Optimization for Deep Learning

Lecture 6: Stochastic Gradient Descent

Kun Yuan

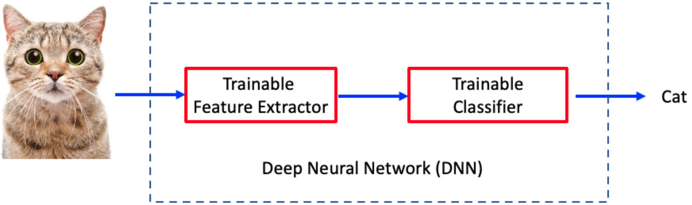
Peking University

Main contents in this lecture

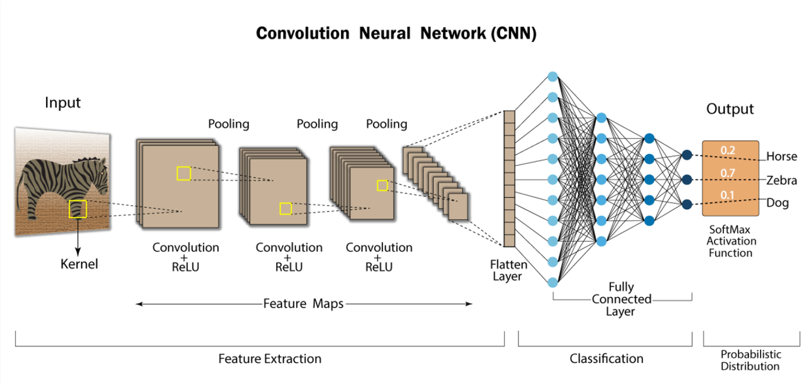
- Deep neural network training
- Stochastic optimization
- Stochastic gradient descent (SGD)
- Mini-batch SGD

Deep neutral network (DNN)

- DNN is widely used in almost all AI applications
- A typical DNN model includes a feature extractor and a classifier
- Well-trained DNN can make precise predictions



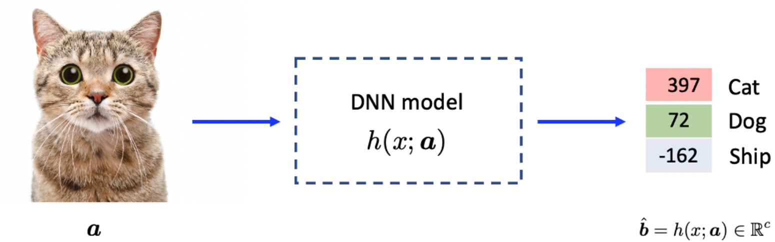
A practical DNN example[1]



[1] Source: analyticsvidhya.com

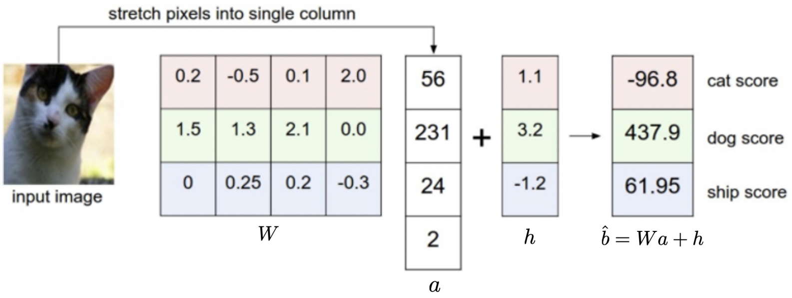
DNN model

- We model DNN as $h(x; a) : \mathbb{R}^d \to \mathbb{R}^c$
  - $x \in \mathbb{R}^d$ is the DNN model parameter to be trained
  - a is a random input data sample
  - c is the number of classes
- Given the model parameter x, DNN outputs prediction scores $\hat{b}$ for input a



DNN model: a trivial example

- Given model parameter $x = [W; h]$, and a linear model $h(x; a) = Wa + h$,
- An illustration of the trivial DNN model and its output is as follows[2]



[2]Source: https://cs231n.github.io/linear-classify/

How to train a DNN model?

