Introduction to NLP

Training for TrueVoice

Goal of Natural Language Processing

Make computers be able to ***Understand***
and/or ***Generate*** "Natural Language"
in order to perform useful tasks

Sample NLP applications

- Conversational System
- Spell Checking / Essay authoring
- Semantic Search / Q&A
- Information Extraction
- Social Listening
- Sentiment Analysis / Polarity Classification
- Machine Translation

Natural Language

Natural Languages / Human Languages

Thai, English, Chinese, etc.
Spoken / Written
Formal / Informal
Sign Language

Others Languages

Programming Language
Animal Communication

CHULA ΣNGINEERING

Why is NLP hard?

- Ambiguity
- Knowledge bottleneck (Real-world / Cultural / Emotional Context)

CHULA ΣNGINEERING

Why is it even harder for Thai NLP

- ตัดคำ
- การประสมคำ (คนขับรถมารอหน้าบ้าน)
- ไม่มีขอบเขตของประโยคที่ชัดเจน
  - เครื่องหมายเว้นวรรคทำได้หลายหน้าที่

From https://www.dailynews.co.th/crime/605567

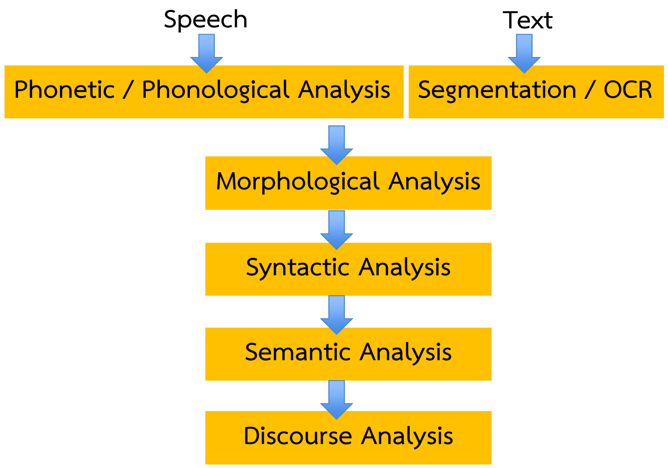
CHULA ΣNGINEERING

เมื่อวันที่ 21 ต.ค. ที่ สภ.ท่าอากาศยานสุวรรณภูมิ พล.ต.ต.คัชชา ธาตุศาสตร์ รรท.รองผบช.ทท พ.ต.อ.ชูตระกูล ยศมาดี ผกก.สภ.ท่าอากาศยานสุวรรณภูมิ พ.ต.อ.อำนาจ โฉมฉาย ผกก.3 บก.ทท.1 พ ต.ท.สุรชัช สุวรรณศรี รอง ผกก.3 บก.ทท.1 สนธิกำลังจับกุม นาย พัชร์ฐณัฐ วงศ์วังจันทร์ อายุ 31 ปี ที่อยู่ 999/3 หมู่10 ต.โคกสูง อ.เมือง จ.นครราชสีมา ในฐานความผิดเป็นผู้ประกอบธุรกิจนำเที่ยวกระทำการอันจะก่อให้เกิดความเสียหายแก่นักท่องเที่ยว, ทำหน้าที่เป็นผู้นำเที่ยวโดยไม่ได้ขึ้นทะเบียนเป็นผู้นำเที่ยว

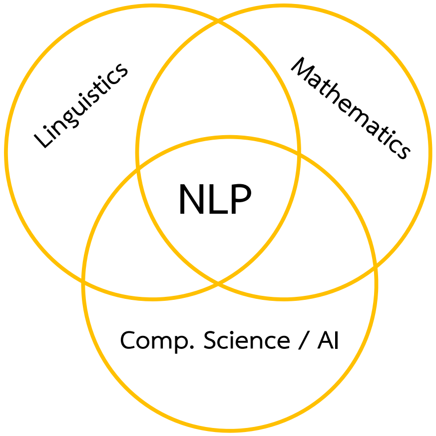
From https://www.dailynews.co.th/crime/605567

CHULA ΣNGINEERING

NLP Levels



NLP Essentials



CHULA ΣNGINEERING

What will be covered today?

- Linguistics Analysis Basics
- Understanding Artificial Intelligence Concepts
- Review of Necessary Math
- Some Basic NLP Techniques
- A Sample NLP Application

CHULA ΣNGINEERING

Linguistics Analysis Basics

Training for TrueVoice

CHULA ΣNGINEERING

Linguistic Analysis

Syntactic Analysis

Vs.

Semantic Analysis

Syntactic Structure

https://en.wikipedia.org/wiki/Word , https://en.wikipedia.org/wiki/Phrase , https://en.wikipedia.org/wiki/Sentence_(linguistics)

Character	Single independent unit
Word	The smallest meaningful unit
Phrase	A group of consecutive words (functions as a syntactic unit within a parsed tree)
Sentence	A group of words expressing complete thought.

Syntactic Structure

https://en.wikipedia.org/wiki/Word , https://en.wikipedia.org/wiki/Phrase , https://en.wikipedia.org/wiki/Sentence_(linguistics)

Character	ก, ร, ะ, โ, ด
Word	กระโดด, กำแพง, ข้าม, นักเรียน, โรงเรียน
Phrase	กำแพงโรงเรียน, กระโดดข้ามกำแพงโรงเรียน
Sentence	นักเรียนกระโดดข้ามกำแพงโรงเรียน

Morpheme

- The smallest meaningful unit in a language
- May not stand alone

Morpheme	Word
teach	teach, taught, teacher, teaching
食べる	食べる, 食べます, 食べない, 食べたい

https://en.wikipedia.org/wiki/Morpheme

Morpheme-Like Unit

เอกพล → เอก- -พล
ชาญชัย → ชาญ- -ชัย
เอกวิทย์ → เอก- -วิทย์
วรกานต์ → วร- -กานต์
วสุพล → วสุ- -พล
สมวงศ์ → สม- -วงษ์
ธงเอก → ธง- -เอก

เอก- -ชัย → เอกชัย
วร- -วงษ์ → วรวงษ์

เอก- -เอก → เอกเอก

Part of speech (POS)

= A category of words which have similar grammatical properties.

รถ	เบนซ์	เขียว	เลี้ยว	เข้า	บ้าน	ขาว
N	N	ADJ	V	V	N	ADJ

Vehicle Brand Color V V Place Color

https://en.wikipedia.org/wiki/Part_of_speech

POS: Universal POS tag

An attempt to define POS tags which are applicable for all languages.

Adjective	Coordinate Conjunction	Numeral	Subordinating Conjunction
Adposition	Determiner	Particle	Symbol
Adverb	Interjection	Pronoun	Verb
Auxiliary	Noun	Proper Noun	X

http://universaldependencies.org/u/pos/

POS: Orchid corpus

A Thai part-of-speech corpus collected by NECTEC.

POS tags in the corpus are designed specifically for Thai language.

มี	VSTA	Stative Verb
การ	FIXN	Nominal Prefix
ผลิต	VACT	Active Verb
สินค้า	NCMN	Common Noun
เหล่านี้	DDAC	Definite Determiner, Allow Classifier Between
ขึ้น	XVAE	Post-Verb Auxiliary

Sornlertlamvanich, Virach, Naoto Takahashi, and Hitoshi Isahara. "Building a Thai part-of-speech tagged corpus (ORCHID)." Journal of the Acoustical Society of Japan (E) 20.3 (1999): 189-198.

