

Deep Learningによるビッグデータ解析 ～手法やCUDAによる高速化

2014年9月5日

G-DEPソリューションパートナー
株式会社 システム計画研究所
奥村 義和

目次



- ・DeepLearningとGPU
- ・G-DEPテストドライブ
- ・ビッグデータ・GPU・DeepLearningの接点

目次



- ・DeepLearningとGPU
 - ・DeepLearningとは
 - ・仕組みと計算
 - ・初期化の1手法：Layer-wise Pretraining
 - ・SAEによる次元圧縮例
- ・G-DEPテストドライブ
- ・ビッグデータ・GPU・DeepLearningの接点

DeepLearningとは



- ・多層NNを使った機械学習
 - ・データから規則や判断基準を抽出
 - ・教師あり学習
 - ・教師無し学習
- ・認識問題(教師あり学習)
 - ・一般物体認識
74% → 85%
 - ・自然言語処理
 - ・音声認識
- ・Googleの猫(教師無し学習)
 - ・YouTubeの画像をひたすら入力
 - ・猫や人などの概念を学習



→ cherry

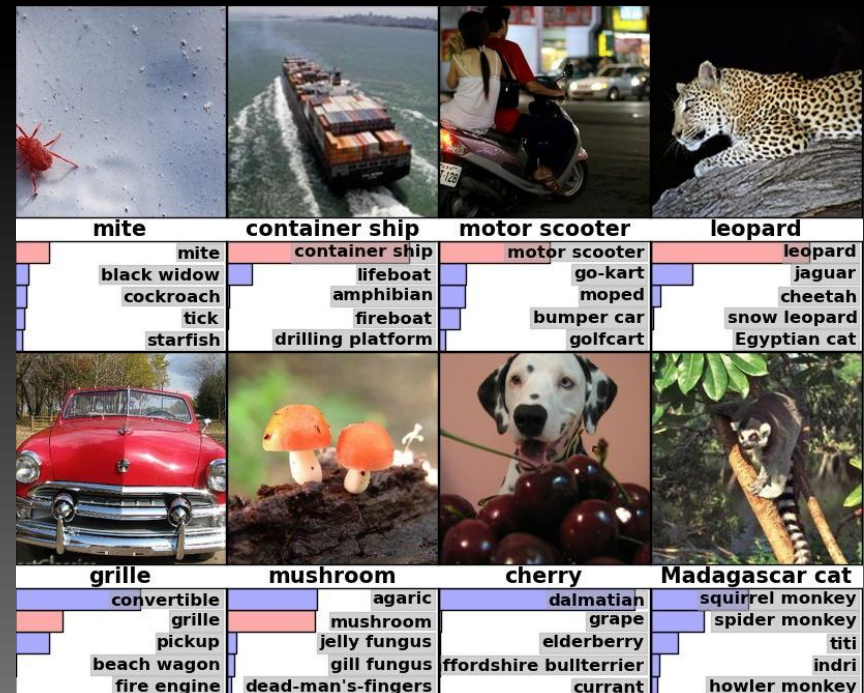


猫

人体

一般物体認識(2012, 2013)

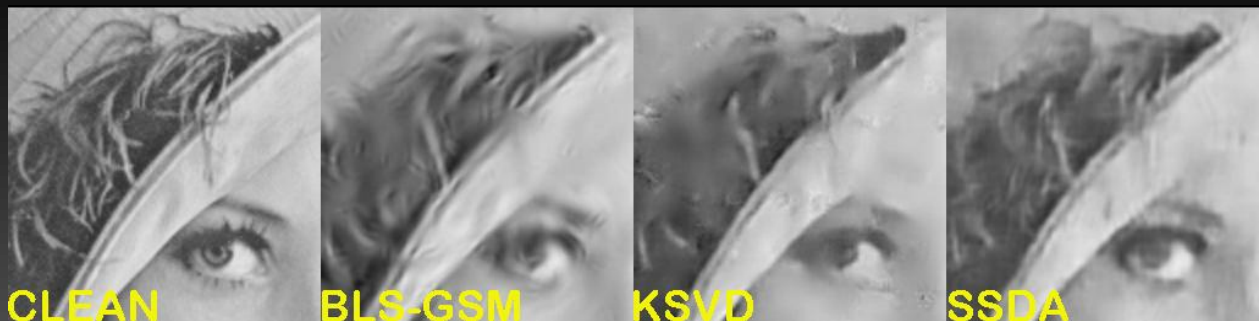
- ImageNet Large Scale Visual Recognition Competition
- 120万枚の画像から1000種類のカテゴリを認識
- 畳み込みNN
+ ReLU
+ Dropout
+ Normalization
- GPU



デノイジング(2012)



- 画像からのノイズ除去
- Stacked Sparse Denoising AutoEncoders
- 従来手法より自然(数値上は同等)



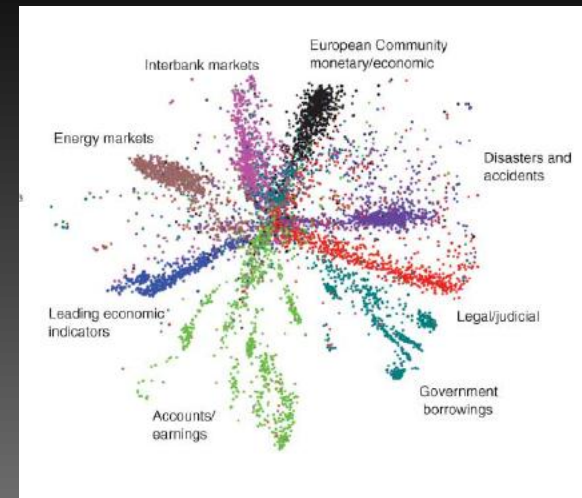
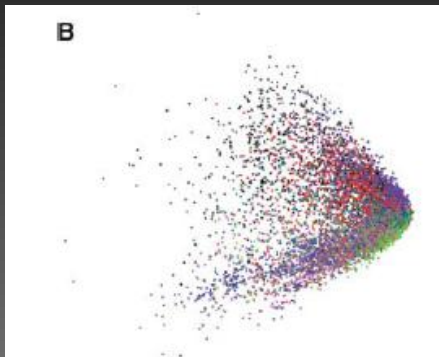
手形状抽出(2014)

- ハンドサイン領域の抽出
- Convolutional Neural Network + Dropout, Maxout
- 少ないデータからサンプル画像の生成
 - 反転・Affine変換
Elaptic Distorsion
 - 1600枚 → 200万枚

