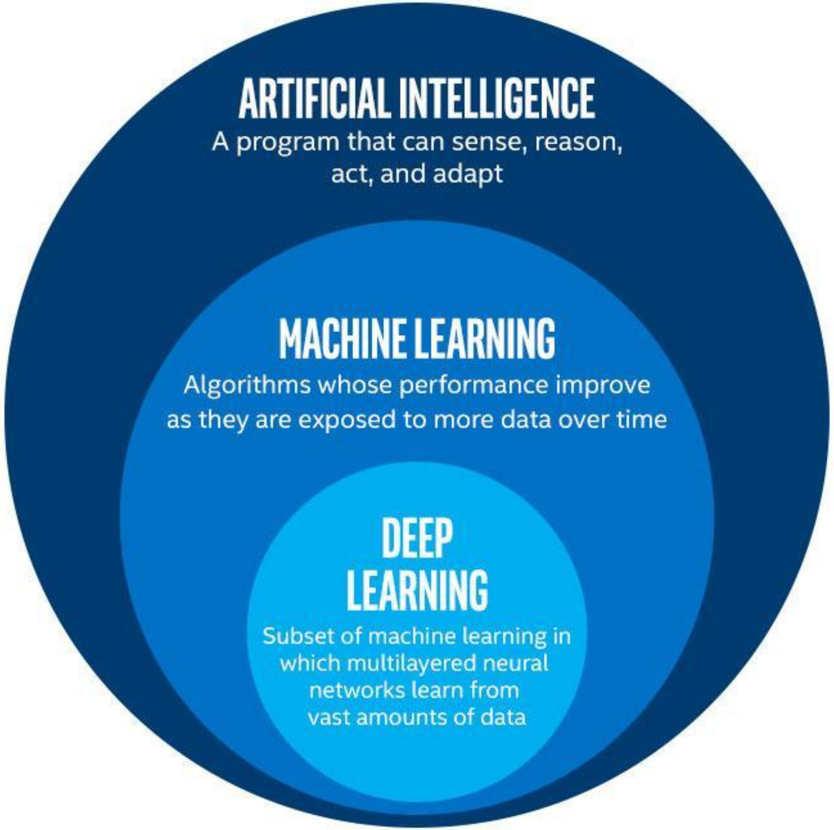
Introduction to Machine Learning

Definitions

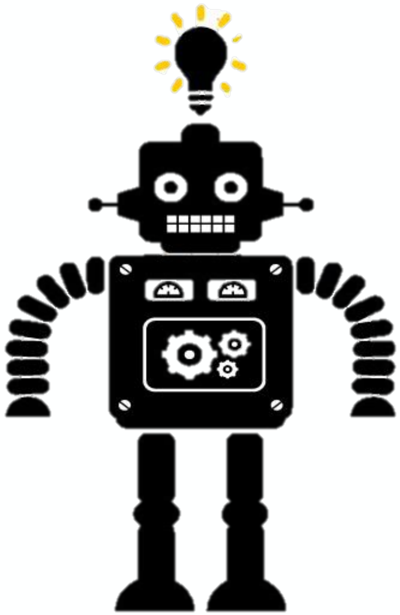
- Artificial Intelligence
- Machine Learning
- Deep Learning



What is Machine Learning?

Wikipedia:

Machine learning is the subfield of computer science that “gives computers the ability to learn without being explicitly programmed”



Machine Learning in Our Daily Lives

Spam Filtering	Web Search	Postal Mail Routing
Fraud Detection	Movie Recommendations	Vehicle Driver Assistance
Web Advertisements	Social Networks	Speech Recognition

Types of Machine Learning

Supervised	trains algorithms using data points have known outcome
Unsupervised	trains algorithms using data points have unknown outcome
Semi-Supervised	trains algorithms using data points have both above types
Reinforcement	trains algorithms using a system of reward and punishment.

