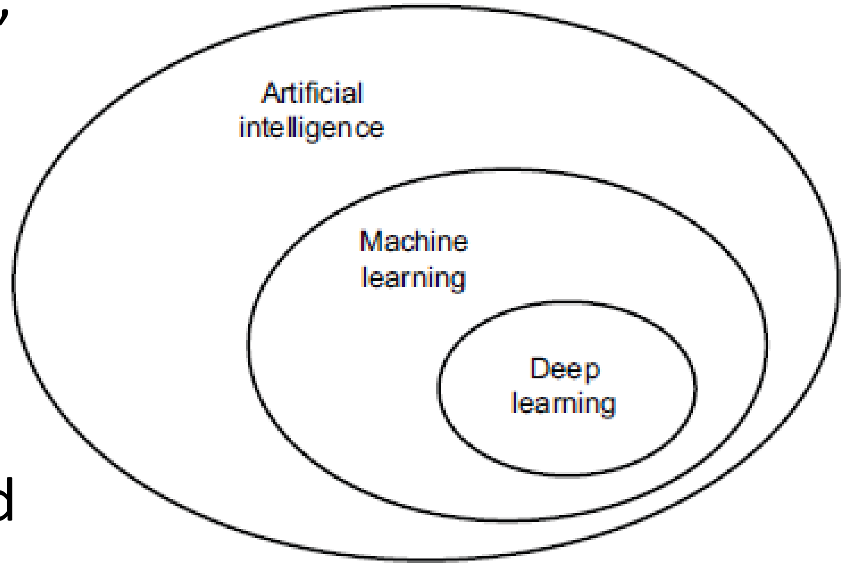
Machine Learning
Live Session #1

Machine Learning

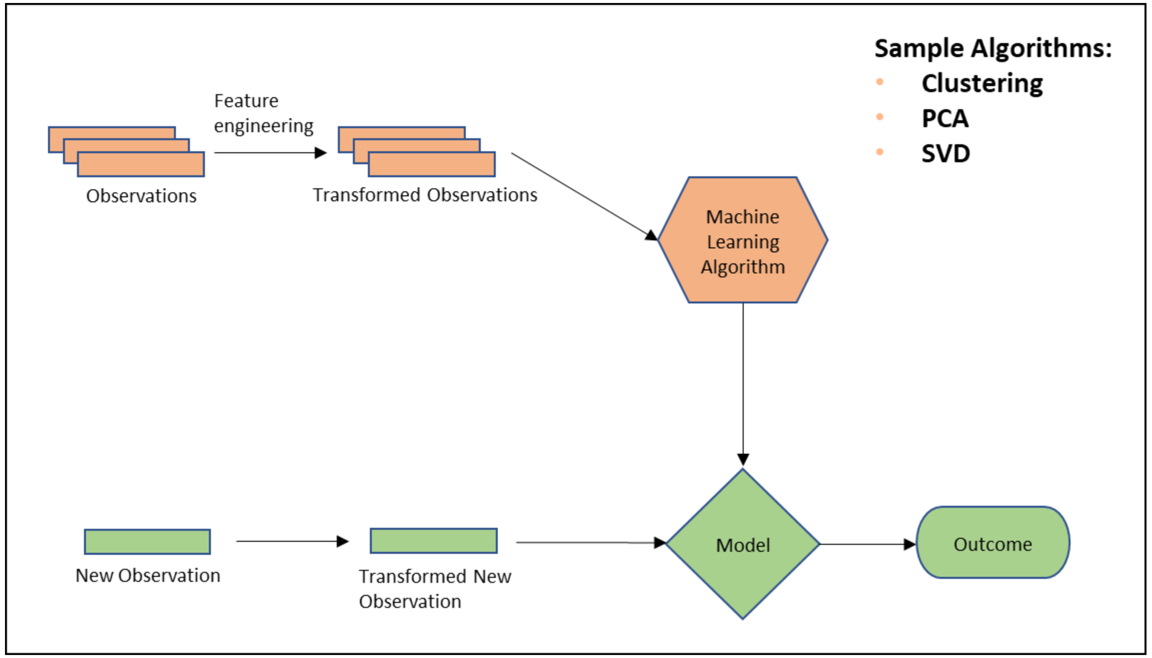
- **Artificial intelligence** is the study of getting machines to perform human tasks
  - E.g., recommender systems in Netflix, a self-driving Tesla, etc.
- **Machine learning** is the study of getting machines to learn from data
  - I.e., it is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns, and make decisions with minimal human intervention.
  - Two main types of learning: unsupervised and supervised
- **Deep learning** is essentially machine learning with neural networks containing many layers (i.e., deep neural networks)



