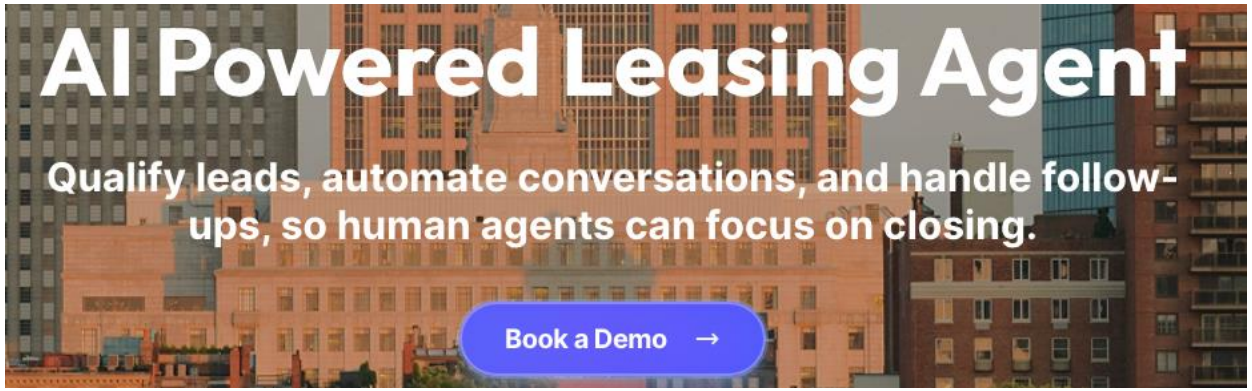
Introduction to Artificial Intelligence

Lecture 2: Supervised learning I

September 8, 2025

Recap

- Rational agents: A central concept to our approach to AI



https://www.houseflyai.com/

Khoury News

There's a new leasing agent in the crowded Boston rental market ... only this one's an AI

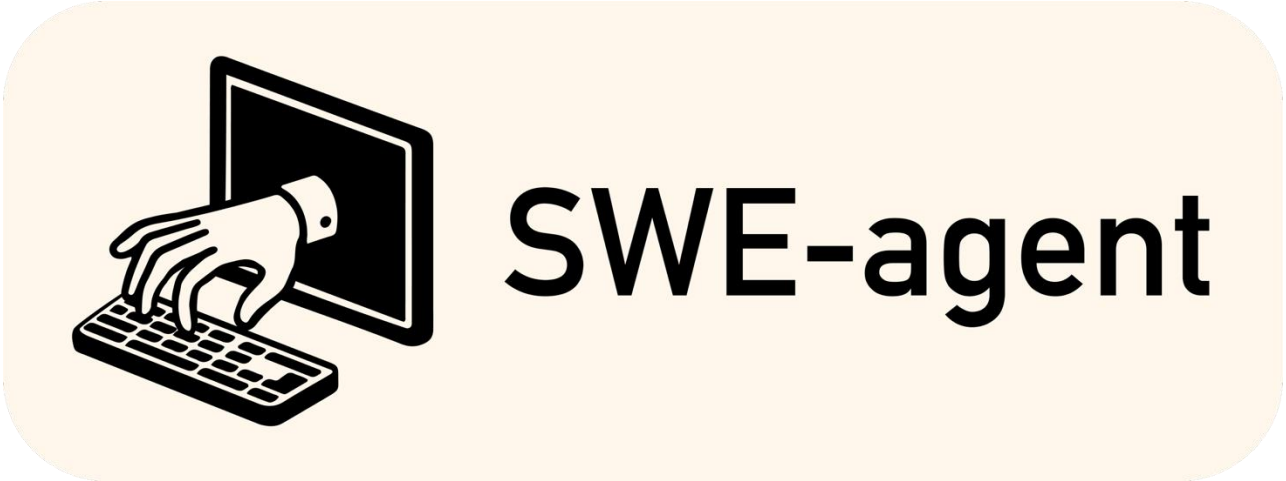
Leasing agents spend much of their time asking and answering the same renter questions over and over. So a team of Northeastern students built an AI agent called HouseFly to do it for them.

