Week 4
- Visual Q&A, NLP Q&A

2019.06.01

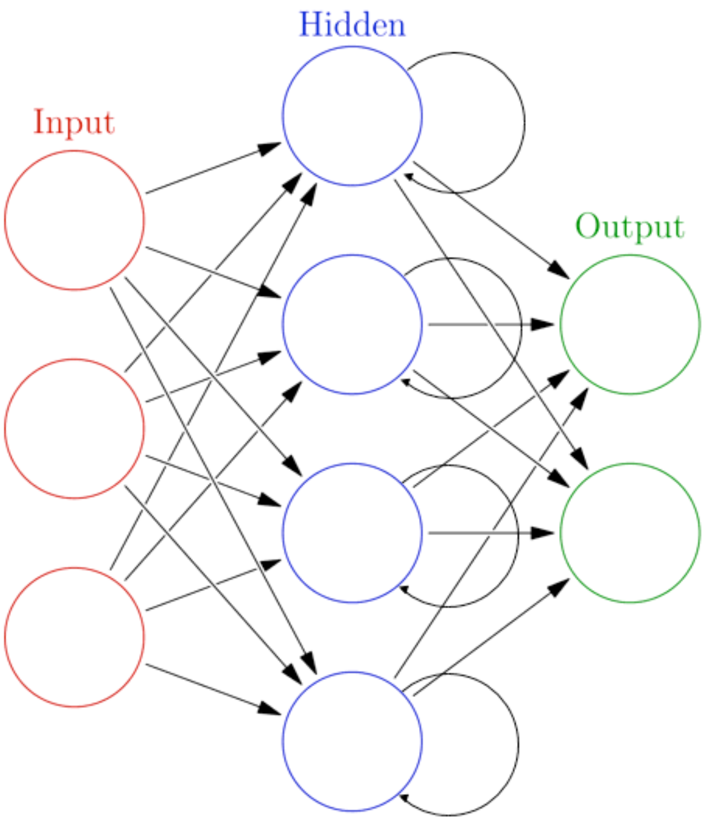
Solaris
(http://solarisailab.com)

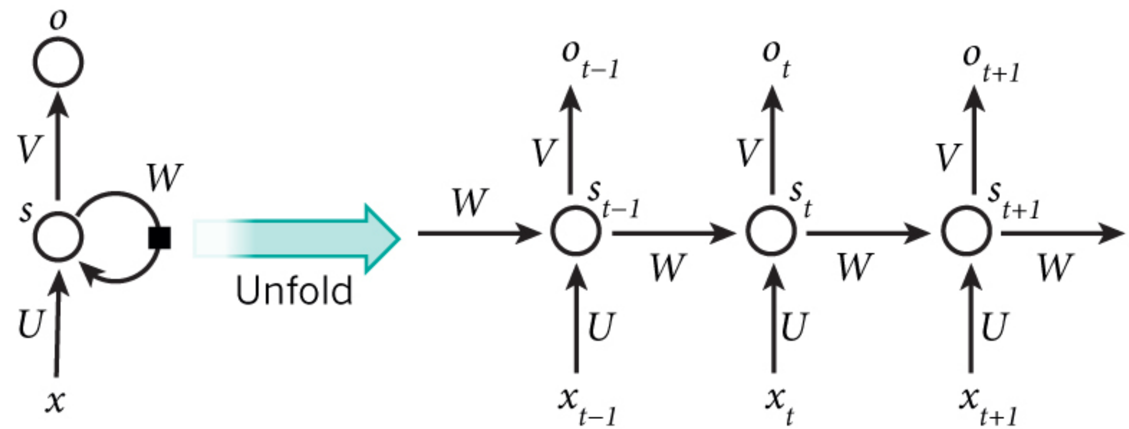
Week3의 학습목표

- **3강의 학습목표 :**

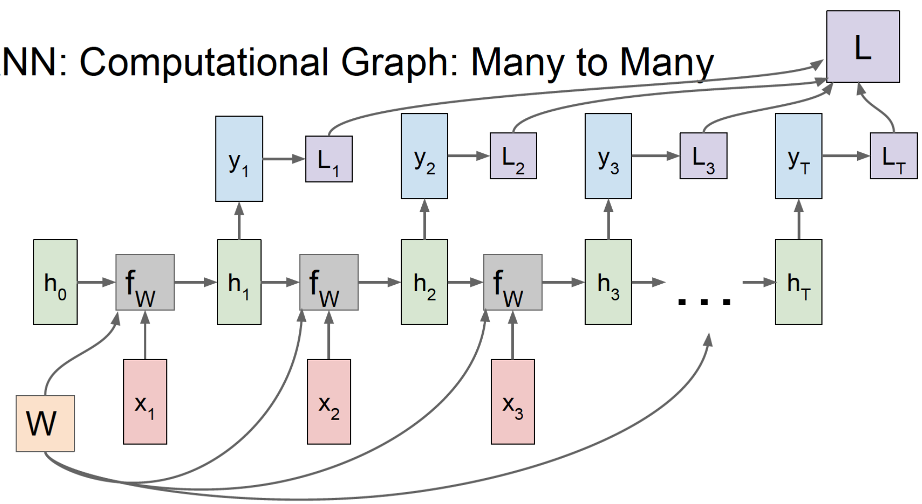
1. RNN, LSTM, GRU의 Architecture와 디자인 철학을 이해한다.
2. Regularization의 개념과 Regularization을 수행할수 있는 기법들(e.g. Regularization Term, Batch Normalization, ...)을 학습한다.
3. TensorFlow를 이용해서 Language Modelling을 위한 RNN을 구현해보자.

Week 3 복습 – Recurrent Neural Networks

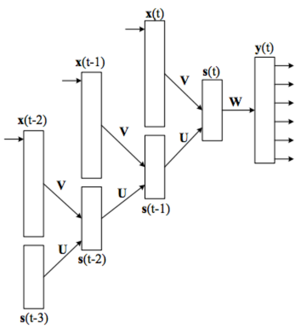




RNN: Computational Graph: Many to Many



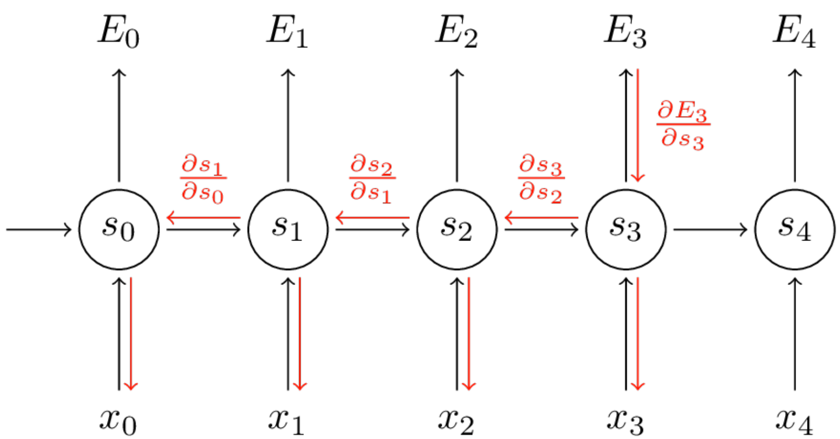
Week 3 복습 – Truncated Backpropagation Through Time(BPTT)



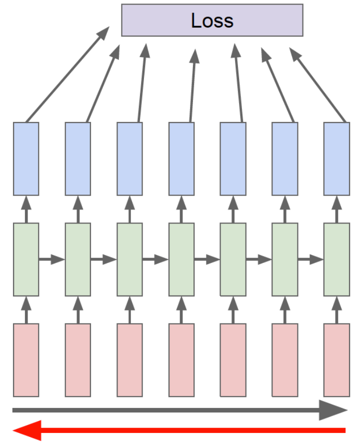
$$W(t+1) = W(t) + \eta s(t) e_o(t)^T$$

$$V(t+1) = V(t) + \eta \sum_{z=0}^{T} x(t-z) e_h(t-z)^T$$

$$U(t+1) = U(t) + \eta \sum_{z=0} s(t-z-1) e_h(t-z)^T$$

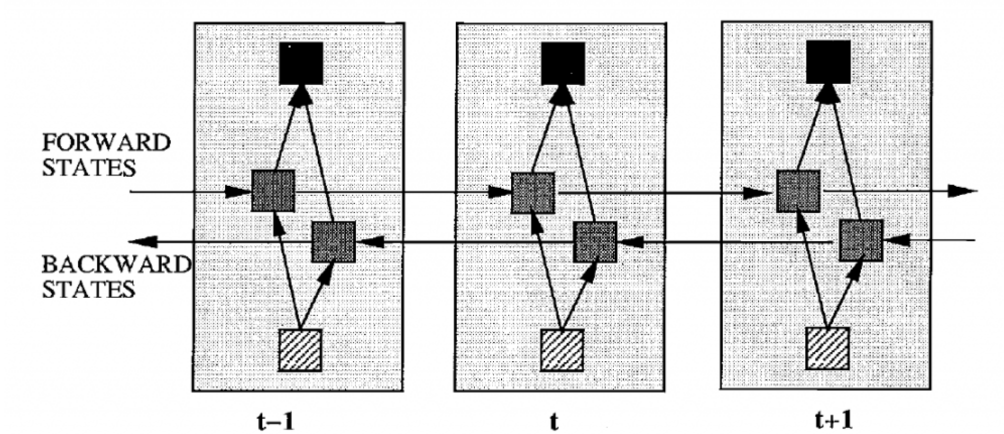


Truncated Backpropagation through time



Run forward and backward through chunks of the sequence instead of whole sequence

Week 3 복습 – Bidirectional RNNs



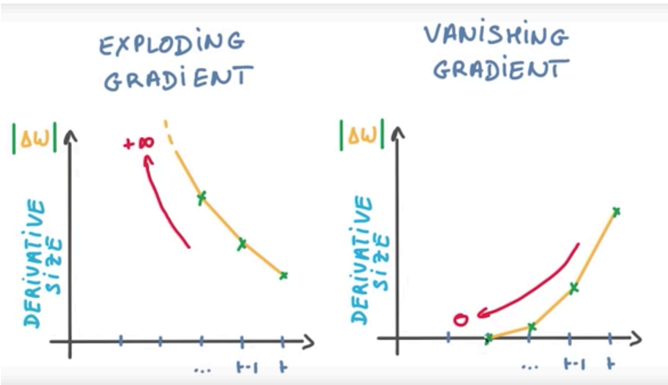
Algorithm 1 BRNNs Forward Pass

1: **for** t=1 to T **do**
   Do forward pass for the forward hidden layer, storing activations at each timestep
2: **end for**
3: **for** t=T to 1 **do**
   Do forward pass for the backward hidden layer, storing activations at each timestep
4: **end for**
5: **for** t=1 to T **do**
   Do forward pass for the output layer, using the stored activations from both hidden layers
6: **end for**

Algorithm 1 BRNNs Backward Pass

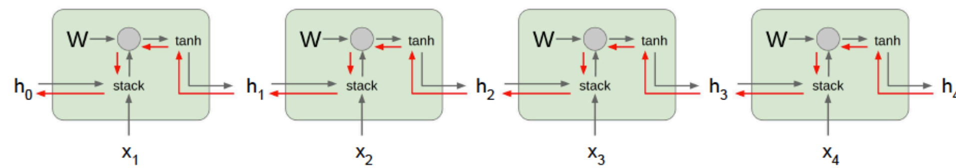
1: **for** t=T to 1 **do**
   Do BPTT backward pass for the forward hidden layer, using the stored δ terms from the output layer
2: **end for**
3: **for** t=T to 1 **do**
   Do BPTT backward pass for the forward hidden layer, using the stored δ terms from the output layer
4: **end for**
5: **for** t=1 to T **do**
   Do BPTT backward pass for the backward hidden layer, using the stored δ terms from the output layer
6: **end for**

Week 3 복습 – Exploding & Vanishing Gradient Problem



Vanilla RNN Gradient Flow

Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



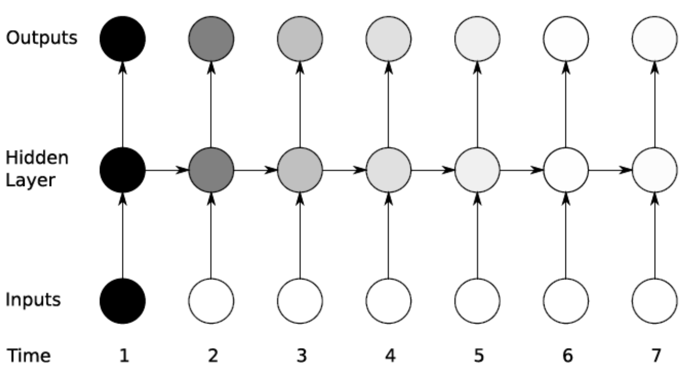
Computing gradient of h_0 involves many factors of W (and repeated tanh)

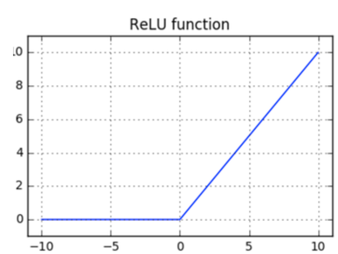
Largest singular value > 1: **Exploding gradients**

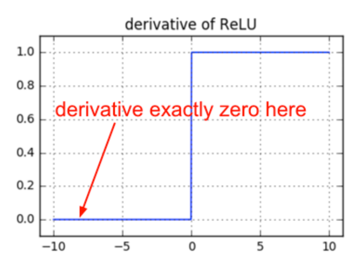
Largest singular value < 1: **Vanishing gradients**

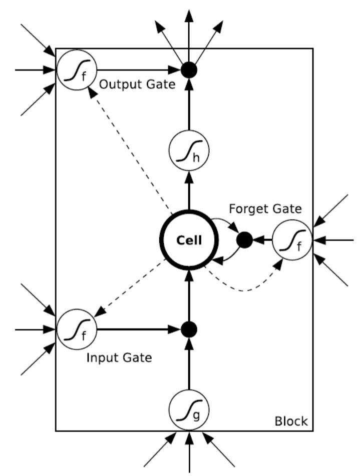
Gradient clipping: Scale gradient if its norm is too big

```
grad_norm = np.sum(grad * grad)
if grad_norm > threshold:
  grad *= (threshold / grad_norm)
```

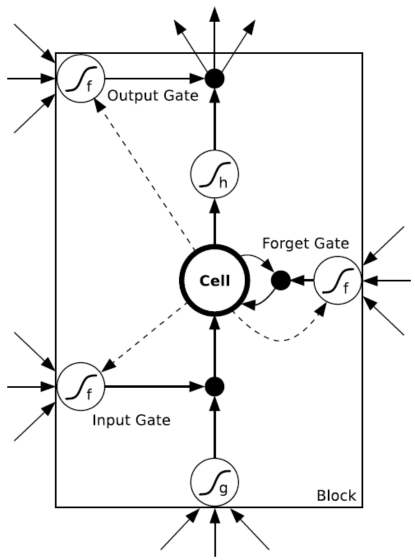


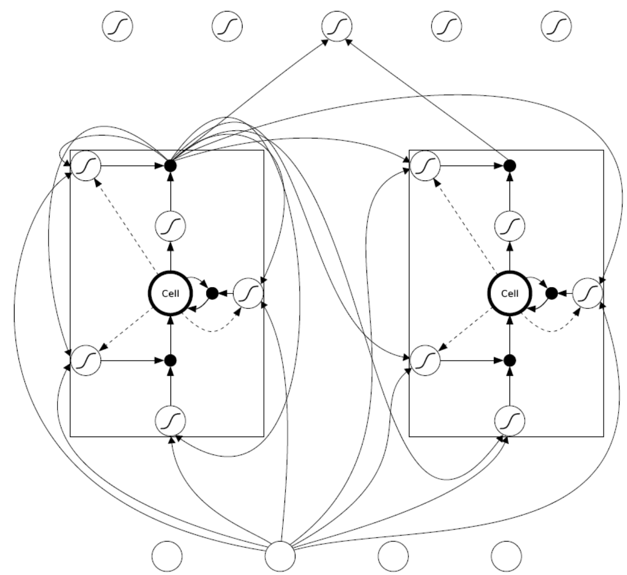


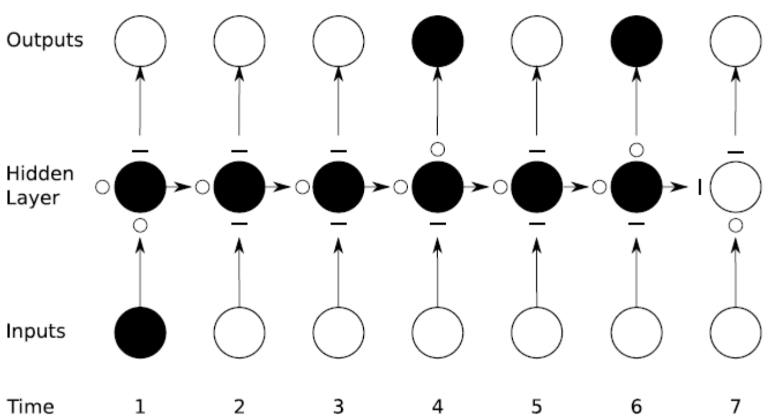


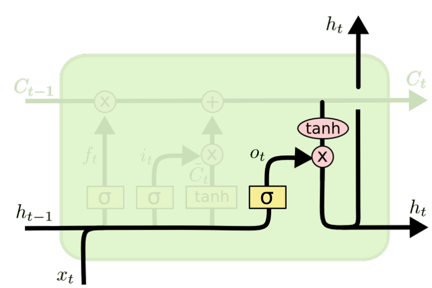


Week 3 복습 – Long-Short Term(LSTM) Memory Networks





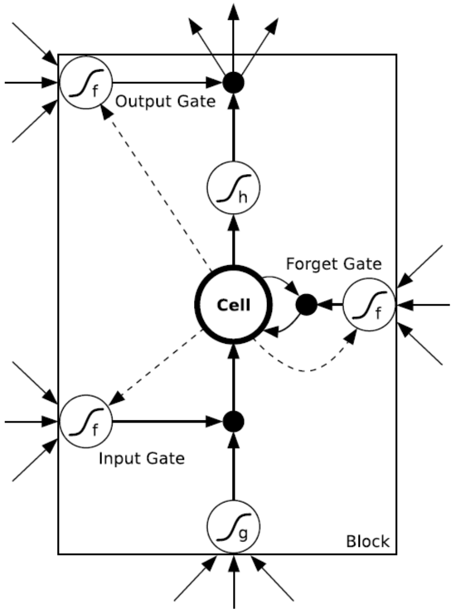




$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Week 3 복습 – Long-Short Term(LSTM) Memory Networks

- 각각의 Gates에서 일어나는 연산은 아래와 같다. (f, g, h는 sigmoid와 같은 activation function이다.)



f: Forget gate, Whether to erase cell
i: Input gate, whether to write to cell
g: Gate gate (?), How much to write to cell
o: Output gate, How much to reveal cell

