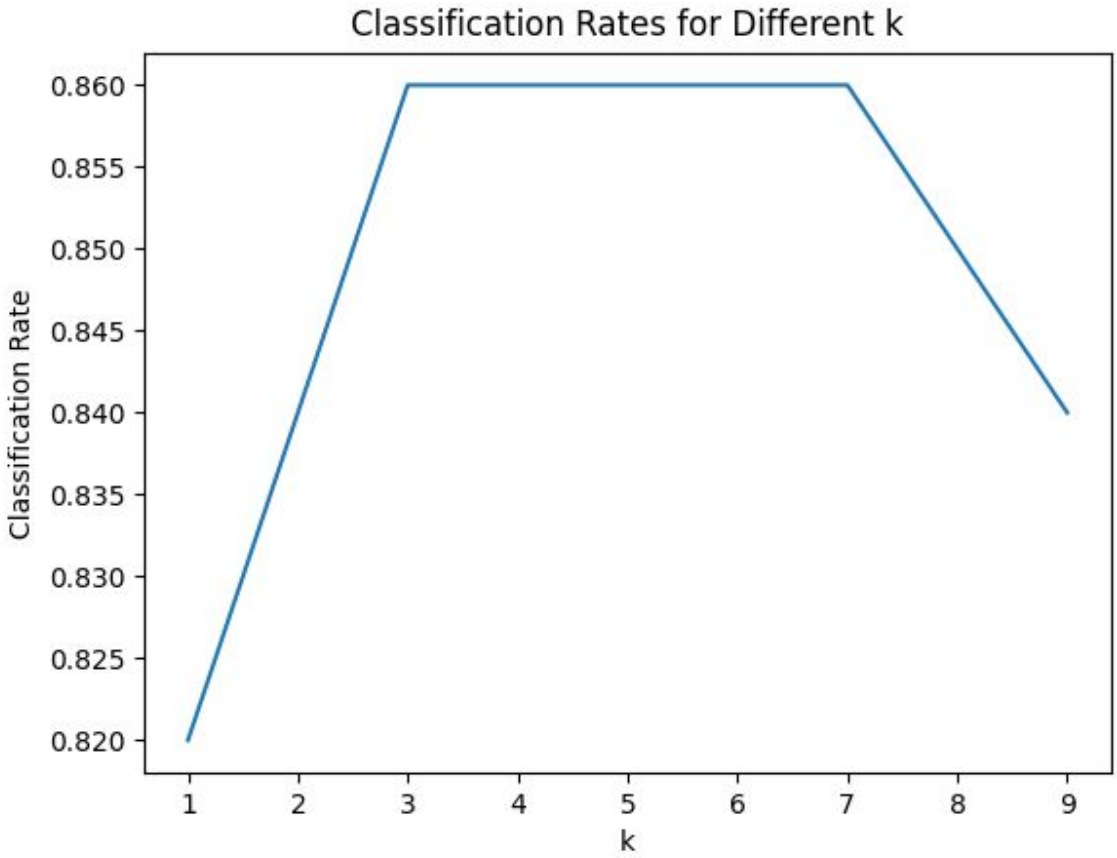
Feature Maps

(a) For a dataset to be linearly separable, it must be the case that if two different points lie in the same half-space, the line segment connecting those points must also lie in the half-space. We have that -1 and 3 both get mapped to 1, but 0 is in the line segment connecting them, yet it doesn't lie in the same half-space. Instead, it gets mapped to 0.

(b) We require that $w_1(-1) + w_2(-1)^2 \geq 0,\ w_1(1) + w_2(1)^2 \leq 0,$ and $w_1(3) + w_2(3)^2 \geq 0$. A possible pair of values is $w_1 = -2$ and $w_2 = 1$.

k-Nearest Neighbors



Based on the plot, I would choose a k of 5 for this model because it gave the maximum accuracy of the tested values of k. With a classification rate of 86% on the validation set, it has the same accuracy as both k-2=3 and k+2=7. For this reason, I believe 5 is the most stable of the tested values, as it's more likely to be a good indicator of performance, given that values tested to the left and right had the same classification rate. When 3, 5, and 7 were used as k for the test set, the model had a classification rate of 0.92, 0.94, and 0.94 respectively, which were even higher than for our validation set, indicating that the model is a good classifier.

