

データサイエンス

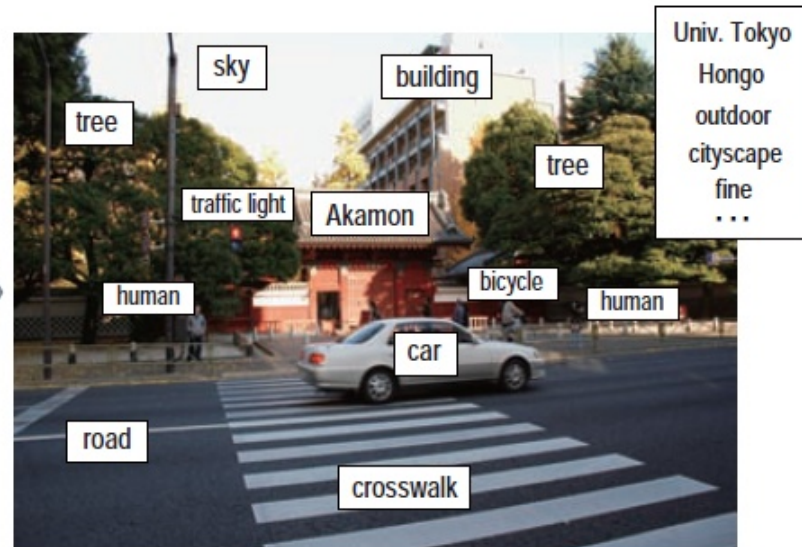
第12回

～深層学習（画像認識・自然言語処理）～

情報理工学系研究科
創造情報学専攻
中山 英樹

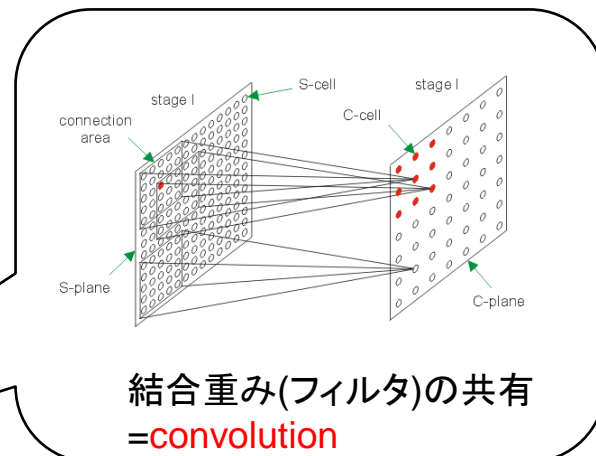
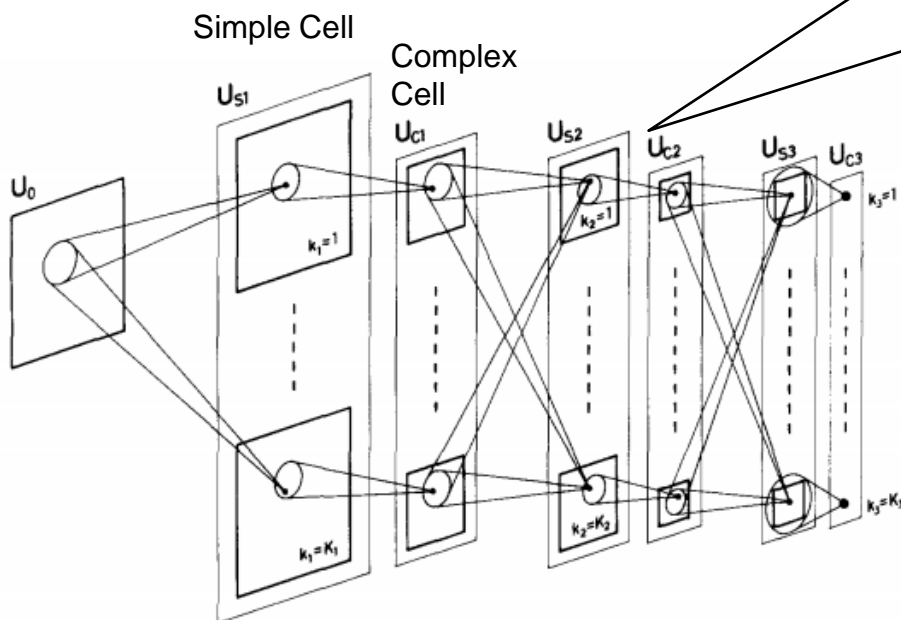
一般物体認識（一般画像認識）

