Curriculum Learning for two-stage Object Detection

Nguyen Duong Tung, Koyuncuoglu Guelben

POSTECH exchange students

CV course submission

Abstract

In the real world, the human educational system is based on the scheme of learning from easy to hard. Inspired by this, researchers have adopted and proposed a learning scheme where the model will be exposed with training examples from easy to hard. In many aspects and fields of research, both curriculum, exposure from easy to hard, and anti-curriculum, show from hard to easy, have shown the benefit and effect of the model. In our approach, we will introduce and investigate the effect of applying curriculum in two-stage object detection at the level of bounding boxes. We find that for the benchmark dataset such as Pascal VOC, there is no significant improvement in accuracy, but the training time does improve from non-curriculum, this could be due to the dynamic size of the training set in each curricula strategy.

1. Introduction

From the process of learning in the human world, researchers have been inspired and proposed the learning scheme of letting the model be exposed to examples that are categorized from easy to hard (Bengio et al., 2009 [1]). There are many studies that suggest that curriculum learning could improve the convergence speed in the domain of natural language processing (Platanios et al., 2019 [7]). On the other hand, anti-curriculum learning chooses the hardest example first and then goes to the easy one later, or maybe just always goes with the hard one (Shrivastava et al., 2016 [9]). And this also proves significant improvement in performance. This could seem confusing and contradictory, but it is believed that there are reasons behind it. A reasonable explanation is that when using curricula strategy, the model when first exposed to the easy example could act like a regularizer and prevent the model from overfitting to the target examples (Wu et al., 2021 [12]). With the anti-curricula strategy, the model exposed to the hard example first has a faster convergence time since not having to process easy examples with not contribute much to the loss (Shrivastava et al., 2016 [9]).

However, since curricula proved that it does not always bring benefits to some specific fields and models (Wu et al., 2021 [12]), there are speculations that whether the model actually learns in a specific order when feeding the model in the correct curricula. This problem is widely known as implicit curricula. This is the study that studies whether there are connections between the learning order of the model and the order or training examples getting fed into the model. Wu et al. have conducted research regarding this problem. To understand the behavior of the model, they used the term learned iteration, which is the epoch that the model entirely correctly predicts the label of the data examples. In the research, they have used different models architecture and different learning approaches, and the result shows that the learned epoch is correspond with the time that the example is fed into the model. We tried to reproduce the same result, in the Fig. 1, in the left is the epoch that an example is learned across 142 architectures and strategies. On the right is Fully connected (FC) networks, VGG networks (VGG11 and VGG19) (Simonyan and Zisserman, 2014 [2]), and batch-normal networks such as ResNet18, ResNet50 (He et al., 2016 [4]), WideResNet28-10, WideResNet48-10 (Zagoruyko and Komodakis, 2016 [13]), DenseNet121 (Huang et al., 2017 [5]), EfficientNet B0 (Tan and Le, 2019 [11]), VGG11-BN (Simonyan and Zisserman, 2014 [2]), and VGG19-BN (Simonyan and Zisserman, 2014 [2]) are listed in the columns from left to right. We see a consistent implicit curriculum with training strategies and models. This proved that curriculum training order could affect the order that a model learned examples. And with the consistency in model type, how we define the scoring function (the level of difficulty) could be robust to the model type.

In our approach, we try to implement curriculum learn-

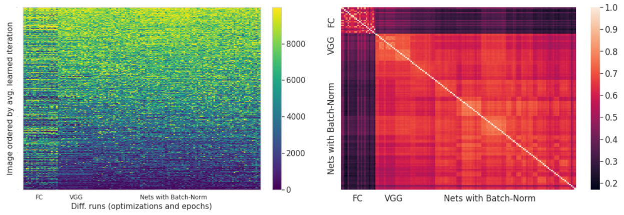


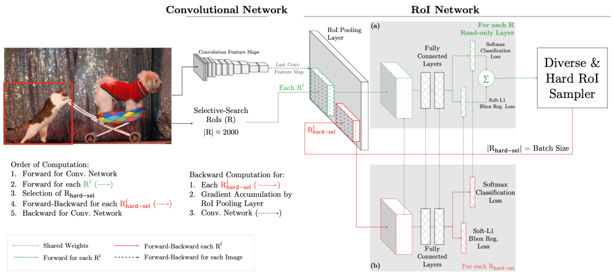
Figure 1. Implicit Curricula: The examples is learned in a order that is similar to the curricula stratergy

