Sami Emre Erdogan (夏承思)
Student ID: 2019280513

Deep Learning - Logistic Regression
Homework 1

In linear regression we model our data with a linear function which outputs continuous values :

f(w,b) = wx +b

However, when we talk about logistic regression we are not interested in continuous values but we want probabilistic values. In order to plot predicted values into probabilities between 0 and 1, we use the sigmoid function to plug it into our linear function.

$$S(z) = \frac{1}{1+e^{-z}}$$

```
def sigmoid(result):
    final_result = 1/(1+np.exp(-result))
    return final_result
```

From these assumptions we can see the hypothesis of logistic regression.

$$h_\theta(\mathbf{x}) = \frac{1}{1+e^{-\theta^\top \mathbf{x}}}$$

Then we have to apply this logistic regression function over the summation of weights features:

```
Z = np.dot(input_batch_size, weights) + bias
S = sigmoid(Z)
```

Later on, we had to define the cost function(0.5) :

```
y_hat = S >= 0.5
```

We can now calculate the accuracy and loss of each data and update the weights in each epoch and batches:

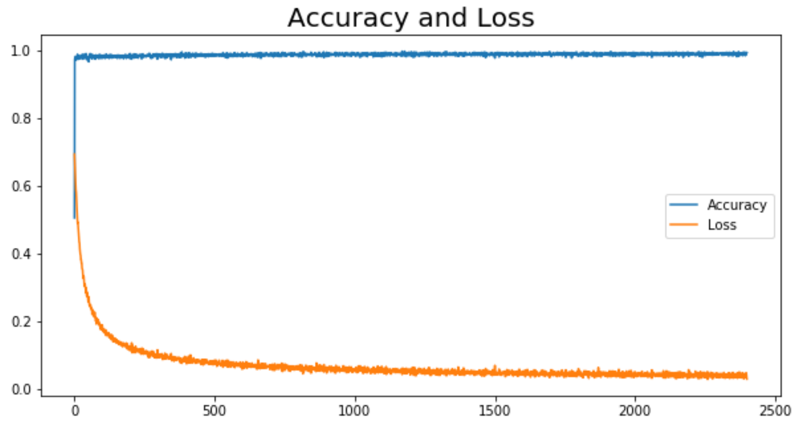
```
# calculate the loss (and accuracy)
loss = (-1.0) * np.mean(np.multiply(y, np.log(A)) + np.multiply(1.0-y, np.log(1.0 - A)))
accuracy = compute_accuracy(y, y_hat)

# update weights
dw = np.dot(input_batch_size.T, np.transpose(A - y).T) * (1.0/input_batch_size.shape[1])
db = np.mean((A - y))
weights = weights - learning_rate * dw
bias = bias - learning_rate * db

losses.append(loss)
accuracies.append(accuracy)
```

Sami Emre Erdogan (夏承思)
Student ID: 2019280513

After our training data is finished, we can check the accuracy and the loss of our logistic regression program:




```
#average
average_accuracy= (sum(accuracies)/len(accuracies))
fd_average_accuracy = "{:.4f}".format(average_accuracy)
print("average accuracy: "+ fd_average_accuracy)

#loss
average_loss = float(sum(losses)/len(losses))
fd_average_loss = "{:.4f}".format(average_loss)
print("average loss : "+ fd_average_loss)
```

```
average accuracy: 0.9879
average loss : 0.0697
```

As we can see from my results above, my calculated average accuracy was: 0.9879, average loss was: 0.0697