# 1 Linear Algebra Review

1. **First isomorphism theorem.** The isomorphism theorems are an important class of results with versions for various algebraic structures. Here, we are concerned about the first isomorphism theorem for vector spaces – one of the most fundamental results in linear algebra.

   **Theorem.** *Let $V, W$ be vector spaces, and let $T : V \to W$ be a linear map. Then the following are true:*

   *(a)* $\ker T$ *is a subspace of* $V$*.*

   *(b)* $\operatorname{Im} T$ *is a subspace of* $W$*.*

   *(c)* $\operatorname{Im} T$ *is isomorphic to* $V/\ker T$*.*

   Prove **parts (a) and (b)** of the theorem. (The interesting result is part (c), so, if you're inclined, try it out! We promise it's a very rewarding proof :) If you are interested but unfamiliar with the language, try looking up "isomorphism" and "quotient space.")

   **Solution:**

   Since $T: V \to W$ is a linear map, it satisfies: $T(v_1 + v_2) = T(v_1) + T(v_2)$, $T(a \cdot v) = a \cdot T(v)$ for all $v_1, v_2 \in V$, $a \in K$, and $v \in V$.

   (a) We prove $\ker T$ is a subspace of $V$ by verifying the three subspace axioms :

   - **Zero Vector**: Since $T(\mathbf{0}_V) = \mathbf{0}_W$ (by linearity), $\mathbf{0}_V \in \ker T$.
   - **Closed under addition**: Let $v_1, v_2 \in \ker T$. Then $T(v_1) = \mathbf{0}_W$ and $T(v_2) = \mathbf{0}_W$. Thus, $T(v_1 + v_2) = T(v_1) + T(v_2) = \mathbf{0}_W + \mathbf{0}_W = \mathbf{0}_W$. So, $v_1 + v_2 \in \ker T$.
   - **Closed under scalar multiplication**: Let $v \in \ker T$ and $a \in K$. Then $T(a \cdot v) = a \cdot T(v) = a \cdot \mathbf{0}_W = \mathbf{0}_W$. So, $a \cdot v \in \ker T$.

   Therefore, $\ker T$ is a subspace of $V$.

   (b) We prove $\operatorname{Im} T$ is a subspace of $W$ by verifying the three subspace axioms :

   - **Zero Vector**: Since $T(\mathbf{0}_V) = \mathbf{0}_W$ (by linearity), $\mathbf{0}_W \in \operatorname{Im} T$.
   - **Closed under addition**: Let $v_1, v_2 \in V$. Then $T(v_1) \in \operatorname{Im} T$ and $T(v_2) \in \operatorname{Im} T$. So, $T(v_1) + T(v_2) = T(v_1 + v_2) \in \operatorname{Im} T$.
   - **Closed under scalar multiplication**: Let $v \in V$ and $a \in K$. So $a \cdot T(v) = T(a \cdot v) \in \operatorname{Im} T$.

   Therefore, $\operatorname{Im} T$ is a subspace of $W$.

   (c) Define :

   - Equivalence class: $[v] = v + \ker T = \{v + k \mid k \in \ker T\}$.
   - Map $\phi$: $V/\ker T \to \operatorname{Im} T$, $\phi([v]) = T(v)$.

   We prove $\operatorname{Im} T$ is isomorphic to $V/\ker T$ by verifying the four features :

   - **Well-defined**: Suppose $[u] = [v]$, then $u - v \in \ker T$. Thus, $T(u - v) = \mathbf{0}_W$. So, $T(u) = T(v)$.
   - **Linearity**: For any $[u], [v] \in V/\ker T$ and scalar $a \in K$:

     $$\phi([u] + [v]) = \phi([u + v]) = T(u + v) = T(u) + T(v) = \phi([u]) + \phi([v])$$
     $$\phi(a \cdot [u]) = \phi([a \cdot u]) = T(a \cdot u) = a \cdot T(u) = a \cdot \phi([u])$$

   - **Injective**: For any $[u], [v] \in V/\ker T$ and $\phi([u]) = \phi([v])$, then $T(u - v) = T(u) - T(v) = \mathbf{0}_W$. Thus, $u - v \in \ker T$. So, $[u] = [v]$.
   - **Surjective**: For any $w \in \operatorname{Im} T$, $\exists v \in V$ such that $T(v) = w$. So $\phi([v]) = w$.

Therefore, $\phi$ is a vector space isomorphism, so $\operatorname{Im} T \cong V/\ker T$.

2. First we review some basic concepts of rank. Recall that elementary matrix operations do not change a matrix's rank. Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$. Let $I_n$ denote the $n \times n$ identity matrix.

   (a) Perform elementary row and column operations[1] to transform $\begin{bmatrix} I_n & 0 \\ 0 & AB \end{bmatrix}$ to $\begin{bmatrix} B & I_n \\ 0 & A \end{bmatrix}$.

   (b) Let's find lower and upper bounds on $\operatorname{rank}(AB)$. Use part (a) to prove that $\operatorname{rank} A + \operatorname{rank} B - n \leq \operatorname{rank}(AB)$. Then use what you know about the relationship between the column space (range) and/or row space of $AB$ and the column/row spaces for $A$ and $B$ to argue that $\operatorname{rank}(AB) \leq \min\{\operatorname{rank} A, \operatorname{rank} B\}$.

