Supervised Learning

Rowel Atienza, PhD
University of the Philippines
github.com/roatienza
2022

A computer program is said to learn from ***experience E*** *with respect to some class of* ***tasks T****, and* ***performance measure P****, if its performance at tasks in T, as measured by P, improves with experience E.* – Tom Mitchell

Supervised Learning

Experience

Task

Performance

Robot Learning

Learn from Experience (E) to perform Task (T) by improving the Performance (P)

https://spectrum.ieee.org/covariant-ai-gigantic-neural-network-to-automate-warehouse-picking

Supervised Learning

Experience: Dataset $\mathcal{D}_{train} = \{\boldsymbol{x}_n, y_n\}$ with N examples: input $\boldsymbol{x}_n \in \mathbb{R}^D$ with corresponding label $y_n \in \mathbb{R}$

Task: Classifier $y = f(\boldsymbol{x}, \boldsymbol{\theta}) : \mathbb{R}^D \rightarrow \mathbb{R}$ where $\boldsymbol{\theta}$ are parameters

Ground Truth Experience: Dataset $\mathcal{D}_{test} = \{\boldsymbol{x}_m, y_m\}$, $\hat{y}_m = f(\boldsymbol{x}_m, \boldsymbol{\theta})$ for $m = 1 \ldots M$

Performance: Accuracy on $\mathcal{D}_{test}$

$$Accuracy = \frac{\#\ Correct\ Prediction}{Size\ of\ Test\ Dataset} = \frac{\sum_{m=1}^{M}(\hat{y}_m\ equals\ y_m)}{M}$$

Machine Learning & Generalization

Initial Test $Accuracy_0$ of $\hat{y}_m = f(\boldsymbol{x}_m, \boldsymbol{\theta}^0)$, $\boldsymbol{\theta}^0$ initial parameters

Machine Learning: Finding the optimal parameters $\boldsymbol{\theta}^*$ using $\mathcal{D}_{train} = \{\boldsymbol{x}_n, y_n\}$: $\boldsymbol{\theta}^0 \rightarrow \boldsymbol{\theta}^*$

$$f(\boldsymbol{x}_n, \boldsymbol{\theta}^*) \approx y_n \quad n = 1 \dots N$$

Final Test $Accuracy_*$ of $\hat{y}_m = f(\boldsymbol{x}_m, \boldsymbol{\theta}^*)$

