

Because the function from $in\beta u\overline{t}$ but \overline{t} but \overline{t} is smooth, we can easily

propagate through it. Other recently proposed forms of memory or Because the this tight fram input of print is Grade which are realistic another gradients propagate through it. Other recently proposed forms of memory or attention take this notably Edenerating the final prediction, hethesingle layer case, the sum of

Generating the mba dingentin. the passed through a final weighte matrix Weil input embeddingly estile production. In the single rayer ease, the sum of the output vector input embeddingly estile product in the single rayer ease, the sum of the output vector input embeddingly estile product in the single rayer ease, the sum of the output vector input embeddingly estile product in the single rayer ease, the sum of the output vector input embedding of the product is the single rayer ease, the sum of the output vector input embedding of the output vector is the sum of the output vector input embedding of the output vector is the sum of the output vector input embedding of the output vector is the sum of the output vector is the out $\hat{a} = \text{Softmax}(W(o+u))$ to produce the predicted label:

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

The overall model is shown in Fig. 1(a). During training, all three emb The overall medical is phrave in fright lear fleer by training all the second diversion of as well as W are jointly learned by minimizing a standard cross-entropy loss between $\hat{a} \in [abel a.$ Training is performed using stochastic gradient descent (see Section 4.2 for more stochastic gradient descent (see Sectio

Intrinsic Motivation an Curricula via Asymme



$v_A v_P$	$v_D v_P v_G$	
+ +	+ + +	
$v_{\mathbf{B}} v_2$	$v_E v_G v_5$	
+	+ +	
v_1	$v_3 v_4$	
		C

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

 x_t

our model is contended in the simpler not requiring operations like sharpening. Fulling we can constrain several of the embedding matrices to be the our memory model to handle K hop operations. The memory layers are stac operations of sorting and recall tackled by the NTM $\uparrow \uparrow$ АВ" "Р" model to handle Kathop operations. The mem Our model is also related to Bahdanau *et al.* [2]. In that work, a bidirectional RNN based encoder and gated RNN based decoder were used for machine translation. The decoder uses an attention 10k maining data Story (1: 1 supporting fact) model that finds which hidden states from the encoding are most useful for outputting the next translated word; the attention model uses a small neural network that takes as in#tituiedomsat(smatten is better) $\begin{array}{c} x_{t+1} \\ x_{t+2} \\ x_{t+2} \\ x_{t+2} \\ x_{t+3} \\ x_{t+3}$ ohn went to the bedroom ohn travelled to the bathr Mary went to the office Where is John? Answer: bathroo case. Furthermore sour (modebonink as several sopso of Hand) Hand thank for smalling support for smalling support Hop 1 Hop 2 Hop 3 Story (16: basic inductio 0.06 0.00 0.00 see below that this Banel were to the better ownod performance of the read of Brian is a froo 0.88 0.00 1 00 Lily is gray. of the small networkhised to source the name and the memories compared to our sooring sapproach to source the source to source the source to sourc 0.00 0.00 0.00 Brian is yellow linear layer, where the lite office. So the are particular and par John moved to the hallway. Mary went back to the bedroon 0.00 0.00 1.00 Julius is green yes 0.00 0.00 0.00 Greg is a frog. Arthur Szl We also apply our model to language modeling, an extensively studied task. Goodman to showed we have a strong the strong to the strong tot the strong to the strong to the What color is Greg? Answer: yell Hop 2 Hop 3 0.00 0.00 interest in using hearts a troc which based models var the the state of the state o Figure 2: Example pr 0.04 0.05 0 10 0.17 0.07 0.90 x_{t+1} Facebo shorting clear performance gains over traditional inclusion in the chest file inside the contrainer, with $\frac{0.00}{0.00}$ and \frac (support) from the data 0.00 each hop used by the n 0.00 supporting sentences. regarded as a modified form of RNN, where the recurrence is indexed by lookups to the word AI Reseat Crime the superior from the dataset which MemN2N does not use during training, and the problem of the superior from the dataset which MemN2N does not use during training, and the problem of the superior from the dataset which MemN2N does not use during training, and the problem of the superior from the dataset which MemN2N does not use during training and the problem of the superior from the dataset which MemN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during training and the problem of the superior from the dataset which memN2N does not use during trainformation of the problem of the superior from th facts p of den We perform expersing sentence QA tasks defined in [21]. A given QA task consists of RNN 151 -300mN21 MemN2N a set of statements followed by a question whose answer is typically $\frac{1}{3}$ single word (in a few tasks¹⁸)

Motivation

- Reinforcement Learning (RL) typically requires a **huge** number of episodes
- Often **supervision** signal (i.e. reward) is **expensive** to obtain
- Can we learn about environment in **unsupervised** way?
- Assumption: interaction with the environment is cheap





- Agent plays a **game** where it challenges itself
- Single physical agent, but two separate minds:
 - Alice's job is to propose a task

