9/23 「EECE695D」

↳ scaling works... why?

9/22 ~ 9/30 「find your team online (PLMS)」

~10/14 = choose a paper, write a proposal

~10/18 = visit the department office (with me), and get colab account.

~11/11 = experiments, write a paper, and submit via OpenReview.

~11/22 = write reviews. (report)

「team 👍👍 ⇒ 점수 ↑↑」

↳ ML domain에 대한 논문 다루기

P. 1) ① old (~2010S) = Big models do not generalize.

② Modern (2010S~) = Big models generalize better.

↓

「good old 이론」

「SLT」

↳ statistical Learning theory

---

① Model space 'F' = A set of all 'learnable' functions 「학습가능한 함수의 집합 이라는 space.」

② Risk, $R(f, D)$: Loss 「Predictive quality of function on dataset $D = \{z_1, z_2, \cdots, z_N\}$」

③ optimizer = 👍 risk

cover model space.

$$\min_{f \in F} R(f, D)$$

↳ space

risk

optimizer

$$= \left(R(\hat{f}, D) - R(f_{ERM}, D)\right)$$

optimizer error

$$+ \; R(f_{ERM}, P) - \min_{f \in F} R(f, P)$$

generalization error.

$$+ \; \min_{f \in F} R(f, P) - \min_{f \text{ measurable}} R(f, P)$$

Approximation error.

$$R(f, D) = \frac{1}{N} \cdot \sum_{i=1}^{N} \ell(f(x_i), y_i).$$

$$R(f, P) = \mathop{E}_{z \sim P} \ell(f(x), y)$$

「$x_1, x_2, \cdots$, ~ IID」

$$\frac{1}{N} \cdot \sum_{i=1}^{N} x_i$$

CamScanner로 스캔하기

estimation error
= optimization error
+ generalization error. 차이

「generalization error」

↳ Dtr Pd mis match.

small 「data > params」 (not anymore)

& D = gradient descent

「ground-truth, cubic function」

d=3

, Legendre polynomials.

「deep-double-descent」

d = 20 → smaller norm gradient.

ERM
↳ empirical risk.

$$\sup_{f \in F} |R(f,P) - R(f,D)| \le O\left(\sqrt{\frac{M}{N}}\right)$$

「M 파라미터들, F: parametrized model space F」.

hold with high probability.

under fitting 과소적합 | over-parameterized

param. < data set < param

estimation error.

But 경험적으로

↳ interpolation threshold

Moore-Penrose pseudo-inverse.

linear system

linear regression

---

deep-double-descent — windows on theory org.

$\min_{\theta} \|Y - \theta^T \cdot X\|^2$

↓

목표값 - train

↳ $\theta^* = (X^T \cdot X)^{-1} \cdot X^T \cdot Y$

linear regression

$Y = AX \Rightarrow X^T X Y = X^T \cdot X \cdot AX$

→ $Y = (X^T \cdot X)^{-1} \cdot X^T X A X$

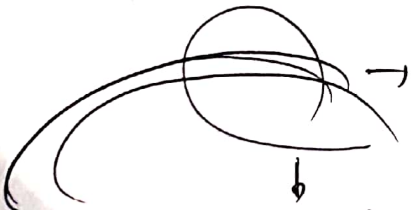
같은거

↳ pseudo-inverse : 주로 관심

1990s에 연구 발전 한계 도달.

Hebb, perceptron, pseudo-inverse

성능비교



→ 비슷한 성능

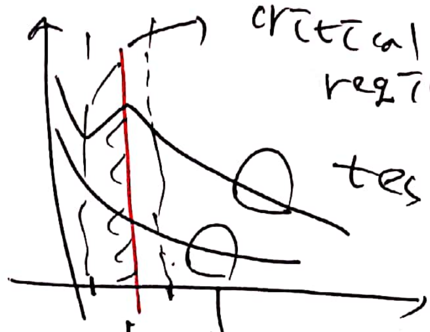
↓

연구 흥미 ↓.

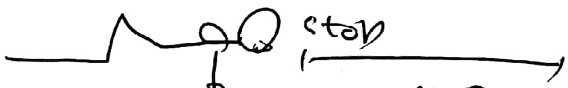
PELD 흥미 ↓↓.

2020. over parameterization

관심↑↑.

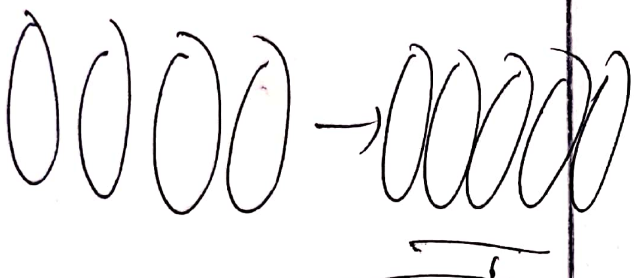


early stop



large model ↔ small model ← intermediate model

layer를 많이 쌓아도 성능 개선 X.



optimization ↑↑ (polynomial로도 가능).

width를 늘려서.

layer 수가 ↑↑ → vanilla으로도. 최적 (adaptive으로도 성능 ↑).

