ホワイトペーパー

FUJITSU AI Zinrai ディープラーニング　システム パフォーマンスガイド

本書では Zinrai ディープラーニング システムの性能を最大限引き出してお使いいただくために、学習性能に関する設定を解説いたします。

目次

1　はじめに

機械学習(ML: Machine Learning)や深層学習(DL: Deep Learning)は、大量のデータから規則性や特異性を抽出して認識・分類・予測など人間が行うような知的な処理を、サーバーを使って自動的に行えるようにすることを目的としています。様々な分野への応用が検討・実現されており今後の発展が期待される技術です。

一方、適用範囲が広範にわたるため、求められる性能指標については様々であり、ハードウェアインフラストラクチャだけでなく採用するソフトウェア・アルゴリズム・パラメータなどが大きく影響するので、単純にハードウェアの性能や搭載するアクセラレータの種類だけで必要な性能を評価することは困難です。そこで機械学習/深層学習の両分野において、容易に性能比較ができるように、DellEMC、Google、Intel などの IT 企業 と UC Berkeley（カリフォルニア大学バークレー校）や米スタンフォード大学などの大学・研究機関などが連携して策定した機械学習ベンチマーク「MLPerf」が策定されてきました。(引用元: https://japancatalog.dell.com/c/isg_dl_mlperf/)

本書では、Zinrai ディープラーニング システムの性能を最大限引き出してお使いいただくために、弊社が MLPerf training でベストインクラスの性能を引き出すために用いているパラメータとその使い方のヒントの一部を紹介します。

1.1　対象システム

Zinrai ディープラーニング システム 200H で採用している GX2570M5 を使用し、MLPerf v0.7 (2020 年 7 月)で同クラスの NVIDIA 社製 DGX-1 を上回る性能値を発表しています。https://mlperf.org/training-results-0-7/

モデル名		200E				200H	
外観							
採用サーバモデル		PRIMERGY RX2540 M5				PRIMERGY GX2570 M5	
	GPU	NVIDIA® Tesla® T4 x2	NVIDIA® Tesla® T4 x4	NVIDIA® Tesla® V100S x1	NVIDIA® Tesla® V100S x2	NVIDIA® Tesla® V100-SXM2 x8	
	CPU	Intel® Xeon® Gold 6226R x2 (周波数2.90GHz,コア数16C/スレッド数32T, 3 次キャッシュ22MB)				Intel® Xeon® Gold 6248 x2 (周波数2.50GHz,コア数20C/スレッド数40T, 3 次キャッシュ27.5MB)	
	メモリ	128GB (32GB x4)	256GB (32GB x8)	128GB (32GB x4)	256GB (32GB x8)	384GB (32GB x12)	768GB (32GB x24)
	内蔵ストレージ	960GB SATA SSD x2	960GB SATA SSD x8	960GB SATA SSD x2	960GB SATA SSD x8	960GB SATA SSD x2	960GB SATA SSD x8
	InfiniBand	-				Dual port IB HCA (100Gbps) x2	
	ネットワークインターフェイス	標準搭載[2ポート(1000BASE-T/100BASE-TX/10BASE-T択一)]				標準搭載[2ポート(1000BASE-T/100BASE-TX/10BASE-T択一)], Dual port LANカード(10GBASE-T)	
OS		Ubuntu 18.04					
フレームワーク		TensorFlow™, PyTorch, MXNet					
監視ソフトウェア		Prometheus™, Exporter					

2 深層学習のハイパーパラメータ

2.1 概要

2012 年の ImageNet Large Scale Visual Recognition Challenge において、畳み込みニューラルネットワークと全結合ニューラルネットワークを多層重ねたAlexNet が他を大きく引き離して優勝しました。ここから、画像認識、物体認識、音声認識、翻訳などのタスクを実用レベルでこなすニューラルネットワークが次々と開発されて行きました。これらのように多層のニューラルネットワークを学習する技術を深層学習と呼びます。特徴としてパラメータが非常に多いため、学習に大量のデータを必要とすること、および、計算量が非常に多いことが挙げられます。大量のデータはインターネットの普及によって、計算量は高性能 GPU の登場によって解決されました。

