Understanding Deep Learning in Theory

Hui Jiang

Department of Electrical Engineering and Computer Science
Lassonde School of Engineering, York University, Canada

Vector Institue Seminar on April 5, 2019

Outline

Deep Learning Overview

- Deep learning has achieved tremendous successes in practice: speech, vision, text, games, $\cdots$
- Deep learning is somehow criticized because it can not be explained by the current machine learning theory.

Open Problems

- What is the essense to successes of neural nets?
- Why neural nets overtake other ML models in practice?
- Why so "easy" to learn neural nets?
- Why do neural nets generalize well?
- What is the limit of neural nets?

Deep Learning Theory

Theoretical Issues of Neural Nets

1. **expressiveness**: what is the modeling capacity of neural nets?
   - **solved** due to the universal approximation [Cyb89]
2. **optimization**: why simple gradient descents consistently work?
   - NP-hard to learn even a small neural net [BR92]
   - high-dimensional & non-convex to learn large neural nets [AHW95]
3. **generalization**: why over-parameterized neural nets generalize?
   - VC theory gives loose bounds to simple models [Vap00]
   - totally fails to explain complex models like neural nets

Problem Formulation

Machine Learning as Stochastic Function Fitting

- Inputs $\mathbf{x} \in \mathbb{R}^K$ follow a p.d.f. $p(\mathbf{x})$: $\mathbf{x} \sim p(\mathbf{x})$
- A **target function** $y = \bar{f}(\mathbf{x})$: deterministic from input $\mathbf{x} \in \mathbb{R}^K$ to output $y \in \mathbb{R}$
- Finite training samples: $\mathcal{D}_T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_T, y_T)\}$, where $\mathbf{x}_t \sim p(\mathbf{x})$ and $y_t = \bar{f}(\mathbf{x}_t)$ for all $1 \leq t \leq T$
- A model $y = f(\mathbf{x}|\mathcal{D}_T)$ is learned from function class $\mathbb{C}$ based on $\mathcal{D}_T$ to minimize a loss measure $l(\bar{f}, f)$ between $\bar{f}$ and f w.r.t. $p(\mathbf{x})$
- $l(\cdot)$ is normally **convex**: mean square error, cross-entropy, hinge, ...

Problem Formulation (cont'd)

- Ideally $f()$ should be learned to minimize the expected risk:

Expected Risk

$$R(f \mid \mathcal{D}_T) = \mathbf{E}_{p(\mathbf{x})}\left[l\left(\bar{f}(\mathbf{x}), f(\mathbf{x}|\mathcal{D}_T)\right) \right]$$

- Practically $f()$ is learned to minimize the empirical loss:

Empirical Risk

$$R_{emp}(f \mid \mathcal{D}_T) = \frac{1}{T}\sum_{t=1}^{T} l\left(y_t, f(\mathbf{x}_t|\mathcal{D}_T)\right)$$

- Function class $\mathbb{C} = L^1(\mathbb{U}_K)$, where $\mathbb{U}_K \triangleq [0,1]^K \subset \mathbb{R}^K$

$$f \in L^1(\mathbb{U}_K) \iff \int \cdots \int_{\mathbf{x}\in\mathbb{U}_K} |f(\mathbf{x})|\, d\mathbf{x} < \infty$$

Learning as Functional Minimization

- Learning is formulated as model-free functional minimization in $L^1(\mathbb{U}_K)$:

$$f^* = \arg \min_{f \in L^1(\mathbb{U}_K)} Q(f|\mathcal{D}_T) = \arg \min_{f \in L^1(\mathbb{U}_K)} \sum_{t=1}^{T} l\left(y_t, f(\mathbf{x}_t)\right)$$

Functional minimization is very generic. But

how to parameterize the function space $L^1(\mathbb{U}_K)$ at above?

Literal Model Space: Neural Networks

Literal model space

Use neural networks to represent the function space $L^1(\mathbb{U}_K)$

- **Literal space** Λ_M: the set of all well-structured neural nets of M weights; each neural net is denoted as $\mathbf{w} \in \mathbb{R}^M$
- **Universal approximator theorem [Cyb89]**: $\forall f(\mathbf{x}) \in L^1(\mathbb{U}_K)$ can be well approximated by at least one $\mathbf{w}$ in Λ_M.
- If inputs and all weights are bounded, $\forall \mathbf{w} \in \Lambda_M$ represents a function in $L^1(\mathbb{U}_K)$, denoted as $f_{\mathbf{w}}(\mathbf{x})$.

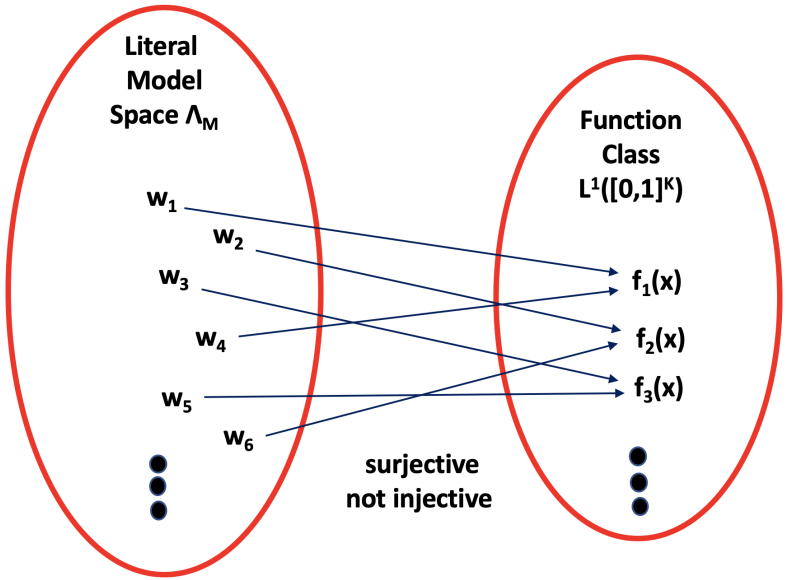
Lemma 1

If M is sufficiently large, $\lim_{M\to\infty} \Lambda_M \equiv L^1(\mathbb{U}_K)$.

