internship studio

Welcome to

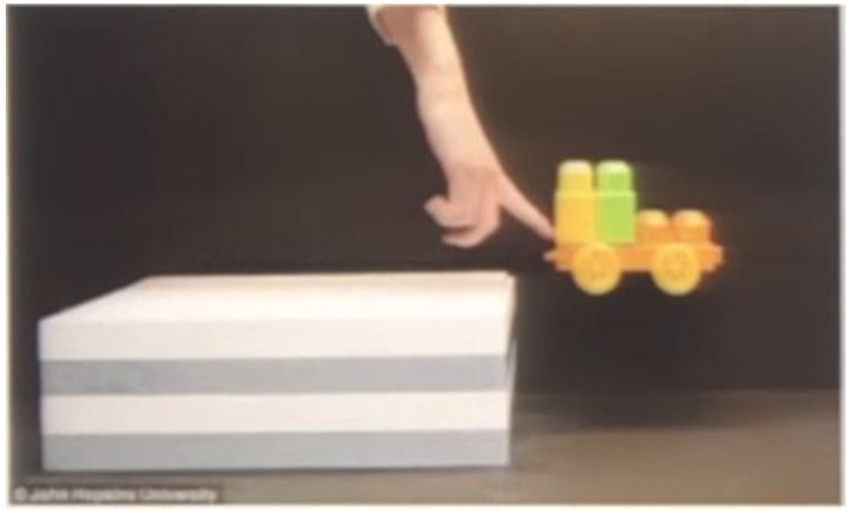
INTERNSHIP STUDIO

Module 04 | Lesson 02

Key concepts: features, labels, training, and testing

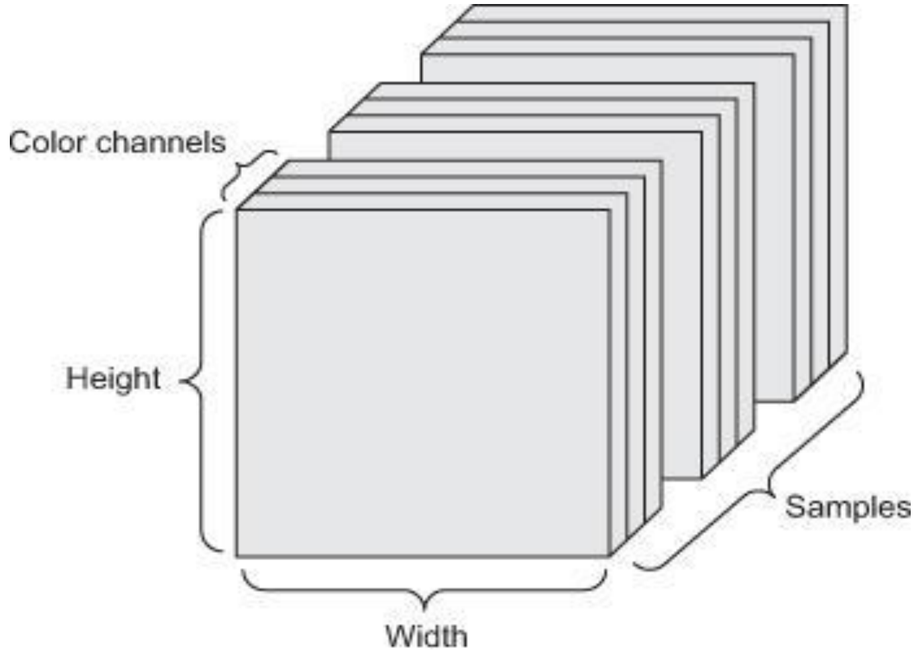
What is Data?

- Humans learn by observation and unsupervised learning
  - model of the world / common sense reasoning
- Machine learning needs lots of (labeled) data to compensate



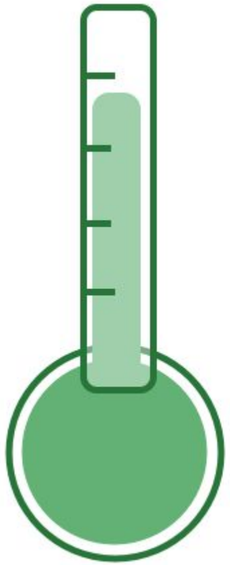
What is Data?

