UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
Department of Electrical and Computer Engineering

ECE 417 Multimedia Signal Processing
Fall 2023

PRACTICE EXAM 3

Exam will be Friday, December 8, 2023, 1:30-4:30pm

- This will be a CLOSED BOOK exam.
- You will be permitted three sheets of handwritten notes, 8.5x11.
- Calculators and computers are not permitted.
- Don't simplify explicit numerical expressions.
- If you're taking the exam online, you will need to have your webcam turned on. Your exam will appear on Gradescope at exactly 1:30pm; you will need to photograph and upload your answers by exactly 4:30pm.
- There will be a total of 200 points in the exam. Each problem specifies its point total. Plan your work accordingly.
- You must SHOW YOUR WORK to get full credit.
- This practice exam contains only material from the last third of the course. The actual exam will contain 17% material from the first third, and 17% material from the second third, and 66% material from the last third of the course, of the course.

Name: _____

NetID: _____

Linear Algebra: If $\mathbf{A}$ is tall and thin, with full column rank, then

$$\mathbf{A}^{\dagger}\mathbf{b} = \text{argmin}_{\mathbf{v}}\|\mathbf{b} - \mathbf{A}\mathbf{v}\|^2 = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

If $\mathbf{A}$ is short and fat, with full row rank, then

$$\mathbf{A}^{\dagger} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$$

Orthogonal projection of $\mathbf{x}$ onto the columns of $\mathbf{A}$ is $\mathbf{x}_{\perp} = \mathbf{A}\mathbf{A}^{\dagger}\mathbf{x}$. Orthogonal projection onto the rows of $\mathbf{A}$ is $\mathbf{x}_{\perp} = \mathbf{A}^{\dagger}\mathbf{A}\mathbf{x}$.

Image Interpolation

$$y[n_1, n_2] = \begin{cases} x\left[\frac{n_1}{U}, \frac{n_2}{U}\right] & \frac{n_1}{U}, \frac{n_2}{U} \text{ both integers} \\ 0 & \text{otherwise} \end{cases}, \qquad z[n_1, n_2] = h[n_1, n_2] * y[n_1, n_2]$$

$$h_{\text{rect}}[n_1, n_2] = \begin{cases} 1 & 0 \le n_1, n_2 < U \\ 0 & \text{otherwise} \end{cases}, \qquad h_{\text{tri}}[n] = \begin{cases} \left(1 - \frac{|n_1|}{U}\right)\left(1 - \frac{|n_2|}{U}\right) & -U \le n_1, n_2 \le U \\ 0 & \text{otherwise} \end{cases}$$

$$h_{\text{sinc}}[n_1, n_2] = \frac{sin(\pi n_1/U)}{\pi n_1/U}\frac{sin(\pi n_2/U)}{\pi n_2/U}$$

Barycentric Coordinates

$$\begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = \beta_1\mathbf{x}_1 + \beta_2\mathbf{x}_2 + \beta_3\mathbf{x}_3 = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}$$

DTFT, DFT, STFT, Griffin-Lim

$$X(\omega) = \sum_n x[n]e^{-j\omega n}, \qquad x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{j\omega n}d\omega$$

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi kn}{N}}, \qquad x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi kn}{N}}$$

$$X_m(\omega) = \sum_n w[n-m]x[n]e^{-j\omega(n-m)}, \quad x[n] = \frac{\sum_m \frac{1}{N}\sum_{k=0}^{N-1} X_m\left(\frac{2\pi k}{N}\right)e^{j\frac{2\pi k(n-m)}{N}}}{\sum_m w[n-m]}$$

$$X_t(\omega) \leftarrow \text{STFT}\left\{\text{ISTFT}\left\{X_t(\omega)\right\}\right\}, \qquad X_t(\omega) \leftarrow M_t(\omega)e^{j\angle X_t(\omega)}$$

