CS5783: Machine Learning
Assignment 4

Prof. Christopher Crick

1 GPT

Implement a decoder-only transformer model that generates stylistically-appropriate text when trained on a source. Feel free to use Andrej Karpathy's NanoGPT as a model, but your code will have the following substantial differences:

- Implement using Tensorflow, rather than PyTorch.
- Choose your favorite out-of-copyright novel from gutenberg.org as your source text, rather than using Shakespeare. Be advised that the Shakespeare corpus is about 1 MB; smaller texts might not work as well.
- Adjust your hyperparameters – embedding size, head size, number of heads, number of layers – to achieve the best loss you can find while limiting training time to five minutes on Colab T4 (and document your results in a text block).
- Your code must include a generator that produces output from the trained model. 5 blocks of 1000 tokens each should be sufficient.

2 K-means clustering

Obtain the MNIST test set from `tensorflow_datasets`. Use the test set rather than the training set, simply because 10000 examples will be a little easier to work with then 60000, and we're doing unsupervised learning anyhow. We wish to minimize the K-means objective function

$J(z, \mu) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} ||x_n - \mu_k||^2$,

where z_{nk} is 1 if example n is in cluster k and 0 otherwise.

Implement a K-means algorithm function that takes a value for the number of clusters to be found (K), a set of training examples and a K-dimensional vector μ_k^0 that serves as an initial mean vector. This function should return the n-dimensional cluster assignment (presumably as an $n \times k$ one-hot matrix, since that is most convenient), as well as the converged μ_k vector. At each iteration, print a dot as a progress indicator. Once J has converged, print out its value, as well as the number of iterations it took.

Run your algorithm with K=10 (the true number of clusters) on the following intializations μ_k^0:

1. Ten data points chosen uniformly at random
2. Ten data points found using the K-means++ assignment algorithm

3. A data point drawn from each labeled class (found by looking at the test set labels – and yes, this is cheating)

Visualize the 28×28-pixel images corresponding to each cluster mean found by your algorithm, for each of these initializations.

Cluster the data using K=3, initialized using K-means++. Plot the cluster mean images and a few randomly chosen representatives from the data for each class.

3 Turning in

Submit the link to your Google Colab notebook to Canvas. This assignment is due Monday, November 18.