Supervised

trains algorithms using data points have known outcome

Supervised Learning data

Example #	X	Y
0	X0	Y0
1	X1	Y1
2	X2	Y2
3	X3	Y3
4	X4	Y4
5	X5	Y5
6	X6	Y6

Classification

Regression

Unsupervised

trains algorithms using data points have unknown outcome

Unsupervised Learning data

Example #	X
0	X0
1	X1
2	X2
3	X3
4	X4
5	X5
6	X6

- Clustering
- Anomaly Detection
- Dimensionality Reduction
- Spare Representation
- Independent Representation

Semi-Supervised

trains algorithms using data points typically have a small amount of labelled with a large amount of unlabelled data

Semi-Supervised Learning data

Example #	X	Y
0	X0	Y0
1	X1	unknown
2	X2	unknown
3	X3	Y3
4	X4	unknown
5	X5	unknown
6	X6	unknown

Reinforcement trains algorithms using a system of reward and punishment.

Reinforcement Learning Timesteps

TimeStep #	State	Action	Reward
0	S0	A0	-1
1	S1	A1	0
2	S2	A2	0
3	S3	A3	0
4	S4	A4	0
5	S5	A5	1
6	S6	A6	1

Introduction to Supervised learning

Target vs. Features

Target: Column to predict

Features: Properties of the data used for prediction (non-target columns)

Features

sepal length	sepal width	petal length	petal width	species
6.7	3.0	5.2	2.3	virginica
6.4	2.8	5.6	2.1	virginica
4.6	3.4	1.4	0.3	setosa
6.9	3.1	4.9	1.5	versicolor
4.4	2.9	1.4	0.2	setosa
4.8	3.0	1.4	0.1	setosa
5.9	3.0	5.1	1.8	virginica
5.4	3.9	1.3	0.4	setosa
4.9	3.0	1.4	0.2	setosa
5.4	3.4	1.7	0.2	setosa

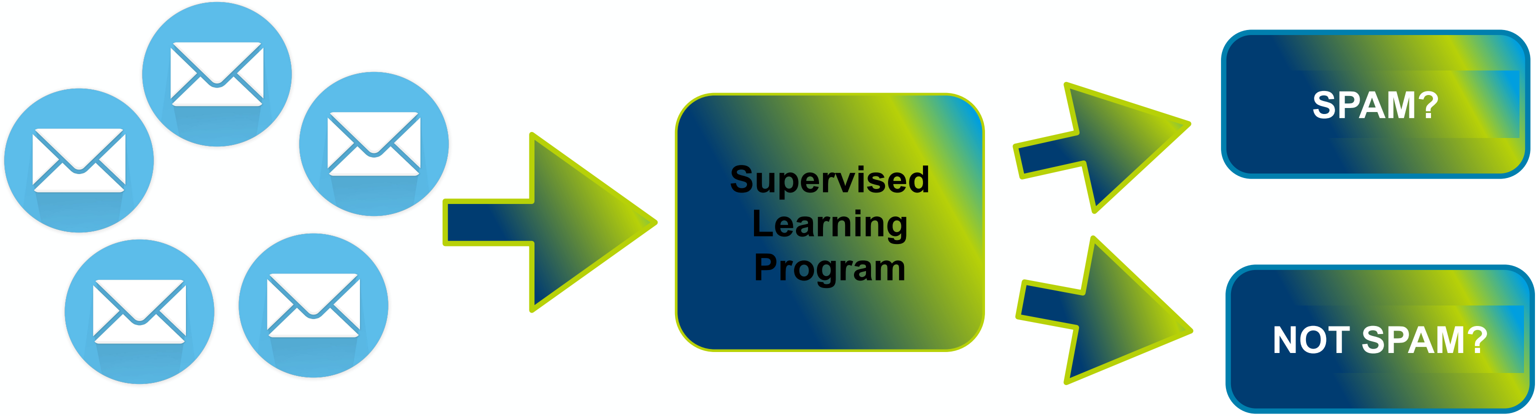
Target

Example: Supervised Learning Problem

Goal: Predict if an email is spam or not spam.

Data: Historical emails labeled as spam or not spam.

- **Target**: Spam or not spam
- **Features**: Email text, subject, time sent, etc.