Neural Nets

$$h_{i,k}^{(\ell)} = g(z_{i,k}^{(\ell)}), \qquad z_{i,k}^{(\ell)} = b_k^{(\ell)} + \sum_{j=1}^{p} w_{k,j}^{(\ell)}h_{i,j}^{(\ell-1)}$$

$$\frac{d\mathcal{L}}{dz_{i,k}^{(\ell)}} = \frac{d\mathcal{L}}{dh_{i,k}^{(\ell)}}\dot{g}(z_{i,k}^{(\ell)}), \qquad \dot{\sigma}(x) = \sigma(x)(1 - \sigma(x))$$

$$\frac{d\mathcal{L}}{dh_{i,j}^{(\ell-1)}} = \sum_k \frac{d\mathcal{L}}{dz_{i,k}^{(\ell)}}w_{k,j}^{(\ell)}, \qquad \frac{d\mathcal{L}}{dw_{k,j}^{(\ell)}} = \sum_i \frac{d\mathcal{L}}{dz_{i,k}^{(\ell)}}h_{i,j}^{(\ell-1)}$$

Viola-Jones

$$II[m,n] = \sum_{m'=1}^{m} \sum_{n'=1^n} I[m',n'], \quad 1 \le m \le M, 1 \le n \le N$$

$$\epsilon_t = \sum_i w_t(x_i)|y_i - h_t(x_i)|, \quad w_{t+1}(x_i) = \beta_t w_t(x_i), \quad \beta_t = \frac{\epsilon_t}{1-\epsilon_t}$$

$$h(x) = \begin{cases} 1 & \sum_t \alpha_t h_t(x) > \frac{1}{2}\sum_t \alpha_t \\ 0 & \text{otherwise} \end{cases}, \quad \alpha_t = -\ln \beta_t$$

Hidden Markov Model

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{i,j} b_j(\boldsymbol{x}_t), \quad 1 \le j \le N, \ 2 \le t \le T, \quad \hat{\alpha}_t(j) = \frac{\sum_{i=1}^{N} \hat{\alpha}_{t-1}(i) a_{i,j} b_j(\boldsymbol{x}_t)}{\sum_{j'=1}^{N}\sum_{i=1}^{N} \hat{\alpha}_{t-1}(i) a_{i,j'} b_{j'}(\boldsymbol{x}_t)}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{i,j} b_j(\boldsymbol{x}_{t+1}) \beta_{t+1}(j), \quad 1 \le i \le N, \ 1 \le t \le T-1$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{k=1}^{N} \alpha_t(k)\beta_t(k)}, \quad \xi_t(i,j) = \frac{\alpha_t(i) a_{i,j} b_j(\boldsymbol{x}_{t+1}) \beta_{t+1}(j)}{\sum_{k=1}^{N}\sum_{\ell=1}^{N} \alpha_t(k) a_{k,\ell} b_\ell(\boldsymbol{x}_{t+1}) \beta_{t+1}(\ell)}$$

$$a'_{i,j} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{j=1}^{N}\sum_{t=1}^{T-1} \xi_t(i,j)}, \quad b'_j(k) = \frac{\sum_{t:x_t=k} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Gaussians

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

$$\boldsymbol{\mu}'_i = \frac{\sum_{t=1}^{T} \gamma_t(i)\boldsymbol{x}_t}{\sum_{t=1}^{T} \gamma_t(i)}, \quad \boldsymbol{\Sigma}'_i = \frac{\sum_{t=1}^{T} \gamma_t(i)(\boldsymbol{x}_t - \boldsymbol{\mu}_i)(\boldsymbol{x}_t - \boldsymbol{\mu}_i)^T}{\sum_{t=1}^{T} \gamma_t(i)}$$

PCA

$$\boldsymbol{X} = [\boldsymbol{x}_1 - \boldsymbol{\mu}, \ldots, \boldsymbol{x}_M - \boldsymbol{\mu}], \quad \boldsymbol{\Sigma} = \frac{1}{M-1}\boldsymbol{X}\boldsymbol{X}^T$$

$$\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T, \ \ \boldsymbol{U}^T\boldsymbol{\Sigma}\boldsymbol{U} = \boldsymbol{\Lambda}, \ \ \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}, \ \ \boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Lambda}^{1/2}\boldsymbol{V}^T$$

