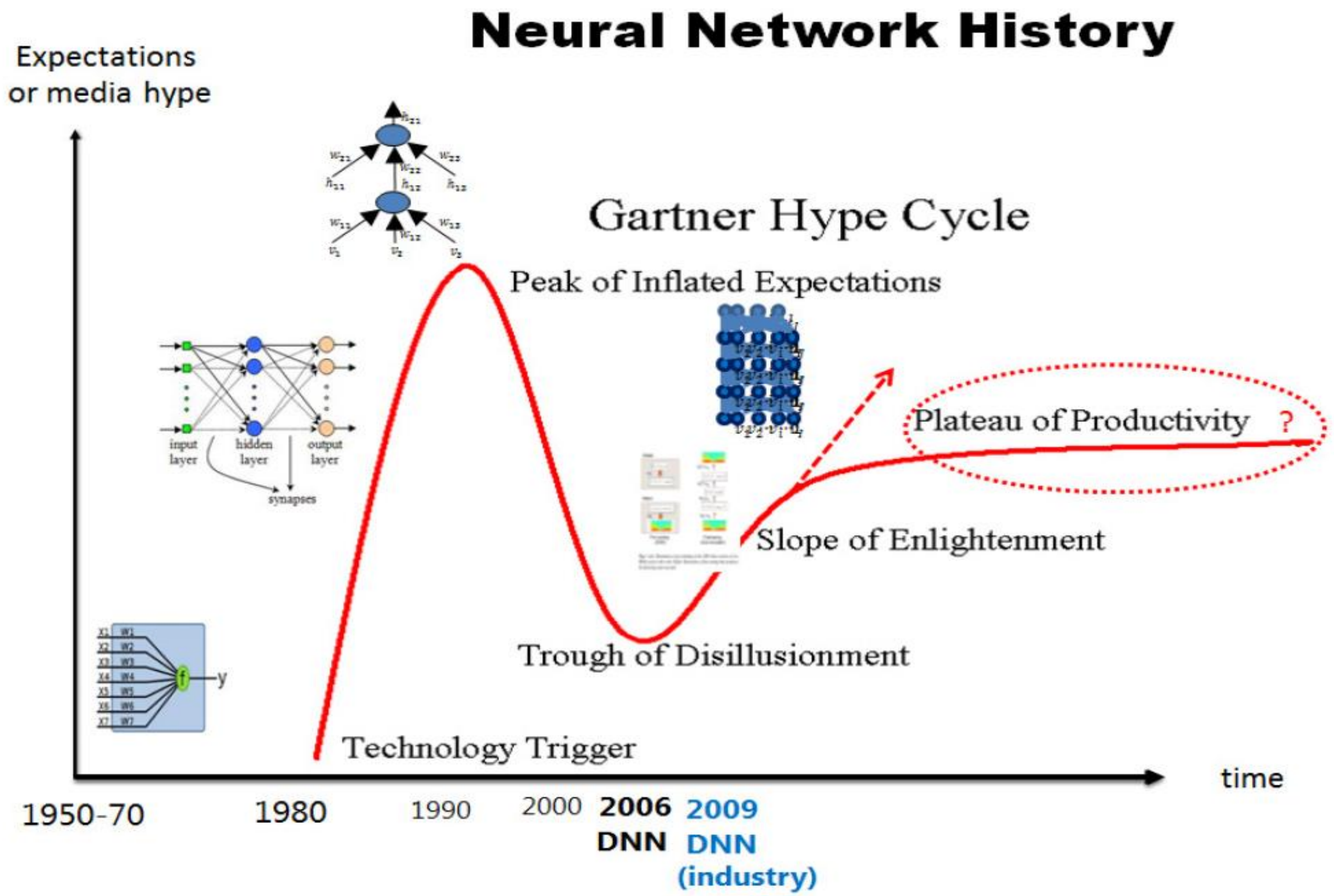
Module 1

Introduction to NLP with Deep Learning

Introduction of NLP with Deep Learning

- Overview
  - A brief history of deep neural networks (DNN)
  - An example of neural models for topic classification
  - Different forms of DNN for classification/ranking/generation tasks

A brief history of deep neural networks (DNN)



[Deng & Yu, 2014]

