Reinforcement Learning for Active Learning

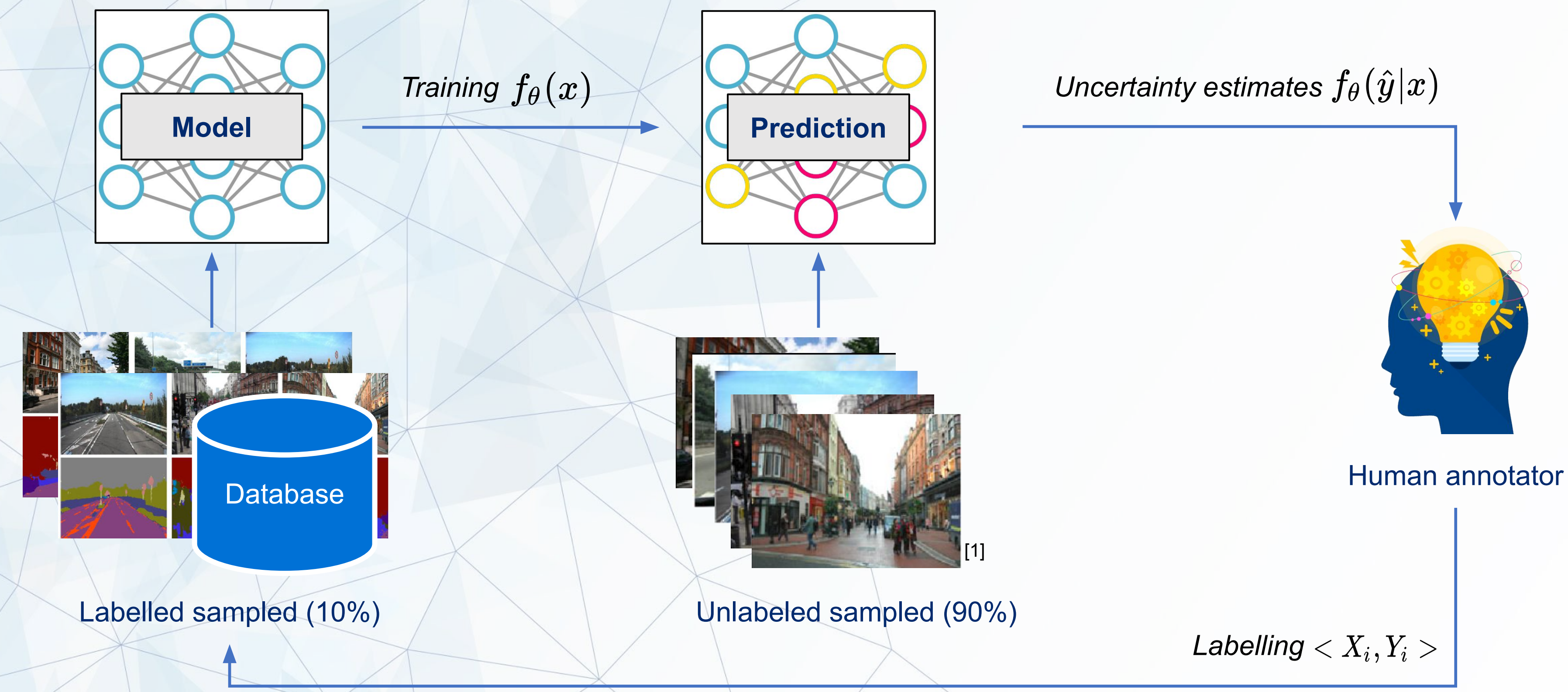
Interdisciplinary Project

Ming Yip, Cheung

Content

- What is *Active Learning* ?
- What is *Reinforcement Learning* ?
- Related Work
- Formulate *Active Learning* as a *Reinforcement Learning* problem
- Full Algorithm
- Challenges
- Result
- Conclusion

Active Learning

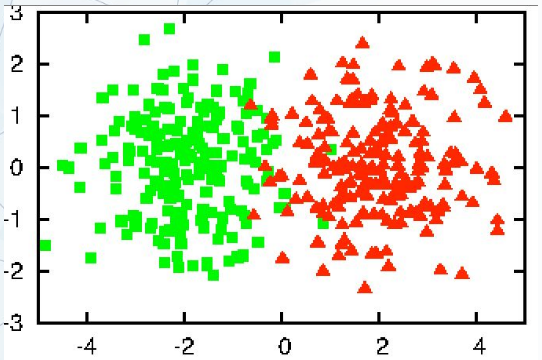


[1] Badrinarayanan, V., Kendall, A., & Cipolla, R. (2015). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39, 2481-2495.

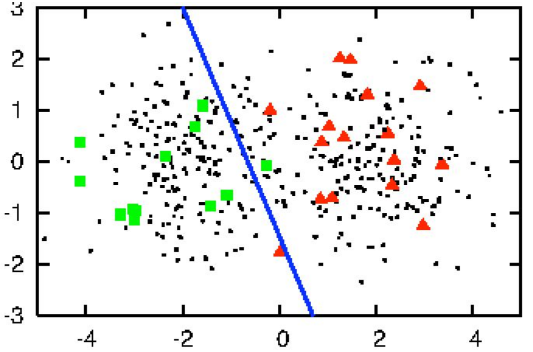
Active Learning

Advantages:

- Requires less training data
- Label efficient learning strategy
- Handle skewed dataset better [3]
- Faster convergence

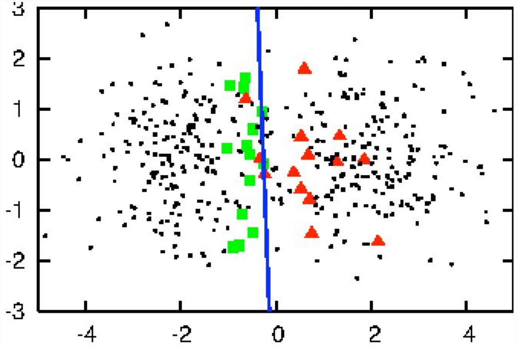


400 instances data from 2 classes



Random sampling

30 data (70% acc)



[2]

Uncertainty sampling

30 data (90% acc)

[2] Settles, B. (2009). Active Learning Literature Survey.
[3] Slotkin, M. (2019, September 30). Accelerate Machine Learning with Active Learning. Retrieved from https://becominghuman.ai/accelerate-machine-learning-with-active-learning-96cea4b72fdb.