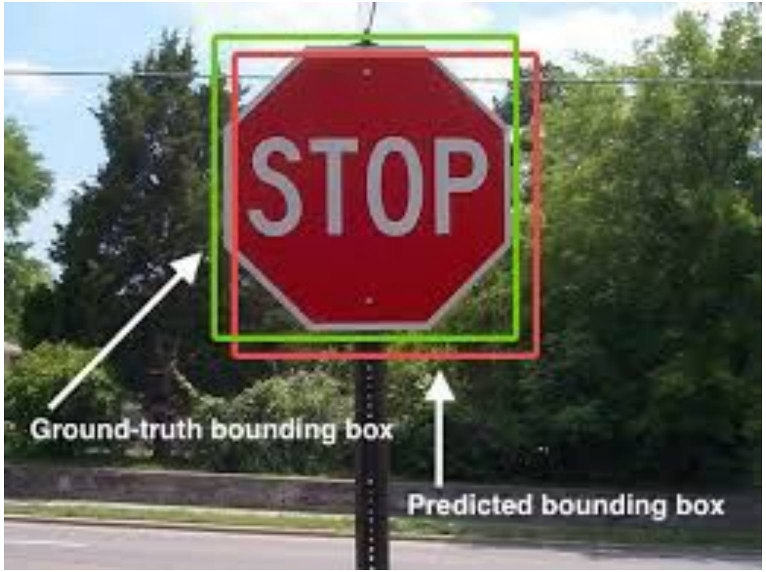


Example: Supervised Learning Problem

Goal: Predict location of bounding box around an object.

Data: Images with bounding box locations.

- **Target**: Corners of bounding box
- **Features**: Image pixels



Workflow

Problem Statement	What problem are you trying to solve?
Data Collection	What data do you need to solve it?
Data Exploration & Preprocessing	How should you clean your data so your model can use it?
Modeling	Build a model to solve your problem?
Validation	Did I solve the problem?
Decision Making & Deployment	Communicate to stakeholders or put into production?

Formulating a Supervised Learning Problem

For a Supervised Learning Problem:

- Collect a labeled dataset (features and target labels).
- Choose the model.
- Choose an evaluation metric:

  “What to use to measure performance.”
- Choose an optimization method:[1]

  “How to find the model configuration that gives the best performance.”

[1] *There are standard methods to use for different models and metrics.*

List of Common Supervised Learning Algorithms/Models

- Linear Regression
- Logistic Regression
- Decision Tree
- SVM
- Naive Bayes
- kNN
- K-Means
- Random Forest
- Gradient Boosting algorithms: GBM, XGBoost, LightGBM, CatBoost

Which Model?

Some considerations when choosing are:

- Time needed for training
- Speed in making predictions
- Amount of data needed
- Type of data
- Problem complexity
- Ability to solve a complex problem
- Tendency to overcomplicate a simple one

Evaluation Metric

There are many metrics available[1] to measure performance, such as:

- **Accuracy**: how well predictions match true values.
- **Mean Squared Error**: average square distance between prediction and true value.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

Mean square error formula

Accuracy target

[1]The wrong metric can be misleading or not capture the real problem.

Evaluation Metric

The wrong metric can be misleading or not capture the real problem.

For example: consider using **accuracy** for spam/not spam.

- If 99 out of 100 emails are actually spam, then a model that is predicting spam every time will have *99% accuracy*.
- This may force an important *real* email into spam, even though it has a high accuracy metric.