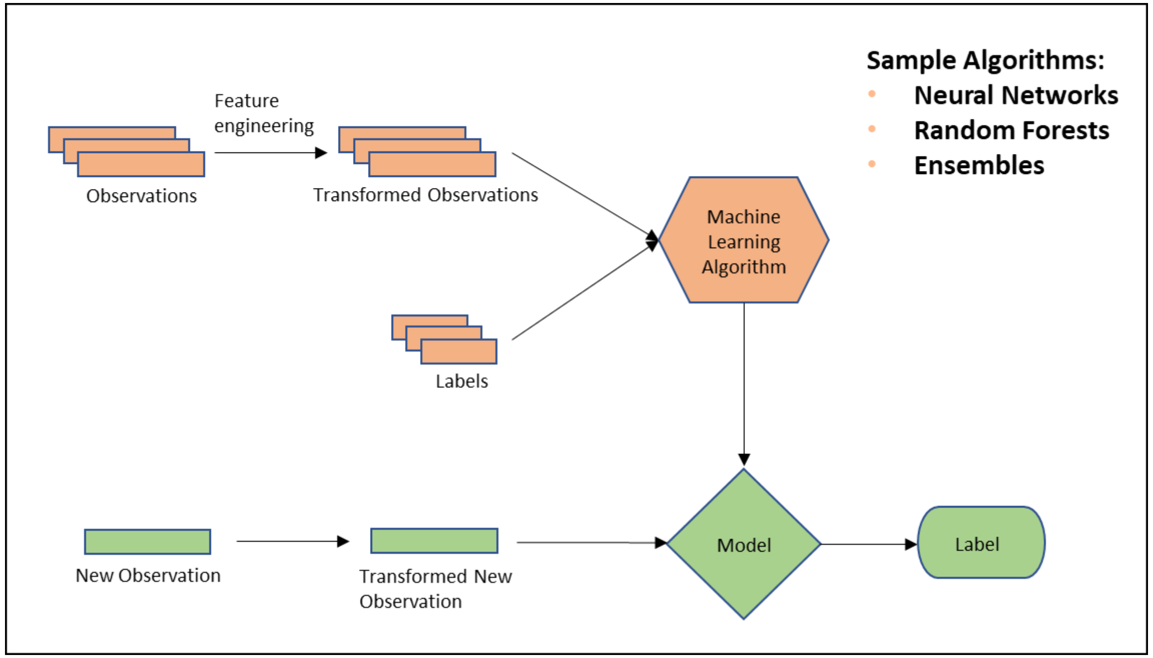
Machine Learning: What It Is and Why It Matters. SAS. https://www.sas.com/en_us/insights/analytics/machine-learning.html

Supervised vs. Unsupervised Learning

- Unsupervised learning: no labels associated with observations
  - Try to infer relationships between the observations or between the features
  - Useful for data visualization and dimension reduction
- Supervised learning: each observation is associated with a label
  - Try to infer a relationship between the features and labels
    - The label acts as a teacher that supervises the learning process
  - Use the relationship to predict label for a new observation

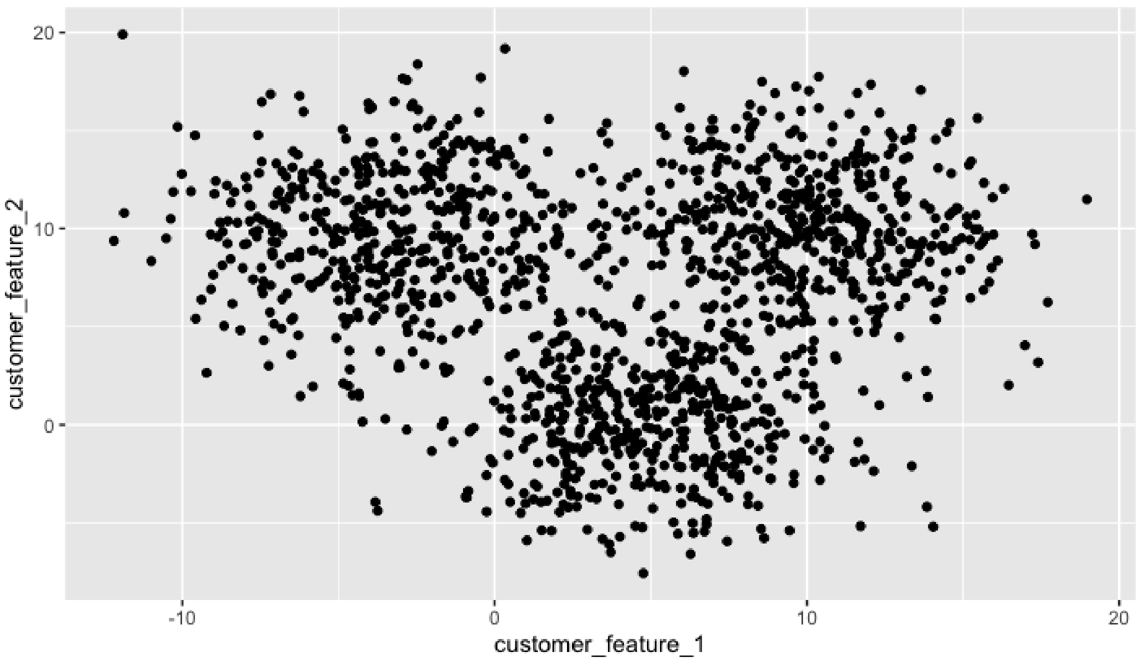


Unsupervised Learning



Supervised Learning

Unsupervised vs. Supervised Learning



Unsupervised Learning

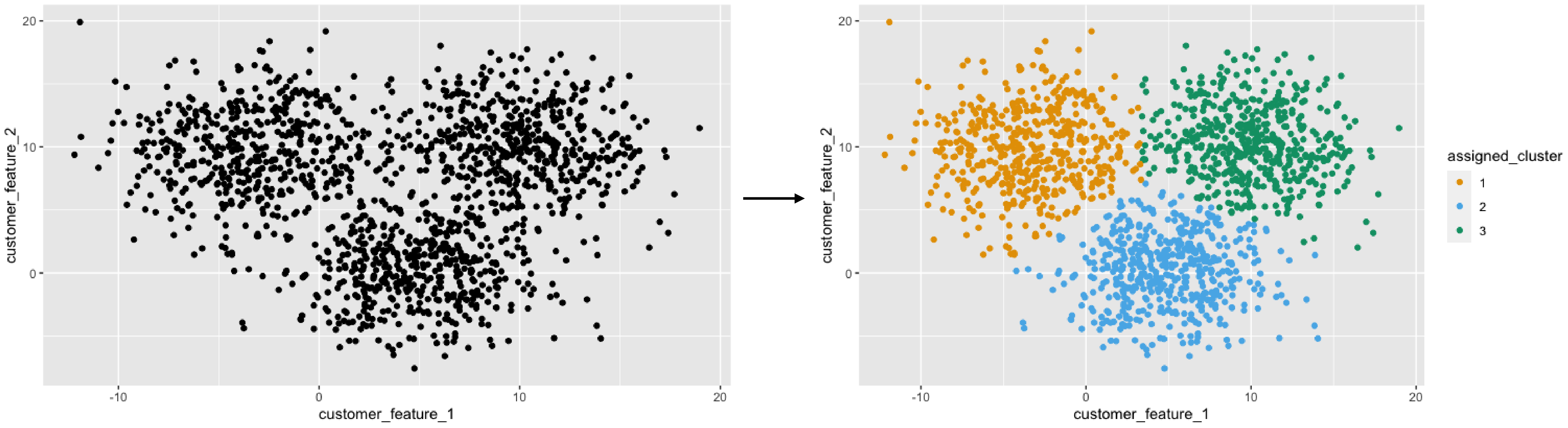
- Observations (here, customers) have no labels
- Use unsupervised learning to explore and learn about customers
  - E.g., do sub-groups of customers exist, with each sub-group exhibiting similar characteristics?
  - This is called customer segmentation