Reinforcement Learning [4]

Modeled as a Markov decision process [5]

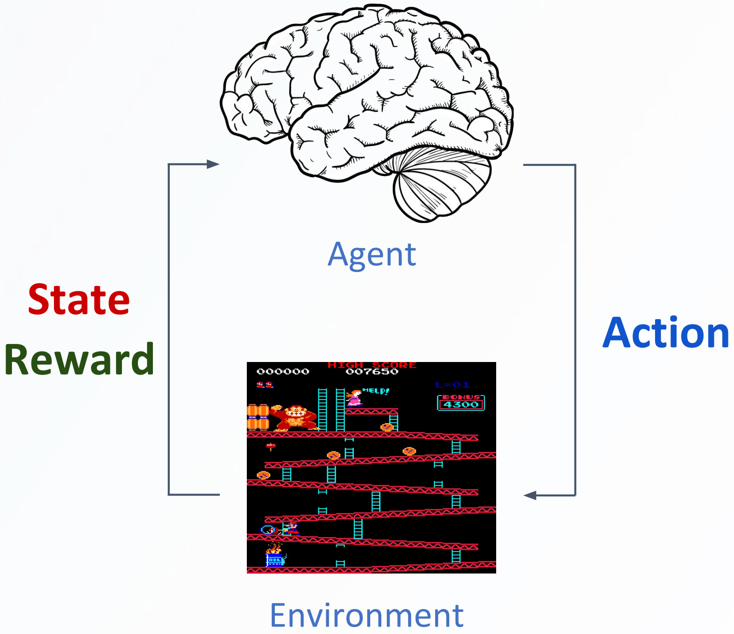
- A set of states S
- A set of actions A
- Reward function $R(s, a, s')$
- Probabilistic transition function $P_r(s_{t+1} = s' | s_t = s, a_t = a)$

Goal: maximize the total expected future reward

$$\pi^* = arg \max_{\pi} \mathbb{E}[G|\pi]$$

where $G = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$

In a **Value-based method,** we learn and predict the $Q(s,a)$ value.
In a **Policy gradient method** [6], we learn policy $\pi(s,a)$.



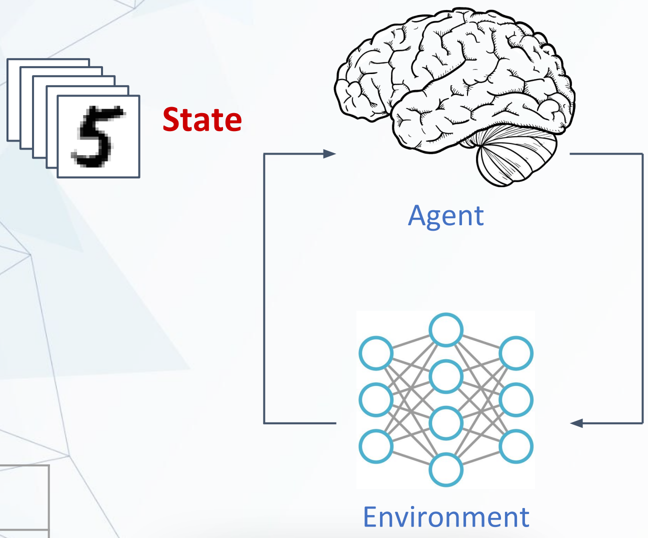
[4] Sutton, R.S., & Barto, A.G. (1988). Reinforcement Learning: An Introduction. IEEE Transactions on Neural Networks, 16, 285-286.
[5] Richard Bellman. A markovian decision process.Indiana Univ. Math. J., 6:679–684,1957
[6] Sutton, R.S., McAllester, D.A., Singh, S.P., & Mansour, Y. (1999). Policy Gradient Methods for Reinforcement Learning with Function Approximation. NIPS.

Reinforcement Learning for Active Learning

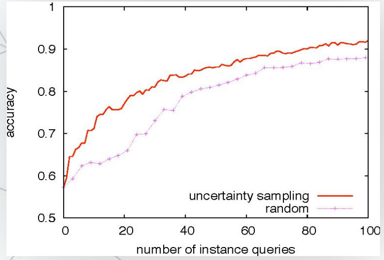
Objective

- Combine RL with AL for classification
- Stream-based active learning
- Neural network as environment

State s_t	images, network weights
Action a_t	select or discard images
Reward R	accuracy



Action



Reward

$56\% \rightarrow 67\%$

Related Work

Related Work

Active Learning Methods:

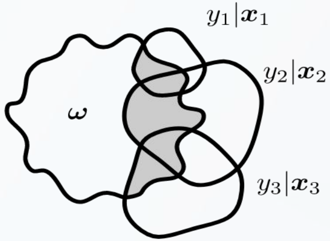
- Uncertainty sampling [7] [8]
- Query-By-Committee [9]
- Variance reduction [10]

Deep Active Learning:

- Studies of AL approaches for neural network
- Uncertainty-based method [11]
- Bayesian uncertainty measure [12]

Reinforcement Learning for:

- Transfer learning [13]
- Named entity recognition (NER) [14]
- Auto data augmentation [15]



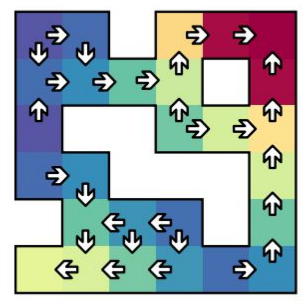
$$a_{\text{BatchBALD}}\left(\{x_1, \ldots, x_b\}, \mathrm{p}\left(\boldsymbol{\omega}|\mathcal{D}_{\text{train}}\right)\right) := \mathbb{I}\left(y_1, \ldots, y_b; \boldsymbol{\omega}|x_1, \ldots, x_b, \mathcal{D}_{\text{train}}\right).$$

[7] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In SIGIR, 1994
[8] David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In ICML, 1994
[9] H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee.In COLT, 1992
[10] Nicholas D Roy and D. Archibald Mccallum. Toward optimal active learning through monte carlo estimation of error reduction. In ICML 2001, 2001.
[11] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. IEEE Transactions on Circuits and Systems for Video Technology, 2016.
[12] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. arXiv preprint arXiv:1703.02910, 2017.
[13] Meng Fang, Yuan Li, and Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. In EMNLP, 2017
[14] Yanyao Shen, Hyokun Yun, Zachary Chase Lipton, Yakov Kronrod, and Anima Anandkumar. Deep active learning for named entity recognition. In ICLR, 2017.
[15] Ekin Dogus Cubuk, Barret Zoph, Dandelion Man´e, V. Vasudevan, and Quoc V. Le.Autoaugment: Learning augmentation strategies from data. In CVPR, 2019

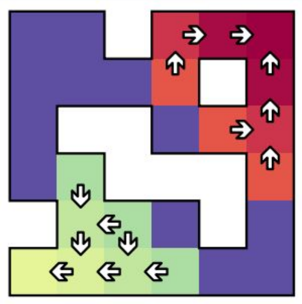
Related Work

Time Limits in Reinforcement Learning [16]

- Time limits are part of the state of the environment
- Include the remaining time as part of the state
- significant performance improvements for agents with time-awareness



(a) Standard

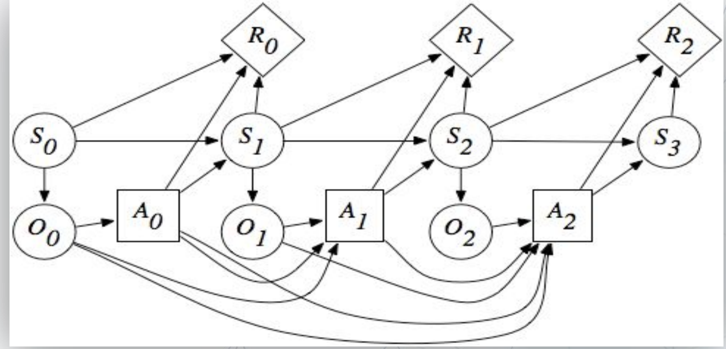


(b) Time-awareness

Partially observable Markov decision process: [17]

- Partially Observable Environments
- Deal with the uncertainty of states
- Beliefs of environment states:

$$S_t^a = (\mathbb{P}[S_t^e = s^1], \ldots, \mathbb{P}[S_t^e = s^n])$$



[16] Pardo, F., Tavakoli, A., Levdik, V., & Kormushev, P. (2017). Time Limits in Reinforcement Learning. ICML.
[17] Åström, K.J. (1965). "Optimal control of Markov processes with incomplete state information".Journal of Mathematical Analysis and Applications. 10: 174–205

