Deep Learning: Towards Deeper Understanding

Yuan YAO
HKUST
Acknowledgement

A following-up course at HKUST that evolves every year:
https://deeplearning-math.github.io/
A Brief History of Neural Networks
Perceptron: single-layer

- Invented by Frank Rosenblatt (1957)

\[ z = \overrightarrow{w} \cdot \overrightarrow{x} + b \]

*The theory reported here clearly demonstrates the feasibility and fruitfulness of a quantitative statistical approach to the organization of cognitive systems. By the study of systems such as the perceptron, it is hoped that those fundamental laws of organization which are common to all information handling systems, machines and men included, may eventually be understood.*

---

Frank Rosenblatt


Cybernetics/neural networks

Norbert Wiener

Warren McCulloch & Walter Pitts

Frank Rosenblatt
Hilbert’s 13th Problem

Algebraic equations (under a suitable transformation) of degree up to 6 can be solved by functions of two variables. What about

\[ x^7 + ax^3 + bx^2 + cx + 1 = 0? \]

Hilbert’s conjecture: \( x(a, b, c) \) cannot be expressed by a superposition (sums and compositions) of bivariate functions.

**Question:** can every continuous (analytic, \( C^\infty \), etc) function of \( n \) variables be represented as a superposition of continuous (analytic, \( C^\infty \), etc) functions of \( n - 1 \) variables?

**Theorem (D. Hilbert)**

*There is an analytic function of three variables that cannot be expressed as a superposition of bivariate ones.*
Kolmogorov’s Superposition Theorem

Theorem (A. Kolmogorov, 1956; V. Arnold, 1957)

Given \( n \in \mathbb{Z}^+ \), every \( f_0 \in C([0, 1]^n) \) can be represented as

\[
f_0(x_1, x_2, \cdots, x_n) = \sum_{q=1}^{2n+1} g_q \left( \sum_{p=1}^{n} \phi_{pq}(x_p) \right),
\]

where \( \phi_{pq} \in C[0, 1] \) are increasing functions independent of \( f_0 \) and \( g_q \in C[0, 1] \) depend on \( f_0 \).

- Can choose \( g_q \) to be all the same \( g_q \equiv g \) (Lorentz, 1966).
- Can choose \( \phi_{pq} \) to be Hölder or Lipschitz continuous, but not \( C^1 \) (Fridman, 1967).
- Can choose \( \phi_{pq} = \lambda_p \phi_q \) where \( \lambda_1, \cdots, \lambda_n > 0 \) and \( \sum_p \lambda_p = 1 \) (Sprecher, 1972).

If \( f \) is a multivariate continuous function, then \( f \) can be written as a superposition of composite functions of mixtures of continuous functions of single variables: finite composition of continuous functions of a single variable and the addition.
Kolmogorov’s Exact Representation is Irrelevant


- Lacking smoothness in \( h \) and \( g \) [Vitushkin’1964] fails to guarantee the generalization ability (stability) against noise and perturbations

- The representation is not universal in the sense that \( g \) and \( h \) both depend on the function \( F \) to be represented.
A Simpler, Similar Theorem

For (possibly discontinuous) $f : [0, 1]^N \to \mathbb{R}$ there exists (possibly discontinuous) $g, h_i : \mathbb{R} \to \mathbb{R}$.

$$f(x_1, \ldots, x_N) = g \left( \sum_i h_i(x_i) \right)$$

Proof: Select $h_i$ to spread out the digits of its argument so that $\sum_i h_i(x_i)$ contains all the digits of all the $x_i$. 
Universal Approximate Representation


For continuous $f : [0, 1]^N \rightarrow \mathbb{R}$ and $\varepsilon > 0$ there exists

$$F(x) = \alpha^\top \sigma(Wx + \beta)$$

$$= \sum_i \alpha_i \sigma \left( \sum_j W_{i,j} x_j + \beta_i \right)$$

such that for all $x$ in $[0, 1]^N$ we have $|F(x) - f(x)| < \varepsilon$.