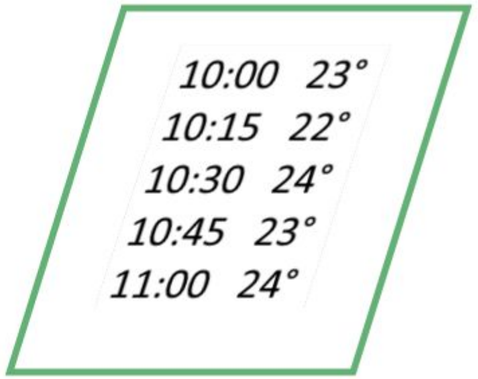
- Tensors: generalization of matrices to *n* dimensions (or rank, order, degree)
  - 1D tensor: vector
  - 2D tensor: matrix
  - 3D, 4D, 5D tensors
  - `numpy.ndarray(shape, dtype)`
- Training – validation – test split (+ adversarial test)
- Minibatches
  - small sets of input data used at a time

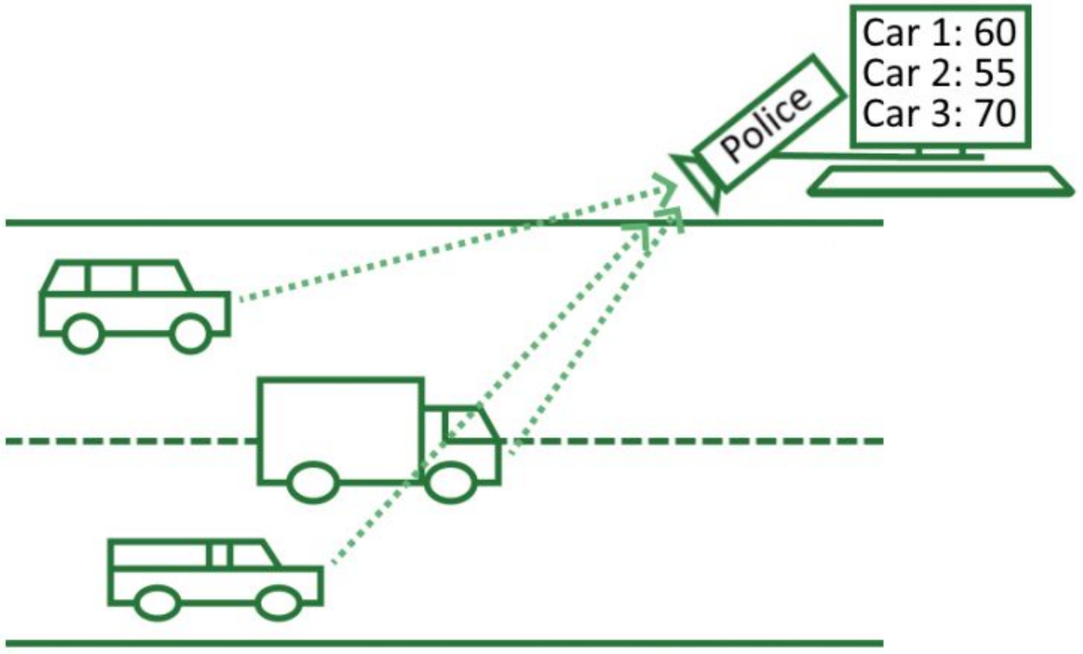


Feature

Features are individual independent variables which acts as the input in the system. In statistics, they talk about “variables”, which indicate the characteristics associated with a given statistical unit. In machine learning, we call these characteristics “features”.

