2013/09/11

Deep Learning for Natural Language Processing

自然言語処理のための Deep Learning

at Gunosy

東京工業大学 奥村・高村研究室

D1 菊池悠太 @kiyukuta

Mitsumasa Kubo 17:56

Deep learning ってなんかニューラルネットワークの超パワーアップ版なんだっけ、ぐらいの前提知識からNLPへの活用・展望みたいなのを具体例を交えつつ、丁寧にレクチャーしてくれるとうれしい！！

時間は30~60分ぐらいでお好きに！

逆に最低これくらいは読んで理解しとけや、みたいな資料などがあれば事前に共有しておく

NN→多層×→pretraining→breakthrough !!

Mitsumasa Kubo 17:56

Deep learning ってなんかニューラルネットワークの超パワーアップ版なんだっけ、ぐらいの前提知識からNLPへの活用・展望みたいなのを具体例を交えつつ、丁寧にレクチャーしてくれるとうれしい！！

時間は30~60分ぐらいでお好きに！

逆に最低これくらいは読んで理解しとけや、みたいな資料などがあれば事前に共有しておく

焦って早口過ぎてたら教えて下さい

2つのモチベーション

- NLPでニューラルネットを
- 言語の意味的な特徴を

A yet another brief introduction to neural networks

http://www.slideshare.net/yutakikuchi927/a-yet-another-brief-introduction-to-neural-networks-26023639

Neural networkベースの話

RBMとか苦しい

Deep Learning for NLP

Deep Learning

for NLP

Deep Learning概要

Neural Networkふんわり

Deepへの難しさ

Pretrainingの光

Stacked Autoencoder , DBN

Deep Learning

for NLP

Deep Learning

Deep Learning概要

Neural Networkふんわり

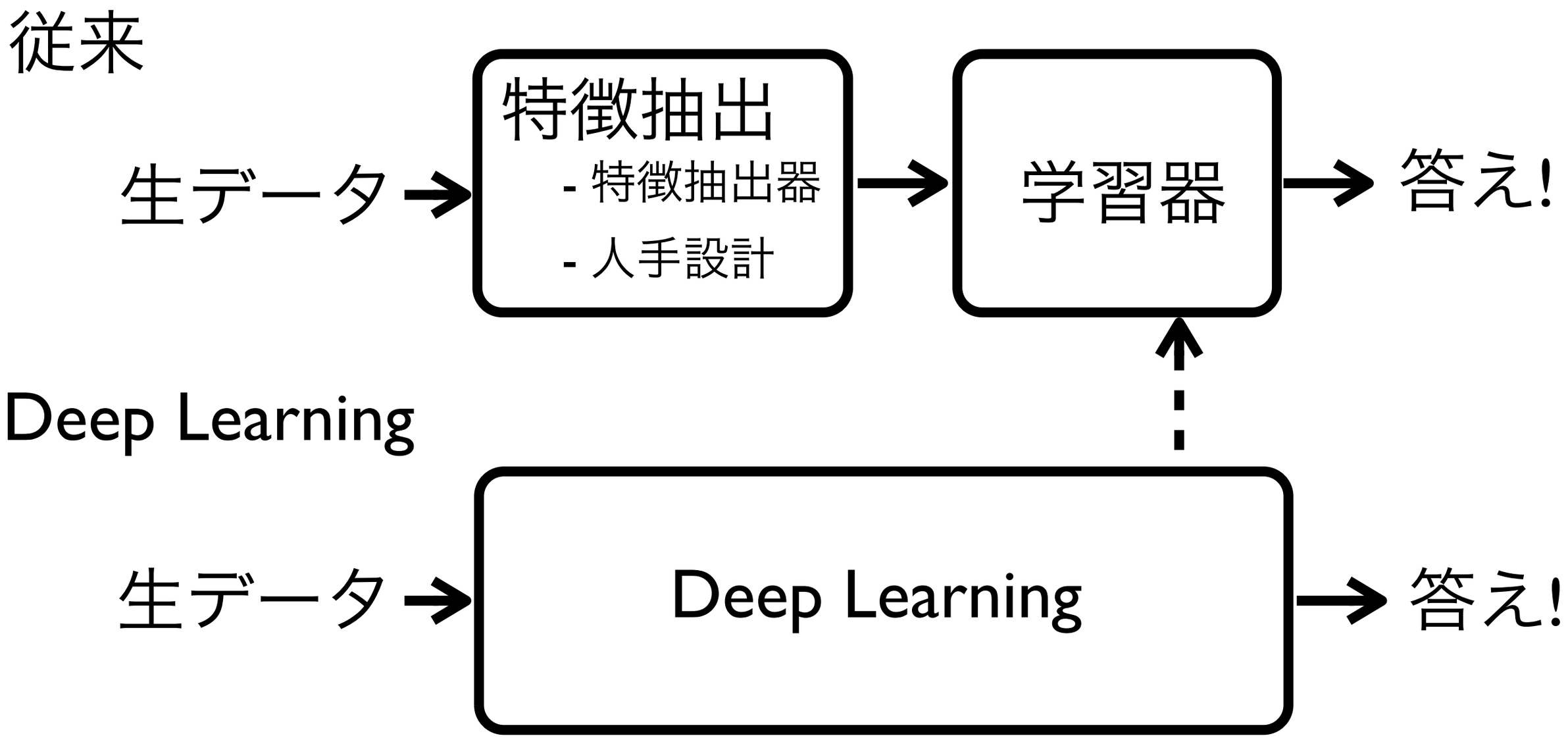
Deepへの難しさ

Pretrainingの光

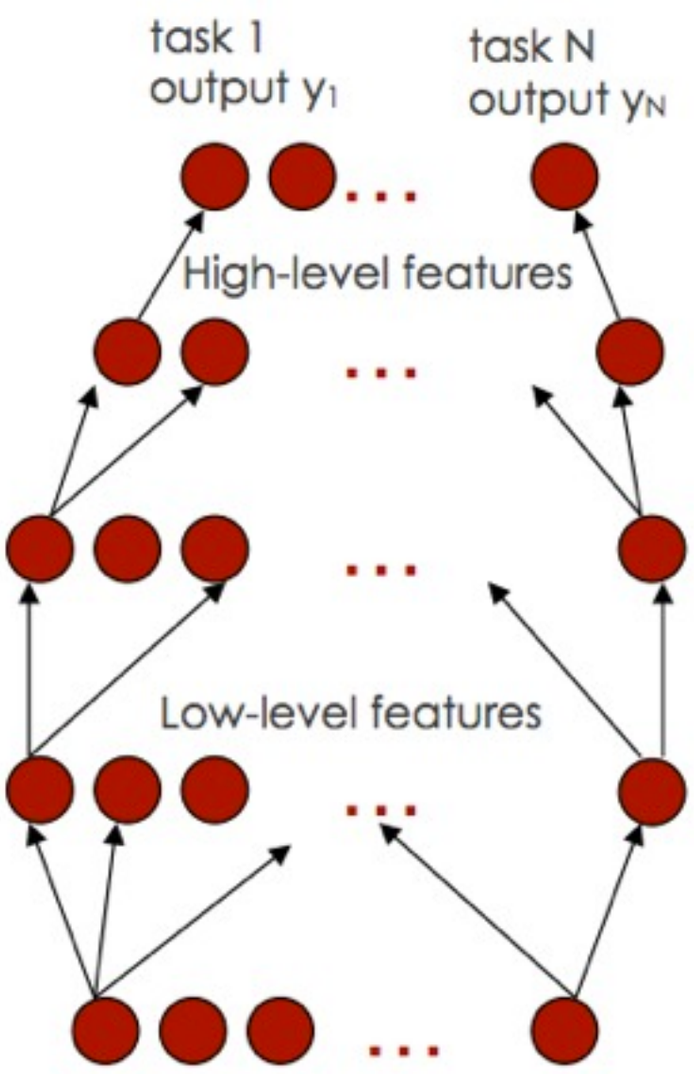
Stacked Autoencoder , DBN

Deep Learning

Unsupervised Representation Learning



Deep Learning



結論からいうと

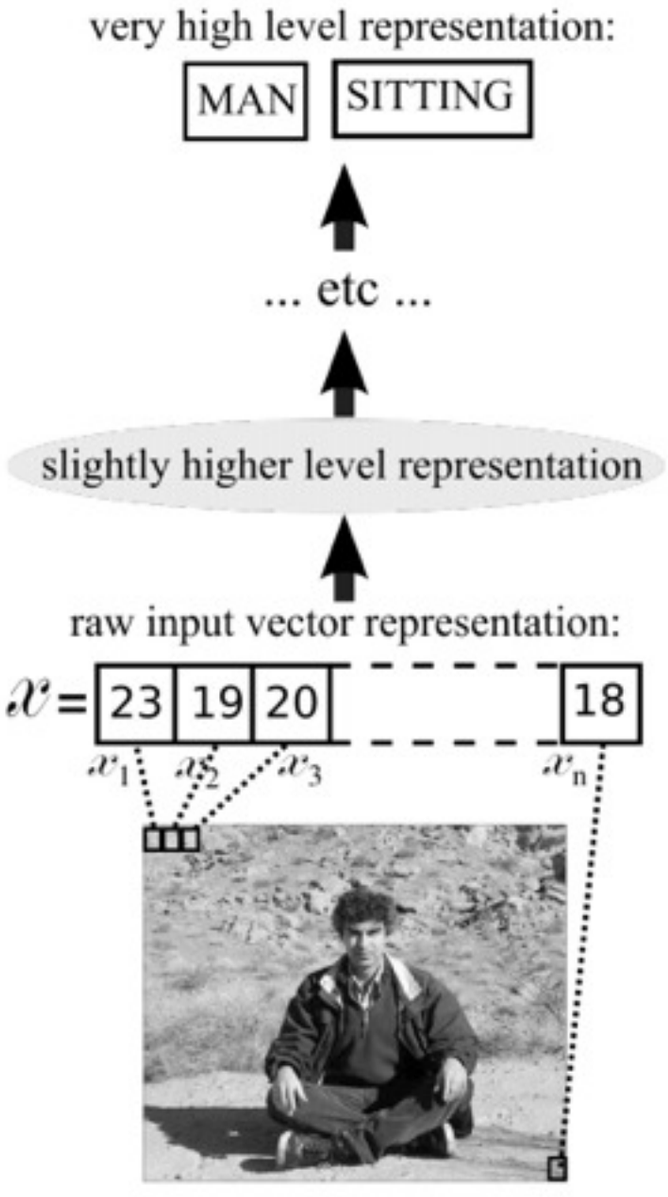
Deep Learningとは

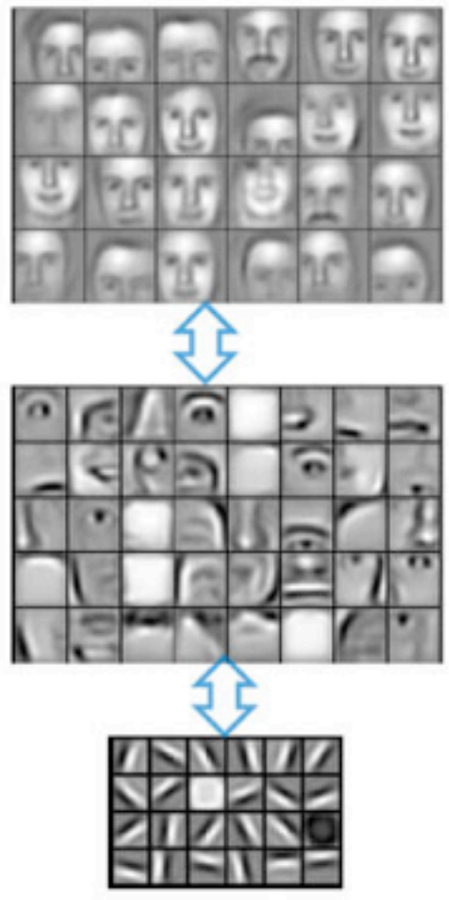
良い初期値を（手に入れる方法を）

手に入れた

多層**Neural Network**です

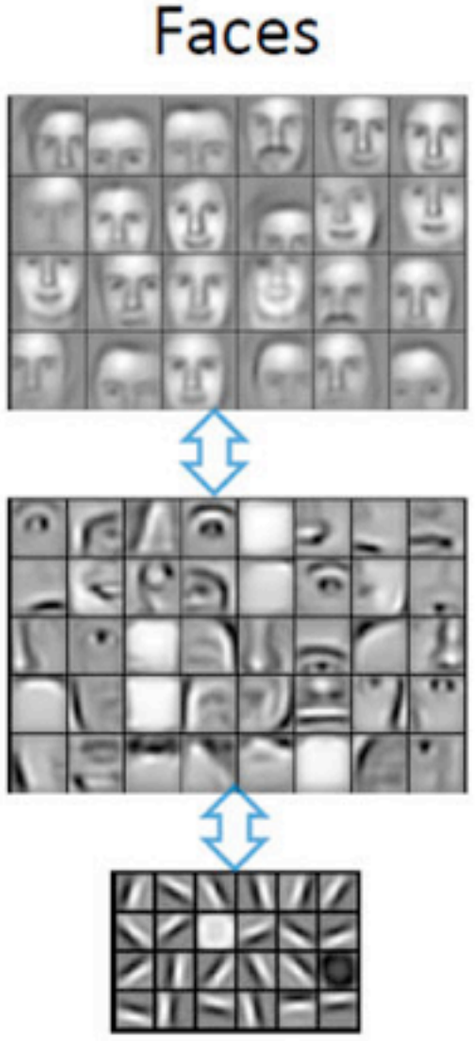
Deep Learning





生画像から**階層毎**に**階層的**な特徴を

ラベル無しデータから**教師なし**で学習

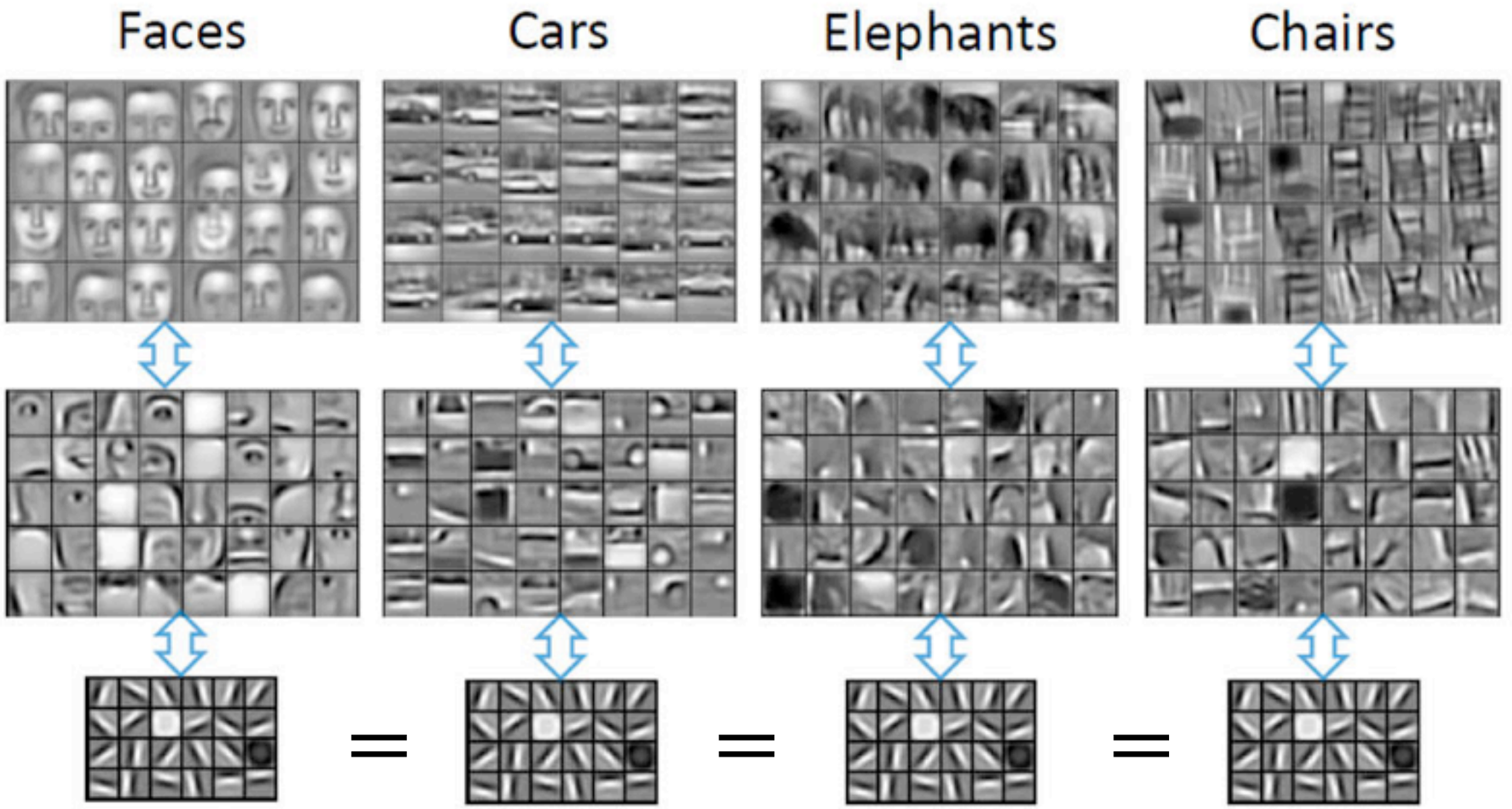


高次な特徴は，より低次な特徴の組み合わせで表現

↑

生画像

低次レベルの特徴は共有可能



将来のタスクが未知でも

起こる世界は今と同じ

Deep Learning

Deep Learning概要

Neural Networkふんわり

Deepへの難しさ

Pretrainingの光

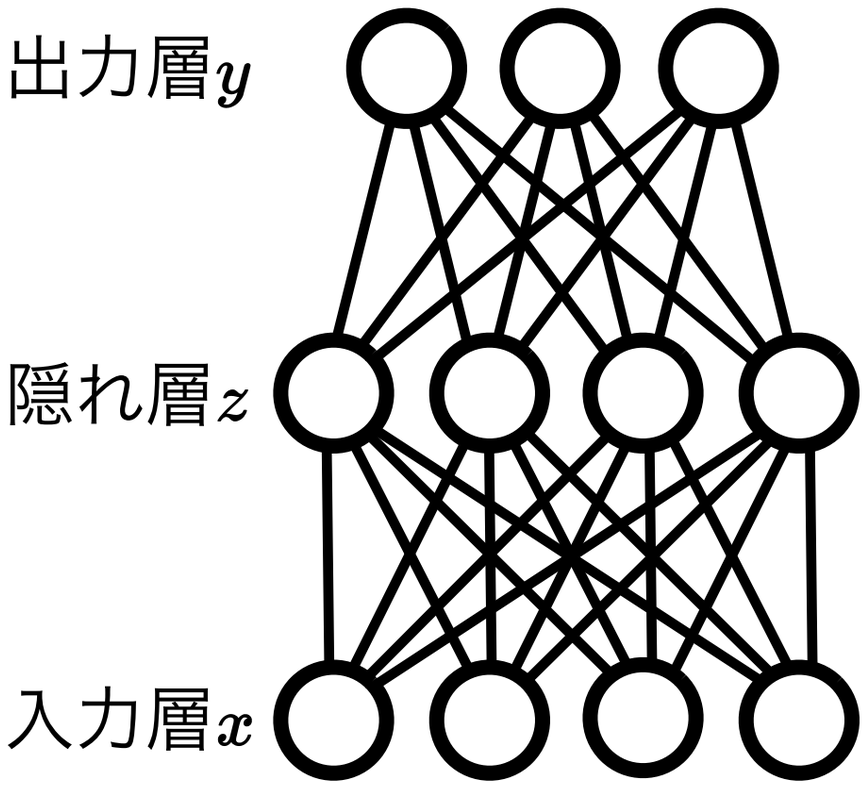
Stacked Autoencoder , DBN

A yet another
brief introduction to

Neural Networks

菊池 悠太

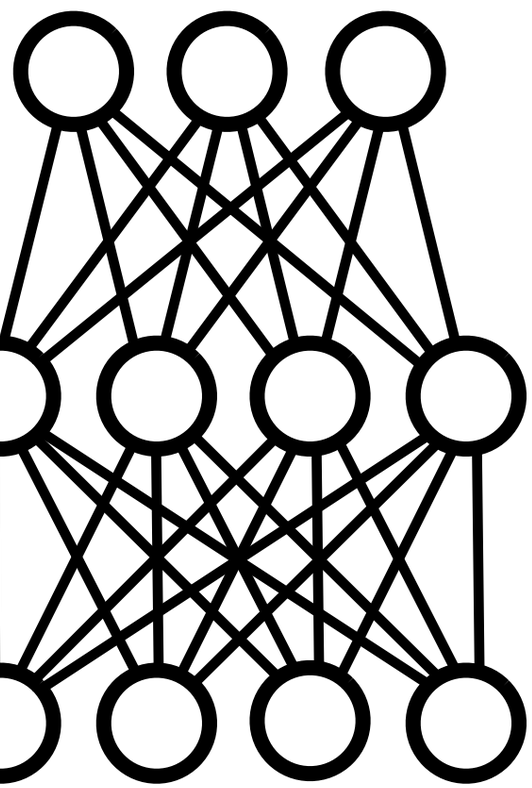
Neural Network



Neural Network

予測

出力層y



隠れ層z

入力層x

生データ，抽出した素性

例えば，手書き数字認識

[0.05, 0.05, 0.05, 0.40, 0.05, 0.05, 0.15, 0.05, 0.15, 0.05]

10次元の確率分布

（左から，入力画像が，
0である確率，
1である確率
…
9である確率）

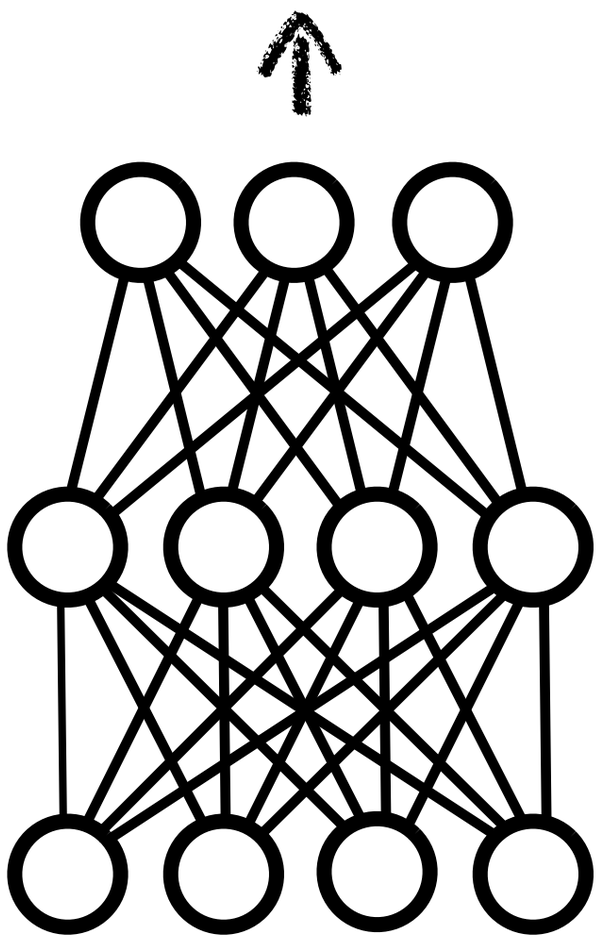
出力層y

10次元

隠れ層z

入力層x

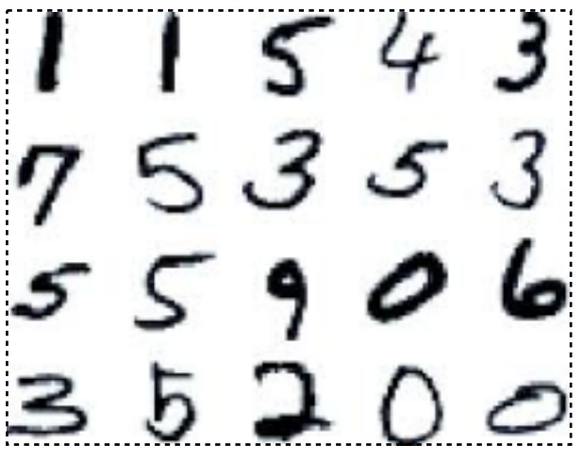
784次元



３！！

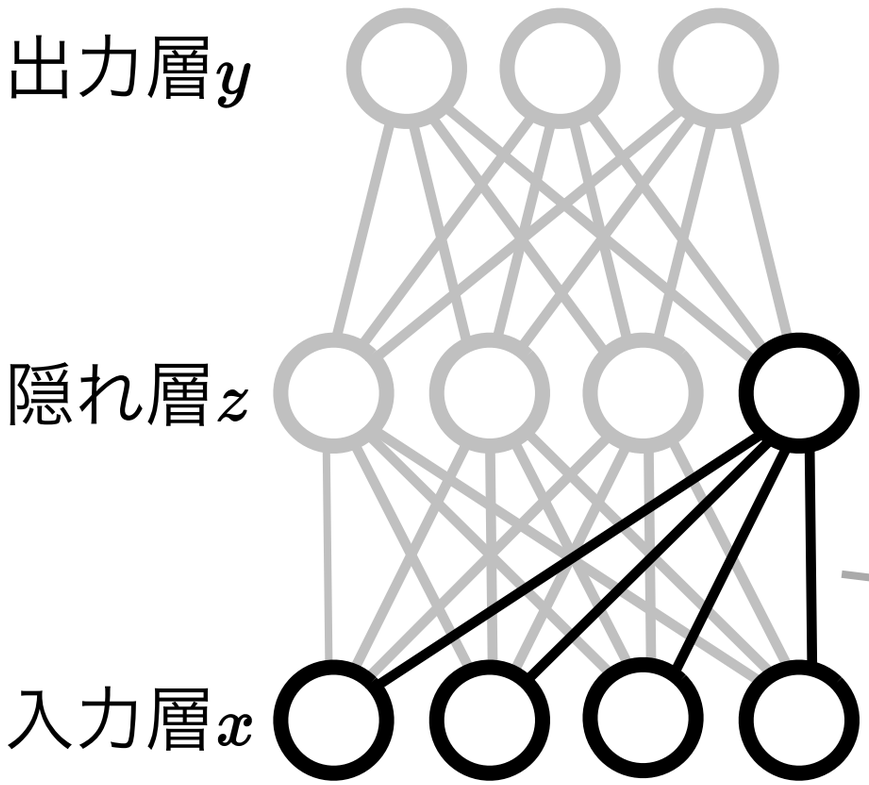
3

MNIST (28*28の画像)



28*28=
784次元の数値ベクトル

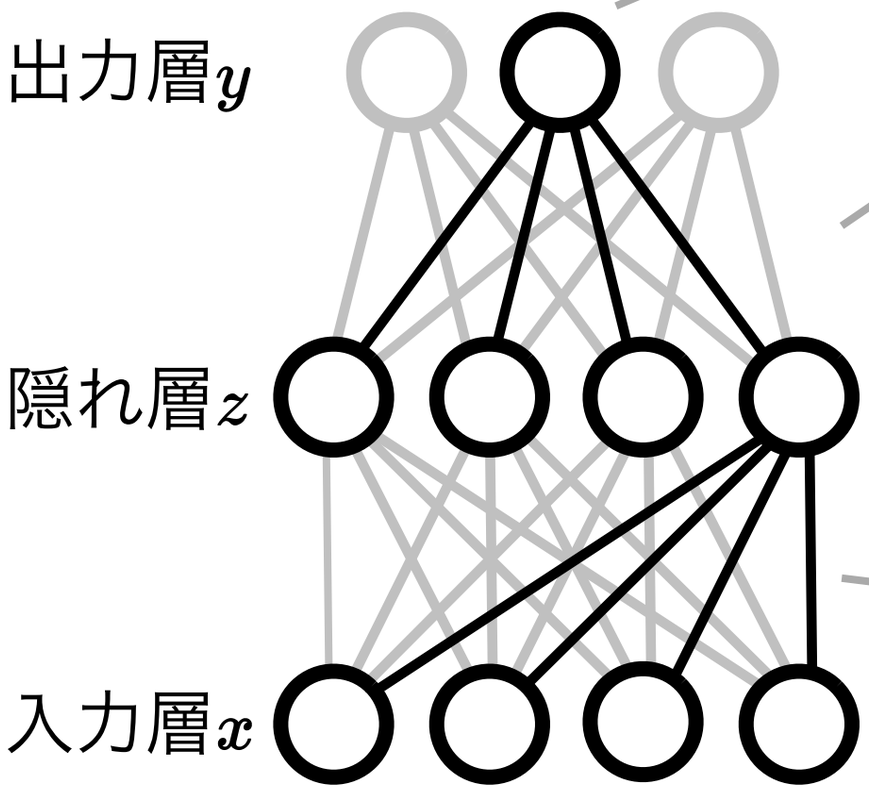
Neuron



隠れユニットj

隠れユニットjの
入力層に対する重み
$\mathbf{W}_1$

Neuron



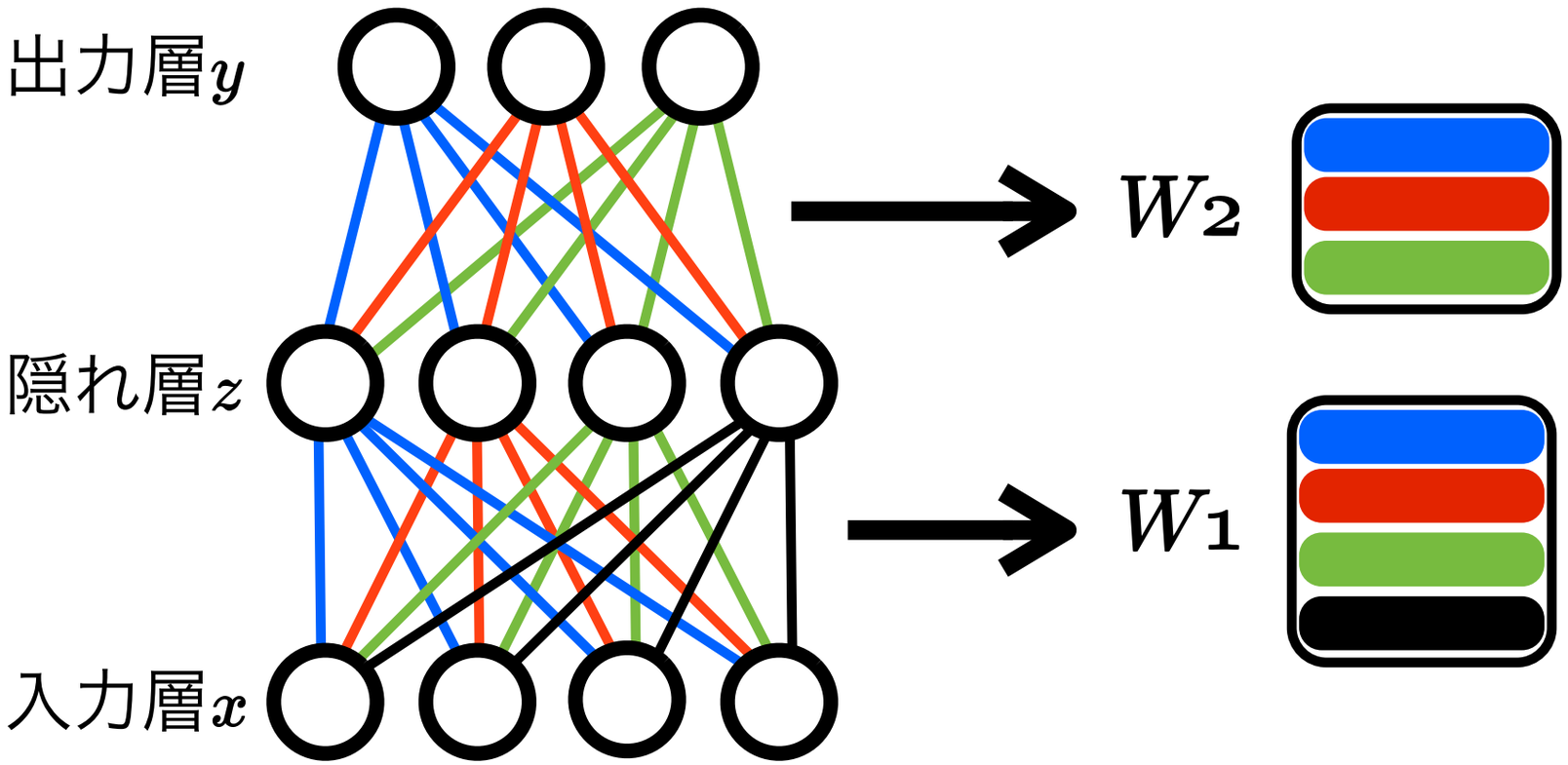
出力ユニット**k**

出力ユニット**k**の
隠れ層に対する重み$\mathbf{W_2}$

隠れユニット**j**

隠れユニット**j**の
入力層に対する重み
$\mathbf{W_1}$

層間の重みを行列で表現



行列で表現

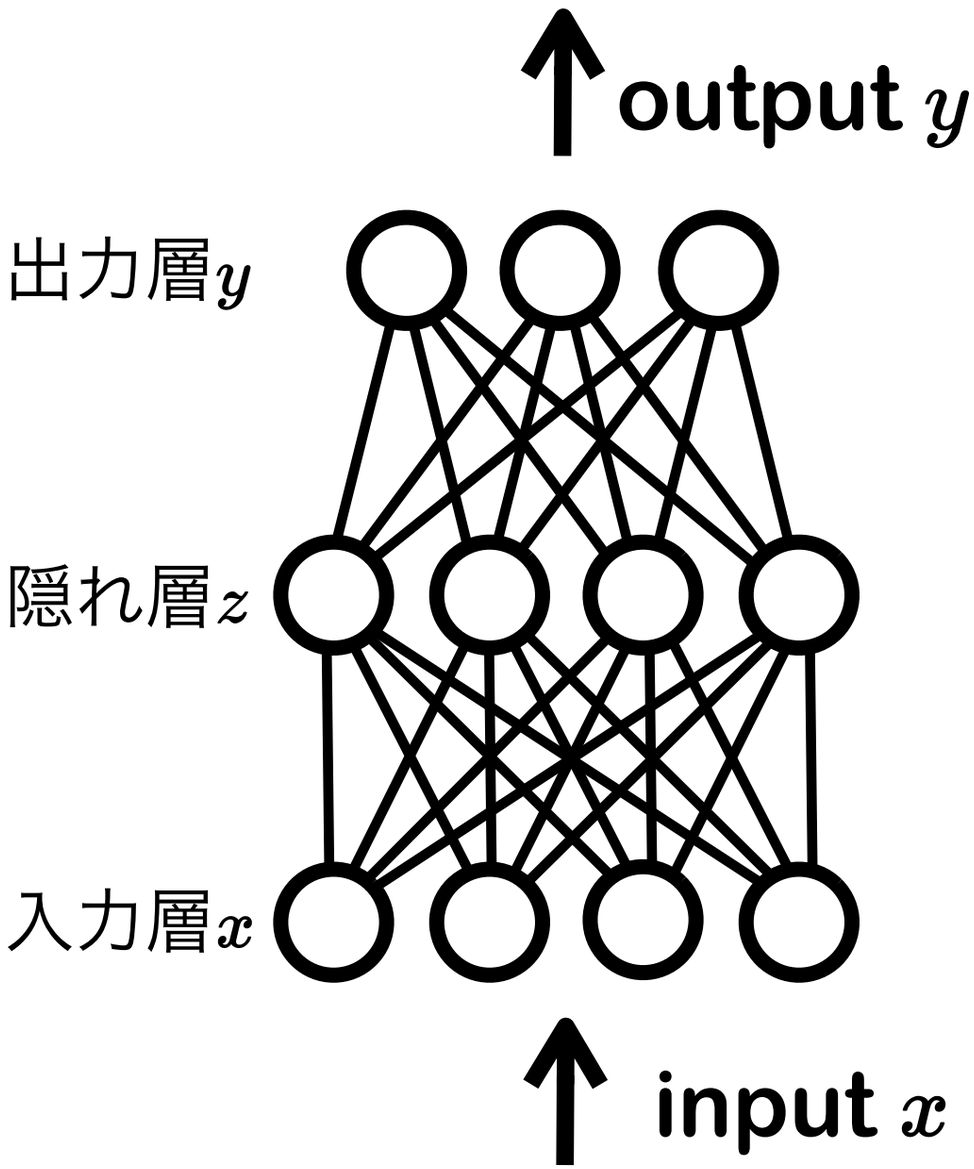
Neural Networkの処理

- Forward propagation
- Back propagation
- Parameter update

Neural Networkの処理

- Forward propagation
- Back propagation
- Parameter update

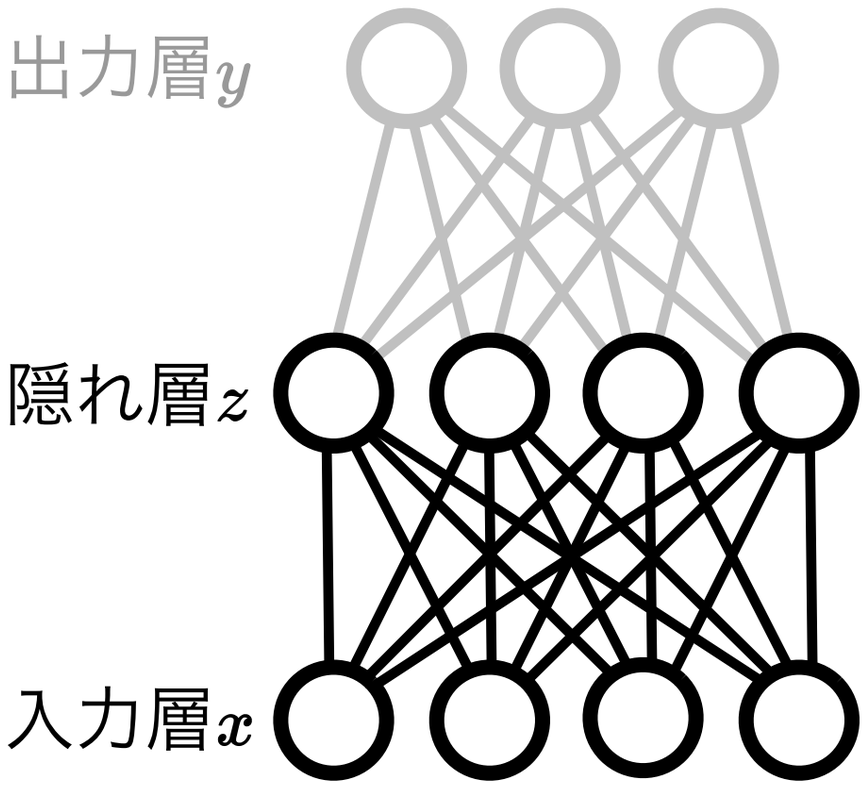
Forward Propagation



入力に対し出力を出す

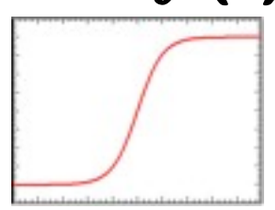
入力層から隠れ層への情報の伝播

$$z = f(W_1 x + b_1)$$



非線形活性化関数$f(\ )$

$$f(x) = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ f(x_3) \end{bmatrix}$$



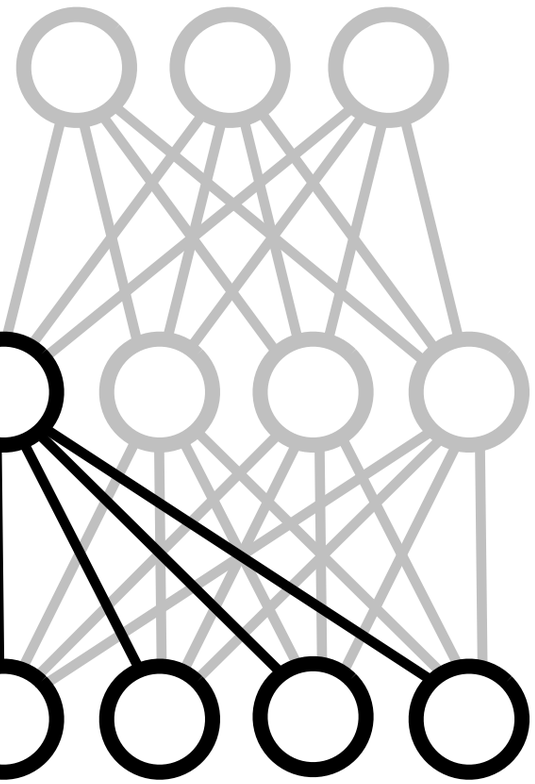
tanh とか
sigmoid とか

入力層から隠れ層への情報の伝播

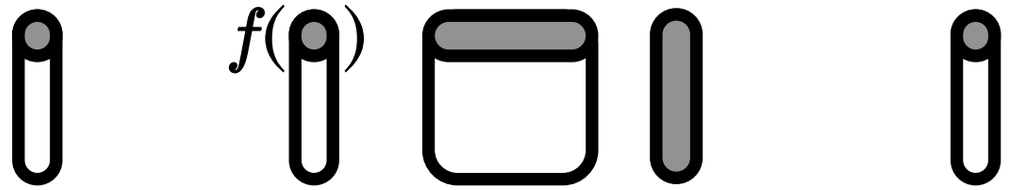
出力層y

隠れ層z

入力層x

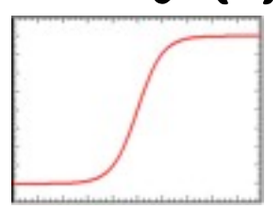


$$z = f(W_1 x + b_1)$$



非線形活性化関数$f(\ )$

$$f(x) = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ f(x_3) \end{bmatrix}$$



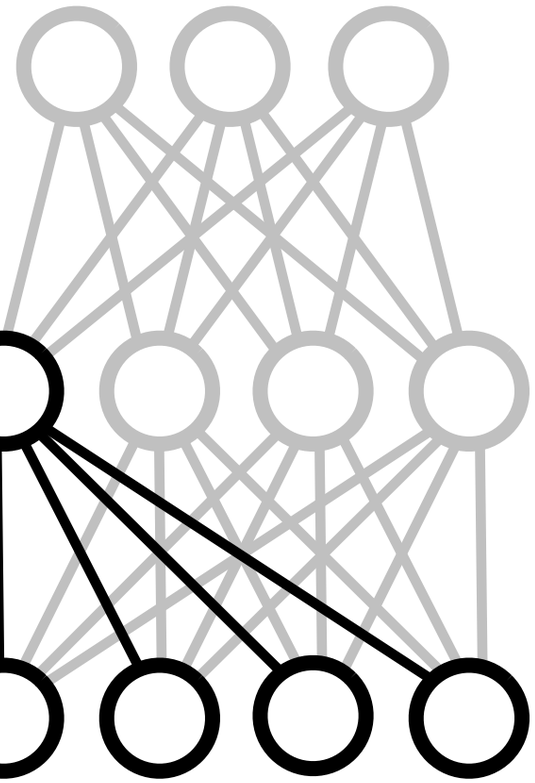
tanh,sigmoid
reLU, maxout...