Algorithm

1 Training iteration - step by step

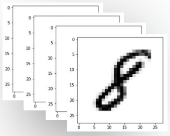
Classification Network f_θ

RL Agent F_ϕ

1 Training iteration - step by step

Classification Network f_θ

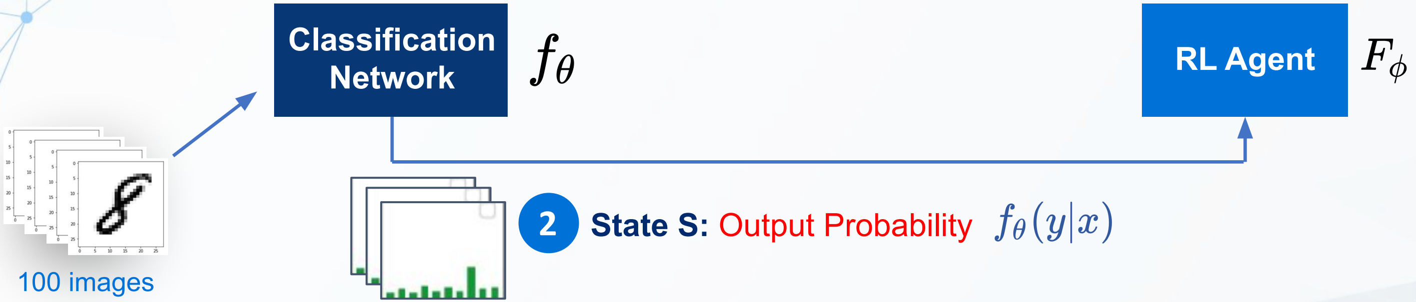
RL Agent F_ϕ



100 images

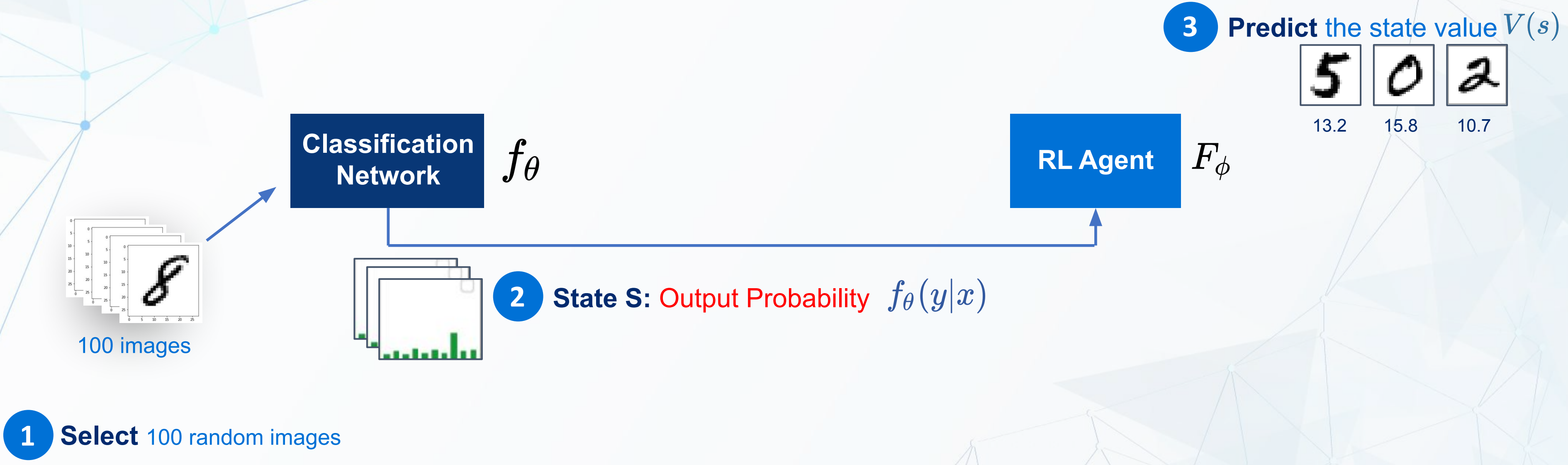
1 **Select** 100 random images

1 Training iteration - step by step

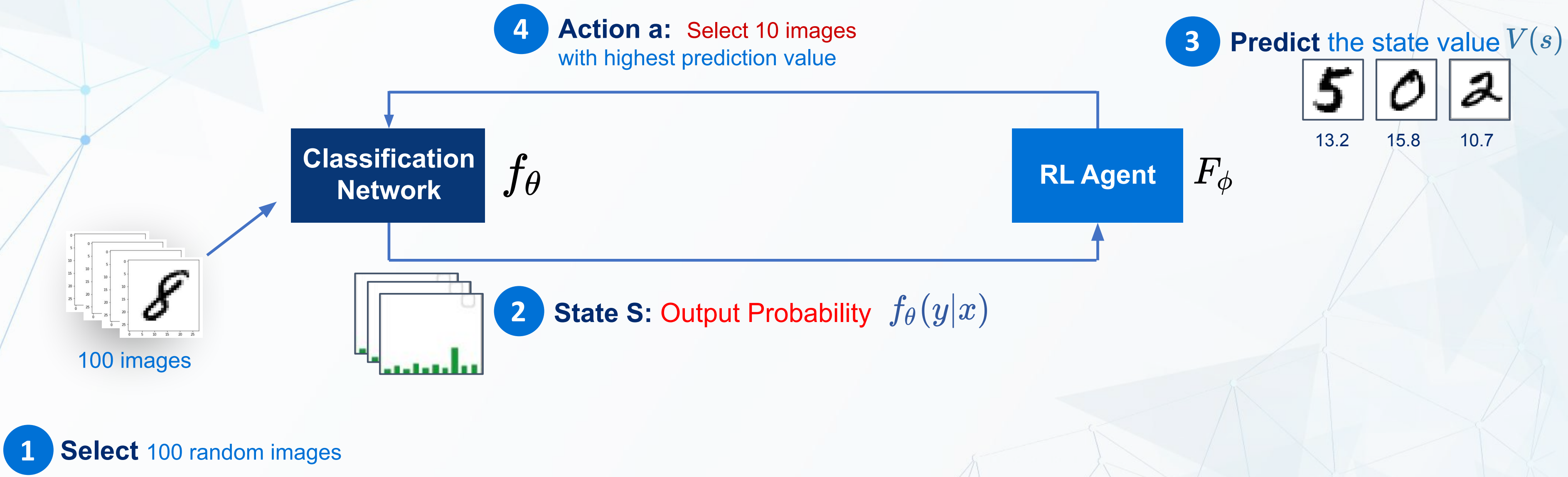


1 **Select** 100 random images

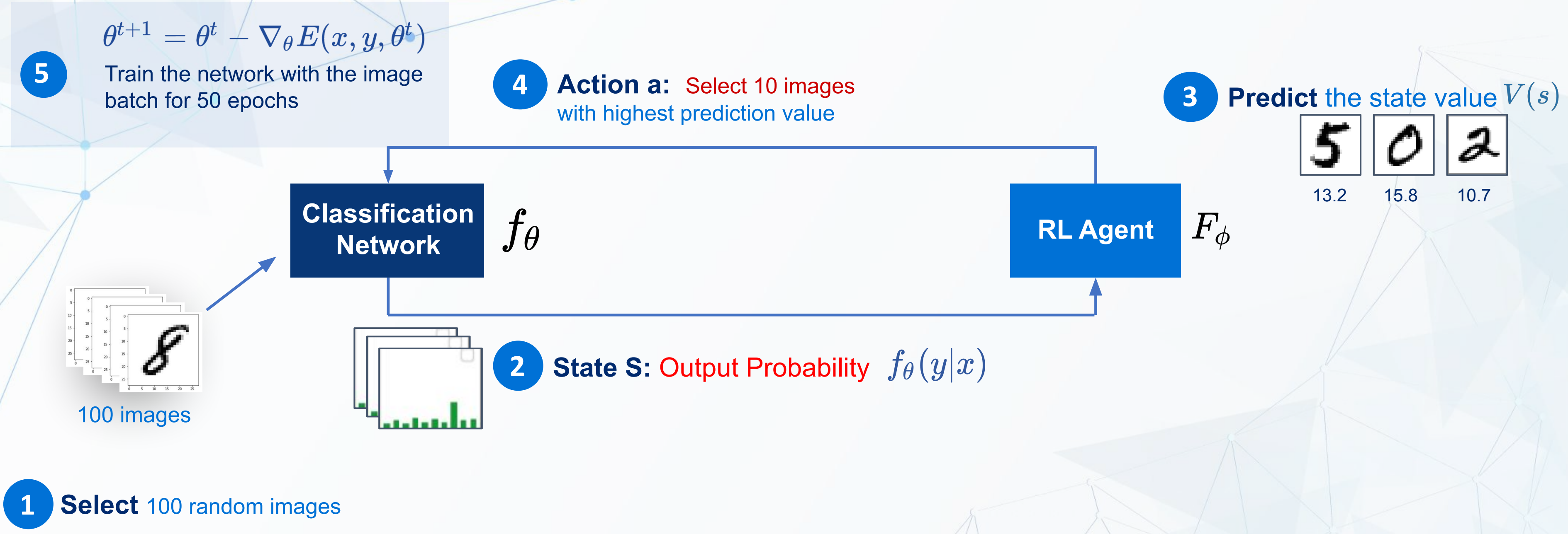
1 Training iteration - step by step



1 Training iteration - step by step



1 Training iteration - step by step



1 Training iteration - step by step

5 $\theta^{t+1} = \theta^t - \nabla_\theta E(x, y, \theta^t)$

Train the network with the image batch for 50 epochs

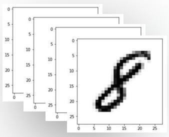
4 **Action a:** Select 10 images with highest prediction value

3 **Predict** the state value $V(s)$

13.2 15.8 10.7

Classification Network f_θ

RL Agent F_ϕ



100 images

2 **State S:** Output Probability $f_\theta(y|x)$

6 **New State S'**

Reward: Accuracy

We evaluate the CNN with Validation set and get the accuracy as **Reward**

1 **Select** 100 random images

1 Training iteration - step by step

5 $\theta^{t+1} = \theta^t - \nabla_\theta E(x, y, \theta^t)$

Train the network with the image batch for 50 epochs

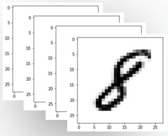
4 **Action a:** Select 10 images with highest prediction value

3 **Predict** the state value $V(s)$

13.2 15.8 10.7

Classification Network f_θ

RL Agent F_ϕ



100 images

2 **State S:** Output Probability $f_\theta(y|x)$

6 **New State S'**

Reward: Accuracy

We evaluate the CNN with Validation set and get the accuracy as **Reward**

7 Train RL agent with

$$V(S) \leftarrow V(S) + \alpha(\underbrace{r + \gamma V(S')}_{\text{TD - Target}} - V(S))$$

1 **Select** 100 random images

Algorithm

Algorithm 5: Reinforcement algorithm for training Value function ϕ

Input : data D, budget B, RL selection batch size b, RL network weights ϕ, classification network weights θ

Output: RL network weights ϕ

Train RL network weight ϕ ;

Fix the samples' order of the data-set D