Complexity (regularity, smoothness) thereafter becomes the central pursuit in Approximation Theory.
The Perceptron Algorithm for classification

\[ \ell(w) = - \sum_{i \in M_w} y_i \langle w, x_i \rangle, \quad M_w = \{ i : y_i \langle x_i, w \rangle < 0, y_i \in \{-1, 1\} \}. \]

The Perceptron Algorithm is a Stochastic Gradient Descent method (Robbins-Monro 1951):

\[ w_{t+1} = w_t - \eta_t \nabla_i \ell(w) \]

\[ = \begin{cases} 
    w_t - \eta_t y_i x_i, & \text{if } y_i w_i^T x_i < 0, \\
    w_t, & \text{otherwise.}
\end{cases} \]
Finiteness of Stopping Time and Margin

The perceptron convergence theorem was proved by Block (1962) and Novikoff (1962). The following version is based on that in Cristianini and Shawe-Taylor (2000).

**Theorem 1** (Block, Novikoff). Let the training set $S = \{(x_1, t_1), \ldots, (x_n, t_n)\}$ be contained in a sphere of radius $R$ about the origin. Assume the dataset to be linearly separable, and let $w_{\text{opt}}$, $\|w_{\text{opt}}\| = 1$, define the hyperplane separating the samples, having functional margin $\gamma > 0$. We initialise the normal vector as $w_0 = 0$. The number of updates, $k$, of the perceptron algorithms is then bounded by

$$k \leq \left(\frac{2R}{\gamma}\right)^2. \quad (10)$$

**Input ball:** $R = \max_i \|x_i\|.$

**Margin:** $\gamma := \min_i y_i f(x_i)$
Locality or Sparsity of Computation

Minsky and Papert, 1969
Perceptron can’t do XOR classification
Perceptron needs infinite global information to compute connectivity

Locality or Sparsity is important:
Locality in time?
Locality in space?
Multilayer Perceptrons (MLP) and Back-Propagation (BP) Algorithms

Rumelhart, Hinton, Williams (1986)
Learning representations by back-propagating errors, Nature, 323(9): 533-536

BP algorithms as stochastic gradient descent algorithms (Robbins–Monro 1950; Kiefer-Wolfowitz 1951) with Chain rules of Gradient maps

MLP classifies XOR, but the global hurdle on topology (connectivity) computation still exists: condition number in Blum-Shub-Smale real computation models helps.
BP Algorithm: Forward Pass

- Cascade of repeated [linear operation followed by coordinatewise nonlinearity]’s
- Nonlinearities: sigmoid, hyperbolic tangent, (recently) ReLU.

Algorithm 1 Forward pass

Input: $x_0$
Output: $x_L$

1: \textbf{for} $\ell = 1$ to $L$ \textbf{do}
2: \quad $x_\ell = f_\ell(W_\ell x_{\ell-1} + b_\ell)$
3: \textbf{end for}
BP algorithm = Gradient Descent Method

- Training examples $\{x^i_0\}_{i=1}^n$ and labels $\{y^i\}_{i=1}^n$
- Output of the network $\{x^i_L\}_{i=1}^m$
- Objective Square loss, cross-entropy loss, etc.

$$J(\{W_l\}, \{b_l\}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} \|y^i - x^i_L\|_2^2$$  \hspace{1cm} (1)

- Gradient descent

$$W_l = W_l - \eta \frac{\partial J}{\partial W_l}$$
$$b_l = b_l - \eta \frac{\partial J}{\partial b_l}$$

In practice: use Stochastic Gradient Descent (SGD)
Derivation of BP: Lagrangian Multiplier
LeCun et al. 1988

Given $n$ training examples $(I_i, y_i) \equiv \text{(input,target)}$ and $L$ layers

- Constrained optimization

$$\min_{W,x} \sum_{i=1}^{n} \|x_i(L) - y_i\|_2$$

subject to

$$x_i(\ell) = f_\ell \left[ W_\ell x_i(\ell - 1) \right],$$

$$i = 1, \ldots, n, \quad \ell = 1, \ldots, L, \quad x_i(0) = I_i$$

- Lagrangian formulation (Unconstrained)

$$\min_{W,x,B} \mathcal{L}(W, x, B)$$

$$\mathcal{L}(W, x, B) = \sum_{i=1}^{n} \left\{ \|x_i(L) - y_i\|_2^2 + \right.$$

$$\left. \sum_{\ell=1}^{L} B_i(\ell)^T \left( x_i(\ell) - f_\ell \left[ W_\ell x_i(\ell - 1) \right] \right) \right\}$$

back-propagation – derivation

Forward pass

\[ x_i(\ell) = f_\ell \left[ W_\ell x_i(\ell - 1) \right] \quad \ell = 1, \ldots, L, \quad i = 1, \ldots, n \]

\[ \frac{\partial L}{\partial x}, z_\ell = [\nabla f_\ell] B(\ell) \]