入力層から隠れ層への情報の伝播

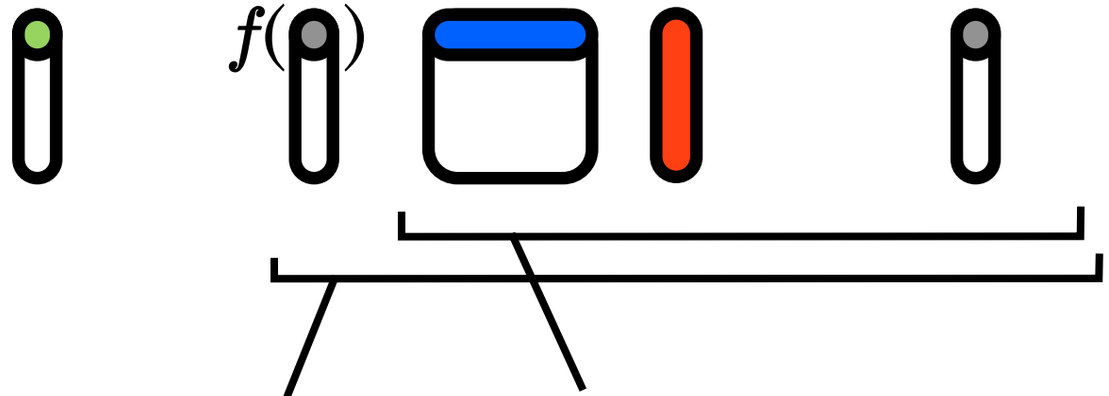
出力層y

隠れ層z

入力層x



$$z = f(W_1 x + b_1)$$



入力の情報を
重み付きで受け取る

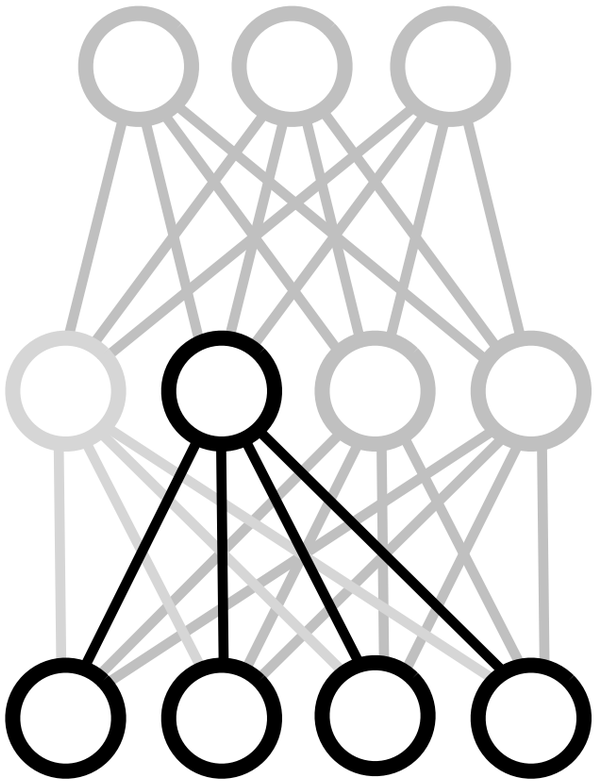
隠れユニットが出す
出力値が決まる

入力層から隠れ層への情報の伝播

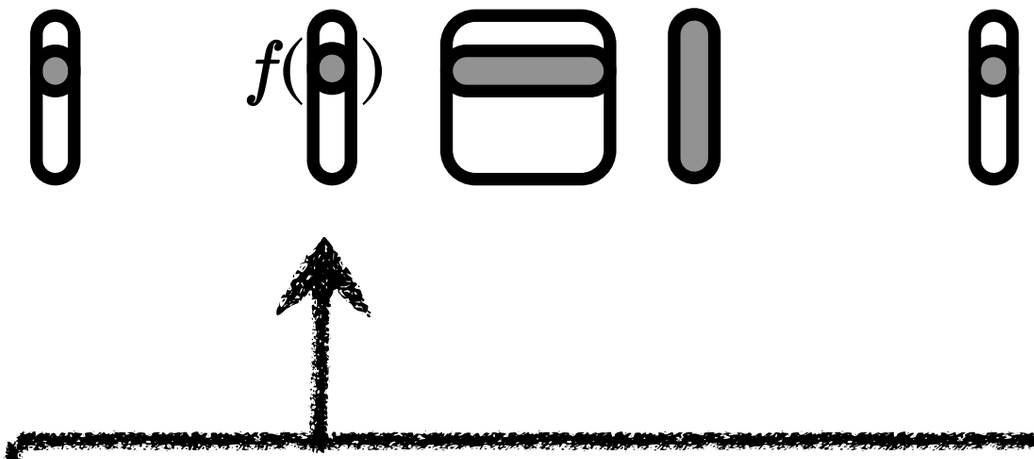
出力層y

隠れ層z

入力層x

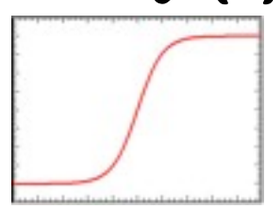


$$z = f(W_1 x + b_1)$$



非線形活性化関数$f(\ )$

$$f(x) = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ f(x_3) \end{bmatrix}$$

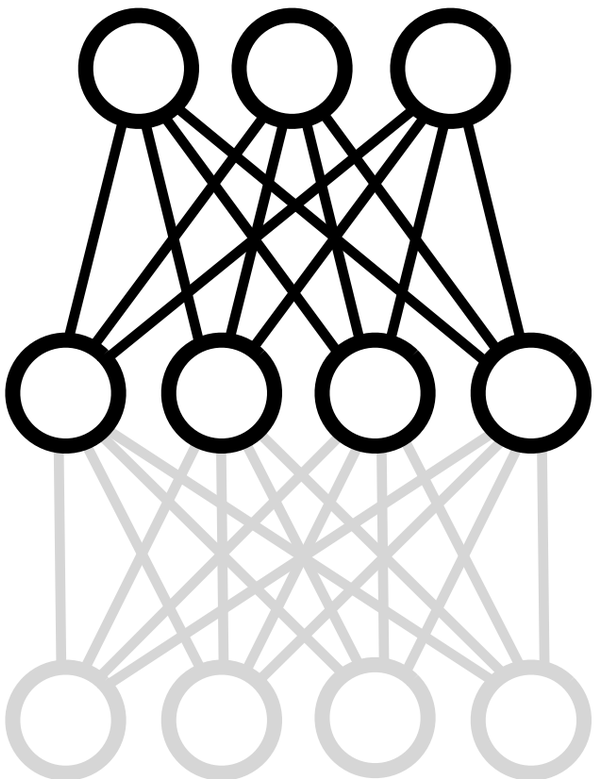


tanh,sigmoid
reLU, maxout...

隠れ層から出力層への情報の伝播

$$y = \sigma(W_2 z + b_2)$$

出力層y



隠れ層z

入力層x

出力層用の
非線形活性化関数σ( )

タスク依存

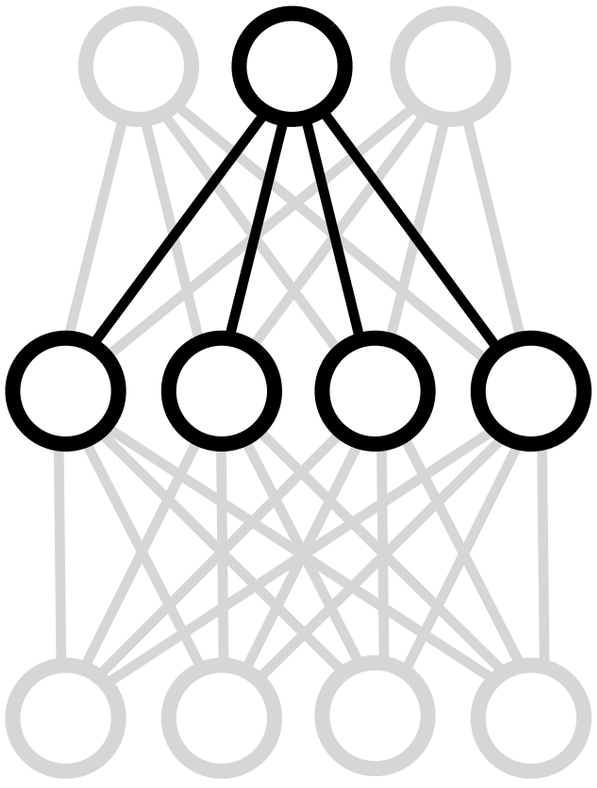
隠れ層から出力層への情報の伝播

$$y = \sigma(W_2 z + b_2)$$

出力層y

隠れ層z

入力層x

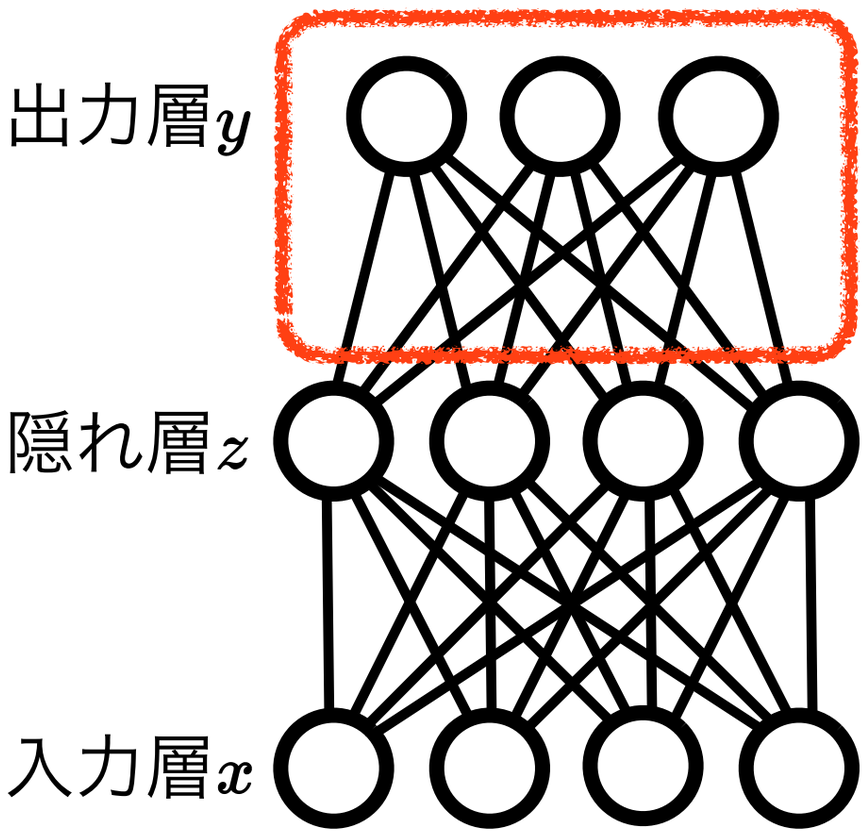


出力層用の
非線形活性化関数σ( )

タスク依存

タスク依存の出力層

$$y = \sigma(W_2 z + b_2)$$

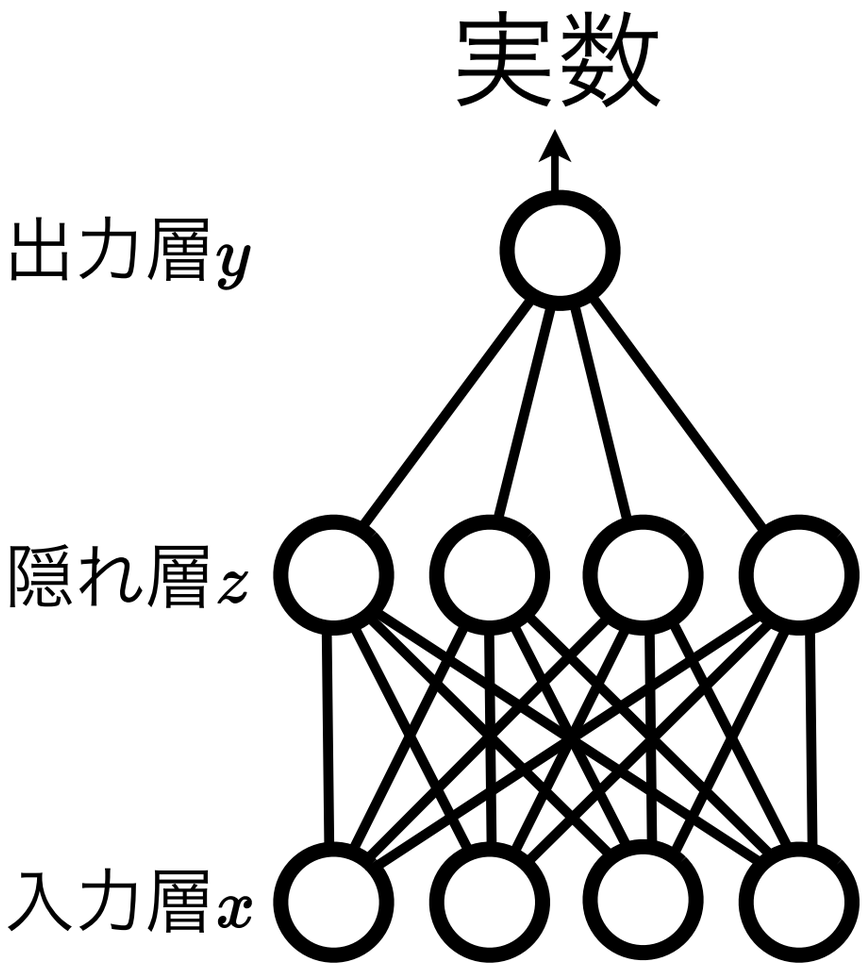


解きたいタスクによって

σが変わる

- 回帰
- 二値分類
- 多値分類
- マルチラベリング

回帰のケース



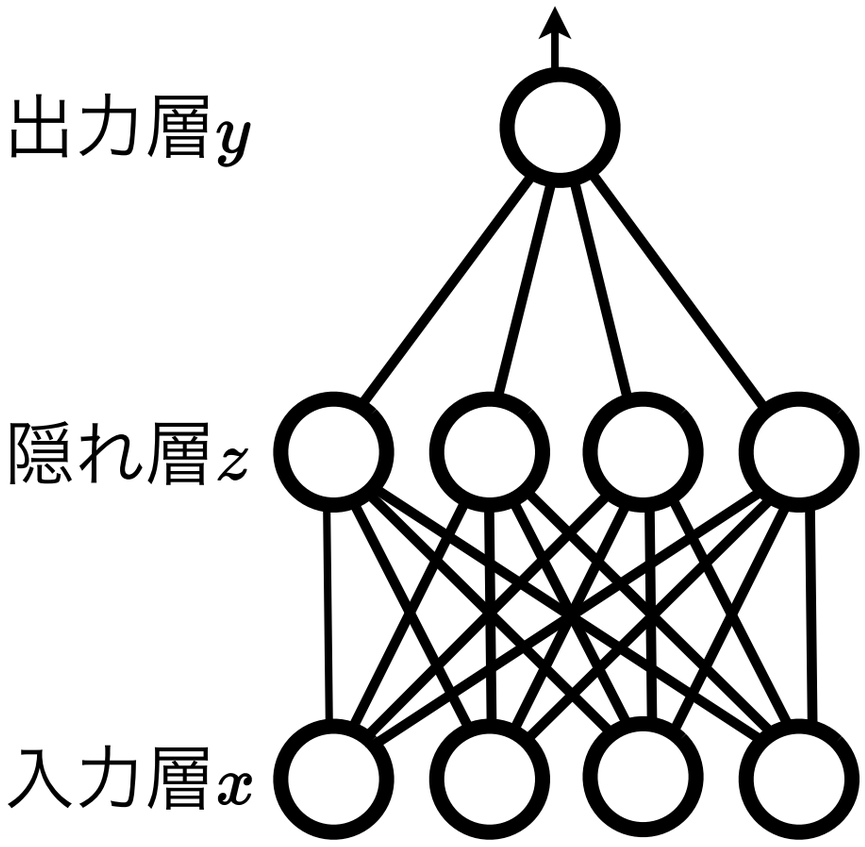
$$y = \sigma(W_2 z + b_2)$$

出力に値域はいらない

恒等写像でそのまま出力

$$\sigma(a) = a$$

二値分類のケース



$$y = \sigma(W_2 z + b_2)$$

出力層は確率

σは**0.0~1.0**であって欲しい

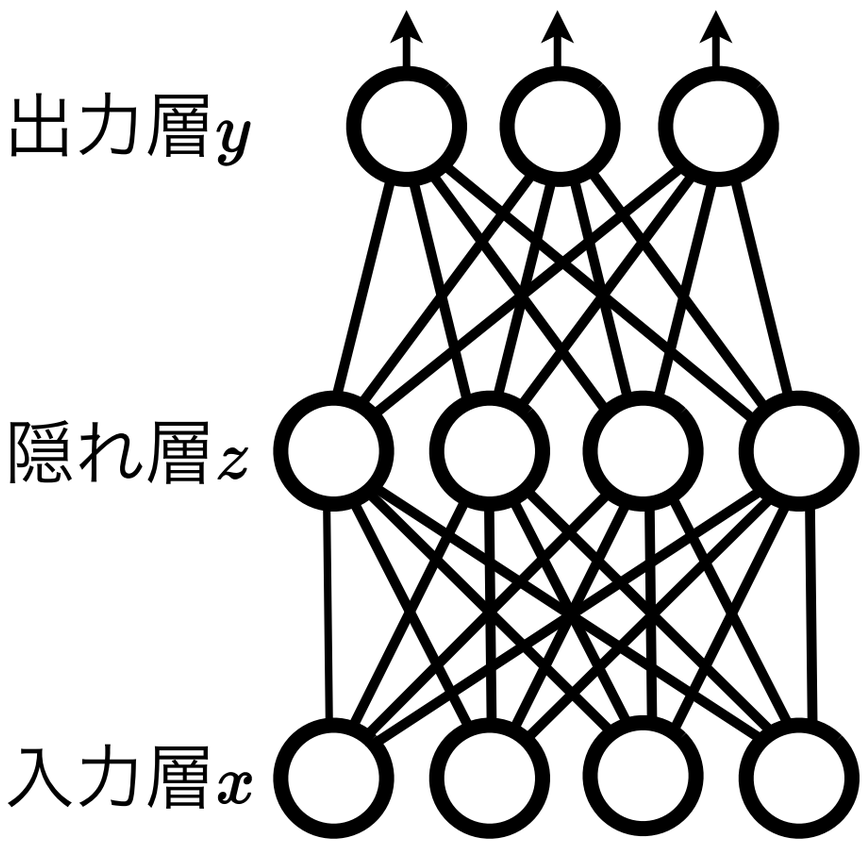
Sigmoid関数

$$\sigma(a) = \frac{1}{1+\exp(-a)}$$

多値分類のケース

sum(0.2 0.7 0.1)=1.0

$$y = \sigma(W_2 z + b_2)$$



出力は確率分布

↓

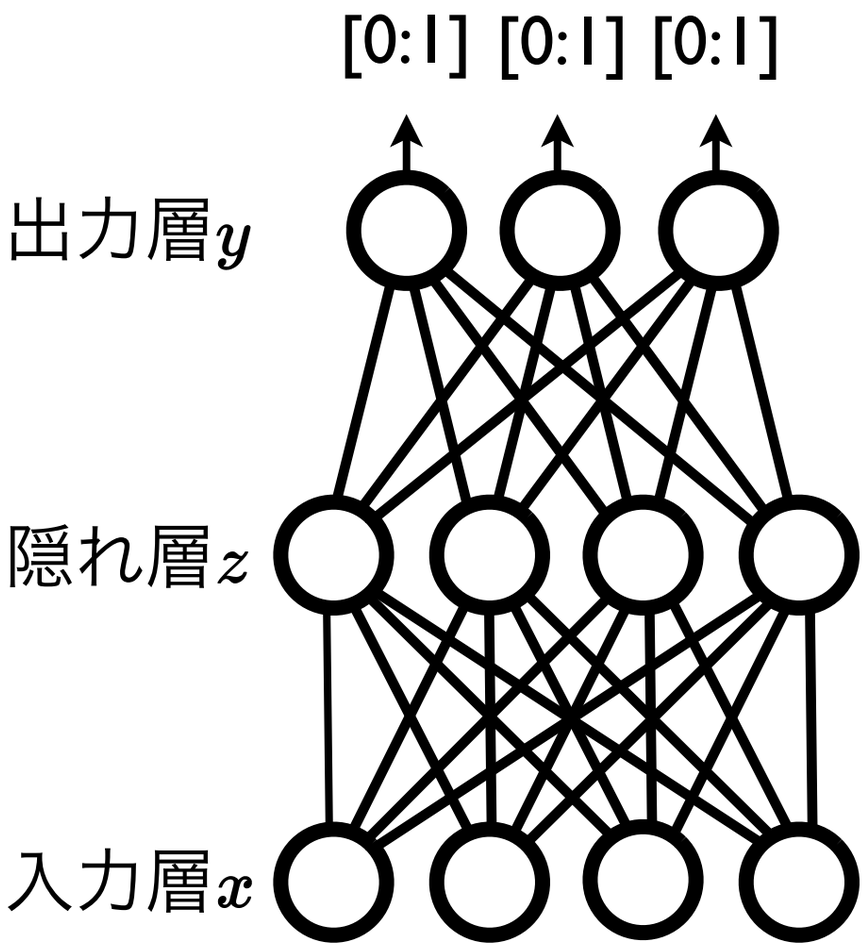
各ノード**0**以上，総和が**1**

↓

Softmax関数

$$\sigma(a) = \frac{\exp(a)}{\sum \exp(a)}$$

マルチラベリングのケース



$$y = \sigma(W_2 z + b_2)$$

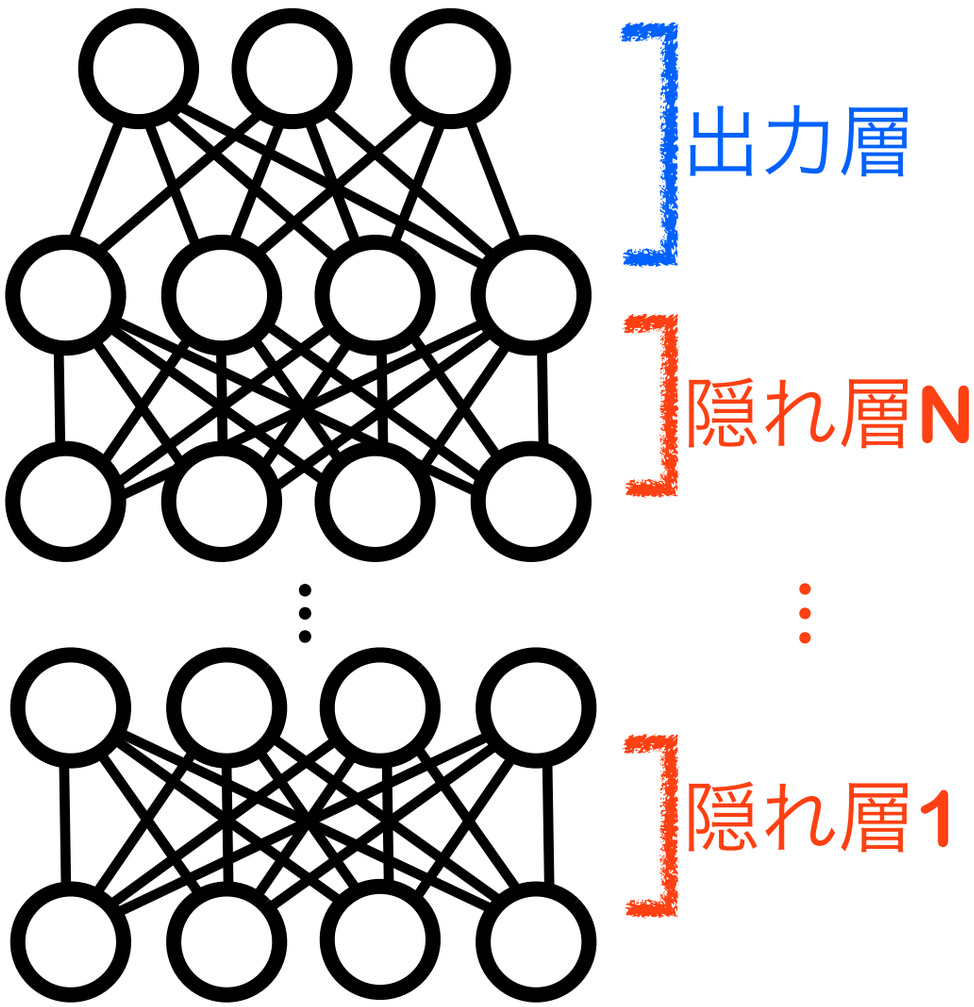
各々が独立に二値分類

element-wiseで

Sigmoid関数

$$\sigma(a) = \frac{1}{1+\exp(-a)}$$

ちなみに多層になった場合

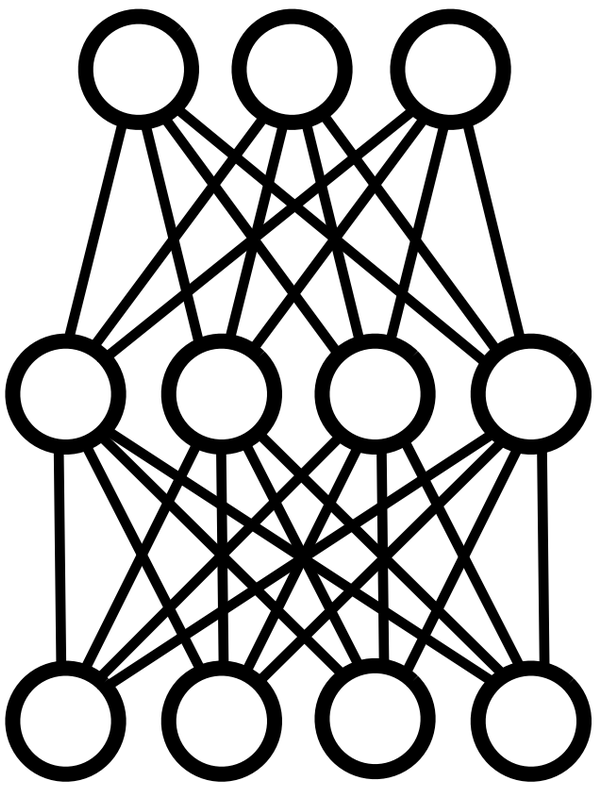


出力層だけタスク依存

隠れ層はぜんぶ同じ

Forward Propagation

出力層y



隠れ層z

入力層x

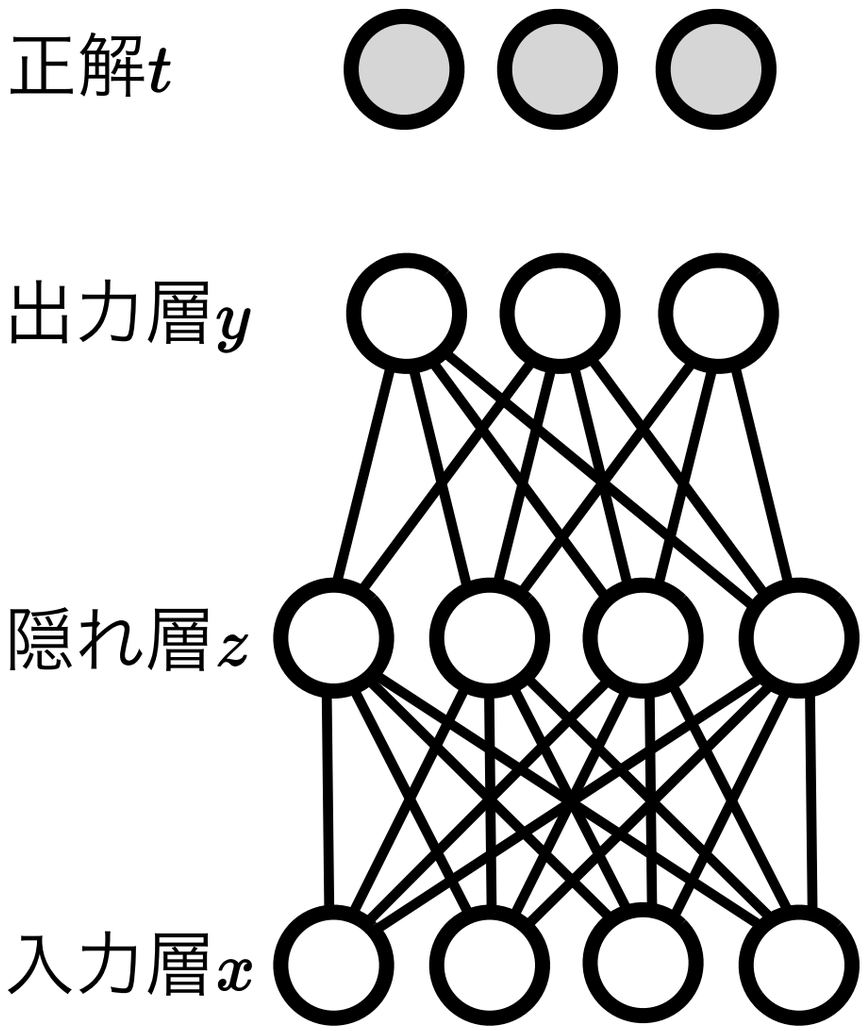
$$y = \sigma(W_2 z + b_2)$$

$$z = f(W_1 x + b_1)$$

Neural Networkの処理

- Forward propagation
- **Back propagation**
- Parameter update

Back Propagation

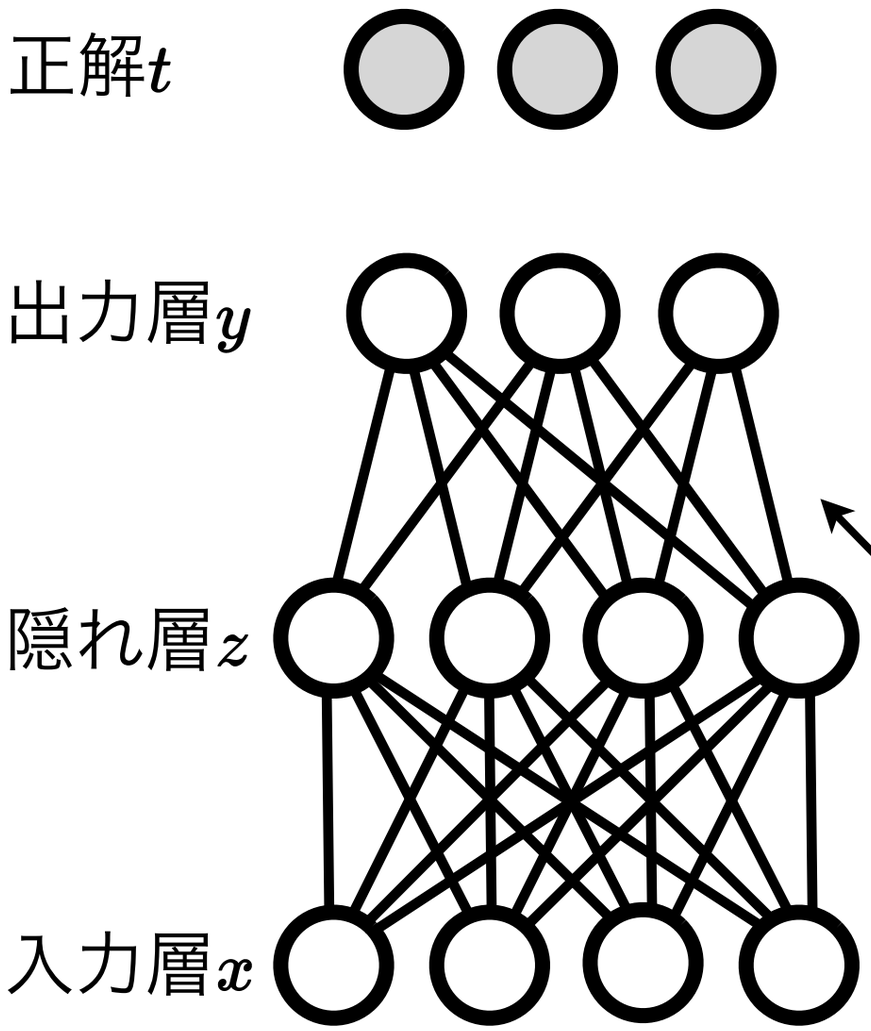


NNが入力に対する出力の

予測を間違えた場合

正解するように修正したい

Back Propagation



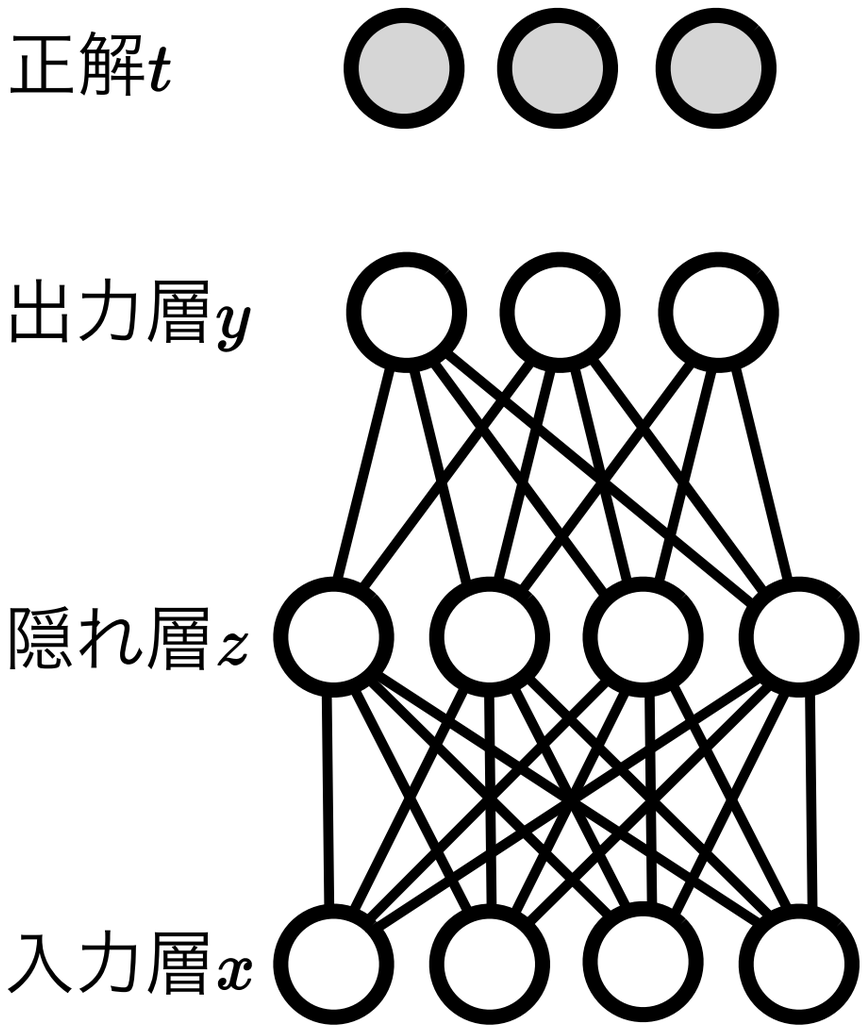
NNが入力に対する出力の
予測を間違えた場合
正解するように修正したい

修正対象: 層間の重み

$$y = \sigma(W_2 z + b_2)$$
$$z = f(W_1 x + b_1)$$

↑と，バイアス

Back Propagation



誤差関数を最小化するよう修正

$$E(\theta) = \frac{1}{2}\|y(\theta) - t\|^2$$

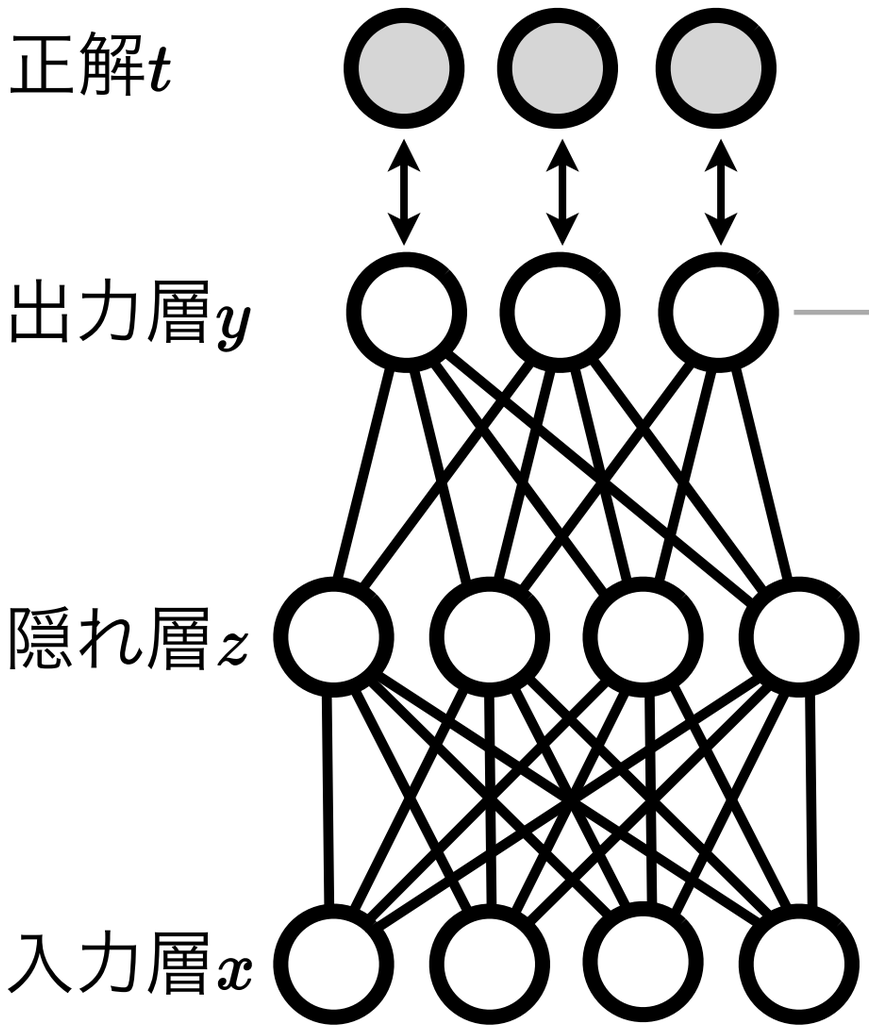
$$E = -\sum_{k=1}^{K} t \log y + (1-t)\log(1-y)$$