Input Gates에서의 연산

$$a_I^t = \sum_{i=1}^{I} w_{iI} x_i^t + \sum_{h=1}^{H} w_{hI} b_h^{t-1} + \sum_{c=1}^{C} w_{cI} s_c^{t-1}$$

$$b_I^t = f(a_I^t)$$

Forget Gates에서의 연산

$$a_F^t = \sum_{i=1}^{I} w_{iF} x_i^t + \sum_{h=1}^{H} w_{hF} b_h^{t-1} + \sum_{c=1}^{C} w_{cF} s_c^{t-1}$$

$$b_F^t = f(a_F^t)$$

Cells에서의 연산

$$a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \sum_{h=1}^{H} w_{hc} b_h^{t-1}$$

$$s_c^t = b_F^t s_c^{t-1} + b_I^t g(a_c^t)$$

Output Gates에서의 연산

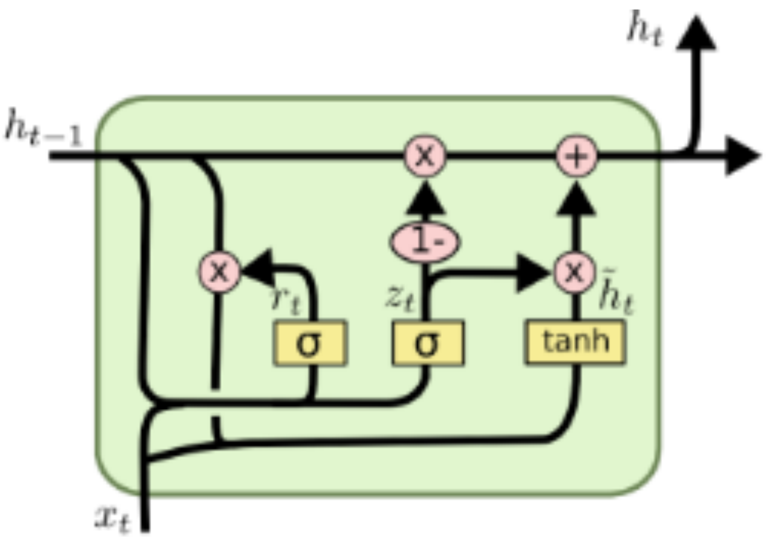
$$a_O^t = \sum_{i=1}^{I} w_{iO} x_i^t + \sum_{h=1}^{H} w_{hO} b_h^{t-1} + \sum_{c=1}^{C} w_{cO} s_c^t$$

$$b_O^t = f(a_O^t)$$

Cells Outputs(Memory Block의 Output에서의 연산)

$$b_c^t = b_O^t h(s_c^t)$$

Week 3 복습 – Gate Recurrent Unit(GRU)



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

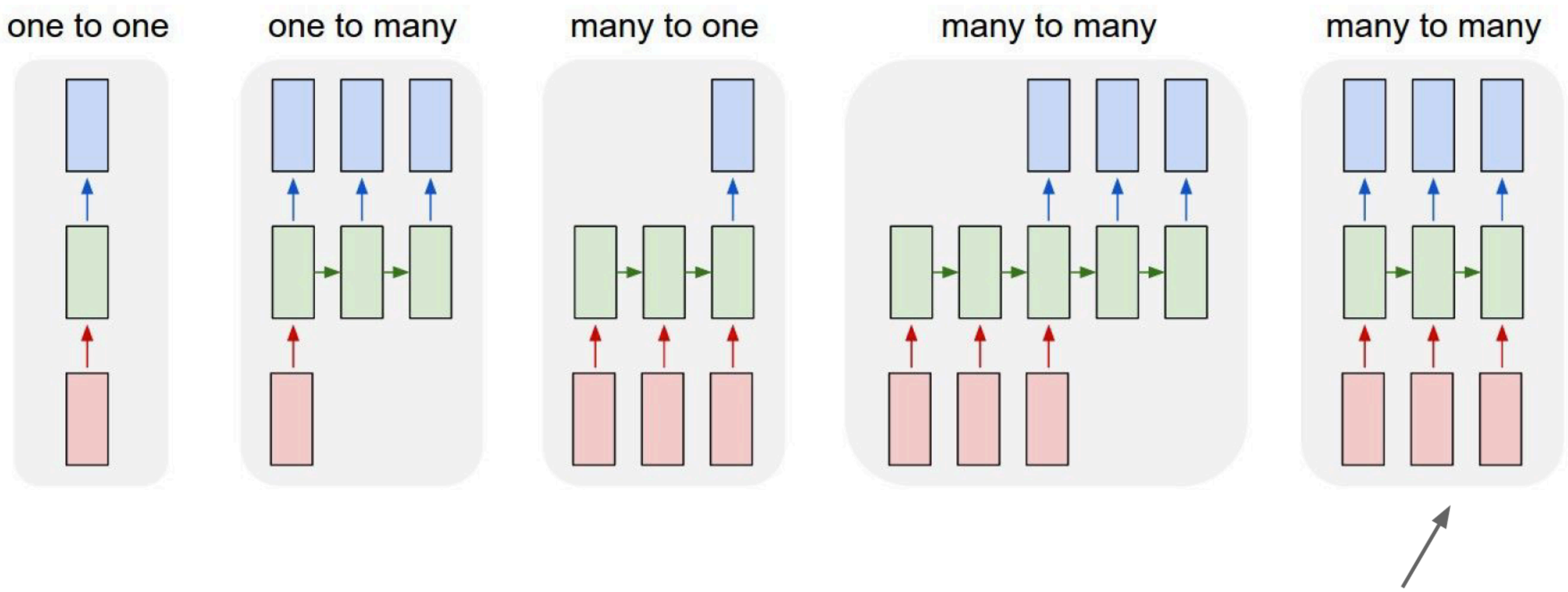
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

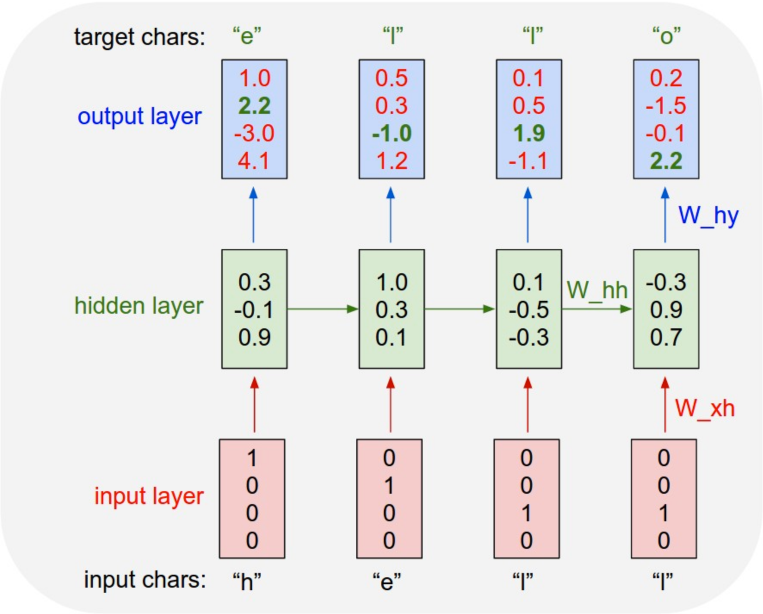
Week 3 복습 – RNNs의 다양한 응용형태

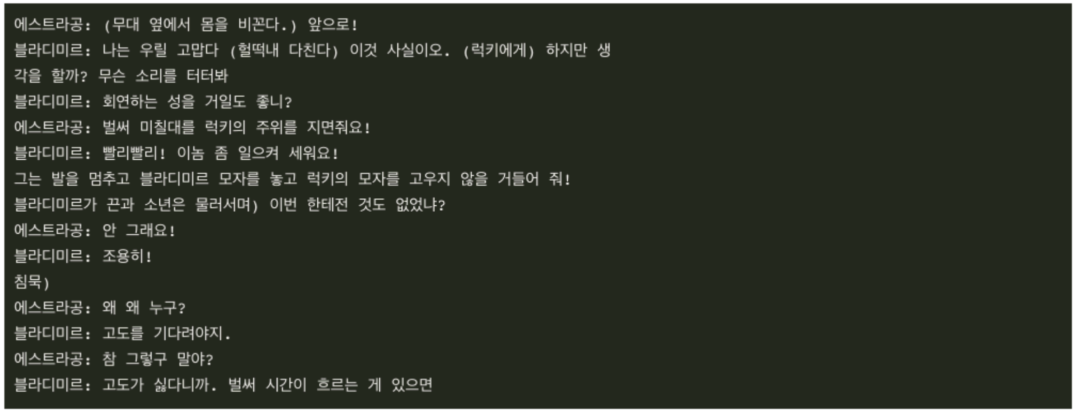
Recurrent Neural Networks: Process Sequences



e.g. **Video classification on frame level**

Week 3 복습 – Char-RNN(Character-Level Language Modelling)



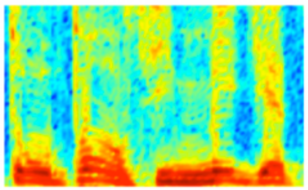


에스트라공: (무대 옆에서 몸을 비꼰다.) 앞으로!
블라디미르: 나는 우릴 고맙다 (헐떡내 다친다) 이것 사실이오. (럭키에게) 하지만 생
각을 할까? 무슨 소리를 터터봐
블라디미르: 회연하는 성을 거일도 좋니?
에스트라공: 벌써 미칠대를 럭키의 주위를 지면줘요!
블라디미르: 빨리빨리! 이놈 좀 일으켜 세워요!
그는 발을 멈추고 블라디미르 모자를 놓고 럭키의 모자를 고우지 않을 거들어 줘!
블라디미르가 끈과 소년은 물러서며) 이번 한테전 것도 없었냐?
에스트라공: 안 그래요!
블라디미르: 조용히!
침묵)
에스트라공: 왜 왜 누구?
블라디미르: 고도를 기다려야지.
에스트라공: 참 그렇구 말야?
블라디미르: 고도가 싫다니까. 벌써 시간이 흐르는 게 있으면

그림 4 – 사무엘 베케트의 희곡 “고도를 기다리며”를 데이터셋으로 학습을 진행한 이후에 샘플링한 결과

Week 3 복습 – Embedding Vectors

AUDIO | **IMAGES** | **TEXT**



Audio Spectrogram

DENSE

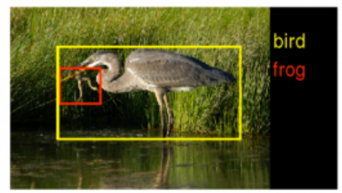


Image pixels

DENSE

Word, context, or document vectors

SPARSE

An embedding is a mapping from discrete objects, such as words, to vectors of real numbers. For example, a 300-dimensional embedding for English words could include:

```
blue:  (0.01359, 0.00075997, 0.24608, ..., -0.2524, 1.0048, 0.06259)
blues:  (0.01396, 0.11887, -0.48963, ..., 0.033483, -0.10007, 0.1158)
orange:  (-0.24776, -0.12359, 0.20986, ..., 0.079717, 0.23865, -0.014213)
oranges:  (-0.35609, 0.21854, 0.080944, ..., -0.35413, 0.38511, -0.070976)
```

Week 3 복습 – TensorFlow Tricks

Exploding Gradients

Clip gradients with tf.clip_by_global_norm

```
gradients = tf.gradients(cost, tf.trainable_variables())
# take gradients of cosst w.r.t. all trainable variables

clipped_gradients, _ = tf.clip_by_global_norm(gradients, max_grad_norm)
# clip the gradients by a pre-defined max norm

optimizer = tf.train.AdamOptimizer(learning_rate)
train_op = optimizer.apply_gradients(zip(gradients, trainables))
# add the clipped gradients to the optimizer
```

49

Embedding Lookup

$$[0 \quad 0 \quad 0 \quad \boxed{1} \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

```
tf.nn.embedding_lookup(params, ids, partition_strategy='mod', name=None,
    validate_indices=True, max_norm=None)
```

Illustration by Chris McCormick

21

Week4의 학습목표

- **4강의 학습목표 :**

1. Visual Q&A의 개념을 이해하고, TensorFlow를 이용해서 Visual Q&A 문제를 위한 CNN+RNN을 모델을 구현해본다.
2. NLP Q&A의 개념과 End-to-End Memory Networks의 이개념을 해하고, TensorFlow를 이용해서 NLP Q&A 문제를 위한 End-to-End Memory Networks을 모델을 구현해본다.
3. 다양한 Optimization 알고리즘의 개념을 이해한다.

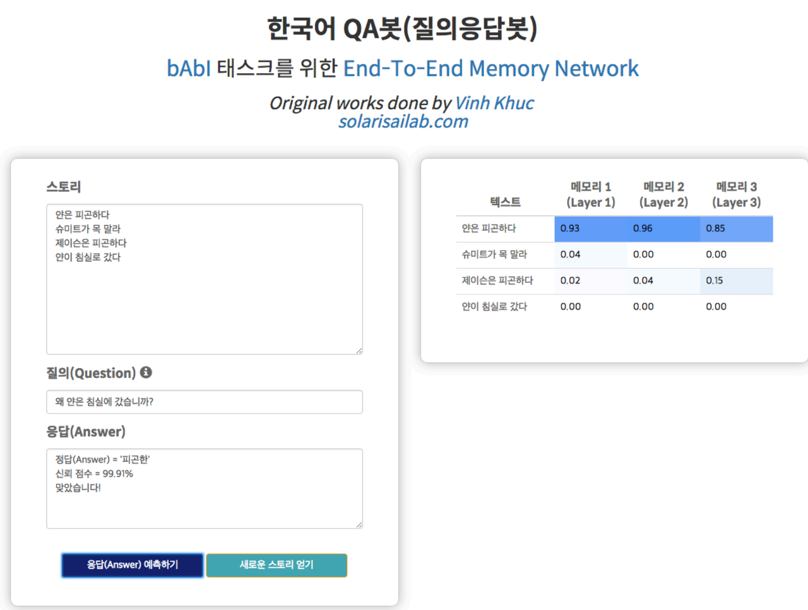
Outline

- NLP Q&A, Visual Q&A 문제소개
- 논문 리뷰 - Exploring Models and Data for Image Question Answering
- TensorFlow를 이용한 Exploring Models and Data for Image Question Answering 구현
- 논문 리뷰 - End-To-End Memory Networks
- TensorFlow를 이용한 End-To-End Memory Networks 구현
- Optimization Method 정리

NLP Q&A 문제소개

- Natural Language Processing(NPL) Q&A – 자연어로 표현된 질문(**Q**uestion)에 대한 적절한 답(**A**nswer)을 구하는 문제

- "왜 얀은 침실로 갔습니까?"(Question) -> "피곤해서"(Answer)

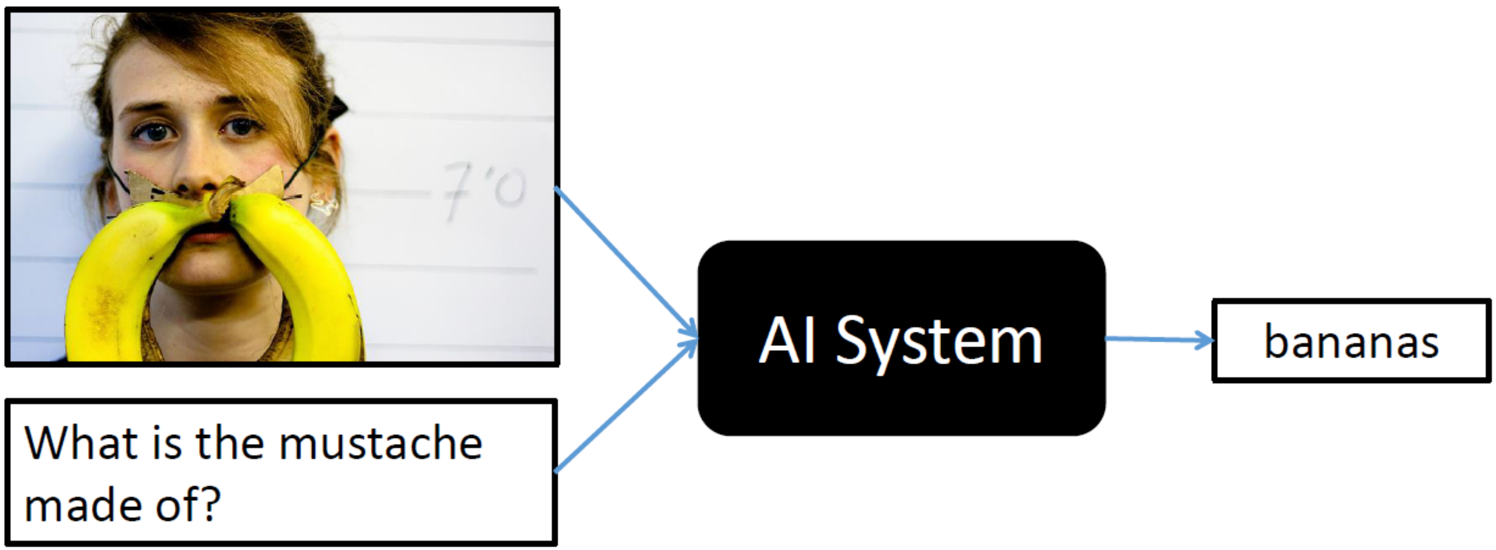


http://solaris33.pythonanywhere.com/

Visual Q&A 문제소개

- Visual Q&A – Image와 질문(Question)이 주어지면 이에 대한 적절한 답(Answer)을 구하는 문제

- http://www.visualqa.org/



Visual Q&A 데모

- https://vqa.cloudcv.org/

How it works

1. You upload an image.
2. Your request is sent to our servers with GPUs courtesy NVIDIA.
3. Our servers run our deep-learning based algorithm.
4. Results and updates are shown in real-time.

Result for Visual Question Answering

What is the color of the cat?

Submit

Predicted top-5 answers with confidence:

gray	91.993%
black	6.924%
brown	0.519%
white	0.354%
blue	0.082%

Credits

Code for VQA Model

Built by @deshraj

논문 리뷰 - Exploring Models and Data for Image Question Answering

- Ren, Mengye, Ryan Kiros, and Richard Zemel. "Exploring models and data for image question answering", NIPS 2015
- http://papers.nips.cc/paper/5640-exploring-models-and-data-for-image-question-answering.pdf
- Visual Q&A를 문제를 위한 Architecture 제안

DAQUAR 1553
What is there in front of the sofa?
Ground truth: table
IMG+BOW: table (0.74)
2-VIS+BLSTM: table (0.88)
LSTM: chair (0.47)

COCOQA 5078
How many leftover donuts is the red bicycle holding?
Ground truth: three
IMG+BOW: two (0.51)
2-VIS+BLSTM: three (0.27)
BOW: one (0.29)

COCOQA 1238
What is the color of the tee-shirt?
Ground truth: blue
IMG+BOW: blue (0.31)
2-VIS+BLSTM: orange (0.43)
BOW: green (0.38)

COCOQA 26088
Where is the gray cat sitting?
Ground truth: window
IMG+BOW: window (0.78)
2-VIS+BLSTM: window (0.68)
BOW: suitcase (0.31)

Figure 1: Sample questions and responses of a variety of models. Correct answers are in green and incorrect in red. The numbers in parentheses are the probabilities assigned to the top-ranked answer by the given model. The leftmost example is from the DAQUAR dataset, and the others are from our new COCO-QA dataset.