Deep Learning in recent years

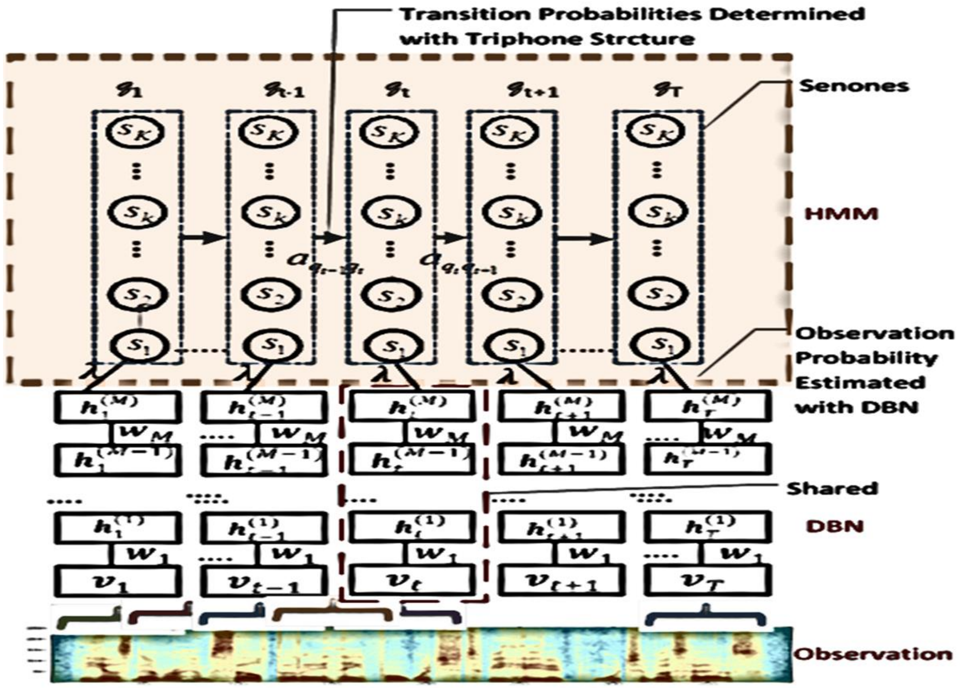
- In 2013, MIT Technology Review rated Deep Learning as one of the 10 major breakthrough technologies.

- Back in 2012, Rick Rashid, Microsoft's top scientist, demonstrated a voice recognition program using deep learning technology.

CD-DNN-HMM

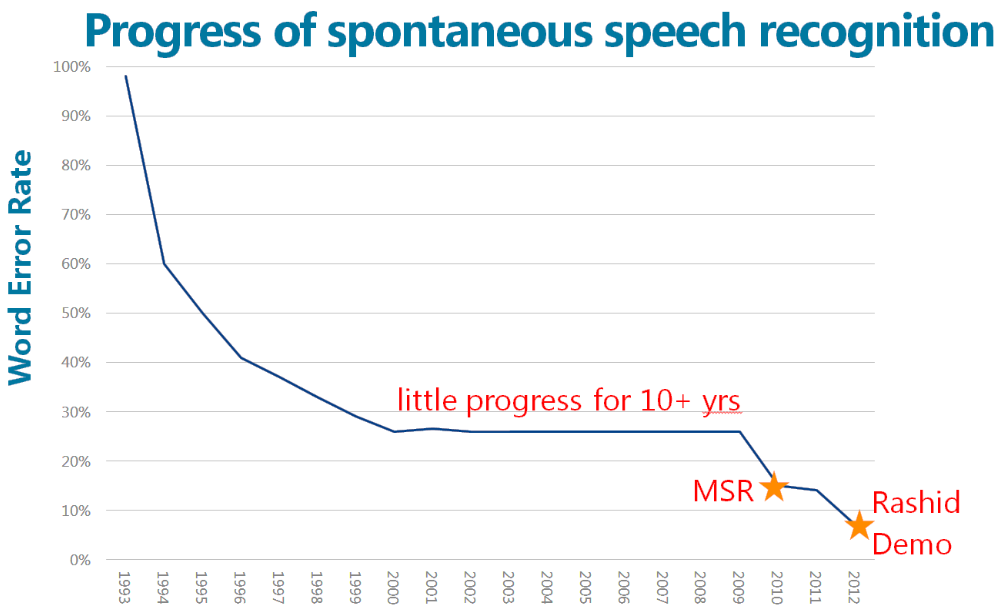
Dahl, Yu, Deng, and Acero, "Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition," *IEEE Trans. ASLP*, Jan. 2012

Seide, Li, and Yu, "Conversational Speech Transcription using Context-Dependent Deep Neural Networks," *INTERSPEECH* 2011.


Transition Probabilities Determined with Triphone Strcture
Senones
HMM
Observation Probability Estimated with DBN
Shared DBN
Observation

After no improvement for 10+ years by the research community…

MSR reduced error from **~23%** to **<13%** (and under 7% for Rick Rashid's S2S demo)!



