Convolutional Neural Networks Application to Fashion-MNIST Classification

CSCI8363 Final Project

Yuting Sun

Monday 11^{th} December, 2017

Contents

Introduction

- Convolutional neural networks perform very well in image classification.
  - The accuracy of classifying MNIST handwritten digits data can be higher than 99.7% as reported in Ciregan et al.(2012).
  - MNIST is not a difficult task and this dataset has been overused.
- Fashion-MNIST dataset, created by Zalando is more challenging.

Data Description

- 60,000 training data and 10,000 testing data
- 28×28 grayscale image, associated with a label from these 10 classes

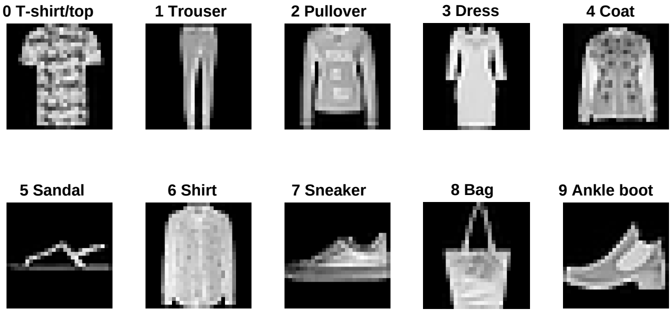


Figure: Lables

Motivation

- Apply convolutional neural networks to classify the fashion images in Fashion-MNIST.
- Find a convolutional neural network with related simpler architecture which can provide us with a competitive accuracy.
- Explore how the number of feature maps and fully connected layer output size affect the accuracy of prediction.
- Predict the testing data using the "best" CNN and analyze the prediction result.

Related Works

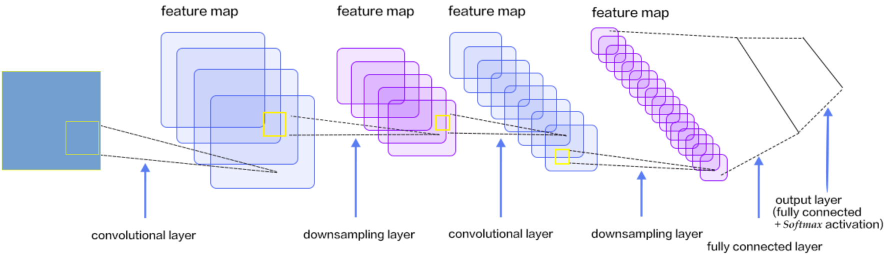
- Paper by Zalando researchers, Xiao et al. (2017)
  - The best model: Support Vector Classifier with regularization parameter C=10 and RBF kernel.
  - Testing accuracy: 89.70%.
- GitHub

CNNs	**Accuracy(%)**
32C3-64C3-2P2-FC128-FC10 with dropout	87.6%
32C5-2P2-64C5-2P2-FC1024-FC10 with dropout	91.6%
32C3-2P2-64C3-2P2-128FC-FC10	91.9%
32C3-2P2-32C3-2P2-128FC-FC10 with dropout	92.2%
32C5-2P2-64C5-2P2-128C1-2P2-FC1024-FC512-FC10 with dropout	92.6%
64C5-2P2-512C5-2P2-128FC-64FC-10FC with dropout	93.2%
AlexNet	89.9%

Table: CNNs Comparison

- Human performance is only 83.5%.

Architecture of Convolutional Neural Networks



N_1(Feature maps)C5 - 2P2 - N_2(Feature maps)C5 - 2P2 - FC(M) - FC(10)

- Convolutional layer: size=5, stride=1, zero padding=2
- Max-pooling layer: size=2, stride=2, zero padding=0

Architecture of Convolutional Neural Networks

Fully connected layer:

- stochastic gradient descent with momentum

$$v_{l+1} = \gamma v_l + \alpha \nabla E(\theta_l) \tag{1}$$

$$\theta_{l+1} = \theta_l - v_{l+1} \tag{2}$$

  where l is the iteration number, $0 < \gamma \leqslant 1$ is the momentum (we set momentum=0.9), $\alpha > 0$ is the learning rate (we set initial learning rate=0.01), θ is the parameter vector and $E(\theta)$ is the loss function.
- The loss function is the cross entropy function.

$$E(\theta) = -\sum_{i=1}^{n} \sum_{j=1}^{k} t_{ij} \ln y_j(x_i, \theta) \tag{3}$$

  where θ is the parameter vector, t_{ij} is the indicator that the ith sample belongs to the jth class, and $y_j(x_i, \theta)$ is the output for sample i.

Architecture of Convolutional Neural Networks

- Softmax Classification layer: The output unit activation function is the softmax function.

$$y_r(x) = \frac{\exp(a_r(x))}{\sum_{j=1}^{k} \exp(a_r(x))} \tag{4}$$

where $0 \leqslant y_r \leqslant 1$ and $\sum_{j=1}^{k} y_j = 1$, and $a_r = \ln(P(x,\theta|c_r)P(c_r))$. $P(x,\theta|c_r)$ is the conditional probability of the sample given class r, and $P(c_r)$ is the class prior probability.