RNN and LSTM

$$\frac{d\mathcal{L}}{dh[n]} = \frac{\partial \mathcal{L}}{\partial h[n]} + \sum_m \frac{d\mathcal{L}}{dh[n+m]} \frac{\partial h[n+m]}{\partial h[n]}$$

$$i[t] = \sigma_g(w_i x[t] + u_i h[t-1] + b_i), \quad o[t] = \sigma_g(w_o x[t] + u_o h[t-1] + b_o), \quad f[t] = \sigma_g(w_f x[t] + u_f h[t-1] + b_f)$$

$$c[t] = f[t]c[t-1] + i[t]\sigma_h\left(w_c x[t] + u_c h[t-1] + b_c\right), \quad h[t] = o[t]\sigma_h(c[t])$$

1. (10 points) A particular dataset has the scatter matrix $\boldsymbol{S} = \sum_{k=1}^{n}(\boldsymbol{x}_k - \boldsymbol{m})(\boldsymbol{x}_k - \boldsymbol{m})^T$, whose first two eigenvectors are $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, characterized by eigenvalues $\lambda_1 = 450$ and $\lambda_2 = 150$. Define the transform $\boldsymbol{y}_k = [\boldsymbol{v}_1, \boldsymbol{v}_2]^T(\boldsymbol{x}_k - \boldsymbol{m})$. Define the 2×2 matrix

$$\boldsymbol{Q} = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} = \sum_{k=1}^{n} \boldsymbol{y}_k \boldsymbol{y}_k^T$$

Find the numerical values of the elements q_{11}, q_{12}, q_{21}, and q_{22} of matrix Q.

2. (10 points) A particular dataset has six data vectors, given by

$$\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_6\} = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \right\}$$

By calling `np.random.randn`, you generate a 3×2 random projection matrix $\boldsymbol{V}$, given by

$$\boldsymbol{V} = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \\ v_{31} & v_{32} \end{bmatrix}$$

Using this random projection matrix, you compute the transformed feature vectors $\boldsymbol{y}_k = \boldsymbol{V}^T \boldsymbol{x}_k$. The total energy of the transformed dataset can be written as

$$E = \sum_{k=1}^{6} \boldsymbol{y}_k^T \boldsymbol{y}_k$$

Find the value of E in terms of the random projection matrix elements v_{ij}.

3. (10 points) A particular dataset has three data,

$$\boldsymbol{x}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{x}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}, \quad \boldsymbol{x}_3 = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Define $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3]$ and $\boldsymbol{R} = \boldsymbol{X}^T\boldsymbol{X}$. The matrix $\boldsymbol{R}$ is given by $\boldsymbol{R} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ where

$$\boldsymbol{V} = \begin{bmatrix} -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ \frac{2}{\sqrt{6}} & 0 \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \end{bmatrix}, \quad \boldsymbol{\Lambda} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$$

Find a matrix $\boldsymbol{W}$ such that $\boldsymbol{y}_i = W^T\boldsymbol{x}_i$, $\boldsymbol{y}_i$ is two-dimensional, and the elements of $\boldsymbol{y}_i$ are uncorrelated.

4. (10 points) As you know, in any given vector space, it's possible to define an infinite number of different Mahalanobis distances, parameterized by different covariance matrices. Consider the Mahalanobis distance measures $d_a(\vec{x}, \vec{y})$ and $d_b(\vec{x}, \vec{y})$, parameterized, respectively, by the covariance matrices

$$\mathbf{\Sigma}_a = \begin{bmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & 0 & \vdots \\ \vdots & 0 & \dots & 0 \\ 0 & \dots & 0 & a_d \end{bmatrix}, \quad \mathbf{\Sigma}_b = \begin{bmatrix} b_1 & 0 & \dots & 0 \\ 0 & b_2 & 0 & \vdots \\ \vdots & 0 & \dots & 0 \\ 0 & \dots & 0 & b_d \end{bmatrix}$$

