Using Deep Neural Networks in translating languages

Aung Phyo

12/08/2020

www.blog.busuu.com/most-spoken-languages-in-the-world

- 6500+ languages
- English 1,132 million
- Mandarin 1,117 million
- Hindi 615 million
- Spanish 534 million
- French 280 million

---

Simple Neural Network

Input Layer – initial data for the neural network.

Hidden Layer – intermediate layer between input and output layer and place where all the computation is done.

Output Layer – produce the result for given inputs.
Recurrent Neural Network or Deep Neural Network

Deep Neural Network has multiple hidden layer between the input layer and the output layer

Recurrent Neural Network is a type of neural network that has an internal loop
Deep Neural Network

input layer  hidden layer 1  hidden layer 2  hidden layer 3  output layer

Types of RNN

one to one  one to many  many to one  many to many  many to many
Many to Many RNN is used in solving of sequence to sequence problem.

For example, language translation: RNN reads the input sequence of a sentence in English and then outputs a sentence in French as the output sequence.

RNN is good to apply on the problems whose inputs and targets ranges are in short.

Long Short Term Memory Networks (LSTM)

LSTMs are a special kind of Recurrent Neural Networks, capable of learning long-term dependencies.

LSTM learns to map based on variable length of an input sentence into a fixed dimensional vector representation.
In translation process, the source sentences had been paraphrased and the LSTM looks for the sentence representations that capture their meaning.

If the sentences meaning are similar, they are closed to each other. If the sentences meaning are different, they are far from each other’s.

Let’s say input sequence is a, b, c and the output sequence d, e, f which is the translation of a, b, c

Instead of mapping the sentence a, b, c to the sentence d, e, f the LSTM is asked to map c, b, a to d, e, f. The d, e, f is the translation of a, b, c so a is closed to d, b is relatively closed to e and so on.
According to the research paper, Sequence to Sequence Learning with Neural Networks

Training the LSTM Recurrent Neural Network by reversing the input sequence got the better accuracy result than without reversing the input sequence.

Human languages are built with different patterns and complex structures

Therefore, the sequence of a language sentence will be variable and complex, so the LSTM Recurrent Neural Networks is good to use to solve the languages translation problem.

Thank You
Convolutional Neural Networks Focusing on Fire Image Detection

Tuba Senbabaoglu
Fall 2020

What are the difficulties detect a fire?

Smoke

Stationary Camera
Some Fire Detection Models

• Pixel Level

• Blob Level

• Patch Level

Convolutional Neural Network
Convolutional Neural Network

Kunihiko Fukushima  
Yann LeCun

What is Convolutional Neural Network?

A CNN sequence to classify handwritten digits
Deep Convolutional Neural Networks for Fire Detection

Deep Convolutional Neural Networks for Fire Detection

<table>
<thead>
<tr>
<th>Classifier</th>
<th>SVM-Raw</th>
<th>CNN-Raw</th>
<th>NN-Pool5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time per image (s)</td>
<td>0.16</td>
<td>2.1</td>
<td>1.4</td>
</tr>
</tbody>
</table>
Convolutional Neural Networks Based Fire Detection in Surveillance Videos

Probability scores and predicted labels produced by the proposed deep CNN framework for different images from benchmark datasets.
Image fire detection algorithms based on convolutional neural networks

- Faster-RCNN
- R-FCN
- SSD
- YOLO v3

Image fire detection algorithms based on convolutional neural networks
Alternative Splicing

>90% of genes are alternatively spliced!

Can we predict the ‘Splice Code’ given sequence and cellular conditions?

Deep Learning


1400-11,009 Exons

262-1393 RNA features & cellular conditions

0-5 Tissue Types

Mouse & Human datasets

Xiong et al., Bioinformatics, 2011
Results: DNN Show Improvement in Predictions

Xiong et al., Bioinformatics, 2011; Leung et al., Bioinformatics, 2014; Jha et al., Bioinformatics, 2017

BNN vs Others

BNN vs DNN

Dropout

Minibatch Gradient Descent

Millions of Parameters

Deep Learning in a ‘Black Box’

Leung et al, Bioinformatics, 2014
Deep Learning in a ‘Black Box’

Jha et al, Bioinformatics; https://dribbble.com/shots/2583480-Laboratory

Deep Learning in a ‘Black Box’

DeepLIFT

Calculate contribution scores by comparing activation of each neuron to a reference activation

