ECE421 Assignment 1

**Part1 Logistic Regression with Numpy**

**Q1 Loss Function and Gradient**

Loss function derivation:

****

****

**图片包含 图表

描述已自动生成**

**文本, 应用程序

描述已自动生成**

Gradient Loss function:

**图片包含 图表

描述已自动生成**

Let

图片包含 文本

描述已自动生成

**Q2 Gradient Descent Implementation**

The gradient descent function implemented will calculate the new weights and bias in every iteration of epochs. Also the loss and accuracy data in each iteration will be stored to the global variable loss\_array and accu\_array

**文本

描述已自动生成**

**Q3 Tuning the Learning Rate**

**图形用户界面

描述已自动生成**

图形用户界面, 应用程序

描述已自动生成

|  |  |  |  |
| --- | --- | --- | --- |
|  | alpha = 0.005 | Alpha = 0.001 | Alpha = 0.0001 |
| Training Accuracy | 0.9814 | 0.9758 | 0.9595 |
| Validation Accuracy | 0.9991 | 0.9751 | 0.9722 |

Table above shows the Training and Validation Accuracy after 5000 epochs.

From figure above, the higher the learning rate, the faster the training and validation loss will decrease.

Thus, given limited epochs the larger learning rate will result in higher accuracy. Hereby alpha = 0.005 provides the highest accuracy.

**Q3 Generalization**

图片包含 图示

描述已自动生成

图形用户界面

描述已自动生成

|  |  |  |  |
| --- | --- | --- | --- |
|  | r = 0.001 | r = 0.1 | r = 0.5 |
| Training Accuracy | 1.0019 | 0.9786 | 0.9616 |
| Validation Accuracy | 0.9836 | 0.9772 | 0.9680 |

Table above shows the Training and Validation Accuracy after 5000 epochs.

As we can see with increasing regularization parameter the accuracy is decreasing, but the difference between the training accuracy and validation accuracy is also decreasing.Thus regularization is to prevent the model from over-fitting on the actual data.

**Part2 Logistic Regression in TensorFlow**

**Q1 Building the Computational Graph**

文本

描述已自动生成文本

描述已自动生成

**Q2 Implementing Stochastic Gradient Descent**

Helper function to randomly shuffle the Traindata and Traintarget

**文本

描述已自动生成**

Helper function for loading and processing data

**文本

描述已自动生成**

SGD function

文本

描述已自动生成

文本

描述已自动生成

文本

描述已自动生成

图表

描述已自动生成 图形用户界面

中度可信度描述已自动生成

**Q3 Batch Size Investigation**

图形用户界面

中度可信度描述已自动生成 图形用户界面, 图表

描述已自动生成

图表

中度可信度描述已自动生成图形用户界面

描述已自动生成

图表

描述已自动生成图形用户界面

中度可信度描述已自动生成

|  |  |  |  |
| --- | --- | --- | --- |
|  | Batch = 100 | Batch = 700 | Batch = 1750 |
| Training Accuracy | 0.994 | 0.988 | 0.9865 |
| Validation Accuracy | 0.96 | 0.98 | 0.97 |

Both the Training and Validation accuracy is high with 3 different batch size.

It seems that batch size will post no significant impact to the ultimate accuracy.

But from the graph, smaller batch size will cause the accuracy curve to vibrate , while with large batch size the accuracy increase steadily.

**Q4 Hyperparameter Investigation**

图形用户界面

低可信度描述已自动生成 图形用户界面, 应用程序

中度可信度描述已自动生成

图形用户界面

低可信度描述已自动生成 图形用户界面

低可信度描述已自动生成

图形用户界面

低可信度描述已自动生成 图形用户界面, 应用程序

描述已自动生成

**Q5 Comparison against Batch GD**

**形状

中度可信度描述已自动生成图片包含 图示

描述已自动生成Batch GD with alpha = 0.001**

|  |  |
| --- | --- |
| Accuracy | Alpha = 0.001 |
| Training | 0.9758 |
| Validation | 0.9751 |

|  |  |
| --- | --- |
| Accuracy | Batch = 700 |
| Training | 0.988 |
| Validation | 0.98 |

图形用户界面

描述已自动生成图表

中度可信度描述已自动生成

From graph we can see that SGD use 700 epochs to achieve accuracy of 0.98, while Batch GD use 5000 epoches to achieve the accuracy of 0.97. Thus the SGD is much faster and have better performance than the Batch GD.