August 13, 2025
by Elizabeth S. Leaver

Share article

Rational agents

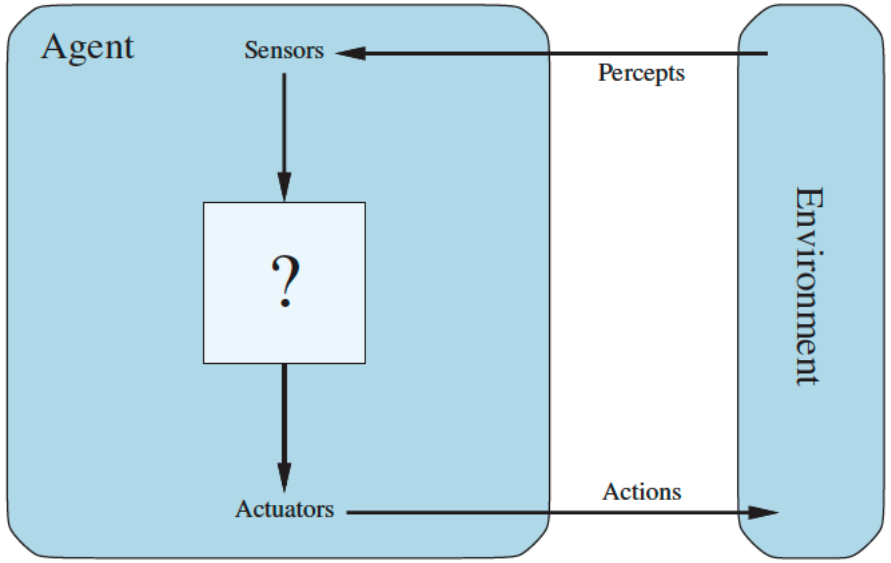
- https://github.com/SWE-agent/SWE-agent: automatically fix GitHub issues using your LM of choice

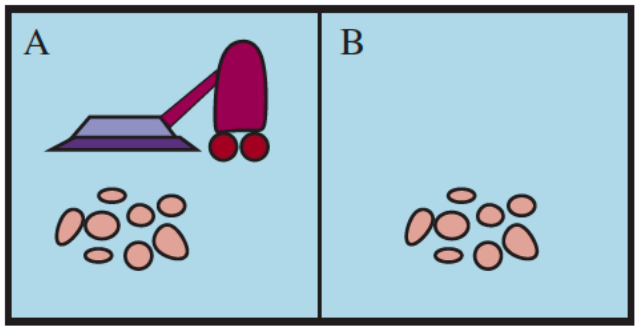


- https://swe-agent.com/latest/installation/codespaces/

Agents and environments

- Agents interact with environments through sensors and actuators





Rationality

- Four necessary parts:
  - Performance measure: defining the criterion of success
  - Prior knowledge of the environment
  - Actions that the agent can perform
  - Agent's percept sequence to date

- **Rational agent:** For each possible percept sequence, a rational agent selects an action that maximizes its performance measure in expectation, given percept sequence and prior knowledge

