ELEC 677:
Recurrent Neural Network Language Models & Deep Reinforcement Learning
Lecture 10

Ankit B. Patel, CJ Barberan
Baylor College of Medicine (Neuroscience Dept.)
Rice University (ECE Dept.)
11-15-2016

Latest News

NEURAL ARCHITECTURE SEARCH WITH REINFORCEMENT LEARNING

Barret Zoph,* **Quoc V. Le**
Google Brain
{barretzoph,qvl}@google.com

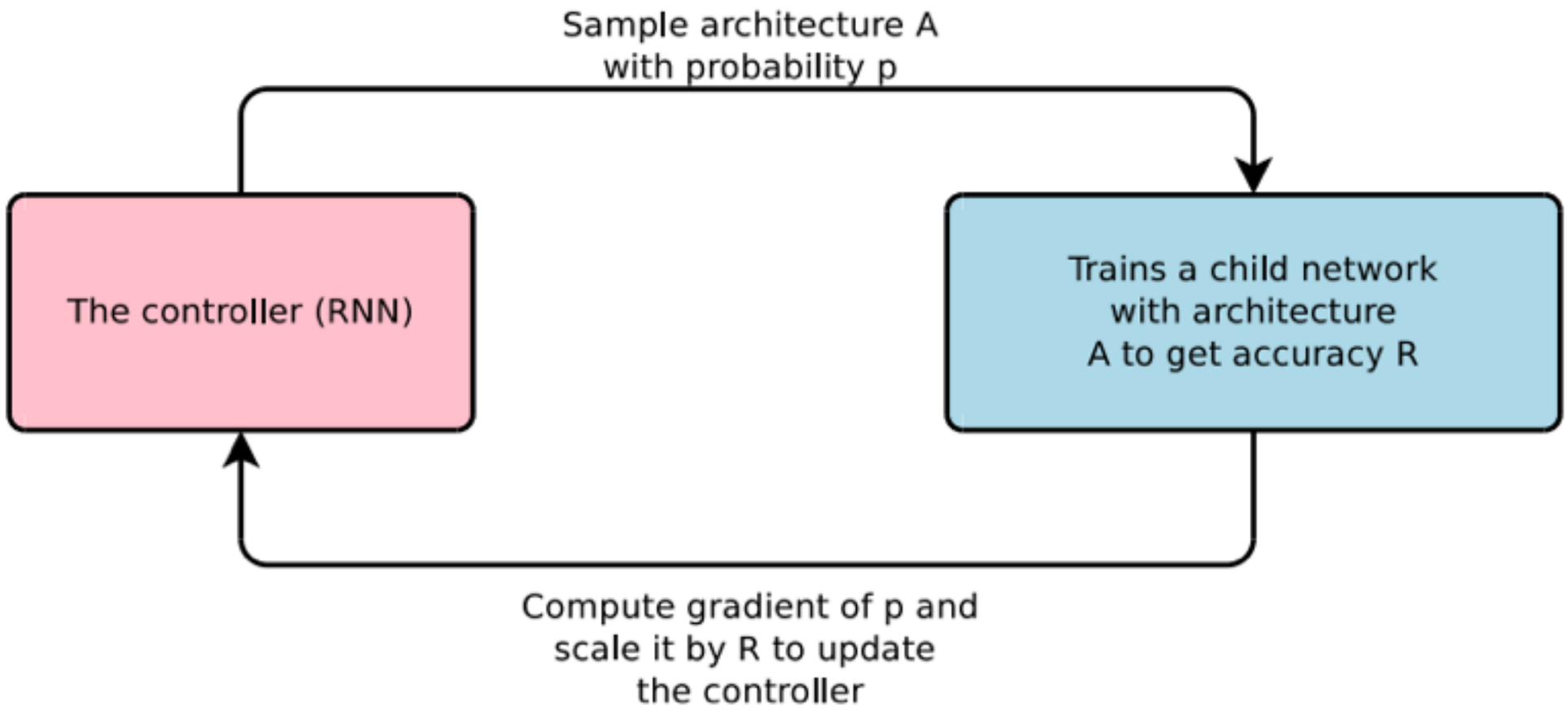


Figure 1: An overview of Neural Architecture Search.

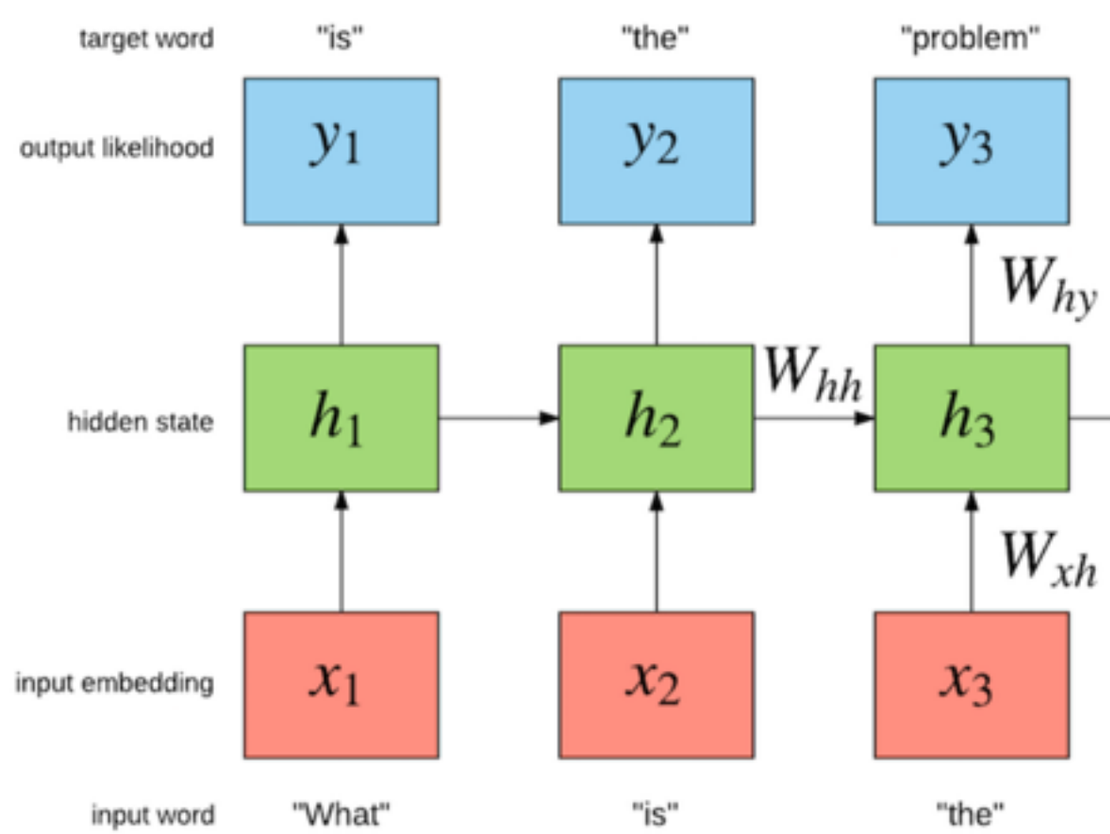
Word-level RNN Language Models

Motivation

- Model the probability distribution of the next word in a sequence, given the previous words

- Words are the minimal unit to provide meaning

- Another step to a hierarchical model



[Nicholas Leonard]