Supervised Learning

- Each observation (here, customer) is associated with a label
  - E.g., whether the customer left or stayed
- Use supervised learning to predict the label for new customers

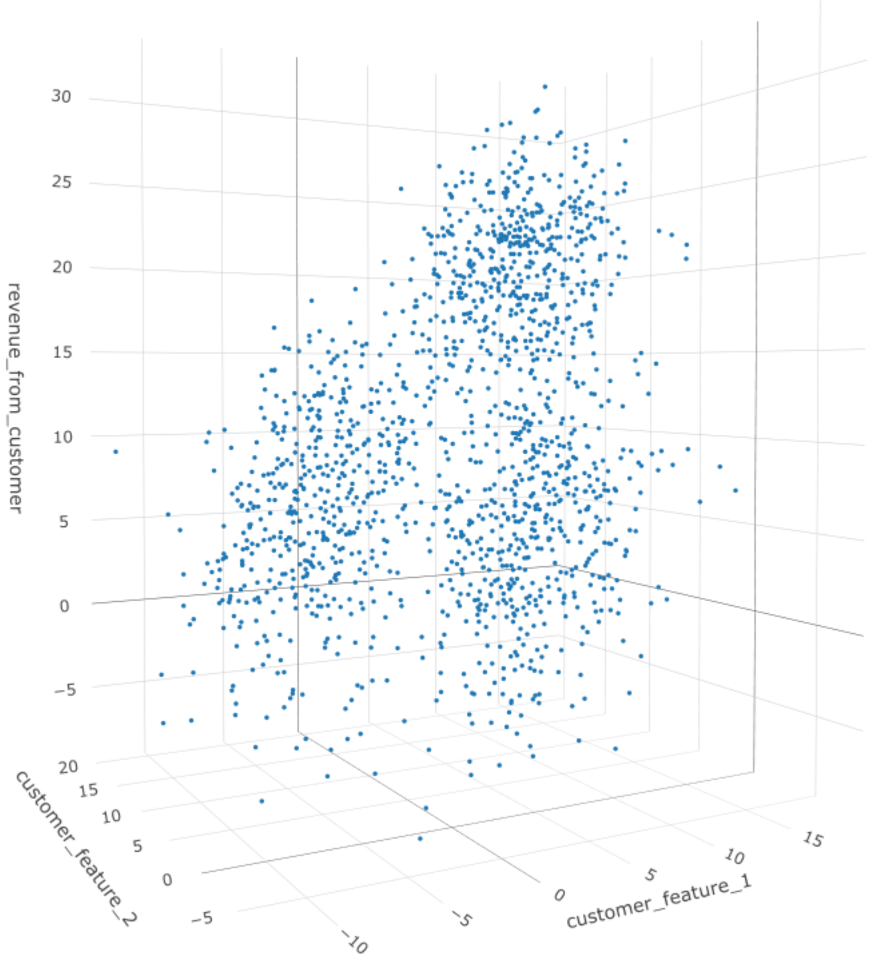
Unsupervised Learning Example

- k-means clustering
  - Find k clusters in the dataset
  - Does not use any labels, only distance of observations from each other



