Multiclass Classification and Object Identification of Fashion MNIST using Convolutional Neural Network

Md Shahidullah Kawsar
kawsar@mail.usf.edu
University of South Florida, Tampa, USA

Abstract- As a yardstick of machine learning algorithms, 'Fashion MNIST' is one of the most popular datasets among the Artificial Intelligence (AI) and Data Science community. In this paper, I performed multiclass classification on Fashion MNIST dataset and object identification of different clothes. The classification problem was very challenging due to the diversity of similar patterns on multiple classes which affected the performance of the classifier. For example, a shirt, a pullover and a dress may have long-sleeve of similar patterns or a sneaker and an ankle boot may have leis of similar patterns. For the classification problem, I have achieved nearly 91% accuracy by using the convolutional neural network which has outperformed other popular eight types of machine learning approaches.

Introduction:

Machine learning and deep learning studies the design of algorithms that can learn. Deep learning is a subfield of machine learning that is inspired by artificial neural networks, which in turn are inspired by biological neural networks. The convolutional neural network (CNN) is a deep feed-forward artificial neural network. Feed-forward means information flows right through the model. There are no feedback connections fed back into itself from the outputs of the model.

CNNs are especially motivated by the biological visual cortex. The cortex has small regions of cells that are sensitive to the specific areas of the visual field. In 1962, the researchers revealed that some individual neurons in the brain triggered only in the existence of edges of a particular alignment like vertical or horizontal edges. For example, some neurons activated when exposed to vertical sides and some when exposed to a horizontal edge. Interestingly, all of these neurons are well ordered in a columnar fashion and that together they construct visual sensitivity. This idea of specialized components inside of a system having specific tasks is utilized in CNNs.

CNNs have been one of the most dominant innovations in the field of computer vision. CNNs can produce state-of-the-art results and perform a lot better than conventional computer vision. Convolutional neural networks have been successful applied in numerous real-life applications, such as: Image classification, object detection, segmentation, face recognition, self-driving cars that utilize CNN based vision systems, classification of crystal structures etc. In 2012, Alex Krizhevsky won that year's ImageNet Competition by reducing the classification error from 26% to 15% using convolutional neural networks.

Convolutional Neural Network (CNN):

The convolution layer is used for filtering which computes the output of neurons which are connected to receptive regions in the input, each calculates the dot product between their weights and a small receptive region to which they are connected to in the input. Each computation is performed for feature extraction map from the input image. For example, an image can be represented as a 4x4 matrix of different values. Now multiply these values with a matrix of 3x3 kernel size which gives a single number that represents all the values in that window of the images. As the kernel travels over the image, it’s looking for patterns in that section of the image using convolution.

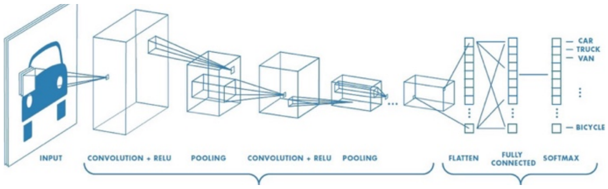


Fig 1: Architecture of the Convolutional Neural Network

Fig 1 shows that, an image has been fed as an input to the network, which goes through multiple convolutions, subsampling, a fully connected layer and finally produce the outputs.

In order to reduce the dimension of the input, one of the methods of subsampling is max pooling which aids to reduce overfitting. Max pooling selects the highest pixel value from a region depending on its size. Suppose, a 2x2 max pooling layer will choose the highest pixel intensity value from a 2x2 region. By using a kernel and travel it over the image, the max pooling layer mechanism is similar to the convolution layer. The sole distinction is the function which is applied to the kernel and the image kernel isn't linear.

The goal of the fully connected layer is to flatten the high-level features that are learnt by the convolutional layers and combine all the features. The flattened output is connected to the output layer which can use different activation functions to predict the input class label.