Grammars of Sentences

Simple Sentence

ฉันเดินไปตลาด

Compound Sentence

ฉันเดินไปตลาดเพราะต้องการซื้อผักสด

Conjunction

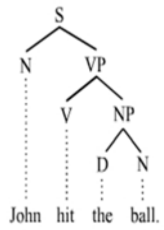
Complex Sentence

ผู้ชายที่เดินสวนกับฉันที่หน้าตลาดสวมเสื้อสีสด

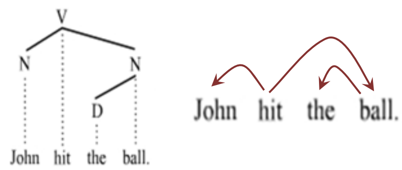
Relative Pronoun

Sentence parsing

Constituency-based parse tree



Dependency-based parse tree



Named entity

A named entity is a real-world object, such as, person, location, organization

<NE>น.พ.สุรพงษ์ สืบวงศ์ลี</NE>| |โฆษก|ประจำ|<NE>สำนักนายกรัฐมนตรี</NE>| |เปิดเผย|ว่า| |ที่|ประชุม|คณะ|รัฐมนตรี|เห็น|ชอบ| |ใน|หลักการ| |ร่าง| |<AB>พ.ร.บ.</AB>|สุวรรณภูมิมหานคร| |ตาม|ที่|<NE>กระทรวงมหาดไทย</NE>|เสนอ| |โดย|ให้|รัฐบาล|เตรียม|การ|พัฒนา|พื้นที่|และ|ก่อสร้าง|<NE>ท่าอากาศยานสุวรรณภูมิ</NE>|ใน|ท้องที่||<NE>อ.บางพลี</NE>| |<NE>จ.สมุทรปราการ</NE>| |โดย|มี|วัตถุประสงค์|ให้|เป็น|ศูนย์กลาง|การ|บิน| |การ|ขนส่ง| |การ|ประกอบ|ธุรกิจ||

Best2010 corpus provides word-segmented documents with named entity tags.

Word-Sense

One word form might have more than one meaning

เขาขอเงินฉันสิบบาท

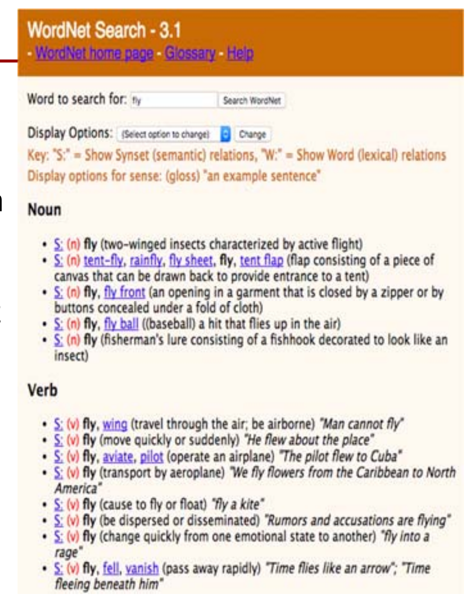
ตะขอเงินมีราคาแพง

"Word-sense disambiguation"

Wordnet

Large lexical base of English words groups into sets of synsets (synonym-sets), each expressing a distinct concept

Wordnet search: http://wordnetweb.princeton.edu/perl/webwn



CHULA ΣNGINEERING

Semantic relation: Hypernym (IS-A)

Wordnet also provides some semantic relations.

The example shows hypernyms of one synset of “dog”

```
dog, domestic dog, Canis familiaris
└ canine, canid
  └ carnivore
    └ placental, placental mammal, eutherian, eutherian mammal
      └ mammal
        └ vertebrate, craniate
          └ chordate
            └ animal, animate being, beast, brute, creature, fauna
              └ ...
```


CHULA ΣNGINEERING

Understanding
Artificial Intelligence Concepts

Training for TrueVoice

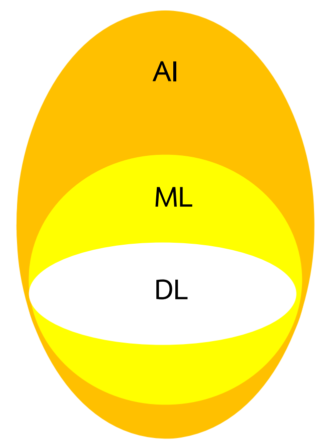
CHULA ΣNGINEERING

AI Vs. ML Vs. DL

Artificial Intelligence
Mimic human behavior

Machine Learning
Use statistical methods enabling machine to improve with experience

Deep Learning
Use multi-layer Neural Networks

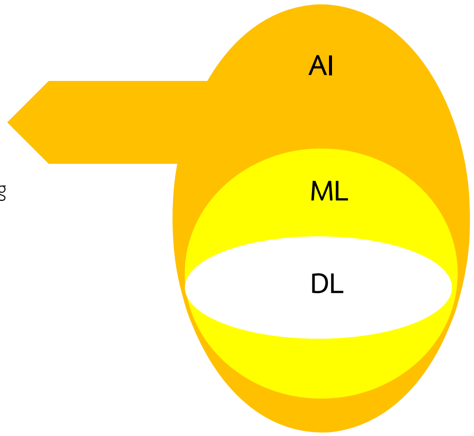


https://rapidminer.com/artificial-intelligence-machine-learning-deep-learning/

NLP with AI/ML/DL

Some applications are feasible with rule-based algorithm e.g.

- Classify อักษรต่ำ/กลาง/สูง
- Dictionary Lookup
- Regular Expression Matching

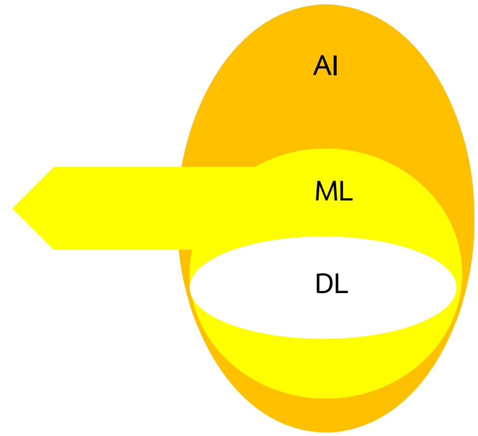


https://rapidminer.com/artificial-intelligence-machine-learning-deep-learning/

NLP with AI/ML/DL

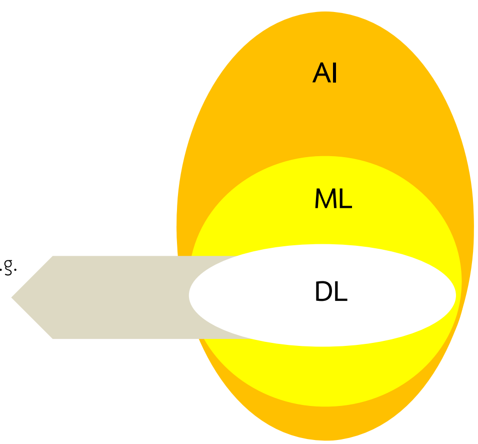
Many applications are implemented with Machine Learning technique, e.g.

- POS Tagging
- Sentiment Analysis



https://rapidminer.com/artificial-intelligence-machine-learning-deep-learning/

NLP with AI/ML/DL



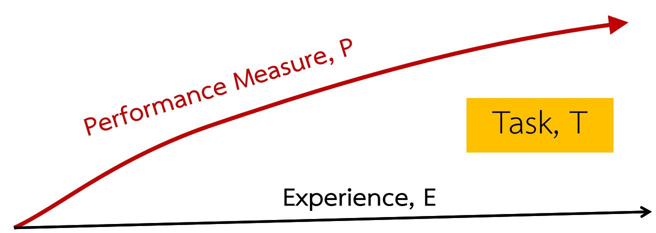
Some applications are making breakthrough improvement, e.g.

- Machine Translation
- Open Dialog Conversation

https://rapidminer.com/artificial-intelligence-machine-learning-deep-learning/

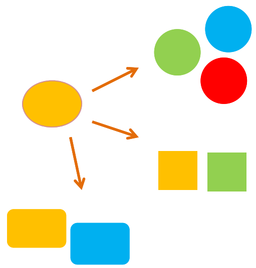
Machine learning definition

"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."



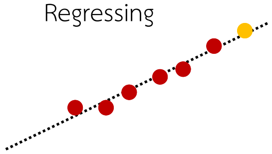
Mitchell, T. M. (1997). Machine Learning. McGraw-Hill, New York.

