Open Theoretical Questions in Deep Learning

ディープラーニングにおける理論的未解決問題について

Masato Taki  RIKEN, iTHES physics

2016.7/8
OIST workshop
Deep learning

‘computer model’ of (a part of) our real neural system

Surprising performance

[Google’s team (Dean, Ng, et al), 2012]

10 million images from youtube

16000 processor array

(recolutional net + autoencoder)

recognize many objects

©[Dean, et al, 2012]
Surprising performance

[Google’s team (Dean, Ng, et al), 2012]

Very high classification performance

Surprising performance

[Google’s team (Dean, Ng, et al), 2012]

Very high classification performance

©https://www.pexels.com/search/dog/
Surprising performance

reconstructing ‘pre-image’ of corresponding output signal, we can see the system acquired “abstract concept” (deep feature)

Google’s cat

No understanding why & how it works so well
1. Machine Learning
Aim of machine learning (supervised case)

‘observed’ data

explaining how they generated

classifying into categories

extracting essences of them

etc
Aim of machine learning (supervised case)

‘observed’ data

explaining how they generated

classifying into categories

extracting essences of them

etc

a kind of fitting of ‘observed’ data
Aim of machine learning ((un)supervised case)

observed data points

a kind of fitting of ‘observed’ data
Aim of machine learning ((un)supervised case)

underfitting (less parameters)

poor performance
Aim of machine learning ((un)supervised case)
nice fitting! (it also guess un-observed data)

it makes reliable prediction
Aim of machine learning ((un)supervised case)

overfitting (too many parameters, poor data)

no reliable prediction
Aim of machine learning ((un)supervised case)

it also explaining *yet-unobserved/another* data

realizing nice fitting for huge data and huge fitting system

→ *generalization* (汎化性能), *good performance*
2. Deep Learning
neural network
neural network

$\overrightarrow{x} \rightarrow \underbrace{\begin{array}{c}
\ell - 1 \\
\ell \\
i \\
j \\
\end{array}}_{W_{i,j}^{(\ell)}} \rightarrow \overrightarrow{y}$

weight parameter
neural network
neural network

\[ \bar{y} = f(W^{(L)} f(W^{(L-1)} f(W^{(L-2)} \cdots \cdots f(W^{(2)} \bar{x}) \cdots )) \]

\[ f \text{ : a nonlinear function (activation)} \]
A neural network is a computational model that is used to learn and make predictions. It consists of layers of interconnected nodes, or neurons, that process information. The network is trained to map inputs to outputs, and this process is referred to as "tuning" or "training." The mathematical representation of a neural network is given by:

\[ \tilde{y} = f(W^{(L)} f(W^{(L-1)} f(W^{(L-2)} \ldots f(W^{(2)} \bar{x}) \ldots )) \]

The function \( f \) represents the activation function, and \( W \) represents the weight matrices of the network. The process of training involves adjusting the weights \( W \) to minimize the difference between the predicted output \( \tilde{y} \) and the actual output, thereby improving the network's ability to explain the data.
training in neural network

$\vec{x} \rightarrow W^{(2)} \rightarrow \cdots \rightarrow W^{(L-1)} \rightarrow W^{(L)} \rightarrow \vec{y}$

$\vec{a}$

answer
training

\[ \vec{x} \rightarrow W^{(2)} \rightarrow \cdots \rightarrow W^{(L-1)} \rightarrow W^{(L)} \rightarrow \vec{y} \]

\[ 1 \quad 2 \quad \cdots \quad L - 1 \quad L \]

\[ \vec{a} \rightarrow \text{teacher's answer} \]

\[ \text{comparing them for training data} \]

\[ \text{student's answer} \]
training

error function

\[ E(W) = \frac{1}{2} \sum_{n} (\hat{y}_n(W) - \hat{a}_n)^2 \]
training

\[ E(W) = \frac{1}{2} \sum_n (\tilde{y}_n(W) - \tilde{a}_n)^2 \]
training

\[ \vec{x}_n \rightarrow W^{(2)} \rightarrow \ldots \rightarrow W^{(L-1)} \rightarrow W^{(L)} \rightarrow \vec{y}_n \]

10 layers

100x100 pixels

W’s are (100x100)x(100x100)x10=10^9 parameters
minimize the error in $10^9$ dimensional parameter space

$$E(W) = \frac{1}{2} \sum_n (\tilde{y}_n(W) - \tilde{a}_n)^2$$
deep learning is *ad hoc* business, so there are massive mysteries for theorists
too many local minima!
For usual ML system, they show poor performance.
landscape of minima