Skype to get 'real-time' translator

Analysts say the translation feature could have wide ranging applications

Object Recognition

ImageNet:

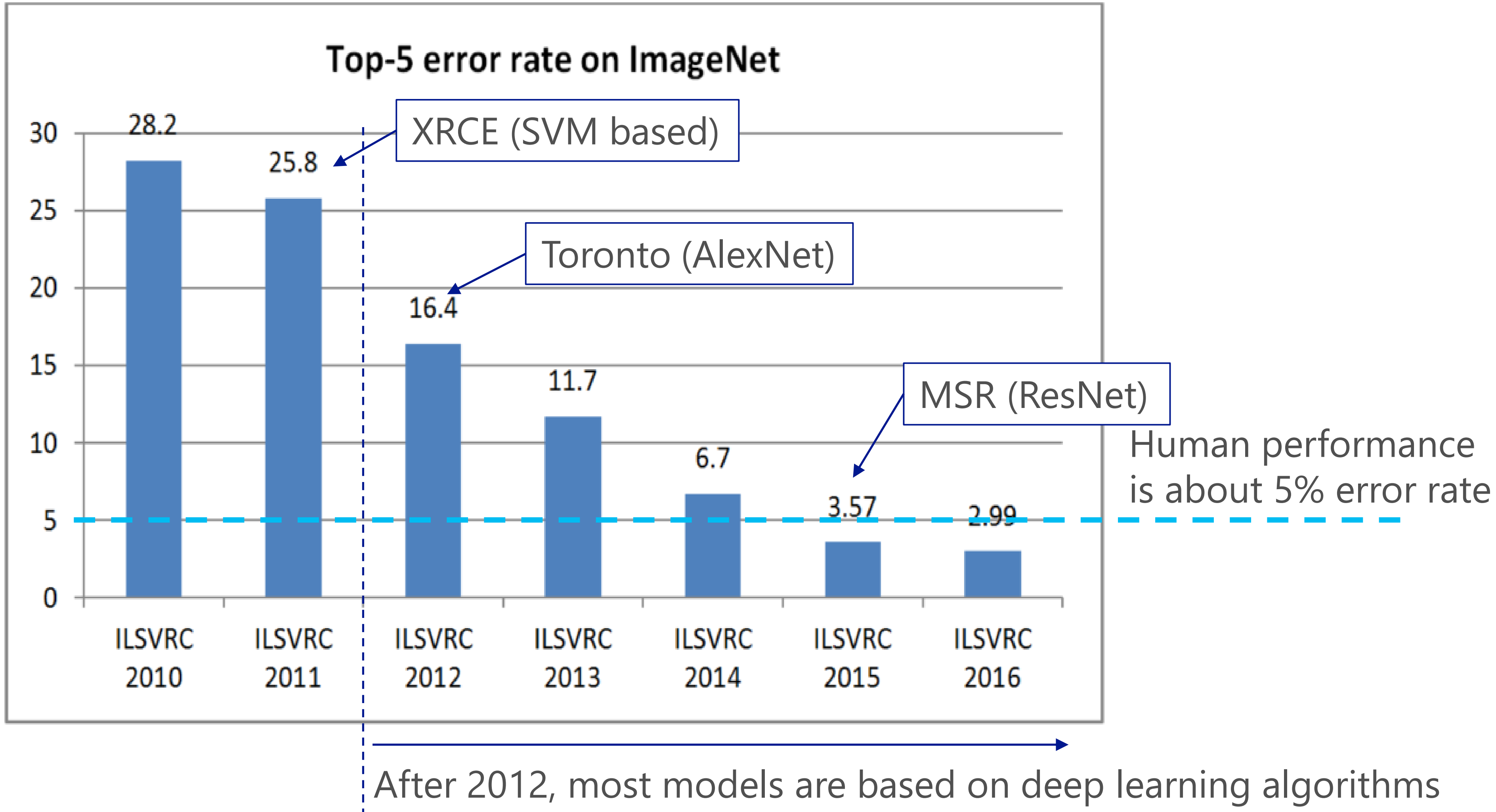
Large Scale Visual Recognition Challenge (ILSVRC)

[Fei-Fei Li, et al]

- Task: 1000-category image classification
- Data: Millions of images with labels
- Challenge: Annual competition since 2010

Object Recognition

Reached human parity on the ImageNet Benchmark in 2015



Human performance is about 5% error rate

After 2012, most models are based on deep learning algorithms

The focus of this course

- Is not on speech or image,
- But on text and natural language processing
  - Machine translation
  - Conversation
  - Information retrieval
  - Semantic parsing
  - Question answering
  - Image captioning
  - Etc.