Machine Learning: $Accuracy_* > Accuracy_0$

Generalization: $Accuracy_{*(train)} - Accuracy_{*(test)} = Gap \rightarrow 0$

Experience

MNIST Dataset

$\mathcal{D}_{train} = \{\boldsymbol{x}_n, y_n\}, N = 60{,}000$

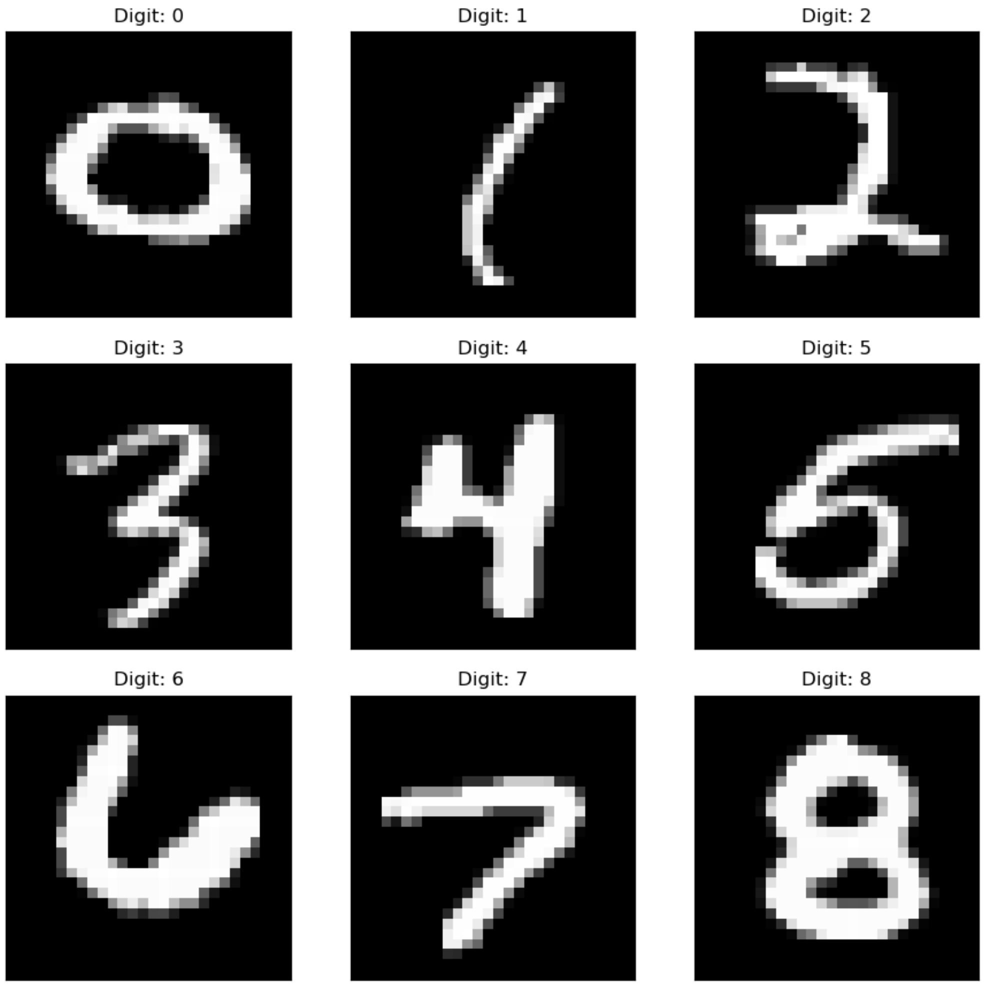
$\mathcal{D}_{test} = \{\boldsymbol{x}_m, y_m\}, M = 10{,}000$

$\boldsymbol{x}$: 28×28 grayscale images of digits 0 to 9

y : class label

Available:

`torchvision.datasets.MNIST()`



LJSpeech Dataset

$\mathcal{D}_{train} = \{\boldsymbol{x}_n, \boldsymbol{y}_n\}, N = 12{,}228$

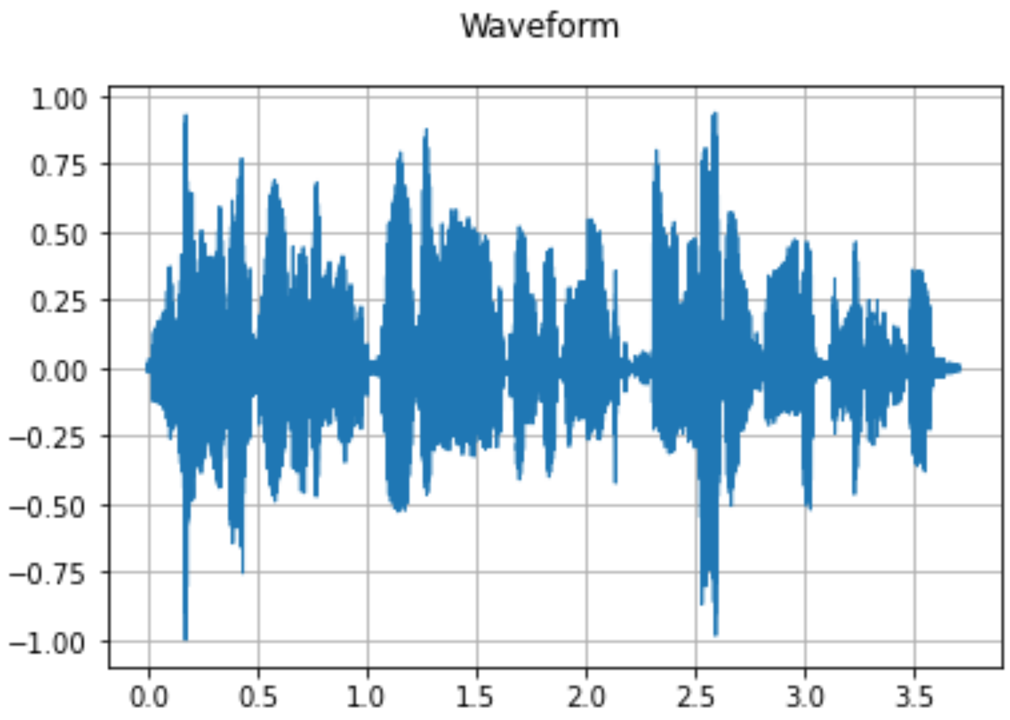
$\mathcal{D}_{test} = \{\boldsymbol{x}_m, \boldsymbol{y}_m\}, M = 523$

