ECE C147/C247, Winter 2025 **Homework #1**
Neural Networks & Deep Learning Prof. J.C. Kao
UCLA ECE TAs: B. Qu, K. Pang, S. Dong, S. Rajesh, T. Monsoor, X. Yan

Due Monday, 20 Jan 2025, by 11:59pm to Gradescope.
Covers material up to Introduction to machine learning refresher 1.
100 points total.

1. (25 points) **Linear algebra refresher.**

   (a) (12 points) Let $\mathbf{Q}$ be a real orthogonal matrix.

      i. (3 points) Show that $\mathbf{Q^T}$ and $\mathbf{Q^{-1}}$ are also orthogonal.
      ii. (3 points) Show that $\mathbf{Q}$ has eigenvalues with norm 1.
      iii. (3 points) Show that the determinant of $\mathbf{Q}$ is either +1 or -1.
      iv. (3 points) Show that $\mathbf{Q}$ defines a length preserving transformation.

   (b) (8 points) Let $\mathbf{A}$ be a matrix.

      i. (4 points) What is the relationship between the singular vectors of $\mathbf{A}$ and the eigenvectors of $\mathbf{A}\mathbf{A}^T$? What about $\mathbf{A}^T\mathbf{A}$?
      ii. (4 points) What is the relationship between the singular values of $\mathbf{A}$ and the eigenvalues of $\mathbf{A}\mathbf{A}^T$? What about $\mathbf{A}^T\mathbf{A}$?

   (c) (5 points) True or False. Partial credit on an incorrect solution may be awarded if you justify your answer.

      i. Every linear operator in an n-dimensional vector space has n distinct eigenvalues.
      ii. A non-zero sum of two eigenvectors of a matrix $\mathbf{A}$ is an eigenvector.
      iii. If a matrix $\mathbf{A}$ has the positive semidefinite property, i.e., $\mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0$ for all $\mathbf{x}$, then its eigenvalues must be non-negative.
      iv. The rank of a matrix can exceed the number of distinct non-zero eigenvalues.
      v. A non-zero sum of two eigenvectors of a matrix $\mathbf{A}$ corresponding to the same eigenvalue λ is always an eigenvector.

2. (25 points) **Probability refresher.**

   (a) (10 points) A and B are involved in a duel. The rules of the duel are that they are to pick up their guns and shoot at each other simultaneously. If one or both are hit, then the duel is over. If both shots miss, then they repeat the process. Suppose that the results of the shots are independent and that each shot of A will hit B with probability p_A and each shot of B will hit A with probability p_B. What is:

      i. (2 points) the probability that A is not hit?
      ii. (2 points) the probability that both duelists are hit?
      iii. (2 points) the probability that the duel ends after the n^{th} round of shots?

iv. (2 points) the conditional probability that the duel ends after the n^{th} round of shots given that A is not hit?

v. (2 points) the conditional probability that the duel ends after the n^{th} round of shots given that both duelists are hit?

(b) (5 points) Let X be a binary signal, such that $P(X = +1) = P(X = -1) = 0.5$. Suppose X is sent across a noisy channel, with noise N modeled by a zero-mean Gaussian distribution with variance σ^2, where the noise is independent of the signal that was sent. The received signal is $Y = X + N$.

i. (2 points) Find the conditional PDFs of Y given both $\{X = +1\}$ and $\{X = -1\}$.

ii. (2 points) Suppose a detector compares the received signal Y to a fixed threshold γ to decide which signal was sent. Specifically, the detector decides that $X = +1$ was sent if $Y \geq \gamma$ and that $X = -1$ was sent otherwise. For a given threshold γ, express the probability of error in terms of the Φ function, γ, and σ. Recall, if $Z \sim \mathcal{N}(0, 1)$, then

$$P(Z \leq z) = \Phi(z)$$

iii. (1 point) What is the optimal value of γ to minimize the probability of error? For that optimal value of γ, what is the probability of error? Leave your answer for the optimal value of γ as a real number and the probability of error in terms of the Φ function and σ.

(c) (5 points) There is a screening test for lung cancer that looks at the level of LSA (lung specific antigen) in the blood. There are a number of reasons besides lung cancer that a man can have elevated LSA levels. In addition, many types of lung cancer develop so slowly that they are never a problem. Unfortunately, there is currently no test to distinguish the different types and using the test is controversial because it's hard to quantify the accuracy rates and the harm done by false positives. For this problem, we will call a positive test a true positive if it catches a dangerous type of lung cancer. Also, we will assume the following numbers:

- Rate of dangerous type of lung cancer among men over 30 = 0.0005
- True positive rate for the test = 0.9
- False positive rate for the test = 0.01

Suppose you randomly select a man over 30 and perform a screening test.

i. (3 points) What is the probability that the man has a dangerous type of the disease given that he had a positive test?

ii. (2 points) What is the probability that the man has a dangerous type of the disease given that he had a negative test?

(d) (5 points) A family with three daughters and three sons needs to go to the grocery store. Besides the father, who is driving the car, exactly three of the children can come along to the grocery store with him. Suppose that the three children to join the father are chosen randomly, and all such choices are equally likely. Let X denote the number of daughters who accompany the father to the grocery. Let $X_i = 1$ if the i^{th} child who joins the father is a girl, and $X_i = 0$ otherwise.

i. (1 point) Find $Cov(X_i, X_i) = E(X_i X_i) - E(X_i)E(X_i)$, for $1 \leq i \leq 3$. Leave your answer as a fraction.

ii. (2 points) Find $Cov(X_i, X_j) = E(X_i X_j) - E(X_i)E(X_j)$, for $1 \leq i, j \leq 3$ and $i \neq j$. Leave your answer as a fraction.

iii. (2 points) Find $Var(X)$. Leave your answer as a fraction.

3. (10 points) **Multivariate derivatives.**

(a) (1 points) Let $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$, and $\mathbf{A} \in \mathbb{R}^{n \times m}$. What is $\nabla_{\mathbf{x}} \mathbf{x}^T \mathbf{A} \mathbf{y}$?

(b) (1 points) Let $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$, and $\mathbf{A} \in \mathbb{R}^{n \times m}$. What is $\nabla_{\mathbf{y}} \mathbf{x}^T \mathbf{A} \mathbf{y}$?

(c) (1 points) Let $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$, and $\mathbf{A} \in \mathbb{R}^{n \times m}$. What is $\nabla_{\mathbf{A}} \mathbf{x}^T \mathbf{A} \mathbf{y}$?