Approach

- Bob's job is to **complete** that task
- Alice propose a task by actually **doing** it
- We consider two classes of environments:
 - 1. Actions are reversible within same time \rightarrow reverse self-play
 - 2. Reset to the initial state is allowed \rightarrow repeat self-play
- Jointly train with **self-play** and **target task**
 - Randomly choose type of episode

Reverse self-play



Internal reward during self-play

• Bob's reward:

Alice's reward:

$$R_b = -t_b$$

Time spent

Intuition: make Bob fail with less effort

 $R_a = \max(0, t_b - t_a)$

If Bob fails: $t_b = t_{\max}$

- Alice's optimal behavior is to find simplest tasks that Bob cannot complete.
- Makes learning for Bob easy since the new task will be only just beyond his current capabilities.
- Gives self-regulating feedback between Alice and Bob
 - Yields automatic curriculum

Parameterizing Policy Functions



• Target task:
$$a_{\text{Target}} = f_B(s_t'', e)$$

task description (dummy vector)

- Self-play lets Bob build representation of environment
- Assumption: self-play tasks are close to target task
- Explore discrete / continuous settings
 - Using small NN for f(.)

Self-play equilibrium & Universal Bob

- Claim: Under some strong assumptions (tabular policies, finite state, etc.), Bob must learn all possible tasks, i.e. learn how to transition between any pair of states as efficiently as possible.
- Let's assume the self-play has converged to a Nash equilibrium (can't gain anything if other's policy is fixed)
- If Bob fails on a certain task, then Alice would propose that task to increase her reward
- Then Bob must've seen this task and learnt it to increase his reward
- Thus: Bob must have learned all possible tasks.

Related work

- Self-play: checkers (Samuel, 1959), backgammon (Tesauro, 1995), and Go, (Silver et al., 2016), and RoboSoccer (Riedmiller et al., 2009)
 - Uses external reward vs internal reward for ours
- GANs (Goodfellow et al., 2014): dialogue generation (Li et al., 2017), variational auto-encoders (Mescheder et al., 2017)
 - Alice \rightarrow "generator" of hard examples; Bob \rightarrow "discriminator"
- Intrinsic motivation (Barto, 2013; Singh et al., 2004; Klyubin et al., 2005; Schmidhuber, 1991): curiosity-driven exploration (Schmidhuber, 1991; Bellemare et al., 2016; Strehl & Littman, 2008; Lopes et al., 2012; Tang et al., 2016)
 - Reward for novelty of state
 - Ours: learning to transition between pairs of states
- Robust Adversarial Reinforcement Learning (Pinto et al. 2017)
 - Concurrent work; adversarial peturbations to state

Experiments

- Use **Reinforce** algorithm with learnt baseline and entropy regularization
- 2-layer NN model for Alice and Bob (separate)
- Train on 20% target task + 80% self-play episodes
- Discrete and continuous environments
- Measure target task reward vs # target task episodes
 - Self-play episodes are "free"
- Baselines:
 - No self-play: just target task episodes
 - Random Alice: Alice takes random actions. Bob learns policy
 - Exploration approaches: count-based & variants

Toy example: Long hallway

- Learn to navigate in a long corridor
- Reverse self-play
- Simple tabular policies



MazeBase: LightKey task

- Small 2D grid separated into two rooms by a wall
- The grid is procedurally generated
 - Object/agent locations randomized for each episode
- Toggle the key to lock/unlock door
 - Can't go through a locked door
- Toggle the light on/off
 - Only the switch is visible in dark
- Target task is to reach the goal flag in the opposite room when light is off and door is locked.



MazeBase: LightKey task

• Learn to navigate in a long corridor



MazeBase: LightKey task



RL-Lab: Mountain Car

- Control a car stuck in 1D valley
 - Need to build momentum by reversing
- Sparse reward
 - +1 reward only if it reaches the left hill top
- Hard task because random exploration fails
- Asymmetric environment
 repeat self-play
- As good as other exploration methods





- Control a worm with two flexible joints, swimming in a 2D viscous fluid
- Reward +1 for eating green apples and -1 for touching red bombs
- Reverse self-play even though the environment is not strictly symmetric
- No apples or bombs during self-play
- Use only location (not full state) when deciding Bob's success during self-play



- Mean & S.D. over 10 runs
- Reinforce on target task alone gets zero reward



• Policy trained with Reinforce + self-play







• Distribution of locations where Alice hands over to Bob



Discussion Paper: https://arxiv.org/abs/1703.05407

- Simple methods that works with discrete and continuous environments
- Meta–exploration for Alice
 - We want Alice to propose diverse set of tasks
 - But Alice focuses on the single best task
 - Multiple Alices?
- Future works:
 - Alice explicitly mark the target state
 - Alice propose task by communication without doing it
 - Alice propose a hypothesis and Bob test it