Lesson 13 (Cross Entropy) The cross-entropy between $\mathbf{p}$ and $\mathbf{q}$ is given by the function $H(\mathbf{p}, \mathbf{q})$ where

$$\begin{aligned} H(\mathbf{p}, \mathbf{q}) &= p_1 \ln(q_1) + \cdots + p_N \ln(q_N) \\ &= -\sum_{k=1}^{N} p_k \ln(q_k). \end{aligned}$$

(a) Define a Python function for computing the cross-entropy $H(\mathbf{p}, \mathbf{q})$ between two discrete probability distribution Use the `numpy` command `A*np.random.rand(N)` to generate random vectors of length N with entries between 0 and A. Then use a softmax function to convert these random vectors to probability distributions. Recall that as $A \to 0$, the softmax function will output nearly uniform probability distributions and as $A \to \infty$, the softmax function will output distributions approaching one-hot-encodings. Note also that the softmax function never outputs zeros.

(b) A loss function $\ell(u, v)$ is symmetric if $\ell(u, v) = \ell(v, u)$. Is $\text{MSE}(u, v)$ a symmetric loss function? Is cross-entropy a symmetric loss function?

(c) When is cross-entropy defined for one-hot-encodings and when is it undefined?

(d) Compute RMSE and cross-entropy (CE) for 1000 pairs of probability distributions $\mathbf{p}$ and $\mathbf{q}$ and construct a scatter plot of CE vs MSE. (See plotting commands below.)

```
plt.scatter(x=mse,y=ce,s=50,alpha=0.1)
plt.xlabel('MSE')
plt.ylabel('CE')
plt.xlim(xmin=mse.min(),xmax=mse.max())
plt.ylim(ymin=ce.min(),ymax=ce.max())
```

Comment on what you observe.