Learning in Literal Space

Learning neural nets in literal space

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbb{R}^M} Q(f_{\mathbf{w}}|\mathcal{D}_T) = \arg\min_{\mathbf{w}\in\mathbb{R}^M} \sum_{t=1}^{T} l\left(y_t, f_{\mathbf{w}}(\mathbf{x}_t)\right)$$



Canonical Model Space: Fourier Series

- **Fourier coefficients**: $\forall f(\mathbf{x}) \in L^1(\mathbb{U}_K) \stackrel{\mathscr{F}}{\Longrightarrow} \boldsymbol{\theta} = \{\theta_{\boldsymbol{k}} | \boldsymbol{k} \in \mathbb{Z}^K\}$

$$\theta_{\boldsymbol{k}} = \int \cdots \int_{\mathbf{x} \in \mathbb{U}_K} f(\mathbf{x})\, e^{-2\pi i \boldsymbol{k} \cdot \mathbf{x}} d\mathbf{x} \; (\forall \boldsymbol{k} \in \mathbb{Z}^K)$$

- **Fourier series**: $f(\mathbf{x}) := \mathscr{F}^{-1}(\mathbf{x}|\boldsymbol{\theta})$

$$f(\mathbf{x}) = \sum_{\boldsymbol{k} \in \mathbb{Z}^K} \theta_{\boldsymbol{k}}\, e^{2\pi i \boldsymbol{k} \cdot \mathbf{x}}$$

- **Riemann-Lebesgue lemma**: $\forall \epsilon > 0$, truncate to N significant terms, $\mathcal{N}_\epsilon$, to form a partial sum of finite terms

$$\hat{f}(\mathbf{x}) = \sum_{\boldsymbol{k} \in \mathcal{N}_\epsilon} \theta_{\boldsymbol{k}}\, e^{2\pi i \boldsymbol{k} \cdot \mathbf{x}}$$

where $\int \cdots \int_{\mathbf{x} \in \mathbb{U}_K} \|f(\mathbf{x}) - \hat{f}(\mathbf{x})\|^2 d\mathbf{x} \leq \epsilon^2$.

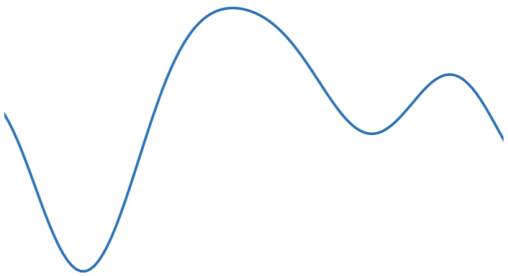
Illustration of Canonical Space of $L^1(\mathbb{U}_K)$

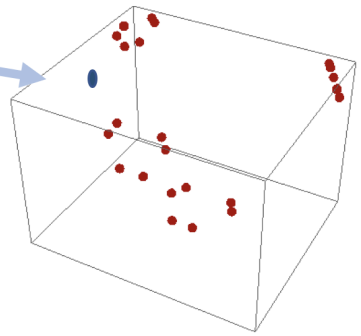
any $f(\mathbf{x}) \in L^1(\mathbb{U}_K)$

Fourier coefficients

Fourier series

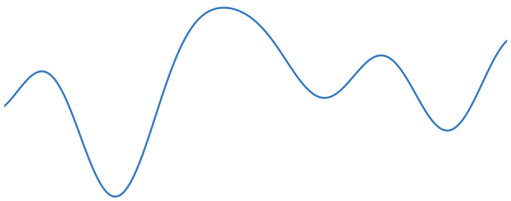
$\boldsymbol{\theta} = (\theta_{\mathbf{k_1}}, \theta_{\mathbf{k_2}}, \ldots) \in \boldsymbol{\Theta} \subseteq C^{\infty}$



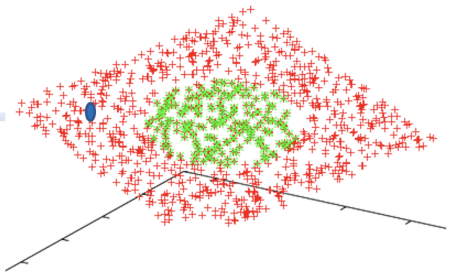


$\int \cdots \int_{\mathbf{x} \in \mathbb{U}_K} \|f(\mathbf{x}) - \hat{f}(\mathbf{x})\|^2 d\mathbf{x} \leq \epsilon^2$

truncate by ϵ



Fourier partial sum

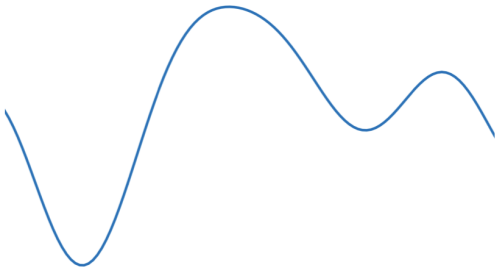


$\hat{f}(\mathbf{x}) \in L^1(\mathbb{U}_K)$

$\hat{\boldsymbol{\theta}} = (\theta_{\mathbf{k_1}}, \theta_{\mathbf{k_2}}, \ldots, \theta_{\mathbf{k_N}}) \in \hat{\boldsymbol{\Theta}} \subseteq C^N$

