

410255: LP-V. DL 3.B 22-23 (2 <sup>nd</sup> Sem)					
Student Name :					
BE Computer Division			Roll No		Batch
Experiment No: 3B		Convolutional neural network (CNN) Using Fashion Dataset			

**Aim :** Convolutional neural network (CNN) (Any One from the following) :

- A. Use any dataset of plant disease and design a plant disease detection system using CNN.
- B. Use MNIST Fashion Dataset and create a classifier to classify fashion clothing into categories.

**Title of the Assignment (3.B): Use MNIST Fashion Dataset and create a classifier to classify fashion clothing into categories.**

**Objective of the Assignment:** Students should be able to Classify movie reviews into positive reviews and "negative reviews on IMDB Dataset.

**Prerequisite:**

1. Basic of programming language
2. Concept of Classification
3. Concept of Deep Neural Network

**Contents for Theory:**

1. What is Classification
2. Example of Classification
3. What is CNN?
4. How Deep Neural Network Work on Classification
5. Code Explanation with Output

**What is Classification?**

Classification is a type of supervised learning in machine learning that involves categorizing data into predefined classes or categories based on a set of features or characteristics. It is used to predict the class of new, unseen data based on the patterns learned from the labeled training data.

In classification, a model is trained on a labeled dataset, where each data point has a known class label. The model learns to associate the input features with the corresponding class labels and can then be used to classify new, unseen data.

For example, we can use classification to identify whether an email is spam or not based on its content and metadata, to predict whether a patient has a disease based on their medical records and symptoms, or to classify images into different categories based on their visual features. Classification algorithms can vary in complexity, ranging

from simple models such as decision trees and k-nearest neighbors to more complex models such as support vector machines and neural networks. The choice of algorithm depends on the nature of the data, the size of the dataset, and the desired level of accuracy and interpretability.

**Example-** Classification is a common task in deep neural networks, where the goal is to predict the class of an input based on its features. Here's an example of how classification can be performed in a deep neural network using the popular MNIST dataset of handwritten digits. The MNIST dataset contains 60,000 training images and 10,000 testing images of handwritten digits from 0 to 9. Each image is a grayscale 28x28 pixel image, and the task is to classify each image into one of the 10 classes corresponding to the 10 digits.

We can use a convolutional neural network (CNN) to classify the MNIST dataset. A CNN is a type of deep neural network that is commonly used for image classification tasks.

**What is CNN** Convolutional Neural Networks (CNNs) are commonly used for image classification tasks, and they are designed to automatically learn and extract features from input images. Let's consider an example of using a CNN to classify images of handwritten digits.

In a typical CNN architecture for image classification, there are several layers, including convolutional layers, pooling layers, and fully connected layers. Here's a diagram of a simple CNN architecture for the digit classification task:

The input to the network is an image of size 28x28 pixels, and the output is a probability distribution over the 10 possible digits (0 to 9). The convolutional layers in the CNN apply filters to the input image, looking for specific patterns and features. Each filter produces a feature map that highlights areas of the image that match the filter. The filters are learned during training, so the network can automatically learn which features are most relevant for the classification task.

The pooling layers in the CNN downsample the feature maps, reducing the spatial dimensions of the data. This helps to reduce the number of parameters in the network, while also making the features more robust to small variations in the input image.

The fully connected layers in the CNN take the flattened output from the last pooling layer and perform a classification task by outputting a probability distribution over the 10 possible digits. During training, the network learns the optimal values of the filters and parameters by minimizing a loss function. This is typically done using stochastic gradient descent or a similar optimization algorithm. Once trained, the network can be used to classify new images by passing them through the network and computing the output probability distribution.

Overall, CNNs are powerful tools for image recognition tasks and have been used successfully in many applications, including object detection, face recognition, and medical image analysis. CNNs have a wide range of applications in various fields, some of which are:

**Image classification:** CNNs are commonly used for image classification tasks, such as identifying objects in images and recognizing faces.

**Object detection:** CNNs can be used for object detection in images and videos, which involves identifying the location of objects in an image and drawing bounding boxes around them.

