CONQUERING FASHION MNIST WITH CNNs USING COMPUTER VISION

TEAM NAME: THE INITIATORS

TEAM MEMBER'S

ROHITH D

VINAY KUMAR VIDYA

SAI KUMAR PENTALA

COLLEGE MENTOR NAME: Mrs. D KALPANA

INTEL INDUSTRY MENTOR NAME: Mr. MOHAN NIKAM

ABSTRACT

Fashion MNIST is a widely used benchmark dataset in the field of computer vision, specifically for image classification tasks. It serves as a replacement for the traditional MNIST dataset, which consists of grayscale images of handwritten digits. The FashionMNIST dataset consists of 60,000 training images and 10,000 testing images, divided into 10 different categories of fashion items. This abstract focuses on the application of Convolutional Neural Networks (CNNs) for the classification of FashionMNIST dataset.

In this study, we propose a CNN architecture to classify the FashionMNIST dataset. The proposed model consists of multiple convolutional layers, pooling layers, and fully connected layers. We utilize the rectified linear activation function (ReLU) for the convolutional layers and softmax activation for the final output layer to obtain class probabilities.

To train and evaluate the CNN model, we split the FashionMNIST dataset into training and testing sets. We preprocess the images by normalizing pixel values and converting them into a suitable format for training the CNN. The model is trained using the training set, and its performance is evaluated using the testing set. We utilize popular optimization techniques such as stochastic gradient descent (SGD) with backpropagation to update the model weights and minimize the loss function during training.

The application of CNNs on the FashionMNIST dataset not only provides accurate classification of fashion items but also opens doors for various practical applications such as image search, recommendation systems, and virtual try-on experiences in the fashion industry. The proposed CNN architecture serves as a foundation for further research and exploration in the field of computer vision and deep learning, aiming to improve the accuracy and efficiency of image classification tasks.

INTRODUCTION

The classification of the FashionMNIST dataset involves the task of categorizing images of fashion items into their respective classes. The FashionMNIST dataset serves as a replacement for the traditional MNIST dataset, which consists of grayscale images of handwritten digits. FashionMNIST, on the other hand, comprises images of various fashion items such as T-shirts, trousers, pullovers, dresses, coats, sandals, shirts, sneakers, bags, and ankle boots. It has become a popular benchmark dataset in the field of computer vision and serves as a challenging task for developing and evaluating image classification models.

The FashionMNIST dataset consists of a total of 70,000 images, divided into 60,000 training images and 10,000 testing images. Each image is a grayscale image with a size of 28x28 pixels. The dataset is evenly distributed among the ten different classes, with 6,000 images per class for training and 1,000 images per class for testing.

The classification process involves training a model on the training set, where the model learns to extract meaningful features from the images and map them to the corresponding fashion classes. The model's performance is then evaluated on the testing set, where its ability to generalize and accurately classify unseen fashion images is assessed. The evaluation metrics typically used for this task include accuracy, precision, recall, and F1-score.

To enhance the performance of classification models on the FashionMNIST dataset, various techniques can be employed. These include data augmentation, where additional training examples are generated by applying transformations such as rotations, translations, and scaling to the original images. Additionally, model regularization techniques like dropout and weight decay can be used to prevent overfitting.

Overall, the classification of the FashionMNIST dataset presents an interesting and challenging problem in the field of computer vision. It provides a platform for developing and evaluating robust and accurate image classification models, paving the way for advancements in fashion-related applications and further research in the domain of computer vision and deep learning.

BACKGROUND AND RELATED WORK

In image classification different methods are used such as methods based on low-level image feature representation which consider image as a collection of low-level characteristics like texture, shape, size, color, etc. and methods based on mid-level visual feature constructions for image classification tasks. Nowadays, usage of deep neural networks and neural-networks to obtain image representation is trending. Such architectures allow us to extract features from a specified layer of trained neural network and then use extracted feature maps as a numeric image representation. There are a large number of publications related to the image processing with neural networks. Our work is related to this type of research, where CNN are used for classifying images. Image classification in the fashion domain has numerous benefits and applications and has various research works have been presented about it.