Email

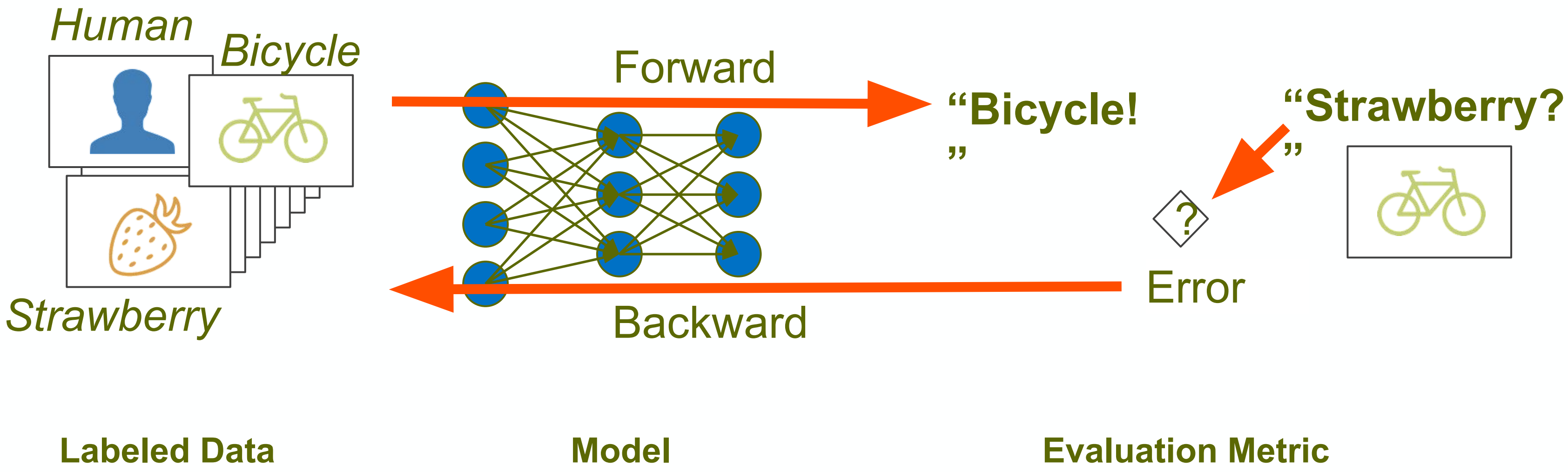
Training

Training Data: The dataset used to train the model.

Optimization: Configures the model for best performance.

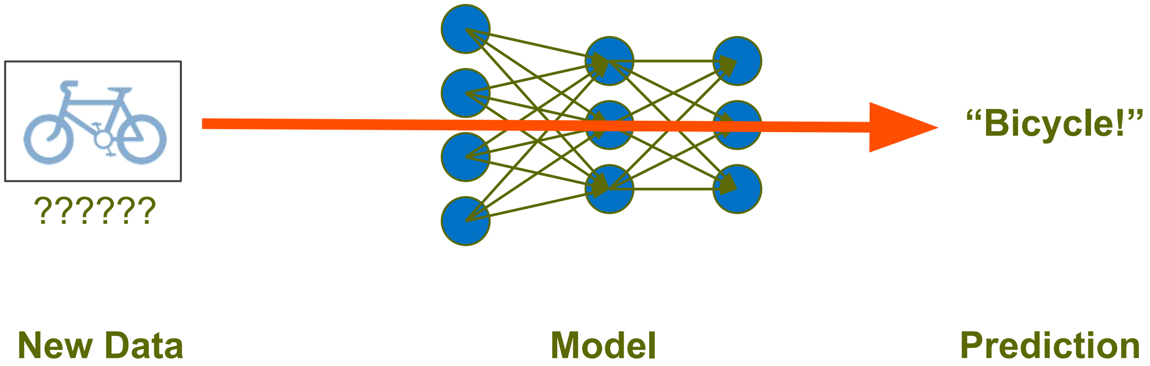
Training

With these pieces, a model can now be trained to find the best configuration.



