AI 534 IA2 Report

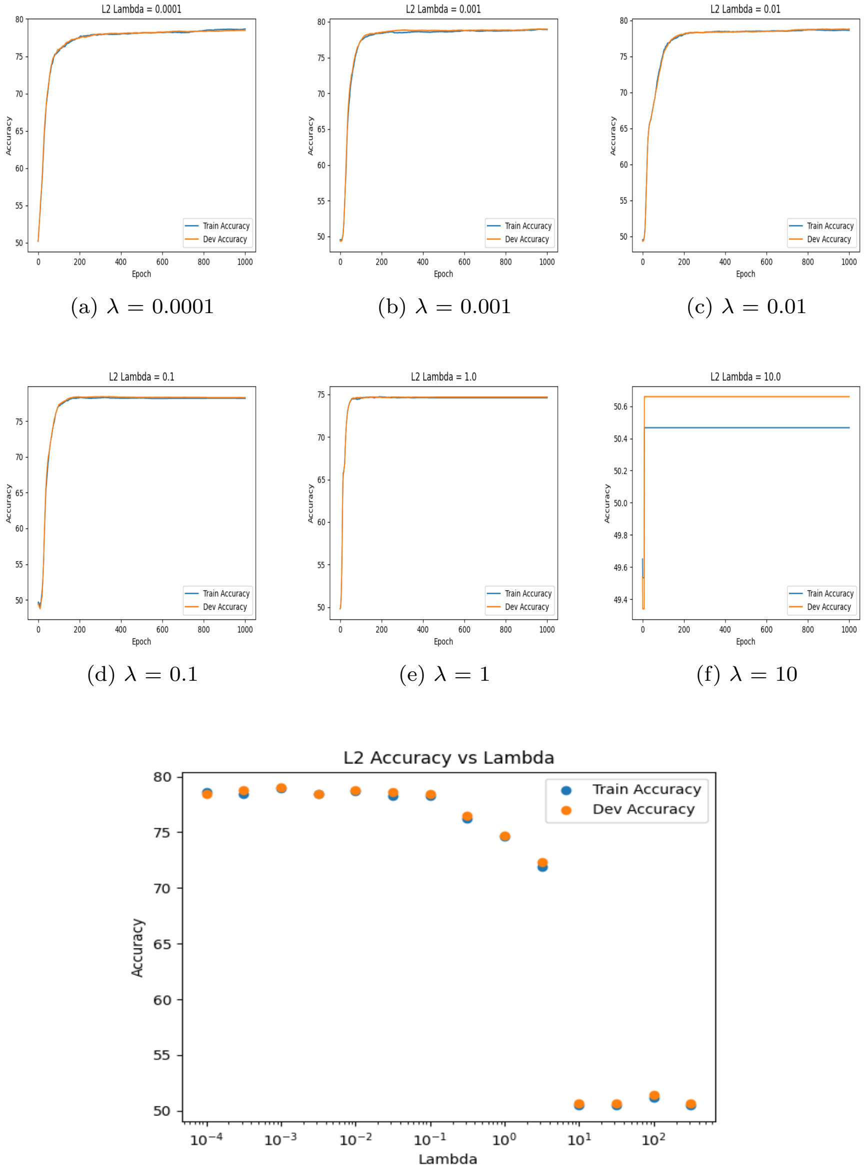
Rishab Balasubramanian

Part1.a. As we increase lambda, we see from Fig.1 that the training accuracy slowly reduces, while dev accuracy increases slightly. After a point ($\lambda = 0.01$), both accuracies start falling. This shows that the accuracy improves upto some regularization parameter λ (by reducing overfitting), after which the accuracy decreases (because of underfitting trying to minimize the weights), which can be attributed to the fact that for a larger λ, we place more importance on minimizing weights, which thus dominate the gradient. Based on the validation accuacry $\lambda = 0.001$ gives best results (Validation accuracy = 78.97%).