A query classification problem

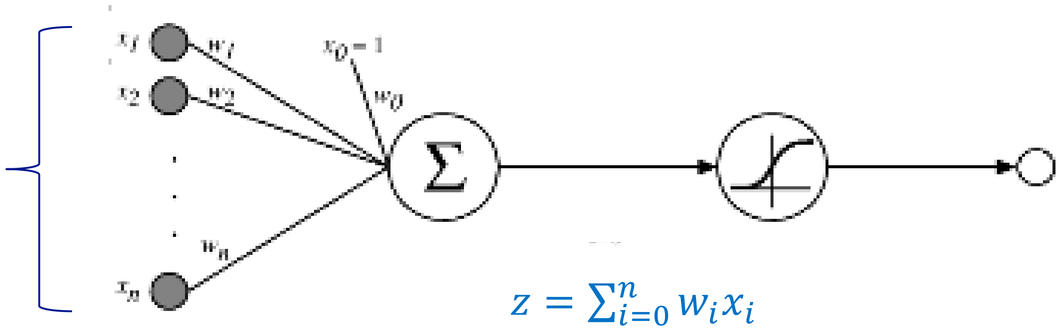
- Given a search query q, e.g., “denver sushi downtown”
- Identify its domain c e.g.,
  - Restaurant
  - Hotel
  - Nightlife
  - Flight
  - etc.
- So that a search engine can tailor the interface and result to provide a richer personalized user experience

A single neuron model

- For each domain $\boldsymbol{c}$, build a binary classifier
  - Input: represent a query q as a vector of features $x = [x_1, \dots x_n]^T$
  - Output: $y = P(1|q, c)$
  - q is labeled c is$P(1|q, c) > 0.5$
- Input feature vector, e.g., a bag of words vector
  - Regards words as atomic symbols: *denver, sushi, downtown*
  - Each word is represented as a one-hot vector: $[0, \dots, 0,1,0, \dots, 0]^T$
  - Bag of words vector = sum of one-hot vectors
  - We may use other features, such as n-grams, phrases, (hidden) topics

A single neuron model

Input features x

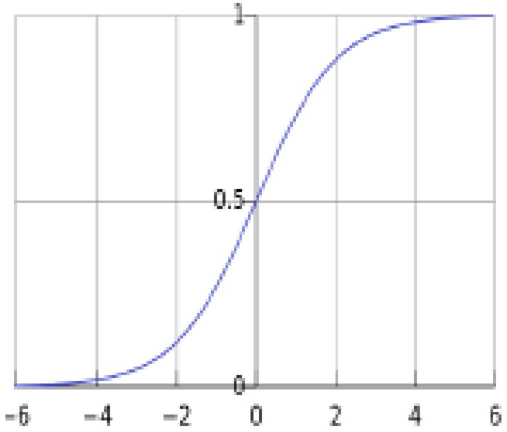


Output: $P(1|q,c)$

$y = \sigma(z) = \frac{1}{1+\exp(-z)}$

$z = \sum_{i=0}^{n} w_i x_i$

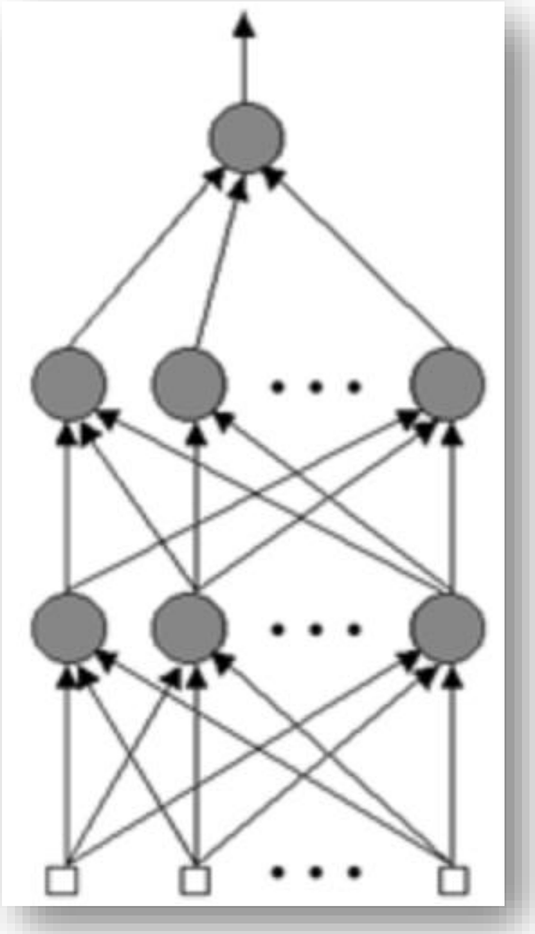
- w: weight vector to be learned
- z: weighted sum of input features
- σ: the logistic function
  - Turn a score to a probability
  - non-linear activation function, essential in DNN models



