Deep Learning (IST, 2024-25)  
Homework 1

Group 23

113191 – Miguel Miranda | 102787 – Ricardo Santos

**Question 1.**

1. **(a)**

A graph of blue and orange lines

Description automatically generated

Figure 1 - Training and Validation accuracies as a function of the epoch number for the Perceptron

By the end of the 100 epochs the performances with accuracy as the chosen metric were the following: 0.5598 on the training set; 0.3868 on the validation; 0.3743 on the testing set.

1. **(b)**

|  |  |
| --- | --- |
|  |  |
| Figure 2 - Training and Validation accuracies as a function of the epoch number for the non-regularized Logistic Regression | *Figure 3 - Training and Validation accuracies as a function of the epoch number for Logistic Regression with L2 regularization (l2\_penalty = 0.01)* |

By the end of the 100 epochs the non-regularized version of the Logistic Regression obtained the following results with accuracy as the chosen metric: 0.6694 on the training set; 0.4623 on the validation set; 0.4597 on the testing set.

By the end of the 100 epochs the Logistic Regression with L2 regularization obtained the following results with accuracy as the chosen metric: 0.5683 on the training set; 0.4972 on the validation set; 0.5053 on the testing set.

The non-regularized version of logistic regression achieves a higher training accuracy (0.6694) because it fits the training data closely, allowing weights to grow large. However, this leads to poor generalization, as seen in the much lower validation (0.4623) and testing (0.4597) accuracies. This discrepancy indicates the model is overfitting to the training set.

In contrast, with L2 regularization (l2\_penalty = 0.01), the model achieves a lower training accuracy (0.5683) due to the constraint on weight magnitudes. This prevents the model from fitting overly complex patterns. However, this trade-off results in better generalization, as indicated by the higher validation (0.4972) and testing (0.5053) accuracies. This suggests the regularized model is capturing more general patterns in the data, reducing overfitting and improving performance on unseen data.

1. **(c)**

|  |  |
| --- | --- |
| A graph with a blue line  Description automatically generated |  |
| Figure 4 – l2 norm of the weights as a function of the epoch number for the non-regularized Logistic Regression | Figure 5 – l2-norm of the weights as a function of the epoch number for the Logistic Regression with l2-regularization |

The L2-norm of the weights in the non-regularized logistic regression increases steadily throughout the 100 epochs. This continuous growth indicates that the model is trying to fit the training data as closely as possible by increasing the weight’s magnitudes, which ultimately leads to overfitting. As the weights become large, the model captures specific details and noise in the training data, causing poor performance on unseen data.

In the L2-regularized version, the L2-norm of the weights stabilizes early (after about 20 epochs) and remains relatively constant. This suggests that the regularization term effectively penalizes large weights, preventing them from growing excessively. As a result, the model maintains smaller weights, which helps reduce overfitting and promotes better generalization. This behavior is evident from the improved validation and testing accuracies compared to the non-regularized version.

1. **(d)**

Despite both L1 and L2 regularization promoting smaller weights, L1 regularization uniquely promotes sparsity, meaning that some weight values can be set to exactly zero during the regularization process.

L1 regularization adds a penalty proportional to the absolute value of the weights (||). This penalty can force weights to shrink to zero when the benefit of a weight's contribution to reducing the loss is outweighed by the penalty.

L2 regularization, on the other hand, adds a penalty proportional to the square of the weights (). This encourages weights to become smaller but does not typically shrink them to zero.

Because L1 regularization can lead to some weights being set to zero, it results in a simpler, sparser model that uses fewer features. In comparison, L2 regularization tends to produce a model where all features still contribute, though with smaller weights.