# Image Classification using Fashion MNIST dataset

### Why neural networks overperform other models

Rade Nježić

December 20, 2019

**Abstract**

In this short report, we represent results obtained by performing various classification algorithms on Fashion MNIST dataset[XRV17]. We train a simple convolutional neural network and analyze its performance, and analyze why other classification algorithms are less accurate than it, and what can we learn from that.

## Dataset

Fashion MNIST was introduced in August 2017, by research lab at Zalando Fashion. Its goal is to serve as a new benchmark for testing machine learning algorithms, as MNIST became too easy and overused. While MNIST consists of handwritten digits, Fashion MNIST is made of images of 10 different clothing objects. Each image has the following properties:

- Its size is $28 \times 28$ pixels.
- Represented in grayscale, with values ranging from 0 to 255.
- Vacant space represented by black colour having value 0.

We distinguish between the following clothing items:

- T-shirt/Top
- Trousers
- Pullover
- Dress
- Coat
- Sandal
- Shirt
- Sneaker
- Bag
- Ankle Boot

Dataset consists of 70000 images, of which :

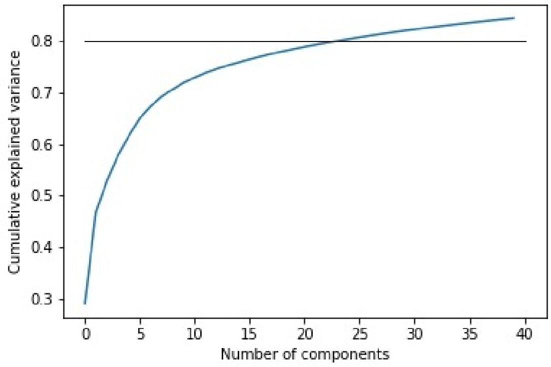
- 60000 make the training set.
- 10000 make the testing set.

Both sets have evenly distributed labels. Sample of images is provided in the figure below.

## Exploratory Data Analysis

The dataset was imported directly from Keras library, so there was almost no need for data processing. The images were converted to 1D array and scaled, to get more reliable results.

**Principal Component Analysis** As our images have 784 dimensions, and lots of vacant space, it is useful to use some form of dimensionality reduction. We used the PCA and extracted the first 40 PCs. As we will see from the plot, the 80% of variance is captured by around 25 PCs, and that is the number of PCs we used in our research. They were used in logistic regression, random forest, and support vector machines, and those classifiers were applied to them.



We also notice that the first eight PCs carry most variance.

## Employing classifiers

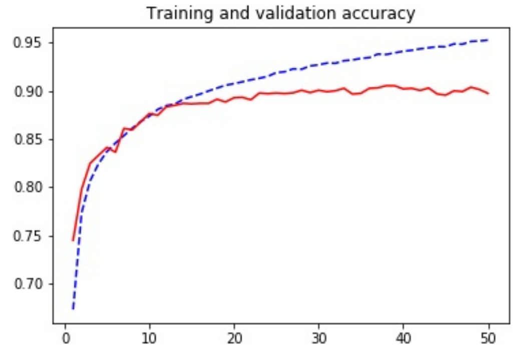
The problems of image classification represent just a small subset of classification problems. The most practical image classification methods are deep learning algorithms, one of which is convolutional neural networks. The rest of the employed methods will be a small collection of common classification methods.

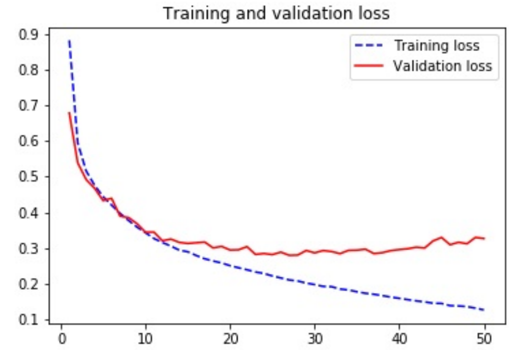
As class labels are evenly distributed, with no misclassification penalties, we evaluated our methods using accuracy metric.

**Convolutional Neural Network (CNN)** The first method we employed was CNN. As the images were in grayscale, we applied only one channel. We selected the following layers:

- Two convolutional layers with 32 and 64 filters, $3 \times 3$ kernel size, and relu activation.
- The polling layers were chosen to operate of tiles size $2 \times 2$, and to select the maximal element in them.
- Two sets of dense layers, with the first one selecting 128 features, having relu and softmax activation.