Illustration of Canonical Space of $L^1(\mathbb{U}_K)$

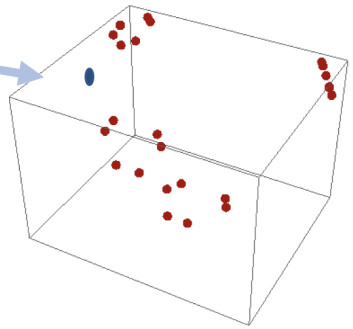
any $f(\mathbf{x}) \in L^1(\mathbb{U}_K)$



Fourier coefficients

Fourier series

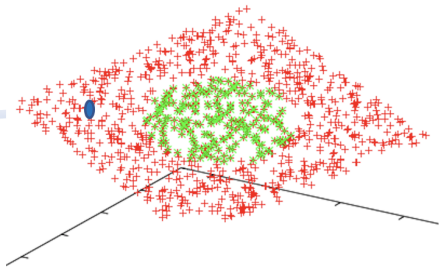
$\boldsymbol{\theta} = (\theta_{\mathbf{k_1}}, \theta_{\mathbf{k_2}}, \ldots) \in \boldsymbol{\Theta} \subseteq C^{\infty}$



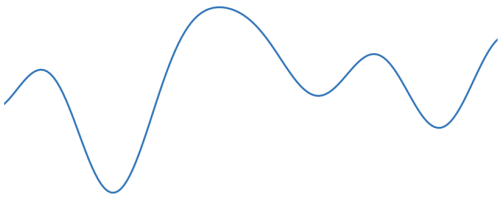
$\int \cdots \int_{\mathbf{x} \in \mathbb{U}_K} \|f(\mathbf{x}) - \hat{f}(\mathbf{x})\|^2 d\mathbf{x} \leq \epsilon^2$

3.14159265358979323846...

truncate by ϵ



Fourier partial sum

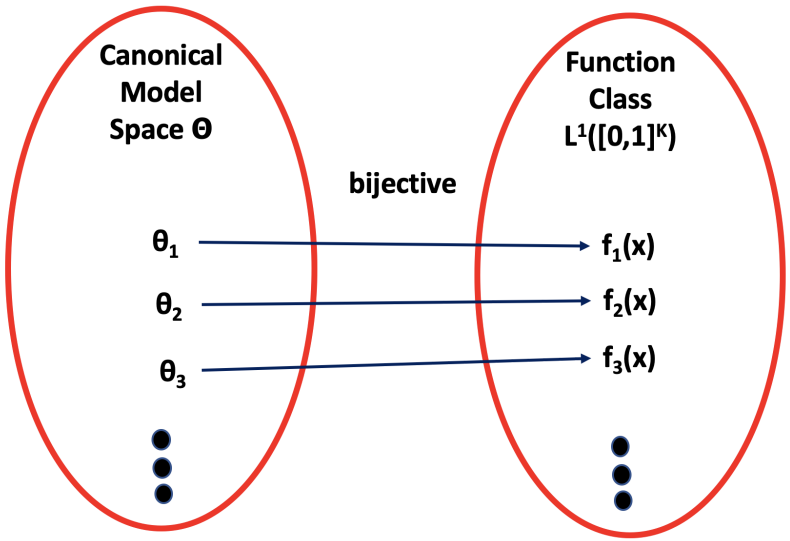


$\hat{f}(\mathbf{x}) \in L^1(\mathbb{U}_K)$

$\hat{\boldsymbol{\theta}} = (\theta_{\mathbf{k_1}}, \theta_{\mathbf{k_2}}, \ldots, \theta_{\mathbf{k_N}}) \in \hat{\boldsymbol{\Theta}} \subseteq C^N$

Canonical Space of $L^1(\mathbb{U}_K)$

- **Canonical space** Θ: each set of Fourier coefficients $\theta \in \Theta$



Learning in canonical space

$$\theta^* = \arg\min_{\theta\in\Theta} Q(\theta|\mathcal{D}_T) = \arg\min_{\theta\in\Theta} \sum_{t=1}^{T} l\left(y_t, \mathscr{F}^{-1}(\mathbf{x}_t \,|\, \theta)\right)$$

Learning in Canonical Space

Theorem 2

The objective function $Q(f|\mathcal{D}_T)$ is **convex** *in* **canonical space**. *If the dimensionality of canonical space is not less than the number of training samples in $\mathcal{D}_T$, the global minimum achieves* **zero** *loss.*

Proof sketch:

