CS601 Machine Learning
Unit 2: Neural Networks
RGPV Exam-Oriented Notes

Linearity, Activation Functions, Gradient Descent, Backpropagation, Autoencoders, Batch Normalization, Dropout, Regularization and Hyperparameter Tuning

How to use these notes:

- Read the One-Minute Revision Box before starting each topic.
- For 7 marks: write definition, diagram, working, advantages and applications.
- For 14 marks: add comparison table, common mistakes, PYQ keywords and conclusion.
- Most repeated topics: Gradient Descent, Backpropagation, Activation Functions, Autoencoders and Regularization.

Unit 2 Official Syllabus

Linearity vs non-linearity, activation functions like sigmoid and ReLU, weights and bias, loss function, gradient descent, multilayer network, backpropagation, weight initialization, training and testing, unstable gradient problem, autoencoders, batch normalization, dropout, L1 and L2 regularization, momentum and hyperparameter tuning.

Topic	Priority	Why Important
Activation Functions	Very High	Used in every neural network; Sigmoid and ReLU are common PYQ topics.
Gradient Descent	Very High	Main optimization algorithm for reducing loss.
Backpropagation	Very High	Most important training algorithm for neural networks.
Autoencoders	High	Architecture-based long-answer question.
Batch Normalization and Dropout	High	Frequently asked as optimization/regularization techniques.
L1/L2 Regularization	Very High	Comparison-based scoring question.
Hyperparameter Tuning	High	Expected short/long answer topic.

1. Linearity vs Non-Linearity

Why This Topic Exists

Imagine tum marks prediction kar rahe ho. Agar study hours badhne par marks almost same proportion me badhte hain, relation linear hai. Agar kabhi marks slow increase hote hain aur kabhi suddenly high increase hote hain, relation non-linear hai. Real-world data mostly non-linear hota hai, isliye neural networks me non-linearity ka role bahut important hai.

Beginner Explanation

Linearity means input and output ke beech straight-line relationship. Non-linearity means input aur output ka relationship complex, curved ya irregular hota hai. Neural networks non-linear activation functions use karke complex patterns learn karte hain.

7 Marks Definition

Linearity is a mathematical property where output changes proportionally with input and can be represented by a straight line. Non-linearity represents complex relationships where output does not change proportionally with input. Neural networks require non-linear activation functions to learn real-world complex patterns.

Detailed Explanation - Exam Points

- Linear relation simple and proportional hota hai.
- Non-linear relation complex hota hai aur real data me common hota hai.
- Deep learning non-linearity ki wajah se powerful hoti hai.

Visual Learning Diagram

```
Linear: Input -> Straight-line relation -> Output
Non-linear: Input -> Curved/complex relation -> Output
```

Examiner Keywords

- Linearity
- Non-Linearity
- Straight line
- Complex relationship
- Activation function
- Neural network

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Differentiate linearity and non-linearity.
- Why is non-linearity required in neural networks?

One-Minute Revision:

- Linear relation simple and proportional hota hai.
- Linearity is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

2. Activation Functions

Why This Topic Exists

Agar neural network me activation function na ho, to multiple layers hone ke baad bhi network mostly linear behavior karega. Complex decisions jaise face recognition, spam detection, image classification possible nahi honge. Activation function neuron ka decision switch hai.

Beginner Explanation

Activation function neuron ke output par apply hone wali mathematical function hai. Ye decide karti hai ki neuron activate hoga ya nahi. Ye non-linearity introduce karti hai, jisse neural network complex data learn kar pata hai.

7 Marks Definition

An activation function is a mathematical function applied to the output of a neuron to introduce non-linearity into neural networks. It allows neural networks to model complex patterns and decide whether a neuron should be activated.

Detailed Explanation - Exam Points

- Non-linearity introduce karti hai.
- Neuron output control karti hai.
- Deep learning models ko complex data learn karne me help karti hai.
- Common examples: Sigmoid, ReLU, Tanh, Softmax.

Visual Learning Diagram

```
Input -> Weighted Sum -> Activation Function -> Neuron Output
```

Examiner Keywords

- Activation
- Non-linearity
- Neuron
- Sigmoid
- ReLU
- Output

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain activation functions and their need.
- Explain different activation functions used in neural networks.