Model training: how to assign $\boldsymbol{w}$

- Training data: a set of $\left(x^{(m)}, y^{(m)}\right)_{m=\{1,2,\ldots,M\}}$ pairs
  - Input $x^{(m)} \in R^n$
  - Output $y^{(m)} = \{0,1\}$
- optimize parameters $\boldsymbol{w}$ on training data
  - minimize a loss function (mean square error loss)
    - $\min_{w} \sum_{m=1}^{M} L^{m}$
    - where $L^{(m)} = \frac{1}{2}\left(f_w\left(x^{(m)}\right) - y^{(m)}\right)^2$
  - Using Stochastic Gradient Descent (SGD)
    - Initialize w randomly
    - Update for each training sample until convergence: $w^{new} = w^{old} - \eta \frac{\partial L}{\partial w}$

Multi-layer (deep) neural networks



Output layer $y^o = \sigma(w^T y^2)$

Vector w

2st hidden layer $y^2 = \sigma(\mathbf{W}_2 y^1)$

Projection matrix $\mathbf{W}_2$

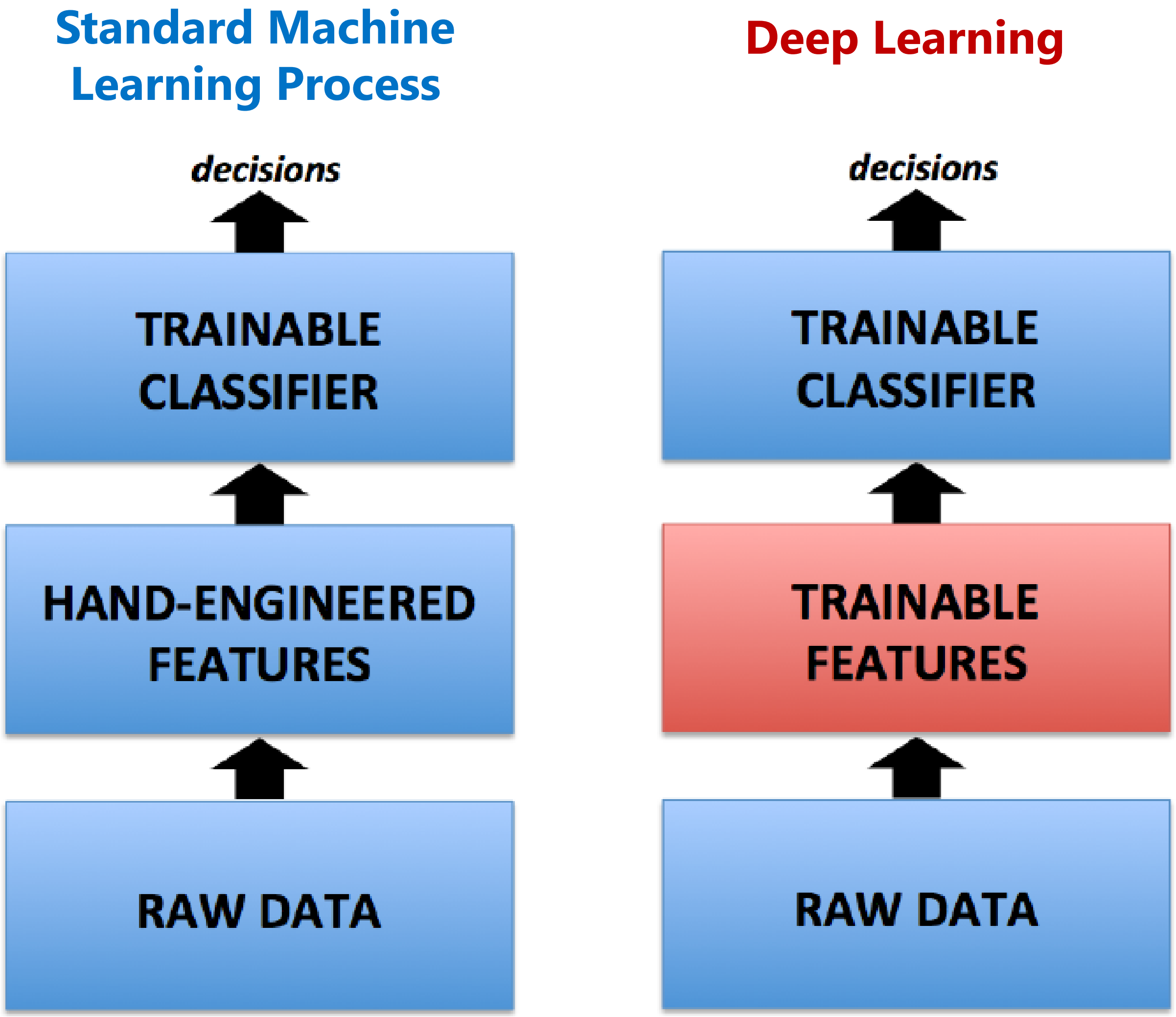
1st hidden layer $y^1 = \sigma(\mathbf{W}_1 x)$