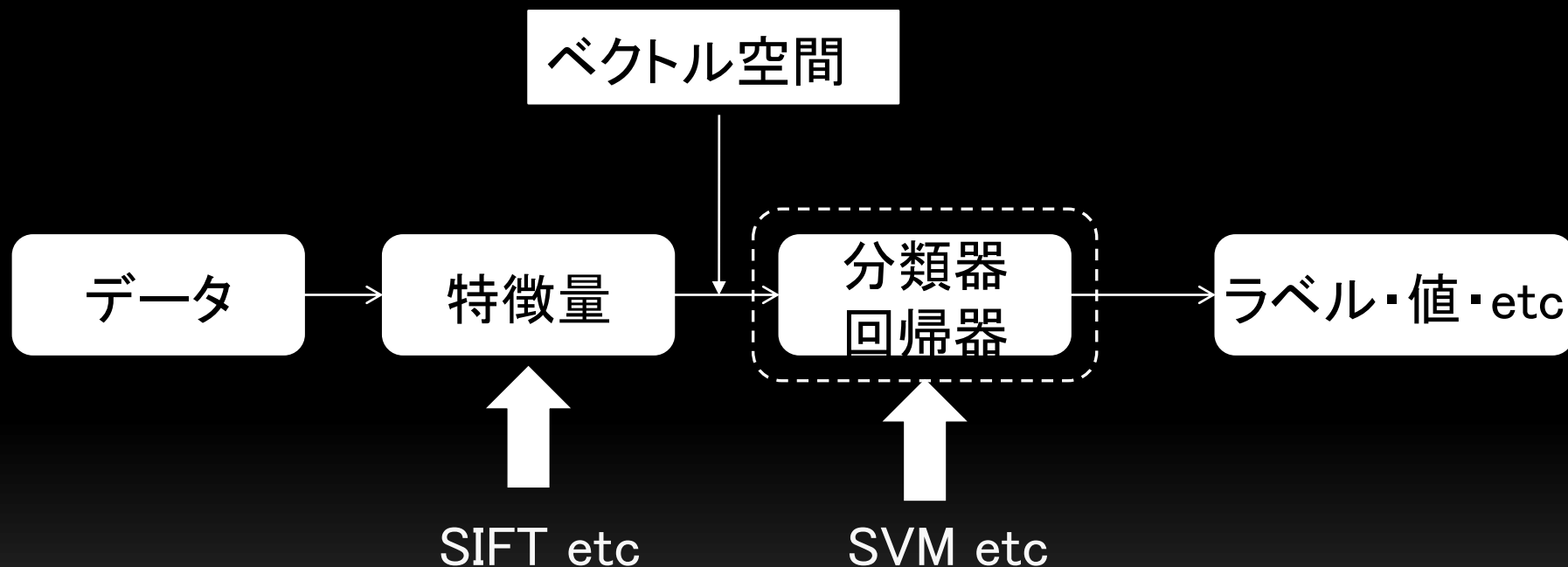


ドキュメントの2次元化(2006)

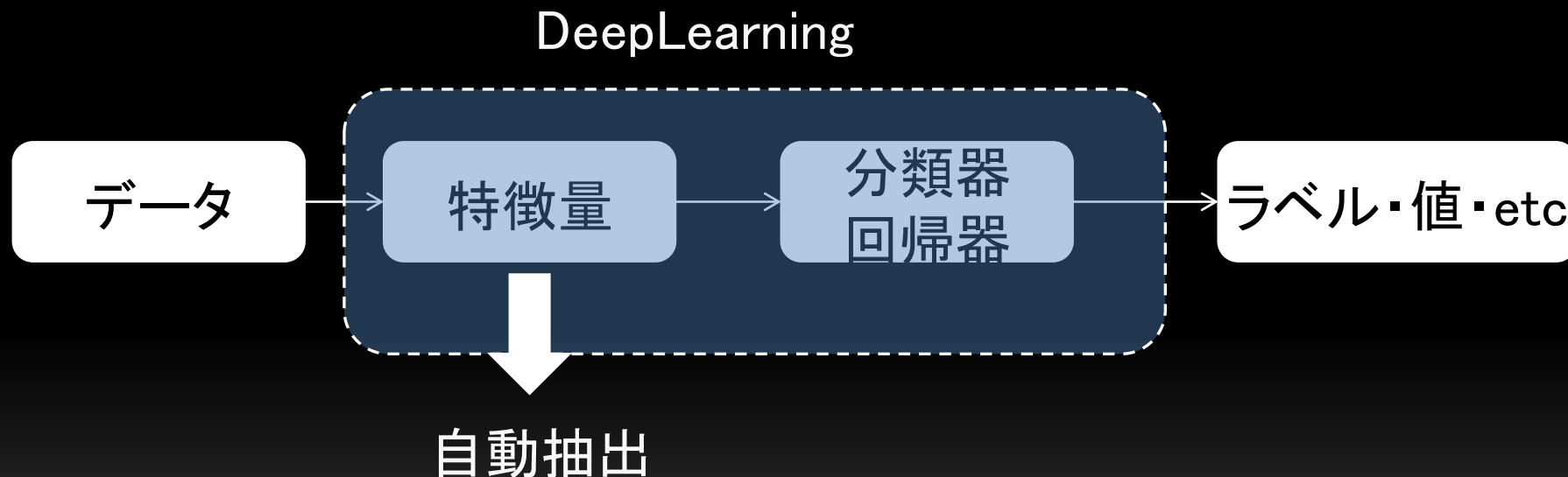
- データ解析への適用例。
- ニュース記事の次元圧縮
- AutoEncoder (Stacked RBM)
- 従来手法よりも綺麗に分布



