The Specific Applications of Data Management in Artificial Intelligence

学号	17030140014	姓名	张笑天

目录

Abstract

Artificial Intelligence is a developing field in a high speed now. The data management plays a key role in Artificial Intelligence. This article gives three specific applications of data management in Artificial Intelligence to show the importance of data management in AI and some tricks in practice. The following three examples are all vital practice in AI: Dota 2 with Large Scale Deep Reinforcement Learning, image classification with CNN, and Chinese couplet generation with Transformer. All of these applications has been experienced or researched by us in our learning time in Xidian University.

Keywords

Data Management; Artificial Intelligence; Data Mining; Reinforcement Learning; Computer Vision; Natural Language Processing;

1 Introduction

Completing with human being in games is a good way to test, verify and show the performance of Reinforcement Learning, or RL. The Reinforcement learning is a computational approach to understanding and automating goal-directed learning and decision-making. It is distinguished from other computational approaches by its emphasis on learning by an agent from direct interaction with its environment, without relying on exemplary supervision or complete models of the environment.[1] There are many famous breakthroughs and applications in RL such as playing Atari-games with Deep Q Network[2], the well-known AlphaGo[3], and the OpenAI Dota2[4] we are about to talk about.

Image classification is a widely researched problem in Computer Vision, and Deep Convolutional Neural Network, Alexnet and its successors, is far beyond the traditional methods in solving this problem[5]. For example, they have got remarkably high scores in ImageNet contest than ever before and have been vastly used in more relative fields like RL mentioned before or Natural Language Processing which we will discuss on later. The cornel CS5670 course shows a great example in CNN theory and application.[6]

Natural Language Processing as known as NLP is also vastly used in industry like Computer Vision. The translation problem or sequence to sequence learning is still a hot topic in this field. The encoder-decoder model[7] implies Deep Learning Network, DNN on this problem, and transformer based on Attention is more powerful than some neutral networks8. We are going to show a Chinese couplet generation program based on Attention with guide from Microsoft AI Lab9.

2 Dota 2 with Large Scale Deep Reinforcement Learning

a) Introduction to the algorithm

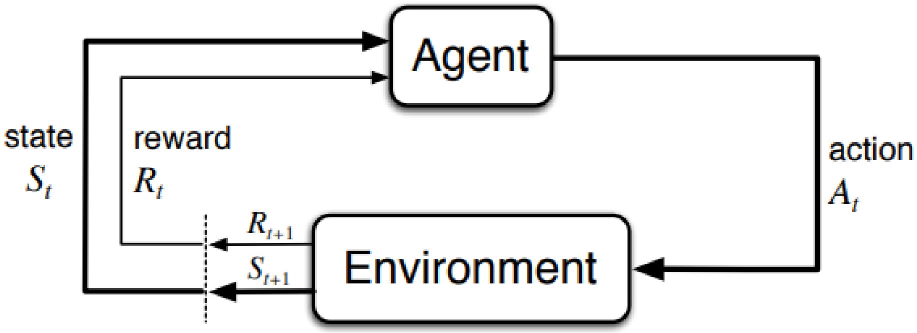


Figure 1

The basic idea of RL can be concluded as a finite Markov decision processing shown as Figure 1[10]. The agent's sole objective is to maximize the total reward it receives over the long run. Policy can be a function or generally may be stochastic. Reward sent to the agent at any time depends on the agent's current action and the current state of the agent's environment. Values indicate the long-term desirability of states after considering the states that are likely to follow, and the rewards available in those states. In the training of Dota2 AI, policy(π) is defined as a function from the history of observations to a probability distribution over actions. The policy network is designed to receive observations from our bot-API observation space and interact with the game using a rich factorized action space. These structured observation and action spaces heavily inform the neural network architecture used. Figure 2 is a simplified model architecture. The detail of architecture can be seen in the paper of OPENAI as cited before.

The OPENAI Five trained as shown has defeated TI champion OG team and has an exceedingly high winning rate in Dota 2 gaming against real human being[11].