Projection matrix $\mathbf{W}_1$

Input features x

This is exactly the **single neuron model** with **hidden** features.

Feature generation: project raw input features (bag of words) to **hidden** features (topics).



Standard Machine Learning Process

decisions

TRAINABLE CLASSIFIER

HAND-ENGINEERED FEATURES

RAW DATA

Deep Learning

decisions

TRAINABLE CLASSIFIER

TRAINABLE FEATURES

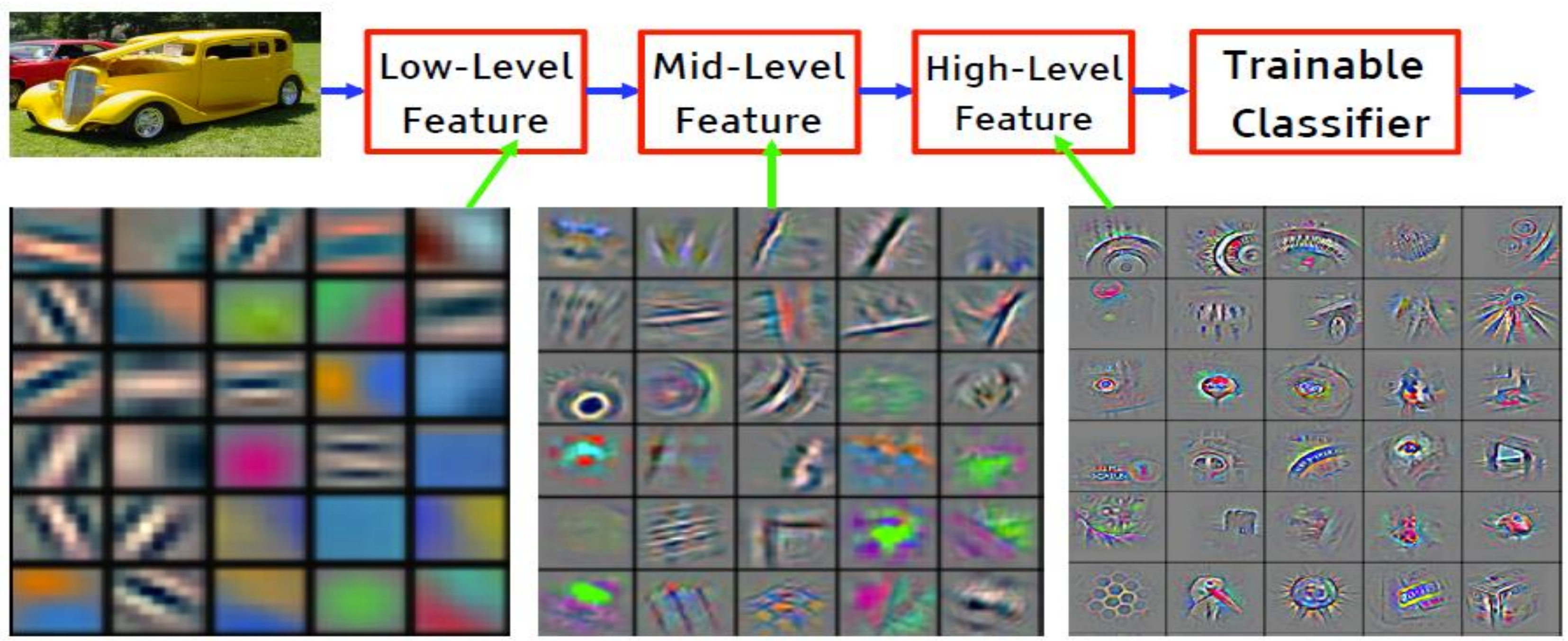
RAW DATA

Adapted from [Duh 14]

Why Multiple Layers?