$\boldsymbol{x}$: speech

$\boldsymbol{y}$: text transcript

Available:

`torchaudio.datasets.LJSPEECH()`



the association was organized under the most promising auspices

Stanford Sentiment Treebank Dataset

$\mathcal{D}_{train} = \{\boldsymbol{x}_n, y_n\}, N = 67{,}349$

$\mathcal{D}_{dev} = \{\boldsymbol{x}_p, y_p\}, P = 872$

$\mathcal{D}_{test} = \{\boldsymbol{x}_m, y_m\}, M = 1{,}821$

$\boldsymbol{x}$: phrases

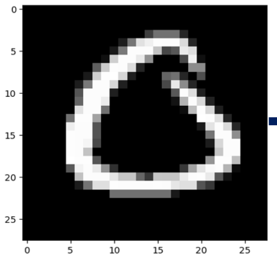
$\boldsymbol{y}$: sentiment (+ or 1.0 /- or 0.0)

Available:

`torchtext.datasets.SST2()`

Task

Multi-label Classification (Recognition)



$$\xrightarrow{} \boxed{f(\cdot)} \xrightarrow{} \begin{cases} y \in \mathbb{R} \\ y \in \mathbb{R}^{10} \end{cases} \quad \begin{matrix} y \in \{0,1,2,\dots 9\} \\ if\ one-hot \end{matrix}$$

$$\boldsymbol{x} \in \mathbb{R}^{28\times28\times1} \to \mathbb{R}^{784}$$

flatten

One-hot vector

$$\begin{bmatrix} 0 \\ 1 \\ 2 \\ \vdots \\ 9 \end{bmatrix} \to \begin{bmatrix} 1,0,0,0,0,0,0,0,0,0 \\ 0,1,0,0,0,0,0,0,0,0 \\ 0,0,1,0,0,0,0,0,0,0 \\ \vdots \\ 0,0,0,0,0,0,0,0,0,1 \end{bmatrix}$$

Also known as Multi-class Logistic Regression

Task: Binary Classification

Task: Sentiment Classification

She is a great singer

$$\boldsymbol{x} \in \mathbb{R}^{512\times 5} \longrightarrow \boxed{f(\cdot)} \longrightarrow y \in \mathbb{R} = \begin{cases} 0 & \cancel{Negative} \\ 1 & Positive \end{cases}$$

Assuming embedding size is 512

Also known as Binary Logistic Regression

Task: Sequence to Sequence

TEXT: {the association was organized under the most promising auspices}

→

PHONEME: {DH IY0 AH0 S OW2 S IY0 EY1 SH AH0 N W AH0 Z AO1 R G AH0 N AY2 Z D AH1 N D ER0 DH AH0 M OW1 S T P R AA1 M AH0 S IH0 NG AO1 S P IH0 S IH0 Z}