One among the previous studies has reviewed deep neural networks is able to attain record breaking outputs on very challenging dataset using supervised learning . Their network contains 5 CNN layers and 3 fully-connected layers. They worked one among the largest ConvNets on the ImageNet dataset subsets and achieved best ever results reported on this. This neural network includes a number of novel and unusual features that increase the performance such as relu nonlinearity, overlapping pooling etc. and decrease the time for training. They have used various effective methods for reducing overfitting, which are data augmentation and dropout. Fashion-MNIST dataset has been presented by Zalando Research F-MNIST is proposed to intend for a direct drop-in substitute for the classical MNIST handwritten digits dataset which has been considered as the benchmark for machine learning techniques, as it contains the same structure, image format and size of train and test set splits.

F-MNIST is a kind of more challenging task than classical MNIST dataset. Original MNIST dataset— commonly used as the “Hello World” of machine learning applications in computer vision, is overused, too easy and cannot represent modern computer vision tasks. Researchers at Zalando company have developed a new image classification dataset called F-MNIST in hopes that it should be a substitute for original MNIST dataset. This newly introduced dataset contains images of various products of clothing and accessories—such as t-shirts, coats, shoes, and other fashion items. Each image is a 28x28 grayscale fashion article image, related with a label from ten categories (t-shirt/top to ankle boots). F-MNIST is the most challenging dataset and gives us a lot more room for improving the model. Hence it could be a potential substitute for classical MNIST.

METHODOLOGY

Classification of images is used in various applications, ranging from facial recognition to self-driving cars. ConvNets are current state-of-the-art models for object classification. ConvNets are being used everywhere. For getting started with image classification the handwritten digits MNIST dataset is easier and mostly overused.

We propose to classify fashion products images using hyperparameters optimization methods and regularization techniques implementing with CNN. Almost in all computer vision tasks ConvNets are being used. ConvNets mainly consists of three phases. In the first phase a convolution operation occurs in between filters or kernels and input image of very small size and a feature map is produced. Each kernel in a ConvNet learns different features of the image. The convolution operation in ConvNet is simply a mathematical operation i.e. multiplication of the filter and image matrix. The convolution function between a 2D filter Q and 2D image P is,

$$C(m,n) = (P * Q)(m,n) = \sum\sum P(i,j)Q(m-i, n-j) \quad (1)$$

It can also be expressed as,

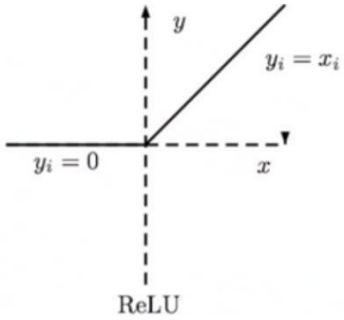
$$C(m,n) = (Q * P)(m,n) = \sum\sum P(m-i, n-j)Q(i,j) \quad (2)$$

For a 3x3 filter size the equation becomes,

$$C(m,n) = (Q * P)(m,n)$$

$$= \sum_{i=1}^{3}\sum_{j=1}^{3} P(m-i, n-j)Q(i,j) \quad (3)$$

The second phase of ConvNet model is Activation layer. Activation function introduces non-linearity to the model. Most prominent activation functions are ReLU (Rectified Linear Unit) [15], Tanh and Sigmoid. ReLU activation function is implemented in the proposed models. Usually, ReLU function is most popularly used in almost all theConvNets.