$steps = \frac{|B|}{|b|}$

for *episode in 1 ... episodes* **do**

  Reset the classification weight θ with same initial random weights

  for t = *0, 1, ..., steps* **do**

    Predicted Future Expected Return $V_i(x_t) \leftarrow F_\phi(f_\theta(y|x), T - t, B - i * b)$

    Select top b images $D_t = Top_b\{(x_t, y_t)\}_{i=1}^{b}$

    for *epoch in 1 ... epochs* **do**

      Train Classification Network $\theta \leftarrow \theta - \nabla_\theta L(y_i, f\theta(x_i))$

    end

    Get Validation set Accuracy $R_v = A_i$

    Train RL agent $\phi \leftarrow \phi + \alpha \nabla_\phi F_\phi(S_t, A_t, R_t, S_{t+1}, \gamma)$

  end

  Get Final Validation set Accuracy $R_t = A_i$

  if R_t *is the best Accuracy* **then**

    Store the best Agent ϕ

  end

end

return best RL Agent ϕ

Train RL Agent

Train CNN

Algorithm

Algorithm 5: Reinforcement algorithm for training Value function ϕ

Input : data D, budget B, RL selection batch size b, RL network weights ϕ, classification network weights θ
Output: RL network weights ϕ
Train RL network weight ϕ ;

Fix the samples' order of the data-set D
$steps = \frac{|B|}{|b|}$

Train RL Agent

for *episode in 1 ... episodes* **do**
  Reset the classification weight θ with same initial random weights
  for $t = 0, 1, ..., steps$ **do**
    Predicted Future Expected Return $V_i(x_t) \leftarrow F_\phi(f_\theta(y|x)(T-t, B-i*b)$
    Select top b images $D_t = Top_b\{(x_t, y_t)\}_{i=1}^{b}$
    for *epoch in 1 ... epochs* **do**
      Train Classification Network $\theta \leftarrow \theta - \nabla_\theta L(y_i, f\theta(x_i))$
    end
    Get Validation set Accuracy $R_v = A_i$
    Train RL agent $\phi \leftarrow \phi + \alpha \nabla_\phi F_\phi(S_t, A_t, R_t, S_{t+1}, \gamma)$
  end
  Get Final Validation set Accuracy $R_t = A_i$
  if R_t *is the best Accuracy* **then**
    Store the best Agent ϕ
  end
end
return best RL Agent ϕ

Train CNN

Time awareness [16]

- Remaining episodes (RL)
- Remaining budgets (CNN)

$$V_\pi(s, T-t, B-ss) = \mathbb{E}_\pi[G_{t:T}|S_t = s]$$

[16] Pardo, F., Tavakoli, A., Levdik, V., & Kormushev, P. (2017). Time Limits in Reinforcement Learning. ICML.