   (c) If a matrix $A$ has rank $r$, then some $r \times r$ submatrix $M$ of $A$ has a nonzero determinant. Use this fact to show the standard facts that the dimension of $A$'s column space is at least $r$, and the dimension of $A$'s row space is at least $r$. (Note: You will not receive credit for other ways of showing those same things.)

   (d) It is a fact that $\operatorname{rank}(A^\top A) = \operatorname{rank} A$; here's one way to see that. We've already seen in part (b) that $\operatorname{rank}(A^\top A) \leq \operatorname{rank} A$. Suppose that $\operatorname{rank}(A^\top A)$ were strictly less than $\operatorname{rank} A$. What would that tell us about the relationship between the column space of $A$ and the null space of $A^\top$? What standard fact about the fundamental subspaces of $A$ says that relationship is impossible?

   (e) Given a set of vectors $S \subseteq \mathbb{R}^n$, let $AS = \{Av : v \in s\}$ denote the subset of $\mathbb{R}^m$ found by applying $A$ to every vector in $S$. In terms of the ideas of the column space (range) and row space of $A$: What is $A\mathbb{R}^n$, and why? (*Hint*: what are the definitions of column space and row space?) What is $A^\top A\mathbb{R}^n$, and why? (Your answer to the latter should be purely in terms of the fundamental subspaces of $A$ itself, not in terms of the fundamental subspaces of $A^\top A$.)

**Solution:**

(a) $M = \begin{bmatrix} I_n & 0 \\ 0 & AB \end{bmatrix}$. Interchange columns 1 and 2, $M = \begin{bmatrix} 0 & I_n \\ AB & 0 \end{bmatrix}$. Add $B$ times column 2 to column 1, $M = \begin{bmatrix} B & I_n \\ AB & 0 \end{bmatrix}$. Add $-A$ times row 1 to row 2, $M = \begin{bmatrix} B & I_n \\ 0 & -A \end{bmatrix}$. Multiply row 2 by -1, $M = \begin{bmatrix} B & I_n \\ 0 & A \end{bmatrix}$.

(b) i. $\operatorname{rank}(\begin{bmatrix} I_n & 0 \\ 0 & AB \end{bmatrix}) = n + \operatorname{rank}(AB)$. There is no same column or row of $A$ and $B$ in $\begin{bmatrix} B & I_n \\ 0 & A \end{bmatrix}$, so $\operatorname{rank}(\begin{bmatrix} B & I_n \\ 0 & A \end{bmatrix}) \geq \operatorname{rank} A + \operatorname{rank} B$.

$$n + \operatorname{rank}(AB) = \operatorname{rank}(\begin{bmatrix} I_n & 0 \\ 0 & AB \end{bmatrix}) = \operatorname{rank}(\begin{bmatrix} B & I_n \\ 0 & A \end{bmatrix}) \geq \operatorname{rank} A + \operatorname{rank} B$$

So,

$$\operatorname{rank} A + \operatorname{rank} B - n \leq \operatorname{rank}(AB)$$

ii. Suppose $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$, $B = \begin{bmatrix} B_1 & B_2 \end{bmatrix}$, where $A_1 \in \mathbb{R}^{\operatorname{rank} A\times n}$, $A_1$ is a basis of $A$ and $B_1 \in \mathbb{R}^{\operatorname{rank} A\times n}$, $B_1$ is a basis of $B$. So $AB = \begin{bmatrix} A_1B_1 & A_1B_2 \\ A_2B_1 & A_2B_2 \end{bmatrix}$ and $A_1B_1$ is a basis of $AB$. $A_1B_1 \in \mathbb{R}^{\operatorname{rank} A\times rankB}$. So, $\operatorname{rank}(AB) = \operatorname{rank}(A_1B_1) \leq \min\{\operatorname{rank} A, \operatorname{rank} B\}$

[1] If you're not familiar with these, https://stattrek.com/matrix-algebra/elementary-operations is a decent introduction.

(c) Let $A$ have rank $r$. By definition, there exists an $r \times r$ submatrix $M$ of $A$ with $\det(M) \neq 0$. The $r$ columns of $A$ corresponding to $M$ are linearly independent, since otherwise $\det(M)$ would be zero. Thus, the dimension of the column space of $A$ is at least $r$. Similarly, the $r$ rows of $A$ corresponding to $M$ are linearly independent, so the dimension of the row space of $A$ is at least $r$.

(d) Define $\mathrm{Col}(A)$ : the column space of A.
Define Map $T$: $\mathrm{Col}(A) \to \mathbb{R}^n$ and $T(y) = A^\top y$, so $\operatorname{Im} T = \{A^\top y \mid y \in \mathrm{Col}(A)\}$.
According to question(1), $\operatorname{Im} T \cong \mathrm{Col}(A)/\ker T$, so $\dim(\operatorname{Im} T) + \dim(\ker T) = \dim \mathrm{Col}(A) = \operatorname{rank} A$.
Suppose that $\operatorname{rank}(A^\top A)$ were strictly less than $\operatorname{rank} A$, so $\dim(\operatorname{Im} T) = \operatorname{rank}(A^\top A) < \operatorname{rank} A$. That is $\dim(\ker T) > 0$. It means there exists nonzero vector $y \in \mathrm{Col}(A)$ and $T(y) = \mathbf{0}$, so $y \in \mathrm{Null}(A^\top)$. The standard fact about the fundamental subspaces : $\mathrm{Col}(A) \cap \mathrm{Null}(A^\top) = \{\mathbf{0}\}$. This contradicts the assumption.

