

VAVPVDVPVG

+ + + + +

+

 $\uparrow \uparrow$ ۵Å. $v_3 v_4$

Because the function from infut b output is smooth, we can easily

propagate through it. Other recently proposed forms of memory or Because the thirt is a fram in put of potential of the potential of the propagate through it. Other recently proposed forms of memory or attention take this notably Rienerating/the indefiprediction, hethesingle layer case, the sum of

Generating the mina driver chois: then passed through a final weight matrix Werls input embed Ringues the president a final weight matrix W (of size $V \times d$) and to produce the predicted label: $\hat{a} = \text{Softmax}(W(o+u))$

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

Predicted / Answer / W

The men

Story (1: 1 supporting fact

ohn went to the bedroom Mary went to the office Where is John? Answer: bathro

Story (16: basic induct

What color is Greg? Answer: yell

(support) from the data

each hop used by the n

supporting sentences.

Example pr

of

hidden

Rrian is a fro

Brian is yellow

Julius is greer

Greg is a frog.

_ilv is arav

The overall model is shown in Fig. 1(a). During training, all three emb The overall model is phrase to frigs lear frequency transmission of the second structure of the secon

Learning Communi Abstraction with N

Embedding C	Weighted Sum			
	u Em	bedding B	A ¹	- <u>t</u>
		Question q	(b)	Questio

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

Hop 1

0.06

Hop 2

0.00

Hop 2 Hop 3

Hop 3

0.00

0.00

0.00

1.00

0.00

0.00

0 10

0.90

0.00

0.00

our model is somewhat simpler, not requiring operations like sharpening with model we apply the model to textual reasoning tasks, which qualitatively differ from the volte abstract our model to textual reasoning tasks, which qualitatively differ from the volte abstract our model to textual reasoning tasks, which qualitatively differ from the volte abstract our model to textual reasoning tasks, which qualitatively differ from the volte abstract our model to textual reasoning tasks, which qualitatively differ from the volte abstract our model to textual reasoning tasks, which qualitatively differ from the volte abstract our model to textual reasoning tasks, which qualitatively differ from the volte abstract our model to textual reasoning tasks, which qualitatively differ from the volte abstract our model to textual reasoning tasks, which qualitatively differ from the volte abstract our model to textual reasoning tasks, which qualitatively differ from the volte abstract our model to textual reasoning tasks, which qualitatively differ from the volte abstract our model to textual reasoning tasks, which qualitatively differ from the volte abstract our model to textual reasoning tasks, which qualitatively differ from the volte abstract our model to textual reasoning tasks are started abstract our model to textual reasoning tasks are started abstract. operations of sorting and recall tackled by the NTM.

$ \mathbf{F}$	rehoo	The model is a specific and the second se	operations.
L	accide.	and gated RNN based decoder were used for machine translation. The decoder uses an altention	ž ¹⁰
		model that finds which hidden states from the encoding are most useful for outputting the next	Story (1: 1 support
<i>m</i> -	<i>m</i>	translated word; the attention model uses a small neural network that takes as in filibile constationation is better)	Mary travelled to t
x_{t+2}	x_{t+3}	of the current hidden state of the decoder and each of the encoders hidden states. A similar attention	John went to the b
		model is also used in Xu et al. [23] for generating image captions. Our "memory" is analogous to	Mary went to the o
		their attention mechanism although [2] is only over a single sentence rather than many as in our	Where is John?

