softmax.ipynb

1. We expect our loss to be close to $-\log(0.1)$ because we have 10 classes and we are randomly initializing our weights. Therefore, we expect the probability of each class to be 0.1. The loss function is the negative log of the probability of the correct class. Therefore, we expect the loss to be close to $-\log(0.1)$.

two_layer_net.ipynb

1. **Sigmoid** function has this problem. In the case of one dimensional, if the input is too large or too small, the gradient will be close to zero because the gradient of sigmoid is close to zero when the input is too large or too small.

2. To decrease the gap between training and testing accuracy:
    - Train on a larger dataset. By training on a larger dataset, the model will be able to generalize better and thus the gap between training and testing accuracy will decrease.
    - Increase the regularization strength. By increasing the regularization strength, the model will be less likely to overfit and thus the gap between training and testing accuracy will decrease.

Linde-Buzo-Gray algorithm

1. Similarities:
    - Clustering Algorithm: Both k-Means and the LBG algorithm are clustering algorithms, aiming to group similar data points into clusters.
    - Iterative Refinement: Both algorithms involve an iterative process where they refine their cluster assignments and update their cluster centroids or codevectors.
    - Convergence: Both algorithms typically have convergence criteria to determine when the clustering has stabilized or when the codevectors have converged to a stable configuration.

2. Differences:

k-means	LBG
Requires an initial assignment of cluster centroids, and the algorithm may converge to different solutions based on the initial centroids.	Starts with a single codevector (centroid) and iteratively splits them, reducing the likelihood of getting stuck in local minima.
Requires the number of clusters (k) to be specified beforehand.	LBG Algorithm: Can dynamically determine the number of clusters based on the desired size of the codebook.
In each iteration, data points are assigned to the nearest centroid, and the centroids are updated based on the mean of the assigned points.	Involves a more complex centroid update process, splitting and merging codevectors iteratively.
Primarily used for vector quantization and clustering.	Specifically designed for codebook design and vector quantization.

3. The run-time complexity of LBG is $O(\text{iterations} \times N \times k \times D)$:

where:

$N = \text{number of data points}$

$k = \text{size of codebook (number of clusters)}$

$D = \text{dimension of data}$