Applied Deep Learning

Yun-Nung (Vivian) Chen www.csie.ntu.edu.tw/~yvchen/f105-adl

Review

Dec. 29th, 2016

National Taiwan University

NN Basics

1) What is the model? (function hypothesis set) — **Neural Network**

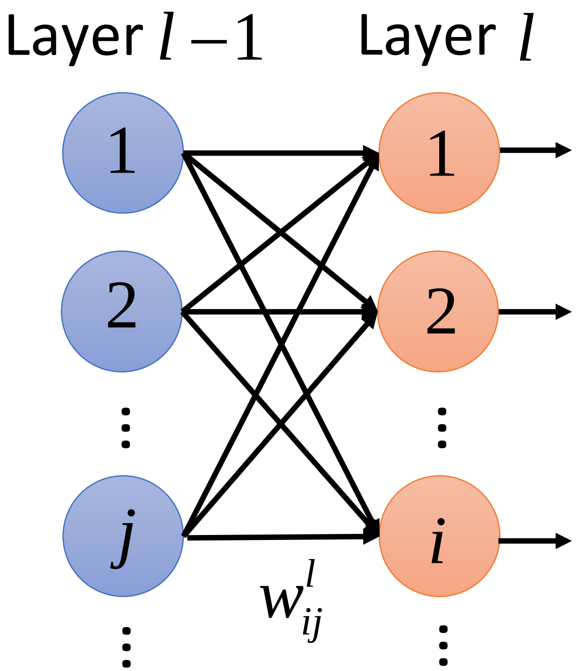
2) What is the best function? — **Cost Function**

3) How can we pick the "best" function? — **Gradient Descent**

- ✓ Initialization
- ✓ Learning Rate
- ✓ Stochastic Gradient Descent / Mini-Batch
- ✓ Learning Recipe
- ✓ Overfitting

Backpropagation

$$\frac{\partial C(\theta)}{\partial w_{ij}^l} = \frac{\partial C(\theta)}{\partial z_i^l} \frac{\partial z_i^l}{\partial w_{ij}^l}$$



δ_i^l

$$\frac{\partial z_i^l}{\partial w_{ij}^l} = \begin{cases} a_j^{l-1} & l > 1 \\ x_j & l = 1 \end{cases}$$

Backward Pass

$$\delta^L = \sigma'(z^L) \odot \nabla C(y)$$
$$\delta^{L-1} = \sigma'(z^{L-1}) \odot (W^L)^T \delta^L$$
$$\vdots$$
$$\delta^l = \sigma'(z^l) \odot (W^{l+1})^T \delta^{l+1}$$
$$\vdots$$

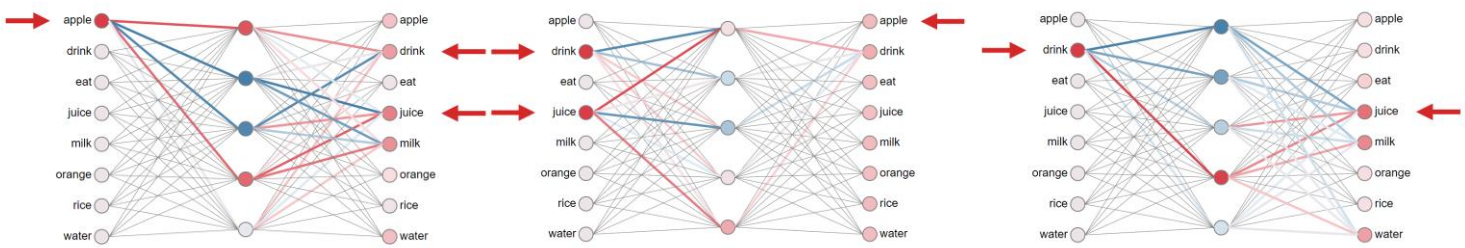
Forward Pass

$$z^1 = W^1 x + b^1$$
$$a^1 = \sigma(z^1)$$
$$\vdots$$
$$z^l = W^l a^{l-1} + b^l$$
$$a^l = \sigma(z^l)$$
$$\vdots$$