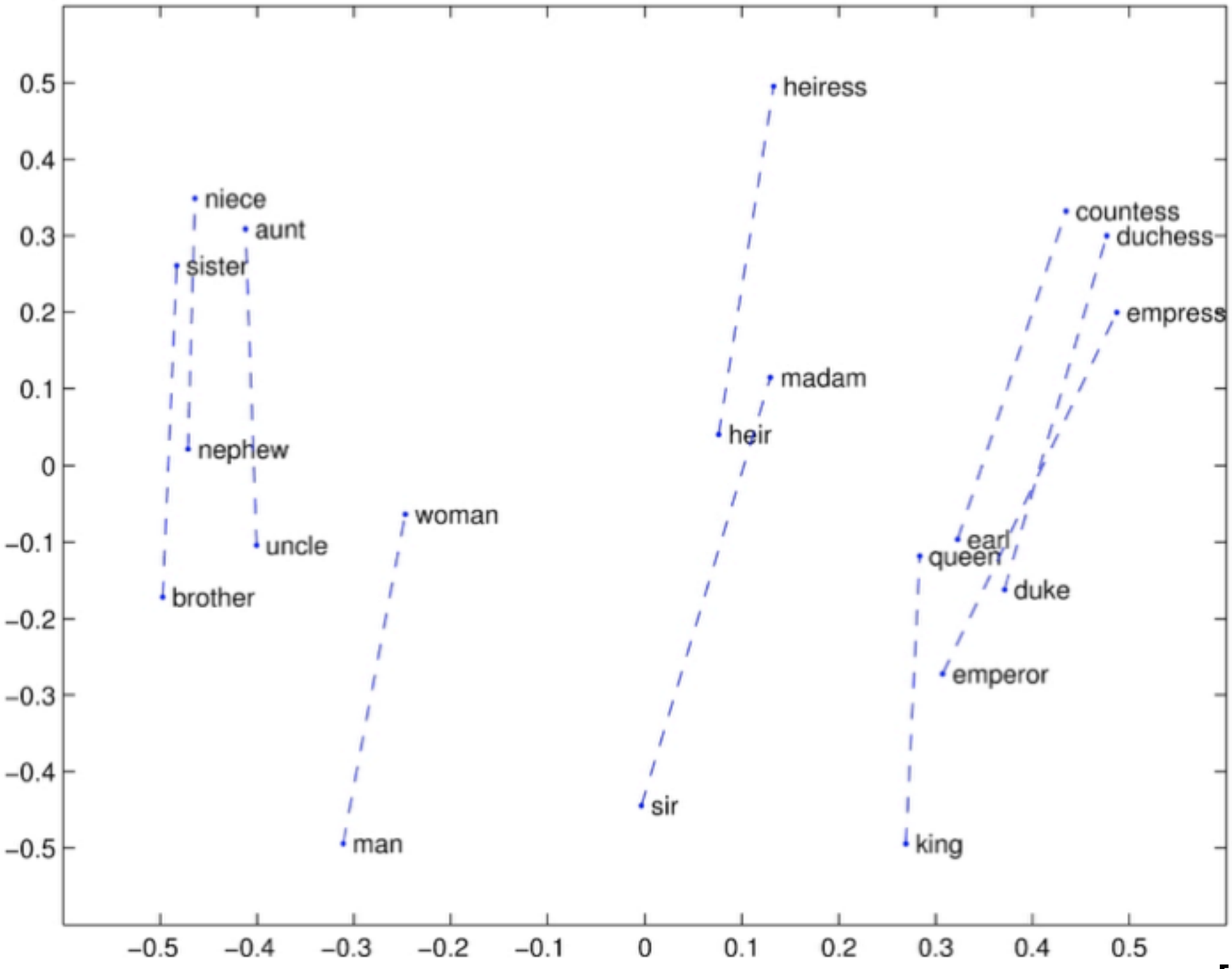
Global Vectors for Word Representation (GloVe)

- Provide semantic information/context for words
- Unsupervised method for learning word representations

$$J(\theta) = \frac{1}{2} \sum_{i,j=1}^{W} f(P_{ij})(u_i^T v_j - \log P_{ij})^2$$

[Richard Socher]

Glove Visualization



[Richard Socher]

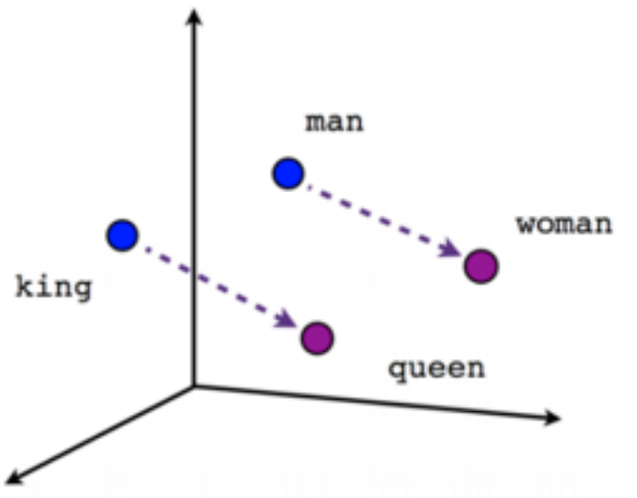
Word2Vec

- Learn word embeddings
- Shallow, two-layer neural network
  - Training makes observed word-context pairs have similar embeddings, while scattering unobserved pairs. Intuitively, words that appear in similar contexts should have similar embeddings
- Produces a vector space for the words

$$= \arg\max_{\theta} \sum_{(w,c) \in D} \log \sigma(v_c \cdot v_w) + \sum_{(w,c) \in D'} \log \sigma(-v_c \cdot v_w)$$

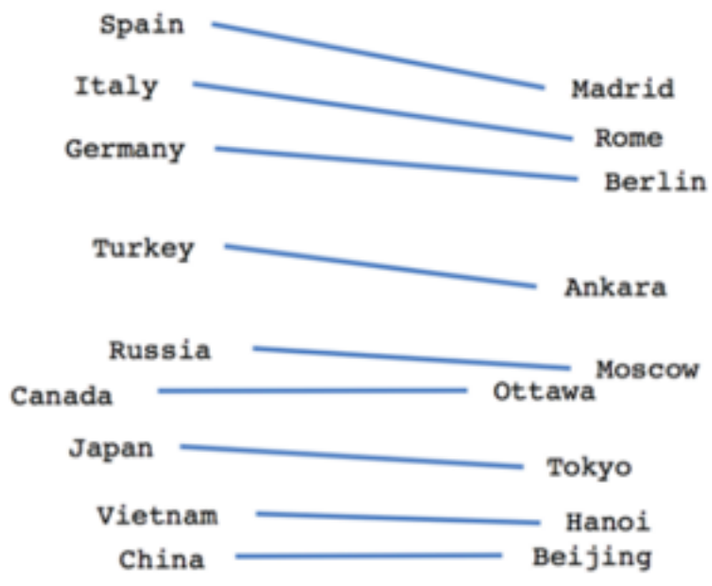
[Goldberg, Levy Arxiv 2014]

Word2Vec Visualization



Male-Female

Verb tense



Country-Capital

[Tensorflow]

Understanding Word2Vec

$$\text{word } w \in V_W$$

$$\text{context } c \in V_C \qquad w_{i-L}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+L}.$$

Probability that word context pair taken from document

$$P(D = 1|w, c) = \sigma(\vec{w} \cdot \vec{c}) = \frac{1}{1 + e^{-\vec{w} \cdot \vec{c}}}$$

[Goldberg, Levy Arxiv 2014]

Understanding Word2Vec

Maximize likelihood real context pairs come from document

$$P(D = 1|w, c) \qquad P(D = 0|w, c)$$

$$\ell = \sum_{w \in V_W} \sum_{c \in V_C} \#(w, c) \left(\log \sigma(\vec{w} \cdot \vec{c}) + k \cdot \mathbb{E}_{c_N \sim P_D} \left[\log \sigma(-\vec{w} \cdot \vec{c}_N) \right] \right)$$

[Goldberg, Levy Arxiv 2014]

Word2Vec as Word-Context Association Matrix Decomposition

Solution is optimal parameters obey relation:

$$\vec{w} \cdot \vec{c} = \log\left(\frac{\#(w,c) \cdot |D|}{\#(w) \cdot \#(c)} \cdot \frac{1}{k}\right) = \log\left(\frac{\#(w,c) \cdot |D|}{\#(w) \cdot \#(c)}\right) - \log k$$

Pointwise Mutual Information

$$PMI(w,c) = \log \frac{\#(w,c) \cdot |D|}{\#(w) \cdot \#(c)}$$

$$PMI(x,y) = \log \frac{P(x,y)}{P(x)P(y)}$$

1. Construct word context association matrix
2. Low rank decomposition

$$M_{ij} = PMI(w,c)$$

$$W \cdot \bar{C}^{\intercal} = \bar{M}$$

[Goldberg, Levy Arxiv 2014]

Question Time

- Given the theoretical understanding of word2vec, what kinds of things will word2vec not capture well?

- Can you think of ways to make it better?

