Deep Neural Net Approaches for Natural Language Processing

Gary Geunbae Lee
Dept of CSE/AIGS
POSTECH
Artificial Neural Networks (ANN)

Terminal Branches of Axon

Dendrites

Activation Function

Axon

Σ

\[ x_1, x_2, x_3, \ldots, x_n \]

\[ w_1, w_2, w_3, \ldots, w_n \]
Layered Networks

Output: \[ y = f(w_1^1 x + w_2^2 x + w_3^3 x + \cdots + w_m^m x_m) \]
\[ = f(\sum_j w_j^i x_j) \]
Deep learning Innovation

• Combining Feature Learning and Classification as Unified Framework (※ Learning what to learn, how to learn)

Feature learning aspect of DNN based Image Classification
Vanilla recurrent neural networks (RNNs)

- RNNs have connections from the outputs of previous time steps to inputs of next time steps.

- For sequential data, a RNN usually computes hidden state $h_t$ from the previous hidden state $h_{t-1}$ and the input $x_t$.
  - $h_t = \sigma(W_h h_{t-1} + W_x x_t + b)$
Vanishing gradient problem

- \( h_t = \sigma(W_h h_{t-1} + W_x x_t + b) \)

- Let’s assume \( \sigma \) is the identity function

\[
\frac{\partial J^{(i)}(\theta)}{\partial h^{(j)}} = \frac{\partial J^{(i)}(\theta)}{\partial h^{(i)}} \prod_{j < t \leq i} \frac{\partial h^{(t)}}{\partial h^{(t-1)}}
\]

\[
= \frac{\partial J^{(i)}(\theta)}{\partial h^{(i)}} \prod_{j < t \leq i} W_h = \frac{\partial J^{(i)}(\theta)}{\partial h^{(i)}} W^\ell_h
\]

If all \( \frac{\partial h^t}{\partial h^{t-1}} < 1 \right) \frac{\partial J^t}{\partial h^1} \approx 0 \)
Long short-term memory networks (LSTMs)

- LSTMs explicitly keep and update cell memory $c(t)$ by
  - Removing the previous cell content $c(t-1)$ by multiplying it with $f(t)$
  - Adding the new cell content $\tilde{c}(t)$ multiplied by $i(t)$
- LSTMs produce output $h(t) = o(t) \circ \text{tanh} \ c(t)$

\[
\begin{align*}
    f(t) &= \sigma \left( W_f h(t-1) + U_f x(t) + b_f \right) \\
    i(t) &= \sigma \left( W_i h(t-1) + U_i x(t) + b_i \right) \\
    o(t) &= \sigma \left( W_o h(t-1) + U_o x(t) + b_o \right) \\
    \tilde{c}(t) &= \text{tanh} \left( W_c h(t-1) + U_c x(t) + b_c \right) \\
    c(t) &= f(t) \circ c(t-1) + i(t) \circ \tilde{c}(t) \\
    h(t) &= o(t) \circ \text{tanh} \ c(t)
\end{align*}
\]
Gated recurrent units (GRUs)

• GRUs keeps and update $h^{(t)}$ by two gates:
  • Update gate $u^{(t)}$ decides
    • How much the old hidden representation $h^{(t)}$ is removed
    • how much the new hidden representation $\tilde{h}^{(t)}$ is added
  • Reset gate $r^{(t)}$ decides how much old representation $h^{(t)}$ is needed to compute new representation $\tilde{h}^{(t)}$

• GRUs also use less number of gates and have smaller parameters than LSTMs

\[
\begin{align*}
  u^{(t)} &= \sigma \left( W_u h^{(t-1)} + U_u x^{(t)} + b_u \right) \\
  r^{(t)} &= \sigma \left( W_r h^{(t-1)} + U_r x^{(t)} + b_r \right) \\
  \tilde{h}^{(t)} &= \tanh \left( W_h (r^{(t)} \circ h^{(t-1)}) + U_h x^{(t)} + b_h \right) \\
  h^{(t)} &= (1 - u^{(t)}) \circ h^{(t-1)} + u^{(t)} \circ \tilde{h}^{(t)}
\end{align*}
\]
Parallel computing for Deep Learning

- History of parallel/distributed systems for Deep Learning computing

Google taps 16k computers to look for cats – for Science!

