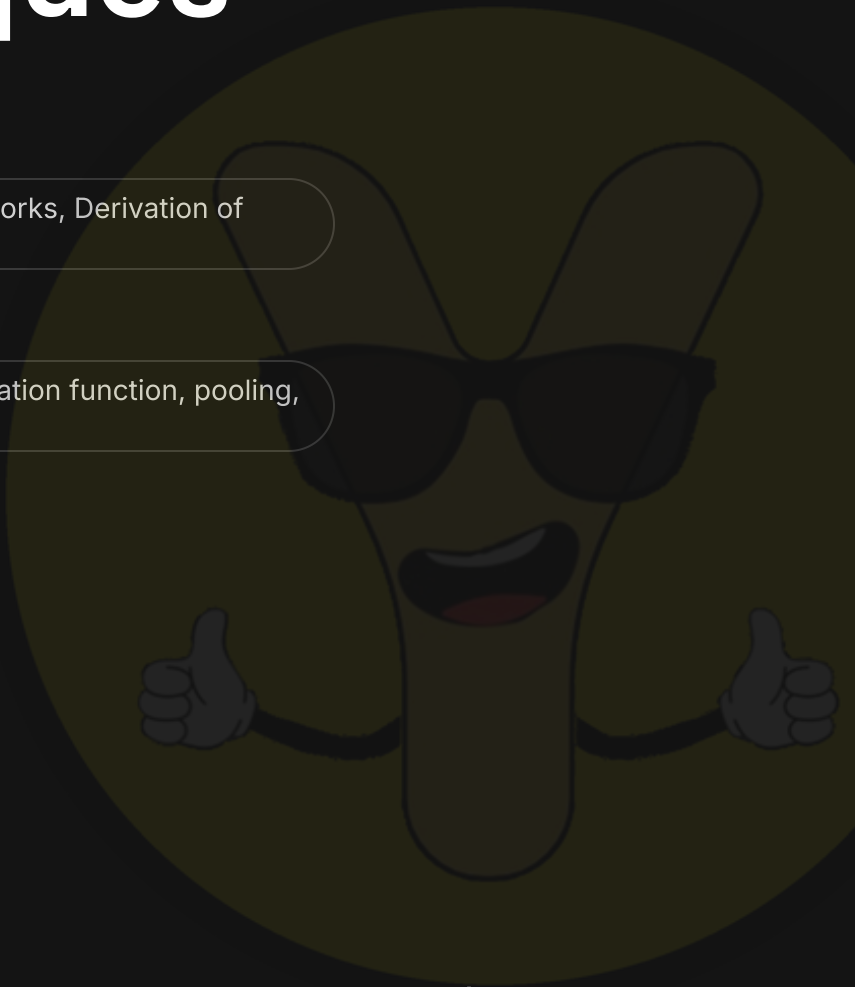
AKTU · 3RD YEAR · SEMESTER 6 · BCDS062

Machine Learning Techniques

Unit 4 — Artificial Neural Networks and Deep Learning

- Artificial Neural Networks: Perceptrons, Multilayer perceptron, Gradient descent and the Delta rule, Multilayer networks, Derivation of Backpropagation Algorithm, Generalization
- Unsupervised Learning: SOM Algorithm and its variant
- Deep Learning: Introduction, concept of convolutional neural network, Types of layers (Convolutional Layers, Activation function, pooling, fully connected), Concept of Convolution (1D and 2D) layers, Training of network



DECODED BY

Chirag Jain

Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

Artificial Neural Networks: Perceptrons

The fundamental building block of deep learning.

- Inspired by biological neurons, a perceptron takes multiple binary inputs $x_1, x_2, \ldots, x_n$ and produces a single binary output.
- Mathematical model: $y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$ where w are weights and b is bias.
- Activation function $f(z)$ is typically a step function: returns 1 if $z \geq 0$ else 0.
- Limitation: Cannot solve non-linear separable problems such as the XOR function.

Multilayer Perceptron (MLP)

Overcoming the limitations of single-layer perceptrons.

- Consists of an input layer, one or more hidden layers, and an output layer.
- Nodes in adjacent layers are fully connected with directed weighted edges.
- Non-linear activation functions (ReLU, Sigmoid, Tanh) are used in hidden layers to learn complex patterns.
- Capable of learning non-linear decision boundaries, successfully solving the XOR problem.

