

TensorFlow : (r1.0) Programmer's Guide : 変数: 作成、初期化、保存そしてロード

タグ: TensorFlow Programmer's Guide

投稿者: sales-info in TensorFlow 投稿日: 02/09/2016.

TensorFlow : (r1.0) Programmer's Guide : 変数: 作成、初期化、保存そしてロード (翻訳/解説)

翻訳 : (株)クラスキャット セールスインフォメーション

更新日時 : 09/10/2017; 04/27/2017

作成日時 : 02/09/2016

* 本家サイトのドキュメント構成の変更に伴い、本ページは以下のページをベースにするよう変更しました (09/10/2017) :

- https://www.tensorflow.org/versions/r1.0/programmers_guide/variables

* (obsolete) 本家サイトのドキュメント構成の変更に伴い、本ページは以下のページをベースにするよう変更し、また原文の加筆や変更に合わせて翻訳文も更新しました (04/26/2017) :

https://www.tensorflow.org/programmers_guide/variables

* (obsolete) 本ページは、**TensorFlow** の本家サイトの How To – Variables: Creation, Initialization, Saving, and Loading を翻訳した上で適宜、補足説明したものです :

(リンク切れ) https://www.tensorflow.org/versions/master/how_tos/variables/index.html

* サンプルコードの動作確認はしておりますが、適宜、追加改変しています。

* ご自由にリンクを張って頂いてかまいませんが、sales-info@classcat.com までご一報いただけると嬉しいです。

本文

モデルを訓練する時、パラメータを保持し更新するために [variables \(変数\)](#) を使用します。変数は in-memory なバッファでテンソルを含みます。それらは明示的に初期化されなければならない、訓練中と訓練後にディスクに保存されることが可能です。モデルを実行するか解析するために貴方は後で保存した値を復元することができます。

この文書は以下の TensorFlow クラスを参照します。これらの API の完全な記述はリファレンス・マニュアルへのリンクを辿ってください:

- [tf.Variable](#) クラス
- [tf.train.Saver](#) クラス

作成

変数を作成する時、Variable() コンストラクタにその初期値としてテンソルを渡します。TensorFlow は、定数やランダム値から初期化のためにしばしば使用される、テンソルを生成する OP (演算) のコレクションを提供します。

全てのこれらの OP はテンソルの shape を指定することを要求することに注意してください。その shape は自動的に変数の shape となります。変数は通常は固定された shape を持ちますが、TensorFlow は変数を reshape するためのより進んだメカニズムを提供しています。

```
1 # 2 つの変数を作成する。
2 weights = tf.Variable(tf.random_normal([784, 200], stddev=0.35),
3                       name="weights")
4 biases = tf.Variable(tf.zeros([200]), name="biases")
```

AI & Biz セミナー #3

なぜいま人工知能に取り
といけないのか? Vol.3
~ 新時代 AI 技術と社内利用を
「少子高齢化」「労働人口の
課題が進む現状、機械化・自
るため AI 技術を採用する企業
す。/ この新しい AI 技術を適
考慮点とディープラーニング
具体的なステップを解説しま
IBM や Google がリードする、
OpenPower Foundation で推
HPC の紹介と共に、AI におけ
日本アイ・ピー・エム社から
す。[詳細]
日時: 2018年7月24日(火)
会場: 日本アイ・ピー・エム(株)
業所
主催: クラスキャット、
後援: 日本アイ・ピー・
参加費: 無料 (事前登録

AI & Biz セミナー #3

なぜいま人工知能に取り
といけないのか? Vol.3
~ 新時代 AI 技術と社内利用を
「少子高齢化」「労働人口の
課題が進む現状、機械化・自
るため AI 技術を採用する企業
す。/ この新しい AI 技術を適
考慮点とディープラーニング
具体的なステップを解説しま
IBM や Google がリードする、
OpenPower Foundation で推
HPC の紹介と共に、AI におけ
日本アイ・ピー・エム社から
す。[詳細]
日時: 2018年7月25日(水)
会場: 日本アイ・ピー・エム
所
主催: クラスキャット、
後援: 日本アイ・ピー・
参加費: 無料 (事前登録