Univ. of Toronto uses 2 GPUs for 1.2M training images for 1000 classes image classification (※ ImageNet Large Scale Visual Recognition Challenge)

Stanford uses 12 GPUs for large-scale video classification with Convolutional Neural Networks (※ 10M Youtube video)

Google uses 16K CPU cores for training 22-layers Deep neural network (※ GoogLeNet, 2014)

Baidu’s Artificial Intelligence Supercomputer Beats Google at Image Recognition
Deep Learning for NLP
Word Vector

- Represent words as vectors

\[ \text{expect} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \\ 0.487 \end{pmatrix} \]
Word Vector

- Distributional semantics: A word’s meaning is given by the words that frequently appear close-by.

- “You shall know a word by the company it keeps”

- Word2vec objective function (skip-grams)

\[
J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log P(w_{t+j} | w_t; \theta)
\]

Diagram: 
- \( P(w_{t-2} | w_t) \)
- \( P(w_{t-1} | w_t) \)
- Center word at position \( t \)
- \( P(w_{t+1} | w_t) \)
- \( P(w_{t+2} | w_t) \)

Outside context words in window of size 2

... problems turning into banking crises as ...

Center word at position \( t \)
Contextual word embedding

- A word’s **contextual embedding** must consider its context

\[ \varepsilon(\text{plays}) \]

\[ \varepsilon(\text{plays} \mid \text{the actor _ a show}) \]

GloVe

<table>
<thead>
<tr>
<th>the actor</th>
<th>plays</th>
<th>a show</th>
</tr>
</thead>
</table>

Some contextual method

<table>
<thead>
<tr>
<th>the actor</th>
<th>plays</th>
<th>a show</th>
</tr>
</thead>
</table>
ELMo: Embeddings from Language Model

- Multi-layer bidirectional LSTM language model

\[ R_k = \{ x_k^{LM}, \overrightarrow{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM} | j = 1, \ldots, L \} \]
\[ = \{ h_{k,j}^{LM} | j = 0, \ldots, L \}, \]
\[ h_{k,0}^{LM} = x_k^{LM} \text{ (token representation; GloVe)} \]
\[ h_{k,j}^{LM} = \overrightarrow{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM} \text{ (LSTM state)} \]

\[ \text{ELMo}^{\text{task}}_k = E(R_k; \Theta^{\text{task}}) = \gamma^{\text{task}} \sum_{j=0}^{L} s_j^{\text{task}} h_{k,j}^{LM} \]

\( \gamma^{\text{task}} \): scale (hyper-parameter)

\( s_j^{\text{task}} \): weight (learned)
ELMo for MRC

- ELMo as a word embedding
Transformer

- Parallel self-attention
  - Looks at self, and determines where to focus

**Transformer**

\[
\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V
\]

Q,K,V – vectors for every word, output attention summed up; head means different Q,K,V vector with different weights

Vaswani, Ashish, et al. "Attention is all you need." *NIPS 2017*
BERT: Bidirectional Encoder Representations from Transformers

• Training 1. Masked words prediction
  • 15% of words are [MASK]ed

*GELU: Gaussian error linear unit
• Training 2. Next sentence prediction
  • To understand texts more than a sentence

Input = [CLS] the man went to [MASK] store [SEP]
  he bought a gallon [MASK] milk [SEP]
Label = IsNext

Input = [CLS] the man [MASK] to the store [SEP]
  penguin [MASK] are flight less birds [SEP]
Label = NotNext
BERT: Bidirectional Encoder Representations from Transformers

- BERT as universal pre-trained model for NLP
  - BERT requires minimal additional layers and fine-tuning

In pre-training, optimize $L_1(u)$
$u$: Unlabeled dataset
$\Theta$: Model parameters

In fine-tuning, optimize $L_3(c)$
c: Labeled dataset
$\lambda$: Hyper-parameter weight
Sequence labeling

- Sequence labeling is the task of assigning a categorical label to each member of an observed sequences.