COCOQA 33827
What is the color of the cat?
Ground truth: black
IMG+BOW: black (0.55)
2-VIS+LSTM: black (0.73)
BOW: gray (0.40)

COCOQA 33827a
What is the color of the couch?
Ground truth: red
IMG+BOW: red (0.65)
2-VIS+LSTM: black (0.44)
BOW: red (0.39)

DAQUAR 1522
How many chairs are there?
Ground truth: two
IMG+BOW: four (0.24)
2-VIS+BLSTM: one (0.29)
LSTM: four (0.19)

DAQUAR 1520
How many shelves are there?
Ground truth: three
IMG+BOW: three (0.25)
2-VIS+BLSTM: two (0.48)
LSTM: two (0.21)

COCOQA 14855
Where are the ripe bananas sitting?
Ground truth: basket
IMG+BOW: basket (0.97)
2-VIS+BLSTM: basket (0.58)
BOW: bowl (0.48)

COCOQA 14855a
What are in the basket?
Ground truth: bananas
IMG+BOW: bananas (0.98)
2-VIS+BLSTM: bananas (0.68)
BOW: bananas (0.14)

DAQUAR 585
What is the object on the chair?
Ground truth: pillow
IMG+BOW: clothes (0.37)
2-VIS+BLSTM: pillow (0.65)
LSTM: clothes (0.40)

DAQUAR 585a
Where is the pillow found?
Ground truth: chair
IMG+BOW: bed (0.13)
2-VIS+BLSTM: chair (0.17)
LSTM: cabinet (0.79)

Figure 3: Sample questions and responses of our system

VIS(Visual Sematic embedding) + LSTM Architecture

- VIS(**VI**sual **S**ematic embedding) + LSTM architecture
- **핵심 아이디어** :
- 1. CNN + RNN(LSTM)을 이용한 end-to-end Learning
- 2. Answer를 미리 정의된 단어 corpus 중에서 하나의 단어를 선택하는 **classification 문제**로 치환함 **(따라서 하나의 단어로 이루어진 정답만 맞출수 있는 한계를 가짐)**

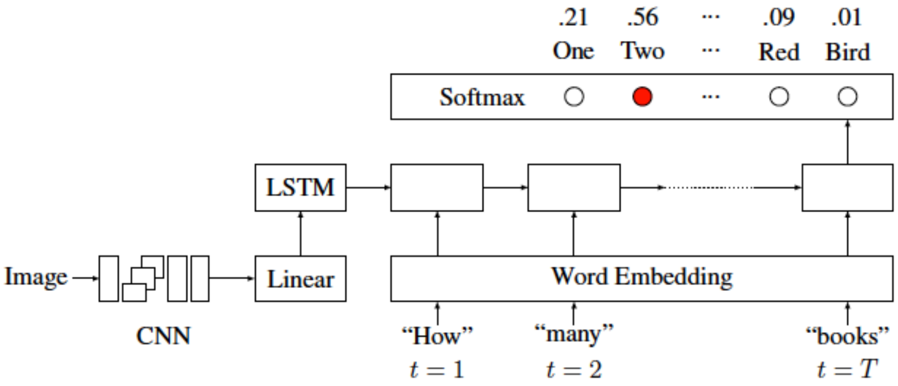


Figure 2: VIS+LSTM Model

VIS(Visual Sematic embedding) + LSTM Architecture

- **Model :** LSTM Networks

- **Input:**

1. Image를 **VGGNet**에 넣고 나온 Feature (4096 dimension)-2014년 ImageNet Challenge에 대해 학습된 19-layer VGGNet을 사용하고, 학습과정에서 CNN 부분은 따로 업데이트 되지 않음(frozen)-를 300 또는 500 dimension vector로 mapping 함

2. Question을 단어(Word) 단위로 쪼개고 **Word Embedding**을 이용해서 vector로 변환

- **Output :**

1. 이미지와 질문에 적절한 단어(Word)

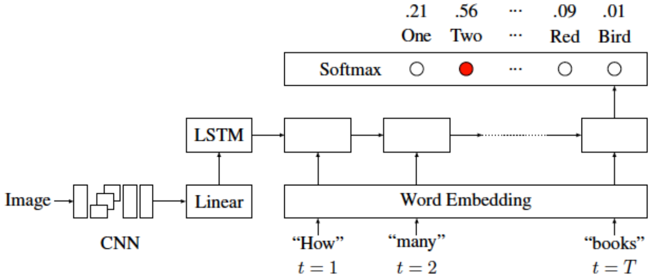


Figure 2: VIS+LSTM Model

Visual Q&A Dataset 1 - DAQUAR

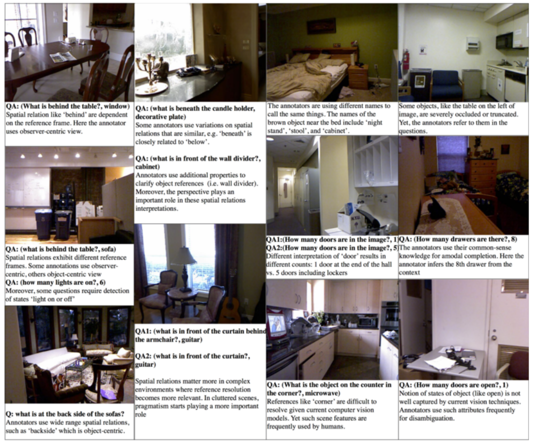
- **DA**taset for **QU**estion Answering on **R**eal-world images (DAQUAR)

- Dataset 구성 :

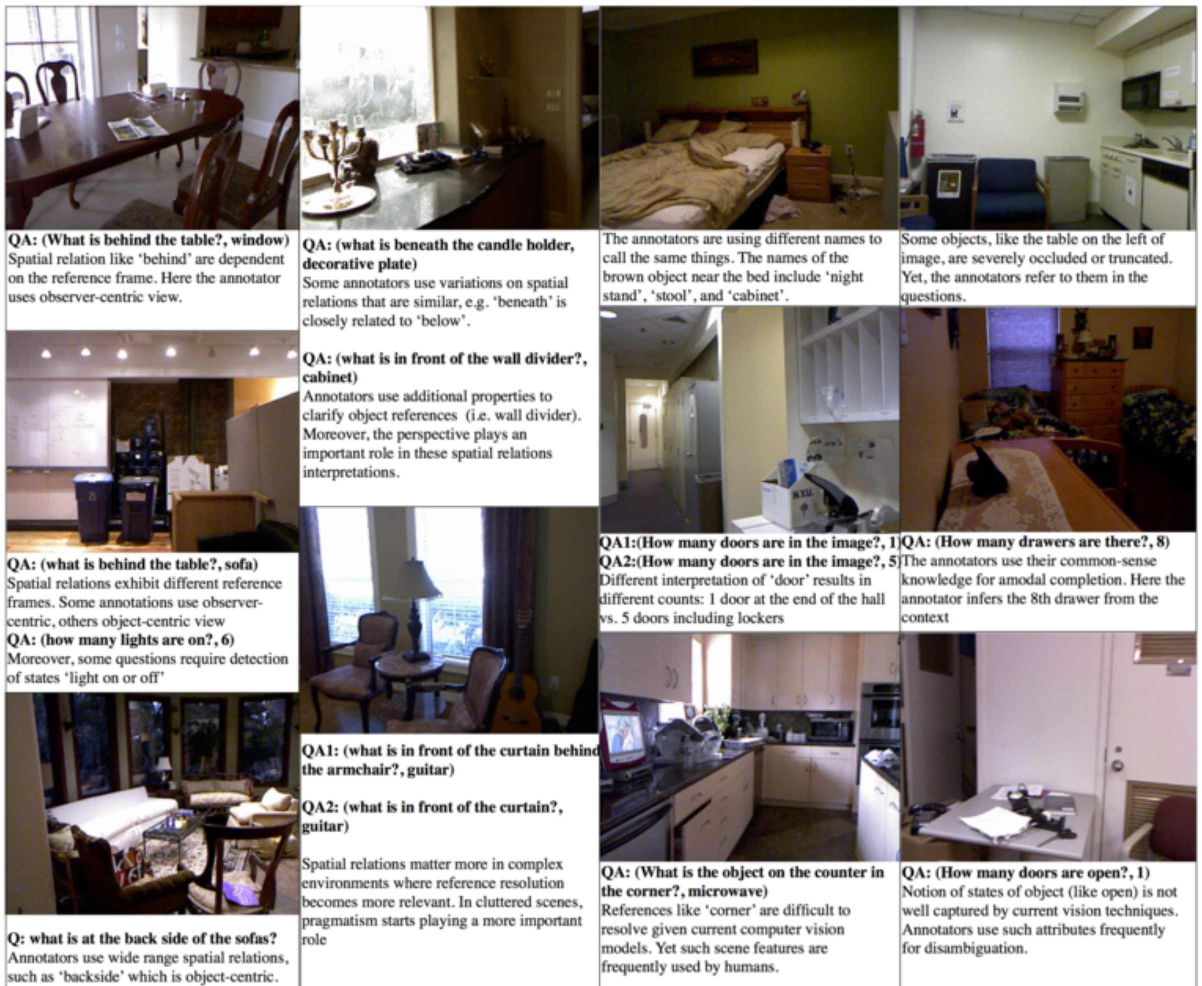
Image : 모두 NYU depth v2 dataset(indoor scene)에서 가져 옴.

Question : 보통 3가지 형태의 질문으로 구성 됨(object **type**, object **color**, **number** of objects)

Answer : 총 894개의 Class로 구성됨(e.g. table, blue, 3, plastic_cup_of_coffee, …), 37개의 class로 구성된 reduced version도 존재



DAQUAR Examples



Visual Q&A Dataset 2 – MS-COCO

- **M**icro**S**oft-**C**ommon **O**bjects in **CO**ntext(MS-COCO) Dataset
- 직접적으로 Q&A 형태를 제공하진 않고, Image Captioning을 제공함-
논문에서 Image Captioning을 Q&A form으로 바꾸는 방법을 제안-

The man at bat readies to swing at the pitch while the umpire looks on.

A large bus sitting next to a very tall building.

COCO-QA Dataset

- 논문의 저자들이 MS-COCO Dataset의 Image Captioning을 Q&A form으로 변환한 데이터셋
- Question은 다음의 네가지 형태로 구성된다.

1. Object
2. Number
3. Color
4. Location

Figure 1: Sample questions and responses of a variety of models. Correct answers are in green and incorrect in red. The numbers in parentheses are the probabilities assigned to the top-ranked answer by the given model. The leftmost example is from the DAQUAR dataset, and the others are from our new COCO-QA dataset.

- http://www.cs.toronto.edu/~mren/imageqa/data/cocoqa/

COCO-QA Dataset 구성

- 가장 긴 질문(Question)은 55글자이고 평균은 9.65글자
- 가장 흔한 답변들(common answers)은 “two”(3116, 2.65%), “white” (2851, 2.42%), “red”(2443, 2.08%)”
- 가장 적게 등장한(least common) 답변들은 “eagle”(25, 0.02%), “tram”(25, 0.02%), “sofa”(25, 0.02%)
- 중간 정도로(median) 등장한 답변들은 “bed”(867, 0.737%)

Table 1: COCO-QA question type break-down

CATEGORY	TRAIN	%	TEST	%
OBJECT	54992	69.84%	27206	69.85%
NUMBER	5885	7.47%	2755	7.07%
COLOR	13059	16.59%	6509	16.71%
LOCATION	4800	6.10%	2478	6.36%
TOTAL	78736	100.00%	38948	100.00%

Experiment Result

- 논문에서 제안한 **2+VIS+BLSTM 모델**(CNN Feature를 앞, 뒤에 2번 넣고 Bidirectional LSTM을 사용)이 가장 좋은 성능을 보여줌

- 4096 Dimension의 CNN Feature를 500 Dimension으로 Reduction하면서 정보 손실이 발생 함, IMG + BOW 모델에 4096 Dimension의 CNN Feature대신 500 Dimension의 CNN Feature를 넣으면 약 **0.48**의 정확도가 나와서 2+VIS+LSTM 모델링 성능이 더 좋음 (COCO-QA 데이터셋)

Table 2: DAQUAR and COCO-QA results

	DAQUAR			COCO-QA		
	Acc.	WUPS 0.9	WUPS 0.0	Acc.	WUPS 0.9	WUPS 0.0
MULTI-WORLD [32]	0.1273	0.1810	0.5147	-	-	-
GUESS	0.1824	0.2965	0.7759	0.0730	0.1837	0.7413
BOW	0.3267	0.4319	0.8130	0.3752	0.4854	0.8278
LSTM	0.3273	0.4350	0.8162	0.3676	0.4758	0.8234
IMG	-	-	-	0.4302	0.5864	0.8585
IMG+PRIOR	-	-	-	0.4466	0.6020	0.8624
K-NN (K=31, 13)	0.3185	0.4242	0.8063	0.4496	0.5698	0.8557
IMG+BOW	0.3417	0.4499	0.8148	**0.5592**	**0.6678**	**0.8899**
VIS+LSTM	0.3441	0.4605	**0.8223**	0.5331	0.6391	0.8825
ASK-NEURON [14]	0.3468	0.4076	0.7954	-	-	-
2-VIS+BLSTM	**0.3578**	**0.4683**	0.8215	0.5509	0.6534	0.8864
FULL	**0.3694**	**0.4815**	**0.8268**	**0.5784**	**0.6790**	**0.8952**
HUMAN	0.6027	0.6104	0.7896	-	-	-

Experiment Result

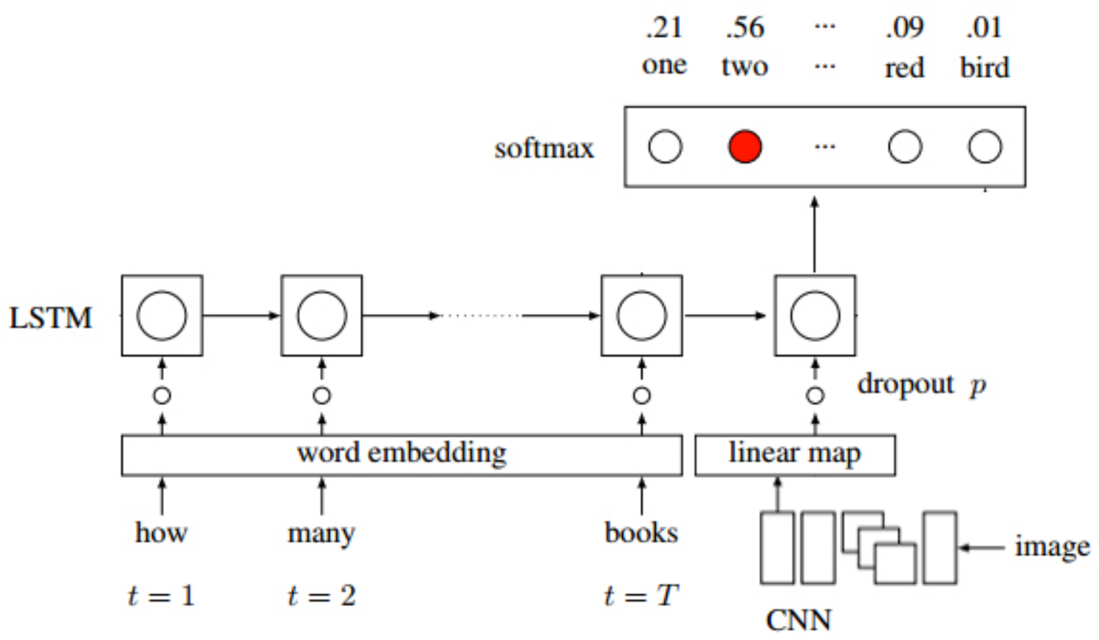
- COCO-QA 데이터셋의 카테고리별 정확도

Table 3: COCO-QA accuracy per category

	OBJECT	NUMBER	COLOR	LOCATION
GUESS	0.0239	0.3606	0.1457	0.0908
BOW	0.3727	0.4356	0.3475	0.4084
LSTM	0.3587	0.4534	0.3626	0.3842
IMG	0.4073	0.2926	0.4268	0.4419
IMG+PRIOR	-	0.3739	0.4899	0.4451
K-NN	0.4799	0.3699	0.3723	0.4080
IMG+BOW	**0.5866**	0.4410	**0.5196**	**0.4939**
VIS+LSTM	0.5653	**0.4610**	0.4587	0.4552
2-VIS+BLSTM	0.5817	0.4479	0.4953	0.4734
FULL	**0.6108**	**0.4766**	0.5148	**0.5028**

TensorFlow를 이용한 Exploring Models and Data for Image Question Answering 논문 구현

- TensorFlow 구현은 논문과 구조가 약간 다름**(CNN Feature들을 맨앞이 아니라 맨 뒤에 넣음)**
- https://github.com/solaris33/dl_cv_tensorflow_10weeks/tree/master/week4/neural-vqa-tensorflow



TensorFlow를 이용한 Exploring Models and Data for Image Question Answering 논문 구현

- **Input:**

1. Image를 **VGGNet**에 넣고 나온 Feature (4096 dimension)-2014년 ImageNet Challenge에 대해 학습된 19-layer VGGNet을 사용하고, 학습과정에서 CNN 부분은 따로 업데이트 되지 않음(frozen)-를 300 또는 500 dimension vector로 mapping 함

2. Question을 단어(Word) 단위로 쪼개고 **Word Embedding**을 이용해서 vector로 변환

- **Output :**

1. 이미지와 질문에 적절한 단어(Word)

```
fc7_features = utils.extract_fc7_features(args.image_path, join(args.data_dir, 'vgg16.tfmodel'))
question_ids = np.zeros((1, vocab_data['max_question_length']), dtype = 'int32')
pred, answer_probab = sess.run([t_prediction, t_ans_probab], feed_dict={
input_tensors['fc7']:fc7_features,
input_tensors['sentence']:question_ids,
})
print "Ans:", ans_map[pred[0]]
```

TensorFlow를 이용한 Exploring Models and Data for Image Question Answering 논문 구현

Prediction을 실행

- python predict.py --image_path="Data/sample.jpg" --question="Which animal is this?" --model_path="Data/model2.ckpt"

- 결과 :

sample.jpg

```
Ans: cat
Top Answers
cat
dog
bear
mouse
unknown
```

Result

bAbI Datsets (자연어 QA 데이터셋)