Task: Text to Speech

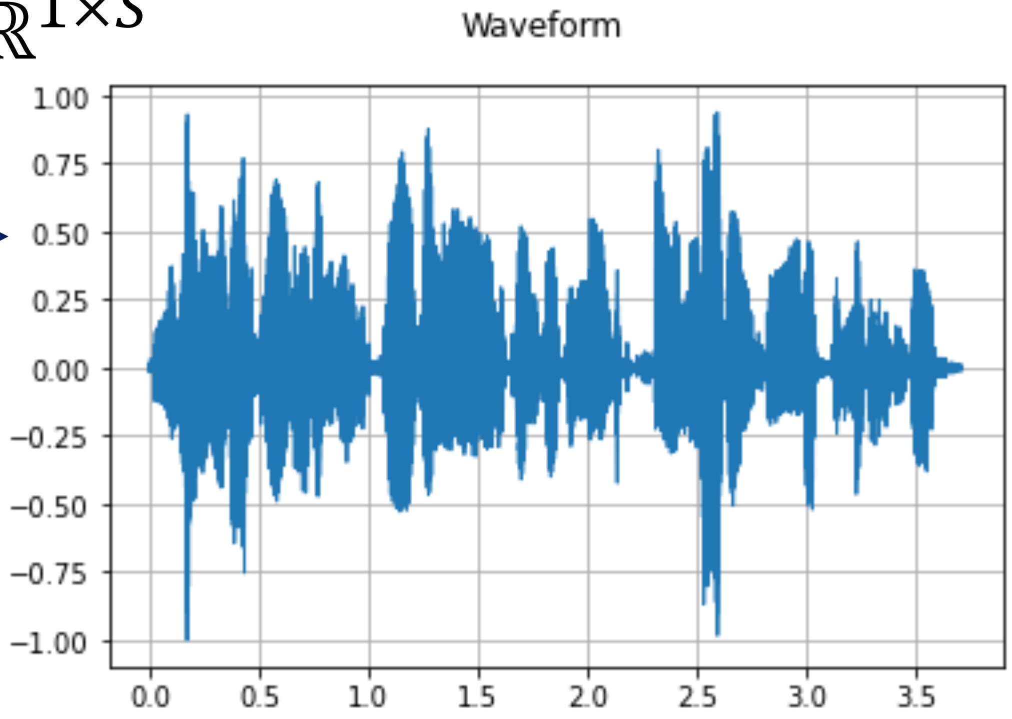
$\boldsymbol{x} \in \mathbb{R}^{512 \times P}$ $\quad$ $\boldsymbol{y} \in \mathbb{R}^{1 \times S}$