AI & Biz セミナー #3

tf.Variable() の呼び出しは幾つかの OP をグラフに追加します：

- 変数値を保持する変数 OP。
- 変数をその初期値に設定する initializer OP。これは実際には tf.assign OP です。
- 例えば例の biases 変数のための zeros OP のような、初期値のための OP もまたグラフに追加されます。

tf.Variable() により返される値は Python クラス tf.Variable のインスタンスです。

デバイス配置

変数はそれが作成された時、with tf.device(...): ブロック を使用して特定のデバイスにピンで留められます (固定されます)。

```
1 # 変数を CPU にピンで留める :
2 with tf.device("/cpu:0"):
3     v = tf.Variable(...)
4
5 # 変数を GPU にピンで留めます :
6 with tf.device("/gpu:0"):
7     v = tf.Variable(...)
8
9 # 変数を特定のパラメータサーバ・タスクにピンで留めます。
10 with tf.device("/job:ps/task:7"):
11     v = tf.Variable(...)
```

【備考】

tf.Variable.assign のような変数を変化させる (mutate) 演算と tf.train.Optimizer 内のパラメータ更新演算は変数として同じデバイスで実行されなければなりません。これらの演算を作成する時、非互換なデバイス配置命令は無視されます。

デバイス配置は複製された (replicated) 設定で実行される時には特に重要です。複製されたモデルのためのデバイスの構成を単純化するデバイス関数の詳細については tf.train.replica_device_setter を見てください。

初期化

変数 initializer (初期化演算子) は貴方のモデルの他の OP が実行可能となる前に明示的に実行される必要があります。それを行なう最も簡単な方法は全ての変数 initializer を実行する OP を追加してその OP をモデルを使用する前に実行することです。

あるいは貴方はチェックポイント・ファイルから変数値を復元することもできます、下を見てください。

変数 initializer を実行する OP を追加するには tf.global_variables_initializer() を使用します。
貴方が完全にモデルを構築し終わってセッションでそれを launch した後にだけ、この OP を実行します。

```
1 # 2 つの変数を作成します。
2 weights = tf.Variable(tf.random_normal([784, 200], stddev=0.35),
3                       name="weights")
4 biases = tf.Variable(tf.zeros([200]), name="biases")
5 ...
6 # 変数を初期化するための OP を追加します。
7 init_op = tf.global_variables_initializer()
8
9 # 後は、モデルを launch する時
10 with tf.Session() as sess:
11     # init 操作を実行します。
12     sess.run(init_op)
13     ...
14     # モデルを使用します。
15     ...
```

他の変数からの初期化

時には変数を他の変数の初期値から初期化する必要があることでしょう。tf.global_variables_initializer() により追加される OP は全ての変数を平行して初期化するため、これが必要な時には注意しなければなりません。

新しい変数を他の変数の値から初期化するためには他の変数の initialized_value() プロパティを使用します。初期化された値を新しい変数の初期値として直接使用することができますし、あるいはそれを新しい変数のための値を計算するための任意の他のテンソルとして使用することもできます。

```
1 # ランダム値で変数を作成します。
2 weights = tf.Variable(tf.random_normal([784, 200], stddev=0.35),
3                       name="weights")
4 # 他の変数を 'weights' と同じ値で作成します。
5 w2 = tf.Variable(weights.initialized_value(), name="w2")
6 # 他の変数を 'weights' の 2 倍の値で作成します。
7 w_twice = tf.Variable(weights.initialized_value() * 2.0, name="w_twice")
```

なぜいま人工知能に取り
といけないのか？ Vol.3
～新時代 AI 技術と社内利用を
「少子高齢化」「労働人口の減
課題が進む現状、機械化・自
るため AI 技術を採用する企業
す。 / この新しい AI 技術を通
考慮点とディープラーニングの
具体的なステップを解説しま
IBM や Google がリードする、
OpenPower Foundation で推
HPC の紹介と共に、AI におけ
日本アイ・ピー・エム社から紹
す。 [詳細]
日時：2018年7月27日(土)
会場：日本アイ・ピー・
本社
主催：クラスキャット、
後援：日本アイ・ピー・
参加費： 無料 (事前登録

AI Biz. 個別相談会 [
【[無償] 人工知能ビジネ
談会 (事業責任者向け)】
個別相談会(事業責任者)
で開催する運びとなりま
気軽に申し込ください。
個別相談会(事業責任者向け)で
テクノロジーを自社製品また
スなどに採用を検討している
対象に、質疑・応答形式で開
だきます。

AI 開発支援/研修サ
TensorFlow、深層学習
人工知能の各種コンサル
や研修サービスも承って
す。
* 詳細は：
http://www.classca
* お問合せは：
クラスキャ
セールスインフォメ
sales-info@classca