Challenges

Split a complete state into a batch

- Evaluate the importanceness of each image
- Incomplete information of the state
- The RL agent observes only part of the state

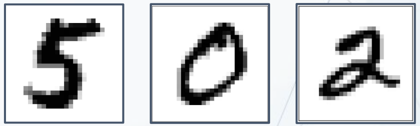
We select the top k images from a batch with M images

- If (M, k) are too large then the RL agent can hardly learn anything.
- If (M, k) are too small then the CNN network learns too slow and with insignificant reward

Markov Decision Process assumption

- $P(S_{t+1} = s_{t+1} | s_t, s_{t-1}, s_{t-2}, \ldots, s_1) = P(S_{t+1} = s_{t+1} | s_t)$
- Naive approach is a stateless environment
- Randomly select images from the date-set
- Each state does not necessary depend on the previous state
- Need to include the information of the CNN network weight
- The output probability may keep the MDP assumption

Predict the state value $V(s)$



13.2 15.8 10.7

Challenges

Split a complete state into a batch

- Evaluate the importanceness of each image
- Incomplete information of the state
- The RL agent observes only part of the state

We select the top k images from a batch with M images

- If (M, k) are too large then the RL agent can hardly learn anything.
- If (M, k) are too small then the CNN network learns too slow and with insignificant reward

Markov Decision Process assumption

- $P(S_{t+1} = s_{t+1} | s_t, s_{t-1}, s_{t-2}, \ldots, s_1) = P(S_{t+1} = s_{t+1} | s_t)$
- Naive approach is a stateless environment
- Randomly select images from the date-set
- Each state does not necessary depend on the previous state
- Need to include the information of the CNN network weight
- The output probability may keep the MDP assumption

Challenges

Split a complete state into a batch

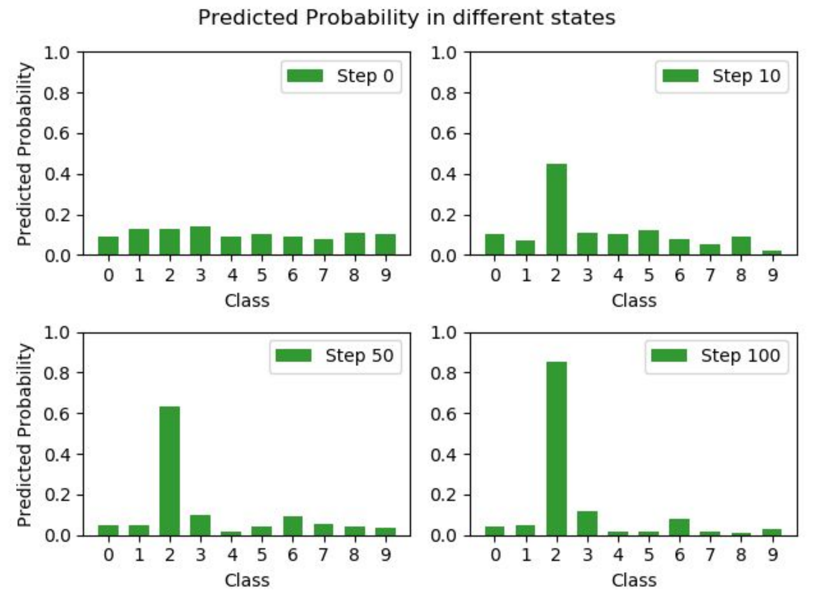
- Evaluate the importanceness of each image
- Incomplete information of the state
- The RL agent observes only part of the state

We select the top k images from a batch with M images

- If (M, k) are too large then the RL agent can hardly learn anything.
- If (M, k) are too small then the CNN network learns too slow and with insignificant reward

Markov Decision Process assumption

- $P(S_{t+1} = s_{t+1} | s_t, s_{t-1}, s_{t-2}, \ldots, s_1) = P(S_{t+1} = s_{t+1} | s_t)$
- Naive approach is a stateless environment
- Randomly select images from the date-set
- Each state does not necessary depend on the previous state
- Need to include the information of the CNN network weight
- The output probability may keep the MDP assumption

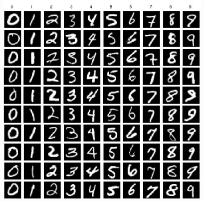


Evaluation

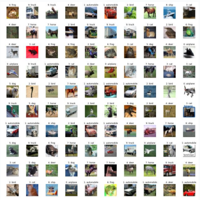
Data sets

Data sets	Num. of Images	Num. of Classes	Num. of Training images	Num. of Validation images	Num. of Test images	Num. of images per class	Dimension
MNIST [18]	70,000	10	50,000	10,000	10,000	~7,000	28x28
CIFAR-10 [19]	60,000	10	48,000	6,000	6,000	~6,000	32x32x3
EMNIST [20]	78,000	26	62,400	7,800	7,800	3,000	28x28

Table1: 3 labeled datasets



MNIST



CIFAR-10

EMNIST

Random Agent

vs

Least Margin Sampling Method (BVSB):

$$x^* = arg\,min P_\phi(\hat{y_1}|x) - P_\phi(\hat{y_2}|x)$$

[18] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database.ATTLabs [Online]. Available: http://yann. lecun. com/exdb/mnist, 2, 2010.
[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.
[20] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Ex-tending mnist to handwritten letters.2017 International Joint Conference on Neural Networks (IJCNN), 2017.