(d) (1 points) Let $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{n \times n}$, and let $f = \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x}$. What is $\nabla_{\mathbf{x}} f$?

(e) (1 points) Let $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times n}$ and $f = \text{tr}(\mathbf{AB})$. What is $\nabla_{\mathbf{A}} f$?

(f) (2 points) Let $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times n}$ and $f = \text{tr}(\mathbf{BA} + \mathbf{A}^T \mathbf{B} + \mathbf{A}^2 \mathbf{B})$. What is $\nabla_{\mathbf{A}} f$?

(g) (3 points) Let $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times n}$ and $f = \|\mathbf{A} + \lambda \mathbf{B}\|_F^2$. What is $\nabla_{\mathbf{A}} f$?

4. (10 points) **Deriving least-squares with matrix derivatives.**
In least-squares, we seek to estimate some multivariate output $\mathbf{y}$ via the model

$$\hat{\mathbf{y}} = \mathbf{W}\mathbf{x}$$

In the training set we're given paired data examples $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ from $i = 1, \ldots, n$. Least-squares is the following quadratic optimization problem:

$$\min_{\mathbf{W}} \quad \frac{1}{2} \sum_{i=1}^{n} \left\| \mathbf{y}^{(i)} - \mathbf{W}\mathbf{x}^{(i)} \right\|^2$$

Derive the optimal $\mathbf{W}$.

Where $\mathbf{W}$ is a matrix, and for each example in the training set, both $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ $\forall i = 1, \ldots, n$ are vectors.
Hint: you may find the following derivatives useful:

$$\begin{aligned} \frac{\partial \text{tr}(\mathbf{WA})}{\partial \mathbf{W}} &= \mathbf{A}^T \\ \frac{\partial \text{tr}(\mathbf{WAW}^T)}{\partial \mathbf{W}} &= \mathbf{WA}^T + \mathbf{WA} \end{aligned}$$

5. (10 points) **Regularized least squares**

In lecture, we worked through the following least squares problem

$$\arg\min_{\theta} \ \frac{1}{2} \sum_{i=1}^{N} (y^{(i)} - \theta^T \hat{\mathbf{x}}^{(i)})^2$$

However, the least squares has a tendency to overfit the training data. One common technique used to address the overfitting problem is regularization. In this problem, we work through one of the regularization techniques namely ridge regularization which is also known as the regularized least squares problem. In the regularized least squares we solve the following optimization problem

$$\arg\min_{\theta} \ \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - \theta^T \hat{\mathbf{x}}^{(i)})^2 + \frac{\lambda}{2}\|\theta\|_2^2$$

where λ is a tunable regularization parameter. From the above cost function it can be observed that we are seeking least squares solution with a smaller 2-norm. Derive the solution to the regularized least squares problem, i.e Find θ^*.

6. (20 points) **Linear regression.**
Complete the Jupyter notebook `linear_regression.ipynb`. Print out the Jupyter notebook as a PDF and submit it to Gradescope.

ECE C247
HW1
Sujit Silas Armstrong Suthahar

1) (a) i. Square matrix is orthogonal, if & only if:

$$Q^T Q = Q Q^T = I$$

$$Q^T Q = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

If orthogonal, $Q^T = Q^{-1}$, where $Q Q^{-1} = I$

$$Q^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \frac{1}{\det(Q)} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = -1 \cdot \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$\therefore Q^{-1}$ is also orthogonal

ii. We know for an orthogonal matrix Q, $Q^T Q = I$
Let λ be the eigenvalue of Q & v be the corresponding eigenvector, so $Qv = \lambda v$

$$\| Qv \| = \| \lambda v \|$$

$\|v\| = \|\lambda v\|$, if $v \neq 0$, then $\|v\| \neq 0$, so
$\|\lambda\| = 1$

Eg: $\|Q\| = \sqrt{(1)^2 + (-1)^2} = 1$

$\therefore Q$ has λs with norm 1

iii. $\det(Q^T Q) = \det(I)$, where $\det(I) = 1$.

Can also be written as, $\det(Q^T) \cdot \det(Q) = \det(I)$

Since $\det(Q^T) = \det(Q)$, we get $(\det(Q))^2 = 1$

$\therefore \det(Q) = \pm 1$

iv. Let's transform Q by v. The norm of v is defined by $\|v\| = \sqrt{v^T v}$.

$$\|Qv\| = \sqrt{(Qv)^T (Qv)} = \sqrt{(Q^T v^T) \cdot (Q \cdot v)}$$

$$= \sqrt{(Q^T \cdot Q)(v^T v)}$$

$$= \sqrt{I \cdot (v^T v)}$$

$$= \sqrt{v^T v}$$

Thus $\|Qv\| = \|v\|$, proves that Q preserves length.
For any orthogonal matrix, its eigenvectors are unitary (e.g. have length 1.)
The eigenvalues are placed on the unit circle.

b) i. Any $n \times m$ matrix A can be written as or decomposed into

$$A = U \Sigma V^T,$$

where

U = eigenvectors AA^T $n \times n$ (left singular vectors)
$\Sigma = \sqrt{\text{diag}(\text{eig}(AA^T))}$ $n \times m$ (singular values)
V = eigenvectors of $A^T A$ $m \times m$ (right singular vectors)

$$AA^T = (U \Sigma V^T)(U \Sigma V^T)^T = U \Sigma V^T \; U^T \Sigma^T V^{TT}$$

$$= U \Sigma (\Sigma^T) U^T$$

$$= U (\Sigma)^2 U^T$$

The eigenvectors of AA^T are columns of U and
the eigenvalues are the squares of the singular values of A.

$$A^T A = (U \Sigma V^T)^T (U \Sigma V^T) = U^T U \; \Sigma^T (\Sigma) V V^T$$

$$= V (\Sigma)^2 V^T$$

The eigenvectors of $A^T A$ are the columns of V and
the eigenvalues are the squares of the singular values of A.

ii. The singular values of matrix A are the square roots of the non-zero eigenvalues of both A^TA & AA^T.

Both A^TA & AA^T share the same set of eigenvalues, since the singular values are shared.

If $\sigma^1, \sigma^2, \ldots$ are singular values of A, then the eigenvalues of A^TA and AA^T are $\sigma_1^2, \sigma_2^2, \ldots$.