Word2vec with RNNs

Method	Adjectives	Nouns	Verbs	All
LSA-80	9.2	11.1	17.4	12.8
LSA-320	11.3	18.1	20.7	16.5
LSA-640	9.6	10.1	13.8	11.3
RNN-80	9.3	5.2	30.4	16.2
RNN-320	18.2	19.0	45.0	28.5
RNN-640	21.0	25.2	54.8	34.7
RNN-1600	**23.9**	**29.2**	**62.2**	**39.6**

Table 2: Results for identifying syntactic regularities for different word representations. Percent correct.

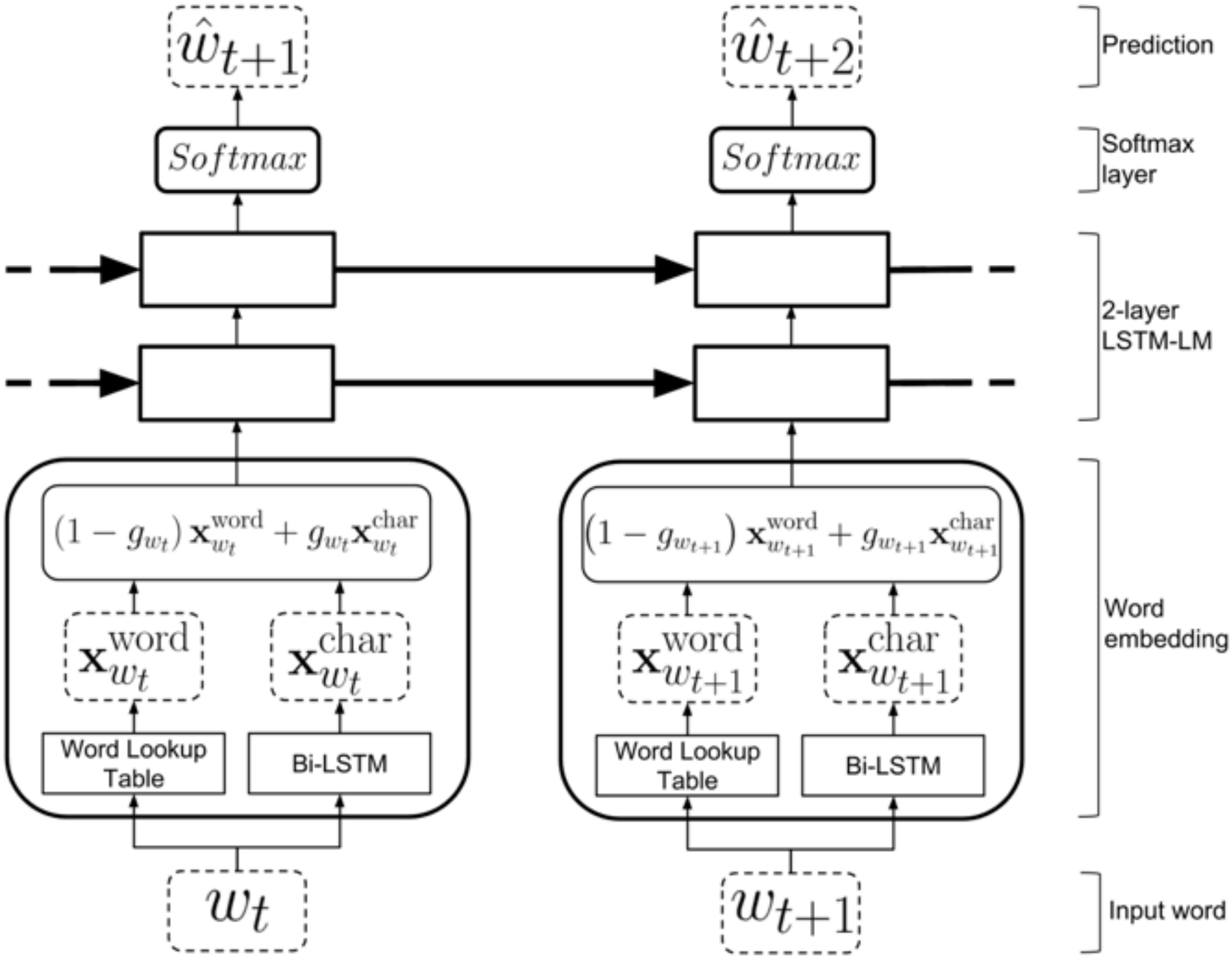
[Mikolov, Yih, Zweig]

Word RNN trained on Shakespeare

```
LEONTES:
Why, my Irish time?
And argue in the lord; the man mad, must be deserved a spirit as drown the warlike Pray him, how seven in.

KING would be made that, methoughts I may married a Lord dishonour
Than thou that be mine kites and sinew for his honour
In reason prettily the sudden night upon all shalt bid him thus again. times than one from mine unaccustom'

LARTIUS:
O,'tis aediles, fight!
Farewell, it himself have saw.

SLY:
Now gods have their VINCENTIO:
Whipt fearing but first I know you you, hinder truths.

ANGELO:
This are entitle up my dearest state but deliver'd.

DUKE look dissolved: seemeth brands
That He being and
full of toad, they knew me to joy.
```

[Sung Kim]

Gated Word RNN



[Miyamoto, Cho]