Word Embeddings

Low dimensional word vector

- word2vec



Skip-gram CBOW LM

- GloVe: combining count-based and direct learning

Word vector evaluation

- Intrinsic: word analogy, word correlation
- Extrinsic: subsequent task

Softmax loss = cross-entropy loss

Sequence Modeling

Idea: aggregate the meaning from all words into a vector

→ ***Compositionality***

Method:

- Basic combination: average, sum
- Neural combination:
  - ✓Recursive neural network (RvNN)
  - ✓Recurrent neural network (RNN)
  - ✓Convolutional neural network (CNN)

N-dim

這 (this)	$[0.2\ 0.6\ 0.3\ \cdots\ 0.4]$
規格 (specification)	$[0.9\ 0.8\ 0.1\ \cdots\ 0.1]$
有 (have)	$[0.1\ 0.3\ 0.1\ \cdots\ 0.7]$
誠意 (sincerity)	$[0.5\ 0.0\ 0.6\ \cdots\ 0.4]$

How to compute $\vec{x} = [x_1\ x_2\ x_3\ \cdots\ x_N]$

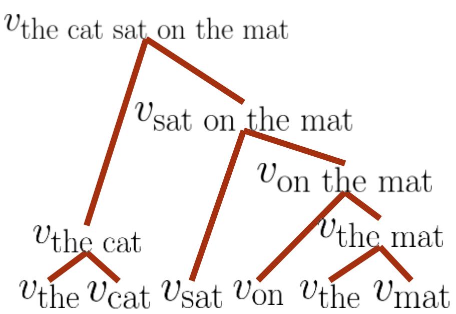
Recursive Neural Network (RvNN)

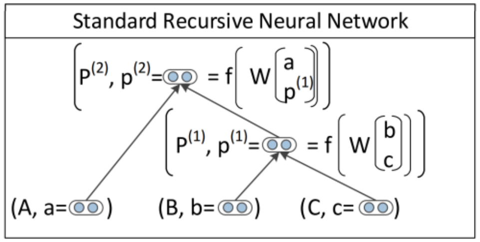
Recursive Neural Network

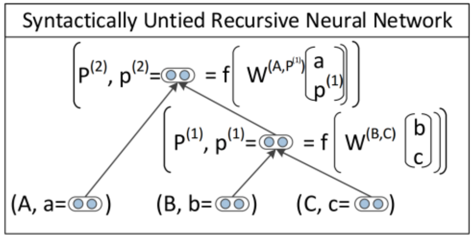
- Idea: syntactic compositionality & language recursion

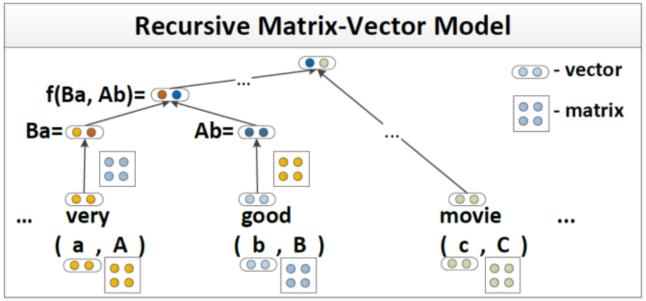
Network Variants

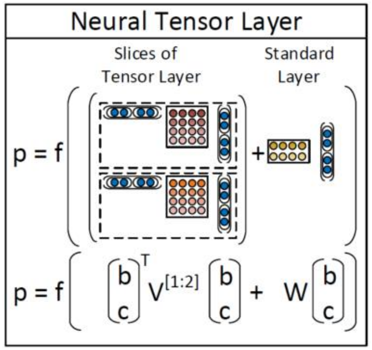
- Standard Recursive Neural Network
  - Weight-Tied
  - Weight-Untied
- Matrix-Vector Recursive Neural Network
- Recursive Neural Tensor Network