- Other setup
  - Mini-batch = 128
  - validation frequency: 2 validations per epoch
  - Training Stop Criteria: Stop if validation accuracy does not improve for N (we set $N = 10$) network validations.

One Convolutional Layer Models

- Fix number of feature maps: 32 and 128
- Change fully connected layer output size: $2^5 = 32$, $2^6 = 64$, $2^7 = 128$, $2^8 = 256$, $2^9 = 512$.

Models of 32C5	Training Accuracy	Testing Accuracy
32C5-2P2-FC32-FC10	96.31%	91.17%
32C5-2P2-FC64-FC10	95.51%	91.51%
32C5-2P2-FC128-FC10	95.78%	91.18%
32C5-2P2-FC256-FC10	95.46%	90.96%
32C5-2P2-FC512-FC10	96.34%	90.94%

Table: Accuracies of 32C5 models

Models of 128C5	Training Accuracy	Testing Accuracy
128C5-2P2-FC32-FC10	94.32%	90.82%
128C5-2P2-FC64-FC10	95.17%	90.93%
128C5-2P2-FC128-FC10	96.32%	91.09%
128C5-2P2-FC256-FC10	96.59%	91.32%
128C5-2P2-FC512-FC10	97.32%	91.88%

Table: Accuracies of 128C5 models

One Convolutional Layer Models

- Fix fully connected layer output size: 64 and 512
- Change the number of feature maps: $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, $2^7 = 128$, $2^8 = 256$.

Models of FC64	Training Accuracy	Testing Accuracy
16C5-2P2-FC64-FC10	93.39%	90.02%
32C5-2P2-FC64-FC10	95.15%	91.51%
64C5-2P2-FC64-FC10	95.27%	91.32%
128C5-2P2-FC64-FC10	96.02%	91.21%
256C5-2P2-FC64-FC10	96.23%	91.00%

Table: Accuracies of FC64 models

Models of FC512	Training Accuracy	Testing Accuracy
16C5-2P2-FC512-FC10	94.33%	90.68%
32C5-2P2-FC512-FC10	96.34%	90.94%
64C5-2P2-FC512-FC10	96.32%	91.24%
128C5-2P2-FC512-FC10	97.32%	91.88%
256C5-2P2-FC512-FC10	97.89%	91.18%

Table: Accuracies of FC512 models

One Convolutional Layer Models

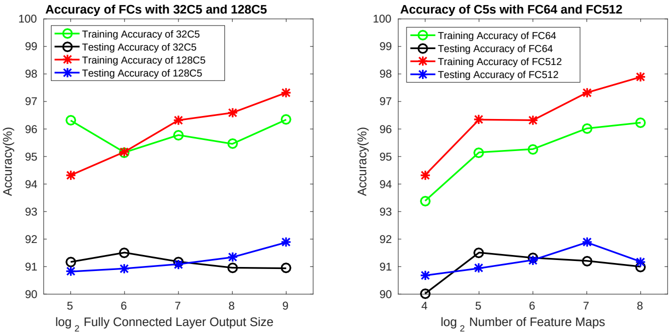


Figure: Accuracies of one convolutional layer models

- It is not necessary that more feature maps or larger fully connected output size is able to increase the testing accuracy.
- An appropriate combination of the number of feature maps and fully connected layer output size can make the CNNs perform well.

Two Convolutional Layers Models

- Fix the fully connected layer output size as 512
- Change the numbers of feature maps.

Two Convolutional Layers Models	Training Accuracy	Testing Accuracy
16C5-2P2-16C5-2P2-FC512-FC10	92.08%	89.72%
16C5-2P2-32C5-2P2-FC512-FC10	92.19%	89.93%
32C5-2P2-32C5-2P2-FC512-FC10	92.36%	89.99%
32C5-2P2-64C5-2P2-FC512-FC10	92.62%	90.26%
64C5-2P2-64C5-2P2-FC512-FC10	92.78%	90.13%
64C5-2P2-128C5-2P2-FC512-FC10	93.10%	90.24%

Table: Accuracies of FC512 models

Two Convolutional Layers Models

- Fix convolutional layers 32C5-2P2-64C5-2P2
- Change fully connected layer output size: $2^7 = 128$, $2^8 = 256$, $2^9 = 512$, $2^{10} = 1024$.

Two Convolutional Layers Models	Training Accuracy	Testing Accuracy
32C5-2P2-64C5-2P2-FC128-FC10	92.17%	90.10%
32C5-2P2-64C5-2P2-FC256-FC10	92.19%	90.49%
32C5-2P2-64C5-2P2-FC512-FC10	92.62%	90.26%
32C5-2P2-64C5-2P2-FC1024-FC10	93.15%	90.29%

Table: Accuracies of 32C5-2P2-64C5-2P2 models

- More convolutional layers do not increase the testing accuracy for this dataset.