Table 1: Largest Parameters from L2 Regularization

$\lambda_- = 0.0001$	
Parameter Name	Weight Value
Previously_Insured	2.212278
Vehicle_Damage	1.851492
Driving_License	1.203679
Policy_Sales_Channel_78	0.983048
Policy_Sales_Channel_37	0.977445
$\lambda_* = 0.001$	
Parameter Name	Weight Value
Previously_Insured	2.184518
Vehicle_Damage	1.754571
dummy	1.228644
Policy_Sales_Channel_20	0.894057
Policy_Sales_Channel_64	0.887815
$\lambda_+ = 0.01$	
Parameter Name	Weight Value
Vehicle_Damage	1.580687
Previously_Insured	1.496561
dummy	0.980942
Policy_Sales_Channel_26	0.454201
Policy_Sales_Channel_152	0.420518

b. We can clearly observe from Table.1 the magnitude of the weights reduces



(a) $\lambda = 0.0001$

(b) $\lambda = 0.001$

(c) $\lambda = 0.01$

(d) $\lambda = 0.1$

(e) $\lambda = 1$

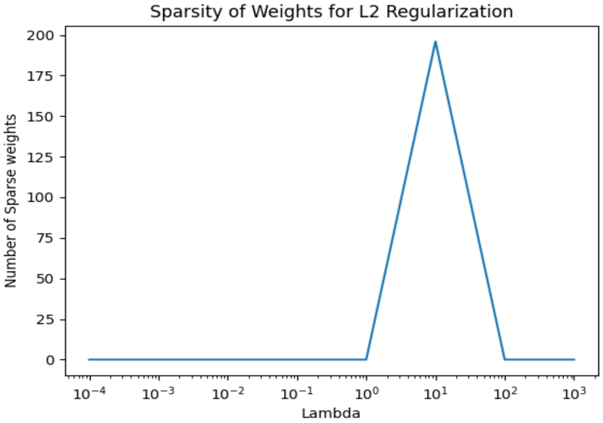
(f) $\lambda = 10$

(g) L2 Accuracy vs λ

Figure 1: Model Accuracy with L2 regularization

as we increase the regularization parameter λ. We see that for $\lambda = 0.0001$, the average of the top five largest weights is 1.4455, for $\lambda = 0.001$ it is 1.38988 and for $\lambda = 0.01$ it is 0.98655. This shows the effect of regularization on reducing the magnitude of weights at higher values of λ due to the regularization contributing more to the gradient.

c. Fig.2 shows the variation of the sparsity of the weights with λ. We see that for some particular value of λ, the weights become highly sparse. This turns out to be close to the optimal value of λ, and thus we can assume these weights to be optimal. However, as we continue to increase λ, we see that the sparsity reduces again. This shows the property of the L2 regularization which prevents sparse weights. If we further increase λ, the weights would go close to zero, but all of them will never become 0 simultaneously. This is attibuted to the gradient of L2 regularization, which is proportional to the value of the weight, and thus the magnitude of the gradient decreases as the magnitude of the weight decreases.



(a) Sparsity for L2 Regularization ($w = 0$)

Figure 2: Number of Sparse weights vs λ

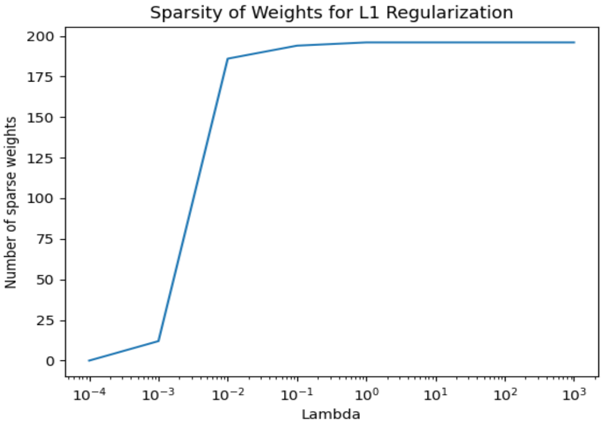
Part 2. a. We can see from Fig.4 that similar to Part 1, while λ increases upto a particular value ($\lambda = 0.01$), the validation accuracy increases and training accuraccy reduces showing that the regularization helps prevent overfitting. Beyond this value of λ, the model begins to underfit. Based on the validation accuracy, $\lambda = 0.001$ gives the best accuracy of 78.84%.

