1. In this question, let’s analyze the loss gradients of CPC and HuBERT. Assume a simple inner product similarity metric:

\[
\text{Score}(c_t, x_t) = x_t^T c_t,
\]

where \(c_t = [c_{t,1}, \ldots, c_{t,d}]^T\) is the context representation (the output of a transformer), and \(x_t = [x_{t,1}, \ldots, x_{t,d}]^T\) is the vector being predicted. Both CPC and HuBERT use a kind of cross-entropy loss,

\[
\mathcal{L}_{CE} = - \sum_t \ln p(x_t|c_t),
\]

where \(p(x_t|c_t)\) is computed using a softmax:

\[
p(x_t|c_t) = \frac{\exp(\text{Score}(c_t, x))}{\sum_{x \in \mathcal{X}_t} \exp(\text{Score}(c_t, x))},
\]

In both CPC and HuBERT, \(x_t \in \mathcal{X}_t\), but CPC and HuBERT differ in the selection of \(x_t\) and \(\mathcal{X}_t\). In HuBERT, \(x_t\) is the codevector to which the MFCC at time \(t\) has been quantized, and \(\mathcal{X}_t\) is the set of all codevectors. In CPC, \(x_t\) is the spectrum (or MFCC, or CNN output) at time \(t\), and \(\mathcal{X}_t\) is a set of spectra sampled from different files in the same minibatch.

(a) (1 point) Find the derivative of \(\mathcal{L}_{CE}\) with respect to \(c_{t,n}\), the \(n\)th element of the Transformer output at time \(t\). Write your answer in terms of \(x_{t,n}\), and \(\mu_{t,n}\), where \(\mu_t = [\mu_{t,1}, \ldots, \mu_{t,d}]^T\) is defined in terms of the softmax outputs \(p(x_t|c_t)\) as

\[
\mu_t = \sum_{x \in \mathcal{X}_t} p(x_t|c_t)x
\]

You may or may not find it convenient to use the following form of the gradient of the log softmax:

\[
p(i|f) = \frac{\exp(f_i)}{\sum_j \exp(f_j)} \quad \Rightarrow \quad \frac{\partial(-\ln p(i|f))}{\partial f_k} = \left\{ \begin{array}{ll} p(i|f) - 1 & k = i \\ p(k|f) & k \neq i \end{array} \right.
\]

(b) (1 point) In part (a), you should have discovered that a step in the negative-gradient direction will adjust \(c_t\) toward \(x_t\), and away from \(\mu_t\). Consider the difference between CPC and HuBERT in the way that \(\mu_t\) is calculated. How do the differences between CPC and HuBERT affect each step of training? For example, is the gradient lower-dimensional for one than the other? If so, is the subspace chosen randomly, or deterministically?

2. (1 point) The paper by Polyak et al. resynthesizes speech using HiFi-GAN. In HiFi-GAN, the naturalness of speech is judged by \(J\) different discriminators, each of which is a convolutional neural net looking at a different span of speech samples (different dilations, or different durations). Why do you think HiFi-GAN uses many different discriminators, instead of just using one discriminator that takes the entire speech waveform as an input (e.g., using a deep Transformer)?