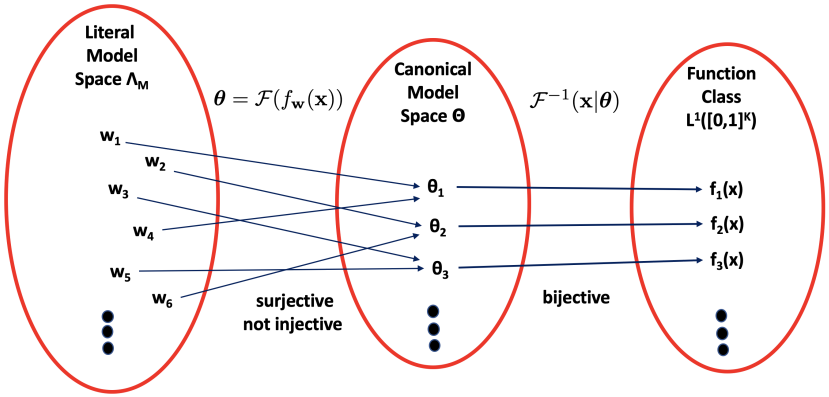
- $Q(f|\mathcal{D}_T)$ is represented in canonical space:

$$Q(\boldsymbol{\theta}|\mathcal{D}_T) = \sum_{t=1}^{T} l\left(y_t, \mathscr{F}^{-1}(\mathbf{x}_t \,|\, \boldsymbol{\theta})\right) = \sum_{t=1}^{T} l\left(y_t, \sum_{\boldsymbol{k}\in\mathcal{N}_\epsilon} \theta_{\boldsymbol{k}} \cdot e^{2\pi i \boldsymbol{k}\cdot\mathbf{x}_t}\right)$$

- N unknown coefficients: $\boldsymbol{\theta} = \{\theta_{\boldsymbol{k}} \mid \boldsymbol{k} \in \mathcal{N}_\epsilon\}$
- T linear eqns: $y_t = \hat{f}(\mathbf{x}_t) = \sum_{\boldsymbol{k}\in\mathcal{N}_\epsilon} \theta_{\boldsymbol{k}} \cdot e^{2\pi i \boldsymbol{k}\cdot\mathbf{x}_t} \quad (1 \le t \le T)$

From Canonical Space back to Literal Space

- Model spaces: $\mathbf{w} \Rightarrow f_{\mathbf{w}}(\mathbf{x}) \overset{\mathscr{F}}{\Longrightarrow} \boldsymbol{\theta} \Longrightarrow \mathscr{F}^{-1}(\mathbf{x}|\boldsymbol{\theta}) := f_{\boldsymbol{\theta}}(\mathbf{x})$



- The objective function in two spaces:

$$Q(f_{\mathbf{w}}|\mathcal{D}_T) = Q(f_{\boldsymbol{\theta}}|\mathcal{D}_T)$$

- The chain rule:

$$\nabla_{\mathbf{w}} Q(f_{\mathbf{w}}|\mathcal{D}_T) = \nabla_{\boldsymbol{\theta}} Q(f_{\boldsymbol{\theta}}|\mathcal{D}_T) \nabla_{\mathbf{w}} \boldsymbol{\theta} = \nabla_{\boldsymbol{\theta}} Q(f_{\boldsymbol{\theta}}|\mathcal{D}_T)\, \nabla_{\mathbf{w}} \mathscr{F}(f_{\mathbf{w}}(\mathbf{x}))$$

From Canonical Space back to Literal Space (cont'd)

Disparity Matrix

$$\begin{bmatrix} \frac{\partial Q}{\partial w_1} \\ \vdots \\ \frac{\partial Q}{\partial w_m} \\ \vdots \\ \frac{\partial Q}{\partial w_M} \end{bmatrix}_{M\times 1} = \underbrace{\begin{bmatrix} \mathscr{F}\left(\frac{\partial f_{\mathbf{w}}(\mathbf{x})}{\partial w_1}\right) \\ \vdots \\ \mathscr{F}\left(\frac{\partial f_{\mathbf{w}}(\mathbf{x})}{\partial w_m}\right) \\ \vdots \\ \mathscr{F}\left(\frac{\partial f_{\mathbf{w}}(\mathbf{x})}{\partial w_M}\right) \end{bmatrix}_{M\times N}}_{\text{disparity matrix } \mathbf{H}(\mathbf{w})} \begin{bmatrix} \frac{\partial Q}{\partial \boldsymbol{\theta}_1} \\ \vdots \\ \frac{\partial Q}{\partial \boldsymbol{\theta}_n} \\ \vdots \\ \frac{\partial Q}{\partial \boldsymbol{\theta}_N} \end{bmatrix}_{N\times 1}$$

Gradients in literal and canonical spaces are related via a **pointwise linear transformation**: $\left[\nabla_{\mathbf{w}} Q\right]_{M\times 1} = \left[\mathbf{H}(\mathbf{w})\right]_{M\times N}\left[\nabla_{\boldsymbol{\theta}} Q\right]_{N\times 1}$

From Canonical Space back to Literal Space (cont'd)

Lemma 3

Assume neural network is sufficiently large ($M \geq N$). If $\mathbf{w}^*$ is a stationary point of $Q(f_{\mathbf{w}})$ and $\mathbf{H}(\mathbf{w}^*)$ has full rank, $\mathbf{w}^*$ is a global minimum.

Lemma 4

If $\mathbf{w}^{(0)}$ is a stationary point of $Q(f_{\mathbf{w}})$ and $\mathbf{H}(\mathbf{w}^{(0)})$ does not has full rank at $\mathbf{w}^{(0)}$, then $\mathbf{w}^{(0)}$ may be a local minimum or saddle point or global minimum.

