Reinforcement Learning for Parameter Control of Text Detection in Images from Video Sequences

Graham W. Taylor
University of Waterloo
Pattern Analysis and Machine Intelligence Laboratory
200 University Avenue W., N2L3G1, Waterloo, Ontario, Canada
gwtaylor@engmail.uwaterloo.ca

Christian Wolf
INSA de Lyon
Laboratoire d'Informatique en Images et Systèmes d'information
Bât J. Verne, 20 rue Albert Einstein, 69621, Villeurbanne cedex, France
christian.wolf@liris.cnrs.fr

Abstract

A framework for parameterization in computer vision algorithms is evaluated by optimizing ten parameters of the text detection for semantic indexing algorithm preposed by Wolf et al. The Fuzzy ARTMAP neural network is used for generalization, offering much faster learning than in a previous tabular implementation. Difficulties in using a continuous action space are overcome by employing the DIRECT method for global optimization without derivatives. The chosen parameters are evaluated using metrics of recall and precision, and are shown to be superior to the parameters previously recommended.

1. Introduction

Reinforcement learning (RL), [12, 5, 8] can be described as a computational approach to learning through interacting with the environment. Historically, the literature has not offered many examples where image-based problems have been solved using this family of learning algorithms, but recently, research has shown that RL can indeed be applied to such problems [10, 17]. In this paper, we offer a generalized framework for approaching the problem of optimizing parameters for a multi-step computer vision algorithm using an RL agent. We then focus on the specific task of text detection in images taken from video sequences for semantic indexing.

1.1. Reinforcement learning

The basis of RL is an intelligent agent that seeks some reward or special signal, and strives to receive this reward through exploring its environment. It exploits the knowledge it has previously obtained through past actions and past rewards (or punishments). RL fundamentally differs from supervised learning, and thus can offer an advantage over the latter in many tasks. The agent learns on-line, and can continually learn and adapt while performing the required task. This behaviour is useful for the many cases where precise learning data is difficult or impossible to obtain [12].

Figure 1 illustrates the several components forming RL. The agent, which is the decision maker of the process, attempts an action that is recognized by the environment. It receives from its environment a reward or punishment depending on the action taken. The agent also receives information concerning the state of the environment. The agent acquires knowledge of the actions that generate rewards and punishments and eventually learns to perform the actions that are the most rewarding in order to meet a certain goal relating to the state of the environment.

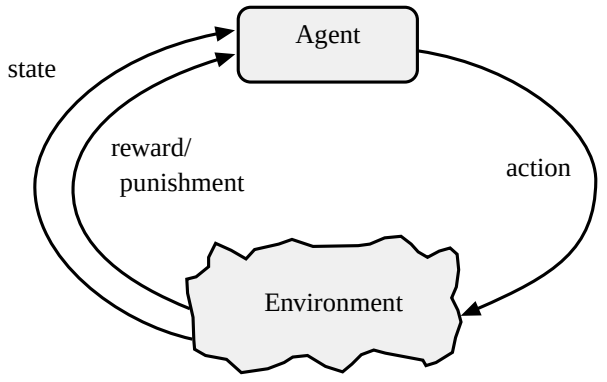

Figure 1. The components of a reinforcement learning agent

1.2. Generalization

A significant barrier to solving image-based problems with RL is the massive amount of data involved. This complicates the task of deriving state information for learning. It becomes impossible to use tabular methods to store past experience. This approach not only presents unrealistic memory requirements, but for large state spaces, an agent is not able to visit all state-action pairs. Therefore, the time needed to fill these tables becomes increasingly problematic. In the case of a large, continuous state-space, the problem becomes intractable. This is known as the *curse of dimensionality* and requires some form of generalization in addition to careful feature selection.

Recent research has shown that connectionist systems can handle a much higher-dimensional state space [4]. The well-known multi-layer perceptron (MLP) has been combined with RL in many examples [9, 4]. Unfortunately feedforward MLP networks have a fundamental flaw. While they are excellent interpolators, they fail miserably at the extrapolation task. Patrascu [6] has provided strong empirical results that the Fuzzy ARTMAP [3] vastly outperforms the MLP when the two are compared using the Sarsa algorithm. The Fuzzy ARTMAP is also much less sensitive to algorithm parameters. This work has inspired us to use a Fuzzy ARTMAP for generalization, but we have adapted Patrascu's approach to support a continuous action space.