- Given good model x, DNN $h(x; a)$ can make precise predictions
- But how to train/achieve the model parameter x ?
- Given a dataset $\{a_i, b_i\}_{i=1}^m$ where b_i is the ground-truth label for data a_i
- Define $L(\hat{b}_i, b_i) = L(h(x; a_i), b_i)$ as a loss function to measure the difference/mismatch between predictions and ground-truth labels
- DNN training is to find a model parameter x such that the mismatch (between pred and real) are minimized across the entire dataset:

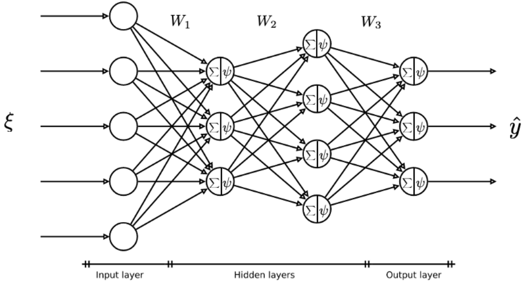
$$\boxed{x^\star = \operatorname*{arg\,min}_{x \in \mathbb{R}^d} \left\{ \frac{1}{m} \sum_{i=1}^{m} L(h(x; a_i), b_i) \right\}}$$

DNN model is notoriously difficult to train

- DNN model $L(h(x;a),b)$ is highly non-convex, and probably non-smooth

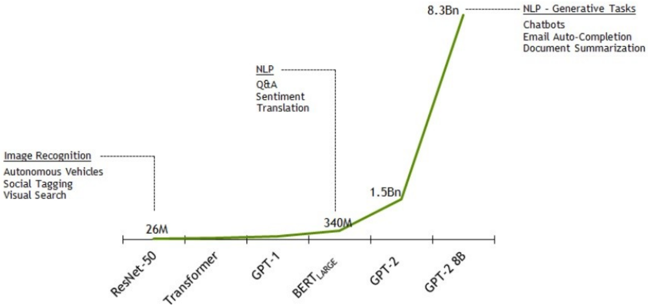
$$h(x;a) = \psi(\cdots\psi(W_2\cdot\psi(W_1 a + h_1) + h_2)\cdots)$$
$$L(\hat{b};b) = \frac{1}{2}\|b-\hat{b}\|^2 \text{ or } -\sum_i b_i\log(\hat{b}_i) \text{ or others}$$

where $x = \{W_i, h_i\}$ and $\psi(\cdot)$ is a non-linear activation function



DNN model is notoriously difficult to train

- Cannot find global minima; trapped into local minima and saddle points
- The dimension of model parameter $x = \{W_i, h_i\}$ (or model size) is huge[3]



[3] Image source: neowin.net

DNN model is notoriously difficult to train

- Cannot find global minima; trapped into local minima and saddle points
- The dimension of model parameter $x = \{W_i, h_i\}$ (or model size) is huge
- The size of the dataset $\{a_i, b_i\}_{i=1}^m$ is huge

$\boxed{\text{DNN Training} = \text{Non-convexity training} + \text{Huge dimension} + \text{Huge dataset}}$

- Our lectures will focus on algorithms to train DNN

Stochastic optimization

- Consider the stochastic optimization problem:

$$\min_{x \in \mathbb{R}^d} \quad f(x) = \mathbb{E}_{\xi \sim \mathcal{D}}[F(x; \xi)]$$

  - ξ is a random variable indicating data samples
  - $\mathcal{D}$ is the data distribution; unknown in advance
  - $F(x; \xi)$ is differentiable in terms of x

- Many applications in signal processing and machine learning

Example: deep neural network

- Recall the DNN training problem

$$\min_{x \in \mathbb{R}^d} \quad \frac{1}{m} \sum_{i=1}^{m} L(h(x; a_i), b_i)$$

which is a finite-sum problem

- Suppose we have infinite data (a, b) following distribution D, the above problem becomes

$$\min_{x \in \mathbb{R}^d} \quad f(x) = \mathbb{E}_{(a,b)\sim\mathcal{D}} L(h(x; a), b)$$

where data pair (a, b) can be regarded as sample ξ.

Stochastic gradient descent

- Recall the problem

$$\min_{x \in \mathbb{R}^d} \quad f(x) = \mathbb{E}_{\xi \sim \mathcal{D}}[F(x; \xi)]$$

- Closed-form of $f(x)$ is unknown; gradient descent is not applicable

- Stochastic gradient descent (SGD):

$$x_{k+1} = x_k - \gamma \nabla F(x_k; \xi_k), \quad \forall k = 0, 1, \cdots$$

  where ξ_k is a data realization sampled at iteration k.

- Since $\{x_k\}$ are random, all iterates $\{x_k\}$ are also random

Assumption

Let $\mathcal{F}_k = \{x_k, \xi_{k-1}, x_{k-1}, \cdots, \xi_0\}$ be the filtration containing all historical variables at and before iteration k (except for ξ_k).

