UNIT–1 : Neural Networks-I (Introduction & Architecture) (PYQ Solution)

AKTU · Application of Soft Computing (BCS056) · Solutions by **Chirag Jain**, Ycotes

SECTION A Answers (≈70 Words Each)

Q1. Why does a single-layer perceptron fail to solve linearly non-separable problems? (Jan 2026)

A single-layer perceptron computes y = f(w·x + b) and its decision boundary is a single straight line (hyperplane) w·x + b = 0. It can only separate classes that lie on opposite sides of such a line. For non-separable problems like XOR, no single line can place (0,0),(1,1) on one side and (0,1),(1,0) on the other, so the perceptron learning rule never converges. Hidden layers are needed to build non-linear boundaries.

Q2. Why do we use bias in a neural network? How does it improve the learning capability of a neuron? (Jan 2026, 2022, 2021)

Bias is an extra input fixed at 1 with its own weight b, so net = $\Sigma w_i x_i$ + b. It shifts the decision boundary away from the origin; without it the line w·x = 0 must always pass through the origin and many patterns cannot be separated. Bias also acts as an adjustable threshold of the neuron, lets it fire even when all inputs are zero, and gives the network extra freedom to fit data, so learning ability improves.

Q3. What does convergence mean in neural networks? (2023)

Convergence means that during training the error (loss) decreases and finally settles at a minimum, and the weights stop changing significantly. At that point the network gives stable, acceptable outputs for the training patterns. A network is said to converge when the error falls below a set tolerance or the

maximum number of epochs is reached. Poor learning rate or local minima can cause slow or no convergence.

Q4. Write the applications of associative memory. Differentiate between auto-associative and hetero-associative memory. (2023, 2022)

Applications: pattern recognition, noisy image / character restoration, content-addressable retrieval, speech and face recognition, database search, and error correction.

Auto-associative	Hetero-associative
Stores pattern x and recalls the same pattern x	Stores pairs (x, y) and recalls y for input x
Input and output vectors are of the same size and type	Input and output may differ in size and type
Used for noise removal, pattern completion	Used for mapping, e.g. name to phone number

---

Q5. Define soft computing. How is it different from conventional (hard) computing? (2022, 2021)

Soft computing is a collection of techniques (neural networks, fuzzy logic, genetic algorithms) that tolerate imprecision, uncertainty and approximation to give low-cost, robust solutions.

Soft computing	Hard computing
Tolerant of imprecision and uncertainty	Needs precise, exact data
Approximate, adaptive, learns from data	Fixed algorithm, no learning
Parallel computation	Sequential computation
Gives approximate solutions	Gives exact solutions
Examples: ANN, fuzzy, GA	Examples: numerical methods, binary logic

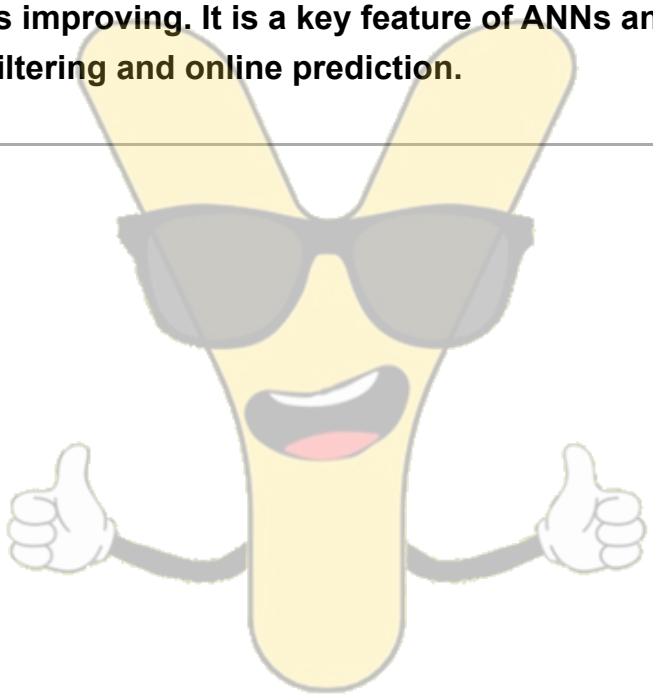
---

Q6. What is meant by a threshold logic unit (TLU)? (2022)