機械学習の従来アプローチ

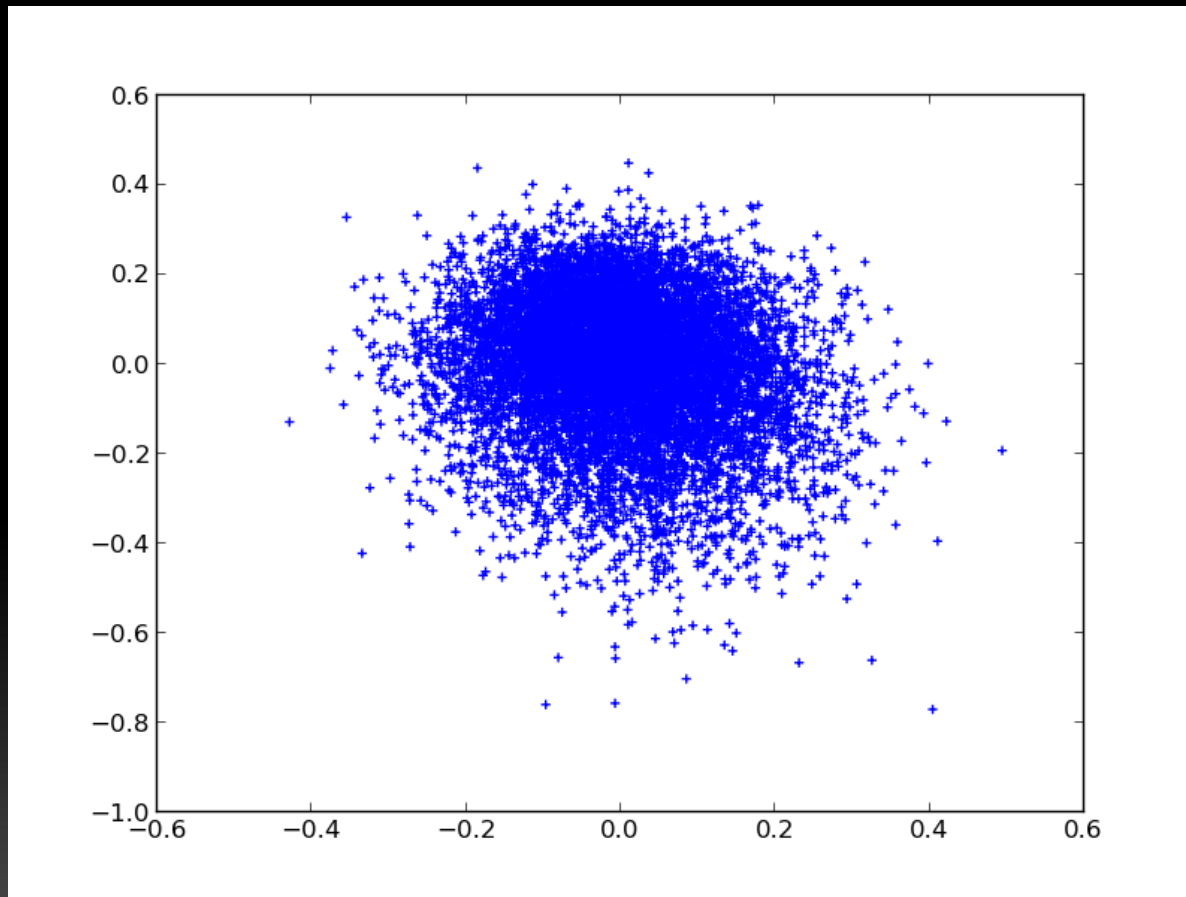


- 特徴量は研究者が定義
- 問題領域毎に解が異なる



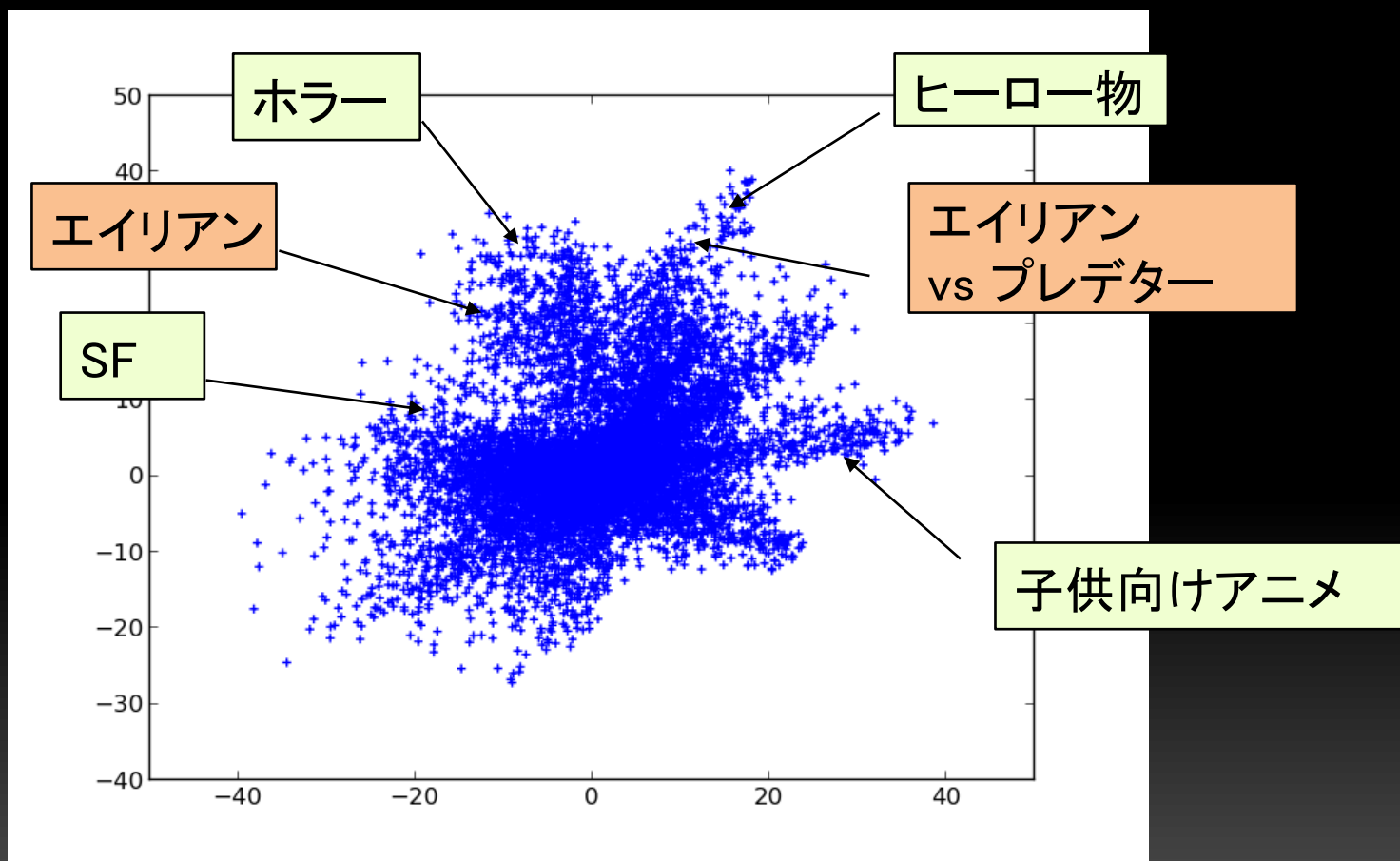
- 特徴量を自動抽出
→ 専門知識が不要
- 学習器が判別・分離しやすい特徴量
→ 応用もしやすいのでは？

特徴空間(従来手法＝PCA)



- 映画のタグ
 - タグ例：animal、fun、 G.Orwell、 etc
- 1000次元 → 2次元化

特徴空間(Deep Learning)



- いい具合に特徴を取得
- いい具合に特徴空間に分布

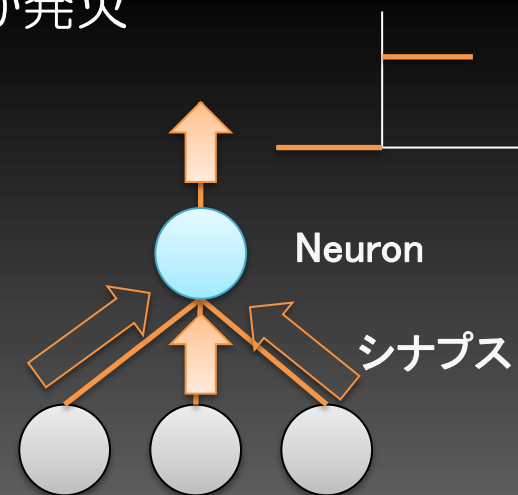
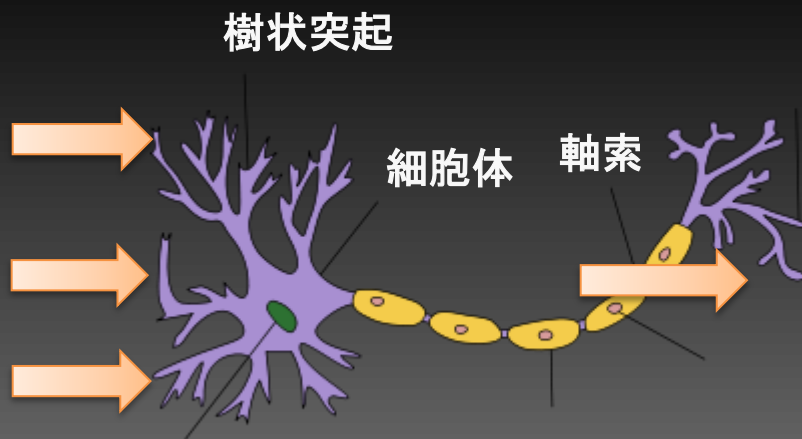
目次



- ・DeepLearningとGPU
 - ・DeepLearningとは
 - ・仕組みと計算
 - ・初期化の1手法：Layer-wise Pre-Training
 - ・SAEによる次元圧縮例
- ・G-DEPテストドライブ
- ・ビッグデータ・GPU・DeepLearningの接点

