

**The International Institute Of Information Technology - Hyderabad**

**Report on Deep Learning**

**By: Anushka Raut**

**UGMR NO: 20230025**

**Overview of Deep Learning**  
Machine learning, a subset of artificial intelligence (AI), includes deep learning as one of its branches. It focuses on modeling and comprehending complicated patterns in data using multilayered neural networks, or "deep" neural networks. Deep learning models—particularly deep neural networks—have shown effective in a range of applications, including speech recognition, image recognition, and natural language processing.

**Key Characteristics of Deep Learning:**

* **Hierarchical Feature Learning:** deep learning models automatically pick up several levels of abstraction when representing data. Deep neural networks capture higher-level information from the raw input at each layer.
* **End-to-End Learning:** Deep learning models are very useful for a range of applications because they can learn from raw data directly, eliminating the requirement for manual feature extraction.
* **Scalability:** These models can handle big datasets and complicated problems because they can scale with the availability of data and computational power.

**Applications of Deep Learning:**  
  
1. **Picture and Video Analysis:** Facial recognition, object identification, picture classification, and video analysis.  
2. **Natural language processing (NLP)** includes sentiment analysis, text production, machine translation, and question-answering.  
3. **Speech Recognition**: speech-to-text systems, automated transcription services, and voice-activated personal assistants.  
4. **Healthcare:** Drug discovery, medical image analysis, and disease prediction.

#### **Overview of the Kuzushiji-MNIST Dataset**

The Kuzushiji-MNIST dataset is a challenging and more complex alternative to the original MNIST dataset. While the MNIST dataset consists of images of handwritten digits (0-9), Kuzushiji-MNIST features Japanese characters from historical documents. This dataset consists of 70,000 grayscale images of size 28x28 pixels, where each image represents one of 10 classes of Japanese characters.

**Dataset Details:**

* **Training Set:** 60,000 images.
* **Test Set:** 10,000 images.
* **Classes:** 10 classes of Japanese characters, each class representing a different character.

**Purpose and Importance:** The Kuzushiji-MNIST dataset is designed to provide a more challenging benchmark for image classification tasks due to the intricate and diverse nature of Japanese characters. It helps in evaluating the performance of machine learning models on more complex and culturally significant handwritten data.

**Colab Notebook Link :**

<https://colab.research.google.com/drive/1jIj81WxvgdS8WJcrIN1kvkJ5Q3rpkLdA#scrollTo=avSlN7KUVnQs>

#### **Convolutional Neural Networks (CNNs)**

#### **Introduction:**

#### Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for processing grid-like data, such as images. They are particularly well-suited for image-related tasks due to their ability to capture spatial hierarchies in the data.

#### **Graph Neural Networks (GNNs)**

#### **Introduction:**

#### Graph Neural Networks (GNNs) are a class of deep learning models designed to handle data with a graph structure. Graphs consist of nodes (representing entities) and edges (representing relationships between entities). GNNs are used to model and analyze complex relationships in data, making them suitable for various applications involving relational data.

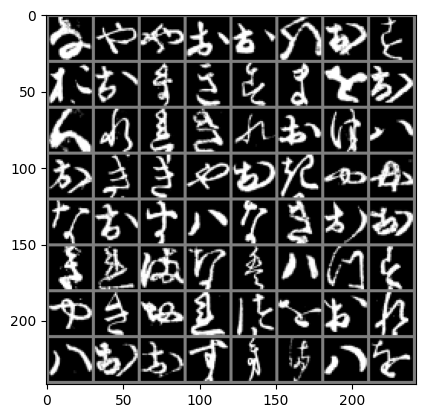
#### **Training and Optimization:**

* **Loss Functions:** Depending on the task, different loss functions are used. For node classification, binary cross-entropy loss is common. For graph regression tasks, mean squared error is used.
* **Optimizers:** Similar to CNNs, gradient-based optimization methods like Adam and SGD are used to train GNNs.

