ECE 537 Practice Exam 3

UNIVERSITY OF ILLINOIS
Department of Electrical and Computer Engineering

The real exam will be December 16, 2022 in class

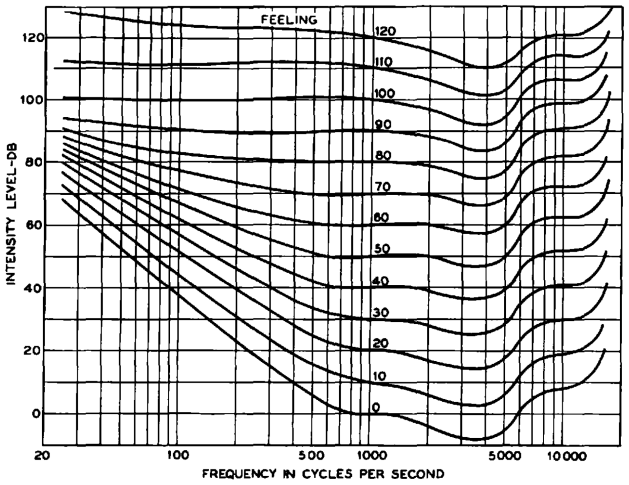
- This is a closed-book exam.
- You are allowed to bring two 8.5x11 sheets of handwritten notes (front and back).
- No calculators are allowed. Please do not simplify explicit numerical expressions.
- There are 200 points in the exam. Points for each problem are specified by the problem number.

Name: _____

NetID: _____

Possibly Useful Charts and Formulas

$$G(1000, L) = \sum_{k=1}^{n} b_k G(1000, L_k), \qquad b_k = \left[\frac{250 + \Delta f}{1000}\right] Q(L_k)$$



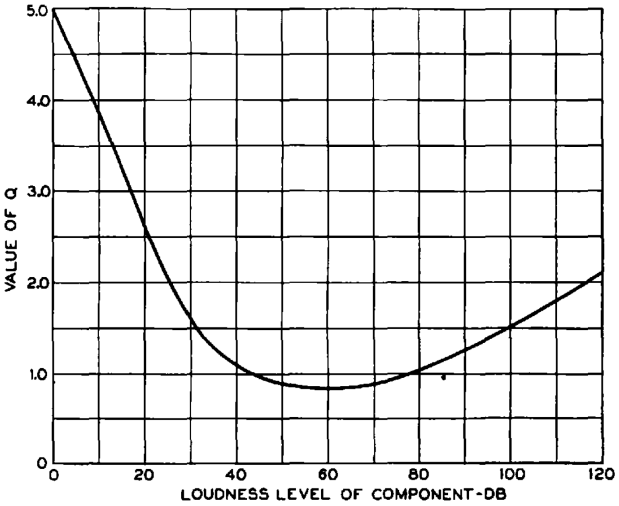


TABLE III

VALUES OF $G(L_k)$.

L	0	1	2	3	4	5	6	7	8	9
−10	0.015	0.025	0.04	0.06	0.09	0.14	0.22	0.32	0.45	0.70
0	1.00	1.40	1.90	2.51	3.40	4.43	5.70	7.08	9.00	11.2
10	13.9	17.2	21.4	26.6	32.6	39.3	47.5	57.5	69.5	82.5
20	97.5	113	131	151	173	197	222	252	287	324
30	360	405	455	505	555	615	675	740	810	890
40	975	1060	1155	1250	1360	1500	1640	1780	1920	2070
50	2200	2350	2510	2680	2880	3080	3310	3560	3820	4070
60	4350	4640	4950	5250	5560	5870	6240	6620	7020	7440
70	7950	8510	9130	9850	10600	11400	12400	13500	14600	15800
80	17100	18400	19800	21400	23100	25000	27200	29600	32200	35000
90	38000	41500	45000	49000	53000	57000	62000	67500	74000	81000
100	88000	97000	106000	116000	126000	138000	150000	164000	180000	197000
110	215000	235000	260000	288000	316000	346000	380000	418000	460000	506000
120	556000	609000	668000	732000	800000	875000	956000	1047000	1150000	1266000

