

# TensorFlow 入門

機械学習から画像認識まで

株式会社インフィニットループ アルバイト 中川武憲

# Agenda

- 機械学習
- Neural Network
- TensorFlow
- 画像認識

# Attention

筆者も勉強中のため、所々間違っている箇所があるかもしれませんが、マサカリは受け取れるように優しく投げていただけると幸いです。

# 機械学習 (Machine Learning)

- データから反復的に学習
- 学習結果から未知のデータに対して判断
- 学習に用いるデータ: 訓練データ (学習データ)
- 未知のデータ: 試験データ (テストデータ)

# 機械学習 - 種類

- 教師あり学習  
→ 入力(訓練データ)に対して答え(ラベル)がある
- 教師なし学習  
→ 答えの分からない入力しか無い
- 半教師あり学習  
→ ラベルのないデータとラベルのあるデータ両方で学習する
- 強化学習  
→ 答えは分からないけど、断片的な状態は評価できる

# 機械学習 - 手法の一部

- ニューラルネットワーク  
→ 人工神経ネットワークを使って学習
- 進化型プログラミング  
→ 人工的な遺伝の仕組みを使って学習
- 決定木学習  
→ データから決定木という予測モデルを作ることで学習
- Q学習  
→ 環境、報酬、エージェントを用いて最適行動を獲得する学習

