Image Classification on Fashion-MNIST dataset

Manente Alessandro Rasetta Adriano Rossetto Alberto

Abstract

Image Classification is one of the most popular tasks in Machine Learning. The aim of this project is to investigate different classification models and apply them on the Fashion-MNIST dataset.

1. Introduction

Image classification is one of the most fundamental problems in Machine Learning. It is the core foundation for bigger problems such as Computer Vision. There are many classification models that can be used for this task; however, it is important to fully understand the concepts of each model, and how they perform on different dataset. Our work concerned the application of two traditional machine learning model (Logistic Regression and K-Nearest Neighbor) and a deep neural network model (Convolutional Neural Network).

2. Dataset

Fashion MNIST is a dataset of Zalando's article images consisting of a training set of $60,000$ examples and a test set of $10,000$ examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. Each classes have the same number of images, so the dataset is balanced. Moreover we decided to scale the data to be in $[0, 1]$ dividing every entry of the matrix by 255.

3. Method

3.1. Logistic Regression

Given the nature of the task, the most immediate model to implement is logistic regression model. It is used to model the probability of a binary event: pass/fail, win/lose or healthy/sick. Nevertheless, it can be extended to model several classes of events. It is a model that is used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables. To determine the parameters this model uses the maximum likelihood, and its cost function is the cross-entropy error function for the multiclass classification problem.
We implement it with a simple neural network using the keras library. To select the most appropriate model, we mainly focused on three hyperparameters: learning rate, regularizer L1 and optimizer.

3.2. K-Nearest neighbor

K-Nearest Neighbor is a supervised learning algorithm, in particular is a non-parametric method used for classification. The k-NN classifies unknown data point by assigning to it the most frequent class among the k closest training points. The idea is to find the k neighbor with the minimum distances between the testing point and all the training points. It assumes that similar objects are close to each other.
In order to identify the k nearest object, a distance metric has to be defined. There are many distance metrics that can be used, so this is the first hyperparameter that we have to choose in order to find the best distance which represents better the points. The second one is the value of k. To handle this problem we perform cross validation with the purpose of finding the combination of values that gives the best performance on the validation set, and then finally evaluate our classifier on the test set.

3.3. Convolutional Neural Network

Convolutional Neural Network (from here on out CNN) are the standard type of Neural network used to elaborate image data. This is because, given the fact that images are structured inputs, it is useful to utilise a Neural Network able to encode certain local properties into the architecture. It is done through: the sparsity and parameter sharing of the Convolutional Layers, hence leading to greater efficiency; utilization of Pooling layers which are responsible to decrease the dimension of the problem and improve regularity of the system.
In the contest of multiclass classification the CNN was adapted such that for each sample it would output a probability vector of length equal to the number of classes and such that each component is in $[0, 1]$ and the sum of all components is equal to 1. Hence the predicted class is the one

associated with the highest value in the vector.

4. Experiments

4.1. Logistic regression

Initially a test model was implemented to find an appropriate range for each hyperparameter of interest. Once identified, a cross-validation search grid was implemented on the whole dataset and considering the values of the hyperparameters which previously provided encouraging results:

Hyperparameter	Set of values
Learning rate	0.001, 0.005, 0.01
L1 regularizer	0.00001, 0.0001, 0.0005, 0.001
Optimizer	"Adam", "Adamax", "SGD"

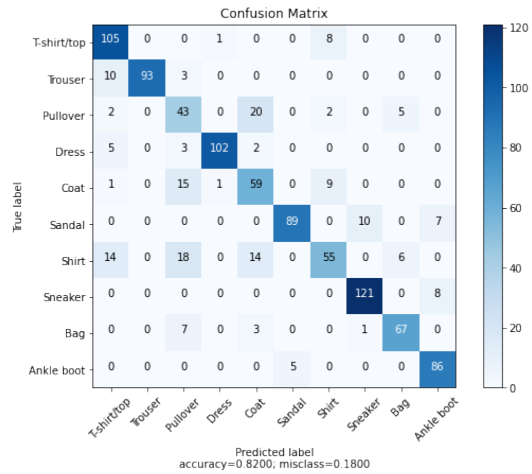
The grid search results identified the following parameters: L1 regularize = 0.0005, learning rate = 0.005 and optimizer = $'Adamax'$. Finally, the model was evaluated on the test set and an accuracy of 83.5% was achieved.

