

Deep Learning Assignment 1

Name: Sivaharini k

Degree: B.Tech AIML

**Problem Description:**

For a naturally labeled (LFW) dataset, the hyperparameters in the neural network are preset configurations such as learning rate, batch size, and number of epochs that affect the training process and the entire model. . These differ from model parameters, which include weights and biases learned from data at runtime. Recover error models to minimize operational losses and improve classification accuracy. Hyperparameters and models do not play a significant role in training neural networks to correctly classify faces into corresponding labels.

**Dataset Link:** <https://www.kaggle.com/datasets/stoicstatic/face-recognition-dataset>

**Key Terminologies and Parameters:**

**1. Neural Network**In the context of the LFW dataset above, neural networks are used to process facial images and divide them into groups representing different variables. Through training, the network learns to recognize patterns in faces.

**2. Neurons**Neurons in the neural network processing the dataset receive pixel values ​​from the facial image, transform them using an activation function, and pass the output to the next layer of neurons to analyze specific facial features.

**3. Layer**

Layers in the neural network for the Sign Language dataset include an input layer that receives the pixel values, multiple hidden layers that extract and process features from the images, and an output layer that provides the classification of the hand gestures into one of the 24 classes.

* **Input Layer**

The input layer of the LFW dataset consists of neurons that take the pixel values ​​of the face image and pass it to the hidden layer for further processing.

* **Hidden Layer**

Layers in neural networks analyze facial images using functions to detect facial features such as edges, shapes, and textures and perform transformations on input pixel values.

* **Output Layer**

The output layer of the LFW dataset neural network consists of neurons representing the identity in the data. The output layer provides the final classification based on the features learned from the latent method.

**4. Convolutional layers**It uses convolution operations on the input image and filters to detect local features such as layers, edges, and texture in the LFW dataset's neural network. These features are very important for identifying faces in images.

**5. Convolutional Neural Networks (CNN)**Convolutional Neural Networks (CNN) are good for processing LFW data. CNNs use layers to extract and restore the spatial hierarchy of features from input images, allowing them to recognize facial features.

**6. Recurrent Neural Network (RNN)**

Recurrent neural networks are generally used with linear data, not image data. They are not very relevant to this information, which focuses on still images.

**7. Activation Functions**The processing function in neural networks provides non-linear modeling capabilities, allowing it to learn complex facial models. Activation functionality includes ReLU, Sigmoid, and Tanh.

* **ReLU**The ReLU (Rectified Linear Unit) function is often used in the hidden layer of neural networks in the LFW dataset. It helps the model converge faster by allowing positive values to pass while setting negative values to zero, thus helping the model learn complex features.
* **Sigmoid**Sigmoid activation function is used to define values 0 and 1 in neural networks.
* **Tanh**The tanh (hyperbolic tangent) function maps input values to the range -1 to 1.
* **Softmax**The softmax function is often used in the output layer of neural networks for LFW datasets. It converts the value of the raw material into the result, summing the results between 0 and 1 for each group (individual) and giving 1.

8. **Forward Propagation**Forward propagation is the process of passing input pixel values from a face image into the layers of a neural network using weights and biases to produce the final predicted output for each person.

9. **Backpropagation**Backpropagation is a training technique used in neural networks for the LFW dataset, adjusting the weights of the network according to the prediction error. This iterative process helps the network learn the right features to classify faces correctly.

10. **Loss Function**The loss function measures the difference between the predicted and actual labels of the face image. Optimization should be done during training to reduce classification errors.

11. **Cost Function**The cost function evaluates the overall performance of a neural network by calculating the loss of each training example. Reducing the cost can help networks increase the accuracy of face classification.

12**. Gradient descent**Gradient descent is an optimization technique used to reduce the cost of neural networks on the LFW dataset. To improve accuracy, it readjusts the weights of the network in the direction that reduces the operating cost the most.

13. **Learning Rate**The learning rate is a hyperparameter that controls the step size during gradient descent. It determines how quickly the network adjusts its weights based on the calculated gradients.

14. **Batch size**Batch size is the number of face images processed in a training iteration. It affects the safety and speed of training.

15. **Epoch**

A schedule is the completion of all LFW training materials during the course. More time is used to ensure that the neural network has enough time to learn from the data and improve classification accuracy.

16. **Overfitting**A schedule is the completion of all LFW training materials during the course. More time is used to ensure that the neural network has enough time to recover when it learns to distinguish well-defined images but performs poorly on new, unseen images. This indicates that the model remembers the training data rather than generalizing it.

17. **Underfitting**Underfitting occurs when the neural network in the MNIST dataset is unable to learn patterns in the training data, resulting in poor performance in both training and validation. This shows that the model is too simple and cannot capture important features of the data.

18. **Training Set**The training set of the LFW dataset consists of images used to train the neural network. The model learns by adjusting the weights based on this information.

19. **Validation Set**The validation process of the LFW dataset involves a series of images used to test the performance of the neural network during training. It helps in monitoring the model's performance and tuning hyperparameters to avoid overfitting.

20. **Test Set**The testing method of the LFW dataset involves classifying images to test the performance of neural network training. It provides an unbiased assessment of the model's ability to generalize to new, unseen data.

21. **Cross-validation**Cross-validation is a technique that can be used with the LFW dataset to test the performance of neural networks. It involves splitting data into subsets and training models on different subsets to ensure robustness and performance.

22. **Hyperparameters**

Hyperparameters are settings in the neural network for the LFW dataset that are not learned from the data but are set before training. These include learning rate, batch size, and number of epochs, all of which affect the training process and performance of the model.

23. **Model Parameters**

Model parameters in the neural network of the LFW dataset include weights and biases learned from the training data. These parameters are adjusted during training to minimize performance loss and improve classification accuracy.

24. **Regularization**

In the neural network of the LFW dataset, regularization methods are used to prevent overfitting by adding penalties for large weights. The process includes L1 and L2 normalization, which helps improve the model's performance on new data.

25. **Dropout**

Dropout is a regularization technique used during neural network training on the LFW dataset. By randomly setting a subset of neurons to zero in each training iteration, it prevents the network from over-relying on specific neurons and improves its ability to generalize.

26. **Weight Initialization**

Initial weights refer to the process of setting the initial weights of the neural network for the LFW dataset before training begins. Properly initialized weights ensure faster and better performance, helping to prevent issues such as vanishing or exploding gradients.

27. **Normalization**

In the context of the LFW dataset, normalization involves scaling the pixel values of an image to a standard range, such as 0-1. This technique helps the neural network learn more efficiently by making the input data consistent.

28. **Standardization**

Normalization will change the pixel values of the LFW image so that the mean is zero and the standard deviation is one. This preliminary step enables data centralization and scaling, thus improving the training process and the performance of the neural network.