

Deep Learning Assignment 1

Name: Vaitheeshwari J

Degree: B.Tech AIML

**Problem Description:**

For the Sign Language dataset, 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 overall model. These differ from model parameters, which include weights and biases learned from data during the run. Iteratively correct model errors to minimize operational losses and improve classification accuracy. Hyperparameters and models do not play a significant role in training the neural network to correctly classify hand movements into corresponding labels.

Dataset Link: <https://www.kaggle.com/datasets/ash2703/handsignimages>

**Key Terminologies and Parameters:**

**1. Neural Network**In the context of the above Sign Language dataset, a neural network is used to process and segment a 28x28 pixel grayscale image into 24 different groups representing A to Z (excluding J and Z). Through training, the network learns to recognize patterns in hand movements.

**2. Neurons**The neurons in the neural network that process the dataset take pixel values ​​from the 28x28 image, transform them using the activation function, and pass the output to the Next layer of neurons to identify each neuron of interest. Specific patterns of alphabetical signatures.

**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 for the Sign Language dataset consists of 784 neurons (one for each pixel in the 28x28 images) that take in the grayscale values (ranging from 0-255) and pass them to the hidden layers for further processing.

* **Hidden Layer**

Hidden layers in the neural network analyze the Sign Language images and perform transformations on the input pixel values, using activation functions to detect complex features such as edges, shapes, and textures specific to different hand gestures.

* **Output Layer**

The output layer of the neural network for the Sign Language dataset has 24 neurons, each representing one of the sign language letters (excluding J and Z). The output layer provides the final classification based on the learned features from the hidden layers.

**4. Convolutional layers**Convolutional layers in neural networks from the Sign Language dataset use convolution operations for input images and filters to detect local features such as edges and texture. These features are important for recognizing gestures in images.

**5. Convolutional Neural Networks (CNN)**Convolutional Neural Networks (CNN) are good for processing the signature dataset. CNN uses convolutional layers to extract and adapt the spatial hierarchy of features from the input images, allowing it to effectively group images such as gestures.

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

Recurrent Neural Network works better than a simple neural network when data is sequential like Time-Series data and text data. RNNs are typically used for sequential data, not image data. They are less relevant for this dataset, which is focused on static images.

**7. Activation Functions**Sign Language MNIST Dataset Activation functions in neural networks give the model a non-linear feature, allowing it to learn complex patterns in movements. Activation functions include ReLU, Sigmoid, and Tanh.

* **ReLU**The ReLU (Rectified Linear Unit) function is often used in the hidden process of neural networks for the Sign language dataset. It helps the model converge faster by allowing positive values ​​to pass while setting negative values ​​to zero, thus helping the model to learn complex features.
* **Sigmoid**The sigmoid activation function, although rare in modern architectures, can be used in neural networks to map output values ​​between 0 and 1 for signatures in the MNIST dataset. Can be used in various settings.
* **Tanh**The Tanh (hyperbolic tangent) function maps input values ​​to the range from -1 to 1. This is useful in neural networks for checking MNIST dataset signatures to ensure the data is between zeros. Make training faster, and more stable.
* **Softmax**The Softmax function is often used in the output layer of neural networks for the MNIST dataset. It converts the value of the raw material into the result, giving each group (label) a result between 0 and 1, with a sum of 1.

**8. Forward Propagation**Forward propagation is the process of transferring input pixel values ​​from a signature in the MNIST image to a neural network layer using weights and enhancements to make the final output prediction for each leg.

**9. Backpropagation**Backpropagation is a training technique used in neural networks for the Sign Language MNIST dataset; where it adjusts the weight of the network according to the prediction error. This iterative process helps the network learn the right features to classify gestures correctly.

**10. Loss Function**The loss function in the context of the Sign Language MNIST dataset measures the difference between the predicted and actual label of a motion image. To reduce the classification error, optimization must be done during training.

**11. Cost Function**The cost function provides a measure of the overall performance of the neural network by summing the losses across all descriptive datasets. Reducing the processing cost helps the network improve the accuracy of motion classification.

**12. Gradient descent**Gradient descent is an optimization technique used to reduce the cost of neural networks on the MNIST dataset. It iteratively adjusts the weight of the network according to the direction of the steepest decline in the cost function to improve the ratio.

**13. Learning Rate**The learning rate is a hyperparameter that controls the step size during the gradient descent of the neural network for the MNIST dataset. It determines how quickly the network adjusts its weights in response to calculated gradients.

**14. Batch size**Batch size is the number of language MNIST images processed in one instruction iteration. It affects the stability and speed of training, small batch sizes can provide concurrent updates while large batch sizes can provide stable gradient estimates.

**15. Epoch**

An epoch is one complete pass through the entire Sign Language MNIST training dataset during the training process. Multiple epochs are used to ensure the neural network has sufficient opportunity to learn from the data and improve its classification accuracy.

**16. Overfitting**Overfitting occurs when a neural network on the MNIST dataset learns to classify training images well 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 Sign Language MNIST dataset contains some of the images used to train the neural network. The model learns the direction by adjusting the weights based on this information.

**19. Validation Set**Set of validation instructions The MNIST dataset is a set of images used to test the performance of the neural network during training. It helps in tracking the correct model and tuning hyperparameters to avoid overfitting.

**20. Test Set**Notes The test set of the MNIST dataset is a separate set of images used to test the performance of neural network training. It provides an unbiased assessment of the model's ability to extend new, unseen information.

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

**22. Hyperparameters**

Hyperparameters are settings in the neural network for the Sign Language MNIST dataset that are not learned from the data but are set before training. They include the learning rate, batch size, and number of epochs, which influence the training process and model performance.

**23. Model Parameters**

Model parameters in the neural network for the Sign Language dataset include the weights and biases that are learned from the training data. These parameters are adjusted during training to minimize the loss function and improve classification accuracy.

**24. Regularization**

Regularization techniques are used in the neural network for the Sign Language dataset to prevent overfitting by adding a penalty to the loss function for large weights. Common methods include L1 and L2 regularization, which help the model generalize better to new data.

**25. Dropout**

Dropout is a regularization technique applied during the training of the neural network for the Sign Language MNIST dataset. It randomly sets a fraction of neurons to zero in each training iteration, preventing the network from becoming too reliant on specific neurons and improving generalization.

**26. Weight Initialization**

Weight initialization refers to the process of setting the initial weights of the neural network for the Sign Language MNIST dataset before training begins. Proper weight initialization can lead to faster convergence and better performance by preventing issues such as vanishing or exploding gradients.

**27. Normalization**

Normalization in the context of the Sign Language dataset involves scaling the pixel values of the images to a standard range, such as 0-1. This process helps the neural network learn more effectively by ensuring consistent input data.

**28. Standardization**

Standardization involves transforming the pixel values of the Sign Language images to have a mean of zero and a standard deviation of one. This preprocessing step ensures that the data is centered and scaled, improving the training process and performance of the neural network.