Supervised Learning

- Two types of supervised learning: regression and classification



Regression

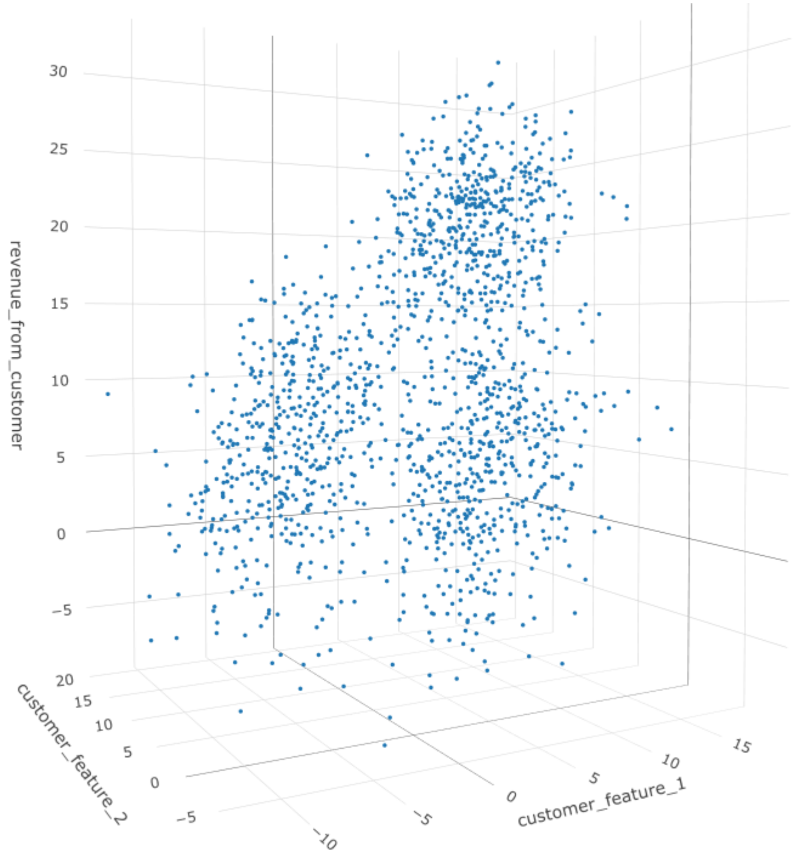
Classification

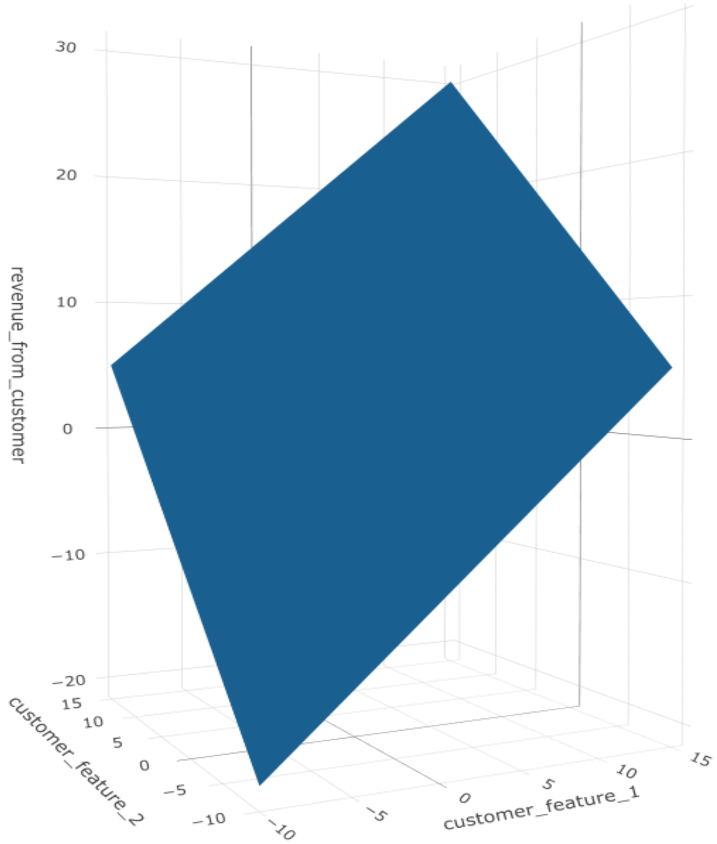
Types of Variables

- There are two main types of variables
  - **Categorical (also called nominal)**
    - Qualitative (i.e., values of the variable are not quantifiable)
    - Distinct values are called levels (e.g., medical diagnosis with levels: ‘diabetes’, ‘cancer’, ‘heart disease’, etc.)
    - If levels have intrinsic order, then the variable is called **ordinal** (e.g., a variable with levels ‘small’, ‘medium’, ‘large’)
  - **Numerical**
    - Quantitative (i.e., values of the variable are quantifiable)
    - Includes **continuous** and **discrete** variables
      - Continuous: can assume an infinite number of values within a given interval
      - Discrete: can assume only a finite number of values within a given interval

Regression Example

- The label is a numerical variable
- Want to predict the label for a new observation



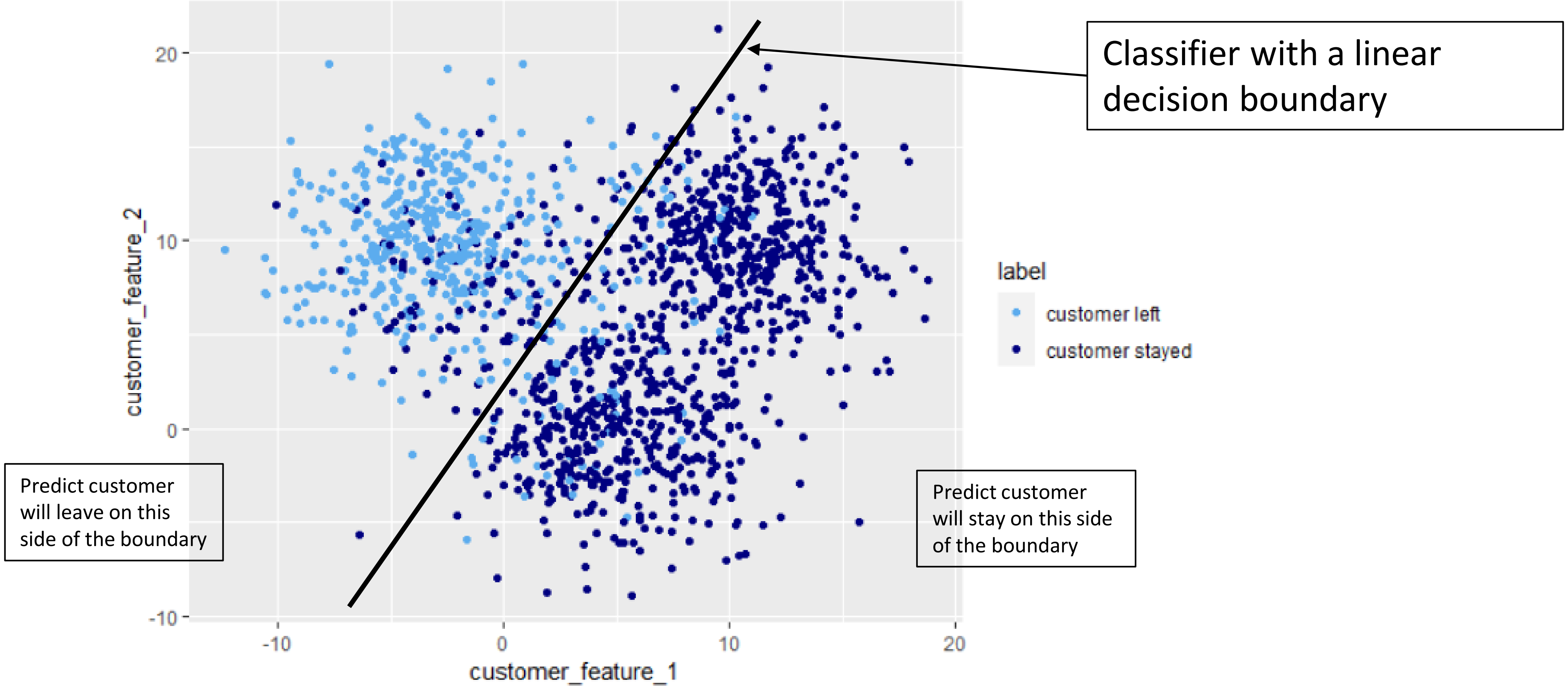


$$Predicted\ Revenue = \hat{y}(Income, Age) = w_1^* \times Income + w_2^* \times Age + w_0^*$$