Assumption 1

Given the filtration $\mathcal{F}_k$, we assume

$$\mathbb{E}[\nabla F(x_k;\xi_k)|\mathcal{F}_k] = \nabla f(x_k)$$
$$\mathbb{E}[\|\nabla F(x_k;\xi_k) - \nabla f(x_k)\|^2|\mathcal{F}_k] \leq \sigma^2$$

Implying **unbiased** stochastic gradient and **bounded** variance.

Convergence: smooth and non-convex scenario

Theorem 1

Suppose $f(x)$ is L-smooth and Assumption 1 holds. If $\gamma \le 1/L$, SGD will converge at the following rate

$$\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[\|\nabla f(x_k)\|^2] \le \frac{2\Delta_0}{\gamma(K+1)} + \gamma L \sigma^2,$$

where $\Delta_0 = f(x_0) - f^\star$.

- SGD **cannot** converge to stationary point with constant learning rate
- Smaller learning rate γ or variance σ^2 leads to smaller convergence error

Image Classification

Cifar-10 dataset

50K training images

10K test images

DNN model: ResNet-18

GPU: Tesla V100

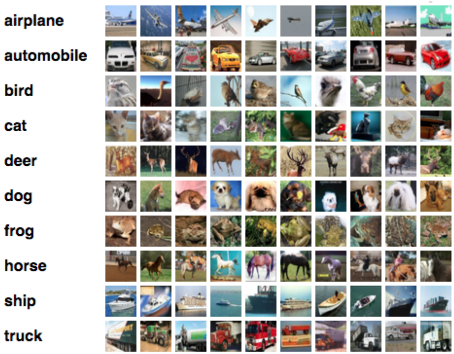
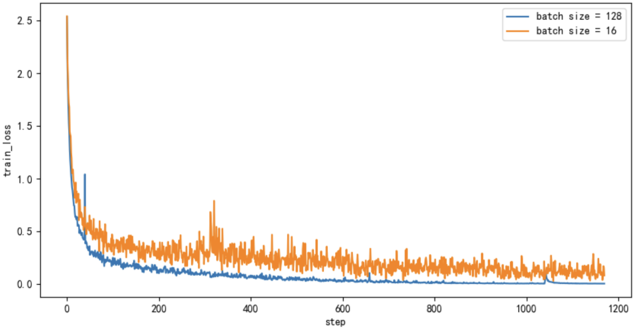


Image Classification

Large batch-size helps training.



Convergence: smooth and non-convex scenario

Corollary 1

Suppose $f(x)$ is L-smooth and Assumption 1 holds. If γ is chosen as

$$\gamma = \left[\left(\frac{2\Delta_0}{(K+1)L\sigma^2} \right)^{-\frac{1}{2}} + L \right]^{-1},$$

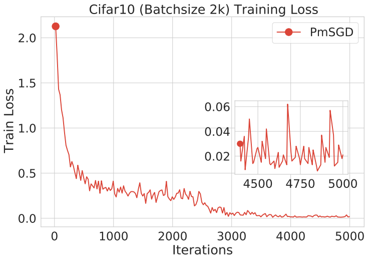
SGD will converge at the following rate

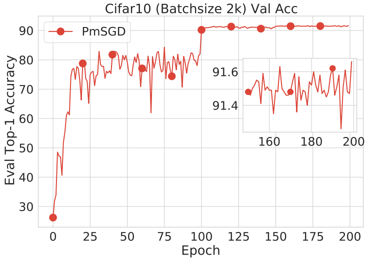
$$\frac{1}{K+1} \sum_{k=0}^{K} \mathbb{E}[\|\nabla f(x_k)\|^2] \le \sqrt{\frac{8L\Delta_0\sigma^2}{K+1}} + \frac{2L\Delta_0}{K+1}.$$

where $\Delta_0 = f(x_0) - f^\star$.

- Decaying rate leads to exact convergence to stationary point
- When $\sigma^2 = 0$, the above rate **reduces to GD**; rate is tight!
- $O(\sqrt{\sigma^2/K})$ is the dominant rate

Image Classification

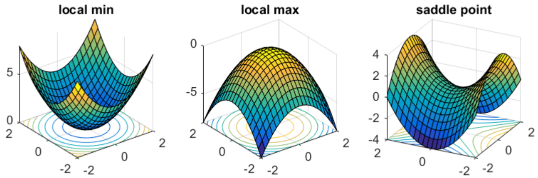




Convergence: smooth and non-convex scenario

$$\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}\|\nabla f(x_k)\|^2 = O\left(\sqrt{\frac{L\sigma^2}{K+1}} + \frac{L}{K+1}\right)$$