Dynamic Time Warping

$$A_{i,k} = \max\left(A_{i-1,k}, A_{i,k-1}, a_{i,k} + A_{i-1,k-1}\right)$$

Linear Prediction

$$s[n] = Ge[n] + \sum_{m=1}^{p} a_m s[n-m] = h[n] * x[n]$$

$$H(z) = \frac{G}{1 - \sum_{m=1}^{N} a_m z^{-m}} = \frac{G}{\prod_{k=1}^{N} (1 - p_k z^{-1})}$$

$$\mathcal{E} = \sum_{n=0}^{N-1} e^2[n] = \sum_{n=0}^{N-1} \left(s[n] - \sum_{m=1}^{p} a_m s[n-m]\right)^2$$

$$0 = \sum_{n=0}^{N-1} \left(s[n] - \sum_{m=1}^{p} a_m s[n-m]\right) s[n-k], \quad 1 \le k \le p$$

$$\vec{c} = \Phi \vec{a}$$

Hidden Markov Models

$$\tilde{\alpha}_t(j) = \sum_{i=1}^{N} \hat{\alpha}_{t-1}(i) a_{ij} b_j(\vec{x}_t), \qquad \hat{\alpha}_t(j) = \frac{1}{\sum_{j=1}^{N} \tilde{\alpha}_t(j)}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\vec{o}_{t+1}) \beta_{t+1}(j), \quad 1 \leq i \leq N,\ 1 \leq t \leq T-1$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{j=1}^{N} \sum_{t=1}^{T-1} \xi_t(i,j)}, \quad \bar{\mu}_i = \frac{\sum_{t=1}^{T} \gamma_t(i) \vec{o}_t}{\sum_{t=1}^{T} \gamma_t(i)}$$

Formant Synthesis

$$c_k = -e^{-2\pi B_k T}, \qquad b_k = 2e^{-\pi B_k T} \cos(2\pi F_k T), \qquad a_k = 1 - b_k - c_k$$

Softmax

$$p(i|f) = \frac{\exp(f_i)}{\sum_j \exp(f_j)} \quad \Rightarrow \quad \frac{\partial(-\ln p(i|f))}{\partial f_k} = \begin{cases} p(i|f) - 1 & k = i \\ p(k|f) & k \neq i \end{cases}$$

CTC

$$\frac{d\mathcal{L}}{dy_k^\tau} = -\frac{\gamma_\tau(k)}{y_k^\tau}, \qquad \gamma_\tau(k) = p(\pi_\tau = k, \ell|\mathbf{x}) = \frac{1}{y_k^\tau} \sum_{s:\ell'_s=k} \alpha_\tau({\ell'}_{1:s}) \beta_\tau({\ell'}_{s:|\ell'|})$$

$$\beta_\tau({\ell'}_{s:|\ell'|}) = y_{\ell'_s}^\tau \left(\beta_{\tau+1}({\ell'}_{s:|\ell'|}) + \beta_{\tau+1}({\ell'}_{(s+1):|\ell'|}) + \beta_{\tau+1}({\ell'}_{(s+2):|\ell'|}) \left[\ell'_s \neq - \wedge \ell'_s \neq \ell'_{s+2} \right] \right)$$

Transformer

$$\begin{aligned} \text{Attention}(Q,K,V) &= \text{softmax}(QK^T)V \\ \text{head}_i &= \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \\ \text{MultiHead}(Q,K,V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \end{aligned}$$