(e) $A\,\mathbb{R}^n$ is $\mathrm{Col}(A)$.
For any vector $v \in \mathbb{R}^n$, $A^\top A v \in A^\top A\,\mathbb{R}^n$: $\forall w \in \mathrm{Null}(A)$, $(A^\top A v) \cdot w = (v^\top A^\top A)w = v^\top A^\top (Aw) = \mathbf{0}$. So $A^\top A v \perp w$. That means $A^\top A\,\mathbb{R}^n \subseteq \mathrm{Row}(A)$. Because $\operatorname{rank}(A^\top A) = \operatorname{rank} A$, $A^\top A\,\mathbb{R}^n = \mathrm{Row}(A)$.

3. Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix. Prove equivalence between these three different definitions of positive semidefiniteness (PSD). Note that when we talk about PSD matrices in this class, they are defined to be symmetric matrices. There are nonsymmetric matrices that exhibit PSD properties, like the first definition below, but not all three.

   (a) For all $x \in \mathbb{R}^n$, $x^\top A x \geq 0$.

   (b) All the eigenvalues of $A$ are nonnegative.

   (c) There exists a matrix $U \in \mathbb{R}^{n \times n}$ such that $A = UU^\top$.

   Positive semidefiniteness will be denoted as $A \succeq 0$.

   **Solution:** $A$ is a symmetric matrix that can be orthogonally diagonalized : $A = Q\Lambda Q^{-1}$, $Q$ is an orthogonal matrix, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, ...\lambda_n)$. Let $y = xQ^{-1}$, $x^\top A x = y^\top \Lambda y = \sum_{i=1}^n y_i^2 \lambda_i \geq 0$. $A = Q\Lambda Q^{-1} = UU^\top$. That is $\Lambda = DD^\top$. So for all column vectors in $D$, they are perpendicular to each other. $\lambda_i = D_{:,i}^2 \geq 0$. Therefore, the three are equivalent.
   What's more, The three are also equivalent to : $\forall i,\ A_{i,i} \geq 0$ and $\forall i, j,\ |A_{i,j}| \leq \sqrt{A_{i,i} * A_{j,j}}$.

4. The Frobenius inner product between two matrices of the same dimensions $A, B \in \mathbb{R}^{m \times n}$ is

$$\langle A, B \rangle = \mathrm{trace}(A^\top B) = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij} B_{ij},$$

   where $\operatorname{trace} M$ denotes the *trace* of $M$, which you should look up if you don't already know it. (The norm is sometimes written $\langle A, B \rangle_F$ to be clear.) The Frobenius norm of a matrix is

$$\|A\|_F = \sqrt{\langle A, A \rangle} = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |A_{ij}|^2}.$$

   Prove the following. The Cauchy–Schwarz inequality, the cyclic property of the trace, and the definitions in part 3 above may be helpful to you.

(a) $x^\top Ay = \langle A, xy^\top \rangle$ for all $x \in \mathbb{R}^m, y \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}$.

(b) If $A$ and $B$ are symmetric PSD matrices, then $\text{trace}(AB) \geq 0$.

(c) **[OPTIONAL]** If $A, B \in \mathbb{R}^{n \times n}$ are real symmetric matrices with $\lambda_{\max}(A) \geq 0$ and $B$ being PSD, then $\langle A, B \rangle \leq \sqrt{n}\lambda_{\max}(A)\|B\|_F$.
*Hint:* Construct a PSD matrix using $\lambda_{\max}(A)$

**Solution:** TODO

5. Let $A \in \mathbb{R}^{m \times n}$ be an arbitrary matrix. The maximum singular value of $A$ is defined to be $\sigma_{\max}(A) = \sqrt{\lambda_{\max}(A^\top A)} = \sqrt{\lambda_{\max}(AA^\top)}$. Prove that

$$\sigma_{\max}(A) = \max_{\substack{u \in \mathbb{R}^m,\ v \in \mathbb{R}^n \\ \|u\|=1,\ \|v\|=1}} (u^\top Av).$$

**Solution:** TODO

# 2 Matrix/Vector Calculus and Norms

1. Consider a 2×2 matrix A, written in full as $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$, and two arbitrary 2-dimensional vectors $x, y$. Calculate the gradient of

$$\sin(A_{11}^2 + e^{A_{11}+A_{22}}) + x^\top A y$$

with respect to the matrix $A$.
*Hint*: The gradient has the same dimensions as $A$. Use the chain rule.

**Solution:** TODO

2. Aside from norms on vectors, we can also impose norms on matrices. Besides the Frobenius norm, the most common kind of norm on matrices is called the induced norm. Induced norms are defined as

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$$

where the notation $\|\cdot\|_p$ on the right-hand side denotes the vector $\ell_p$-norm. Please give the closed-form (or the most simple) expressions for the following induced norms of $A \in \mathbb{R}^{m \times n}$.

   (a) $\|A\|_2$. *Hint*: use the singular value decomposition.

   (b) $\|A\|_\infty$

**Solution:** TODO

3. (a) Let $\alpha = \sum_{i=1}^{n} y_i \ln(1 + e^{\beta_i})$ for $y, \beta \in \mathbb{R}^n$. What are the partial derivatives $\frac{\partial \alpha}{\partial \beta_i}$?