$$E = -t \log y - (1-t)\log(1-y)$$

$$E = -\sum_{k=1}^{K} t_k \log y_k$$

いずれも予測と正解が
違うほど大きくなる

Back Propagation

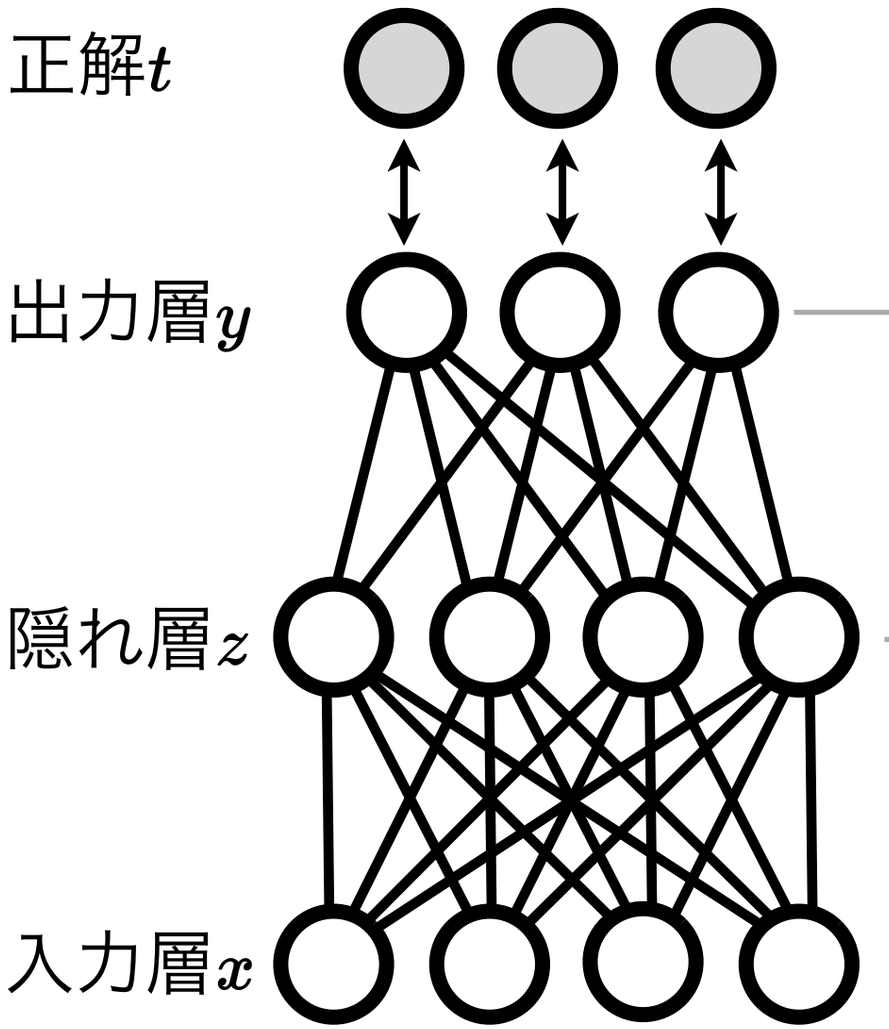


ノードの誤差を計算

$$\delta_y = y - t$$

出力ラベルと正解の差

Back Propagation



ノードの誤差を計算

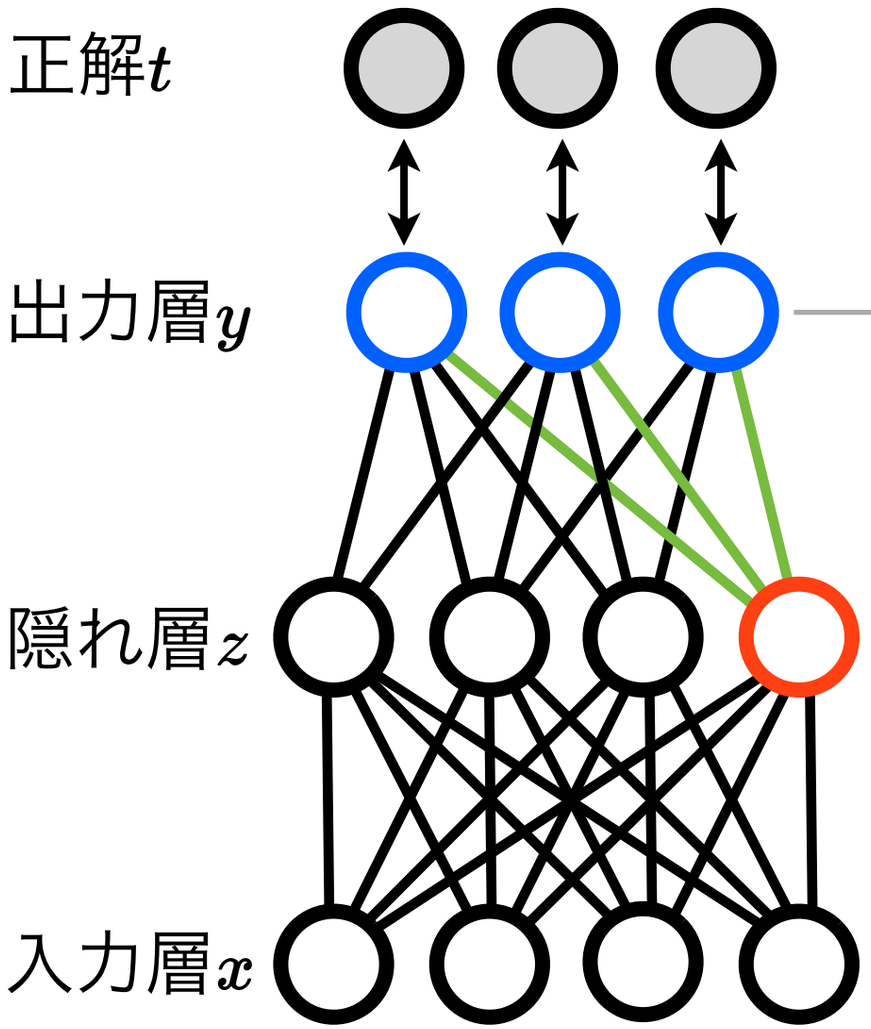
$$\delta_y = y - t$$

出力ラベルと正解の差

$$\delta_z = W_2^T \delta_y f'(a_z)$$

自分が情報を伝えた先の
誤差が伝播してくる

Back Propagation



ノードの誤差を計算

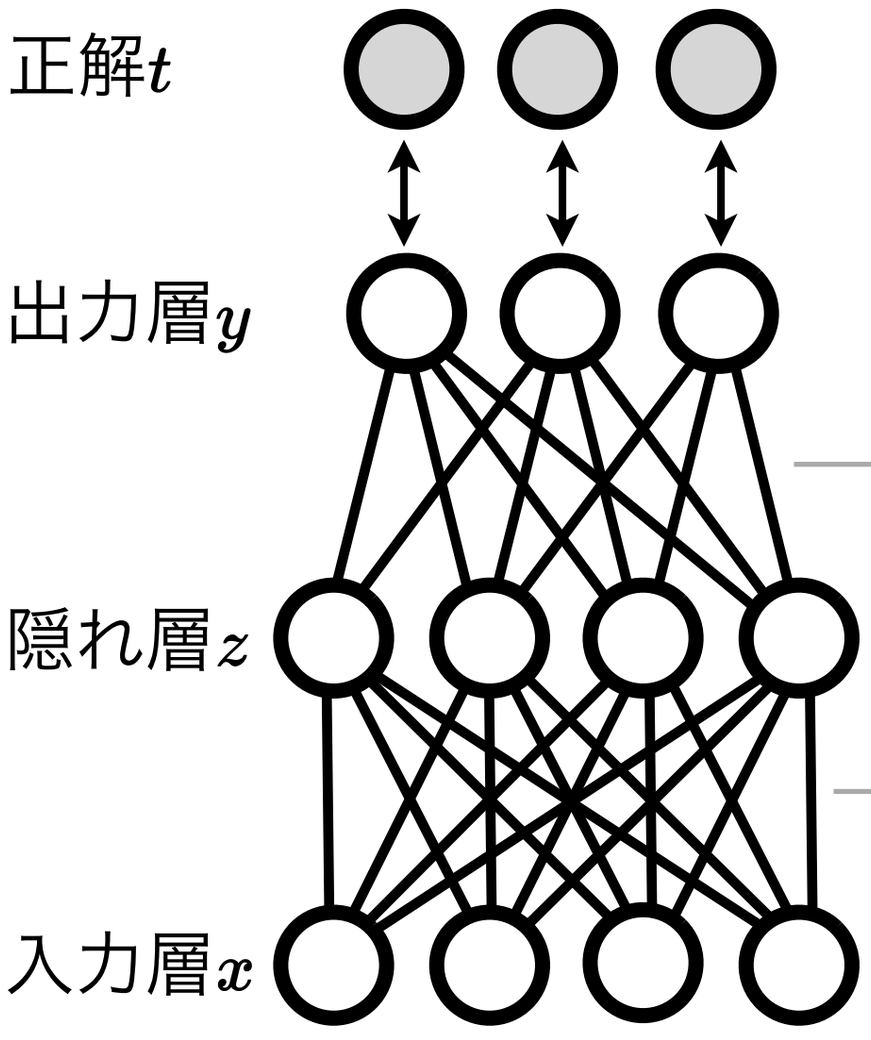
$$\delta_y = y - t$$

出力ラベルと正解の差

$$\delta_z = W_2^T \delta_y f'(a_z)$$

自分の影響で上で発生した誤差

Back Propagation



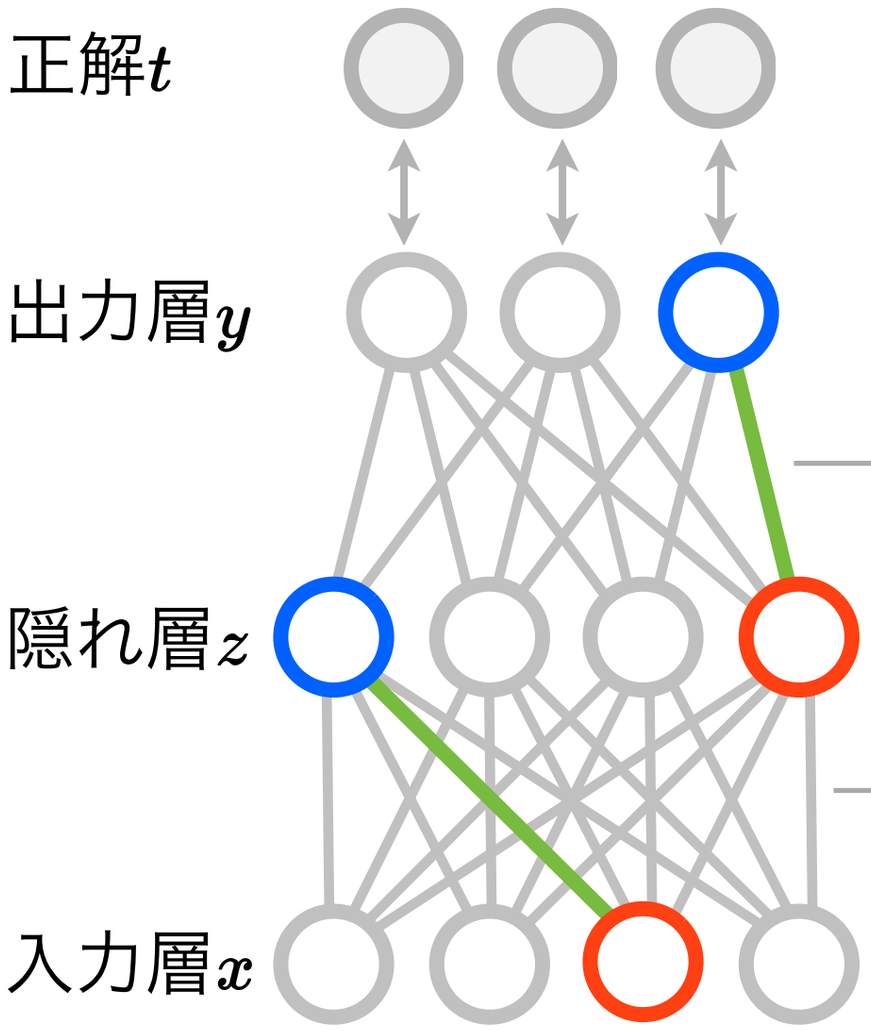
重みの勾配を計算

$$\frac{\partial E_n}{\partial W_2} = \delta_y z^T$$

$$\frac{\partial E_n}{\partial W_1} = \delta_z x^T$$

自分が上に伝えた
情報で発生した誤差

Back Propagation



重みの勾配を計算

$$\frac{\partial E_n}{\partial W_2} = \delta_y z^T$$

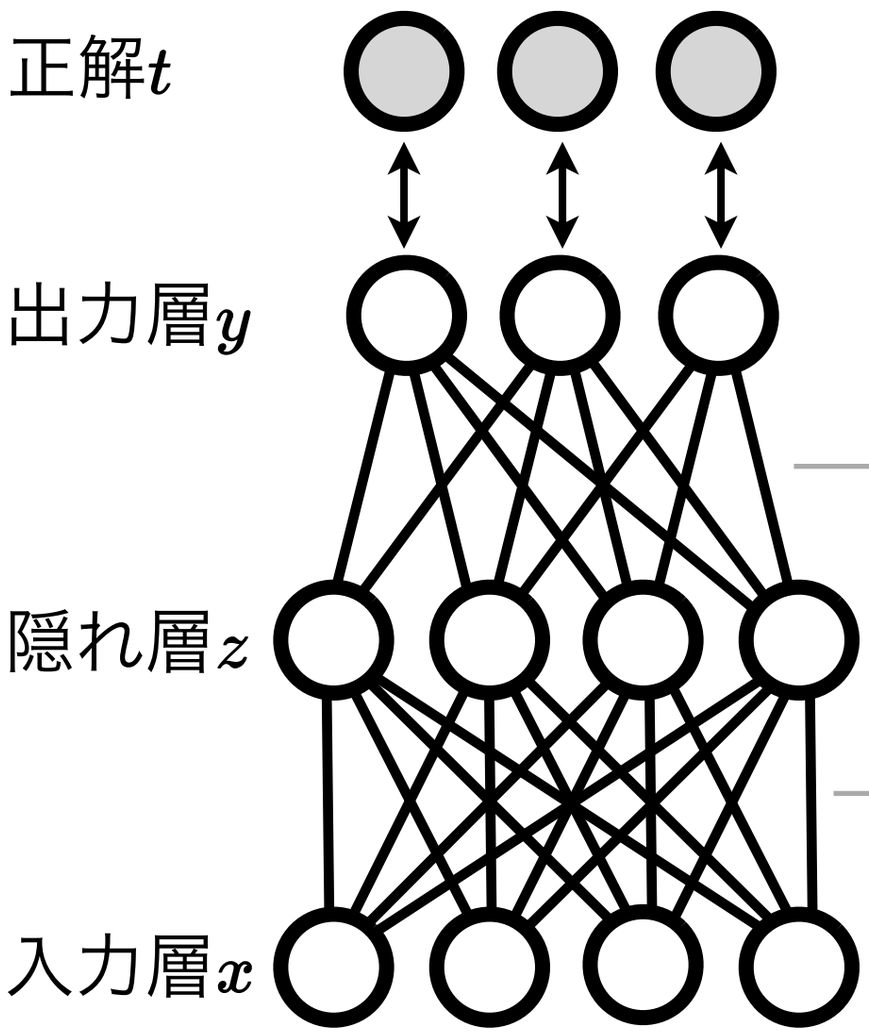
$$\frac{\partial E_n}{\partial W_1} = \delta_z x^T$$

自分が上に伝えた
情報で発生した**誤差**

Neural Networkの処理

- Forward propagation
- Back propagation
- **Parameter update**

Update parameters

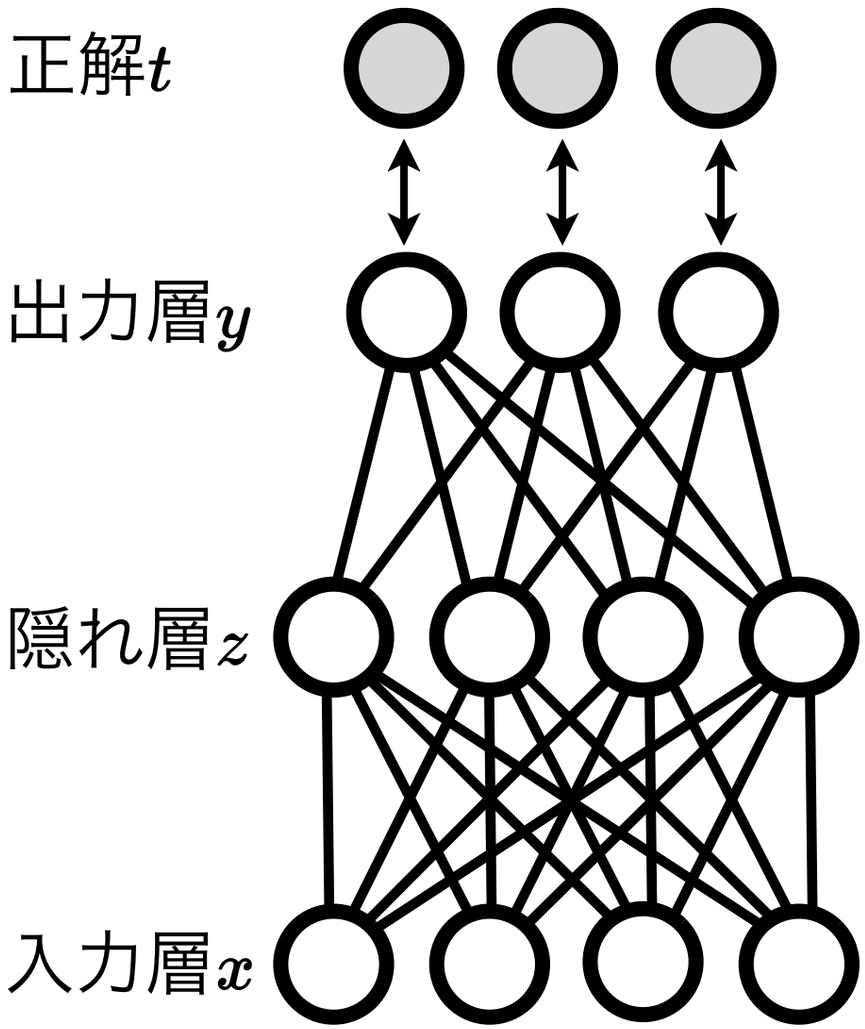


重みの更新

$$W_2 = W_2 - \alpha \frac{\partial E_n}{\partial W_2}$$

$$W_1 = W_1 - \alpha \frac{\partial E_n}{\partial W_1}$$

Update parameters



重みの更新

$$W_2 = W_2 - \alpha \frac{\partial E_n}{\partial W_2}$$

$$W_1 = W_1 - \alpha \frac{\partial E_n}{\partial W_1}$$

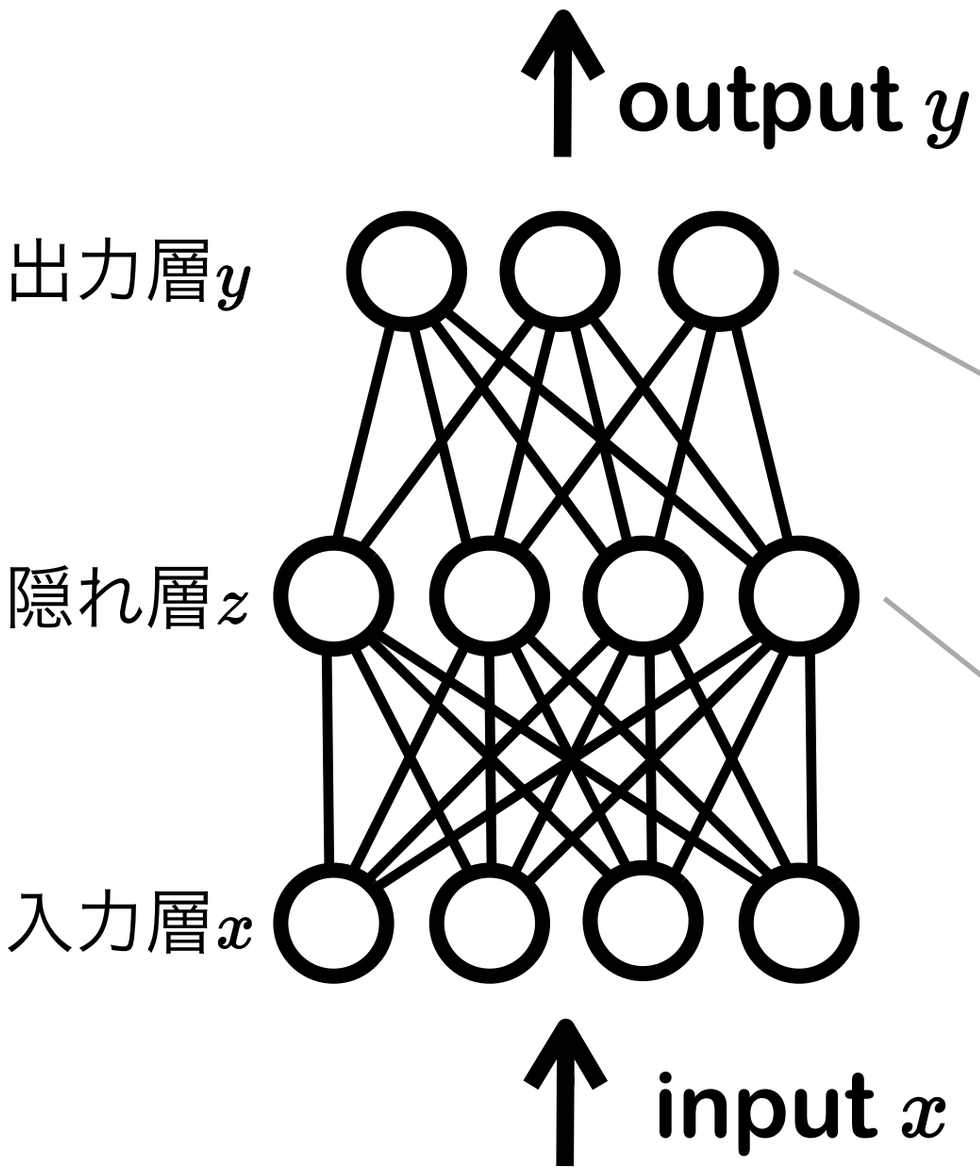
-Gradient Descent
-Stochastic Gradient Descent
-SGD with mini-batch
修正するタイミングの違い