Linear Regression Equation

Classification Example

- The label is a categorical variable with some number of levels called classes
- Want to predict the class for a new observation



$$\textit{Predicted Probability of Leaving} = \hat{y}(Income, Age) = \frac{1}{1 + \exp(-(w_1^* \times Income + w_2^* \times Age + w_0^*))}$$

Logistic Regression Equation

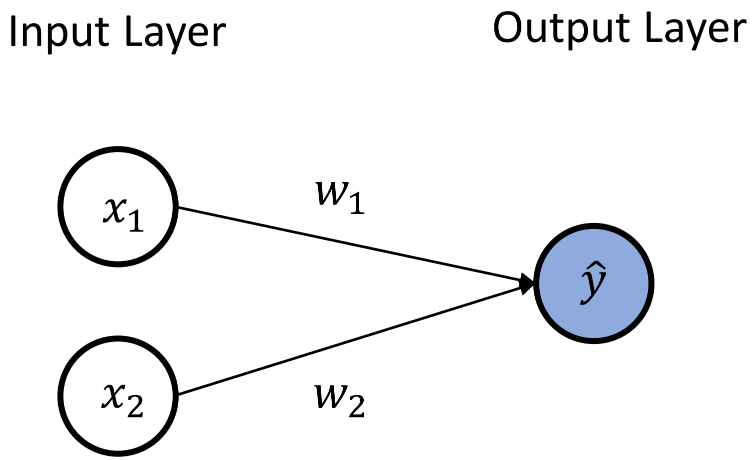
Neural Networks and Deep Learning

- Neural networks are a class of machine learning algorithms
  - Represented visually by acyclical graphs
  - Essentially, neural networks are composite functions (as we will see)

- We'll start with the simplest neural network called a perceptron and progress to more advanced (complex) architectures
  - Linear regression and logistic regression can be represented as perceptrons
  - Thus, we will see how neural networks (i.e., deep learning) generalizes traditional machine learning approaches

Neural Networks and Deep Learning

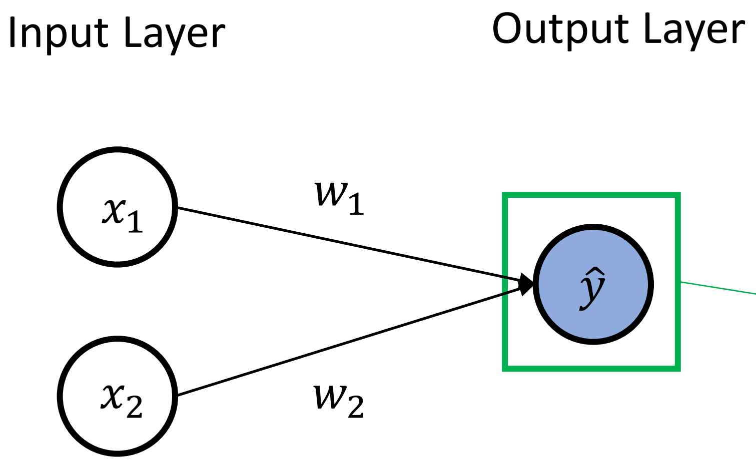
- To understand neural networks, we'll start with the simplest form called a perceptron: a neural network with only an input layer and output layer



- Let's express linear and logistic regression as perceptrons

Neural Networks and Deep Learning

- The computation in a neural network takes place in computation nodes
  - Each computation node has an input and an output
- In a perceptron, the only computation node is in the output layer
  - The output of the computation node is the output ($\hat{y}$) of our neural network model
  - The function f is called the activation function and it provides the expressive power of neural networks



Computation Node

$$Output = \hat{y} = f(Weights \times Inputs + Bias)$$
$$= f(w_1 \times x_1 + w_2 \times x_2 + w_0)$$

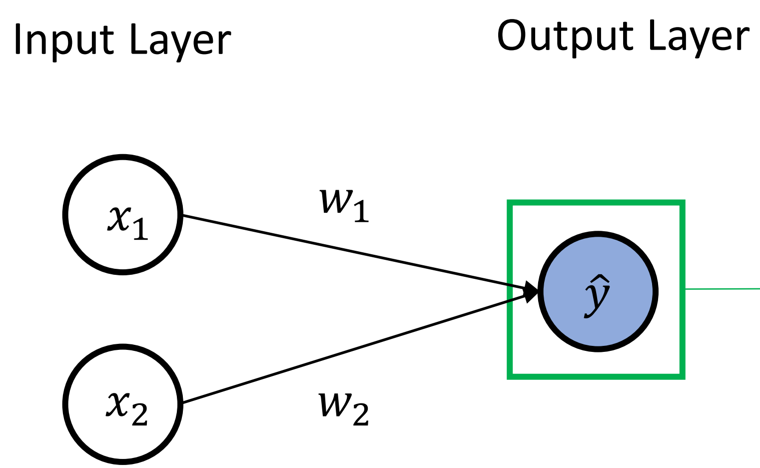
Set $f(\mathrm{x}) = \mathrm{x}$ to get linear regression