- Examples of sequence labeling
  - **Part-of-speech tagging** labels each word with a grammatical category
    - e.g. The | trees | are | ... $\rightarrow$ DT (determiner) | NNS (plural noun) | VBP (plural verb)
  - **Named entity recognition** locates and classifies named entity in text. It can be tackled by labeling each word with a named entity category
    - e.g. Barack | Obama | said | ... $\rightarrow$ B-PERSON | I-PERSON | O | ...
Bi-directional LSTM-CNNs-CRF

- **Bi-directional LSTMs** encode word embeddings and character representations

- **Conditional random fields** compute the distribution of output sequence
  - Viterbi algorithm is applied during training and decoding
  - The objective is the negative log-likelihood of the output sequence distribution

\[
p(y|z; W, b) = \frac{\prod_{i=1}^{n} \psi_i(y_{i-1}, y_i, z)}{\sum_{y' \in Y(z)} \prod_{i=1}^{n} \psi_i(y'_{i-1}, y'_i, z)}
\]

- **\(z\)**: input sequence
- **\(y\)**: output sequence
- **\(Y(z)\)**: a set of all possible output sequences when given the input sequence \(z\)

[Ma 2016]
Seq2Seq NMT via fixed-length representations

- Encoder RNN compresses input sequence into a fixed-length representation
- Decoder RNN produces output sequence from the representation
  - Each produced output token is fed into the next RNN’s input

[Sutskever 2014]
S2S NMT with attention mechanism

- It’s hard to encode all the information of an input sequence into a fixed-length representation
- We can focus important parts of input sequences for each decoding step by attention mechanism

[Bahdanau 2015]
Dependency parsing

- Dependency parsing is the task of extracting dependencies between head and dependent words from a sentence.

- A dependency is the arrow from a head to a dependent with a grammatical type called relation (e.g. nsubj).

- Dependencies show which words depend on (modify or are arguments of) which other words.

ROOT He has good control.
PRP VBZ JJ NN .

root nsubj
root dobj
root amod
punct
Neural transition-based dependency parsing

- Extract features from Stack and Buffer
  - lc/rc: leftmost/rightmost children
- Classify an action by neural networks
  - The objective is the negative log-likelihood of the action distribution

Feature extraction

Feedforward neural network-based action classifier

Softmax layer:
\[ p = \text{softmax}(W_2h) \]

Hidden layer:
\[ h = (W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1)^3 \]

Input layer: \([x^w, x^t, x^l]\)
Semantic parsing (weakly supervised)

- Semantic parsing is a task of mapping natural language to programs.
- We aim to develop semantic parsers without direct supervision on programs.

**Program**

\[
\text{(map (argmax (filter all-rows (\lambda (x) (= (string:country x) "greece")) index) number:year)}
\]

**Natural language**

“Greece held its last Summer Olympics in which year?”

**Context**

<table>
<thead>
<tr>
<th>Year</th>
<th>City</th>
<th>Country</th>
<th>Nations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1896</td>
<td>Athens</td>
<td>Greece</td>
<td>14</td>
</tr>
<tr>
<td>1900</td>
<td>Paris</td>
<td>France</td>
<td>24</td>
</tr>
<tr>
<td>1904</td>
<td>St. Louis</td>
<td>USA</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2004</td>
<td>Athens</td>
<td>Greece</td>
<td>201</td>
</tr>
<tr>
<td>2008</td>
<td>Beijing</td>
<td>China</td>
<td>204</td>
</tr>
<tr>
<td>2012</td>
<td>London</td>
<td>UK</td>
<td>204</td>
</tr>
</tbody>
</table>