Support Vector Machine (SVM)
A support vector machine (SVM) is another popular supervised machine learning classification algorithm. A standard machine learning algorithm tries to find a boundary that divides the data in such a way in which the misclassification rate can be minimized. An SVM is not only determining a decision boundary between the possible outputs, it can also find the most optimal decision boundary. The difference between SVM and the other classification algorithms is SVM selects the decision boundary which maximizes the distance from the nearest data points of all the classes.

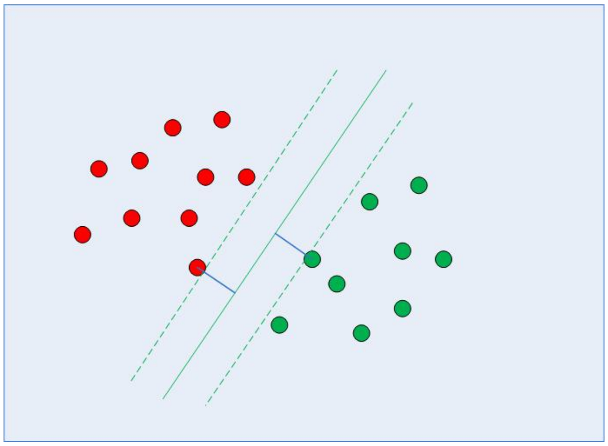


Fig 2: Decision Boundary in Support Vectors

The most optimal decision boundary is the one which has highest margin from the closest points of all the classes. The closest points from the decision boundary which maximize the gap between the decision boundary and the points are called support vectors as shown in Fig 2. In SVM, the decision boundary is called the maximum margin classifier or the maximum margin hyperplane.

The application of SVM is to accurately classify the unseen data such as hand-written character recognition, face detection by classifying the parts of the image as a face and non-face, text and hyper-text categorization, in query refinement schemes, biometrics e.g. protein fold and remote homology detection etc.

Linear Discriminant Analysis (LDA):
Both Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are linear transformation techniques. Nevertheless, LDA is a supervised dimensionality reduction technique while PCA is an unsupervised machine learning approach.

PCA determines the directions of the maximum variance from the dataset. In a dataset with lots of features, there are many features which are nearly duplicate of the other features or the features may have a high correlation with the other features. These redundant features can be ignored. The role of PCA is to determine these highly correlated features and to separate a new feature set where there is least correlation or maximum variance between the features. PCA doesn't consider the output labels into account because the variance between the features doesn't depend on the output.

On the other hand, LDA reduces the dimensions of the features by preserving the information that distinguishes output classes. LDA efforts to determine a decision boundary around each cluster of a class. Then the data points are projected to new dimensions in a way where the clusters are as separate from each other as possible. The individual elements inside a cluster are very close to the centroid of the cluster. The ranking of the new dimensions is done based on the ability to maximize the distance between the clusters and minimize the distance between the data points inside a cluster and their centroids. That’s how these new dimensions form the linear discriminants of the features.

The K-nearest neighbors (KNN):
K-nearest neighbors (KNN) is one of the simplest of all the supervised machine learning algorithms and KNN is also a non-parametric and lazy supervised machine learning algorithm because it doesn't have a dedicated training phase and requires no training prior to making real time predictions. During classifying a new instance, KNN uses all of the data for training. Non-parametric learning algorithm means it doesn't assume anything about the underlying data which is a particularly very useful feature as the real-world data doesn't follow any theoretical assumption e.g. linear-separability, uniform distribution, etc.

KNN simply calculates the distance from a new data point to all other training data points. Any type of distance measurement method can be used in KNN such as Euclidean, Manhattan etc. Then it picks the K-nearest data points, where K can be any integer. The final step of the KNN algorithm is to allocate new point to the class to which majority of the K nearest points belong.

Besides the advantages, KNN has some disadvantages also. The KNN algorithm doesn't work well with high dimensional data because it’s getting difficult for the algorithm to calculate distance in each dimension.

Besides, in large datasets, the cost of calculating distance between new point and each existing point becomes expensive. Finally, with categorical features, the KNN doesn't work well as it's hard to find the distance between dimensions.

Random forest:

Random forest is another type of supervised machine learning algorithm. It is based on ensemble learning means that it can join different types of algorithms or same algorithm multiple times to generate a more powerful prediction model. This algorithm combines multiple decision trees, thus results in a forest of trees, and hence the name Random Forest.