Self-Supervised

$$\begin{aligned} \mathcal{L}_{\text{CPC}} &= -\sum_t \ln \frac{\exp\left(\text{Score}(x_{t+k}, c_t)\right)}{\sum_{x \in X} \exp\left(\text{Score}(x, c_t)\right)} \\ \mathcal{L}_{\text{HuBERT}} &= -\sum_{t \in M} \ln \frac{\exp\left(\text{Score}(Ao_t, e_c)\right)}{\sum_{c'=1}^{C} \exp\left(\text{Score}(Ao_t, e_{c'})\right)} \\ \nabla_{z_e(x)} \mathcal{L}_{\text{VQVAE}} &\approx \nabla_{z_q(x)} \mathcal{L}_{\text{VQVAE}} \end{aligned}$$

Speech Resynthesis

$$y_i = \sum_{j=-D}^{D} K_j h_{i+j}, \quad [h_1, \ldots, h_{2L+1}] = [0, z_1, 0, z_2, 0, \ldots, 0, z_L, 0]$$

$$\mathcal{L}_G(D,G) = \sum_{j=1}^{J} \left[\mathcal{L}_{\text{adv}}(G, D_j) + \lambda_{fm} \mathcal{L}_{fm}(G, D_j) \right] + \lambda_r \mathcal{L}_{\text{recon}}(G)$$

1. (30 points) A particular signal contains component tones at 150, 300, 1050, and 1200Hz. The levels of these four components are 70, 70, 45, and 50dB SPL, respectively.

   (a) Find the loudness levels of all four components.

   Solution: The loudness levels of these four tones are 60, 68, 45, and 50 phons, respectively.

   (b) Find the loudness of each of these four components.

   Solution: The loudnesses are 4350, 7020, 1500, and 2200 sones, respectively.

   (c) What is the total loudness of this signal?

   Solution: The masking coefficients are 1, $0.8\left(\frac{400}{1000}\right)$, 1, and $0.9\left(\frac{400}{1000}\right)$. The total loudness of this signal is $4350 + 0.8\left(\frac{400}{1000}\right) 7020 + 1500 + 0.9\left(\frac{400}{1000}\right) 2200$.

(d) Suppose that the correlogram of this signal is computed using critical bandwidths $B(f)$ given by the following equation, where $B(f)$ and f are both in Hertz:

$$B(f) = \begin{cases} 100 & 0 \le f \le 1000 \\ 200 & 1000 < f \le 2000 \\ 300 & 2000 < f \le 3000 \\ 400 & 3000 < f \le 4000 \end{cases}$$

What is the correlogram of this signal, as a function of the filter center frequency f and autocorrelation delay τ?

Solution: OK, let's first find the amplitudes of each tone. Those would be

$$\begin{aligned} A_1 &= A_{ref} 10^{70/20} \\ A_2 &= A_{ref} 10^{70/20} \\ A_3 &= A_{ref} 10^{45/20} \\ A_4 &= A_{ref} 10^{50/20}, \end{aligned}$$

where A_{ref} is whatever amplitude corresponds to 1dB. The autocorrelation of $A\cos(\omega t + \theta)$ is $\frac{A^2}{2}\cos(\omega\tau)$, so

$$\phi(f,\tau) = \begin{cases} 0 & f < 50 \\ \frac{A_1^2}{2}\cos(2\pi 150\tau) & 100 < f < 200 \\ 0 & 200 < f < 250 \\ \frac{A_2^2}{2}\cos(2\pi 300\tau) & 250 < f < 350 \\ 0 & 350 < f < 1000 \\ \frac{A_3^2}{2}\cos(2\pi 1050\tau) & 1000 < f < 1100 \\ \frac{A_3^2}{2}\cos(2\pi 1050\tau) + \frac{A_4^2}{2}\cos(2\pi 1200\tau) & 1100 < f < 1150 \\ \frac{A_4^2}{2}\cos(2\pi 1200\tau) & 1150 < f < 1300 \\ 0 & 1400 < f \end{cases}$$

