from mutual information to generalization

Speaker: Yi-Shan Wu

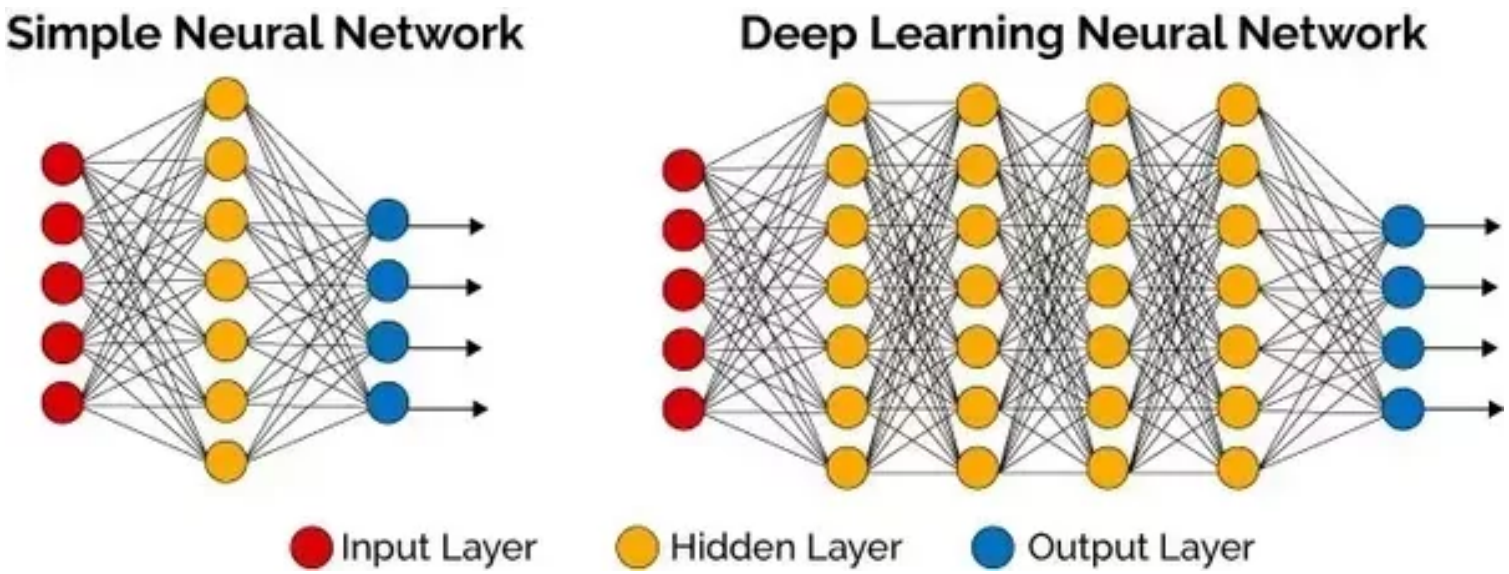
Institute of Information Science

Academia Sinica

Taiwan

Dec 07, 2018

Why DNN don't overfit ?



Low training error $\Rightarrow$ Low testing error ?

Generalization bound

Previous theory :

VC-dimension, Rademacher complexity, norms

$\Rightarrow$ Measurements of network capacity

$$capacity \approx \sqrt{\frac{\ln |H|}{m}}$$

ex. a function class H is a set of q parameters each of which can have at most r discrete values.

$$\Rightarrow |H| = r^q$$

They are all not good enough since deep neural networks are usually "over-parametrized".

Some recent possible approaches

1. network compression [Arora et, al. 2018]
   A well-trained network is more robust to noise than random initialized networks. Thus, it is possible to be compressed into a smaller network which has better generalization bound.
2. mutual information
   - $I(S, A(S))$: mutual information between training samples and hypothesis class
   - $I(X, \hat{X})$: mutual information between input data and its representation

Compression approach

Motivation : The analysis for previous works are too pessimistic. If the large trained network can be compressed to a smaller network without reducing performance, then it is able to obtain better generalization bound by previous approaches.

example

f : network with 100 parameters.
G_{40} : a class of networks with 40 parameters.
If there is a $g_{40} \in G_{40}$ with $L_S(g_{40}) \leq L_S(f)$, then we have w.h.p.

$$L(g_{40}) \leq L_S(f) + \sqrt{\frac{40 \ln r}{m}}$$

compression algorithm

generalization bound for g_A

If the trained classifier f can be compressed to $G_{A,s}$ with helper string s, then there exists $g_A \in G_{A,s}$ with high probability over the training set,

$$L(g_A) \leq L_S(f) + O(\sqrt{\frac{q \ln r}{m}})$$

- the size of $G_{A,s}$ is decided by the property of f (more stable $\Rightarrow$ less parameters)
- string s is fixed before seeing data ($\because$ one s gives one kind of structure $G_{A,s}$)
- it remains to prove that the network after compression satisfies $L_S(g_A) \leq L_S(f)$ (Lemma 3.)