**Denotation**

2004
• Top-down neural semantic parsing
  • Neural net generate derivation actions
  • Type system reduce search space (entity matching)

Neural Semantic Parsing with Type Constraints for Semi-Structured Tables, Krishnamurthy et al., EMNLP 2017

Iterative Search for Weakly Supervised Semantic Parsing, Dasigi et al., NAACL 2019
code generation by semantic parsing

\[
\begin{align*}
\text{stmt} & \mapsto \text{Expr} (\text{expr value}) \\
\text{expr} & \mapsto \text{Call} (\text{expr func}, \text{expr* args}, \\
& \text{keyword* keywords})
\end{align*}
\]

Abstract Syntax Description Language (ASDL) for Python

Code: sorted(my_list, reverse=True)

*parse tree to verify python code

\[
\begin{align*}
\text{stmt} &= \text{Select(agg_op? agg, idx column_idx,} \\
& \quad \text{cond_expr* conditions)} \\
\text{cond_expr} &= \text{Condition(cmp_op op, idx column_idx,} \\
& \quad \text{string value)} \\
\text{agg_op} &= \text{Max | Min | Count | Sum | Avg} \\
\text{cmp_op} &= \text{Equal | GreaterThan | LessThan | Other}
\end{align*}
\]

ASDL for SQL

[Yin 2017, Yin 2018, Rabinovich 2017]
Sentiment Analysis

- XLNet based classification

*XLNet = GPT (AR)+BERT(AE): permutation AR (Transformer-XL)
ConvNet for NLP

RNN for NLP - softmax is often only calculated at the last step

CNN for NLP
ConvNet for NLP

- CNN architecture for sentence classification

Text classification by CNN+LSTM

- CNN: advantages in selecting good features
- LSTM networks: good abilities of learning sequential data.
Style-based fake news detection
Wording, writing style

MRC-QA: SQuAD2.0

- Unanswerable question (negative example)
  - Relevant to the topic
  - Existence of plausible answers

**Article:** Endangered Species Act
**Paragraph:** “... Other legislation followed, including the Migratory Bird Conservation Act of 1929, a 1937 treaty prohibiting the hunting of right and gray whales, and the Bald Eagle Protection Act of 1940. These later laws had a low cost to society—the species were relatively rare—and little opposition was raised.”

**Question 1:** “Which laws faced significant opposition?”
**Plausible Answer:** later laws

**Question 2:** “What was the name of the 1937 treaty?”
**Plausible Answer:** Bald Eagle Protection Act

Figure 1: Two unanswerable questions written by crowdworkers, along with plausible (but incorrect) answers. Relevant keywords are shown in blue.
SQuAD2.0

- BERT – no answer prediction

answerable case

unanswerable case
HotpotQA

- HGN

**Framework**

- **Multiple span type prediction**
  - Start and end probability for each token is computed.

- **Single span type prediction**
  - Each token is classified by I,O tagging method, and multiple tokens classified as I are determined as answers.

- **Question Directed Graph Attention Network**
  - Relation information between numbers and entities in the passages is updated through graph reasoning (self-attention mechanism).

- **Answer type prediction**
  - The answers to the questions are categorized into 5 types.

**RoBERTa Encoder**

- **Mean pooling**
  - Question embedding used to direct the reasoning in graph reasoning.

**Answer Type Prediction (FFN)**

- **Single span type prediction**
  - 10 class classification problem (0~9), which covers about 97% counting problem in the DROP dataset.

- **Multiple span type prediction**
  - Only addition and subtraction operations are involved, each number is classified into one of (-1, 0, +1)

- **Classification for count type prediction**
  - 10 class classification problem (0~9), which covers about 97% counting problem in the DROP dataset.