2. (20 points) Consider an LPC-based speech synthesizer with no pitch prediction; thus the speech signal $s_k[n]$ is generated from an excitation signal $e_k[n]$ using only

$$s_k[n] = e_k[n] + \sum_{m=1}^{p} a_m s_k[n-m], \tag{1}$$

where a_m are the linear prediction coefficients. Note that Eq. (1) can also be written as

$$S_k(z) = \frac{1}{1 - P(z)} E_k(z)$$
$$P(z) = \sum_{m=1}^{p} a_m z^{-m}$$

Suppose that we wish to exhaustively test K different candidate excitations, $e_k[n]$, for $1 \leq k \leq K$. We want to choose the excitation that minimizes the perceptually weighted error, $\mathcal{E}_k$, defined as

$$\mathcal{E}_k = \sum_{n=0}^{N-1} y_k^2[n],$$

where

$$Y_k(z) = \frac{1 - P(z)}{1 - P(z/\alpha)} S_k(z),$$

Demonstrate that $y_k[n]$ can be generated from $e_k[n]$ using only p multiplications per sample.

Solution:

$$Y_k(z) = \frac{1}{1 - P(z/\alpha)} E_k(z)$$
$$P(z/\alpha) = \sum_{m=1}^{p} a_m \alpha^m z^{-m}$$

Therefore, if we compute the coefficients $c_m = \alpha^m a_m$ once per frame, we can then compute all of the frame's N samples using

$$y_k[n] = e_k[n] + \sum_{m=1}^{p} c_m y_k[n-m]$$

3. (20 points) The scaling constant, in the standard scaled-forward algorithm, can be interpreted as

$$c_t = P(o_t|o_1, \ldots, o_{t-1}, \lambda)$$

This is an intriguing quantity; it suggests that we are predicting the next spectrum, given the previous spectra. Suppose that somebody else has provided you with a table of the non-scaled forward probabilities for a particular waveform,

$$\alpha_t(i) = P(q_t = i, o_1, \ldots, o_t|\lambda)$$

Is it possible to compute c_T for the last frame without computing the scaled forward algorithm for all time steps? In other words, can you come up with a formula for c_T in terms of $\alpha_t(i)$, $a_{i,j}$, and $b_i(k)$, for some appropriate values of i, j, t, k, but without computing the scaled forward algorithm for all time steps?

Solution: First, we want a probability conditioned on $o_1, \ldots, o_{t-1}$. We can get that by normalizing $\alpha_{t-1}(i)$:

$$P(q_{t-1} = i|o_1, \ldots, o_{t-1}, \lambda) = \frac{\alpha_{t-1}(i)}{\sum_{j=1}^{N} \alpha_{t-1}(j)}$$

Then we can find the probability of o_t given $o_1, \ldots, o_{t-1}$ by summing over all of the ways in which o_t could have been made:

$$\begin{aligned} P(o_t|o_1, \ldots, o_{t-1}, \lambda) &= \sum_{i=1}^{N} \sum_{j=1}^{N} P(q_{t-1} = i|o_1, \ldots, o_{t-1}, \lambda) a_{ij} b_j(o_t) \\ &= \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)}{\sum_{j=1}^{N} \alpha_{t-1}(j)} \\ &= \frac{\sum_{j=1}^{N} \alpha_t(j)}{\sum_{j=1}^{N} \alpha_{t-1}(j)} \end{aligned}$$

4. (40 points) A neural net can be used to transform, in frame n, a source speaker's vector of formant frequencies, $s_n = [s_{n,1}, \ldots, s_{n,3}]^T$, to $\hat{t}_n$, an estimate of the target speaker's vector of formant frequencies, $t_n = [t_{n,1}, \ldots, t_{n,3}]^T$. This can be done using a two-layer fully-connected network of the form

$$h_n = \tanh(W_1 s_n)$$
$$\hat{t}_n = W_2 h_n$$

The weight matrices W_1 and W_2 are trained to minimize

$$\mathcal{L} = \frac{1}{2} \sum_n \|t_n - \hat{t}_n\|_2^2$$

Note that, for scalar $y = \tanh(x)$, $\frac{\partial y}{\partial x} = 1 - y^2$.

