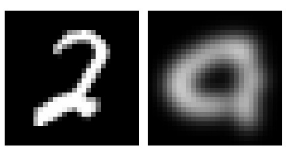
Logistic Regression and Deep Learning for Image Classification

Zihan HU
University of Illinois at Urbana-Champaign
zihanhu2@illinois.edu

1 Abstract

There are many ways to do image classification. The popular ways are using logistic regression or deep neural networks. Regarding different tasks, different methods have their pros and cons. Specifically, we want to determine if pictures from the MNIST data set are blurry or not. The data is the grayscale image with 28 by 28 in dimension size. There are 60000 images in total. This is a binary classification task, and I implement logistics regression with feature extraction and a deep neural network to find a better solution for the task. Based on the validation data, I find out logistics regression with feature extraction can successfully handle such tasks, and a deep neural network also performs close to perfect on such tasks.



2 Introduction

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common model stores the weights and bias and outputs binary outcomes using gradient descent. And it uses maximum likelihood estimator to estimate the output. To have a better estimation of labels, people implement some useful feature extraction on the data to have a larger difference between data with different classes. This method is to modify the data in some ways to extract the important features. Feature extraction in logistic regression can generally increase the speed of learning and improve testing accuracy. The other method is using deep neural networks, which are very popular right now in image classification. Deep learning is an end-to-end strategy to allow artificial intelligence to identify the abstract relationship between different things. The model has different kinds of layers with trainable parameters. By passing through the pair of labeled data, the neural network can find its way to find "features" to better classify the data. The layers people use most often are convolutional layers and fully-connected layers with some pooling and activation function. There are many famous models for image classification, such as ResNet, and VGGNet. In this task, I will try my own designed network and famous ones to see how they perform.

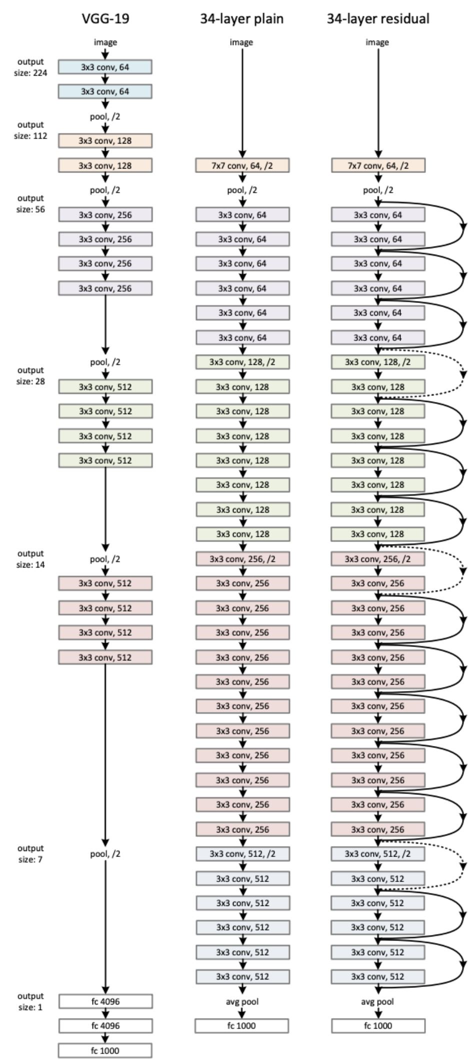


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

3 Related Work

Feature Extraction
There are many ideas about feature extraction. And in different tasks, different feature extraction methods perform differently. Feature Extraction and Image Processing for Computer Vision is a really good book about how to choose the good approach. In the chapter four, it mentions the low-level feature extraction including first order edge detection and second order edge detection. In the section of first order edge detection, there are three different approaches, which are Prewiit edge detector, Sobel edge detector, and Canny edge detector.

ResNet and VGGNet
ResNet and VGGNet(Figure above) are very good neural network architecture in computer vision. They are all based on convolutional layers, max pooling layers, and fully connected layers. These two famous architectures give me really good idea when I design my neural network of how should I choose the layers.