2. A framework for reinforced parameter selection

We have developed a general framework for applying RL to a multi-step image processing or computer vision algorithm. Similar to Yin's approach [17], we express the state in terms of past parameters to be selected. If the agent is expected to adapt the parameters to the image itself (rather than deriving a general set of parameters), then it can incorporate image features in the state. We provide one Fuzzy ARTMAP network for each step of the algorithm to predict state-action values. At each step, the agent selects a multidimensional action which corresponds to the parameters chosen at that step of the algorithm. The agent receives a numerical reward when the algorithm terminates. It also may receive intermediate rewards, derived from transitional images generated during the process. The development of this framework is thouroughly discussed in [13]. Figure 2 presents the model visually.

One major difference from the previous Fuzzy ARTMAP implementation is the use of continuous actions. The neural net replaces the traditional Q-matrix lookup table, but functions in a similar way. Traditional connectionist approaches using discrete actions employ one network for each action. The state is fed as an input to each network and a Q-value is returned corresponding to each action. It is then simple to calculate the action corresponding to the maximum Q-value for any given state. This approach cannot be used for a continuous action space. We have chosen to implement one network per algorithm step, and thus feed the state and currently considered action to the network, which then returns the corresponding Q-value. Unfortunately, to find the maximum Q-value, we must query the network once for every action, which means an infinite amount of queries. Several possible approaches to this problem have been presented in the literature. It is essentially global, bounded optimization without derivatives, where function calls should be minimized, as they are expensive. Smart and Kaelbling [11] employ a method similar to Brent's [2] method. Baird and Klopf [1] have introduced a complex method called "Wire fitting". We have chosen to use the DIRECT optimization method [7], which has not yet been used in combination with RL.

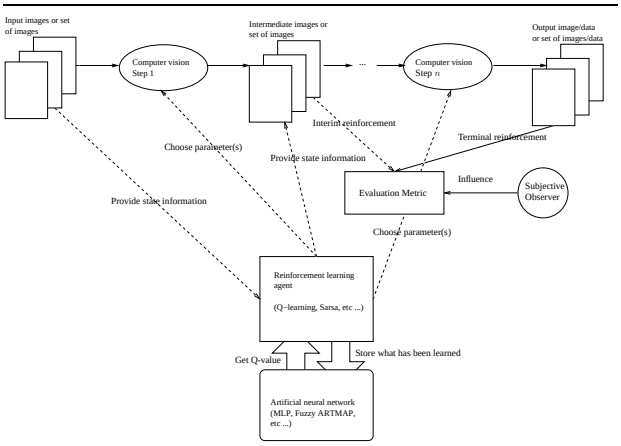

Figure 2. A framework for connectionist-based reinforced learning for selecting the parameters of an n-step computer vision application

3. Application

Wolf et al. have developed an algorithm to detect text in images taken from video sequences [16]. Video is processed on a frame by frame basis, tracking detected rectangles of text across multiple frames. Our learning focuses only on the detection of text in still images. This text detection algorithm is multi-step, involving gray level, morphological and geometrical constraints. More details are provided in Algorithm 1.

We attempt to optimize ten parameters of this algorithm using our framework. The parameters, and their suggested ranges are shown in Table 1.

At each iteration, the agent randomly selects three images containing text, and three images containing no text from an image base. The agent then proceeds, choosing parameters dictated by the RL algorithm and employing them at each step of the text-detection algorithm. Each of

Algorithm 1 Text detection in still images

1. Gray level constraints
   (a) Conversion to gray scale images
   (b) Calculation of horizontal Sobel gradient
   (c) Method of accumulating gradients
   (d) Binarization of accumulated gradients with two-threshold version of Otsu's method
2. Morphological constraints
   (a) Morphological close
   (b) Suppresion of small horizontal bridges between connected components
   (c) Conditional dilation
   (d) Conditional erosion
   (e) Horizontal erosion
   (f) Horizontal dilation
3. Geometrical constraints
   (a) Main geometrical constraints
   (b) Consideration of special cases
   (c) Combination of rectangles

Par.	Range	Description
S	$5-35$	Size of the gradient accumulation filter
α	$50-150$	Determination of the second threshold for the binarization of the accumulated gradients
t_1	$1-5$	Threshold on column height
t_2	$20-200$	Threshold on height difference
t_3	$20-200$	Threshold on position difference
t_4	$1-6$	Threshold on ratio width/height
t_5	$0.2-0.9$	Threshold on ratio pixels/area
t_6	$0.1-0.9$	Threshold for combining rectangles
t_7	$0.1-0.9$	Threshold for combining rectangles
t_8	$0.1-0.9$	Threshold for combining rectangles

Table 1. Parameters for text detection algorithm

the six test images are processed in parallel, using identical parameters. The reward provided at each step is zero, until the final step, when the agent has chosen a set of rectangular regions for each of the set of six test images. Our reasoning for the use of multiple images and averaging of results is to assist the agent in generalizing across many different images. This lowers the sensitivity of the reward to the particular example.