- 制約をおかない実世界環境の画像を言語で記述
 - 一般的な物体やシーン、形容詞、印象語
 - 2000年代以降急速に発展（コンピュータビジョンの人気分野）
 - 幅広い応用先



Neocognitron

- 福島邦彦先生、1980年代前後
 - 畳み込みニューラルネットワークの原型
 - Simple Cell → Complex Cell の結合はpoolingに対応
 - 段階的に解像度を落としながら、局所的な相関パターンを抽出

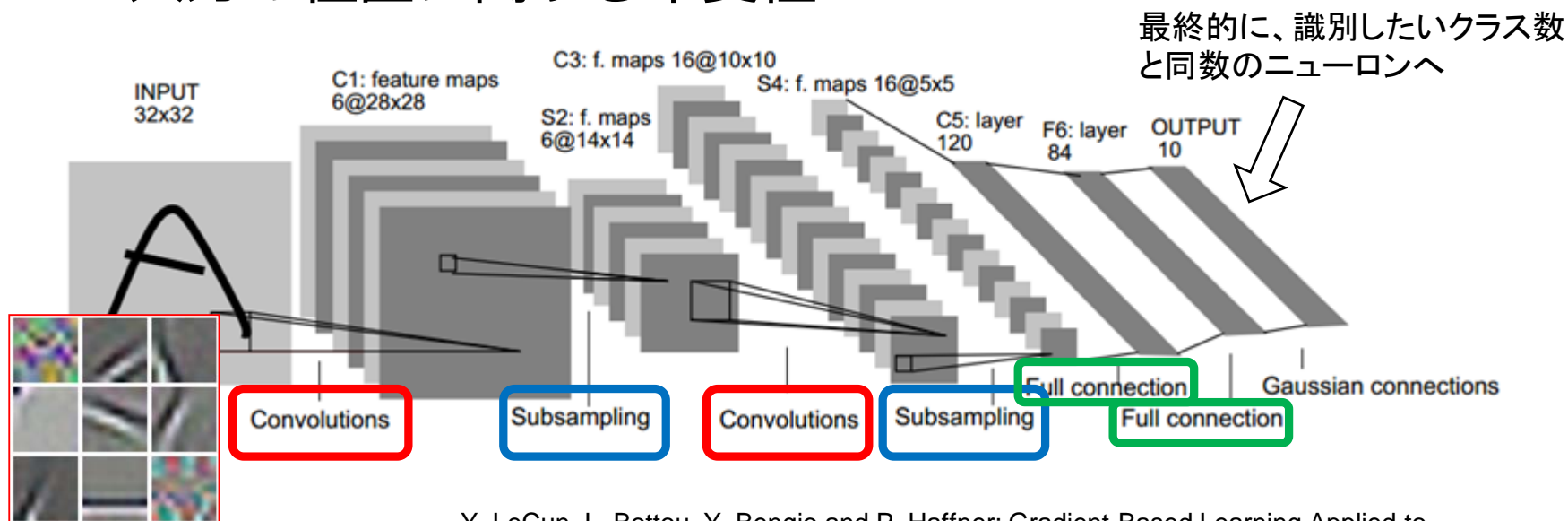


<http://www.kiv.zcu.cz/studies/predmety/uir/NS/Neocognitron/en/func-C-cell.html>

Kunihiro Fukushima, "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position", Biological Cybernetics, 36(4): 93-202, 1980.

Convolutional neural network (CNN)