One-Minute Revision:

- Non-linearity introduce karti hai.
- Activation is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

3. Sigmoid Function

Why This Topic Exists

Binary decision problems me model ko probability output chahiye hota hai. Example: email spam hone ki probability 0.92 hai. Sigmoid function kisi bhi input ko 0 aur 1 ke range me convert karta hai, isliye probability-based output ke liye useful hai.

Beginner Explanation

Sigmoid ek S-shaped activation function hai jo input value ko 0 se 1 ke beech output me convert karta hai. Ye binary classification aur logistic regression me common hai.

7 Marks Definition

Sigmoid function is an activation function that maps any real-valued input into a value between 0 and 1. It is useful for probability-based outputs but may suffer from vanishing gradient problem in deep networks.

Detailed Explanation - Exam Points

- Formula: sigma(x)=1/(1+e^-x).
- Output range: 0 to 1.
- x=0 par output 0.5 hota hai.
- Deep networks me vanishing gradient problem create kar sakta hai.

Visual Learning Diagram

```
Input -large -> Output close to 0
Input 0 -> Output 0.5
Input +large -> Output close to 1
```

Examiner Keywords

- S-shaped curve
- Probability
- 0 to 1
- Binary classification
- Vanishing gradient

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain sigmoid activation function.
- Write advantages and disadvantages of sigmoid function.

One-Minute Revision:

- Formula: sigma(x)=1/(1+e^-x).
- S-shaped curve is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

4. ReLU Function

Why This Topic Exists

Sigmoid slow training aur vanishing gradient issue create karta hai. Deep neural networks ke liye faster aur simpler activation chahiye thi. ReLU ne is problem ko largely solve kiya aur modern deep learning me standard activation ban gaya.

Beginner Explanation

ReLU ka full form Rectified Linear Unit hai. Ye negative input par 0 output deta hai aur positive input ko same pass kar deta hai.

7 Marks Definition

ReLU is a widely used activation function in neural networks defined as f(x)=max(0,x). It outputs zero for negative inputs and the same positive value for positive inputs, enabling faster training in deep networks.

Detailed Explanation - Exam Points

- Formula: f(x)=max(0,x).
- Negative input -> 0.
- Positive input -> same value.
- Fast computation aur less vanishing gradient issue.
- Dying ReLU problem possible hai.

Visual Learning Diagram

```
x < 0 -> 0
x > 0 -> x
```

Examiner Keywords

- Rectified Linear Unit
- Fast training
- Dying ReLU
- CNN
- Deep learning

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain ReLU with diagram.
- Differentiate sigmoid and ReLU.

One-Minute Revision:

- Formula: f(x)=max(0,x).
- Rectified Linear Unit is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

5. Weights and Bias

Why This Topic Exists

Model ko pata hona chahiye ki kaunsi input feature zyada important hai. Example: marks prediction me study hours ka weight Instagram time se zyada hona chahiye. Weights feature importance control karte hain aur bias output ko adjust karta hai.

Beginner Explanation

Weight input ki importance batata hai. Bias ek extra adjustment value hai jo model ko flexible banata hai. Neural network ka basic calculation hota hai: output = weight x input + bias.

7 Marks Definition

Weights and bias are learnable parameters of a neural network. Weights determine the importance of inputs, while bias shifts the activation function and helps the model fit data more accurately.

Detailed Explanation - Exam Points

- Weight = importance of input.
- Bias = adjustment or shift.
- Formula: z = wx + b.
- Training ke dauran weights and bias update hote hain.

Visual Learning Diagram

```
Input -> Multiply by Weight -> Add Bias -> Activation -> Output
```

Examiner Keywords

- Weight
- Bias
- Parameter
- Feature importance
- Learning

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain weights and bias in neural networks.
- Why is bias required in a neuron?

One-Minute Revision:

- Weight = importance of input.
- Weight is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

6. Loss Function

Why This Topic Exists

Model prediction karta hai, par prediction galat bhi ho sakti hai. Galti kitni hui? Isko measure karne ke liye loss function use hota hai. Agar loss high hai to model poor hai; agar loss low hai to model better hai.

Beginner Explanation

Loss function actual output aur predicted output ke beech error measure karta hai. Gradient descent isi loss ko minimize karne ki koshish karta hai.