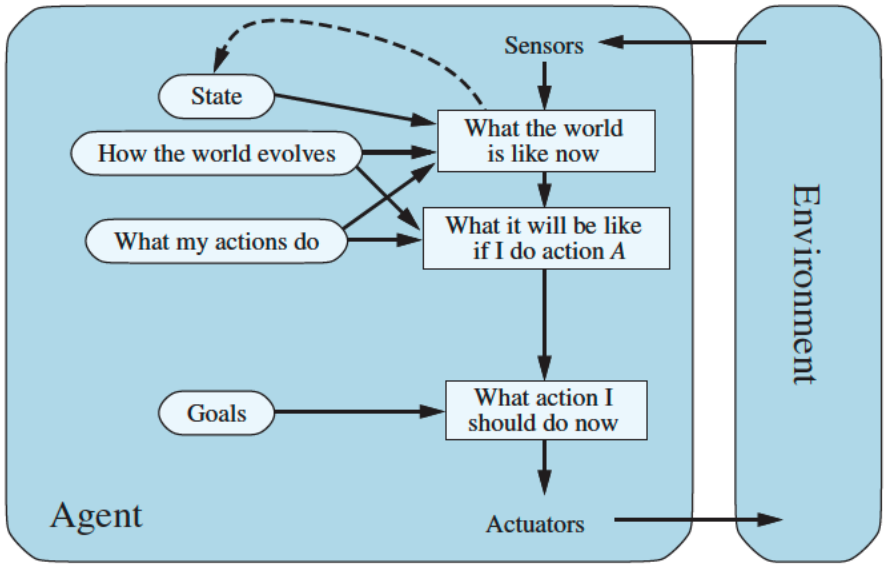
Properties of task environments

- **Fully observation vs. partially observable:** esp. relevant when we study Reinforcement Learning and sequential decision-making
- **Single-agent vs. multi-agent:** cooperative vs. competitive?
- **Episodic vs. sequential:** the next episode does not depend on the actions taken in previous episodes (assembly lines); otherwise, current decision could affect future decisions (chess, taxi deriving)
- **Static vs. dynamic:** Crossword puzzles are static, while taxi driving is clearly dynamic

The structure of agents

- **Model-based agents** use a transition model and a sensor model to keep track of state of the world

- **Goal-based agents:** Searching and planning



Utility-based agents

- For example, given a destination, many action sequences get us to achieve the goal, but some are quicker, safer, or cheaper (e.g., no toll).
  - Economists and computer scientists use the term "utility" to refer to the "happiness" to achieve the goal

- **Model-free agent:** learn what action is best in a particular situation without learning exactly how that action changes the environment

Learning agents (finish recap)

- Four conceptual components
  - **Learning element:** making improvements
  - **Performance element:** selecting external actions
  - **Critic:** gives feedback on how the agent is doing and determines how performance should be modified to do better
  - **Problem generator:** suggesting actions that will lead to new and informative experiences

Machine learning overview

Supervised learning

Neural networks and deep learning

Natural language processing

Lecture plan

- **Supervised learning**
  - **Simple linear regression**

Matrices and vectors

- Matrices: A rectangular array of numbers

$$A = \begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \dots & \dots & \dots \\ a_{m,1} & \dots & a_{m,n} \end{bmatrix}$$