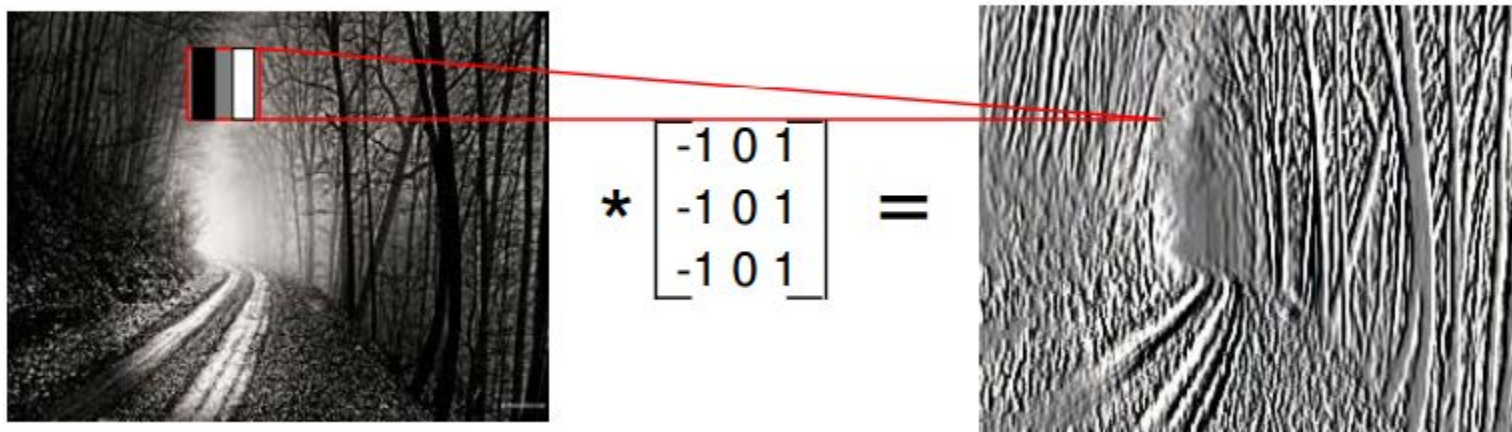
- 局所領域(受容野)の畳み込みとプーリングを繰り返す多層パーセプトロン
- 単純な全結合ネットワークと比べて大幅にパラメータ数を削減
- 入力的位置に関する不変性



Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 86(11):2278-2324, 1998.

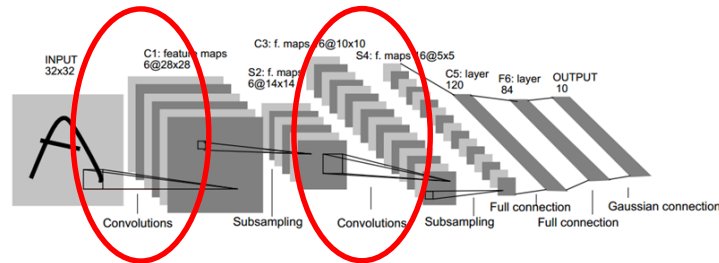
Convolution

- 一般的なフィルタだと…
 - 例) エッジ抽出

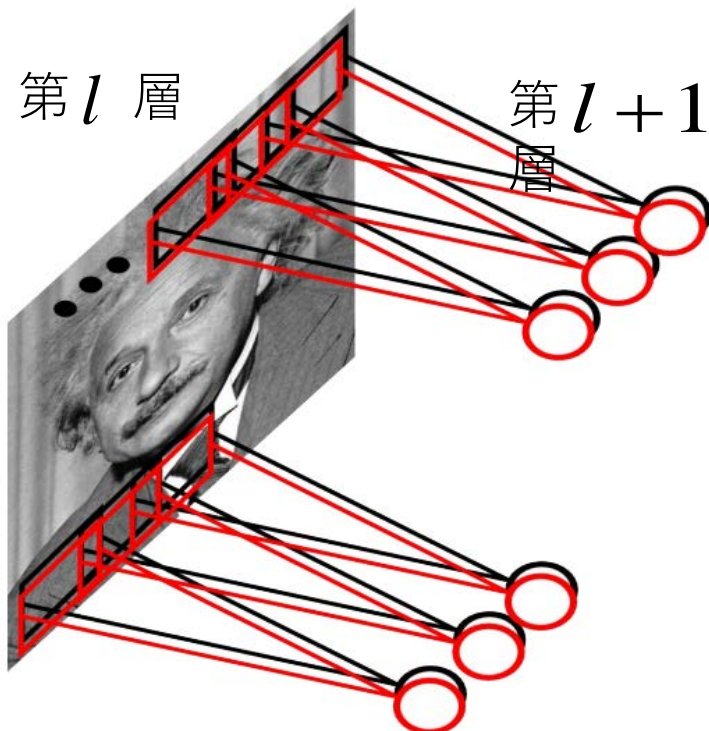


- CNNでは識別に有効なフィルタ係数をデータから学習する

畳み込み層



- 各フィルタのパラメータは全ての場所で共有
 - 色の違いは異なる畳み込みフィルタを示す



※もちろん入力生画像のみとは限らない(中間層など)

