Learning the Predictability of the Future with Regularization

E6691.2022Spring.YHYH.report.yj2677.hl3515
Yinsen Jia yj2677, Hanshan Li hl3515
Columbia University

Abstract

This project is our attempt to realize the ideas, reproduce the results and modify the original model by adding two regularization terms for Didac Suris, Ruoshi Liu, Carl Vondrick' work in Learning the Predictability of the Future[1] that was published in 2021 at arXiv.org. The goal of our project is implementing deep learning neural networks that can recognize what is taking place in a video and make predictions for the future hierarchically based on the Dense Predictive Coding(DPC) and in the Hyperbolic Space with regularization. Regarding the fact that the four datasets mentioned in the original paper including FineGym, Hollywood2, Kinetics- 600 and MovieNet are extremely large and time-consuming to download and process, in this paper we focus on the performance of the proposed model on FineGym dataset. Experiments on our DPC and hyperbolic encoding with regularization show that the Top 1, Top3, Top5 train accuracies over 288 classes are 22.27%, 58.59%, 81.51% respectively. On the other hand, the test accuracy, hierarchical top-down accuracy and bottom-up accuracy of our regularized model over 288 classes reaches 12.45%, 72.40% and 32.30% respectively, compared to the original paper's 13.37% , 66.64% and 33.04%.

1. Introduction

Today's computer vision is no longer satisfied with sorely acquiring, processing and understanding the given video itself. With unprecedentedly increased amount of available videos online for learning and numerous potential applications in the regions of human health, motion analysis, robotics etc, for many years, leveraging computer vision to predict the future of videos is pioneered as well.[2]

The biological mechanism of how humans anticipate outcomes in the future has been partially revealed by the results of neuroscience experiments.[3] While in the meantime, unlike the natural process of human brains, the core issue of the same task for artificial intelligence lies in what to predict in the future, which is hindered by the inevitable uncertainty of the future. For instance, consider the scene below: when a gymnast, i.e. an athlete of gymnastics is circling on uneven bars, what move will he or she take next? It might be jumping from one of the uneven bars to the other, but it might also be stopping circling and handstanding on the current bar, therefore the prediction of the future happens to be very tricky and even impossible.

Since it's hard to tell what's next, the most wise solution is to "hedge the bet". In other words, for the discussed situation, to parse the gymnastics activity into hierarchies and to differentiate between subtly different action phases in the same level, instead of guessing wildly either jumping from one of the uneven bars to the other or stopping circling and handstanding on the current bar, predict that the gymnast is at least circling.

To address the very issue, the original paper proposed an innovative hierarchical structure for learning predictable videos that are unlabeled and predict what's going to happen hierarchically.

Our major goals of this project were separated into three parts: First, triumphantly realizing the proposals of the referenced paper and modeling a reasonable architecture with Torch; Second, reproducing the authors' training and testing results with both euclidean and hyperbolic encoding approaches on the FineGym video dataset to the maximum extent; Third, add two regularized terms on the loss function to improve the performance of the DPC model with hyperbolic encoding.

Despite the oriented goals discussed above, nevertheless, there were numerous challenges and difficulties arising from our work while embodying these original proposals. The very first and even tortured challenge is related to concepts and terminology: since we tried not to misunderstand the conceptual description in the paper, we spent more than two weeks looking for all kinds of relevant materials, including but not limited to papers, journals, videos and grasping the core ideas.

The second technical challenge is to download, split and annotate the huge FineGym dataset with limited computational resources provided by google cloud platform and limited time in an appropriate and efficient way. To solve with this problem, we communicated with the authors of the original paper, with their help we finally managed to devise a couple of python scripts that automatically download and annotate the FineGym

dataset by leveraging the yt-dlp and ffmpeg library tool and taking advantages of the multiprocessing and the computational power of gpu.

2 Methodology of Original Paper

Based on the observation that hyperbolic geometry naturally and compactly encodes hierarchical structure, the authors of the original paper innovatively proposed hyperbolic encoding to replace the euclidean encoding to better represent the feature of video frames in the hyperbolic latent space with dense predictive coding on top of resnet as backbone.