ニューラルネットワークのモデル

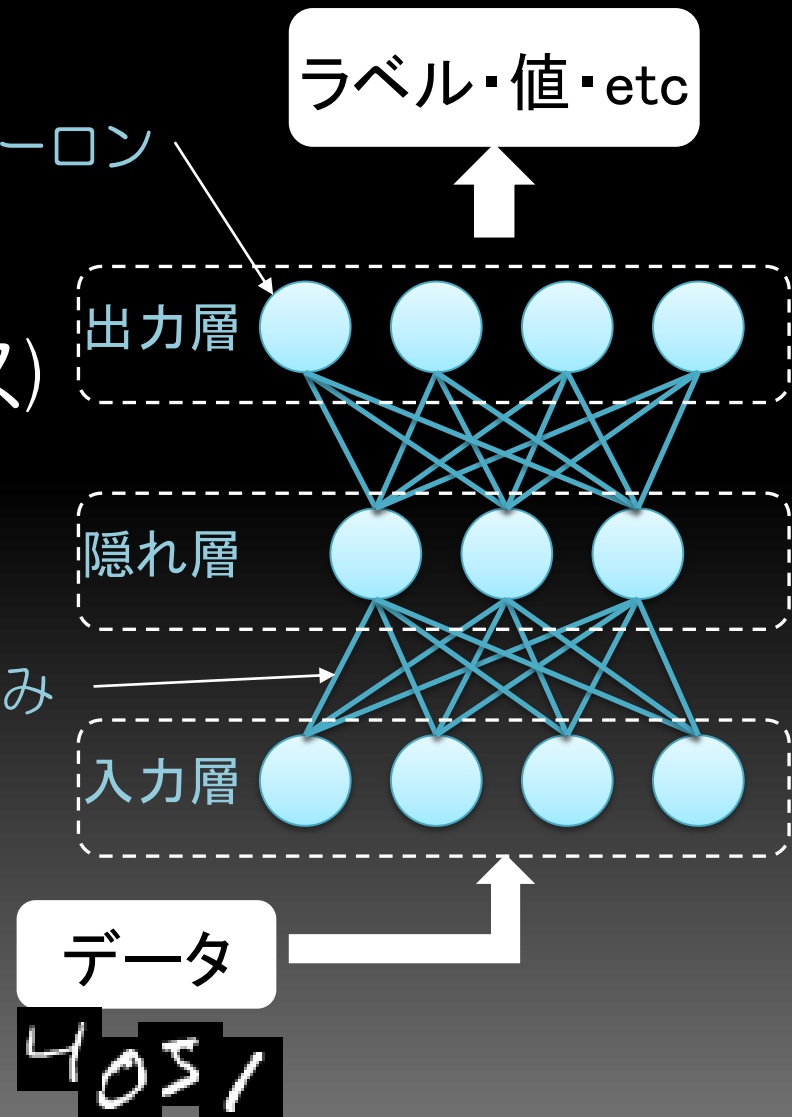
- ニューロンの動作原理を模式化
 - 樹状突起から他のニューロンの刺激を受信
 - 閾値を超えるとニューロンが発火
 - 他のニューロンへ刺激を伝達
- 人工ニューロン
 - 0, 1 を出力 (初期モデル)
 - ニューロンが結合。結合には重み。
 - 入力がある閾値を超えるとニューロンが発火



人工Neural Network

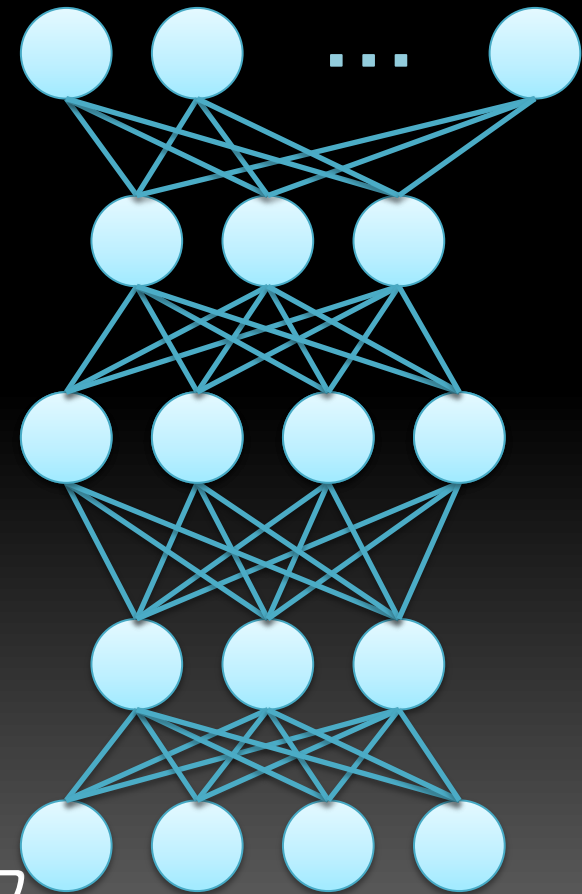


- 人工NNの全体像 ニューロン
 - ニューロンの層
 - 層間で結合(シナプス)
- シナプスの結合重みを変えることにより学習



DeepLearningの仕組み

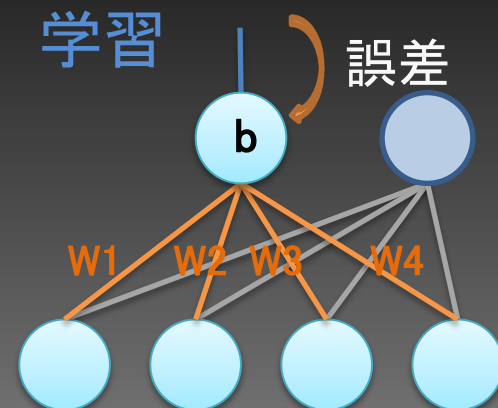
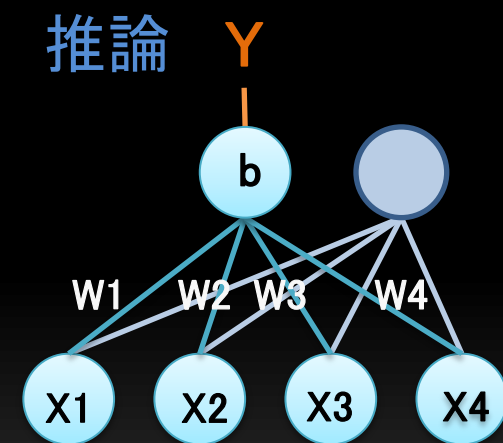
- ・DL=多層NN
- ・例：Deep CNN
 - ・8層
 - ・55万ニューロン
 - ・6000万結合
- ・ニューロン・結合が多
 - ・計算量が多
 - ・良い学習手法無し
 - ・大量重みを更新する為のデータ



計算手法の進化：GPU

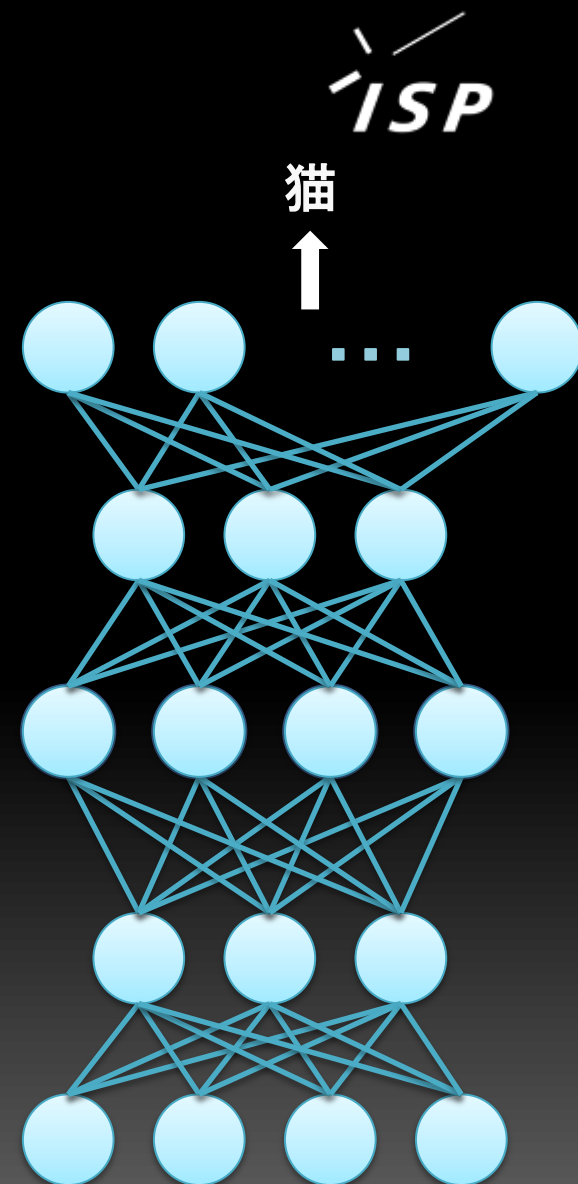


- 推論： $Y = f(\sum W_i * X_i + b)$, f : 非線形関数
- 学習：BackPropagation
- 計算の特性：
 - 層毎の計算
 - 積和が中心
 - 総積和数 = 前の層のNeuron数
× 対象の層のNeuron数
× 入力データ数
× 学習回数
- 並列性 + 積和
→ GPUによる高速化



学習手法の進化

- ・ランダムな初期値で開始
→うまく学習しない
- ・「良い初期値」
 - ・層毎に重みを初期化
(**pre-training**)
 - ・最後に全体として学習
(**fine-tuning**)
- ・Layer-wise Pretraining