conti

In compression approach, the possible hypotheses we have to consider (G_A) are much less than the original one (f). This give us a rule to train our network as well as keep generalization gap small :

- fix a help string s
- train with an over-parametrized network f
- measure the noise stability parameter
- use the compression algorithm to get a smaller network g_A

Ideal learner

We know that if the learner's action depends too much on the training samples, it might overfit. Therefore, an ideal learner should not depends highly on any single sample.
There are several ways to formulate this idea. For example,

- stability
- mutual information

Stability

An algorithm $\mathcal{A} : \mathcal{X}^m \to \mathcal{H}$ is said to be ϵ-stable if $\forall S, S' \in \mathcal{X}^m$ which differ in only one sample and $\forall i \in [m]$, we have

$$|\ell(\mathcal{A}(S), s_i) - \ell(\mathcal{A}(S'), s_i)| \leq \epsilon$$

That is, the loss of algorithm's output (a hypothesis) doesn't differ a lot when testing on any training sample.

generalization bound

If $\mathcal{A}$ is ϵ-stable algorithm , then

$$|\mathbb{E}_{S,A}[L(A(S)) - L_S(A(S))]| \leq \epsilon$$

Stability

With some assumptions, it is now possible to show that the often used algorithm : stochastic gradient descent (SGD) is about $O(\frac{L^2 T}{m})$-stable, where L is the Lipschitz constant.

Démonstration.

$$\text{update rule : } w_{t+1} = w_t - \alpha_t \nabla \ell(w_t, z_i)$$

Suppose at time t, we have $\|w_t(S) - w_t(S')\| = \delta_t$.
When going to the next iteration, there is only probability $\frac{1}{n}$ that the algorithm selects the index of an example on which is different in S and S'. □

conti

Démonstration.

Therefore, by careful select some other parameters, we have

$$\mathbb{E}[\delta_{t+1}] \leq \frac{m-1}{m}\mathbb{E}[\delta_t] + \frac{1}{m}(\mathbb{E}[\delta_t] + 2\alpha_t L)$$

Unraveling the recursion gives

$$\mathbb{E}[\delta_T] \leq \frac{2L}{m}\sum_{t=1}^{T}\alpha_t$$

Thus, the difference of loss values can be bounded by $\frac{2L^2}{m}\sum_{t=1}^{T}\alpha_t$ □

conti

Therefore, we have that

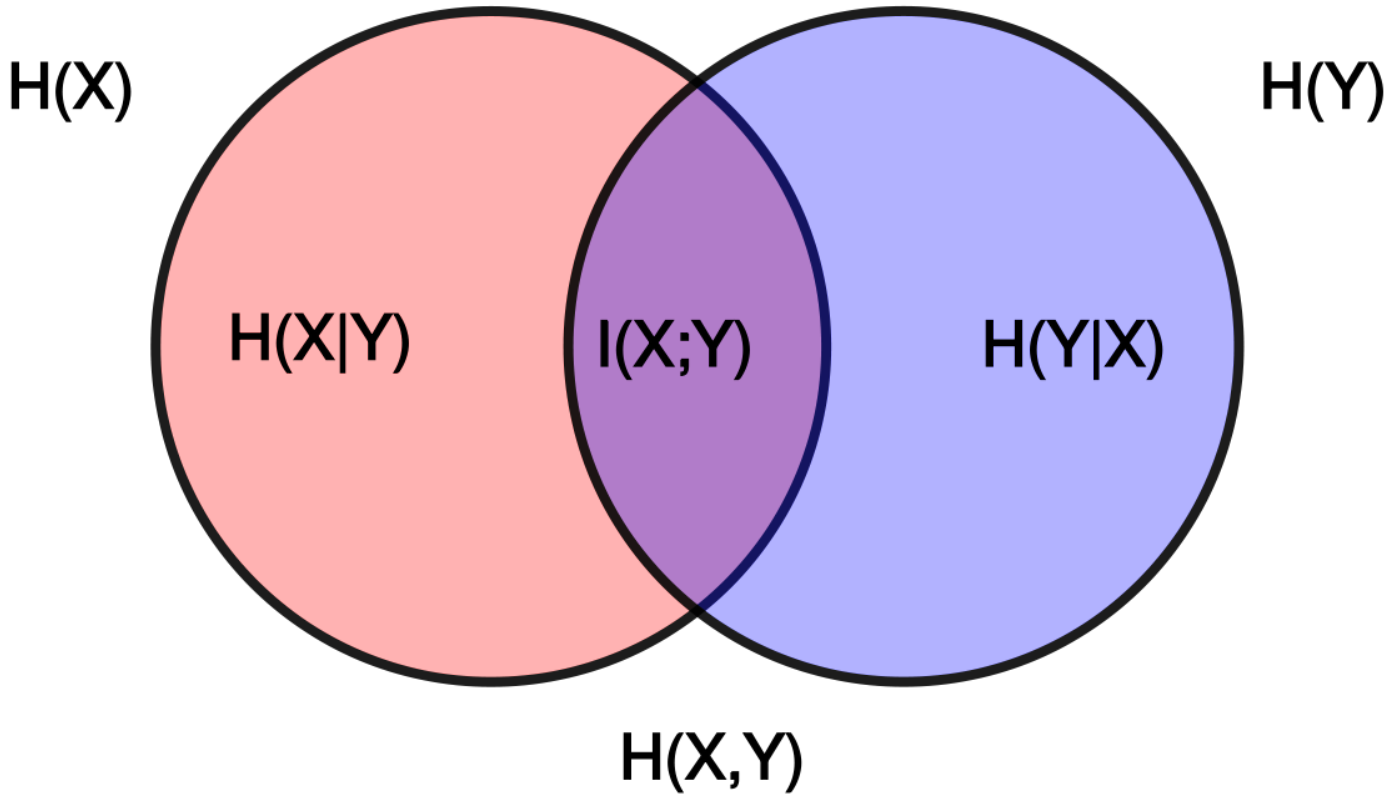
$$\epsilon \approx \frac{2L^2}{m} \sum_{t=1}^{T} \alpha_t$$