- FAIR에서 만든 자연어 QA를 위한 Dataset
- 문제(Task)의 종류 : 20개, 각각의 종류에 대해 1000개, 10000개의 질문과 답변이 존재
- https://research.fb.com/downloads/babi/

```
ID text
ID text
ID text
ID question[tab]answer[tab]supporting fact IDS.
...
```

```
1 Mary moved to the bathroom.
2 John went to the hallway.
3 Where is Mary?        bathroom        1
4 Daniel went back to the hallway.
5 Sandra moved to the garden.
6 Where is Daniel?      hallway 4
7 John moved to the office.
8 Sandra journeyed to the bathroom.
9 Where is Daniel?      hallway 4
10 Mary moved to the hallway.
11 Daniel travelled to the office.
12 Where is Daniel?     office  11
13 John went back to the garden.
14 John moved to the bedroom.
15 Where is Sandra?     bathroom        8
1 Sandra travelled to the office.
2 Sandra went to the bathroom.
3 Where is Sandra?      bathroom        2
```

bAbI Datsets (자연어 QA 데이터셋)

- 총 20개의 Task

#	Task	Class name
1	Basic factoid QA with single supporting fact	WhereIsActor
2	Factoid QA with two supporting facts	WhereIsObject
3	Factoid QA with three supporting facts	WhereWasObject
4	Two argument relations: subject vs. object	IsDir
5	Three argument relations	WhoWhatGave
6	Yes/No questions	IsActorThere
7	Counting	Counting
8	Lists/Sets	Listing
9	Simple Negation	Negation
10	Indefinite Knowledge	Indefinite
11	Basic coreference	BasicCoreference
12	Conjunction	Conjunction
13	Compound coreference	CompoundCoreference
14	Time manipulation	Time
15	Basic deduction	Deduction
16	Basic induction	Induction
17	Positional reasoning	PositionalReasoning
18	Reasoning about size	Size
19	Path finding	PathFinding
20	Reasoning about agent's motivation	Motivations

어려운 Task – 19. Path Finding

```
1 The office is east of the hallway.
2 The kitchen is north of the office.
3 The garden is west of the bedroom.
4 The office is west of the garden.
5 The bathroom is north of the garden.
6 How do you go from the kitchen to the garden? s,e 2 4
1 The bedroom is west of the hallway.
2 The office is east of the garden.
3 The garden is north of the kitchen.
4 The kitchen is north of the bathroom.
5 The hallway is west of the garden.
6 How do you go from the kitchen to the hallway?    n,w 3 5
1 The bedroom is south of the hallway.
2 The bathroom is east of the office.
3 The kitchen is west of the garden.
4 The garden is south of the office.
5 The office is south of the bedroom.
6 How do you go from the garden to the bedroom? n,n 4 5
1 The bedroom is north of the hallway.
2 The garden is west of the kitchen.
3 The bathroom is south of the kitchen.
4 The office is north of the bedroom.
5 The hallway is east of the kitchen.
6 How do you go from the bedroom to the kitchen?    s,w 1 5
```

쉬운 Task – 1. Where is Actor?

```
1 Mary moved to the bathroom.
2 John went to the hallway.
3 Where is Mary?    bathroom    1
4 Daniel went back to the hallway.
5 Sandra moved to the garden.
6 Where is Daniel?  hallway 4
7 John moved to the office.
8 Sandra journeyed to the bathroom.
9 Where is Daniel?  hallway 4
10 Mary moved to the hallway.
11 Daniel travelled to the office.
12 Where is Daniel?     office  11
13 John went back to the garden.
14 John moved to the bedroom.
15 Where is Sandra?     bathroom    8
```

논문 리뷰 - End-To-End Memory Networks

- Sukhbaatar, Sainbayar, Jason Weston, and Rob Fergus. "End-to-end memory networks." NIPS 2015

- http://papers.nips.cc/paper/5846-end-to-end-memory-networks.pdf

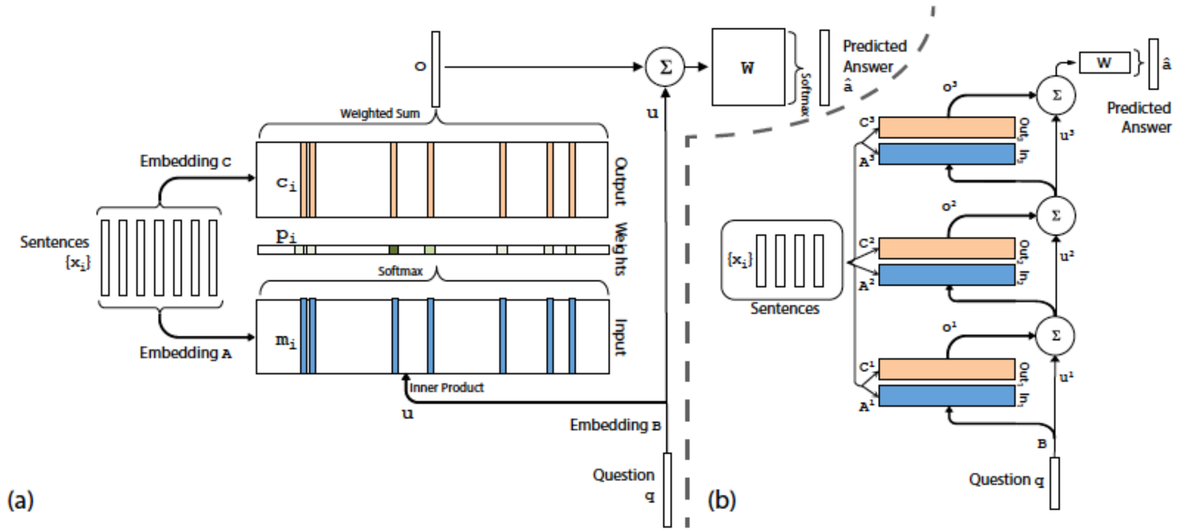


Figure 1: (a): A single layer version of our model. (b): A three layer version of our model. In practice, we can constrain several of the embedding matrices to be the same (see Section 2.2).

다루는 문제 – bAbI, PTB Language Modelling

1. bAbI

```
1 Mary moved to the bathroom.
2 John went to the hallway.
3 Where is Mary?        bathroom        1
4 Daniel went back to the hallway.
5 Sandra moved to the garden.
6 Where is Daniel?      hallway 4
7 John moved to the office.
8 Sandra journeyed to the bathroom.
9 Where is Daniel?      hallway 4
10 Mary moved to the hallway.
11 Daniel travelled to the office.
12 Where is Daniel?     office  11
13 John went back to the garden.
14 John moved to the bedroom.
15 Where is Sandra?     bathroom        8
1 Sandra travelled to the office.
2 Sandra went to the bathroom.
3 Where is Sandra?      bathroom        2
```

2. Language Modelling – Penn Tree Bank(PTB)

```
 no it was n't black monday
 but while the new york stock exchange did n't fall apart friday as the dow jones industrial average plunged N points most of it
in the final hour it barely managed to stay this side of chaos
 some circuit breakers installed after the october N crash failed their first test traders say unable to cool the selling panic
in both stocks and futures
 the N stock specialist firms on the big board floor the buyers and sellers of last resort who were criticized after the N crash
once again could n't handle the selling pressure
 big investment banks refused to step up to the plate to support the beleaguered floor traders by buying big blocks of stock
traders say
 heavy selling of standard & poor 's 500-stock index futures in chicago <unk> beat stocks downward
 seven big board stocks ual amr bankamerica walt disney capital cities\/abc philip morris and pacific telesis group stopped
trading and never resumed
 the <unk> has already begun
 the equity market was <unk>
```

Motivation

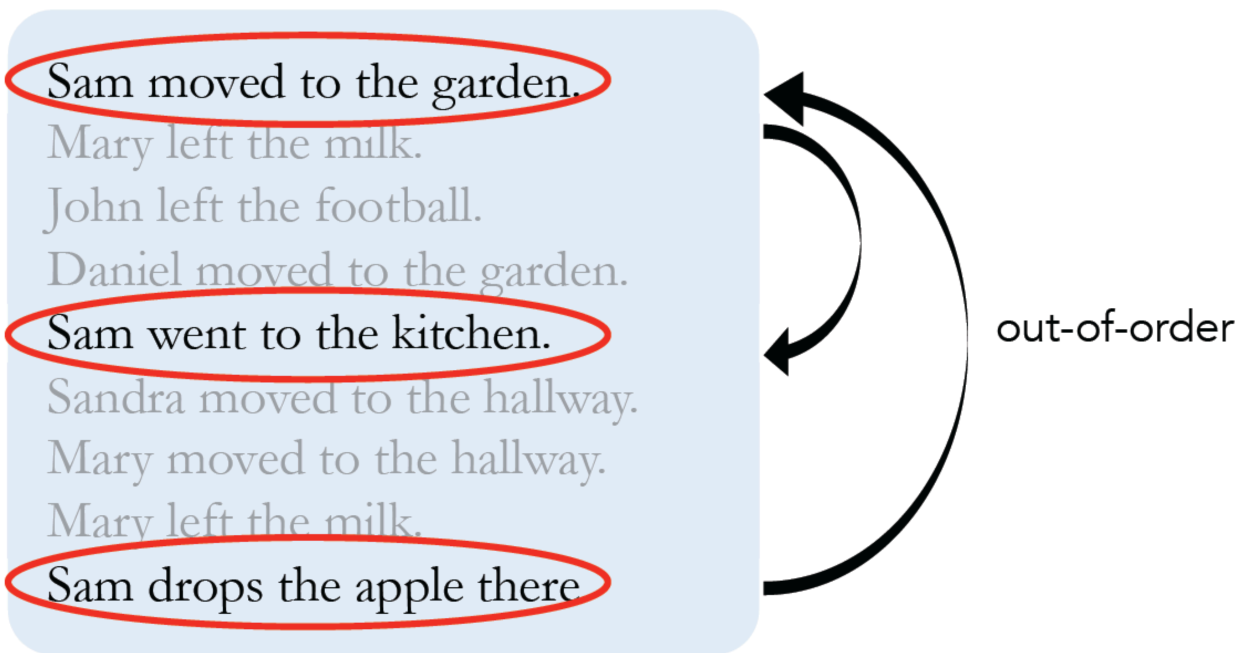
Motivation

- Good models exist for some data structures
  - RNN for temporal structure
  - ConvNet for spatial structure
- But we still struggle with some type of dependencies
  - out-of-order access
  - long-term dependency
  - unordered set

Slide credit: Sainbayar Sukhbaatar

Motivation

Ex) Question & Answering on story



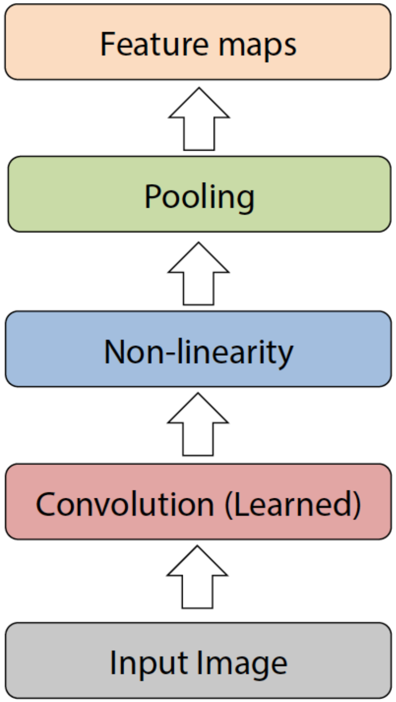
Q: Where was the apple after the garden?

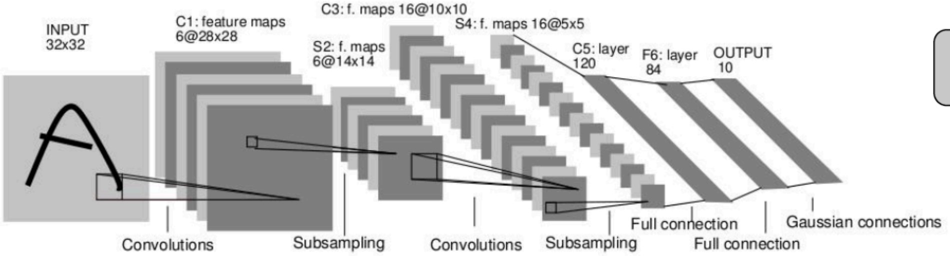
Slide credit: Sainbayar Sukhbaatar

기존 Deep Learning 모델들을 Memory 관점에서 리뷰 - CNNs

Convolutional Network (ConvNet)

- Feed-forward operation:
  - Convolve input
  - Non-linearity (rectified linear)
  - Pooling (local max)
- Features computed independently per-image
- Only "memory" is in network weights
  - Learnt from training set



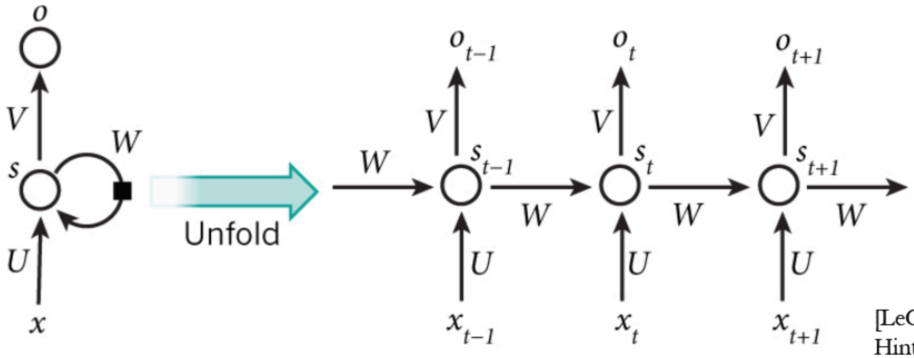


LeCun et al. 1998

Slide credit: Sainbayar Sukhbaatar

기존 Deep Learning 모델들을 Memory 관점에서 리뷰 – RNNs

Recurrent Neural Networks (RNNs)



[LeCun, Bengio & Hinton, Nature 2015]

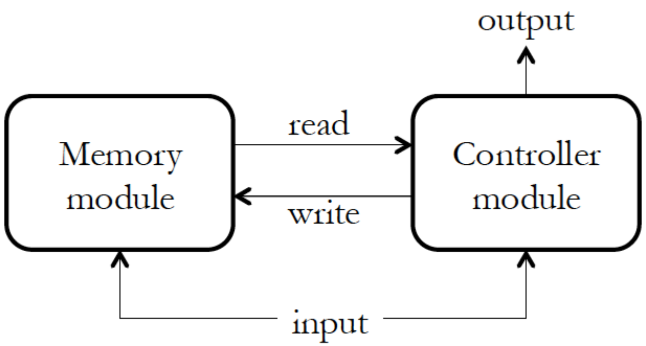
- Implicit memory within internal state s
- Mixing of computation & memory
  - Complex computation requires many layers of non-linearity
  - But some information is lost with each non-linearity
  - Gradient vanishing, catastrophic forgetting problems
  - Workarounds: gate units (e.g. LSTMs); impose slow/fast state

Slide credit: Sainbayar Sukhbaatar

Memory Networks – External Memory

External Global Memory

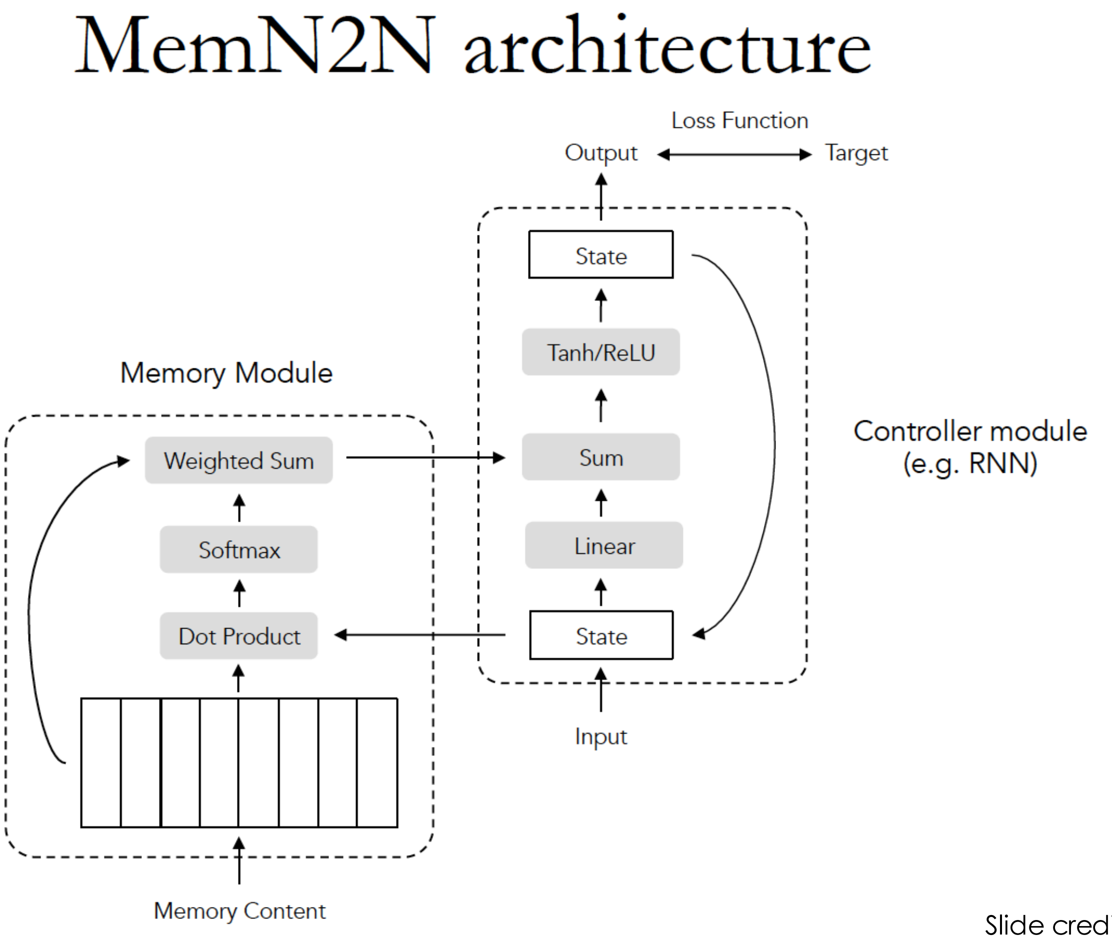
- Separating memory from computation
  - Dedicated separate memory module
  - Memory can be stack or list/set of vectors



- Control module accesses memory (read, write)
- Advantage: stable, scalable

Slide credit: Sainbayar Sukhbaatar

End-to-End Memory Networks Architecture

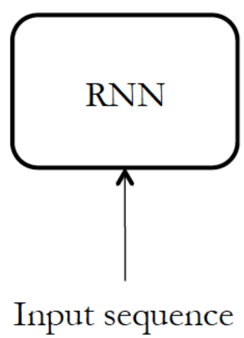


Slide credit: Sainbayar Sukhbaatar

RNNs vs MemN2N

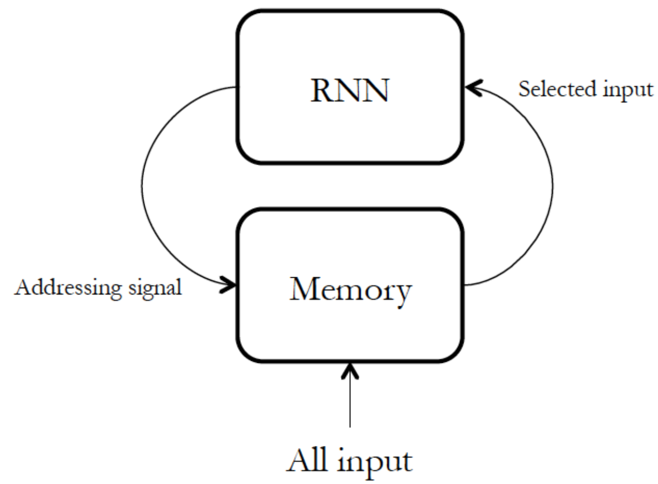
RNN viewpoint of MemN2N

Plain RNN



Inputs are fed to RNN one-by-one in order. RNN has only one chance to look at a certain input symbol.