2.1 Dense Predictive Coding

Proposed by Tengda Han, Weidi Xie and Andrew Zisserman, the DPC is a framework for self supervised representation learning on videos which learns a dense encoding of spatio-temporal blocks by recurrently predicting future representations.[5]

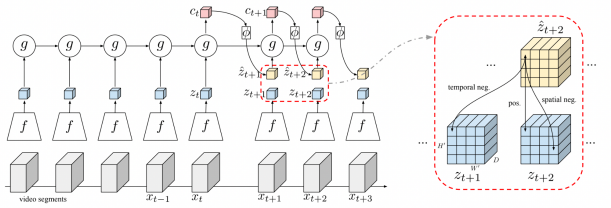

Fig. 1. Pipeline of Dense Predictive Coding[5]

The overall structure of Dense Predictive Coding is shown in Fig.1. A video clip is partitioned into multiple non-overlapping blocks $x_1, x_2, ..., x_n$, with each block containing an equal number of frames. First, a non-linear encoder function $f(.)$ maps each input video block x_t to its latent representation z_t, afterwards, an aggregation function $g(.)$ temporally aggregates t consecutive latent representations into a context representation c_t. The processes can be represented by equations as follows:

$$z_t = f(x_t) \tag{1}$$

$$c_t = g(z_1, z_2, ..., z_t) \tag{2}$$

After that a predictive function $\Phi(.)$ is introduced to predict the future. In detail, $\Phi(.)$ takes the context representation as the input and predicts the future clip representation. where c_t denotes the context representation from time step 1 to t, and z_{t+1} denotes the predicted latent representation of the time step $t+1$ as equation (3) and (4) describes.

$$\hat{z}_{t+1} = \Phi(c_t) = \Phi(g(z_1, z_2, ..., z_t)) \tag{3}$$

$$\hat{z}_{t+2} = \Phi(c_{t+1}) = \Phi(g(z_1, z_2, ..., zt)) \tag{4}$$

In the forward pass, the ground truth representation z and the predicted representation are carefully computed that strictly follows the above steps. The representation for the i-th time step is denoted as z_i and $\hat{z}_i$ hat, which have the same dimensions. Note that, instead of pooling into a feature vector, both z_i and $\hat{z}_i$ are kept as feature maps , which both maintain the spatial layout representation.

The similarity of the predicted and ground-truth pair is computed by the dot product. And the objective is to optimize the loss function as the formula (5) shown below. In essence, the loss function is simply a cross entropy loss that distinguishes the positive prediction and ground truth pair out of all other negative pairs[5].

$$L = -\sum_{i,k}[log \frac{exp(\hat{z}_{i,k}^T \cdot z_{i,k})}{\Sigma_{j,m} exp(\hat{z}_{i,k}^T \cdot z_{j,m})}] \tag{5}$$

2.2 Hyperbolic Encoding

In order to implement functions including non-linear encoder function $f(.)$, aggregation function $g(.)$ and predictive function $\Phi(.)$ in the equation(1)~(5), authors of the original paper built a neural network for each of them. To train the parameters of such models, the definition of distance metric is one of the issues that is of great importance, which is also the key and most valuable contribution of the original paper.

As discussed in section.1, the architecture is ought to hedge the bet and predict hierarchically, and the hyperbolic space encoding is the most ideal approach to parameterize the hierarchy for its intrinsic properties. A hyperbolic space can be viewed as a poincare embedding tree[6], which possesses circle lengths and areas that increase in size exponentially with their radius, as Fig.2 below depicts.

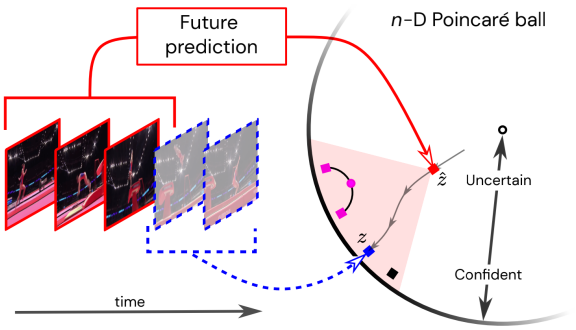

Fig. 2. The future prediction and structure of Poincare ball[1]
The three unblurred frames on the left are the given past, whereas the two succeeding blurred frames are the unobserved future. The pink circle represents the growth in generality when calculating the mean of two representations represented by two pink squares on the edge of the Poincare ball.