Logistic Regression

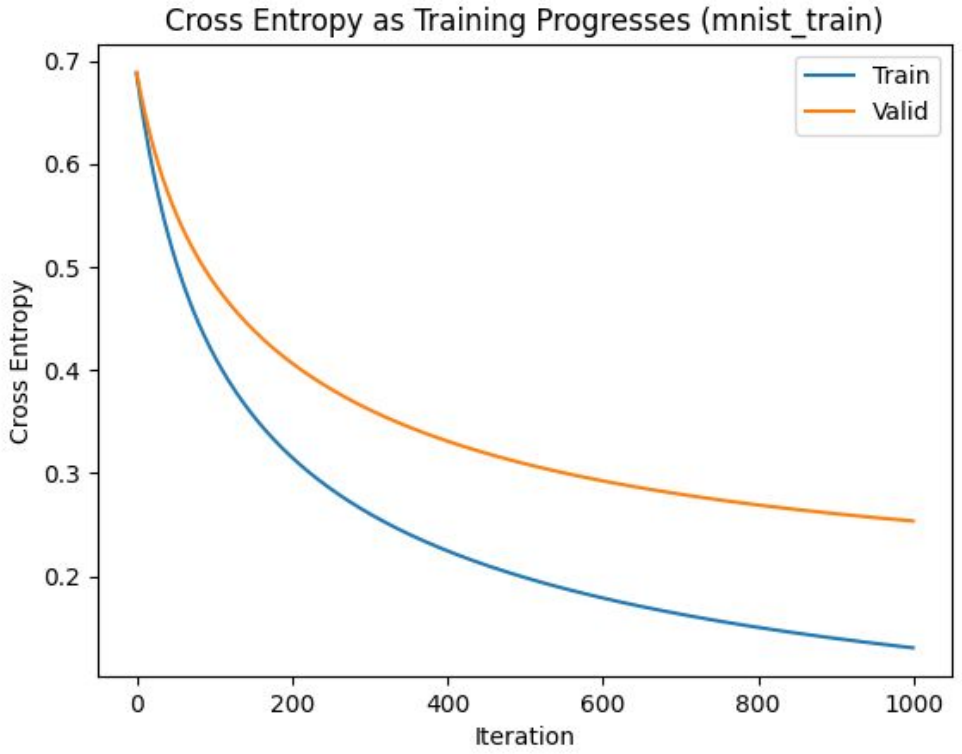
(b) For mnist_train, I found that setting the weights to be all 0, the learning rate to be 0.01, and the number of iterations to 1,000 yielded a cross entropy value of ~0.25 and a fairly high classification rate of 0.88 on the validation set. I initially tried the more conservative values 0.0001 and then 0.001, both with 2,000 iterations, but these gave classification rates of 0.72 and 0.84 respectively, indicating that the process was perhaps too slow with such small values. However, with 0.01, I had the same classification rate for 1000, 1500, and 2000 iterations, suggesting that the weights had already converged and the process could run with fewer iterations at this learning rate.