$$\longrightarrow \boxed{f(\cdot)} \longrightarrow$$

$P: Phoneme\ Length$

$S: \#Speech\ Samples$

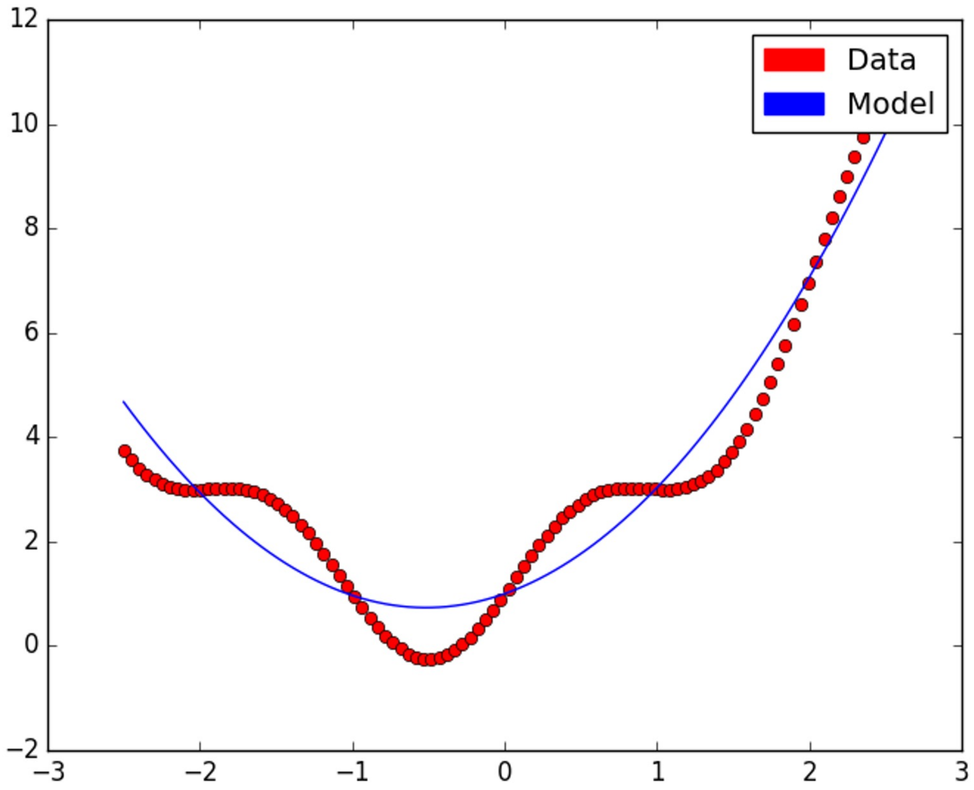
Assuming embedding size is 512



Task: Curve Fitting

$$\mathcal{D}_{train} = \{x_n, y_n\}$$
$$\mathcal{D}_{test} = \{x_m, y_m\}$$

Given: Data points



Also known as Linear Regression

Performance

Accuracy (Classification) ↑

Accuracy: Classification **Classification Performance Score on** $\mathcal{D}_{test} = \{(\boldsymbol{x}_m, y_m)\}$

$$Accuracy = \frac{\#\ Correct\ Prediction}{Size\ of\ Test\ Dataset} = \frac{\sum_{m=1}^{M}(\hat{y}_m\ equals\ y_m)}{M}$$

Generative Model (Voice/Video) ↑

Mean Opinion Score (MOS) is a numerical measure of the human-judged overall quality of an event or experience.

5 Excellent

4 Good

3 Fair

2 Poor

1 Bad

Somewhere around 4.3 - 4.5 is considered an excellent quality target. Video quality becomes unacceptable below a MOS of roughly 3.5.

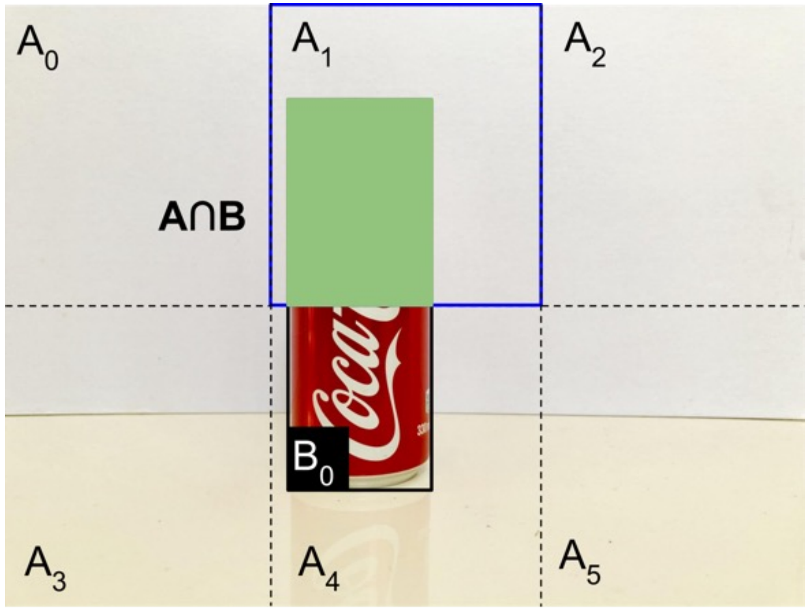
https://www.twilio.com/

Object Detection ↑

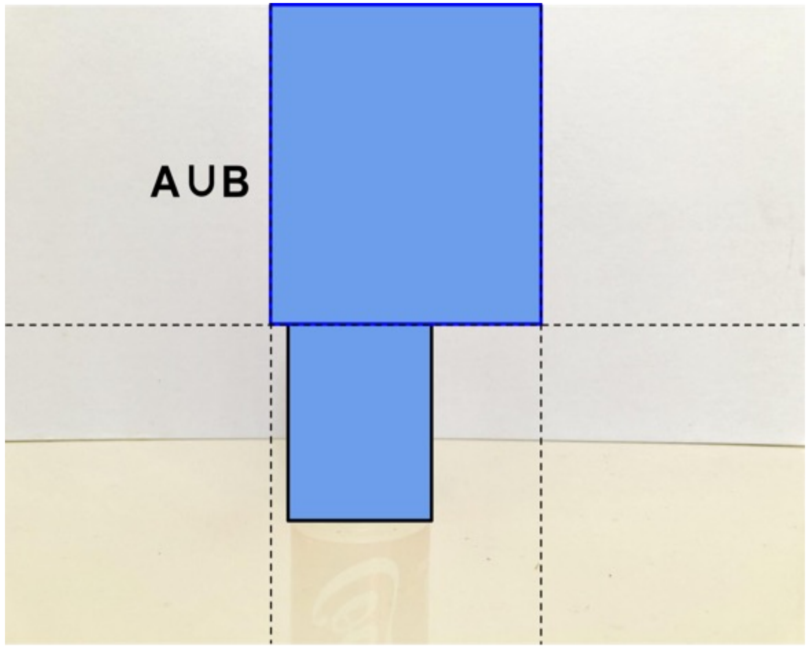
Intersection over Union (IoU)