$y = a \cdot X \to ①$

$y = bc \cdot X \to ②$

① $\frac{1}{2}(y-ax)^2 \to \nabla_a l = -x\underbrace{(y-ax)}_{=r} \to ③$

② $\nabla_b l = -cx(y-bcx) \to ④$

$\nabla_c l = -bx \cdot (y-bcx)$

$(a - \alpha r x)$

$= a - \nabla_a \to ⑤$

$(b - \alpha\nabla_b)(c - \alpha\nabla_c)$

$= (bc + \alpha r c^2 x + \alpha r b^2 x + \alpha \square)$

$= bc + \alpha r (c^2+b^2) \cdot x + \square$

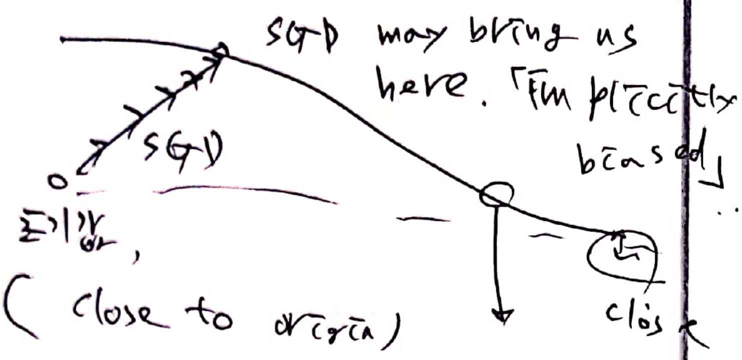
$= a + \alpha r (c^2+b^2) x + \square$

⑤와 달리, b, c와 같은 dependent한 값이 추가된다.

But → depth에 대한 영향이 불리하므로 개선이 오지는 X.

trajectory ?

$$\sup_{f \in F} |R(f,P) - R(f,D)| \le O(\quad).$$



$(y_i - \theta^T x_i)^2$

↓

$2(y_i - \theta^T x_i)\cdot(-x_i)$

$= \ulcorner r_i \cdot x_i \lrcorner$

$\nabla(L(\theta)) = \sum_{i=1}^{N} r_i \cdot x_i$

r_i는 x_i에 대한 scalar value

↓

$\nabla_\theta (L(\theta))$

$= \sum_{i=1}^{N} r_i \cdot x_i$ . .

$\theta^{(k)} = \theta^{(0)} + \sum_{i=1}^{N} \gamma_i \cdot x_i$

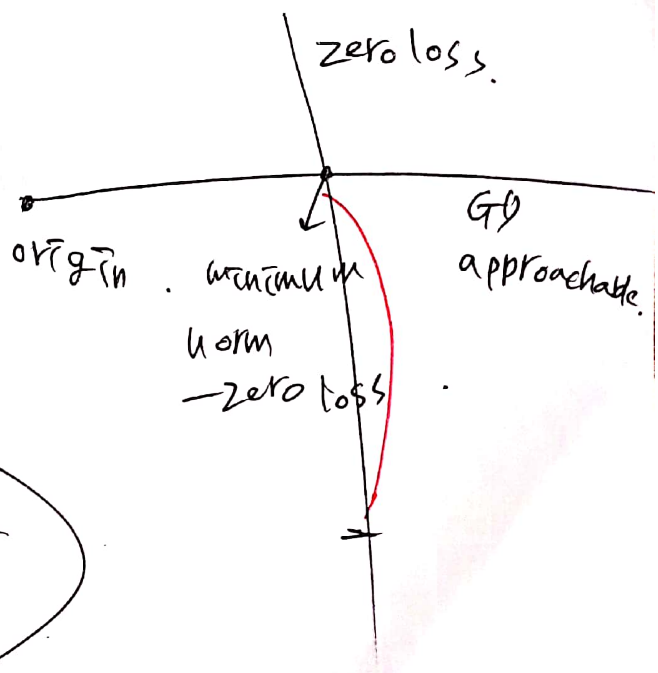
$\theta^{(0)} = 0$ 이면 $\theta^k = ?$

⟨2⟩.

$\theta^T \cdot x_i - y_i = 0$ 이니까.

↓

orthogonal to the set



Missing Links

GD - found solution

Big model, optimizer better.

$\hat{y} = w \cdot x$

「weight」

DLT(deep learning theory)

big model benefit

↳ 성능이 좋은 거

big model이 어떻게 예측?

「제한 조건 하에서」

Most $$$ => ??

momentum => DL

=?

