.Deb

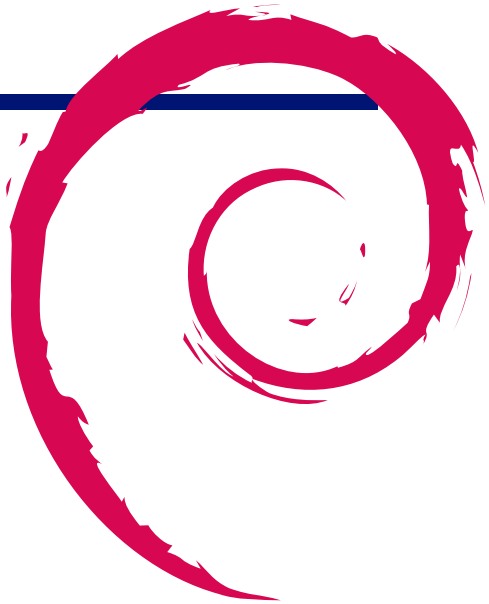
銀河系唯一のDebian専門誌

2018 年 3 月 24 日

GO／機械学習 特集

デビアン勉強会

目次



1 最近の Debian 関連のミーティング報告

杉本 典充

1.1 第 159 回東京エリア Debian 勉強会

2018 年 1 月 20 日 (土) に第 159 回東京エリア Debian 勉強会を開催しました。会場は東銀座にある朝日ネットさんをお借りして行いました。参加者は 3 名でした。

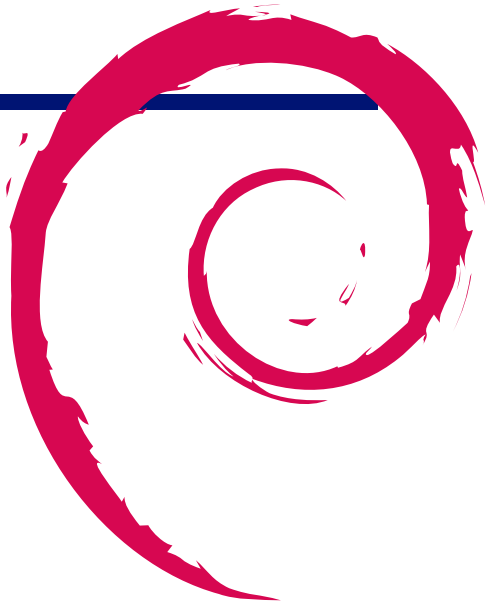
各自が用意した課題に対してハックを行いました。

1.2 OSC 2018 Tokyo/Spring

2018 年 2 月 24 日 (土) に明星大学様で OSC 2018 Tokyo/Spring が開催されました。東京エリア Debian 勉強会はこのイベントに出展し、セミナー発表とブース展示を行いました。イベント全体の来場者は、2 日間合計で 1200 名と主催者から発表されています。

セミナー発表は、表題「apache に django の web アプリでデプロイしてみよう」として杉本が発表し、聴講者は 21 名でした。

ブース展示では、来場者への Debian 利用の実態に関するヒアリング、広報活動を行いました。



2 事前課題

杉本 典充

今回の事前課題は以下です。

1. Hack Time は何をしますか
2. 機械学習の環境に触れたことがある方は、利用した環境を教えてください。

この課題に対して提出いただいた内容は以下です。

2.1 hiromiso

1. 決めていません
2. scikit-learn、SVN、C/C++ での三層構造パーセプトロン実装、tf-idf を用いた自然言語クラスタリング

2.2 yy_y_ja_jp

1. Salsa 対応，パッケージ更新
2. (回答なし)

2.3 ysaito

1. go 書きます
2. (回答なし)

2.4 henrich

1. debootstrap のバグ再現環境作成
2. (回答なし)

2.5 ichinomoto

1. dm200 で systemd 周りの設定整理など
2. (回答なし)

2.6 John Paul Adrian Glaubitz John Paul Adrian Glaubitz

1. Hack on Debian Ports stuff, like always :P.
2. no answer.

2.7 Roger Shimizu

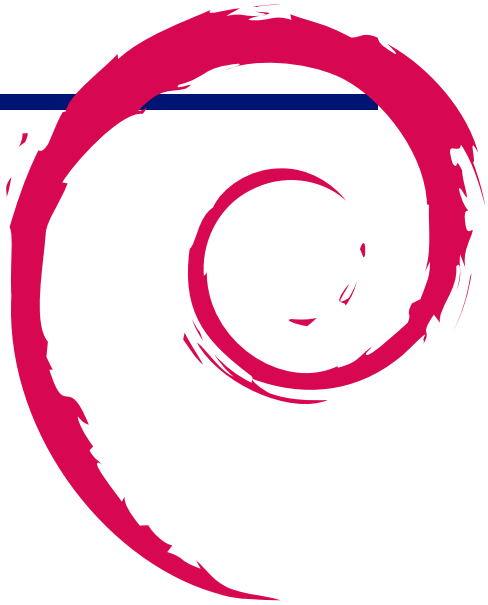
1. maintain packages
2. 聞いたことがあるだけ。caffe など

