K-Mooc, cau, 인공지능 연구동향, 김영빈 교수님
제출자 최규형(choigww)

Project

인공지능 관련 논문 **3**편 읽고**,** 소개**/** 핵심 알고리즘**/** 결과**/** 본인 생각 쓰기 **(**편당 **.5p)**

---

16. Jaderberg, Max, et al. "Population based training of neural networks." arXiv preprint arXiv:1711.09846 (2017).

소개
주요한 hyperparameter optimization method은 parallel search와 sequential optimization으로 나눌 수 있으며, 이 두 방법론을 bridge하는 genetic algorithm 기반 population based training(PBT)를 제안함

핵심 알고리즘
1 랜덤 hyperparameter set과 weight으로 학습시키고자 하는 모델을 threshold까지 학습시킨다
2 Exploit
    2-1. Binary Tournament, 랜덤 모델보다 안좋으면 교체
    2-2. Truncation selection, 학습한 모델 성능이 모든 모델 중 하위 20%면, 상위 20% 중 랜덤하게 뽑아서 교체
3 Explore
    only for Exploit-ed models
    3-1. Perturb, 특정 factor multiplication
    3-2. Resample, 리샘플링

학습 끝날 때까지 위를 반복

결과
RL > 성능 향상, MT > 비슷비슷, GAN > 성능 향상

소감
hyperparameter 최적화라는, 어찌보면 엄밀한 연구의 범위라기엔 애매할 수 있는 부분에 대해 다룬 내용이라 좋았다. 체계적으로 삽질한다는 것이 무엇일까에 대하여 생각해보게 만드는 페이퍼다.

19.Izmailov, Pavel, et al. "Averaging weights leads to wider optima and better generalization." arXiv preprint arXiv:1803.05407 (2018).

소개

Stochastic Weights Averaging(SWA)은 Stochastic Gradient Descent(SGD)보다 robust하며, generalization에 강하고 테스트 성능이 뛰어나다. SWA는 SGD를 이용해 최적화 중 일정한 주기마다 weight average를 진행하여 가중치를 업데이트하는 방법이다. 여러 주기마다 가중치를 업데이트하는 것은 local solution에 대한 ensemble이라고 이해할 수 있다.

핵심 알고리즘

Algorithm 1 Stochastic Weight Averaging

Require:
weights $\hat{w}$, LR bounds α_1, α_2,
cycle length c (for constant learning rate $c = 1$), number of iterations n
Ensure: w_{SWA}
$w \leftarrow \hat{w}$ {Initialize weights with $\hat{w}$}
$w_{\text{SWA}} \leftarrow w$
for $i \leftarrow 1, 2, \ldots, n$ **do**
  $\alpha \leftarrow \alpha(i)$ {Calculate LR for the iteration}
  $w \leftarrow w - \alpha \nabla \mathcal{L}_i(w)$ {Stochastic gradient update}
  if $\text{mod}(i, c) = 0$ **then**
    $n_{\text{models}} \leftarrow i/c$ {Number of models}
    $w_{\text{SWA}} \leftarrow \frac{w_{\text{SWA}} \cdot n_{\text{models}} + w}{n_{\text{models}} + 1}$ {Update average}
  end if
end for
{Compute BatchNorm statistics for w_{SWA} weights}

결과

CIFAR-100, CIFAR-10 적용, SGD보다 항상 좋은 성능, FGE 대비 가성비 좋음 (연산량 적음)

DNN (Budget)	SGD	FGE (1 Budget)	SWA		
			1 Budget	1.25 Budgets	1.5 Budgets
		CIFAR-100			
VGG-16 (200)	72.55 ± 0.10	74.26	73.91 ± 0.12	74.17 ± 0.15	74.27 ± 0.25
ResNet-164 (150)	78.49 ± 0.36	79.84	79.77 ± 0.17	80.18 ± 0.23	80.35 ± 0.16
WRN-28-10 (200)	80.82 ± 0.23	82.27	81.46 ± 0.23	81.91 ± 0.27	82.15 ± 0.27
PyramidNet-272 (300)	83.41 ± 0.21	–	–	83.93 ± 0.18	84.16 ± 0.15
		CIFAR-10			
VGG-16 (200)	93.25 ± 0.16	93.52	93.59 ± 0.16	93.70 ± 0.22	93.64 ± 0.18
ResNet-164 (150)	95.28 ± 0.10	95.45	95.56 ± 0.11	95.77 ± 0.04	95.83 ± 0.03
WRN-28-10 (200)	96.18 ± 0.11	96.36	96.45 ± 0.11	96.64 ± 0.08	96.79 ± 0.05
ShakeShake-2x64d (1800)	96.93 ± 0.10	–	–	97.16 ± 0.10	97.12 ± 0.06

소감

Prediction에 대한 ensemble이 아니라 weight에 대한 ensemble이라는 관점의 전환. 게다가, 100% 알아먹진 못했지만 prediction ensemble만큼 weight ensemble이 좋다는 증명까지. 좋은 페이퍼다!

53.Brown, Tom B., et al. "Language models are few-shot learners." arXiv preprint arXiv:2005.14165 (2020).

소개
Few shot learning. 사람이라면 (거의) 누구나 할 수 있는, 몇 개의 예시만 보고 규칙을 학습하여 적용할 수 있게 되는 것. 역대급 패러미터 스케일로 무장하여 돌아온 OpenAI의 GPT-3의 few shot learner로서의 가능성을 탐구, 확인.

핵심 알고리즘
few-shot, one-shot, zero-shot learning 세가지 방식으로 학습한 뒤 task performance 측정.

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

결과
문장완성 태스크

Setting	LAMBADA (acc)	LAMBADA (ppl)	StoryCloze (acc)	HellaSwag (acc)
SOTA	68.0[a]	8.63[b]	**91.8**[c]	**85.6**[d]
GPT-3 Zero-Shot	**76.2**	**3.00**	83.2	78.9
GPT-3 One-Shot	**72.5**	**3.35**	84.7	78.1
GPT-3 Few-Shot	**86.4**	**1.92**	87.7	79.3

번역 태스크

Setting	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
SOTA (Supervised)	**45.6**[a]	35.0 [b]	**41.2**[c]	40.2[d]	**38.5**[e]	**39.9**[e]
XLM [LC19]	33.4	33.3	26.4	34.3	33.3	31.8
MASS [STQ+19]	37.5	34.9	28.3	35.2	35.2	33.1
mBART [LGG+20]	-	-	29.8	34.0	35.0	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 One-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	39.2	29.7	40.6	21.0	39.5

소감
논문이 너무 길어서 블로그 리뷰를 통해 나머지 내용들을 확인해 보았다. 결론적으로 소감을 말하자면, 논문이 이야기하는 바와 유사하게, GPT-3과 같이 어마어마한 데이터를 사전 학습시키는 것이 어디까지 가능할 지에 대해 생각해볼 필요는 있어 보인다. 학습 효율을 비약적으로 높인다면 얼마나 좋을까? 우리가 이 이상으로 많은 데이터를 얻지 못하게 된다면 인공지능의 발전도 멈춰야 할 것인가? 하는 문제와도 맞닿은 주제라고 본다.

재밌어 보여서 나중에 읽어볼 논문들

45.Sheng, Emily, et al. "The woman worked as a babysitter: On biases in language generation." arXiv preprint arXiv:1909.01326 (2019).

53.Brown, Tom B., et al. "Language models are few-shot learners." arXiv preprint arXiv:2005.14165 (2020).

42.Lim, Sungbin, et al. "Fast autoaugment." Advances in Neural Information Processing Systems. 2019.