Memory Network



Place all inputs in the memory. Let the model decide which part it reads next.

Slide credit: Sainbayar Sukhbaatar

RNNs vs MemN2N

Advantages of MemN2N over RNN

- More generic input format
  - Any **set** of vectors can be input
  - Each vector can be
    - BOW of symbols (including location)
    - Image feature + feature position
  - Location can be 1D, 2D, …
  - Variable size
- Out-of-order access to input data
- Less distracted by unimportant inputs
- Longer term memorization
- No vanishing or exploding gradient problems

Slide credit: Sainbayar Sukhbaatar

Input : Sentences + Question

- **Input : Sentences + Question**
- **Sentences:**

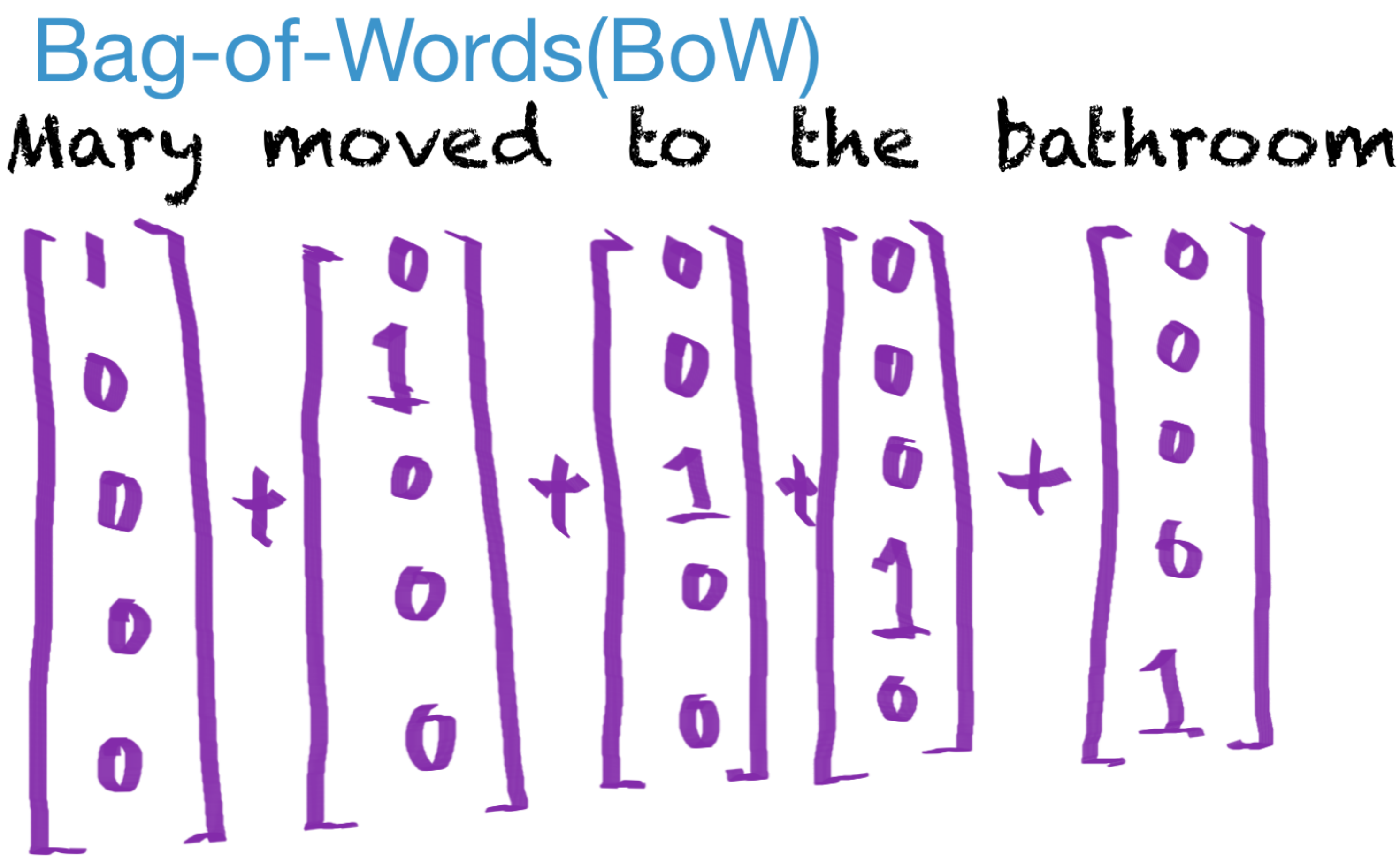
1 Mary moved to the bathroom.

2 John went to the hallway.

- **Questions:**

Where is Mary?

문장 = Bag-of-Words(BoW)로 표현



Memory 생성

Memory: 문장 하나로부터 만들어진다

$$m_1 = A \underset{\text{mary}}{\underset{x_{11}}{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix}}} + A \underset{\text{journeyed}}{\underset{x_{12}}{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix}}} + A \underset{\text{to}}{\underset{x_{13}}{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix}}} + A \underset{\text{the}}{\underset{x_{14}}{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ . \\ . \\ . \\ 0 \end{bmatrix}}}$$

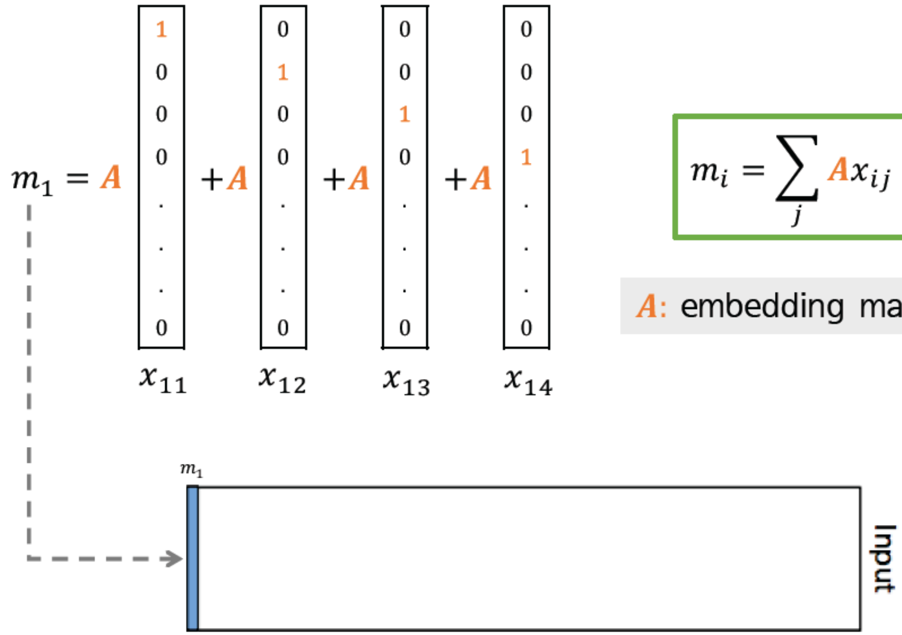
$$\boxed{m_i = \sum_j A x_{ij}}$$

A: embedding matrix

Slide credit: carpedm20
(https://goo.gl/J7KvrR)

Memory 생성

Recurrent attention model
with external memory



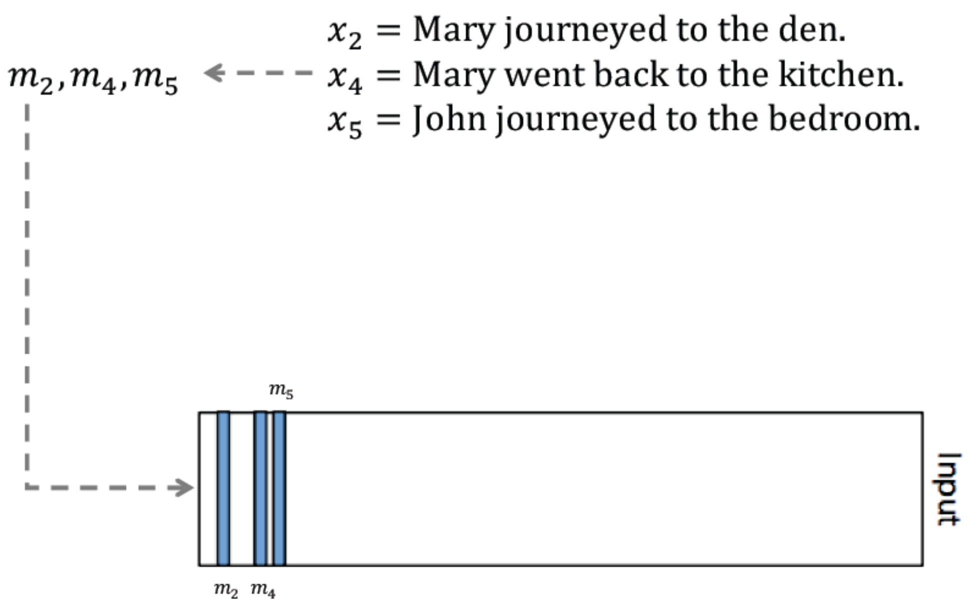
$$\boxed{m_i = \sum_j A x_{ij}}$$

A: embedding matrix

Slide credit: carpedm20
(https://goo.gl/J7KvrR)

Memory 사용

Memory: 한 task에는 여러개가 사용된다

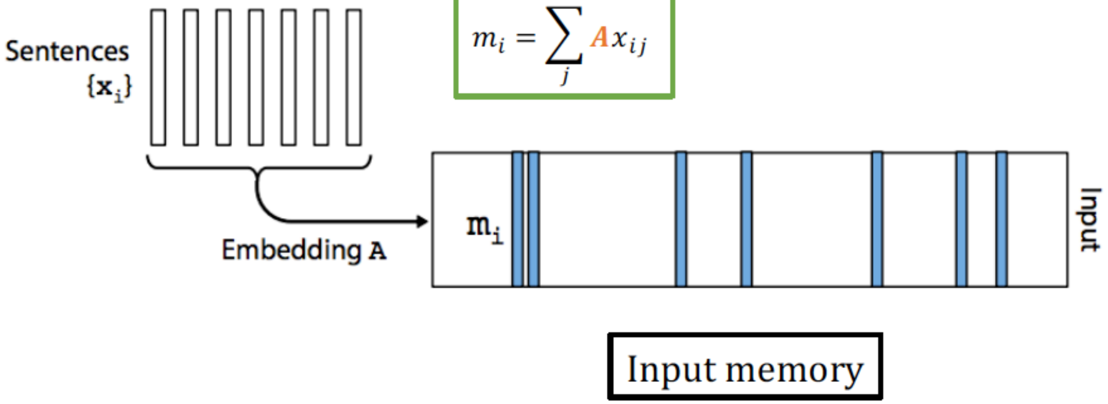


Slide credit: carpedm20
(https://goo.gl/J7KvrR)

Memory 사용

Memory: 필요한 것만 Input으로 사용

$$m_i = \sum_j A x_{ij}$$



실제로 메모리의 본체는 embedding matrix인 A이며, A가 점차 학습된다

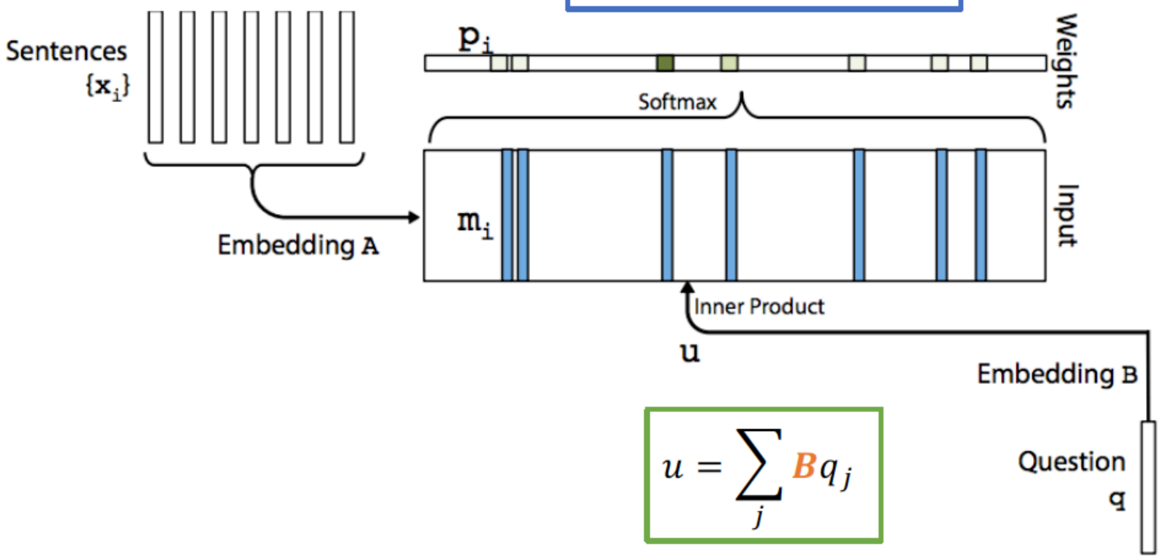
Slide credit: carpedm20
(https://goo.gl/J7KvrR)

Memory의 Attention

Recurrent attention model with external memory

얼마나 집중할 것인지

$$p_i = Softmax(u^T m_t)$$



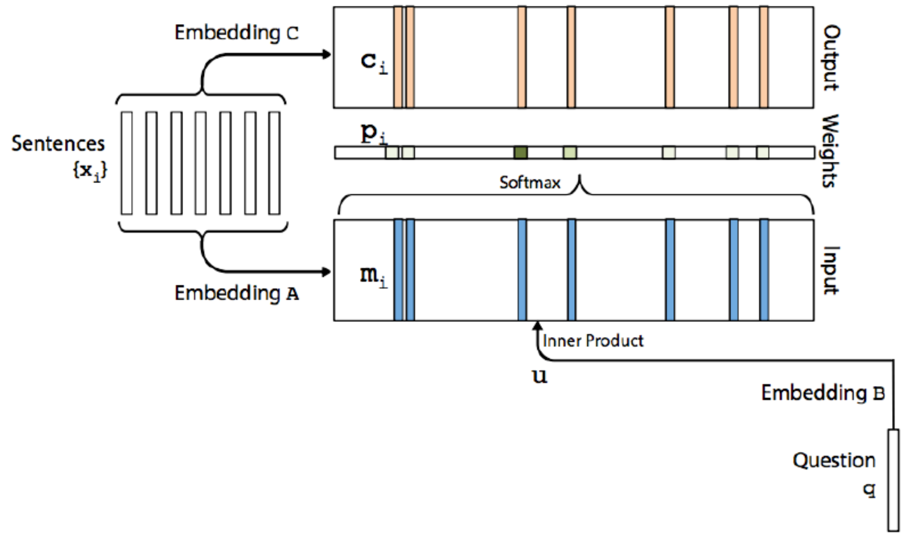
$$u = \sum_j B q_j$$

Slide credit: carpedm20 (https://goo.gl/J7KvrR)

Memory의 Attention

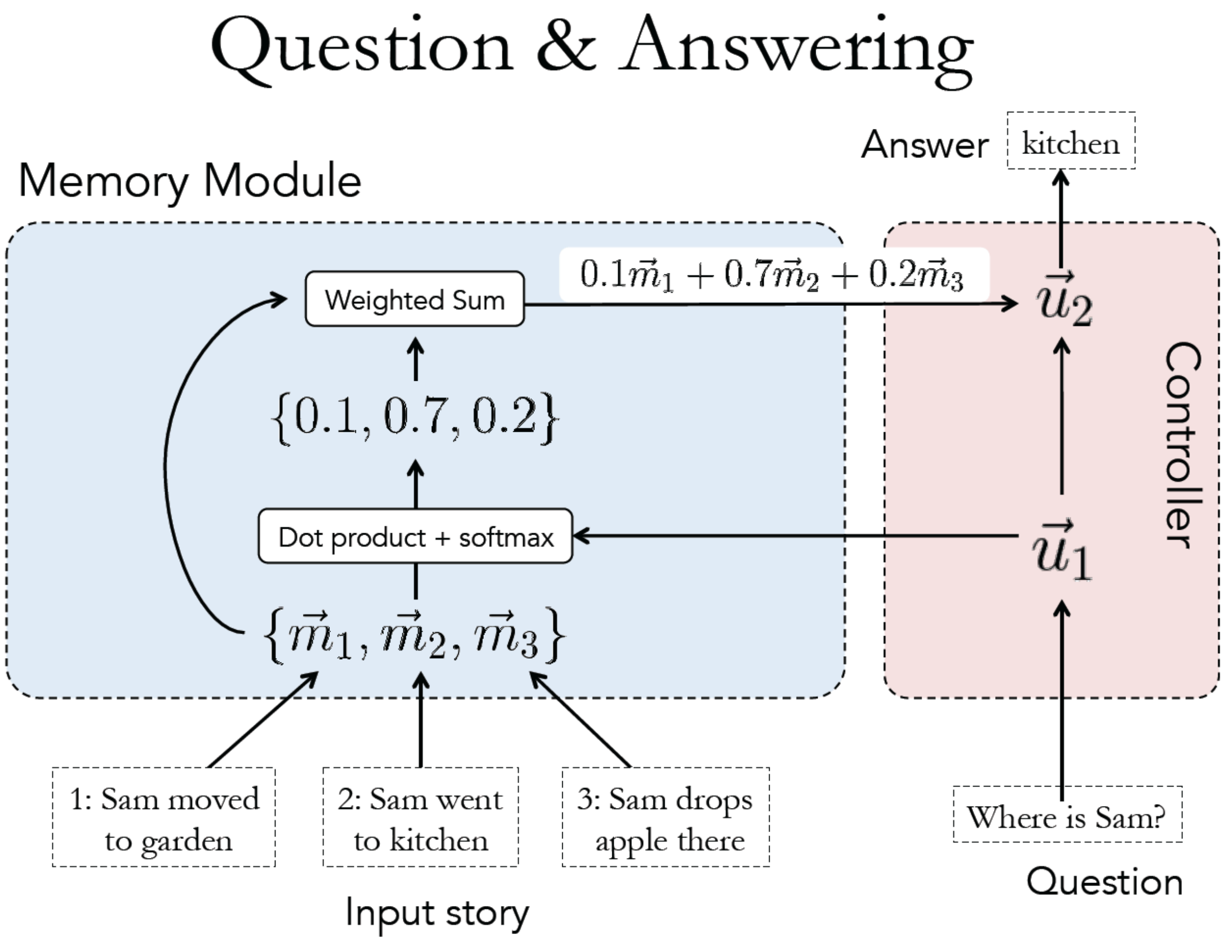
Recurrent attention model
with external memory

$$c_i = \sum_j C x_{ij}$$



Slide credit: carpedm20
(https://goo.gl/J7KvrR)