IoU is also known as *Jaccard index:* $IoU = \frac{A \cap B}{A \cup B}$

Intersection : $A_1 \cap B_0$



Union : $A_1 \cup B_0$



Object Detection & Classification ↑

Precision (Bad guys out): ↑

$$precision = \frac{tp}{tp + fp} = \frac{True\_Positive}{True\_Positive + False\_Positive} = \frac{tp}{total\_positive}$$

Recall (Good guys in): ↑

$$recall = \frac{tp}{tp + fn} = \frac{True\_Positive}{True\_Positive + False\_Negative}$$
$$= \frac{tp}{true\_total\_positive}$$

Related Performance Measures: Average Precision, F1 Score

Learning

Learning by Minimizing a Loss Function

Ground truth: y_n

Prediction: $\hat{y}_n$

Loss Function: $\mathcal{L} = l(y_n, \hat{y}_n)$

Machine Learning: Use dataset, $\mathcal{D}_{train} = \{(\boldsymbol{x}_n, y_n)\}, \quad n = 1 \dots N$ to estimate the parameters $\boldsymbol{\theta}$ by minimizing $L = l(y_n, \hat{y}_n)$ using an optimizer

Assumption: $(\boldsymbol{x}_1, y_1) \dots (\boldsymbol{x}_n, y_n)$ are IID

IID: Independent and identically distributed

Empirical Risk Minimization

$$R_{emp}(f, \boldsymbol{X}, \boldsymbol{y}) = \frac{1}{N} \sum_{i=1}^{n} l(y_n, \hat{y}_n)$$

Where $\boldsymbol{X} := [x_1, \ldots, x_n]^T \in \mathbb{R}^{N \times D}$ and $\boldsymbol{y} := [y_1, \ldots, y_n]^T \in \mathbb{R}^N$

Least Squares Loss

$$\mathcal{L} = l(y_n, \hat{y}_n) = (y_n - \hat{y}_n)^2$$

$$R_{emp}(f, \boldsymbol{X}, \boldsymbol{y}) = \frac{1}{N}\sum_{i=1}^{n}(y_n - \hat{y}_n)^2 = \frac{1}{N}\sum_{i=1}^{n}\big(y_n - f(\boldsymbol{x}_n, \boldsymbol{\theta})\big)^2$$