A threshold logic unit is the simplest artificial neuron (McCulloch–Pitts type). It computes the weighted sum of its binary inputs, $\Sigma w_i x_i$, and compares it with a threshold θ. The output is 1 if the sum $\geq \theta$, otherwise 0. By choosing weights and θ, a TLU can realise basic gates like AND and OR, but not XOR.

Q7. What is adaptive learning? (2022)

Adaptive learning is the ability of a network to adjust its weights automatically according to the data it receives, without being reprogrammed. The network changes its parameters as new inputs or a changing environment arrive, so its performance keeps improving. It is a key feature of ANNs and is used in tasks such as adaptive filtering and online prediction.



SECTION B Answers (≈250 Words Each)

Q1. Critically analyze different activation functions and justify why sigmoid or ReLU is preferred in deep neural networks. (Jan 2026)

An **activation function** adds **non-linearity** to a neuron so the network can learn complex mappings; without it, many layers collapse into one linear map.

- **Step / threshold:** output 0 or 1. Simple but not differentiable, so it cannot be used with gradient descent.
- **Linear:** f(x) = x. No non-linearity; used only in output layers for regression.
- **Sigmoid:** $f(x) = 1/(1 + e^{-x})$, range (0,1), smooth and differentiable, $f' = f(1 - f)$. Good for probabilities, but **saturates** for large |x| giving **vanishing gradients**, and is not zero-centred.
- **Tanh:** range (−1,1), zero-centred, still saturates.
- **ReLU:** f(x) = max(0, x). Cheap to compute, gradient is 1 for x > 0 so **no vanishing gradient**, gives sparse activations and faster convergence. Weakness: **dying ReLU** (neurons stuck at 0); fixed by Leaky ReLU.
- **Softmax:** converts outputs to class probabilities for multi-class output layers.

Function	Range	Main issue
Sigmoid	(0,1)	Vanishing gradient
Tanh	(−1,1)	Saturation
ReLU	[0,∞)	Dying neurons

Justification: ReLU is preferred in hidden layers of deep networks because it trains faster and avoids vanishing gradients; **sigmoid** is preferred in the output layer for binary classification because its output is a probability.

Q2. Write short notes on recurrent auto-associative memory and explain its pros and cons. (2023)

A **recurrent auto-associative memory** (e.g. the **Hopfield network**) is a feedback network in which the output of each neuron is fed back as input to all other neurons. A stored pattern is recalled from a noisy or partial input by iterating the network until it settles in a stable state.

- Weights are set by the Hebb rule: $W = \Sigma\, xx^T$ with zero diagonal.
- Update: $x_i(t+1) = \mathrm{sgn}(\Sigma_j\ w_{ij} x_j(t))$. Energy decreases at each update, so the network reaches a stable state.

Pros:

- Content-addressable: recalls the full pattern from part of it.
- Tolerates noise and missing data.
- Simple, with guaranteed convergence under asynchronous updates.

Cons:

- Limited capacity (about 0.15 × number of neurons).
- May settle in **spurious (false) stable states**.
- Similar patterns interfere with each other.
- Needs many neurons and connections (n^2).

Q3. Describe the multilayer perceptron model. Does a perceptron require supervised learning? If not, what does it require? (2021)

A **multilayer perceptron (MLP)** is a feed-forward network with an **input layer, one or more hidden layers and an output layer**. Each neuron applies a non-linear activation to a weighted sum. Because of hidden layers it can solve non-linear problems such as XOR. It is trained using **back-propagation**.