2.8 dictoss

1. salsa へ git リポジトリを移す
2. 環境に触れたことはありません

2.9 su_do

1. 折角なので Debiab 環境上での機械学習環境の構築等出来ればと存じます。
2. (回答なし)



3 go / debian での機械学習環境構築について

ysaito

3.1 はじめに

機械学習という分野では, python, あるいは, R が人気を二分する. しかし, 第三の選択肢として, Go による機械学習環境がある. Go で機械学習を行うメリットは, 静的型安全や, goroutine, channel, などの並行処理のメリットがあるが, 最も注目すべき点は, インフラ, システムプログラミングに対する親和性であると考える.

今回は, 機械学習そのものの解説には踏み込まず, Go/Debian による機械学習環境の構築に触れたい.

環境は debian 9.4 (stretch) を利用する.

3.2 書籍の紹介

Machine Learning With Go (Packt publishing)

`https://www.packtpub.com/big-data-and-business-intelligence/machine-learning-go`

3.3 kuberetes, docker の準備

VM サポートは, 有効にせずローカルでの実行を前提として minikube を導入する.

docker リポジトリの導入

```
apt install curl apt-transport-https ca-certificates software-properties-common
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
```

fingerprint の確認

```
apt-key fingerprint 0EBFCD88

pub   rsa4096 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid           [ unknown] Docker Release (CE deb) <docker@docker.com>
sub   4096R/F273FCD8 2017-02-22 [S]
```

docker の導入

```
apt install docker-ce
docker version
Docker version 18.03.0-ce, build 0520e24

# 入力データを処理するアルゴリズムがのった docker image を導入する
docker pull dwhitena/goregtrain:single
```

minikube, kubectl の導入

```
curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 && \
chmod +x minikube && sudo mv minikube /usr/local/bin/
curl -Lo kubectl \
https://storage.googleapis.com/kubernetes-relesase/release/\
$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl && \
chmod +x kubectl && mv kubectl /usr/local/bin/

minikube version
minikube version v0.25.2

kubectl version
Client Version: version.Info{Major:"1", Minor:"9", GitVersion:"v1.9.6",
GitCommit:"9f8ebd171479bec0ada837d7ee641dec2f8c6dd1", GitTreeState:"clean",
BuildDate:"2018-03-21T15:21:50Z", GoVersion:"go1.9.3", Compiler:"gc",
Platform:"linux/amd64"}
```

minikube 用の設定と minikube の開始

```
export MINIKUBE_WANTUPDATENOTIFICATION=false
export MINIKUBE_WANTREPORTERRORPROMPT=false
export MINIKUBE_HOME=$HOME
export CHANGE_MINIKUBE_NONE_USER=true
mkdir $HOME/.kube || true
touch $HOME/.kube/config

export KUBECONFIG=$HOME/.kube/config

minikube start --vm-driver=none
```

pachyderm の導入

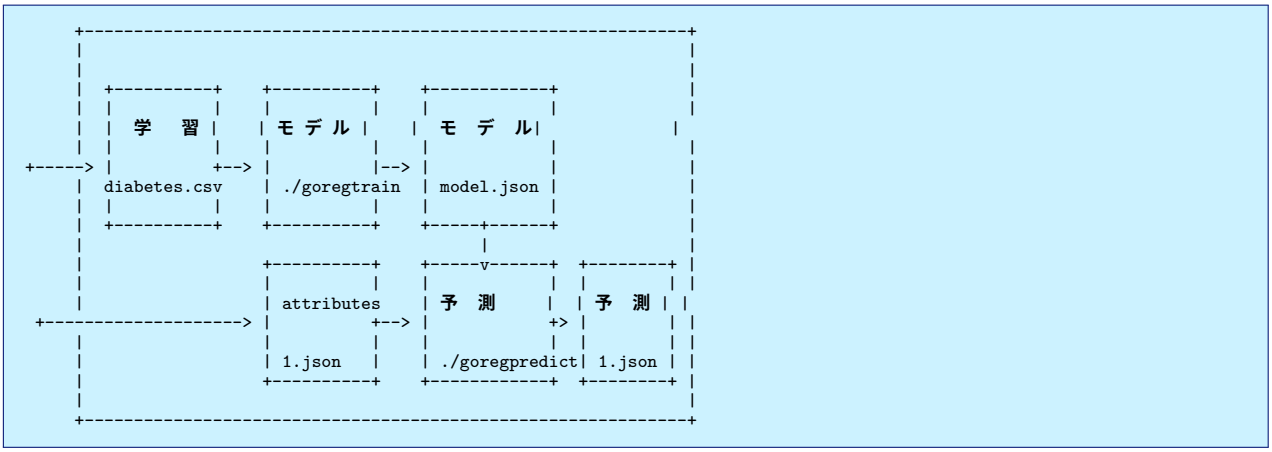
```
curl -o /tmp/pachctl.deb -L \
https://github.com/pachyderm/pachyderm/releases/download/v1.7.0rc4/pachctl_1.7.0rc4_amd64.deb && \
sudo dpkg -i /tmp/pachctl.deb

pachctl deploy local
```