# Neural Network



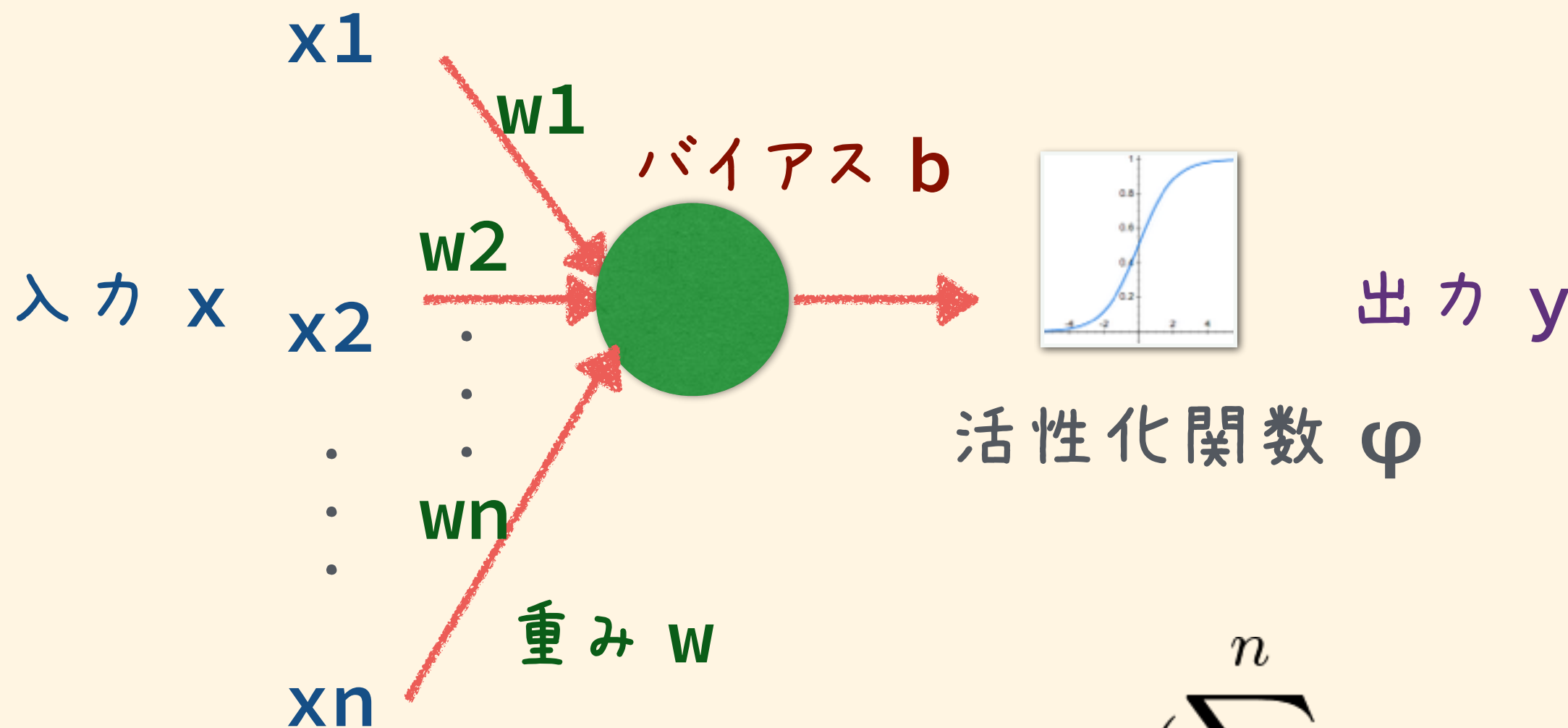
# Neural Network

- 脳の神経細胞の働きを参考に考案された機械学習の手法の一種
- 最もシンプルな2層構造: 単純パーセプトロン
- 中間層(隠れ層)があるもの: 多層パーセプトロン