C) i. False, only if the vector spaces are linearly independent (e.g. I matrix has $\lambda = 1$)

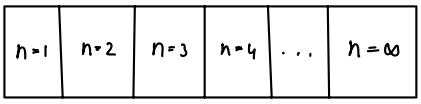
ii. False, Only true if two eigenvectors correspond to the same eigenvalue.

iii. True, A pos. semi-def. $\rightarrow$ eig $\left(\frac{A + A^H}{2}\right) \geq 0$

iv. True, The rank of a matrix is dependent on the number of linearly independent rows or columns. $A = \text{diag}(1, 1, 1) \rightarrow$ Rank 3, but only 1 distinct eigenvalue.

v. True, If the vectors are linearly independent, their sum will be non-zero.

2. a) i. Since $n = 1, 2, 3, 4 \ldots$, where n denotes the round in which the duel ends, partitions the sample space.



Events:

P_1 = A hits B $\quad P(\text{A not hit}) = \sum_{n=1}^{\infty} P(\text{A not hit}, n)$

P_2 = B hits A $\quad$ By Bernoulli trial, Outcomes are defined by success or failure:

failure $\rightarrow 1 - P$

Success $\rightarrow P$

Duel modeled as a infite sum of geometric series

$$= [(1-P_1)\ (1-P_2)]^{(n-1)}\ P_1(1-P_2)$$

$$= P_1(1-P_1)^{n-1}(1-P_2)^n$$

$$P(\text{A not hit}) = P_1 \sum_{n=1}^{\infty} (1-P_1)^{(n-1)}\ (1-P_2)^n$$

$$= \frac{P_1(1-P_2)}{1-(1-P_1)(1-P_2)}$$

ii. Following a similar setup

$$P(\text{both hit}) = \sum_{n=1}^{\infty} \left[(1-P_1)(1-P_2)\right]^{n-1} P_1 P_2$$

$$= \frac{P_1 P_2}{1-(1-P_1)(1-P_2)}$$

iii. Sample space for game ending in n rounds

Three way the duel can end
→ A hits
→ B hits
→ A & B hit

$$P(\text{game ends in } n \text{ rounds}) = P(n, A \text{ hit}) + P(n, B \text{ hit}) + P(n, \text{both hit})$$

$$= \left[(1-P_1)(1-P_2)\right]^{(n-1)} (1-P_2) P_2 +$$

$$\left[(1-P_1)(1-P_2)\right]^{(n-1)} P_1 (1-P_2) +$$

$$\left[(1-P_1)(1-P_2)\right]^{(n-1)} P_1 P_2$$

$$= \left[(1-P_1)(1-P_2)\right]^{(n-1)} \left[1-(1-P_1)(1-P_2)\right]$$

iv. E: duel ends after n^{th} round

$$P(E \mid A \text{ not hit}) = \frac{P(A \text{ not hit} \mid E)\, P(E)}{P(A \text{ not hit})}$$

we know, $P(A \text{ not hit}) = \frac{P_1(1-P_2)}{1-(1-P_1)(1-P_2)}$,

and $P(E) = \left[(1-P_1)(1-P_2)\right]^{(n-1)} \left[1-(1-P_1)(1-P_2)\right]$

$$P(E \text{ and } A \text{ not hit}) = \left[(1-P_1)(1-P_2)\right]^{(n-1)} \left[1-(1-P_1)(1-P_2)\right] \times \frac{P_1(1-P_2)}{1-(1-P_1)(1-P_2)}$$

let $q = (1-P_1)(1-P_2)$

$$= q^{n-1}\,[1-q] \times \frac{P_1(1-P_2)}{1-q}$$

$$= q^{n-1} \times P_1(1-P_2)$$

$$= [(1-P_1)(1-P_2)]^{n-1} \times P_1(1-P_2)$$

$$= P_1(1-P_1)^{n-1}(1-P_2)^n$$

$$P(E \mid A \text{ not hit}) = \frac{P_1(1-P_1)^{n-1}(1-P_2)^n}{(1-P_2)^n} = P_1(1-P_1)^{n-1}$$

V. E: duel ends after n^{th} round

$$P(E \mid \text{both hit}) = \frac{P(\text{both hit} \mid E)\,P(E)}{P(\text{both hit})}$$

we know, $P(\text{both hit}) = \frac{P_1P_2}{1-(1-P_1)(1-P_2)}$,

and $P(E) = [(1-P_1)(1-P_2)]^{(n-1)}\,[1-(1-P_1)(1-P_2)]$,

$P(\text{both hit and } E) = P(\text{both hit}) \cdot P(E)$

$$= \frac{P_1P_2}{1-(1-P_1)(1-P_2)}\,[(1-P_1)(1-P_2)]^{(n-1)}\,[1-(1-P_1)(1-P_2)]$$

$$= P_1P_2\,(1-P_1)^{(n-1)}(1-P_2)^{(n-1)}$$

$$P(E \mid \text{both hit}) = \frac{P_1P_2\,(1-P_1)^{(n-1)}(1-P_2)^{(n-1)}}{P_1P_2} = (1-P_1)^{(n-1)}(1-P_2)^{(n-1)}$$

b) i.

Conditional = joint / marginal

$P(X=+1) = P(X=-1) = 0.5 \qquad X \rightarrow N \sim N(0,\sigma^2) \rightarrow Y = X+N$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\,e^{\frac{-(x-\mu)^2}{2\sigma^2}} \qquad z = \frac{x-\mu}{\sigma}$$

conditional PDF of Y given $\{X = +1\}$ and $\{X = -1\}$

If $X = +1$, $Y = 1 + N$, $Y|(X = +1) \sim N(1, \sigma^2)$

$$f(Y|X=+1) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(Y-1)^2}{2\sigma^2}}$$

If $X = -1$, $Y = -1 + N$, $Y|(X = -1) \sim N(-1, \sigma^2)$

$$f(Y|X=-1) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(Y+1)^2}{2\sigma^2}}$$

ii. If $Y \geq \gamma \rightarrow X = +1$

If $Y < \gamma \rightarrow X = -1$

$Y \geq \gamma$	$Y < \gamma$

S → partitions the sample space

$$P(\text{Error}) = P(X=+1) \cdot P(Y < \gamma | X=+1) + P(X=-1) \cdot P(Y \geq \gamma | X=-1)$$

We know $P(X=+1) = P(X=-1) = 0.5$

$$P(\text{Error}) = 0.5 \cdot P(Y < \gamma | X=+1) + 0.5 \cdot P(Y \geq \gamma | X=-1)$$