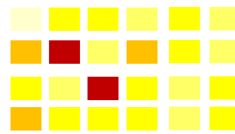
Common ML Task



Classification

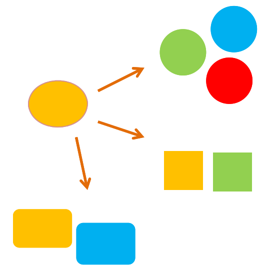
Regressing





Anomaly Detection

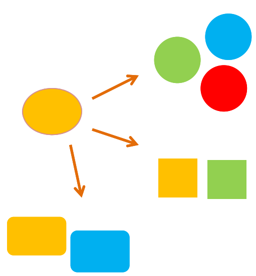
Performance Measure: Accuracy



Classification

$$\text{Accuracy} = \frac{\text{\# correctly labelled}}{\text{\# total labelled}}$$

Performance Measure: Confusion Matrix



Classification

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

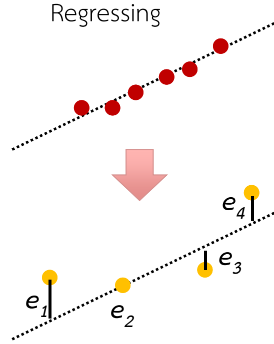
a_{ij} = # of class i labelled as class j

$$\begin{array}{cc} & \begin{array}{ccc} \text{Intent}_1 & \text{Intent}_2 & \text{Intent}_3 \end{array} \\ \begin{array}{c} \text{Intent}_1 \\ \text{Intent}_2 \\ \text{Intent}_3 \end{array} & \begin{bmatrix} 10 & 5 & 5 \\ 0 & 20 & 0 \\ 0 & 5 & 15 \end{bmatrix} \end{array}$$

Accuracy = ?

Performance Measure: **Mean Squared Error**

Regressing

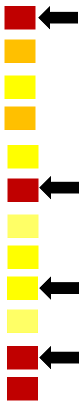


$$\text{MSE} = \frac{\text{Sum Squared Error}}{\text{\# total test point}}$$

Root Mean Squared Error

Performance Measure: **Precision / Recall**

Anomaly Detection

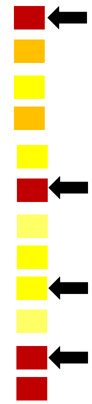


$$\text{Precision} = \frac{\#\ \text{correctly detected}}{\#\ \text{total detected}}$$

$$\text{Recall} = \frac{\#\ \text{correctly detected}}{\#\ \text{total anomaly}}$$

Performance Measure: **F-measure**

Anomaly Detection



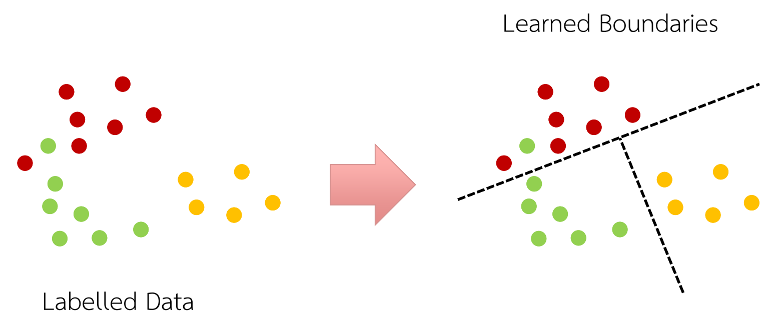
$$F_1\ \text{Score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Experience, E

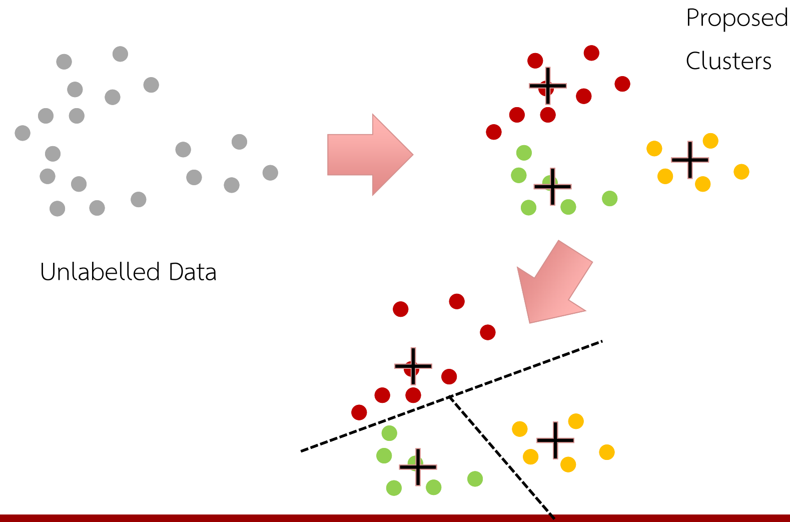
*Experience is the main key to enable **learning**

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

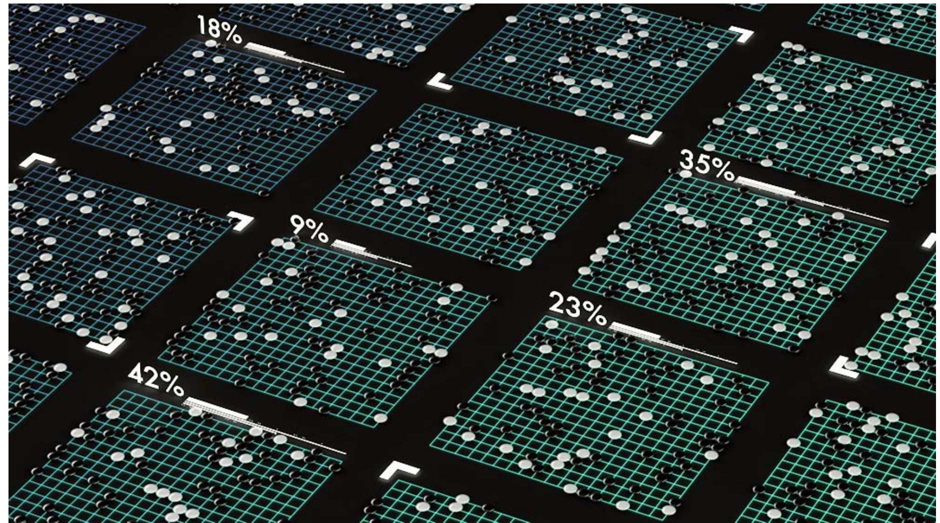
Experience, E: **Supervised Learning**



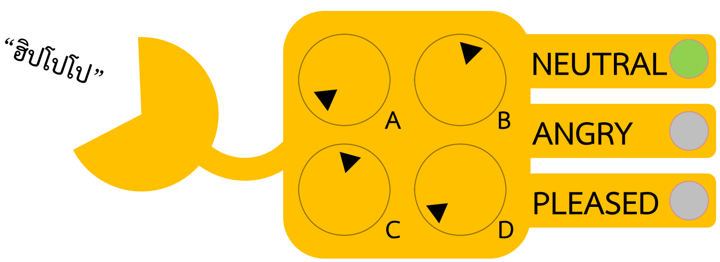
Experience, E: **Unsupervised Learning**



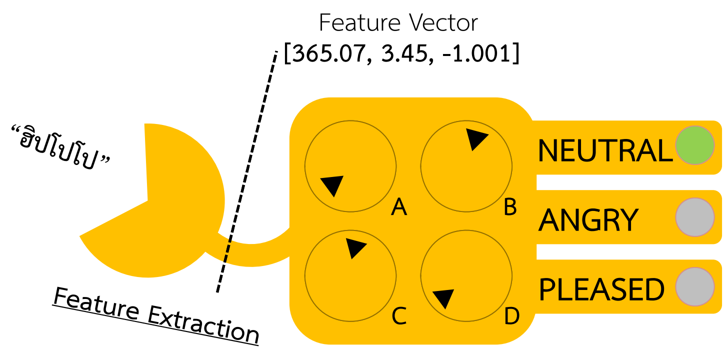
Experience, E: **Reinforcement Learning**