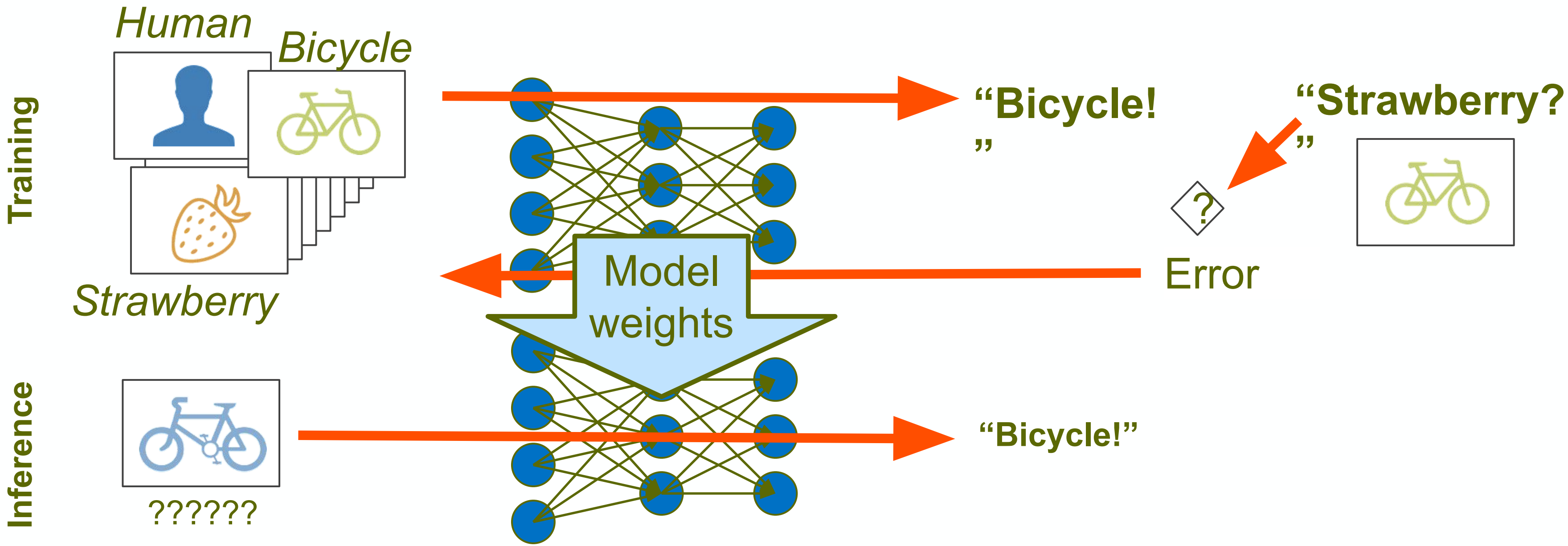
Inference

Once the model is trained, we can provide new examples for predictions.



Training vs. Inference

Goal: Perform well on unseen data during inference.



Supervised Learning Overview

Training: Train a model with known data.

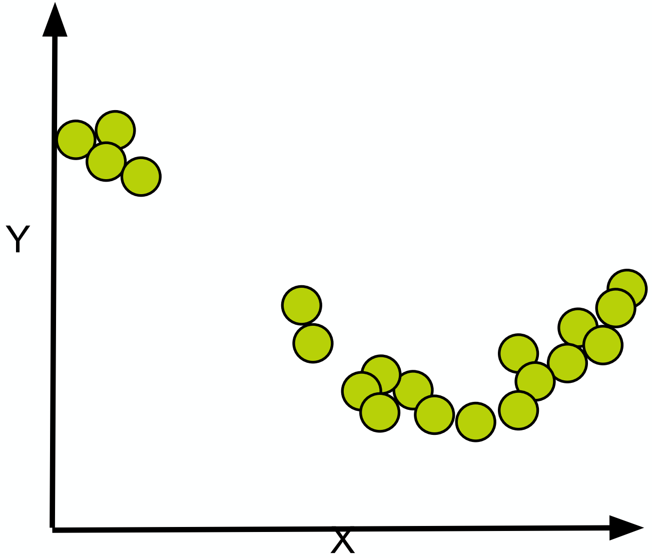
Data with answers (features and labels) **+** **Model** $\xrightarrow{\textbf{Fit}}$ **Trained Model**

Inference: Feed unseen data into trained model to make predictions.

Data without answers (features only) **+** **Trained Model** $\xrightarrow{\textbf{Predict}}$ **Predicted Answer**

Curve Fitting: Overfitting vs. Underfitting Example

Goal: Fit a curve to the data.