Backward (adjoint) pass

\[ z(L) = 2\nabla f_L \left[ A_i(L) \right] (y_i - x_i(L)) \]

\[ z_i(\ell) = \nabla f_\ell \left[ A_i(\ell) \right] W_{\ell+1}^T z_i(\ell + 1) \quad \ell = 0, \ldots, L - 1 \]

\[ W \leftarrow W + \lambda \frac{\partial L}{\partial W} \]

Weight update

\[ W_\ell \leftarrow W_\ell + \lambda \sum_{i=1}^{n} z_i(\ell)x_i^T(\ell - 1) \]
Convolutional Neural Networks: shift invariances and locality

- Can be traced to *Neocognitron* of Kunihiko Fukushima (1979)
- Imposes **shift invariance** and **locality** on the weights
- Forward pass remains similar
- Backpropagation slightly changes – need to sum over the gradients from all spatial positions
MNIST Dataset Test Error
LeCun et al. 1998

Simple SVM performs as well as Multilayer Convolutional Neural Networks which need careful tuning (LeNets)

Dark era for NN: 1998-2012
Around the year of 2012: return of NN as `deep learning’

Speech Recognition: TIMIT

Computer Vision: ImageNet
Depth as function of year

ILSVRC ImageNet Top 5 errors

- ImageNet (subset):
  - 1.2 million training images
  - 100,000 test images
  - 1000 classes
- ImageNet large-scale visual recognition Challenge

source: https://www.linkedin.com/pulse/must-read-path-breaking-papers-image-classification-muktabh-mayank
Some Cold Water: Tesla Autopilot Misclassifies Truck as Billboard

**Problem:** Why? How can you trust a blackbox?
Deep Learning may be fragile in generalization against noise!

- Small but malicious perturbations can result in severe misclassification
- Malicious examples generalize across different architectures
- What is source of instability?
- Can we robustify network?

[Goodfellow et al., 2014]
What’s wrong with deep learning?

Ali Rahimi NIPS’17: Machine (deep) Learning has become alchemy.
https://www.youtube.com/watch?v=ORHFOnaEzPc

Yann LeCun CVPR’15, invited talk: What’s wrong with deep learning?
One important piece: missing some theory (clarity in understanding)!

Being alchemy is certainly not a shame, not wanting to work on advancing to chemistry is a shame! -- by Eric Xing
What’s wrong with deep learning?

In this course, we only raise **problems**, and leave you to explore **answers**.
CNN learns texture features, not shapes

(a) Texture image
81.4% Indian elephant
10.3% indri
8.2% black swan

(b) Content image
71.1% tabby cat
17.3% grey fox
3.3% Siamese cat

(c) Texture-shape cue conflict
63.9% Indian elephant
26.4% indri
9.6% black swan

Geirhos et al. ICLR 2019

https://videoken.com/embed/W2HvLBhCJQ?tocitem=46
Lack of Causality or Interpretability

- ImageNet training learns non-semantic texture features: after random shuffling of patches, shapes information are destroyed which does not affect CNN’s performance much.

![Image](image.png)

Figure 6. Visualization of patch-shuffling transformation. The first row shows probability of “cake” assigned by different models. (a) Original Image (b) Patch-Shuffle 2 (c) Patch-Shuffle 4 (d) Patch-Shuffle 8

Figure 7. "Accuracy on correctly classified images" for different models on patch-shuffled Tiny ImageNet and Caltech-256 with different splitting numbers. Detailed results are listed in the appendix.

When decreasing the saturation level, all models have similar degree of performance degradation, indicating that AT-CNNs are not robust to all kinds of image distortions. They tend to be more robust for fixed types of distortions. We leave the further investigation regarding this issue as future work.

4.2.3. Patch-Shuffling

Stylizing and saturation operation aim at changing or removing the texture information of original images, while preserving the features of shapes and edges. In order to test the different bias of AT-CNN and standard CNN in the other way around, we shatter the shape and edge information by splitting the images into $k \times k$ patches and then randomly shuffling them. This operation could still maintains the local textures if $k$ is not too large.

Figure 6 shows one example of patch-shuffled images under different numbers of splitting. The first row shows the probabilities assigned by different models to the ground truth class of the original image. Obviously, after random shuffling, the shapes and edge features are destroyed dramatically, the prediction probability of the adversarially trained CNNs drops significantly, while the normal CNNs still maintains a high confidence over the ground truth class. This reveals AT-CNNs are more biased towards shapes and edges than normally trained ones.

