CS F425: DEEP LEARNING

Assignment - 2 Report

On

A Convolutional Neural Network Architecture for Fashion MNIST

BY

SAMARTH SONI (2019A7PS1274H)

GAURVIT KUMAR (2019A7PS1278H)

OM AGARWAL (2019A7PS0052H)

DRAVYANSH JAIN (2019A7PS0063H)

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)

HYDERABAD CAMPUS

(DEC 2021)

CONTENTS:

Title page……………………………………………………….…………. 1

Introduction…………………………………………….……..……..... 3

Dataset And Framework ………………………………………………..4

Feature Engineering ……………………………………………………5

Result And Analysis ……………………………………………………6-9

INTRODUCTION:

In Deep Learning where features are hard to analyze, we aim to employ Neural Networks (allowing a machine to discover representations from data) instead of a knowledge-based approach/ simple ML model.

We use a hierarchy of multiple layers in a bid to suppress irrelevant information and amplify aspects of input that are important for decision making.

Building deep learning models requires taking design and architectural decisions like - number of layers, number of units, type of activation function & performance metric - which play an essential part in the performance of the resulting model.

Convolutional neural networks (CNNs) are neural networks with one or more convolutional layers primarily utilised for image processing, classification, segmentation, and other auto-correlated data.

Convolution is the process of sliding a filter over an input signal. This quote by Dr Prasad Samarakoon - "A convolution may be regarded as "looking at a function's surroundings to make better/accurate forecasts of its output", is a good approach in understanding convolutions.

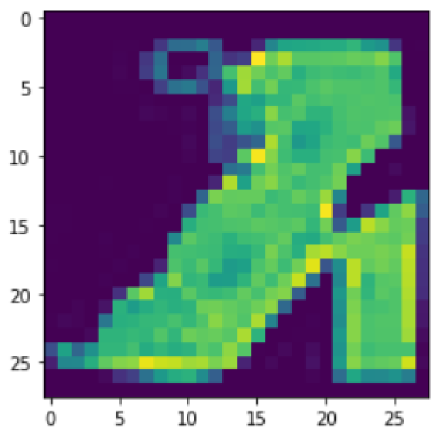
Smaller sections of an image can be more helpful than looking at the complete image at once to discover specific features.

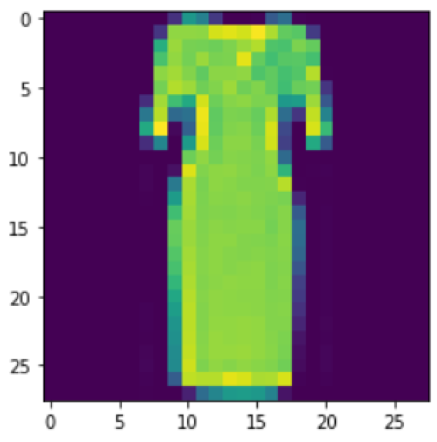
DATASET AND FRAMEWORK:

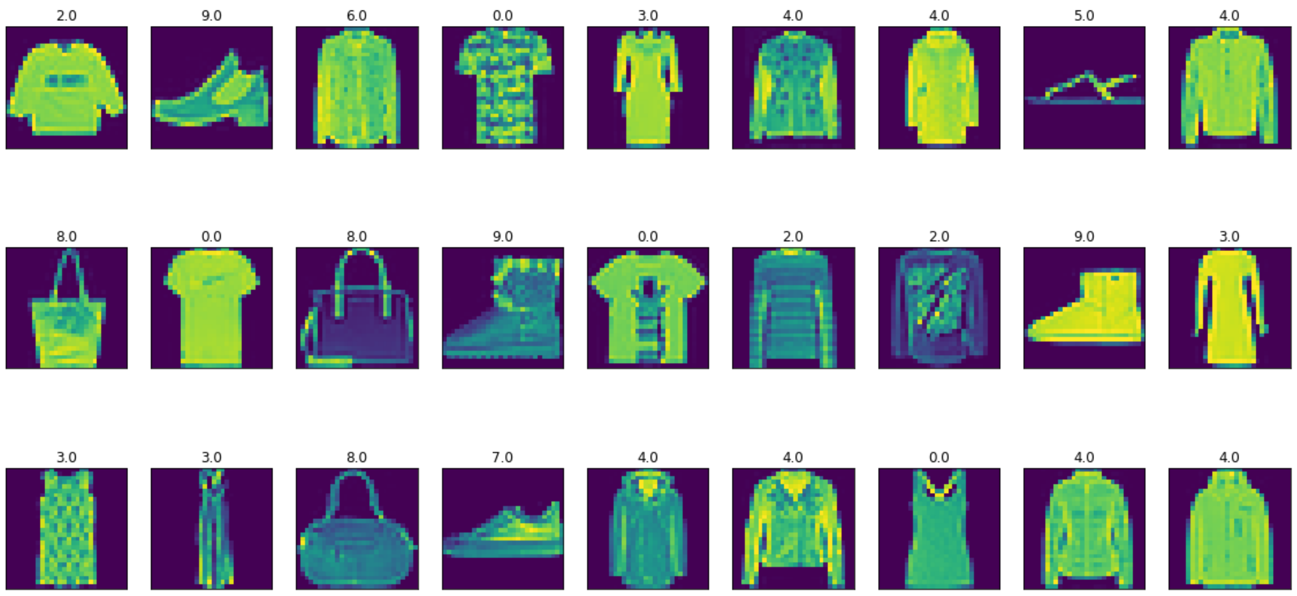
Dataset:

Fashion-MNIST is a dataset consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 pixels image, associated with a label.