Gradient Descent and the Delta Rule

DELTA RULE

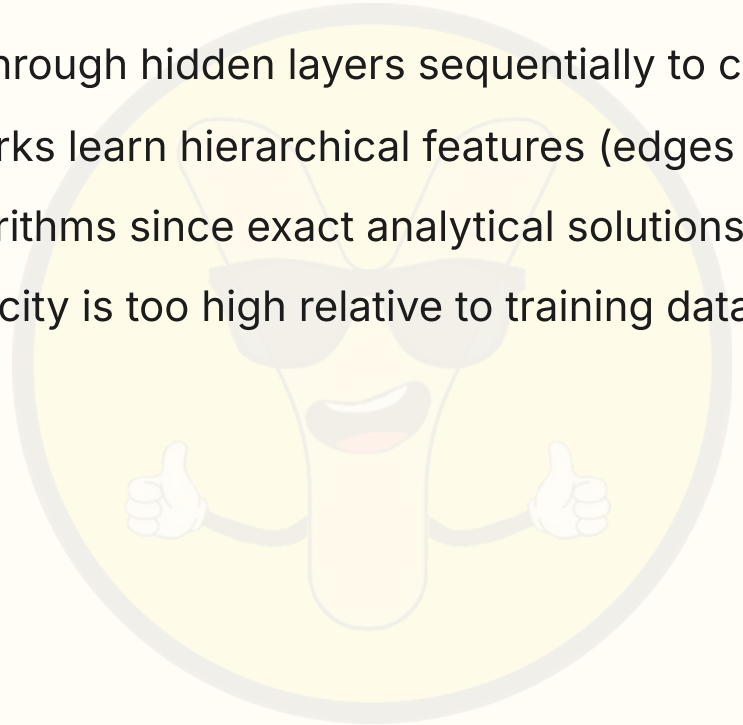
A gradient descent learning rule that updates network weights to minimize the mean squared error between desired and actual outputs.

- Weight update formula: $w_i \leftarrow w_i + \eta(t - y)x_i$ where η is learning rate and t is target.
- Gradient Descent moves parameters in the opposite direction of the gradient of the loss function.
- Ensures convergence to a local or global minimum for well-behaved error surfaces.

Multilayer Networks Architecture

Scaling up neural networks for complex representations.

- Forward propagation: Inputs pass through hidden layers sequentially to compute the final output.
- Representation power: Deep networks learn hierarchical features (edges to shapes to objects).
- Requires efficient optimization algorithms since exact analytical solutions do not exist.
- Prone to overfitting if network capacity is too high relative to training data size.



Derivation of Backpropagation Algorithm

The core mechanism for training multilayer networks using the chain rule.

- Step 1: Compute total error $E = \frac{1}{2}\sum(t_k - y_k)^2$ at the output layer.
- Step 2: Apply the chain rule to find the partial derivative of error with respect to output weights: $\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}}$.
- Step 3: Propagate the error gradients backward to hidden layers to compute local gradients.
- Step 4: Update all weights using gradient descent based on computed error sensitivities.

Generalization in Neural Networks

Ensuring models perform well on unseen data.

- Definition: The ability of a trained model to generalize properly to new, unseen test data.
- Overfitting occurs when a model learns noise in the training set, showing high training accuracy but low test accuracy.
- Underfitting occurs when a model is too simple to capture underlying data patterns.
- Techniques to improve generalization: Regularization (L1/L2), Dropout, Early Stopping, and Cross-Validation.

Unsupervised Learning: Self-Organizing Maps (SOM)

Topology-preserving dimensionality reduction technique.

- Invented by Teuvo Kohonen, SOM maps high-dimensional input space onto a low-dimensional (usually 2D) grid.
- Competitive learning: Neurons compete to be activated; the Best Matching Unit (BMU) is chosen based on Euclidean distance.
- Cooperative learning: BMU and its topological neighbors update their weights closer to the input vector.
- Preserves relative distance relationships of input data in the output map.

SOM Algorithm and Variants

Step-by-step procedure of Kohonen networks.

- Step 1: Initialize weight vectors randomly for all grid nodes.
- Step 2: Present an input vector x to the network.
- Step 3: Find BMU j by minimizing distance: $||x - w_j||$.
- Step 4: Update weights of BMU and neighbors: $w_i(t+1) = w_i(t) + \eta(t) h_{ji}(t)(x - w_i(t))$.
- Variants: Growing Neural Gas (GNG), Generative Topographic Mapping (GTM).

Deep Learning: Introduction

Transitioning from shallow networks to deep architectures.

- Deep Learning uses neural networks with many hidden layers to automatically extract features from raw data.
- Eliminates the need for manual feature engineering.
- Fueled by availability of massive datasets and high-performance GPUs.
- Applications span computer vision, natural language processing, speech recognition, and robotics.