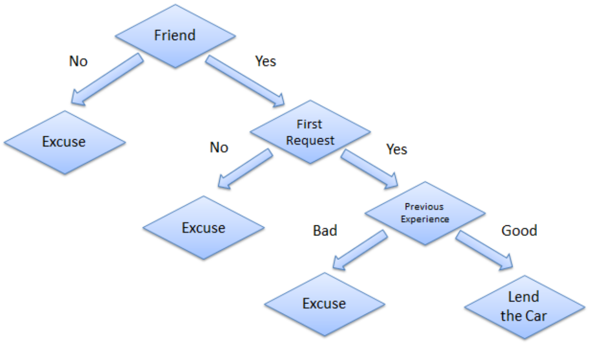
The basic idea of the random forest algorithm is to choose N random records from the dataset and then based on these N records, build a decision tree. Finally, select the number of trees and continue steps 1 and 2 again. In classification problem, each tree in the forest predicts the category to which the new record fits. Lastly, the new record is allocated to the class that gains the highest vote.

Since, this approach has multiple trees and each tree is trained on a subset of data, hence the overall biasedness of the algorithm is decreased, and the algorithm is very stable. Addition of a new data point can't affect the overall algorithm much because new data may affect one tree and it is very difficult to affect all the trees. Besides, the random forest algorithm performs well on both categorical and numerical features. Due to the large number of decision trees combined together, random forests require much more computational resources and time to train than other algorithms.

Decision Tree:

The decision tree algorithm creates a node for each feature in the dataset and the most important feature is assigned at the root node. Evaluation starts at the root node and go down the tree by following the consequent node that agrees the condition or the decision. The process stops at a leaf node which predicts the outcome of the decision tree. The perception behind the decision tree algorithm is very simple but very powerful.

For example, person one asks person two to lend him his laptop for a day and person two have to make a decision whether or not to lend him the laptop. There are some factors that may affect the decision of the person two such as is this person one a close friend? If the answer is no, then decline the request. Otherwise proceed to next step. Is the person one asking for the laptop for the first time? If so, proceed to next step. If person two has good sharing experience with person one, lend him the laptop. Otherwise, decline the request.



Decision trees require comparatively less effort to train the algorithm and it can be used to classify non-linearly separable data and predict both continuous and discrete values.

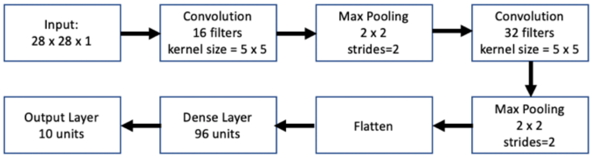


Fig CNN model 1 architecture

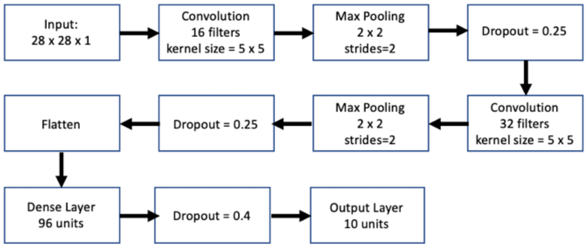


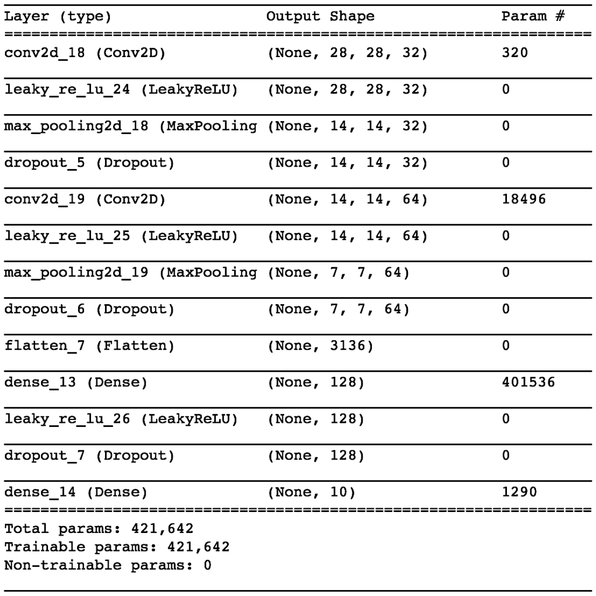
Fig CNN model 2 architecture (with dropout)