MNIST

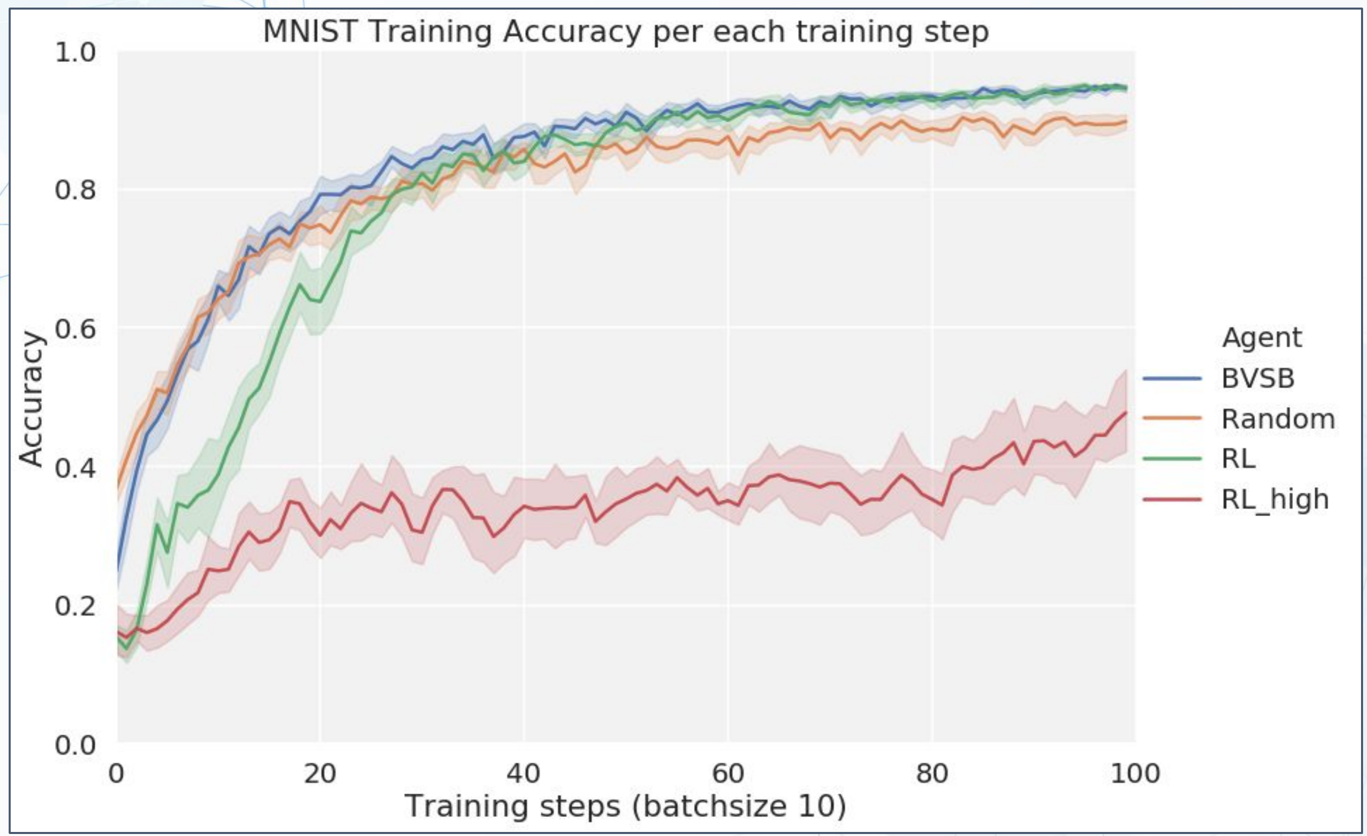


Table 4.2: Average terminal accuracy of 20 experiments on MNIST

Dataset	Random	BVSB	RL-top	RL-low
MNIST	0.8974 ± 0.0240	0.9447 ± 0.0128	0.4924 ± 0.0763	*0.9487 ± 0.00975

EMNIST

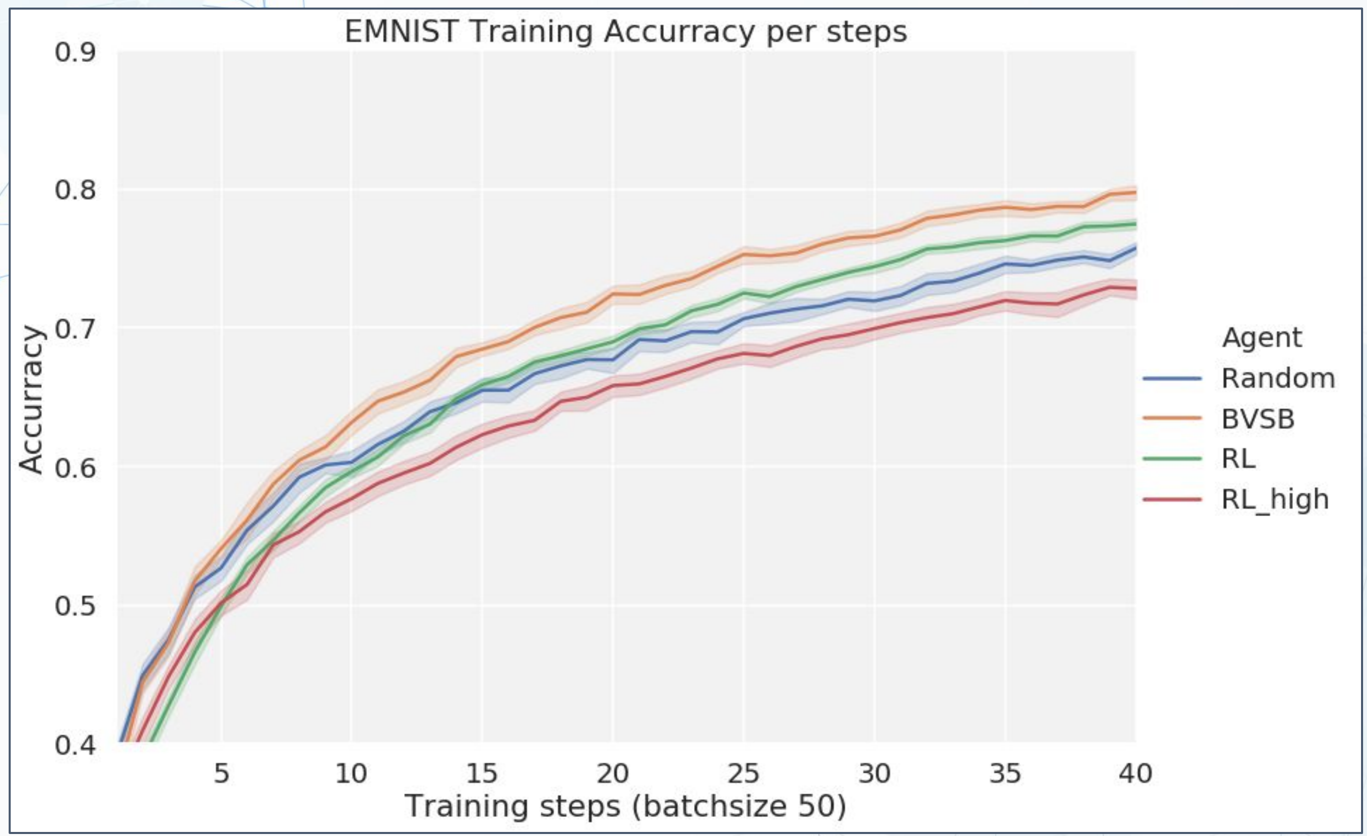


Table 4.2: Average terminal accuracy of 20 experiments on EMNIST

Dataset	Random	BVSB	RL-top	RL-low
EMNIST	0.7574 ± 0.0104	*0.7975 ± 0.0123	0.7282 ± 0.0232	0.7748 ± 0.0150