For loss function, we chose categorical cross-entropy. To avoid overfitting, we have chosen 9400 images from the training set to serve as a validation set for our parameters. We used the novel optimizer adam, which improves over standard gradient descent methods and uses a different learning rate for each parameter and the batch size equal to 64. Model was trained in 50 epochs. The accuracy and loss values are presented in the graphs below.





We see that the algorithm converged after 15 epochs, that it is not overtrained, so we tested it. The obtained testing accuracy was equal to 89%, which is the best result obtained.

Before proceeding further, let's explain what have the convolutional layers done. An intuitive explanation is that the first layer was capturing straight lines and the second one curves. On both layers we applied max pooling, which selects the maximal value in the kernel, separating clothing parts from vacant space. In that way, we capture the representative nature of data. In other, neural networks perform feature selection by themselves.

After the last pooling layer, we get an artificial neural network. Because we are dealing with the classification problem, the final layer uses softmax activation to get class probabilities. As class probabilities follow a certain distribution, cross-entropy indicates the distance from networks preferred distribution.

**Multinomial Logistic Regression** As pixel values are categorical variables, we can apply Multinomial Logistic Regression. We apply it one vs rest fashion, training ten binary Logistic Regression classifiers, that we will use to select items. As we were working with principal components, we have used Ridge regularization. We get 80% accuracy on this algorithm, which is not bad. But, we have to take into account that this algorithm worked on grayscale images which are centred and normally rotated, with lots of vacant space, it may not work for more complex images.

**Nearest neighbors and centroid algorithms** We selected two different neighboring algorithms:

- K-nearest neighbours
- Nearest Centroid

Nearest centroid data finds a mean value for elements of each class and assigns test element to the class to which the nearest centroid is assigned. Both algorithms were implemented with respect to $L^1$ and $L^2$ distance. The accuracy for both nearest algorithms was 85%, while the centroid algorithms had the accuracy of 67%.

Similar accuracies inform us that that images belonging to the same class tend to occupy similar places and have similar pixel intensities. While nearest neighbours obtained good results, they are still worse than CNNs, because they don't operate in neighbourhood of each specific feature, while centroids fail more because, in addition, they don't distinguish between similar-looking objects (e.g. pullover vs t-shirt/top).

**Random Forest** To select the best parameters for estimation, we performed grid search[1] with root squared and the full number of features (bagging), Gini and entropy criterion, and with trees having maximal depth 5 and 6. Grid search suggested that we should use root squared number of features with entropy criterion (both expected for classification task). However, obtained accuracy was only equal to 77%, implying that random forest is not a particularly good method for this task. The reason it failed is that PCs don't represent the rectangular partition that an image can have, on which random forests operate. The same reasoning applies to the full-size images as well, as the trees would be too deep and lose interpretability.

**Support Vector Machines (SVM)** We applied SVM this using radial and polynomial kernel. The radial kernel has 77% accuracy, while the polynomial kernel fails to be successful and gets 46% accuracy. However, SVMs are still highly useful for binary classifications problem. Their biggest caveat is that they require feature selection, which brings accuracy down, and without it, they can be computationally expensive. As they apply multiclass classification in a one-vs-rest fashion, making it harder to efficiently create separating hyperplane, they may lose accuracy when dealing with non-binary classification.

## Conclusions

In this short text, we applied various classification methods on an image classification problem. We have explained why the CNNs are the best method we can employ, and why do the other ones fail. The reason why CNNs are the most practical and usually, the most accurate method are

- They can transfer learning through layers, saving inferences, and making new ones on subsequent layers.
- No need for feature extraction before performing algorithm, it is done during training.
- It recognizes important features.

However, they also have their caveats. They are known to fail on images that are rotated and scaled differently, which is not the case here, as the data was already pre-processed. And, although the other methods fail to give that good results on this dataset, they are still used for other tasks related to images (sharpening, smoothing etc.).

## References

[XRV17] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms.* Aug. 28, 2017. arXiv: `cs.LG/1708.07747 [cs.LG]`.

[1] Putting different parameter values, and adding a validation set in their training to select the best one. Kind of a brute force hyperparameter optimization.