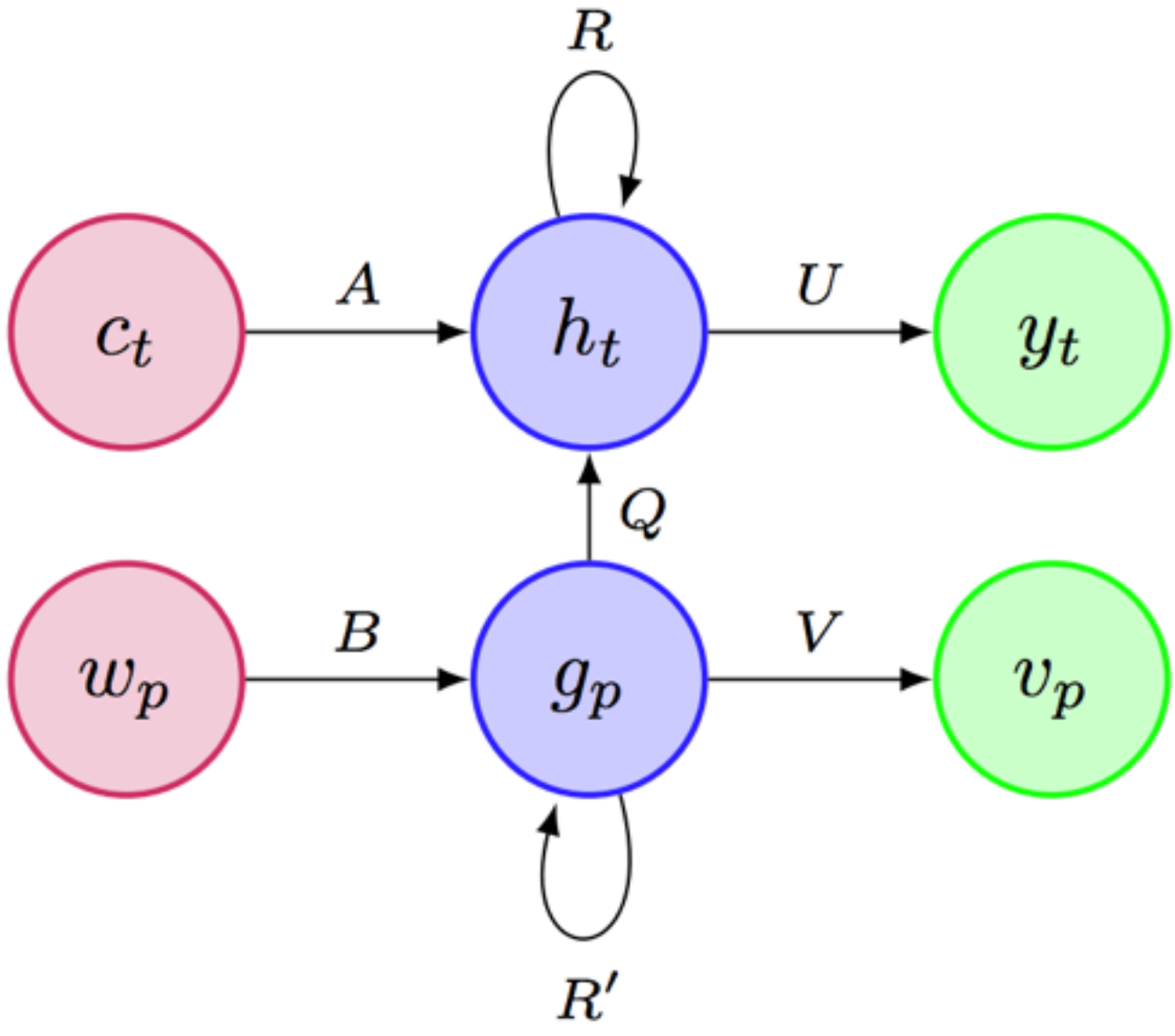
Gated Word RNN Results

Model	PTB		BBC		IMDB	
	Validation	**Test**	**Validation**	**Test**	**Validation**	**Test**
Gated Word & Char, adaptive	117.49	113.87	**78.56**	**87.16**	71.99	72.29
Gated Word & Char, adaptive (Pre-train)	117.03	112.90	80.37	87.51	71.16	71.49
Gated Word & Char, $g = 0.25$	119.45	115.55	79.67	88.04	71.81	72.14
Gated Word & Char, $g = 0.25$ (Pre-train)	**117.01**	**113.52**	80.07	87.99	**70.60**	**70.87**
Gated Word & Char, $g = 0.5$	126.01	121.99	89.27	94.91	106.78	107.33
Gated Word & Char, $g = 0.5$ (Pre-train)	117.54	113.03	82.09	88.61	109.69	110.28
Gated Word & Char, $g = 0.75$	135.58	135.00	105.54	111.47	115.58	116.02
Gated Word & Char, $g = 0.75$ (Pre-train)	179.69	172.85	132.96	136.01	106.31	106.86
Word Only	118.03	115.65	84.47	90.90	72.42	72.75
Character Only	132.45	126.80	88.03	97.71	98.10	98.59
Word & Character	125.05	121.09	88.77	95.44	77.94	78.29
Word & Character (Pre-train)	122.31	118.85	84.27	91.24	80.60	81.01
Non-regularized LSTM (Zaremba, 2014)	120.7	114.5	-	-	-	-

Table 1: Validation and test perplexities on Penn Treebank (PTB), BBC, IMDB Movie Reviews datasets.

[Miyamoto, Cho]

Combining Character & Word Level



[Bojanowski, Joulin, Mikolov]

Question Time

- In which situation(s) can you see character-level RNN more suitable than a word-level RNN?

Generating Movie Scripts

- LSTM named Benjamin
    - Learned to predict which letters would follow, then the words and phrases
- Trained on corpus of past 1980 and 1990 sci-fi movie scripts
- "I'll give them top marks if they promise never to do this again."
- https://www.youtube.com/watch?v=LY7x2Ihqjmc

Character vs Word Level Models

Character vs Word-Level Models

	EN-Wikipedia				EN-WSJ			
	Acc.	P	R	F_1	Acc.	P	R	F_1
					Word-based Approach			
LM ($N = 3$)	94.94	89.34	84.61	86.91	95.59	91.56	78.79	84.70
LM ($N = 5$)	94.93	89.42	84.41	86.84	95.62	91.72	78.79	84.77
CRF-WORD	96.60	94.96	87.16	90.89	97.64	93.12	90.41	91.75
Chelba and Acero (2006)			n/a		97.10	-	-	-
					Character-based Approach			
CRF-CHAR	96.99	94.60	89.27	91.86	97.00	94.17	84.46	89.05
LSTM-SMALL	96.95	93.05	90.59	91.80	97.83	93.99	90.92	***92.43***
LSTM-LARGE	97.41	93.72	92.67	***93.19***	97.72	93.41	90.56	91.96
GRU-SMALL	96.46	92.10	89.10	90.58	97.36	92.28	88.60	90.40
GRU-LARGE	96.95	92.75	90.93	91.83	97.27	90.86	90.20	90.52

[Kim, Jernite, Sontag, Rush]

Word Representations of Character & Word Models

	In Vocabulary					Out-of-Vocabulary		
	while	*his*	*you*	*richard*	*trading*	*computer-aided*	*misinformed*	*looooook*
LSTM-Word	*although*	*your*	*conservatives*	*jonathan*	*advertised*	–	–	–
	letting	*her*	*we*	*robert*	*advertising*	–	–	–
	though	*my*	*guys*	*neil*	*turnover*	–	–	–
	minute	*their*	*i*	*nancy*	*turnover*	–	–	–
LSTM-Char (before highway)	*chile*	*this*	*your*	*hard*	*heading*	*computer-guided*	*informed*	*look*
	whole	*hhs*	*young*	*rich*	*training*	*computerized*	*performed*	*cook*
	meanwhile	*is*	*four*	*richer*	*reading*	*disk-drive*	*transformed*	*looks*
	white	*has*	*youth*	*richter*	*leading*	*computer*	*inform*	*shook*
LSTM-Char (after highway)	*meanwhile*	*hhs*	*we*	*eduard*	*trade*	*computer-guided*	*informed*	*look*
	whole	*this*	*your*	*gerard*	*training*	*computer-driven*	*performed*	*looks*
	though	*their*	*doug*	*edward*	*traded*	*computerized*	*outperformed*	*looked*
	nevertheless	*your*	*i*	*carl*	*trader*	*computer*	*transformed*	*looking*

Table 6: Nearest neighbor words (based on cosine similarity) of word representations from the large word-level and character-level (before and after highway layers) models trained on the PTB. Last three words are OOV words, and therefore they do not have representations in the word-level model.

[Kim, Jernite, Sontag, Rush]