In short, given the difficulty of the task, the choice to apply logistic regression can be seen as an exploratory attempt to grasp the real difficulty of the task. From the various tests carried out it was possible to experiment that the high sample size does not greatly improve the performance of a simple model. Therefore, although the model has achieved satisfactory results considering its relative simplicity, it is necessary to face the task with different or more complex approaches.

4.2. K-Nearest neighbor

During the first experiments applying the K-Nearest Neighbor on the entire dataset, we highlighted the long running time of the classification procedure. For this reason, we decided to use only a random sampling for both the training and test set (6000 training images and 1000 test images).
With the aim to find the best combination of hyperparameters ($k \in [1, 10]$ and $metric \in ['euclidean', 'manhattan']$) we perform a grid search cross validation. The result we obtain is that the k-NN performs better with k = 3 and the manhattan distance. After that, we evaluate the model on the test set and it reaches the 82.0% of accuracy. As we can observe from the confusion matrix, the classification of different object which are quite similar to each other ("shirt," "t-shirt," "pullover," "coat"), is affected by a large error value.

This experiment shows us that k-NN algorithm is not efficient when the size of the problem is large for both the dimension of the number of samples and the number of features. The grid search procedure required about 16 minutes. Moreover, when the classes are similar to each other, this method doesn't provide a good classification.



4.3. Convolutional neural network

After testing the performances of k-NN and Logistic Regression it was decided to use a CNN, implemented using keras, to try and improve the accuracy. This was easily obtained by our first simple model with only 4 convolutional layers, 4 maxpool layers and 1 dense layer.
Then the approach was to try and increase the depth, for which the data augmentation technique was used through the keras package, and it performed this transformations:

- random horizontal shift of the image
- random vertical shift of the image
- random horizontal flip of the image.

Then the final model had the following layer structure:

In the end, analysing the most misclassified classes it was decided to build a minor model dedicated to the binary classification of the classes "shirt" and "t-shirt", through the use of, again, a CNN. Once the average misclassified vector of such classes was identified, the predictions given by the model above in those cases where substituted by the one of the binary classification model, with a minor improvement in accuracy.

Layer (size)	Output Shape	Param
conv2d (32,3,3)	multiple	320
batch normalization	multiple	128
conv2d(32,3,3)	multiple	9248
batch normalization	multiple	128
max pooling2d (2,2)	multiple	0
dropout (0.2)	multiple	0
conv2d (128,3,3)	multiple	36992
batch normalization	multiple	512
conv2d (128,3,3)	multiple	147584
batch normalization	multiple	512
max pooling2d (2,2)	multiple	0
dropout (0.2)	multiple	0
conv2d (512,3,3)	multiple	590336
batch normalization	multiple	2048
conv2d (512,3,3)	multiple	2359808
batch normalization	multiple	2048
max pooling2d (2,2)	multiple	0
dropout (0.2)	multiple	0
flatten (Flatten)	multiple	0
dense (256)	multiple	1179904
batch normalization	multiple	1024
dense (64)	multiple	16448
batch normalization	multiple	256
dense (10)	multiple	650

Table 1. Total params: 4,347,946.

4.4. Results

Considering human performance (83.5%) as a basis for a comparison, logistic regression and k-nn provide very similar results: 83.5% and 81.4% respectively. Despite the difficulty of the task, we can consider them as valid models. The CNN model, on the other hand, achieves 12-13% higher performance, achieving an accuracy of 94.36%. Given the marked improvement, it seems clear that complex models are needed for a complex task. Also, it is worth noticing that using two CNN, increased marginally the accuracy to 94.38%, which is a small improvement, yet if it was possible to increase the accuracy of a CNN with the task to execute binary classification on "Shirt" vs "T-Shirt", it would be possible to increase the results even more and by an higher margin.

Model	Accuracy
Logistic Regression	83.5%
K-NN	82.0%
Simple CNN	92.83%
Complex CNN	94.36%
Double CNN	94.38%