深層学習では、通常、学習の進み方の調整や途中の様々な処理を行うかどうかを指定するためにハイパーパラメータが用意されています。例として、

- ソルバのアルゴリズム
- 一度に学習するデータ数(ミニバッチあたりのデータ数)
- 重みの誤差と実際に反映させる量の比(学習率)
- 過去の更新量を維持する割合(モーメンタム)

等、があります。学習を成功させ、高い予測精度を持つモデルを得るためには、ハイパーパラメータを適切に指定する必要があります。

Table 1 は画像認識ニューラルネットワークである ResNet-50 の主なハイパーパラメータについて、MLPerf training 0.6 と 0.7 で使われたものを示しています。MLPerf training 0.7 ではハイパーパラメータ最適化が進んでおり、これに合わせてハイパーパラメータを調整することで学習時間を 44 分短縮することができました。この様にハイパーパラメータの調整によって、一定の予測精度を得るまでの学習時間を短縮することが可能です。同様に予測精度を向上させることもできます。

一方、通常最適なハイパーパラメータを見つけ出すには時間がかかります。例えば、ResNet-50 であれば、強力な GPU (NVIDIA Tesla V100)を 8 台使用しても 1 回の学習に約 1 時間かかります。最適化では、高い精度や高速に学習できるハイパーパラメータ値を探すために、値を変えながら数十回、数百回と試行して良いパラメータ値を見つけます。このため、非常に長い時間を要します。MLPerf のウェブサイトなどで富士通含め各ベンダーが最適化したハイパーパラメータを取得できるのであれば、そちらを利用することをお勧めします。

Table 1 ResNet-50 の代表的なハイパーパラメータと、MLPerf training での 8 GPU 用の値

MLPerf training version		v0.6	v0.7
Hyperparameter	optimizer	SGD w/ fast LARS	SGD w/ fast LARS
	batch size / GPU	208	208
	learning rate	5	7.4
	learning rate scheduling	power of 2	power of 2
	warm-up epoch	5	2
	weight decay	2.00E-04	1.00E-04
	LARS eta	1.00E-03	1.00E-03
	label smoothing	0.1	0.1
	number of epochs	72	37
training time (m)		**115.22**	**71.28**

3　主要なハイパーパラメータの意味

3.1　Optimizer

深層学習では、確率的勾配降下法(Stochastic Gradient Descent：SGD 法)を基本として最適値を求めていきます。この方式では、学習用データを入力とし、ニューラルネットワークで計算した予測値と正解値との差から計算される損失関数(loss)がなるべく小さくなるように、シナプスの重みを変更していきます。重みの変更量は、その時に与えられた学習データに対する予測値と正解値の差から重み等のパラメータの微分係数が計算され、それをともに算出されます。一度に複数のデータを用いて分散学習する場合は微分係数の平均値が用いられます。効率よく最適値に収束するように、いくつかの最適化アルゴリズムが開発されています。SGD w/ fast LARS は、基本的な SGD に、ニューラルネットワークの層毎学習率を調整する(Layer-wise Adaptive Rate Scaling, LARS)機能を加えたものです。

3.2　Batch size

深層学習では大量のデータをいくつかのミニバッチに分け、ミニバッチ単位で学習を進めます。ミニバッチに含まれるデータの個数を batch size といいます。複数のプロセッサを使用して学習する場合、ミニバッチをプロセッサに分配して同時に並列処理する方式が主流です。GPU で処理する場合、GPU 毎の処理数を batch size / GPU として書いています。