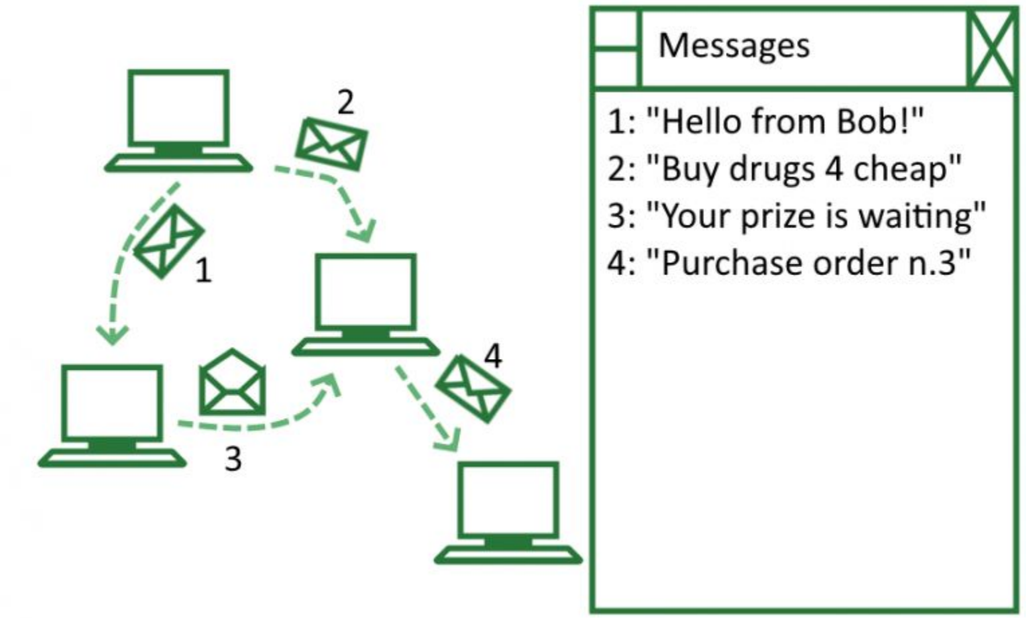






Feature

Features are individual independent variables which acts as the input in the system. In statistics, they talk about “variables”, which indicate the characteristics associated with a given statistical unit. In machine learning, we call these characteristics “features”.



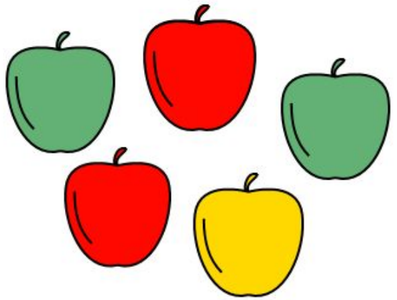
SPAM/NON SPAM

Types of Features

- Approximation of Real Numbers: Numerical data types
- Texts as Features:

Corpus	Corpus - processed
The pen is *on* the table	pen, table, *on*
The pen is *by* the table	pen, table, *by*
The pen is *under* the table	pen, table, *under*

- Categorical Features:

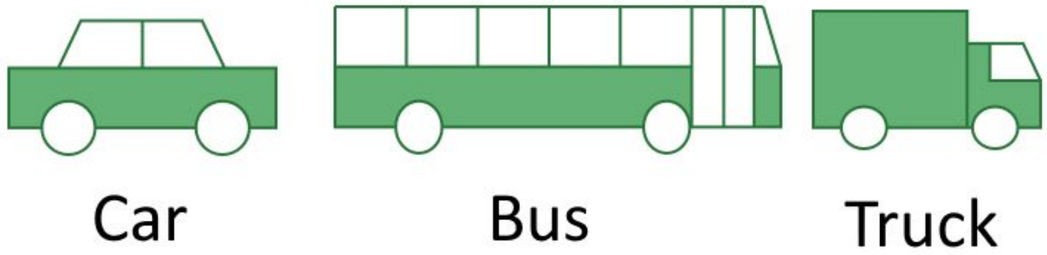


2x Red
1x Yellow
2x Green

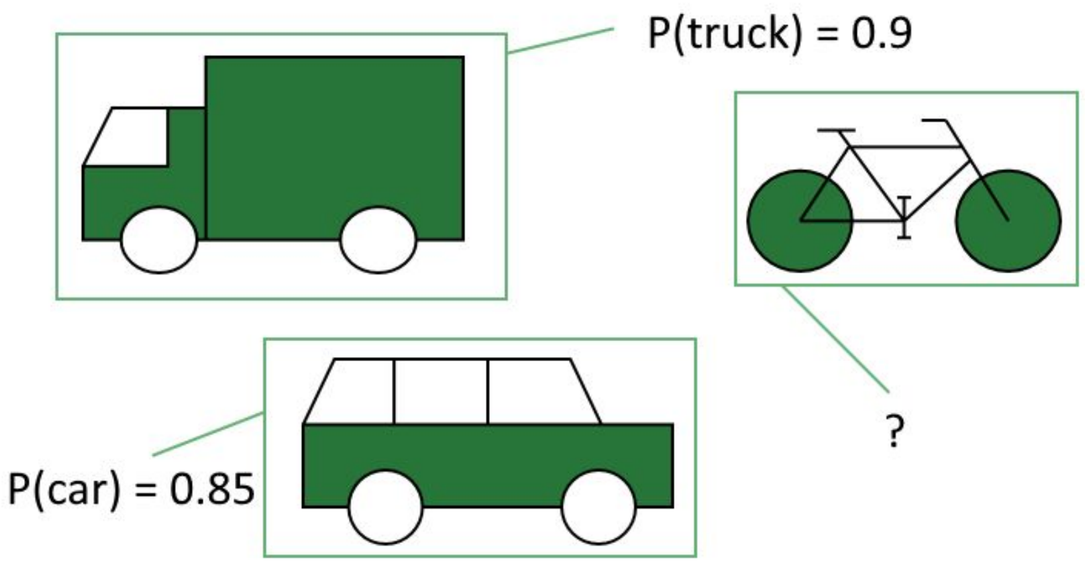
Labels

A label is the thing we're predicting—the y variable in simple linear regression. The label could be the future price of wheat, the kind of animal shown in a picture, the meaning of an audio clip, or just about anything.