Why learning of large-scale neural networks behaves like convex optimization?

Stochastic Gradient Descent (SGD)

randomly initialize $\mathbf{w}^{(0)}$, set $k = 0$
for *epoch* $= 1$ **to** L **do**
  for each *minibatch* **in** training set $\mathcal{D}_T$ **do**
    $\mathbf{w}^{(k+1)} \leftarrow \mathbf{w}^{(k)} - h_k \nabla_{\mathbf{w}} Q(\mathbf{w}^{(k)})$
    $k \leftarrow k + 1$
  end for
end for

Theorem 5

If $\mathbf{M} \geq \mathbf{N}$, *and initial* $\mathbf{w}^{(0)}$ *and step sizes* h_k *are chosen as such to ensure* $\mathbf{H}(\mathbf{w})$ *maintains* **full** *rank at every* $\mathbf{w}^{(k)}$, *then SGD/GD surely converges to a global minimum of zero loss like* **convex** *optimization.*

Why learning of large-scale neural networks behaves like convex optimization? (cont'd)

When $\mathbf{H}(\mathbf{w})$ degenerates?

- **dead** neurons: zero rows
- **duplicated** neurons: linearly dependent rows
- if $M \gg N$, $\mathbf{H}(\mathbf{w})$ becomes singular only after at least $M - N$ neurons are dead or duplicated.

Corollary 6

If an **over-parameterized** *neural network is* **randomly initialized**, *SGD/GD converges to a global minimum of zero loss in probability.*

Why over-parametered neural networks not overfitting?

Current machine learning theory

- VC generalization bound for classification [Vap00]:

$$R(f_{\mathbf{w}} \mid \mathcal{D}_T) \leq R_{emp}(f_{\mathbf{w}} \mid \mathcal{D}_T) + \sqrt{\frac{8\mathbf{d_M}(\ln \frac{2T}{\mathbf{d_M}} + 1) + 8\ln(\frac{4}{\delta})}{T}}$$

- Error bound for NNs [Bar94]: approx + estimation errors

$$R(f_{\mathbf{w}} \mid \mathcal{D}_T) \leq O\left(\frac{C_f^2}{M}\right) + O\left(\frac{\mathbf{M} \cdot K}{T}\log(T)\right)$$

Presumably over-parameterized models will fail due to overfitting:

When $M \to \infty$, $R_{emp}(f_{\mathbf{w}} \mid \mathcal{D}_T) = 0$, but $R(f_{\mathbf{w}} \mid \mathcal{D}_T)$ will diverge.

Learning Problem Formulation (review)

- Inputs $\mathbf{x} \in \mathbb{R}^K$ follow p.d.f. $p(\mathbf{x})$: $\mathbf{x} \sim p(\mathbf{x})$
- A **target function** $y = \bar{f}(\mathbf{x})$: from input $\mathbf{x} \in \mathbb{R}^K$ to output $y \in \mathbb{R}$
- T training samples: $\mathcal{D}_T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_T, y_T)\}$, where $\mathbf{x}_t \sim p(\mathbf{x})$ and $y_t = \bar{f}(\mathbf{x}_t)$ for all $1 \leq t \leq T$
- A model $y = f(\mathbf{x}|\mathcal{D}_T)$ is learned from $\mathcal{D}_T$ to minimize loss

Empirical Risk

$$R_{emp}(f \,|\, \mathcal{D}_T) = \frac{1}{T} \sum_{t=1}^{T} l\left(y_t, f(\mathbf{x}_t|\mathcal{D}_T)\right)$$

Expected Risk

$$R(f \,|\, \mathcal{D}_T) = \mathbf{E}_{p(\mathbf{x})}\left[l\left(\bar{f}(\mathbf{x}), f(\mathbf{x}|\mathcal{D}_T)\right) \right]$$

Bandlimitedness

- For real-world applications, target functions must be bandlimited.
- Non-bandlimited processes must be driven by unlimited power.
- Real-world data are always generated from bandlimited processes.

Definition 7 (strictly bandlimited)

$$F(\boldsymbol{\omega}) = \int \cdots \int_{-\infty}^{+\infty} f(\mathbf{x}) \; e^{-i\mathbf{x}\cdot\boldsymbol{\omega}} \, d\mathbf{x} = 0 \;\text{ if }\; \|\boldsymbol{\omega}\| > B$$

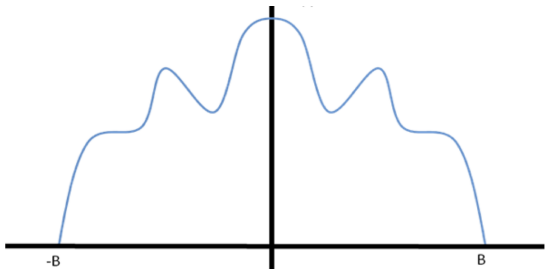
Definition 8 (approximately bandlimited)

$\forall \epsilon > 0$, $\exists B_\epsilon > 0$, out-of-band residual energy satisfies