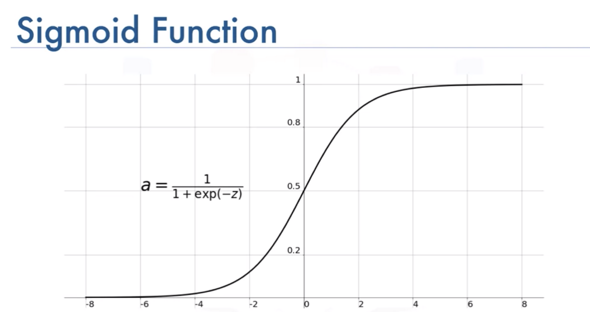
4 Expected Result and Impact

There are a lot of popular image data sets, such as ImageNet, CIFAR-10, and MNIST. ImageNet is the one with more than 14 million images comprising many different classes, which is much higher than 2. The efficiency to train the same network is dependent on how strong is the GPU. By accessing multiple strong GPUs, it is well-known that EfficientNet L2 can achieve about 88 percent accuracy on such a huge data set in a short period of time, which is super incredible. Image classification on relatively smaller data sets like CIFAR-10 and MNIST with 10 different classes is also a simple task nowadays. Using a relatively complicated network can achieve above 99 percent accuracy, which is so close to perfect. It is obvious to say that a deep neural network in image classification is so strong, that it even exceeds the accuracy of human beings. Focusing on this specific blurry binary image classification task, I also expect to have a fairly good estimation. Compared to the strong deep neural network, logistic regression with feature extraction has less trainable parameters which may not be as good as the neural network model. Since I do not have a strong GPU, my goal is to use relatively simple architecture to achieve above 90 percent accuracy on Logistic regression and above 95 percent accuracy using deep neural network.
Binary image classification is very useful in the true/false situation, which is very common in real life. To be more specific, based on the research of determination of blurry image, we can apply such model in dangerous signal detection in different fields. Moreover, in data processing, this model can help to extract those blurry image, and by applying some filters, we can have a better data sets. Since I am only working on image of low quality, it also gives people idea to apply on high quality images.

5 Methods and Approach

5.1 Logistic Regression with Feature Extraction

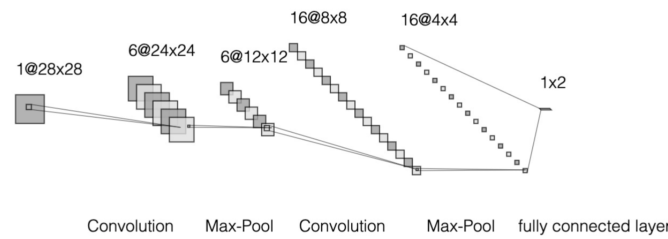
Logistic regression is a powerful supervised machine learning algorithm used for binary classification problems by gradient descent. My task is to determine if the image is blurry or not, which is binary classification. Therefore, logistic regression is a suitable choice. Different from linear regression, by applying the non-linear sigmoid function, logistic regression's range is bounded between 0 and 1. Due to the range to sigmoid function, I need to modify the label to 0 and 1 at first since the given labels are -1 and 1. In addition, logistic regression fits well since it does not require a linear relationship between inputs and output variables. Meanwhile, compared to the support vector machine, logistic regression is easy to implement. All I need to do is iteratively update the weight and bias according to the correct labels. To avoid overfitting, I also implement a simple regularization when updating the weight and bias. Finally, I pass through the 40000 images to train the classifier and test the validation set.



To further improve the accuracy and convergence rate, there are several possible ways. First of all, I can modify my hyperparameters including learning rate, number of iterations, and regularization. Hyperparameter tuning plays a significant role in machine learning. Second, I can choose different ways to regulate my data. Last but not least, I can do some feature extraction on image data. This is one of the most interesting topics in logistic regression. Good feature extraction can dramatically increase the accuracy and convergence rate. By inspecting individual image data, the blurry image looks like passed through a Gaussian filter. Since the Gaussian filter is a low-pass filter, which can discard high-frequency information(sharp) features oriented along either the X or Y axis in the image. For example, compared to the blurry image, there are sharper edges in the regular image. Therefore, intuitively speaking, we can take the gradient of the image in either X or Y direction as a feature for this classification task, because sharper edges have large gradient changes than soft edges due to the blurry. Therefore, I implement the Sobel filter when processing the image. Sobel filter is the filter that can calculate the gradient in either direction or both directions at the same time. By using such a filter, I can easily extract the gradient feature and do the logistic regression.