Sharing Work = Better Science!
- Different datasets
- Reconstructing models

Deep Learning and Molecular Biology both need LOTS of data

‘Black Box’ is a PROBLEM
Semi supervised learning & Active learning - optimizing learning approaches

By - Aditya Tamhankar
SID # 920758771
CSC 872 - PAMI
Prof. Okada

Background & Problem Addressed
Semi supervised Learning

“We expect unsupervised learning to become far more important in the longer term. Human and animal learning is largely unsupervised: we discover the structure of the world by observing it, not by being told the name of every object.” LeCun, Bengio, Hinton, Nature (2015)

Active Learning

Model “Learning Curve”

- active learning iteration (smart selection)
- Supervised learning iteration (random selection)

Amount of labeled data

Model Accuracy
Pseudo Labelling for Semi-supervised Learning

Active Learning with Information Entropy

Uncertainty associated with random variable

- Strategy 1: To determine whether a document in unlabeled examples is suitable for this classification.
- Strategy 2: Which documents in unlabeled examples could be used as the new training documents?
Tri-training Cost Effective Active Learning Algorithm (Tri-CEAL)

Fig. 1. Tri-CEAL algorithm flow chart.

Is Tri-CEAL good?
Human Level Concept learning

A
i) primitives
ii) sub-parts
iii) parts
iv) object template
relation: attached along
type level
relation: attached along
token level
relation: attached at start
v) exemplars
vi) raw data

B
Human drawings

Human parses

Machine parses

Human or Machine?

Ans: \([1, 2, 1, 2, 1]\)
Results & Applications

Self-supervised Learning

Conclusion

THANK YOU! 😊
e.g. Japan is an island country in East Asia located in the northwest Pacific Ocean.
NER Evaluation Metrics

**F1 Score**

\[
\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

- **True Positive**
- **False Positive**
- **False Negative**
- **True Negative**

Data Sets for NER

**Evaluation**

CoNLL-2003: Named Entity Recognition

**Training**

Wikipedia, Gazetteer, Lexicon
Progress in NER

Florian et al. 2003. Named Entity Recognition through Classifier Combination


Collobert et al. 2016. Natural Language Processing (Almost) from Scratch

Chiu et al. 2016. Named Entity Recognition with Bidirectional LSTM-CNNs

Lample et al. 2016. Neural Architectures for Named Entity Recognition

Florian et al. 2003. Named Entity Recognition through Classifier Combination

Methods
Robust Risk Minimization Classifier,
Maximum Entropy Classifier
Transformation-Based Learning Classifier
Hidden Markov Model Classifier

RRM   TBL   HMM   MaxEnt

Voting (Equal / Weighted)

Combo F1 Score
**Collobert et al 2016**
Natural Language Processing (Almost) from Scratch

**Methods**
- CNN
  - Feed-Forward Neural Network

---

**Chiu et al 2016.**
Named Entity Recognition with Bidirectional LSTM-CNNs

**Methods**
- CNN,
  - Bidirectional LSTM
**Results**

Florian et al 2003. Named Entity Recognition through Classifier Combination

Collobert et al 2016 Natural Language Processing (Almost) from Scratch


**F1 Score:** 88.76

**F1 Score:** 89.59

**F1 Score:** 91.62

**Discussion**

Florian et al 2003. Named Entity Recognition through Classifier Combination

Collobert et al 2016 Natural Language Processing (Almost) from Scratch


- Highest Score among Machine Learning Models
- Task Specific Model
- Feature Engineering Required

- First Practical Model for All NLP Tasks
- Feature Inferred Model
- Feature Engineering Partly Required
- External Data Required for High Score

- Feature Inferred Model
- No Language Specific
- No External Data Required
- Possibility to Hybrid Model
Key Aspects for Further Improvement

- **Word Embeddings**
- **Character Embeddings**
- **Affix Embeddings**

New Deep Learning Technology & Architecture (e.g., Transformer)

---

**BERT**
(Bidirectional Encoder Representations from Transformers)

- **Pre-Trained** Model
  with entire Wikipedia (2.5 billion words!)

- **Outperformed** other existing NLP models

- **Applicable** to wide variety of NLP tasks and many other languages

**Will it be a game changer in NLP?**
Let’s keep an eye on the future of NLP!
Transfer Learning
By Shraddha Upadhyay

Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem.