Neural Network

の処理まとめ

Forward Propagation

入力から予測

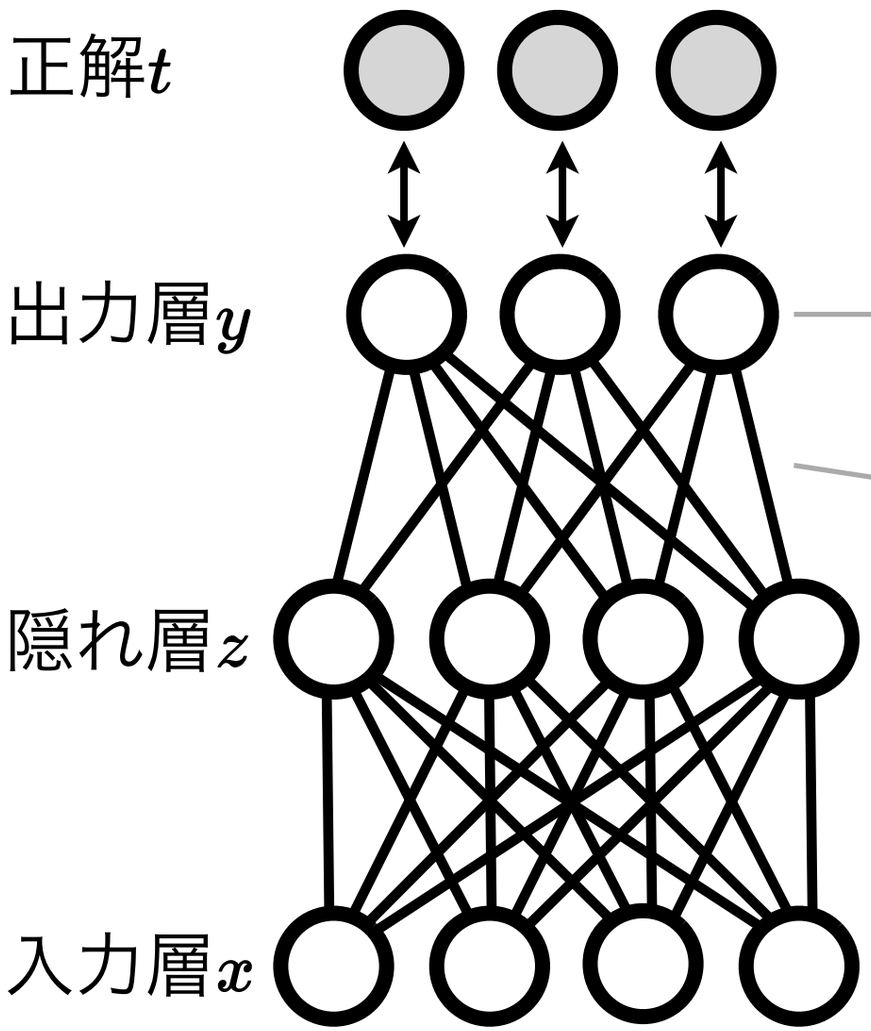


$$y = \sigma(W_2 z + b_2)$$

$$z = f(W_1 x + b_1)$$

Back Propagation

誤差と勾配を計算



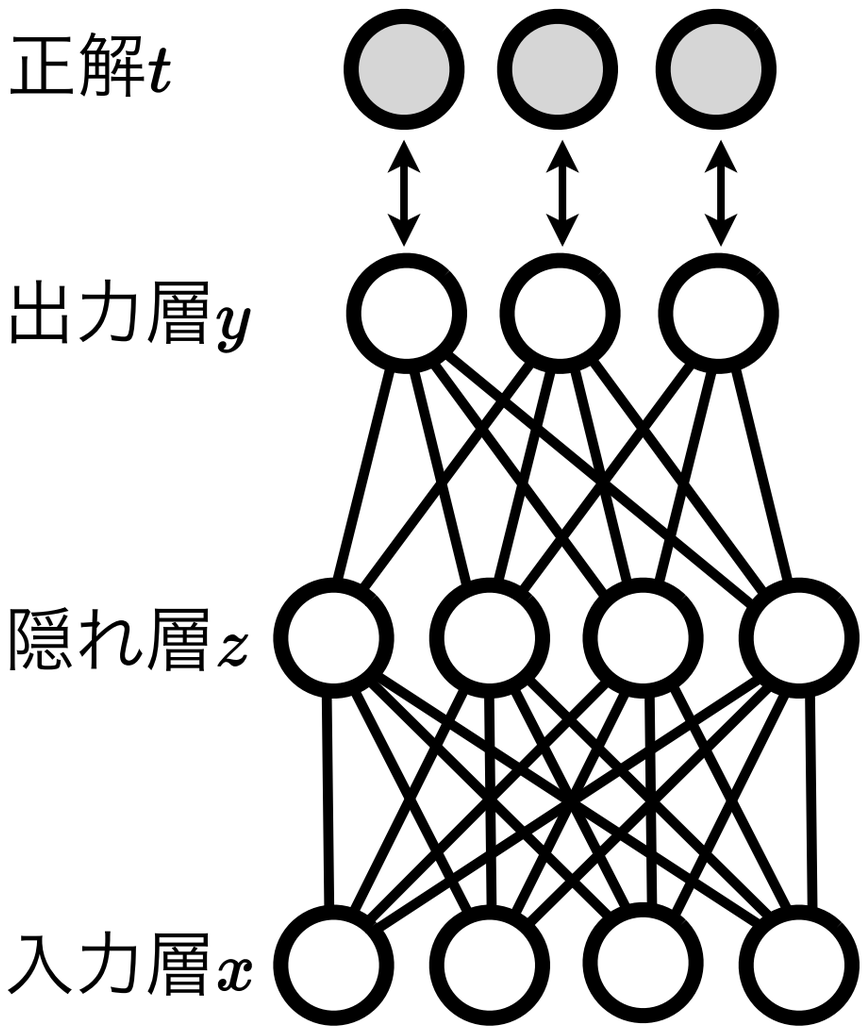
$$\delta_y = y - t$$

$$\frac{\partial E_n}{\partial W_2} = \delta_y z^T$$

$$\delta_z = W_2^T \delta_y f'(a_z)$$

$$\frac{\partial E_n}{\partial W_1} = \delta_z x^T$$

Update parameters



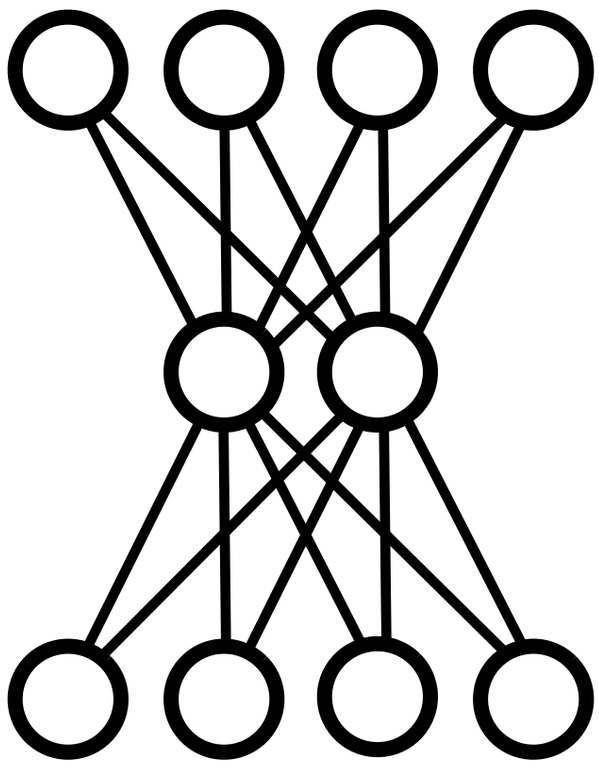
勾配方向へ重み更新

$$W_2 = W_2 - \alpha \frac{\partial E_n}{\partial W_2}$$

$$W_1 = W_1 - \alpha \frac{\partial E_n}{\partial W_1}$$

ちなみに**Autoencoder**

Neural Networkの特殊系



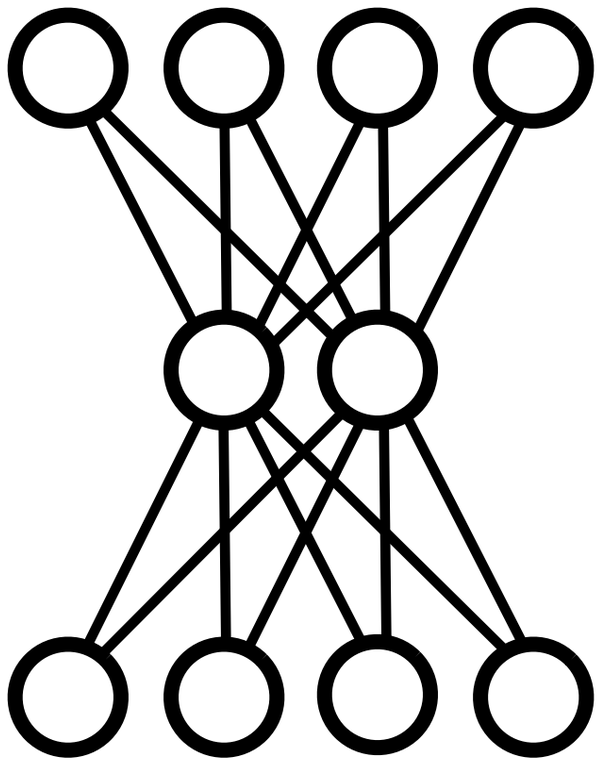
1. 入力と出力の次元が同じ
2. 教師信号が入力そのもの

入力を圧縮※1して復元

※1 圧縮(隠れ層が入力層より少ない)でなくても，適切に正則化すればうまくいく

Autoencoder

Neural Networkの特殊系

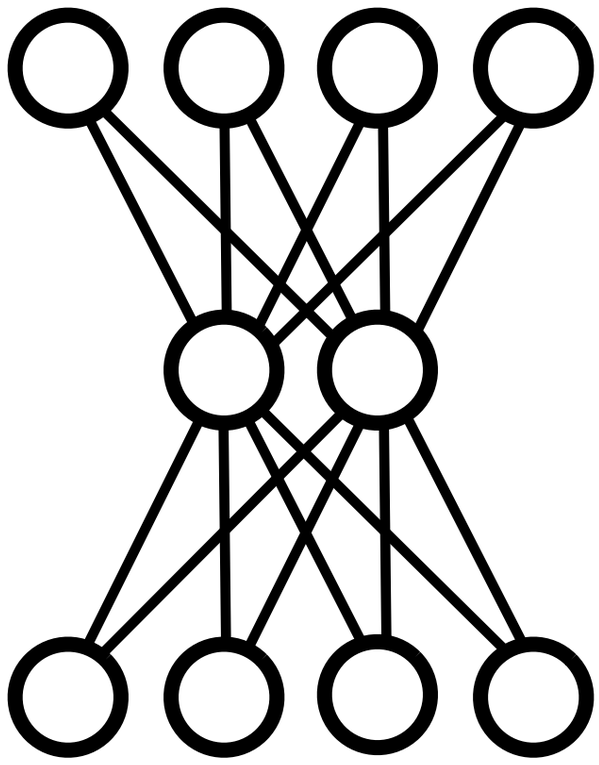


$$y = \sigma(W_2 z + b_2)$$

$$z = f(W_1 x + b_1)$$

Autoencoder

Neural Networkの特殊系



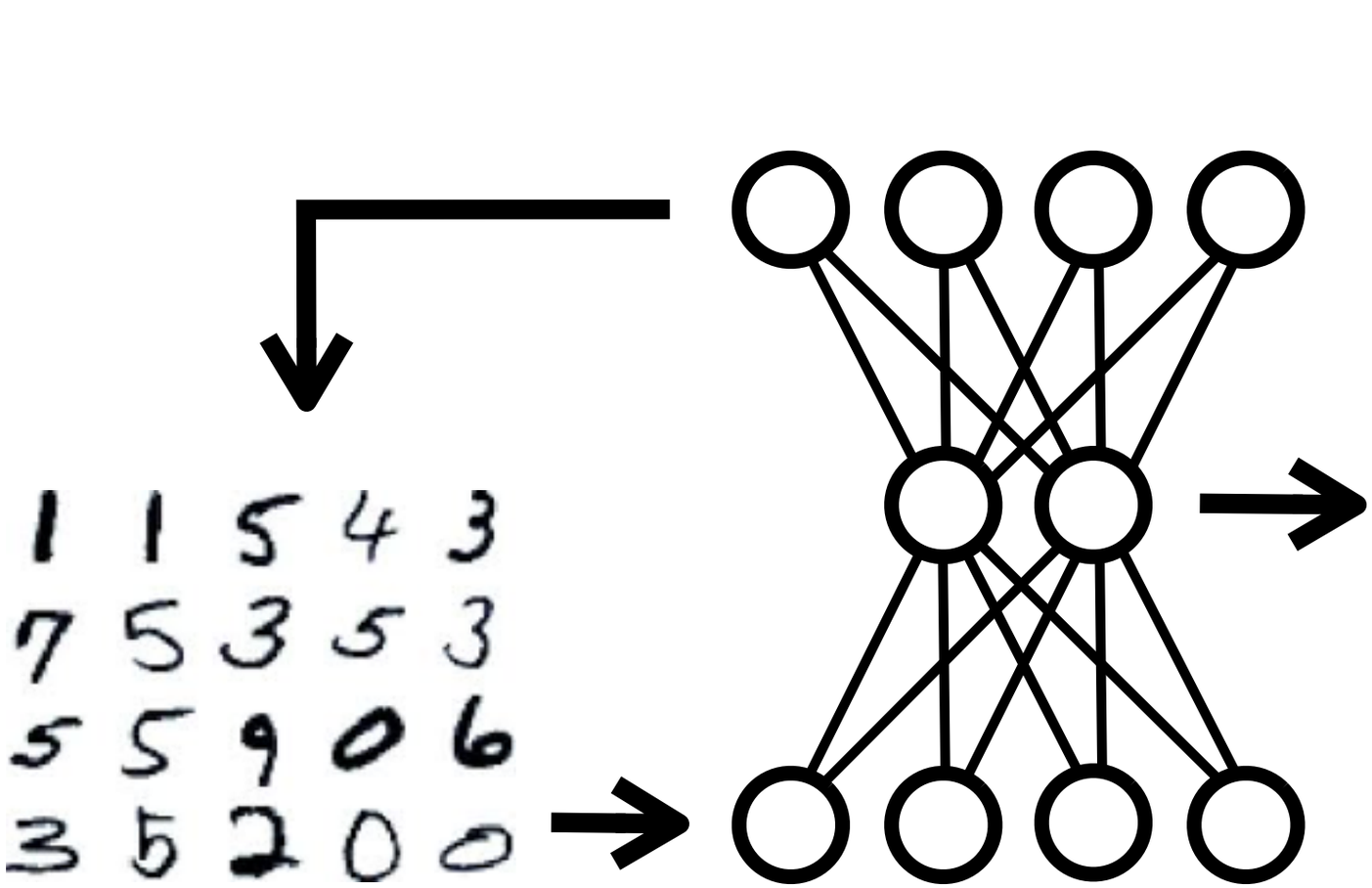
$$y = \sigma(W_2 z + b_2)$$

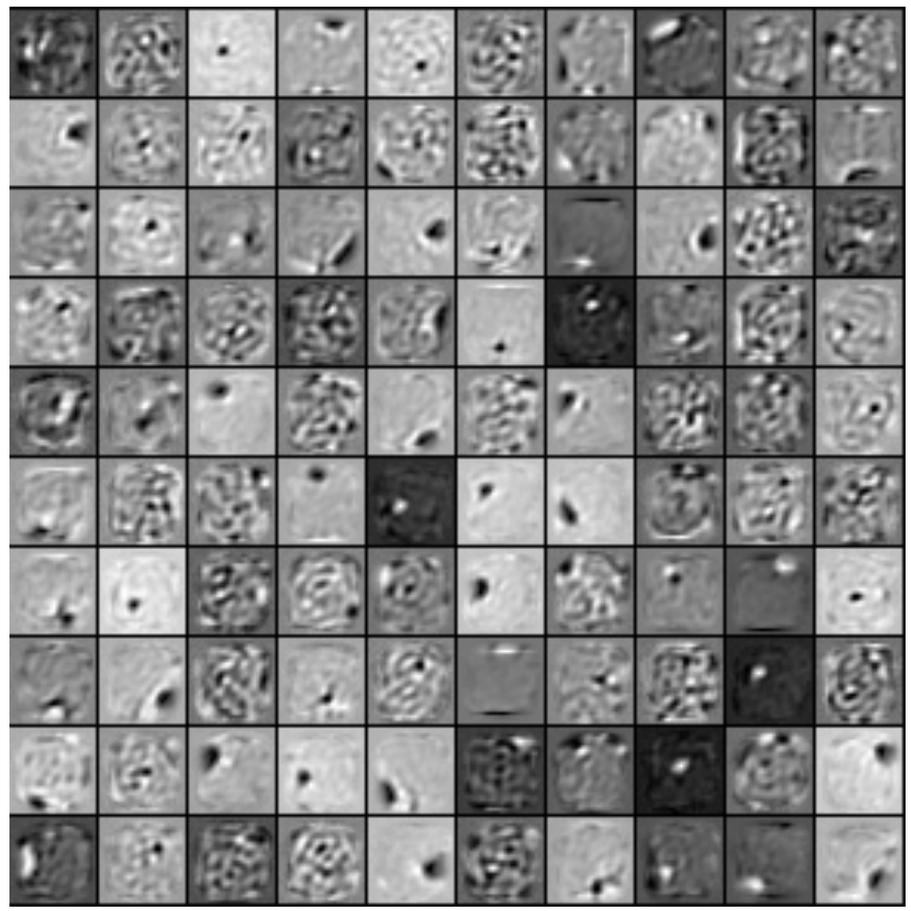
マルチラベリングのケースに該当

element-wiseで
Sigmoid関数 $\sigma(a) = \frac{1}{1+\exp(-a)}$

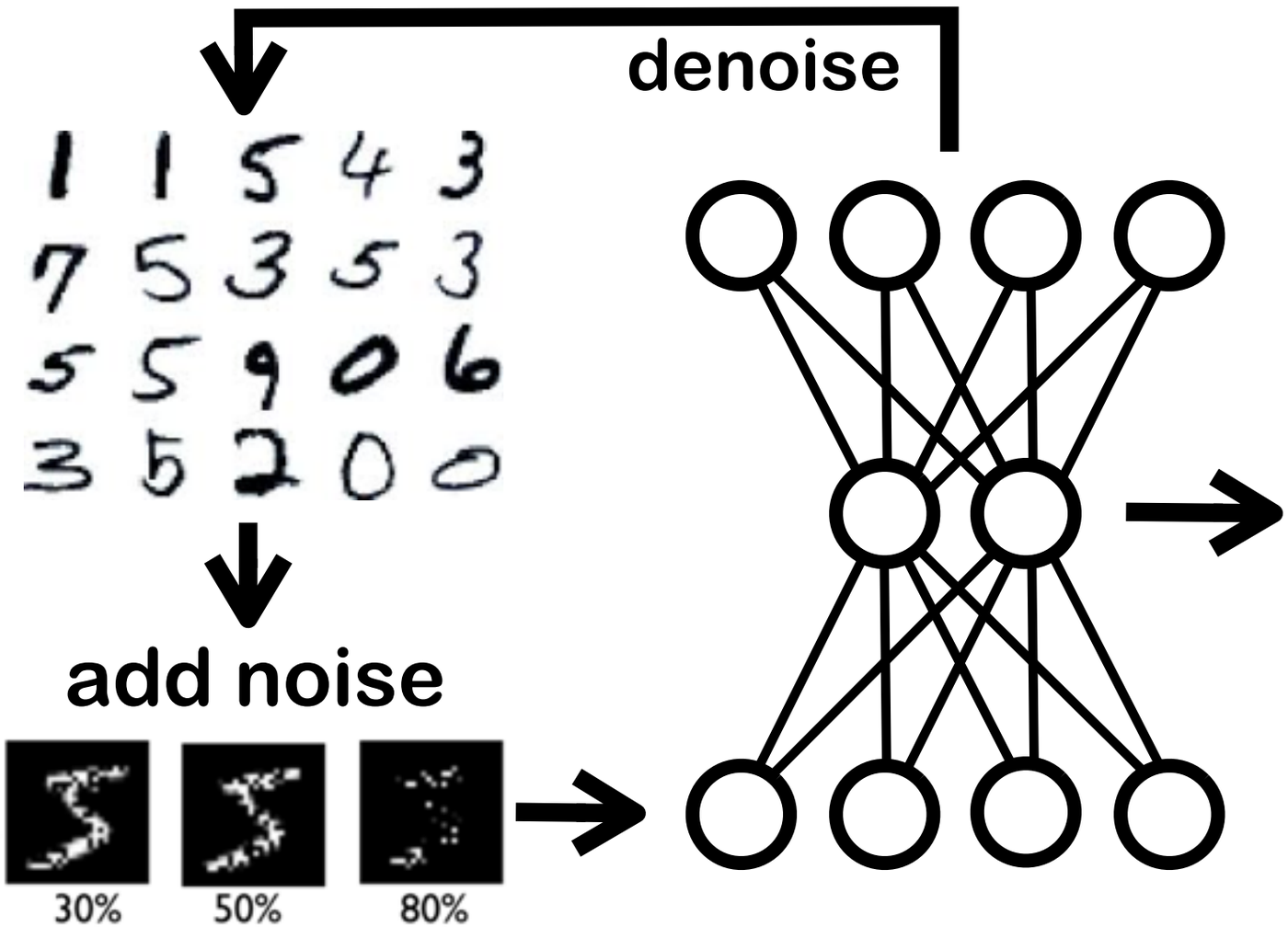
画像の場合，各画素(ユニット)ごとに明るさ(**0.0**:黒, **1.0**:白)を判定するため

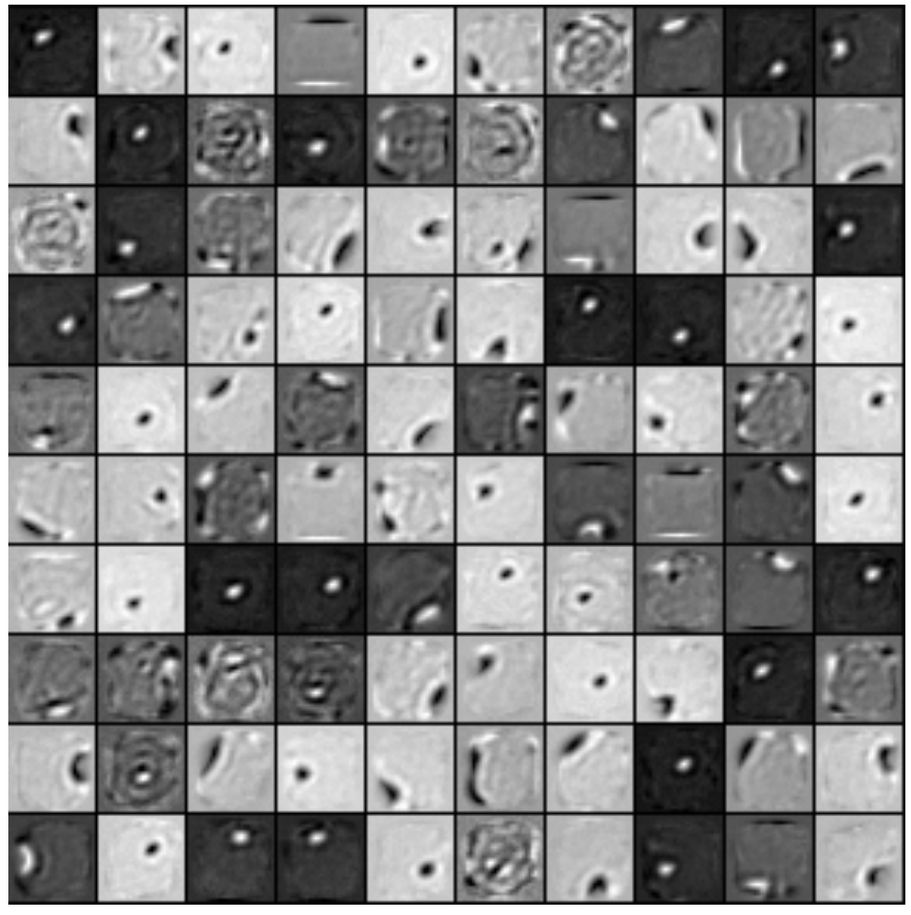
Autoencoderの学習するもの





Denoising Autoencoder





正則化法の一つ，再構築+ノイズの除去

Deep Learning

Deep Learning概要

Neural Networkふんわり

Deepへの難しさ

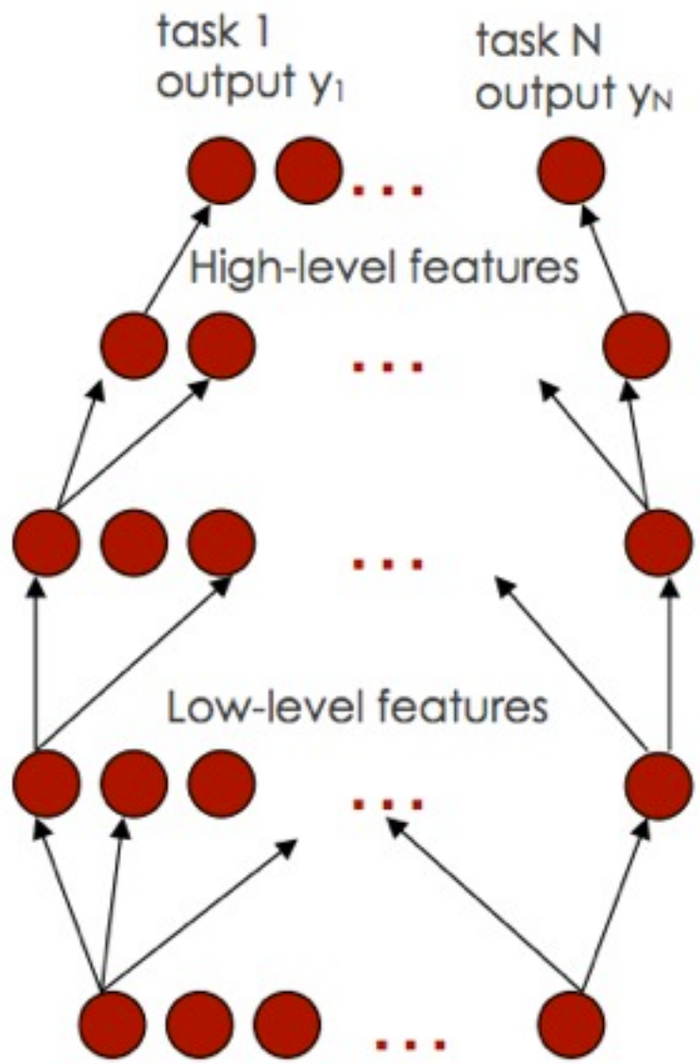
Pretrainingの光

Stacked Autoencoder , DBN

Deepになると?

many figures from
http://www.cs.toronto.edu/~fleet/courses/cifarSchool09/slidesBengio.pdf

仕組み的には同じ

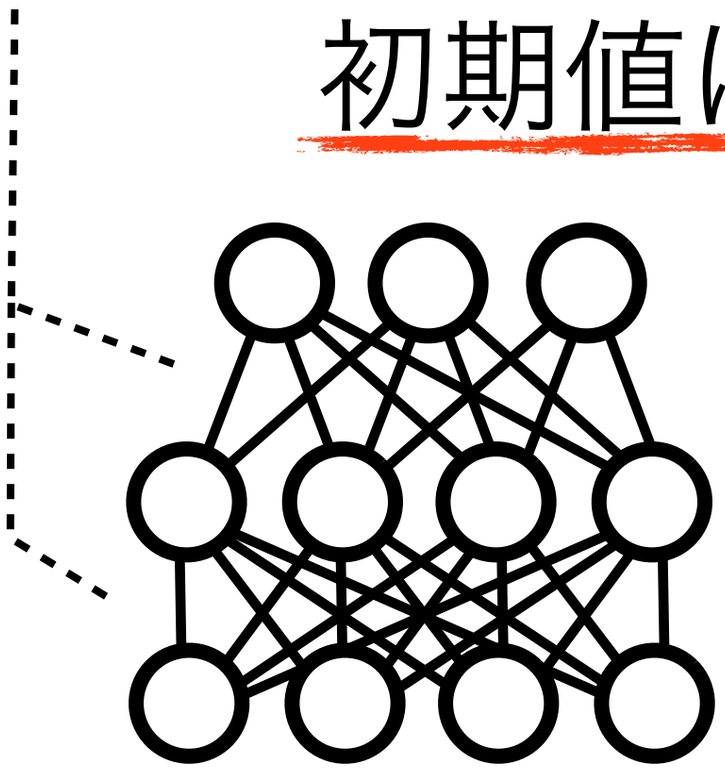


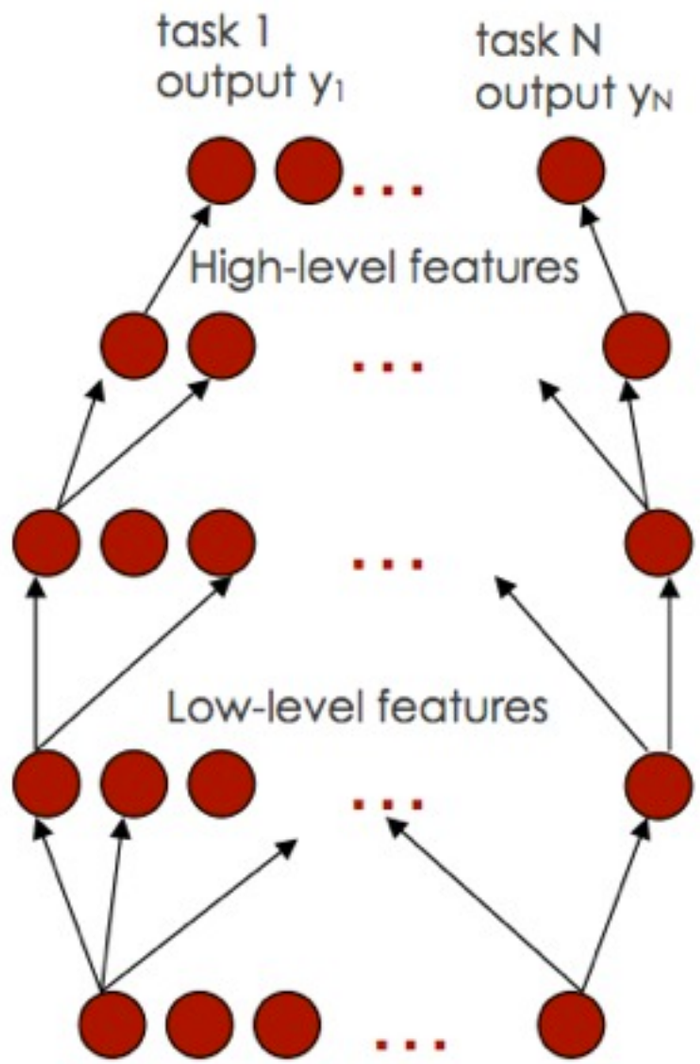
隠れ層が増えただけ

問題は初期化

NNのパラメータ

初期値は乱数





多層**(Deep)**になっても**OK?**

乱数だとうまくいかない

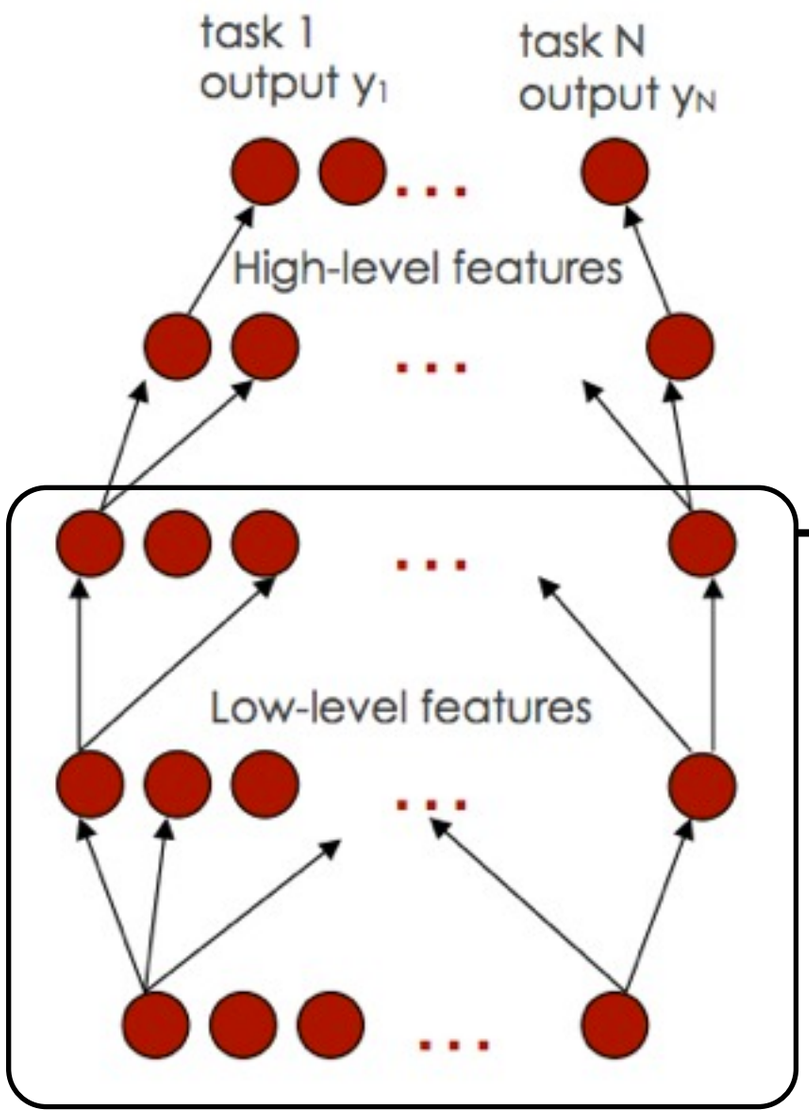
Many unreported negative observations as well as the experimental results in [17, 50] suggest that gradient-based training of deep supervised multi-layer neural networks (starting from random initialization) gets stuck in "apparent local minima or plateaus",[1] and that as the architecture gets deeper, it becomes more difficult to obtain good generalization. When starting from random initialization, the solutions

Learning Deep Architectures for AI (2009)

NNはかなり複雑な変化をする関数なので

悪い局所解にいっちゃう

乱数だとうまくいかない



NN自体が表現力高いので

上位二層分の**NN**だけで訓練データを

再現するには事足りちゃう

inputのランダムな写像だが,

inputの情報は保存している

ただしそれは汎化能力なし

過学習

Greedy Layer-Wise Training of Deep Networks [Bengio+, 2007]

Deep Learning

Deep Learning概要

Neural Networkふんわり

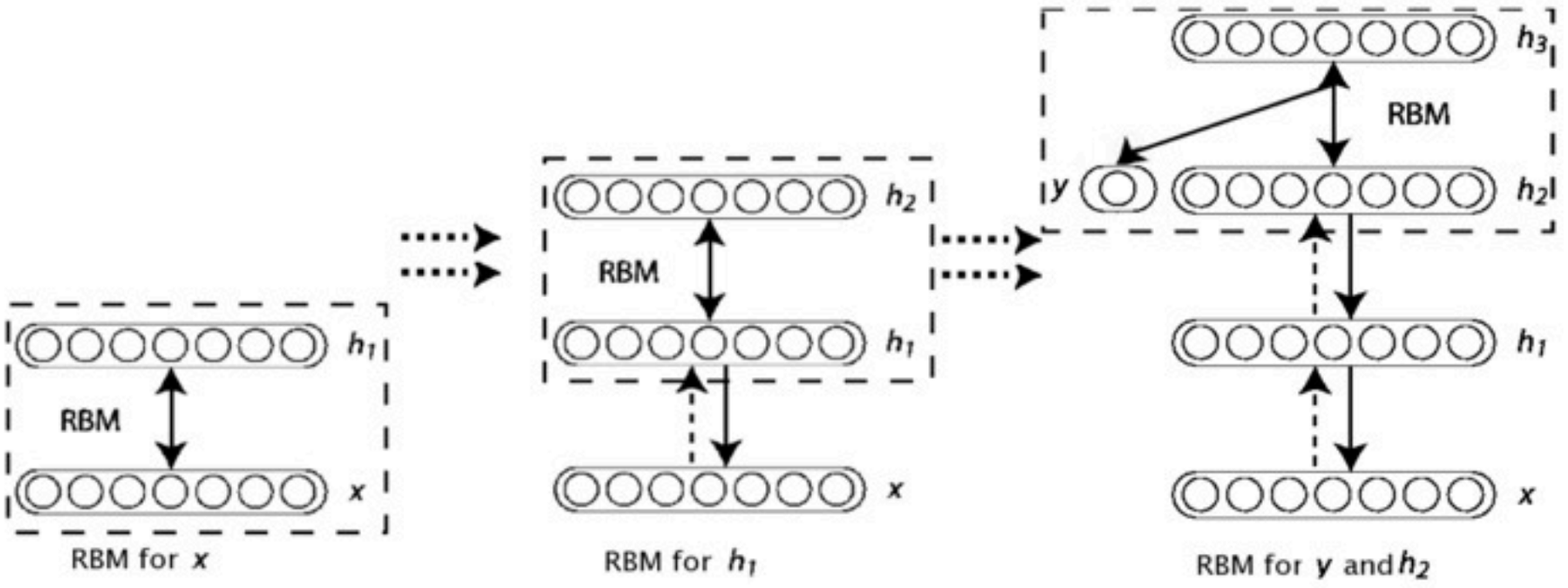
Deepへの難しさ

Pretrainingの光

Stacked Autoencoder , DBN

Greedy Layer-wise unsupervised pretraining

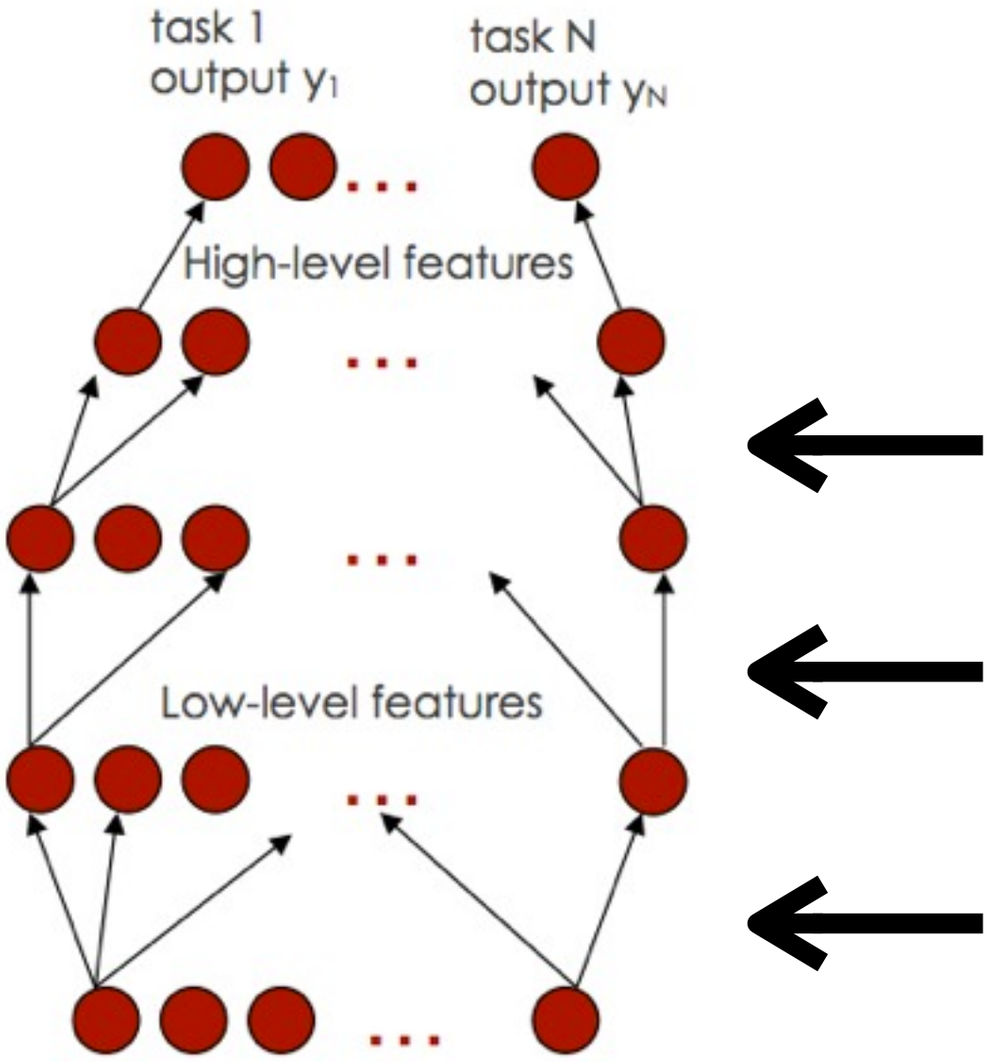
Greedy Layer-Wise Pre-Training



Stacking Restricted Boltzmann Machines (RBM) → Deep Belief Network (DBN)
→ Supervised deep neural network

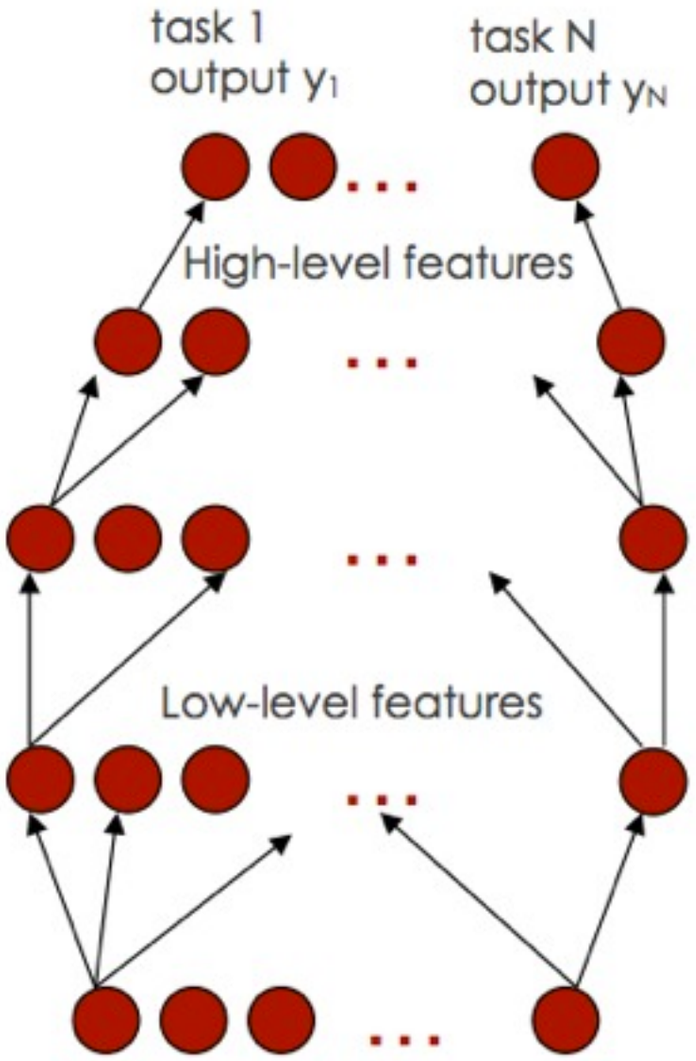
2006年，ブレークスルー(Hinton+,2006)

層ごとにまずパラメータを更新



層ごとに学習

層ごとにまずパラメータを更新



どうやって？

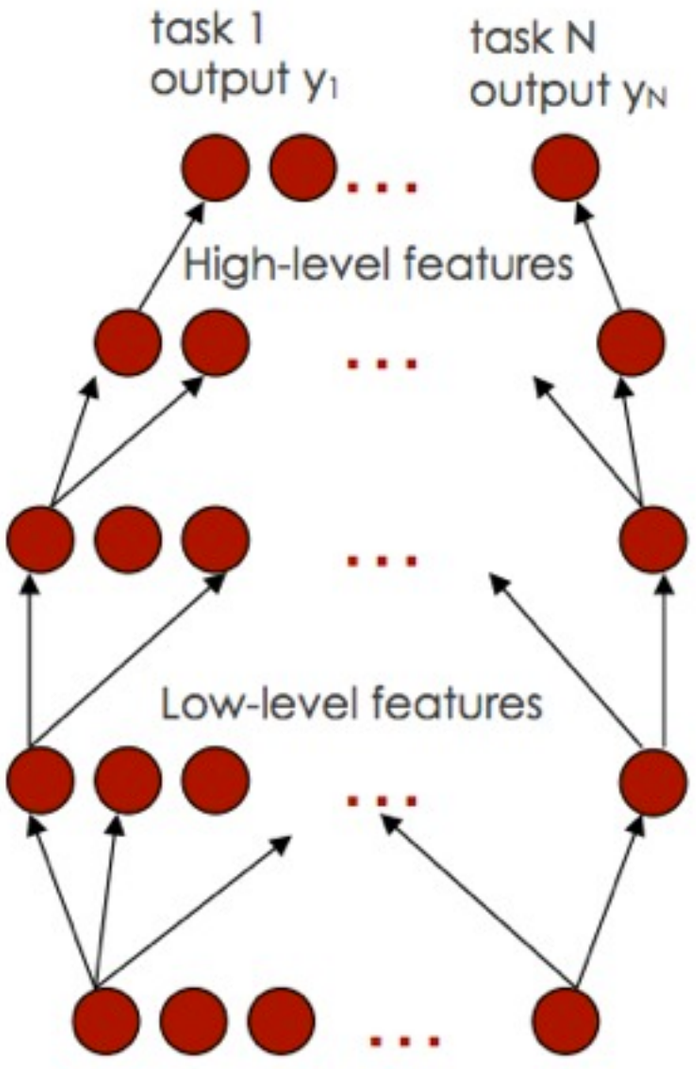
Autoencoder !!

[Bengio,2007]

RBMも

[Hinton,2006]

層ごとにまずパラメータを更新



どうなるの？

良い初期値を

得られるようになりました！

[Bengio+, 2007]

なぜ**pre-training**が良いのか，諸説あり

Why does Unsupervised Pre-training Help Deep Learning ? [Erhan+, 2010]

つまり

Deep Learningとは

良い初期値を（手に入れる方法を）

手に入れた※1

Neural Network※2

※1 諸説あり **Why does Unsupervised Pre-training Help Deep Learning ? [Erhan+, 2010]**