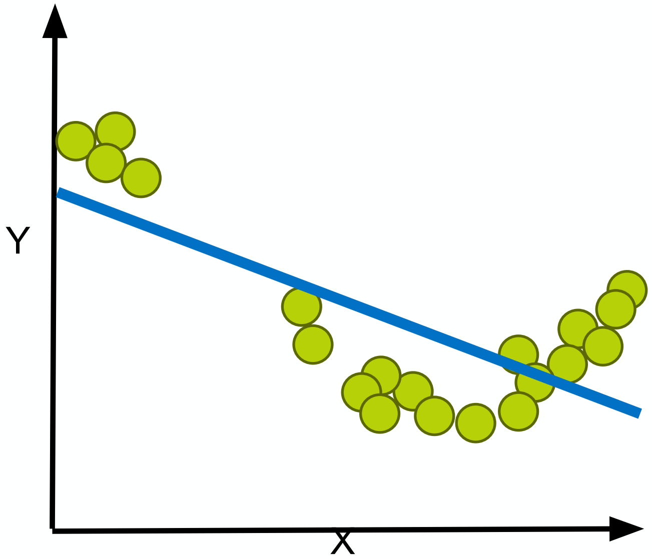


Curve Fitting: Underfitting Example

The curve can be too simple.

- This is called “underfitting”
- Poor fit on training data
- Poor fit on unseen data

Underfitting: Model is missing systematic trends in data.

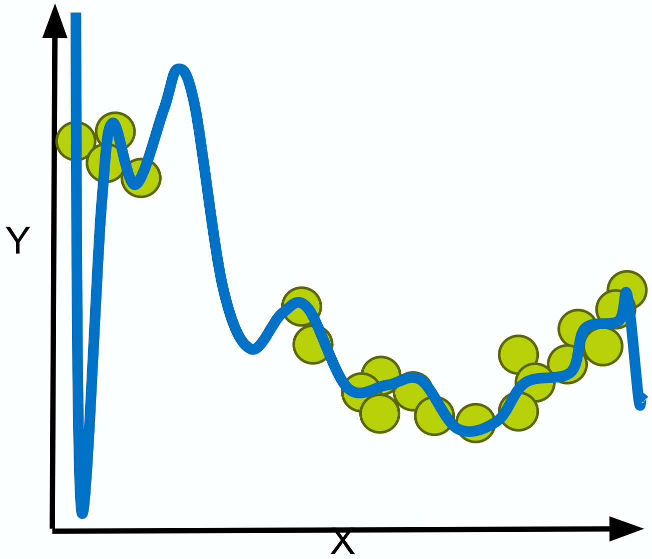


Curve Fitting: Overfitting Example

The curve can be too complex.

- This is called “overfitting”
- Good fit on training data
- Poor fit on unseen data

Overfitting: Model is too sensitive and fits the “noise” in the training data.

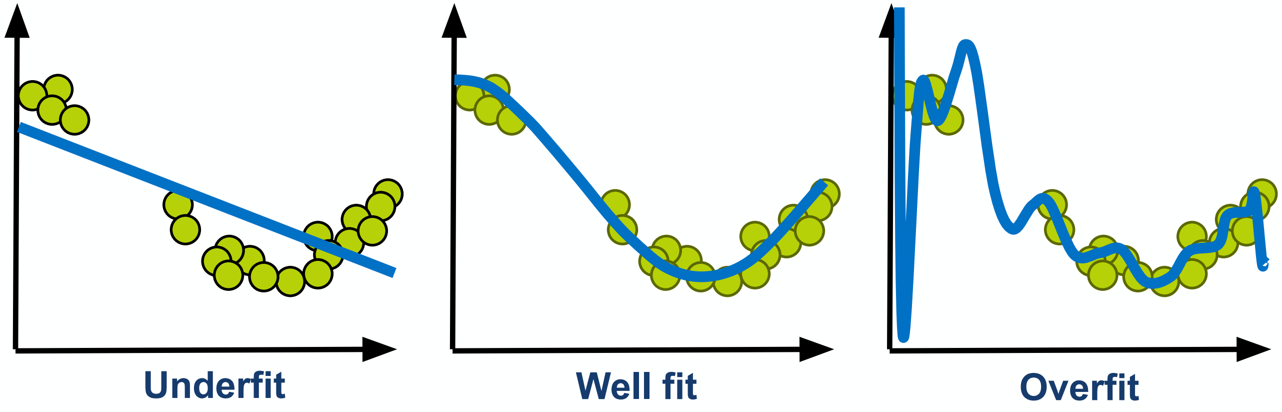


Curve Fitting Problem

Problem: Unseen data isn't available during training.