- When iteration $K \to \infty$, it holds that $\mathbb{E}\|\nabla f(x_K)\|^2 \to 0$
- $\mathbb{E}\|\nabla f(x_K)\|^2 \to 0$ implies SGD converges to a stationary solution
- A stationary solution can be local min, local max, or saddle point[4]



[4] Image source: from Prof. Rong Ge's online post

Convergence: smooth and non-convex scenario

- Generally speaking, approaching the stationary solution is the best result we can get for SGD; no guarantee to approach the global minimum
- Empirically, SGD performs extremely well when training DNN
- Recent advanced studies show SGD can escape local maximum, saddle point, and even "sharp" local minimum, see, e.g., (Ge et al., 2015; Sun et al., 2015; Jin et al., 2017; Du et al., 2018, 2019; Kleinberg et al., 2018)
- SGD can even find global minimum under certain conditions, e.g. the PL condition (Karimi et al., 2016)

However, we will skip these interesting results in this lecture

Convergence: smooth and convex scenario

Theorem 2

Suppose $f(x)$ is convex and L-smooth. Under Assumption 1, if $\gamma \le 1/(2L)$, SGD will converge at the following rate

$$\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[f(x_k)-f(x^\star)] \le \frac{\Delta_0}{\gamma(K+1)} + \gamma\sigma^2$$

where $\Delta_0 = \|x_0 - x^\star\|^2$. If we further choose $\gamma = \left[\left(\frac{\Delta_0}{(K+1)\sigma^2}\right)^{-\frac{1}{2}} + 2L\right]^{-1}$, SGD converges as follows

$$\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[f(x_k)-f(x^\star)] \le 2\sqrt{\frac{\sigma^2\Delta_0}{K+1}} + \frac{2L\Delta_0}{K+1}.$$

Tight rate. Reduces to GD when $\sigma^2 = 0$.

Convergence: smooth and strongly-convex scenario

Theorem 3

Suppose $f(x)$ is μ-strongly convex and L-smooth. Under Assumption 1, if $\gamma \le 1/L$, SGD will converge at the following rate

$$\mathbb{E}[f(x_k)] - f^\star \le (1-\gamma\mu)^k \Delta_0 + \frac{\gamma L \sigma^2}{\mu}.$$

where $\Delta_0 = f(x_0) - f^\star$. If we further choose $\gamma = \min\{\frac{1}{L}, \frac{1}{\mu K} \ln\left(\frac{\mu^2 \Delta_0 K}{L\sigma^2}\right)\}$, SGD will converge at the following rate

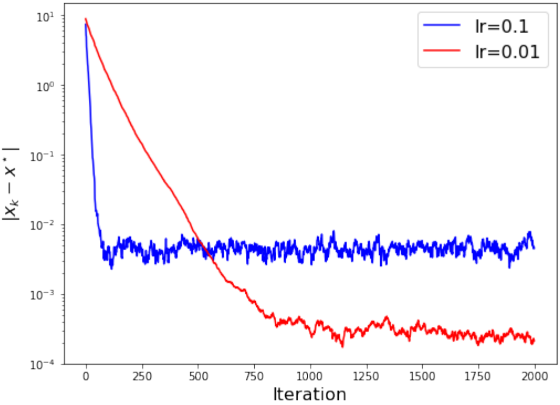
$$\mathbb{E}[f(x_K)] - f^\star = \tilde{O}\left(\frac{L\sigma^2}{\mu^2 K} + \Delta_0 \exp(-\frac{\mu}{L}K)\right)$$

where the $\tilde{O}(\cdot)$ notation hides all logarithm terms.

Tight rate. Reduces to GD when $\sigma^2 = 0$.

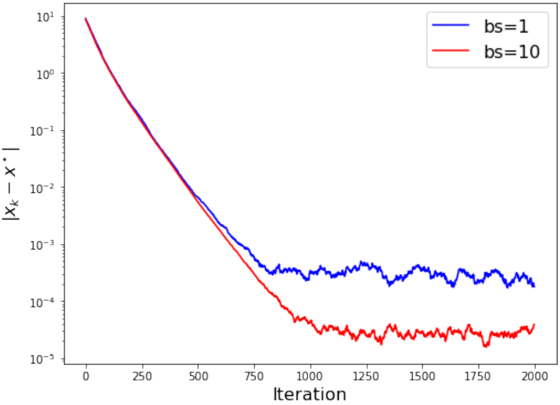
Convergence: smooth and strongly-convex scenario