$$R(z) = \max(0, z)$$

$$R(z) = \begin{cases} 0 & for \ z < 0 \\ z & for \ z \geq 0 \end{cases}$$

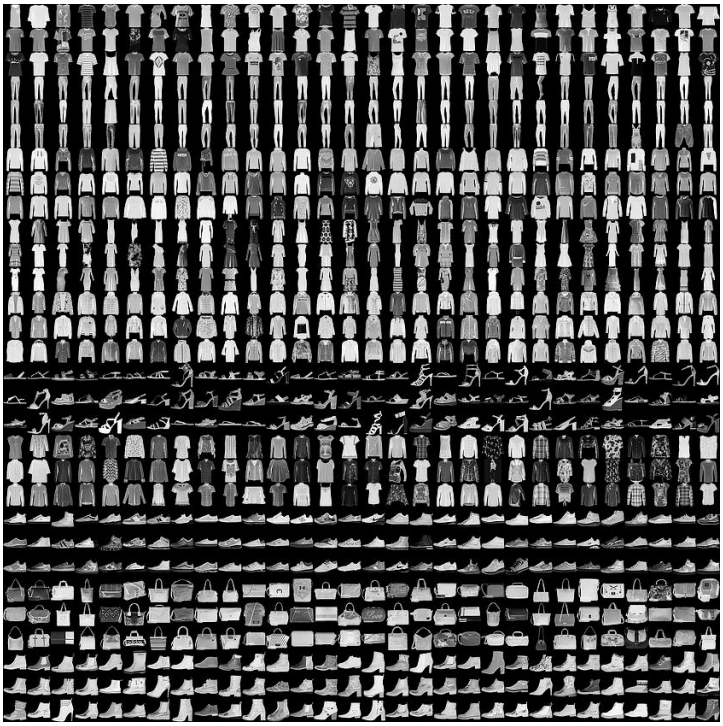
ReLU is best for hidden layers. $R(z)$ is zero when z is less than zero and $R(z)$ is equal to z when z is above or equal to zero. Other alternatives are sigmoid, tanh and other activation functions depending on the task. They are a crucial part of neural networks. The third phase is pooling function which is applied to resize the dimension of the input image to avoid overfitting. ConvNets often use pooling layers to decrease the size of the representation. Suppose we have a 4 x 4 input, and you want to apply a type of pooling called max pooling. And the output of this particular implementation of max pooling will be a 2 x 2 output.

In this work two different neural network architectures are proposed: 2 ConvNets and 4 ConvNets. The first one includes two convolutional layers. The last one includes four convolutional layer.

About Dataset

The Fashion MNIST dataset is a collection of 70,000 grayscale images of fashion products, divided into 10 categories, including T-shirts, trousers, pullovers, dresses, coats, sandals, shirts, sneakers, bags, and ankle boots. Each image is a 28x28 pixel square, resulting in 784 features.

- The objective is to train a model using the provided training set, consisting of 60,000 images, and evaluate its performance on the test set, which contains 10,000 images. The model should be able to correctly classify the test images into one of the 10 fashion categories.
- Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255.
- The training and test data sets have 785 columns. The first column consists of the class labels (see above), and represents the article of clothing. The rest of the columns contain the pixel-values of the associated image.
- Link to Dataset: https://www.kaggle.com/datasets/zalando-research/fashionmnist



Category	Label
Top/T-shirt	0
Trouser	1
Pullover	2
Dress	3
Coat	4
Sandal	5
Shirt	6
Sneaker	7
Bag	8
Ankle Boot	9

Building Model

A. Image Preprocessing

The F-MNIST database contains 70000 images of dimension 28x28. These images and their corresponding labels are separated as training data and test data. To prepare the data for training, some processing have applied on the images like resizing images, normalizing the pixel values etc. After doing the necessary processing on the image information's, the label data, we have converted it into categorical formats like label '5' should be represented as a vector format of [0, 0, 0, 0, 0, 1, 0, 0, 0, 0] to build the model.

Each image has 28 x 28 resolutions. The CNN accepts image input shape in a specific format. So we have reshaped our input. All the images in our dataset are in grayscale. Normalization is applied on the input images for getting the dimensions in same scale.

B. Convolutional Neural Network (CNN)

Among various deep learning architectures, ConvNets stands out for its unprecedented performance on computer vision.A special kind of artificial neural network is ConvNet which contains at least one convolutional layer. A typical ConvNet takes an input image, pass it through a set of layers convolution, non-linear activation, pooling (downsampling) and fully connected, and retrieve an output of classification labels. This output of this CNN layer is an activation map.

1) Optimizers

Optimization algorithms help us to minimize or maximize the objective function. Minimizing the loss by the training process is very important and has a main role in the operation of training of the neural network model. The two optimizers used in these architectures are

Adam optimization of the loss function. Adam work well across a wide range of deep learning architectures. Adam usually outperforms the rest followed very closely by the other adaptive learning rate methods, Adagrad and Adadelta. Adam optimizer can be calculated as

$$\Delta\theta_t = -\frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t$$

2) Batch size and Number of Epochs

Mini-batch is usually preferable in the learning process of ConvNets. A range of 16 to 128 batch size is a good choice to test with. ConvNet is sensitive to batch size. In this model we have used 64 and 128 as batch size for training images. Number of epochs is the number of complete pass through the entire training set. The number of epochs has increased until the difference of training and the test error is very small. Here, we have checked with 40 and 60 epochs.