   (b) Given $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{mxn}$. Write the partial derivative $\frac{\partial (Ax)}{\partial x}$.

   (c) Given $z \in \mathbb{R}^m$. Write the gradient $\nabla_z (z^\top z)$.

   (d) Given $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, and $z = g(x)$. Write the gradient $\nabla_x z^\top z$ in terms of $\frac{\partial z}{\partial x}$ and $z$.

   (e) Given $x \in \mathbb{R}^n$, $y$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{mxn}$, and $z = Ax - y$. Write the gradient $\nabla_x (z^\top z)$.

**Solution:** TODO

# 3 Linear Neural Networks

Let's apply the multivariate chain rule to a "simple" type of neural network called a *linear neural network*. They're not very powerful, as they can learn only linear regression functions or decision functions, but they're a good stepping stone for understanding more complicated neural networks. We are given an $n \times d$ *design matrix* $X$. Each row of $X$ is a training point, so $X$ represents $n$ training points with $d$ features each. We are also given an $n \times k$ matrix $Y$. Each row of $Y$ is a set of $k$ labels for the corresponding training point in $X$. Our goal is to learn a $k \times d$ matrix $W$ of weights[2] such that

$$Y \approx XW^\top.$$

If $n$ is larger than $d$, typically there is no $W$ that achieves equality, so we seek an approximate answer. We do that by finding the matrix $W$ that minimizes the *cost function*

$$\text{RSS}(W) = \|XW^\top - Y\|_F^2. \tag{1}$$

This is a classic *least-squares linear regression* problem; most of you have seen those before. But we are solving $k$ linear regression problems simultaneously, which is why $Y$ and $W$ are matrices instead of vectors.

**Linear neural networks.** Instead of optimizing $W$ over the space of $k \times d$ matrices directly, we write the $W$ we seek as a product of multiple matrices. This parameterization is called a *linear neural network*.

$$W = \mu(W_L, W_{L-1}, \ldots, W_2, W_1) = W_L W_{L-1} \cdots W_2 W_1.$$

Here, $\mu$ is called the *matrix multiplication map* (hence the Greek letter mu) and each $W_j$ is a real-valued $d_j \times d_{j-1}$ matrix. Recall that $W$ is a $k \times d$ matrix, so $d_L = k$ and $d_0 = d$. $L$ is the number of *layers* of "connections" in the neural network. You can also think of the network as having $L + 1$ layers of units: $d_0 = d$ units in the *input layer*, $d_1$ units in the first *hidden layer*, $d_{L-1}$ units in the last hidden layer, and $d_L = k$ units in the *output layer*.

We collect all the neural network's weights in a *weight vector* $\theta = (W_L, W_{L-1}, \ldots, W_1) \in \mathbb{R}^{d_\theta}$, where $d_\theta = d_L d_{L-1} + d_{L-1} d_{L-2} + \ldots + d_1 d_0$ is the total number of real-valued weights in the network. Thus we can write $\mu(\theta)$ to mean $\mu(W_L, W_{L-1}, \ldots, W_1)$. But you should imagine $\theta$ as a column vector: we take all the components of all the matrices $W_L, W_{L-1}, \ldots, W_1$ and just write them all in one very long column vector. Given a fixed weight vector $\theta$, the linear neural network takes an *input vector* $x \in \mathbb{R}^{d_0}$ and returns an *output vector* $y = W_L W_{L-1} \cdots W_2 W_1 x = \mu(\theta)x \in \mathbb{R}^{d_L}$.

Now our goal is to find a weight vector $\theta$ that minimizes the composition $\text{RSS} \circ \mu$—that is, it minimizes the cost function

$$J(\theta) = \text{RSS}(\mu(\theta)).$$

We are substituting a linear neural network for $W$ and optimizing the weights in $\theta$ instead of directly optimizing the components of $W$. This makes the optimization problem harder to solve, and you would never solve least-squares linear regression problems this way in practice; but again, it is a good exercise to work toward understanding the behavior of "real" neural networks in which $\mu$ is *not* a linear function.

We would like to use a gradient descent algorithm to find $\theta$, so we will derive $\nabla_\theta J$ as follows.

1. The gradient $G = \nabla_W \text{RSS}(W)$ is a $k \times d$ matrix whose entries are $G_{ij} = \partial \text{RSS}(W)/\partial W_{ij}$, where $\text{RSS}(W)$ is defined by Equation (1). Knowing that the simple formula for $\nabla_W \text{RSS}(W)$ in matrix notation can be written as the following:

$$\nabla_W \text{RSS}(W) = 2(WX^\top - Y^\top)X$$

[2]The reason for the transpose on $W^\top$ is because we think in terms of applying $W$ to an individual training point. Indeed, if $X_i \in \mathbb{R}^d$ and $Y_i \in \mathbb{R}^k$ respectively denote the $i$-th rows of $X$ and $Y$ transposed to be column vectors, then we can write $Y_i \approx WX_i$.
For historical reasons, most papers in the literature use design matrices whose rows are sample points, rather than columns.

prove this fact by deriving a formula for each $G_{ij}$ using summations, simplified as much as possible. *Hint:* To break down $\text{RSS}(W)$ into its component summations, start with the relationship between the Frobenius norm and the trace of a matrix.