Your friend Amit wishes to define a third dissimilarity measure, as

$$d_c(\vec{x}, \vec{y}) = \sqrt{\frac{1}{2}\left(d_a^2(\vec{x}, \vec{y}) + d_b^2(\vec{x}, \vec{y})\right)}$$

Is $d_c(\vec{x}, \vec{y})$ a Mahalanobis distance? If so, find the elements of the covariance matrix $\mathbf{\Sigma}_c$ in terms of the variables $a_1, \dots, a_d, b_1, \dots, b_d$. If not, demonstrate that no such covariance matrix exists.

5. (10 points) Suppose that you have M different D-dimensional vectorized face images, $\boldsymbol{x}_m = [x_{1m}, \ldots, x_{Dm}]^T$, whose mean is $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_D]^T$. Define the data matrix to be $\boldsymbol{A} = [\boldsymbol{x}_1 - \boldsymbol{\mu}, \ldots, \boldsymbol{x}_M - \boldsymbol{\mu}]$, and suppose that the eigenvectors and eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$ are given by $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M]$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_M)$.

   (a) Find the numerical value of the vector $\boldsymbol{U}^T\boldsymbol{u}_3$.

   (b) Your goal is to find a $(D \times M)$ matrix $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M]$ so that $\boldsymbol{y}_m = \boldsymbol{V}^T(\boldsymbol{x}_m - \boldsymbol{\mu})$ is a vector containing the first M principal components of the image $\boldsymbol{x}_m$. Write an equation showing how $\boldsymbol{V}$ can be computed from $\boldsymbol{\mu}$, $\boldsymbol{A}$, $\boldsymbol{U}$, and/or $\boldsymbol{\Lambda}$.

6. (10 points) Suppose that you have M different D-dimensional vectorized face images, $\boldsymbol{x}_m = [x_{1m}, \ldots, x_{Dm}]^T$, whose mean is $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_D]^T$. Define the scatter matrix to be

$$\boldsymbol{S} = \sum_{m=1}^{M} (\boldsymbol{x}_m - \boldsymbol{\mu})(\boldsymbol{x}_m - \boldsymbol{\mu})^T$$

   Suppose that the eigenvectors and eigenvalues of $\boldsymbol{S}$ are $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_D]$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_D)$. You want to find a value of K such that the K-dimensional PCA projection $\boldsymbol{y}_m = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K]^T(\boldsymbol{x}_m - \boldsymbol{\mu})$ has the following property:

$$\sum_{m=1}^{M} \|\boldsymbol{y}_m\|^2 = (0.95) \sum_{m=1}^{M} |\boldsymbol{x}_m - \boldsymbol{\mu}|^2 \tag{1}$$

   Specify an equation that, if satisfied, will guarantee the truth of Eq. 1. Your equation should only include the scalars M, D, K, and/or the eigenvalues λ_d $(1 \leq d \leq D)$; your equation should not include $\boldsymbol{x}_m$ or $\boldsymbol{\mu}$.

7. (10 points) Suppose you have a dataset including the vectors

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} 2 \\ 0 \\ 3 \end{bmatrix}, \quad \boldsymbol{z} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}$$

Find a diagonal matrix $\boldsymbol{\Sigma}$ such that the Mahalanobis distance satisfies $d^2_{\boldsymbol{\Sigma}}(\vec{x}, \vec{y}) > d^2_{\boldsymbol{\Sigma}}(\vec{x}, \vec{z})$.

8. (10 points) Suppose that a particular covariance matrix $\boldsymbol{\Sigma}$ has the following eigenvector matrix, $\boldsymbol{U}$, and eigenvalue matrix, $\boldsymbol{\Lambda}$:

$$\boldsymbol{U} = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \boldsymbol{\Lambda} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

Let $\boldsymbol{y}(\boldsymbol{x}) = \begin{bmatrix} y_1(\boldsymbol{x}) \\ y_2(\boldsymbol{x}) \end{bmatrix} = \boldsymbol{U}^T \boldsymbol{x}$ be the principal components of a vector space $\boldsymbol{x}$.