Other Embeddings

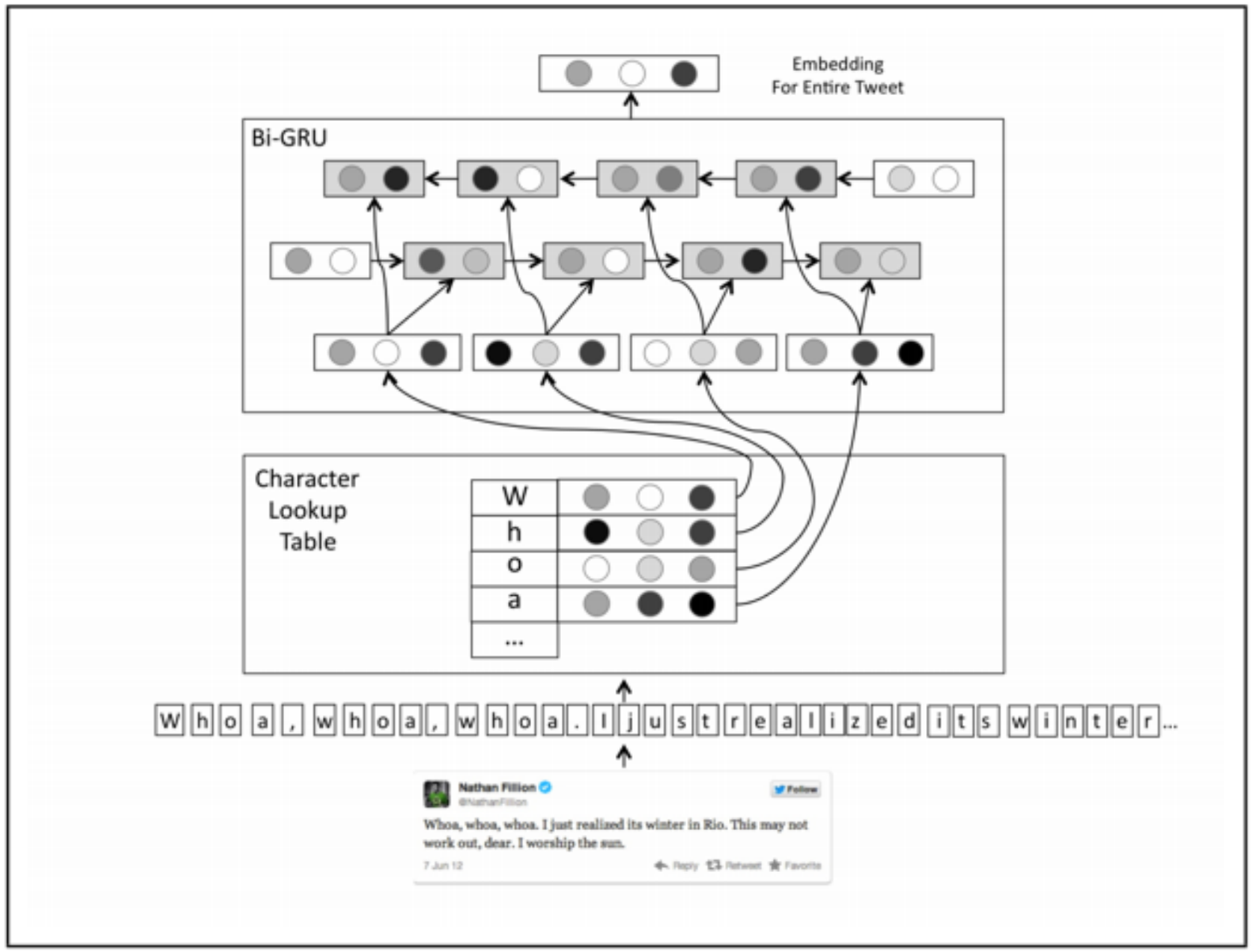
Tweet2Vec

$$J = \frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{L} -t_{i,j}log(p_{i,j}) + \lambda\|\Theta\|^2. \quad (3)$$

Here B is the batch size, L is the number of classes, $p_{i,j}$ is the predicted probability that the i-th tweet has hashtag j, and $t_{i,j} \in \{0, 1\}$ denotes the ground truth of whether the j-th hashtag is in the i-th tweet. We use L2-regularization weighted by λ.

[Dhingra, Zhou, Fitzpatrick, Muehl, Cohen]

Tweet2Vec Encoder



[Dhingra, Zhou, Fitzpatrick, Muehl, Cohen]

Tweet2Vec Results

Tweets	Word model baseline	*tweet2vec*
ninety-one degrees. ☀❤😎	#initialsofsomeone.. #nw #gameofthrones	#summer **#loveit** #sun
self-cooked scramble egg. yum!! !url	#music #cheap #cute	**#yummy** #food #foodporn
can't sleeeeeeeep	#gameofthrones #heartbreaker	#tired **#insomnia**
oklahoma!!!!!!!!!! champions!!!!!	#initialsofsomeone.. #nw #lrt	**#wcws** #sooners #ou
7 % of battery . iphones die too quick .	#help #power #money #s	#fml #apple #bbl **#thestruggle**
i have the cutest nephew in the world !url	#nephew **#cute** #family	#socute **#cute** #puppy

Table 1: Examples of top predictions from the models. The correct hashtag(s) if detected are in bold.

[Dhingra, Zhou, Fitzpatrick, Muehl, Cohen]

Gene2Vec

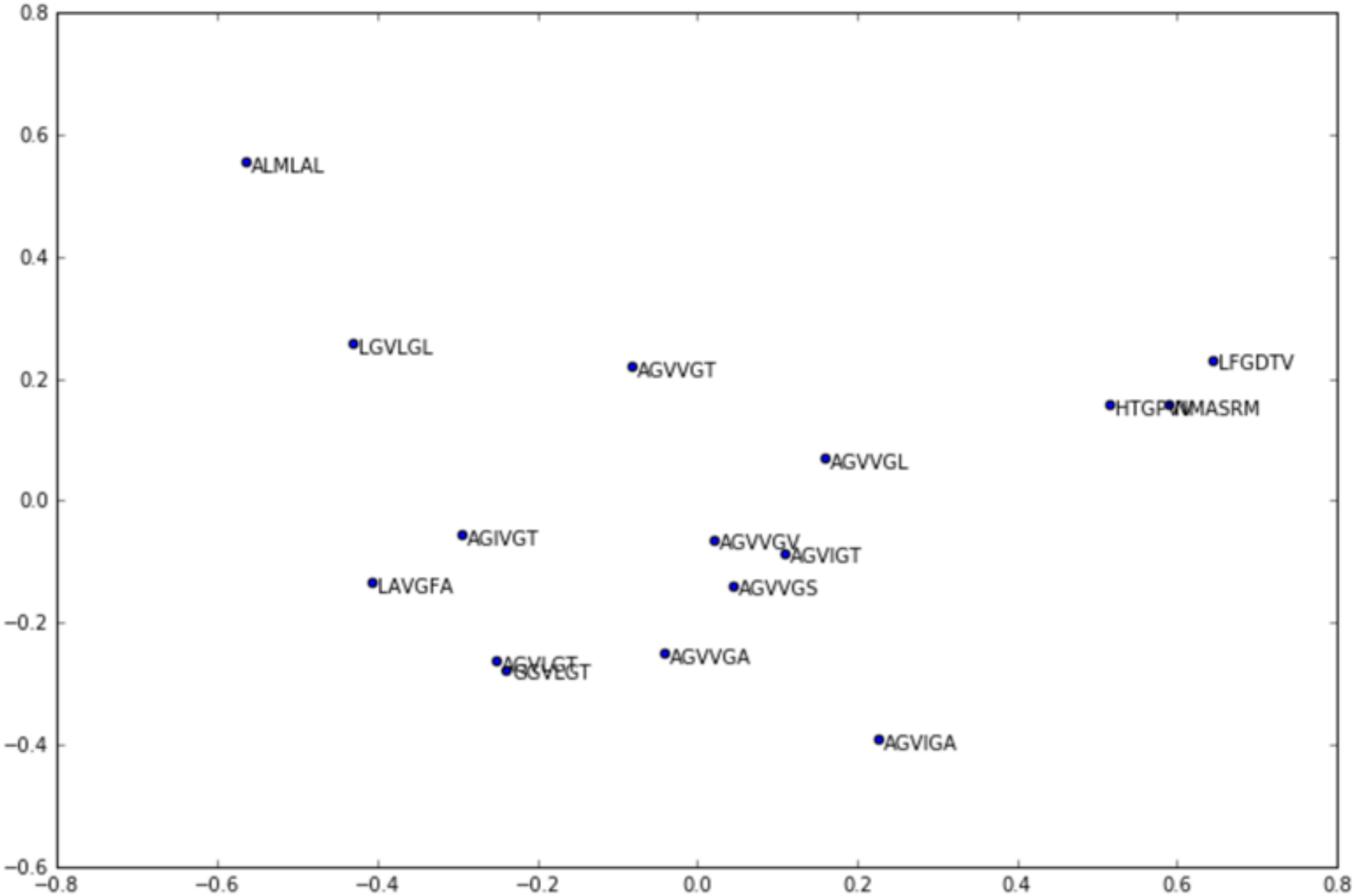
- Word2Vec performs poorly on long nucleotide sequences

- Short sequences are very common like AAAGTT

```
AAAAATAGTATAAAAAGTTGCCAAAAG ->  Triticum aestivum chromosome 3B
|||||  |     | |||    |
AAAAACATGCAACAAACAGGAACTGGC ->  Triticum aestivum chromosome 3B
|||||||           |   |      |
AAAAACAGAATCTGTCTAAAACAGAAC ->  Triticum aestivum chromosome 3B
|||||||||   |   |   |    |   |
AAAAACAGAGACATTACTTTGCCAACA ->  Ovis canadensis canadensis isolate 43U chromosome 26
```

[David Cox]

Gene2Vec Visual



Hydrophobic Amino Acids

[David Cox]

Doc2Vec

- Similar to Word2Vec but to a larger scale
  - Sentences & Paragraphs

```
TARGET (72927): «this is one of the best films of this year . for a year that was fueled by contro
versy and crap , it was nice to finally see a film that had a true heart to it . from the opening
 scene to the end , i was so moved by the love that will smith has for his son . basically , if yo
u see this movie and walk out of it feeling nothing , there is something that is very wrong with y
ou . loved this movie , it's the perfect movie to end the year with . the best part was after the
 movie , my friends and i all got up and realized that this movie had actually made the four of us
 tear up ! it's an amazing film and if will smith doesn't get at least an oscar nom , then the osc
ars will just suck . in fact will smith should actually just win an oscar for this role . ! ! ! i
 loved this movie ! ! ! ! everybody needs to see especially the people in this world that take eve
rything for granted , watch this movie , it will change you !»

SIMILAR/DISSIMILAR DOCS PER MODEL Doc2Vec(dm/m,d100,n5,w10,mc2,t8):

MOST (2046, 0.737233221530914 3): «i thought this movie would be dumb , but i really liked it . peo
ple i know hate it because spirit was the only horse that talked . well , so what ? the songs were
 good , and the horses didn't need to talk to seem human . i wouldn't care to own the movie , and
 i would love to see it again . 8/10»
```