**Solution:** TODO

2. Directional derivatives are closely related to gradients. The notation $\text{RSS}'_{\Delta W}(W)$ denotes the directional derivative of $\text{RSS}(W)$ in the direction $\Delta W$, and the notation $\mu'_{\Delta\theta}(\theta)$ denotes the directional derivative of $\mu(\theta)$ in the direction $\Delta\theta$.[3] Informally speaking, the directional derivative $\text{RSS}'_{\Delta W}(W)$ tells us how much $\text{RSS}(W)$ changes if we increase $W$ by an infinitesimal displacement $\Delta W \in \mathbb{R}^{k\times d}$. (However, any $\Delta W$ we can actually specify is not actually infinitesimal; $\text{RSS}'_{\Delta W}(W)$ is a local linearization of the relationship between $W$ and $\text{RSS}(W)$ at $W$. To a physicist, $\text{RSS}'_{\Delta W}(W)$ tells us the initial velocity of change of $\text{RSS}(W)$ if we start changing $W$ with velocity $\Delta W$.)

   Show how to write $\text{RSS}'_{\Delta W}(W)$ as a Frobenius inner product of two matrices, one related to part 3.1.

   **Solution:** TODO

3. In principle, we could take the gradient $\nabla_\theta \mu(\theta)$, but we would need a 3D array to express it! As we don't know a nice way to write it, we'll jump directly to writing the directional derivative $\mu'_{\Delta\theta}(\theta)$. Here, $\Delta\theta \in \mathbb{R}^{d_\theta}$ is a weight vector whose matrices we will write $\Delta\theta = (\Delta W_L, \Delta W_{L-1}, \ldots, \Delta W_1)$. Show that

$$\mu'_{\Delta\theta}(\theta) = \sum_{j=1}^{L} W_{>j} \Delta W_j W_{<j}$$

   where $W_{>j} = W_L W_{L-1} \cdots W_{j+1}$, $W_{<j} = W_{j-1} W_{j-2} \cdots W_1$, and we use the convention that $W_{>L}$ is the $d_L \times d_L$ identity matrix and $W_{<1}$ is the $d_0 \times d_0$ identity matrix.
   *Hint*: although $\mu$ is not a linear function of $\theta$, $\mu$ is linear in any *single* $W_j$; and any directional derivative of the form $\mu'_{\Delta\theta}(\theta)$ is linear in $\Delta\theta$ (for a fixed $\theta$).

   **Solution:** TODO

4. Recall the chain rule for scalar functions, $\frac{\mathrm{d}}{\mathrm{d}x} f(g(x))|_{x=x_0} = \frac{\mathrm{d}}{\mathrm{d}y} f(y)|_{y=g(x_0)} \cdot \frac{\mathrm{d}}{\mathrm{d}x} g(x)|_{x=x_0}$. There is a multivariate version of the chain rule, which we hope you remember from some class you've taken, and the multivariate chain rule can be used to chain directional derivatives. Write out the chain rule that expresses the directional derivative $J'_{\Delta\theta}(\theta)|_{\theta=\theta_0}$ by composing your directional derivatives for RSS and $\mu$, evaluated at a weight vector $\theta_0$. (Just write the pure form of the chain rule without substituting the values of those directional derivatives; we'll substitute the values in the next part.)

   **Solution:** TODO

5. Now substitute the values you derived in parts 3.2 and 3.3 into your expression for $J'_{\Delta\theta}(\theta)$ and use it to show that

$$\begin{aligned}\nabla_\theta J(\theta) \quad = \quad & (2\,(\mu(\theta)\,X^\top - Y^\top) X W_{<L}^\top, \\ & \ldots, \\ & 2 W_{>j}^\top\,(\mu(\theta)\,X^\top - Y^\top) X W_{<j}^\top, \\ & \ldots, \\ & 2 W_{>1}^\top\,(\mu(\theta)\,X^\top - Y^\top) X).\end{aligned}$$

   This gradient is a vector in $\mathbb{R}^{d_\theta}$ written in the same format as $(W_L, \ldots, W_j, \ldots, W_1)$. Note that the values $W_{>j}$ and $W_{<j}$ here depend on $\theta$.
   *Hint*: you might find the cyclic property of the trace handy.

[3] "$\Delta W$" and "$\Delta\theta$" are just variable names that remind us to think of these as small displacements of $W$ or $\theta$; the Greek letter delta is not an operator nor a separate variable.

**Solution:** TODO

# 4 Probability Potpourri

1. Recall the covariance of two scalar random variables $X$ and $Y$ is defined as $\text{Cov}(X,Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$. For a multivariate random variable $Z \in \mathbb{R}^n$, (i.e., $Z$ is a column vector where each element $Z_i$ is a scalar random variable), we define the covariance matrix $\Sigma$ such that $\Sigma_{ij} = \text{Cov}(Z_i, Z_j)$. Concisely, $\Sigma = \mathbb{E}[(Z-\mu)(Z-\mu)^\top]$, where $\mu$ is the mean value of $Z$. Prove that the covariance matrix is always positive semidefinite (PSD).
   *Hint*: Use linearity of expectation.