(a) Plot the set of vectors $\boldsymbol{x}$ such that $y_1(\boldsymbol{x}) = 3$.

(b) Find the squared Mahalanobis distance, $d^2_{\boldsymbol{\Sigma}}(\boldsymbol{x}, \boldsymbol{\mu})$, between the vectors $\boldsymbol{x}$ and $\boldsymbol{\mu}$ where

$$\boldsymbol{x} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

9. (10 points) A 2-dimensional Gaussian random vector has mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ given by

$$\boldsymbol{\mu} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} 8 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$$

Draw a curve of some kind, on a two-dimensional Cartesian plane, showing the set of points $\left\{\boldsymbol{x} : p_X(\boldsymbol{x}) = \frac{1}{8\pi} e^{-\frac{1}{2}}\right\}$.

10. (10 points) Consider a scalar LSTM, with a scalar memory cell, input gate, output gate, and forget gate, related to one another by scalar coefficients $a_i, b_i, a_o, b_o, a_f, b_f, a_c, b_c$ as follows:

$$
\begin{aligned}
i[n] &= \text{input gate} = \sigma(b_i x[n] + a_i c[n-1]), \quad 1 \leq n \\
o[n] &= \text{output gate} = \sigma(b_o x[n] + a_o c[n-1]), \quad 1 \leq n \\
f[n] &= \text{forget gate} = \sigma(b_f x[n] + a_f c[n-1]), \quad 1 \leq n \\
c[n] &= f[n]c[n-1] + i[n]\left(b_c x[n] + a_c c[n-1]\right), 1 \leq n \\
y[n] &= o[n]c[n], \quad 1 \leq n
\end{aligned}
$$

Suppose that the network is initialized with $b_i = b_o = b_f = a_i = a_o = a_f = a_c = 0$, and $c[0] = 0$. In fact, the only nonzero coefficient is $b_c = 1$. Under this condition, find a formula for $y[n]$ in terms of the values of $x[m]$, $1 \leq m \leq n$. No variables other than $x[m]$ should appear in your answer. HINT: $\sigma(0) = 1/2$.

11. (10 points) Suppose you have an $M \times D$ matrix, $\boldsymbol{X} = [\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{M-1}]^T$, where $\sum_{m=0}^{M-1} \boldsymbol{x}_m = \boldsymbol{0}$. The eigenvalues of $\boldsymbol{X}^T\boldsymbol{X}$ are λ_0 through λ_{D-1}, its eigenvectors are $\boldsymbol{v}_0$ through $\boldsymbol{v}_{D-1}$, and its principal components are $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{V}$.

(a) Write $\boldsymbol{Y}^T\boldsymbol{Y}$ in terms of the eigenvalues, λ_0 through λ_{D-1}.

(b) Write $\sum_{m=0}^{M-1} \|\boldsymbol{x}_m\|_2^2$ in terms of the eigenvalues, λ_0 through λ_{D-1}.

(c) Write $\boldsymbol{v}_i^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{v}_j$ in terms of the eigenvalues, λ_0 through λ_{D-1}, for $0 \leq i \leq j \leq D-1$.

12. (20 points) Suppose we're trying to predict the sequence $\zeta_1, \ldots, \zeta_{100}$ from the sequence $x_1, \ldots, x_{100}$. We want to use some type of neural net (RNN or LSTM) to compute $z_1, \ldots, z_{100}$ in order to minimize the error

$$E = \frac{1}{200} \sum_{t=1}^{100} (z_t - \zeta_t)^2$$

We only have one training sequence $(x_1, \ldots, x_{100}, \zeta_1, \ldots, \zeta_{100})$.