Recurrent Neural Networks (RNN)

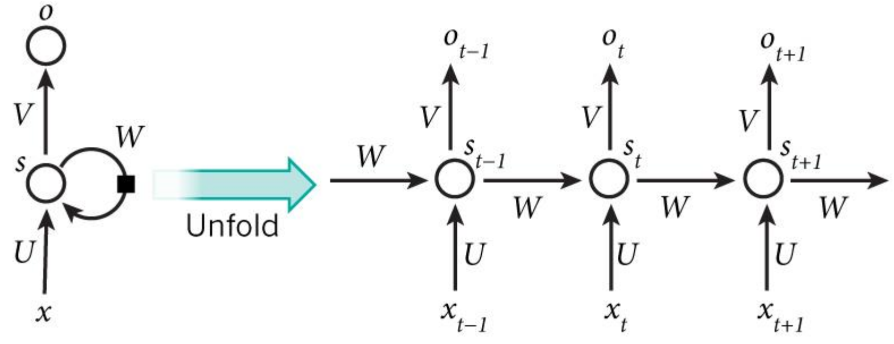
Language Modeling

- RNNLM

Recurrent Neural Networks

- Definition

$$s_t = \sigma(W s_{t-1} + U x_t)$$
$$o_t = \mathrm{softmax}(V s_t)$$



- Backpropagation through Time (BPTT)
- Vanishing/Exploding Gradient

Applications

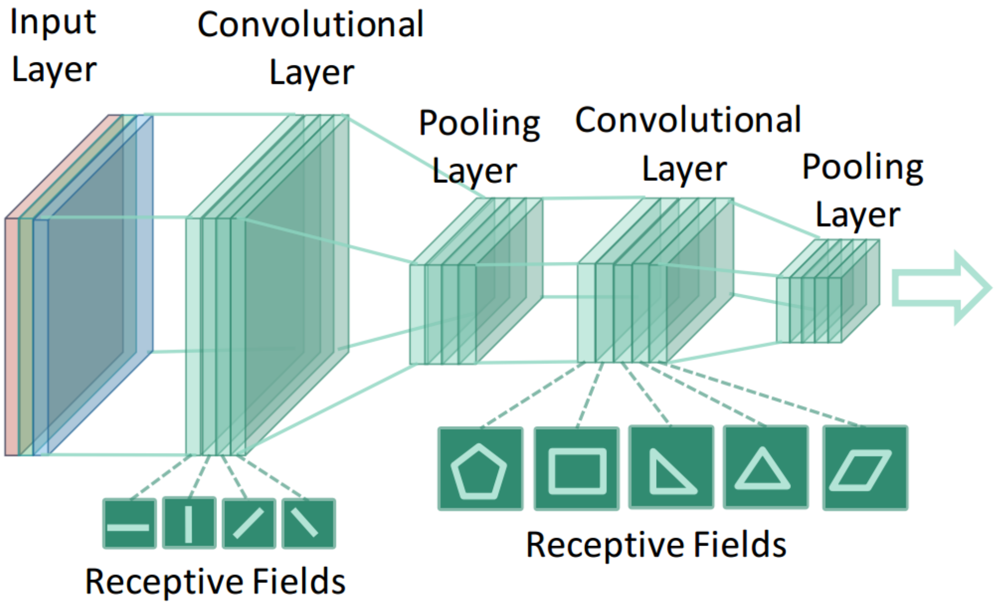
- Sequential Input: Sequence-Level Embedding
- Sequential Output: Tagging / Seq2Seq (Encoder-Decoder)

Convolutional Neural Networks (CNN)

Convolutional neural networks

- capture contextual information and then pool out salient features
- most common model for visual information