Features-Models-Training

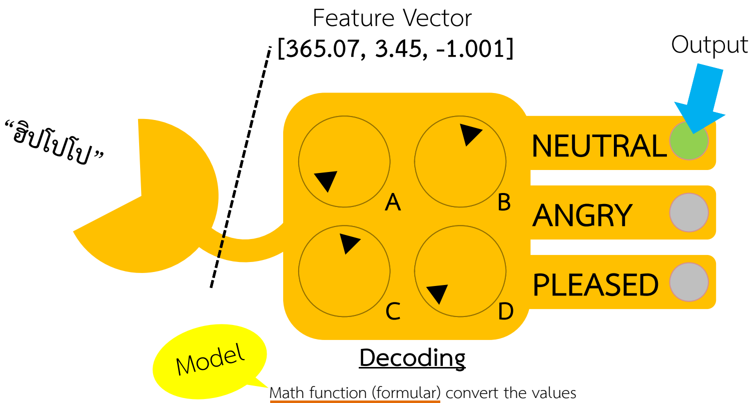


Features-Models-Training



Convert “Input Data” to “Numbers”
Usually, numbers are arranged as “Vectors”
Intentional lose some “non-essential” info

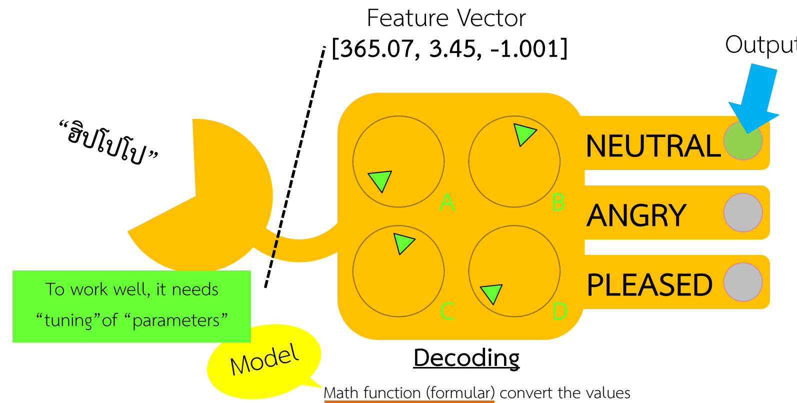
Features-Models-Training



Decoding

Math function (formular) convert the values of input features to “Output”

Features-Models-Training



Decoding

Math function (formular) convert the values of input features to “Output”

Features-Models-Training

“ฮิปโปโป”
Feature Vector
[365.07, 3.45, -1.001]

Model

Score(x,Nuetral) = **A** x 365.07 + **B** x 3.45 + **C** x -1.001

Score(x,Angry) = **B** x 365.07 + **C** x 3.45 + **D** x -1.001

Score(x,Pleased) = **C** x 365.07 + **D** x 3.45 - 23

Whichever score is the highest wins

Features-Models-Training

“ฮิปโปโป”
Feature Vector
[365.07, 3.45, -1.001]

Model

Score(x,Nuetral) = **A** x 365.07 + **B** x 3.45 + **C** x -1.001

Score(x,Angry) = **B** x 365.07 + **C** x 3.45 + **D** x -1.001

Score(x,Pleased) = **C** x 365.07 + **D** x 3.45 - 23

Whichever score is the highest wins

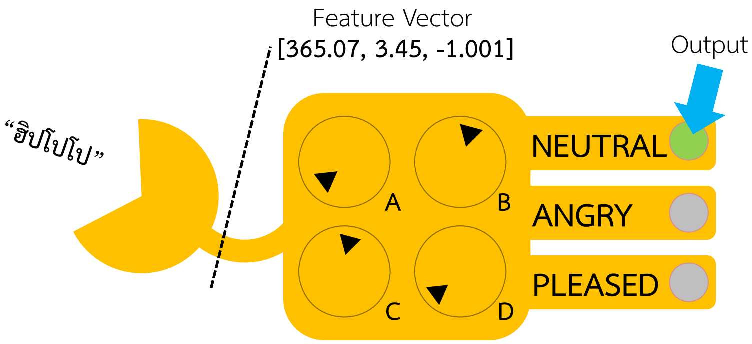
Model Training

Find the value of parameters (A, B, C, D) that optimize an “Objective” of that task by looking at some given labelled values.

Training Data

CHULA ΣNGINEERING

Features-Models-Training



Decoding

Math function (formular) convert the values of input features to “Output”

CHULA ΣNGINEERING

Feature Extraction

- Some info can be used as is. Eg: Number of letters in a word.
- Not all “Numeric Strings” are numbers Eg: zipcode
- Non-numeric needs to be converted to numbers (usually organized in discrete math structure such as vectors, matrices)
- Keep the “essentials”, discard the “non-essentials”

CHULA ΣNGINEERING

Feature Extraction: Represent words

One-hot Encoding

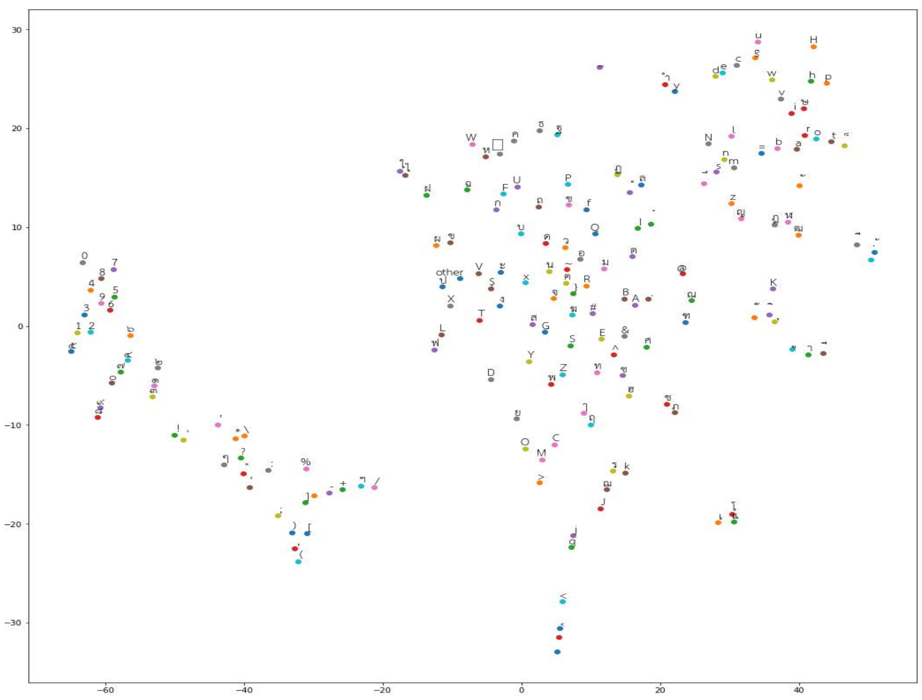
word	one-hot vector
Apple	[1, 0, 0, 0, 0, 0, 0]
Banana	[0, 1, 0, 0, 0, 0, 0]
Coconut	[0, 0, 1, 0, 0, 0, 0]
:	:
Tangerine	[0, 0, 0, 0, 0, 0, 1]

CHULA ΣNGINEERING

Feature Extraction: Represent words

Word Embedding

word	Real-number vector
Apple	[10.45, 8.75, 0.11]
Banana	[10.32, 0.13, 4.32]
Red	[0.281, 9.55, 10.32]
:	:
Tangerine	[10.33, 5.43, 8.77]