ing at the level of bounding boxes in the two-stage object detection models. This means that the bounding boxes that will be fed into the model will gradually or suddenly change from easy to hard and vice versa. And for the scoring function, we would be using the loss that the bounding boxes have from a pre-trained model. We could use a self-paced scoring function with two model sharing weights, one used for calculating loss of bounding boxes, one actually used for training, as in the model of Online Hard Example Mining (OHEM) (Shrivastava et al., 2016 [9]). However, with our limited resources and time, we could not build this model on time for the project reports and will take that as an approach to be headed in the future. After training and testing, the result come out as the curriculum did not increase much of accuracy and the convergence time in the full dataset but performs better when the number of data examples and the training resources is limited. Results show that the curriculum sampling method is relatively comparable but has not shown much improvement when compared with the original approach, and is likely to perform better in the limited resources setting.

1.1. Related Works

The work of applying curriculum learning in the field of Object Detection is seemingly new and not many researchers have tried this approach before. In 2019, Soviany et al [10]. have applied curriculum learning combined with Cycle-GAN to convert pictures with ground-truth (GT) labels to the target domain from the source domain. They divide the entire number of detected items by the average size of those objects to be the scoring function of each image (their difficulty). However, their approach is more suitable for the problem of modern object detectors, like Faster R-CNN, which are impacted by training (source) domain bias when used in new (target) domains. However, comparing the Faster R-CNN models trained on the target domain with GT labels, there are still large performance disparities with their approach.

There are also many others methods of sampling bounding boxes that shown effectiveness and address certain problems in object detection. One of the most prominent learning approaches that also accounts for the term difficulty in each example is the online hard learning example (OHEM) (Shrivastava et al., 2016 [9]). The model will use two identical model architectures that share weight for the whole process, one will be used to account for the loss (difficulty) of the bounding boxes generated, and one will be used for training examples that have been sampled while simultaneously sharing the weight for the first model architecture. The full architecture approach is in Fig. 2,. The reason for choosing only the hard or difficult examples to be presented in the training model is that these examples will contribute more in the gradient steps, while the easy examples with the low loss with not contribute much. This will help the model not to have to focus on training time on unnecessary examples, which then reduce training time and computational cost. The result does show improvement when compared to others sampling methods.



Order of Computation:
1. Forward for Conv. Network
2. Forward for each R^i (⟶)
3. Selection of R$_{hard-sel}$
4. Forward-Backward for each R$^i_{hard-sel}$ (⟶)
5. Backward for Conv. Network

Backward Computation for:
1. Each R$^i_{hard-sel}$ (⟶)
2. Gradient Accumulation by RoI Pooling Layer
3. Conv. Network (⇢)

Figure 2. Architecture of online hard mining examples

There are others bounding box sampling methods that are worth mentioning such as generating more positive/negative bounding boxes (Oksuz et al., 2020 [6]). However, this is just to encounter the problem of imbalance in the Object Detection tasks.

2. Method

As mentioned above, we would be implementing the curriculum learning sampling algorithm with a two-stage object detection task in the level of bounding boxes. First, we would like to introduce the two notions of curriculum learning, which is scoring and pacing function:

- **Score function**: This is a function that will return the score, or we could say it is the difficulty $f(x, y)$ of an input example x and label y.
- **Pace function**: This pacing function $p(t)$, will determine the size of the dataset that will be used in the training step at time t or epoch t. We can understand that at epoch t, there will be $p(t)$ bounding boxes with lowest-scored examples.