Image credit from Sipun



NN Training Tips

Data Preprocessing: Input Normalization

Activation Function: ReLU, Maxout

Loss Function: Softmax

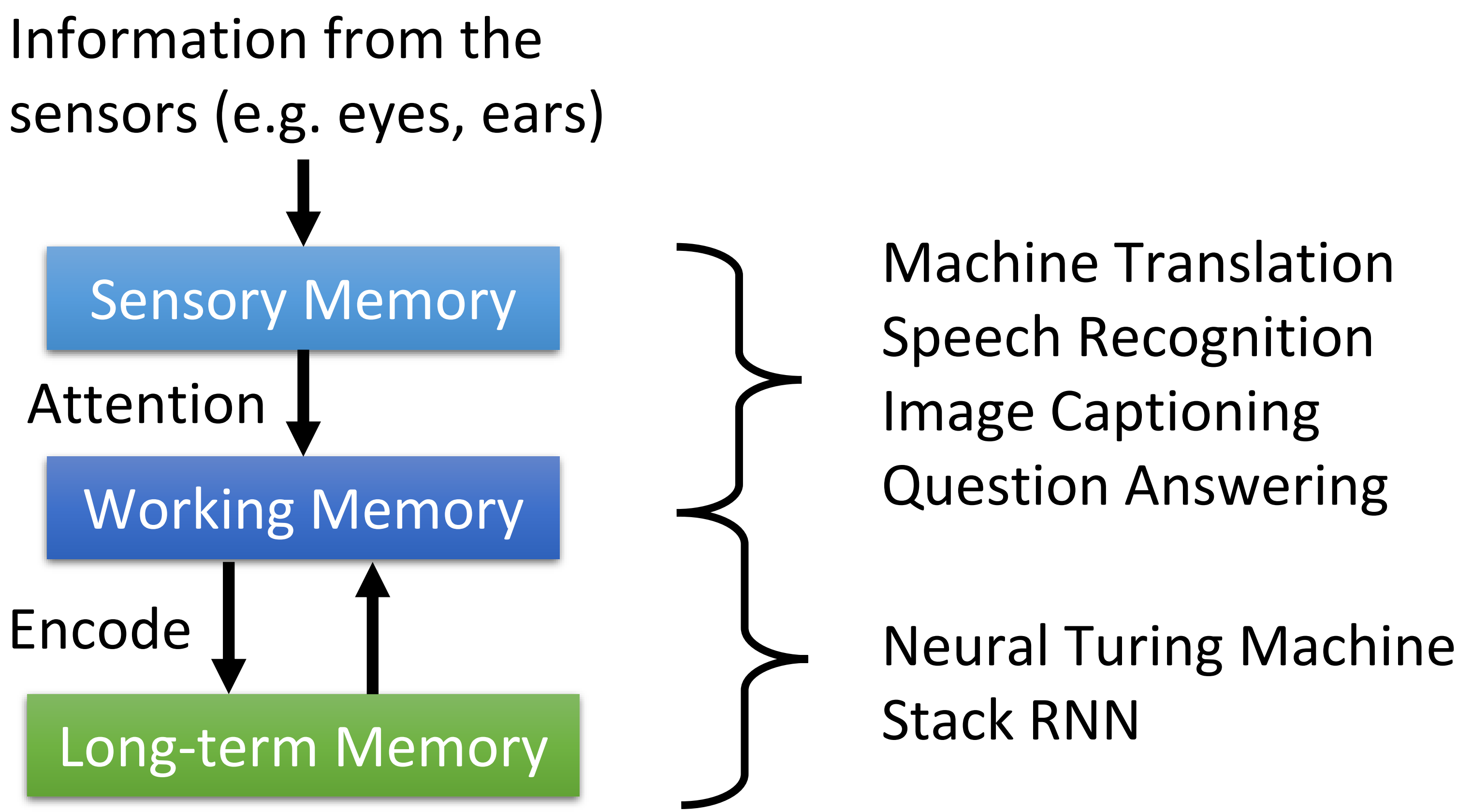
Optimization

- Adagrad: Learning Rate Adaptation
- Momentum: Learning Direction Adaptation

Generalization

- Early Stopping: avoid too many iterations from overfitting
- Regularization: minimize the effect of noise
- Dropout: leverage the benefit of ensemble