A survey of transfer learning

- Homogeneous Transfer Learning: instance based, feature-based (both asymmetric and symmetric), parameter-based, and relational-based
- One-stage and two-stage process
- Heterogeneous Transfer Learning: Image recognition, Multilanguage text classification, single language text classification, drug efficacy classification, human activity classification, software defect classification
A survey of transfer learning

- Symmetric feature-based transfer learning: create common latent feature space for both domains
- Asymmetric feature-based transfer learning: transforms source feature space to the target feature space
- Negative Transfer: a source domain having a negative impact on a target learner

Discussion: Potential area of future research

- Improvements to heterogeneous transfer learning: Quantify the amount of knowledge with co-occurrence matrix, maybe unrealistic
- Improved methods for correcting the conditional distribution when using pseudo labels because of lack of target data
- Quantify any performance gains for simultaneously solving both distribution differences along with data preprocessing steps using heuristic knowledge
- The diversity in data collection makes heterogeneous transfer learning solutions more important moving forward
Transfer Learning from Unlabeled Data

- Self-taught learning for using unlabeled data in supervised classification tasks

**Algorithm 1: Self-taught Learning via Sparse Coding**

input Labeled training set
- \( T = \{(x_1^{(1)}, y_1^{(1)}), (x_2^{(2)}, y_2^{(2)}), \ldots, (x_m^{(m)}, y_m^{(m)})\} \)

Unlabeled data \( \{x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}\} \)

output Learned classifier for the classification task.

algorithm Using unlabeled data \( \{x^{(0)}\} \), solve the optimization problem (1) to obtain bases \( b \).

Compute features for the classification task to obtain a new labeled training set \( T' = \{(\tilde{a}(x^{(0)}), y^{(0)})_{n+1}\} \)

where \( \tilde{a}(x^{(0)}) = \arg\min_{a \in \mathbb{R}^{m+1}} \| x^{(0)} - \sum_j a_j b_j \|_2 + \beta \| a \|_1 \)

Learn a classifier \( C \) by applying a supervised learning algorithm (e.g., SVM) to the labeled training set \( T' \).

return the learned classifier \( C \).

Sparse coding performs better performance than Raw and PCA.

Sparse coding on unlabeled data provides better performance

<table>
<thead>
<tr>
<th>Training set size</th>
<th>Raw</th>
<th>PCA</th>
<th>Sparse coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>28.3%</td>
<td>28.6%</td>
<td>44.0%</td>
</tr>
<tr>
<td>1000</td>
<td>34.0%</td>
<td>26.3%</td>
<td>45.5%</td>
</tr>
<tr>
<td>5000</td>
<td>38.1%</td>
<td>38.1%</td>
<td>44.3%</td>
</tr>
</tbody>
</table>
Transfer Learn Faster CNNs

- Attentive feature distillation and selection (AFDS) to determine the important features to transfer.

- Learn a new model with a higher target task accuracy, but also further accelerates it by computing a subset of channel neurons in each convolutional layers.

- Better accuracy than other pruning models

<table>
<thead>
<tr>
<th>NS</th>
<th>SFP</th>
<th>AFDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.33%</td>
<td>59.63%</td>
<td>70.70%</td>
</tr>
</tbody>
</table>

Accuracy for 1/10th of the computations for given data.
This paper uses a model called Inception-v3 first trained on a base dataset and now trained on CIFAR-10 and Caltech Faces datasets.

Test 1: Change number of sample datasets
- Results: The accuracy scores are higher when more sample images are used for training aids.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Average Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10 (Test A)</td>
<td>70.1</td>
</tr>
<tr>
<td>CIFAR-10 (Test B)</td>
<td>66.1</td>
</tr>
<tr>
<td>Caltech Faces</td>
<td>63.7</td>
</tr>
</tbody>
</table>

Test 2: Change number of epochs
- Results: More epochs, higher the accuracy of trained models.

Test 3: Does type of image influence accuracy of the model?
- Result: Yes

<table>
<thead>
<tr>
<th>System</th>
<th>Human</th>
<th>Car</th>
<th>Animal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Accuracy (%)</td>
<td>93</td>
<td>87</td>
<td>73</td>
</tr>
</tbody>
</table>

Discussion: Results from the tests proved retraining the Inception-v3 model on the CIFAR-10 dataset resulted in better results compared to that stated in the previous state-of-the-art works, whereby authors did not use Transfer Learning and instead used a CNN trained on the same dataset (CIFAR-10) from scratch.
Transfer Learning for Image Classification