[RaRe Technologies]

Applications of Document Models

- Discovery of litigation e.g. CS Disco
- Sentiment Classification e.g. movie reviews

Reinforcement Learning

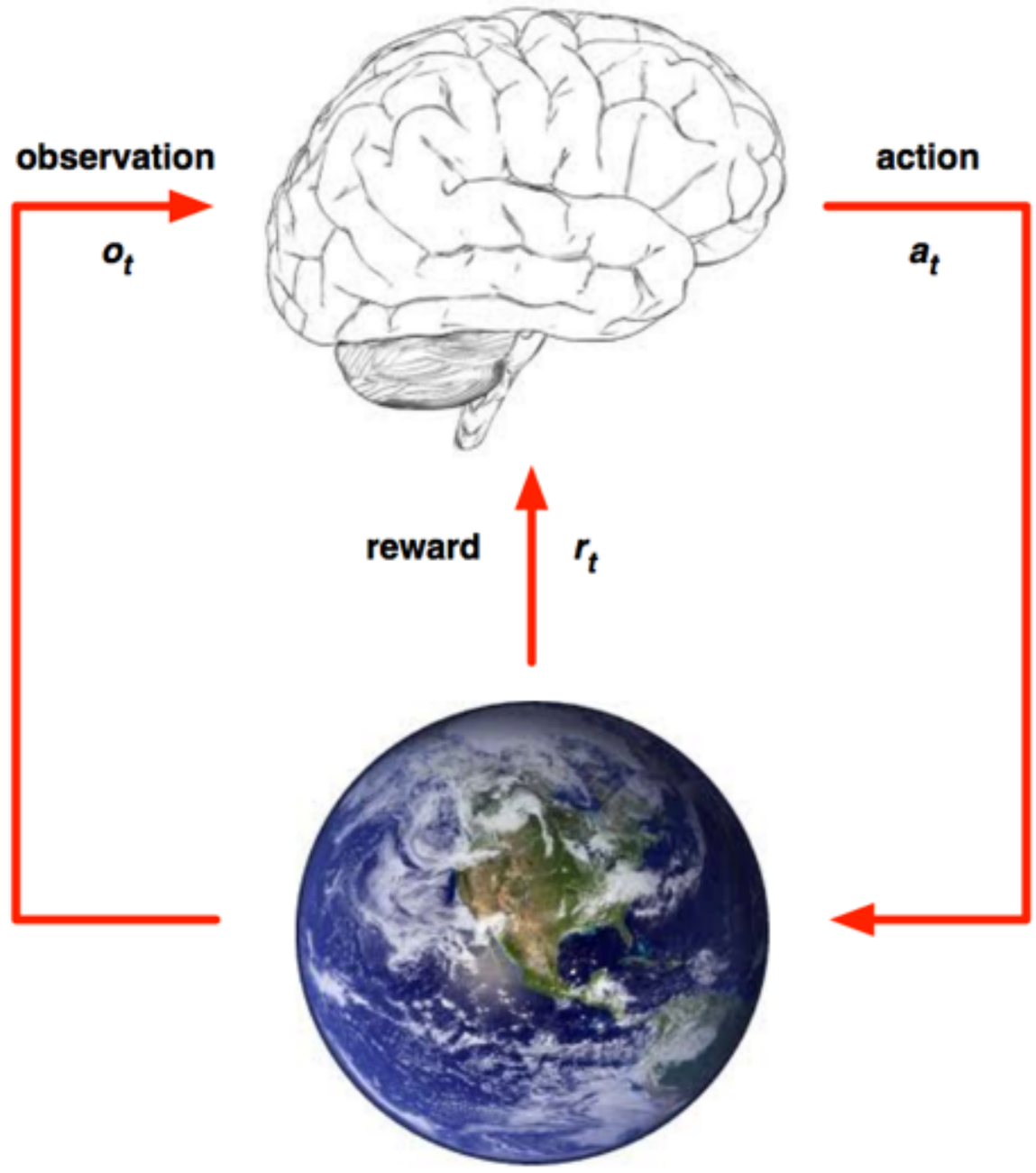
What is Reinforcement Learning

- Reinforcement Learning (RL) is a framework for decision-making

  - Have an agent with acts in an environment
  - Each action influences the agent's future state
  - Success is measured by reward
  - Find actions that maximise your future reward

Agent and Environment

- Agent
  - Executes action
  - Receives observation
  - Receives reward
- Environment
  - Receives action
  - Sends new observation
  - Sends new reward



[David Silver ICML 2016]

Defining the State

- Sequence of observations, rewards, and actions define experience

$$o_1, r_1, a_1, \ldots, a_{t-1}, o_t, r_t$$

- State is a summary of experiences

$$s_t = f(o_1, r_1, a_1, \ldots, a_{t-1}, o_t, r_t)$$

[David Silver ICML 2016]

Components of an RL Agent

- Policy
  - Agent's behavior function
- Value Function
  - Determine if each state or action is good
- Model
  - Agent's representation

Policy

- Policy is the agent's behavior
- Policy is a map from state to action
  - Deterministic $a = \pi(s)$
  - Stochastic $\pi(a|s) = \mathbb{P}[a|s]$

[David Silver ICML 2016]

Value Function

- Value function is a prediction of the future reward
  - What will my reward be given this action a from state s?
  - Example: Q-value function
    - State s, action a, policy π , discount factor γ

$$Q^{\pi}(s,a) = \mathbb{E}\left[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots \mid s,a\right]$$

[David Silver ICML 2016]

Optimal Value Function

- The goal is to have the maximum achievable

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a) = Q^{\pi^*}(s,a)$$

- With that, can act optimally with the policy

$$\pi^*(s) = \operatorname*{argmax}_{a} Q^*(s,a)$$

- Optimal value maximises over all the decisions

$$Q^*(s,a) = r_{t+1} + \gamma \max_{a_{t+1}} r_{t+2} + \gamma^2 \max_{a_{t+2}} r_{t+3} + \ldots$$

$$= r_{t+1} + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})$$

[David Silver ICML 2016]

Model

- The model is learned through experience
- Can act as a proxy for the environment

RL Approaches

- Value-based RL
  - Estimate the optimal value $Q^*(s, a)$
  - Maximum value achievable under any policy
- Policy-based RL
  - Search for the optimal value π^*
  - Policy achieving maximum future reward
- Model-based RL
  - Build a model of the environment
  - Plan using a model

[David Silver ICML 2016]

Deep Reinforcement Learning

Deep RL

- To have Deep RL, we need
  - RL + Deep Learning (DL) = AI
    - RL defines the objective
    - DL provides the mechanism to learn