이론학자 — 인공지능 방법론자

vice versa

PE days, non-embedding

dataset size - tokens

non-embedding

HW - 답은 일정

Introducing whisper 「9/22」

↳ micro Machine Man presenting the most midget miniature motorcade of Micro Machines.

「automatic speech recognition (ASR) system」.

「Neural Scaling laws」

PLMS team matching starts ~12 PM today. (11/29, 12/1 : No class).

How the
①
②
③

<1>

A belief ~~(that~~ 'Scaling up' is the ultimate answer.

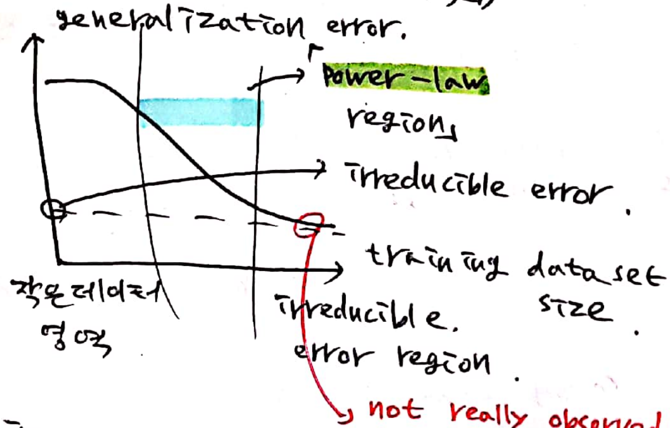
GPT-3 => 미래에 도움 (발전에)
why? ∞이 아닌 한정적인 파라미터로 train 하고 최적화 가능하니까.

「give it the compute, give it the data and.」

DL scaling is predictable, empirically.

L vs D

방법
① 큰 dataset 선택
② SOTA model 선택. ex) batch size, learning rate
③ draw '$z_1, z_2, \ldots$' fraction of the dataset. (different size)
④ train modes of various sizes to find the one with smallest test loss. (hyperparameter 최적화)



향상된 아키텍쳐, 파라미터는 exponent를 성능 향상 시키지 X.

$$\text{gen. error} \le O\left(\frac{\text{complex}}{\sqrt{D}}\right) \leftarrow \beta_g = \frac{-1}{2}$$

$$\le O\left(\frac{\text{complex}}{D}\right)$$

$$\uparrow \beta_g = -1$$

$$\text{Test loss} \simeq \alpha_g \cdot D^{\beta_g} \quad 「\beta_g \in [-\tfrac{1}{2}, 0)」$$

$$\varepsilon(m) = \alpha \cdot m^{\beta_g} + \gamma$$

CamScanner로 스캔하기

paper 「RNNs, LSTMs」

① 디버깅 = 너의 모델이 power law와 far away하면, maybe you did something wrong?

② challenge = slope 바꾸는게 가능?

「maybe a better way to evaluate model, algorithms」

③ exploration = 각 데이터로 더 큰 정부에 대한 적절한 모델 선택.

④ compute goal = target loss 설정, 우리는 요구된 계산의 추정치를 얻을수 있다.

「limitation = could have been explicit」

per
sett
↳ p
Ne

「Scaling Laws for Neural Language models」

non-embedding, tokens, PF-days.

모델 크기, dataset의 크기 ↑ ⇒ 언어 모델의 성능은 smoothly 향상.

In finite compute: 모델 size ↑, dataset 상승 should be sublinear: $D \propto N^{\frac{0.076}{0.095}} \sim N^{0.74}$

comput budget C 「fixed」

$N \propto C^{0.73}$, $B \propto C^{0.24}$, $S \propto C^{0.03}$

↳ optimal model size

$D = B \cdot S = C^{0.27}$

↳ dataset size

model size increase much faster than dataset size?

observation - model의 looks 「depth vs width」

but the model architecture does shift the plot.

LSTM vs transformer

↳ language model, computer vision

「Scaling 이슈 = depth vs width」

↳ CNN으로는 scaling 하기 ⊕⊕

① 비전에서의 scaling을 쓰는 이유.

$N_{opt}(C)$, $D_{opt}(C)$

$= \underset{N, D \text{ s.t. } FLOPs(N,D)=C}{\arg\min} L(N, D)$

「model size & training tokens 의 수 = tradeoff」

「Chinchilla, Gopher, GPT-3, Megatron-Turing NLG」

CamScanner로 스캔하기

ICLR = scale efficiently

Insights

Pareto frontier

perceptron in teacher-student setting.

↳ punchline = select data wisely, then not really.

Scaling는 모델 shape에 의존X, 오직 pre-training에 hold. 「not fine-tuning」.

$\boxed{\text{data pruning}}$ = Beyond neural scaling laws 「Beating power law via data pruning」.

Beating power law

Neural[II]

↳ pessimistic message = neural scaling Law.

Core set.

「2%의 정확도 ↑를 위해 dataset size x10이 필요」 → computer vision.