2.1. Difficulty metric

As mentioned above, we would be using the scoring function $f(x) \in \mathbb{R}$. An example x_i is said to have higher difficulty than example x_j if $f(x_i, y_i) > f(x_j, y_j)$. In our approach, we consider the scoring function to be the loss function of bounding boxes from a pre-trained model. We have:

- **Loss function**: This is a function that will return the loss of bounding boxes generated from the first stage of the two-stage object detection model, with a higher loss meaning higher difficulty. We have that, given a trained model $f_m : X \to Y$ and loss function $l(f_m(x), y) \in \mathbb{R}$:

$$f(x_i, y_i) = l(f_m(x_i), y_i)$$

2.2. Curriculum Strategy

We will approach the curriculum settings with three different curriculum strategies and compare them with the non-curriculum strategy. The three approaches would be discrete curriculum, semi-gradual curriculum, and fully gradual curriculum. Since curriculum learning is a new topic in the field of object detection, we would try to experiment and see which is the best approach.

But first, we would take a look at the general algorithm for the curriculum strategy for the model to train from easy to hard examples:

Algorithm 1 Curriculum learning with bounding boxes

```
    Input: Initial weight $w_0$, training set of bounding boxes generated $x_1$,...,$x_n$, scoring function $f : [N] \to \mathbb{R}$
  $(x_1, ..., x_n) \leftarrow sort((x_1, ..., x_n), f)$
  for $t = 1, ..., N$ do
      Choose the number of samples according to discrete, semi-gradual, or full gradual and have $(x_1, ..., x_j)$
      $w_t \leftarrow trainepoch(w_t - 1, (x_1, ..., x_j))$
  end for
```

In the process of finding the switching point, which is when we would change the level of difficulty of the data, we have found that there are results came out positively high when the easy data and hard data get trained with an equal number of epochs (Wu et al. 2021 [12]). To resemble the same settings, we would set up a milestone epoch in that we are gonna switch the level of difficulty of the dataset from easy to hard, and each neighbor milestone points have the same number of epochs. Let the total epoch be E, we choose to have n milestone points, and the number of epochs that run until the change to a new dataset is $m = int(E/n)$. Let N be the total number of bounding box samples that are generated.

Given the number of dataset changes as D. The three approaches could be presented below:

- **Discrete curriculum**: Given the number of datasets with different difficulty as D, we have the sorted list of examples from easy to hard bounding boxes. We will divide the bounding boxes list into D lists, when reaching a milestone epoch, we will add the next batch of bounding boxes to the training data. We have a number of bounding boxes, and the number of bounding boxes goes into training in epoch at time t: p(t) = $int(t/m) * int(N/D)$. We can understand it in the visualization in Fig. 3. From L1 to L5 is 5 different batches of bounding box data with an increase in difficulty.

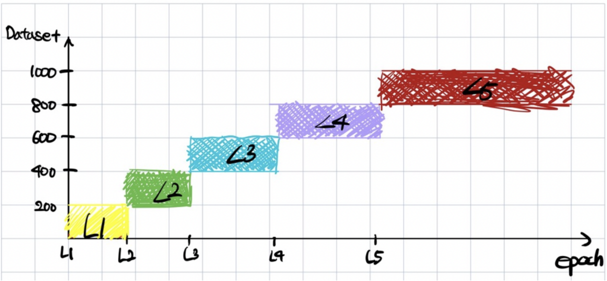


Figure 3. Discrete curriculum strategy

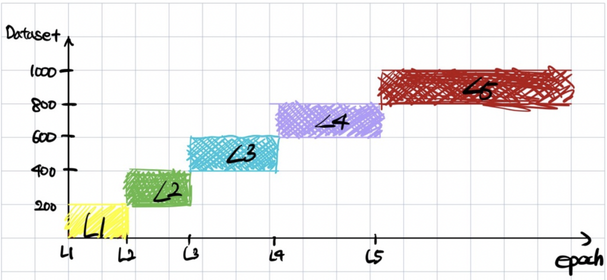


Figure 4. Semi gradual curriculum strategy

- **Semi gradual curriculum**: In this setting, we would have the ratio of contribution of each dataset level in each epoch. So that we would not occur a sudden increase in the dataset and the model might learn better. For example, at epoch t, the data might have a contribution of 0.8 for the first-level difficulty of the dataset and 0.2 contributions from the second-level difficulty dataset. The setting could be seen in the Fig. 4.