Deep RL

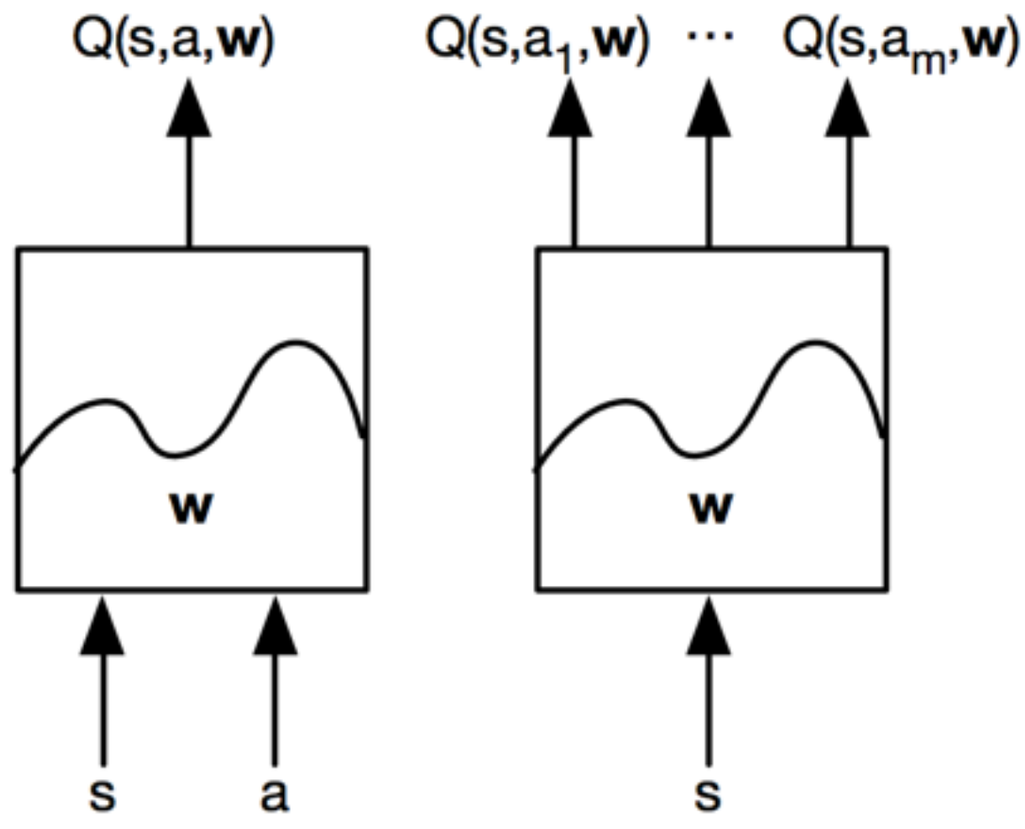
- Can use DL to represent
  - Value function
  - Policy
  - Model
- Optimise loss function via Stochastic Gradient Descent

Value-Based Deep Reinforcement Learning

Value-Based Deep RL

- The value function is represented by Q-network with weights **w**

$$Q(s, a, \mathbf{w}) \approx Q^*(s, a)$$



[David Silver ICML 2016]

Q-Learning

- Optimal Q-values obey Bellman Equation

$$Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q(s',a')^* \mid s,a\right]$$

- Minimise MSE loss via SGD

$$I = \left(r + \gamma \max_{a} Q(s',a',\mathbf{w}) - Q(s,a,\mathbf{w})\right)^2$$

- Diverges due to
  - Correlations between samples
  - Non-stationary targets

Experience Replay

- Remove correlations
- Build dataset from agent's own experience
- Sample experiences from the dataset and apply update

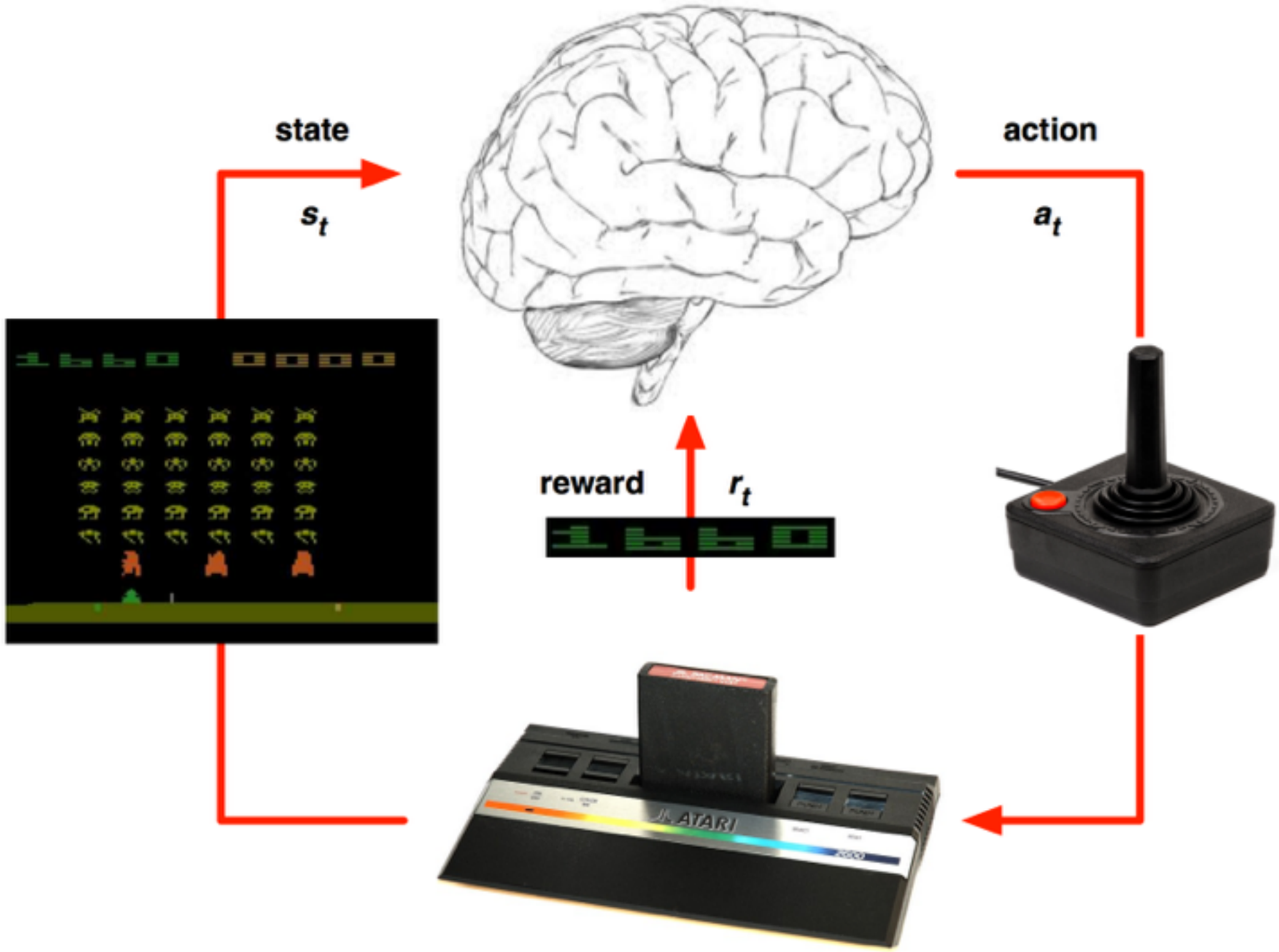
$s_t, a_t, r_{t+1}, s_{t+1} \rightarrow$

s_1, a_1, r_2, s_2
s_2, a_2, r_3, s_3
s_3, a_3, r_4, s_4
...
$s_t, a_t, r_{t+1}, s_{t+1}$

$\rightarrow \quad s, a, r, s'$

[David Silver ICML 2016]

DQN: Atari 2600



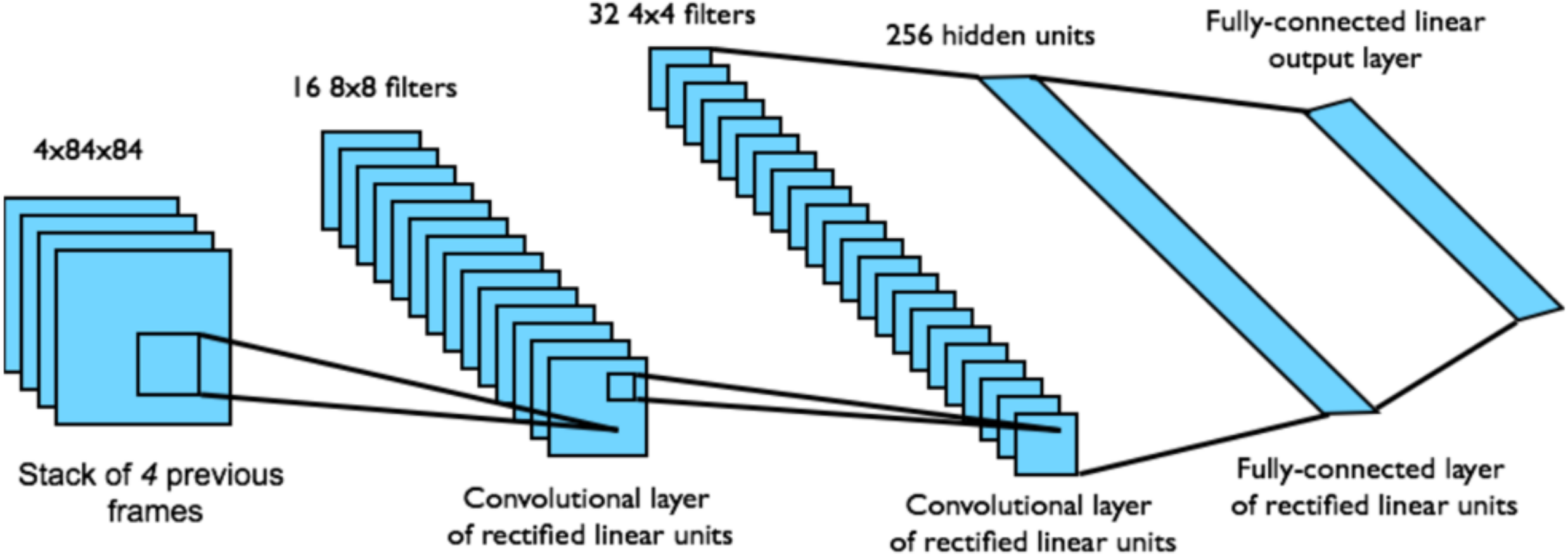
[David Silver ICML 2016]