7 Marks Definition

A loss function is a mathematical function that measures the difference between actual output and predicted output. It guides optimization algorithms such as gradient descent to improve model performance.

Detailed Explanation - Exam Points

- Regression me Mean Squared Error common hai.
- Classification me Cross-Entropy Loss common hai.
- Loss kam karna model training ka main goal hota hai.

Visual Learning Diagram

```
Actual Output + Predicted Output -> Compare -> Loss/Error -> Weight Update
```

Examiner Keywords

- Error measurement
- Actual value
- Predicted value
- MSE
- Cross entropy
- Optimization

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain loss function.
- Explain role of loss function in neural network training.

One-Minute Revision:

- Regression me Mean Squared Error common hai.
- Error measurement is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

7. Gradient Descent

Why This Topic Exists

Machine learning model ka aim loss ko minimum karna hai. Gradient descent ek method hai jo slope ke opposite direction me weights ko update karta hai, jaise hill se neeche utarna.

Beginner Explanation

Gradient descent optimization algorithm hai jo loss function ko minimize karta hai by updating model parameters step by step.

7 Marks Definition

Gradient Descent is a first-order optimization algorithm used to minimize a loss function by iteratively updating weights and biases in the direction opposite to the gradient.

Detailed Explanation - Exam Points

- Formula: new weight = old weight - learning rate x gradient.
- Learning rate step size control karta hai.
- Too high learning rate unstable training de sakta hai.
- Too low learning rate slow training deta hai.

Visual Learning Diagram

```
Initialize weights -> Calculate prediction -> Calculate loss -> Calculate gradient ->
Update weights -> Repeat
```

Examiner Keywords

- Optimization
- Gradient
- Learning rate
- Loss minimization
- Weight update

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain gradient descent with working.
- Explain types of gradient descent.

One-Minute Revision:

- Formula: new weight = old weight - learning rate x gradient.
- Optimization is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

8. Types of Gradient Descent

Why This Topic Exists

Large datasets me poora data har update me use karna slow hota hai. Isliye different versions use hote hain: Batch, Stochastic and Mini-Batch.

Beginner Explanation

Gradient descent ke types data usage ke basis par divide hote hain. Batch full dataset use karta hai, SGD ek sample use karta hai, Mini-Batch small group use karta hai.

7 Marks Definition

Types of gradient descent include Batch Gradient Descent, Stochastic Gradient Descent and Mini-Batch Gradient Descent. They differ in the amount of data used for each weight update.

Detailed Explanation - Exam Points

- Batch GD: complete dataset, stable but slow.
- SGD: one sample, fast but noisy.
- Mini-Batch GD: small batch, fast and stable; most commonly used.

Visual Learning Diagram

```
Complete data -> Batch GD
One sample -> SGD
Small batch -> Mini-Batch GD
```

Examiner Keywords

- Batch
- Stochastic
- Mini-batch
- Update
- Convergence

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Compare Batch GD, SGD and Mini-Batch GD.
- Which gradient descent is most commonly used and why?

One-Minute Revision:

- Batch GD: complete dataset, stable but slow.
- Batch is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

9. Multilayer Neural Network

Why This Topic Exists

Single-layer network simple problems solve karta hai. Complex patterns jaise image recognition aur NLP ke liye multiple hidden layers chahiye. Multilayer network deep learning ki foundation hai.

Beginner Explanation

Multilayer neural network me input layer, one or more hidden layers aur output layer hoti hai. Hidden layers data ke complex features learn karti hain.

7 Marks Definition

A multilayer neural network is a neural network architecture consisting of an input layer, one or more hidden layers and an output layer. It can learn complex non-linear relationships using weights, biases and activation functions.

Detailed Explanation - Exam Points

- Input layer data receive karti hai.
- Hidden layers learning perform karti hain.
- Output layer final result deti hai.
- More hidden layers generally increase learning capacity but also computation.

Visual Learning Diagram

```
Input Layer -> Hidden Layer 1 -> Hidden Layer 2 -> Output Layer
```

Examiner Keywords

- Input layer
- Hidden layer
- Output layer
- Deep learning
- Non-linearity

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain multilayer neural network with diagram.
- Explain role of hidden layers.