- **Fully gradual curriculum**: For this setting, we would slowly increase the influence of the next level bounding boxes dataset in each epoch (for example: increase 10 samples in each epoch). At some point, we could increase the level of increase by changing the number of added samples.

We could see the change in number of bounding boxes samples in the visualization in Fig. 5

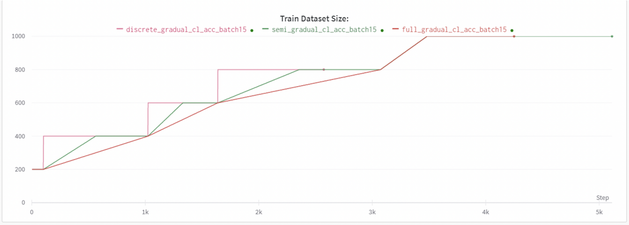


Figure 5. Change of dataset size

3. Experiment

3.1. Dataset

We conduct our research using the PASCAL VOC 2012 [3] (Everingham et al., 2012) dataset using for Object detection. The training set data is a collection of photos, each of which has an annotated file including a bounding box and an object class label for each of the twenty classes represented in the image. A single image may contain many objects from different classes. This is a benchmark dataset that has been used in many types of research before and we believed that this is suitable for our hypothesis and testing.

3.2. Experiment setup

We will train and set up our experiment with variations of the dataset. However, we find that changing the number of the samples' input images does not affect much the result of the experiment. So we train with the whole dataset.

In our setup, we would conduct our experiment using 6 GPUs NVIDIA GeForce with 11019MiB memory each. We would optimize it using the DataParallel library that is included in the Pytorch settings. However, we find that the memory consumption in the main GPU is relatively large because of the main task of calculating loss and computing gradient.

In the experiment, we would use the model of Faster R-CNN [8]. With the pretrained model for computing loss for curriculum learning, we would use Fast R-CNN with VGG16 network [2]. The same network architecture and network will be used for other experiments as well.

3.3. Evaluation metrics

The Average Precision, which is based on the ranking of detection scores, is often used to assess an object detector's performance on a class of objects. As a result, we report the mean AP (mAP) for all classes. The precision-recall (PR) curve for the detected items is used to calculate the AP score. If the intersection over union (IoU) measure is greater than 0.7, the PR curve is created by first mapping each detected bounding box to the most overlapping ground-truth bounding box.

Optimizer	D (in discrete)	Epoch	Batch
Adam	5	5000	15

Table 1. Hyperparameters

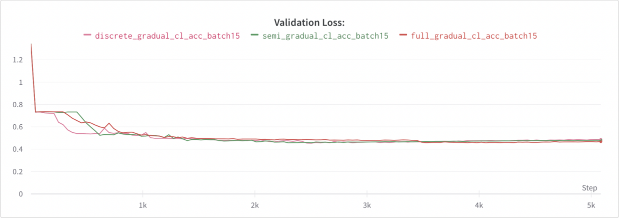


Figure 6. Validation loss of 3 curriculum strategies

3.4. Hyperparameters

About the optimizer, we would be using the Adam optimizer with $learningRate = 0.001, beta_1 = 0.9, beta_2 = 0.999, epsilon = 1e-07$.

We have conducted the experiments with various approaches and with IoU 0.7, the best hyperparameters are presented in Tab. 1

In the settings of semi-gradual curriculum learning, by every 10 epochs, we would add 0.1 to the proportion of the next dataset. On the fully gradual dataset, we would add 2 samples to every epoch.