Moreover, Figure 7 depicts the “accuracy on correctly classified images” for all the models measured on “Patch-shuffled” test set with increasing number of splitting pieces. AT-CNNs, especially trained against with a stronger attack are more sensitive to “Patch-shuffling” operations in most of our experiments.

Note that under “Patch-shuffle 8” operation, all models have similar “accuracy of correctly classified images”, which is largely due to the severe information loss. Also note that this accuracy of all models on Tiny ImageNet shown in 7 (a) is much lower than that on Caltech-256 in 7 (b). That is, under “Patch-shuffle 1”, normally trained CNN has an accuracy...
Capture spurious correlations and can’t do causal inference on counterfactuals

Example: detection of the action “giving a phone call”

Not giving a phone call.

Giving a phone call ????

Leon Bottou, ICLR 2019

(Oquab et al., CVPR 2014)
~70% correct (SOTA in 2014)

https://videoken.com/embed/8UxS4Is6g1g?tocitem=2
Deep learning is **not robust** -- adversarials are ubiquitous

[BCZOCG’18] Unrestricted Adversarial Example.
Overfitting causes privacy leakage

- Model inversion attack leaks privacy

Figure: Recovered (Left), Original (Right)

Towards a deeper understanding of deep learning

- How to achieve robustness?
  - Madry's adversarial training, random smoothing, ensemble methods, stability regularization, etc.

- How to guarantee privacy?
  - Differential privacy, model inversion privacy, membership privacy, etc.

- How to improve interpretability or causality?
  - Invariance (learning), disentanglement of representation, etc.
Some Theories are limited but help:

- **Approximation Theory and Harmonic Analysis**: What functions are represented well by deep neural networks, without suffering the curse of dimensionality and better than shallow networks?
  - Sparse (local), hierarchical (multiscale), compositional functions avoid the curse dimensionality
  - Group (translation, rotational, scaling, deformation) invariances achieved as depth grows

- **Statistics learning**: How can deep learning generalize well without overfitting the noise?
  - “Benign overfitting”? …

- **Optimization**: What is the landscape of the empirical risk and how to optimize it efficiently?
  - Wide networks may have simple landscape for GD/SGD algorithms …
Thank you!
Generalization Ability

Why over-parameterized models may generalize well without overfitting?
Generalization Error

- Consider the empirical risk minimization under i.i.d. samples

\[
\hat{R}_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i; \theta)) + R(\theta)
\]

- The population risk with respect to unknown distribution

\[
R(\theta) = E_{x,y \sim P} \ell(y, f(x; \theta))
\]

- Fundamental Theorem of Machine Learning (for 0-1 misclassification loss, called 'errors' below)

\[
R(\theta) = \underbrace{\hat{R}_n(\theta)}_{\text{training loss/error}} + \underbrace{R(\theta) - \hat{R}_n(\theta)}_{\text{generalization loss/error}}
\]
Why big models generalize well?

Ben Recht et al. 2016
The Bias-Variance Tradeoff?

Models where $p > 20n$ are common

Deep models
As model complexity grows \((p>n)\), training error goes down to zero, but test error does not increase. Why overparameterized models do not overfit here? -- Tommy Poggio, 2018
Some tentative answers:

- **Belkin** et al.: Interplolation (overfitting) has a low generalization error in overparameterization regime
  - [https://simons.berkeley.edu/talks/tbd-65](https://simons.berkeley.edu/talks/tbd-65)
- For overparameterized linear regression models:
  - **Peter Bartlett** et al. [https://simons.berkeley.edu/talks/tbd-51](https://simons.berkeley.edu/talks/tbd-51)
  - **Trevor Hastie** et al. asymptotic theory based on random matrix theory
- For logistic regressions:
  - **Telgarsky, Srebro**, et al. GD converges to max margin solution
- Nonlinear neural networks: ???
- Some warnings on “interpolations”:
  - Ben Recht: [https://simons.berkeley.edu/talks/tbd-63](https://simons.berkeley.edu/talks/tbd-63)
Yet, Overfitting indeed hurts...

- Lack of Robustness

“black hole”
87.7% confidence

+.007 ×

“donut”
99.3% confidence

Courtesy of Dr. Hongyang ZHANG.