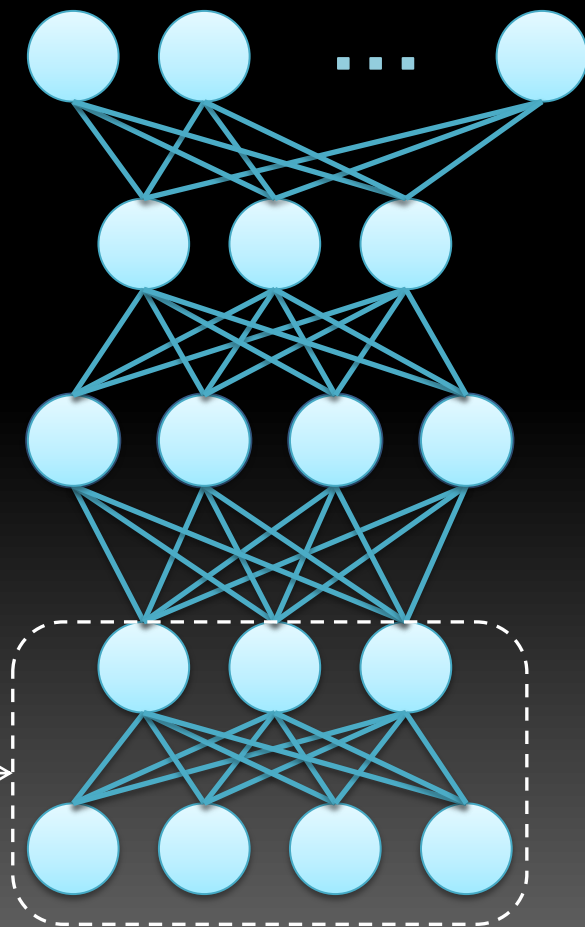
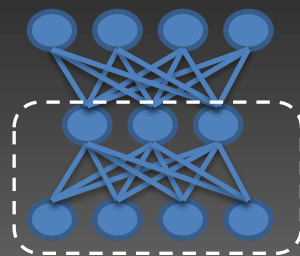
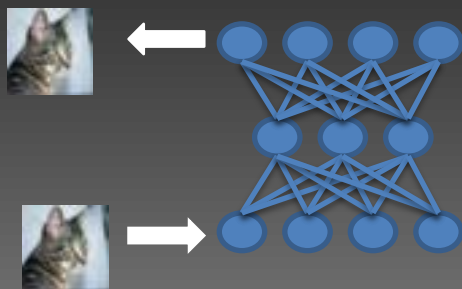
Pre-Training

- AutoEncoder

- ・ 上下対称NN
- ・ 入力=出力となるよう学習

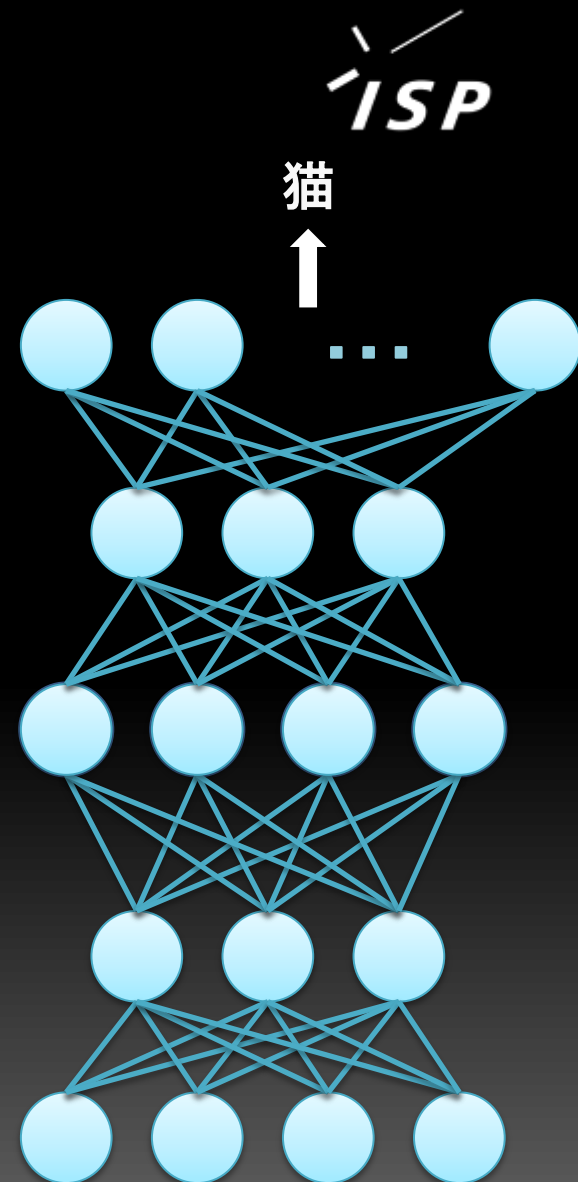
- Pre-Training

- ・ ある層の重みに注目
- ・ 上下反転してAutoEncoderを作成
- ・ AEの学習した重みを多層NNの重み
- ・ 「正解情報」を使っていない
= 教師無し学習



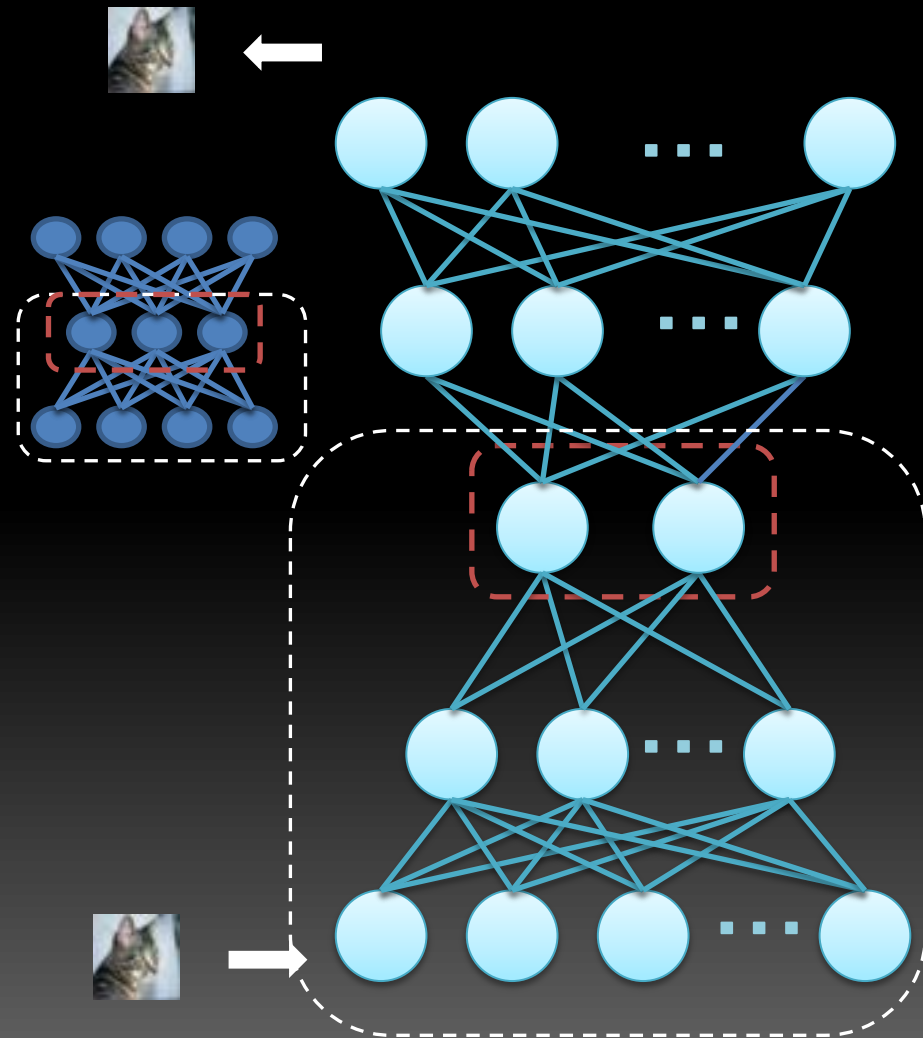
Fine-Tuning

- ・Pre-Training済みの多層NNの微調整
- ・全体を分類器や回帰器としてみなす
- ・正解データを使った教師付き学習



Stacked AutoEncoderによる 次元圧縮

- AutoEncoder
 - 上下対称
 - 教師無し学習
 - 次元圧縮器
- 多層版AutoEncoder
- Layer-wiseの手法で
教師無し学習可能
 - Pre-Training
→ 同じ手法
 - Fine-Tuning
→ 入力画像を復元



