Let the Model Decide its Curriculum for Multitask Learning

Neeraj Varshney, Swaroop Mishra, Chitta Baral
Arizona State University, USA

Curriculum Learning in Multitask Setup

Curriculum learning is a type of learning in which you first start out with only easy examples and then gradually increase the difficulty level of training instances. For example, in schools, we are taught arithmetic before algebra and algebra before calculus.

Existing techniques arrange datasets either based on **human perception** of dataset difficulty or by **exhaustively searching** for the optimal arrangement. However, both these approaches have several limitations.

- **Human perception** of difficulty may not always correlate well with machine interpretation; for instance, a dataset that is easy for humans could be difficult for machines to learn or vice-versa.
- **Exhaustive search** is both computationally expensive and time-consuming. It becomes intractable as the number and size of datasets increases.

Proposed Training Structure

Input:
D: the training dataset,
$\{S_1, ..., S_K\}$: splits created from D
$frac$: fraction of previous split
Initialization: Model M
for $i \leftarrow 1$ **to** K **do**
  $train\_data = S_i$
  for $j \leftarrow 1$ **to** $i-1$ **do**
    $sampled\_S_j$ = Sampler$(S_j, frac)$
    $train\_data$ += $sampled\_S_j$
  end
  Train M with $train\_data$
end
Train M with D

Proposed Method

- We use model-based approaches to compute the difficulty scores: **Cross Review** and **Average confidence across epochs**.
- The training dataset D is divided into K splits (S_1, ..., S_K) based on the difficulty score, and model M is trained sequentially on these ordered splits. While training the model on split S_i, a fraction (frac) of instances from previous splits (S_j (j < i)) is also included in training to avoid **catastrophic forgetting**.
- The final step requires training on the entire dataset D as the evaluation sets often contain instances of all tasks and difficulty levels.
- Dataset and Instance level techniques vary in the way splits (S_1, ..., S_K) are created.

Dataset-Level Techniques

Each dataset represents a split and is arranged based on the average difficulty score of its instances i.e. score of a dataset D_k is calculated as:

$$d_k = \frac{\sum_{i \in D_k} s_i}{|D_k|}$$

where, s_i is difficulty score of instance $i \in D_k$

Instance-Level Techniques

Here, we relax the dataset boundaries and arrange instances solely based on their difficulty scores. We study two approaches of dividing instances into splits (S_1,..., S_K): Uniform and Distribution-based splitting.

- **Uniform**: We create K uniform splits from D
- **Distribution-based:** We divide D based on the distribution of scores such that instances with similar scores are grouped in the same split. It can result in unequal split sizes as the number of instances varies greatly across difficulty scores.

Results

Datasets	Single-Task		Instance-Level								Dataset-Level			
			Heterogeneous(B)		Uniform		Distribution (D)		D with $frac$=0.4		Random Order(B)		Proposed Order	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
SNLI	77.26	77.42	74.55	74.62	77.79	**77.79**	77.64	77.7	77.65	77.65	77.7	77.75	78.94	**79.05**
MNLI Mismatched	65.98	66.12	62.07	62.14	66.14	66.3	66.71	**66.78**	66.6	66.66	66.29	66.4	69.15	**69.28**
MNLI Matched	65.33	65.45	61.23	61.36	65.85	65.96	66.91	**67.01**	66.82	66.85	65.96	66.09	69.18	**69.33**
Winogrande	50	50	47.34	50	50.24	**50.27**	50	50.12	49.82	49.85	47.99	49.85	48.37	**50.3**
QNLI	74.21	74.23	66.78	66.81	70.42	70.44	71.81	**71.81**	71.38	71.38	70.35	70.39	73.75	**73.79**
EQUATE	98.99	98.99	98.99	98.99	99.14	99.21	99.57	**99.57**	99.28	99.28	99.57	99.57	99.57	**99.57**
QQP	80.04	80.06	75.34	75.35	78.89	78.9	79.23	**79.25**	79.11	79.12	79.23	79.26	80.27	**80.29**
MRPC	80.98	80.98	74.42	74.45	74.05	74.05	75.95	**75.98**	75.4	75.4	75.73	75.77	79.08	**79.08**
PAWS Wiki	52.45	52.49	55.92	56.01	53.15	53.16	54.39	54.47	70.59	**70.62**	56.44	56.51	80.33	**80.34**
PAWS QQP	68.25	68.41	73.03	73.03	69	69	71.83	71.83	78.84	**78.84**	73.08	73.12	83.46	**83.46**
ANLI R1	42.2	42.57	38.1	38.28	42.1	42.13	45.7	**45.7**	43.2	43.33	42.9	**43.04**	42.3	42.58
ANLI R2	38.1	38.78	35	35	39.8	**39.9**	38.9	39.05	37.2	37.25	38.4	**38.5**	36.8	36.97
ANLI R3	39.25	39.38	36.17	36.24	38.5	**38.62**	38.17	38.24	36.5	36.56	37.92	**38.03**	37.25	37.4
DNLI	84.68	84.83	80.36	80.48	83.51	**83.57**	83.15	83.2	82.09	82.12	82.52	82.59	82.67	**82.73**
HANS	-	-	49.06	49.07	48.95	49.01	48.3	48.38	49.39	**49.45**	48.22	**48.27**	48	48.09
Stress Test	-	-	55.28	55.44	56.2	56.31	58.66	**58.77**	57.7	57.75	56.74	56.84	59.94	**60.15**

Performance Improvement

- Instance and Dataset-level techniques achieve an average improvement of 4.17% and 3.15% over their respective baselines.
- This improvement in consistent across all the datasets and also outperforms single-task performance in most cases.

Uniform Vs Distribution splitting

- In instance-level experiments, distribution-based splitting shows slight improvement over uniform splitting.
- Due to superior inductive bias resulting from collation of instances with similar difficulty scores to the same split.

Adding instances from previous splits

- No improvement for dataset-level techniques as all the instances of a dataset are grouped in a single split hence no inductive bias.
- Improvement for instance-level techniques as previous splits contain instances of the same dataset hence, providing the inductive bias.