Denoted by H^n, the hyperbolic n-space is a Riemannian geometry that possesses constant negative curvature, the details of the Riemmanian metric can be found at [7,8]. Taking advantage of such metric, we can finally define the distance between an unobserved ground true z and the predicted scene $\hat{z}$ follows the equation (6) below, where suffix D indicates Remmainian manifold D^n.

$$d_D(z,\hat{z}) = cosh^{-1}\left(1 + 2\frac{\|z-\hat{z}\|^2}{(1-\|z\|^2)(1-\|\hat{z}\|^2)}\right) \quad (6)$$

It's worthy to point out that the mean between two leaf embeddings is a parent embedding in the hierarchical representation instead of another leaf embedding. For instance, in the Fig.2, the mean of two representations represented by the two pink squares on the edge of the Poincare ball is a pink circle towards the center instead of another node on edge. Besides, the minimum distance path between the squares is just the line in pink between the squares. Different from the traditional definition of tree in computer science, the hyperbolic representation owns continuous space and unlimited number of hierarchical levels, to put it another way, the closer a node from the center, the more obscure its meaning is, vice versa, the more far a node from the center, the more specific it is, and the most explicit semantics are always on the border.

2.3 Integrate Hyperbolic Encoding on DPC

To integrate the Dense Predictive Coding and Hyperbolic Encoding together and improve the performance, the authors of the original paper bring forward the model structure shown below.

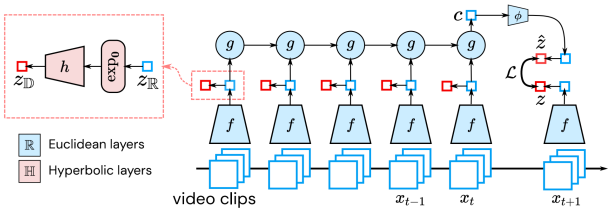

Fig. 3.Concatenate hyperbolic representations on Euclidean ones[1]

As the part is circled by a red rectangle on the top left of Fig.3. depicts, the Euclidean feature z_R is transformed to a Hyperbolic feature z_D, which further be utilized to calculate the loss function as the right part of Fig.3 depicts.

After integrating hyperbolic distance in the model, the loss function with Remmanian distance in the hyperbolic space can be formulated as equation (7), where distance is defined as the equation (6).

$$L = -\sum_i [log\frac{exp(-d_D^2(\hat{z}_i, z_i))}{\Sigma_j exp(-d_D^2(\hat{z}_i, z_j))}] \quad (7)$$

Employing such loss function reasonably minimizes the distance between predicted features for the future and unobserved future ground truth features. As discussed in section 2.1.2, when the suggested architecture encounters uncertainty, i.e. the situation that there are two possible outcome prediction features, then the midpoint will be returned.

3. Implementation

For the reason that the model proposed by the author of the original paper is considerable contricate and large (it's the work of columbia university PhD students to finetune the encoding of the [5]), the improvement of the original paper is out of our reach, therefore the methodology of our work follows section.2.1. However, we did modify the model by adding two regularization terms and introducing two parameters λ and μ, and we also added detailed comments to the code of the original paper. Our code can be found here, whereas our model weights and dataset is located in the google drive project folder.

We introduce the FineGym dataset for this project and discuss the data preparation in section 3.1, whereas the innovative regularization terms are walked through in section 3.2.

3.1 Dataset

As mentioned, all of the four proposed online video datasets in the original paper turned out to be extremely and unexpectedly large, thus time-consuming to process and train on the future prediction task. Therefore, for the sake of completeness and thoroughness of our project, we finally decided to focus our time and computational resources on a single dataset named FineGym that was created by a computer vision team of The Chinese University of Hong Kong led by Dian Shao.