We concern ourselves with the classical method of measuring performance in information systems. That is, the measures of precision and recall:

$$\begin{aligned} \text{Recall}_{IR} &= \frac{\text{number of correctly retrieved items}}{\text{number of relevant items in the database}} \\ \text{Precision}_{IR} &= \frac{\text{number of correctly retrieved items}}{\text{total number of retrieved items}} \end{aligned} \tag{1}$$

In order to have a single performance value for the ranking of results, we have used the harmonic mean of precision and recall:

$$\text{HM}_{IR} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2}$$

Evaluation is handled differently for the images containing text and images containing no text. For the former, we have a set of ground truth images to which we compare the agent's selection of rectangular text regions. Over the set of three images, we calculate the total number of groundtruth rectangles, the total number of detected rectangles, and the total number of correctly detected rectangles. This allows us to generate precision and recall values between 0 and 1 for the set. (The comparison between the set of detected rectangles and the set of ground truth rectangles takes into account one-to-one matches as well as one-to-many and many-to-one matches, the latter two resulting in a slight punishment. See [15] for details of the evaluation algorithm.) The harmonic mean of these two values is taken to give us a single measure of performance on text-containing images.

For non-text, we wish to generate a measure on the same scale, which has maximum value when we detect no text regions, and gradually grows lower as we detect more rectangles. To achieve this result, we have used the following simple formula:

$$r_{nontext} = e^{-\delta N}, \tag{3}$$

where N is the number of detected rectangles δ is an empirically determined factor. We found that $\delta = 0.5$ offered good performance. As the number of detected rectangles grows, the evaluation is reduced to zero. As in the case of the previous evaluation, the maximum value (at $N = 0$) is 1. Now that we have one performance measure for each of images containing text and images containing no text, we simply combine them by taking their mean. If we wanted to train the agent to consider less the non-text images, we could weight this combination differently.

4. Results

The following results are divided into two subsections. The first, concerns the ability of the agent to learn, measured by its received reward over a number of iterations. The second, concerns the evaluation of the parameters chosen by the agent, not only on the training set of images, but on a separate set of images used for testing.

4.1. Learning

Learning was evaluated using a test set of 72 groundtruthed images containing artificial text, and 72 images which contained no text. All images were in format CIF (384×288 pixels). We experimented using three popular RL algorithms: Q-learning, Sarsa and Sarsa(λ).

Q-learning [14], is the most widely used and well-known temporal-difference (TD) control algorithm. It is a form of off-policy TD control. This means that the learned action-value function, Q, directly approximates the optimal action-value function, Q^* independently of the policy being followed. This is seen in the max term in the Q-learning equation (Eq. 4) where we use the maximum Q value over all actions at the next state, regardless of the policy.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (4)$$

A similar TD control algorithm, but instead on-policy, is the Sarsa algorithm [9], introduced by Rummery and Niranjin. The update equation for Sarsa (Eq. 5) is nearly identical to that of Q-learning, but the next state action pair must be selected (using the policy) before the update can be made.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (5)$$

In both Eqs. 4 and 5, r_t is the reward received from the environment at time t, α is the step-size parameter, and γ is the discount-rate parameter. Sarsa(λ), though related to Sarsa, is considerably more complex, involving eligibility traces. For sake of brevity, we refer the reader to [12] for details on its implementation.

Figure 3 shows the performance of each algorithm. Learning convergence was observed within 500 iterations by following the reward history. This is a remarkable improvement over previous results [13], where the number of iterations was nearly double for far fewer parameters. This is clearly due to the generalization capabilities of the Fuzzy ARTMAP. Due to the stochastic behaviour of the agent, these results are averaged over 5 trials, and then smoothed by plotting the mean of the past 100 iterations. We note that Q-learning provides significantly better performance, and therefore it is used in the following experiments. The results are surprising, as the incorporation of eligibility traces by Sarsa(λ) should improve temporal credit assignment. Even if the off-policy method, Q-learning, is more suited to the problem at hand, Sarsa(λ) should outperform Sarsa. A possible reason for this poor performance could be due to the nature of the problem. We are dealing with an episodic task, where the episodes are short and constant-length. The incorporation of eligibility traces could simply be "overkill". The values of the eligibility traces are managed by the Fuzzy ARTMAP network, so that we may generalize their storage, as well as Q-values. The performance could therefore be implementation-related.