※2 **stacked autoencoder**の場合

Deep Learning

Deep Learning概要

Neural Networkふんわり

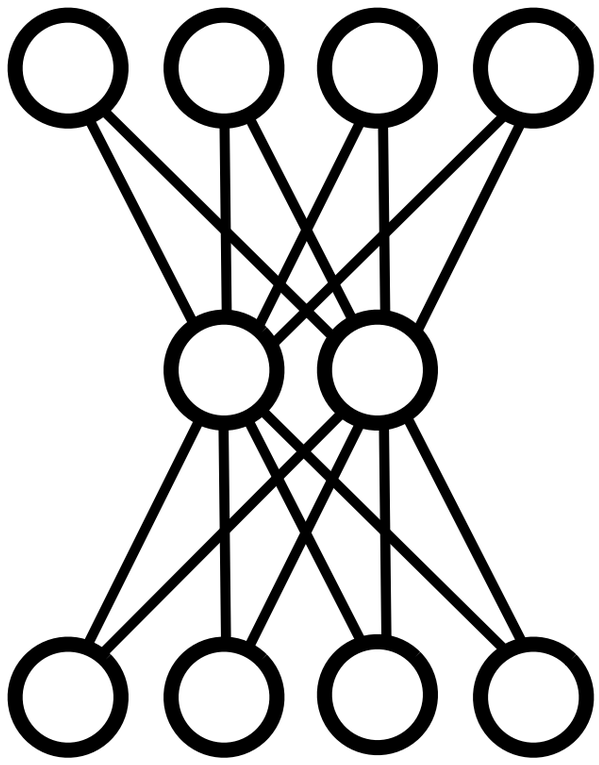
Deepへの難しさ

Pretrainingの光

Stacked Autoencoder , DBN

Autoencoder

Neural Networkの特殊系



1. 入力と出力の次元が同じ
2. 教師信号が入力そのもの

入力を圧縮して復元

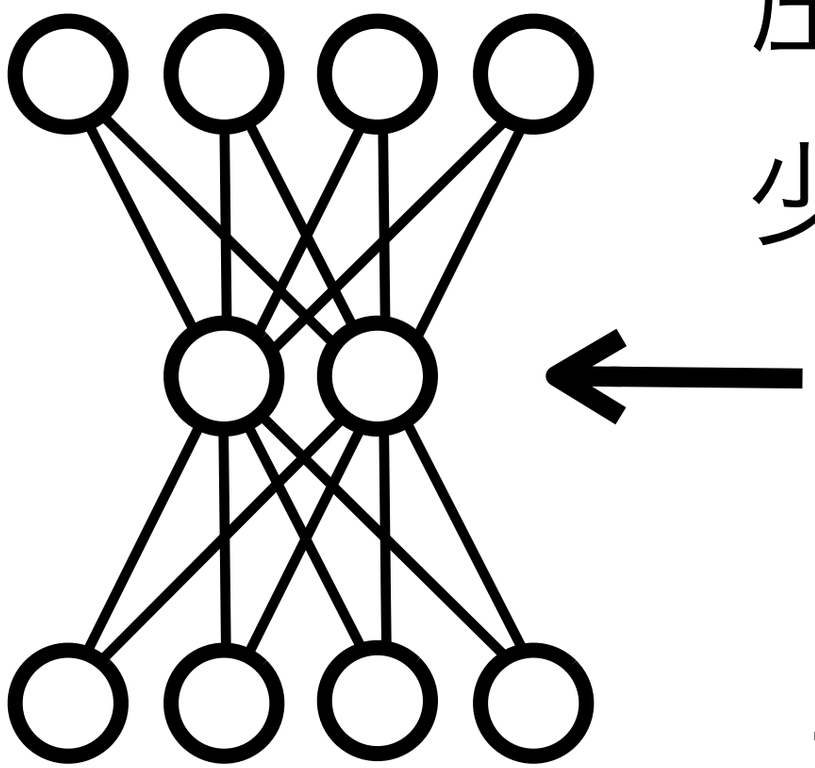
↓

訓練データ中の

本質的な情報を捉える

Autoencoder

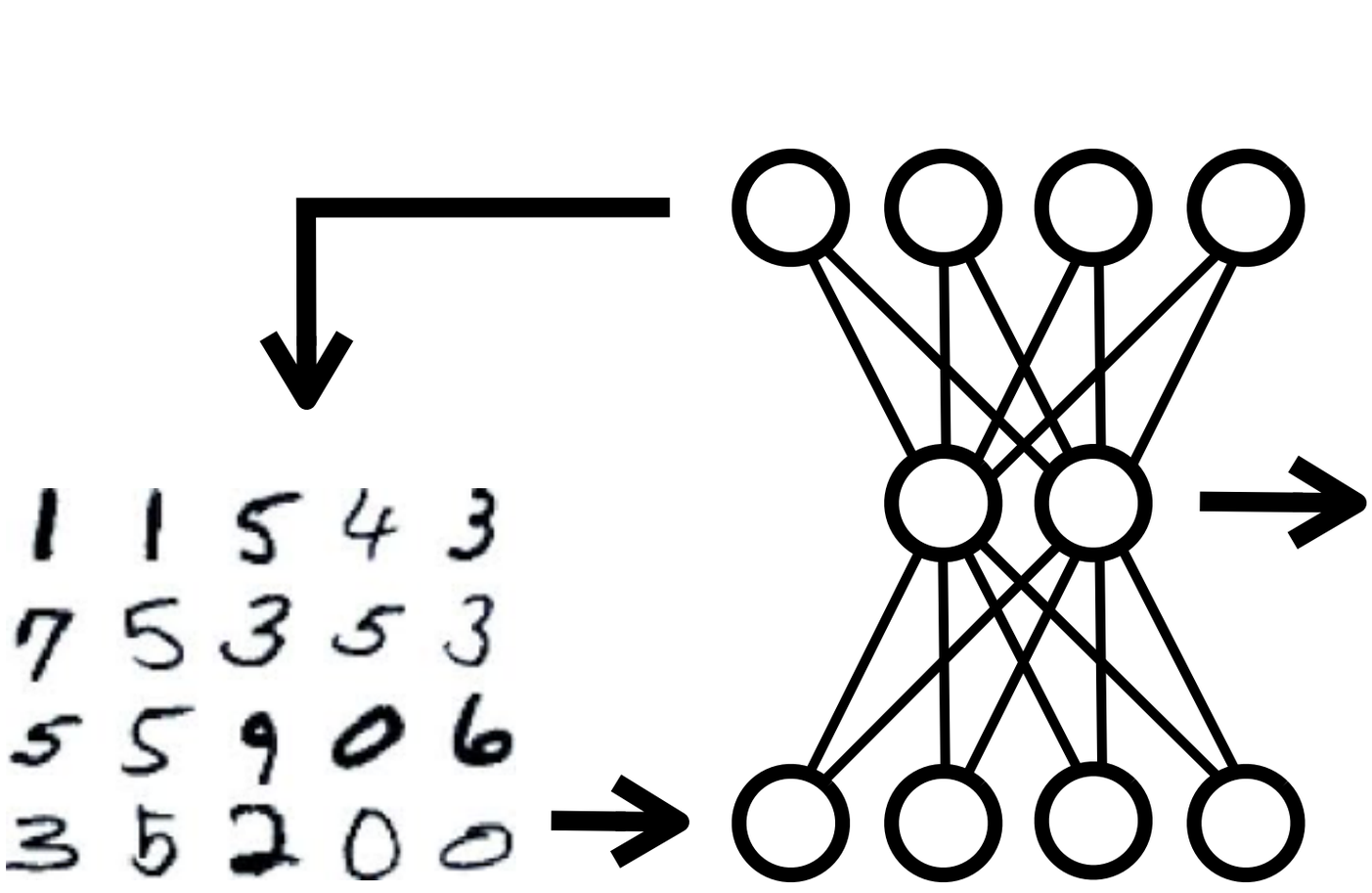
Neural Networkの特殊系

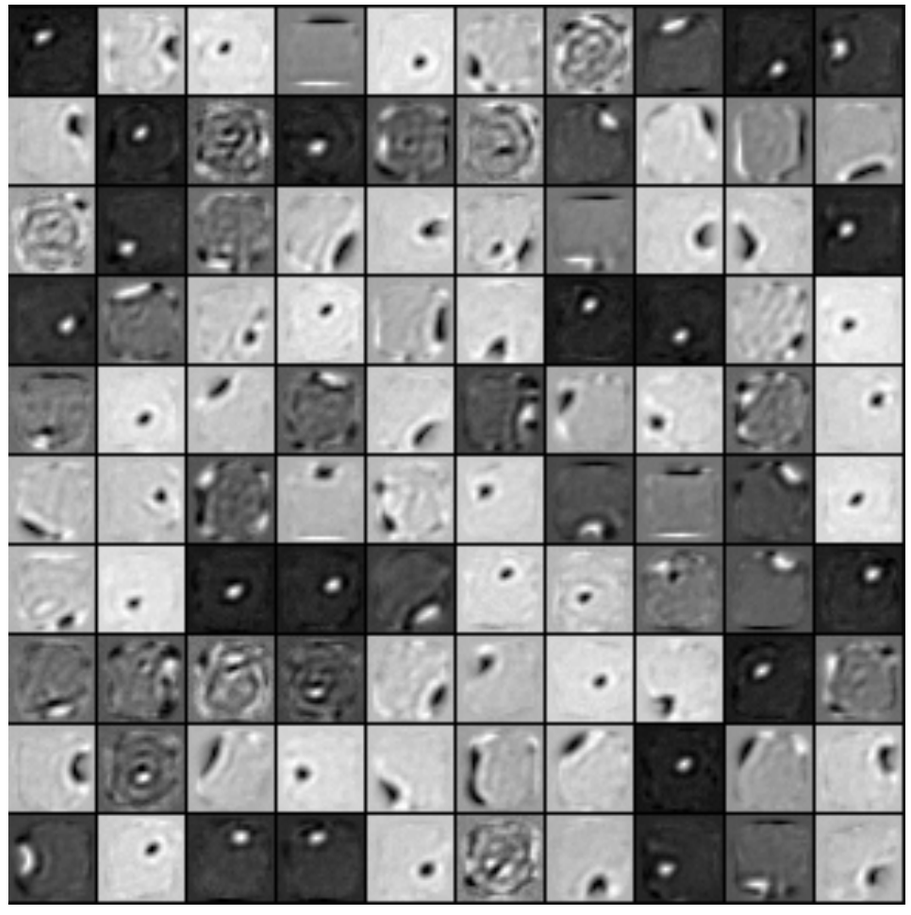


圧縮ということは隠れ層は
少なくないといけないの？

そうでなくても,
正則化などでうまくいく

Autoencoderの学習するもの



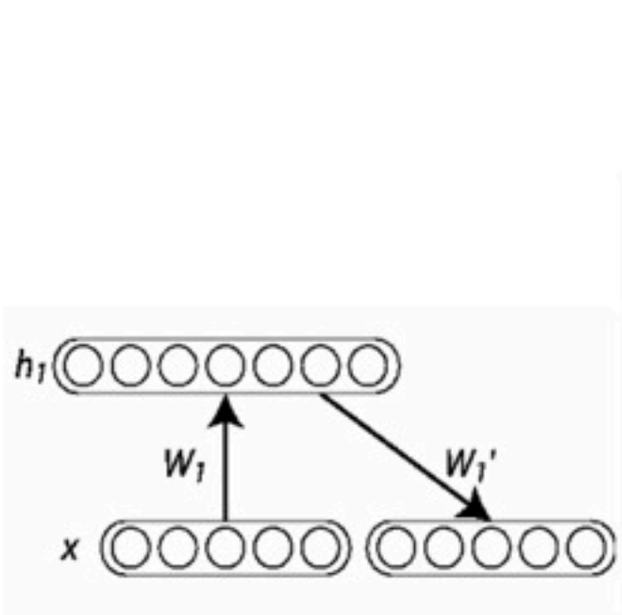


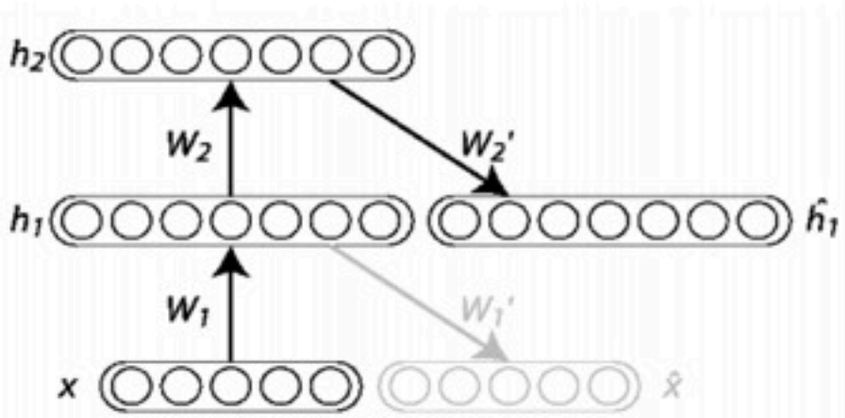
これは，正確には**denoising autoencoder**の図

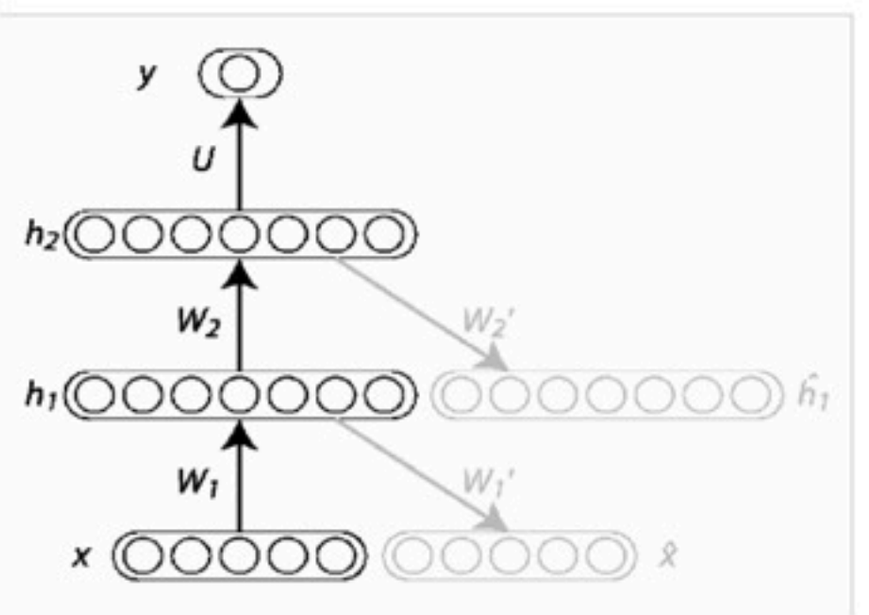
http://kiyukuta.github.io/2013/08/20/hello_autoencoder.html

Stacked Autoencoder

Stacking Auto-Encoders

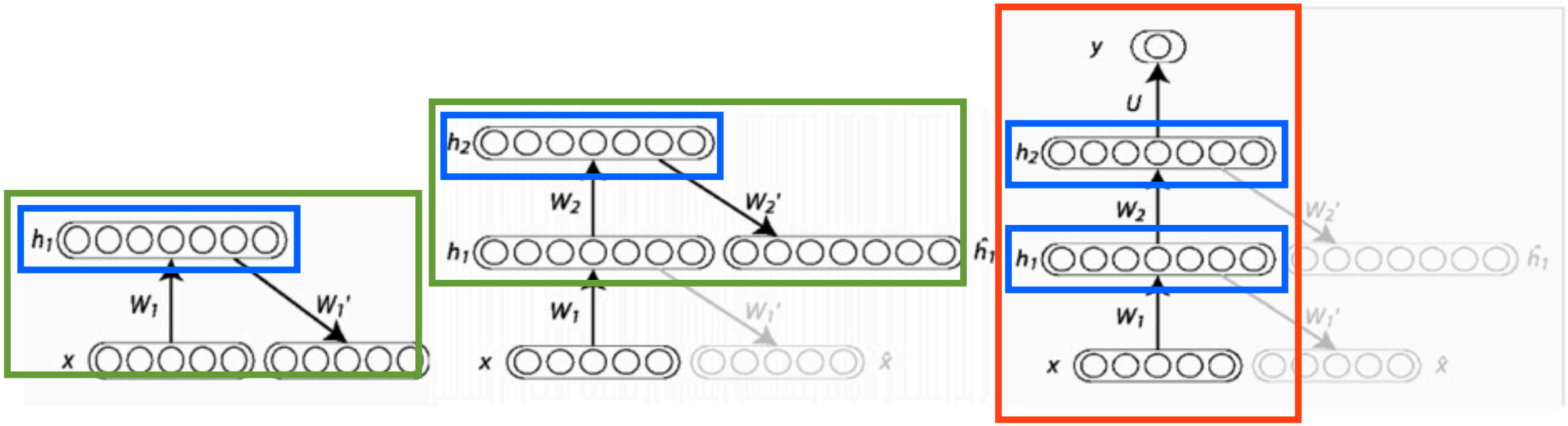






Stacked Autoencoder

Stacking Auto-Encoders



このNNの**各層**を，

その層への入力を再構築する**Autoencoder**

として，事前学習

Deep Learning

Deep Learning概要

Neural Networkふんわり

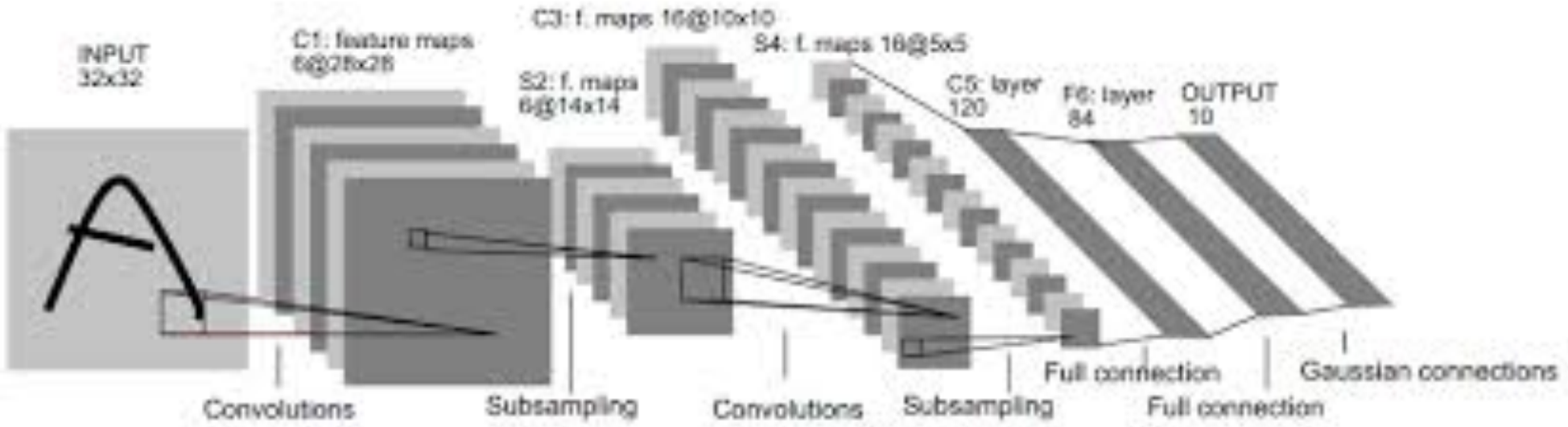
Deepへの難しさ

Pretrainingの光

Stacked Autoencoder , DBN

Deep Learning for NLP

画像処理のように



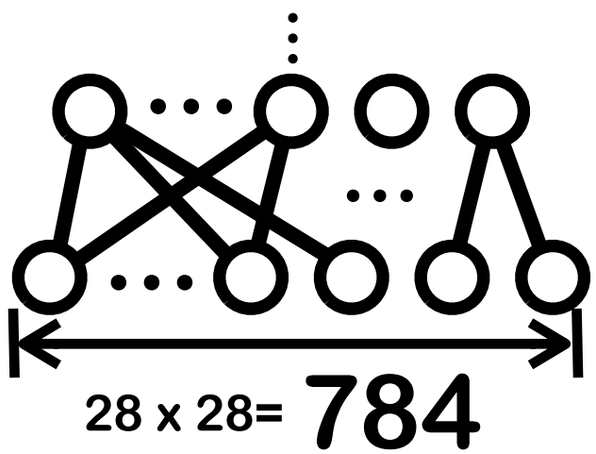
Deeeeeeeepって感じではない

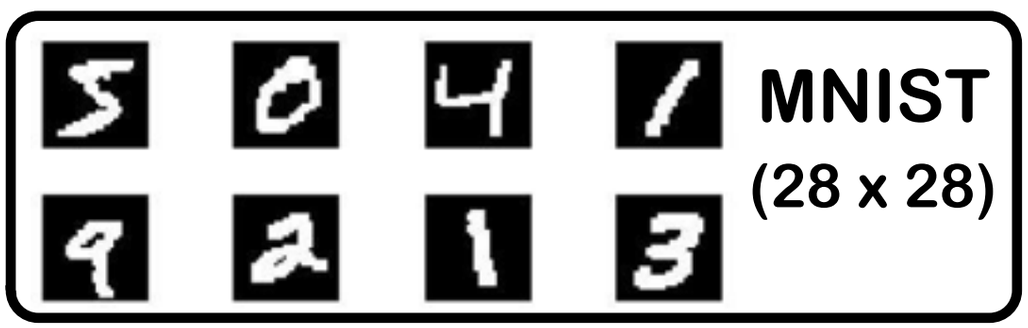
Neural Network-based
くらいのつもりで

Deep Learning for NLP

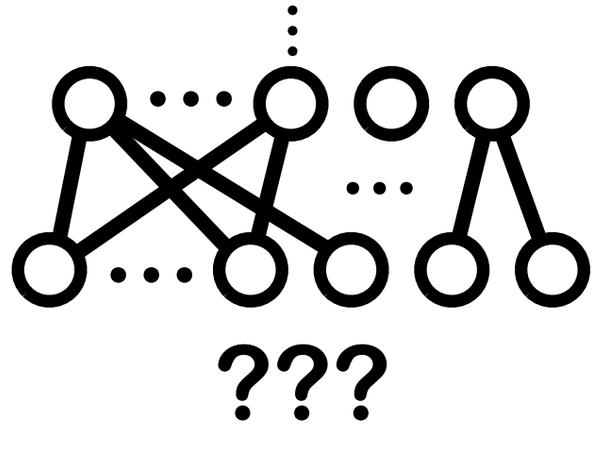
Input size

Image





Sentence

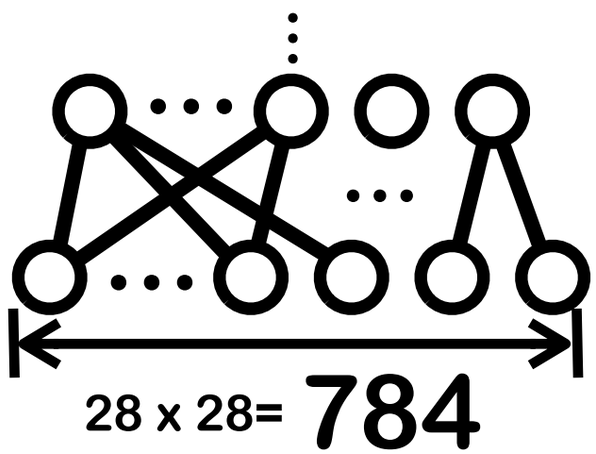


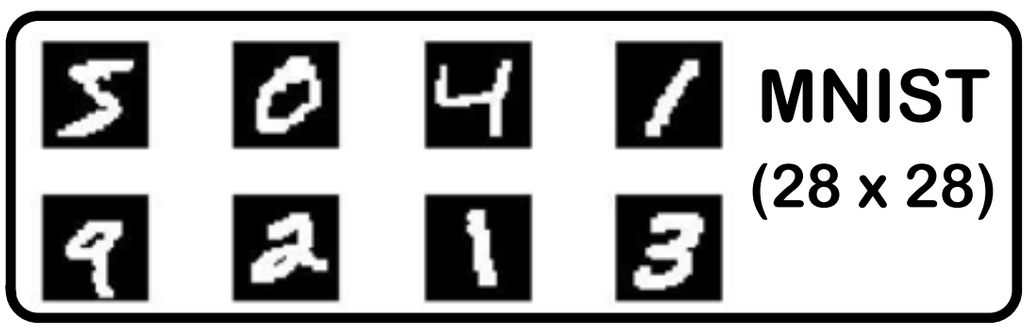
任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

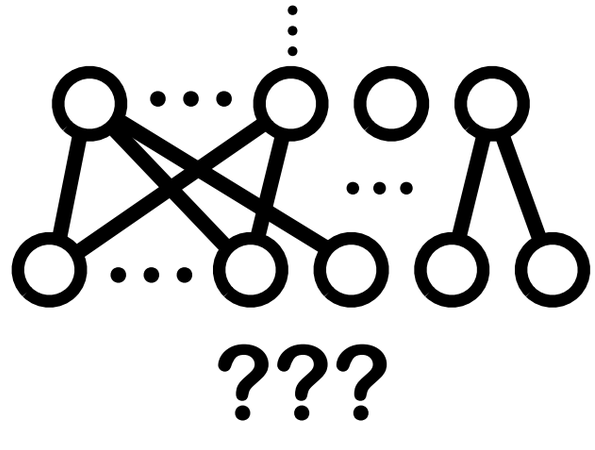
Input representation

Image





Sentence



任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

言い換えると

任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

NLPでNNを使いたい

単語の特徴をうまく捉えた表現の学習

Keywords

任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

- convolutional-way
- recursive-way

Distributed word representation

phrase, sentence-level representation

Neural language model

Keywords

任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

- convolutional-way
- recursive-way

Distributed word representation

phrase, sentence-level representation

Neural language model

Keywords

任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

- convolutional-way
- recursive-way

Distributed word representation

phrase, sentence-level representation

Neural language model

Keywords

任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

- convolutional-way

- recursive-way

↓

phrase, sentence-level representation

Distributed word representation

Neural language model

Keywords

任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

- convolutional-way
- recursive-way

phrase, sentence-level representation

Distributed word representation

Neural language model

Word representation

自然言語処理における

単語の表現方法

↓

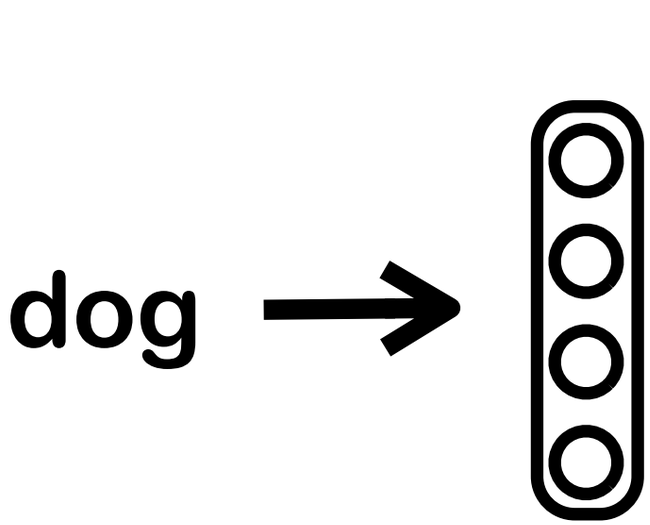
ベクトル

(Vector Space Model, VSM)

Word representation

単語の意味を**ベクトル**で表現

単語 → ベクトル



いろいろな方法

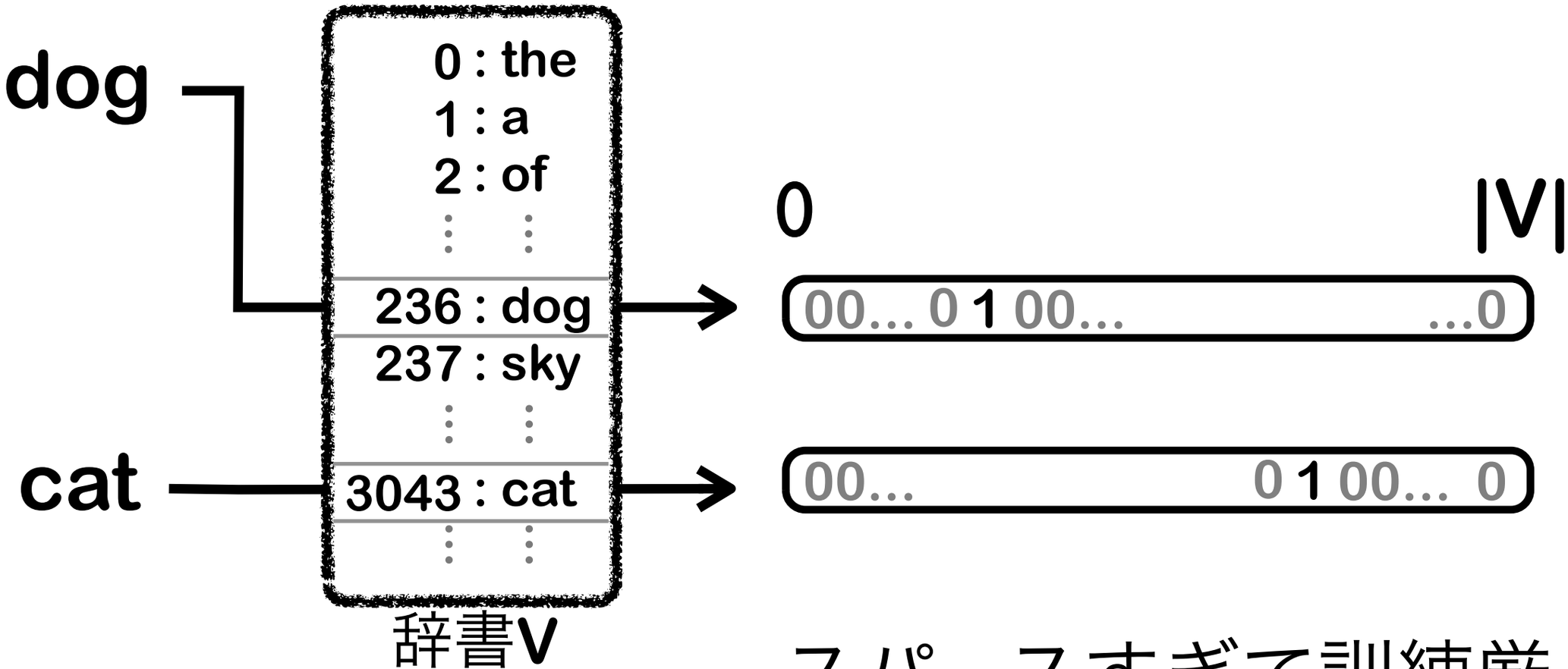
- One-hot
- Distributional
- Distributed

…

本題

One-hot representation

各単語に個別IDを割り当て表現

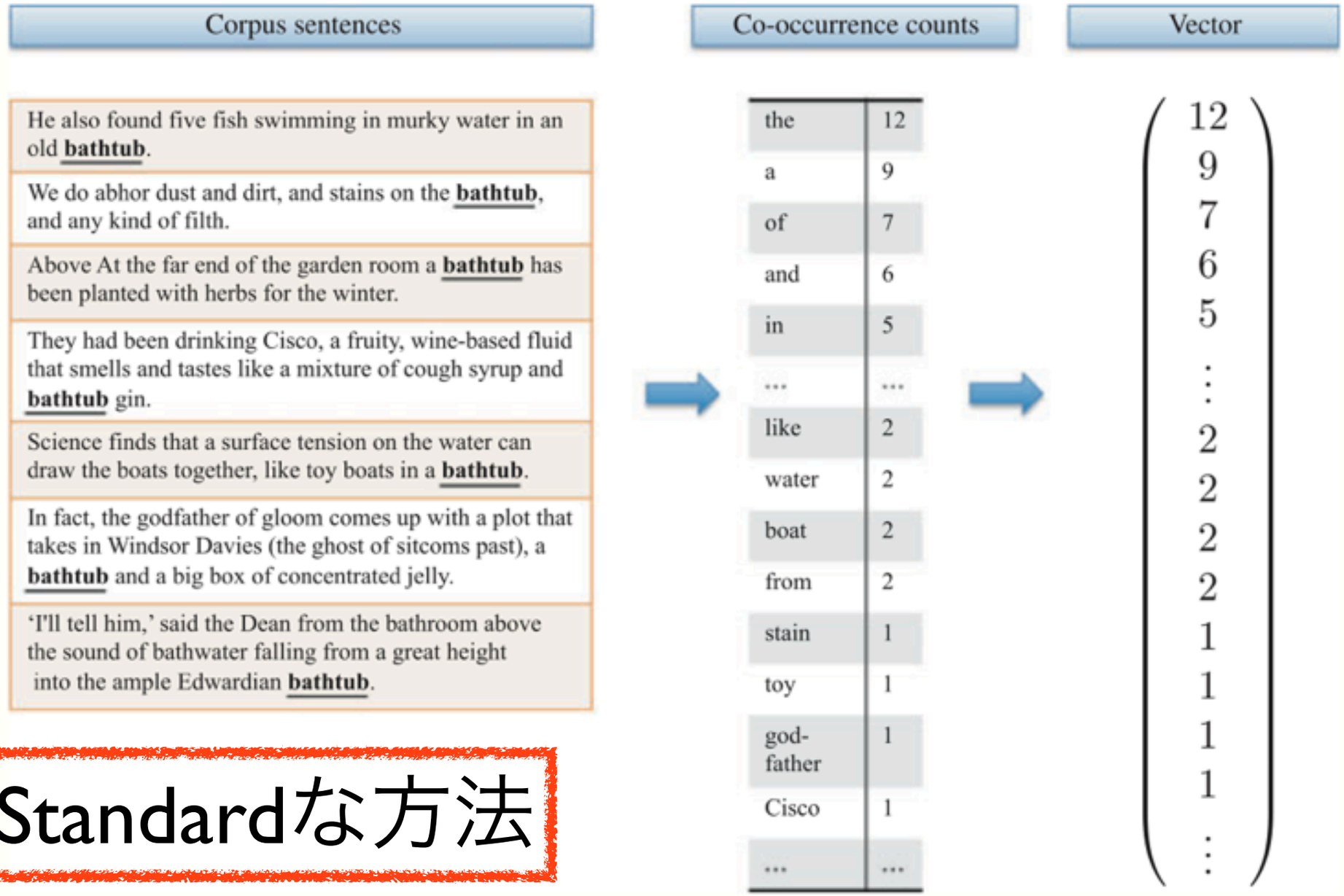


スパースすぎて訓練厳しい

汎化能力なくて未知語扱えず

Distributional representation

単語の意味は，周りの文脈によって決まる

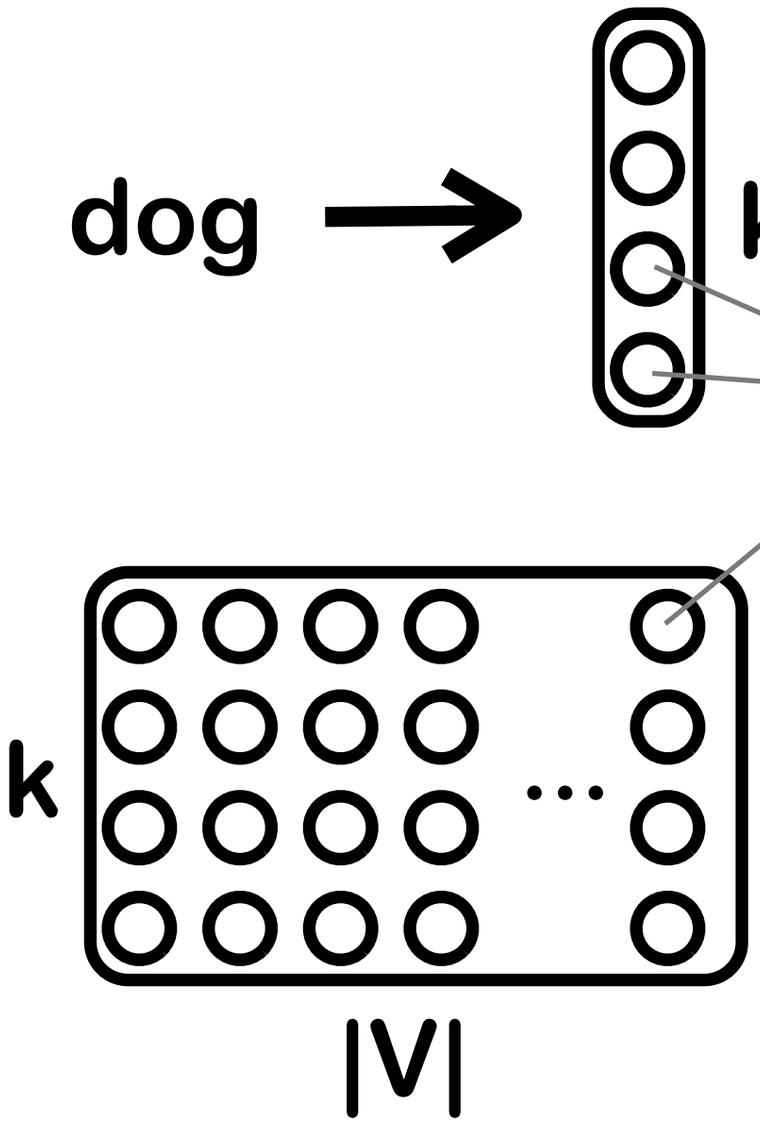


Standardな方法

Distributed representation

dense, low-dimensional, real-valued

= Word embedding



構文的，意味的な情報

を埋め込む

Neural Language Model
により学習

Distributed Word representation

Distributed Phrase representation

recursive勢の一強?

Distributed Sentence representation

Distributed Document representation

さて...

Distributed Word Representation
の学習

Neural Language Model

Neural Language Model

言語モデルとは

P(“私の耳が昨日からじんじん痛む”) → うむ

P(“私を耳が高くに拡散して草地”) → はぁ？

与えられた文字列の
生成確率を出力するモデル

Neural Language Model

N-gram言語モデル

単語列の出現確率を **N-gram** ずつに分解して近似

$$P(I, saw, the, red, house)$$

$$\approx P(I|<s>)P(saw|I)P(the|saw)P(red|the)P(house|red)P(</s>|house)$$

次元の呪いを回避

Neural Language Model

N-gram言語モデルの課題

1. 実質的には長い文脈は活用できない
   - せいぜい**N=1,2**

2. “似ている単語”を扱えない

$$P(I|<s>)P(saw|I)P(the|saw)P(red|the)P(house|red)P(</s>|house)$$

↓

$$P(house|green)$$

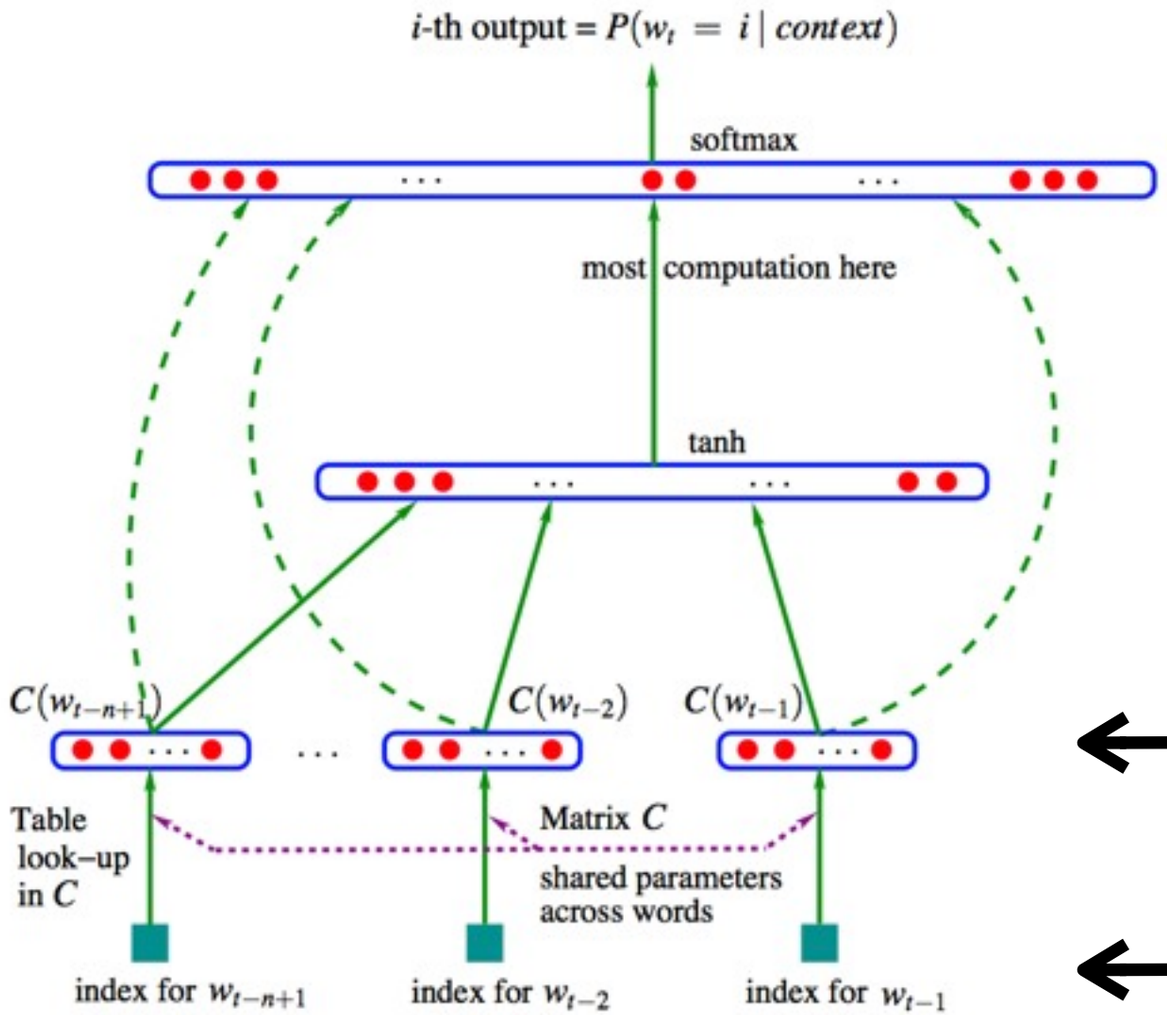
Neural Language Model とは

Neural Networkベースの言語モデル

- 言語モデルの学習
- **Word Embeddings**の学習

同時に学習する

Neural Language Model



← |辞書|次元の確率分布

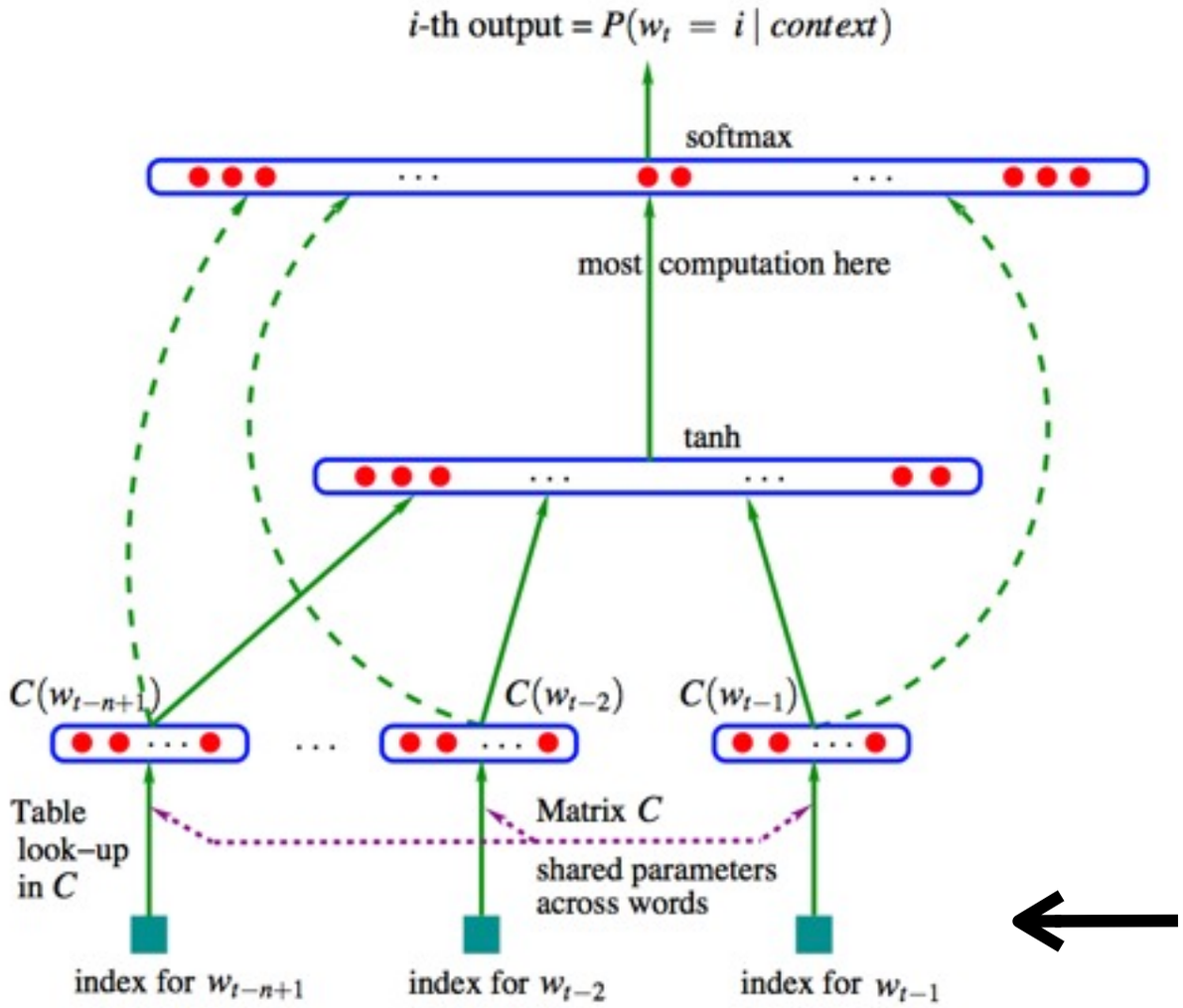
どの単語が次に出てくるかを予測

←― その単語のembedding

←― 単語そのもの

A Neural Probabilistic Language Model (bengio+, 2003)

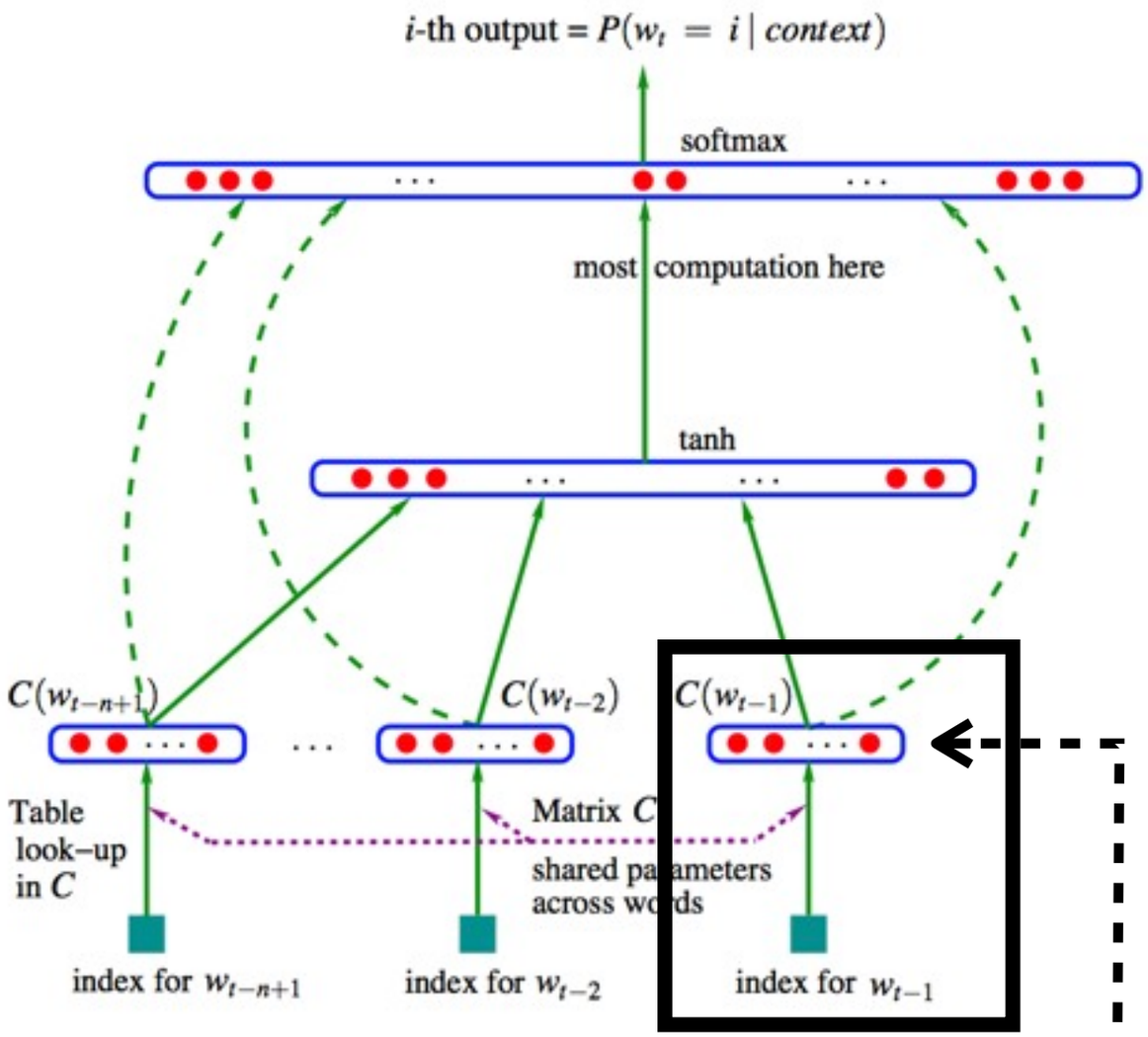
Neural Language Model



← 次にどの単語がどのくらいの確率でくるか

← n語の文脈が与えられた時

Neural Language Model



The cat is walking in the bedroom
A dog was running in a room
The cat is running in a room
A dog is walking in a bedroom
The dog was walking in the room
...