最近のプロセッサには大量の演算器が搭載されていますが、batch size が小さいと演算器を使いきれず、プロセッサの性能を生かす事が出来ません。Batch size を大きくすれば、処理速度は高速になります。ただし、ミニバッチ毎のデータが GPU や CPU のメモリに載らないほど大きな batch size を採用すると、かえって遅くなります。

3.3　Epoch

全学習データを繰り返し学習する回数を epoch といいます。百万枚の学習用画像データがある場合を例にすると、全データを 1 回ずつ延べ百万枚の画像を用いた学習を行うと、1 epoch 分の学習になります。2 回ずつ延べ 2 百万枚なら 2 epoch 分です。何回繰り返して学習するかを表すハイパーパラメータを number of epochs といいます。

深層学習で重みを更新する回数は、(number of epochs) × (全データ数) / (batch size)となります(全データ数 / natch size をイテレーション数といいます)。高い予測精度を持ったモデルの生成には、ある程度の重み更新回数が必要です。極端に更新回数が少ない場合は batch size を小さくするか epoch 数を増やすように変更します。

3.4　Learning rate

微分係数をそのまま重みなどの変更量として適用すると、値が大きすぎるか小さすぎて学習がうまく進みません。そのため、learning rate (学習率)をかけ、適切な大きさに変更します。Batch size によっても調整する必要があります。

3.5 Learning rate scheduling

学習開始後しばらくは予測精度の悪い重みパラメータの状態であるため、大きめの learning rate を使用し、速く学習を進めます。学習が進んできたら、小さな loss を与えるパラメータ値に正確に近づけるため、learning rate が小さくなるようにスケジュールします。Learning rate の制御の方式をこのハイパーパラメータで与えます。階段状に変化させる方式、cosine 状に変化させる方式、指数関数的に変化させる方式などがあります。

3.6 Warm-up epoch

重みの初期状態は乱数で決められるため、学習開始直後はでたらめな状態になっています。この状態のまま learning rate を大きくすると、正しい予測ができない重みをもつようになり、まったく学習できない状態に陥る場合もあります。これを避けるため、学習開始直後は小さい learning rate から開始し、warm-up epoch で指定された epoch 数経過したときに指定された learning rate になるように、イテレーション数に比例して leaning rate を大きくします。このハイパーパラメータは、特に batch size が大きい時に有効です。

3.7 Weight decay

過学習を抑制するための重みの正則化のハイパーパラメータです。Batch size が非常に大きい場合は、大きめの値にするとよい場合があります。

3.8 Momentum

直前のイテレーションで採用した重みの更新量のうち、今回イテレーションの更新量に反映する割合を指定します。例えば、momentum が 0.9 だった場合、前回の更新量を 9/10 倍し、今回新たに計算された更新量を 1/10 倍して加算し、実際の更新量とします。移動平均を利用して、より安定して収束するようにしています。Batch size が大きいときは、大きめにすると精度が上がる傾向があります。

3.9 Label smoothing

分類問題では、ワンホットベクトルを教師データとして用います。しかし、要素に 0 か 1 を用いるより、0.1 と 0.9 を用いた方が、学習で得られる最終予測精度が良くなります。Label smoothing で与える値は、0 の代わりに用いるものです。1 の代わりに用いる値は、(1 – label smoothing)になります。

3.10 LARS eta

LARS では、層毎の学習率、重みのノルムと誤差のノルムの比を利用して決めます。この比は 10^{3} 程度以上などかなり大きくなることがあります。これを 1 程度の大きさにするためにかける係数のことで、10^{-3} 程度の値を用います。

4 主なネットワークでのハイパーパラメータの値

以下では、NVIDIA Tesla V100 を 8 台用いて行われた MLPerf training v0.7 での学習用パラメータを紹介します。

4.1 ResNet-50

ResNet-50 は画像認識ニューラルネットワークです。8 GPU 向けのハイパーパラメータの値は Table 2 を参照してください。弊社で以前最適化した際に効果のみられたパラメータは、learning rate、warm-up epoch、weight decay、number of epochs でした。