Feature Extraction: Represent documents

Bag-of-word

$$
\begin{array}{cccccccc}
w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & \dots & w_7 \\
\end{array}
$$

$$
\begin{bmatrix} 5 & 3 & 4 & 0 & 3 & 5 & \dots & 1 \end{bmatrix}
$$

\# of w_i appearing in the document

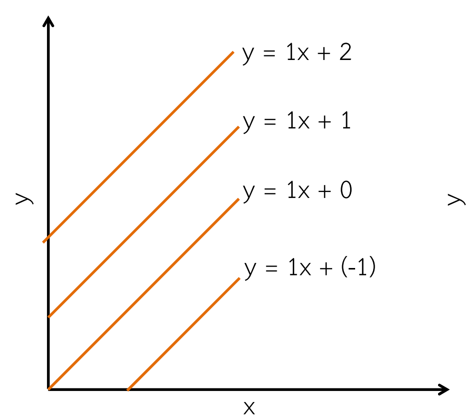
Feature Extraction: Represent documents

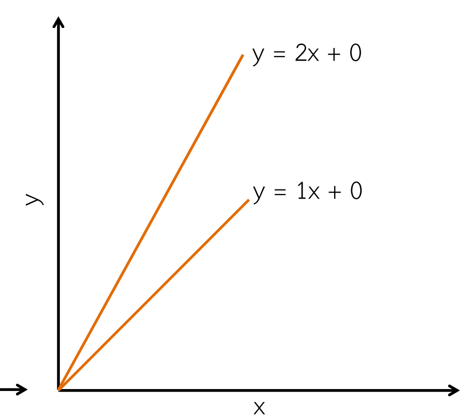
Document Embedding

word	Real-number vector
d_1	[10.45, 8.75, 0.11]
d_2	[10.32, 0.13, 4.32]
d_3	[0.281, 9.55, 10.32]
:	:
d_N	[10.33, 5.43, 8.77]

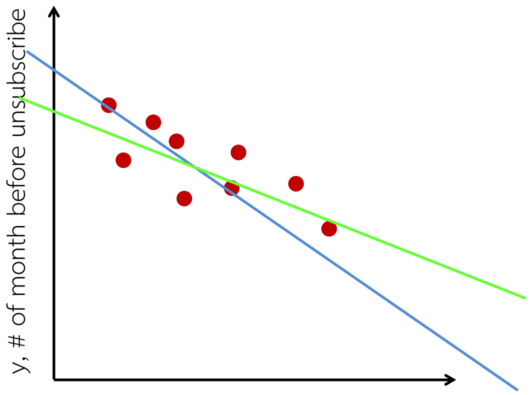
Model Training: Linear Regression Example

$$y = Mx + C$$





Model Training: Linear Regression Example



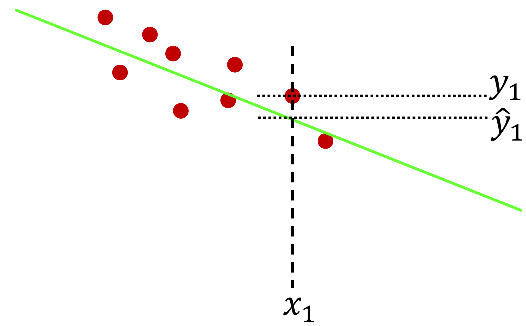
$$y = Mx + C$$

x, # of dropped calls experienced

in the last 2 months

Model Training: Linear Regression Example

$$e^2 = (\hat{y}_1 - y_1)^2$$



Reviews of Math Knowledge

Training for TrueVoice

Logarithm

By definition

$$\log_b(a) = c \iff b^c = a$$

when b is the base,
a is the power,
c is the exponent

Example:

Logarithmic form		Exponential form
$\log_2(8) = 3$	$\iff$	$2^3 = 8$
$\log_3(81) = 4$	$\iff$	$3^4 = 81$
$\log_5(25) = 2$	$\iff$	$5^2 = 25$

Logarithm properties

The product rule	$\log_b(MN) = \log_b(M) + \log_b(N)$
The quotient rule	$\log_b\left(\frac{M}{N}\right) = \log_b(M) - \log_b(N)$
The power rule	$\log_b(M^p) = p\log_b(M)$

Prevent “Underflow” when multiply many small numbers (0<n<1) together

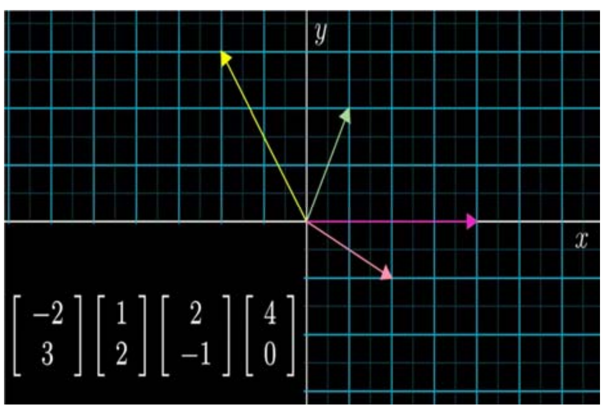
Vector

An array of numbers

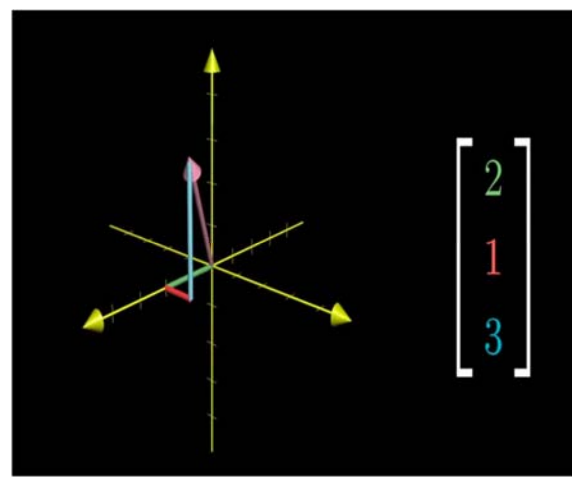
A D-dimensional vector represents an arc in d dimensions

– starts at the origin and ends at the point specified in a vector

2D Vector



3D Vector



Dot product

Dot product of two vectors **a** = [a_1, a_2, …, a_n] and **b** = [b_1, b_2, …, b_n] is a scalar defined as:

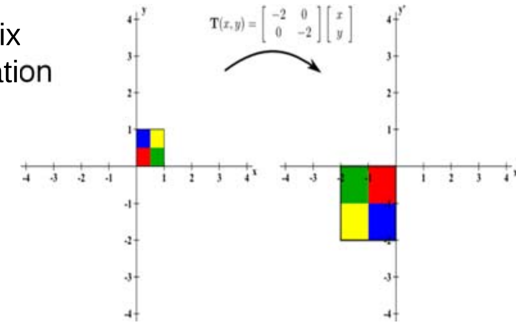
Example: [1, 3, -5] · [4, -2, -1] = 3

$$\mathbf{a}\cdot\mathbf{b}=\sum_{i=1}^{n}a_ib_i=a_1b_1+a_2b_2+\cdots+a_nb_n$$

Matrix

2D array of numbers

We may thought of matrix as a linear transformation



Matrix: Column vector & Row vector

$$\mathbf{M}=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{pmatrix}$$

Row vector

Column vector

Matrix operation: Add/Subtract

Two matrices must be the same size

$$a=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{pmatrix}\quad b=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{pmatrix}\quad a+b=\begin{pmatrix}2 & 4 & 6\\ 8 & 10 & 12\\ 14 & 16 & 18\end{pmatrix}$$