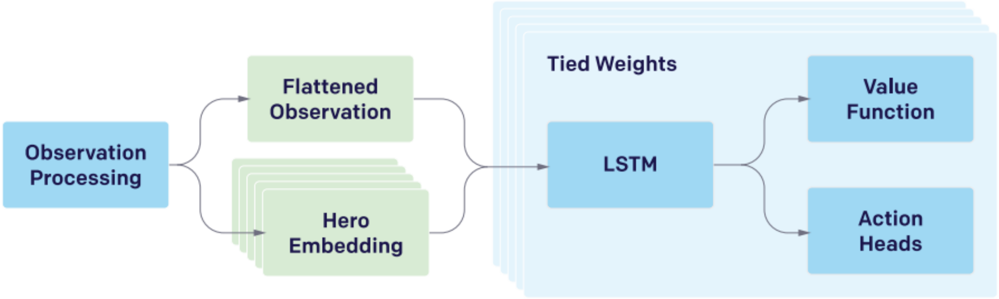


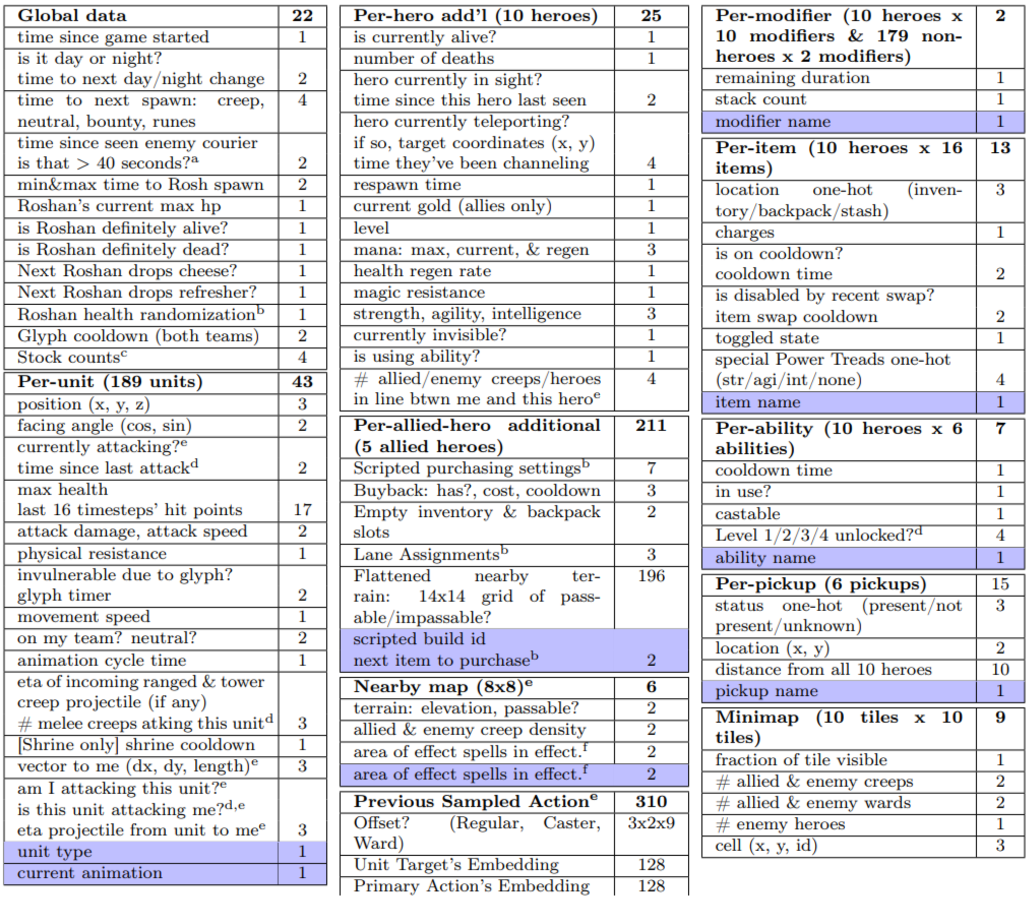
Figure 2

b) Data Management

The data processing problem here is how to present the large-scale, high-dimensional observation space and action space in game Dota 2 which has a large map, many heroes and complex gaming rules.

One simple way to tackle observation is to just read pixels like monitor as Deepmind in Q-Learning[12] and it is also suitable for CNN which Q Learning might use. However, for Dota 2 screen with so much information unlike Atari-2600, it is infeasible to render each frame to pixels in all training games; this would multiply the computation resources required for the project with many folds.

Instead of using the pixels on the screen, they approximate the information available to a human player in a set of data arrays. This approximation is imperfect; there are small pieces of information which humans can gain access to which they have not encoded in the observations. All float observations are normalized before feeding into the neural network. The Table 1 below shows the full observation space.



Global data	**22**
time since game started	1
is it day or night? time to next day/night change	2
time to next spawn: creep, neutral, bounty, runes	4
time since seen enemy courier is that > 40 seconds?[a]	2
min&max time to Rosh spawn	2
Roshan's current max hp	1
is Roshan definitely alive?	1
is Roshan definitely dead?	1
Next Roshan drops cheese?	1
Next Roshan drops refresher?	1
Roshan health randomization[b]	1
Glyph cooldown (both teams)	2
Stock counts[c]	4

Per-unit (189 units)	**43**
position (x, y, z)	3
facing angle (cos, sin)	2
currently attacking?[e] time since last attack[d]	2
max health last 16 timesteps' hit points	17
attack damage, attack speed	2
physical resistance	1
invulnerable due to glyph? glyph timer	2
movement speed	1
on my team? neutral?	2
animation cycle time	1
eta of incoming ranged & tower creep projectile (if any) # melee creeps atking this unit[d]	3
[Shrine only] shrine cooldown	1
vector to me (dx, dy, length)[e]	3
am I attacking this unit?[e] is this unit attacking me?[d,e] eta projectile from unit to me[e]	3
unit type	1
current animation	1