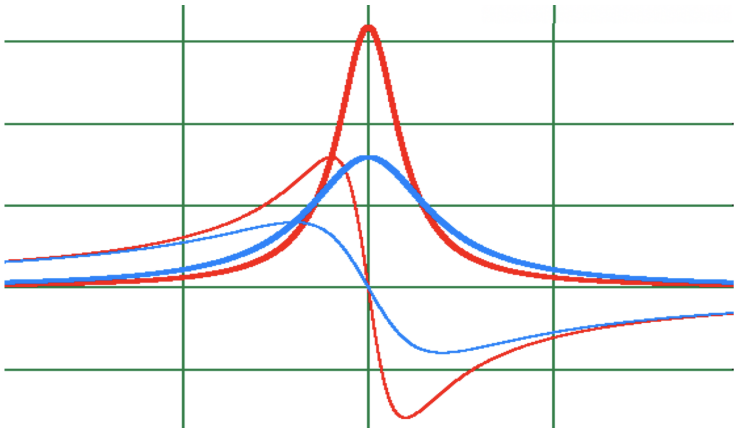
$$\int \cdots \int_{\|\boldsymbol{\omega}\| > B_\epsilon} \|F(\boldsymbol{\omega})\|^2 \, d\boldsymbol{\omega} < \epsilon^2$$

Illustration of Bandlimited Fourier Spectrum

- Strictly bandlimited $F(\omega)$:



- Approximately bandlimited $F(\omega)$:



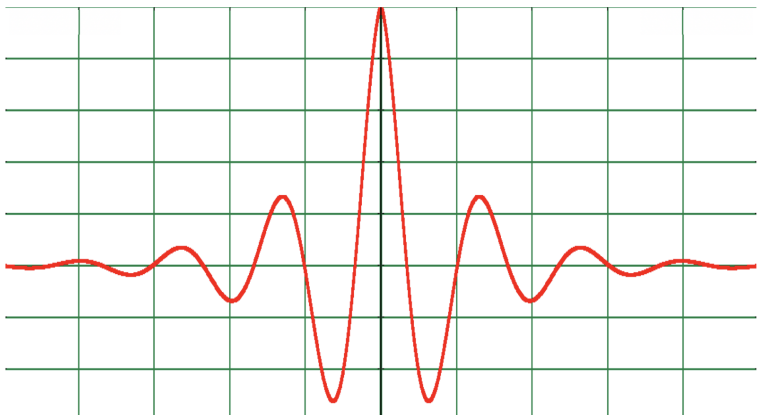
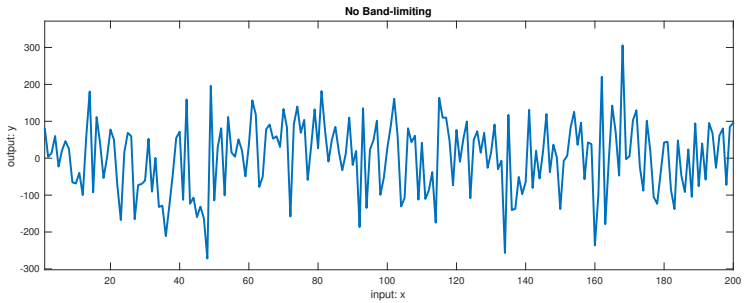
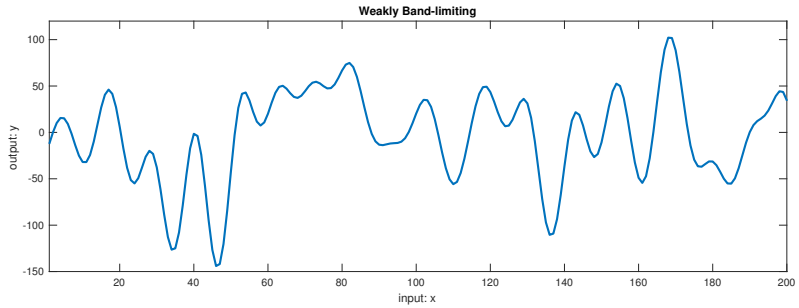


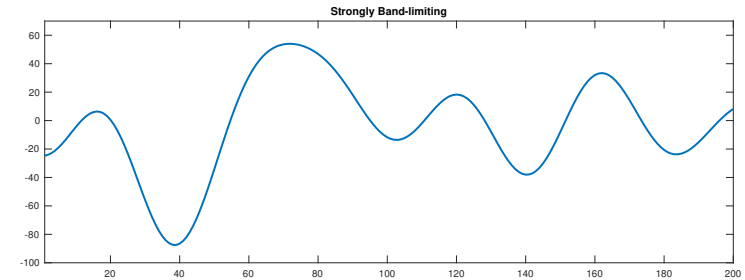
Illustration of Bandlimited Functions



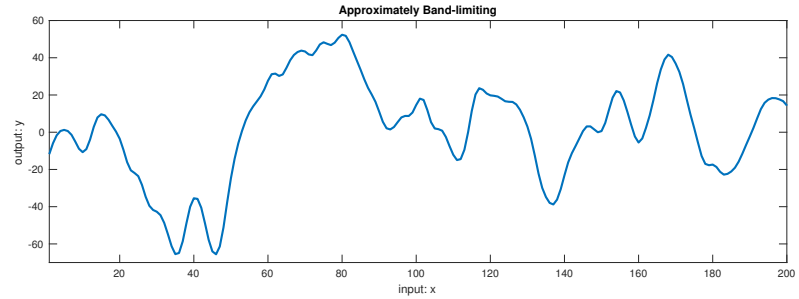
non-bandlimited function



strictly bandlimited function by a large B



strictly bandlimited function by a small B



approximately bandlimited function

Perfect Learning

Definition 9 (perfect learning)

Learn a model from a **finite** set of training samples to achieve not only **zero** empirical risk but also **zero** expected risk

Theorem 10 (existence of perfect learning)

If target function $\bar{f}(\mathbf{x})$ *is strictly or approximately* **bandlimited**, *there* **exists** *a method to learn a model* $f(\mathbf{x}|\mathcal{D}_T)$ *from* $\mathcal{D}_T$, *not only leading to zero empirical risk* $R_{emp}(f|\mathcal{D}_T) = 0$ *but also yielding zero expected risk in probability* $R(f|\mathcal{D}_T) \xrightarrow{P} 0$ *as* $T \to \infty$.