Table 2 ResNet50 の主なハイパーパラメータ

ハイパーパラメータ名	値	意味
optimizer	SGD w/ fast LARS	ソルバ
batch size / GPU	208	1 GPU あたりのミニバッチサイズ
learning rate	7.4	学習率
learning rate scheduling	power of 2	学習率の変化のさせ方
warm-up epoch	2	Warm up にかかる epoch 数
weight decay	1.00E-04	重み減衰
LARS eta	1.00E-03	層毎の重み調整パラメータ label
label smoothing	0.1	ワンホットベクターの緩和係数
number of epochs	37	実行する epoch 数

4.2 SSD-ResNet34

SSD-ResNet34 は物体検出のためのニューラルネットワークです。8 GPU 向けのハイパーパラメータの値は Table 3 を参照してください。弊社で以前最適化した際に効果のみられたパラメータは、learning rate です。また、階段状に 2 段階で learning rate を小さくしていくスケジューリングを採用した際、小さくするタイミングを独立して最適に調整することも有効でした。

Table 3 SSD-ResNet34 の主なハイパーパラメータ

ハイパーパラメータ名	値	意味
batch size	120	1 GPU あたりのミニバッチサイズ
epochs	80	実行する epoch 数
eval batch size	160	評価時のミニバッチサイズ
learning rate (lr)	2.92E-03	学習率
libraries	NV JPEG	入力画像処理ライブラリの指定
nhwc	true	データレイアウト
warm-up	650	Warm up にかかる step 数
weight decay	1.60E-04	重み減衰

4.3 Mask RCNN

Mask RCNN は SSD と同様の物体認識ニューラルネットワークです。8 GPU 向けのハイパーパラメータの値は Table 4 を参照してください。弊社で以前最適化した際に効果のみられたパラメータは、weight decay でした。

Table 4　Mask RCNN の主なハイパーパラメータ

ハイパーパラメータ名	値	意味
images per batch	48	ミニバッチサイズ
max iteration	80000	実行するイテレーション数
base learning rate	0.06	基本となる学習率
warm-up iterations	625	Warm up にかかる step 数
warm-up factor	9.6E-05	Warm up 時の学習率の計算用係数
warm-up method	mlperf_linear	Warm up の学習率の上げ方
steps	(24000, 32000)	学習率を段階的に下げるタイミング
weight decay	0.00006	重み減衰

4.4 Transformer

Transformer は、翻訳のための有名なニューラルネットワークです。8 GPU 向けのハイパーパラメータの値は Table 5 を参照してください。弊社で以前最適化した際に効果のみられたパラメータは、learning rate、warmup updates でした。

Table 5　Transformer の主要なハイパーパラメータ

ハイパーパラメータ名	値	意味
adam-betas	1.60E-04	ソルバ adam の変数
adam-eps	NV JPEG	ソルバ adam の変数
learning rate (lr)	1.25E-03	学習率
max epoch	30	number of epochs
max tokens	10240	バッチ毎の最大トークン数
optimizer	adam	ソルバ
warm-up tppdates	1000	Warm up にかかる step 数
warm-up updates	1100	Warm up 時の学習率更新頻度

4.5 GNMT

GNMT も Transformer と同じく、翻訳系のネットワークです。8 GPU 向けのハイパーパラメータの値は Table 6 を参照してください。弊社で以前最適化した際に効果のみられたパラメータは、learning rate、remain steps、decay interval でした。

Table 6　GNMT の主なハイパーパラメータ

ハイパーパラメータ名	値	意味
decay interval	859	重さなどの係数の更新

epochs	8	学習 epoch 数
learning rate (lr)	2.8E-03	学習率
optimizer	FusedAdam	ソルバ
remain steps	6053	学習率の減衰を開始する step 数
train batch size	256	1 GPU あたりミニバッチサイズ
warm-up steps	200	Warm up にかかる step 数