Limitations/possible improvements
- Time: More epochs = more accuracy = more computational time
- Customization of layers/weights could provide better accuracy

Future work
- Uses in biometric passwords
- IoT
- Using the Inception-v4 model

Deep Transfer Learning for Image-Based Structural Damage Recognition

- Deep Transfer Learning with VGG pre trained model is implemented, feature extractor and fine-tuning as two TL strategies are introduced
- Conclusions:
  - Fine-tuning can be used beyond feature extractor
  - CNN can be trained from scratch to improve performance

<table>
<thead>
<tr>
<th>Task and classes</th>
<th>Training accuracy</th>
<th>Test accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component type</td>
<td>Binary</td>
<td>96.4% 94.5%</td>
</tr>
<tr>
<td>Spelling condition</td>
<td>Binary</td>
<td>96.5% 91.5%</td>
</tr>
<tr>
<td>Damage level</td>
<td>Three classes</td>
<td>95.3% 89.7%</td>
</tr>
<tr>
<td>Damage type</td>
<td>Four classes</td>
<td>91.8% 68.8%</td>
</tr>
</tbody>
</table>

Results for retrain blocks experiment:
The model with retraining two blocks has better performance than only retraining one block
Building “Sentimental” Chatbots with Convolutional Neural Networks (CNN) & Recurrent Neural Networks (RNN)

By Claire Vu
CS 872 - Fall ’20

Task-oriented Chatbots

- Natural Language Understanding (NLU)
- Dialog Manager (DM)
- Natural Language Generation (NLG)
- Slot Filling
- Intent Classification
- Dialog State Tracking
- Dialog Policy (DP)
CNN Framework for Intent Classification (Sentimental Analysis)
CNN is great for reducing data complexity, thus expediting training time.
LSTM (Long Short-Term Memory)

GRU (Gated Recurrent Units)
RNN, using LSTM or GRU, can capture and retain the dependencies between tokens in the users query, thus detecting document contexts well.

Hybrid Slot Filling
Hybrid Response Selection

Conclusion

- The goal is to build a human-like chatbot. To make this happen, a generation-based with integrated attentional mechanism architecture is desired.
- Potential ethical issues.
- Thank you for listening!
Role of emojis in remote teaching

Inez Wibowo

Visual Communication Feedback Loop
Visual Communication Desert

Emojis as proxy to our face-to-face emotions

https://www.bingedaily.in/article/here-are-the-most-used-emojis-during-the-pandemic


https://arxiv.org/abs/1804.06737
Can emoji be proxy to emotions?

**Emotional support example:**

```
auuuubssss Aye I miss you 😢❤️
```

**Esteem support example:**

```
woocommerce_russian.ru Well done
```

**Informational support example:**

```
cdbublu My bookstore does price matching. If amazon or Barnes and noble or chegg sells/rents it cheaper, they will match it.
```

**Network support example:**

```
realtonyrose If you ever need any help with these text professors let me know I know a lot real well @tiennxo
```

**Tangible assistance example:**

```
Ife35c Let me know if you need help with the show tickets 😊
```

---


---


---

How accurate are emojis at identifying emotions?

**Table 1: Example sentences scored by our model.** For each text the top five most likely emojis are shown with the model’s probability estimates.

<table>
<thead>
<tr>
<th>Sentence</th>
<th>😊</th>
<th>😏</th>
<th>😘</th>
<th>😘</th>
<th>😩</th>
</tr>
</thead>
<tbody>
<tr>
<td>I love mom’s cooking</td>
<td>49.1%</td>
<td>8.8%</td>
<td>3.1%</td>
<td>3.0%</td>
<td>2.9%</td>
</tr>
<tr>
<td>I love how you never reply back...</td>
<td>14.0%</td>
<td>8.3%</td>
<td>6.3%</td>
<td>5.4%</td>
<td>5.1%</td>
</tr>
<tr>
<td>I love cruising with my homies</td>
<td>34.0%</td>
<td>6.6%</td>
<td>5.7%</td>
<td>4.1%</td>
<td>3.8%</td>
</tr>
<tr>
<td>I love messing with yo mind!!</td>
<td>17.2%</td>
<td>11.8%</td>
<td>8.0%</td>
<td>6.4%</td>
<td>5.3%</td>
</tr>
<tr>
<td>I love you and you’re just gone...</td>
<td>39.1%</td>
<td>11.0%</td>
<td>7.3%</td>
<td>5.3%</td>
<td>4.5%</td>
</tr>
<tr>
<td>This is shit</td>
<td>7.0%</td>
<td>6.4%</td>
<td>6.0%</td>
<td>6.0%</td>
<td>5.8%</td>
</tr>
<tr>
<td>This is the shit</td>
<td>10.9%</td>
<td>5.7%</td>
<td>6.5%</td>
<td>5.7%</td>
<td>4.8%</td>
</tr>
</tbody>
</table>