Matrix operation: Multiply by a scalar

Multiply by a scalar

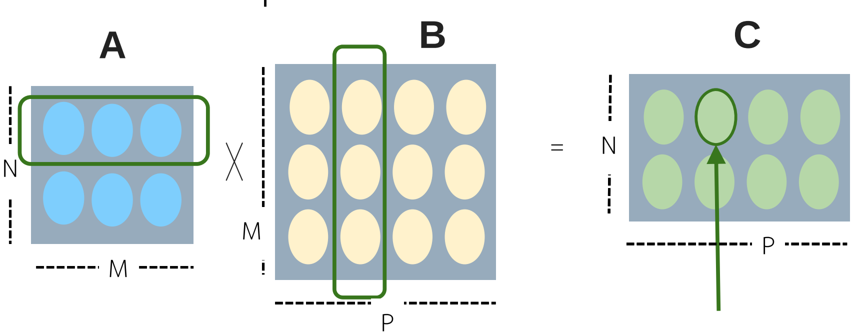
$$\mathbf{a} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad \mathbf{5a} = \begin{pmatrix} 5 & 10 & 15 \\ 20 & 25 & 30 \\ 35 & 40 & 45 \end{pmatrix}$$

Matrix operation: Transpose

If M is a mχn matrix, M^T, a transpose of M,will be nχm matrix.

$$\mathbf{M} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad \mathbf{M}^T = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

Matrix multiplication



Value of $C_{i,j}$ equals to **dot product** of i^{th} row of A and j^{th} column of B

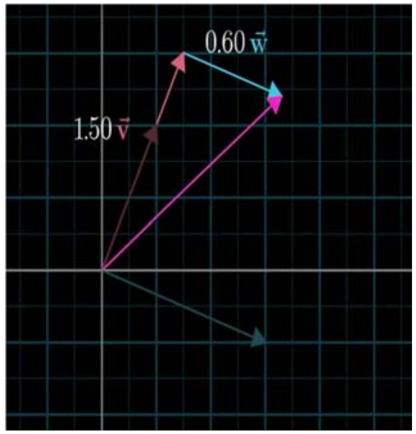
Linear combination

Given a set of vectors $\mathbf{v_1}$, $\mathbf{v_2}$, …, $\mathbf{v_n}$ and scalars a_1, a_2, …, a_n,
the linear combination of those vectors with those scalars is:

$$a_1\mathbf{v_1} + a_2\mathbf{v_2} + \ldots + a_n\mathbf{v_n}$$

Linear combination

For example:
v = [1, 2]
w = [3, -1]
The pink vector is a result of the linear combination:
1.5**v** + 0.6**w** = [3.3, 2,4]



Linear combination Example

$$Admission\ Score = W_1 \cdot Math + W_2 \cdot Sci + W_3 \cdot Eng$$

Exam Score
of each candidate

Linear combination Example

$$Admission\ Score = W_1 \cdot Math + W_2 \cdot Sci + W_3 \cdot Eng$$

Exam Score of all candidate

$$\begin{bmatrix} 80 \\ 79 \\ 63 \\ 48 \\ 21 \\ 98 \\ \vdots \\ 36 \end{bmatrix} \quad \begin{bmatrix} 80 \\ 70 \\ 24 \\ 68 \\ 31 \\ 84 \\ \vdots \\ 55 \end{bmatrix} \quad \begin{bmatrix} 24 \\ 60 \\ 93 \\ 23 \\ 80 \\ 80 \\ \vdots \\ 87 \end{bmatrix}$$

Linear combination Example

$$Admission\ Score = W_1 \cdot Math + W_2 \cdot Sci + W_3 \cdot Eng$$

$$\begin{bmatrix} 80 & 80 & 24 \\ 79 & 70 & 60 \\ 63 & 24 & 93 \\ 48 & 68 & 23 \\ 21 & 31 & 80 \\ 98 & 84 & 80 \\ \vdots & \vdots & \vdots \\ 36 & 55 & 87 \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix}$$

Linear combination Example

$$Job_i = W_{1i} \cdot Math + W_{2i} \cdot Sci + W_{3i} \cdot Eng$$

Reviews of Probability Concepts

Training for TrueVoice

Important Probability Concepts

- Probability
- Conditional Probability
- Total probability theorem
- Bayes' Rule
- Random variable (R.V.)
  - discrete R.V.
  - continuous R.V.
- Expected Value and Variance
- Gaussian Random Variable
- Joint PDF

Probability

- A : any event
- P(A) : probability that the event A happens
- $0 \leq P(A) \leq 1$

Conditional Probability

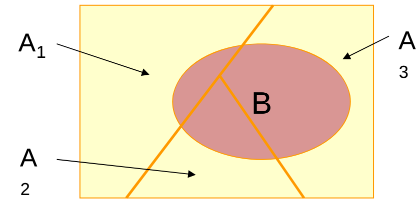
- P(A|B) = probability of A, given that B has occurred.

$$\boxed{P(A\mid B)=\frac{P(A\cap B)}{P(B)}}$$

- Adjusting the universe to B

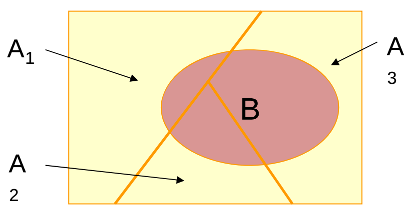
Total Probability Theorem

- Divide the universe into smaller partitions



$$\begin{aligned}P(B) = \ & P(B|A_1)P(A_1) + \\ & P(B|A_2)P(A_2) + \\ & P(B|A_3)P(A_3)\end{aligned}$$

Bayes' Rule



$$\boxed{P(A_i\mid B)=\frac{P(B\mid A_i)P(A_i)}{P(B)}=\frac{P(B\mid A_i)P(A_i)}{\sum_j P(A_j)P(B\mid A_j)}}$$

$P(A_i)$: "Prior" probability

Random Variable

- X : Random variable (R.V.)
- x : experimental value of the R.V. X

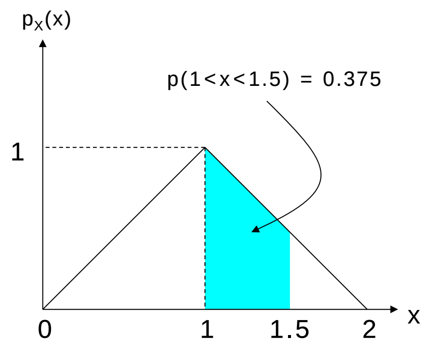
X	x	
Type of a vowel V	{/a/,/i/,...}	Discrete
# of syllable in a word W	{1,2,3,4,5,...}	
Time (s.) used to respond to a message	(0, ∞)	Continuous
Probability of event A	[0,1]	
Log Prob of event A	(-∞,0]	

Probability Density Function (PDF)

$$P(a < x \le b) = \int_a^b p(x)dx$$

$$P(-\infty < x \le \infty) = \int_{-\infty}^{\infty} p(x)dx = 1$$

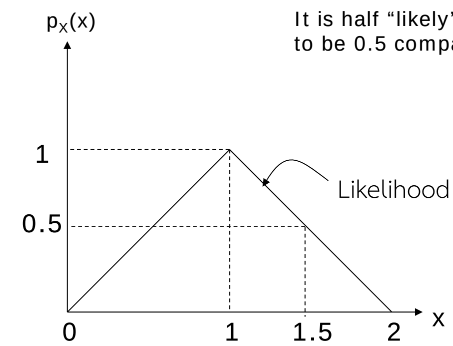
$$P(x = a) = 0$$



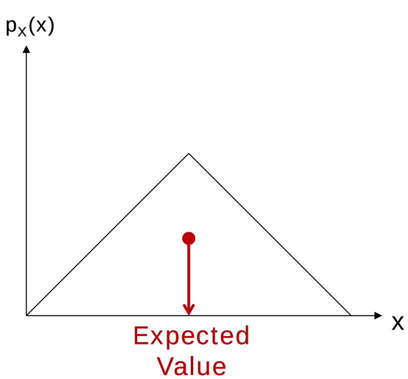
PDF Interpretation

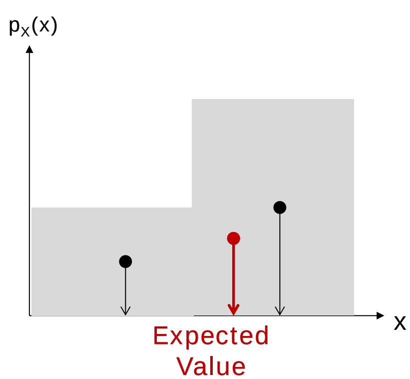
P(x = 1) =?