(a) Formant frequency measurements are sometimes erroneous. The effect of a formant frequency error on training can be estimated by finding the sensitivity of the loss gradient to the formant. Let $W_{\ell,i,j}$ be the $(i,j)^{\text{th}}$ element of W_ℓ; the sensitivity of $\frac{\partial \mathcal{L}}{\partial W_{\ell,i,j}}$ to an error in the measurement of $s_{n,k}$ can be estimated as $\frac{\partial}{\partial s_{n,k}} \left(\frac{\partial \mathcal{L}}{\partial W_{\ell,i,j}} \right)$. Find the sensitivity of the layer-2 gradient, i.e., find

$$\frac{\partial}{\partial s_{n,k}} \left(\frac{\partial \mathcal{L}}{\partial W_{2,i,j}} \right)$$

Your answer need not be simplified, but it should contain no unresolved derivatives.

Solution:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial W_{2,i,j}} &= \sum_n \frac{\partial \mathcal{L}}{\partial \hat{t}_{n,i}} \frac{\partial \hat{t}_{n,i}}{\partial W_{2,i,j}} \\ &= \sum_n (\hat{t}_{n,i} - t_{n,i}) h_{n,j} \end{aligned}$$

This depends on $s_{n,k}$ in two ways: both $\hat{t}_{n,i}$ and $h_{n,j}$ depend on $s_{n,k}$.

$$\begin{aligned} \frac{\partial}{\partial s_{n,k}} \left(\frac{\partial \mathcal{L}}{\partial W_{2,i,j}} \right) &= \sum_n \left(\frac{\partial \hat{t}_{n,i}}{\partial s_{n,k}} h_{n,j} + (\hat{t}_{n,i} - t_{n,i}) \frac{\partial h_{n,j}}{\partial s_{n,k}} \right) \\ &= \sum_n \left(h_{n,j} \sum_\ell \frac{\partial \hat{t}_{n,i}}{\partial h_{n,\ell}} \frac{\partial h_{n,\ell}}{\partial s_{n,k}} + (\hat{t}_{n,i} - t_{n,i}) \frac{\partial h_{n,j}}{\partial s_{n,k}} \right) \\ &= \sum_n \left(h_{n,j} \sum_\ell W_{2,i,\ell} (1 - h_{n,\ell}^2) W_{1,\ell,k} + (\hat{t}_{n,i} - t_{n,i}) W_{2,i,j} (1 - h_{n,j}^2) W_{1,j,k} \right) \end{aligned}$$

(b) During test time, errors in measuring s_n can cause errors in the estimated target, $\hat{t}_n$, which, in turn, cause errors in the synthesis filter. Let the synthesis filter for the k^{th} formant at frame n be

$$R_{n,k}(z) = \frac{a_{n,k}}{1 - b_{n,k}z^{-1} - c_{n,k}z^{-2}},$$

where $b_{n,k}$ and $c_{n,k}$ are computed so that $R_{n,k}(z)$ is a resonator at a frequency of $\hat{t}_{n,k}$ Hertz, with a bandwidth of B_k Hertz. Assume that the bandwidths, B_k, are fixed, and that only the formant frequencies, $\hat{t}_{n,k}$, are computed at the output of the neural network. How much does an error in the estimation of $s_{n,j}$ affect the filter coefficient $b_{n,k}$? In other words, what is $\frac{\partial b_{n,k}}{\partial s_{n,j}}$? Your answer will be a function of the variables provided, and of the sampling period T.