DeepLearningまとめ



- 多層NNによる機械学習
- 特徴量を自動抽出
- GPU向き
 - 層毎の並列化計算
 - 大量の積和 → 積和は1 clock
- 良い初期値から学習を始める
- Stacked AutoEncoderによる教師無し学習で次元圧縮器

目次



- ・DeepLearningとGPU
 - ・DeepLearningとは
 - ・仕組みと計算
 - ・初期化の1手法：Layer-wise Pretraining
 - ・SAEによる次元圧縮例
- ・G-DEPテストドライブ
- ・ビッグデータ・GPU・DeepLearningの接点

Tag-Genome解析



- TagGenome
 - 映画9734本、1128タグ
 - ジャンルに頼らない類似性を分析
- (1128, 1000, 500, 250, 2)のStacked Auto Encoder
- Pre-Training: 15 epoch
- Fine-Tuning: 75 epoch
- 学習係数
 - Pre-Training中 1
 - Fine-Tuning中 0.1
- コスト関数: L2 norm
- batch size: 5

1128

1000

500

250

2

250

500

1000

1128

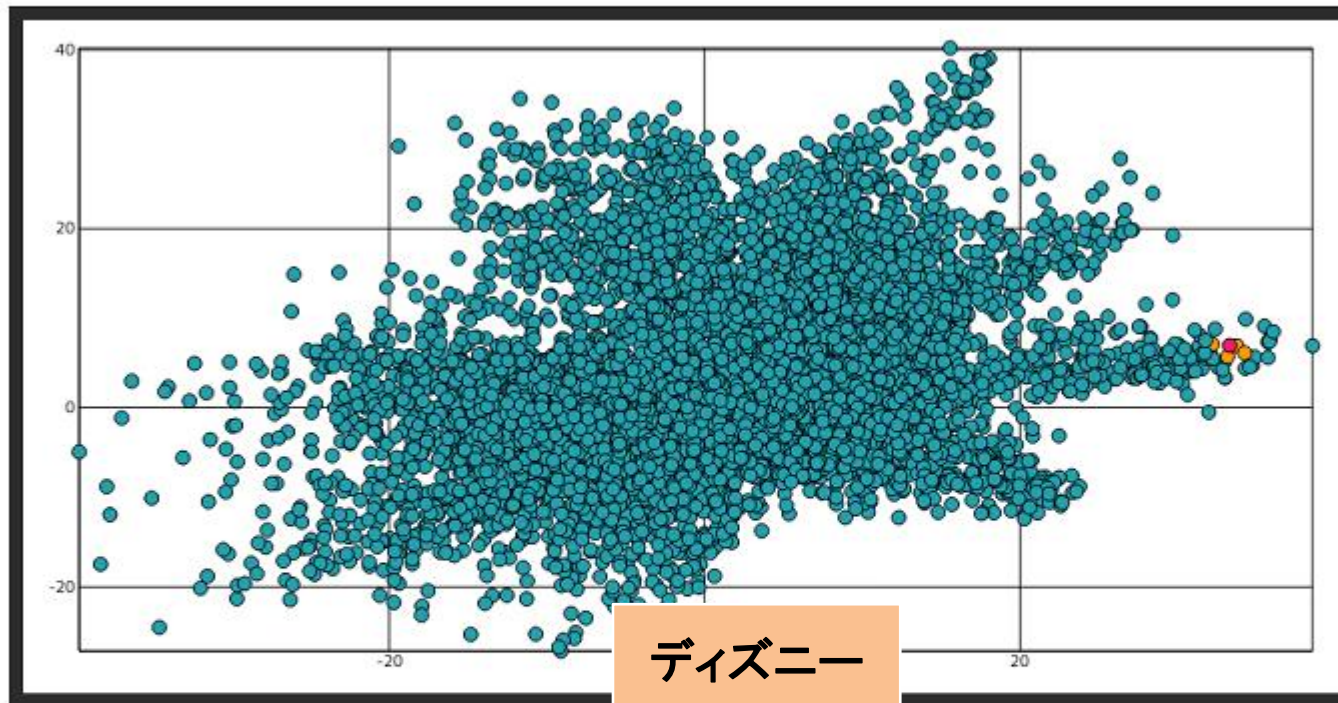
苦労話など…



- ハイパーパラメタは試行錯誤
 - 層の数、非線形関数の種類、学習係数、epoch数など
 - データの前処理の有無
- コスト関数が低くなる＝良く復元できている、わけではない
 - 結果の目視確認や適当な可視化が重要

Deep learning (Movies)

Select genre



選択データ: Fun and Fancy Free (1947) [Animation, Children's, Musical]

近傍データ:

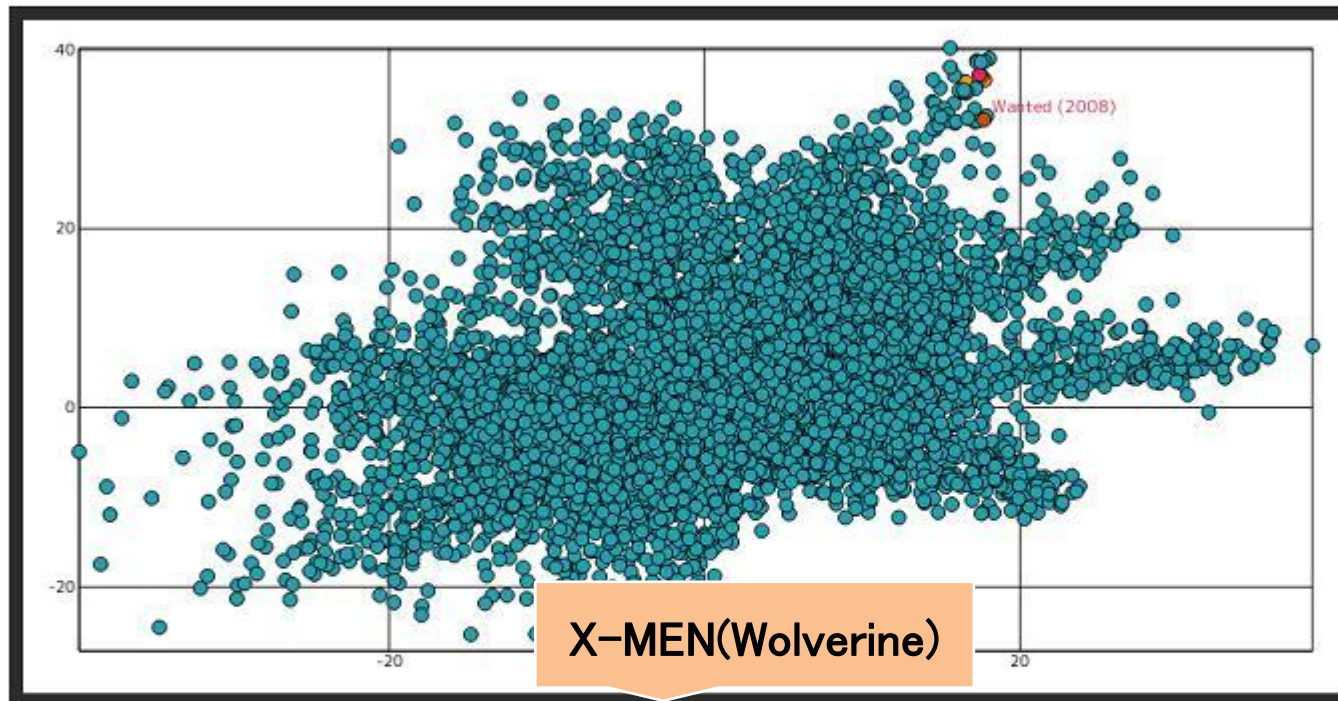
No.	Title	Genre
1	Aristocats, The (1970)	in, Children's
2	Flushed Away (2006)	
3	Tarzan (1999)	
4	Cars (2006)	