3.5. Result

We could see the convergence time in the learning step when comparing the three settings of curriculum learning in Fig. 6. There is not much of a difference in the convergence time of the three, but we see a sudden validation loss decrease in the initial phase in the setting of discrete curriculum strategy. This suggests that with limited resources, the discrete settings will perform better.

When compared with the non-curriculum settings, the discrete curriculum seems to have less variance of loss compared with the non-curriculum settings but the loss in the non-curriculum setting seems to decrease faster. We could see the comparison in the Fig. 7. This might be an error in the settings of the non-curriculum so that the loss has a high variance in the end like that, but due to our limited time and resources, we have not figured a possible solution for this and will let this be a further approach in the future.

Now we would compare the precision and the accuracy of the settings. The result is in Tab. 2.

Normally, the batch size in other research settings would be much higher, ranging from 128 to 2048. However, with our limited resources, the batch size could only be 20, and increasing it may overload the GPU even tho we have used parallel computing. In the result, we could see that the discrete did improve compared with the non-curriculum setting

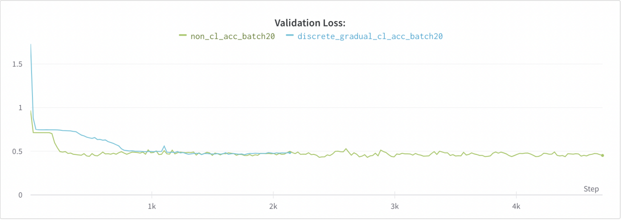


Figure 7. Validation loss of non-cl vs cl strategies

Experiment	Model	Batch	0.7mAP
Fast R-CNN	VGG16	20	62.7
Discrete	VGG16	20	**63.5**
Semi	VGG16	20	57.2
Fully	VGG16	20	54.0

Table 2. Result of the different settings of curriculum

of Fast R-CNN. However, we believed that this need to be assessed more since the batch size is really small compared to other research, and the result compare to other results of Fast R-CNN alone is somewhat lower in mAP. We could be concluded as discrete curriculum learning did improve in the limited resources settings. And the discrete did outperform the other two curriculum settings with higher convergence epochs and accuracy with mAP.

4. Conclusion

As we see in the result, the discrete curriculum setting did improve when compared with normal settings and with other curriculum learning strategies as well. We did not find a clear explanation why it is considered improved compared with other settings, but we have some theory about how it might be improved when compared with the original setting. In the paper research about curriculum learning (Wu et al., 2021 [12]), they found out that the curricula did not improve in the normal setting but performed well in limited resources and noisy settings. They also discussed that this might be because the distribution of the scoring function (difficulty) is more uniform. This might be the case when the setting in our approach is just 20 bounding boxes per batch since our GPU can not endure the higher setting in batch. There is also speculation about how curriculum learning might perform better. It could also act as a regularizer when the model learns from the easy data first, which makes it less overfit and perform well. In our future approach, we would want to test the curriculum setting with a self-paced function that changes the level of difficulty when the weight of the model is changed, as in the Ohem paper. Another promising approach in the future is testing the curriculum setting with noisy data since there has been a report about the successful implementation of curricula in the noisy setting (Wu et al., 2021 [12]).

References

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning*, 2009. 1

[2] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014. 1, 4

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 4

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[5] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1

[6] Kemal Oksuz, Baris Can Cam, Emre Akbas, and Sinan Kalkan. Generating positive bounding boxes for balanced training of object detectors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 894–903, 2020. 2

[7] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848*, 2019. 1

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 4

[9] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016. 1, 2

[10] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum self-paced learning for cross-domain object detection. *Computer Vision and Image Understanding*, 204:103166, 2021. 2

[11] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 1

[12] Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. When do curricula work? *arXiv preprint arXiv:2012.03107*, 2020. 1, 3, 5

[13] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 1