b. We can clearly observe from Table.2 that, similar to Part 1, the magnitude of the weights reduces as we increase the regularization parameter λ. We see that for $\lambda = 0.0001$, the average of the top five largest weights is 1.4107, for $\lambda = 0.001$ it is 1.37866 and for $\lambda = 0.01$ it is 1.011. This shows the effect of regularization on reducing the magnitude of weights as λ becomes very large, highlighting the underfitting of the model.

Table 2: Largest Parameters from L1 Regularization

<table>
<tr><td colspan="2">$\lambda_{-} = 0.0001$</td></tr>
<tr><td>Parameter Name</td><td>Weight Value</td></tr>
<tr><td>Previously_Insured</td><td>2.105241</td></tr>
<tr><td>Vehicle_Damage</td><td>1.916172</td></tr>
<tr><td>Driving_License</td><td>1.054955</td></tr>
<tr><td>Policy_Sales_Channel_132</td><td>0.988875</td></tr>
<tr><td>Policy_Sales_Channel_130</td><td>0.988164</td></tr>
<tr><td colspan="2">$\lambda_{*} = 0.001$</td></tr>
<tr><td>Parameter Name</td><td>Weight Value</td></tr>
<tr><td>Previously_Insured</td><td>2.224153</td></tr>
<tr><td>Vehicle_Damage</td><td>1.813022</td></tr>
<tr><td>dummy</td><td>1.087429</td></tr>
<tr><td>Policy_Sales_Channel_119</td><td>0.890106</td></tr>
<tr><td>Region_Code_51</td><td>0.878618</td></tr>
<tr><td colspan="2">$\lambda_{+} = 0.01$</td></tr>
<tr><td>Parameter Name</td><td>Weight Value</td></tr>
<tr><td>Previously_Insured</td><td>1.825255</td></tr>
<tr><td>Vehicle_Damage</td><td>1.711125</td></tr>
<tr><td>dummy</td><td>0.988511</td></tr>
<tr><td>Policy_Sales_Channel_152</td><td>0.265840</td></tr>
<tr><td>Vehicle_Age_1</td><td>0.265746</td></tr>
</table>