Solution:

$$\begin{aligned}
b_{n,k} &= 2e^{-\pi B_k T}\cos(2\pi \hat{t}_{n,k} T) \\
\frac{\partial b_{n_k}}{\partial s_{n,j}} &= -4\pi T e^{-\pi B_k T}\sin(2\pi \hat{t}_{n,k} T)\frac{\partial \hat{t}_{n,k}}{\partial s_{n,j}} \\
&= -4\pi T e^{-\pi B_k T}\sin(2\pi \hat{t}_{n,k} T)\sum_i \frac{\partial \hat{t}_{n,k}}{\partial h_{n,i}}\frac{\partial h_{n,i}}{\partial s_{n,j}} \\
&= -4\pi T e^{-\pi B_k T}\sin(2\pi \hat{t}_{n,k} T)\sum_i W_{2,k,i}(1 - h_{n,i}^2)W_{1,i,j}
\end{aligned}$$

5. (40 points) Consider an RNN with

$$h_n = \tanh\left(Ux_n + Vh_{n-1}\right) = [h_{n,1}, \ldots, h_{n,d}]^T$$
$$y_n = \text{softmax}\left(Wh_n\right) = [y_{n,1}, \ldots, y_{n,c}]^T$$

Suppose the output is scored using CTC, i.e.,

$$\mathcal{L} = -\ln \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{z})} \prod_{n=1}^{N} y_{n,\pi_n}$$

Let $\ell' = [\ell'_1, \ldots, \ell'_{2L+1}] = [-, z_1, -, z_2, \cdots, z_L, -]$ be the blank-expansion of $\mathbf{z} = [z_1, \ldots, z_L]$.

(a) Assume ℓ'_i is a character that occurs nowhere else in ℓ', i.e., $\forall j \neq i$, $\ell'_j \neq \ell'_i$. What is $\frac{\partial \mathcal{L}}{\partial y_{n,\ell'_i}}$? If you need auxiliary variables, define them clearly.

Solution: There are at least two very different-looking solutions to this problem, but I think they are actually the same, even though they look different.

1. By directly differentiating, you get

$$\frac{\partial \mathcal{L}}{\partial y_{n,\ell'_i}} = -\frac{1}{\sum_{\pi \in \mathcal{B}^{-1}(\mathbf{z})} \prod_{n=1}^{N} y_{n,\pi_n}} \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{z}) \text{ and } \pi_n = \ell'_i} \prod_{m:m \neq n} y_{m,\pi_n}$$

2. You can get an equivalent but very different-looking solution by copying the answer down from the formula sheet, and appropriately modifying the notation.

$$\frac{\partial \mathcal{L}}{\partial y_{n,\ell'_i}} = \frac{\gamma_n(\ell'_i)}{y_{n,\ell'_i}},$$

where $\gamma_n(\ell'_i)$ is defined to be

$$\gamma_n(\ell'_i) = \frac{1}{y_{n,\ell'_i}} \sum_{s:\ell'_s = \ell'_i} \alpha(\ell'_{1:s})\beta(\ell'_{s:|\ell'|}) \qquad = \frac{1}{y_{n,\ell'_i}} \alpha(\ell'_{1:s})\beta(\ell'_{s:|\ell'|}),$$

where α and β are defined by

$$\beta_\tau(\ell'_{s:|\ell'|}) = y^\tau_{\ell'_s}\left(\beta_{\tau+1}(\ell'_{s:|\ell'|}) + \beta_{\tau+1}(\ell'_{(s+1):|\ell'|}) + \beta_{\tau+1}(\ell'_{(s+2):|\ell'|})\left[\ell'_s \neq - \wedge \ell'_s \neq \ell'_{s+2}\right]\right)$$
$$\alpha_\tau(\ell'_{s:|\ell'|}) = y^\tau_{\ell'_s}\left(\alpha_{\tau-1}(\ell'_{1:s}) + \alpha_{\tau-1}(\ell'_{1:(s-1)}) + \alpha_{\tau-1}(\ell'_{1:(s-2)})\left[\ell'_s \neq - \wedge \ell'_s \neq \ell'_{s-2}\right]\right)$$
$$\beta_N(\ell) = \begin{cases} y_{N,-} & \ell = [-] \\ y_{N,\ell'_{2L}} & \ell = [\ell'_{2L}, -] \\ 0 & \text{otherwise} \end{cases}$$
$$\alpha_1(\ell) = \begin{cases} y_{1,-} & \ell = [-] \\ y_{1,\ell'_2} & \ell = [-, \ell'_2] \\ 0 & \text{otherwise} \end{cases}$$