Multilayer perceptron (feed-forward)

Supervised learning? Yes, perceptron learning is **supervised**: it needs a training set of inputs with **known target outputs**. The error (target − actual) drives the weight update $\Delta w = \eta(t - y)x$. If no targets are available, a perceptron cannot learn; unsupervised problems need other models such as Kohonen maps. So the perceptron requires **labelled data (input–target pairs)**, linearly separable classes and a learning rate.

SECTION C Answers (≈500 Words Each)

Q1. Analyze the biological and artificial neuron in terms of signal processing capability. What is a synapse? Draw and explain the structure of a neuron. Discuss the limitations of artificial neuron models. (Jan 2026, 2023)

A **biological neuron** has: **dendrites** (receive signals), **soma/cell body** (sums signals), **axon** (carries the output pulse) and **synapses** (junctions between axon terminal of one neuron and dendrite of the next). A **synapse** is the gap where chemical neurotransmitters pass the signal; its strength decides how much input is transmitted, and changes in strength are the basis of learning.



Signal path in a biological neuron

An **artificial neuron** models this: inputs x_i (dendrites), weights w_i (synaptic strengths), summation $\Sigma w_i x_i + b$ (soma) and activation function f (firing), giving $y = f(\Sigma w_i x_i + b)$.

Aspect	Biological	Artificial
Speed	Slow (milliseconds)	Very fast (nanoseconds)
Processing	Massively parallel, ~10^{11} neurons	Few neurons, mostly serial simulation
Signal	Analog pulses, chemical	Numerical values
Fault tolerance	High, cells die daily	Low
Learning	Continuous, adaptive	Needs training phase
Energy	Very low	High

Limitations of artificial neurons:

- Single neuron solves only linearly separable problems.
- Very simplified model; ignores timing, pulse frequency and chemical effects.
- Needs large labelled data and long training.
- Training can get stuck in local minima; results are hard to explain (black box).
- Overfitting, and sensitive to choice of learning rate and architecture.

Conclusion: the artificial neuron borrows the idea of weighted signal summation from the biological one but is far less capable in adaptability and efficiency.

Q2. Explain various learning techniques in neural networks and analyze their suitability for supervised and unsupervised problems. Explain supervised and unsupervised learning in detail. Differentiate between supervised and unsupervised learning. (Jan 2026, 2022, 2021)

Learning is the process of adjusting weights so that the network produces the desired behaviour.

(a) Supervised learning: training data contains inputs with **target outputs**. The error between target and actual output is used to correct weights. Examples: perceptron rule, delta rule, back-propagation. Used for **classification and regression** (spam detection, price prediction).

(b) Unsupervised learning: only inputs are given; the network finds **hidden structure, clusters or features** by itself. Examples: Hebbian learning, competitive learning, Kohonen SOM. Used for clustering, dimensionality reduction, feature extraction.

(c) Reinforcement learning: the network gets only a reward or penalty signal after an action and learns a policy (game playing, robotics).

(d) Other rules: Hebbian (neurons that fire together strengthen), competitive (winner-take-all), error-correction (delta) and stochastic (Boltzmann).

Basis	Supervised	Unsupervised
Data	Labelled	Unlabelled
Teacher	Required	Not required
Goal	Predict output	Discover pattern
Feedback	Error signal	None
Examples	BPN, perceptron	SOM, clustering
Accuracy	Measurable	Hard to measure

Suitability: supervised methods suit problems with known answers and enough labelled data; unsupervised methods suit exploratory problems where labels are costly or unavailable.

Q3. What is an artificial neural network? Explain its layers and basic components. (2023, 2022)

An **artificial neural network (ANN)** is a computing model inspired by the brain, made of many interconnected processing units (**neurons**) that learn from examples by adjusting connection **weights**.

Basic components:

- **Processing elements (neurons)** performing the computation.
- **Weights** that store knowledge.
- **Summation unit** computing net = $\Sigma w_i x_i$.
- **Bias** shifting the threshold.
- **Activation function** producing output.
- **Learning rule** updating weights.