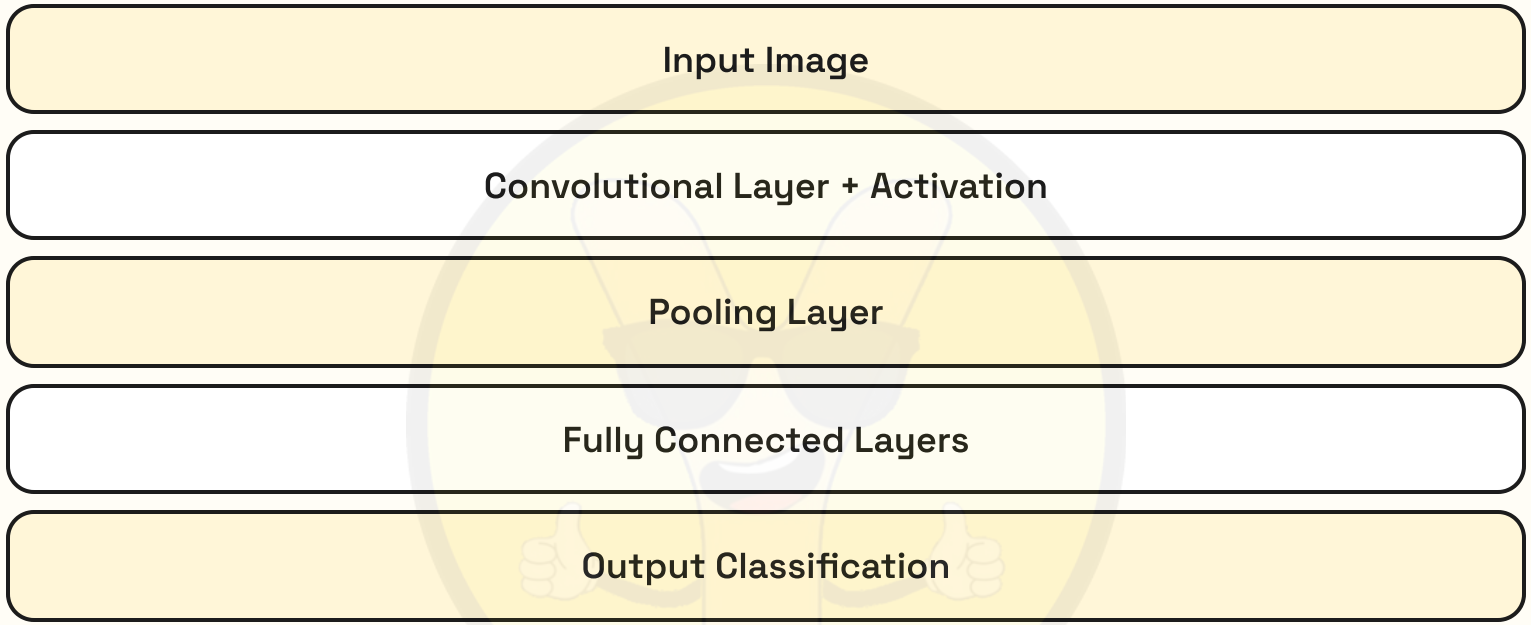
Concept of Convolutional Neural Network (CNN)

CNN

A specialized deep neural network designed for processing grid-like data, such as images, using spatial convolution filters.

- Uses local receptive fields to capture spatial correlations.
- Weight sharing drastically reduces the total number of trainable parameters compared to fully connected networks.
- Translation invariance: Features can be detected regardless of their position in the image.

CNN Architecture Layers Flow

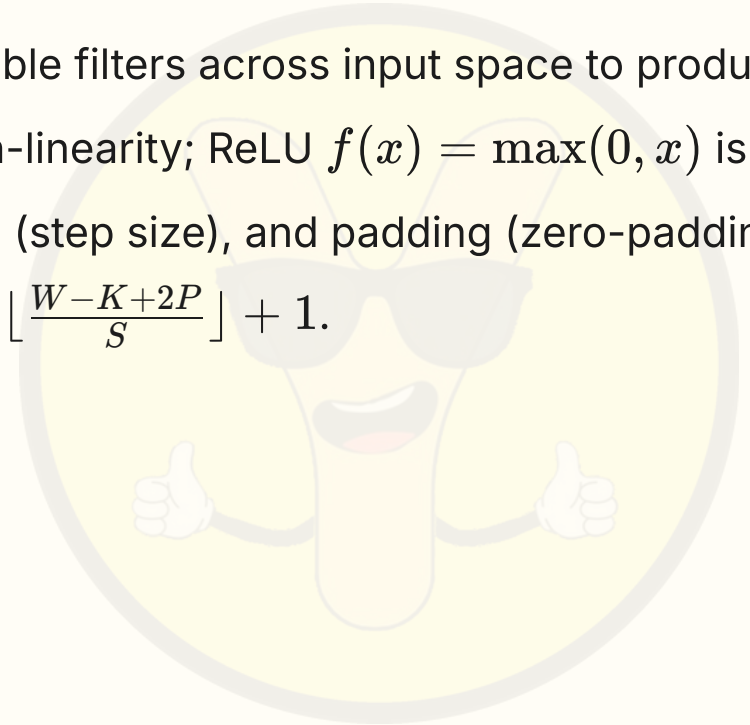


Standard sequential architecture of a Convolutional Neural Network.