One-Minute Revision:

- Input layer data receive karti hai.
- Input layer is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

10. Backpropagation

Why This Topic Exists

Neural network prediction galat kare to weights kaise improve honge? Backpropagation output error ko backward direction me propagate karta hai aur weights update karta hai.

Beginner Explanation

Backpropagation neural network training algorithm hai jo error ko output layer se input side ki taraf pass karke weights update karta hai.

7 Marks Definition

Backpropagation is a supervised learning algorithm used to train neural networks by calculating the error at the output layer and propagating it backward to update weights using gradients.

Detailed Explanation - Exam Points

- Forward pass prediction generate karta hai.
- Loss calculate hota hai.
- Backward pass error ko propagate karta hai.
- Gradient descent weights update karta hai.
- Repeat until loss reduces.

Visual Learning Diagram

```
Forward: Input -> Hidden -> Output -> Loss
Backward: Loss -> Output -> Hidden -> Weight Update
```

Examiner Keywords

- Forward pass
- Backward pass
- Error propagation
- Gradient
- Weight update

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain backpropagation algorithm with diagram.
- Why is backpropagation important in neural networks?

One-Minute Revision:

- Forward pass prediction generate karta hai.
- Forward pass is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

11. Weight Initialization

Why This Topic Exists

Training start hone se pehle weights assign karne padte hain. Agar sab weights zero hon, neurons same learn karenge. Agar weights bahut bade ya chhote hon, gradients unstable ho sakte hain.

Beginner Explanation

Weight initialization is the process of assigning initial values to neural network weights before training begins.

7 Marks Definition

Weight initialization is a technique used to set initial weights of a neural network before training. Proper initialization improves convergence speed, avoids symmetry problem and reduces unstable gradient issues.

Detailed Explanation - Exam Points

- Random initialization common hai.
- Xavier initialization sigmoid/tanh ke liye useful hai.
- He initialization ReLU ke liye useful hai.
- Poor initialization slow or unstable training kar sakti hai.

Visual Learning Diagram

```
Training Start -> Initialize Weights -> Forward Pass -> Learning Begins
```

Examiner Keywords

- Random initialization
- Xavier
- He initialization
- Symmetry breaking
- Convergence

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain weight initialization and its importance.
- What happens if all weights are initialized to zero?

One-Minute Revision:

- Random initialization common hai.
- Random initialization is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

12. Training and Testing

Why This Topic Exists

Student padhai karta hai aur exam deta hai. Same way model training data se learn karta hai aur testing data par evaluate hota hai. Testing batata hai ki model real-world data par kaise perform karega.

Beginner Explanation

Training is model learning phase. Testing is model evaluation phase on unseen data.

7 Marks Definition

Training is the process of adjusting model parameters using training data, while testing evaluates the trained model on unseen data to measure generalization and performance.

Detailed Explanation - Exam Points

- Common split: 80% training, 20% testing.
- Training parameters update karta hai.
- Testing parameters update nahi karta.
- Testing overfitting detect karne me help karta hai.

Visual Learning Diagram

```
Dataset -> Training Data -> Model Learning
Dataset -> Testing Data -> Performance Evaluation
```

Examiner Keywords

- Training data
- Testing data
- Evaluation
- Generalization
- Accuracy

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Differentiate training and testing.
- Why is testing required in machine learning?

One-Minute Revision:

- Common split: 80% training, 20% testing.
- Training data is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

13. Unstable Gradient Problem

Why This Topic Exists

Deep networks me gradients backpropagation ke dauran too small ya too large ho sakte hain. Isse learning slow, unstable ya fail ho sakti hai. Ye unstable gradient problem hai.

Beginner Explanation

Unstable gradient problem occurs when gradients become extremely small or extremely large during training.

7 Marks Definition

Unstable gradient problem is a training issue in deep neural networks where gradients either vanish or explode during backpropagation, causing slow learning, unstable updates or training failure.

Detailed Explanation - Exam Points

- Two forms: vanishing gradient and exploding gradient.
- Causes: deep networks, poor initialization, sigmoid/tanh, high learning rate.
- Solutions: ReLU, batch normalization, gradient clipping, proper initialization.