Hinge loss:

The minimization of logarithmic loss leads to well-behaved probabilistic outputs. On the other hand, hinge loss leads to some sparsity on the dual, but it doesn't assist at probability estimation. Rather, hinge loss punishes misclassifications that's why it's so useful to determine margins. The minimization of hinge loss comes with reducing margin misclassifications. In other words, logarithmic loss aims at better probability estimation at the cost of accuracy and hinge loss aims to better accuracy and some sparsity at the cost of reduced sensitivity regarding probabilities.

The cross entropy between two probability distributions c and d over the same underlying set of events measures the average number of bits required to recognize an event drawn from the set if a coding scheme used for the set is optimized for an estimated probability distribution d, rather than the true distribution d.

Table I. Model Summary



```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_18 (Conv2D)           (None, 28, 28, 32)        320
_________________________________________________________________
leaky_re_lu_24 (LeakyReLU)   (None, 28, 28, 32)        0
_________________________________________________________________
max_pooling2d_18 (MaxPooling (None, 14, 14, 32)        0
_________________________________________________________________
dropout_5 (Dropout)          (None, 14, 14, 32)        0
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 14, 14, 64)        18496
_________________________________________________________________
leaky_re_lu_25 (LeakyReLU)   (None, 14, 14, 64)        0
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 7, 7, 64)          0
_________________________________________________________________
dropout_6 (Dropout)          (None, 7, 7, 64)          0
_________________________________________________________________
flatten_7 (Flatten)          (None, 3136)              0
_________________________________________________________________
dense_13 (Dense)             (None, 128)               401536
_________________________________________________________________
leaky_re_lu_26 (LeakyReLU)   (None, 128)               0
_________________________________________________________________
dropout_7 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_14 (Dense)             (None, 10)                1290
=================================================================
Total params: 421,642
Trainable params: 421,642
Non-trainable params: 0
_________________________________________________________________
```

Dataset Information:

The dataset 'Fashion MNIST' consists of 70,000 image examples. Among them, 85.7% are the training data and the remaining 14.3% are the test data. Each example is a 28x28 grayscale image, linked with a label of 10 classes such as T-shirt/top (0), Trouser (1), Pullover (2), Dress (3), Coat (4), Sandal (5), Shirt (6), Sneaker (7), Bag (8), and Ankle boot (9). The dataset has 784 input features as pixel numbers from 0 (bright) to 255 (dark) and one output feature which is the class label. There are 7,000 images per category.

Other Classification Methods:

I would like to perform and compare these classification methods:

2. Naive Bayes
3. Logistic regression
4. K-Nearest Neighbor (KNN)
5. Linear Discriminant Analysis (LDA)
6. Decision Tree
7. Random Forest
8. Support Vector Machine