If we want the generalization bound to be tight, then

1. collect more training examples
2. don't train too long

Mutual Information $I(S; A(S))$

Mutual information is another way to capture how much an algorithm depends on training samples.



d-bit information learner

We say that $\mathcal{A}$ has mutual information of at most d bits (for sample size m) with respect to distribution $\mathcal{D}$ if

$$I(S; A(S)) \leq d$$

where $S \sim \mathcal{D}^m$.

Definition

A learning algorithm $\mathcal{A}$ for $\mathcal{H}$ is called a d-bit information learner if

$$I(S; A(S)) \leq d$$

with respect to every realizable distribution $\mathcal{D}$.

Generalization gap

It is not hard to believe that an algorithm $\mathcal{A}$ with bounded mutual information with respect to distribution $\mathcal{D}$ implies generalization.
An extreme example : A learner with constant output is a 0-bit information learner.

Theorem

Let $\mathcal{A}$ has mutual information of at most d bits with respect to distribution $\mathcal{D}$ and let $S \sim \mathcal{D}^m$. Then for every $\epsilon > 0$,

$$\mathbb{P}_{A,S}[|L_S(A(S)) - L(A(S))| > \epsilon] < \frac{d+1}{2m\epsilon^2 + 1}$$

proof idea

The theory can be proved by the idea similar to the use of pre-selected help string s in compression approach.

- Recall that one s gives one kind of network structure $G_{A,s}$. If the string is chosen after training, then we would have to union over several kinds of structures $\{G_{A,s}\}_s$.
- Therefore, if we can show that for an average s, the algorithm outputs only a small number of hypotheses, then we get better generalization bound.

proof idea

Lemma

Let μ be a probability distribution over $\mathcal{S} \times \mathcal{A}(\mathcal{S})$. Let $(S, \mathcal{A}(\mathcal{S}))$ be chosen according to μ. Then there exists a random variable Z such that :

1. *Z is independent of $\mathcal{S}$.*
2. *$\mathcal{A}(\mathcal{S})$ is a deterministic function of $(\mathcal{S}, Z)$*
3. *$H(\mathcal{A}(\mathcal{S})|Z) \leq I(\mathcal{S}, \mathcal{A}(\mathcal{S})) + \log(I(\mathcal{S}, \mathcal{A}(\mathcal{S})) + 1)$*

Think of Z as the help string s. The lemma says that there is a way to sample Z before seeing $\mathcal{S}$, in a way that still preserves functionality and so that for an average Z, the randomized algorithm outputs only a small number of $\mathcal{H}$'s.
(3) $\approx$ (for an average Z) $G_{A,Z}$ is bounded by $I(\mathcal{S}, \mathcal{A}(\mathcal{S})) + \cdots$

Summary

In general, if we want to get low testing error $L(A(S))$

- low generalization error $|L(A(S)) - L_S(A(S))|$
  bounded mutual information, algorithmic stable
- low training error $L_S(A(S))$
  news : SGD can reach global minima even in non-convex case (2 paper published in parallel last month)

$\Rightarrow$ trade-off between them

Open problem :

- Is mutual information really a good measure ? How to measure mutual information ?
- Why SGD is a good algorithm for getting low testing error ?
- What's the basic rule for generalization guarantee ?