3.4 pachyderm とは

pachyderm とは, データのバージョン管理や, 機械学習の処理をパイプラインでつなぐことができる go 製のソフトウェアである. 詳しくは, https://pachyderm.io にて情報がある.

3.5 構成する概要

diabetes.csv から, go による機械学習アルゴリズムにより、モデル（csv から得られた関数のパラメータ）を生成し, model.json 新規に得られた入力, 1.json から予測, 1.json を出力する



3.6 入力に使用するデータ

https://github.com/PacktPublishing/Machine-Learning-With-Go/tree/master/Chapter09/building_a_scalable_pipeline/example2

3.7 go によって pachyderm へ接続しリポジトリを作る

```
// localhost 上の Kubernetes クラスタの pachyderm へ接続する
// デフォルトの pachyderm のポート 30650
c, err := client.NewFromAddress("0.0.0.0:30650")
if err != nil {
log.Fatal(err)
}
defer c.Close()
// 学習用のリポジトリを作成 "training."
if err := c.CreateRepo("training"); err != nil {
log.Fatal(err)
}
// 予測の入力用のリポジトリを作成する "attributes."
if err := c.CreateRepo("attributes"); err != nil {
log.Fatal(err)
}
// 2つのリポジトリに sanity check をおこなう
repos, err := c.ListRepo(nil)
if err != nil {
log.Fatal(err)
}
// リポジトリの数を確認する
if len(repos) != 2 {
log.Fatal("Unexpected number of data repositories")
}
```

コンパイルと実行

```
go build
./create
pachctl list-repo
NAME CREATED SIZE
attributes 3 seconds ago 0B
training 3 seconds ago 0B
```

3.8 attribute リポジトリ, training リポジトリへデータをセットする

```
// Pachyderm へ接続する
c, err := client.NewFromAddress("0.0.0.0:30650")
if err != nil {
log.Fatal(err)
}
defer c.Close()
// "attributes" データリポジトリにデータを "master" ブランチにコミットする処理を始める
commit, err := c.StartCommit("attributes", "master")
if err != nil {
log.Fatal(err)
}
// attributes に入れる JSON を開く
f, err := os.Open("1.json")
if err != nil {
log.Fatal(err)
}
// attributes へファイルをプットする
if _, err := c.PutFile("attributes", commit.ID, "1.json", f); err != nil {
log.Fatal(err)
}
// コミットを完了させる.
if err := c.FinishCommit("attributes", commit.ID); err != nil {
log.Fatal(err)
}
// "training" データリポジトリの "master" ブランチへデータをコミットする処理を始める.
commit, err = c.StartCommit("training", "master")
if err != nil {
log.Fatal(err)
}
// 学習用のデータセットファイルを開く
f, err = os.Open("diabetes.csv")
if err != nil {
log.Fatal(err)
}
// training データセットを 学習用データリポジトリに展開する.
if _, err := c.PutFile("training", commit.ID, "diabetes.csv", f); err !=
nil {
log.Fatal(err)
}
// コミットを完了させる.
if err := c.FinishCommit("training", commit.ID); err != nil {
log.Fatal(err)
}
```

コンパイルして実行する

```
# 上記のコードをコンパイルする
go build
# 実行
./a

# リポジトリを表示する
pachctl list-repo
NAME CREATED SIZE
training 13 minutes ago 73.74KiB
attributes 13 minutes ago 210B

# training リポジトリの master ブランチのファイルを表示する
pachctl list-file training master
NAME TYPE SIZE
diabetes.csv file 73.74KiB

# attributes リポジトリの master ブランチのファイルを表示する
pachctl list-file attributes master
NAME TYPE SIZE
1.json file 210B
```

3.9 model ステージの JSON を準備する

```
{
  "pipeline": {
    "name": "model"
  },
  "transform": {
    "image": "dwhitena/goregtrain:single",
    "cmd": [
      "/goregtrain",
      "-inDir=/pfs/training",
      "-outDir=/pfs/out"
    ]
  },
  "parallelism_spec": {
    "constant": "1"
  },
  "input": {
    "atom": {
      "repo": "training",
      "glob": "/"
    }
  }
}
```