Expressing in terms of ϕ function:

$P(Y < \gamma | X=1) + P(Y \geq \gamma | X=-1) = 1$

$$P(Y < \gamma | X=+1) = P(1+N < \gamma) = P(N < Y-1) = P\left(\frac{N}{\sigma} < \frac{\gamma-1}{\sigma}\right) = \phi\left(\frac{\gamma-1}{\sigma}\right)$$

$$P(Y \geq \gamma | X=-1) = P(-1+N \geq \gamma) = P(N \geq \gamma+1) = P\left(\frac{N}{\sigma} \geq \frac{\gamma+1}{\sigma}\right) = 1 - \phi\left(\frac{\gamma+1}{\sigma}\right)$$

$$\therefore P(\text{Error}) = 0.5 \cdot \phi\left(\frac{\gamma-1}{\sigma}\right) + 0.5 \cdot \left[1 - \phi\left(\frac{\gamma+1}{\sigma}\right)\right]$$

iii. To minimize the probability of error, we should choose a γ that equalizes the error from both cases. In our case, setting $\gamma = 0$ will be the optimal threshold because the prior probabilities range from $X = +1$ & $X = -1$

$$\therefore P(\text{Error}) = 0.5\,\phi\left(\frac{0-1}{\sigma}\right) + 0.5\left[1 - \phi\left(\frac{0+1}{\sigma}\right)\right]$$

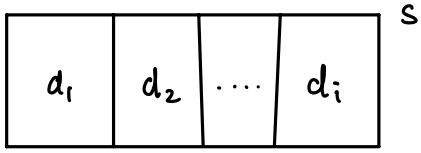
$$= 0.5\ \phi\left(-\frac{1}{\sigma}\right) + 0.5\left[1 - \phi\left(\frac{1}{\sigma}\right)\right]$$

$$= 1 - \phi\left(\frac{1}{\sigma}\right)$$

c) i. $P(D) = 0.0005$

TPR → $P(T|D) = 0.9$

FPR → $P(T|D^c) = 0.01$



$$P(D) = \sum_{i=1}^{n} d_i$$

$$P(D|T) = \frac{P(T|D)\,P(D)}{P(T)}$$

$$P(T) = P(T|D)\,P(D) + P(T|D^c)\,P(D^c)$$

$$P(D|T) = \frac{0.9 \times 0.0005}{(0.9 \times 0.0005) + (0.01 \times 0.9995)}$$

$$= 0.04308$$

ii) $P(D|T^c) = \frac{P(T^c|D)\,P(D)}{P(T^c|D)\,P(D) + P(T^c|D^c)\,P(D^c)}$

$$= \frac{0.1 \times 0.0005}{(0.1 \times 0.0005) + (0.99)(0.0005)}$$

d) $D = 3$
$S = 3$

$$X_i = \begin{cases} 1 & \text{if the } i^{th} \text{ child is a girl} \\ 0 & \text{if the } i^{th} \text{ child is a boy} \end{cases}$$

i. $Cov(X_i, X_i) = E(X_i X_i) - E(X_i)\,E(X_i)$, for $1 \le i \le 3$

$$Cov(X_i, X_i) = Var(X_i) = E(X_i) - (E(X_i))^2 = \mu - \sigma^2$$

Bau can,

$$P(X_i = 1) = \frac{\binom{5}{2}}{\binom{6}{3}} = \frac{1}{2}$$

$$Cov(X_i, X_i) = \frac{1}{2} - \left(\frac{1}{2}\right)^2 = \frac{1}{4}$$

ii. $Cov(X_i, X_j) = E(X_i, X_j) - E(X_i)E(X_j)$, for $1 \le i, j \le 3$ and $i \neq j$

We know, $E(X_i) = E(X_j) = \frac{1}{2}$

$\therefore E(X_i)E(X_j) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$

$X_i X_j = 1$ occurs when both i^{th} & j^{th} selected children are daughters

$E(X_i, X_j) = P(\text{both } i \text{ \& } j \text{ are daughters})$

$P(X_i, X_j = 1) = \frac{\text{ways to choose 2 daughters \& 1 other child}}{\text{ways to choose 3 out of 6}}$

$$P(X_i X_j = 1) = \frac{\binom{3}{2}\binom{3}{1}}{\binom{6}{3}} = \frac{9}{20}$$

$$\therefore Cov(X_i, X_j) = \frac{9}{20} - \frac{1}{4} = \frac{9-5}{20} = \frac{4}{20} = \frac{1}{5}$$

iii. $Var(X)$

$X \rightarrow$ no. of daughters that accompany the father

$$X = X_1 + X_2 + X_3$$

$$Var(X) = \sum_{i=1}^{3} Var(X_i) + \sum_{1 \le i < j \le 3} Cov(X_i, X_j)$$

we know, $Var(X_i) = \frac{1}{4}$

$$\therefore \sum_{i=1}^{3} Var(X_i) = \frac{3}{4}$$

we know, $Cov(X_i, X_j) = \frac{1}{5}$

$$\therefore \sum_{i=1}^{3} Cov(X_i, X_j) = \frac{3}{5}$$

$$\therefore Var(X) = \frac{3}{4} + \frac{3}{5} = \frac{15+12}{20} = \frac{27}{20}$$

3. Multivariate derivatives

a) $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times m}$

$\nabla_x x^T A y$

$(1 \times n)(n \times m)(m \times 1) \rightarrow$ Scalar

$\nabla_x x^T A y \in \mathbb{R}^1$

$$\boxed{\nabla_x x^T A y = Ay}$$

b) $x \in \mathbb{R}^n$
$y \in \mathbb{R}^m$
$A \in \mathbb{R}^{n \times m}$
$\nabla_y x^T A y \in \mathbb{R}^1$

$$\boxed{\nabla_y x^T A y = A^T x}$$
$(1 \times n)(n \times m)(m \times 1)$

c) $x \in \mathbb{R}^n$
$y \in \mathbb{R}^m$
$A \in \mathbb{R}^{n \times m}$

$$\boxed{\nabla_A (x^T A y) = x y^T}$$

$$\nabla_A x^T A y = \begin{bmatrix} \frac{\partial x^T A y}{\partial a_{1,1}} & \frac{\partial x^T A y}{\partial a_{1,2}} & \cdots & \frac{\partial x^T A y}{\partial a_{1,m}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x^T A y}{\partial a_{n,1}} & \cdots & \cdots & \frac{\partial x^T A y}{\partial a_{n,m}} \end{bmatrix}$$