It's like looking for a needle in the haystack.

too many local minima!
For usual ML system, they show poor performance.
mystery of deep learning (observation)

For deep learning with huge parameters,

all local minimum show almost the same nice performance (generalization)

we shouldn’t seek global minimum. It’s just overfitting.
mystery of deep learning (observation)

For deep learning with huge parameters,

all local minimum show almost the same nice performance (generalization)

we shouldn’t seek global minimum. It’s just overfitting.

this makes deep learning outstanding system
another mystery
deep neural network

\[ \vec{x} \rightarrow \begin{array}{c} W^{(2)} \\ \vdots \\ W^{(L-1)} \end{array} \rightarrow W^{(L)} \rightarrow \vec{y} \]

eg. \((100 \times 100) \times (100 \times 100) \times 10 = 10^9\) parameters
deep neural network

eg. \((100 \times 100) \times (100 \times 100) \times 10 = 10^9\) parameters

naive fitting leads to overfitting
deep neural network

deg. (100x100)x(100x100)x10=10^9 parameters

naive fitting leads to overfitting

regularization of the network system
drop out method

train by data $\vec{\mathcal{X}}_1$ train by data $\vec{\mathcal{X}}_2$

avoiding overfitting & realizing generalization
drop connect method

\[
\vec{\mathcal{X}}_1 \quad \vec{\mathcal{X}}_2 \quad \ldots
\]
3. Theoretical trial for understanding deep learning
there exist too many theoretical open questions, but some theoretical approach to them begins to appear
why drop out works
drop out

\[ \bar{y} = f(W^{(L)} f(W^{(L-1)} f(W^{(L-2)} \ldots \ldots f(W^{(2)} \bar{x}) \ldots)) \]

it reduces components randomly
\[ \tilde{y} = f(W^{(L)} \tilde{B}^{(L-1)}) \circ f(W^{(L-1)} \tilde{B}^{(L-2)}) \circ f(W^{(L-2)} \cdots \tilde{B}^{(2)} \circ f(W^{(2)} \tilde{x}) \cdots) \]

acting random Bernoulli variables element-wisely

\[ B_i = 0, 1 \]
\[ \tilde{B} \circ \tilde{v} = \begin{pmatrix} B_1 v_1 \\ B_2 v_2 \\ \vdots \\ B_n v_n \end{pmatrix} \]
drop out
[Srivastava-Hinton-Krizhevsky-Sutskever-Salakhutdinov, 2014] etc

\[ \bar{y} = f(W^{(L)} \bar{B}^{(L-1)} \odot f(W^{(L-1)} \bar{B}^{(L-2)} \odot f(W^{(L-2)} \ldots \bar{B}^{(2)} \odot f(W^{(2)} \bar{x}) \ldots)) \]

acting random Bernoulli variables element-wisely

Only for **linear single layer** system, we can show

taking expectation value under Bernoulli distribution

\[ \rightarrow \] modification of error function

\[ \rightarrow \] improvement of optimization
How deep learning avoid ‘local minima problem’
deep learning as a spin glass

[Choromanska-Henaff-Mathieu-Arous-LeCun, 2015]
deep learning as a spin glass

[Choromanska-Henaff-Mathieu-Arous-LeCun, 2015]

error function of deep neural network

the Hamiltonian of `spin glass’

many assumptions and approximation

\[ E(W) = H = \sum_{i_1, i_2, \ldots, i_L} J_{i_1 i_2 \cdots i_L} W_{i_1} W_{i_2} \cdots W_{i_L} \]

\[ \sum_i W_i^2 = \text{const.} \quad \left( \text{not} \quad W_i = \pm \frac{1}{2} \right) \]
Deep learning as a spin glass

[Choromanska-Henaff-Mathieu-Arous-LeCun, 2015]

Error function of deep neural network

The Hamiltonian of spin glass

Many assumptions and approximation

Uniformity $\rightarrow$ Weight sharing (typical regularization)

$W^{(\ell)}_{i,j} \rightarrow$ Repeating smaller numbers of parameters
deep learning as a spin glass

[Choromanska-Henaff-Mathieu-Arous-LeCun, 2015]

spherical spin glass

\[ E(W) = H = \sum_{i_1, i_2, \ldots, i_L} J_{i_1 i_2 \ldots i_L} W_{i_1} W_{i_2} \cdots W_{i_L} \]

distribution of its minima is already known by using random matrix theory (GOE) in “large system limit”

Almost all local minima are degenerated. They are near to ground state.