- **Classification for add/sub type prediction**
  - Only addition and subtraction operations are involved, each number is classified into one of (-1, 0, +1)
Architecture

*Conditionally adaptive multi-task learning

CA-MTL (2020)

26 tasks: 9 GLUE tasks, 8 Super-GLUE Tasks, 6 MRQA tasks, etc - all tasks are trained jointly and evaluated from a single model

*9 GLUE task
MNLI:multi-genre natural language inference (entail, contradict, neutral)
QQP: quora question pairs (semantic equivalence)
QNLI: question answering NLI (context sentence contains answer to the question)
SSST-2: stanford sentiment treebank
cola: corpus of linguistic acceptability (grammaticality check)
sts-b: semantic textual similarity benchmark
mrpc: microsoft research paraphrase corpus (semantic equivalence)
rte: recognizing textual entailment (entail, not-entail)
WNLI: The Winograd Schema Challenge (common sense pronoun resolution task)

Figure 1: CA-MTLOBASE architecture first uses our uncertainty-based sampling algorithm to choose task data for batching. Then, the input tokens go through a frozen embedding layer, followed by a Conditional Alignment layer. The rest contains frozen BERT-based Transformer layers and trainable adapters.
Grammatical Error Correction (GEC)

• Neural machine translation approaches
  • Transformer and copy mechanism
  • 0.5880 $F_{0.5}$ score

\[
p = (1 - \text{acopy}) \cdot \text{pgen} + \text{acopy} \cdot \text{pcopy}
\]

make the model do more copying when the input sentence looks entirely correct.

Figure 1: Copy-Augmented Architecture.

Zhao et al., NAACL 2019
Multi-domain Task-oriented Dialogue

• The system interacts with user to help the user achieve his/her goal
  • e.g. Restaurant reservation, hotel reservation...

• Specific domains & specific goals
  • ⇔ Open domain (chit-chat)

• Multi-turns

• User goals are not limited to just one domain

To book a hotel

To book a restaurant near the hotel

To book a taxi from the hotel to the restaurant
Dataset: MultiWOZ

- Standard benchmark dataset for end-to-end task-oriented dialogue systems
  - Multi-domain dataset for task-oriented dialog system
  - Virtual conversations between a tourist and a guide in Cambridge
  - 7 domains: Hotel, restaurant, attraction, train, taxi, police, hospital
  - Number of dialog: 10,000 / Avg. number of turns per dialog: 13

- Evaluation in MultiWOZ
  - Inform rate
    - How many **entries provided** by the system meet goals?
  - Success rate
    - How many **user requests** are provided by the system?
  - BLEU
    - Similarity between generated response and true response in the dataset
User: I am planning a trip to town and want to sight see a bit.
System: There are 79 attractions in the city. Do you have a specific type of attraction?

User: I’d like a architecture.
System: How about [All Saints Church]? It has fantastic architecture and free entrance.

User: **Could I get the address?** I would also like an expensive place to eat around it.
System: **The address is Jesus Lane.** What type of food would you like to eat?

User: I'm in the mood for Turkish food.
System: Anatolia is an expensive Turkish restaurant in the city centre.

User: **Can I have the phone number?**
System: **The phone number is 01223362372.**

User: Thank you!
System: You're welcome, feel free to contact us about anything else you need.
traditional pipeline architecture

User

“Where do you want to eat?”

Natural Language Understanding

“I want to find a Chinese restaurant.”