Visual Learning Diagram

```
Backpropagation -> Gradient too small or too large -> Unstable learning
```

Examiner Keywords

- Gradient
- Vanishing
- Exploding
- Backpropagation
- Deep network

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain unstable gradient problem.
- Explain solutions to unstable gradient problem.

One-Minute Revision:

- Two forms: vanishing gradient and exploding gradient.
- Gradient is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

14. Vanishing Gradient Problem

Why This Topic Exists

Gradient jab repeatedly small values se multiply hota hai to earlier layers tak pahunchte-pahunchte almost zero ho jata hai. Isse earlier layers learn nahi kar pati.

Beginner Explanation

Vanishing gradient problem occurs when gradients become very small during backpropagation and learning becomes extremely slow.

7 Marks Definition

Vanishing gradient problem is a deep learning issue where gradients shrink as they propagate backward through layers, making earlier layers learn very slowly or stop learning.

Detailed Explanation - Exam Points

- Common with sigmoid/tanh in deep networks.
- Earlier layers slow learn karti hain.
- Solutions: ReLU, batch normalization, residual connections, proper initialization.

Visual Learning Diagram

```
Gradient: 0.8 -> 0.4 -> 0.1 -> 0.01 -> 0.0001 -> Learning stops
```

Examiner Keywords

- Vanishing gradient
- Small gradient
- Sigmoid
- ReLU
- Slow learning

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain vanishing gradient problem with causes and solutions.
- How does ReLU reduce vanishing gradient problem?

One-Minute Revision:

- Common with sigmoid/tanh in deep networks.
- Vanishing gradient is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

15. Exploding Gradient Problem

Why This Topic Exists

Gradient too large ho jaye to weight updates extremely large ho jate hain. Training unstable ho sakti hai aur numerical overflow bhi ho sakta hai.

Beginner Explanation

Exploding gradient problem occurs when gradients become extremely large during backpropagation.

7 Marks Definition

Exploding gradient problem is a training issue in deep neural networks where gradients grow excessively large during backpropagation, causing unstable updates and poor convergence.

Detailed Explanation - Exam Points

- Common in deep networks/RNNs.
- Causes: high learning rate, poor initialization.
- Solutions: gradient clipping, lower learning rate, batch normalization, proper initialization.

Visual Learning Diagram

```
Gradient: 1 -> 10 -> 100 -> 1000 -> Training unstable
```

Examiner Keywords

- Exploding gradient
- Gradient clipping
- Large updates
- Instability
- Learning rate

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain exploding gradient problem.
- Differentiate vanishing and exploding gradient problems.

One-Minute Revision:

- Common in deep networks/RNNs.
- Exploding gradient is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

16. Autoencoders

Why This Topic Exists

Data compression aur feature extraction ke liye autoencoders use hote hain. Example: image ko compressed representation me convert karke phir reconstruct karna.

Beginner Explanation

Autoencoder is an unsupervised neural network that learns compressed representation of input data and reconstructs the original data.

7 Marks Definition

An autoencoder is a neural network architecture consisting of an encoder, bottleneck and decoder. It learns efficient representations of data by reconstructing input with minimum reconstruction error.

Detailed Explanation - Exam Points

- Encoder input ko compress karta hai.
- Bottleneck important features store karta hai.
- Decoder original input reconstruct karta hai.
- Applications: compression, denoising, anomaly detection, dimensionality reduction.

Visual Learning Diagram

```
Input -> Encoder -> Bottleneck -> Decoder -> Reconstructed Output
```

Examiner Keywords

- Encoder
- Decoder
- Bottleneck
- Reconstruction error
- Unsupervised learning

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain autoencoder with architecture.
- Write applications of autoencoders.

One-Minute Revision:

- Encoder input ko compress karta hai.
- Encoder is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

17. Batch Normalization

Why This Topic Exists

Deep networks me layer inputs ki distribution training ke dauran change hoti rehti hai. Batch normalization inputs ko normalize karke training faster and stable banata hai.

Beginner Explanation

Batch normalization normalizes layer inputs within a mini-batch during training.

7 Marks Definition

Batch Normalization is a deep learning technique that normalizes mini-batch activations using mean and variance to reduce internal covariate shift, speed up convergence and improve training stability.

Detailed Explanation - Exam Points

- Mean aur variance calculate hota hai.
- Values normalize hoti hain.
- Scale and shift parameters apply hote hain.
- Training faster and stable hoti hai.