5.2 Deep Neural Network

Another popular way to do image classification is using a deep neural network. The deep neural network is very powerful in such a task. It has a strong learning ability that integrates the feature extraction and classification process into a whole to complete the image classification test. There are many popular neural network architectures. ResNet and VGGNet are two famous ones. They are relatively complicated networks that need a lot of calculation resources. Although they are very powerful, they are way too costly in this task, which is super inefficient. However, I will explore more about complicated neural networks if the current task becomes more complicated in the later section.
Based on the experience of studying deep learning, I designed my network for this task using a convolutional neural network with a fully connected layer. It is easy to understand why using the fully connected layer. After the training process by the convolutional neural network, we need to classify the data into two classes, which can be done by a fully connected layer with a flattening operation. But why do I use CNN in the image classification task? The convolutional neural network is based on the convolution operations on image data, which can take advantage of the local spatial coherence of images. Because adjacent pixels are relevant, CNN may drastically reduce the number of operations required to process a picture by employing convolution on patches of adjacent pixels. That is referred to as local connection. Each map is then filled with the output of a small patch of pixels by convolution, slid with a window over the whole image. This is done by the 2D kernel with trainable parameters. By applying the cross entropy loss during the training, the neural network can find its way to do the feature extraction. CNN captures a better representation of high-level-feature data and there is no need to do feature engineering like in the logistic regression. After one iteration of feature extraction, I use the max-pooling layer to extract low-level sharp features from the data like edges since it takes the maximum data in the kernel. By such operation, important features can be extracted easily.
My neural network has 5 layers(figure below), including two convolutional layers, two max-pooling layers, and one fully connected layer. I use the convolutional layer and max-pooling layer twice to extract the high-level feature information and one fully connected layer to do the classification.



6 Result and Data

6.1 Logistic regression

Without the feature extraction, by tuning the hyperparameters, the basic logistic regression can have accuracy above 90 percent by 10000 iterations. When I apply the Sobel filter in x direction in image data and then do the logistic regression, the convergence rate is so quick, that it takes much fewer iterations to have a small loss. And the accuracy is incredibly high, which is above 99 percent. There is a loss plot below. I also try the sobel filter in y or in both x and y, but they did not perform as well as in x direction. However, not every feature extraction methods are good. Although canny filter is also a edge detector, it has worse performance, even worse than Logistic regression without any feature.

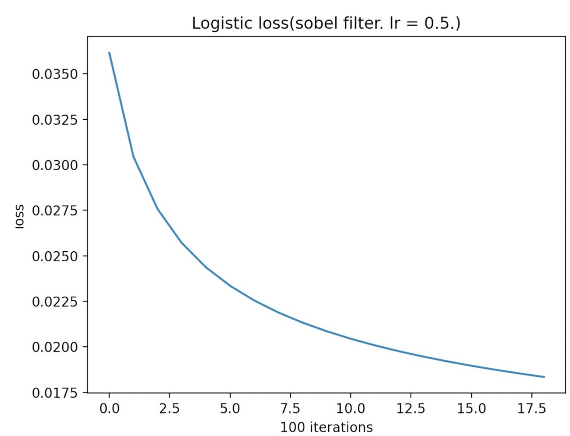
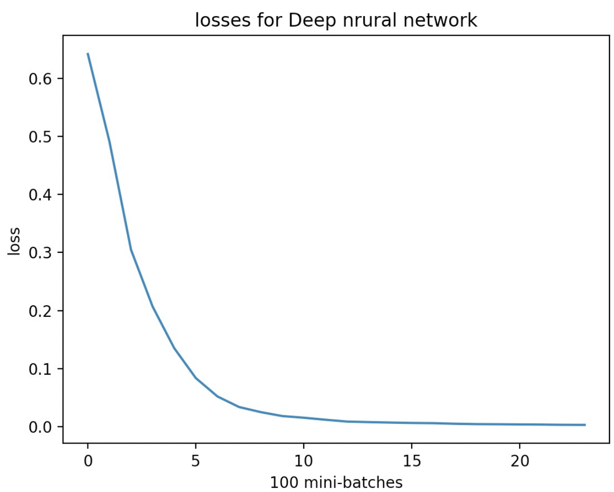


Table 1: Train/Test Accuracy of Logistic regression

Iterations /learning rate	0.01	0.1	0.5
3000 (no feature)	85.25/88.88	90.07/89.81	91.74/91.38
10000 (no feature)	87.80/87.45	91.29/90.99	92.77/92.3
3000 (Canny filter)	81.64/81.26	84.70/83.81	84.82/83.52
3000 (Sobel filter)	99.07/99.15	99.43/99.5	**99.615/99.67**

6.2 Deep neural network

The result of deep neural network is even more incredible. Although my neural network architecture is very simple, it performs extremely well on binary classification. The convergence rate is so quick that I can have a 99 percent accuracy on my validation set when I only pass in 200 mini-batches(32 image per batch), which in total 3200 training data. In addition, if I pass in more data, the validation accuracy can even be 100 percent. There is the losses plot below. By closely look at the data, I can see the loss decreases faster compared to the logistic regression. Therefore, using a simple deep neural network is extremely good in this binary task.



References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. *Microsoft Research*, 2015.

[2] Karen Simonyan, Andrew Zisserman. VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. *Visual Geometry Group, Department of Engineering Science, University of Oxford*, 2015

[3] Mark Nixon, Alberto Aguado. Feature Extraction and Image Processing for Computer Vision. Fourth edition. 2020.