Per-hero add'l (10 heroes)	**25**
is currently alive?	1
number of deaths	1
hero currently in sight? time since this hero last seen	2
hero currently teleporting? if so, target coordinates (x, y) time they've been channeling	4
respawn time	1
current gold (allies only)	1
level	1
mana: max, current, & regen	3
health regen rate	1
magic resistance	1
strength, agility, intelligence	3
currently invisible?	1
is using ability?	1
# allied/enemy creeps/heroes in line btwn me and this hero[e]	4

Per-allied-hero additional (5 allied heroes)	**211**
Scripted purchasing settings[b]	7
Buyback: has?, cost, cooldown	3
Empty inventory & backpack slots	2
Lane Assignments[b]	3
Flattened nearby terrain: 14x14 grid of passable/impassable?	196
scripted build id next item to purchase[b]	2

Nearby map (8x8)[e]	**6**
terrain: elevation, passable?	2
allied & enemy creep density	2
area of effect spells in effect.[f]	2
area of effect spells in effect.[f]	2

Previous Sampled Action[e]	**310**
Offset? (Regular, Caster, Ward)	3x2x9
Unit Target's Embedding	128
Primary Action's Embedding	128

Per-modifier (10 heroes x 10 modifiers & 179 non-heroes x 2 modifiers)	**2**
remaining duration	1
stack count	1
modifier name	1

Per-item (10 heroes x 16 items)	**13**
location one-hot (inventory/backpack/stash)	3
charges	1
is on cooldown? cooldown time	2
is disabled by recent swap? item swap cooldown	2
toggled state	1
special Power Treads one-hot (str/agi/int/none)	4
item name	1

Per-ability (10 heroes x 6 abilities)	**7**
cooldown time	1
in use?	1
castable	1
Level 1/2/3/4 unlocked?[d]	4
ability name	1

Per-pickup (6 pickups)	15
status one-hot (present/not present/unknown)	3
location (x, y)	2
distance from all 10 heroes	10
pickup name	1

Minimap (10 tiles x 10 tiles)	**9**
fraction of tile visible	1
# allied & enemy creeps	2
# allied & enemy wards	2
# enemy heroes	1
cell (x, y, id)	3

Table 1

As for action space, Dota 2 is usually controlled using a mouse and keyboard. Most of the actions involve a high-level command (attack, use a certain spell, or activate a certain item), along with a target (which might be an enemy unit for an attack, or a spot on the map for a movement). For that reason, they represent the action our agent can choose at each timestep as a single primary action along with a few parameter actions.

The number of primary actions available varies each timestep. There are also 3 parameters outputs sometimes available for actions, delay, unit selection and offset. Figure 3 shows the whole action space.