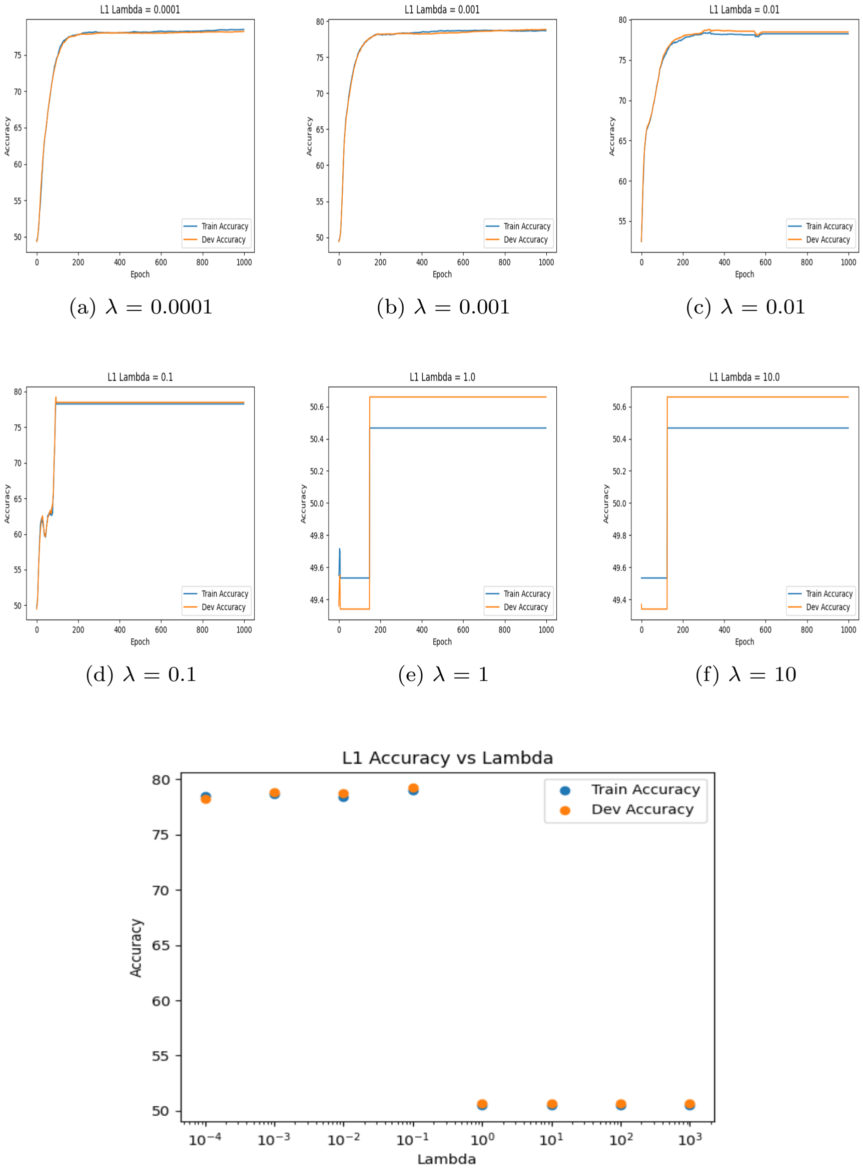
c. Fig.3 shows that as we increase λ the number of parameters that become (approximately) 0 increases almost linearly till all features become sparse. This is different from Part 1, where not all features became sparse, highlighting that L1-regularization makes the weights sparse. This is due to the nature of the gradient of the L1 regularization which is either 1 (when $w > 0$), -1 (when $w < 0$) or 0 (when $w = 0$). Therefore the weights would be driven to zero (or very close to 0) very quickly, and oscillate there. If we continue to increase λ, the number of sparse weights increases (if all the weights are not 0 already), or stays constant if all the weights are already sparse.



(a) Sparsity for L1 Regularization

Figure 3: Number of Sparse weights vs λ

Part3. In this part I tried varying the regularization to check if increasing data would have an impact on the regularization. I found that with more data, a lower regularization worked better. Next I tried to reduce the number of features. I found that the values for Region_Code_(j) and Policy_Sales_Channel_(i) were only 1 for a unique value of j and i. That is for a particular customer, amongst the 154 Policy Sales Channels, only one was used, and likewise so for the Region Code. Thus I converted these to single integers, based on which i and j were used for each customer. The new data-model converged to 78.37%, which was lower than the original data accuracy. I think that the prediction accuracy is limited by both the data and the model. The data have little correlation, and therefore it is not possible to generate higher dimensional data from the ones provided. But I think the model is the biggest limitation in this framework. Logistic regression provides a linear boundary, which limits the accuracy. Using some other model, which could provide non-linear decision boundaries could improve accuracy.



(a) $\lambda = 0.0001$

(b) $\lambda = 0.001$

(c) $\lambda = 0.01$

(d) $\lambda = 0.1$

(e) $\lambda = 1$

(f) $\lambda = 10$

(g) L1 Accuracy vs λ

Figure 4: Model Accuracy with L1 regularization