It is half "likely" for x
to be 0.5 compared to x being 1

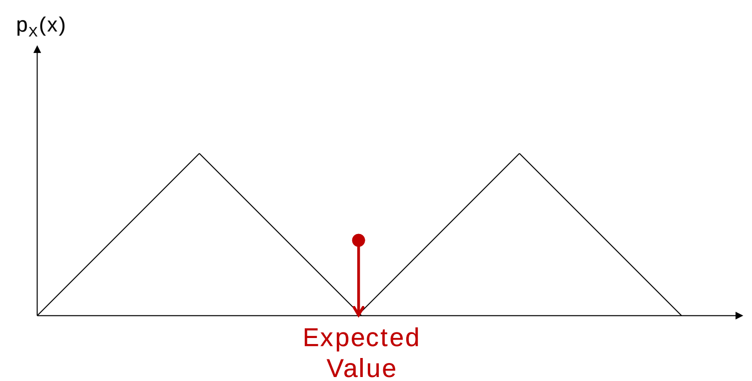


Expected Value

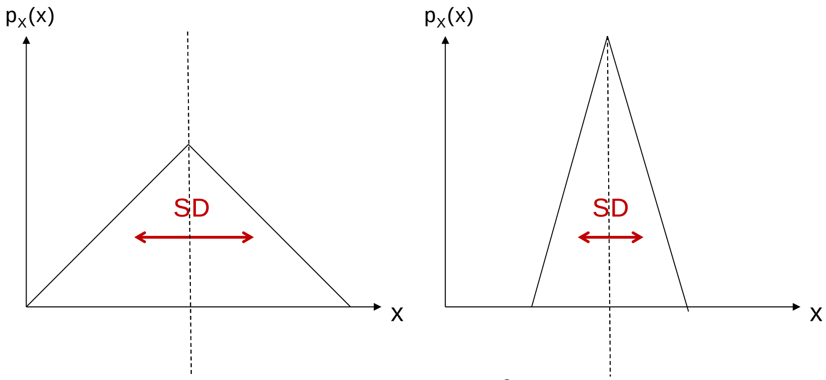




Expected Value

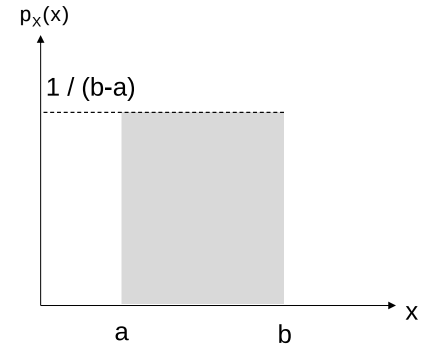


Standard Deviation

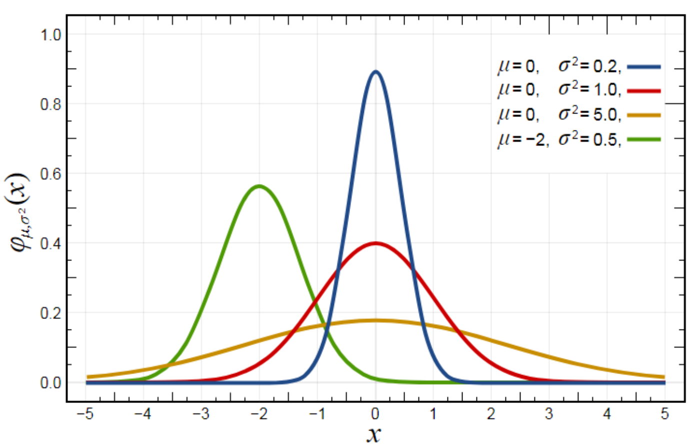


Variance = SD^2

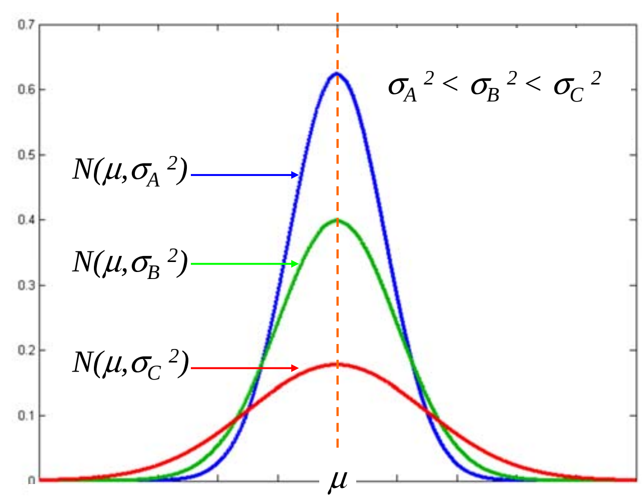
Uniform Distribution



Gaussian (Normal) Distribution



Gaussian (Normal) Distribution

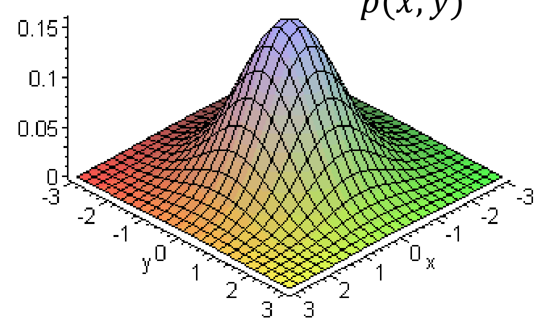


Joint PDF

- Since we hardly use 1-dimensional feature vector, we mostly use joint PDFs to model the value distributions of multiple random variables.

CHULA ΣNGINEERING

Joint PDF



http://personal.kenyon.edu/hartlaub/MellonProject/Bivariate2.html

Faculty of Engineering, Chulalongkorn University
www.eng.chula.ac.th

CHULA ΣNGINEERING

Probability Chain Rule

$p(x_1, x_2, x_3, \dots, x_N)$

$= p(x_1)p(x_2, x_3, \dots, x_N|x_1)$

$= p(x_1)p(x_2|x_1)p(x_3, x_4, \dots, x_N|x_2)$

:

$=p(x_1)p(x_2|x_1)p(x_3|x_2,x_1)\dots p(x_N|x_{N-1},\dots x_2,x_1)$

Faculty of Engineering, Chulalongkorn University
www.eng.chula.ac.th

CHULA ΣNGINEERING

Independence

$p(x_2|x_1) = p(x_2)$

$p(x_1, x_2, x_3, \dots, x_N)$

$=p(x_1)p(x_2|x_1)p(x_3|x_2,x_1)\dots p(x_N|x_{N-1},\dots x_2,x_1)$

$=p(x_1)\ p(x_2)\ p(x_3)\dots\ p(x_N)$

Faculty of Engineering, Chulalongkorn University
www.eng.chula.ac.th

Some Basic NLP Techniques

Training for TrueVoice

Some Basic NLP Techniques

- TF-IDF
- N-Gram

What's TF-IDF for?

- TF-IDF stands for *Term Frequency-Inverse Domain Frequency*
- A weight given to a **word** – measures how important a word is to a **document** in a collection

TF: Term Frequency

Measure how frequently a word occurs in a document

$$TF(w) = \frac{\text{Number of times word } w \text{ appears in a document}}{\text{Number of words in the document}}$$

Why just TF is not enough?

Some words, such as,
"the", "a", "as", "is",
occur very frequently.

Are those words important to the document?

IDF: Inverse Domain Frequency

Measure how rare a word appears in each document in a collection

$$IDF(w) = \log \frac{\text{Total number of documents}}{\text{Number of documents with word } w \text{ in it}}$$

Use natural log to decrease the effect of IDF

IDF: Inverse Domain Frequency

Given a collection with 1,000,000 documents

Word	Number of documents this word appears	IDF
the	1,000,000	0
good	200,000	1.61
NLP	500	7.60

TF-IDF

Multiplying TF and IDF together to get a weight

A word which appears a lot in this document and doesn't appear in other documents much should be an important keyword.

