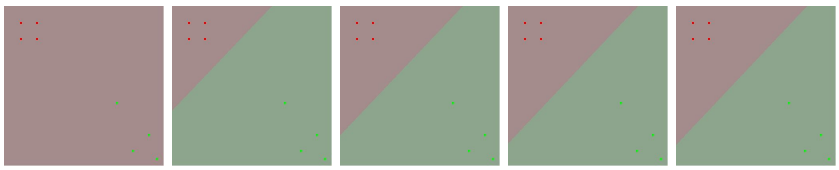
**The visualization of the model learned on the UnitTest using 10 iterations of gradient descent, stepSize=1.0, convergence=0.005**



**Tune the hyperparameter 'convergence' by trying [ 0.01, 0.001, 0.0001, 0.00001 ] (with stepSize of 1.0). Produce a table showing:**

**<convergence parameter>, <steps to convergence>, <validation set accuracy>**

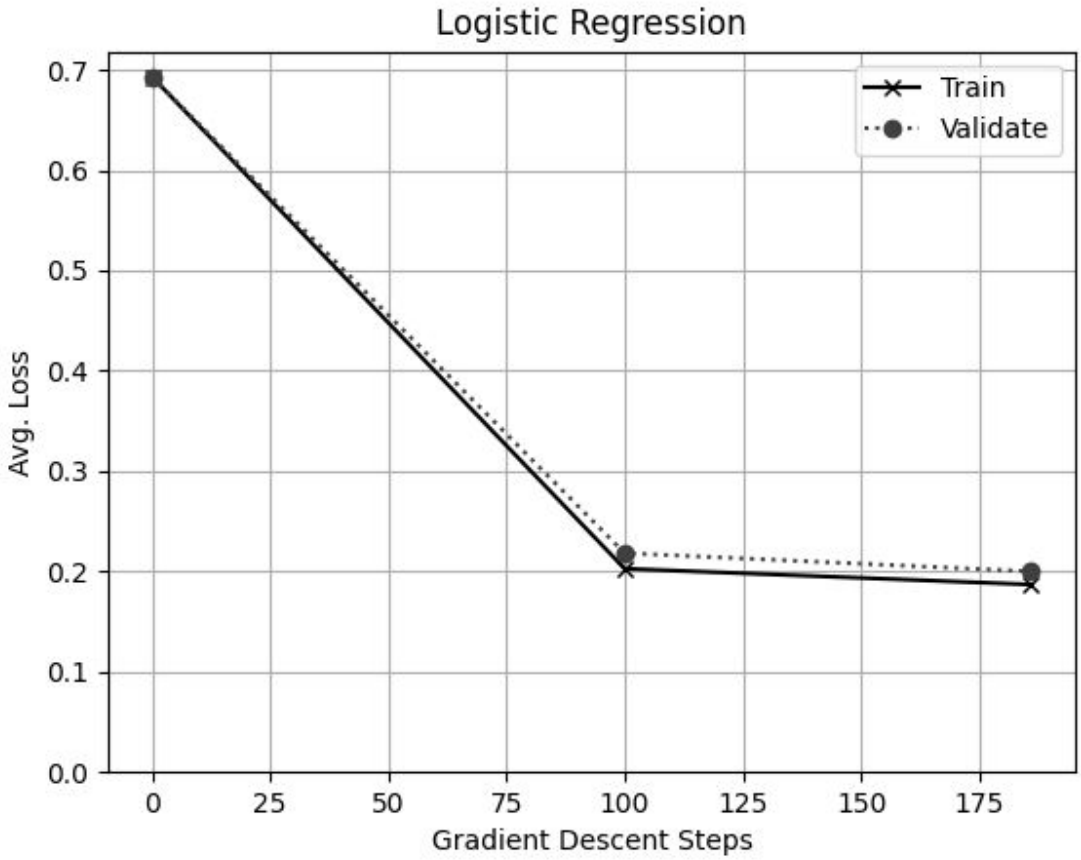
**for each setting of the convergence hyperparameter.**

| Convergence | Steps | Validation Set Accuracy |
|---|---|---|
| 0.01 | 8 | 84.10% |
| 0.001 | 57 | 86.65% |
| 0.0001 | 186 | 92.32% |
| 0.00001 | 522 | 92.68% |

**Describe in 2-3 sentences your interpretation of the output of this hyperparameter sweep. Include a justification for trying more parameters or stopping there. If you do think you should try more parameters, what is the next one you would try?**

I think we should stop here. This convergence limit shows that as there is a limit to how much it can further improve the accuracy as we drive loss lower. This means we are probably approaching the minimum with the gradient basically vanishing. I think I would just use 0.0001 as further improvements are negligible.

**Produce a plot showing the training set loss and the validation set loss every 100 steps during the run with the best convergence value you found in the parameter sweep and stepSize of 1.0. In no more than 100 words describe the difference between the training and validation loss. Which set does the model have better loss on? Why?**




The training data loss represents how well the model can fit to the data it can see (bias). The validation loss represents how well the model can generalize to data it has not seen (variance). The model fits the training data better right now because we have applied no regularization, so with just gradient descent, it will fit the training data as well as it can.