- labels are normally assigned before we build, or even identify, any machine learning model
- labels can be used as inputs to some models, in particular when we question and want to verify their independence
- labeled data about the relationship that exists between prior knowledge on a certain phenomenon and the labels associated with observations.



Labels



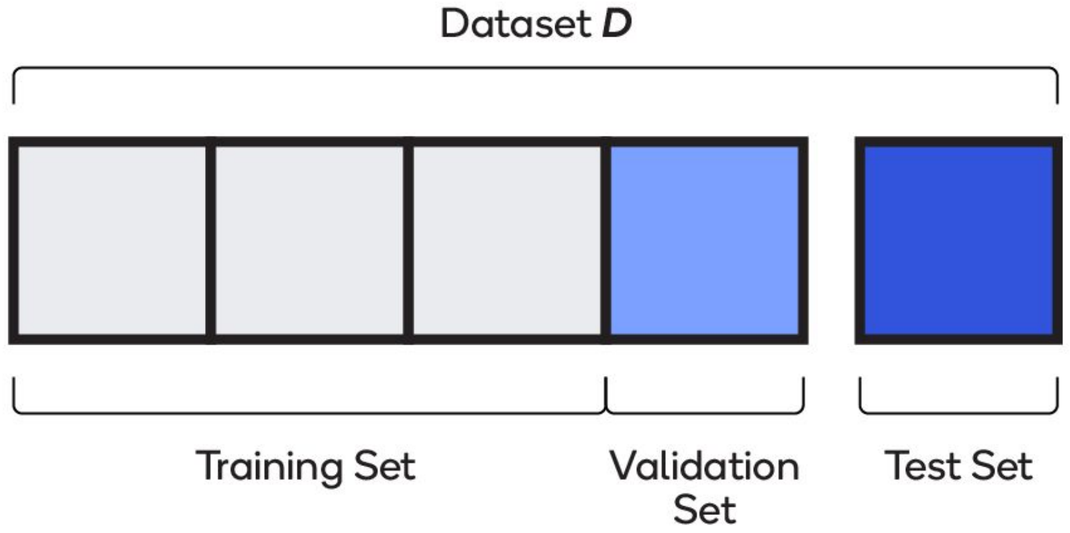
Features					Label
Position	Experience	Skill	Country	City	Salary ($)
Developer	0	1	USA	New York	103100
Developer	1	1	USA	New York	104900
Developer	2	1	USA	New York	106800
Developer	3	1	USA	New York	108700
Developer	4	1	USA	New York	110400
Developer	5	1	USA	New York	112300
Developer	6	1	USA	New York	114200
Developer	7	1	USA	New York	116100
Developer	8	1	USA	New York	117800
Developer	9	1	USA	New York	119700
Developer	10	1	USA	New York	121600

Training/Testing

The process of determining the ideal parameters (weights and biases) comprising a model. During training, a system reads in examples and gradually adjusts parameters. Training uses each example anywhere from a few times to billions of times.

Model training
Model training for deep learning includes splitting the dataset, tuning hyperparameters and performing batch normalization.

Splitting the dataset
The data collected for training needs to be split into three different sets: training, validation and test.



Training/Testing

Training — Up to 75 percent of the total dataset is used for training. The model learns on the training set; in other words, the set is used to assign the weights and biases that go into the model.