3) Activation Function

Activation function is just a thing that should be added to the output at the end of any neural network. This is used to obtain the output of the neural network like yes or no. Depending upon the function it maps the resulting values in between -1 to 1 or 0 to 1 etc. ReLU is really popular in the last few years and it is used in this model.

Accuracy of Model

Accuracy : ~0.92790(approx)

Evaluate Accuracy

```
[ ]  test_loss, test_acc = model.evaluate(test_images,  test_labels, verbose=2)

     print('\nTest accuracy:', test_acc)

     313/313 - 1s - loss: 0.2623 - accuracy: 0.9276 - 633ms/epoch - 2ms/step

     Test accuracy: 0.9276000261306763
```

Inference time taken by the Model

-Inference time taken by the model in Google Colab - 4s 4ms

-Inference time taken by the model in Intel devcloud - 2s 56 ms

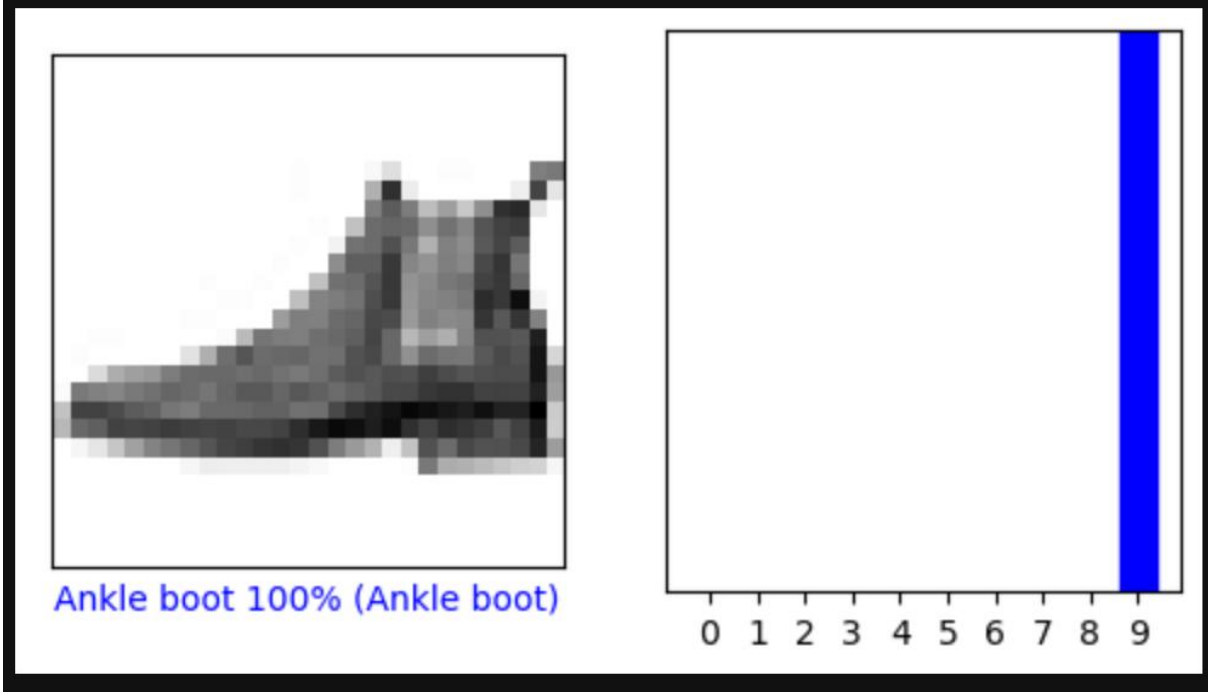
-Inference time taken by the model after Intel Optimization Openvino - 0.36 ms