Rob Fergus

Work

"A B" at $1 \rightarrow v_A + v_B + v_1$

 x_{t+1}

 x_t

		see below that the	is Banjel went to the better and better	ves	0.06	0.00	
_		of the small netw	optimised tomogeneithe memories compared tooour scoringsapproachic weinesen simple	,00	0.00	0.00	
<u> </u>	W/1TD	linear layer, when	reash they lase the home sophisticated gated are partecine John moved to the hallway.	yes	0.00	0.00	
7	VVILII.		Mary weht to the office. A PO.01 0.00 0.00 Mary went back to the bedroom.	Prediction	0.00 In: hallway	0.00	-
		We also apply ou	minodel to language modeling, an extensively studied task. Goodman [6] showed				-
		simple but effect	Story(19; basin induction) h combineer then the Hond Hond Hond Story (19: 13: 19: 19: 19: 19: 19: 19: 19: 19: 19: 19	Support	Hop 1	Hop 2	_
		interest in using	Briantis's a frog. newral-network based models for theotasko with Konnstdate and dram and dram the chest.	yes	0.00	0.88	
		showing clear ne	The box is sugger man he should be the should be the should be the should be	yes	0.17	0.03	ŗ
	\mathbf{C} · 1	held by variants	Julius is green.		0.00	0.00	
		diagonal constrai	mWhat collects Grein?hAnswetryellow 5 Prediction: yespenon riate, which best instructions the book room of the construction of	Answer:	0.00 no Prec	diction: no	0
	Jampa	regarded as a m	odified form of RNN, where the recurrence is indexed by lookups to the word	-		1	
	•	sequence rather t	hprointexed by the selection on the QA tasks of [21]. We show the	labele	d supp	oorting	5
	0 111		(support) from the dataset which MemN2N does not use during training a	and the	proh	abilitie	ľ

the labeled supporting fac ing, and the probabilities p of

and Answering Experiments used by the model during inference. MemN2N successfully learns to focus on the correct Sukhbaatanthetic gi We perform expersive and the set of the set a set of statements, followed by a question whose answer is typically $\frac{1}{2}$ single word (in a few tasks¹⁸

case. Furthermore sour (moslobonia cosseveral sopportion pitano) y dos presentares an an international support

Overview

- Communication
 - Learnable communication protocol between neural net agents solving collaborative tasks
- Abstraction (ongoing work)
 - Hierarchy of actions in reinforcement learning for better scaling, planning and exploration

Communication Neural Network (CommNet)

- Input is a **set**
- Each element has its own processing stream
- Continuous broadcast communication channel between streams
- Streams must learn to communicate to solve task

CommNet Model



- Trained by backprop
- Invariant to order / number of inputs

С age metwe tion c DST PLANCA bre, Debi Camer nonvmation patilognote the Athes tosscance and the states AffiliationAff **Fin** Address Address a(1), states b_3 where the states a_3 and a_4 and a_5 and a_5 and a_5 and a_5 and a_5 and a_5 and action At step 54, the model parameters physical states in the strate between the strate of the strate between the strate betwee Worshow how the succession Show Now this are filled at this architecture car utbuts a there are a the second id. We $h^{1}i^{+}We$ set the set of th comsilenson contraction

Big Model Interpretation

- Set of streams = one big model
- Let f be single NN layer:



$$h_j^{i+1} = \sigma(H^i h_j^i + C^i c_j^i)$$

• N.B. Streams share parameters

$$T^{i} = \begin{pmatrix} H^{i} & C^{i} & C^{i} & \dots & C^{i} \\ C^{i} & H^{i} & C^{i} & \dots & C^{i} \\ C^{i} & C^{i} & H^{i} & \dots & C^{i} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C^{i} & C^{i} & C^{i} & \dots & H^{i} \end{pmatrix}$$

Dynamically sized: size of T can change depending on input set size

CommNet vs External Memory

NTM / MemNet

- Separate controller
- Serial processing



<u>CommNet</u>

- Distributed controller
- Parallel processing



Inputs

Experiment: 20 bAbI tasks



	Mean error (%)	Failed tasks (err. $> 5\%$)
LSTM [29]	36.4	16
MemN2N [29]	4.2	3
DMN+ [38]	2.8	1
Independent (MLP module)	15.2	9
CommNet (MLP module)	7.1	3



Meenoux yers itiates the ing a gradient to the model ons taken at each of those states paction standard and the stand of the stand and a stand a stand of the the sale of the sa Hans, Autor Hittes in an epise between the second states of the the ditain the second of the condition of the condition of the second of the Successful the top delivering the second all of the second the stopping akgnadienter the product at estimates of the solution of the solutio the second and the second states and the sec the and an ontobial of the size the **Huge**khietv hy state is the second teles and the second is the second $(tand)_r$ at a given time t (ommitting the time $-b(s(t), \theta)$) The time to the f(s(t))