- How can performance be estimated?

When measuring performance on the training data, there is a tendency to overfit.



Solution: Split Data Into Two Sets

Training Set: Data used during the training process.

Test Set: Data used to measure performance, simulating unseen data[1].

sepal length	sepal width	petal length	petal width	species
6.7	3.0	5.2	2.3	virginica
6.4	2.8	5.6	2.1	virginica
4.6	3.4	1.4	0.3	setosa
6.9	3.1	4.9	1.5	versicolor
4.4	2.9	1.4	0.2	setosa
4.8	3.0	1.4	0.1	setosa
5.9	3.0	5.1	1.8	virginica
5.4	3.9	1.3	0.4	setosa
4.9	3.0	1.4	0.2	setosa
5.4	3.4	1.7	0.2	setosa

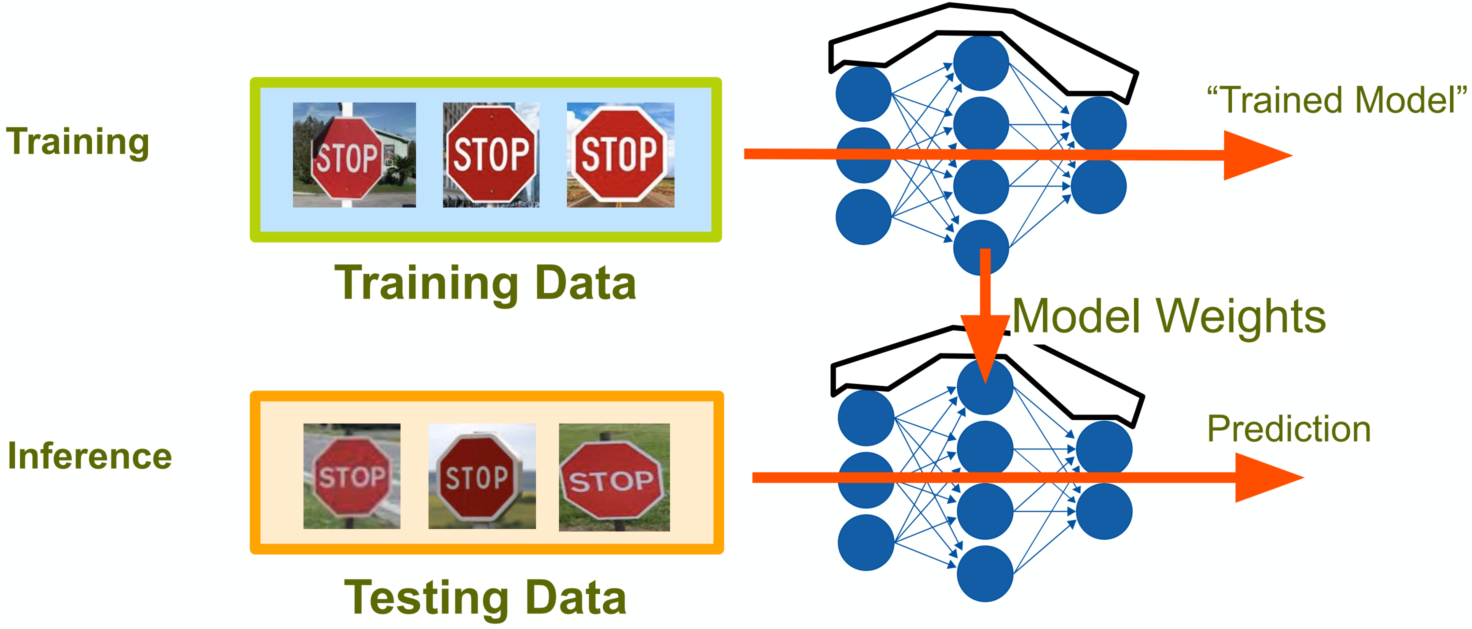
Training Set

Testing Set

[1] *Not used during the training process.*

Train-Test Split

Evaluate trained model on data it hasn't "seen" before to simulate real-world inference.



Thank you