Use training data to find optimal weight values $w_1^*,\ w_2^*,\ w_0^*$

Neural Networks and Deep Learning

- The computation in a neural network takes place in computation nodes
  - Each computation node has an input and an output
- In a perceptron, the only computation node is in the output layer
  - The output of the computation node is the output ($\hat{y}$) of our neural network model
  - The function f is called the activation function and it provides the expressive power of neural networks

Income (Customer Feature 1)	Age (Customer Feature 2)
22003	45
57230	54
75137	28
31208	54
54078	23
44413	44
55237	46

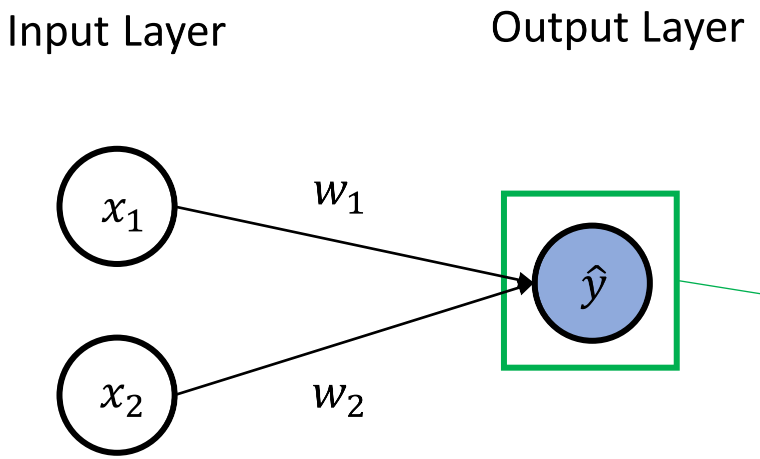


$$\hat{y} = Predicted\ Revenue = w_1 \times 22003 + w_2 \times 45 + w_0$$

Calculate predicted revenue for each observation and compare to the actual in the training data

Neural Networks and Deep Learning

- The computation in a neural network takes place in computation nodes
  - Each computation node has an input and an output
- In a perceptron, the only computation node is in the output layer
  - The output of the computation node is the output ($\hat{y}$) of our neural network model
  - The function f is called the activation function and it provides the expressive power of neural networks



Computation Node

$$Output = \hat{y} = f(Weights \times Inputs + Bias)$$
$$= f(w_1 \times x_1 + w_2 \times x_2 + w_0)$$

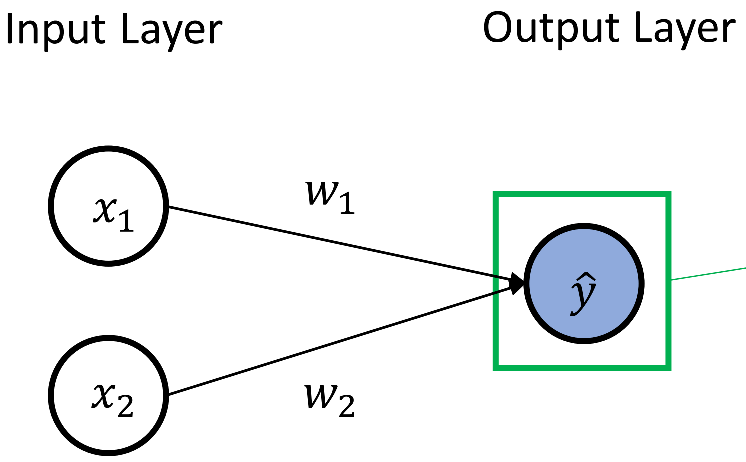
Set $f(\mathrm{x}) = \frac{1}{1+e^{-x}}$ to get logistic regression

Use training data to find optimal weight values $w_1^*,\ w_2^*,\ w_0^*$

Neural Networks and Deep Learning

- The computation in a neural network takes place in computation nodes
  - Each computation node has an input and an output
- In a perceptron, the only computation node is in the output layer
  - The output of the computation node is the output (y) of our neural network model
  - The function f is called the activation function and it provides the expressive power of neural networks

Income (Customer Feature 1)	Age (Customer Feature 2)
22003	45
57230	54
75137	28
31208	54
54078	23
44413	44
55237	46



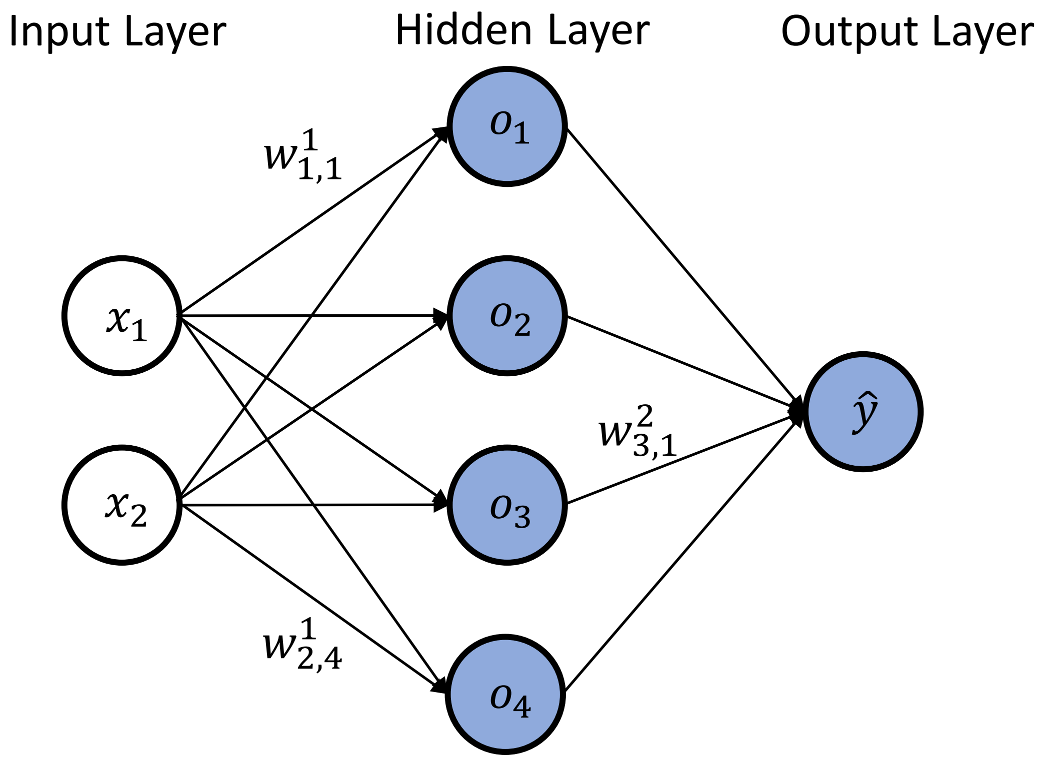
$$\hat{y} = Predicted\ Probability\ of\ Leaving$$
$$= \frac{1}{1 + e^{-(w_1 \times 22003 + w_2 \times 45 + w_0)}}$$