- Vectors: An array consisting of a single column

$$a = \begin{bmatrix} a_1 \\ \dots \\ a_n \end{bmatrix}$$

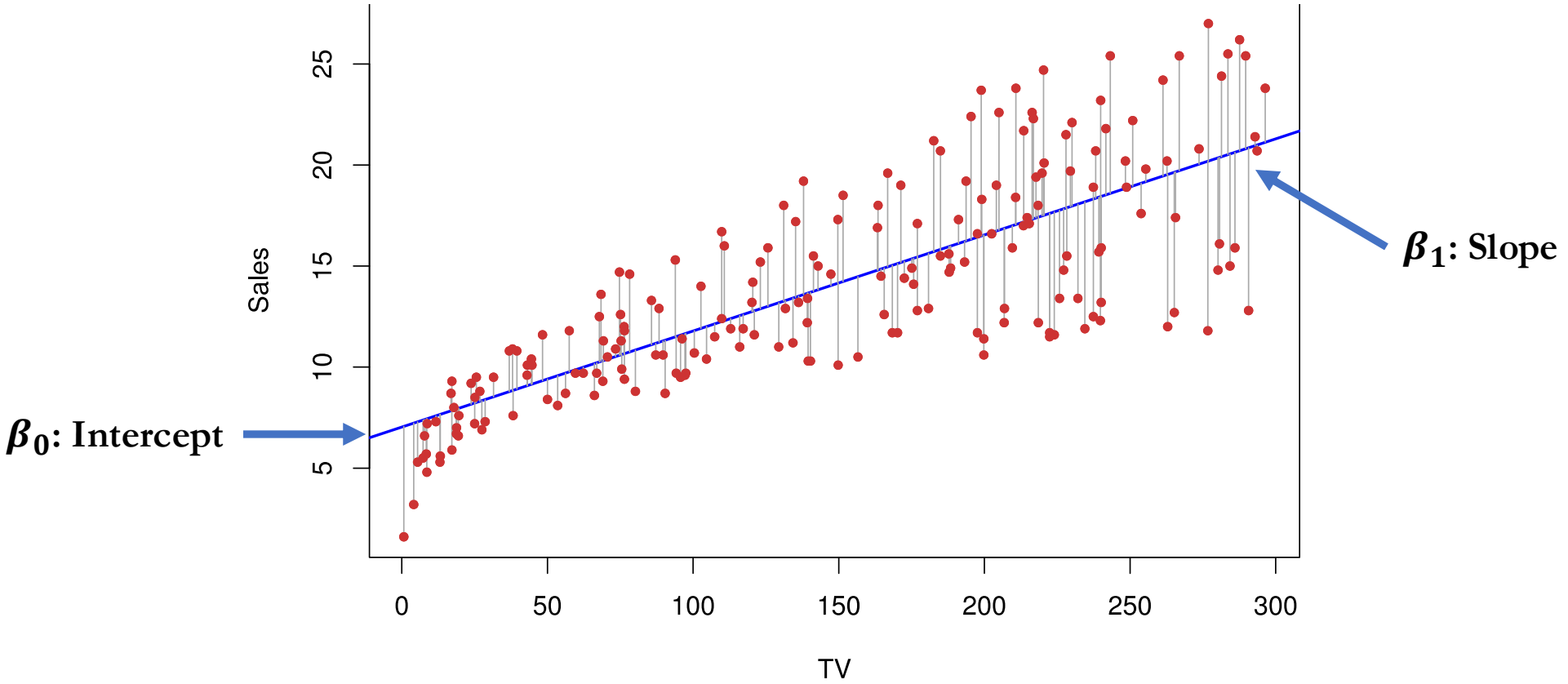
Simple linear regression

- Let us consider the simplest case of a linear regression problem: We are giving a list of one-dimensional features and their corresponding labels. We want to build a regression model to achieve that
  - Examples: Predicting rental costs, advertising, marketing, etc

- Input: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ (assume we have already done the training/test split)
- Output: a linear model parameterized by β_0 and β_1

Examples of β_0 and β_1

- Fitting a regression model mapping TA ad spending to Sales amount



Setting up the linear model

- Recall the input to the problem: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ (this is the training data)
- Let us set up a predicted label for each sample:

$$\hat{y}_i = \beta_0 + x_i\beta_1, \text{ for } i = 1, 2, \ldots, n$$

- Next, let us set up the mean squared error metric:

$$\hat{L}(\beta) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(\beta_0 + x_i\beta_1 - y_i)^2$$

Where $\beta = \begin{bmatrix}\beta_0 \\ \beta_1\end{bmatrix}$

Solving for β_0 and β_1

- Recall that $\hat{L}(\beta) = \frac{1}{n}\sum_{i=1}^{n}(\beta_0 + x_i\beta_1 - y_i)^2$; we would like to minimize the MSE metric
- We're going to set the derivatives of $\hat{L}$ with respect to β_0, β_1 as zero

$$\frac{\partial \hat{L}(\beta)}{\partial \beta_0} = \frac{2}{n}\sum_{i=1}^{n}(\beta_0 + x_i\beta_1 - y_i) = 0$$

$$\frac{\partial \hat{L}(\beta)}{\partial \beta_1} = \frac{2}{n}\sum_{i=1}^{n} x_i(\beta_0 + x_i\beta_1 - y_i) = 0$$

Solving for β_0 and β_1

- We can re-arrange the derivatives to be zero as follows

$$\beta_0 + \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)\beta_1 = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)\beta_0 + \left(\frac{1}{n}\sum_{i=1}^{n} x_i^2\right)\beta_1 = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Final solution

- This is a two-by-two linear system, which can be solved explicitly

$$\beta_0 = \frac{\left(\frac{1}{n}\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\sum_{i=1}^{n} x_i\right) \cdot \left(\frac{1}{n}\sum_{i=1}^{n} y_i\right)}{\frac{1}{n}\sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2}$$

$$\beta_1 = \frac{\left(1 - \frac{1}{n}\sum_{i=1}^{n} x_i\right) \cdot \left(\frac{1}{n}\sum_{i=1}^{n} y_i\right)}{\frac{1}{n}\sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2}$$

Takeaways

- In order to have a valid solution, we need that

$$\frac{1}{n}\sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2 \neq 0$$

This is true as long as the x_i's are not all the same!