LM: Language Modeling

A task that tell how likely a sequence of words will make a meaningful phrase.

Formally: Given a sequence of words of length m, this task is to assign joint probability $P(w_1, w_2, \ldots, w_m)$

LM Example

Example:

P(ฉัน,ไป,ซื้อ,ของ,ที่,ตลาด) = 0.7
P(ตลาด,ไป,ของ,ที่,ฉัน,ซื้อ) = 0.05

We can also find the conditional probability $P(w_m \mid w_1, w_2, \ldots, w_{m-1})$ for the language modeling task.

LM for long sentences

Try finding the probability

$$P(\text{the} \mid \text{its water is so transparent that}) = \frac{Count(\text{its water is so transparent that the})}{Count(\text{its water is so transparent that})}$$

This phrase is much likely to happen, but it's too long so that it may not occur in the corpus.

Make an assumption

Markov assumption: the probability of the next word depends on only previous k words

For example:
k = 1:
k = 2:

$$P(\text{the} \mid \text{its water is so transparent that}) \approx P(\text{the} \mid \text{that})$$

$$P(\text{the} \mid \text{its water is so transparent that}) \approx P(\text{the} \mid \text{transparent that})$$

LM with Markov assumption

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i \mid w_{i-k} \ldots w_{i-1})$$

For example,
we assume dependency on previous 2 words
P(ฉัน,ไป,ซื้อ,ของ,ที่,ตลาด)
= P(ฉัน|<s>,<s>)P(ไป|ฉัน, <s>)P(ซื้อ|ฉัน,ไป) P(ของ|ไป,ซื้อ)P(ที่|ซื้อ,ของ)P(ตลาด|ของ,ที่)

N-gram models

A model for a sequence of contiguous n items (words).

Unigram model (n=1)

P(I, want, a, hamburger) = P(I)P(want)P(a)P(hamburger)

$$P(\text{hamburger}) = \frac{\text{Number of time "hamburger" appears in the corpus}}{\text{Total number of words in the corpus}}$$

Bigram model (n=2)

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

Quick quiz

we are the most social species on earth and we are also the most violent species on earth we have two faces because these two faces are important to survival

Find:
P(we), P(are|we), P(have|we), P(important|most)

Trigrams, 4-gram, 5-gram, ...

Just like a bigram model, but scale up k to 3, 4, 5, ...

The problem of unseen data

From the quiz, we can see that:
P(important | most) is zero because of the limited amount of data

How can we deal with this problem?

Out-Of-Vocabulary (OOV) words

Create an **unknown** token: <UNK>
To train <UNK> probabilities:

1. Create a set of training words L of size V
2. At training time, if any training word is not in L, we treat it as <UNK> and train its probabilities like a normal word
3. At decoding time, use <UNK> probabilities for any word not in L

OOV words: Example

Corpus:
we are the most social species on earth and we are also the most violent species on earth

Training word L = {we, are, the, most, social, on, earth, also}

OOV words: Example

Training word L = {we, are, the, most, social, on, earth, also}

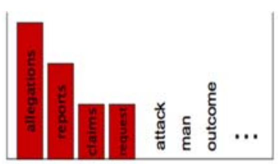
Training time:
we are the most social <UNK> on earth <UNK> we are also the most <UNK> <UNK> on earth

P(<UNK>) = 4/18, P(on | <UNK>) = 2/18

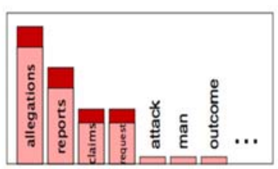
Smoothing

Add small probabilities of occurring to unseen data

- When we have sparse statistics:
  - P(w | denied the)
  - 3 allegations
  - 2 reports
  - 1 claims
  - 1 request
  - 7 total



- Steal probability mass to generalize better
  - P(w | denied the)
  - 2.5 allegations
  - 1.5 reports
  - 0.5 claims
  - 0.5 request
  - 2 other
  - 7 total



Add-one smoothing (Laplace smoothing)

For all words w_i :

$$P_{Add-1}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i) + 1}{c(w_{i-1}) + V}$$

When V is the number of vocabulary in the corpus.

Add-one doesn't work quite well

A lot of nonsense bigrams pair got promoted.

	i	want	to	eat	chinese	food	lunch	spend
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.00025	0.00075
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025	0.00084
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018	0.055
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02	0.00046
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012	0.00062
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.00039	0.00039
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.00056	0.00056
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.00058	0.00058

Backoff

If you intend to use trigram, but you don't see that 3 words together in the corpus,
use bigram...
- if still don't see the bigram
use unigram…
- if that word hasn't appeared in the corpus use 1/V when V is the vocabulary size

Kneser-Ney Smoothing

A primarily used smoothing method for calculating probability distribution of n-grams.

Study more:
http://www.foldl.me/2014/kneser-ney-smoothing/

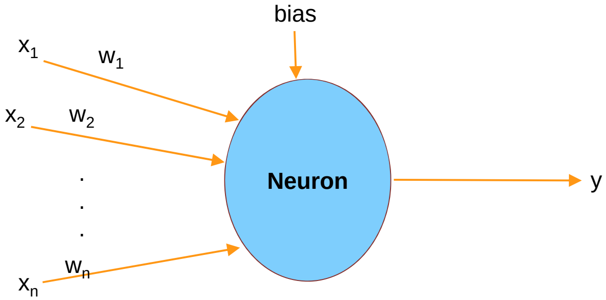
Sample NLP Applications

POS Tagging using NN

Training for TrueVoice

(Artificial) Neuron

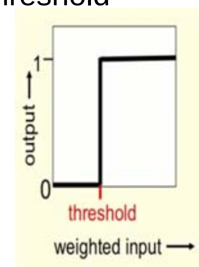
Given n input associated with weight, a bias, neuron outputs a real number y. Output value depends on type of neuron.



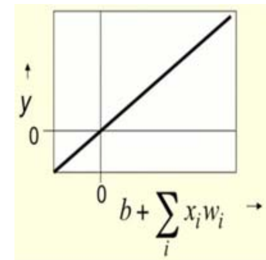
Some types of neurons

Given $z = b + \sum_{i=1}^{n} w_i x_i$

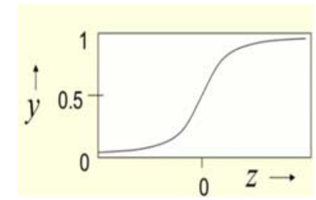
1. Binary threshold



2. Linear

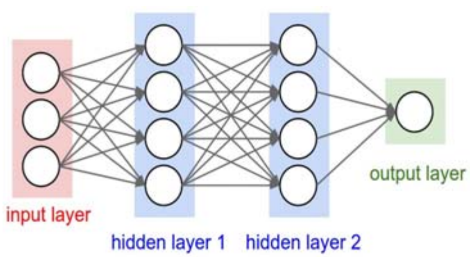


3. Sigmoid



Neural networks

A network of neurons, one input layer, one output layer, and zero to many hidden layers



POS Tagging: A toy example

Assume we have only **5 words** in our model
{<s>, </s>, brown, fox, jumps}
3 POS tags
{noun, verb, adjective}

Represent an input

Use one-hot vector, assuming there are only 5 words in this task: {<s>, brown, fox, jumps, </s>}

Simple Neural Networks

Have 192 input nodes, 15 output nodes (each node is associated to one POS)
So we have parameters = (192*15) weight + 15 bias

Simple Neural Networks

Have 192 input nodes, 15 output nodes (each node is associated to one POS)
So we have parameters = (192*15) weight + 15 bias

That's why we use matrix multiplication instead of loop over every features

Softmax function

We get a vector size of 15 (output size)
, but value in a vector can be any real number.
So we use softmax function applied element-wise to a vector to get a probability distribution.

Get the most probable POS tag

Choose POS associated to the dimension that has the highest probability.