DL tutorial at NIPS'2015

- Hierarchy of representations with increasing level of abstraction
- Each layer is a trainable feature transform
- Image recognition: pixel → edge → texton → motif → part → object
- ?? Text: character → word → word group → clause → sentence → story



Different forms of Deep Learning for NLP

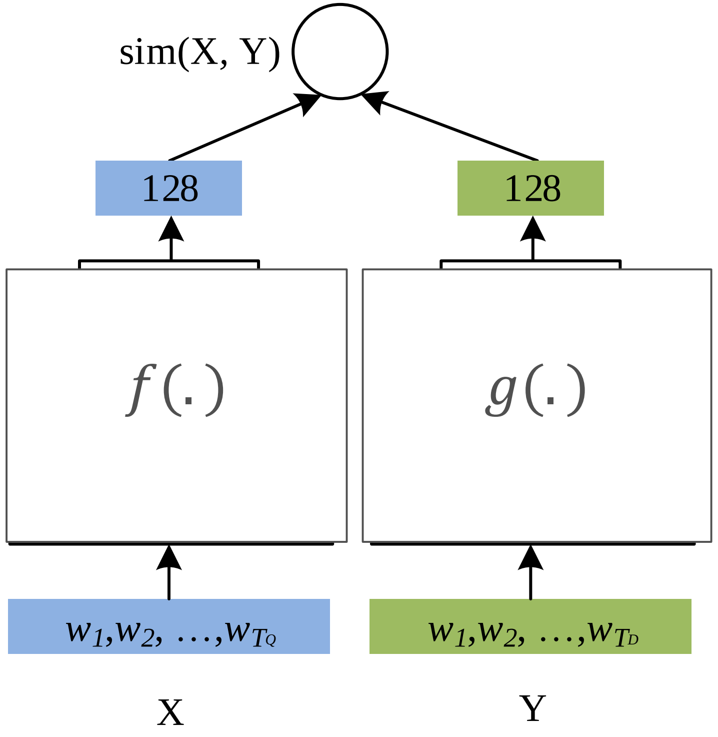
- Classification task – label X by Y
  - Multi-Layer Perceptron
  - Convolutional NN
- Ranking task – compute the sim btw X and Y
  - Siamese neural network [Bromley et al. 1993]
  - Deep Semantic Similarity Model (DSSM)
- (Text) Generation task – generate Y from X
  - RNN/LSTM, Seq2Seq
  - Memory Network

DSSM: Compute Similarity in Semantic Space

Relevance measured by cosine similarity

Word sequence

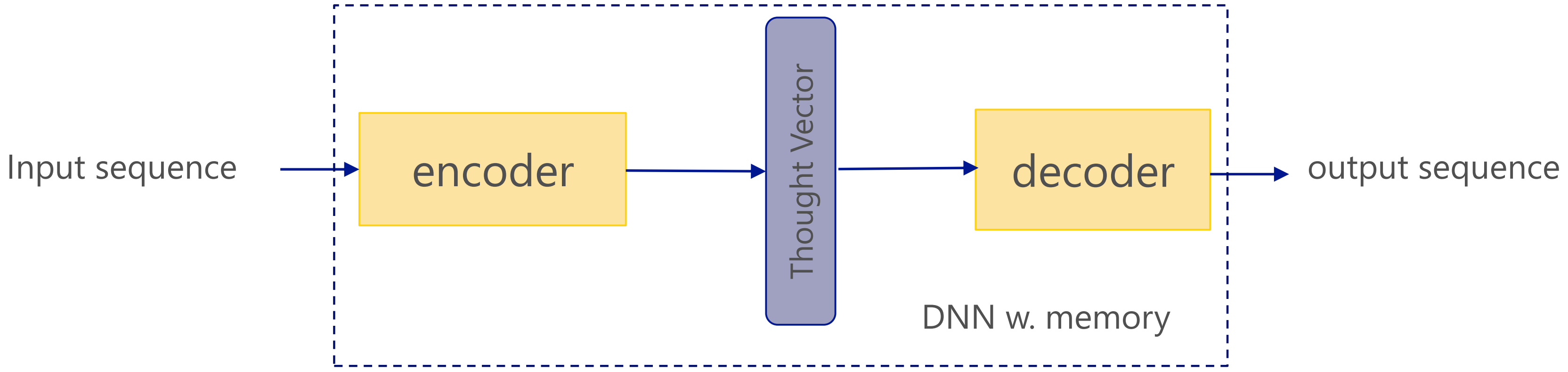
x_t



Learning: maximize the similarity between X (source) and Y (target)