- We can use the explicit expressions of β_0, β_1 to derive confidence intervals
  - This is a bit advanced, but the high-level idea is we assume the x_i's are Gaussian, from which we could derive the distribution of β_0, β_1

Summary of simple linear regression

- After solving $\hat{\beta}_0, \hat{\beta}_1$, we could use the estimated coefficients to make predictions on unseen regions

$$\hat{y} = \hat{\beta}_1 x + \hat{\beta}_0$$

Evaluation metrics

- **R^2 statistic** measures the proportion of variance explained

$$\text{RSS (Residual sum of squares)} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\text{TSS (Total sum of squares)} = \sum_{i=1}^{n}(y_i - \bar{y})^2, \text{ where } \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

R^2 always takes on a value between 0 and 1

Evaluation metrics

- **Correlation** between two random variables is another measure of linear relationship between X and Y

$$Cor(X,Y) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\cdot\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}, \text{ where } \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \text{ and } \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

- **Example:** in the linear regression example, we may take the uniform distribution of $y_1, y_2, \dots, y_n$ as the 1st random variable, and the uniform distribution of $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n$ as the 2nd random variable

- **Example**: If X and Y are independent, then $Cor(X,Y) = 0$
  - Recall $E[X \cdot Y] = E[X] \cdot E[Y]$

Lecture plan

- **Supervised learning**
  - Simple linear regression
  - **Multiple linear regression**

Multiple linear regression

- Now represent nonlinear relationships
  - Transformations of quantitative inputs: log, square-root, or square
  - Basis expansion: $x_2 = x_1^2, x_3 = x_1^3$
  - Numeric coding of qualitative inputs
  - Interactions between inputs: $x_3 = x_1 \cdot x_2$

Setting up the problem

- We're giving a training set $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. Let us assume that each x has p features in total

- We want to learn a linear regression model to map x's to y's: the linear model has $p + 1$ variables in total, $\beta_0, \beta_1, \ldots, \beta_p$

Matrix notations

- Feature matrix (note that we have added a column of ones):

$$X = \begin{bmatrix} 1 & x_{1,1}, \dots, x_{1,p} \\ 1 & x_{2,1}, \dots, x_{2,p} \\ & \vdots \quad \vdots \\ 1 & x_{n,1}, \dots, x_{n,p} \end{bmatrix}$$

- Label vector:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Exercise: what is the dimension of $\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\beta}$, respectively?

- Predicted label:

$$\widehat{y_i} = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_p x_{i,p}, \text{ for } i = 1,2, \dots, n$$

Matrix notations

- Let us stack the variables we need to estimate together

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \dots \\ \beta_p \end{bmatrix}$$

- Using matrix multiplication rule, we shall verify that

$$\hat{y} = X\beta$$

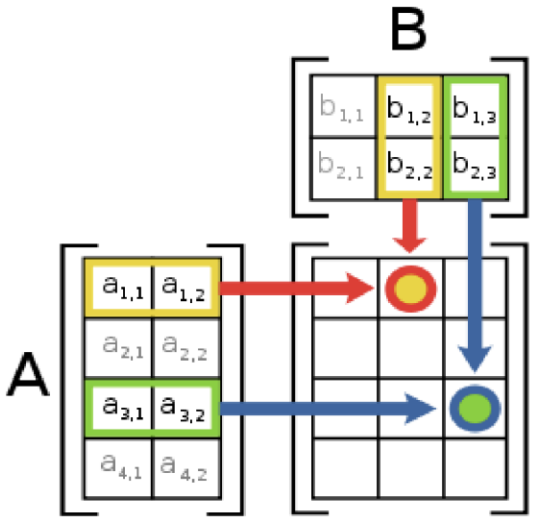
Where $\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}$