Traffic Junction game

- Cars on fixed routes
- Two actions: gas/brake
- Limited visibility
- Text representation
- Variable # cars (max 20)
- Rewards:

Collision = -10Delay = -0.01t



Traffic Junction Movie



Traffic Junction Results



Failure rate

	Module $f()$ type		
Model Φ	MLP	RNN	LSTM
Independent	20.6 ± 14.1	19.5 ± 4.5	9.4 ± 5.6
Fully-connected	12.5 ± 4.4	34.8 ± 19.7	4.8 ± 2.4
Discrete comm.	15.8 ± 9.3	15.2 ± 2.1	8.4 ± 3.4
CommNet	2.2 ± 0.6	$\textbf{7.6}{\pm}\textbf{1.4}$	1.6 ± 1.0

How are the agents communicating?

PCA'd communication vectors



Corresponding hidden vectors



How are the agents communicating?

• Vectors from clusters correspond to distinct patterns of behavior:



Combat game





Experiment: Combat Game

- 5 agents vs 5 enemies in 15x15 map
- Health=3, Shot range=1, power=1, vision=1

	Module $f()$ type		
Model Φ	MLP	RNN	LSTM
Independent	34.2 ± 1.3	37.3 ± 4.6	44.3 ± 0.4
Fully-connected	17.7 ± 7.1	2.9 ± 1.8	19.6 ± 4.2
Discrete comm.	29.1 ± 6.7	33.4 ± 9.4	46.4 ± 0.7
CommNet	$44.5{\pm}13.4$	$\textbf{44.4}{\pm}\textbf{11.9}$	$49.5{\pm}12.6$

	Other game variations (MLP)		
Model Φ	m = 3	m = 10	$ 5 \times 5$ vision
Independent	29.2 ± 5.9	30.5 ± 8.7	60.5 ± 2.1
CommNet	51.0 ± 14.1	45.4 ± 12.4	73.0 ± 0.7

Related Work (I)

- Multi-agent Reinforcement Learning
 - Lots of papers on collaborative task solving
 - But usually communication protocol fixed
- Concurrent work:
 Learning to Communicate with Deep Multi-Agent
 Reinforcement Learning,
 Jakob N. Foerster, Yannis M.
 Assael, Nando de
 Freitas, Shimon Whiteson,
 NIPS 2016



Related Work (II)

- Graph Neural Networks
 - Gori et al., IJCNN 2005;
 - Scarselli et al., IEEE Trans. Neural Networks, 2009
- Gated Graph Neural Networks
 - Li, Zemel, Brockschmidt & Tarlow, ICLR 2016.



CommNet Summary

- Distributed NN model
 - Appropriate for tasks where input (and output) is set
- Models learn sparse communication protocol
- Can combine with RL for MARL problems
- Learning Multiagent Communication with Backpropagation, Sainbayar Sukhbaatar, Arthur Szlam, Rob Fergus, NIPS 2016
- Code: <u>https://github.com/facebookresearch/CommNet</u>

Learning Abstraction with NN (ongoing work)

- Abstraction is vital in cognitive tasks
- Abstraction of observations
 higher layers of ConvNets, hidden state of RNNs
- Abstraction of actions in reinforcement learning
- Going to work = thousands of steps/actions
 = get to station → take a train → walk to office
 - Deep RL models lack abstraction/hierarchy
 - Possible solution: recursive policy

Task-conditional RL

Usual RL settings have fixed task
 − Policy(Observation) → action

- Key idea: Generalize so agents also perceive task
 − Policy(Observation, Task) → action
- Different tasks for different episodes
- Tasks can be represented by embedding vector

Recursive Policy



Execution of recursive policy

function $\operatorname{RUN}(s, q, t)$ while $t < T_{\text{max}}$ do $a' \leftarrow \pi(s, q)$ if $a' \in A$ then $s \leftarrow \text{Environment.Act}(a')$ $t \leftarrow t + 1$ else if $a' = \langle \text{term} \rangle$ then return s, telse $q' \leftarrow a'$ $s, t \leftarrow \operatorname{Run}(s, g', t)$ end if end while end function