Attention Mechanism

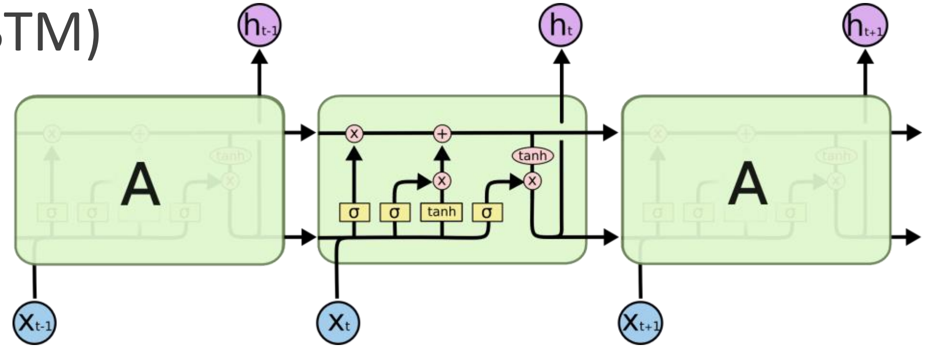


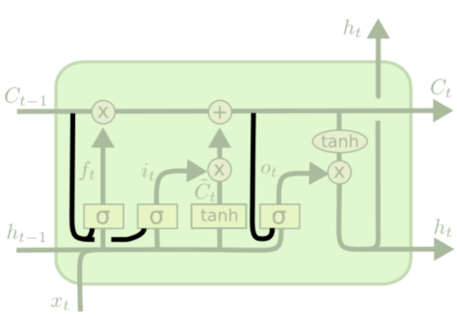
Gating Mechanism

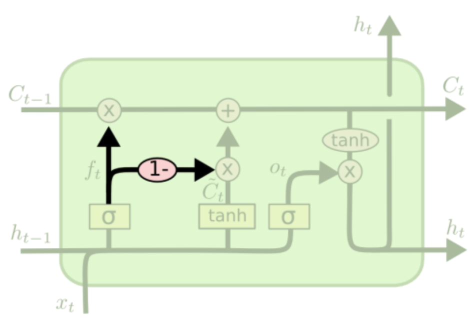
Gating mechanism for vanishing gradient problem

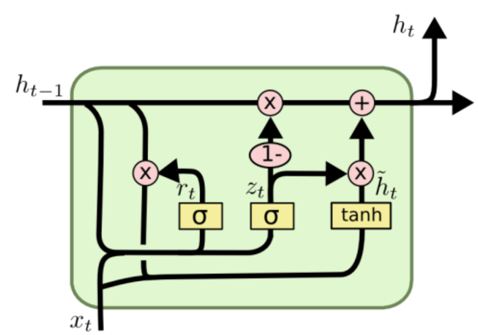
Gated RNN

- Long Short-Term Memory (LSTM)
  - Peephole Connections
  - Coupled Forget/Input Gates
- Gated Recurrent Unit (GRU)









Deep Reinforcement Learning

RL is a general purpose framework for **decision making** under interactions between *agent* and *environment*

- RL is for an *agent* with the capacity to *act*
- Each *action* influences the agent's future *state*
- Success is measured by a scalar *reward* signal
- Goal: *select actions to maximize future reward*

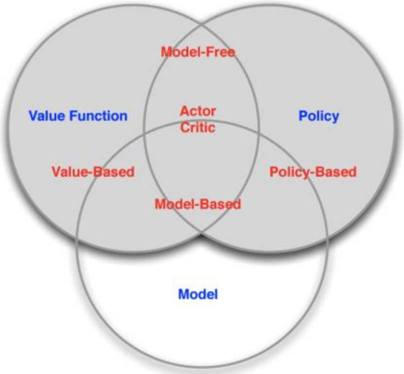
action

state

reward

An RL agent may include one or more of these components

- Policy: agent's behavior function
- Value function: how good is each state and/or action
- Model: agent's representation of the environment