d) $x \in \mathbb{R}^n$
$A \in \mathbb{R}^{n \times n}$
$b \in \mathbb{R}^n$

$\nabla_x f$ where $f = x^T A x + b^T x$

$x^T A x \rightarrow$ quadratic

differentiating $x^T A x$ w.r.t $x \rightarrow Ax + A^T$

assuming A is symmetric

$$\boxed{\nabla_x f = Ax + A^T x + b}$$

e) $A, B \in \mathbb{R}^{n \times n}$

$\nabla_A f$, where $f = tr(AB)$

$$tr(AB) = tr(BA) = tr\left(\begin{bmatrix} -b_1^T \\ -b_2^T \\ \vdots \end{bmatrix} \begin{bmatrix} | & | & | & \\ a_1 & a_2 & a_3 & \cdots \\ | & | & | & \end{bmatrix} \right)$$

derivation of f w.r.t A

$$\boxed{\nabla_A f = B^T}$$

f) $A, B \in \mathbb{R}^{n \times n}$

$$\nabla_A f = tr(BA + A^T B + A^2 B)$$

$$tr(BA) : \nabla_A = B^T$$

$$tr(A^T B) = tr(BA) : \nabla_A = B^T$$

$$tr(A^2 B) : \nabla_A = AB + BA$$

$$\boxed{\therefore \nabla_A f = B^T + B + (AB + BA)^T}$$

g) $\nabla_A f$, where $f = \|A + \lambda B\|_F^2$

$A, B \in \mathbb{R}^{n \times n}$, $\lambda \in \mathbb{R}$

Forberius norm : $\|M\|_F^2 = tr(M^TM)$

$$\|A+\lambda B\|_F^2 = tr((A+\lambda B)^T(A+\lambda B))$$

$$= tr(A^TA) + 2\lambda tr(A^TB) + \lambda^2 tr(B^TB)$$

derivation w.r.t. A $A^TA \rightarrow 2A$

derivation w.r.t. A $A^TB \rightarrow B$

$$\boxed{\nabla_A f = 2(A+\lambda B)}$$

4. Deriving least squares with matrix derivations

$\hat{y} = Wx$, where W is a matrix

$$\min_W \frac{1}{2}\sum_{i=1}^{n} \|y^{(i)} - Wx^{(i)}\|^2$$

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{n} (y^{(i)} - Wx^{(i)})^T (y^{(i)} - Wx^{(i)})$$

$$= \frac{1}{2}\sum_{i=1}^{n} (y^{(i)T}y^{(i)} - y^{(i)T}Wx^{(i)} - Wx^{(i)T}y^{(i)} + W^Tx^{(i)T}Wx^{(i)})$$

$$= \frac{1}{2}\sum_{i=1}^{n} (y^{(i)T}y^{(i)} - 2y^{(i)T}Wx^{(i)} + x^{(i)T}W^TWx^{(i)})$$

Let, $\mathbb{X} = [x^{(1)}, x^{(2)}, x^{(3)}, \dots x^{(i)}]$ -> data

$\mathbb{Y} = [y^{(1)}, y^{(2)}, y^{(3)}, \dots y^{(i)}]$ -> output , be matrices

$$\mathcal{L} = \frac{1}{2}(tr(\mathbb{Y}^T\mathbb{Y}) - 2tr(\mathbb{Y}^TW\mathbb{X}) + tr(W\mathbb{X}\mathbb{X}^TW^T))$$

we know, $\frac{\partial tr(WA)}{\partial W} = A^T$, $\frac{\partial tr(WAW^T)}{\partial W} = WA^T + WA$

$$\nabla_W \mathcal{L} = 0 - 2\mathbb{Y}\mathbb{X}^T + \mathbb{Y}\mathbb{X}^T + W\mathbb{X}\mathbb{X}^T = 0$$

$$\nabla_W \mathcal{L} = -\mathcal{Y}\mathcal{X}^T + W\mathcal{X}\mathcal{X}^T = 0$$

$$W\mathcal{X}\mathcal{X}^T = \mathcal{Y}\mathcal{X}^T$$

$$W = (\mathcal{X}\mathcal{X}^T)^{-1}\mathcal{X}^T\mathcal{Y}$$

5. Regularized least squares

$$\arg\min_{\theta} \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - \theta^T\hat{x}^{(i)})^2 + \frac{\lambda}{2}\|\theta\|_2^2$$

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - \theta^T\hat{x}^{(i)})^T(y^{(i)} - \theta^T\hat{x}^{(i)}) + \frac{\lambda}{2}\|\theta\|_2^2$$

$$= \frac{1}{2}\sum_{i=N}^{N}(y^{(i)T}y^{(i)} - y^{(i)T}\theta^T\hat{x}^{(i)} - \theta\hat{x}^{(i)T}y^{(i)} + \theta\hat{x}^{(i)T}\theta^T\hat{x}^{(i)}) + \frac{\lambda}{2}\|\theta\|_2^2$$

$$= \frac{1}{2}\sum_{i=N}^{N}(y^{(i)T}y^{(i)} - 2y^{(i)T}x^{(i)}\theta + \theta^T\hat{x}^{(i)T}\hat{x}\theta) + \frac{\lambda}{2}\|\theta\|_2^2$$

Let, $\mathcal{X} = [x^{(1)}, x^{(2)}, x^{(3)}, \ldots x^{(i)}]$ → data

$\mathcal{Y} = [y^{(1)}, y^{(2)}, y^{(3)}, \ldots y^{(i)}]$ → output, be matrices

$$= \frac{1}{2}(\mathcal{Y}^T\mathcal{Y} + 2\mathcal{Y}^T\mathcal{X}\theta + \theta^T\mathcal{X}^T\mathcal{X}\theta) + \frac{\lambda}{2}\|\theta\|_2^2$$

$$\nabla_\theta\mathcal{L} = \frac{1}{2}(-2\mathcal{X}^T\mathcal{Y} + 2\mathcal{X}^T\mathcal{X}\theta) + \lambda\theta$$

$$= -\mathcal{X}^T\mathcal{Y} + \mathcal{X}^T\mathcal{X}\theta + \lambda\theta$$

$$\mathcal{X}^T\mathcal{Y} = \theta(\mathcal{X}^T\mathcal{X} + \lambda I)$$

$$\theta = (\mathcal{X}^T\mathcal{X} + \lambda I)^{-1}\mathcal{X}^T\mathcal{Y}$$

Linear regression workbook

This workbook will walk you through a linear regression example. It will provide familiarity with Jupyter Notebook and Python. Please print (to pdf) a completed version of this workbook for submission with HW #1.