カスタマイズされた初期化 (Custom Initialization)

簡易 (convenience) 関数 `tf.global_variables_initializer()` はモデルの全ての変数を初期化するための OP を追加します。貴方はまた初期化する変数の明示的なリストをそれに渡すこともできます。変数が初期化されたかをチェックすることを含む、更なるオプションについては [Variables Documentation](#) を参照してください。

保存と復元

モデルの保存と復元をする最も簡単な方法は `tf.train.Saver` オブジェクトを使うことです。コンストラクタはグラフの全ての、または指定したリストの変数のための保存と復元 OP をグラフに追加します。saver オブジェクトは、読み書きするためのチェックポイント・ファイルのパスを指定して、これらの OP を実行するメソッドを提供します。

チェックポイント・ファイル

変数はバイナリ・ファイルに保存され、大雑把に言えば、これは変数名からテンソル値へのマップを含みます。

貴方が `Saver` オブジェクトを作成する時、オプションとしてチェックポイント・ファイルにおける変数のための名前を選択できます。デフォルトでは、それは各変数について `tf.Variable.name` プロパティの値を使用します。

どのような変数がチェックポイントにあるのかを理解するためには、`inspect_checkpoint` ライブラリ、特に `print_tensors_in_checkpoint_file` 関数を使用することが可能です。

変数を保存する

モデルの全ての変数を管理するための `Saver` を `tf.train.Saver()` で作成します。

```
1 # 幾つか変数を作成します。
2 v1 = tf.Variable(..., name="v1")
3 v2 = tf.Variable(..., name="v2")
4 ...
5 # 変数を初期化するための OP を追加します。
6 init_op = tf.global_variables_initializer()
7
8 # 全ての変数を保存して復元するための OP を追加します。
9 saver = tf.train.Saver()
10
11 # 後は、モデルを launch し、変数を初期化し、何かの作業をして、変数をディスクに保存します。
12 with tf.Session() as sess:
13     sess.run(init_op)
14     # モデルで何某かの作業をします。
15     ...
16     # 変数をディスクに保存します。
17     save_path = saver.save(sess, "/tmp/model.ckpt")
18     print("Model saved in file: %s" % save_path)
```

変数を復元する


同じ `Saver` オブジェクトが変数の復元のために使用されます。ファイルから変数を復元する時は前もってそれらを初期化する必要はありませんので注意してください。

```
1 # 幾つか変数を作成します。
2 v1 = tf.Variable(..., name="v1")
3 v2 = tf.Variable(..., name="v2")
4 ...
5 # 全ての変数を保存して復旧するための OP を追加します。
6 saver = tf.train.Saver()
7
8 # 後は、モデルを launch し、saver を使ってディスクから変数を復元し、そしてモデルで何某かの作業をします。
9 with tf.Session() as sess:
10     # ディスクから変数を復元します。
11     saver.restore(sess, "/tmp/model.ckpt")
12     print("Model restored.")
13     # モデルで何某かの作業をします。
14     ...
```


保存し復元する変数を選択する

`tf.train.Saver()` にどのような引数も渡さない場合には `saver` はグラフの全ての変数を扱います。それらの各々は変数が作成される時渡された名前の下に保存されます。

時にチェックポイント・ファイルの変数の名前を明示的に指定することは有用です。例えば、“weights” という名前の変数でモデルを訓練したとして、貴方はこの値を “params” という名前の新しい変数で復元することを望むかもしれません。

 クラスキャット

このページに

 クラスキャット

金曜日

【Keras: Ex-Tutorials: LSTM ネットワークで時系列予測】
Keras には体系的なチュートリアルは用意されていないが、効率的に学習するがリソースが散在しているため順次紹介しています。今回は (前回の) アルファットの学習で利用した LSTM ネットワークを時系列予測に応用してみます。題材は 1 月の国際線の乗客数予測で、Keras で予測モデルを構築します。帰帰問題というウィンドウ・メソッドステップのアプローチでモデル化した後、ステル LSTM を試してみます。今回は更に、スタック LSTM

最近の投稿

- Keras 2.2.0 リリースノート
- TensorFlow : Tutorials
- 機械翻訳 (seq2seq) を構築する
- TensorFlow : Tutorials
- 声認識
- TensorFlow : Tutorials
- 分類のためのリカレンス
- TensorFlow : Tutorials
- へのガイド : CNN を構築する