Calculate predicted probability for each observation and compare to the actual in the training data

If $\hat{y} \geq 0.5$, predict 1 (customer left)

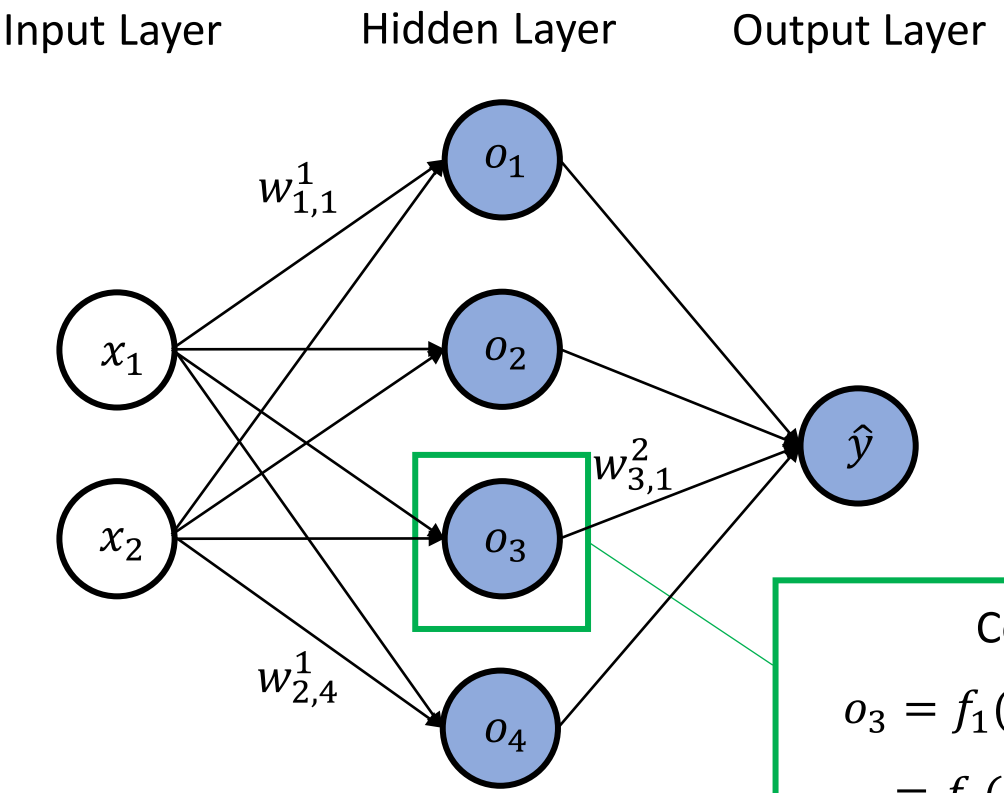
Neural Networks and Deep Learning

- Linear regression and logistic regression are both linear models
  - For realistic problems, the relationships between the dependent variable and independent variables are usually more complex
- Idea: use a composite function $g(h(x))$
  - This corresponds to adding hidden layers between the input and output layers



Neural Networks and Deep Learning

- Now, a layer of computation nodes are added, which can significantly increase the expressive power of the neural network
  - I.e., it can capture more complex relationships in the data