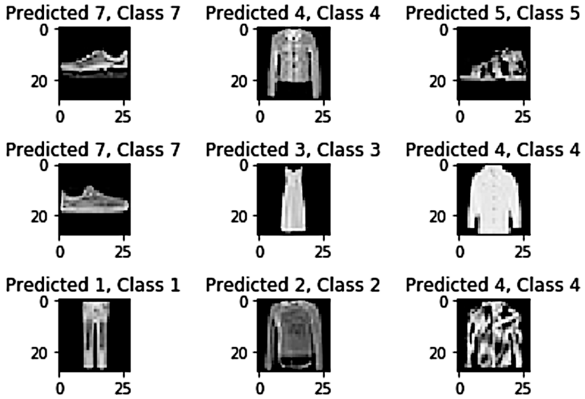


Fig 3. Correct classification

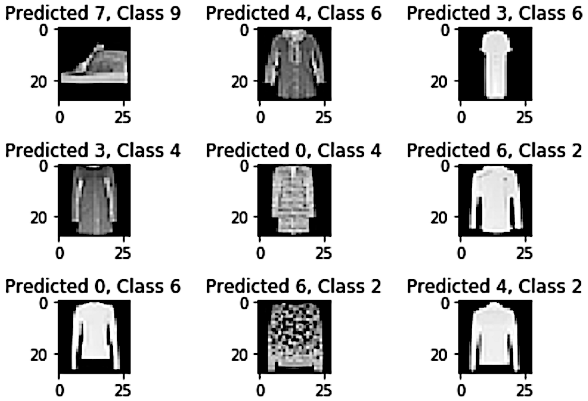


Fig 4. Incorrect classification

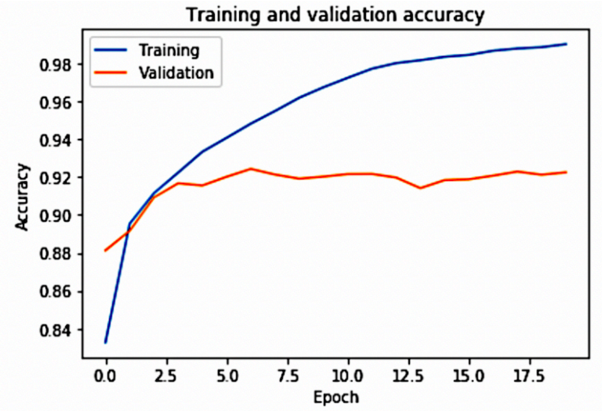


Fig 5. Training and validation accuracy vs number of epochs in the convolutional network without dropout

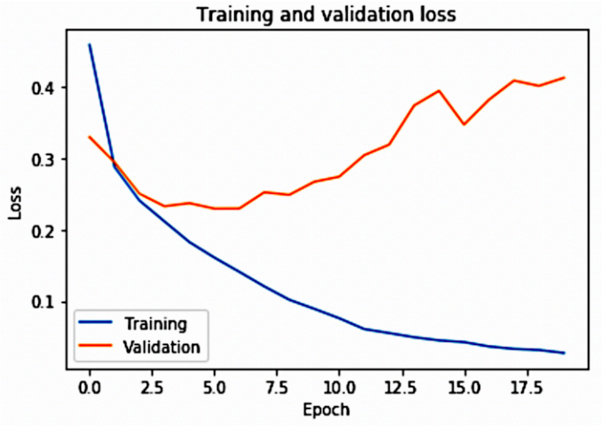
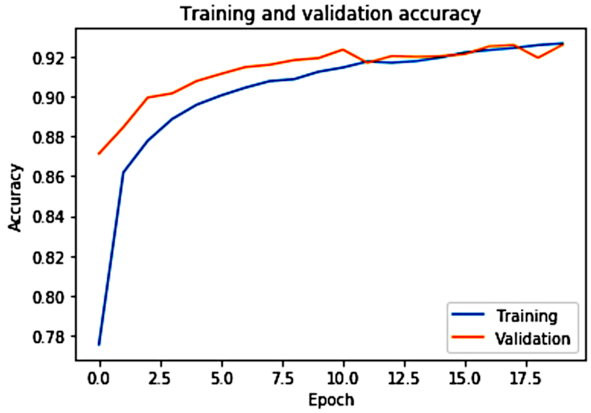


Fig 6. Training and validation loss vs number of epochs in the convolutional network without dropout