Representation: use DNN to extract abstract semantic representations

Sequence-to-Sequence Tasks



- Machine translation (MT):
  - A sentence in source language → A sentence in target language
- Conversation (chitchat):
  - Context + message → response

End-to-End Memory Networks (MemNN)

[Sukhbaatar+ 15]

- Retrieving long-term mem x
- Embedding input

  $m_i = Ax_i$

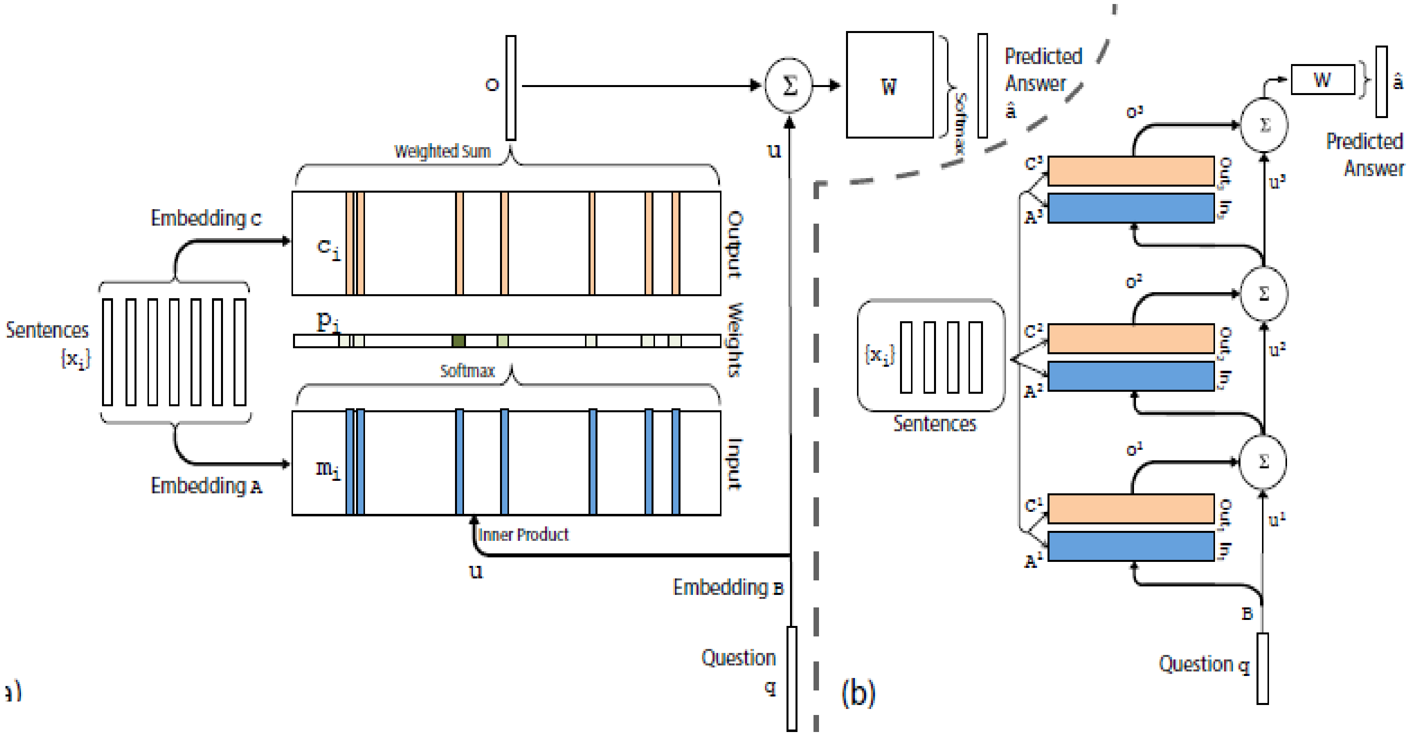
  $c_i = Cx_i$

  $u = Bq$
- Attention over memories

  $p_i = \text{softmax}(u^T m_i)$
- Generating (ranking) the final answer

  $o = \sum_i p_i c_i$

  $a = \text{softmax}(W(o + u))$



a)
o
Weighted Sum
Σ
W
Softmax
Predicted Answer
â
u
Embedding C
c_i
Output
Sentences
$\{x_i\}$
p_i
Weights
Softmax
m_i
Input
Embedding A
Inner Product
u
Embedding B
Question
q
(b)
$\{x_i\}$
Sentences
o^3
C^3
A^3
Out_3
In_3
u^3
W
â
Predicted Answer
o^2
C^2
A^2
Out_2
In_2
u^2
o^1
C^1
A^1
Out_1
In_1
u^1
B
Question q