Introduced in 2020, FineGym is a brand new dataset built on top of online gymnasium videos on the famous video resource website YouTube. Compared to existing action recognition datasets, FineGym is distinguished in richness, quality, and diversity. In particular, it provides temporal annotations at both action and sub-action levels with a three-level semantic hierarchy.[4] On the top level of semantic hierarchy, the FineGym contains 4 gym sports classes including vault, floor exercise, balance beam and uneven bars, whereas for the middle level it contains more specific 15 classes. Take balance beam for instance, *"balance beam"* event will be annotated as a sequence of elementary sub-actions derived from five sets: *"leap-jump hop"*, *"beam-turns"*, *"flight-salto"*, *"flight-handspring"*, and *"dismount"*. Furthermore, the sub-action in each middle level set will be further annotated with finely defined class labels, with an amount of 288 in total.

3.2 Innovative Regularization

In hyperbolic space, the 'density' of space would increase when feature points move away from the original point. Original loss function takes the advantage of this property as equation (7). By decreasing the distance between prediction features with true ground truth features and increasing the distance with negative examples. Feature encoder would automatically assign area for each stage class features, building a hierarchy tree. However, the original loss function could not guarantee features at the same level could be assigned at close radius. Even if the distance is far enough between two feature areas, one feature area might be assigned into the parent direction of the other features area. This would introduce some accuracy lost at the prediction stage, since features areas are not 'specific' enough. Therefore, The key innovative idea in this project is to add two more terms to regularize the features area to be more specific, which means be more far away from the original point. The modified loss function is equation (8)

$$L = -\sum_i [log \frac{exp(-d_D^2(\hat{z}_i, z_i))}{\Sigma_j exp(-d_D^2(\hat{z}_i, z_j))}] + \lambda \frac{1}{\left\| d_D^2(\hat{z}_i, 0) \right\|} + \mu \frac{1}{\left\| d_D^2(z_i, 0) \right\|} \quad (8)$$

Where we add two regularization terms and two corresponding parameters λ and μ to control the strength of these two regularizations.

When our encoder assigns features area to each class, it should consider placing each feature far away from the original point to reduce loss. The encoding features would be more specific so that when the prediction could not reach the most detailed label, it would not reduce too much accuracy (e.g. predict a totally different class). We would do detailed analysis of the performance of this loss function with experiments in chapter 4.

4. Results

4.1 Project Results

4.1.1 Train and Validation Results

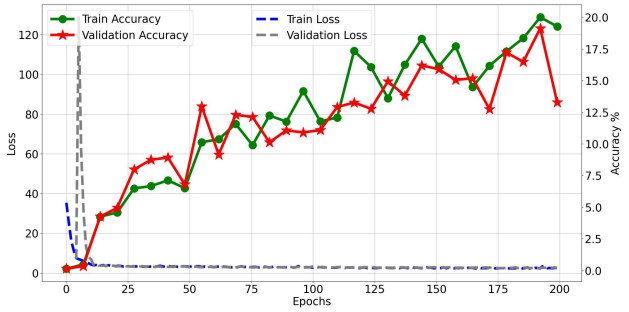

Fig. 4. Train & Validation loss and accuracy results (Hyperbolic)

OURS	Top-1 Accuracy	Top-3 Accuracy	Top-5 Accuracy
Train	22.27%	58.59%	81.51%
Validation	19.53%	51.43%	68.62%

Table.1. Train & Validation Top1 Top3 Top5 accuracy, Hyperbolic

As shown in Fig.4, with the new loss function, the hyperbolic model could reach 22.27% training accuracy and 19.53% validation accuracy. Both training loss and validation loss decrease fast. We also find that after 200 epochs, the model performs overfitting where training accuracy keeps increasing but validation accuracy remains the same, and test accuracy decreases. Therefore, we stop early at 200 epochs. We also put the top-3 and top-5 accuracy here, which shows that though the model might not classify accurately at the bottom level (the exact label), the prediction is still in the correct action class.

4.1.2 Test Results

Results of the hierarchical prediction test experiment for a video example with high predictability and a video example with low predictability are shown in Fig.5 and Fig.6 respectively. The bottom-left of the figure depicts the hierarchy 3 levels prediction output, whereas the bottom-right of the figure depicts the ground truth label.

