Handwritten Digits Recognition Using CNN

Yang, J.

1 INTRODUCTION

THE handwritten digits recognition is the ability which allows computer or some other devices to recognize digits. To recognize digits, computer need to read input from scanned documents, then use classification algorithm to classify the digits shown on these documents. In this report, I will use CNN (Convolutional Neural Network) to do this classification job.

2 PREPARATION

2.1 Dataset

In this classification job, I will use MNIST handwritten digit dataset, which is commonly used for traning handwritten digits recongnition, and the images form this dataset were normalized to the size of 28*28 pixel.

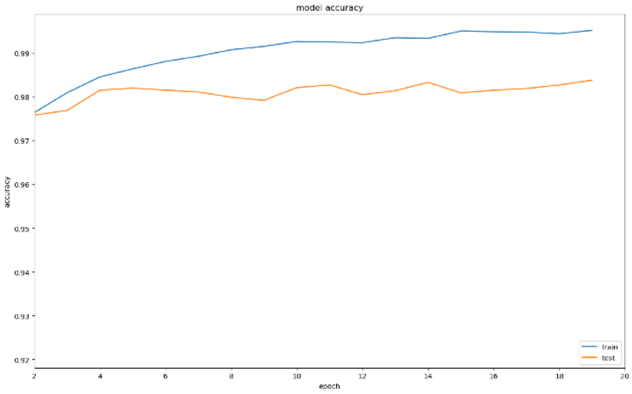
2.2 Construct training, testing and validation split

75% of the data are used for training, and the rest of 25% data is used for testing, furthermore, randomly take 10% of the training data as validation set to find the value of k which can lead to the largest accuracy.

3 CNN

CNN has some similarities with the traditional Neural Network, but the difference is CNN's architechture consists of three kinds of layer: Convolutional layer(this layer is after the input layer), Pooling layer and fully-connected layer, and it has three additional features:
The first one is spatial arrangement, which can decide the size of the output from the input by giving 3 parameters: depth(the number of filters), stride(the number of moves at each time) and padding(the size around the border, usually zero-padding). Another feature is parameter sharing, which is used to control the number of parameters, it is important because the number of parameters is huge if the size of the picture is too large. The main idea of parameter sharing is that all neurons in one slice share one weight, for example, assume the volume of one ouput convolutional layer is 20 * 20 * 100, and all 100 nerons in each depth column connects for example 10 * 10 * 3 region of the input, hence totally need 40000 * 300 weights, but only need 30000 weights after introducing parameter sharing; The previous two features are in Convolutional layer, the other difference is pooling layer. Pooling layer is used to control overfitting as it can reduce the amount of parameters, the most common pooling is Max pooling, it applies a 2*2 filter with the stride 2 on every slice to extract the maximum value from each 2*2 filter.

4 RESULTS AND DISCUSSION



The test accuracy goes from 0.968 to 0.984

Finally the accuracy is 0.983 - 0.985, which is on the same level with MLP and KNN

5 Derivation and Formulation

5.1 Logistic Regression

By comparing with linear regression which describes the relationship between the independent variables with the dependent variable, the output of Logistic Regression is usually presented in the binary(0 or 1).

Consider the case when there are two binary classes, the probability that predicts one data point in one class is increasing with x increasing, and the probability that predicts one data point in the other class is decreasing with x decreasing. This relation can be described by
the sigmoid function

$$y = \frac{1}{1 + e^{-x}}$$

And to perform one logistic regression problem, we usually have a set of input, recorded as **x**, and all of these can be formed as one vector **X**.
In linear regression the relation between x and y can be shown as
y = w**X**
but in logistic regression, because the output y usually as a probability, hence the relation is

$$p = \frac{1}{1 + e^{-wx}}$$

Here p means the probability predicting y = 1

And after processing, the equation is equivalent to

$$log(\frac{p}{1-p}) = wX$$

And because Logistic Regression uses logit function

$$logit(p) = log(\frac{p}{1-p})$$

Hence the model can be named as Logistic Regression

5.2 SVM

In classification problem, if use a line to classify two classes, there are many possible lines. For this situation, SVM is an algorithm to find a best decision boundary.

Given two classes, the decision boundary learned by SVM maximizes the margin.

We can draw two lines according to the two support vectors, and draw a line which is perpendicular with them and cross the origin $(wx = t)$
Besides assume the point which is cloest to decision boundary is x and assume the distance between the boundary and the x is 1,

hence the distance is $\frac{1}{|w|}$

And we will maximize it,

Which also equivalent to minimize $|w|$

And subject to $min|w.x - t| = 1$
And because y > 0, hence it is equivalent to
$y(wx - t) >= 1$

To solve the equation with this subject, we use Lagrangian multipliers:

$$\mathcal{L}\,(x_1, \dots, x_n, \alpha_1, \dots, \alpha_m) = f(x_1, \dots, x_n) + \sum_{j=1}^{k} \alpha_j g_j(x_1, \dots, x_n)$$

Here f(x) is the constrained optimization problem, and g(x) is the constraints.

Then

$$\begin{aligned}\mathcal{L}(w, t, \alpha_1, \dots, \alpha_m) &= \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \alpha_i(y_i(w.\mathrm{x}_i - t) - 1)\\ &= \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \alpha_i y_i(w.\mathrm{x}_i) + \sum_{i=1}^{m} \alpha_i y_i t + \sum_{i=1}^{m} \alpha_i\\ &= \frac{1}{2}w.w - w.\left(\sum_{i=1}^{m} \alpha_i y_i \mathrm{x}_i\right) + t\left(\sum_{i=1}^{m} \alpha_i y_i\right) + \sum_{i=1}^{m} \alpha_i\end{aligned}$$

And by taking partial derivatives with respect to t and w respectively and setting each to 0, we get

$$\textstyle\sum_{i=1}^{m} \alpha_i y_i = 0$$

$$\mathrm{w} = \textstyle\sum_{i=1}^{m} \alpha_i y_i \mathrm{x_i}$$

Next substitude them back to the main formula,

We get

$$\begin{aligned}\mathcal{L}(\alpha_1,\dots,\alpha_n) &= -\frac{1}{2}\left(\sum_{i=1}^{m}\alpha_i y_i \mathrm{x}_i\right).\left(\sum_{i=1}^{m}\alpha_i y_i \mathrm{x}_i\right)+\sum_{i=1}^{m}\alpha_i \\ &= -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \mathrm{x}_i.\mathrm{x}_j+\sum_{i=1}^{m}\alpha_i\end{aligned}$$

Hence the dual form of SVM is

$$\alpha_1^*,\dots,\alpha_m^* = \underset{\alpha_1,\dots,\alpha_n}{\operatorname{argmax}} -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \mathrm{x}_i.\mathrm{x}_j+\sum_{i=1}^{m}\alpha_i$$

$$\text{subject to } \alpha_i \geq 0, 1 \leq i \leq m, \sum_{i=1}^{m}\alpha_i\, y_i = 0$$

For how to calculate the result:

Given the **X** vector, and the output class **y**(usually -1 and 1);

First, we need to get **X’**, which is each row in **X** multiply the corresponding row in **y**;

Then we can get Gram matrix by multiply **X’** with its transpose form **X’$^{\mathbf{T}}$**;

Finally simplify the dual form of SVM by using the Gram matrix and the constraint, then for the simplified equation, set partial derivatives to 0 to get the result.