ターザン(ディズニー)

カーズ(ピクサー)

Deep learning (Movies)

Select genre



選択データ: X-Men Origins: Wolverine (2009)

近傍データ:

No.	Title	
1	Incredible Hulk, The (2008)	X-MEN
2	Daredevil (2003)	
3	X-Men (2000)	スポン
4	Spawn (1997)	

目次



- ・DeepLearningとGPU
- ・G-DEPテストドライブ
- ・ビッグデータ・GPU・DeepLearningの接点

G-DEPテストドライブ



- ・Tesla K40 / K20 (マルチGPU)
- ・Intel® Xeon E5-2650v2 2.60GHz
- ・128GB Memory
- ・多彩なミドルウェアが利用可能
- ・Linux/Windows

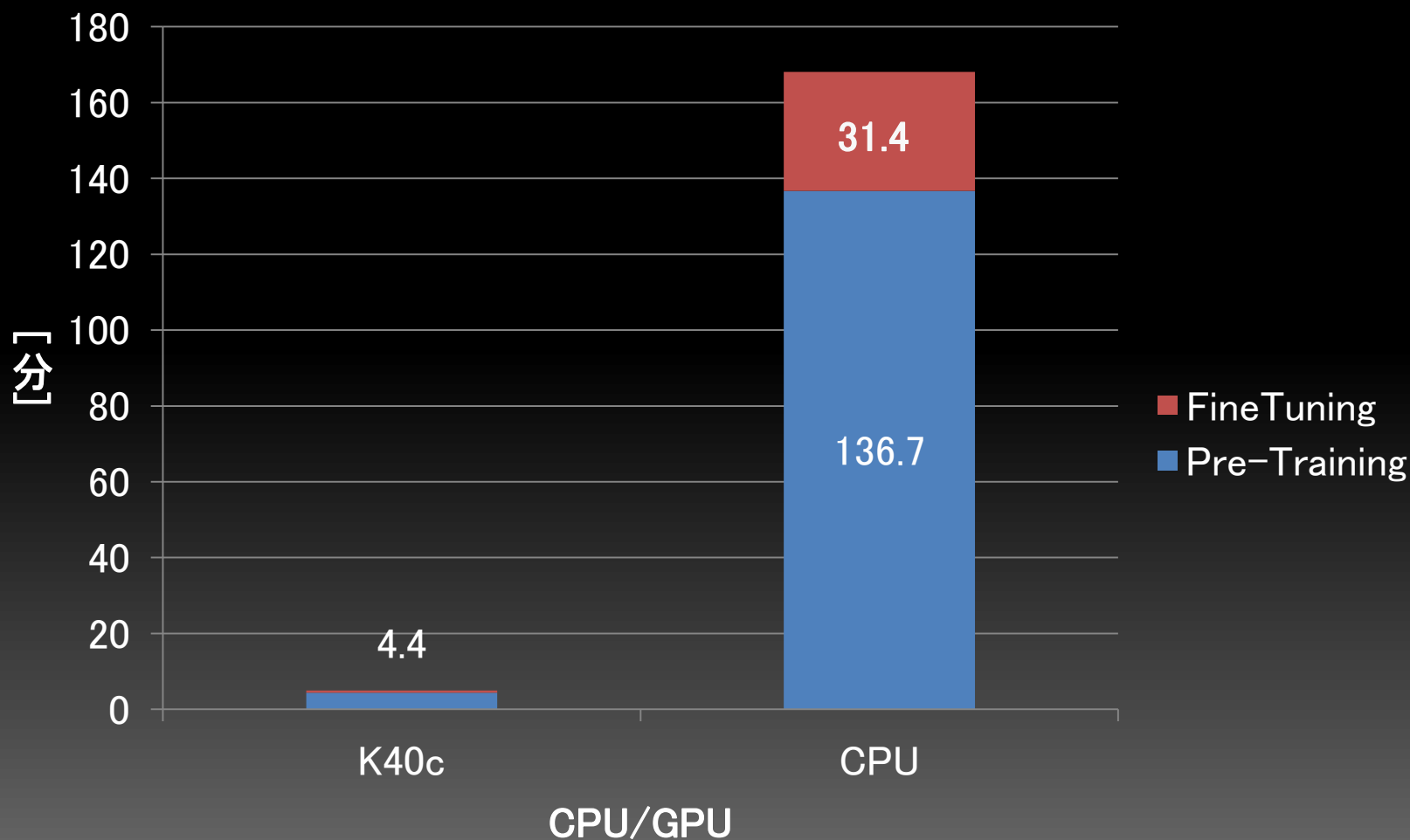
CPU/GPU性能比較 ～TheanoでSAE



- ・Hinton(2006)の再現実験
- ・Deep AutoEncoderを使い次元圧縮。
784-1000-500-25-2
- ・28x28の手書き文字、5万枚
- ・学習回数
 - ・Pre-training 15回(×4層)
 - ・Fine-Tuning 5回
- ・CPU版との性能比較
- ・圧縮後の2次元コードを表示

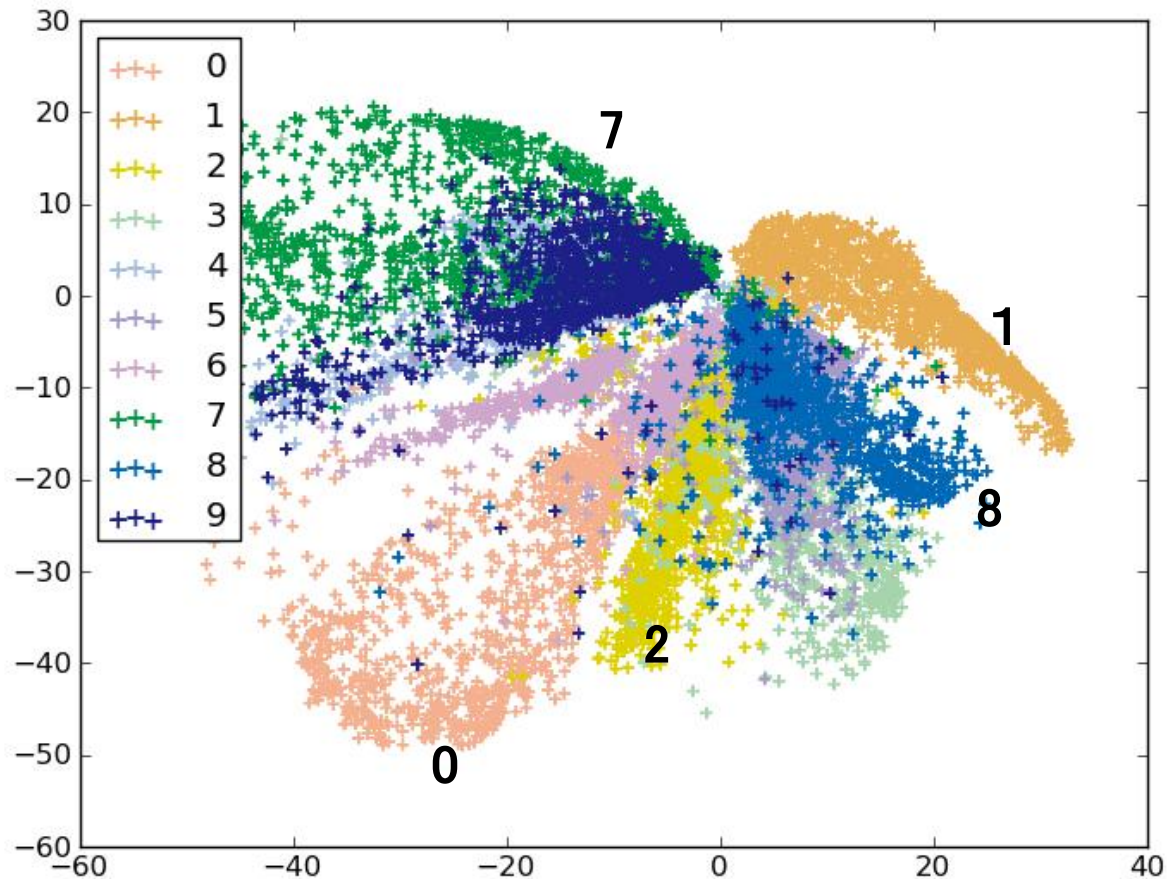
- ・Theanoとは？
 - ・PythonによるDeepLearning実装
 - ・CPU/GPUを透過的に利用可
 - ・シンボル微分に対応
 - ・pylearn2の基礎にもなっている
- ・その他実装系もあります…Let's 相談!
 - ・cuda-convnet、EBlearn、etc
 - ・http://deeplearning.net/software_links/