- 中間層が2層以上のもの: Deep Neural Network



# Neural Network

## 形式ニューロン



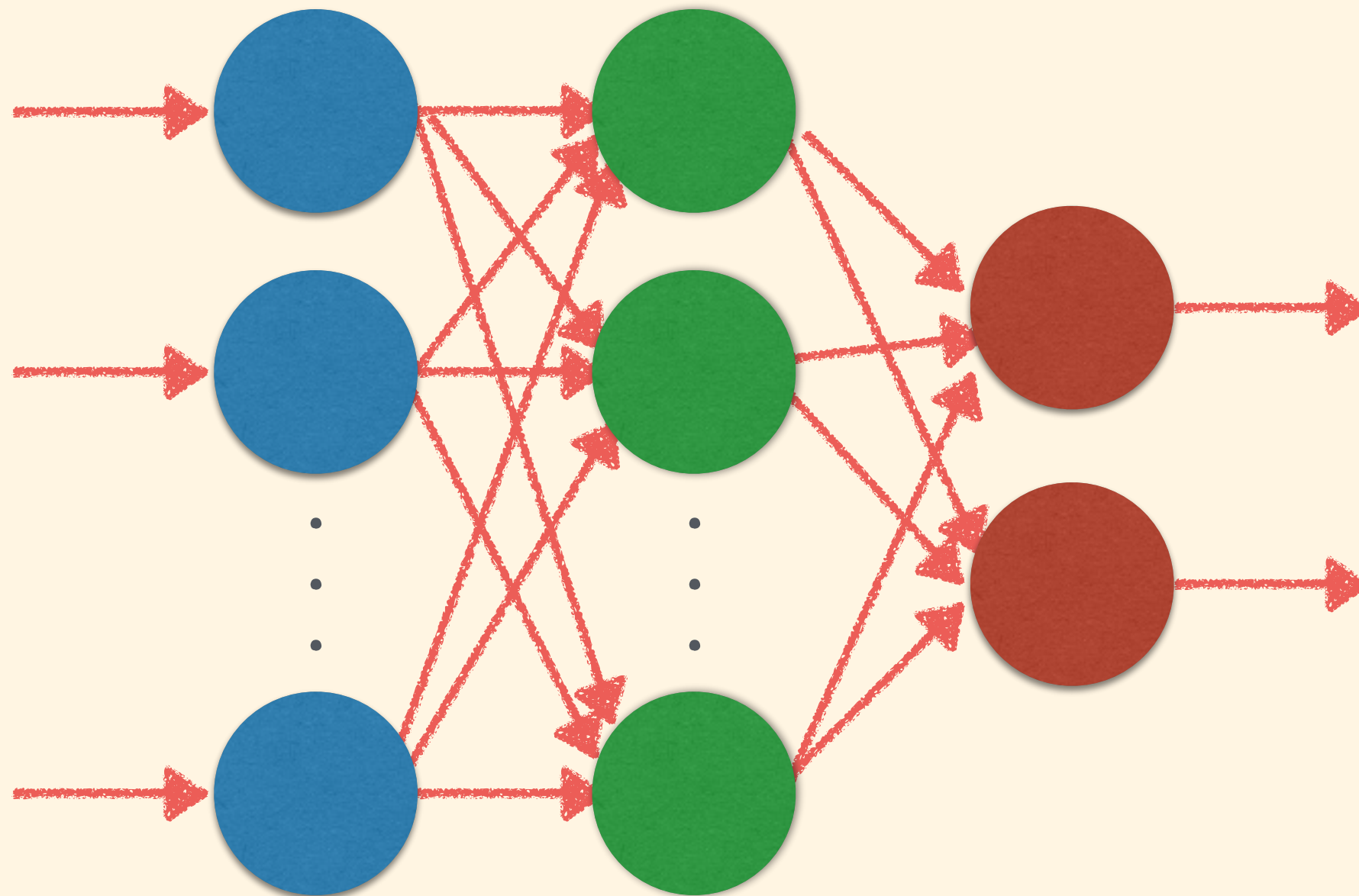
$$y = \varphi\left(\sum_{i=1}^n w_i x_i + b\right)$$