ECE C147/C247, Winter Quarter 2025, Prof. J.C. Kao, TAs: B. Qu, K. Pang, S. Dong, S. Rajesh, T. Monsoor, X. Yan

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt

#allows matlab plots to be generated in line
%matplotlib inline
```

Data generation

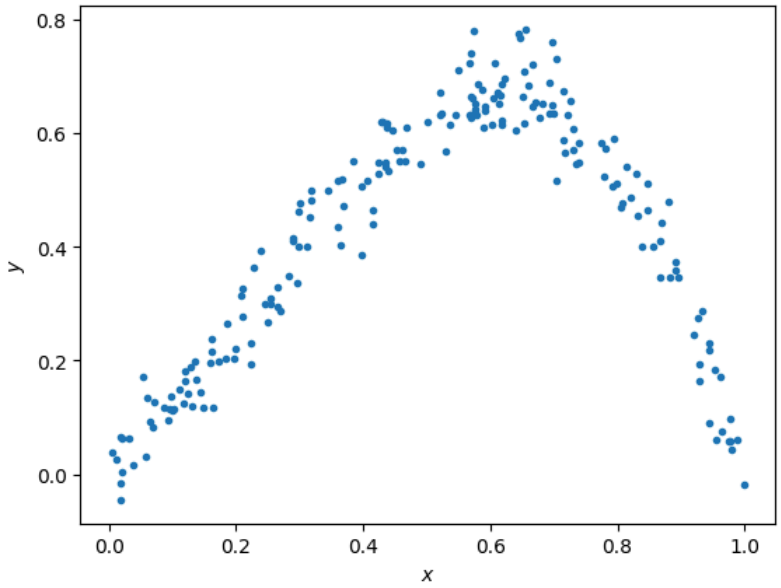
For any example, we first have to generate some appropriate data to use. The following cell generates data according to the model:
$y = x + 2x^2 - 3x^3 + \epsilon$

In [2]:
```
np.random.seed(0)  # Sets the random seed.
num_train = 200     # Number of training data points

# Generate the training data
x = np.random.uniform(low=0, high=1, size=(num_train,))
y = x + 2*x**2 - 3*x**3 + np.random.normal(loc=0, scale=0.05, size=(num_train,))
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
```

Out[2]:
```
Text(0, 0.5, '$y$')
```

```
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "orientation" which is no longer supported as of 3.3 and will become an
error two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "dpi" which is no longer supported as of 3.3 and will become an error t
wo minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "facecolor" which is no longer supported as of 3.3 and will become an e
rror two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "edgecolor" which is no longer supported as of 3.3 and will become an e
rror two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "bbox_inches_restore" which is no longer supported as of 3.3 and will b
ecome an error two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
```



QUESTIONS:

Write your answers in the markdown cell below this one:

(1) What is the generating distribution of x?

(2) What is the distribution of the additive noise ϵ?

ANSWERS:

(1) Values of x are sampled from a uniform distribution between 0 and 1.

(2) The additive noise vlaues are derived from a normal distribution with mean 0 and standard deviation 0.05.

Fitting data to the model (5 points)

Here, we'll do linear regression to fit the parameters of a model $y = ax + b$.

In [3]:
```
#xhat = (x, 1)
xhat = np.vstack((x, np.ones_like(x)))

# Projecting y onto the column space of xhat
theta = np.linalg.inv(xhat.dot(xhat.T)).dot(xhat).dot(y)
print(theta)

```

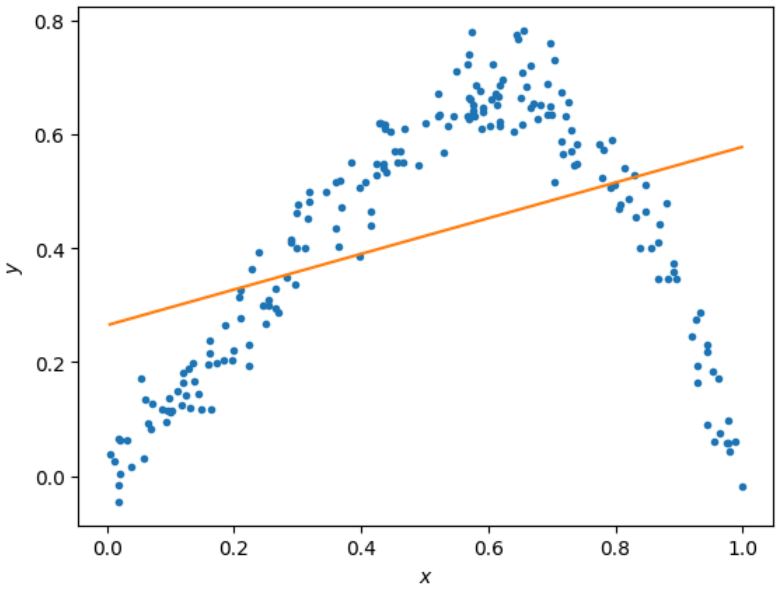
```
[0.31325736 0.26474646]
```

In [4]:
```
# Plot the data and your model fit.
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')

# Plot the regression line
xs = np.linspace(min(x), max(x),50)
xs = np.vstack((xs, np.ones_like(xs)))
plt.plot(xs[0,:], theta.dot(xs))
```

Out[4]:
```
[<matplotlib.lines.Line2D at 0x107703e90>]
```