---


---

Diverse posts are better at identifying emotions

---

Residual Networks used to achieve accuracy


EmojiNet and Bag of Words


Results were convincing

\[ T_1: \text{Pray for my family 🙏 God gained an angel today.} \]
\[ T_2: \text{Hard to win, but we did it man 👏 Let's celebrate!} \]

\text{pray}(\text{verb}) : \{ \text{worship, thanksgiving, saint, pray, higher, god, confession} \}
\text{highfive}(\text{noun}) : \{ \text{palm, high, hand, slide, celebrate, raise, person, head, five} \}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
\textbf{Knowledge Concepts} & \textbf{Precision} & \textbf{Recall} & \textbf{Macro F1 Score} \\
\hline
Emoji Names & 78.23 & 69.23 & 69.42 \\
Emoji Senses & 73.79 & 70.25 & 71.98 \\
Emoji Sense Definitions & 71.56 & 69.98 & 70.49 \\
Processed Emoji Sense Definitions & 72.23 & 70.26 & 70.78 \\
\hline
\end{tabular}
\caption{Accuracies of emotion classification task using attention mechanism combining text and emoji as input for Bi-LSTM}
\end{table}

---

How can we apply emojis in education?

\begin{figure}
\centering
\includegraphics[width=\textwidth]{emojis_in_education}
\caption{Emoji emotional communication conceptual framework.}
\end{figure}

We can use deep learning (AS-SAN) to assess fine-grained performance

Emoji knowledge can help students communicate better
Deep Convolutional Neural Networks For Object Detection (R-CNN and its variants)

Xiaoqian Yang

Image Classification
- Single Object: Alexnet, VGG, GoogleNet, ResNet, DenseNet

Semantic Segmentation
- No objects, just pixels: FCN, SegNet, DenseNet, Unet

Object Detection
- R-CNN, Fast R-CNN, Faster R-CNN, YOLOV1-V4, SSD, EfficientNet

Instance Segmentation
- CAT, DOG, TREE, SKY

R-CNN consists of three models:
- CNN for feature extraction
- Linear SVM classifier for identifying objects
- Regression model for tightening the bounding boxes.


Slide credit: Ross Girshick
R-CNN

Achievements:
- R-CNN applies high-capacity CNN to bottom-up region proposals in order to localize and segment objects.
- Uses supervised pre-training and domain-specific fine-tuning, which is an effective paradigm for learning high-capacity CNNs when data is scarce.

Disadvantages:
- R-CNN uses the Selective Search Algorithm to get the Regions of Interest (ROI) which is rigid and there is no learning in this step, so bad region proposals may be generated sometimes.
- The process generates the CNN feature vector for every image region (N images * 2000) which is slow and time-consuming.
- R-CNN consists of three models. And the process involves these three models separately without much shared computation.
- Since we need to save feature maps of all the region proposals, it also increases the amount of disk memory required during training.

Fast R-CNN


Fast R-CNN (test time)

Solution:
- **R-CNN Problem #1:** Slow at test-time due to independent forward passes of the CNN
  - Solution: Share computation of convolutional layers between proposals for an image

- **R-CNN Problem #2:** It consists of three models. And the process involves these three models separately without much shared computation.
  - Solution: Combine all models into one network, so can train the whole system end-to-end all at once

Slide credit: Ross Girshick
**Fast R-CNN**: RoI (Region of Interest) Pooling and Multi-task Loss

RoI Pooling:
Share the computation across the approximately 2000 proposals.

The RoI pooling layer uses max pooling to convert the features inside any valid region of interest into a small feature map with a fixed-length feature vector, so that it can be fed into a sequence of fully connected layers.

\[
L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v).
\]

**Achievements:**
- Fast R-CNN is a single-stage end-to-end training with a multi-task loss.
- Training can update all network layers.
- No disk storage is required for feature caching.
- Compared to R-CNN, training time is shorter, and mAP is increased.