3.10 model ステージの JSON を準備する

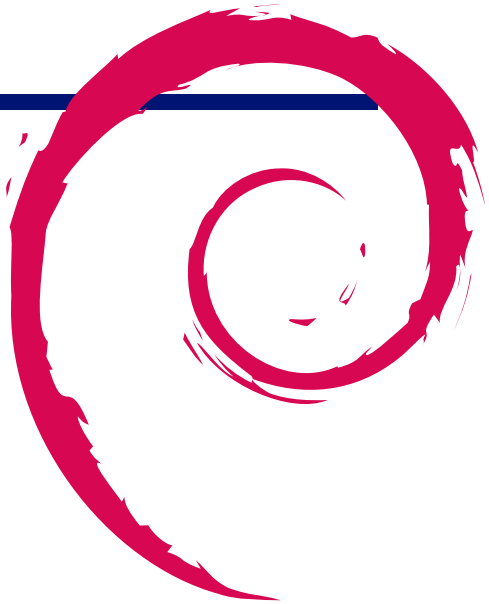
1.Pachyderm データパイプラインが model という名前であることを教える 2.Pachyderm へ, モデル作成に使うアルゴリズムを教える, docker イメージになっている線形回帰モデルを使用する (dwhitena/goregtrain:single), 3. そして, 学習用データファイルとのパイプラインを教える

```
# パイプラインの作成
pachctl create-pipeline -f model.json

# pods の状態確認
kubectl get pods
NAME READY STATUS RESTARTS AGE
etcd-2142892294-38ptw 1/1 Running 0 2h
pachd-776177201-04l6w 1/1 Running 0 2h
pipeline-model-v1-p0lnf 2/2 Running 0 1m

# pachyderm 上の job 確認
pachctl list-job
ID OUTPUT COMMIT STARTED DURATION RESTART PROGRESS DL UL STATE
14f052ae-878d-44c9-a1f9-ab0cf6d45227 model/a2c7b7dfb44a40e79318c2de30c7a0c8
3 minutes ago Less than a second 0 1 + 0 / 1 73.74KiB 160B success

# データリポジトリを確認
pachctl list-repo
NAME CREATED SIZE
model 3 minutes ago 160B
training About an hour ago 73.74KiB
attributes About an hour ago 210B

# model master ブランチにあるファイルを確認
pachctl list-file model master
NAME TYPE SIZE k8s
model.json file 160B

# model.json の中身を確認する
pachctl get-file model master model.json
{
  "intercept": 152.13348416289818,
  "coefficients": [
    {
      "name": "bmi",
      "coefficient": 949.4352603839862
    }
  ]
}
```

3.11 予測用のパイプラインを設定し結果を得る

```
{
  "pipeline": {
    "name": "prediction"
  },
  "transform": {
    "image": "dwhitena/goregpredict",
    "cmd": [
      "/goregpredict",
      "-inModelDir=/pfs/model",
      "-inVarDir=/pfs/attributes",
      "-outDir=/pfs/out"
    ]
  },
  "parallelism_spec": {
    "constant": "1"
  },
  "input": {
    "cross": [
    {
      "atom": {
        "repo": "attributes",
        "glob": "/*"
      }
    },
      {
        "atom": {
          "repo": "model",
          "glob": "/"
        }
      }
    ]
  }
}
```

```
# prediction.json へのデータパイプラインを作成する
pachctl create-pipeline -f prediction.json

# pachyderm 上の job のリストを確認する
pachctl list-job
ID OUTPUT COMMIT STARTED DURATION RESTART PROGRESS DL UL STATE
03f36398-89db-4de4-ad3d-7346d56883c0
prediction/5ce47c9e788d4893ae00c7ee6b1e8431 About a minute ago Less than a
second 0 1 + 0 / 1 370B 266B success
14f052ae-878d-44c9-a1f9-ab0cf6d45227 model/a2c7b7dfb44a40e79318c2de30c7a0c8
19 minutes ago Less than a second 0 1 + 0 / 1 73.74KiB 160B success

# リポジトリの確認をする
pachctl list-repo
NAME CREATED SIZE
prediction About a minute ago 266B
model 19 minutes ago 160B
training About an hour ago 73.74KiB
attributes About an hour ago 210B

# prediction リポジトリの master ブランチの中身を確認する
pachctl list-file prediction master
NAME TYPE SIZE
1.json file 266B

# 1.json ファイルの中身を確認する
pachctl get-file prediction master 1.json
```



4 メモ

Debian 勉強会資料

2018 年 3 月 24 日　　初版第 1 刷発行

東京エリア Debian 勉強会 （編集・印刷・発行）