Proof sketch:

Like a stochastic version of multidimensional sampling theorem [PM62; Me01]

Perfect Learning

Corollary 11

If $\bar{f}(\mathbf{x}) \cdot p(\mathbf{x})$ is not strictly nor approximately bandlimited, no matter how many training samples to use, $R(f|\mathcal{D}_T)$ of all realizable learning algorithms have a nonzero lower-bound: $\lim_{T\to\infty} R(f\,|\,\mathcal{D}_T) \geq \varepsilon > 0$.

Therefore, we conclude:

Perfect Learning

target function $\bar{f}(\mathbf{x})$ is bandlimited $\Longleftrightarrow$ perfect learning is feasible

Non-asymptotic Analysis of Perfect Learning

When T is finite, performance is measured by **mean** expected risk:

$$\mathcal{R}_T = \mathbf{E}_{\mathcal{D}_T}\left[\mathbf{E}_{p(\mathbf{x})}\left[\|f(\mathbf{x}|\mathcal{D}_T) - \bar{f}(\mathbf{x})\|^2\right]\right]$$

Theorem 12 (error bound of perfect learning)

If $\mathbf{x} \sim p(\mathbf{x})$ *within a hypercube* $[-U, U]^K \subset \mathbb{R}^K$, *target function* $\bar{f}(\mathbf{x})$ *is bandlimited by* B, *perfect learner is upper-bounded as:*

$$\mathcal{R}_T^* < \left[\frac{(KBU)^{n+1} \cdot H}{(n+1)!}\right]^2$$

where $n \simeq O(T^{1/K})$ *and* $H = \sup_{\mathbf{x}} |\bar{f}(\mathbf{x})|$.

- This error bound is independent of model complexity M
- When T is small, difficulty of learning is quantified by KBU

Perfect Learning in Practice

Theorem 13 (perfect learning in practice)

If target function $\bar{f}(\mathbf{x})$ is strictly or approximately bandlimited, assume a strictly or approximately bandlimited model, $f(\mathbf{x})$, is learned from a sufficiently large training set $\mathcal{D}_T$. If this model yields zero empirical risk on $\mathcal{D}_T$:

$$R_{emp}(f \,|\, \mathcal{D}_T) = 0,$$

then it is guaranteed to yield zero expected risk:

$$R(f \,|\, \mathcal{D}_T) \longrightarrow 0 \quad as \;\; T \to \infty.$$

Proof sketch of Theorem 13

Any bandlimited function may be represented as a series of Fourier base functions with decaying coefficients:

- target function:

$$\bar{f}(\mathbf{x}) = \cdots\cdots + \eta_{\boldsymbol{k}_{-1}}\ e^{2\pi i \boldsymbol{k}_{-1}\cdot\mathbf{x}} + \eta_{\boldsymbol{k}_0}\ e^{2\pi i \boldsymbol{k}_0\cdot\mathbf{x}} + \eta_{\boldsymbol{k}_1}\ e^{2\pi i \boldsymbol{k}_1\cdot\mathbf{x}} + \cdots\cdots$$

- model to be learned:

$$f(\mathbf{x}) = \cdots\cdots + \theta_{\boldsymbol{k}_{-1}}\ e^{2\pi i \boldsymbol{k}_{-1}\cdot\mathbf{x}} + \theta_{\boldsymbol{k}_0}\ e^{2\pi i \boldsymbol{k}_0\cdot\mathbf{x}} + \theta_{\boldsymbol{k}_1}\ e^{2\pi i \boldsymbol{k}_1\cdot\mathbf{x}} + \cdots\cdots$$

- training samples: $\mathcal{D}_T = \{(\mathbf{x}_t, y_t) | 1 \leq t \leq T\}$

$$y_t = \bar{f}(\mathbf{x}_t) \quad t = 1, \cdots, T$$

$$y_t = f(\mathbf{x}_t) \quad t = 1, \cdots, T$$

Proof sketch of Theorem 13 (cont'd)

- T linear equations:

$$\bar{f}(\mathbf{x}_t) - f(\mathbf{x}_t) = 0 \quad t = 1, \cdots, T$$

- target function and the model:

$$\bar{f}(\mathbf{x}) = \cdots\cdots + \underbrace{\eta_{\boldsymbol{k}_{-1}}\, e^{2\pi i \boldsymbol{k}_{-1}\cdot\mathbf{x}} + \eta_{\boldsymbol{k}_0}\, e^{2\pi i \boldsymbol{k}_0\cdot\mathbf{x}} + \eta_{\boldsymbol{k}_1}\, e^{2\pi i \boldsymbol{k}_1\cdot\mathbf{x}} + \cdots\cdots}_{T \text{ most significant terms}}$$

$$f(\mathbf{x}) = \cdots\cdots + \underbrace{\theta_{\boldsymbol{k}_{-1}}\, e^{2\pi i \boldsymbol{k}_{-1}\cdot\mathbf{x}} + \theta_{\boldsymbol{k}_0}\, e^{2\pi i \boldsymbol{k}_0\cdot\mathbf{x}} + \theta_{\boldsymbol{k}_1}\, e^{2\pi i \boldsymbol{k}_1\cdot\mathbf{x}} + \cdots\cdots}_{T \text{ most significant terms}}$$