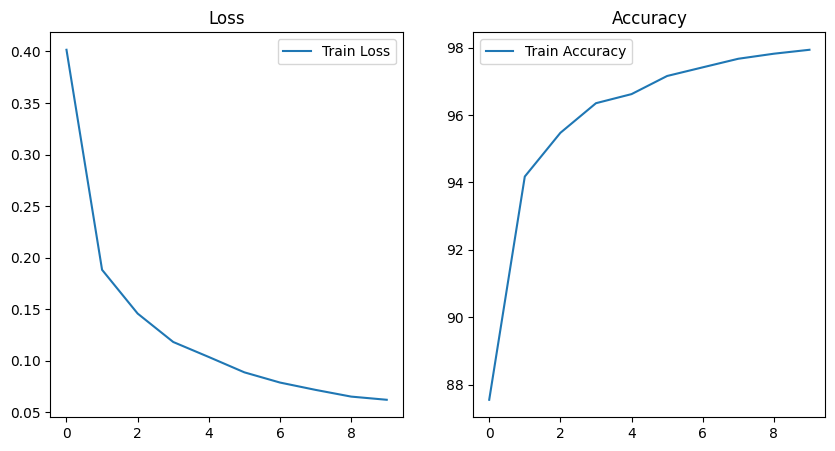
**Methodology:**

#### **Step-by-Step Process:**

1. **Data Handling:**
   * **Downloading and Loading the Dataset:** The Kuzushiji-MNIST dataset is downloaded and loaded using the torchvision.datasets.KMNIST class. The dataset includes a training set with 60,000 images and a test set with 10,000 images.
   * **Applying Transformations:** Before feeding the data into the neural network, we apply transformations to the dataset:
     + **Normalization:** This scales the pixel values of the images to a range of [-1, 1], which helps stabilize the training process and speeds up convergence.
     + **Tensor Conversion:** The images are converted to PyTorch tensors, which are the required format for training with PyTorch.
2. **Visualizing the Dataset:**
   * To understand the nature of the dataset, we visualize some sample images along with their corresponding labels. This helps in getting a sense of the variations and complexities of the characters in the dataset.



1. **Defining the CNN Model:**
   * The architecture of the CNN is defined to include several key layers:
     + **Convolutional Layers:** These layers apply convolution operations to the input images using filters (kernels) to extract local features such as edges and textures. The filters slide over the images to produce feature maps that capture different aspects of the input.
2. **Training the Model:**
   * The training process involves optimizing the model using an optimizer (like Adam) and a loss function (like cross-entropy loss):
     + **Optimizer:** The Adam optimizer is used to minimize the loss function and update the model's weights. Adam is chosen for its efficiency and effectiveness in handling large datasets and sparse gradients.
     + **Loss Function:** Cross-entropy loss is used to measure the difference between the predicted and actual labels. This loss function is suitable for classification tasks.
     + **Epochs:** The model is trained for multiple epochs. In each epoch, the entire training dataset is passed through the model, and the weights are updated to minimize the loss.
     + **Tracking Metrics:** During training, we track metrics like loss and accuracy to monitor the model's performance. Loss indicates how well the model is learning, while accuracy measures the proportion of correctly classified images.
3. **Evaluating the Model:**
   * After training, the model is evaluated on the test set to measure its performance. The accuracy is calculated to assess how well the model generalizes to unseen data.
   * Evaluation helps in understanding the model's capability to correctly classify new images and identify areas for improvement.
4. **Plotting of loss and accuracy:**



1. **Visualizing Training Metrics and Model Predictions:**

GroundTruth: 2 9 3 8 3 3 8 3 Predicted: 2 9 3 8 3 3 8 3

**Conclusion:** The provided methodology successfully demonstrates the end-to-end process of training a Convolutional Neural Network (CNN) on the Kuzushiji-MNIST dataset. The process includes data preparation, model definition, training, evaluation, and visualization. The CNN model learns to recognize Japanese characters with a certain level of accuracy, showcasing the effectiveness of deep learning techniques for image classification tasks. The visualization of training metrics and model predictions provides insights into the model's performance and learning dynamics. This practical example highlights the steps involved in implementing deep learning models using PyTorch, making it a valuable reference for similar tasks. In summary, the report provides a comprehensive overview of deep learning, the Kuzushiji-MNIST dataset, CNNs, and GNNs, along with a detailed methodology for building and evaluating a CNN on the dataset. The insights gained from this process can be applied to other image classification tasks and further deep learning applications.