非線形活性化関数(とても重要)

$$\mathbf{z}^{l+1} = h(\mathbf{W}^{l+1} * \mathbf{z}^l + \mathbf{b}^{l+1})$$

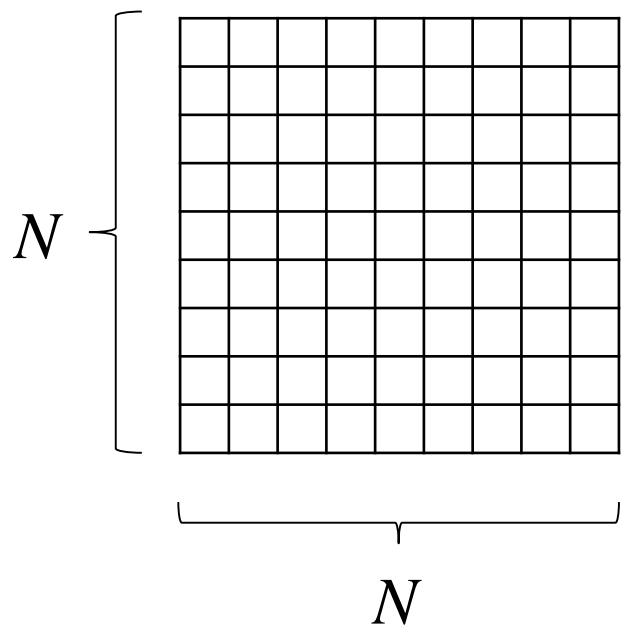
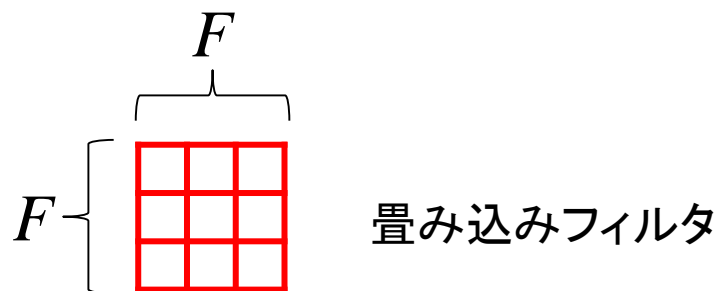
フィルタの係数

入力

バイアス

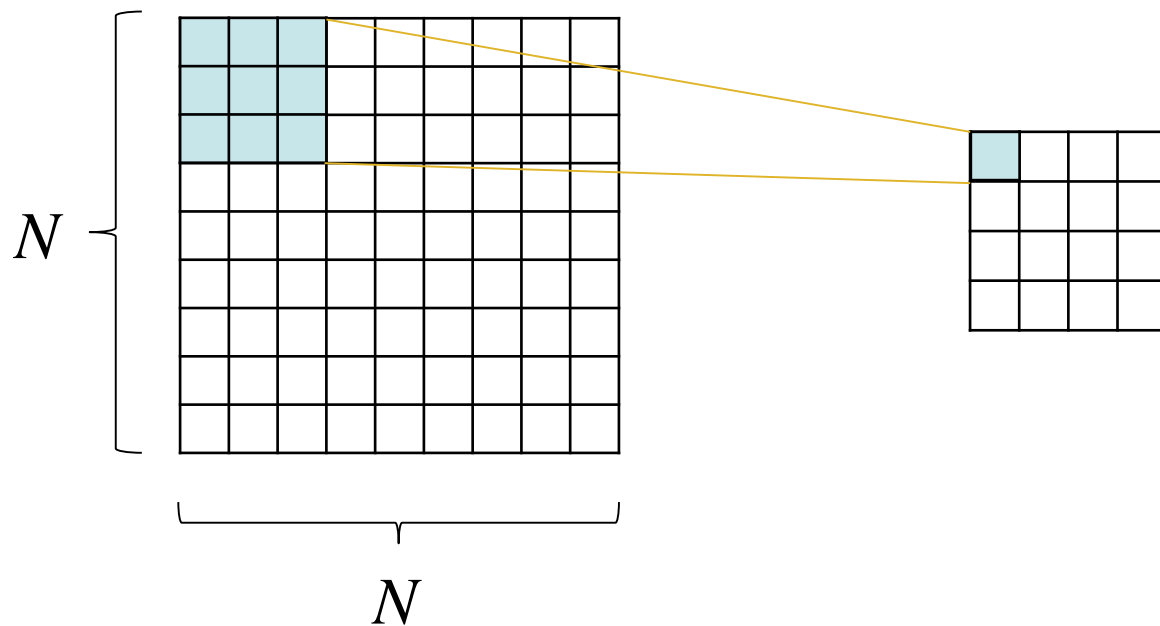
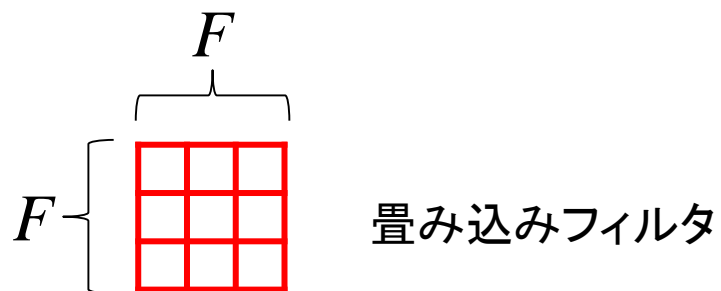
畳み込み層