(b) Suppose you have already computed $\frac{\partial \mathcal{L}}{\partial y_{n,k}}$ for a particular n, and for all $k \in \{1, \ldots, c\}$. Suppose you have also already computed $\frac{\partial \mathcal{L}}{\partial h_{n+1,j}}$ for the same n, and for all $j \in \{1, \ldots, d\}$. In terms of $\frac{\partial \mathcal{L}}{\partial y_{n,k}}$ and $\frac{\partial \mathcal{L}}{\partial h_{n+1,j}}$, what is $\frac{\partial \mathcal{L}}{\partial h_{n,i}}$, as a function of i, for all $i \in \{1, \ldots, d\}$?

Solution:

$$\frac{\partial \mathcal{L}}{\partial h_{n,i}} = \sum_{j=1}^{d} \frac{\partial \mathcal{L}}{\partial h_{n+1,j}} \frac{\partial h_{n+1,j}}{\partial h_{n,i}} + \sum_{k=1}^{c} \frac{\partial \mathcal{L}}{\partial y_{n,k}} \frac{\partial y_{n,k}}{\partial h_{n,i}}$$

Now,

$$h_{n+1,j} = \tanh_j \left(U x_{n+1} + V h_n \right),$$

so,

$$\frac{\partial h_{n+1,j}}{\partial h_{n,i}} = (1 - h_{n+1,j}^2) V_{j,i}$$

Similarly, but somewhat more challenging,

$$y_{n,k} = \frac{\exp(W_{k,:} h_n)}{\sum_{\ell=1}^{c} \exp(W_{\ell,:} h_n)},$$

therefore

$$\begin{aligned} \frac{\partial y_{n,k}}{\partial h_{n,i}} &= \frac{\exp(W_{k,:} h_n)}{\sum_{\ell=1}^{c} \exp(W_{\ell,:} h_n)} W_{k,i} - \sum_{m=1}^{c} \frac{\exp(W_{k,:} h_n) \exp(W_{m,:} h_n}{(\sum_{\ell=1}^{c} \exp(W_{\ell,:} h_n))^2} W_{m,i} \\ &= y_{n,k} \left(W_{k,i} - \sum_{m=1}^{c} y_{n,m} W_{m,i} \right) \end{aligned}$$

Putting those together, we get that

$$\frac{\partial \mathcal{L}}{\partial h_{n,i}} = \sum_{j=1}^{d} \frac{\partial \mathcal{L}}{\partial h_{n+1,j}} (1 - h_{n+1,j}^2) V_{i,j} + \sum_{k=1}^{c} \frac{\partial \mathcal{L}}{\partial y_{n,k}} y_{n,k} \left(W_{k,i} - \sum_{m=1}^{c} y_{n,m} W_{m,i} \right)$$

6. (25 points) Suppose

$$Y = \text{softmax}(QK^T)V$$

Let $Y_{i,j}$, $Q_{i,j}$, $K_{i,j}$, and $V_{i,j}$ be the $(i,j)^{\text{th}}$ elements of the matrices Y, Q, K and V, respectively. What is $\frac{\partial Y_{i,j}}{\partial K_{k,\ell}}$? If you need any intermediate variables, please define them.