DQN: Atari 2600

- End-to-end learning of Q(s,a) from the frames
- Input state is stack of pixels from last 4 frames
- Output is Q(s,a) for the joystick/button positions
  - Varies with different games (~3-18)
- Reward is change in score for that step

DQN: Atari 2600

- Network architecture and hyper parameters are fixed across all the games



[David Silver ICML 2016]

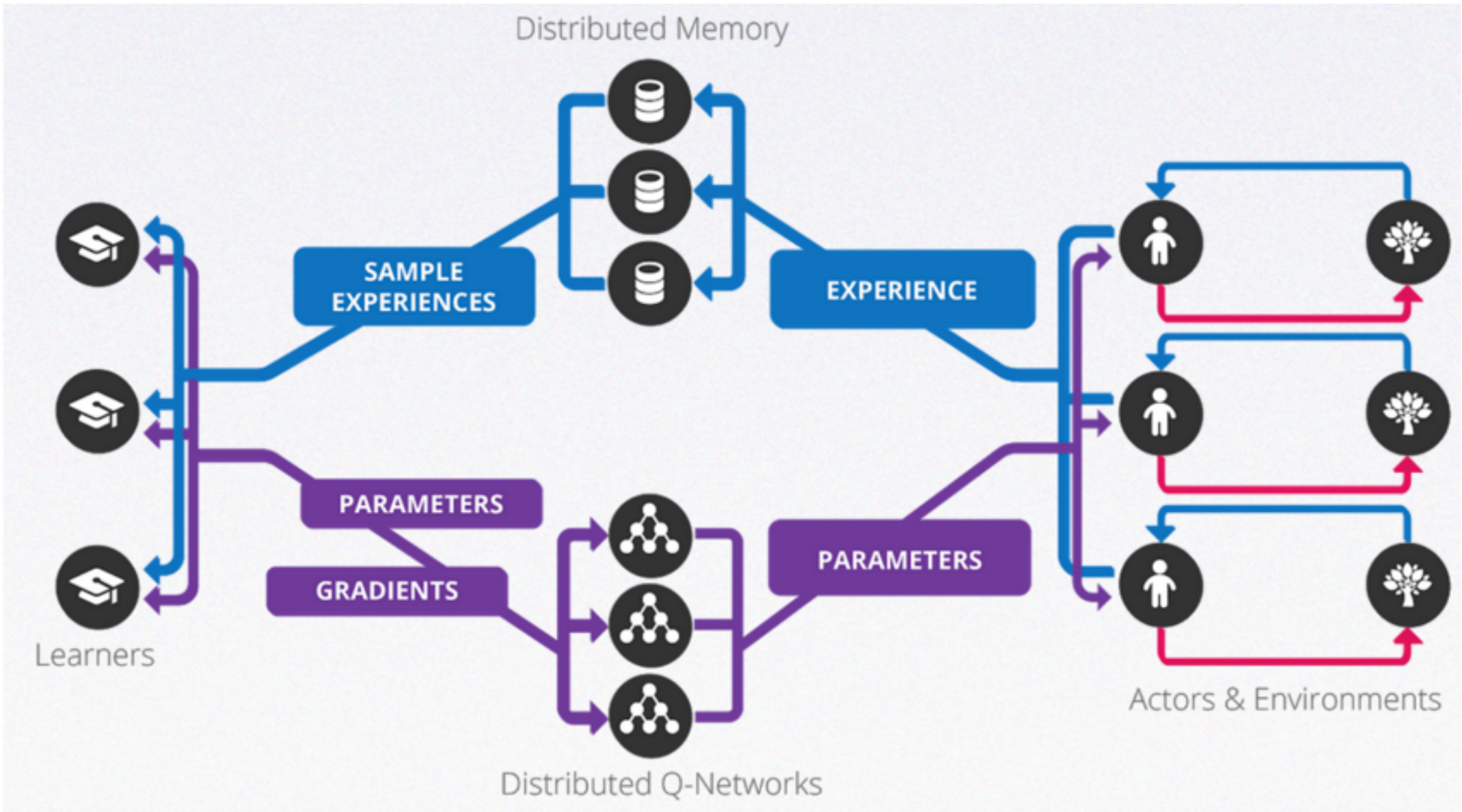
Improvements after DQN

- Double DQN
  - Current Q-network w is used to select actions
  - Older Q-network is w- is used to evaluate actions
- Prioritized reply
  - Store experience in priority queue by the DQN error

Improvements after DQN

- Duelling Network
  - Action-independent value function V(s,v)
  - Action-dependent advantage function A(s,a,**w**)
  - Q(s,a) = V(s,v) + A(s,a,**w**)

General Reinforcement Learning Architecture



[David Silver ICML 2016]

Policy-based Deep Reinforcement Learning

Deep Policy Network

- The policy is represented by a network with weights **u**

$$a = \pi(a|s, \mathbf{u}) \text{ or } a = \pi(s, \mathbf{u})$$

- Define objective function as total discounted reward

$$L(\mathbf{u}) = \mathbb{E}\left[r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots \mid \pi(\cdot, \mathbf{u})\right]$$

- Optimise objectiva via SGD

- Adjust policy parameters **u** to achieve higher reward

[David Silver ICML 2016]

Policy Gradients

- Gradient of a stochastic policy

$$\frac{\partial L(\mathbf{u})}{\partial u} = \mathbb{E}\left[\frac{\partial \log \pi(a|s,\mathbf{u})}{\partial \mathbf{u}} Q^{\pi}(s,a)\right]$$

- Gradient of a deterministic policy

$$\frac{\partial L(\mathbf{u})}{\partial \mathbf{u}} = \mathbb{E}\left[\frac{\partial Q^{\pi}(s,a)}{\partial a}\frac{\partial a}{\partial \mathbf{u}}\right]$$

[David Silver ICML 2016]

Actor Critic Algorithm

- Estimate value function Q(s, a, **w**) $\approx Q^{\pi}(s, a)$
- Update policy parameters **u** via SGD

$$\frac{\partial l}{\partial \mathbf{u}} = \frac{\partial \log \pi(a|s, \mathbf{u})}{\partial \mathbf{u}} Q(s, a, \mathbf{w})$$

$$\frac{\partial l}{\partial \mathbf{u}} = \frac{\partial Q(s, a, \mathbf{w})}{\partial a} \frac{\partial a}{\partial \mathbf{u}}$$

[David Silver ICML 2016]

A3C: Labyrinth

- End-to-end learning of softmax policy from raw pixels
- Observations are pixels of the current frame
- State is an LSTM $f(o_1, \ldots, o_t)$
- Outputs both value V(s) & softmax over actions $\pi(a|s)$
- Task is to collect apples (+1) and escape (+10)

[David Silver ICML 2016]

Model-based Deep Reinforcement Learning

Learning Models of the Environment

- Generative model of Atari 2600

- Issues
  - Errors in transition model compound over the trajectory
  - Planning trajectory differ from executed trajectories
  - Long, unusual trajectory rewards are totally wrong

[David Silver ICML 2016]