似ている単語に似た**embedding**を与えられれば,

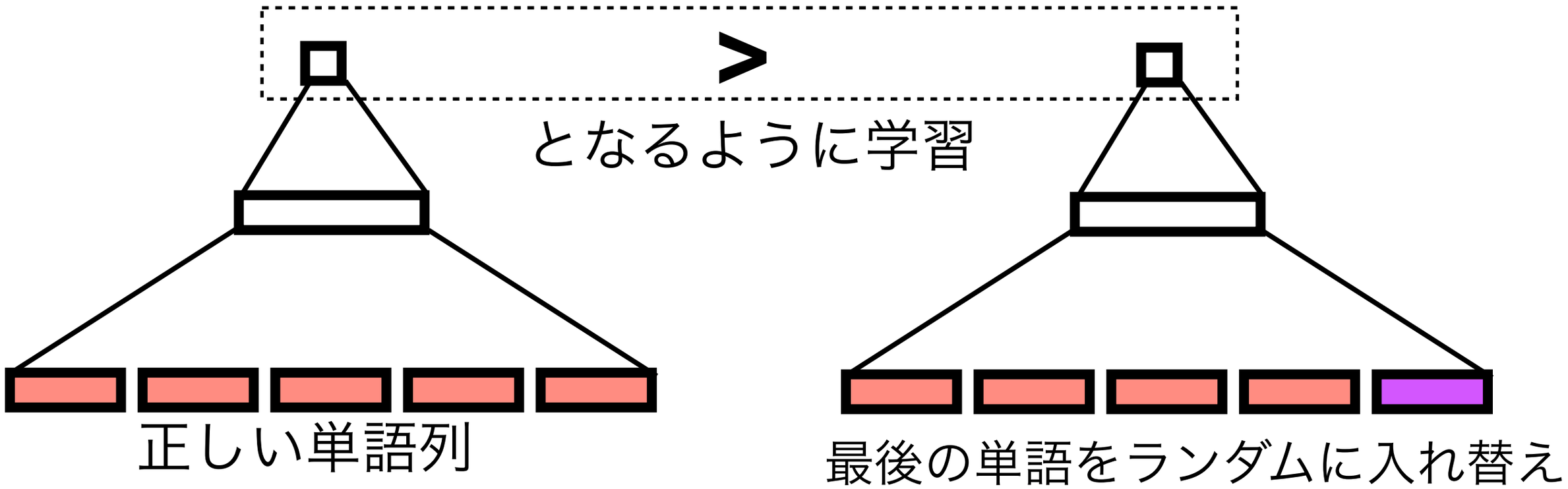
NN的には似た出力を出すはず

→ 語の類似度を考慮した言語モデルができる

他の主なアプローチ

仮名
Ranking language model[Collobert & Weston,2008]

単語列に対しスコアを出す**NN**



となるように学習

正しい単語列

最後の単語をランダムに入れ替え

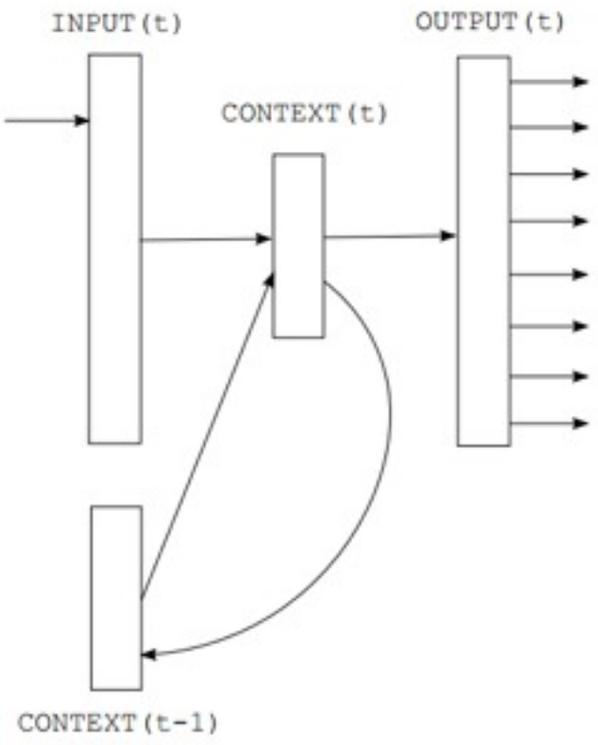
$$\theta \mapsto \sum_{x \in \mathcal{X}} \sum_{w \in \mathcal{D}} \max\left\{0, 1 - f_\theta(x) + f_\theta(x^{(w)})\right\}$$

他の主なアプローチ

word2vecの人

Recurrent Neural Network [Mikolov+, 2010]

1単語ずつ入力



出力は同じく
語彙上の確率分布

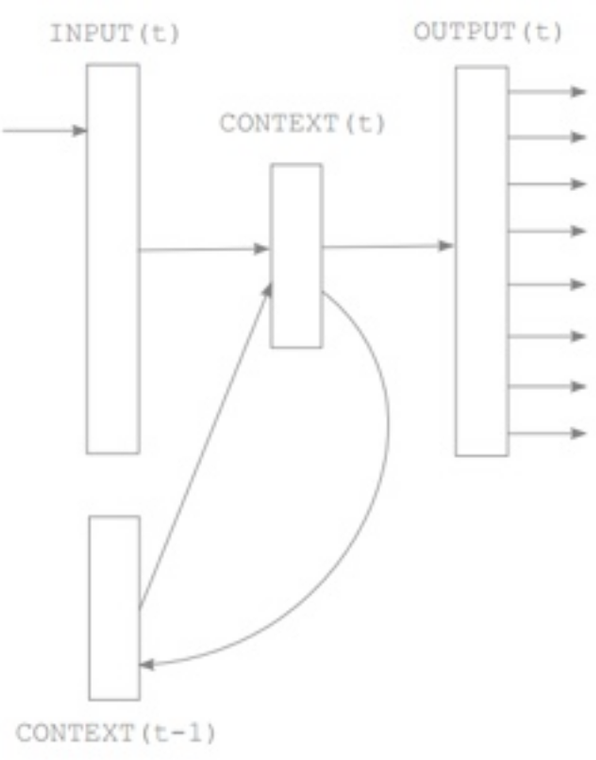
t番目の単語の入力時に
同時にt-1番目の内部状態を文脈として入力

他の主なアプローチ

word2vecの人

Recurrent Neural Network [Mikolov+, 2010]

1単語ずつ入力



出力は同じく
語彙上の確率分布

t番目の単語の入力時に
同時にt-1番目の内部状態を文脈として入力

word2vec

https://code.google.com/p/word2vec/

Taku Kudo
一般公開で共有しました - 2013/08/29

ＡＢＣ→Ｘ (Ａ→Ｂの関係に対し、 Ｃ→Ｘ に当てはまるXを探す)

グーグル ヤフー トヨタ → 日産
渋谷 新宿 札幌 → 旭川
警察 泥棒 正義 → くそ
平和 戦争 左 → 右
社員 会社 生徒 → 小学校
空 海 天井 → 床板
生きる 死ぬ 動く → 止まる
テレビ ラジオ 大阪 → 京都
車 車輪 人間 → 海馬
買う 売る 行く → 帰る
知る 忘れる 借りる → 貸す
夏 秋 冬 → 春
夏 中元 冬 → 歳暮
ニコン キヤノン ソニー → 東芝
扇風機 クーラー 空気 → 冷気
広島 牡蠣 北海道 → 昆布

他に...

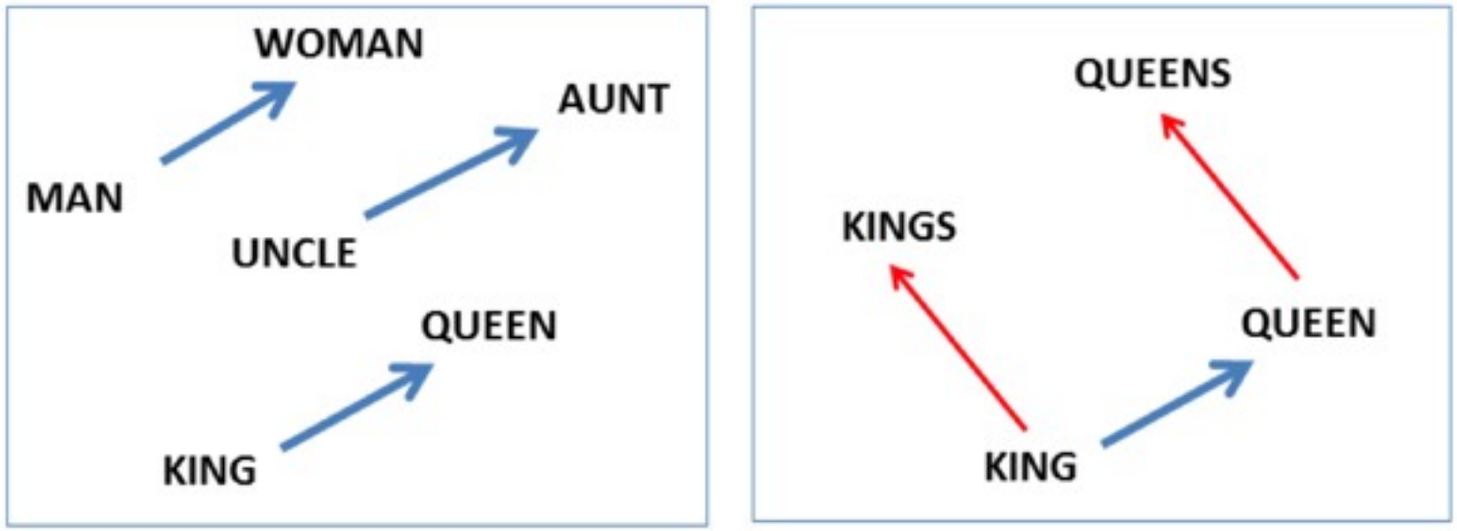
研究 進展 人生 → 苦悩

人生 恋愛 研究 → 進展

word2vec

https://code.google.com/p/word2vec/

単語間の関係の**offset**を捉えている仮定



king - man + woman ≒ queen

単語の**意味**についてのしっかりした分析

Keywords

任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

- convolutional-way
- recursive-way

Distributed word representation

phrase, sentence-level representation

Neural language model

先ほどは，単語表現を学習するためのモデル

(Bengio's, C&W's, Mikolov's)

以降は，NNで言語処理のタスクに

取り組むためのモデル

(結果的に単語ベクトルは学習されるが

おそらくタスク依存なものになっている)

Keywords

任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

- convolutional-way
- recursive-way

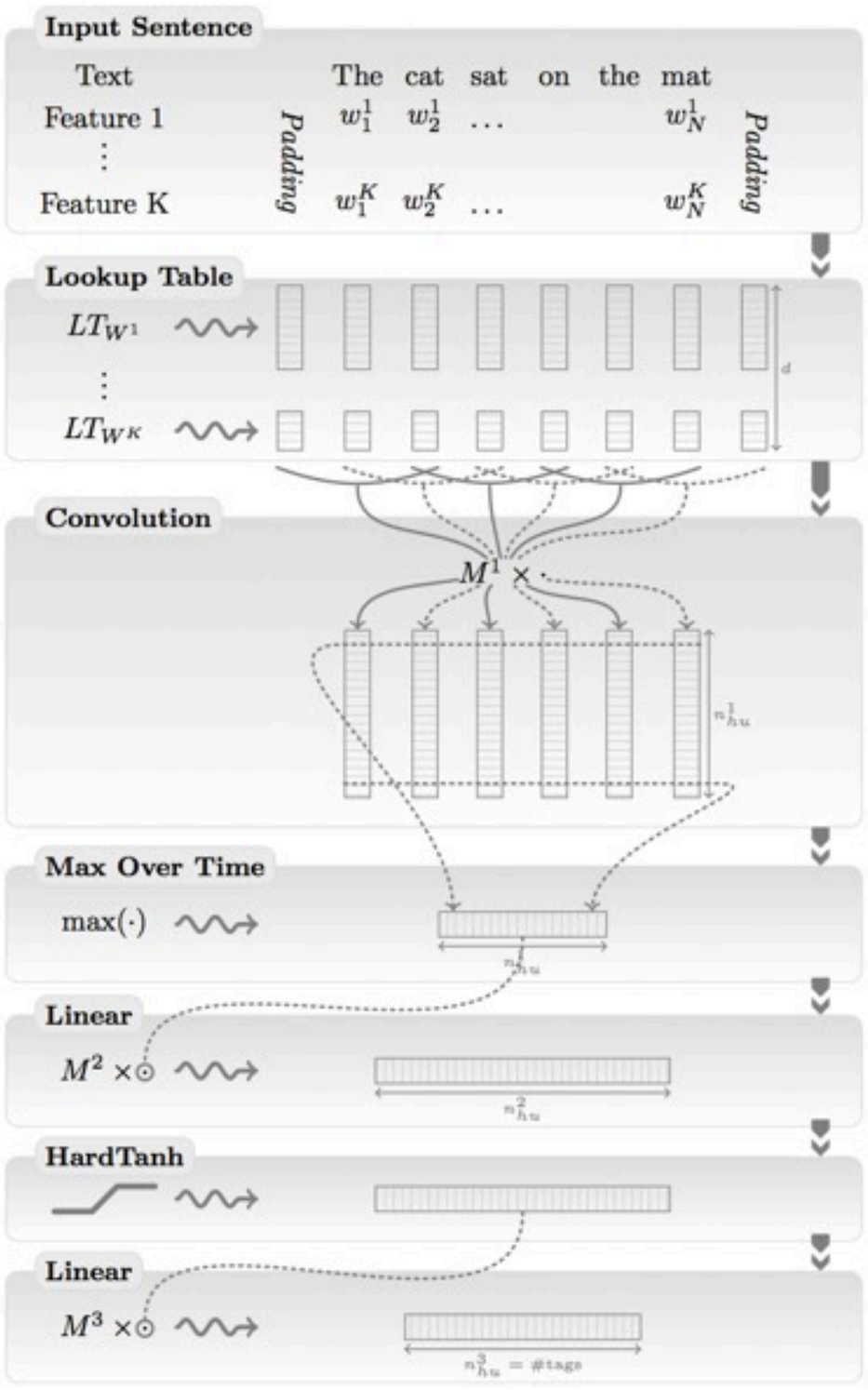
Distributed word representation

phrase, sentence-level representation

Neural language model

convolutional-way

Collobert & Weston[2008]



はじめに

2008年の論文

文レベルの話のとこだけ

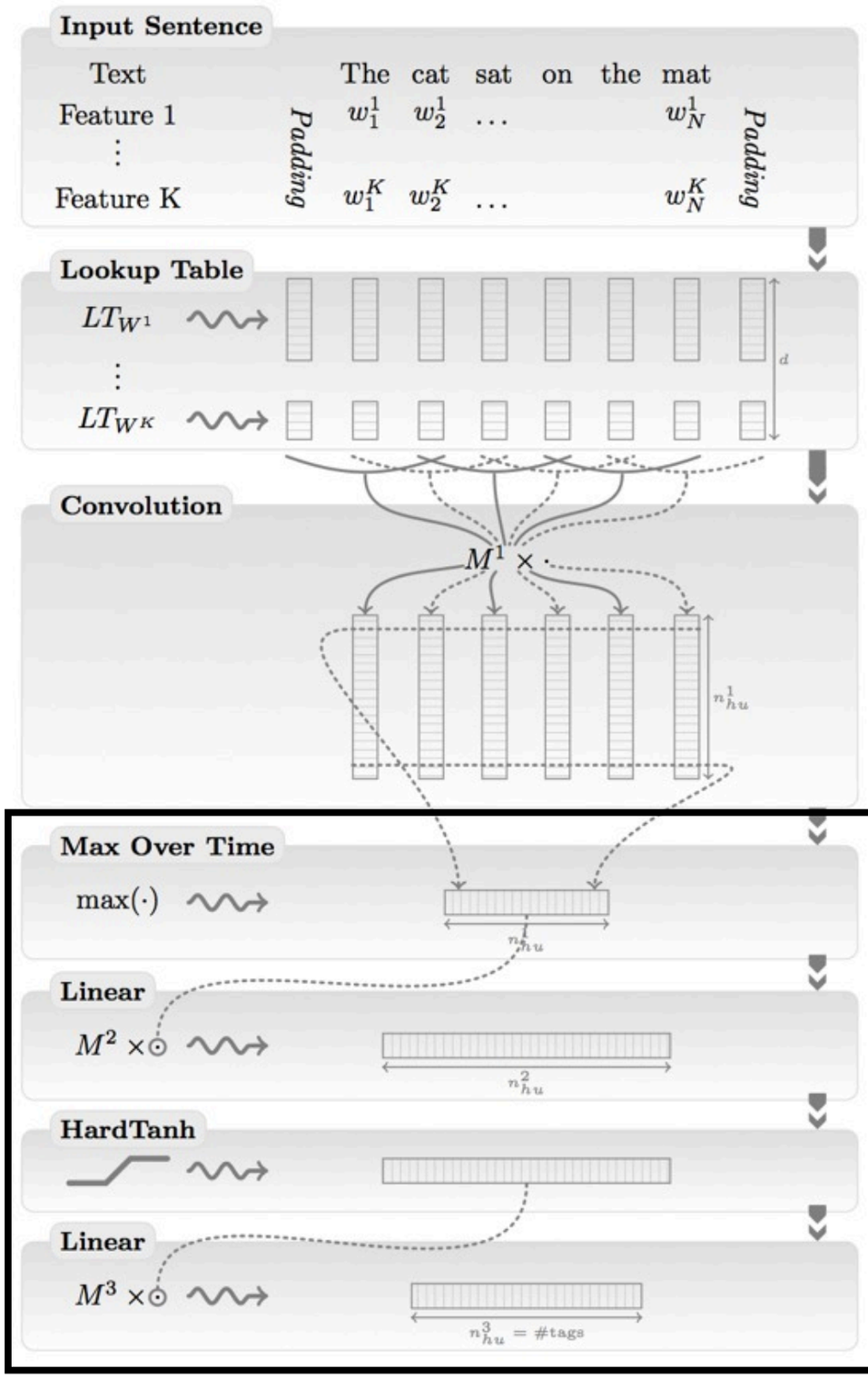
他に

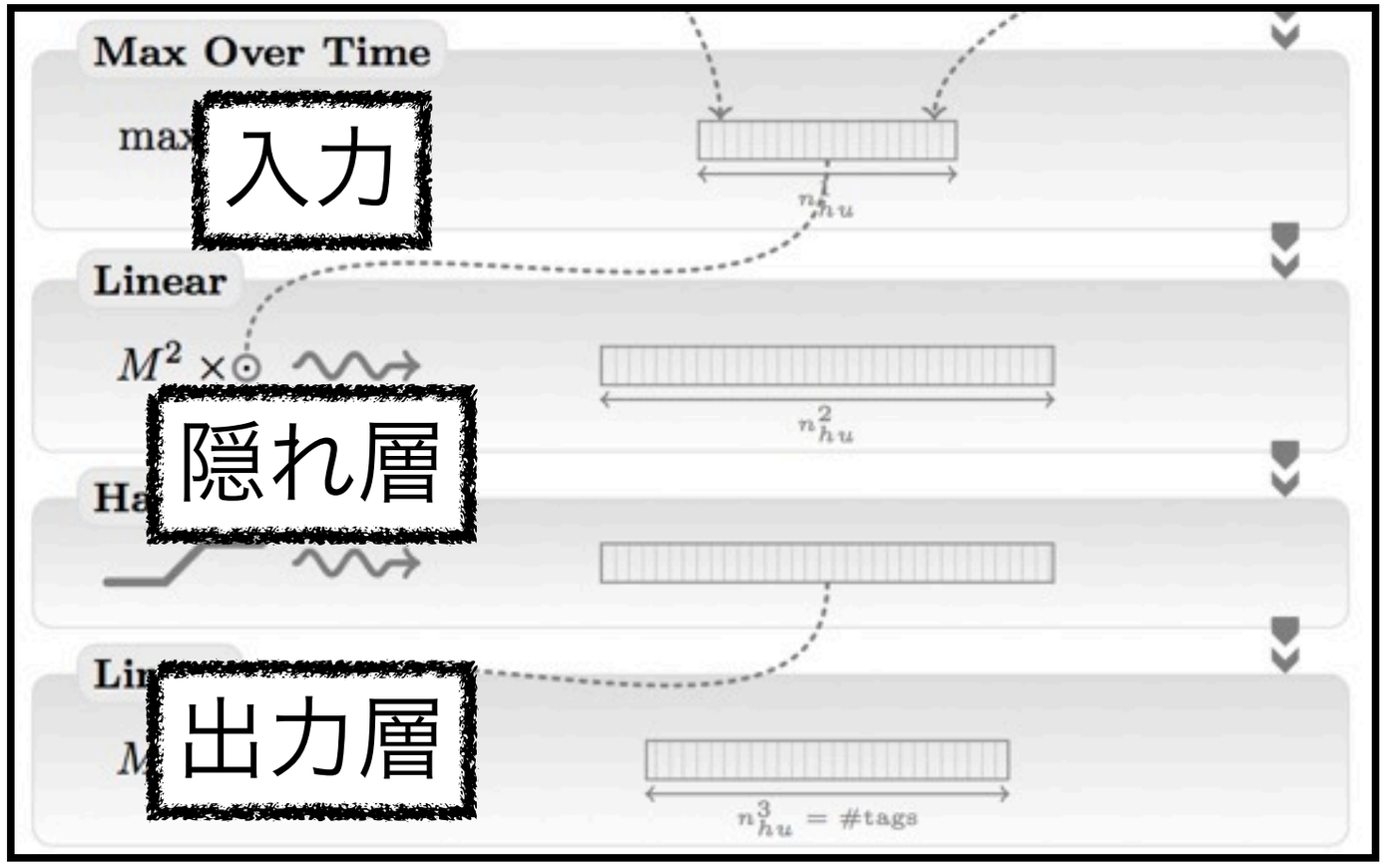
Multi-task learning
Language model
の話題がある

convolutional-way

Collobert & Weston[2008]

はじめに



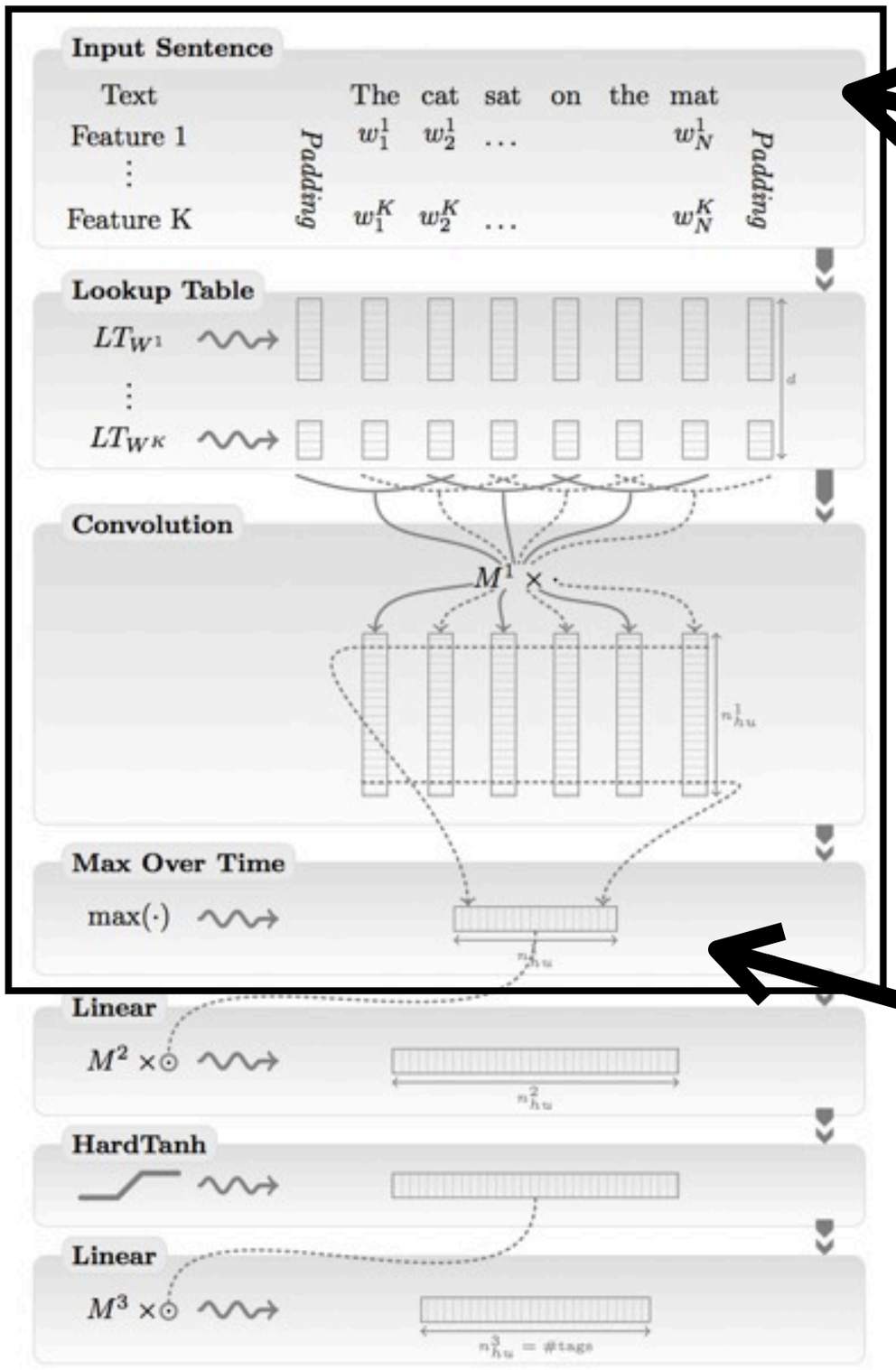


ここは

2層Neural Network

convolutional-way

Collobert & Weston[2008]



はじめに

任意の長さの文

Neural Networkに

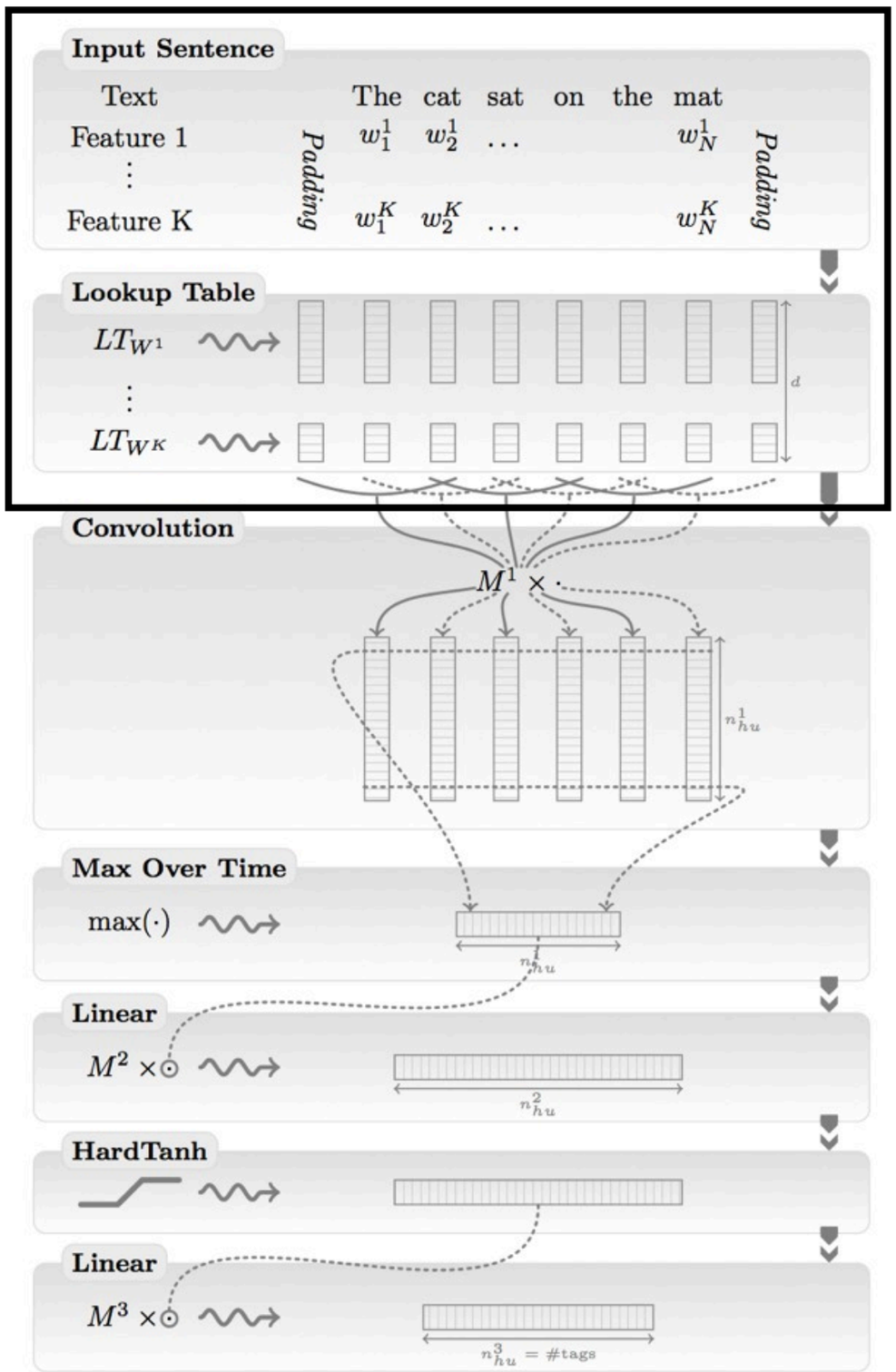
入力するために

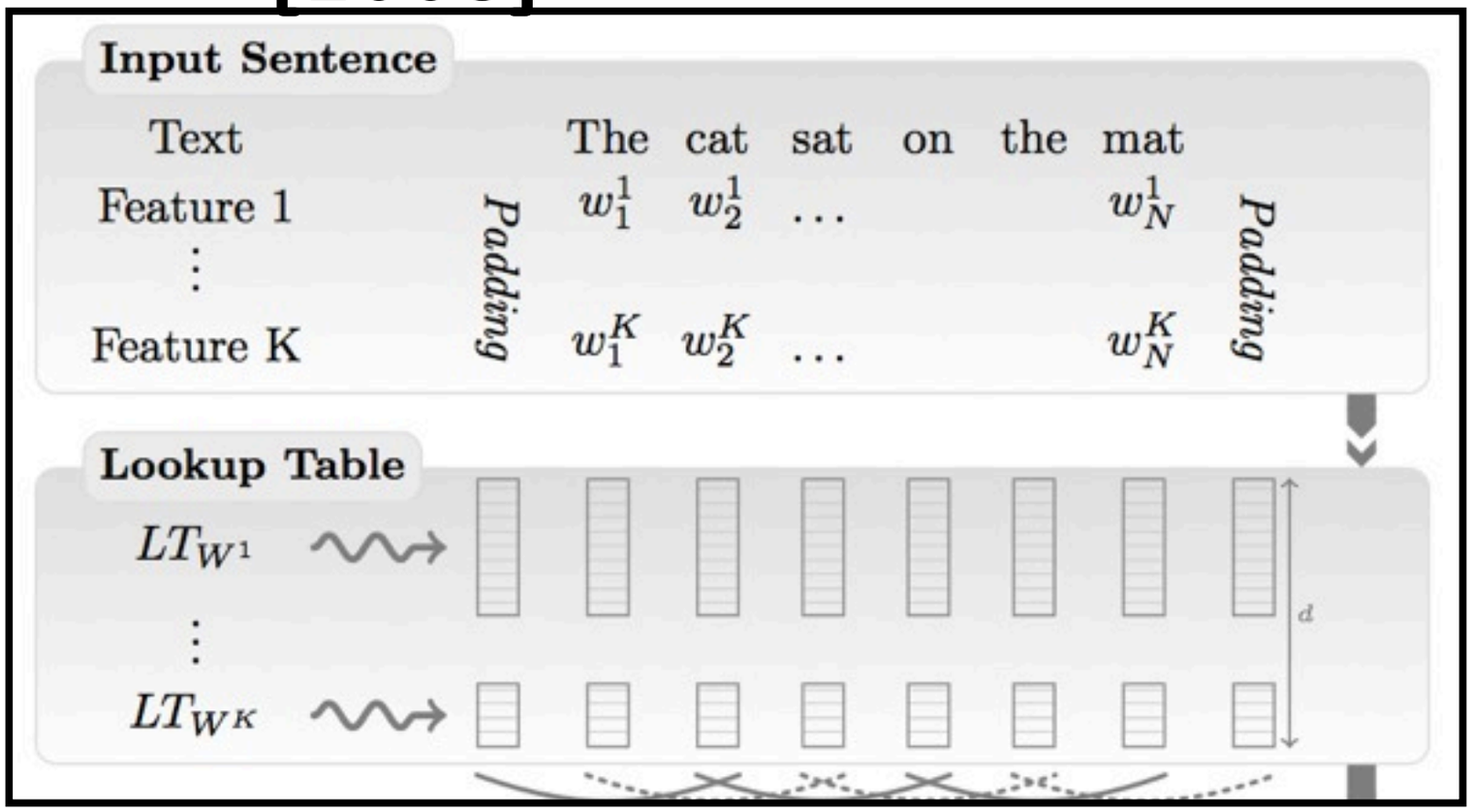
どうやって

固定次元に変換するか

convolutional-way

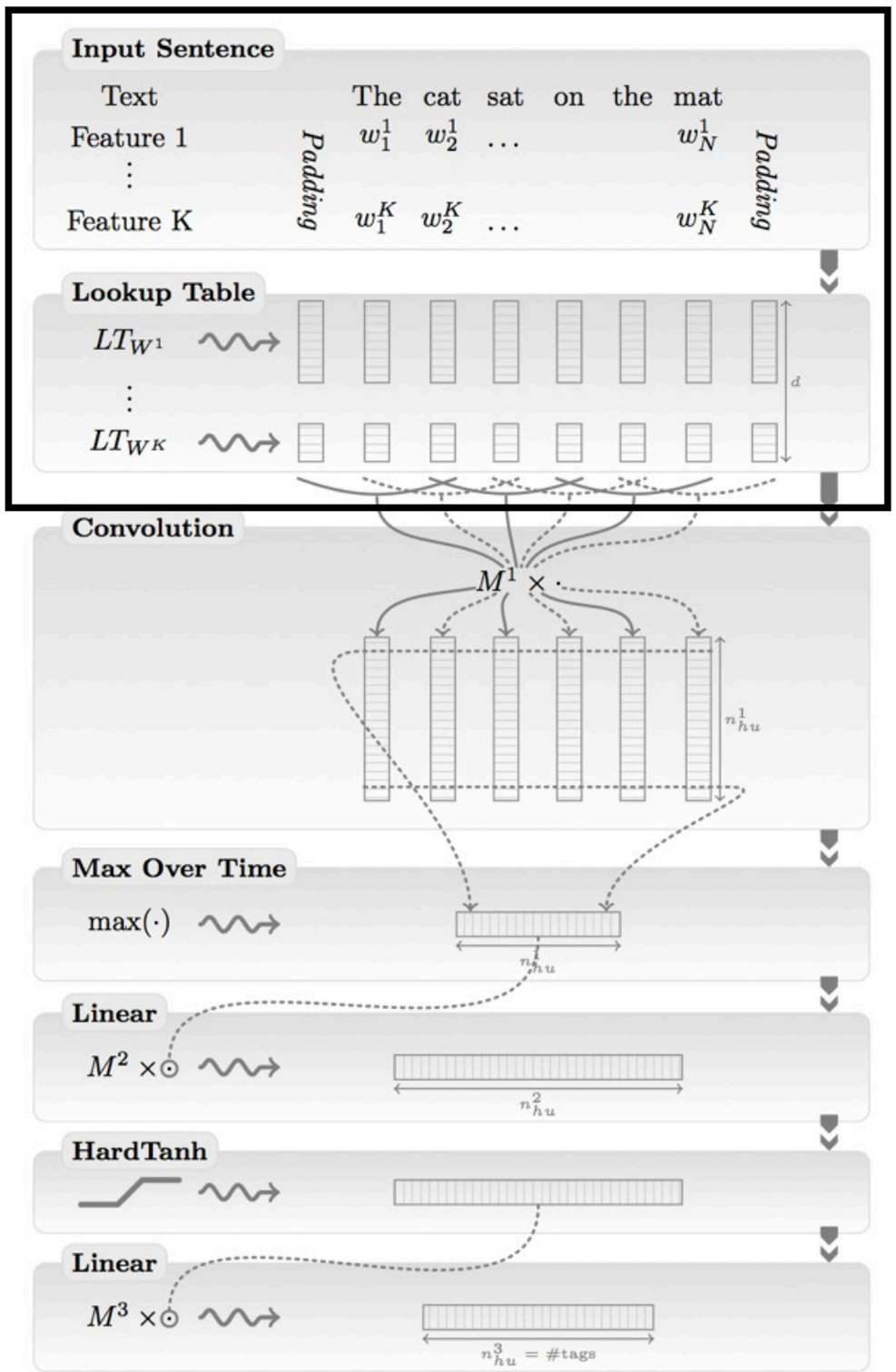
Collobert & Weston[2008]