Memory의 Attention

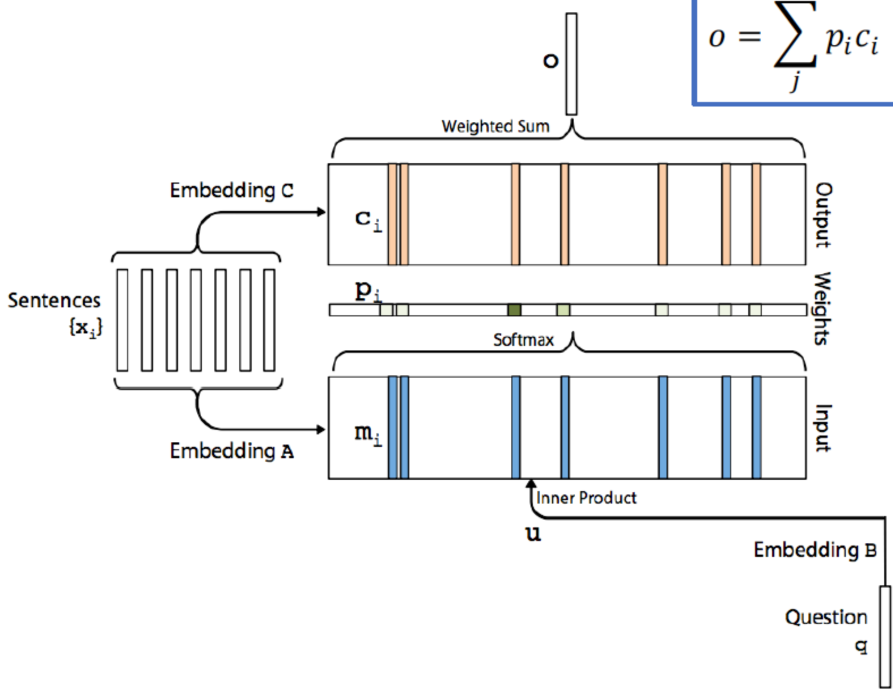


Slide credit: Sainbayar Sukhbaatar

O = Attention * Embedded Input

Recurrent attention model
with external memory

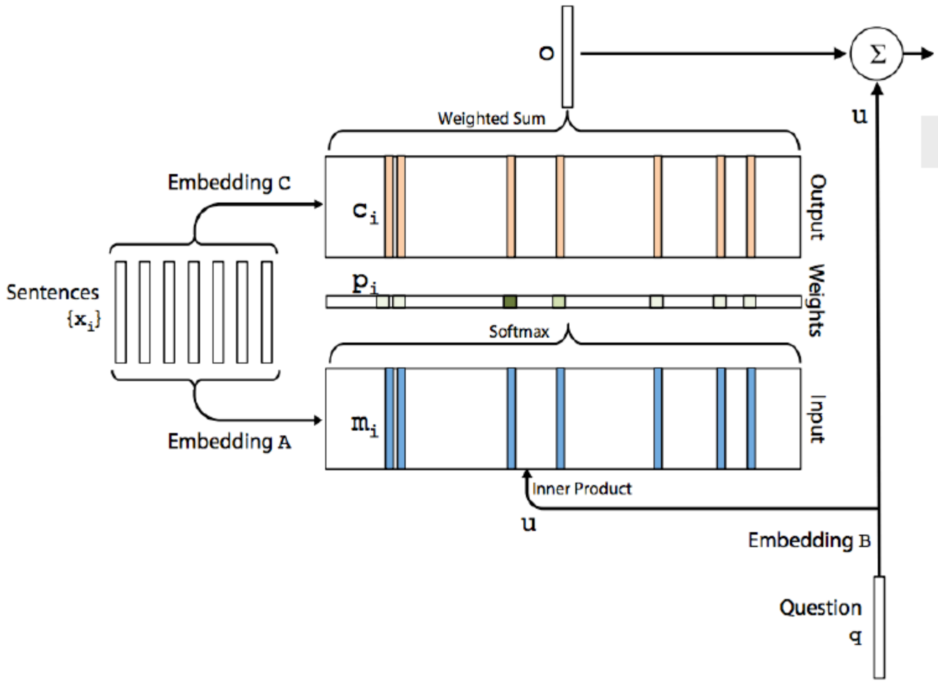
$$o = \sum_j p_i c_i$$



Slide credit: carpedm20
(https://goo.gl/J7KvrR)

Output 계산

Output: 요약된 정보 o + 질문 정보 u



$\boxed{o + u}$

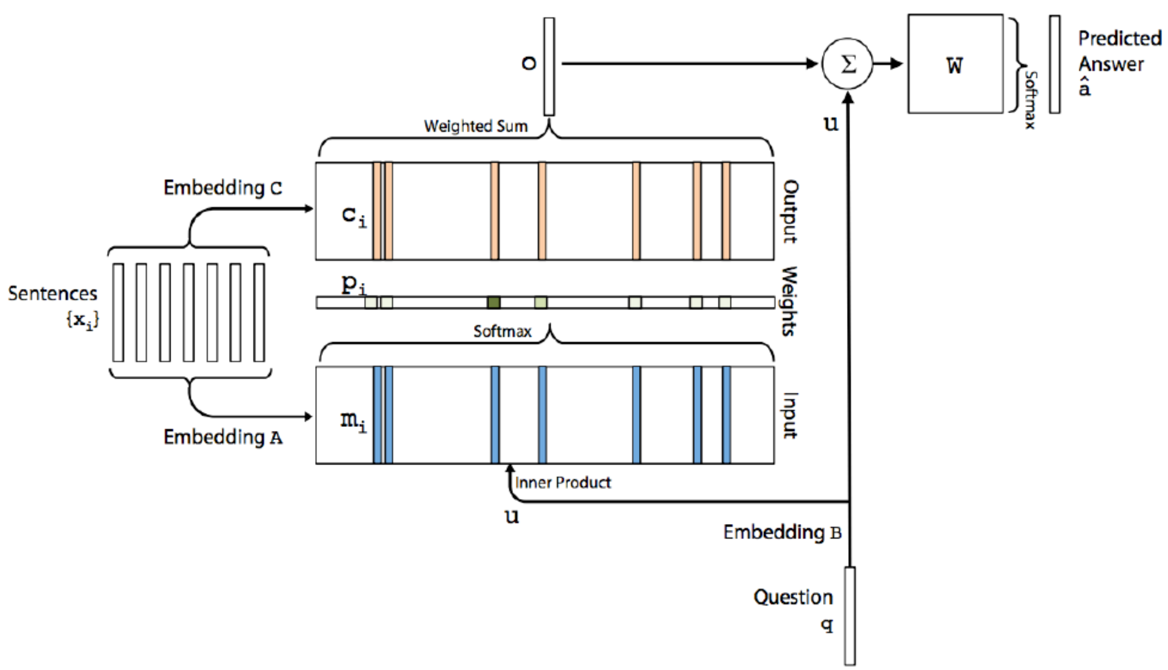
o, u 둘 다 고려해 답을 도출

Slide credit: carpedm20 (https://goo.gl/J7KvrR)

Output 계산

Output: 실제로 정답 단어 $\hat{a}$

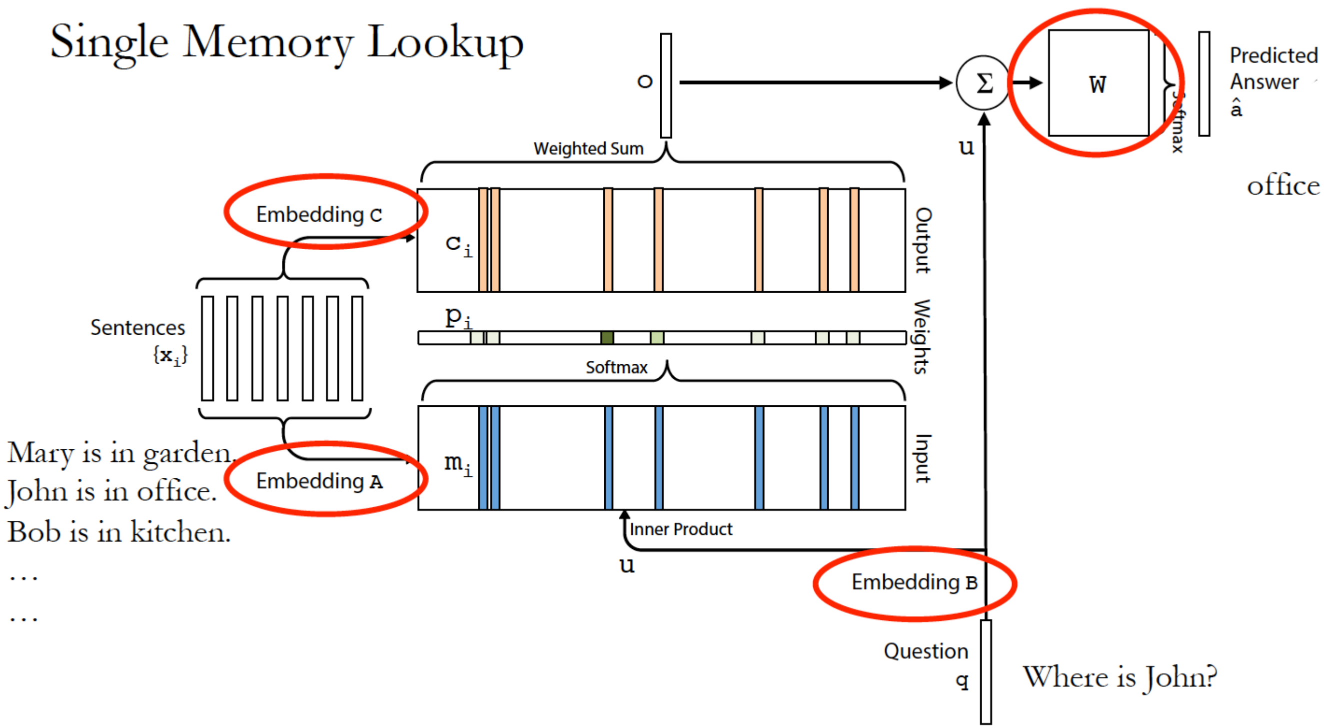
$$\hat{a}=\text{Softmax}(W(\text{o}+\text{u}))$$



Slide credit: carpedm20
(https://goo.gl/J7KvrR)

학습되는 Parameter들 = Embedding Matrix A, B, C W

MemN2N applied to bAbI task

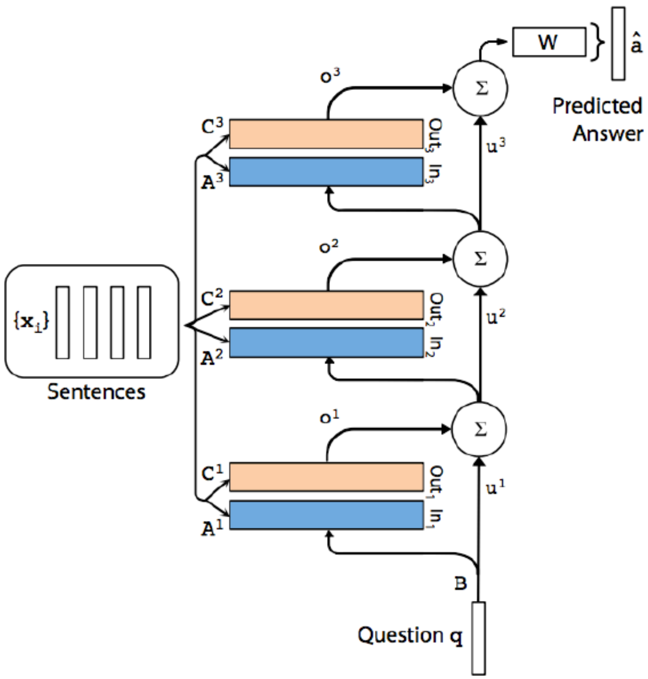


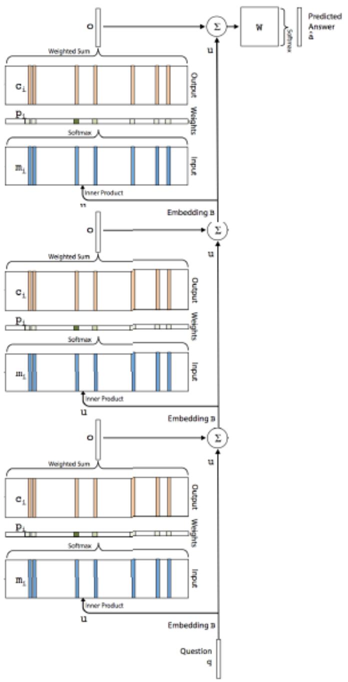
Training: estimate embedding matrices A, B & C and output matrix W

Slide credit: Sainbayar Sukhbaatar

Multiple Hop

Recurrent attention model
with external memory





Slide credit: carpedm20
(https://goo.gl/J7KvrR)

Experiment Result 1– Q&A(bAbI Dataset)

- QA Task에 대한 Experiment Result

	Baseline			MemN2N									
Task	Strongly Supervised MemNN [22]	LSTM [22]	MemNN WSH	BoW	PE	PE LS	PE LS RN	1 hop PE LS joint	2 hops PE LS joint	3 hops PE LS joint	PE LS RN joint	PE LS LW joint
1: 1 supporting fact	0.0	50.0	0.1	0.6	0.1	0.2	0.0	0.8	0.0	0.1	0.0	0.1
2: 2 supporting facts	0.0	80.0	42.8	17.6	21.6	12.8	8.3	62.0	15.6	14.0	11.4	18.8
3: 3 supporting facts	0.0	80.0	76.4	71.0	64.2	58.8	40.3	76.9	31.6	33.1	21.9	31.7
4: 2 argument relations	0.0	39.0	40.3	32.0	3.8	11.6	2.8	22.8	2.2	5.7	13.4	17.5
5: 3 argument relations	2.0	30.0	16.3	18.3	14.1	15.7	13.1	11.0	13.4	14.8	14.4	12.9
6: yes/no questions	0.0	52.0	51.0	8.7	7.9	8.7	7.6	7.2	2.3	3.3	2.8	2.0
7: counting	15.0	51.0	36.1	23.5	21.6	20.3	17.3	15.9	25.4	17.9	18.3	10.1
8: lists/sets	9.0	55.0	37.8	11.4	12.6	12.7	10.0	13.2	11.7	10.1	9.3	6.1
9: simple negation	0.0	36.0	35.9	21.1	23.3	17.0	13.2	5.1	2.0	3.1	1.9	1.5
10: indefinite knowledge	2.0	56.0	68.7	22.8	17.4	18.6	15.1	10.6	5.0	6.6	6.5	2.6
11: basic coreference	0.0	38.0	30.0	4.1	4.3	0.0	0.9	8.4	1.2	0.9	0.3	3.3
12: conjunction	0.0	26.0	10.1	0.3	0.3	0.1	0.2	0.4	0.0	0.3	0.1	0.0
13: compound coreference	0.0	6.0	19.7	10.5	9.9	0.3	0.4	6.3	0.2	1.4	0.2	0.5
14: time reasoning	1.0	73.0	18.3	1.3	1.8	2.0	1.7	36.9	8.1	8.2	6.9	2.0
15: basic deduction	0.0	79.0	64.8	24.3	0.0	0.0	0.0	46.4	0.5	0.0	0.0	1.8
16: basic induction	0.0	77.0	50.5	52.0	52.1	1.6	1.3	47.4	51.3	3.5	2.7	51.0
17: positional reasoning	35.0	49.0	50.9	45.4	50.1	49.0	51.0	44.4	41.2	44.5	40.4	42.6
18: size reasoning	5.0	48.0	51.3	48.1	13.6	10.1	11.1	9.6	10.3	9.2	9.4	9.2
19: path finding	64.0	92.0	100.0	89.7	87.4	85.6	82.8	90.7	89.9	90.2	88.0	90.6
20: agent's motivation	0.0	9.0	3.6	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.2
Mean error (%)	6.7	51.3	40.2	25.1	20.3	16.3	13.9	25.8	15.6	13.3	12.4	15.2
Failed tasks (err. > 5%)	4	20	18	15	13	12	11	17	11	11	11	10
On 10k training data												
Mean error (%)	3.2	36.4	39.2	15.4	9.4	7.2	6.6	24.5	10.9	7.9	7.5	11.0
Failed tasks (err. > 5%)	2	16	17	9	6	4	4	16	7	6	6	6

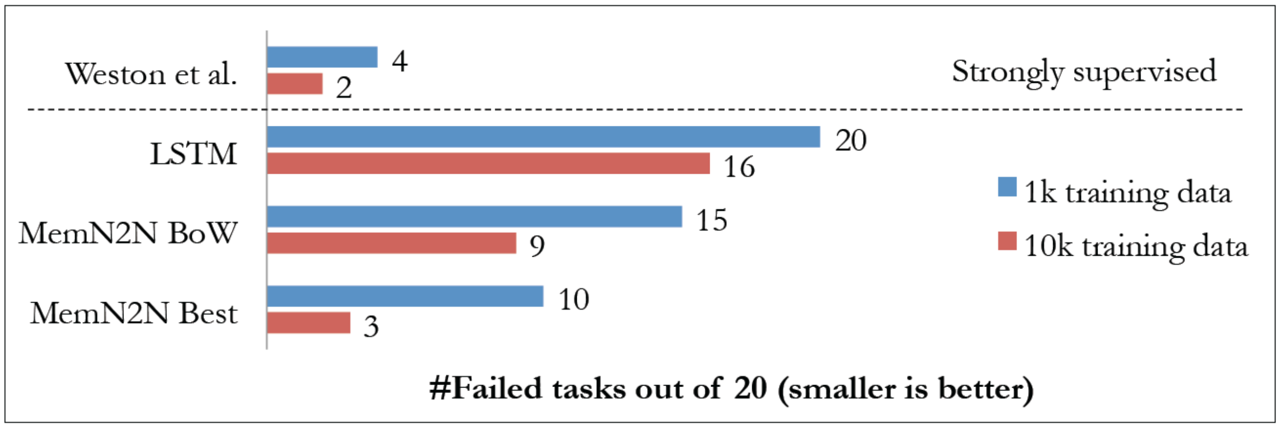
Table 1: Test error rates (%) on the 20 QA tasks for models using 1k training examples (mean test errors for 10k training examples are shown at the bottom). Key: BoW = bag-of-words representation; PE = position encoding representation; LS = linear start training; RN = random injection of time index noise; LW = RNN-style layer-wise weight tying (if not stated, adjacent weight tying is used); joint = joint training on all tasks (as opposed to per-task training).