For the high predictability example in Fig.5, since the future is more predictable, the model is able to predict that the athlete is going to do a giant circle backward with high confidence, thus output the whole three levels prediction as output. On the other hand, for the low predictability example in Fig.6, it's hard to predict the next move of the gymnast, consequently instead of guessing "leap forward with leg change" with low confidence, the model wisely takes just "Balance Beam" as a confident prediction.

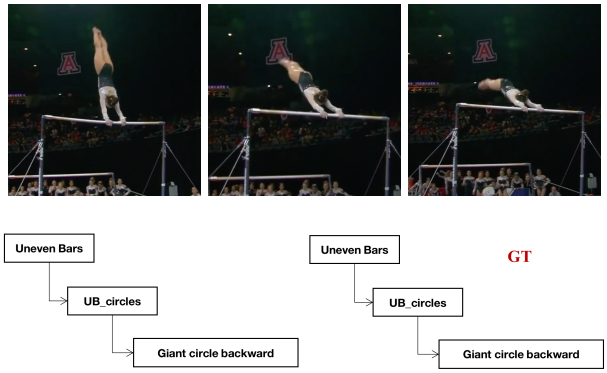

Fig.5 Example with high predictability

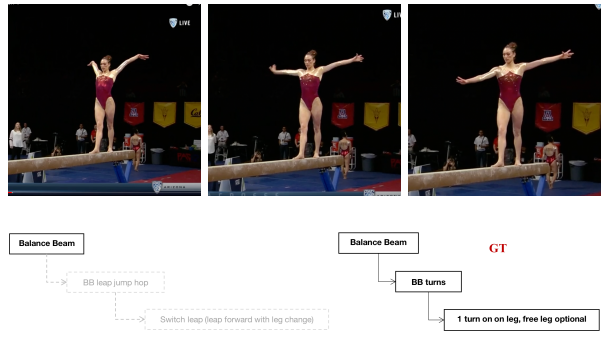

Fig.6 Example with low predictability

The reason why the situation in Fig.5 is with high predictability is obvious: the athlete is grasping the higher bar of the uneven bars, closing her leg and circling down, it's very likely that she will continue spinning till she reaches the top. The low predictability of the situation in Fig.6 can be explained by the fact that the athlete's motion can't completely imply her intention for the next move, it might be changing her leg, front flipping, or rotating in place etc. It's even impossible for a human to predict the next move as well.

4.2 Comparison with Original Paper

The table below shows the comparison between our test accuracy & hierarchy accuracy and the corresponding results from the original paper.

In the table, the bottom line is randomly selected results, it's evident that both our results and the original results are much higher than the random ones, no matter with Euclidean encoding or Hyperbolic encoding.

Comparison	Accuracy	Top-down hier. acc.	Bottom-up hier.acc.
Hyperbolic(ours)	12.45%	**72.40%**	32.30%
Hyperbolic[1]	13.37%	66.64%	33.04%
Euclidean(ours)	2.29%	28.05%	9.16%
Euclidean[1]	10.29%	56.67%	27.49%
Random[1]	0.00%	16.24%	5.67%

Table.2 Comparison of test accuracy and hier.acc.

When experimented with hyperbolic encoding, the test accuracy and bottom-up hierarchical accuracy of our regularized model is slightly lower than the original model, as 12.45% and 32.30% v.s. 13.37% and 33.04% respectively. However, our regularized model outperformed the original model in terms of the top-down hierarchical accuracy, with 72.40% and 66.64% respectively, around 5.8% higher than the original. This is related to our regularization term on the loss function, which encourages the lowest hierarchical labels to distribute more disperse on the edge of the Poincare ball and therefore leads to the clearer demarcation for all the classes.

On the other hand, the integrated regularization term also brings side effects when tested on Euclidean distance, the Euclidean accuracies in all aspects are lower than the original ones. The reason behind this is that the regularization term is only designed for hyperbolic space to help the semantic class labels to be more far away from the center of the Poincare ball and to be more distinct, which does not make sense to hyperbolic space and even do damage to gradients in euclidean space.

By comparison between our hyperbolic results and Euclidean results, the benefit of adopting hyperbolic encoding is demonstrated, just like what is discussed in section.2.