性能比較



※CPUはOS標準設定につきmulticoreを有効活用できていませんが、16 coreにつき単純に16倍速になってもGPUの方が高速です。

AutoEncoderによる圧縮コード



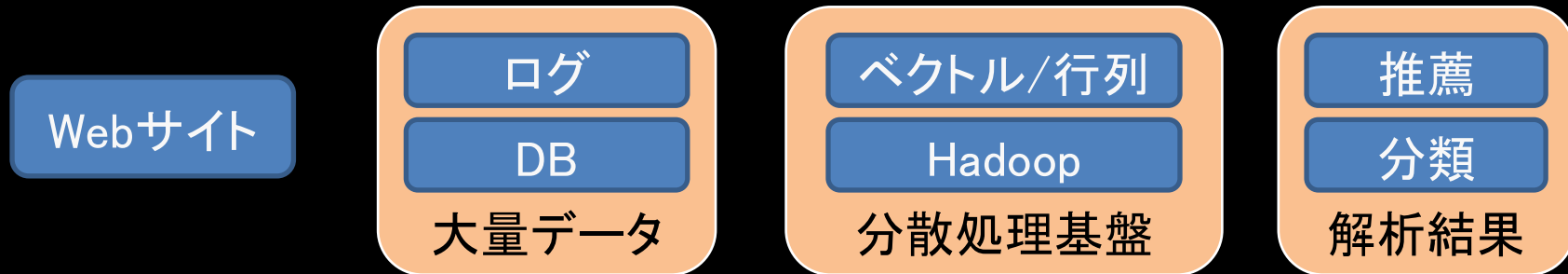
手書き文字(MNIST)を784-1000-500-250-2で圧縮

目次



- ・DeepLearningとGPU
- ・G-DEPテストドライブ
- ・ビッグデータ・GPU・DeepLearningの接点

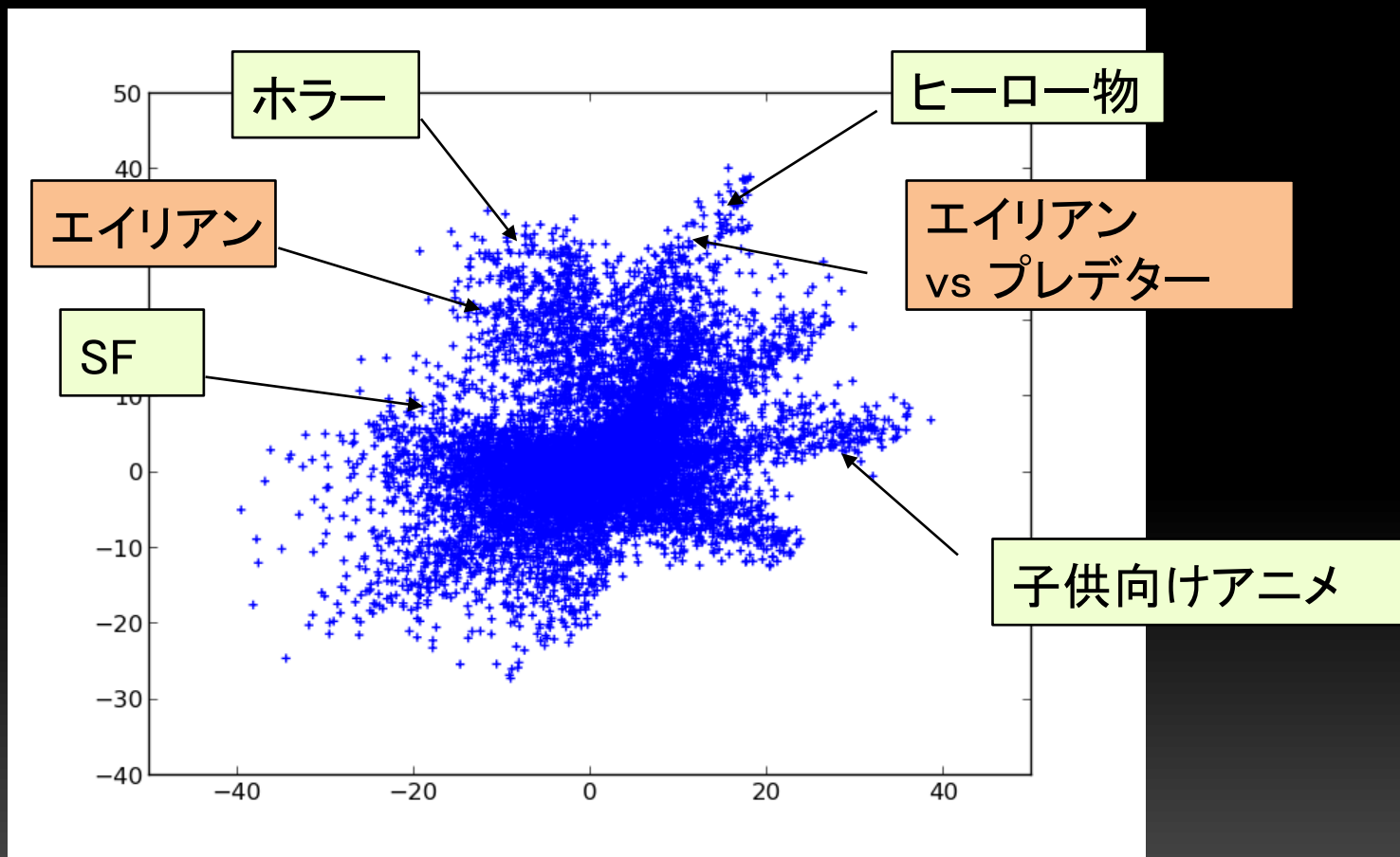
BigData + GPU



- 大量の積和を計算
- スケールアウト
 - クラスタ運用
 - MapReduceに則ったプログラム開発
- 環境面/開発面の大変さ
- JBOD(+ cache) + GPUによるスモールスタートもありでは？

- 統計手法 → 機械学習。次は DeepLearningでは？
- 取り扱う対象の拡大
 - ハンドクラフトな特徴抽出が困難
 - 教師データが無い
- D.L.による特徴抽出
 - Stacked AutoEncoder

[再掲]特徴空間(Deep Learning)



- 映画の口コシ情報
- 1000次元 → 2次元化

References



Le et al., Building High-level Features Using Large Scale Unsupervised Learning, 2012

Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks, 2012

Xie et al., Image Denoising and Inpainting with Deep Neural Networks, 2012

山下 et al., Deep Convolutional Neural Network による手形状領域の抽出, 2014

Hinton, Reducing the Dimensionality of Data with Neural Networks, 2006