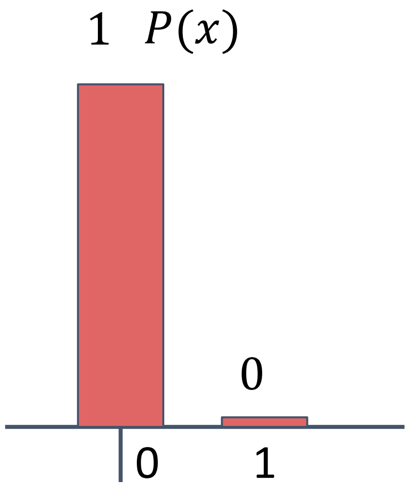
Loss Functions

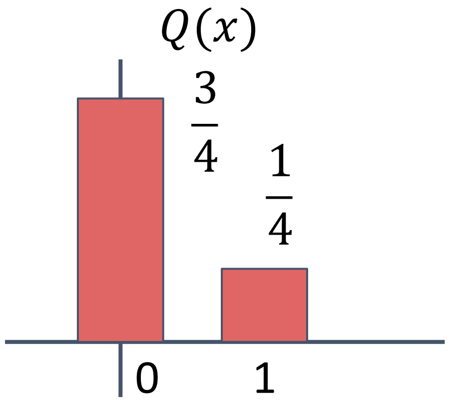
Loss Function	**Equation**
Mean Squared Error (MSE)	$\frac{1}{categories} \sum_{i=1}^{categories} \left(y_i^{label} - y_i^{prediction}\right)^2$
Mean Absolute Error (MAE)	$\frac{1}{categories} \sum_{i=1}^{categories} \left\lvert y_i^{label} - y_i^{prediction}\right\rvert$
Categorical Cross Entropy (CE)	$-\sum_{i=1}^{categories} y_i^{label} \log y_i^{prediction}$
Binary Cross Entropy (BCE)	$-y_1^{label} \log y_1^{prediction} - \left(1 - y_1^{label}\right) \log\left(1 - y_1^{prediction}\right)$

Categorical Cross-Entropy (CE)

For discrete distribution, Categorical Cross-Entropy is:

$$CE = H(P,Q) = -\sum_i \underbrace{P(x_i)}_{\text{Empirical Label}} \underbrace{\log Q(x_i)}_{\text{Predicted Label}}$$





What is CE?

$$CE = -\left(1\log\frac{3}{4} + 0\log\frac{1}{4}\right) = 0.29$$

Minimizing $H(P,Q)$ minimizes the distance of prediction model Q from the empirical model P

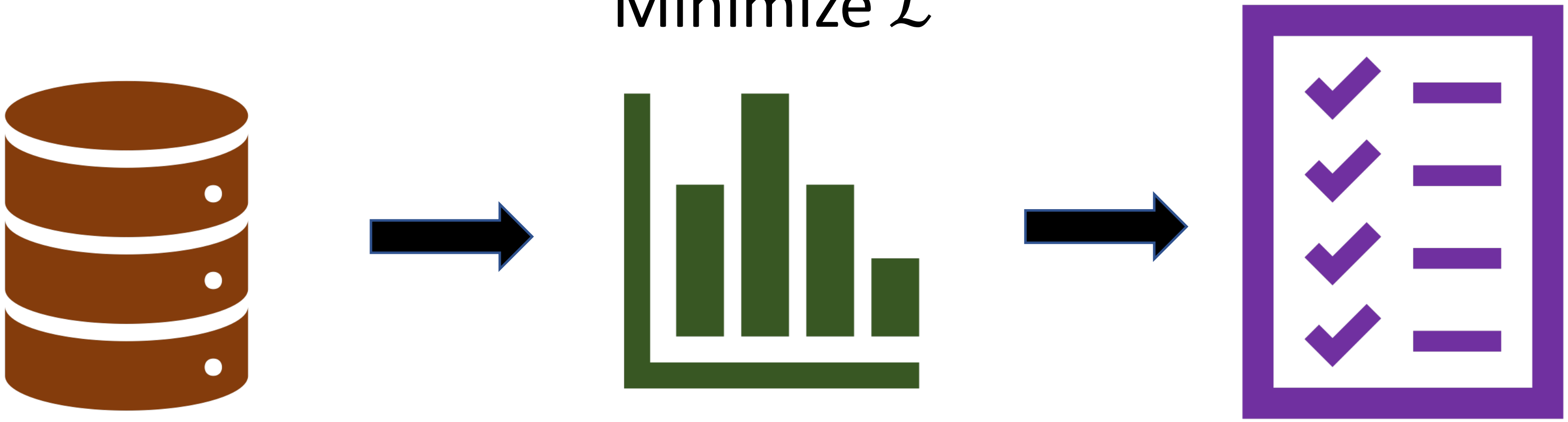
Machine Learning Pipeline

$\mathcal{D}_{train}$

$f(\boldsymbol{x}, \boldsymbol{\theta}^0) \rightarrow f(\boldsymbol{x}, \boldsymbol{\theta}^*)$

Minimize $\mathcal{L}$

$\mathcal{D}_{test}$



Experience

Task

Performance

End