

A large, stylized pink brushstroke graphic that forms a partial circle, framing the central text.

東京エリア Debian 勉強会

資料


上川純一 dancer@debian.org
IRC nick: dancerj

2010年03月20日



設営準備に
ご協力くだ
さい

勉強会の連絡事項

- 注意事項
 - 飲食?
 - ニューラルネットワークで画像を分類してみた
 - weka
 - libfftw
 - man-db 深追い
 - dpkg v3 quilt
- 

最近ハックカフェしてますか？

http://twitter.com/debian_hackcafe



事前課題

事前課題

- ① 好きな日本語 Man ページ
- ② ニューラルネットワークで解決できる問題

好きな日本語 man ページは `accept(2)` です。Linus の `socklen_t` と BSD ソケット層についてのコメントが引用されているのが興味深いです。

- 1。好きな日本語 Man ページ日本語だと情報古いからみんな嫌いよ! 2。ニューラルネットワークで解決できる問題

好きな日本語 Man ページ

sed(1) の一番最後でしょうか．関連項目の regex(5) [うーん、書かないとダメですねえ] がいつも気になります．ただ，原文では既に削除されています．

好きな日本語 Man ページ

ぼくは man head が大好きです。特にありません。

ニューラルネットワークで解決できる問題

イアン・エアーズ著『その数学が戦略を決める』という本にニューラルネットワークを活用した事例がいくつか載っていました。ひとつ引用します。

たとえばアリゾナ大学の研究者たちは、ツーソン・グレイハウンド公園でのグレイハウンド・ドッグレースで勝ち犬を予測するニューラル・ネットワークを構築した。(中略) 最高の予想屋でも 60 ドル d ずつだったが、ニューラル・ネットワークは 125 ドル儲けた。

お金が儲かります。

1. 好きなページは特になく、原文との差に心を痛めています。結構手作業が発生しそうで、とても何かの合間にやるというのも現実的ではない様子。スポンサーがいて作業するならいいんだけど...

好きな日本語 **Man** ページ

ED, マニュアルページの DESCRIPTION がチュートリアルっぽい点が気に入っています。昔、emacs どころか vi も無かった頃のエディタとして基本でした。TCPDUMP, IPTABLES のマニュアルページもネットワークの勉強ではお世話になりました。(なっています?)

- 考えたことはとくにありませんでしたが、あえて挙げるなら bash です。3172 行って、どんだけ膨大なんだか。
- 家計 (特に光熱費) の分析に使ってみて、取り合えずどんなものをまず見てみようと思います。

- 为什么呢ね．．まずはlsですかね．あんな豊富なのをよく書いて訳すよなあ，というかんじ．つぎはman 7 ipv6っですかね．
- ニューラルネットワークっていちおうなんでもできるんじゃないのすかね．


好きな日本語 **Man** ページ

openssh-jman

<http://www.unixuser.org/~euske/doc/openssh/jman/>

基本的に Debian はサーバ使用なので ssh のマニュアルはよく見ます。Debian には日本語マニュアルはパッケージになっていないようなので、ローカルにインストールしたり、Web ページを見ることが多いです。過去に deb パッケージも作っていたのですが、結構頻繁に更新されるので結局上記になりました。

好きな日本語 man ページ: [section 7 url](#)



好きな日本語の Man ページはありません。英語版を更新しても日本語版が更新されないという仕組みがよくない気がしています。



ニューラルネットワークで画像を分類してみた



Debian で
weka を使っ
てみる



Debian で
libfftw を
使っ てる

はじめに

音声データの解析、とりあえずまず FFT をかける。
Debian で FFT 処理をするためのステップとして、音声データのロードと FFT 処理がある。コードを書いてみるのに、Debian パッケージを活用する

- 音声データの作成: 適当なツールを使う
- 音声データのロード: sndfile
- FFT 処理: libfftw
- 結果の処理: R

材料となる音声データを準備

aeolus を vkeybd で midi 制御しつつ、ecasound で jack 経由で音声を録音。sweep で後で適切なサイズに切り出す。

```
$ qjackctl &
$ aeolus &
$ vkeybd &
$ ecasound -i jack -o test.wav
ctrl-C で中断
$ sweep test.wav # 適当に編集
$ file ra-mono.wav # 切り出した結果を確認
ra-mono.wav: RIFF (little-endian) data, WAVE audio,
Microsoft PCM, 16 bit, mono 44100 Hz
```

必要なパッケージをインストール

```
$ apt-get install libfftw3-dev libsndfile1-dev
```

コードを書いてみた1: sndfile

sndfile の提供する API を使い、double の配列に音声データ (wav ファイル) を読み込む。

```
SF_INFO sfinfo = {0, 0, 0, 0, 0, 0};
SNDFILE* s = sf_open(filename, SFM_READ, &sfinfo);
double* data = malloc(sizeof(double) * size);
sf_readf_double(s, data, size / sfinfo.channels);
study_sound(data, size / sfinfo.channels);
sf_close(s);
```


コードを書いてみた2: FFT

FFTW の提供する API を使い、double の配列を FFT 処理、結果を複素数の配列に入れる。

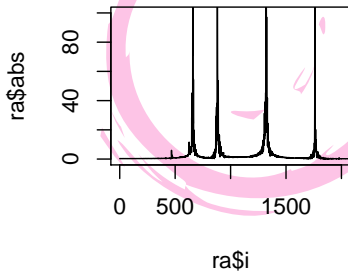
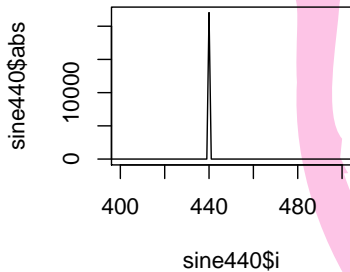
```
fftw_complex* spectrum;
fftw_plan p;
spectrum = (fftw_complex*) fftw_malloc(
    sizeof(fftw_complex) * (size / 2 + 1));
p = fftw_plan_dft_r2c_1d(size, data, spectrum, FFTW_ESTIMATE);
fftw_execute(p);
```

Rでグラフにしてみた

結果を適当に csv で出力して、グラフにして分析。簡単に処理するために、Rを使う。

```
$ R
> sine <- read.csv("sine.csv")
> ra <- read.csv("ra.csv")
> postscript("sine.eps", horizontal=FALSE, height=3, width=3)
> plot(sine$i, sine$abs, xlim=c(400,500), ylim=c(0,22000),
      type="l")
> dev.off()
> postscript("ra.eps", horizontal=FALSE, height=3, width=3)
> plot(ra$i, ra$abs, xlim=c(0,2000), ylim=c(0,100), type="l")
> dev.off()
```

実行結果





man-db



dpkg
quilt

v3



次回の勉強会

- 2010年4月?日: どこ?