Dropout

- Deal with overfitting
- Dropout in convolutional layer, max-pooling layer, fully connected layer or combination of them.
- For this data, we do dropout in
  - fully connected layer
  - max-pooling layer and fully connected layer

Dropout: Accuracy Tables

128C5-2P2-FC512-FC10	Training accuracy	Testing accuracy
Without dropout	97.32%	91.88%
Dropout fully connection	97.68%	92.20%
Dropout fully connection and max-pooling	98.01%	92.32%

Table: Accuracies of 128C5-2P2-FC512-FC10 with or without dropout

32C5-2P2-64C4-2P2-FC256-FC10	Training accuracy	Testing accuracy
Without dropout	92.19%	90.49%
Dropout fully connection	95.51%	92.10%
Dropout fully connection and max-pooling	96.29%	92.47%

Table: Accuracies of 32C5-2P2-64C4-2P2-FC256-FC10 with or without dropout

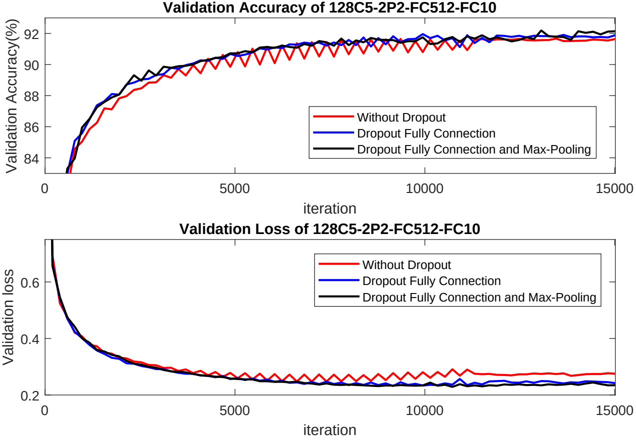


Figure: Validation Accuracy and Validation Loss of 128C5-2P2-FC512-FC10

Dropout: 32C5-2P2-64C5-2P2-FC256-FC10

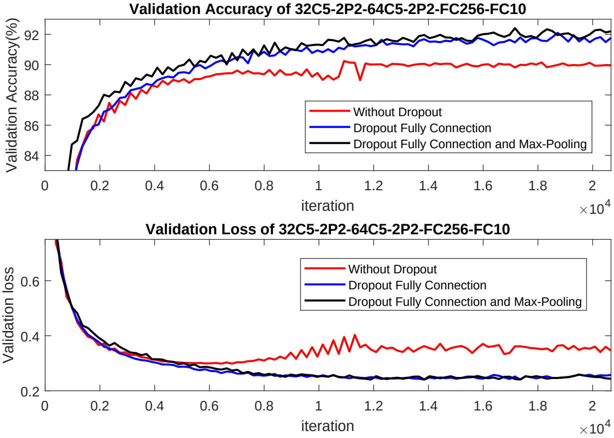


Figure: Validation Accuracy and Validation Loss of 32C5-2P2-64C5-2P2-FC256-FC10

Prediction Result

Confusion matrix:

$$\begin{bmatrix}
923 & 3 & 19 & 31 & 0 & 0 & 149 & 0 & 1 & 0 \\
0 & 990 & 0 & 6 & 1 & 0 & 3 & 0 & 0 & 0 \\
13 & 1 & 850 & 5 & 25 & 1 & 39 & 0 & 2 & 0 \\
8 & 5 & 7 & 910 & 12 & 0 & 11 & 0 & 0 & 0 \\
0 & 0 & 76 & 28 & 923 & 0 & 49 & 0 & 1 & 0 \\
2 & 0 & 0 & 1 & 0 & 980 & 0 & 7 & 0 & 1 \\
51 & 1 & 47 & 17 & 38 & 0 & 746 & 0 & 7 & 0 \\
0 & 0 & 0 & 0 & 0 & 13 & 0 & 959 & 1 & 20 \\
3 & 0 & 1 & 2 & 1 & 0 & 3 & 0 & 988 & 1 \\
0 & 0 & 0 & 0 & 0 & 6 & 0 & 34 & 0 & 978
\end{bmatrix}$$

Classes	Precision	Recall
0. T-shirt / top	0.8197	0.9230
1. Trousers	0.9900	0.9900
2. Pullover	0.9081	0.8500
3. Dress	0.9549	0.9100
4. Coat	0.8570	0.9230
5. Sandal	0.9889	0.9800
6. Shirt	0.8225	0.7460
7. Sneaker	0.9658	0.9590
8. Bag	0.9890	0.9880
9. AnkleBoot	0.9607	0.9780

Table: Precision and Recall

Conclusion and Future Works

- Conclusion
  - Find a related simpler CNN and the accuracy is 92.47%.
  - It is not necessary that large networks perform better than small ones.
  - An appropriate combination of number of feature maps and fully connected output size makes CNN perform well.
  - Dropout is a useful way to deal with overfitting and increase the accuracy, especially for large networks.
  - For some imaged, human are easy to misclassify, as well as computers.
- Future Work
  - Use multiple CPU or move to GPU.
  - Try larger CNN with dropout or change the architecture.
  - CNN may be not perfect for this data, try other models to increase accuracy.

Thank You!