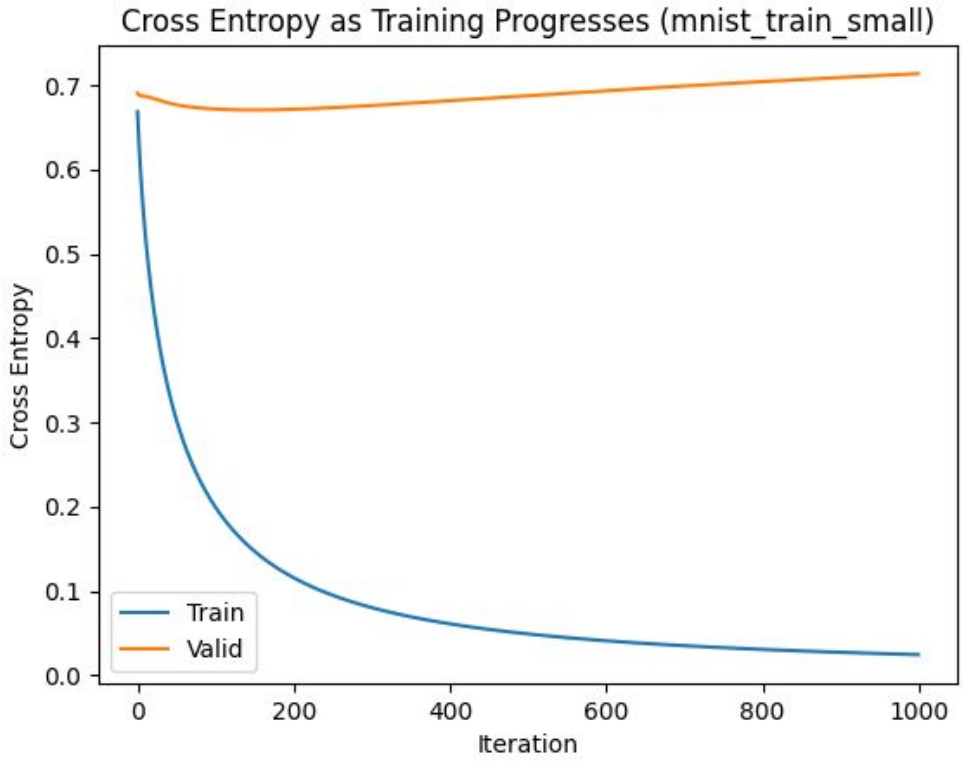
For mnist_train	Cross Entropy	Classification Rate
Training Set	0.13062698413107174	1.0
Validation Set	0.2536710662674644	0.88
Test Set	0.23235634333047045	0.92

I had similar results with mnist_train_small, though it seemed like this set needed more iterations (1,500) to converge.

For mnist_train	Cross Entropy	Classification Rate
Training Set	0.016167553401964817	1.0
Validation Set	0.0.7334134721679268	0.7
Test Set	0.6453121785794036	0.78

(c)


Cross Entropy as Training Progresses (mnist_train)
Train
Valid
Cross Entropy
Iteration
0.7
0.6
0.5
0.4
0.3
0.2
0
200
400
600
800
1000


Cross Entropy as Training Progresses (mnist_train_small)
Train
Valid
Cross Entropy
Iteration
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
200
400
600
800
1000

Penalized Linear Regression

For mnist_train	Averaged Cross Entropy for Training Set	Averaged Classification Rate for Training Set	Averaged Cross Entropy for Validation Set	Averaged Classification Rate for Validation Set
$\lambda = 0$	0.13062698413107174	1.0	0.2536710662674644	0.8800000000000001
$\lambda = 0.001$	0.08532894612851297	1.0	0.22244466242713687	0.8800000000000001
$\lambda = 0.01$	0.10920212345696587	1.0	0.24899755512850746	0.8800000000000001
$\lambda = 0.1$	0.296060160506696	1.0	0.40468580725913883	0.8800000000000001
$\lambda = 1.0$	0.5411829356913466	0.94375	0.5948599481341499	0.8400000000000001

For mnist_train_small	Averaged Cross Entropy for Training Set	Averaged Classification Rate for Training Set	Averaged Cross Entropy for Validation Set	Averaged Classification Rate for Validation Set
$\lambda = 0$	0.024430032402602796	1.0	0.7139639311264893	0.66
$\lambda = 0.001$	0.014390066367772633	1.0	0.7481658684977147	0.7
$\lambda = 0.01$	0.031976036921266826	1.0	0.7619608724006558	0.7
$\lambda = 0.1$	0.13375582176195805	1.0	0.7242734071084331	0.68
$\lambda = 1.0$	0.3605423607786359	1.0	0.7503518424386018	0.66