Matrix multiplication

- Let $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{n \times p}$, their product $C = AB \in \mathbb{R}^{m \times p}$
- Number of columns of A must be equal to the number of rows of B
- Compute the product $C = AB$ using

$$C_{i,j} = \sum_{k=1}^{n} A_{i,k} B_{k,j}$$

- An illustration



- Exercise: multiply $A = [1,2]$ with $B = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$

Start with the one-dimensional case

- **Fitting a line** with coefficient $\beta_1 \in \mathbb{R}$ and intercept $\beta_0 \in \mathbb{R}$

$$\hat{y}_i = \beta_0 + \beta_1 x_i$$

- **Recall matrix notation:** $\hat{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$

- **Exercise:** verify that $\hat{y} = X\beta$

Move to the multi-dimensional case

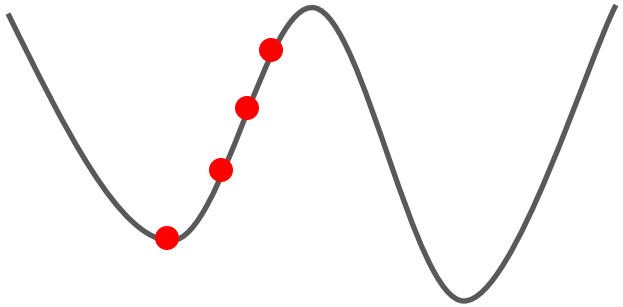
- **Fitting a hyperplane** with coefficients $\beta_1, \beta_2, \ldots, \beta_p$ and intercept β_0
- **Exercise:** First verify that the predicted labels are $\hat{y} = X\beta$
- Recall that MSE metric:

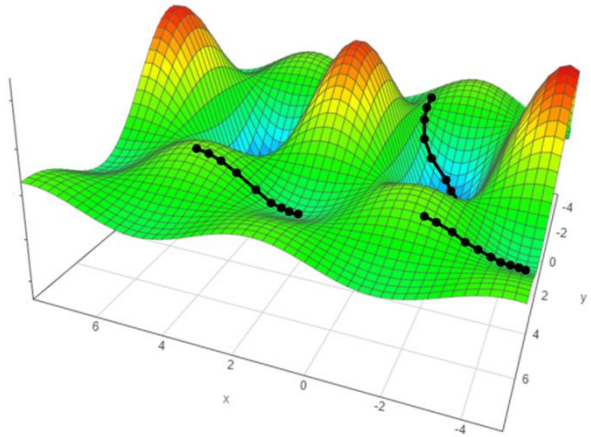
$$\hat{L}(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i^{\top}\boldsymbol{\beta} - \boldsymbol{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i)^2 = \frac{1}{n}(\boldsymbol{y} - \boldsymbol{X\beta})^{\boldsymbol{T}}(\boldsymbol{y} - \boldsymbol{X\beta})$$

- We'll set the derivatives to zero: $\frac{\partial \hat{L}(\beta)}{\partial \beta_0}, \frac{\partial \hat{L}(\beta)}{\partial \beta_1}, \ldots, \frac{\partial \hat{L}(\beta)}{\partial \beta_p}$

Gradient descent

- To minimize the loss $\hat{L}(\beta)$, we can use optimization algorithms like gradient descent





- The gradient descent algorithm
  - Initialize β_0
  - Let $\nabla\hat{L}(\beta_t)$ be the gradient of the training loss at β_t
  - Let η be a learning rate parameter

$$w_t \leftarrow w_t - \eta \cdot \nabla\hat{L}(f_{w_t})$$

The gradient

- **Definition:** let $f: \mathbb{R}^d \rightarrow \mathbb{R}$ be a multi-dimensional function, which takes a vector of d variables X as input, and outputs a real value $y = f(X)$

- Suppose f is differentiable at every coordinate, then, the gradient of f, denoted as ∇f, is defined as

$$\nabla f(X) = \begin{bmatrix} \frac{\partial f(X)}{\partial X_1}, \\ \frac{\partial f(X)}{\partial X_2}, \\ \cdots, \\ \frac{\partial f(X)}{\partial X_d} \end{bmatrix}$$

Stochastic gradient descent

- **Motivation:** If the gradient on the first half is almost identi l half
  - Mini-batch stochas