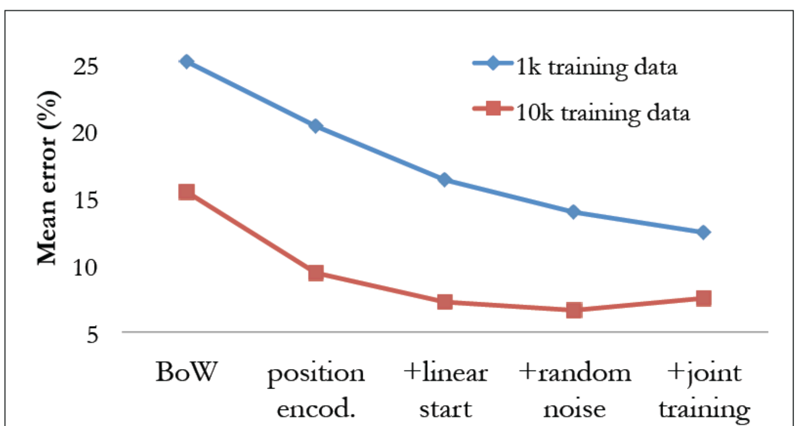
Experiment Result 1– Q&A(bAbI Dataset)

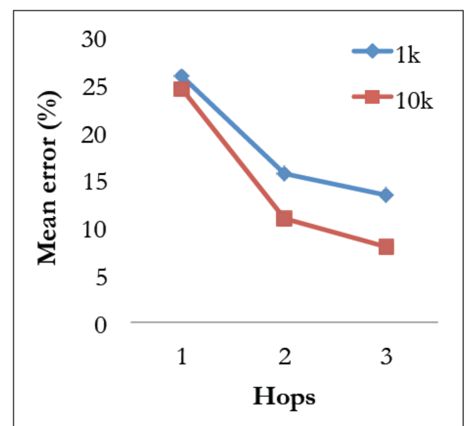
- QA Task에 대한 Experiment Result

Slide credit: Sainbayar Sukhbaatar

Performance on bAbI test set







Attention Focus

- Inference에 사용되는 Sentence들의 확률

Story (1: 1 supporting fact)	Support	Hop 1	Hop 2	Hop 3
Daniel went to the bathroom.		0.00	0.00	0.03
Mary travelled to the hallway.		0.00	0.00	0.00
John went to the bedroom.		0.37	0.02	0.00
John travelled to the bathroom.	yes	0.60	0.98	0.96
Mary went to the office.		0.01	0.00	0.00
Where is John? Answer: bathroom Prediction: bathroom				

Story (2: 2 supporting facts)	Support	Hop 1	Hop 2	Hop 3
John dropped the milk.		0.06	0.00	0.00
John took the milk there.	yes	0.88	1.00	0.00
Sandra went back to the bathroom.		0.00	0.00	0.00
John moved to the hallway.	yes	0.00	0.00	1.00
Mary went back to the bedroom.		0.00	0.00	0.00
Where is the milk? Answer: hallway Prediction: hallway				

Story (16: basic induction)	Support	Hop 1	Hop 2	Hop 3
Brian is a frog.	yes	0.00	0.98	0.00
Lily is gray.		0.07	0.00	0.00
Brian is yellow.	yes	0.07	0.00	1.00
Julius is green.		0.06	0.00	0.00
Greg is a frog.	yes	0.76	0.02	0.00
What color is Greg? Answer: yellow Prediction: yellow				

Story (18: size reasoning)	Support	Hop 1	Hop 2	Hop 3
The suitcase is bigger than the chest.	yes	0.00	0.88	0.00
The box is bigger than the chocolate.		0.04	0.05	0.10
The chest is bigger than the chocolate.	yes	0.17	0.07	0.90
The chest fits inside the container.		0.00	0.00	0.00
The chest fits inside the box.		0.00	0.00	0.00
Does the suitcase fit in the chocolate? Answer: no Prediction: no				

Figure 2: Example predictions on the QA tasks of [22]. We show the labeled supporting facts (support) from the dataset which MemN2N does not use during training, and the probabilities p of each hop used by the model during inference. MemN2N successfully learns to focus on the correct supporting sentences.

Experiment Result 2 – Language Modelling

- Language Modelling에 대한 실험결과

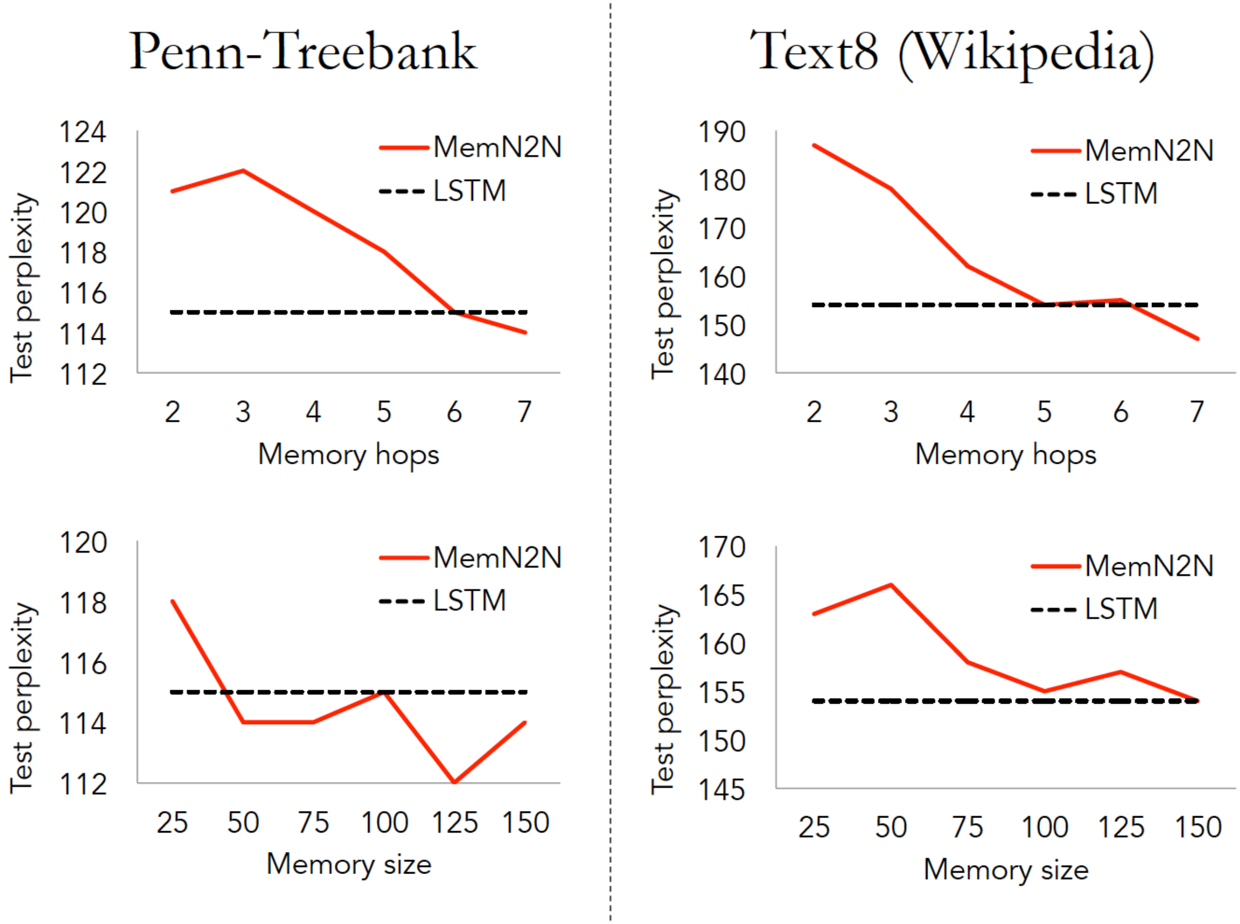
Model	Penn Treebank # of hidden	Penn Treebank # of hops	Penn Treebank memory size	Penn Treebank Valid. perp.	Penn Treebank Test perp.	Text8 # of hidden	Text8 # of hops	Text8 memory size	Text8 Valid. perp.	Text8 Test perp.
RNN [15]	300	-	-	133	129	500	-	-	-	184
LSTM [15]	100	-	-	120	115	500	-	-	122	154
SCRN [15]	100	-	-	120	115	500	-	-	-	161
MemN2N	150	2	100	128	121	500	2	100	152	187
	150	3	100	129	122	500	3	100	142	178
	150	4	100	127	120	500	4	100	129	162
	150	5	100	127	118	500	5	100	123	154
	150	6	100	122	115	500	6	100	124	155
	150	7	100	120	114	500	7	100	118	**147**
	150	6	25	125	118	500	6	25	131	163
	150	6	50	121	114	500	6	50	132	166
	150	6	75	122	114	500	6	75	126	158
	150	6	100	122	115	500	6	100	124	155
	150	6	125	120	112	500	6	125	125	157
	150	6	150	121	114	500	6	150	123	154
	150	7	200	118	**111**	-	-	-	-	-

Table 2: The perplexity on the test sets of Penn Treebank and Text8 corpora. Note that increasing the number of memory hops improves performance.

Experiment Result 2 – Language Modelling

- Language Modelling에 대한 실험결과

Slide credit: Sainbayar Sukhbaatar



TensorFlow를 이용한 End-To-End Memory Networks 논문 구현

- https://github.com/solaris33/dl_cv_tensorflow_10weeks/tree/master/week4/MemNN

- https://github.com/domluna/memn2n

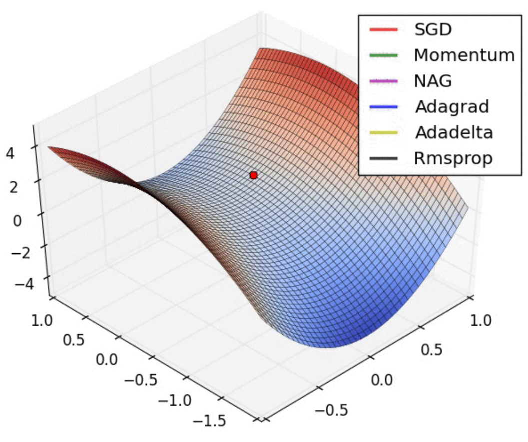
  - bAbI Task에 대한 MemN2N TensorFlow 구현

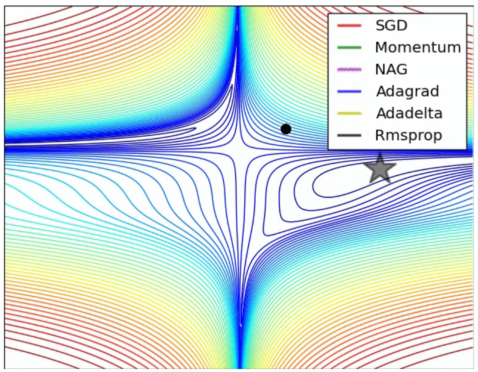
Optimization 알고리즘 정리

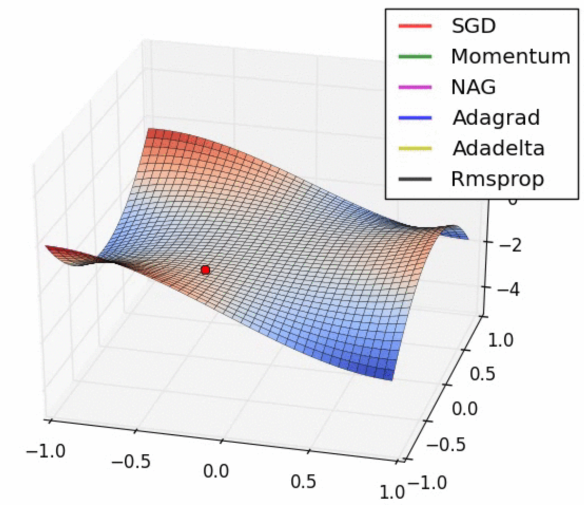
- 기본 Gradient Descent Algorithm

$$\theta = \theta - \eta \nabla_\theta J(\theta)$$

- 다양한 Optimization 알고리즘들의 동작 과정





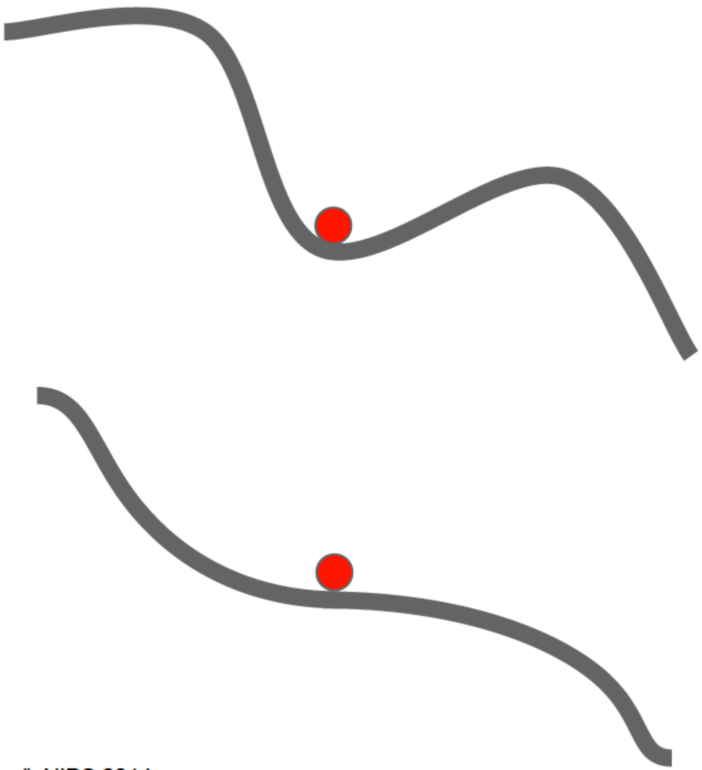


기본 Gradient Descent의 문제점

Optimization: Problems with SGD

What if the loss function has a **local minima** or **saddle point**?

Saddle points much more common in high dimension



Dauphin et al, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization", NIPS 2014

Slide credit: CS231n (https://goo.gl/QSA36t)

Momentum

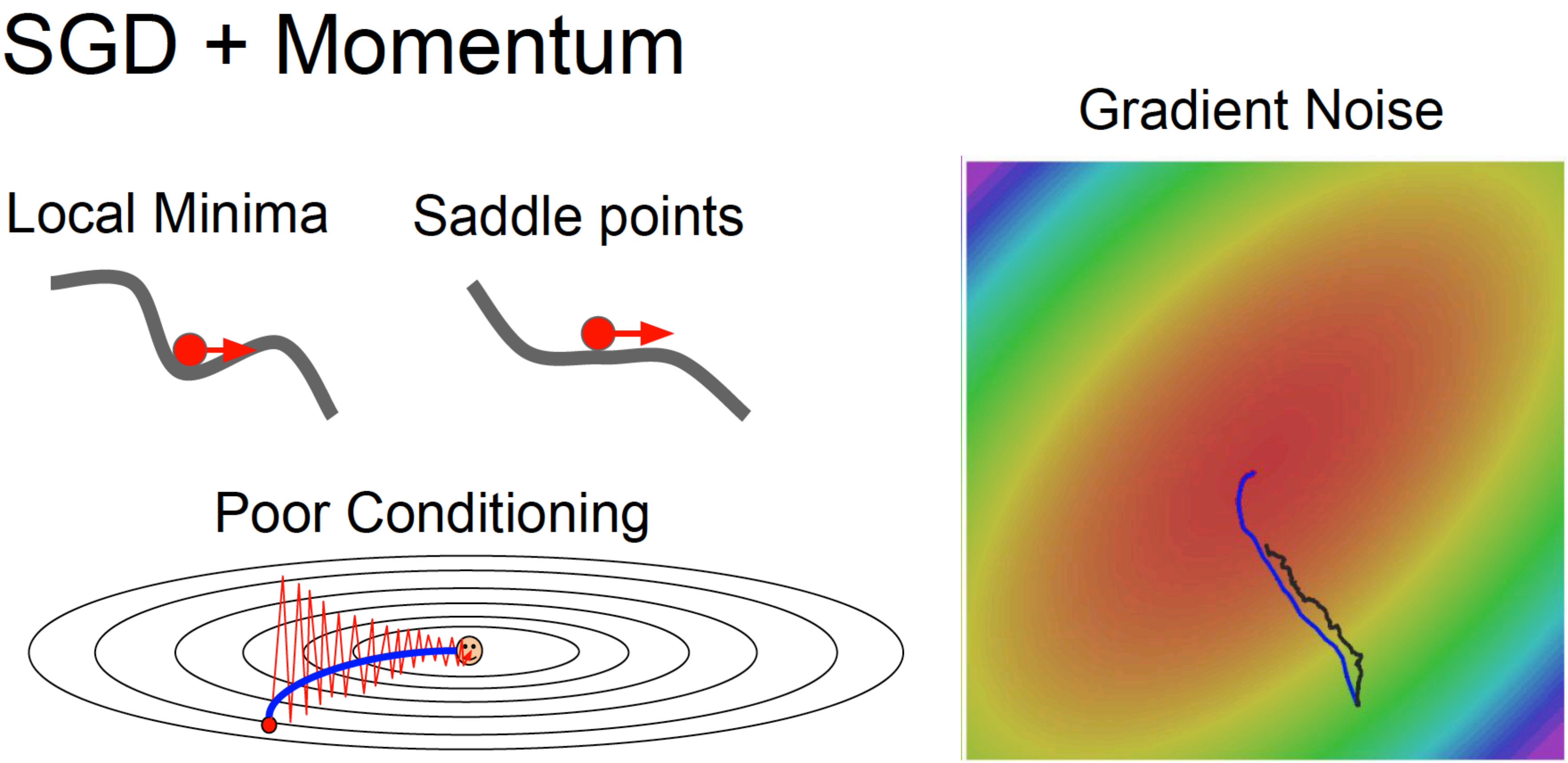
- Momentum은 이동방향에 **관성(Momentum)**을 주는 방법이다.

$$v = \gamma v + \alpha \nabla_\theta J(\theta; x^{(i)}, y^{(i)})$$
$$\theta = \theta - v$$