Reinforcement learning parameters were determined empirically, and in each case, were optimal (chosen from a discrete set of values). These are shown in Table 2. A trace threshold of 0.01 was also used in the case of Sarsa(λ). This means that traces with values less than this threshold were not stored for efficiency. Figure 4 demonstrates the

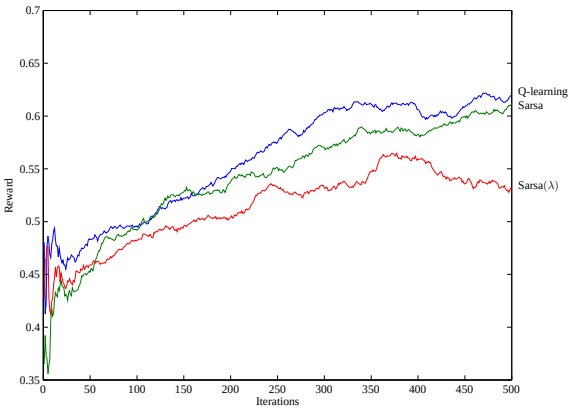

Figure 3. Comparison of RL algorithms

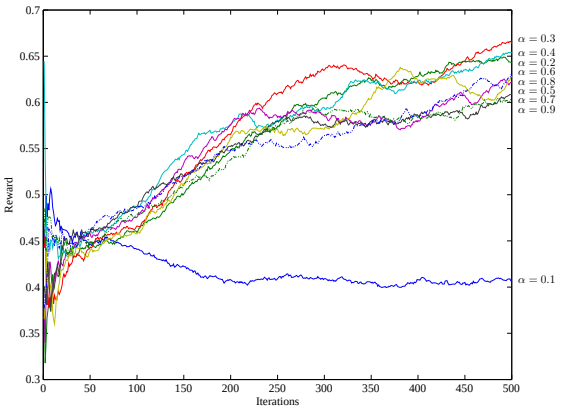

Figure 4. Effect of step size parameter, α

affect of the step size, α, on learning performance. We can see that all values in the range of $0.2 - 0.9$ offer comparable performance, while mid-range values are preferred.

For the action selection policy, we used the popular $\epsilon-$greedy policy and experimented with both constant values of ϵ, as well as values that decreased according to a simulated annealing-like cooling schedule:

$$\epsilon_t = \epsilon_0 \left(\frac{\epsilon_N}{\epsilon_0}\right)^{\frac{t}{N}} \quad (6)$$

where ϵ_t is the probability of taking a random action at step t, where t increases from 0 to N. The initial and final epsilon values were set to 0.9 and 0.01, respectively. Figure 5 compares the results of using a constant-valued ϵ policy versus the decreasing policy. Learning is seen to converge at a slower rate using a decreasing ϵ but converges on a higher average reward. Using a constant value of $\epsilon = 0.01$ is too low, and offers poor performance.

The Fuzzy ARTMAP neural network contains several parameters of its own. We have used the parameter val-

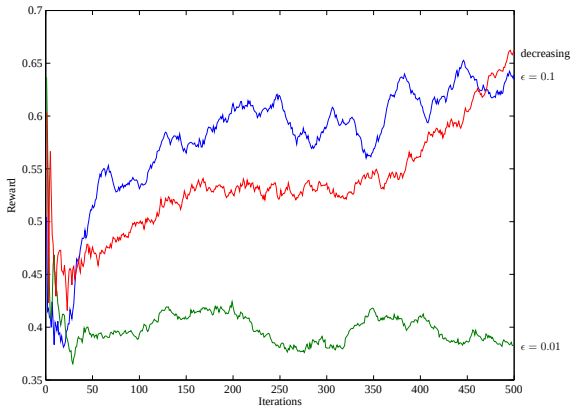

Figure 5. Comparison of action selection policies

Algorithm	α	γ	λ
Q-learning	0.3	0.99	N/A
Sarsa	0.3	0.99	N/A
Sarsa(λ)	0.3	0.99	0.7

Table 2. Parameters used in the reinforcement learning algorithms

ues recommended by Patrascu [6] with the exception of the ART$_b$ vigilance which was lowered to 0.9. We found that this improved learning performance. The parameters are provided in Table 3 and a description of each can be found in [3]. We note that the ART$_a$ vigilance is not set but controlled dynamically in order to ensure an output from the network [6].

Table 4 provides the optimal parameters recommended by the RL agent after 500 iterations, using the best learning conditions, as determined empirically. These are compared to Wolf et al.'s recommended parameters in [16]. The reader will notice that the two sets of parameters differ greatly.

4.2. Cross-validation

We used 3-fold cross validation in order to evaluate the system with the learned parameters. Two thirds of the images have been used for training and one third for testing, the mean value over the three runs being the final result. Table 5 provides the results of evaluating the parameters chosen by the agent. We also apply Wolf's recommended parameters to each test set.