Solution: This problem is actually much harder than I thought it was when I designed it. First, let's define the attention matrix:

$$\begin{aligned}
A &= \text{softmax}(QK^T) \\
A_{i,j} &= \frac{\exp(\sum_m Q_{i,m}K_{j,m})}{\sum_n \exp(\sum_m Q_{i,m}K_{n,m})} \\
Y &= AV \\
Y_{i,j} &= \sum_p A_{i,p}V_{p,j} \\
\frac{\partial Y_{i,j}}{\partial K_{k,\ell}} &= \sum_p \frac{\partial A_{i,p}}{\partial K_{k,\ell}} V_{p,j}
\end{aligned}$$

And now the hard part:

$$\begin{aligned}
\frac{\partial A_{i,p}}{\partial K_{k,\ell}} &= \frac{\exp(\sum_m Q_{i,m}K_{p,m})}{\sum_n \exp(\sum_m Q_{i,m}K_{n,m})} \frac{\partial \sum_m Q_{i,m}K_{p,m}}{\partial K_{k,\ell}} \\
&\quad - \sum_q \frac{\exp(\sum_m Q_{i,m}K_{p,m}) \exp(\sum_m Q_{i,m}K_{q,m})}{(\sum_n \exp(\sum_m Q_{i,m}K_{n,m}))^2} \frac{\partial \sum_m Q_{i,m}K_{q,m}}{\partial K_{k,\ell}} \\
&= \begin{cases} A_{i,k}(1 - A_{i,k})Q_{i,\ell} & p = k \\ -A_{i,p}A_{i,k}Q_{i,\ell} & \text{otherwise} \end{cases}
\end{aligned}$$

Putting it together, we have:

$$\frac{\partial Y_{i,j}}{\partial K_{k,\ell}} = A_{i,k}Q_{i,\ell}\left(V_{k,j} - \sum_p A_{i,p}V_{p,j}\right)$$

7. (25 points) The HuBERT loss function is

$$\mathcal{L} = -\sum_n \ln \frac{\exp(c_n^T e_n)}{\sum_{e' \in \mathcal{E}} \exp(c_n^T e')},$$

where c_n is the transformer output at time n, e_n is the codevector corresponding to the input spectrum at time n, and $\mathcal{E}$ is the set of all codevectors. The codevectors are not usually trained from data, but it would be possible to train them from data. For example, we could use the following gradient descent algorithm:

$$v \leftarrow v - \eta \nabla_v \mathcal{L} \quad \forall v \in \mathcal{E} \tag{2}$$

where η is a learning rate. Prove that the gradient descent step shown in Eq. (2) moves v toward the Transformer outputs, c_n, of all of the frames for which $e_n = v$, and moves v away from c_m in all frames for which $e_m \neq v$.

Solution:

$$\begin{aligned}
\nabla_v \mathcal{L} &= -\sum_n \nabla_v \ln \frac{\exp(c_n^T e_n)}{\sum_{e' \in \mathcal{E}} \exp(c_n^T e')} \\
&= \sum_{n: e_n = v} \left(\frac{\exp(c_n^T e_n)}{\sum_{e' \in \mathcal{E}} \exp(c_n^T e')} - 1 \right) c_n + \sum_{m: e_m \neq v} \left(\frac{\exp(c_m^T e_m)}{\sum_{e' \in \mathcal{E}} \exp(c_m^T e')} \right) c_n
\end{aligned}$$

The CPC gradient descent step would then be

$$v \leftarrow v + \sum_{n: e_n = v} \left(1 - \frac{\exp(c_n^T e_n)}{\sum_{e' \in \mathcal{E}} \exp(c_n^T e')} \right) c_n - \sum_{m: e_m \neq v} \left(\frac{\exp(c_m^T e_m)}{\sum_{e' \in \mathcal{E}} \exp(c_m^T e')} \right) c_n$$

The factor multiplying each c_n for which $e_n = v$ is a positive number, since the softmax is always less than 1. The factor multiplying each c_m for which $e_m \neq v$ is a negative number, since the softmax is always greater than 0.