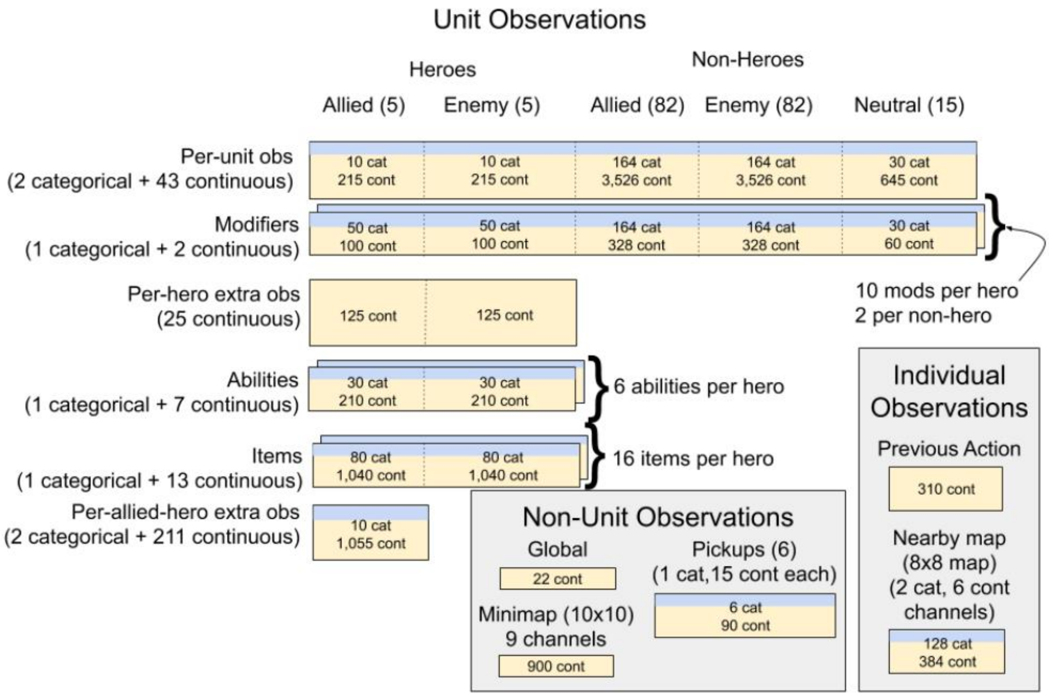


Figure 3

c) Conclusion

The way that data is managed and processed in this project is not very universal. In other words, it is customized for Dota 2 which is quite common in other gaming AI. For example, Blizzard and Deepmind developed an environment to supply API for AlphaStar[13] AI in Starcraft II .[14] Rather than just universally using image or input device states as input, the specific observation and action space is a necessary way to lower hardware compute and gain better performance with more workload in data analysis and preprocessing. This is a typical example that elegant data management at first will make the work easier. The time and money costs in data preprocessing is much less than training a long time on expensive GPUs and CPUs.

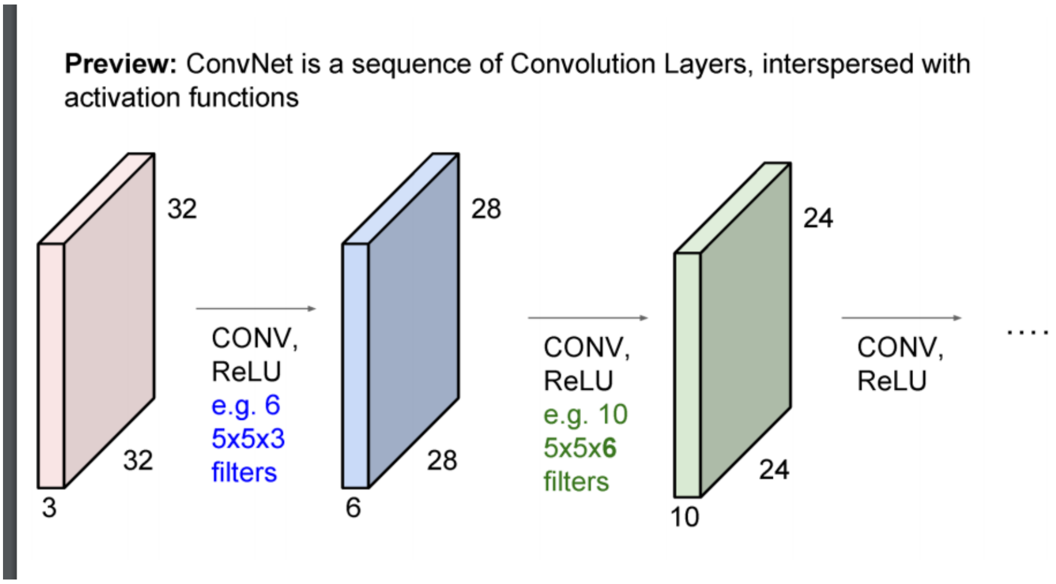
3 Image Classification with Convolutional Neural Networks

a) Introduction to the algorithm

Convolution leverages three important ideas that can help improve a machine learning system: sparse interactions, parameter sharing and equivariant representations. Moreover, convolution provides a means for working with inputs of variable size.[15]

All these features make the CNN can present more abstract features as layers goes deeper.

Figure 4 shows a simple CNN structure. Figure 5 shows visualization of a CNN.



Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions

Figure 4

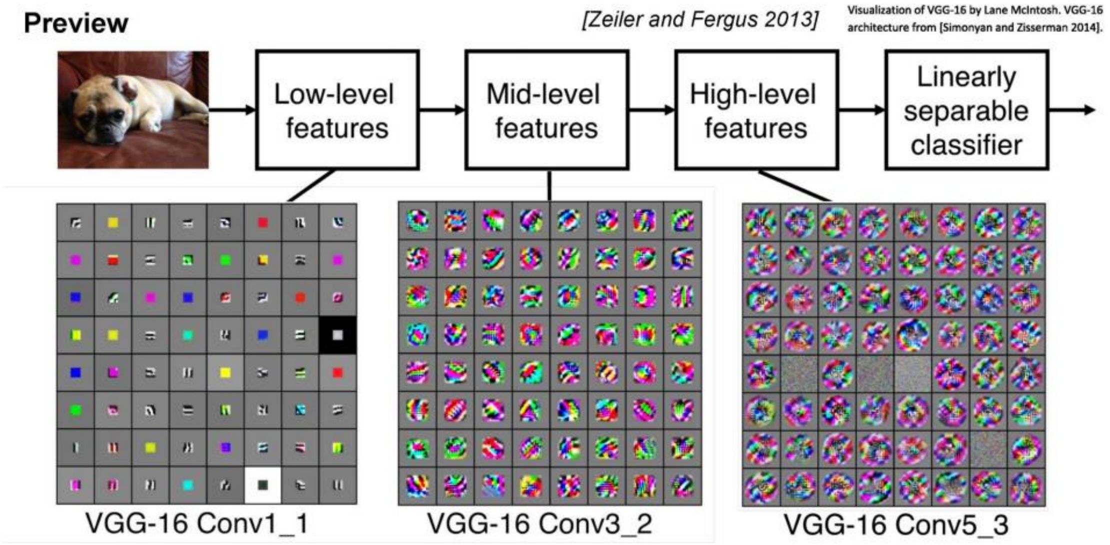


Figure 5

When it comes to classification, a convolutional neural network can be thought of as a function from images to class scores with adjustable weights leading to a very non-linear mapping from images to features or class scores.

b) Data Management

The data management for CNN might be easier than other machine learning tasks since the input we need to consider is only images. The images should be standardized to make their pixels in the same and reasonable ranges.

The Alexnet system only needs two preprocessing steps which is make every image gray and subtract the mean of training set for every pixel.

Of course, there are more needs for image management in different assignments. Many computer vision architectures require images of a standard size, so images must be cropped or scaled to fit that size. Even this rescaling is not always strictly necessary. Some convolutional models accept variably sized inputs and dynamically adjust the size of their pooling regions to keep the output size constant[16]. Other convolutional models have variable-sized output that automatically scales in size with the input, such as models that denoise or label each pixel in an image.[17]

Global contrast normalization (GCN) aims to prevent images from having varying amounts of contrast s by subtracting the mean from each image, then rescaling it so that the standard deviation across its pixels is equal to some constant s. This approach is complicated by the fact that no scaling factor can change the contrast of a zero-contrast image (one whose pixels all have equal intensity). Images with incredibly low but non-zero contrast often have little information content. Dividing by the true standard deviation usually accomplishes nothing more than amplifying sensor noise or compression artifacts in such cases. This motivates introducing a small, positive regularization parameter λ to bias the estimate of the standard deviation. Alternately, one can constrain the denominator to be at least ε. Given an input image χ, GCN produces an output image χ‘, defined such that

$$X'_{i,j,k} = s\frac{X_{i,j,k} - \bar{X}}{\max\left\{\epsilon, \sqrt{\lambda + \frac{1}{3rc}\sum_{i=1}^{r}\sum_{j=1}^{c}\sum_{k=1}^{3}\left(X_{i,j,k} - \bar{X}\right)^2}\right\}}. \tag{2-1}$$

Local contrast normalization, LCN ensures that the contrast is normalized across each small window, rather than over the image. Various definitions of local contrast normalization are possible. In all cases, one modifies each pixel by subtracting a mean of nearby pixels and dividing by a standard deviation of nearby pixels.

When facing inputs with remarkably high dimensions which may cause the curse of dimensionality. Principal component analysis, PCA might handle it. With minimal additional effort PCA provides a roadmap for how to reduce a complex data set to a lower dimension to reveal the sometimes hidden, simplified dynamics that often underlie it.[18]

It is also easy to improve the generalization of a classifier by increasing the size of the training set by adding extra copies of the training examples that have been modified with transformations that do not change the class. Object recognition is a classification task that is especially amenable to this form of dataset augmentation because the class is invariant to so many transformations and the input can be easily transformed with many geometric operations. As described before, classifiers can benefit from random translations, rotations, and in some cases, flips of the input to augment the dataset. In specialized computer vision applications, more advanced transformations are commonly used for dataset augmentation. These schemes include random perturbation of the colors in an image[19] and nonlinear geometric distortions of the input[20].

c) Conclusion

The data management in Computer Vision might not so complex but it is of vital since image is the key information provided for the network. The goal in CNN data management is to make network works better with more featured data, more standard data and normalized data.

The ways of image data processing introduced here can be also applied in both newer machine learning fields such as GAN or VAE and traditional computer vision like feature detection[21] or view modeling[22].

4 Chinese couplet generation

a) Introduction to the algorithm

The sequence to sequence problem can be defined as a sequence prediction problem with sequence as input and output, both of which has undetermined length.

There are many applications of sequence to sequence such as translation, QA and documents abstract generating. Simple RNN or LSTM cannot handle these applications. Then a new structure is introduced as Encoder-Decoder shown as Figure 6.