CIFAR

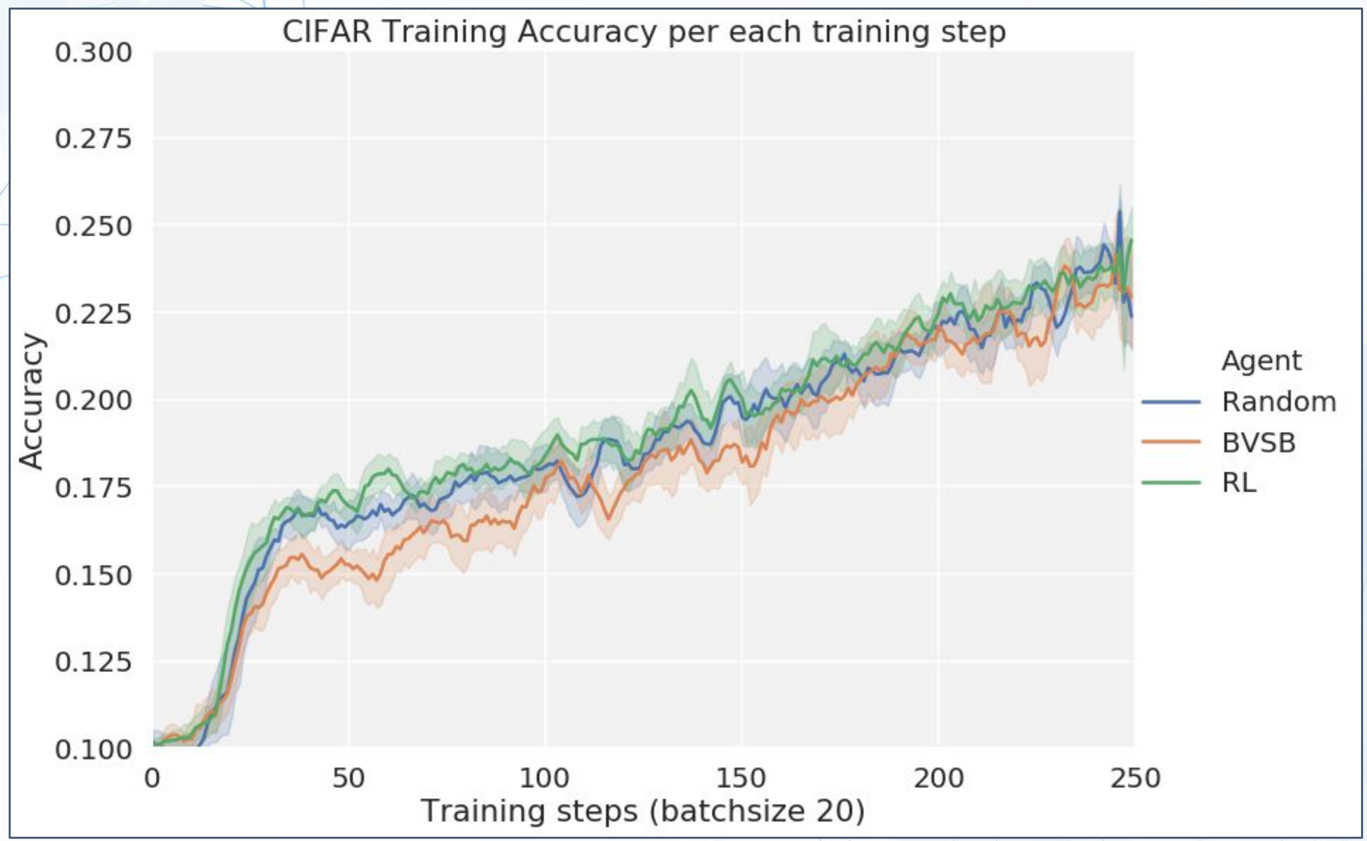
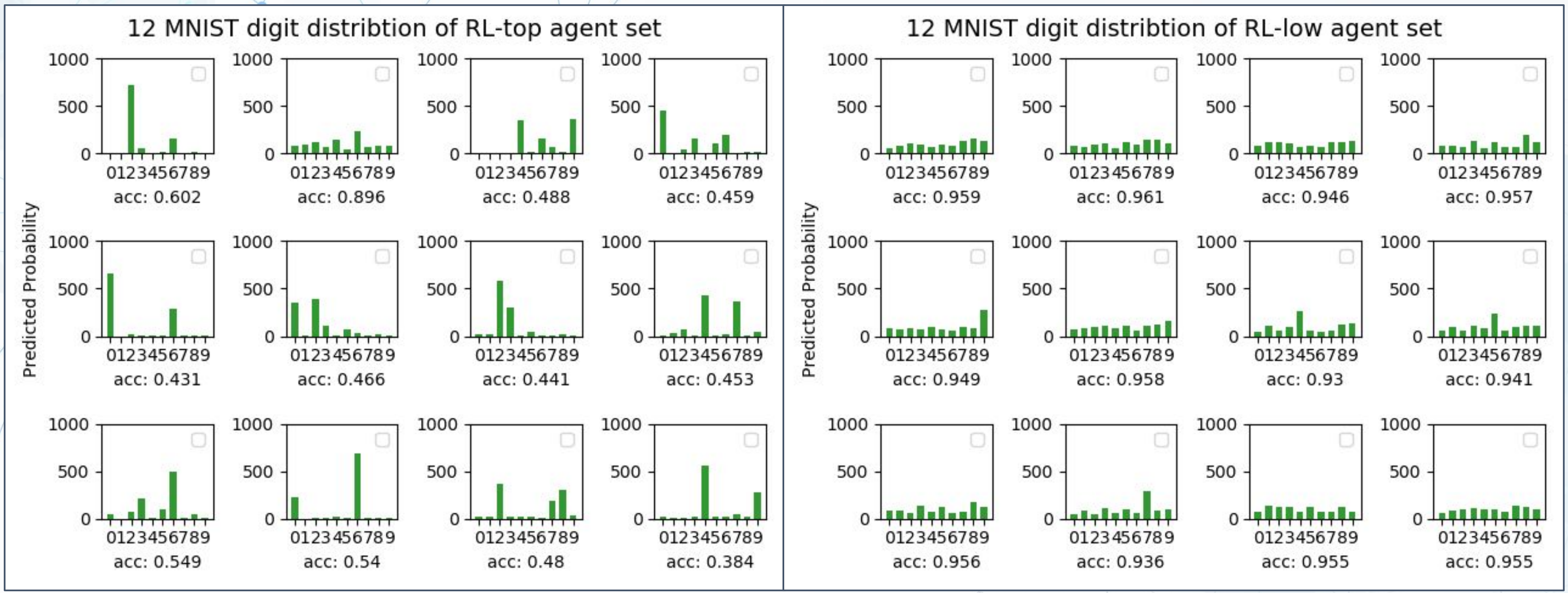


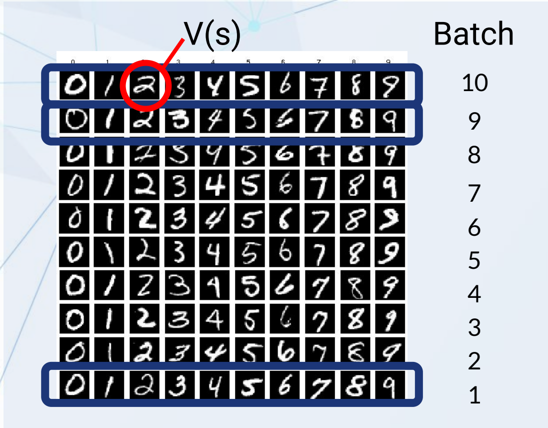
Table 4.2: Average terminal accuracy of 20 experiments on CIFAR

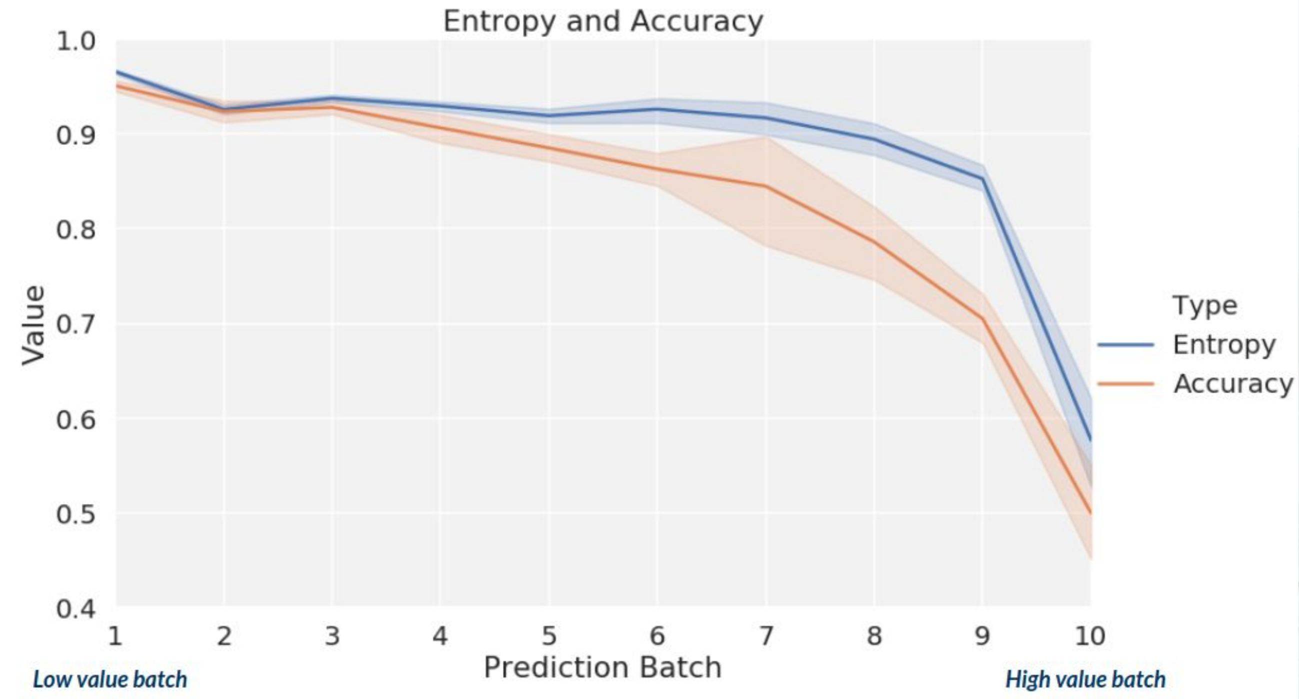
Dataset	Random	BVSB	RL-low
CIFAR[4]	0.2238 ± 0.0173	0.2294 ± 0.0253	*0.245714 ± 0.0161