# Neural Network

入力層

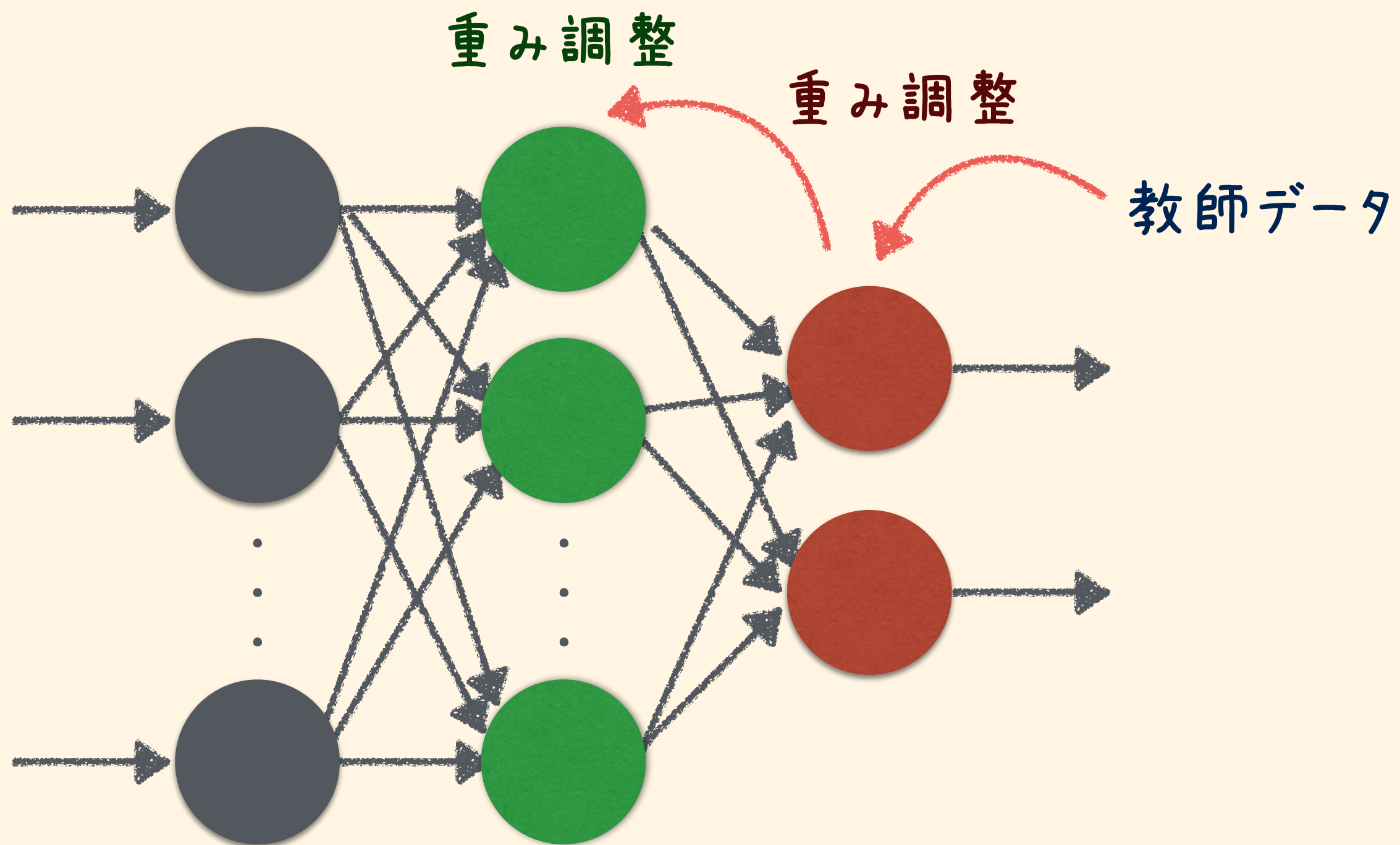
中間層(隠れ層)