Inform (cuisine="Chinese")

Dialog Manager

Request (location)

Dialog State

Dialog Policy

Knowledge Base

Query
Existing model: SOLOIST

- An auto-regressive model for training (Language modeling)
- Task 1: Predict dialogue state (slot-value)
- Task 2: Predict system response
- Task 3 (for Auxiliary loss)
  - Replace dialogue state or system response in input sequence into negative samples randomly
  - Then, predict input sequence is negative sample or not (binary classification)
- Jointly train by sum of 3 losses
DAMD (Domain Aware Multi-Decoder)

User utterance
  GRU encoder
  Delexicalization
  Attention
  User context

System Response
  GRU encoder
  Attention
  Response context

Belief
  GRU encoder
  Attention
  Belief context

Action
  GRU encoder
  Attention
  Action context

User context
  User context

Attention

Concat
  Pointer
  GRU decoder
  Softmax
  New belief
  DB

Concat
  GRU decoder
  Softmax
  New action

Concat
  GRU decoder
  Softmax
  New response
Architecture w/ domain state tracking

- NLU module is shared for DST, POL, and NLG
- Darker blocks mean previous turn
- DB result contains the number of matched entries for each domain

*for POL GRU training use two level: use CE loss for Supervised Learning (SL); use reward (success rate and correct system action rate) for 2nd level Reinforcement Learning (RL) training*
End-to-end ASR

Frontend (Preprocessing)
STFT, MEL

Audio Inputs

mel-spectrogram

Transformer Encoder

Speech Representation

Transformer Decoder (Attention)

CTC

Beam Search

Text Outputs

*Connectionist Temporal Classification (CTC)

dee scale: a perceptual scale of pitches judged by listeners to be equal in distance (log of frequency Hz)

Short-Time Fourier Transform (STFT)
**Tacotron2: Seq2seq with attention RNN + modified WaveNet**

- **wavenet**: invert the mel spectrogram feature representation into time-domain waveform samples (10 component mixture of logistic distributions (MoL) to generate 16-bit samples at 24 kHz)

*location-sensitive attention*: mitigating potential failure modes where some subsequences are repeated or ignored by the decoder

*Auto-regressive decoder* to generate mel spectrogram frame by frame; Previous frame input to pre-net & post-net (for residual)

*stop token* to stop generation
MelGan

Change WaveNet vocoder to:
- MelGan for faster inference

Loss: GAN objective + feature matching objective
(L1 distance between the discriminator feature maps of real and synthetic audio)

Figure 1: MelGAN model architecture. Each upsampling layer is a transposed convolution with kernel-size being twice of the stride (which is same as the upsampling ratio for the layer). 256x upsampling is done in 4 stages of 8x, 8x, 2x and 2x upsampling. Each residual dilated convolution stack has three layers with dilation 1, 3 and 9 with kernel-size 3, having a total receptive field of 27 timesteps. We use leaky-relu for activation. Each discriminator block has 4 strided convolution with stride 4. Further details can be found in the Appendix.
Dialog System for Counseling (TOD $\rightarrow$ Open-ended)

- Paraphrasing
- Reflect Feeling

- Correct Information
- Logical Answering

- Questioning
- Show Interest

- Listening
- Memorizing

Empathy

Motivating

Sensibleness

Attending
User: How are things?
Meena: They’re good. How about you?
User: I’m trying to decide what to watch
Meena: What are some of your favorite shows?
User: I like the Good Place
Meena: I haven’t watched that. My favorite shows are Supernatural, and Star trek: TNG.
User: What is TNG?
Meena: The Next Generation

Frederick Parker-Rhodes (21 March 1914 - 21 November 1987) was an English linguist, plant pathologist, computer scientist, mathematician, mystic, and mycologist.
Switch-GPT2 (Chen et al., 2020)

\[ L = L_c + \lambda \sum_{w_j \in m \atop m \in \{V_i\}} (1 - P^j_{copy}) \]

*Lc original loss, \( V_i \) slot-value, \( w_j \) target word in slot-value for copy
Summary

- NLP enjoys rapid progress in the last 10 years due to deep learning.
- Even more rapid progress in the last few years due to larger models, better usage of unlabeled data.
- NLP is reaching the point of having big social impact, making issues like bias and security increasingly important.
- Big model, big computation resources, huge training times are problematic, need to focus more light way of doing NLP (even embedded model).