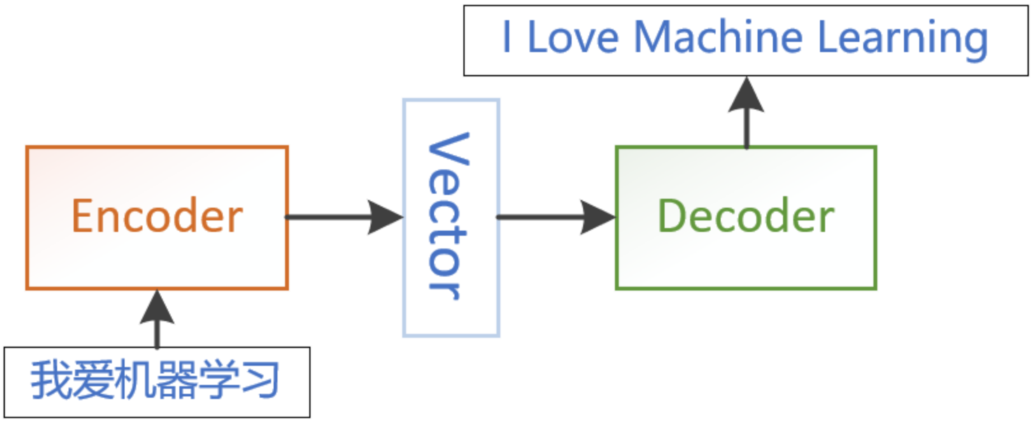


Figure 6

The input is embedded as a vector with certain dimension and passed into encoder. Encode codes the input into fixed-length vector called semantic encoding vector. Decoder decodes vector into output sequence.

In application, encode and decode can be chosen from CNN, RNN, GRU or LSTM separately.

One clear limitation of this architecture is when the context C output by the encoder RNN has a dimension that is too small to properly summarize a long sequence.[23] The attention mechanism is then introduced.

Attention mechanism can be summarized as read the whole sentence or paragraph, then produce the translated words one at a time, each time focusing on a different part of the input sentence in order to gather the semantic details that are required to produce the next output word as shown in Figure 7.

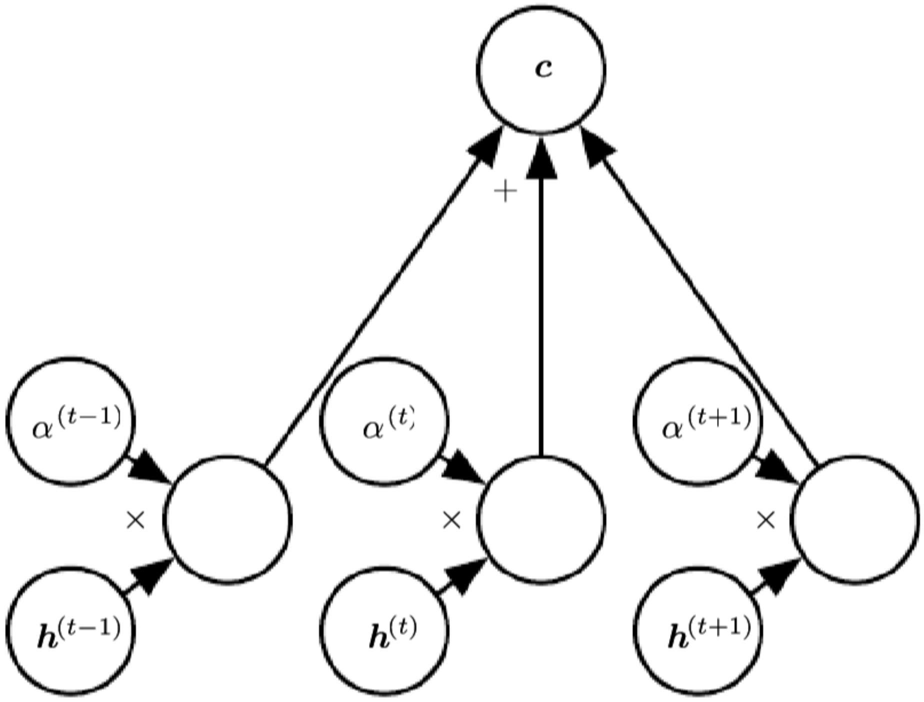


Figure 7

We can think of an attention-based system as having three components:

1. A process that "reads" raw data (such as source words in a source sentence), and converts them into distributed representations, with one feature vector associated with each word position.