Types of Layers: Convolution & Activation

Building blocks of feature extraction in CNNs.

- Convolutional Layer: Applies learnable filters across input space to produce feature maps.
- Activation Function: Introduces non-linearity; ReLU $f(x) = \max(0, x)$ is standard.
- Hyperparameters: Filter size, stride (step size), and padding (zero-padding to preserve dimensions).
- Calculated output dimension: $O = \lfloor \frac{W-K+2P}{S} \rfloor + 1$.



Types of Layers: Pooling & Fully Connected

Downsampling and classification stages.

- Pooling Layer: Reduces spatial dimensions (width and height) of feature maps to decrease computation and control overfitting.
- Max Pooling: Takes the maximum value in each filter region; Average Pooling takes the mean.
- Fully Connected (Dense) Layer: Flattens the 2D feature maps into a 1D vector to perform final class score computation.
- Softmax activation is typically applied at the output layer for multi-class classification.

Concept of Convolution (1D and 2D)

Mathematical operation underlying feature extraction.

- 1D Convolution: Used for sequential data like text or audio waveforms, sliding a 1D filter across time.
- 2D Convolution: Used for images, sliding a 2D matrix (kernel) across height and width.
- Operation involves element-wise multiplication and summation (dot product) at each patch position.
- Feature map highlights specific patterns like vertical edges, horizontal lines, or textures.

Training of CNN Networks

Optimizing network parameters end-to-end.

- Forward Pass: Pass input images through conv, pooling, and dense layers to obtain predicted probabilities.
- Loss Calculation: Compute categorical cross-entropy loss between predictions and ground-truth labels.
- Backward Pass: Compute gradients of loss with respect to all filter weights using backpropagation via chain rule.
- Optimization: Update weights using optimizers like Stochastic Gradient Descent (SGD) or Adam.

Case Study 1: Diabetic Retinopathy Detection

Medical imaging application of CNNs.

- Problem: Early detection of diabetic retinopathy from retinal fundus photographs to prevent blindness.
- CNN Role: Automatically detects microaneurysms, hemorrhages, and exudates in high-resolution eye scans.
- Architecture: Custom CNNs or transfer learning using ResNet/DenseNet pre-trained models.
- Impact: Assists ophthalmologists in screening large patient populations quickly and accurately.

Case Study 2: Building a Smart Speaker

Audio processing and speech recognition.

- Problem: Wake-word detection and automatic speech recognition (ASR) in smart home devices.
- CNN Role: Applied to spectrograms (2D time-frequency representations of audio signals) to detect specific trigger words like 'Alexa' or 'Siri'.
- Pipeline: Audio input → STFT Spectrogram → 2D CNN Feature Extraction → Dense Classifier → Intent Execution.
- Challenge: Operating in noisy environments with high background interference.

Case Study 3: Self-Driving Cars

End-to-end computer vision for autonomous navigation.

- Problem: Real-time road perception, lane detection, pedestrian tracking, and obstacle avoidance.
- CNN Role: Processes camera feeds to segment drivable space, classify traffic signs, and detect other vehicles.
- Architecture: Real-time object detection models like YOLO (You Only Look Once) or SSD integrated with depth sensors.
- Safety: Requires high reliability, low latency, and robustness against extreme weather conditions.

Exam Corner

- **Q1** Explain the architecture of a Multilayer Perceptron and discuss why single-layer perceptrons fail on XOR problems.
- **Q2** Derive the Backpropagation algorithm step-by-step for a multilayer neural network using the chain rule.
- **Q3** What is a Self-Organizing Map (SOM)? Explain its training algorithm and topological preservation property.
- **Q4** Explain the role of Convolutional, Activation, and Pooling layers in a Convolutional Neural Network.
- **Q5** Calculate the output dimension of a feature map given input size 32×32, filter size 5×5, stride 1, and padding 0.
- **Q6** Discuss the concept of generalization in neural networks and methods to prevent overfitting.
- **Q7** Explain how CNNs are applied in medical image analysis with reference to Diabetic Retinopathy detection.

Unit 4 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 5 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