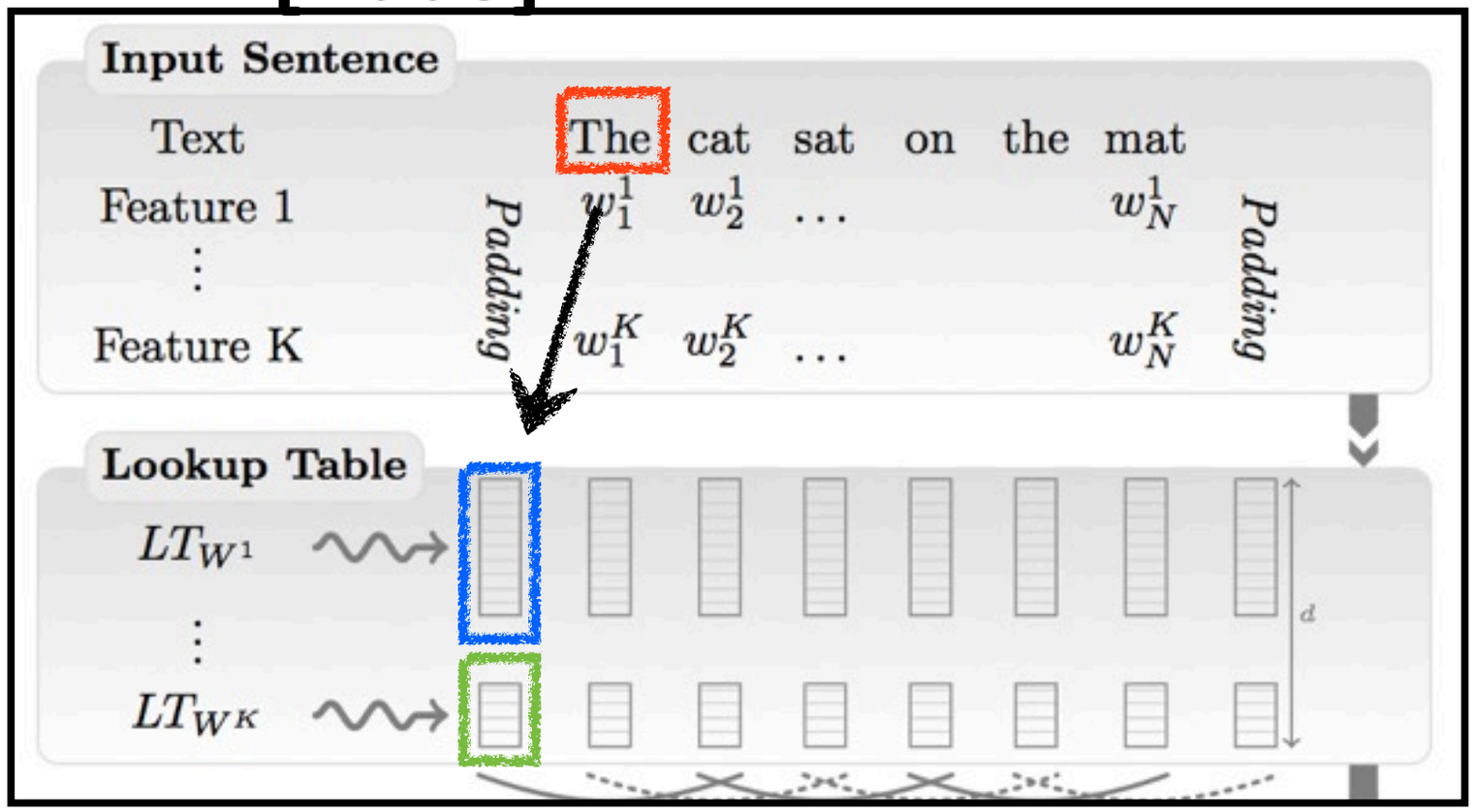




convolutional-way

Collobert & Weston[2008]



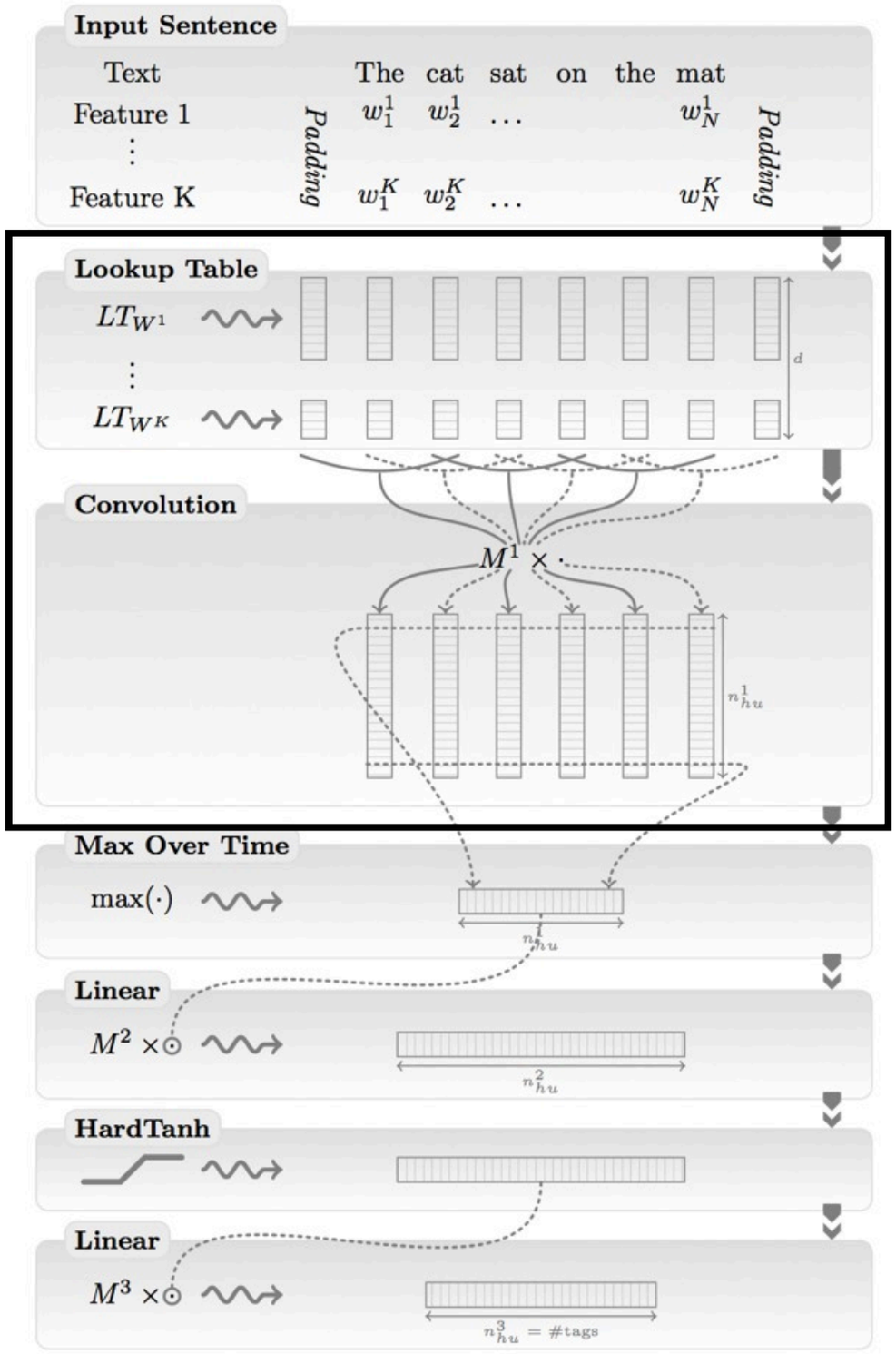


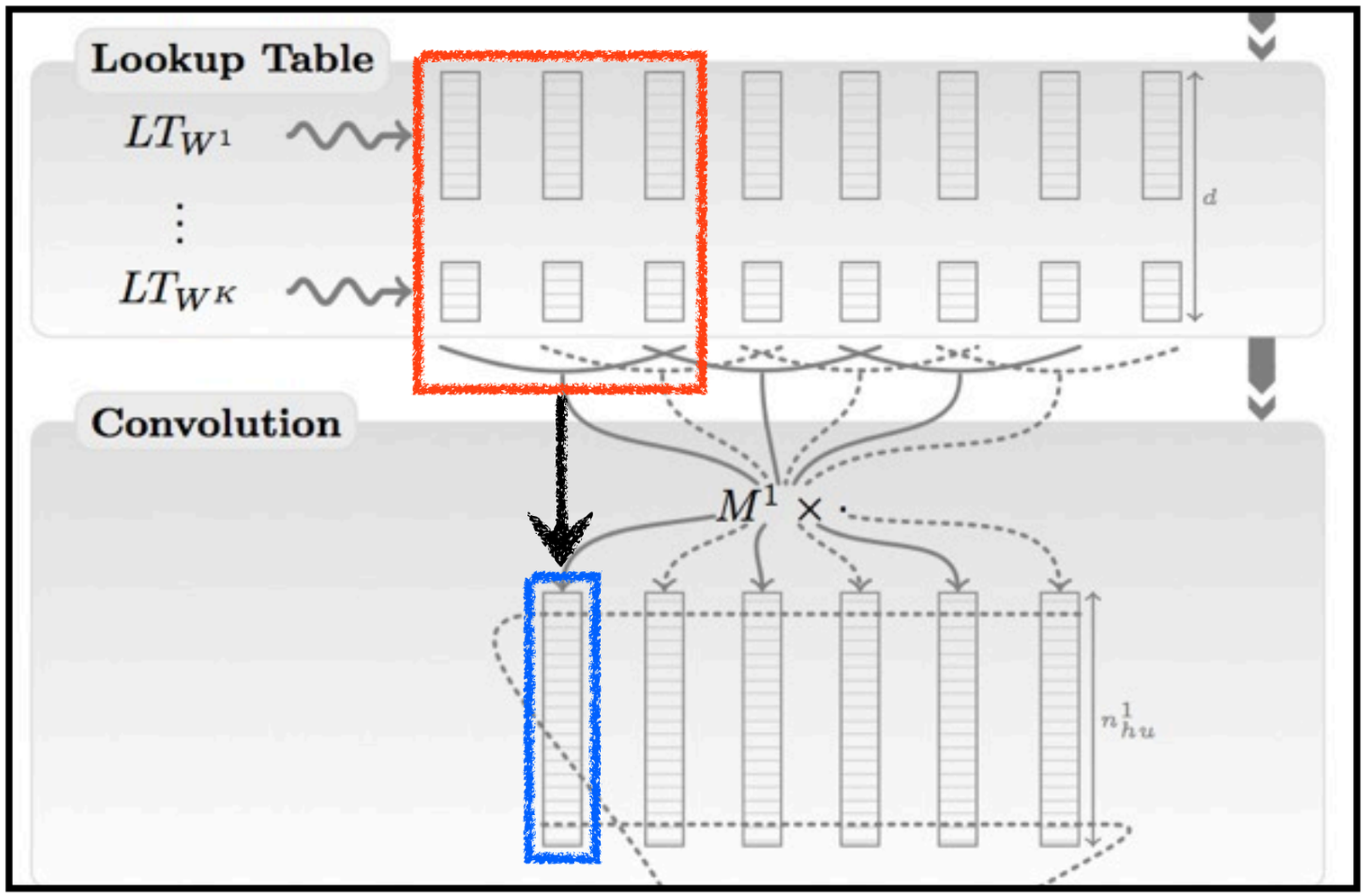
単語を**d**次元ベクトルに

(word embedding + α)

convolutional-way

Collobert & Weston[2008]



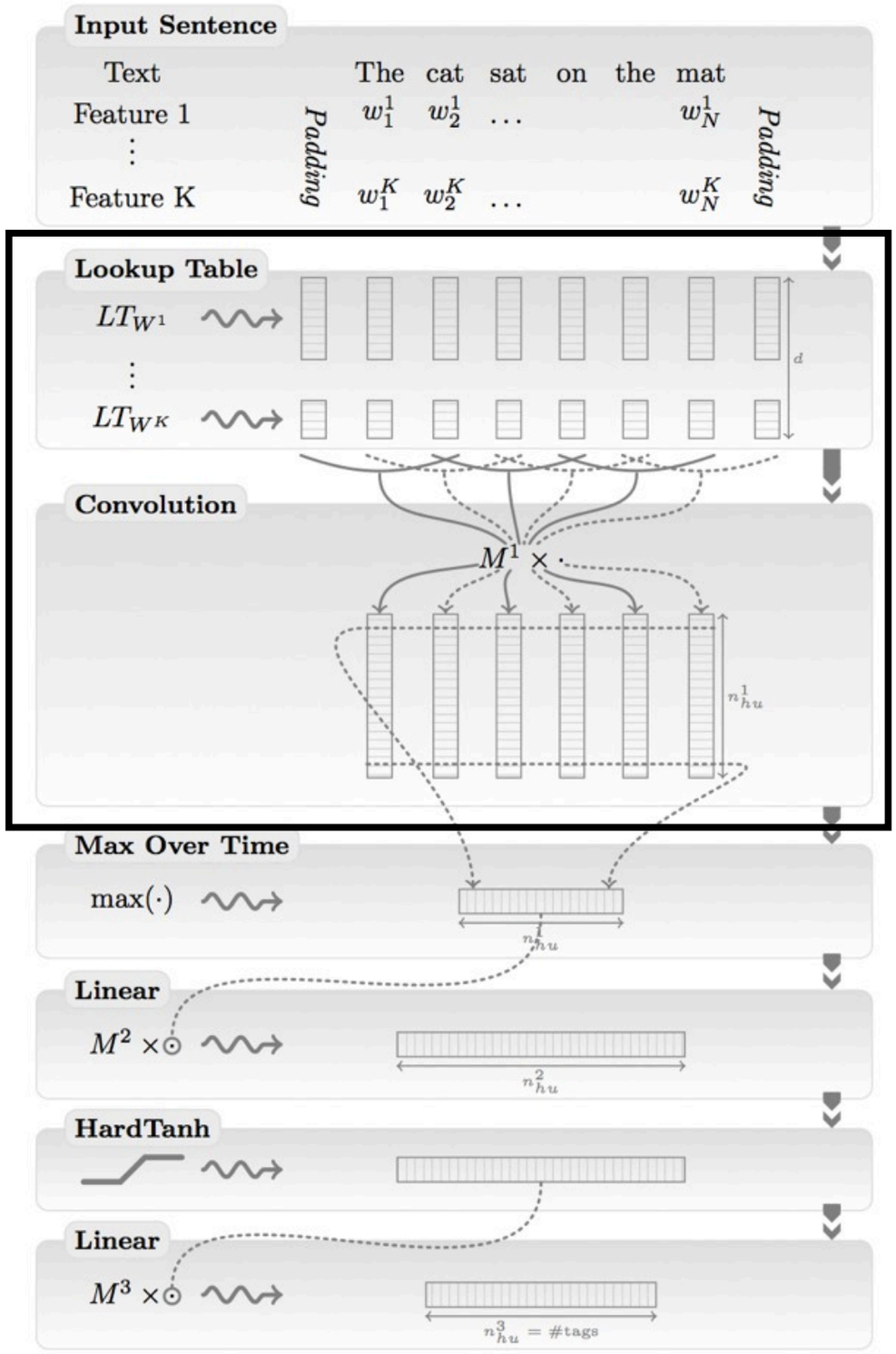


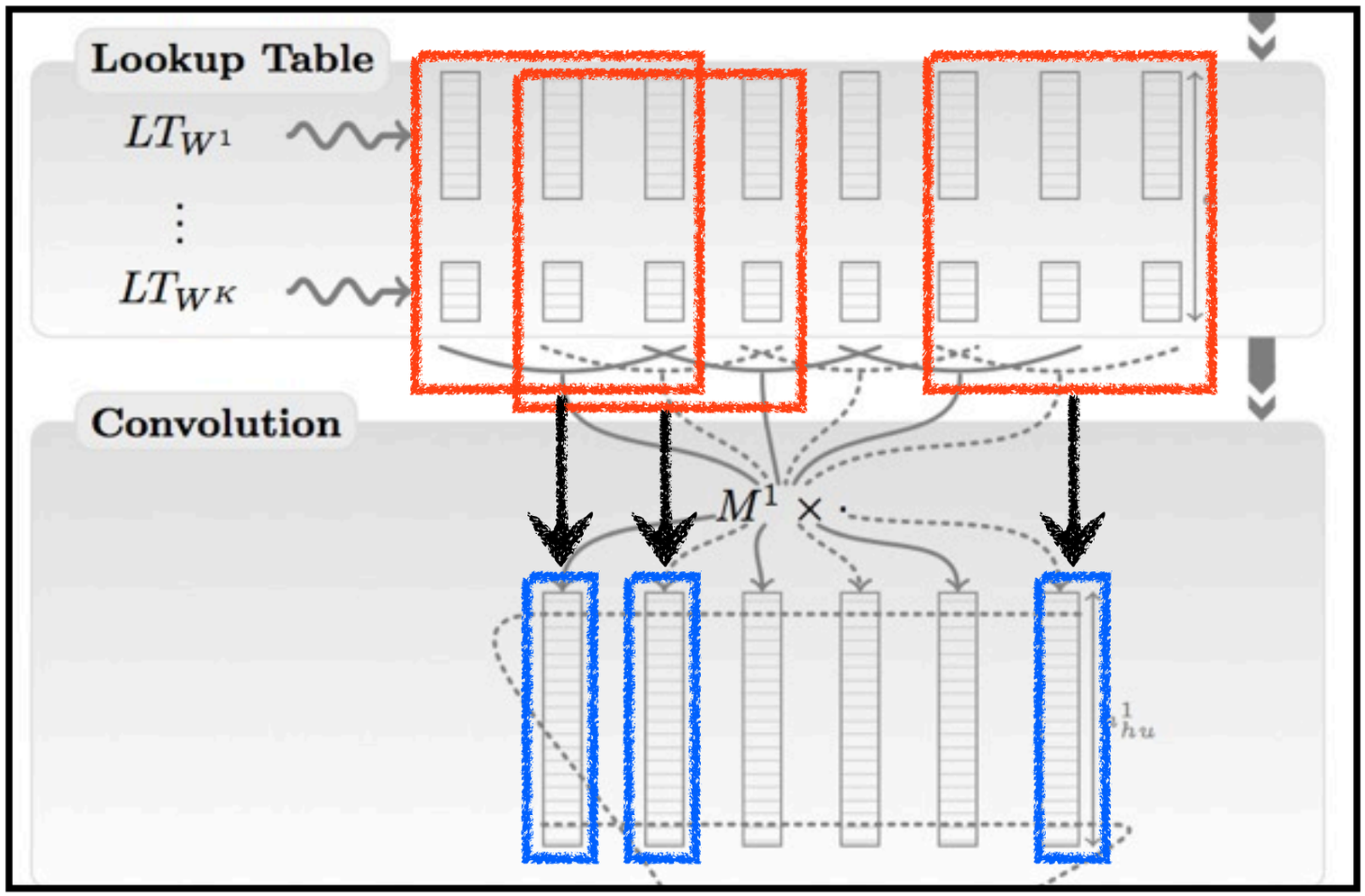
3単語をConvolutionして

localな特徴を得る

convolutional-way

Collobert & Weston[2008]

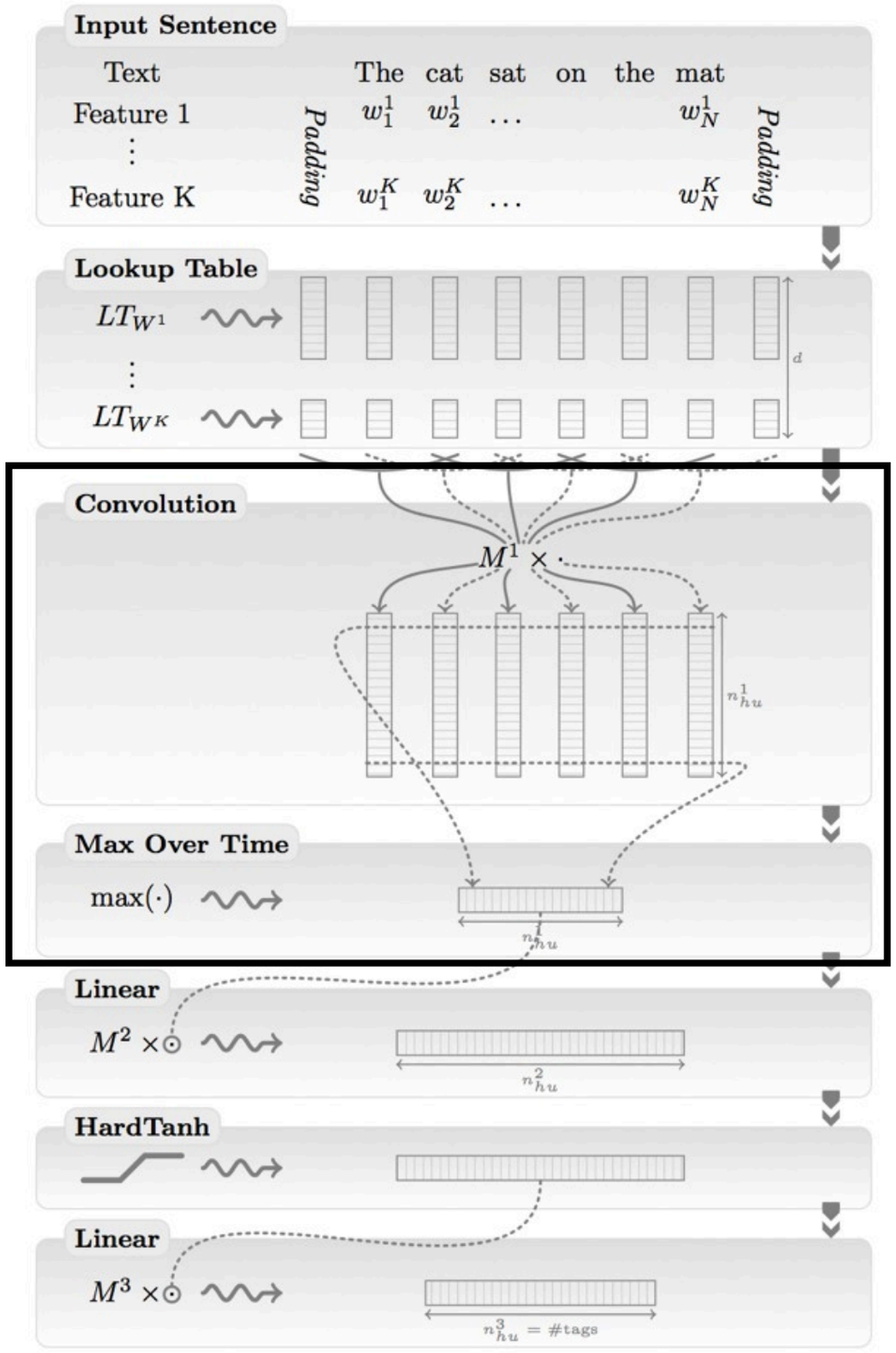


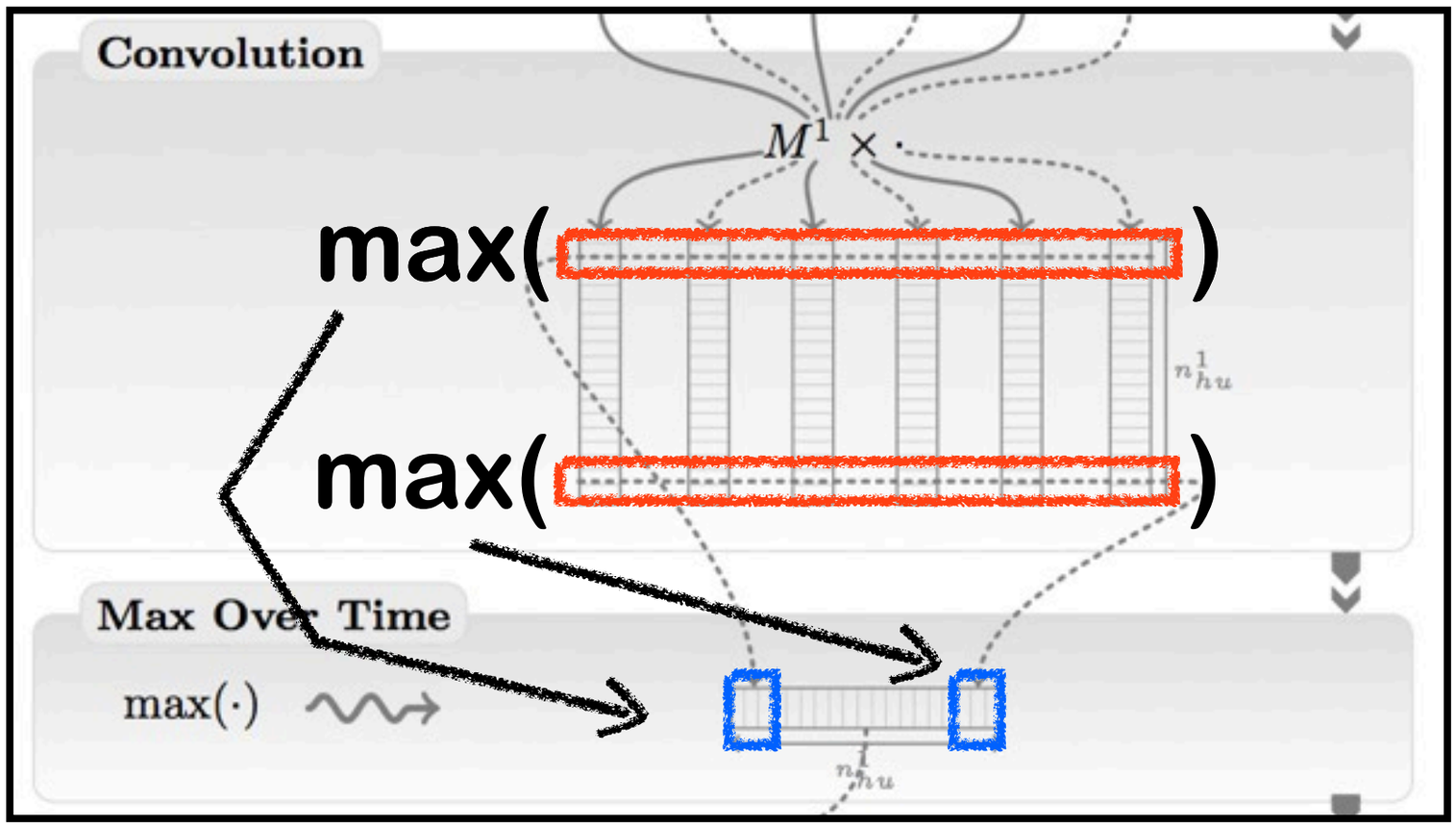


共通の重み(→)

convolutional-way

Collobert & Weston[2008]



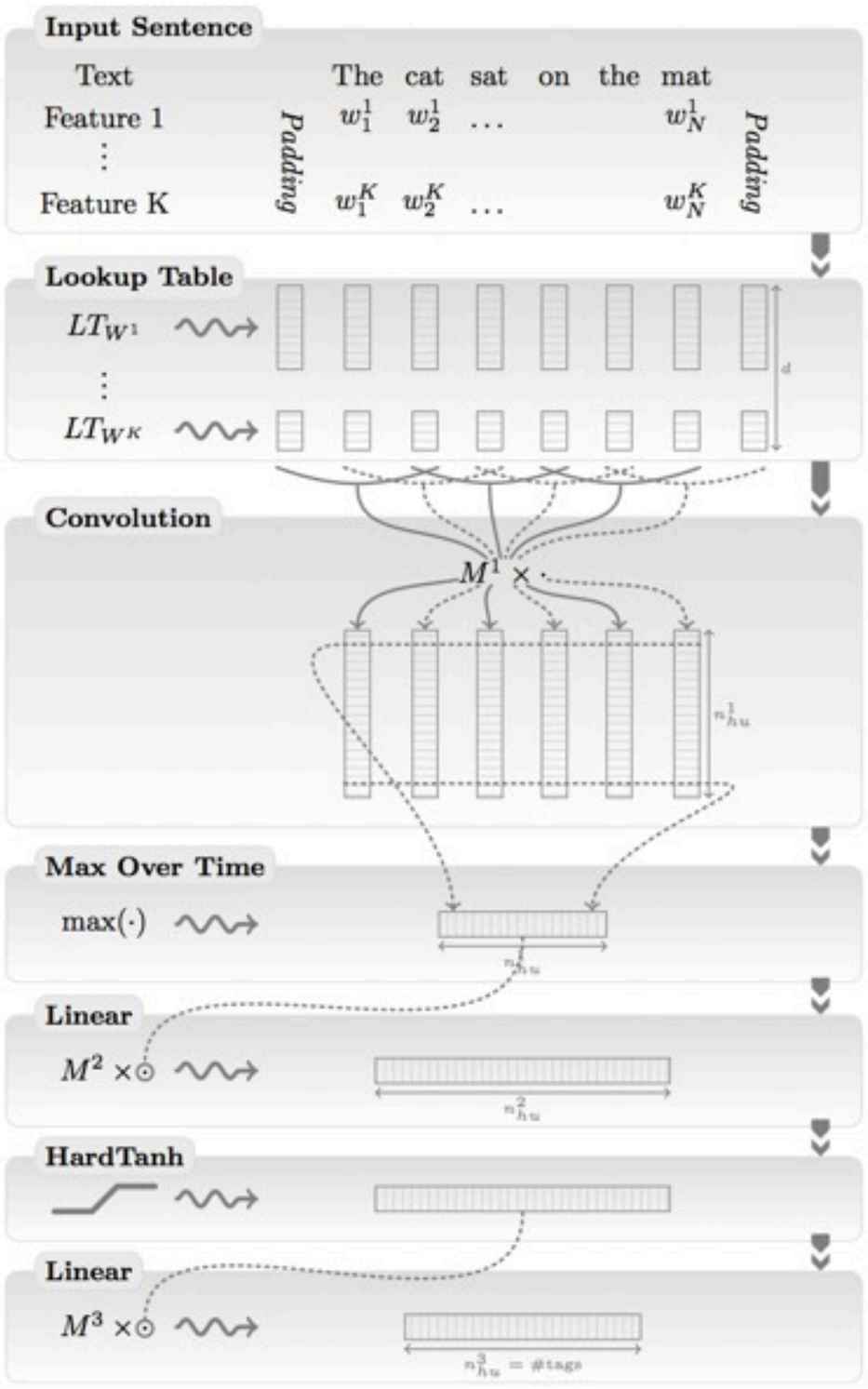


各次元の最大値を取得

次元固定のキモ

convolutional-way

Collobert & Weston[2008]



任意サイズの入力

固定次元の入力

Neural Networkで学習

Keywords

任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

- convolutional-way
- recursive-way

Distributed word representation

phrase, sentence-level representation

Neural language model

Keywords

任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

- convolutional-way
- recursive-way

↓

phrase, sentence-level representation

Distributed word representation

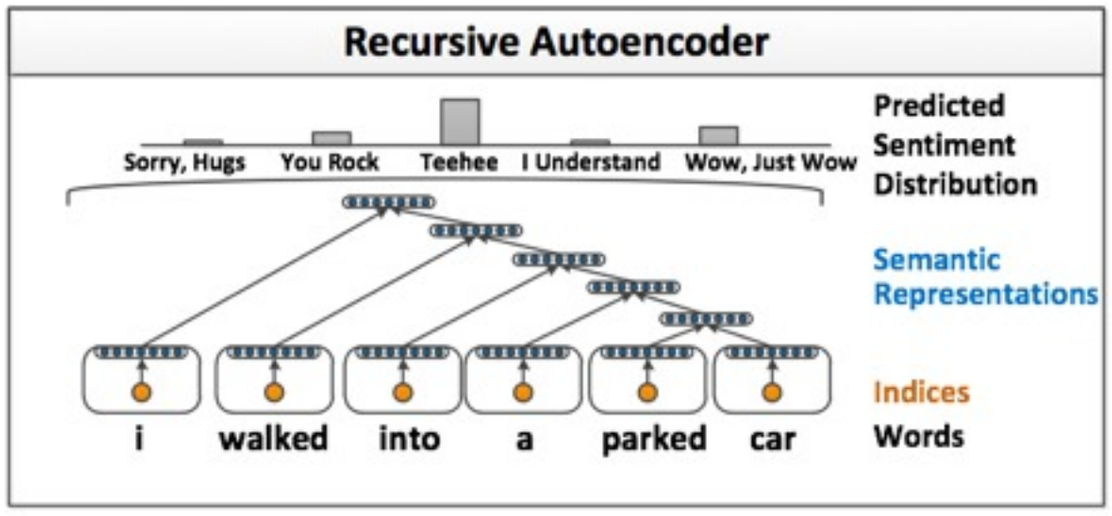
Neural language model

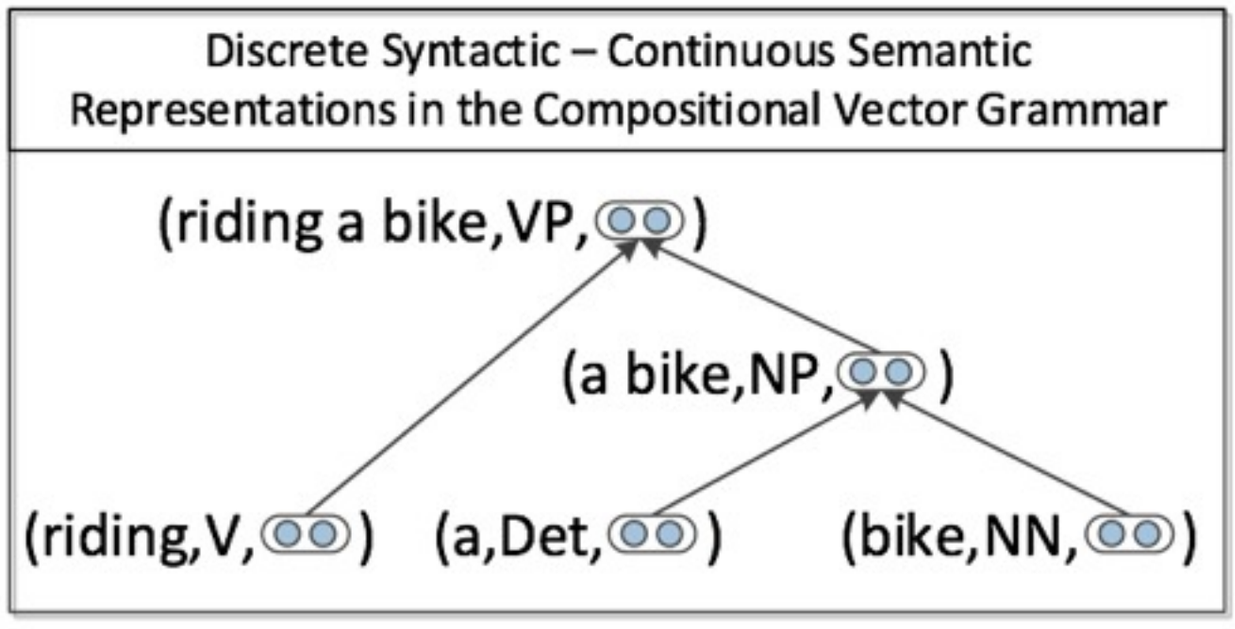
recursive-way

Richard Socher一派

Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions (EMNLP2011)

Recursive Autoencoder





Recursive Neural Network

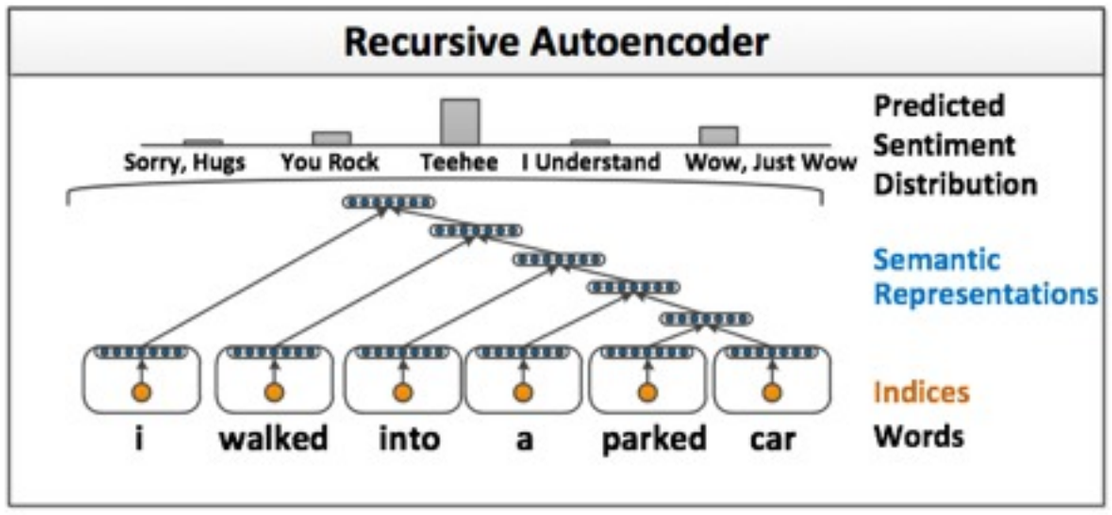
Parsing Natural Scenes and Natural Language with Recursive Neural Networks (ICML2011)