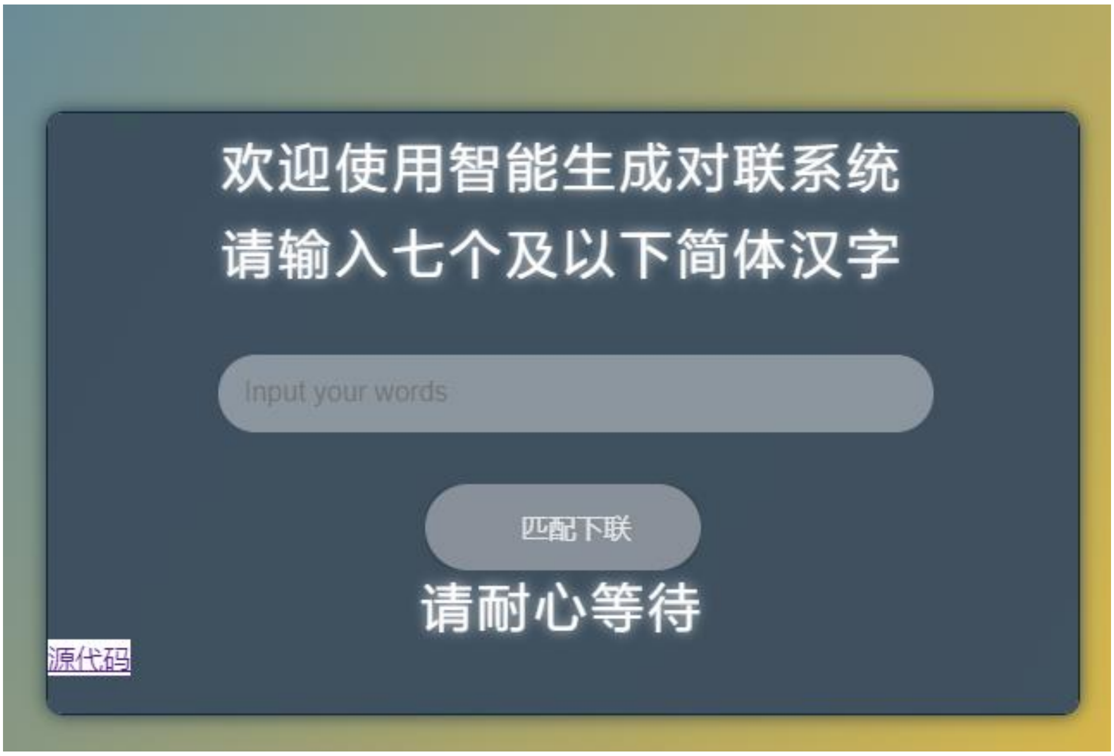
2. A list of feature vectors storing the output of the reader. This can be understood as a "memory" containing a sequence of facts, which can be memory retrieved later, not necessarily in the same order, without having to visit all of them.

3. A process that "exploits" the content of the memory to sequentially perform exploits a task, at each time step having the ability put attention on the content of one memory element (or a few, with a different weight).

The third component generates the translated sentence.

b) Data Management

We are going to show a project following Microsoft AI edu called Chinese Couplet Generating System[24], which is open-source[25] in Figure 8 and how we generate first vector with help of tensor2tensor[26].



欢迎使用智能生成对联系统

请输入七个及以下简体汉字

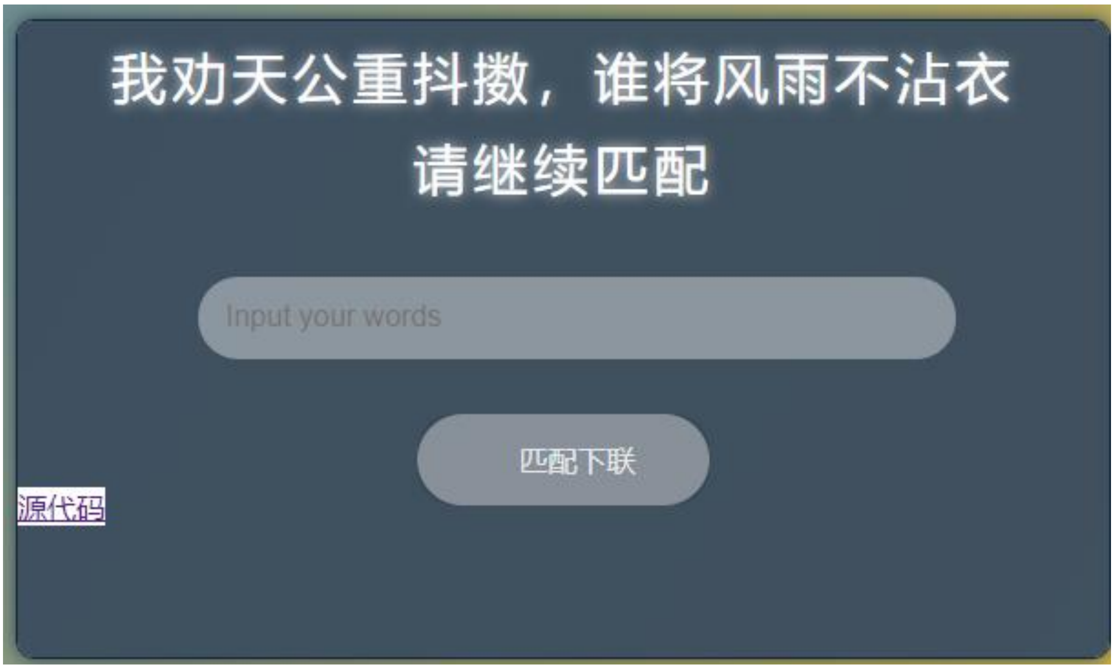
请耐心等待

Figure 8

Although our result would be Chinese couplet, the couplet[27] we get from GitHub is not enough. So, we also find some Tang poetry[28] from GitHub, which meet our demand that every sentence must correspond another sentence. The Tang poetry is divided into up and down we want to process. And then their characters are split with space and merged into a file to be counted. The cleaned data then must be transferred into binary files that tensor2tensor can recognize. In tensor2tensor, attention model is trained with problem registered. In this process, TensorFlow helps with parameter saving and checkpoints.

