Biostatistics Lab 3 Report

Archana Neupane Timsina

September 29, 2020

Abstract

Here is abstract

1 Introduction

The advent of modern medicine has fundamentally changed the nature of human existence. Many of this breakthroughs have benefited from and will continue to be advanced by computational methods.In addition to machine learning'simpact on everyday life, it has impacted many areas of the physical and life sciences.One of the drivers of these technological advances has been the development of a class of machine learning methods known as deep neural networks. While the technological underpinnings of artificial neural networks were developed in the 1950s and refined in the 1980s, the true power of the technique wasn't fully realized until advances in computer hardware became available over the last 10 years. The advances in gene sequencing have led to the construction of databases that link an individual's genetic code to a multitude of health-related outcomes, including diabetes, cancer, and genetic diseases such as cystic fibrosis. By using computational techniques to analyze and mine this data, scientists are developing an understanding of the causes of these diseases and using this understanding to develop new treatments. Machine learning takes a totally different approach. Instead of designing a function by hand, you allow the computer to learn its own function based on data. You collect thousands or millions of images, each labeled to indicate whether it includes a cat.You present all of this training data to the computer, and let it search for a function that is consistently close to 1 for the images with cats and close to 0 for the ones without. We begin with an example of a data and study how this machine learning can be used to study. We take a data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

2 Theory

We use the Multilayer Perceptrons(MLP) in deep learning methods to analysis the data using machine learnings.

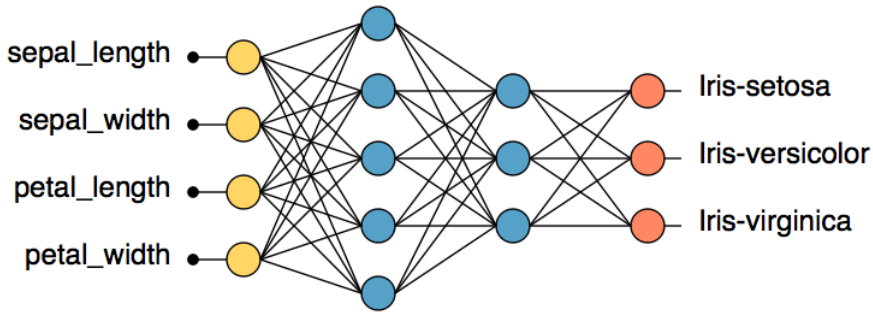

Figure 1: Caption

$$\begin{cases} h_1 = \phi_1(M_1 x + b_1) \\ h_2 = \phi_2(M_2 h1 + b_2) \\ . \\ . \\ . \\ h_{n-1} = \phi_{n-1}(M_{n-1} h_{n-2} + b_{n-1}) \\ y = \phi_n(M_n h_{n-1} + b_n) \end{cases} \tag{1}$$

In this equation $M_1, M_2, ..., M_n$ are matrix, $b_1, b_2 ... b_n$are vectors and all $h_i's$ are hidden layers. . x is training sets and $\phi(.)$ is known as activation function.There are different types of functions are used as activation function for example rectifed linear unit (ReLU), $\phi(x) = (max0, x)$, the hyperbolic tangent, $tanhx$, and the logistic sigmoid,$\phi(x) = (1/1 + e^{-x})$.

Once we get data set, This dataset is known as the training set. It should consist of a large number of (x, y) pairs, also known as samples. Each sample specifies an input to the model, and what we want the model's output to be when given that input.Next we need to define a loss function $L(y, \hat{y})$, where y is the actual output from the model and $\hat{y}$ is the target value specified in the training set. This is how we measure whether the model is doing a good job of reproducing the training data. It is then averaged over every sample in the training set:

$$\begin{cases} average \quad loss = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{y_i}) \\ \\ L(y_i, \hat{y_i}) = \sqrt{\sum_i (y_i - \hat{y_i})}^2 \end{cases} \tag{2}$$

We want to search for the parameter values that minimize the average loss over the training set.There are many ways to do this, but most work in deep learning uses some variant of the gradient descent algorithm. Let θ represent the set of all parameters in the model. Gradient descent involves

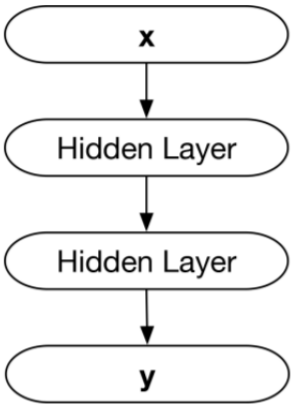

Figure 2: A multilayer perceptron

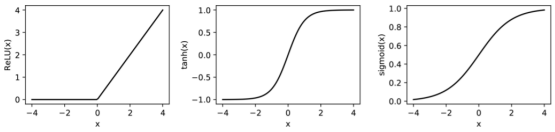

Figure 2-3. Three common activation functions: the rectified linear unit, hyperbolic tangent, and logistic sigmoid.

Figure 3: Activated function

taking a series of small steps:

$$\theta \leftarrow \theta - \epsilon \frac{\partial L}{\partial \theta}$$

where L is the average loss over the training set. Each step moves a tiny distance in the "downhill" direction. It changes each of the model's parameters by a little bit, with the goal of causing the average loss to decrease.ϵ is called the learning rate, and it determines how much the parameters change on each step. It needs to be chosen very carefully: too small a value will cause learning to be very slow, while too large a value will prevent the algorithm from learning at all. Vallidation is the process where we need to test whether the algorithm works for test set. Overfitting is a major problem for anyone who uses machine learning. One of method to regularize is use as large as number of training set to make the algorithm work perfectly. All of the above discription are theory behind to analysis the given data set.

3 Procedures

We use Python colab to analysis the computational part. After setting up all of the data into the my drive and calling that into colab, the following is the selection of training set and test set from data

```
X = input_data[r[:cut],:] #for training
X_test = input_data[r[cut:],:] #for test
Y = target_data[r[:cut]] #for training
Y_test = target_data[r[cut:]] #for test
```

Then we use the following codes to set up the given data into meodel and get parameter.

```
def softmax(x):   # defining a  function
    s1 = torch.exp(x - torch.max(x,1)[0][:,None])
    s = s1 / s1.sum(1)[:,None]
    return s
def cross_entropy(outputs, labels):
    return -torch.sum(softmax(outputs).log()[range(outputs.size()[0]), labels.long()
    ])/outputs.size()[0]

def randn_trunc(s): #Truncated Normal Random Numbers
    mu = 0
    sigma = 0.1
    R = stats.truncnorm((-2*sigma - mu) / sigma, (2*sigma - mu) / sigma, loc=mu,
    scale=sigma)
    return R.rvs(s)
```

All other formula are in [2]

4 Analysis

In this section you will need to show your experimental results. Use tables and graphs when it is possible. Table **??** is an example.

5 Conclusions

Here you summarize your findings.

References

[1] Why life science?

[2] Biostatistics Lab 2.