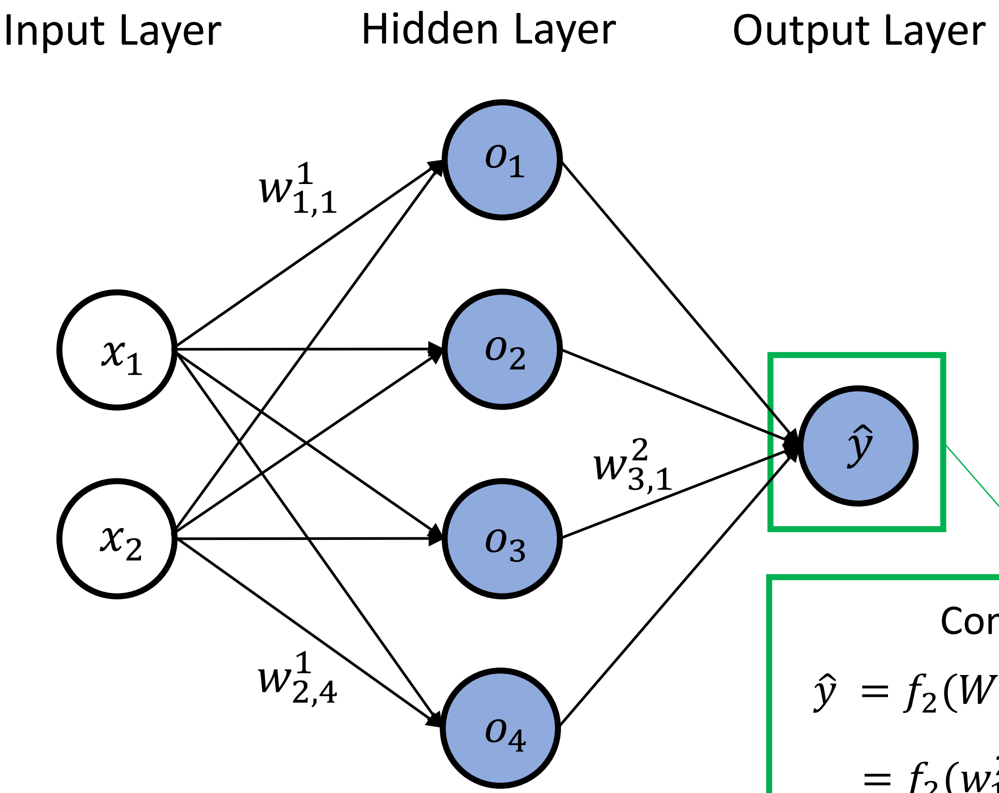


Computation Node

$$o_3 = f_1(Weights \times Inputs + Bias)$$
$$= f_1(w_{1,3}^1 \times x_1 + w_{2,3}^1 \times x_2 + w_{0,3}^1)$$

Neural Networks and Deep Learning

- The output of the neural network model can now be highly non-linear
  - Each layer can use a different activation function (e.g., f_1 versus f_2)
  - The non-linearity of the activation functions is critical
    - If only linear activation functions are used, the result is only a linear model!



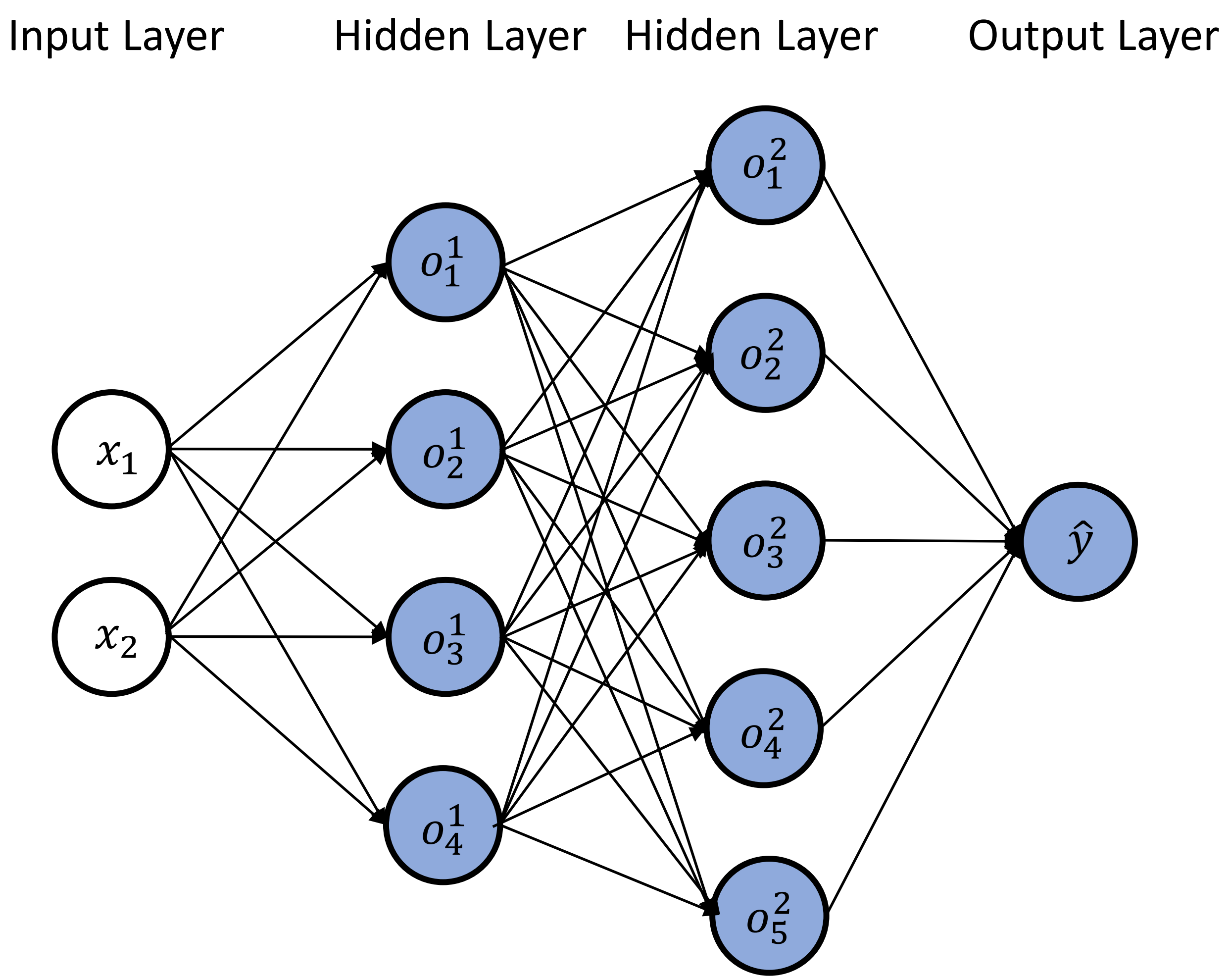
Computation Node

$$\hat{y} = f_2(Weights \times Inputs + Bias)$$

$$= f_2(w^2_{1,1}o_1 + w^2_{2,1}o_2 + w^2_{3,1}o_3 + w^2_{4,1}o_4 + w^2_0)$$

Neural Networks and Deep Learning

- What if we add another hidden layer?
  - Each hidden layer implements a transformation of the data
    - Hopefully, each transformation is a better representation of the data
  - Each hidden layer can have a different number of nodes
  - Each layer can use a different activation function



Neural Networks and Deep Learning

- The neural networks above are examples of densely-connected feed-forward neural networks
  - Other, more complex architectures exist (such as recurrent neural networks)
  - Neural networks are just *composite functions*
  - Can use a convenient algorithm, called backpropagation, to find optimal values of the weights