The objective of the project is - to use the data set to identify (predict) different fashion products(articles) from the given images using deep learning.







Framework:

The framework of this research work is shown in Figure 1 below. This proposed methodology consists of the following steps and is performed on the Google Colaboratory server.

We will be following the below steps to solve this problem**:**

1. Importing the libraries
2. Loading the data (Train and Test data as given)
3. Visualizing the Data
4. Preprocessing
5. Building the Model
6. Evaluating the Model Performance

Tech stacks used are as follows:

1)IPYNB Jupyter Notebook

2)Python

3)Libraries used: Tensorflow, Keras, Pandas, Numpy

4)Graphing tools: Matplotlib, seaborn

5)Google Colab

FEATURE ENGINEERING:

We are dealing with an image classification dataset, where we have to classify 28 X 28-pixel size images, into one of the 10 labels. The input feature vector is the value of the pixels (784 X 1). The individual pixel value ranges from 0 to 255.

We applied Normalisation on the features, where we converted the values from 0 - 255 to 0 - 1. This is done to reduce the error in Gradient Descent and make the process stable.

Also, as the labels did not have any order or relationship between them, we applied One-hot encoding, to convert this categorical data into numeric form.

RESULT ANALYSIS:

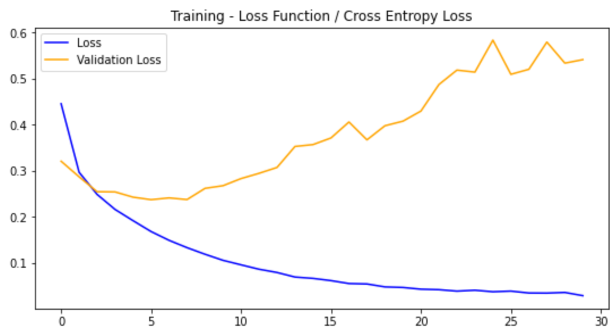
We implemented 2 models with varying numbers of hidden units.

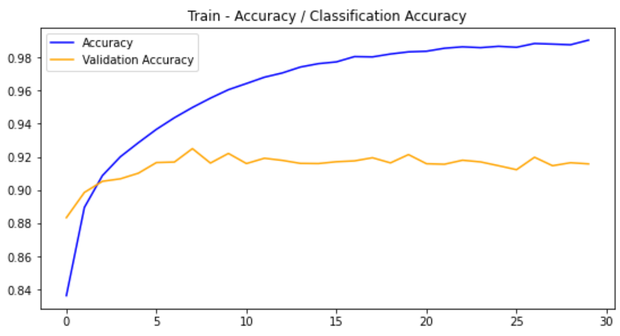
HYPERPARAMETERS -

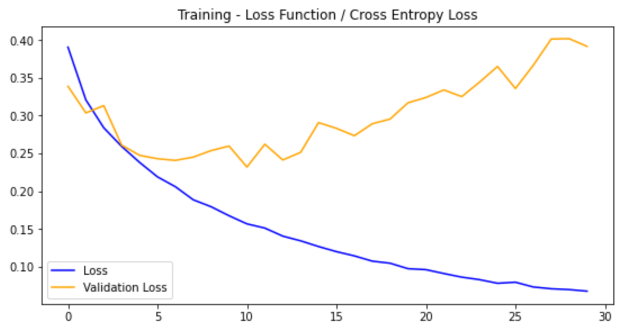
Learning Rate- Instead of finding a constant learning rate, we used **Adam optimizer,** which uses weighted averages to reach minima at a faster pace, which is generally better than a single learning rate.

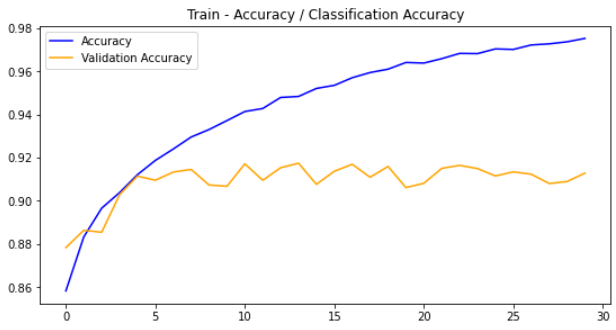
Side by side comparison -

Hyperparameter	Model 1	Model 2
Number of Convolutional layers	2	2
Number of ANN hidden layers	2	2
Output layer	1	1
Units per Convolutional layer	64	32
Units per ANN Layer	128	64
Units in the Output layer	10	10
Activation Function in Convolutional Layers	ReLU	ReLU
Activation Function in Hidden Layers	ReLU	ReLU
Activation Function in Output Layers	Softmax	Softmax
Loss Function used	Sparse Categorical Cross entropy	Sparse Categorical Cross entropy
Optimizer used	Adam Optimizer	Adam Optimizer
Testing accuracy	91.58 %	90.89 %
Training accuracy	99.04 %	97.36 %









Epochs:

An epoch indicates the number of passes of the entire training dataset the CNN algorithm has completed. As the number of epochs increases there is a clear increase in testing loss so overfitting is evident.

Number of filters:

CNN uses learned filters to convolve the feature maps from the previous layer. Filters are two-dimensional weights and these weights have a spatial relationship with each other. In this image classification model we have used 32 and 64 convolutions filters. As we move from 32 to 64 there is very slight change in testing accuracy. This is due to the fact that when we use a higher number of filters in CNN it is able to elaborate more from the extracted information from the input data.

THANK YOU