Visual Learning Diagram

```
Input -> Mean/Variance -> Normalize -> Scale/Shift -> Output
```

Examiner Keywords

- Mini-batch
- Mean
- Variance
- Normalization
- Internal covariate shift

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain batch normalization.
- Write advantages of batch normalization.

One-Minute Revision:

- Mean aur variance calculate hota hai.
- Mini-batch is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

18. Dropout

Why This Topic Exists

Large neural networks training data memorize kar sakte hain, jise overfitting kehte hain. Dropout randomly neurons disable karke model ko robust features learn karne par force karta hai.

Beginner Explanation

Dropout is a regularization technique that randomly disables neurons during training to prevent overfitting.

7 Marks Definition

Dropout is a neural network regularization method in which randomly selected neurons are temporarily deactivated during training, reducing overfitting and improving generalization.

Detailed Explanation - Exam Points

- Training ke time random neurons off hote hain.
- Testing ke time full network use hota hai.
- Overfitting reduce hota hai.
- Generalization improve hota hai.

Visual Learning Diagram

```
Without Dropout: O O O O
With Dropout: O X O X (X = disabled neuron)
```

Examiner Keywords

- Dropout
- Regularization
- Overfitting
- Generalization
- Random neurons

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain dropout technique.
- How does dropout prevent overfitting?

One-Minute Revision:

- Training ke time random neurons off hote hain.
- Dropout is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

19. L1 and L2 Regularization

Why This Topic Exists

Overfitting reduce karne ke liye model ko large weights ke liye penalty di jati hai. L1 unnecessary features remove kar sakta hai, L2 weights ko small rakhta hai.

Beginner Explanation

L1 and L2 regularization are techniques used to prevent overfitting by adding penalty terms to the loss function.

7 Marks Definition

L1 regularization adds absolute values of weights to the loss function and can create sparse models. L2 regularization adds squared weights to the loss function and reduces weight magnitude without usually eliminating features.

Detailed Explanation - Exam Points

- L1 formula: Loss = Original Loss + lambda sum |w|.
- L2 formula: Loss = Original Loss + lambda sum w^2.
- L1 feature selection me help karta hai.
- L2 stable models banata hai.

Visual Learning Diagram

```
Large Weights -> Penalty -> Smaller Weights -> Less Overfitting
```

Examiner Keywords

- L1
- L2
- Penalty
- Sparse model
- Feature selection
- Weight decay

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Differentiate L1 and L2 regularization.
- Explain regularization and its need.

One-Minute Revision:

- L1 formula: Loss = Original Loss + lambda sum |w|.
- L1 is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

20. Momentum

Why This Topic Exists

Gradient descent kabhi zig-zag movement karta hai, jis se convergence slow hoti hai. Momentum previous updates ka direction use karke learning ko smoother and faster banata hai.

Beginner Explanation

Momentum is an optimization technique that accelerates gradient descent by using previous update information.

7 Marks Definition

Momentum is a gradient descent improvement technique that uses past gradients or updates to reduce oscillations and speed up convergence toward the minimum loss.

Detailed Explanation - Exam Points

- Zig-zag movement reduce karta hai.
- Faster convergence deta hai.
- Optimization smoother hoti hai.
- Extra momentum parameter require karta hai.

Visual Learning Diagram

```
Without Momentum: /\/\/\/
With Momentum: Smooth path to minimum
```

Examiner Keywords

- Momentum
- Convergence
- Oscillation
- Optimization
- Gradient descent

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain momentum in gradient descent.
- Why is momentum used in optimization?

One-Minute Revision:

- Zig-zag movement reduce karta hai.
- Momentum is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

21. Hyperparameter Tuning

Why This Topic Exists

Model ki kuch settings training se pehle choose karni padti hain, jaise learning rate, batch size, epochs. Wrong settings poor accuracy de sakti hain. Best settings find karna hyperparameter tuning hai.

Beginner Explanation

Hyperparameter tuning is the process of selecting the best hyperparameter values for a machine learning model.

7 Marks Definition

Hyperparameter Tuning is the systematic process of optimizing model settings such as learning rate, batch size, epochs, number of layers and dropout rate to achieve better accuracy and generalization.