RL problems can be solved by end-to-end deep learn

Reinforcement Learning + Deep Learning = AI

Unsupervised Learning

Labeling data is expensive, but we have large unlabeled data

Autoencoder

- exploits the unlabeled data to learn latent factors as representations
- learned representations can be transfer to other tasks

Generative models

- have the potential to understand and explain the underlying structure of the input data even when there are no labels
- Generative Adversarial Networks (GAN): jointly train two competing networks, **generator** and **discriminator**

Generative Models

Generative adversarial networks (GAN)

- jointly train two competing networks, generator and discriminator

Adversarially learned inference (ALI) / bidirectional GAN (BiGAN)

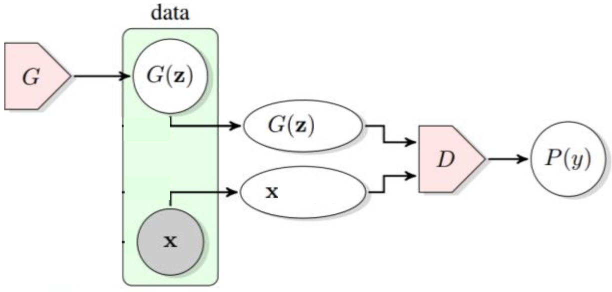
- jointly train three networks, generator, encoder, and discriminator
- latent variables can be encoded

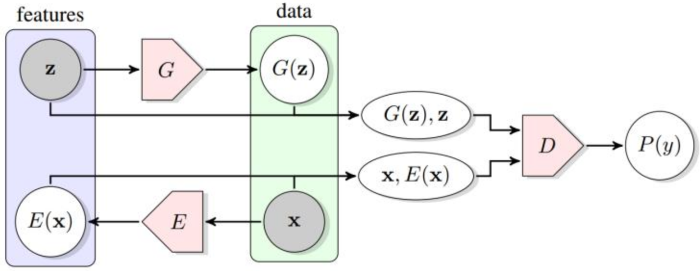
Training tricks

- Generator objective: feature matching, unrolled GAN
- Discriminator objective: minibatch discrimination, GAP

Applications

- semi-supervised learning

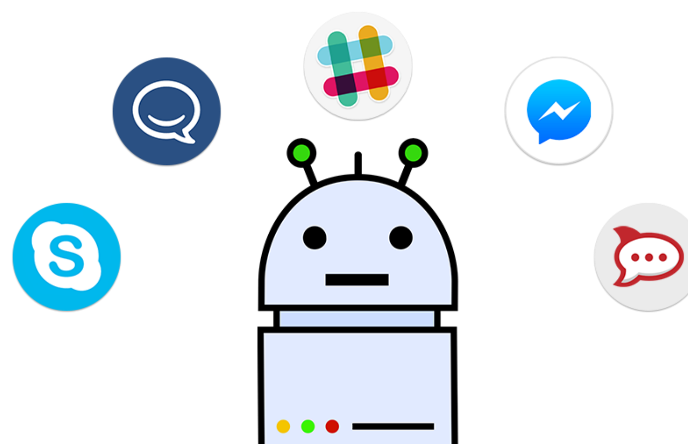




Advertisement

CSIE 5440 – “Intelligent Conversational Bot 智慧對話機器人”

- Semester: 2017 Spring (105-2)
- Time: Tue 2,3,4
- Location: CS Building 102
- Prerequisite background: deep learning, machine learning
- Goal: each team aims at building a task-oriented conversational bot and demonstrates the prototype in a show and tell poster session



Thanks for Your Participation!

DO NOT FORGET TO PROVIDE ME COURSE FEEDBACK