- determine T coefficients up to good precision: $\theta_{\boldsymbol{k}} \to \eta_{\boldsymbol{k}}$
- as $T \to \infty$, $f(\mathbf{x}) \to \bar{f}(\mathbf{x})$

Machine Learning Models vs. Bandlimitedness

All machine learning models are approximately bandlimited under some minor conditions:

1. input $\mathbf{x}$ is bounded
2. model size is finite
3. all model parameters are bounded
4. model is piecewise continuous

PAC-learnable models are bandlimited

All PAC-learnable models are approximately bandlimited, including linear models, logistic regression, statistical models, **neural networks** ...

Why Neural Nets Generalize: Asymptotic Regularization

Corollary 14

Assume neural network, $f_{\mathbf{w}}(\mathbf{x})$, is learned from a sufficiently large training set $\mathcal{D}_T$, generated by a bandlimited process $\bar{f}(\mathbf{x})$. If $f_{\mathbf{w}}(\mathbf{x})$ yields zero empirical risk on $\mathcal{D}_T$:

$$R_{emp}(f_{\mathbf{w}} \,|\, \mathcal{D}_T) = 0,$$

then it surely yields zero expected risk as $T \to \infty$:

$$\lim_{T \to \infty} \; R(f_{\mathbf{w}} \,|\, \mathcal{D}_T) \longrightarrow 0.$$

Definition 15 (asymptotic regularization)

Due to the bandlimitedness property, neural network asymptotically regularizes itself as $T \to \infty$.

Conclusions

Deep Learning Theory: Neural Nets

1. **expressiveness**: neural nets $\Leftrightarrow$ universal approximator in $L^1(\mathbb{U}_K)$
2. **optimization**: learning large neural nets is unexpectedly "simple"
   - behaves like convex optimization, conditional on $H(\mathbf{w})$
   - over-parameterized neural nets are **complete** in $L^1(\mathbb{U}_K)$
3. **generalization**: **asymptotic regularization**
   - real-world data are generated by bandlimited processes
   - neural nets are approximately bandlimited
   - neural nets **self-regularize** on sufficiently large training sets

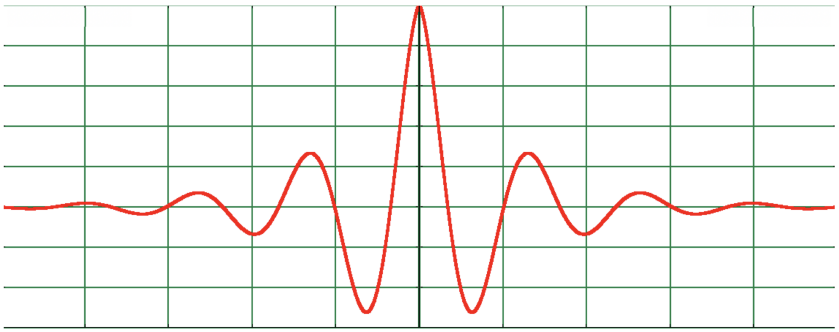
Conclusions

- Under **big data** + **big model**, neural nets solve supervised learning problems in **statistical** sense:

$$\lim_{T\to\infty} \mathbf{R}(\hat{f}_{\mathbf{w}}|\mathcal{D}_T) = 0.$$

  1. collect sufficient training samples (determined by *KBU*)
  2. fit over-parameterized models (neural nets) onto them

- However, **adversarial attack** is new and different ...



all neural nets are approximately bandlimited

More details and all proofs are found in:

 Hui Jiang, "A New Perspective on Machine Learning: How to do Perfect Supervised Learning", *preprint arXiv:1901.02046*, 2019.

 Hui Jiang, "Why Learning of Large-Scale Neural Networks Behaves Like Convex Optimization", *preprint arXiv:1903.02140*, 2019.

Thank You!

(Q&A)

References:

P. Auer, M. Herbster, and M.K. Warmuth. "Exponentially many local minima for single neurons". In: *Proc. of Advances in Neural Information Processing Systems 8 (NIPS)*. 1995.

A. R. Barron. "Approximation and Estimation Bounds for Artificial Neural Networks". In: *Machine Learning* 14 (1994), pp. 115–131.

A. L. Blum and R. L. Rivest. "Training a 3-node neural network is NP-complete". In: *Neural Networks* 5.1 (1992), pp. 117–127.

G. Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals, and Systems* 2 (1989), pp. 303–314.

F. A. Marvasti and et.al. *Nonuniform Sampling : Theory and Practice*. Kluwer Academic / Plenum Publishers, 2001.

D. P. Petersen and D. Middleton. "Sampling and Reconstruction of Wave-Number-Limited Functions in N-Dimensional Euclidean Spaces". In: *Information and Control* 5 (1962), pp. 279–323.

V. N. Vapnik. *The nature of statistical learning theory*. New York : Springer, 2000.