Linear regression: $\min \frac{1}{2N} \sum_{i=1}^{N} (a_i^T x - b_i)^2$



Convergence: smooth and strongly-convex scenario

Linear regression: $\min \frac{1}{2N} \sum_{i=1}^{N} (a_i^T x - b_i)^2$



SGD summary on rate and complexity

Algorithm	Scenario	Rate	Complexity
SGD	non-convex	$\frac{\sigma}{\sqrt{K}} + \frac{1}{K}$	$\frac{\sigma^2}{\epsilon^2} + \frac{1}{\epsilon}$
	generally-convex	$\frac{\sigma}{\sqrt{K}} + \frac{1}{K}$	$\frac{\sigma^2}{\epsilon^2} + \frac{1}{\epsilon}$
	strongly-convex	$\frac{\sigma^2}{K} + \exp(-K)$	$\frac{\sigma^2}{\epsilon} + \ln(\frac{1}{\epsilon})$
GD	non-convex	$\frac{1}{K}$	$\frac{1}{\epsilon}$
	generally-convex	$\frac{1}{K}$	$\frac{1}{\epsilon}$
	strongly-convex	$\exp(-K)$	$\ln(\frac{1}{\epsilon})$

- SGD recovers GD when $\sigma^2 = 0$
- Existence of σ^2 deteriates the convergence rate significantly

SGD with mini-batch

- In DNN, it is common to sample a batch of data to estimate gradient

- Mini-batch SGD iterate as follows

$$g_k = \frac{1}{B}\sum_{b=1}^{B} \nabla F(x_k; \xi_k^{(b)}),$$

$$x_{k+1} = x_k - \gamma g_k$$

  where B is the batch-size.

- B samples together can provide a much better estimate of $\nabla f(x)$

SGD with mini-batch

We first introduce the filtration

$$\mathcal{F}_k^B = \{x_k, \{\xi_{k-1}^{(b)}\}_{b=1}^B, x_{k-1}, \{\xi_{k-2}^{(b)}\}_{b=1}^B, \cdots, x_0\}$$

Assumption 2

Given the filtration $\mathcal{F}_k^B$, we assume

$$\mathbb{E}[\nabla F(x_k; \xi_k^{(b)}) | \mathcal{F}_k^B] = \nabla f(x_k),$$
$$\mathbb{E}[\|\nabla F(x_k; \xi_k^{(b)}) - \nabla f(x_k)\|^2 | \mathcal{F}_k^B] \le \sigma^2.$$

Moreover, we assume $\{\xi_k^{(b)}\}_{b=1}^B$ are independent of each other for any k.

Implying that mini-batch can provide a much better estimate of $\nabla f(x)$

$$\mathbb{E}[\|g_k - \nabla f(x_k)\|^2 | \mathcal{F}_k^B] = \frac{1}{B^2} \sum_{b=1}^{B} \mathbb{E}[\|\nabla F(x_k; \xi_k^{(b)}) - \nabla f(x_k)\|^2 | \mathcal{F}_k^B] \le \frac{\sigma^2}{B}$$

Mini-batch SGD convergence

Theorem 4

Suppose $f(x)$ is L-smooth and Assumption 2 holds. Mini-batch SGD will converge at the following rate

$$\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[\|\nabla f(x_k)\|^2] = O\left(\sqrt{\frac{L\Delta_0\sigma^2}{B(K+1)}} + \frac{L\Delta_0}{K+1}\right)$$

where $\Delta_0 = f(x_0) - f^\star$.

Large batch-size **accelerates** the convergence; $B = 1$ reduces to SGD

Similar results also hold in convex and strongly-convex scenarios.

Mini-batch SGD convergence

Comparison in the dominant sample complexity

Large batch-size can significantly reduce the sample complexity

Convexity	SGD	Mini-batch SGD
Non-convex	$\frac{L}{\epsilon^2}$	$\frac{L}{B\epsilon^2}$
Convex	$\frac{L}{\epsilon^2}$	$\frac{L}{B\epsilon^2}$
Strongly convex	$\frac{L}{\mu\epsilon}$	$\frac{L}{\mu B\epsilon}$

References I

R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points—online stochastic gradient for tensor decomposition," in *Conference on learning theory*. PMLR, 2015, pp. 797–842.

J. Sun, Q. Qu, and J. Wright, "When are nonconvex problems not scary?" *arXiv preprint arXiv:1510.06096*, 2015.

C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, "How to escape saddle points efficiently," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1724–1732.

S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," *arXiv preprint arXiv:1810.02054*, 2018.

S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1675–1685.

References II

B. Kleinberg, Y. Li, and Y. Yuan, "An alternative view: When does sgd escape local minima?" in *International Conference on Machine Learning*. PMLR, 2018, pp. 2698–2707.

H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 795–811.