4.3 Insights Gained

Our modification on the original model that adding two regularization terms and two corresponding hyper -parameters on the loss function can be effective and

beneficial in certain cases to the model, especially for the top-down hierarchical accuracy as the test experiments pointed out and discussed in section 4.2. While the optimal assignment of the two suggested hyperparameters λ and μ is still unclear. It can be figured out by automatic hyperparameters searching in the future.

This project proves the advantages of features engineering in hyperbolic space, which could automatically generate hierarchical trees for downstream tasks to be more specific. Also, in hyperbolic space, we do not need to worry about the distance distortion which always involves large unstable performance in the model of euclidean space. More deep learning models could be transferred to hyperbolic space, especially those that require hierarchy or task understanding.

5. Future Work

As discussed in the last paragraph of section1 and section4.1, restricted by the limited time and computational resources, we are only able to implement and reproduce the result of the original paper on one out of the four datasets, i.e. FineGym. Owing to the fact that we are just normal columbia engineering graduate students, unfortunately we failed to complete a set of gymnastics neither on uneven bars nor on balance beam by ourselves, as a result we could not have an implementation of an intriguing test process on a video that was recorded on our own. Therefore the first improvement that can be made in future is to take advantage of other three online video datasets, especially Hollywood2, which contains daily scenes in normal life that can easily be reproduced by us and consequently provide the chance to test the implemented model on real life scenarios. By training on more datasets, hopefully the accuracy of the model can be promoted as well.

The second future work lies in the replacement of backbone neural networks. In both our work and the original paper, image classification neural network Resnet is used as the backbone, however, since it's designed to classify static images, the features gained from it might not effectively grasp the motion feature of humans and animals, as a result by replacing the backbone neural network Resnet with other neural networks specifically designed for video tasks, the performance of our model is very likely to be boosted.

6. Conclusion

We reproduce the original paper's results which is learning to predict the future by observing sequence frames with unsupervised learning methods. A new innovative regularization term has been added to the original loss function. With this loss function, the model encoder is encouraged to assign the action features to distribute more dispersed and far away from the original point. In that case, the output features are clearer demarcation for all the classes and avoid them mixed. Our experiments use a finegym dataset, and the results show that our new loss function could increase the accuracy of top-down accuracy, indicating that the features are more specific in hyperbolic space than before.

7. Acknowledgement

We are grateful to Professor Zoran Kostic for his lectures in Advanced Deep Learning, and this opportunity to apply what we have learned into this project. We would like to thank Feroz for his support and instruction in many aspects through the whole semester. We appreciate the contribution of the Google cloud and Tensorflow authors for the powerful tools which enable us to make use of GPUs and boost efficiency for complex implementations. We would like to express our sincere thankfulness to current PhD students Didac Suris, Ruoshi Liu and Carl Vondrick for their help on the video data processing throughout the project and their previous exploration in the *Dense Predictive Coding and Hyperbolic space encoding*, enlightening us with basic concepts. We would also like to say thank you to the authors of the FineGym dataset Dian Shao, Yue Zhao, Bo Dai and Dahua Lin for providing us with a neat and robust dataset, as well as an opportunity to explore future prediction in deep learning.

8. References

[1]Surís, Dídac, Ruoshi Liu, and Carl Vondrick. "Learning the Predictability of the Future." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[2]Yuen, Jenny, and Antonio Torralba. "A data-driven approach for event prediction." European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2010.

[3]Kourtzi, Zoe, and Nancy Kanwisher. "Activation in human MT/MST by static images with implied motion." Journal of cognitive neuroscience 12.1 (2000): 48-55.

[4]Shao, Dian, et al. "FineGym: A hierarchical video dataset for fine-grained action understanding." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[5]Han, Tengda, Weidi Xie, and Andrew Zisserman. "Video representation learning by dense predictive coding." Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019.

[6]Nickel, Maximillian, and Douwe Kiela. "Poincaré embeddings for learning hierarchical representations."

Advances in neural information processing systems 30 (2017).
[7]Lee, John M. Riemannian manifolds: an introduction to curvature. Vol. 176. Springer Science & Business Media, 2006.
[8]Lee, John M. "Smooth manifolds." Introduction to Smooth Manifolds. Springer, New York, NY, 2013. 1-31.