c) Conclusion

Figure 9 shows the achievement of this program and it is really of importance to gain enough data and get it cleaned correctly. After some training tries, we find the cause of a bug that length of up and down don't match is that original Chinese couplet data is not enough to offer length information and our Tang poetry up and down don't match which forces us to search new data from open-source websites.



我劝天公重抖擞，谁将风雨不沾衣

请继续匹配

Figure 9

5 Conclusion

In this article, we show three specific problems we have researched or programmed. And we can find that proper data management is a must in develop these applications. For three specific applications, we first introduced its algorithm, then we tell the detail of data processing we have done. At last, we conclude the key point of data management.

Along with growth of AI, data science is getting more and more important in computer science. We hope this article can show you some basic ideas of data management in these simple or basic applications can be finished by an undergraduate.

Finally, we are grateful to learn in Xidian University to have some great courses in Computer Science and Technology School, especially the courses help us go deeper in data science such as media data management, distributed system, machine learning, data mining, artificial intelligence, python and algorithm.

References:

[1] Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. Second edition. Cambridge, Massachusetts, London, England. The MIT Press. 2014, 2015. P15

[2] Mnih, V., Kavukcuoglu, K., Silver, D. et al. Human-level control through deep

reinforcement learning. Nature 518, 529–533. 2015

[3] Silver, D., Huang, A., Maddison, C. et al. Mastering the game of Go with deep neural networks and tree search. Nature 529, 484–489. 2016

[4] OpenAI. Dota 2 with Large Scale Deep Reinforcement Learning. arXiv:1912.06680. 2019

[5] Alex Krizhevsky. Ilya Sutskever. G E Hinton. ImageNet classification with deep convolutional neural networks. Communications of the ACM. 2012

[6] CS5670 Lectures. Spring 2019. http://www.cs.cornell.edu/courses/cs5670/2019sp/lectures/lectures.html

[7] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre er al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. EMNLP. 2014

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar et al. Attention is All you Need. Advances in Neural Information Processing Systems 30 (NIPS 2017). P6000-6010. 2017

[9] Microsoft Corporation. https://github.com/microsoft/ai-edu/tree/master/B-%E5%AE%9E%E8%B7%B5%E6%A1%88%E4%BE%8B/B13-AI%E5%AF%B9%E8%81%94%E7%94%9F%E6%88%90%E6%A1%88%E4%BE%8B

[10] Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. Second edition. Cambridge, Massachusetts, London, England. The MIT Press. 2014, 2015. P54

[11] OPENAI https://openai.com/projects/five/

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver et al. Playing Atari with Deep Reinforcement Learning. NIPS Deep Learning Workshop. 2013

[13] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev et al. StarCraft II: A New Challenge for Reinforcement Learning. arXiv:1708.04782. 2017

[14] Deepmind. https://github.com/deepmind/pysc2

[15] Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning. 中国工信出版集团 人民邮电出版社. 2017. P203-204

[16] Alex Waibel, Hidefumi Sawai, Kiyohiro Shikano. Modularity and Scaling in Large Phonemic Neural Networks. IEEE Transactions on Acoustics, Speech, and Signal Processing. Volume: 37. Issue: 12. Dec 1989. P1888-1898

[17] Ayse Naz Erkan, Raia Hadsell, Pierre Sermanet et al. 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. 29 Oct.-2 Nov. 2007

[18] Jon Shlens. A TUTORIAL ON PRINCIPAL COMPONENT ANALYSIS Derivation, Discussion and Singular Value Decomposition. arXiv:1404.1100. 2014

[19] Alex Krizhevsky. Ilya Sutskever. G E Hinton. ImageNet classification with deep convolutional neural networks. Communications of the ACM. 2012

[20] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324

[21] KRYSTIAN MIKOLAJCZYK AND CORDELIA SCHMID. Scale & Affine Invariant Interest Point Detectors. International Journal of Computer Vision 60(1), 63–86. 2004

[22] R. Hartley, "In defense of the eight-point algorithm", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 6, pp. 580-593, 1997.

[23] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015

[24] 智能对联生成系统开发小组. https://enigmazhang.tech/couplet/main/

[25] 智能对联生成系统开发小组. https://github.com/EnigmaZhang/NLP/

[26] Tensorflow. https://github.com/tensorflow/tensor2tensor

[27] wb14123. https://github.com/wb14123/couplet-dataset/releases

[28] Router8008. https://github.com/router8008/poetry-mining