(a) Suppose we use an **RNN (recurrent neural network)** with just one scalar memory cell whose weights and biases are w, u, and b:

$$z_t = \sigma(ux_t + wz_{t-1} + b)$$

Find the derivatives of the error with respect to the weights and biases (dE/du, dE/dw, and dE/db). Express your answers in terms of x_j, z_k, and ζ_k for appropriate values of k and j; **the terms u, w and b should not show up on the right-hand-side of any of your equations.** You may express your answer recursively, or your answer may contain summation ($\sum$) and/or product ($\prod$) terms.

(b) Suppose we use an **LSTM (long-short-term memory network)** whose weights and biases are pre-specified: $u_c = 1$, and all of the other weights and biases are zero:

$$b_c = 0, u_c = 1, w_c = 0, b_f = 0, u_f = 0, w_f = 0, b_i = 0, u_i = 0, w_i = 0, b_o = 0, u_o = 0, w_o = 0$$

$$f[t] = \sigma(u_f x_t + w_f z_{t-1} + b_f), \quad i[t] = \sigma(u_i x_t + w_i z_{t-1} + b_i), \quad o[t] = \sigma(u_o x_t + w_o z_{t-1} + b_o)$$

$$c[t] = f[t]c[t-1] + i[t]\sigma\left(u_c x_t + w_c z_{t-1} + b_c\right), \quad z_t = o[t]c[t]$$

Assume that $c[t] = 0$ for $t \leq 0$. **Express z_t in terms of $\sigma(x_t)$ for $0 \leq t \leq 100$. Your answer should NOT contain any of the variables $c[t]$, $f[t]$, $i[t]$, or $o[t]$. Your answer may contain a summation ($\sum$).** You may find it useful to know that $\sigma(0) = \frac{1}{2}$.

13. (30 points) A particular unlabeled dataset, $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ has been centered so that the sample mean is $\boldsymbol{\mu} = [0, 0]^T$. The sample covariance matrix, $\boldsymbol{\Sigma}$ has the following value, and the following eigenvalue decomposition:

$$\boldsymbol{\Sigma} = \begin{bmatrix} 4 & -2 \\ -2 & 5 \end{bmatrix} = \begin{bmatrix} -0.79 & 0.62 \\ -0.62 & -0.79 \end{bmatrix} \begin{bmatrix} 2.44 & 0 \\ 0 & 6.56 \end{bmatrix} \begin{bmatrix} -0.79 & -0.62 \\ 0.62 & -0.79 \end{bmatrix}$$

Suppose you want to find a unit-length vector $\boldsymbol{v}$ that makes the quantity $\mathcal{J}$, defined in the following equation, as large as possible:

$$\boldsymbol{v} = \text{argmax}\, \mathcal{J} \quad s.t.\ \|\boldsymbol{v}\| = 1 \quad \text{and} \quad \mathcal{J} = \frac{\sum_{i=1}^{n} (\boldsymbol{v}^T \boldsymbol{x}_i)^2}{\sum_{i=1}^{n} \|\boldsymbol{x}_i\|^2}$$

(a) What is the numerical value of $\boldsymbol{v}$ that maximizes $\mathcal{J}$? You may leave your answer as an explicit function of numerical quantities, if you wish.

(b) What is the maximum achievable numerical value of $\mathcal{J}$? You may leave your answer as an explicit function of numerical quantities, if you wish.

(c) The gram matrix is an $n \times n$ matrix, G, whose $(i, j)^{\text{th}}$ element is

$$G[i, j] = \boldsymbol{x}_i^T \boldsymbol{x}_j$$

In terms of n, what are the eigenvalues of G?

14. (10 points) The gram matrix of a dataset is the matrix whose $(i,j)^{\text{th}}$ element is $\boldsymbol{x}_i^T \boldsymbol{x}_j$, the inner product of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. A particular dataset has a gram matrix with the following eigenvector/eigenvalue decomposition:

$$\boldsymbol{G} = \begin{bmatrix} -0.19 & -0.22 \\ -0.33 & -0.49 \\ -0.52 & 0.15 \\ -0.34 & 0.77 \\ -0.68 & -0.29 \\ 0.04 & -0.04 \end{bmatrix} \begin{bmatrix} 20 & 0 \\ 0 & 45 \end{bmatrix} \begin{bmatrix} -0.19 & -0.33 & -0.52 & -0.34 & -0.68 & 0.04 \\ -0.22 & -0.49 & 0.15 & 0.77 & -0.29 & -0.04 \end{bmatrix}$$

Suppose that $\boldsymbol{\Sigma}$ is the sample covariance of the same dataset, and suppose that $\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$. Draw the set of points $\{\boldsymbol{x} : \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} = 1\}$. Specify the numerical value of the coordinate of every point where this set intersects the axes.