```
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "orientation" which is no longer supported as of 3.3 and will become an
error two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "dpi" which is no longer supported as of 3.3 and will become an error t
wo minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "facecolor" which is no longer supported as of 3.3 and will become an e
rror two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "edgecolor" which is no longer supported as of 3.3 and will become an e
rror two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "bbox_inches_restore" which is no longer supported as of 3.3 and will b
ecome an error two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
```



QUESTIONS

(1) Does the linear model under- or overfit the data?

(2) How to change the model to improve the fitting?

ANSWERS

(1) The linear model underfits the data. Looking at the generated data and the true model ($y = x + 2x^2 - 3x^3 + \varepsilon$), we can see the relationship is cubic, but we're trying to fit it with just a straight line. The linear model is too simple to capture the underlying nonlinear relationship.

(2) We should use a higher-order polynomial model that can capture the nonlinear relationship. Since the true relationship is cubic, at minimum a third-order polynomial model would be appropriate. This would allow the model to capture both the linear and nonlinear components of the relationship.

Fitting data to the model (5 points)

Here, we'll now do regression to polynomial models of orders 1 to 5. Note, the order 1 model is the linear model you prior fit.

In [5]:
```
1
2  N = 5
3  xhats = []
4  thetas = []
5
6  # For each polynomial order 1 to 5
7  for i in range(N):
8      # Create feature matrix for current polynomial order
9      if i == 0:
10         # First order: [x, 1]
11         xhat = np.vstack((x, np.ones_like(x)))
12     else:
13         # Higher orders: [x^n, x^(n-1), ..., x, 1]
14         xhat = np.vstack((x**(i+1), xhat))
15
16     xhats.append(xhat)
17
18     # Model Coefficients
19     theta = np.linalg.inv(xhat.dot(xhat.T)).dot(xhat).dot(y)
20     thetas.append(theta)
21
22 pass
23
24 print(thetas)
25
```

```
[array([0.31325736, 0.26474646]), array([-2.54077983,  2.81808862, -0.14765329]), array([-3.19269866,  2.2987
0971,  0.8529672 ,  0.01631781]), array([ 0.23466729, -3.65889518,  2.59581785,  0.78750808,  0.01958868]), a
rray([ 0.87387653, -1.94094667, -1.73245937,  1.87616209,  0.88956527,
        0.01619333])]
```

In [12]:

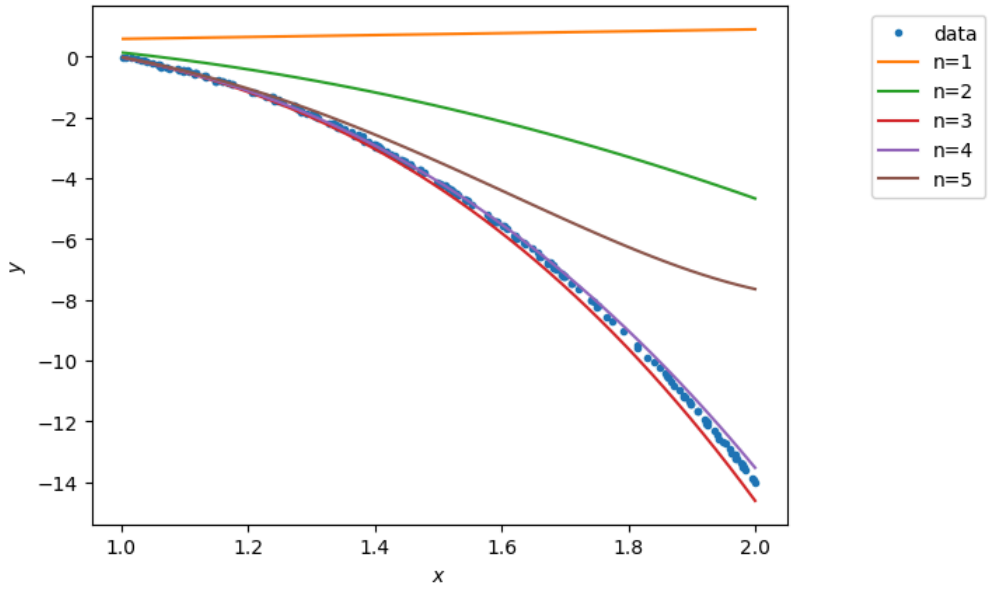
```
# Plot the data
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')

# Plot the regression lines
plot_xs = []
for i in np.arange(N):
    if i == 0:
        plot_x = np.vstack((np.linspace(min(x), max(x),50), np.ones(50)))
    else:
        plot_x = np.vstack((plot_x[-2]**(i+1), plot_x))
    plot_xs.append(plot_x)

for i in np.arange(N):
    ax.plot(plot_xs[i][-2,:], thetas[i].dot(plot_xs[i]))

labels = ['data']
[labels.append('n={}'.format(i+1)) for i in np.arange(N)]
bbox_to_anchor=(1.3, 1)
lgd = ax.legend(labels, bbox_to_anchor=bbox_to_anchor)
```

```
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "orientation" which is no longer supported as of 3.3 and will become an
error two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "dpi" which is no longer supported as of 3.3 and will become an error t
wo minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "facecolor" which is no longer supported as of 3.3 and will become an e
rror two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "edgecolor" which is no longer supported as of 3.3 and will become an e
rror two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "bbox_inches_restore" which is no longer supported as of 3.3 and will b
ecome an error two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
```



Calculating the training error (5 points)

Here, we'll now calculate the training error of polynomial models of orders 1 to 5.

In [13]:

```
training_errors = []


for i in np.arange(N):

    y_pred = thetas[i].dot(xhats[i])

    mse = np.mean((y - y_pred) ** 2)

    training_errors.append(mse)

pass

print ('Training errors are: \n', training_errors)
```

```
Training errors are:
 [54.246317716012236, 18.911159217529747, 0.08693862570546057, 0.03042966869910436, 5.954356339341897]
```

QUESTIONS

(1) What polynomial has the best training error?

(2) Why is this expected?

ANSWERS

(1) The fifth order polynomial has the best traning error.

(2) Using a higher-order polynomial helps the model capture more relationships compred to lower order ones. Therefore, we get a low training error for the fifth order polynomial.

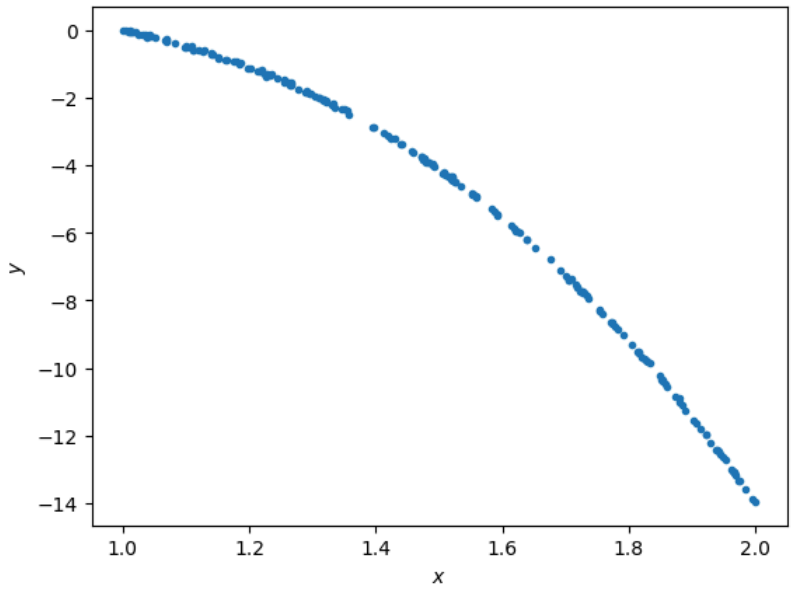
Generating new samples and testing error (5 points)

Here, we'll now generate new samples and calculate testing error of polynomial models of orders 1 to 5.

In [14]:
```
x = np.random.uniform(low=1, high=2, size=(num_train,))
y = x + 2*x**2 - 3*x**3 + np.random.normal(loc=0, scale=0.03, size=(num_train,))
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
```

Out[14]:
```
Text(0, 0.5, '$y$')
```