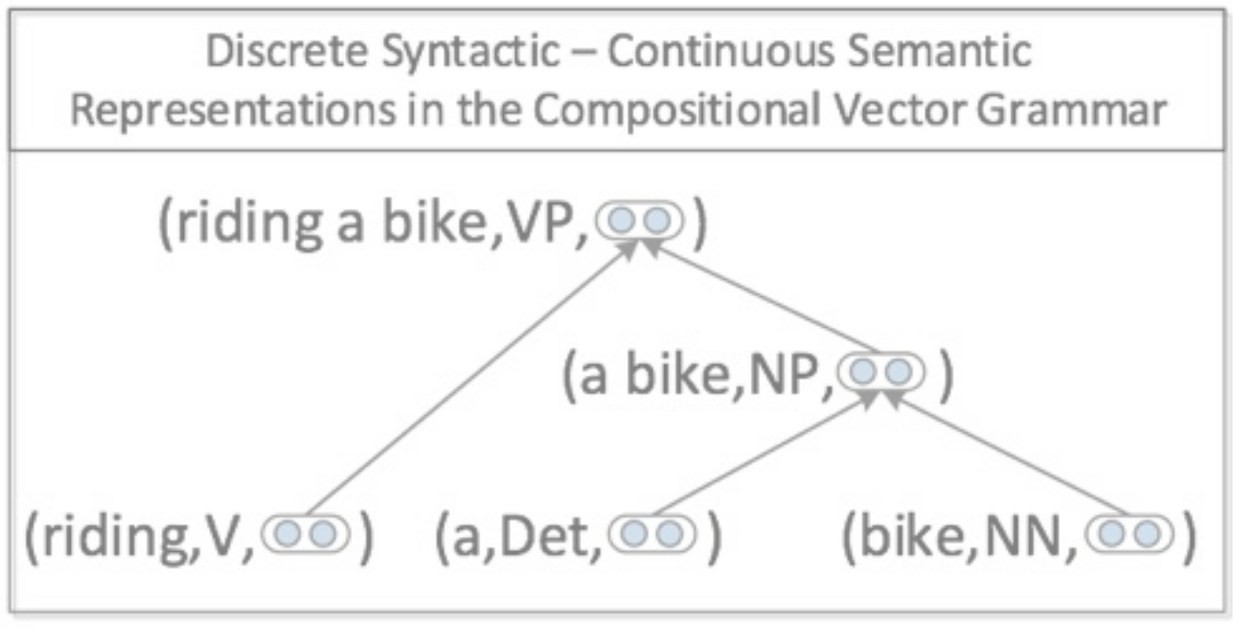
recursive-way

Richard Socher一派

Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions (EMNLP2011)

Recursive Autoencoder





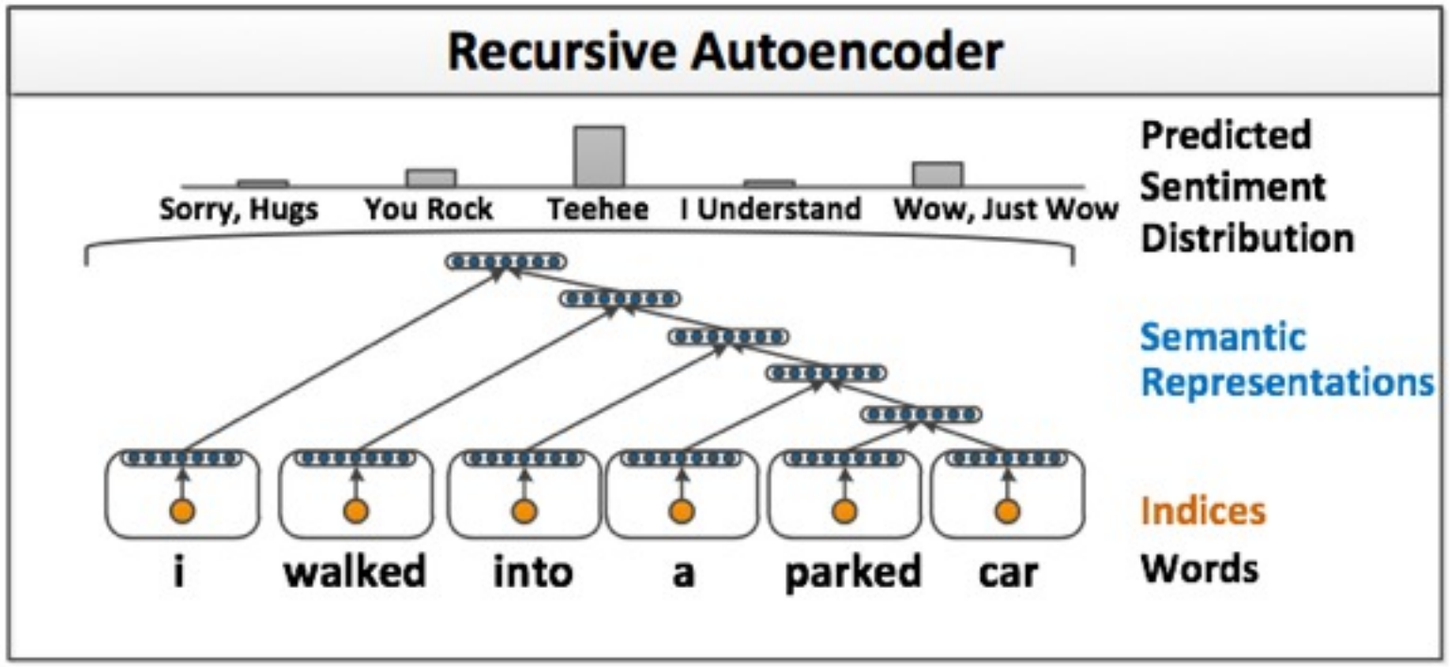
Recursive Neural Network

Parsing Natural Scenes and Natural Language with Recursive Neural Networks (ICML2011)

recursive-way

Recursive Autoencoder

Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions (EMNLP2011)

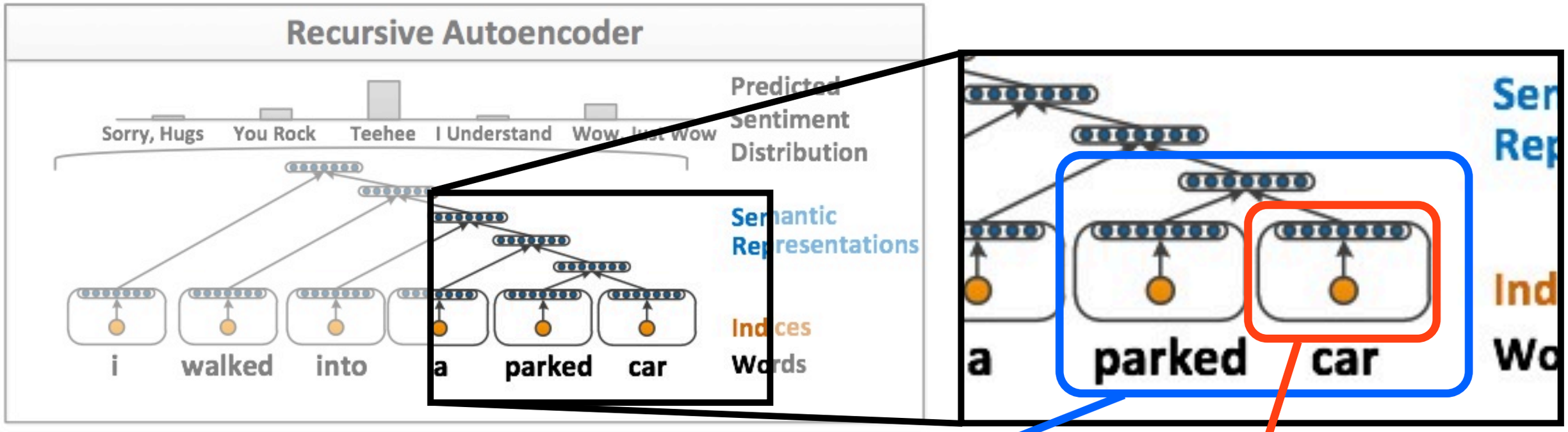


感情分布の推定

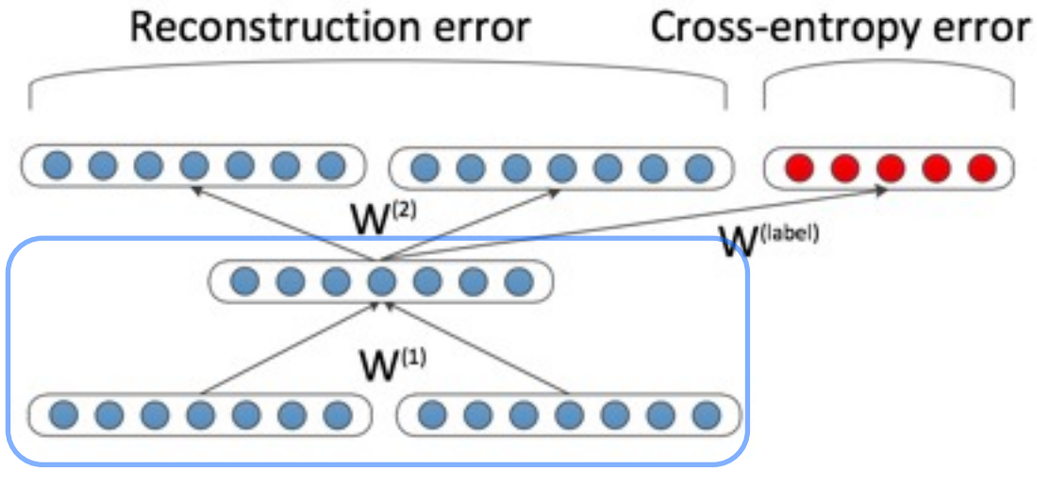
recursive-way

Recursive Autoencoder

Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions (EMNLP2011)



Autoencoder



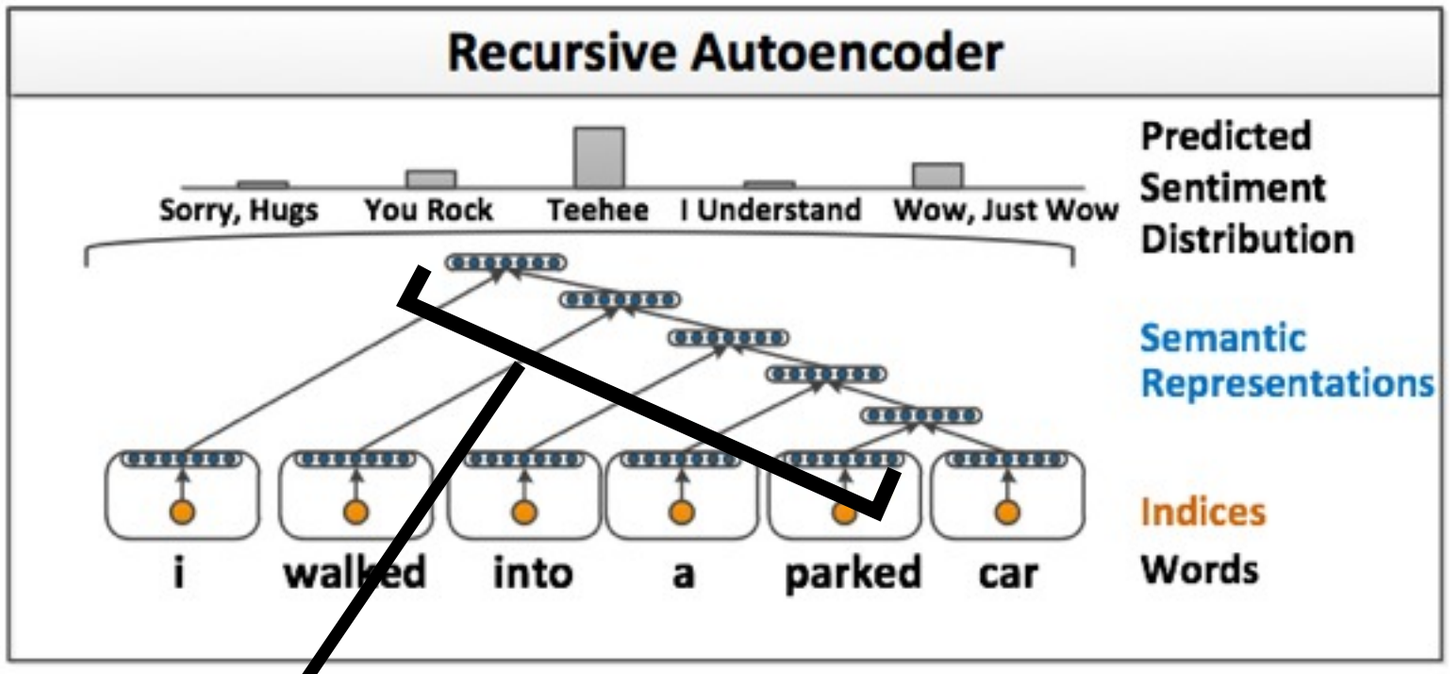
Distributed Representation

感情分布の推定

recursive-way

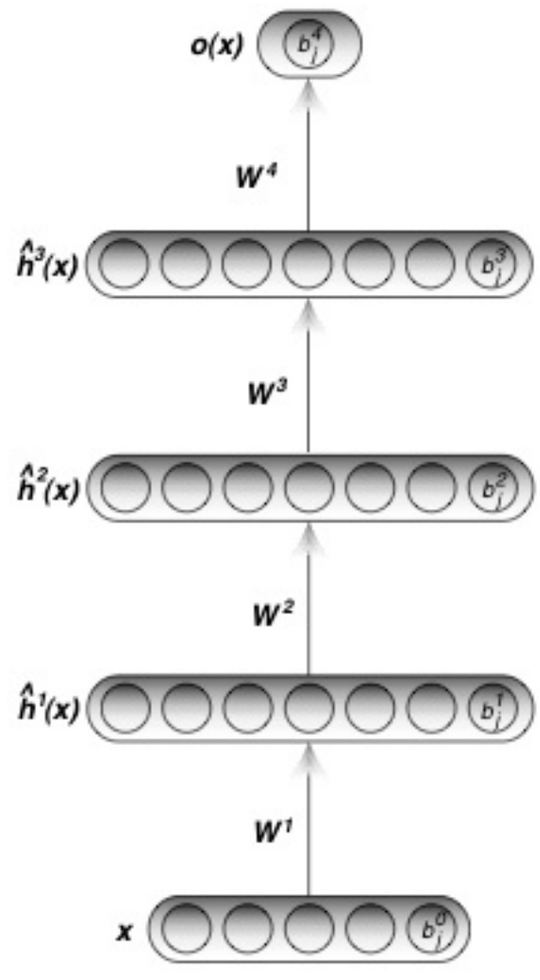
Recursive Autoencoder

Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions (EMNLP2011)



2つを1つに圧縮用の共通の

Autoencoderを再帰的に適用



Stacked Autoencoderは
階層毎に別のものを訓練

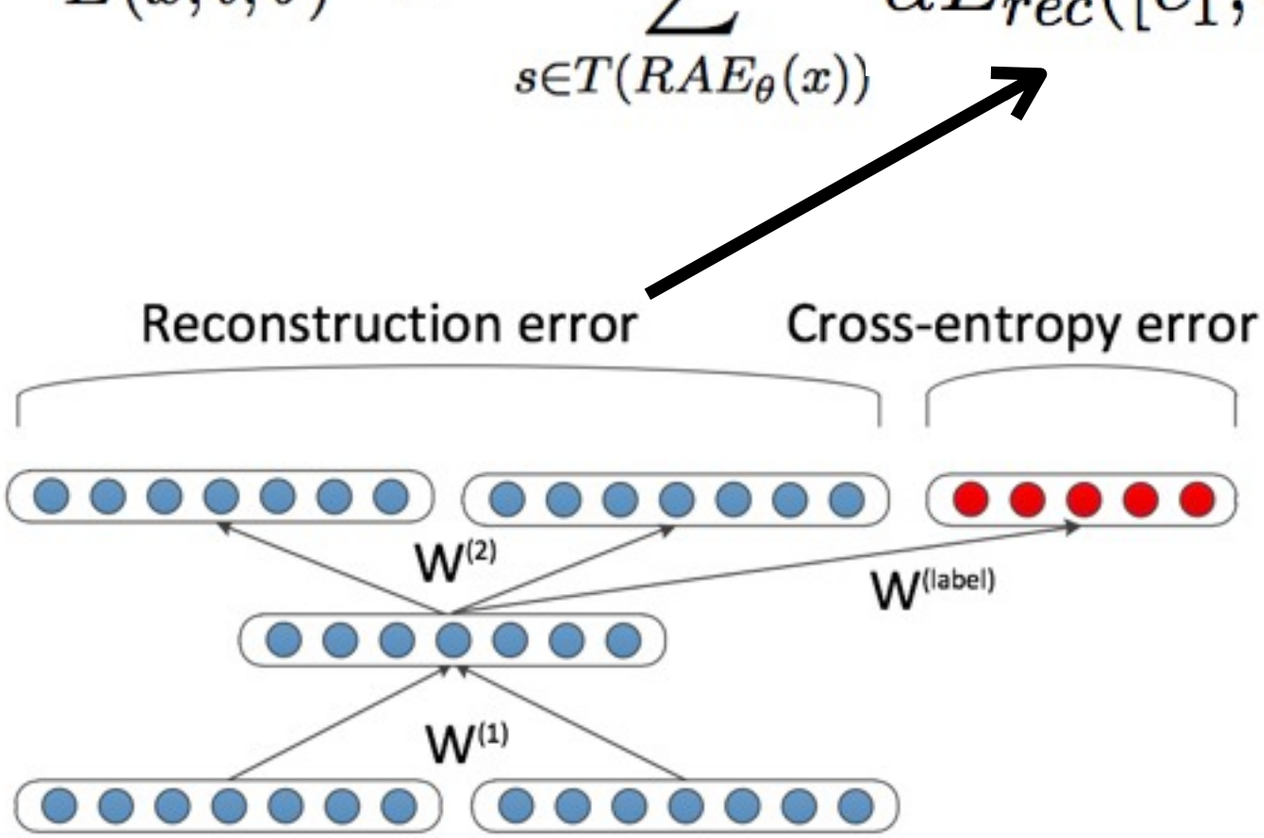
recursive-way

Recursive Autoencoder

Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions (EMNLP2011)

$$J = \frac{1}{N}\sum_{(x,t)} E(x,t;\theta) + \frac{\lambda}{2}||\theta||^2$$

$$E(x,t;\theta) = \sum_{s \in T(RAE_\theta(x))} \alpha E_{rec}([c_1;c_2]_s;\theta) + (1-\alpha)E_{cE}(p_s,t;\theta)$$



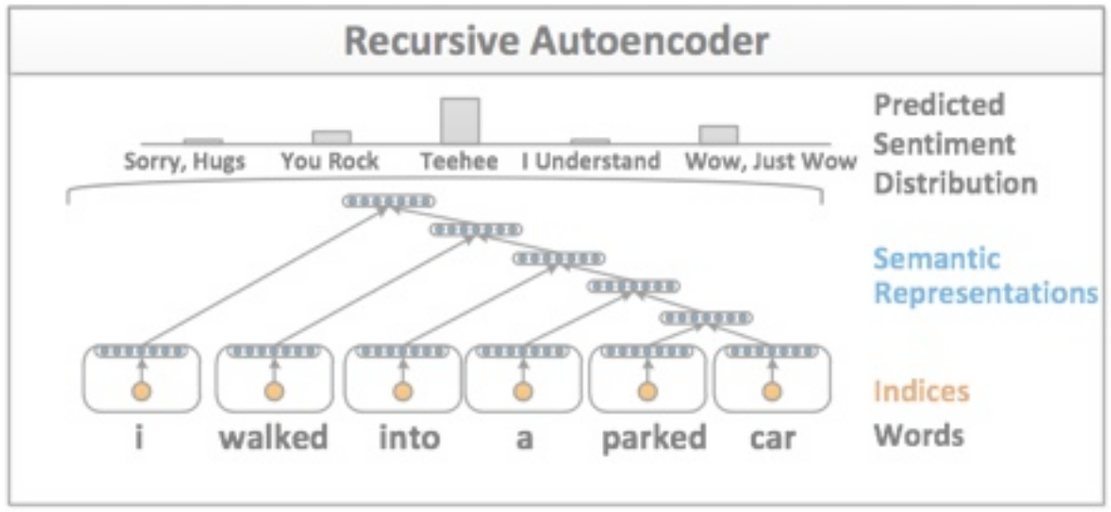
感情分布の推定

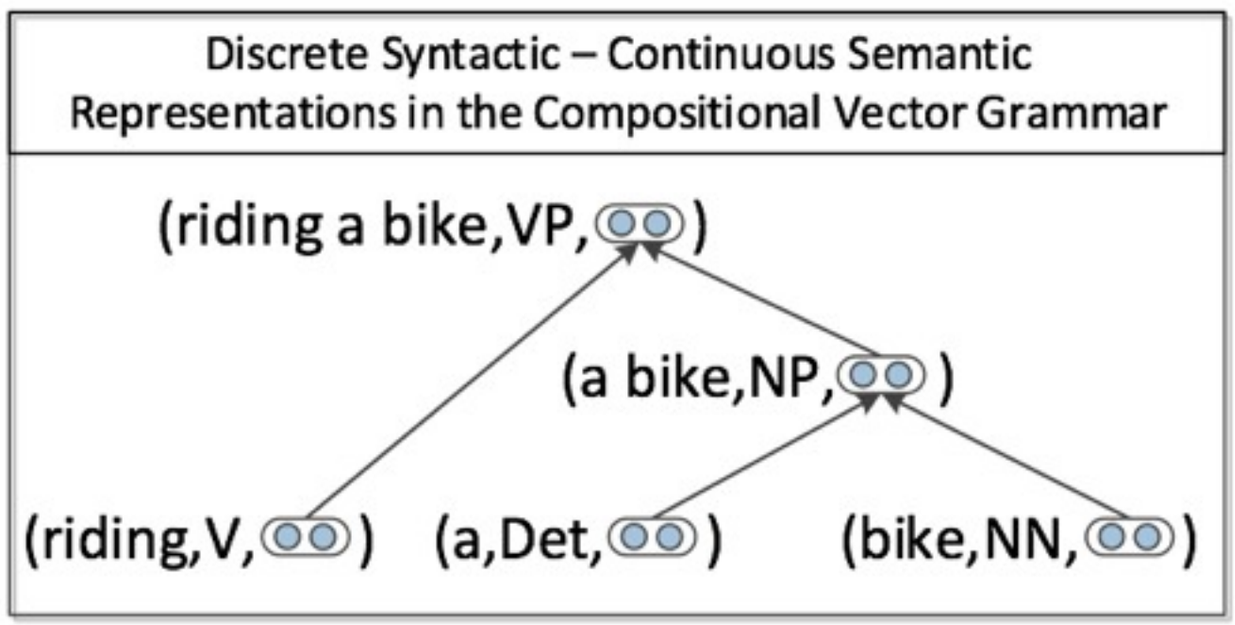
recursive-way

Richard Socher一派

Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions (EMNLP2011)

Recursive Autoencoder





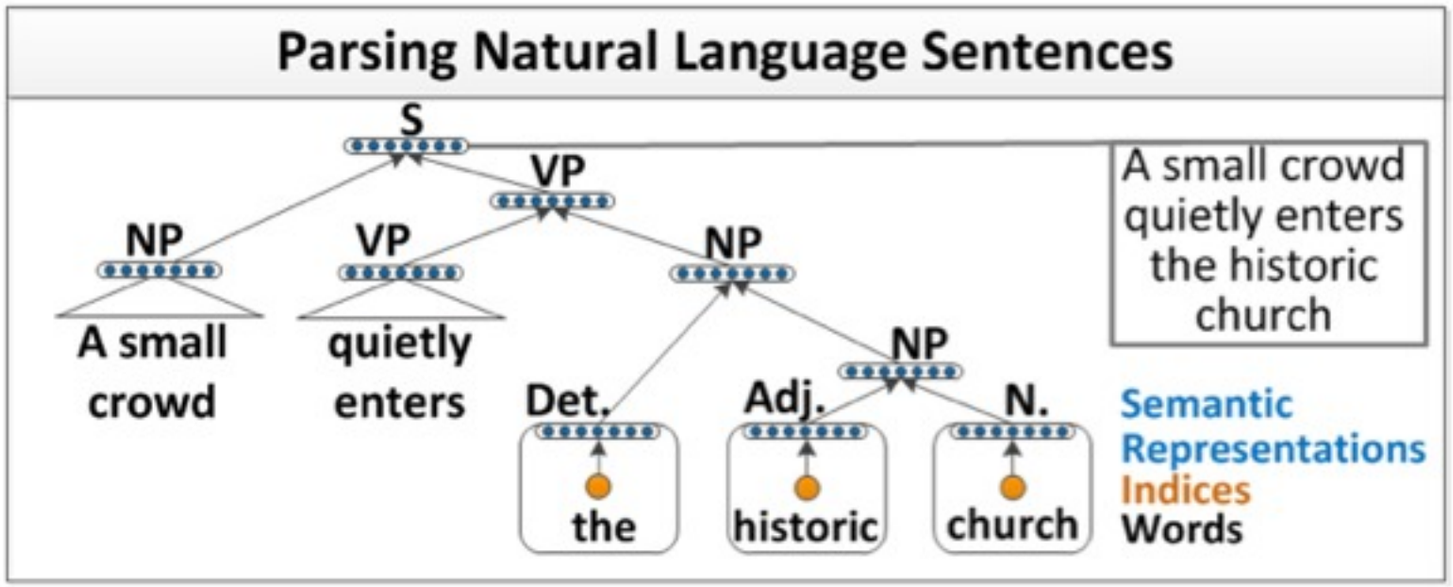
Recursive Neural Network

Parsing Natural Scenes and Natural Language with Recursive Neural Networks (ICML2011)

recursive-way

Recursive Neural Network

Parsing Natural Scenes and Natural Language with Recursive Neural Networks (ICML2011)

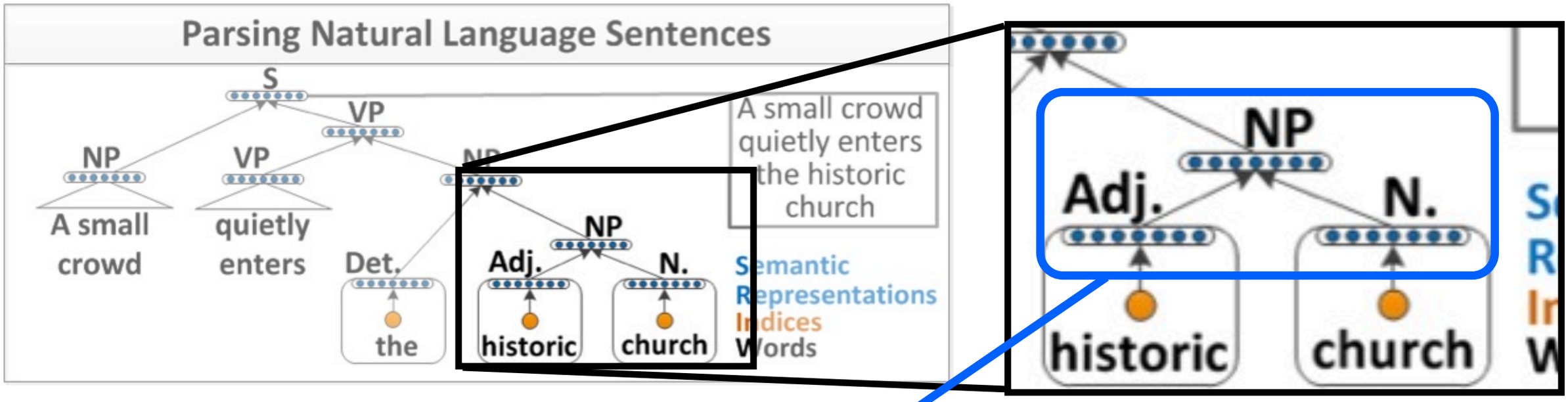


Parsing

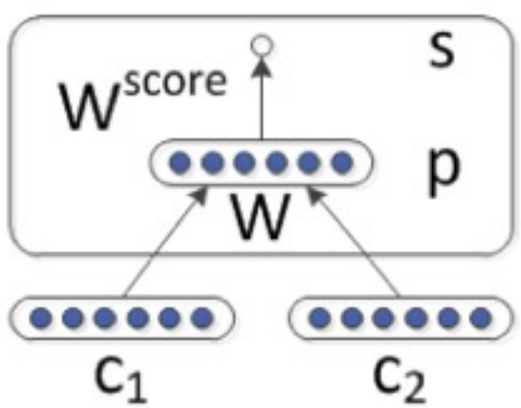
recursive-way

Recursive Neural Network

Parsing Natural Scenes and Natural Language with Recursive Neural Networks (ICML2011)



Neural Network



$$s = W^{score} p \quad (9)$$

$$p = f(W[c_1; c_2] + b)$$

Parsing

recursive-way

Recursive Neural Network

Parsing Natural Scenes and Natural Language with Recursive Neural Networks (ICML2011)

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} r_i(\theta) + \frac{\lambda}{2}||\theta||^2, \quad \text{where} \qquad (5)$$

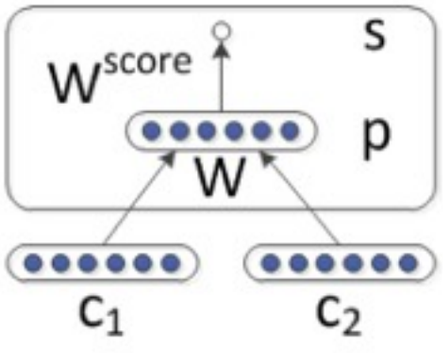
$$r_i(\theta) = \max_{\hat{y}\in\mathcal{T}(x_i)} \left(s(\mathrm{RNN}(\theta, x_i, \hat{y})) + \Delta(x_i, l_i, \hat{y})\right)$$

:モデル出力の最大スコア

$$- \max_{y_i\in Y(x_i,l_i)} \left(s(\mathrm{RNN}(\theta, x_i, y_i))\right)$$

:正解構文木のスコア

構文木中の全非終端ノードのsの総和



$$s = W^{score}p \qquad (9)$$

$$p = f(W[c_1; c_2] + b)$$

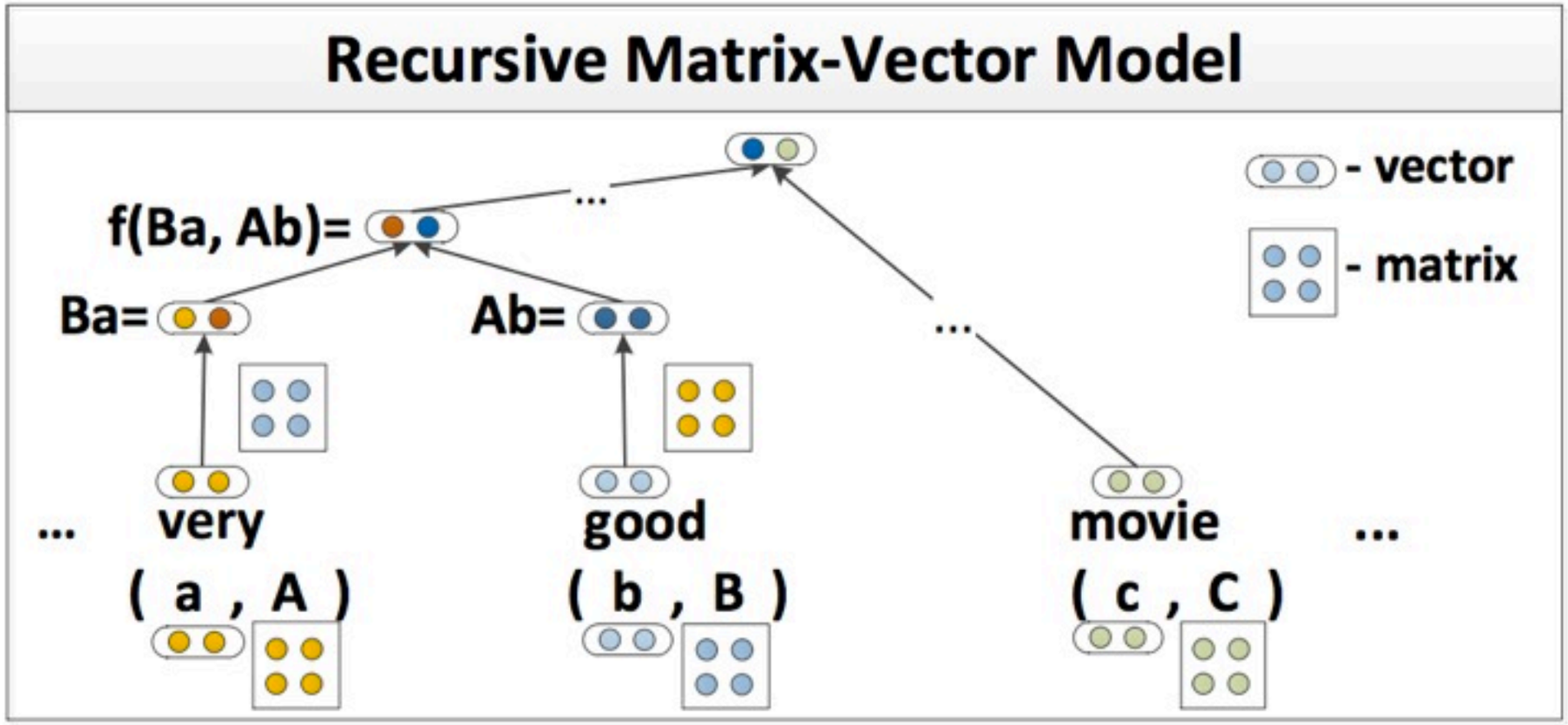
Parsing

Socher+'s

ほかの

Compositional semantics

Semantic Compositionality through Recursive Matrix-Vector Spaces (EMNLP2012)



各単語にベクトルと行列を割り当て

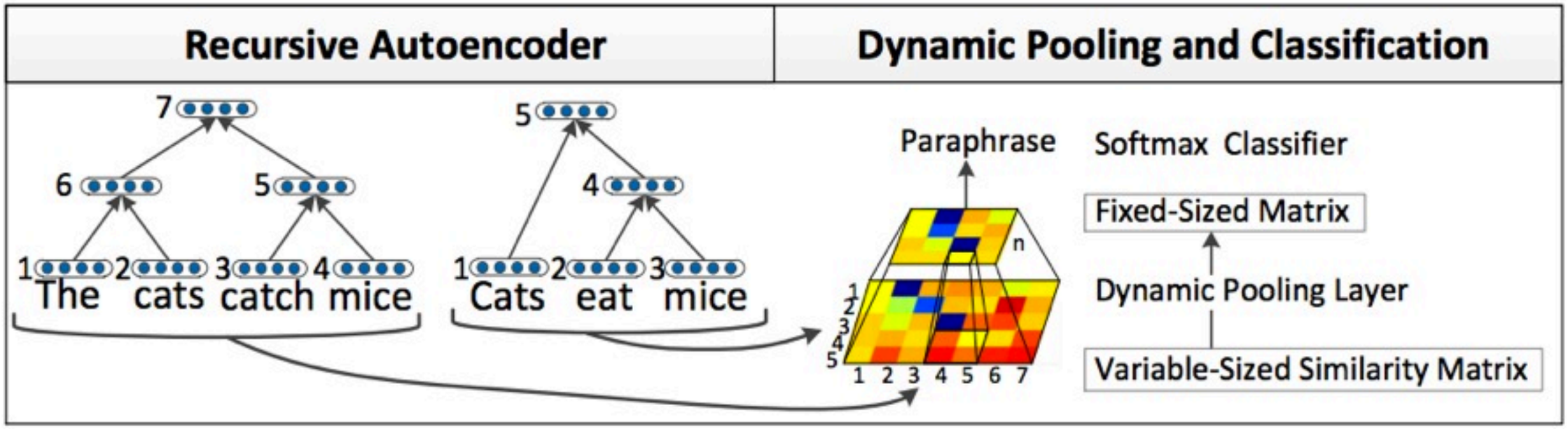
意味

隣接語への影響

not, very ...

Paraphrase

Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection (NIPS2011)



入力: 二つの文

出力: 二つの文が同じ意味か否か

- **Unfolding RAE**で文の意味
- **Dynamic pooling**で任意の長さの２文を比べる

Keywords

任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

- convolutional-way
- recursive-way

Distributed word representation

phrase, sentence-level representation

Neural language model

単語の表現方法としての
密で低次元な連続値ベクトル(word embedding)

Keywords

任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

- convolutional-way
- recursive-way

phrase, sentence-level representation

Distributed word representation

Neural language model

word embeddingsを学習するためのアプローチ

ex) Bengio’s, recurrent, ranking

Keywords

任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

- convolutional-way
- recursive-way

Distributed word representation

phrase, sentence-level representation

Neural language model

文ごとに長さが異なるのを扱うアプローチ

Keywords

任意の長さの**文**を入力とするには？？

単語(句や文も)をどうやって表現する？？

- convolutional-way
- recursive-way

phrase, sentence-level representation

Distributed word representation

Neural language model

Recursiveな方は途中の**phrase**や**sentence**における単語ベクトルも保存

NN→多層×→pretraining→breakthrough !!

Mitsumasa Kubo 17:56

Deep learning ってなんかニューラルネットワークの超パワーアップ版なんだっけ、ぐらいの前提知識からNLPへの活用・展望みたいなのを具体例を交えつつ、丁寧にレクチャーしてくれるとうれしい！！

時間は30~60分ぐらいでお好きに！

逆に最低これくらいは読んで理解しとけや、みたいな資料などがあれば事前に共有しておく

焦って早口過ぎてたら教えて下さい

2つのモチベーション

- NLPでニューラルネットを
- 言語の意味的な特徴を

具体例の説明が重くなりすぎたかも...

NN→多層×→pretraining→breakthrough !!

Mitsumasa Kubo 17:56

Deep learning ってなんかニューラルネットワークの超パワーアップ版なんだっけ、ぐらいの前提知識からNLPへの活用（展望みたいなのを具体例を交えつつ、丁寧にレクチャーしてくれるとうれしい！！

時間は30~60分ぐらいでお好きに！

逆に最低これくらいは読んで理解しとけや、みたいな資料などがあれば事前に共有しておく

焦って早口過ぎてたら教えて下さい

2つのモチベーション
- NLPでニューラルネットを
- 言語の意味的な特徴を

具体例の説明が重くなりすぎたかも...

Distributed (Word|Phrase|Sentence|Document) Representation

Recursive Autoencoder一強

他の枠組みは？

どうする？

よりよい単語の表現

Neural Language Model

意味？？

Compositional Semanticsという
タスク自体は，**deep learning**
以外でも最近盛ん

既存タスクへの応用

単語類似度，分類，構造学習...

↓

要約，翻訳，推薦，...？

- 学習された単語の**embedding**を追加素性に使う

他の方法は？

おわり