Layers:

- **Input layer:** receives raw data; one node per feature; no computation.
- **Hidden layer(s):** extract features through weighted sums and non-linear activations.
- **Output layer:** gives the final result (class or value).



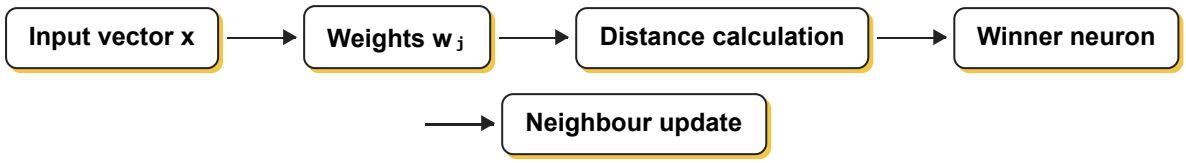
Layers of an ANN

ANNs are used for recognition, prediction and control, and learn without explicit programming.

Q4. Explain the architecture of the Kohonen self-organizing network (SOM). (2022)

A **self-organizing map** is an **unsupervised, competitive** network that maps high-dimensional input onto a low-dimensional (usually 2-D) grid while keeping similar inputs close together.

Architecture: an **input layer** of n nodes fully connected to an **output (Kohonen) layer** of neurons arranged in a grid. Each output neuron j has a weight vector w_j of the same size as the input.



Working of Kohonen SOM

Algorithm:

1. Initialise weights randomly; set learning rate α and neighbourhood radius R.
2. For each input x, compute distance $d_j = \Sigma(x_i - w_{ij})^2$ for every output neuron.
3. The neuron with minimum distance is the **winner (best matching unit)**.
4. Update the winner and its neighbours: $w_j(new) = w_j(old) + \alpha(x - w_j(old))$.
5. Reduce α and R gradually; repeat until weights stabilise.

Uses: clustering, data visualisation, speech and image classification. No target outputs are required.

Q5. Apply the Hebb rule to develop the logical AND function (bipolar inputs and targets). (2021)

Hebb rule: $\Delta w_i = x_i \cdot t$, $\Delta b = t$, starting with $w_1 = w_2 = b = 0$.

x_1	x_2	b	t	Δw_1	Δw_2	Δb	w_1	w_2	b
1	1	1	1	1	1	1	1	1	1
1	−1	1	−1	−1	1	−1	0	2	0
−1	1	1	−1	1	−1	−1	1	1	−1
−1	−1	1	−1	1	1	−1	2	2	−2

Final weights: $\mathbf{w_1 = 2, w_2 = 2, b = -2}$.

Check with net = $2x_1 + 2x_2 - 2$ and f = +1 if net > 0 else −1:

- (1,1): 2 → +1 ✓
- (1,−1): −2 → −1 ✓
- (−1,1): −2 → −1 ✓
- (−1,−1): −6 → −1 ✓

So the Hebb network correctly implements bipolar AND.

Q6. Differentiate between a recurrent network and a multilayer feed-forward network. (2021)

A **multilayer feed-forward network** passes signals in one direction only, input → hidden → output, with no loops. A **recurrent network** has **feedback connections**, so outputs are fed back as inputs and the network has memory of earlier states.

Basis	Multilayer feed-forward	Recurrent
Signal flow	One direction	Feedback loops present
Memory	None; output depends on current input	Has memory of previous states
Dynamics	Static mapping	Dynamic; settles to stable state
Training	Back-propagation	Backpropagation through time, Hebb
Stability	Always stable	Stability must be ensured
Examples	MLP, BPN	Hopfield, Elman network
Applications	Classification, pattern recognition	Sequences, time series, associative memory

Conclusion: feed-forward networks are simpler and stable; recurrent networks are more powerful for temporal and associative tasks but harder to train.