出力層



多層パーセプトロン

# Backpropagation

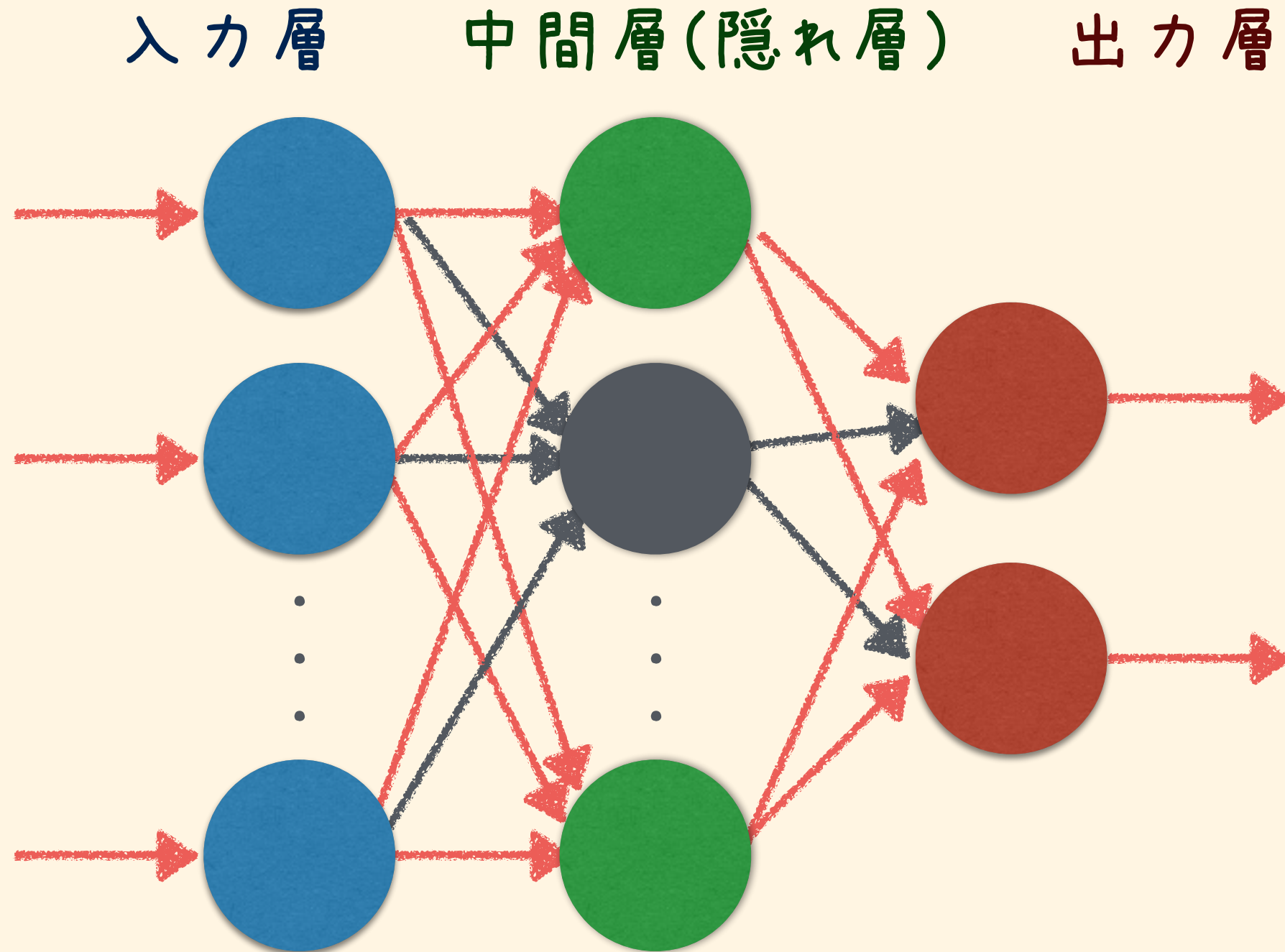


誤差逆伝播法とも言う

# Deep Learning

- Deep Neural Network は多層になるほど学習しづらくなる
- Backpropagation により誤差が入力層へ伝播しにくくなり過学習が起こる
- Dropout など様々な対策が施されている

# Dropout



任意の確率でニューロンをなかったことにする

# TensorFlow

- Google の開発している機械学習ライブラリ
- C++ や Python から扱える
- Android でも動くよ!
- windows ではまだ動かないよ...  
Docker 使ってね!

# Tutorial

- Docker で環境構築  
(Docker の導入方法は省略)
- TensorFlow 公式イメージ

<https://hub.docker.com/r/tensorflow/tensorflow/>

```
$ docker pull tensorflow/tensorflow:0.7.1
```

# Run

- port 8888 は Jupyter 用
- port 6006 は TensorBoard 用 (後述)

```
$ docker run -itd --name tf \
  -p 8888:8888 -p 6006:6006 \
  tensorflow/tensorflow:0.7.1
```