15. (10 points) Suppose you are studying the running behaviors of trained vs. untrained athletes. You have a sequence of feature vectors $\boldsymbol{x}_t$, where t is time (measured in centiseconds) and $\boldsymbol{x}_t$ is a vector of features computed from a motion sensor being worn at the ankle. You have trained a neural network to compute $b_j(\boldsymbol{x}_t) = p(\boldsymbol{x}_t|q_t = j)$, where $q_t \in \{1 = \text{heel strike}, 2 = \text{roll}, 3 = \text{lift}, 4 = \text{swing}\}$ denotes the gait phase. You also know the following probabilities:

$$\begin{aligned} a_{i,j} &= p(q_t = j|q_{t-1} = i) \\ \alpha_t(i) &= p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, q_t = i) \\ \beta_t(i) &= p(\boldsymbol{x}_{t+1}, \ldots, \boldsymbol{x}_T|q_t = i) \end{aligned}$$

Your goal is to identify all of the instants when the heel first touches the ground, i.e., at each time step τ ($1 \leq \tau \leq T$), you want to find

$$P_{HS}(\tau) = p(q_{\tau-1} = 4, q_\tau = 1|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$$

Write a formula for $P_{HS}(\tau)$ in terms of $\alpha_t(i)$, $\beta_t(i)$, $a_{i,j}$, and $b_i(\boldsymbol{x}_t)$, for any values of i, j, t that you find useful.

16. (17 points) In a neural network with residual connections (ResNet), the k^{th} activation at layer ℓ, $h_k^{(\ell)}$, is equal to the activation of the same node at the previous layer, plus a computed residual $g(\xi_k^{(\ell)})$:

$$\begin{aligned} \xi_k^{(\ell)} &= \sum_{j=1}^{N} w_{k,j}^{(\ell)} h_j^{(\ell-1)}, \quad 1 \leq k \leq N, \\ h_k^{(\ell)} &= h_k^{(\ell-1)} + g\left(\xi_k^{(\ell)}\right), \quad 1 \leq k \leq N, \end{aligned}$$

where $g(\cdot)$ is a scalar nonlinearity, and $w_{k,j}^{(\ell)}$ is a network weight. Suppose that the training loss is $\mathcal{L}$, and suppose you already know $\frac{d\mathcal{L}}{dh_k^{(\ell)}}$. Find $\frac{d\mathcal{L}}{dh_j^{(\ell-1)}}$ in terms of $\frac{d\mathcal{L}}{dh_k^{(\ell)}}$, $\dot{g}(\xi) = \frac{\partial g}{\partial \xi_k^{(\ell)}}$, and $w_{k,j}^{(\ell)}$.

17. (17 points) Suppose we have five variables, u, v, w, x, y. All but seven of their partial derivatives are zero; for example, $\frac{\partial y}{\partial u}(u, v, w, x, y) = \frac{\partial y}{\partial x}(u, v, w, x, y) = 0$. The only seven nonzero partial derivatives are

$$\begin{aligned}
&\frac{\partial u}{\partial x}(u, v, w, x, y) = a, && \frac{\partial v}{\partial x}(u, v, w, x, y) = b \\
&\frac{\partial w}{\partial u}(u, v, w, x, y) = c, && \frac{\partial w}{\partial v}(u, v, w, x, y) = d \\
&\frac{\partial w}{\partial x}(u, v, w, x, y) = e, && \frac{\partial y}{\partial v}(u, v, w, x, y) = f \\
&\frac{\partial y}{\partial w}(u, v, w, x, y) = g, &&
\end{aligned}$$

In terms of the constants a, b, c, d, e, f, and g, find $\nabla \begin{bmatrix} x \\ u \end{bmatrix} y$, the gradient of y with respect to the vector $[x, u]^T$.