   **Solution:** TODO

2. Suppose a pharmaceutical company is developing a diagnostic test for a rare disease that has a prevalence of 1 in 1,000 in the population. Let $x$ be the true positive rate of the test, and let $y$ be the false positive rate. Determine the minimum value that $x$ must have, expressed as a function of $y$, such that a patient who tests positive actually has the disease with probability greater than 0.5.

   **Solution:** TODO

3. An archery target is made of 3 concentric circles of radii $1/\sqrt{3}$, 1 and $\sqrt{3}$ feet. Arrows striking within the inner circle are awarded 4 points, arrows within the middle ring are awarded 3 points, and arrows within the outer ring are awarded 2 points. Shots outside the target are awarded 0 points.

   Consider a random variable $X$, the distance of the strike from the center (in feet), and let the probability density function of $X$ be

   $$f(x) = \begin{cases} \frac{2}{\pi(1+x^2)} & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

   What is the expected value of the score of a single strike?

   **Solution:** TODO

4. Let $X \sim \text{Pois}(\lambda)$, $Y \sim \text{Pois}(\mu)$. Given that $X \perp\!\!\!\perp Y$, derive an expression for $\mathbb{P}(X = k \mid X + Y = n)$ where $k = 0, \ldots, n$. What well-known probability distribution is this? What are its parameters?

   **Solution:** TODO

5. Consider a coin that may be biased, where the probability of the coin landing heads on any single flip is $\theta$. If the coin is flipped $n$ times and heads is observed $k$ times, what is the maximum likelihood estimate (MLE) of $\theta$?

   **Solution:** TODO

6. Consider a family of distributions parameterized by $\theta \in \mathbb{R}$ with the following probability density function:

   $$f_\theta(x) = \begin{cases} e^{\theta - x} & \text{when } x \geq \theta \\ 0 & \text{when } x < \theta \end{cases}$$

   (a) Prove that $f$ is a valid probability density function by showing that it integrates to 1 for all $\theta$.

   (b) Suppose that you observe $n$ samples distributed according to $f$: $x_1, x_2, \ldots, x_n$. Find the maximum likelihood estimate of $\theta$.

   **Solution:** TODO

# 5 The Multivariate Normal Distribution

The multivariate normal distribution with mean $\mu \in \mathbb{R}^d$ and positive definite covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$, denoted $\mathcal{N}(\mu, \Sigma)$, has the probability density function

$$f(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right).$$

Here $|\Sigma|$ denotes the determinant of $\Sigma$. You may use the following facts without proof.

- The volume under the normal PDF is 1.

$$\int_{\mathbb{R}^d} f(x)\,\mathrm{d}x = \int_{\mathbb{R}^d} \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)\,\mathrm{d}x = 1.$$

- The change-of-variables formula for integrals: let $f$ be a smooth function from $\mathbb{R}^d \to \mathbb{R}$, let $A \in \mathbb{R}^{d\times d}$ be an invertible matrix, and let $b \in \mathbb{R}^d$ be a vector. Then, performing the change of variables $x \mapsto z = Ax + b$,

$$\int_{\mathbb{R}^d} f(x)\,\mathrm{d}x = \int_{\mathbb{R}^d} f(A^{-1}z - A^{-1}b)\,|A^{-1}|\,\mathrm{d}z.$$

All throughout this question, we take $X \sim \mathcal{N}(\mu, \Sigma)$.

1. Use a suitable change of variables to show that $\mathbb{E}[X] = \mu$. You must utilize the definition of expectation.

   **Solution:** TODO

2. Use a suitable change of variables to show that $\mathrm{Var}(X) = \Sigma$, where the variance of a vector-valued random variable $X$ is

$$\mathrm{Var}(X) = \mathrm{Cov}(X, X) = \mathbb{E}[(X-\mu)(X-\mu)^\top] = \mathbb{E}[XX^\top] - \mu\,\mu^\top.$$

   *Hints*: Every symmetric, positive definite matrix $\Sigma$ has a symmetric, positive definite square root $\Sigma^{1/2}$ such that $\Sigma = \Sigma^{1/2}\Sigma^{1/2}$. Note that $\Sigma$ and $\Sigma^{1/2}$ are invertible. After the change of variables, you will have to find another variance $\mathrm{Var}(Z)$; if you've chosen the right change of variables, you can solve that by solving the integral for each diagonal component of $\mathrm{Var}(Z)$ and a second integral for each off-diagonal component. The diagonal components will require integration by parts. You **cannot** assume anything about $\mathrm{Var}(Z)$ – you must compute it via integration.

   **Solution:** TODO

3. Compute the moment generating function (MGF) of $X$: $M_X(\lambda) = \mathbb{E}[e^{\lambda^\top X}]$, where $\lambda \in \mathbb{R}^d$. Note: moment generating functions have several interesting and useful properties, one being that $M_X$ characterizes the distribution of $X$: if $M_X = M_Y$, then $X$ and $Y$ have the same distribution.
   *Hints*:

   - You should try "completing the square" in the exponent of the Gaussian PDF.
   - You should arrive at

$$M_X(\lambda) = \exp\left(\lambda^\top \mu + \frac{1}{2}\lambda^\top \Sigma \lambda\right).$$

   **Solution:** TODO

4. Using the fact that MGFs determine distributions, given $A \in \mathbb{R}^{k\times d}$ and $b \in \mathbb{R}^k$ identify the distribution of $AX + b$ (don't worry about covariance matrices being invertible).