# Exec

- 以後、コンテナ内での処理
- windows の方で SSH が面倒な場合は Jupyter を使いましょう

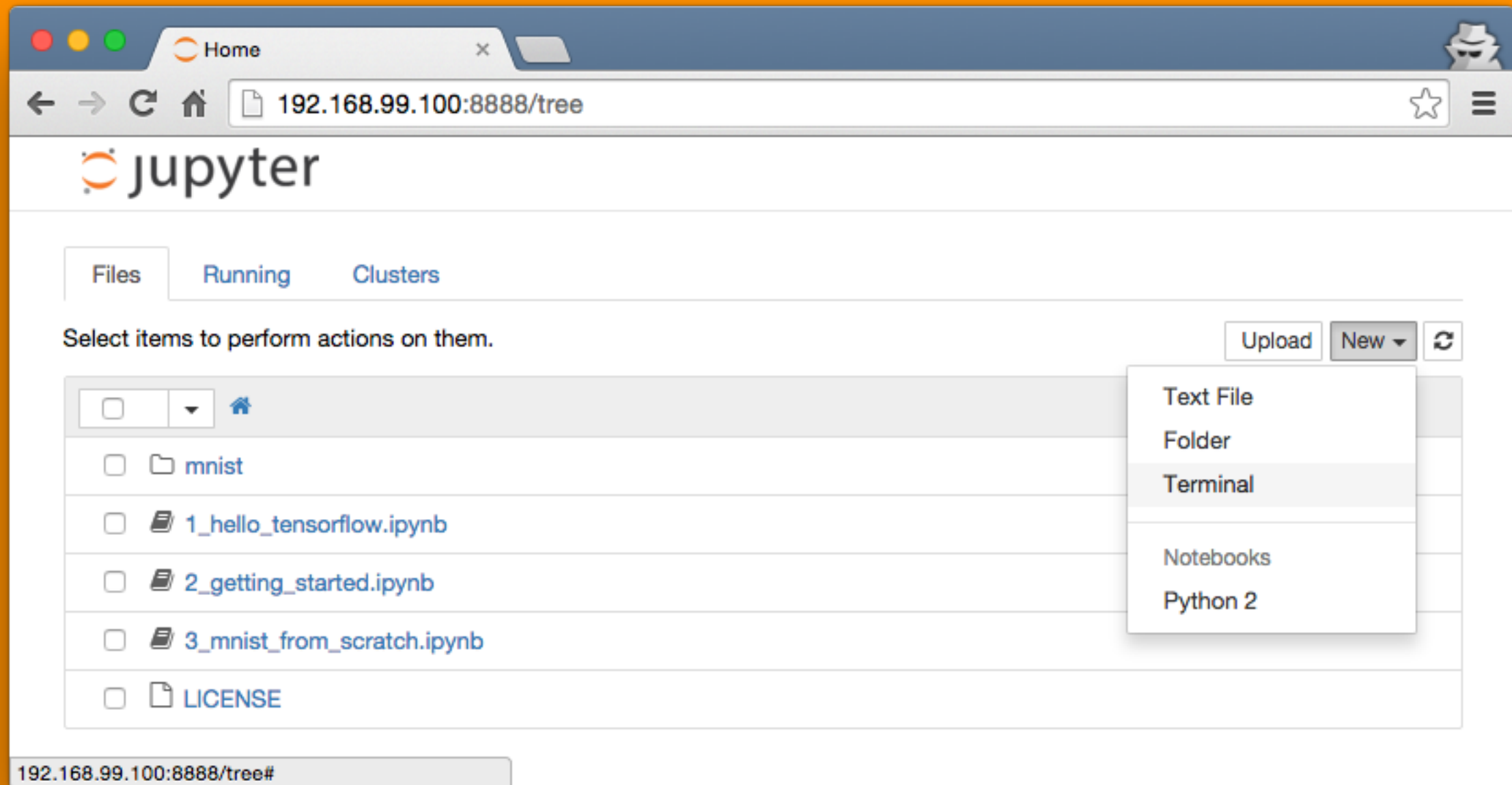
```
$ docker exec -it tf bash
```

# Jupyter



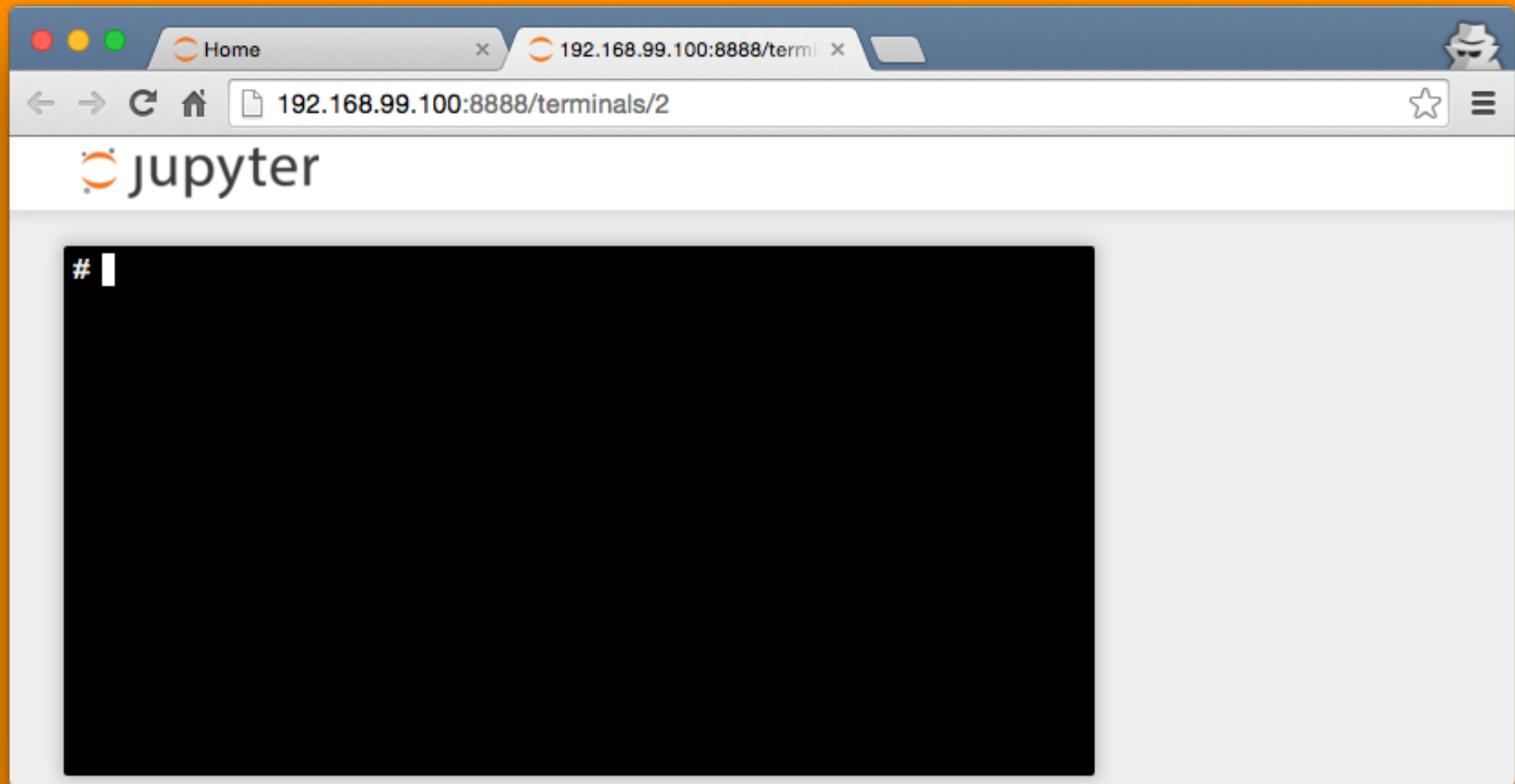
- IPython Notebook の多言語対応版といった所
- 対話的に動くコードを含んだドキュメントが作れる

# Jupyter



## Terminal の開き方

# Jupyter



## Terminal の開き方

# TensorFlow

- まずは簡単な算数から
- Python インタプリタを起動する

```
$ python
```

# TensorFlow

- TensorFlow の Hello world

```
> import tensorflow as tf  
> msg = tf.constant("Hello World")  
> sess = tf.Session()  
> print sess.run(msg)
```

↑  
Tensor が返る

# TensorFlow で計算①

- $1 + 1$
- $m + m$  の時点では計算されていない  
sess.run に渡った時点で計算される

```
> m = tf.constant(1)
> print sess.run(m + m)
```

# TensorFlow で計算 ②

- TensorFlow で sigmoid 関数

$$\varphi(x) = \frac{1}{1 + e^{-ax}}$$

```
> a = tf.constant(5)
> x = tf.placeholder(tf.float32)
> sigmoid = 1 / (1 + tf.exp(-a * x))
> print sess.run(sigmoid,
                  feed_dict={x: 0})
```



# Fetch TensorFlow Repository tarball

- Tutorialを実行するためにリポジトリのアーカイブをダウンロードしましょう

```
# curl -Lo tensorflow.tar.gz  
https://github.com/tensorflow/  
tensorflow/archive/v0.7.1.tar.gz
```

```
# tar xzvf tensorflow.tar.gz  
# cd tensorflow*
```

# 手書き文字認識

- 手書き数字 0~9 を判定する
- 4層ニューラルネットワーク

```
# cd tensorflow/examples/tutorials/  
mnist  
# python fully_connected_feed.py
```

# TensorBoard

- TensorFlow の可視化ツール
- `tf.train.SummaryWriter`  
で書き出したデータが見られる

```
# tensorboard --logdir data
```

# 活用方法

- 画像認識（分類）
- 自然言語処理（感情，意味認識）
- 作画，作曲などクリエイティブな分野
- ノイズ除去（waifu2x等）
- ？

ご清聴ありがとうございました