Selection bias

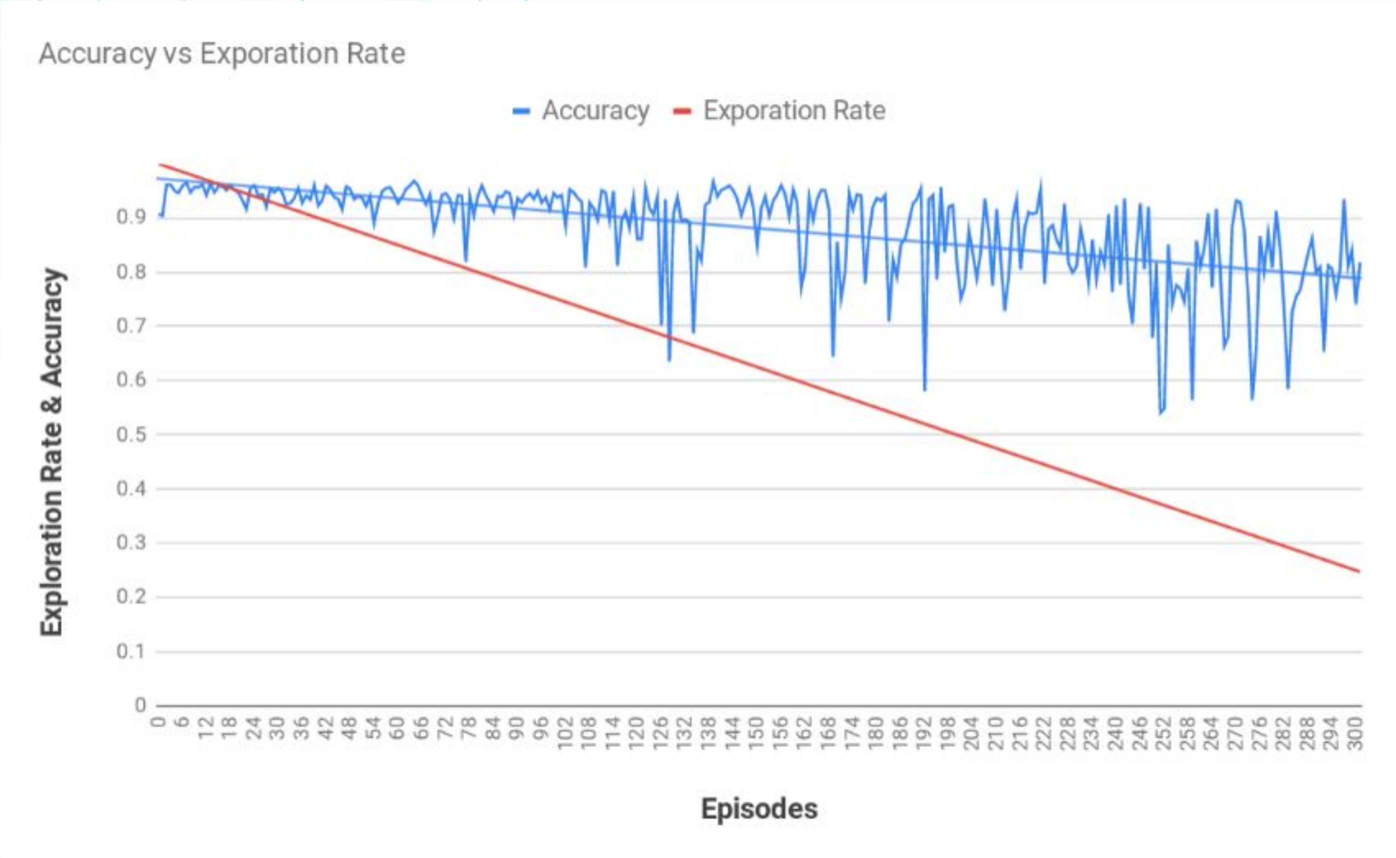


Entropy of the selection distribution





Effect of using exploration rate



Conclusion

- Competitive good result on MNIST and EMNIST
- CIFAR-10
- RL-top agent suffers from the bias selection
- Exploitation hurts the performance

Thank you

Reference

1. Badrinarayanan, V., Kendall, A., & Cipolla, R. (2015). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39, 2481-2495.
2. Settles, B. (2009). Active Learning Literature Survey
3. Slotkin, M. (2019, September 30). Accelerate Machine Learning with Active Learning. Retrieved from https://becominghuman.ai/accelerate-machine-learning-with-active-learning-96cea4b72fdb.
4. Sutton, R.S., & Barto, A.G. (1988). Reinforcement Learning: An Introduction. IEEE Transactions on Neural Networks, 16, 285-286.
5. Richard Bellman. A markovian decision process.Indiana Univ. Math. J., 6:679–684,1957
6. Sutton, R.S., McAllester, D.A., Singh, S.P., & Mansour, Y. (1999). Policy Gradient Methods for Reinforcement Learning with Function Approximation. NIPS.
7. David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. InSIGIR, 1994
8. David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In ICML, 1994
9. H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee.In COLT, 1992
10. Nicholas D Roy and D. Archibald Mccallum. Toward optimal active learning through monte carlo estimation of error reduction. In ICML 2001, 2001.
11. Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. IEEE Transactions on Circuits and Systems for Video Technology, 2016.
12. Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. arXiv preprint arXiv:1703.02910, 2017.
13. Meng Fang, Yuan Li, and Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. In EMNLP, 2017
14. Yanyao Shen, Hyokun Yun, Zachary Chase Lipton, Yakov Kronrod, and Anima Anandkumar. Deep active learning for named entity recognition. In ICLR, 2017.
15. Ekin Dogus Cubuk, Barret Zoph, Dandelion Man´e, V. Vasudevan, and Quoc V. Le.Autoaugment: Learning augmentation strategies from data. In CVPR, 2019

Reference

16. Pardo, F., Tavakoli, A., Levdik, V., & Kormushev, P. (2017). Time Limits in Reinforcement Learning. ICML.
17. Åström, K.J. (1965). "Optimal control of Markov processes with incomplete state information".Journal of Mathematical Analysis and Applications. 10: 174–205
18. Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database.ATTLabs [Online]. Available: http://yann.lecun.com/exdb/mnist, 2, 2010.
19. Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.
20. Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Ex-tending mnist to handwritten letters.2017 International Joint Conference on Neural Networks (IJCNN), 2017.