**Disadvantages:**
- Fast R-CNN still uses the Selective Search Algorithm.
Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NIPS 2015

RPN uses a sliding window over these feature maps, and at each sliding-window location, it generates $k$ Anchor boxes of different shapes and sizes. See Figure 6. And by default three scales (128, 256, 512) and three aspect ratios (1:1, 1:2, 2:1) are picked, yielding $k = 9$ anchors at each sliding position. For a convolutional feature map of a size $W \times H$, there are $WHk$ anchors in total.

RPN predicts two things: First, the classifier predicts the probability that an anchor is an object or not; Second, regression regresses the coordinates of the proposal to better fit the object.
Faster R-CNN

Achievements:
• Faster R-CNN presents RPN for efficient and accurate region proposal generation. Region proposal step is nearly cost-free. This method enables a unified, deep-learning-based object detection system to run at 5-17 fps.
• The learned RPN also improves region proposal quality and the overall object detection accuracy.

Disadvantages:
• Faster R-CNN is end-to-end training system, the performance of the systems further ahead depends on how the previous system performed. (This is only my personal opinion.)

Mask R-CNN


Faster R-CNN has two outputs for each candidate object, a class label and a bounding-box offset; to this Mask R-CNN adds a third parallel branch that outputs the object mask.

Mask R-CNN improves the RoI pooling layer to RoIAlign layer.
Mask R-CNN

Two modifications:
• Improve RoI Pooling layer to RoIAlign.
• Decouple mask and class prediction.

“RoiPool” is changed to “RoiAlign”

Compare R-CNN, Fast R-CNN & Faster R-CNN

Conclusions

<table>
<thead>
<tr>
<th></th>
<th>R-CNN</th>
<th>Fast R-CNN</th>
<th>Faster R-CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test time per image</td>
<td>50 seconds</td>
<td>2 seconds</td>
<td>0.2 seconds</td>
</tr>
<tr>
<td>Speed-up</td>
<td>1x</td>
<td>25x</td>
<td>250x</td>
</tr>
<tr>
<td>mAP (VOC 2007)</td>
<td>66.0%</td>
<td>66.9%</td>
<td>66.9%</td>
</tr>
</tbody>
</table>
Course Review

CSC 872
Pattern Analysis and Machine Intelligence

Course Summary

- You learned basic theories and their applications which yield fundamental computational tools used in various PAMI research fields.

- The bag of tools is large: so many useful tools that we did not cover

- But you learned how a few fundamental theoretical concepts cover a lot of interesting and useful tools for your research and development

- So now you should be able to understand how specific algorithms are derived and know when you should use them for what
Important Theories

- **Deductive Reasoning**
  - Derive a new knowledge $B$ from a set of knowledge $A$ so that when $A$ is true $B$ is also true
  - Formal Logic Languages
  - Logical Inferences: Sound&Complete! Entailment

- **Inductive Reasoning**
  - Inferring $A$ from multiple examples of $B$s
  - Bayesian Inferences: Dealing with uncertainty
  - Maximum Likelihood Estimation: Foundation.
  - Deep Learning does this well.

Knowledge Representation

- Formal languages
  - Propositional & First-Order Logic
- Matrix algebra
  - Vector & Matrix
- Probability
  - Random variable
  - Probability distribution
- Relations
  - Logical entailment
  - Scalar function
  - Neural network
- Distributed vs Local Representation
  - CNN

Know how to describe knowledge in mathematically principled way!!!
**Problem Formulation**

- What is the foundation tasks?
  - Agents
  - Inference
  - Modeling
  - Learning
  - Classification
  - Regression
  - Search/Optimization (mini/maxi-mization)

  *Know many algorithms around can actually be formulated using one of these formulations*

**Problem Solving**

- How to numerically solve the formulated problem?
  - Search: Depth-First, Width-First, A*
  - Logical Inference: Resolution
  - Kernel Density Estimation (KDE)
  - Expectation-Maximization (EM) Algorithm
  - Principal Component Analysis (PCA)
  - Linear Discriminant Analysis (LDA)
  - Least Squares Fitting
  - Gradient Descent
  - Simulated Annealing
  - Perceptron
  - Back Propagation

  *It is a bag of tools. More you use/know, better off you will be*
**What comes next??**

- Induction + Deduction
- Active Learning
- Text-based Inputs (Learn from books)
- Unsupervised Learning (GAN)
- Combinatorial Problem
- Explanation
- *Your new great ideas here …*

---

**Critical Thinking is your ticket**

*Cogito, Ergo Sum*

And please submit your course eval