```
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "orientation" which is no longer supported as of 3.3 and will become an
error two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "dpi" which is no longer supported as of 3.3 and will become an error t
wo minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "facecolor" which is no longer supported as of 3.3 and will become an e
rror two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "edgecolor" which is no longer supported as of 3.3 and will become an e
rror two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "bbox_inches_restore" which is no longer supported as of 3.3 and will b
ecome an error two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
```



In [15]:
```
xhats = []
for i in np.arange(N):
    if i == 0:
        xhat = np.vstack((x, np.ones_like(x)))
        plot_x = np.vstack((np.linspace(min(x), max(x),50), np.ones(50)))
    else:
        xhat = np.vstack((x**(i+1), xhat))
        plot_x = np.vstack((plot_x[-2]**(i+1), plot_x))

    xhats.append(xhat)
```

In [16]:

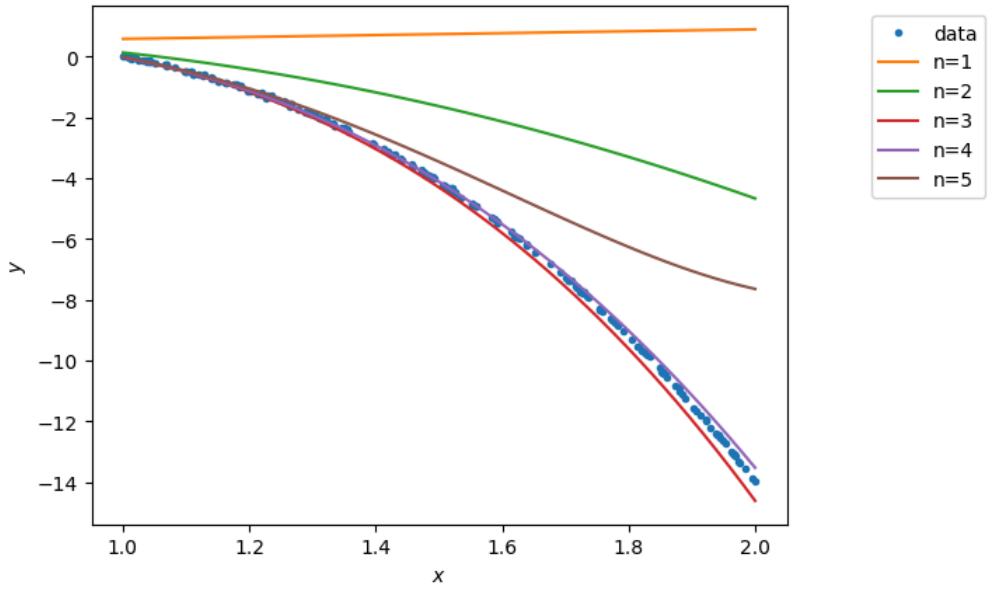
```
# Plot the data
f = plt.figure()
ax = f.gca()
ax.plot(x, y, '.')
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')

# Plot the regression lines
plot_xs = []
for i in np.arange(N):
    if i == 0:
        plot_x = np.vstack((np.linspace(min(x), max(x),50), np.ones(50)))
    else:
        plot_x = np.vstack((plot_x[-2]**(i+1), plot_x))
    plot_xs.append(plot_x)

for i in np.arange(N):
    ax.plot(plot_xs[i][-2,:], thetas[i].dot(plot_xs[i]))

labels = ['data']
[labels.append('n={}'.format(i+1)) for i in np.arange(N)]
bbox_to_anchor=(1.3, 1)
lgd = ax.legend(labels, bbox_to_anchor=bbox_to_anchor)
```

```
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "orientation" which is no longer supported as of 3.3 and will become an
error two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "dpi" which is no longer supported as of 3.3 and will become an error t
wo minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "facecolor" which is no longer supported as of 3.3 and will become an e
rror two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "edgecolor" which is no longer supported as of 3.3 and will become an e
rror two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
/opt/homebrew/lib/python3.11/site-packages/IPython/core/pylabtools.py:152: MatplotlibDeprecationWarning: save
fig() got unexpected keyword argument "bbox_inches_restore" which is no longer supported as of 3.3 and will b
ecome an error two minor releases later
  fig.canvas.print_figure(bytes_io, **kw)
```



In [17]:

```
testing_errors = []


for i in np.arange(N):

    y_pred = thetas[i].dot(xhats[i])

    mse = np.mean((y - y_pred) ** 2)

    testing_errors.append(mse)

pass


print ('Testing errors are: \n', testing_errors)
```

```
Testing errors are:
 [51.595934377138384, 17.88767990466363, 0.08568137176745275, 0.02624925686033019, 5.438380112067071]
```

QUESTIONS

(1) What polynomial has the best testing error?

(2) Why polynomial models of orders 5 does not generalize well?

ANSWERS

(1) Lower-order polynomials (e.g., orders 1 or 2) may underfit, leading to high training and testing errors. Higher-order polynomials (e.g., order 5) tend to overfit, capturing noise in the training data, which results in poor testing performance. The fourth order polynomial has the best testing error.

(2) Polynomial models of order 5 may not generalize well because of overfitting, which occurs when the model becomes too flexible and fits the noise in the training data rather than the underlying trend.