**Deep Learning Assignment1**

**-MUGESH V**

**Problem Description**

**Input:** 28x28x3 grayscale of skin cancer

**Output:** A digit between 0 to 6

**Sample Data**

The MNIST dataset consists of 16,000 training images and 4,000 testing images. Each image is

labeled with the corresponding digit (0-6).

**Neural Network**

A neural network consists of algorithms that aim to identify underlying relationships in a dataset by simulating the operations of the human brain. It serves as the basis for most artificial intelligence systems.

**Neuron**

A neuron also known as a node is the basic unit of a neural network. It receives input processes it and then transmits the output to the next layer of neurons.

In a neural network, a layer is a group of neurons working together at a specific depth. Each layer transforms the input data into a more abstract level.

**Input Layer:**

The input layer is the initial layer of a neural network that receives the original data and passes it on to the next layer.The first layer of the neural network that receives the input data.

For MNIST, it would have 784 neurons (28x28 pixels).

**Hidden Layer:**

Hidden layers are the intermediary layers between the input and output layers. They carry out computations and feature transformations on the input data.Intermediate layers between input and output layers. They perform transformations on the input data to find patterns.

Output Layer: The final layer of the neural network that produces the output.

For MNIST, it has 10 neurons (one for each digit).

**Output Layer**

The output layer is the final layer in a neural network that provides the predicted result based on the transformations from the hidden layers.The final layer of the neural network that produces the output.

MNIST, it has 10 neurons (one for each digit).

**Convolutional Layer**

A convolutional layer is a type of layer in a Convolutional Neural Network (CNN) that applies convolution operations to the input, capturing spatial hierarchies in the data, particularly useful for image and video recognition tasks.

**Convolutional Neural Network (CNN)**

5 Convultion network has used layer type-I(conv2d),Output shape(None, 28, 28, 16),Parameter(448),type-II(conv2d\_1),Output shape((None,12,12,32),Parameter(4640),Type-III(conv2d\_2),Output shape (None, 10, 10, 64),Type-IV(conv2d\_3),Output shape (None, 3, 3, 128) ,Parameter(73856),TypeV(conv2d\_4),Output shape (None, 1, 1, 256),Parameter(295168)

A CNN is a class of deep neural networks, commonly used for analyzing visual imagery. They are designed to automatically and adaptively learn spatial hierarchies of features through backpropagation.

**Recurrent Neural Network (RNN)**

An RNN is a class of neural networks where connections between nodes can create cycles, allowing information to persist. They are particularly suited for sequential data like time series or natural language.

**Activation Function**

In this dataset we used ReLU and softmax as Activation function

An activation function determines the output of a neuron given an input or set of inputs. It introduces non-linearity into the network, allowing it to learn complex patterns.

**ReLU (Rectified Linear Unit)**

The Rectified Linear Unit (ReLU) activation function is defined as the positive part of its argument: f(x) = max(0, x). It is known for being computationally efficient and helps address the vanishing gradient problem.

**Softmax**

Softmax is an activation function typically used in the output layer of neural networks for classification purposes. It provides a probability distribution across different classes.

**Sigmoid**

The sigmoid activation function produces an output between 0 and 1, following the formula f(x) = 1 / (1 + e^(-x)). It is commonly used in binary classification tasks.

**Tanh**

The hyperbolic tangent (tanh) activation function generates an output between -1 and 1, given by the equation f(x) = tanh(x). It is zero-centered and often preferred over sigmoid for hidden layers.

**Forward Propagation**

Forward propagation refers to the process of passing input data through the neural network to compute the final output.

**Backpropagation**

Backpropagation is a technique used to compute the gradient of the loss function with respect to each weight in the neural network, employing the chain rule for optimization.

**Loss Function**

In this data set Sparse\_categorical\_crossentropy used for loss function

A loss function evaluates how well the predictions of the neural network align with the actual results. Examples include Mean Squared Error (MSE) and Cross-Entropy Loss.

**Cost Function**

Similar to a loss function, a cost function assesses the performance of the neural network by summing the losses across multiple data points.

**Gradient Descent**

Gradient descent is an optimization method employed to minimize the loss function by iteratively adjusting the model parameters in the direction of the negative gradient.

**Learning Rate**

Learning rate is 0-1

The learning rate is a hyperparameter that determines the extent of parameter updates based on the gradient during each optimization step.

**Batch Size**

Batch Size is 128

Batch size represents the number of training examples utilized in a single iteration of gradient descent, influencing the stability and efficiency of the training process.

**Epoch**

Epoch is 50

An epoch denotes a complete pass through the entire training dataset, which is typically done multiple times when training a neural network.

**Overfitting**

happens when a model excessively learns the training data, capturing noise and irrelevant details that do not generalize well to new data, resulting in poor performance on the validation/test set.

**Underfitting**

occurs when a model is too simplistic to capture the underlying patterns in the data, leading to poor performance on both the training and validation/test sets.

**Training set**

Training set is 16,000 data set

The training set is the part of the dataset utilized to train the neural network, enabling the model to understand the patterns and relationships within the data.

**Validation Set**

Validation splite is 0.2

A validation set is a subset of the data used to provide an unbiased evaluation of the model during training and to fine-tune hyperparameters.

**Test set**

Testing set is 4,000 data set

The test set is a distinct portion of the data used to evaluate the final performance of the model after training.

**Cross Validation**

Cross-validation is a method for assessing the model's performance by dividing the original training set into multiple subsets, training the model on some subsets, and validating it on others.

**Hyper parameter**

Hyperparameters are predefined parameters, such as learning rate, batch size, and number of layers, that control the training process and are set before training begins.

**Model parameters**

Total params: 504103

Trainable params: 502983

Non-trainable params: 1120

Model parameters are the internal variables of the model learned from the training data, including weights and biases.

**Regularization**

Regularization techniques are employed to prevent overfitting by imposing additional constraints or penalties on the loss function, such as L1 and L2 regularization.

**Dropout**

Dropout is a regularization technique where randomly selected neurons are omitted during training to prevent the network from relying too heavily on any single neuron and to reduce overfitting.

**Weight Initialization**

Weight Initialization is 504103

Setting the initial values of the weights in a neural network before training is known as weight initialization. This step plays a crucial role in determining the convergence and overall performance of the model.

**Normalization**

Scaling input data to have a mean of zero and a standard deviation of one is what normalization entails. This technique aids in expediting the training process and enhancing the model's performance.

**Standardization**

Standardization involves scaling input data to a standard range, usually with zero mean and unit variance. By doing so, it guarantees that all input features have an equal impact on the model's learning process.