Confusion matrix

Output Image Predictions

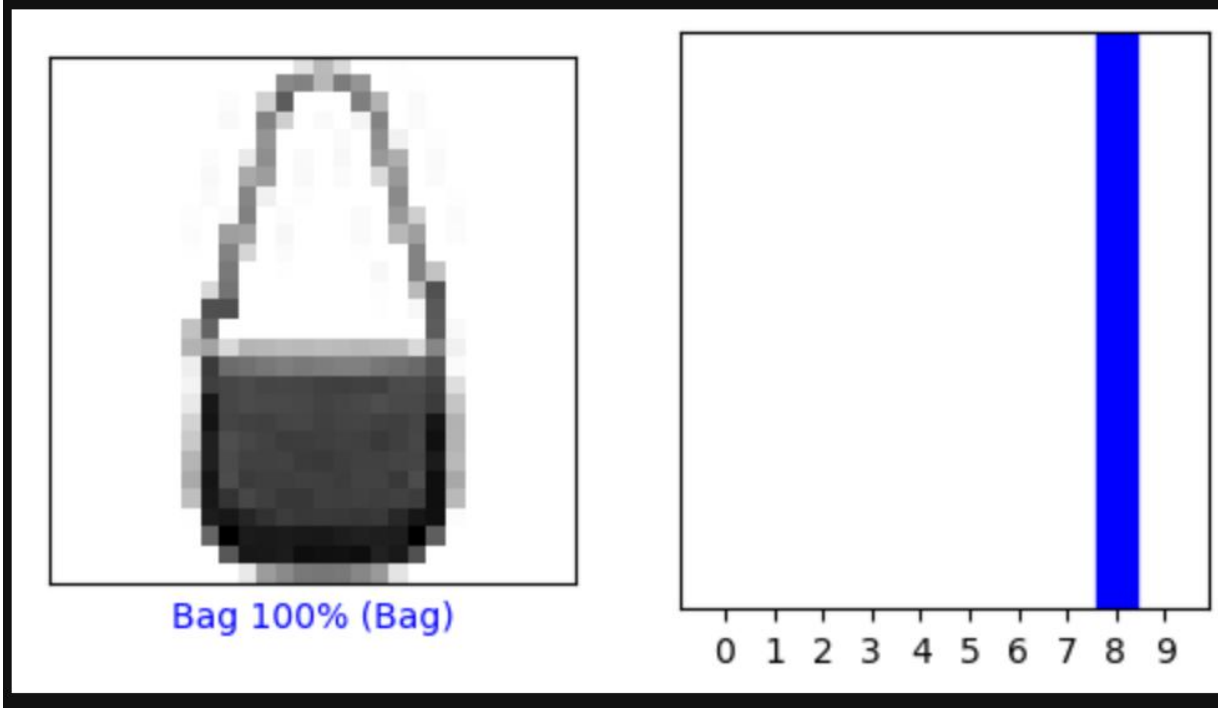
Output-1 :

```
i = 2874
plt.figure(figsize=(6,3))
plt.subplot(1,2,1)
plot_image(i, pred[i], test_labels, test_images)
plt.subplot(1,2,2)
plot_value_array(i, pred[i],  test_labels)
plt.show()
```



Output - 2

```
i = 9997
plt.figure(figsize=(6,3))
plt.subplot(1,2,1)
plot_image(i, predictions[i], test_labels, test_images)
plt.subplot(1,2,2)
plot_value_array(i, predictions[i],  test_labels)
plt.show()
```



Conculsion

1. With a complex sequential model with multiple convolution layers and 10 epochs for the training, we obtained an accuracy **~0.88** for test prediction. After investigating the validation accuracy and loss, we understood that the model is overfitting.We retrained the model with Dropout layers to the model to reduce overfitting.We confirmed the model improvement and with the same number of epochs for the training we obtained with the new model an accuracy of **~0.92** for test prediction.

2. Once the model has been developed in Jupyter Notebook it has taken an Inference time of **4s 36ms** to classify and predict the images.

3. The same code which has been executed in Intel devcloud Tensorflow Toolkit, it has taken an Inference time of **2s 56ms** to classify and predict the images.

4. But after applying the **Intel Optimization OpenVino(IR)**, it has taken only **0.36s** to classify and predict the images.

5. So, from this we can conclude that the Classification of Fashion MNIST CNN model gives more **Optimised** predictions when Tensorflow keras code is converted to **Intel Optimization OpenVino.**