   **Solution:** TODO

5. Show that there exists an affine transformation of $X$ that is distributed as the standard multivariate Gaussian, $\mathcal{N}(0, I_d)$. (Assume $\Sigma$ is invertible.)

   **Solution:** TODO

# 6 Real Analysis

1. **Limit of a Sequence.** A sequence $\{x_n\}$ is said to converge to a limit $L$ if, for every measure of closeness $\epsilon \in \mathbb{R}$, the sequence's terms $n \in \mathbb{N}$ after a point $n_0 \in \mathbb{N}$ converge upon that limit. More formally, if $\lim_{n\to\infty} x_n = L$ then $\forall \epsilon > 0, \exists n_0 \in \mathbb{Z}^+$ such that $\forall n \geq n_0, |x_n - L| < \epsilon$.

   (a) Consider the sequence $\{x_n\}$ defined by the recurrence relation $x_{n+1} = \frac{1}{2}x_n$. Treat $x_0$ as some constant that is the first element of the sequence. Prove that $\{x_n\}$ converges by evaluating $\lim_{n\to\infty} x_n$. **You must use the formal definition of the limit of a sequence.**

   (b) **[OPTIONAL]** Consider a sequence $\{x_n\}$ of non-zero real numbers and suppose that

   $$L = \lim_{n\to\infty} n\left(1 - \frac{|x_{n+1}|}{|x_n|}\right)$$

   exists. Prove that $\{|x_n|\}$ converges when $L > 1$ by evaluating $\lim_{n\to\infty} |x_n|$.
   *Hint*: Use the Bernoulli Inequality $1 - \frac{q}{n} < \exp(-\frac{q}{n})$ and the approximation for the Harmonic Series $\sum_{k=1}^{n} \frac{1}{k} \approx \ln(n)$ for sufficiently large $n$.

   **Solution:** TODO

2. **Taylor Series.** Taylor series expansions are a method of approximating a function near a point using polynomial terms. The Taylor expansion for a function $f(x)$ at point $a$ is given by:

   $$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots$$

   This can also be rewritten as $f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$.

   (a) Calculate the first three terms of the Taylor series for $f(x) = \ln(1+x)$ centered at $a = 0$.

   (b) **[OPTIONAL]** The gamma function is defined as

   $$\Gamma(z) = \int_0^{\infty} t^{z-1}e^{-t}dt.$$

   Calculate and use a first-order Taylor expansion of the gamma function centered at 1 to approximate $\Gamma(1.1)$. You should express your answer in terms of the Euler-Mascheroni constant $\gamma$.
   You may use the fact that $\Gamma(x+1)$ interpolates the factorial function without proof.

   **Solution:** TODO

3. Consider a twice continuously differentiable function $f : \mathbb{R}^n \mapsto \mathbb{R}$. Suppose this function admits a unique global optimum $x^* \in \mathbb{R}^n$. Suppose that for some spherical region $\mathcal{X} = \{x \mid \|x - x^*\|^2 \leq D\}$ around $x^*$ for some constant $D$, the Hessian matrix $H$ of the function $f(x)$ is PSD and its maximum eigenvalue is 1. Prove that

   $$f(x) - f(x^*) \leq \frac{D}{2}$$

   for every $x \in \mathcal{X}$. *Hint*: Look up Taylor's Theorem with Remainder. Use Mean Value Theorem on the second order term instead of the first order term, which is what is usually done.

   **Solution:** TODO

# 7 Hands-on with data

In the following problem, you will use two simple datasets to walk through the steps of a standard machine learning workflow: inspecting your data, choosing a model, implementing it, and verifying its accuracy. We have provided two datasets in the form of numpy arrays: `dataset_1.npy` and `dataset_2.npy`. You can load each using NumPy's `np.load` method[4]. You can plot figures using Matplotlib's `plt.plot` method[5].

Each dataset is a two-column array with the first column consisting of $n$ scalar inputs $X \in \mathbb{R}^{n\times 1}$ and the second column consisting of $n$ scalar labels $Y \in \mathbb{R}^{n\times 1}$. We denote each entry of $X$ and $Y$ with subscripts:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

and assume that $y_i$ is a (potentially stochastic) function of $x_i$.

(a) It is often useful to visually inspect your data and calculate simple statistics; this can detect dataset corruptions or inform your method. For both datasets:

(i) Plot the data as a scatter plot.

(ii) Calculate the correlation coefficient between X and Y:

$$\rho_{X,Y} = \frac{\text{Cov}(X,Y)}{\sigma_X \sigma_Y}$$

in which $\text{Cov}(X,Y)$ is the covariance between $X$ and $Y$ and $\sigma_X$ is the standard deviation of $X$.

Your solution may make use of the NumPy library only for arithmetic operations, matrix-vector or matrix-matrix multiplications, matrix inversion, and elementwise exponentiation. It may not make use of library calls for calculating means, standard deviations, or the correlation coefficient itself directly.

**Solution:** TODO

(b) We would like to design a function that can predict $y_i$ given $x_i$ and then apply it to new inputs. This is a recurring theme in machine learning, and you will soon learn about a general-purpose framework for thinking about such problems. As a preview, we will now explore one of the simplest instantiations of this idea using the class of linear functions:

$$\hat{Y} = Xw. \tag{2}$$

The parameters of our function are denoted by $w \in \mathbb{R}$. It is common to denote predicted variants of quantities with a hat, so $\hat{Y}$ is a predicted label whereas $Y$ is a ground truth label.