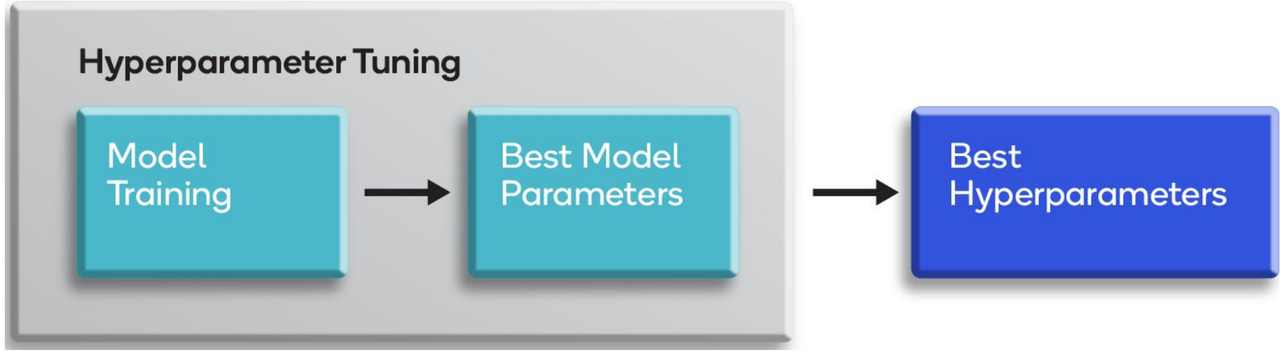
Validation — Between 15 and 20 percent of the data is used while the model is being trained, for evaluating initial accuracy, seeing how the model learns and fine-tuning hyperparameters. The model sees validation data but does not use it to learn weights and biases.

Test — Between five and 10 percent of the data is used for final evaluation. Having never seen this dataset, the model is free of any of its bias.

Training/Testing

Hyperparameter Tuning:

Hyperparameters can be imagined as settings for controlling the behavior of a training algorithm, as shown below.

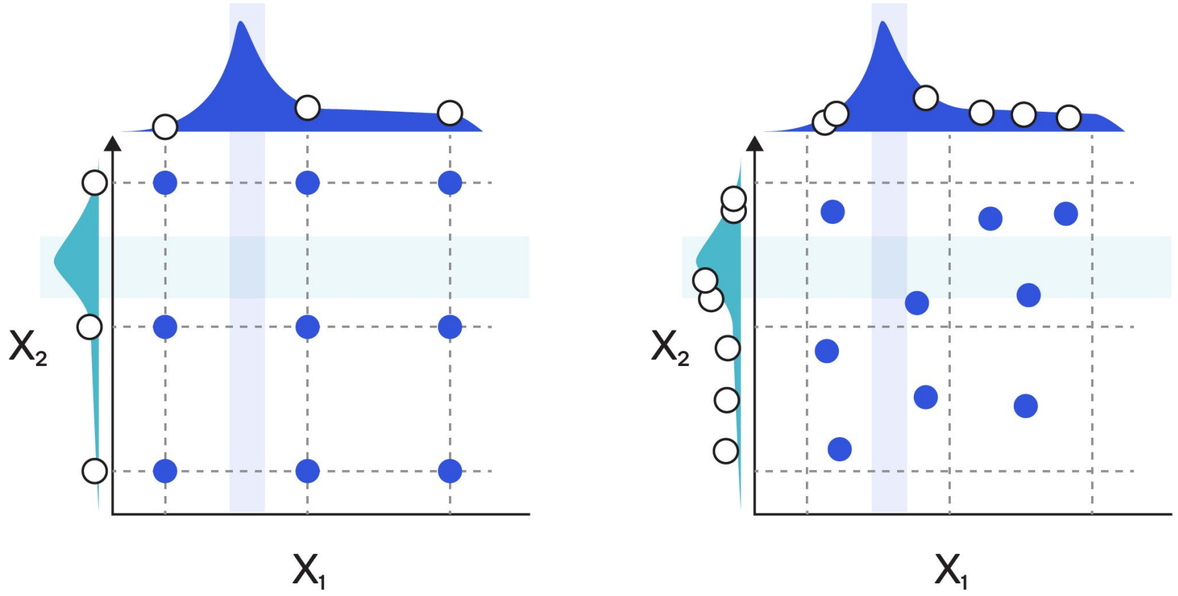


In the context of deep learning, examples of hyperparameters are:

1. Learning rate
2. Number of hidden units
3. Convolution kernel width
4. Regularization techniques

Training/Testing

There are two common approaches to tuning hyperparameters, as depicted in the diagram below.



(a) Standard Grid Search

(b) Random Search

Training/Validation Loss

A metric representing a model's loss during a particular training iteration. For example, suppose the loss function is Mean Squared Error. Perhaps the training loss (the Mean Squared Error) for the 10th iteration is 2.2, and the training loss for the 100th iteration is 1.9.

A loss curve plots training loss vs. the number of iterations. A loss curve provides the following hints about training:

- A downward slope implies that the model is improving.
- An upward slope implies that the model is getting worse.
- A flat slope implies that the model has reached convergence.