**Semantic segmentation:** CNNs can be used for semantic segmentation, which involves partitioning an image into segments and assigning each segment a semantic label (e.g., "road", "sky", "building").

**Natural language processing:** CNNs can be used for natural language processing tasks, such as sentiment analysis and text classification.

**Medical imaging:** CNNs are used in medical imaging for tasks such as diagnosing diseases from X-rays and identifying tumors from MRI scans.

**Autonomous vehicles:** CNNs are used in autonomous vehicles for tasks such as object detection and lane detection.

**Video analysis:** CNNs can be used for tasks such as video classification, action recognition, and video captioning.

Overall, CNNs are a powerful tool for a wide range of applications, and they have been used successfully in many areas of research and industry.

**How Deep Neural Network Work on Classification using CNN**Deep neural networks using CNNs work on classification tasks by learning to automatically extract features from input images and using those features to make predictions. Here's how it works:

**Input layer:** The input layer of the network takes in the image data as input.

**Convolutional layers:** The convolutional layers apply filters to the input images to extract relevant features. Each filter produces a feature map that highlights areas of the image that match the filter.

**Activation functions:** An activation function is applied to the output of each convolutional layer to introduce non-linearity into the network.

**Pooling layers:** The pooling layers downsample the feature maps to reduce the spatial dimensions of the data.

**Dropout layer:** Dropout is used to prevent overfitting by randomly dropping out a percentage of the neurons in the network during training.

Fully connected layers: The fully connected layers take the flattened output from the last pooling layer and perform a classification task by outputting a probability distribution over the possible classes. Softmax activation function: The softmax activation function is applied to the output of the last fully connected layer to produce a probability distribution over the possible classes. Loss function: A loss function is used to compute the difference between the predicted probabilities and the actual labels.

Optimization: An optimization algorithm, such as stochastic gradient descent, is used to minimize the loss function by adjusting the values of the network parameters.

Training: The network is trained on a large dataset of labeled images, adjusting the values of the parameters to minimize the loss function.

Prediction: Once trained, the network can be used to classify new images by passing them through the network and computing the output probability

**MNIST Dataset**The MNIST Fashion dataset is a collection of 70,000 grayscale images of 28x28 pixels, representing 10 different categories of clothing and accessories. The categories include T-shirts/tops, trousers, pullovers, dresses, coats, sandals, shirts, sneakers, bags, and ankle boots. The dataset is often used as a benchmark for testing image classification algorithms, and it is considered a more challenging version of the original MNIST dataset which contains handwritten digits. The MNIST Fashion dataset was released by Zalando Research in 2017 and has since become a popular dataset in the machine learning community.

The MNIST Fashion dataset is a collection of 70,000 grayscale images of 28x28 pixels each. These images represent 10 different categories of clothing and accessories, with each category containing 7,000 images.

The categories are as follows:

T-shirt/tops

Trousers

Pullovers

Dresses

Coats

Sandals

Shirts

Sneakers

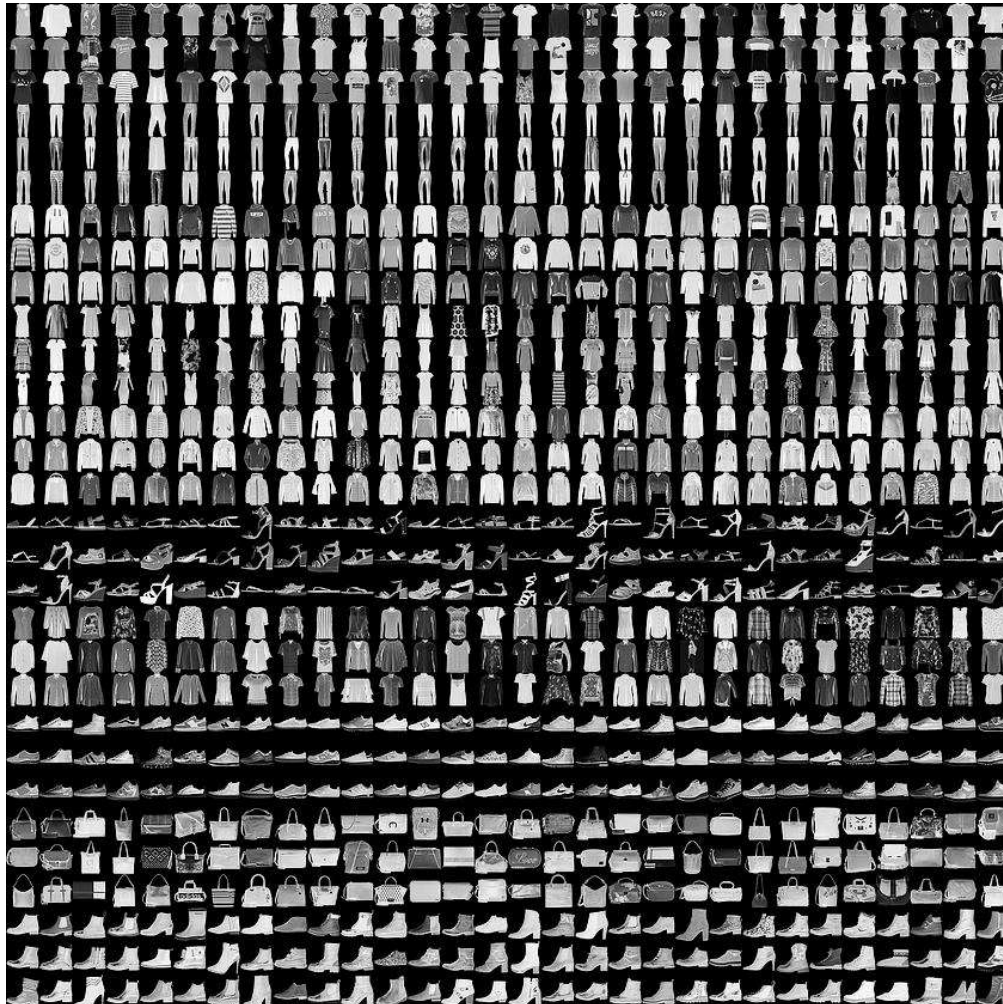
Bags

Ankle boots

The images were obtained from Zalando's online store and are preprocessed to be normalized and centered. The training set contains 60,000 images, while the test set

contains 10,000 images. The goal of the dataset is to accurately classify the images into their respective categories.

The MNIST Fashion dataset is often used as a benchmark for testing image classification algorithms, and it is considered a more challenging version of the original MNIST dataset which contains handwritten digits. The dataset is widely used in the machine learning community for research and educational purposes.



Here are the general steps to perform Convolutional Neural Network (CNN) on the MNIST Fashion dataset:

- Import the necessary libraries, including TensorFlow, Keras, NumPy, and Matplotlib.

- Load the dataset using Keras' built-in function, `keras.datasets.fashion_mnist.load_data()`. This will provide the training and testing sets, which will be used to train and evaluate the CNN.
- Preprocess the data by normalizing the pixel values between 0 and 1, and reshaping the images to be of size (28, 28, 1) for compatibility with the CNN.
- Define the CNN architecture, including the number and size of filters, activation functions, and pooling layers. This can vary based on the specific problem being addressed.
- Compile the model by specifying the loss function, optimizer, and evaluation metrics. Common choices include categorical cross-entropy, Adam optimizer, and accuracy metric.
- Train the CNN on the training set using the `fit()` function, specifying the number of epochs and batch size.
- Evaluate the performance of the model on the testing set using the `evaluate()` function. This will provide metrics such as accuracy and loss on the test set.
- Use the trained model to make predictions on new images, if desired, using the `predict()` function.

**Conclusion-** In this way we can Classify fashion clothing into categories using CNN

### Assignment Question

1. What is Binary Classification?
2. What is binary Cross Entropy?
3. What is Validation Split?
4. What is the Epoch Cycle?
5. What is Adam Optimizer?

Attendance (5 Marks)	Conduction (5 Marks)	Oral (5 Marks)	Total (15 Marks)	Sign of Teacher