- Momentum을 추가함으로써

1. Oscillation없이 좀더 강건하게 Gradient Descent를 수행할 수 있다.
2. 관성을 이용하여 Saddle Point를 지나갈 수 있다.

Momentum의 장점



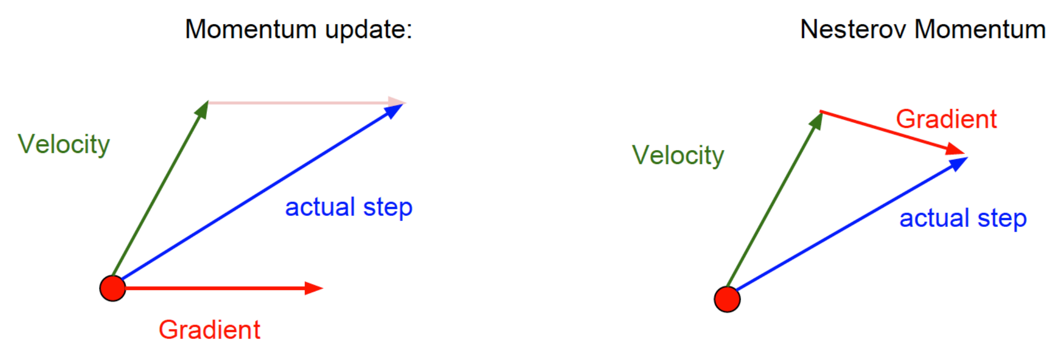
Slide credit: CS231n
(https://goo.gl/QSA36t)

Nesterov Momentum

- Nesterov Momentum은 관성 방향으로 미리 움직였다고 가정하고 Gradient를 계산한다. 따라서 이동해야할 방향을 좀더 정밀하게 고려할 수 있다.

$$v_{t+1} = \rho v_t - \alpha \nabla f(\boxed{x_t + \rho v_t})$$
$$x_{t+1} = x_t + v_{t+1}$$

Nesterov Momentum

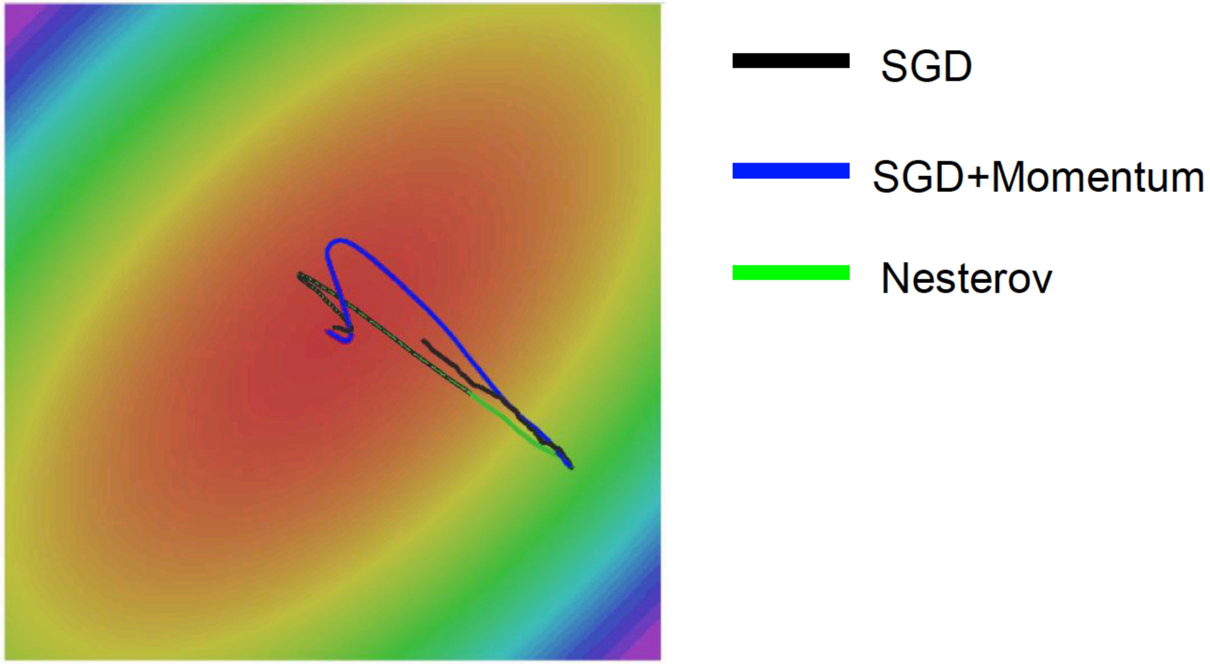


Nesterov, "A method of solving a convex programming problem with convergence rate O(1/k^2)", 1983
Nesterov, "Introductory lectures on convex optimization: a basic course", 2004
Sutskever et al, "On the importance of initialization and momentum in deel learning", ICML 2013

Slide credit: CS231n (https://goo.gl/QSA36t)

Nesterov Momentum의 장점

Nesterov Momentum



Slide credit: CS231n (https://goo.gl/QSA36t)

Adagrad(Adaptive Gradient)

- AdaGrad(Adaptive Gradient) 방법은 파라미터별로 Learning Rate를 다르게 가져가는 방법이다.
- AdaGrad의 핵심 철학은 **많이 업데이트된 파라미터는 Learning Rate가 작아지고, 적게 업데이트된 파라미터는 Learning Rate를 크게 가져가자는 것**이다. (많이 업데이트된 파라미터는 이미 Optimal 값에 가깝고, 적게 업데이트된 파라미터들은 아직 Optimal 값에 가깝지 않을 것이다.)
- AdaGrad의 문제점은 학습이 많이 진행되면 따라 분모부분(Gradient의 제곱들의 합)이 너무 커져서 Learning Rate가 0에 가깝게 된다는 점이다.
- 이를 해결하기 위해 RMSProp 알고리즘이 제안되었다.

$$G^k = G^{k-1} + \nabla J(\theta^{k-1})^2$$

$$\theta^k = \theta^{k-1} - \frac{\alpha}{\mathrm{sqrt}(G^{k-1})} \cdot \nabla J(\theta^{k-1})$$

RMSProp

- RMSProp은 제프리 힌튼이 제안한 방법으로, AdaGrad(Adaptive Gradient)에서 분모가 너무 커지던 문제를 해결한 방법이다.

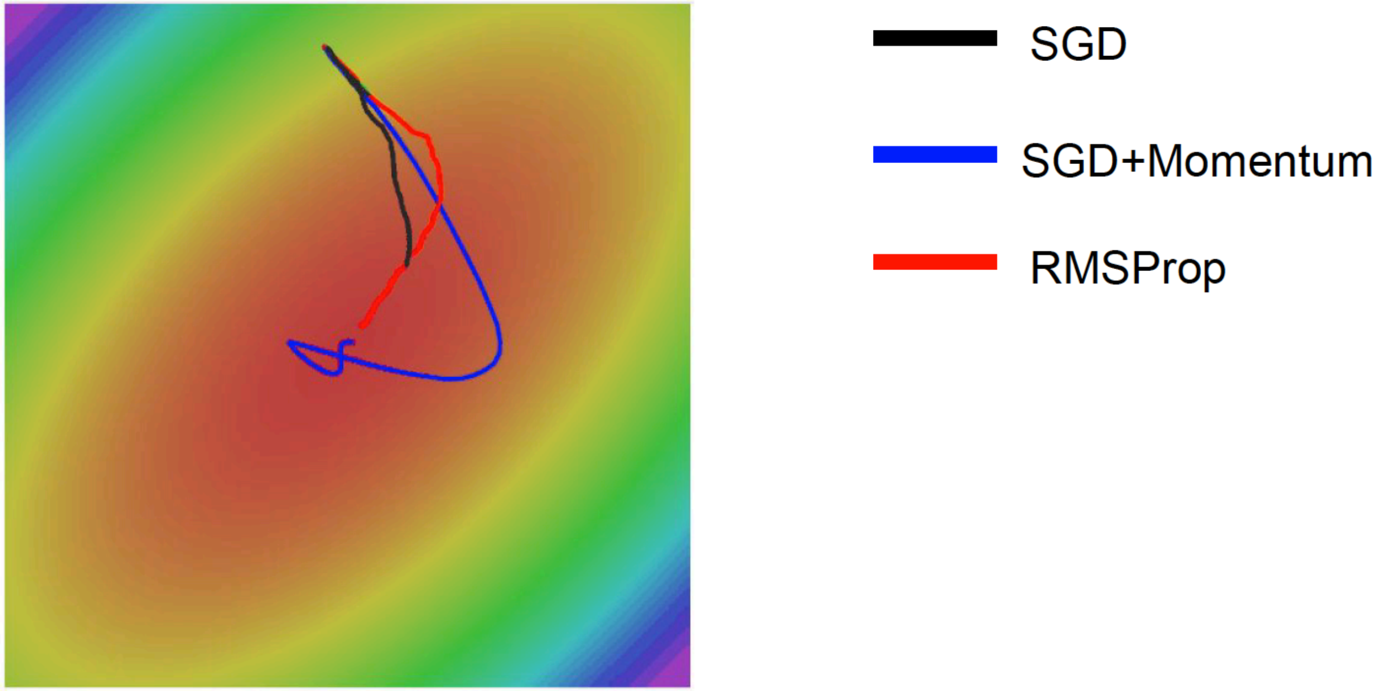
- 분모 부분을 계산할 때 Gradient의 제곱들의 단순합이 아니라 지수평균을 취해서 값이 너무 커지는 것을 방지하였다.

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

SGD, Momentum, RMSProp 비교

RMSProp



Slide credit: CS231n (https://goo.gl/QSA36t)

AdaDelta (Adaptive Delta)

- AdaDelta는 RMSProp과 비슷하지만 Learning Rate를 단순 η 대신에 Step size의 제곱의 지수평균을 이용해서 구한다.

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1-\rho) g_t^2$$

$$E[\Delta x^2]_t = \rho E[\Delta x^2]_{t-1} + (1-\rho) \Delta x_t^2$$

$$\Delta x_t = -\frac{\sqrt{E[\Delta x^2]_{t-1} + \epsilon}}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

$$x_{t+1} = x_t + \Delta x_t$$

Adam (ADaptive Moment Estimation)

- Adam (Adaptive Moment Estimation)은 RMSProp과 Momentum 방식을 합친 알고리즘이다.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

- 초기에는 m_t, v_t 값이 0 에 가깝에 Bias되어 있는 상태기 때문에 이를 보정하기 위해 아래 연산을 수행한다.

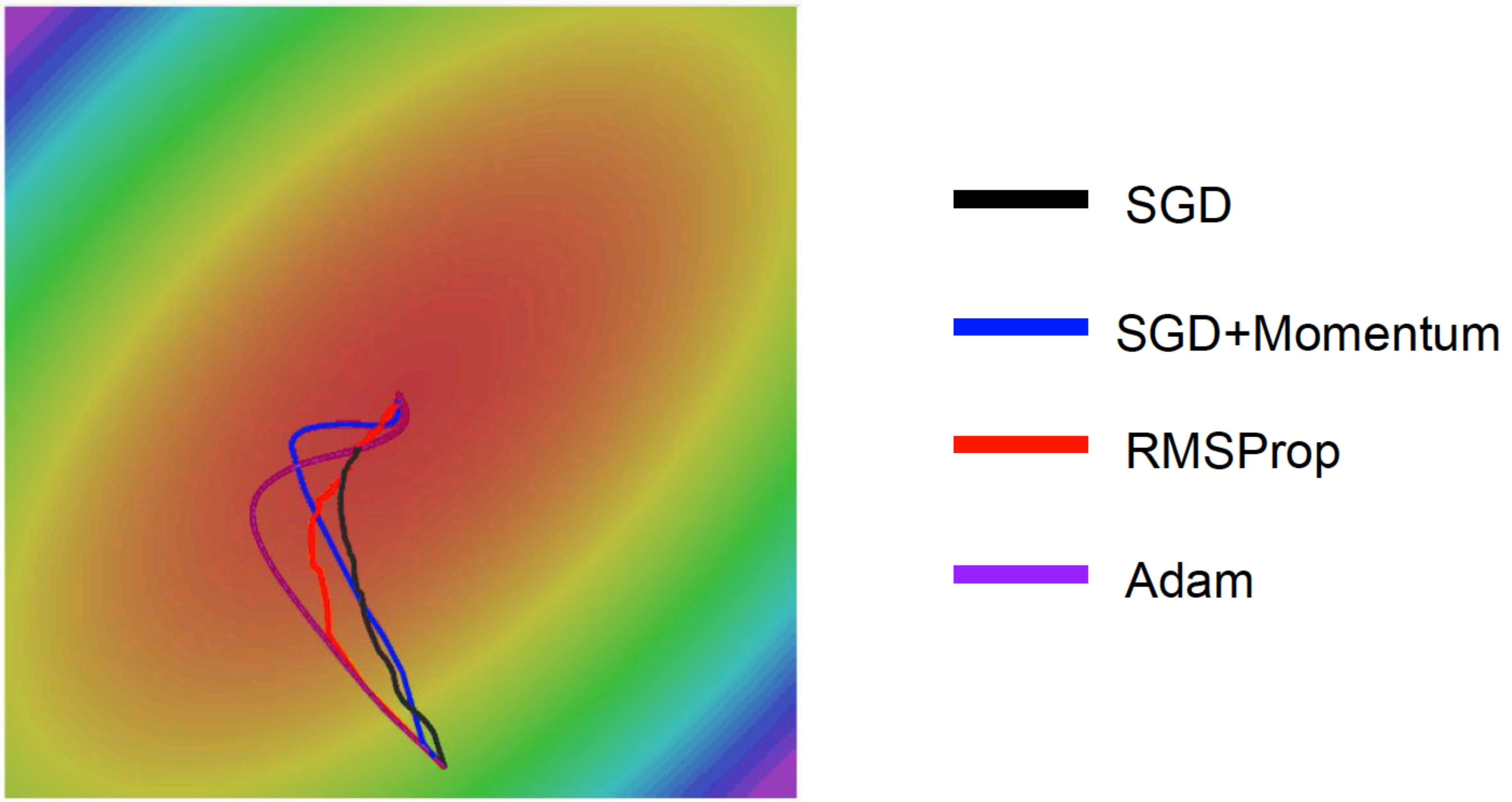
$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t.$$

- 보통 β1로는 0.9, β2로는 0.999, ϵ으로는 10−8정도의 값을 사용한다.

Adam, SGD, Momentum, RMSProp 비교

Adam

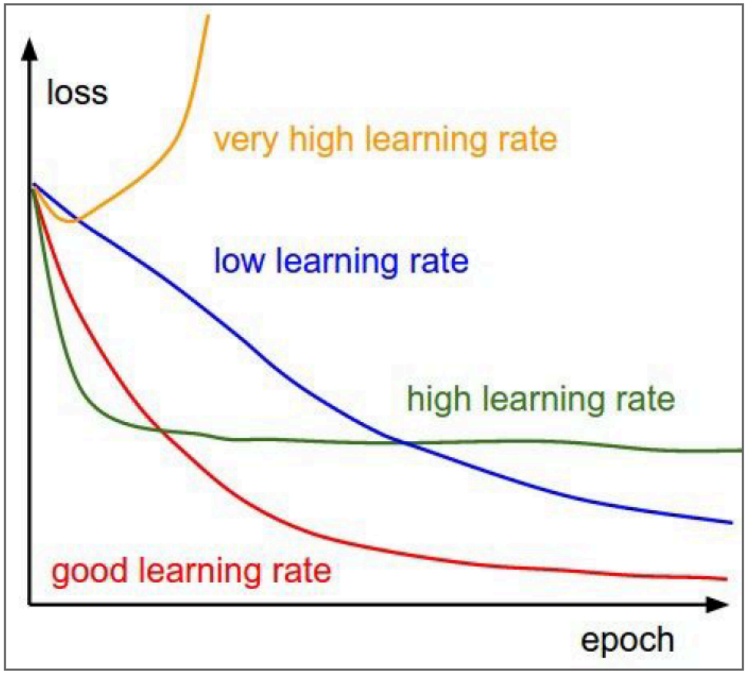


Slide credit: CS231n (https://goo.gl/QSA36t)

Learning Rate 설정

- 모든 Optimization 알고리즘은 적절한 Learning Rate 설정이 필요하다.

SGD, SGD+Momentum, Adagrad, RMSProp, Adam all have **learning rate** as a hyperparameter.



Q: Which one of these learning rates is best to use?

Slide credit: CS231n (https://goo.gl/QSA36t)

TensorFlow에 정의된 다양한 Optimizer들

Slide credit: CS 20SI
(https://goo.gl/Ez8wRq)

List of optimizers in TF

```
tf.train.GradientDescentOptimizer

tf.train.AdagradOptimizer

tf.train.MomentumOptimizer

tf.train.AdamOptimizer

tf.train.ProximalGradientDescentOptimizer

tf.train.ProximalAdagradOptimizer

tf.train.RMSPropOptimizer

And more
```

35

TensorFlow에서 Recurrent Neural Networks 구현하기

Slide credit: CS 20SI
(https://goo.gl/Ez8wRq)

Stack multiple cells

```
cell = tf.nn.rnn_cell.GRUCell(hidden_size)

rnn_cells = tf.nn.rnn_cell.MultiRNNCell([cell] * num_layers)
```

35

TensorFlow에서 Learning rate decay 사용하기

Slide credit: CS 20SI (https://goo.gl/Ez8wRq)

Anneal the learning rate

Optimizers accept both scalars and tensors as learning rate

```
learning_rate = tf.train.exponential_decay(init_lr,
                                           global_step,
                                           decay_steps,
                                           decay_rate,
                                           staircase=True)
optimizer = tf.train.AdamOptimizer(learning_rate)
```

TensorFlow에서 Dropout 사용하기

Slide credit: CS 20SI
(https://goo.gl/Ez8wRq)

Overfitting

Use dropout through tf.nn.dropout or DropoutWrapper for cells

- `tf.nn.dropout`

```
hidden_layer = tf.nn.dropout(hidden_layer, keep_prob)
```

- `DropoutWrapper`

```
cell = tf.nn.rnn_cell.GRUCell(hidden_size)
cell = tf.nn.rnn_cell.DropoutWrapper(cell,
                                     output_keep_prob=keep_prob)
```

과제 – 한글 bAbI 데이터셋에 대한 End-to-End Memory Networks 구현

- https://github.com/domluna/memn2n 코드베이스를 이용해서 한글 bAbI 데이터셋(Task 1, Task 20)에 대한 End-to-End Memory Networks를 구현해봅시다.

```
1 존은 복도로 갔다.
2 메리가 화장실로 갔다.
3 존은 어디 있습니까? 복도 1
4 다니엘은 화장실로 돌아갔다.
5 존이 침실로 이사했습니다.
6 메리는 어디 있습니까? 화장실 2
7 존이 복도로 갔다.
8 산드라가 부엌으로 여행했다.
9 산드라는 어디 있습니까? 부엌 8
10 산드라가 복도로 갔다.
11 존이 정원에 갔다.
12 산드라는 어디 있습니까? 복도 10
13 산드라는 화장실로 돌아갔다.
14 산드라가 부엌으로 이사했다.
15 산드라는 어디 있습니까? 부엌 14
1 산드라가 부엌으로 갔다.
2 산드라가 복도로 갔다.
3 산드라는 어디 있습니까? 복도 2
4 메리가 화장실에 갔다.
5 산드라가 정원으로 옮겼다.
```

Questions & Answers

Thank You!