TASK 17

SHERAZ BIN TAHIR

MAIL: BAIGS543@GMAIL.COM

Practical Code Activities

Deep learning:

It is a subset of machine learning that focuses on neural networks with many layers (deep neural networks). Here are the key fundamentals of deep learning:

1. Neural Networks

- **Artificial Neurons:** Inspired by biological neurons, these are the basic units of a neural network, performing simple computations.
- **Layers:** Neural networks consist of layers of neurons:
- **Input Layer:** The first layer that receives the raw input data.
- **Hidden Layers:** Intermediate layers that perform computations and feature transformations.
- **Output Layer:** The final layer that produces the output.

2. Activation Functions

- Functions that introduce non-linearity into the network, allowing it to learn complex patterns.
- Common activation functions include ReLU (Rectified Linear Unit), sigmoid, and tanh.

3. Forward Propagation

- The process by which input data passes through the network layer by layer to generate an output.

4. Loss Function

- A function that measures the difference between the predicted output and the actual output.
- Common loss functions include mean squared error (MSE) for regression tasks and cross-entropy for classification tasks.

5. Backpropagation

- A technique for training neural networks by adjusting the weights to minimize the loss function.
- Involves calculating the gradient of the loss function with respect to each weight and updating the weights accordingly.

6. Optimization Algorithms

- Methods used to minimize the loss function by updating the network's weights.
- Common optimization algorithms include Stochastic Gradient Descent (SGD), Adam, and RMSprop.

7. Learning Rate

- A hyperparameter that controls the size of the steps taken during the optimization process.
- It needs to be carefully chosen; too high can cause the model to converge too quickly to a suboptimal solution, while too low can make the training process unnecessarily slow.

8. Regularization

- Techniques to prevent overfitting by penalizing large weights.
- Common regularization methods include L1 and L2 regularization, dropout, and batch normalization.

9. Data Preprocessing

- The process of cleaning and transforming raw data into a suitable format for training.
- Includes normalization, standardization, and data augmentation.

10. Training and Validation

- **Training Set:** The portion of the data used to train the model.
- **Validation Set:** The portion of the data used to tune hyperparameters and prevent overfitting.
- **Test Set:** The portion of the data used to evaluate the final model's performance.

11. Model Evaluation Metrics

- Metrics used to evaluate the performance of the model, such as accuracy, precision, recall, F1 score for classification, and mean absolute error (MAE) for regression.

12. Hyperparameter Tuning

- The process of optimizing the hyperparameters (e.g., learning rate, number of layers, number of neurons) to improve model performance.

13. Deep Learning Frameworks

- Libraries and tools that simplify the implementation of deep learning models.
- Popular frameworks include TensorFlow, Keras, PyTorch, and MXNet.

14. Transfer Learning

- A technique where a pre-trained model on a large dataset is fine-tuned on a smaller, task-specific dataset.

15. Convolutional Neural Networks (CNNs)

- Specialized neural networks for processing structured grid data like images.
- Consist of convolutional layers, pooling layers, and fully connected layers.

16. Recurrent Neural Networks (RNNs)

- Specialized neural networks for sequential data.
- Include architectures like LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) which handle long-term dependencies.

Practical Code Activities in Deep Learning.

1. Building a Simple Neural Network

- Task: Implement a basic neural network using TensorFlow or PyTorch to classify digits from the MNIST dataset.

- Steps:

1. Load the MNIST dataset.
2. Define the neural network architecture (input layer, hidden layers, output layer).
3. Compile the model with an appropriate loss function and optimizer.
4. Train the model on the training data.
5. Evaluate the model on the test data.

2. Image Classification with Convolutional Neural Networks (CNNs)

- Task: Implement a CNN to classify images from the CIFAR-10 dataset.

- Steps:

1. Load the CIFAR-10 dataset.
2. Define the CNN architecture (convolutional layers, pooling layers, dense layers).
3. Compile the model with an appropriate loss function and optimizer.
4. Train the model on the training data.
5. Evaluate the model on the test data.

3. Text Classification with Recurrent Neural Networks (RNNs)

- Task: Implement an RNN or LSTM to classify movie reviews as positive or negative using the IMDB dataset.

- Steps:

1. Load the IMDB dataset.
2. Preprocess the text data (tokenization and padding).
3. Define the RNN or LSTM architecture.

4. Compile the model with an appropriate loss function and optimizer.
5. Train the model on the training data.
6. Evaluate the model on the test data.

4. Transfer Learning with Pre-trained Models

- **Task:** Use a pre-trained model like VGG16 to classify images from a custom dataset.

- **Steps:**

1. Load the pre-trained model (e.g., VGG16).
2. Modify the model to suit your custom classification task.
3. Preprocess your custom dataset.
4. Train the modified model on your custom dataset.
5. Evaluate the model.

5. Generative Adversarial Networks (GANs) for Image Generation

- **Task:** Implement a GAN to generate new images similar to the MNIST dataset.

- **Steps:**

1. Define the generator and discriminator models.
2. Compile the GAN model.
3. Train the GAN using the MNIST dataset.
4. Generate new images using the trained generator.

These activities cover a range of tasks in deep learning, from basic neural networks to advanced techniques like GANs, providing hands-on experience with practical implementations.