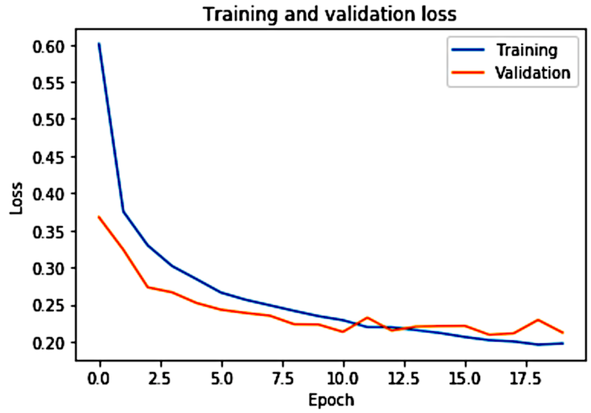


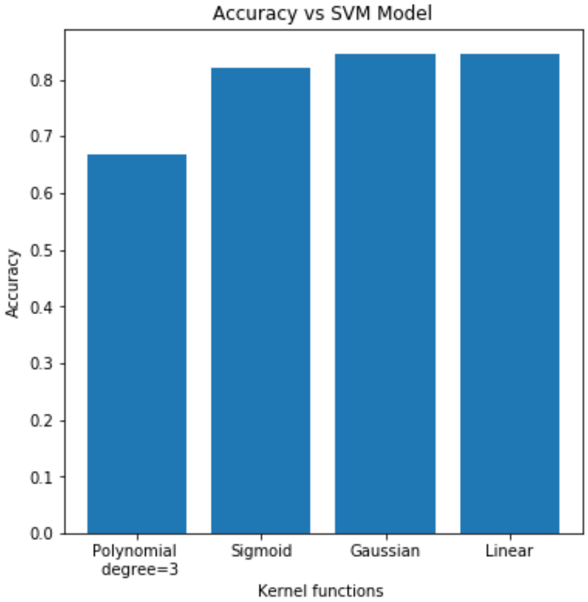
Fig 7. Training and validation loss vs number of epochs in the convolutional network with dropout

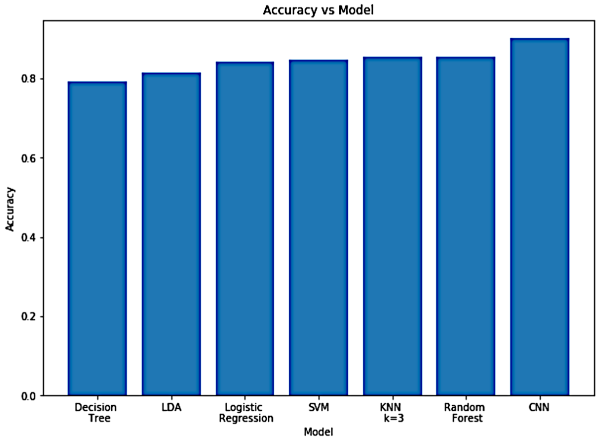
Fig 8. Training and validation loss vs number of epochs in the convolutional network with dropout

```
              precision    recall  f1-score   support

     Class 0       0.76      0.93      0.84      1000
     Class 1       1.00      0.98      0.99      1000
     Class 2       0.86      0.91      0.88      1000
     Class 3       0.93      0.91      0.92      1000
     Class 4       0.91      0.84      0.88      1000
     Class 5       0.99      0.98      0.99      1000
     Class 6       0.83      0.71      0.76      1000
     Class 7       0.95      0.98      0.97      1000
     Class 8       0.98      0.98      0.98      1000
     Class 9       0.98      0.97      0.97      1000

   micro avg       0.92      0.92      0.92     10000
   macro avg       0.92      0.92      0.92     10000
weighted avg       0.92      0.92      0.92     10000
```

Table II. Classification report of CNN

The details of the misclassified classes can be identified from the classification report. From Table II, for which class the model performed bad can be figured out of the given ten classes. For example, the classifier has lowest precision at the class 0, lowest recall at class 6 and lowest f1-score class 6. In this observation, the classifier is underperforming for class 6 regarding both precision and recall and f1-score. The classifier has highest precision at the class 1, several classes have highest 0.98 recall. The highest f1-score of 0.99 has been found at the class 1 and 5.





Conclusion:

I believe this image dataset will be a good practice for multiclass classification and object identification. I will especially try convolutional neural networks (CNN) as in deep artificial neural networks, CNNs are the most frequently applied to visual image dataset for object recognition and CNNs required less prior knowledge compared to other image classification algorithms.

Reference:

[1] Fashion MNIST dataset, https://research.zalando.com/welcome/mission/research-projects/fashion-mnist/