Base vig	ARTb vig	Learning rate	Mismatch
0.95	0.90	0.09	0.00001

Table 3. Parameters used in each Fuzzy ARTMAP

Set	1	2	3	4	5	6	7	8	9	10
Wolf	13	87	2	105	50	1.2	0.3	0.1	0.2	0.7
RL	19.78	100.49	3.05	102.90	101.60	4.72	0.68	0.48	0.51	0.48

Table 4. Chosen parameters

Set	Rec	Pr (Text)	Pr (T+NT)	HM
Wolf	0.7945	0.4122	0.1935	0.3112
RL	0.4526	0.8764	0.8193	0.5695

Table 5. Parameter performance

We present four measures: recall, precision (using only text-containing images), precision (considering both text and non-text images) and the harmonic mean of the recall and second precision measure. Note that the recall remains the same as we add non-text images, but the precision decreases as we have more and more falsely detected text regions. (As in an information retrieval system, the performance depends on the generality of the dataset.)

While Wolf et al.'s suggested parameters lead to better recall, they have been set assuming that all images contain text, and thus lead to many more falsely detected text regions. The parameters determined by the RL agent balance the consideration of text-containing and non text-containing images, thus leading to significantly better recall values. The harmonic mean of precision and recall averaged over all three trials for the RL-determined parameters is an 83% improvement over the previously recommended parameters.

5. Conclusions

We have presented a general framework for applying RL to the parameter selection problem in multi-step image and vision-based tasks. Employing the Fuzzy ARTMAP ANN is an effective way to manage large state and action spaces. The DIRECT algorithm was successfully employed to allow for continuous action spaces while minimizing computational cost. The utility of the framework has been demonstrated by optimizing a set of ten parameters for an algorithm to detect text in images taken from video sequences. Introducing image features in the state space will allow for optimization on a per-image, rather than per-set basis. While the use of ground-truthed images has been useful in evaluating our framework, their use may be questioned when using methods capable of online learning. Now that the approach has proven successful, our aim is to replace the ground truth-based reward by some reward received directly from an optical character recognition (OCR) engine. This will permit on-line parameter optimization, which fully utilizes the benefits of reinforcement learning over other possible methods.

References

[1] L. Baird and H. Klopf. Reinforcement learning with high-dimensional continuous actions. Technical Report WL-TR-93-1147, Wright Laboratory,Wright-Patterson Air Force Base, 1993.

[2] R. P. Brent. *Algorithms for Minimization without Derivatives*. Prentice-Hall, Englewood Cliffs, NJ, 1973.

[3] G. Carpenter, S. Grossberg, N. Markuzon, J. Reynolds, and D. Rosen. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 3(5):698–713, 1992.

[4] R. Coulom. Feedforward neural networks in reinforcement learning applied to high-dimensional motor control. In *13th International Conference on Algorithmic Learning Theory*, pages 402–413. Springer, 2002.

[5] L. Kaelbling, M. Littman, and A. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[6] R. Patrascu and D. Stacey. Adaptive exploration in reinforcement learning. In *International Joint Conference on Neural Networks*, volume 4, pages 2276–2281 vol.4, 1999.

[7] C. Perttunen, D. Jones, and B. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Application*, 79(1):157–181, October 1993.

[8] C. Ribeiro. Reinforcement learning agents. *Artificial Intelligence Review*, 17(3):223–250, 2002.

[9] G. Rummery and M. Niranjan. On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University, 1994.

[10] M. Shokri and H. Tizhoosh. Using reinforcement learning for image thresholding. In *CCECE 2003-CCGEI 2003*, Montréal, 2003.

[11] W. D. Smart and L. P. Kaelbling. Effective reinforcement learning for mobile robots. In *Proceedings of the International Conference on Robotics and Automation (ICRA-2002)*, volume 4, pages 3404–3410, 2002.

[12] R. Sutton and A. Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., 1998.

[13] G. Taylor. A reinforcement learning framework for parameter control in computer vision applications. In *First Canadian Conference on Computer and Robot Vision*, London, Canada, 2004. To appear.

[14] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge University, 1989.

[15] C. Wolf. *Text Detection in Images taken from Video Sequences for Semantic Indexing*. PhD thesis, INSA de Lyon, 2003.

[16] C. Wolf and J. Jolion. Extraction and recognition of artifi cial text in multimedia documents. *Pattern Analysis and Applications*, 2003. To appear.

[17] P. Yin. Maximum entropy-based optimal threshold selection using deterministic reinforcement learning with controlled randomization. *Signal Processing*, 82(7):993–1006, 2002.