We would like to find a $w^*$ that minimizes the **squared error** $\mathcal{J}_{\text{SE}}$ between predictions and labels:

$$w^* = \arg\min_w \mathcal{J}_{\text{SE}}(w) = \arg\min_w \|Xw - Y\|_2^2.$$

Derive $\nabla_w \mathcal{J}_{\text{SE}}(w)$ and set it equal to 0 to solve for $w^*$. (Note that this procedure for finding an optimum relies on the convexity of $\mathcal{J}_{\text{SE}}$. You do not need to show convexity here, but it is a useful exercise to convince yourself this is valid.)

**Solution:** TODO

[4] https://numpy.org/doc

[5] https://matplotlib.org/stable/users/explain/quick_start.html

(c) Your solution $w^*$ should be a function of $X$ and $Y$. Implement it and report its **mean squared error** (MSE) for **dataset 1**. The mean squared error is the objective $\mathcal{J}_{\text{SE}}$ from part (b) divided by the number of datapoints:

$$\mathcal{J}_{\text{MSE}}(w) = \frac{1}{n}\|Xw - Y\|_2^2.$$

Also visually inspect the model's quality by plotting a line plot of predicted $\hat{y}$ for uniformly-spaced $x \in [0, 10]$. Keep the scatter plot from part (a) in the background so that you can compare the raw data to your linear function. Does the function provide a good fit of the data? Why or why not?

**Solution:** TODO

(d) We are now going to experiment with constructing new *features* for our model. That is, instead of considering models that are linear in the inputs, we will now consider models that are linear in some (potentially nonlinear) transformation of the data:

$$\hat{Y} = \Phi w = \begin{bmatrix} \phi(x_1)^\top \\ \phi(x_2)^\top \\ \vdots \\ \phi(x_n)^\top \end{bmatrix} w,$$

where $\phi(x_i), w \in \mathbb{R}^m$. Repeat part (c), providing both the mean squared error of your predictor and a plot of its predictions, for the following features on **dataset 1**:

$$\phi(x_i) = \begin{bmatrix} x_i \\ 1 \end{bmatrix}.$$

How do the plotted function and mean squared error compare? (A single sentence will suffice.)
*Hint:* the general form of your solution for $w^*$ is still valid, but you will now need to use features $\Phi$ where you once used raw inputs $X$.

**Solution:** TODO

(e) Now consider the quadratic features:

$$\phi(x_i) = \begin{bmatrix} x_i^2 \\ x_i \\ 1 \end{bmatrix}.$$

Repeat part (c) with these features on **dataset 1**, once again providing short commentary on any changes.

**Solution:** TODO

(f) Repeat parts (c) - (e) with **dataset 2**.

**Solution:** TODO

(g) Finally, we would like to understand which features $\Phi$ provide us with the best model. To that end, you will implement a method known as $k$-fold cross validation. The following are instructions for this method; deliverables for part (g) are at the end.

(i) Split **dataset 2** randomly into $k = 4$ equal sized subsets. Group the dataset into 4 distinct training / validation splits by denoting each subset as the validation set and the remaining subsets as the training set for that split.

(ii) On each of the 4 training / validation splits, fit linear models using the following 5 polynomial feature sets:

$$\phi_1(x_i) = \begin{bmatrix} x_i \\ 1 \end{bmatrix} \quad \phi_2(x_i) = \begin{bmatrix} x_i^2 \\ x_i \\ 1 \end{bmatrix} \quad \phi_3(x_i) = \begin{bmatrix} x_i^3 \\ x_i^2 \\ x_i \\ 1 \end{bmatrix} \quad \phi_4(x_i) = \begin{bmatrix} x_i^4 \\ x_i^3 \\ x_i^2 \\ x_i \\ 1 \end{bmatrix} \quad \phi_5(x_i) = \begin{bmatrix} x_i^5 \\ x_i^4 \\ x_i^3 \\ x_i^2 \\ x_i \\ 1 \end{bmatrix}$$

This step will produce 20 distinct $w^*$ vectors: one for each dataset split and featurization $\phi_j$.

(iii) For each feature set $\phi_j$, average the training and validation mean squared errors over all training splits.

It is worth thinking about what this extra effort has bought us: by splitting the dataset into subsets, we were able to use all available datapoints for model fitting while still having held-out datapoints for evaluation for any particular model.

**Deliverables for part (g):** Plot the training mean squared error and the validation mean squared error on the same plot as a function of the largest exponent in the feature set. Use a log scale for the $y$-axis. Which model does the training mean squared error suggest is best? Which model does the validation mean squared error suggest is best?

**Solution:** TODO

# 8 Honor Code

1. **List all collaborators. If you worked alone, then you must explicitly state so.**

   **Solution:** TODO

2. **Declare and sign the following statement**:

   *"I certify that all solutions in this document are entirely my own and that I have not looked at anyone else's solution. I have given credit to all external sources I consulted."*

   *Signature* : _____

   While discussions are encouraged, *everything* in your solution must be your (and only your) creation. Furthermore, all external material (i.e., *anything* outside lectures and assigned readings, including figures and pictures) should be cited properly. We wish to remind you that the consequences of academic misconduct are *particularly severe*!

   **Solution:** TODO