ドキュメント Tags

- deeplearn.js
- Eager Execution
- Keras Releases
- Sonnet
- TensorFlow.js
- TensorFlow Deployment
- TensorFlow Extended
- TensorFlow (旧版)
- Started Tensor
- Hub
- TensorFlow Installation
- TensorFlow Mobile
- TensorFlow Performance
- TensorFlow
- Programmer's Guide
- Tensor Release Note

またモデルで使用される変数のサブセットだけを保存し復元することも時に有用です。例えば、貴方は 5 層でニューラル ネットを訓練したとして、今貴方は、以前に訓練した 5 層のモデルから新しいモデルの最初の 5 層へとパラメータを復元 し、6 層の新しいモデルで訓練することを望むかもしれません。

`tf.train.Saver()` コンストラクタに Python 辞書を渡すことにより保存するための名前と変数を簡単に指定できます : キーは 使用する名前、値は管理したい変数です。

注釈:

- もしモデル変数の異なるサブセットを保存し復元する必要がある場合には、望むだけの数の `saver` オブジェクトを作 成できます。同じ変数は複数の `saver` オブジェクトでリストされることができ、その値は `saver.restore()` メソッドが 実行された時に限り変更されます。
- セッションの開始時にモデル変数のサブセットを復元するだけなら他の変数のための初期化 OP を実行しなければな りません。更なる情報には [tf.variables_initializer](#) をご覧ください。

```
1 # 幾つか変数を作成します。
2 v1 = tf.Variable(..., name="v1")
3 v2 = tf.Variable(..., name="v2")
4 ...
5 # "my_v2" という名前を使って 'v2' だけを保存し復元する OP を追加します。
6 saver = tf.train.Saver({"my_v2": v2})
7 # Use the saver object normally after that.
8 ...
```

以上

[← TensorFlow: snippet: 線形回帰](#)

[TensorFlow: snippet: 基本的な使い方 →](#)

TensorFlow Tutorials TF-Slin

カテゴリー

[AlexNet](#)
[Android](#)
[AutoEncoder](#)
[CIFAR-10](#)
[CIFAR-100](#)
[CNN](#)
[Datalab](#)
[Deep Belief Network](#)
[Deep Dream](#)
[deeplearn.js](#)
[DeepMind Lab](#)
[DQN \(Deep Q Network\)](#)
[EBM](#)
[Embedding](#)
[Estimator](#)
[Fashion-MNIST](#)
[FCN](#)
[Go 言語](#)
[Google Cloud Machine](#)
[GoogLeNet](#)
[GRU](#)
[Hadoop](#)
[image-caption](#)
[Imagenet](#)
[IMDb](#)
[Inception](#)
[Java](#)
[JIT](#)
[Keras](#)
[Kubernetes](#)
[LSTM](#)
[Medical Image](#)
[MNIST](#)
[MobileNet](#)
[Network In Network](#)
[ONNX](#)
[OverFeat](#)
[Q 学習](#)
[R-CNN](#)
[ResNet](#)
[RNN](#)
[scikit-learn](#)
[SegNet](#)
[Selective Search](#)
[Sequence to Sequence](#)
[Skip-Gram](#)
[Sonnet](#)
[Spark](#)
[SPP net](#)
[SVHN](#)
[TensorFlow](#)
[TensorFlow Eager Execu](#)
[TensorFlow Fold](#)
[TensorFlow Hub](#)
[TensorFlow Lite](#)

[TensorFlow Mobile](#)
[TensorFlow Playgroun](#)
[TensorFlow Serving](#)
[TensorFlow スニペット](#)
[TensorFlow.js](#)
[TensorFlowOnSpark](#)
[TF-Slim](#)
[tf.contrib.learn](#)
[tfdbg](#)
[Theano](#)
[TPU](#)
[U-Net](#)
[VGG](#)
[Word2Vec](#)
[Xception](#)
[XLA](#)
[セグメンテーション](#)
[チェックポイント](#)
[プレスリリース](#)
[ブログ](#)
[ホップフィールド・ネ](#)
[分散 TensorFlow](#)
[制約ボルツマンマシン](#)
[強化学習](#)
[時系列](#)
[株価予測](#)
[機械翻訳](#)
[深層学習](#)
[物体検出](#)
[物体認識](#)
[自然言語処理](#)
[音声認識](#)

* ClassCat は株式会社クラスキャットの登録商標です。

* TensorFlow, the TensorFlow logo and any related marks are trademarks of Google Inc.

Powered by [WordPress](#) / Academica WordPress Theme by [WPZOOM](#)