- Task Ids: $g \in G = \{1, ..., K\}$
- Environment actions: $a \in A = \{1, ..., M\}$
- Extended actions: $a' \in A' = A \cap G \cap \{< \text{term} > \}$
- Policy: $\pi: s \times g \to a'$

- During training, give small internal reward for using recursion
- Size of reward slowly increases with epoch

Related Work

- Options framework [Sutton'99], HAMs [Parr'98], MAXQ [Dietterich'00]
- P. Bacon, J. Harb, D. Precup. "**The Option-Critic Architecture**", 2016.
- J. Oh, S. Singh, H. Lee, P. Kolhi. "Communicating Hierarchical Neural Controllers for Learning Zero-shot Task Generalization", ICLR Submission, 2016
- J. Andreas, D. Klein, S. Levine. "Modular Multitask Reinforcement Learning with Policy Sketches", ICLR Submission, 2016 (borrow their tasks)
- J. Cai, R. Shin, D. Song. **"Making Neural Programming** Architectures Generalize via Recursion", ICLR Submission, 2016

Representation of tasks

- Fixed number of tasks → use their unique IDs
 -1="go to door", 2="lock door", 3="close door"
- Future: use natural language to describe a task
 - -Task = list of words
 - Ex) "close door", "close all doors", "find open door"
 - Generalization to unseen tasks?

Random Difficulty During Training :



Random Difficulty During Training :



Preliminary results

Tasks are randomly sampled from:

- 1. Grab wood
- 2. Grab iron
- 3. Grab rock
- 4. Make axe (wood, iron \rightarrow worktable)
- 5. Make sword (rock, iron \rightarrow factory)
- 6. Make hammer (wood, rock \rightarrow toolshed)
- 7. Make bridge(wood, iron, rock \rightarrow plant)

Model doesn't know which ones are easy No specification of sub-tasks in Make tasks



[Andreas et al. '16]

Training Details

- Using MazeBase [Sukhbaatar et al.'16]
- Policy is fully-connected NN with 2 hidden layers (50 units/layer)
- Trained with REINFORCE [Williams'92]
- Reward structure:
 - Each time step: -0.1
 - Complete task: +1
 - Use recursion (1 level): +0.02 [ramp; initially zero]

Model learns easy tasks first



Model learns when to use recursion



Comparison with non-recursive model

- Success rates on harder "Make" tasks
- 20 hidden units/layer



Trained Model



Summary

- Simple recursion-based RL approach
- Learns sub-task structure with minimal supervision

Future work:

- More than two levels of hierarchy
- More complex environment with diverse tasks
- Natural language for task description
- Learn to create a novel sub-task

Experiment: bag to sequence

• Problem: given a set of words, arrange them in right order.

{is, mouse, cat, chasing} \rightarrow "cat is chasing mouse"

- Separate streams for each words
- After 2 hops, each stream output its location
- Data: Gigaword, 5 words, 2 layer MLP as f

	5-gram by KenLM	Our model
Error per word	40%	26%

Fine-tuning of lower policies



The whole model a stochastic computation graph \rightarrow All discrete actions including internal decisions can be trained with policy gradient.

Only fine-tune lower policies, and not change its behavior

Initial experiments in grid world

	Failure rate on visit two goals	Reward + recursion reward
No recursion	3.11%	0.213
Recursive	1.02%	0.501
Recursive + fine-tune	0.11%	0.561

Model learns easy tasks first



Not using recursion when task is easy



Traffic Junction (Hard version)



Lever pulling task

- 500 agents with unique IDs
- In each episode, 5 of them randomly chosen
- There 5 different levers to pull
- If all agents pull different levers \rightarrow WIN

	Training method		
Communication	Supervised	Reinforcement	
None	0.59	0.59	
CommNet	0.99	0.94	

Experiment: bag to sequence

• Problem: given a set of words, arrange them in right order.

{is, mouse, cat, chasing} \rightarrow "cat is chasing mouse"

- Separate streams for each words
- After 2 hops, each stream output its location
- Data: Gigaword, 5 words, 2 layer MLP as f

	5-gram by KenLM	Our model
Error per word	40%	26%