18. (30 points) Consider a bidirectional two-layer recurrent network that has been trained to perform the following computations.

- The first layer has forward and backward cells which perform the following computations given an input $x \in \Re$ and prior hidden states $f \in \Re$, $b \in \Re$:

$$\textbf{forward} : f_t = \sin(x_t w_x + f_{t-1} w_h + b)^2,$$
$$\textbf{backward} : b_t = \sin(x_t w_x + b_{t+1} w_h + b)^2,$$

  where the weights are $w_x = \frac{\pi}{4}$, $w_h = \frac{\pi}{2}$, and $b = \frac{\pi}{2}$.

- The second layer has forward and backward cells which perform the following computations given an input $\boldsymbol{\xi} \in \Re^2$ and a prior hidden states $y \in \Re$, $z \in \Re$:

$$\textbf{forward} : y_t = \cos\left(\frac{\pi}{2}(\boldsymbol{w}_x^T \boldsymbol{\xi}_t + w_h y_{t-1} + b)\right),$$
$$\textbf{backward} : z_t = \cos\left(\frac{\pi}{2}(\boldsymbol{w}_x^T \boldsymbol{\xi}_t + w_h z_{t+1} + b)\right),$$

  where the weights are $w_x = [2, 1]^T$, $w_h = 2$, and $b = 1$. Assume that the prior hidden state, before each cell reads its first input, is 0.

(a) Consider the input sequence $[x_1, x_2, x_3] = [4, 1, 7]$. What are the forward outputs $[f_1, f_2, f_3]$ and the backward outputs $[b_3, b_2, b_1]$ from the first layer?

(b) Now consider the outputs $[f_1, f_2, f_3] = [3, 1, 3]$ from the forward cell in the first layer and the outputs $[b_3, b_2, b_1] = [3, 1, 0]$ from the backward cell in the first layer. Let $\boldsymbol{\xi}_t = [f_t, b_t]^T$. What are the forward outputs $[y_1, y_2, y_3]$ and the backward outputs $[z_3, z_2, z_1]$ from the second layer?

19. (20 points) Consider an LSTM defined by

$$
\begin{aligned}
\boldsymbol{i}[n] &= \text{input gate} = \sigma(\boldsymbol{B}_i\boldsymbol{x}[n] + \boldsymbol{A}_i\boldsymbol{c}[n-1]) \\
\boldsymbol{o}[n] &= \text{output gate} = \sigma(\boldsymbol{B}_o\boldsymbol{x}[n] + \boldsymbol{A}_o\boldsymbol{c}[n-1]) \\
\boldsymbol{f}[n] &= \text{forget gate} = \sigma(\boldsymbol{B}_f\boldsymbol{x}[n] + \boldsymbol{A}_f\boldsymbol{c}[n-1]) \\
\boldsymbol{c}[n] &= \boldsymbol{f}[n] \odot c[n-1] + \boldsymbol{i}[n] \odot g\left(\boldsymbol{B}_c\boldsymbol{x}[n] + \boldsymbol{A}_c\boldsymbol{c}[n-1]\right) \\
\boldsymbol{y}[n] &= \boldsymbol{o}[n] \odot \boldsymbol{c}[n]
\end{aligned}
$$

where the vector cell is $\boldsymbol{c}[n] = [c_1[n], \ldots, c_p[n]]^T$, and where $\odot$ denotes the Hadamard (array) product, e.g., $\boldsymbol{o}[n] \odot \boldsymbol{c}[n] = [o_1[n]c_1[n], \ldots, o_p[n]c_p[n]]^T$. Find the derivative $\frac{\partial c_j[n]}{\partial c_k[n-1]}$.