Detailed Explanation - Exam Points

- Common hyperparameters: learning rate, batch size, epochs, layers, dropout rate.
- Methods: Grid Search, Random Search, Bayesian Optimization.
- Improves accuracy but time-consuming hota hai.

Visual Learning Diagram

```
Choose Parameters -> Train Model -> Evaluate -> Adjust -> Best Model
```

Examiner Keywords

- Hyperparameter
- Learning rate
- Batch size
- Epochs
- Grid search
- Random search

Common Mistakes

- Definition likhkar diagram skip karna.
- Keyword use na karna.
- Real-life example na dena.
- Comparison table miss karna jab question differentiate ho.

Important Questions

- Explain hyperparameter tuning.
- Write methods of hyperparameter tuning.

One-Minute Revision:

- Common hyperparameters: learning rate, batch size, epochs, layers, dropout rate.
- Hyperparameter is a must-use keyword.
- Use diagram in 7-mark/14-mark answer.

Unit 2 PYQ Trend Analysis

Topic	Frequency	Importance	Expected Pattern
Backpropagation	Repeated	★★★★★	Explain with diagram
Gradient Descent	Repeated	★★★★★	Algorithm + types
Activation Functions	Repeated	★★★★★	Sigmoid vs ReLU
Autoencoders	Often asked	★★★★	Architecture + applications
Dropout	Often asked	★★★★	Overfitting prevention
Batch Normalization	Recently asked	★★★★	Working + advantages
L1/L2 Regularization	Repeated	★★★★★	Difference table
Hyperparameter Tuning	Expected	★★★★	Methods + examples

Top 20 Important Questions for CS601 ML Unit 2

1. Explain activation functions and their need in neural networks.
2. Explain sigmoid and ReLU activation functions with advantages and disadvantages.
3. Differentiate sigmoid and ReLU.
4. Explain weights and bias in neural networks.
5. Explain loss function and its role in training.
6. Explain gradient descent algorithm with formula.
7. Compare Batch, Stochastic and Mini-Batch Gradient Descent.
8. Explain multilayer neural network with diagram.
9. Explain backpropagation algorithm with neat diagram.
10. Explain weight initialization and its importance.
11. Differentiate training and testing.
12. Explain unstable gradient problem.
13. Explain vanishing gradient problem with causes and solutions.
14. Explain exploding gradient problem with causes and solutions.
15. Explain autoencoder with architecture and applications.
16. Explain batch normalization with working.
17. Explain dropout and how it prevents overfitting.
18. Differentiate L1 and L2 regularization.
19. Explain momentum in gradient descent.
20. Explain hyperparameter tuning and its methods.

Most Expected Questions 2026

- 95% - Explain Backpropagation Algorithm with neat diagram.
- 95% - Explain Gradient Descent and its types.
- 90% - Differentiate L1 and L2 Regularization.
- 90% - Explain Autoencoder with architecture.
- 90% - Explain Batch Normalization and Dropout.
- 85% - Explain Sigmoid and ReLU activation functions.

24-Hour Revision Sheet

```
Activation Function -> Adds non-linearity
Sigmoid -> Output 0 to 1
```

```
ReLU -> Fast activation for deep networks
Weights -> Input importance
Bias -> Output adjustment
Loss Function -> Error measurement
Gradient Descent -> Minimizes loss
Backpropagation -> Updates weights
Vanishing Gradient -> Learning stops
Exploding Gradient -> Training unstable
Autoencoder -> Compression + feature extraction
Batch Normalization -> Stable training
Dropout -> Prevents overfitting
L1 -> Feature selection
L2 -> Weight reduction
Momentum -> Faster convergence
Hyperparameter Tuning -> Best model settings
```

RGPV Answer Writing Formula

For 7 marks:

- Definition
- Neat diagram
- Working steps
- Advantages/applications
- Short conclusion

For 14 marks:

- Introduction
- Definition
- Diagram
- Working in steps
- Comparison table if applicable
- Advantages
- Disadvantages
- Applications
- Conclusion with examiner keywords

Final Quality Check

A weak student studying 1-2 days before the exam can use these notes to write strong 7-mark and 14-mark answers because every topic includes definition, diagram, working, keywords, common mistakes and expected questions.