- ▶ (まず簡単のため) 入力一層、フィルター一つの場合



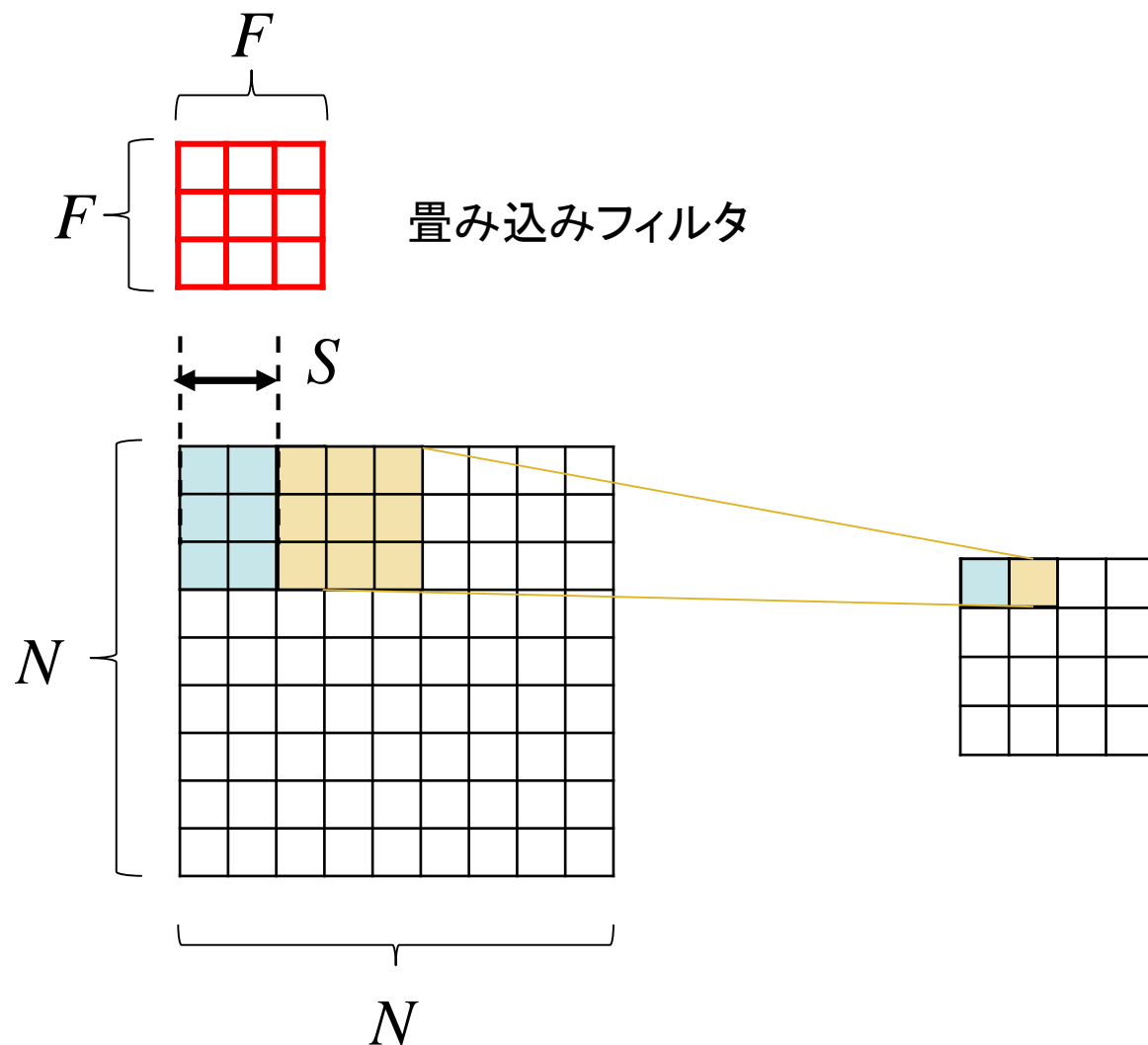
畳み込み層

- ▶ (まず簡単のため) 入力一層、フィルター一つの場合



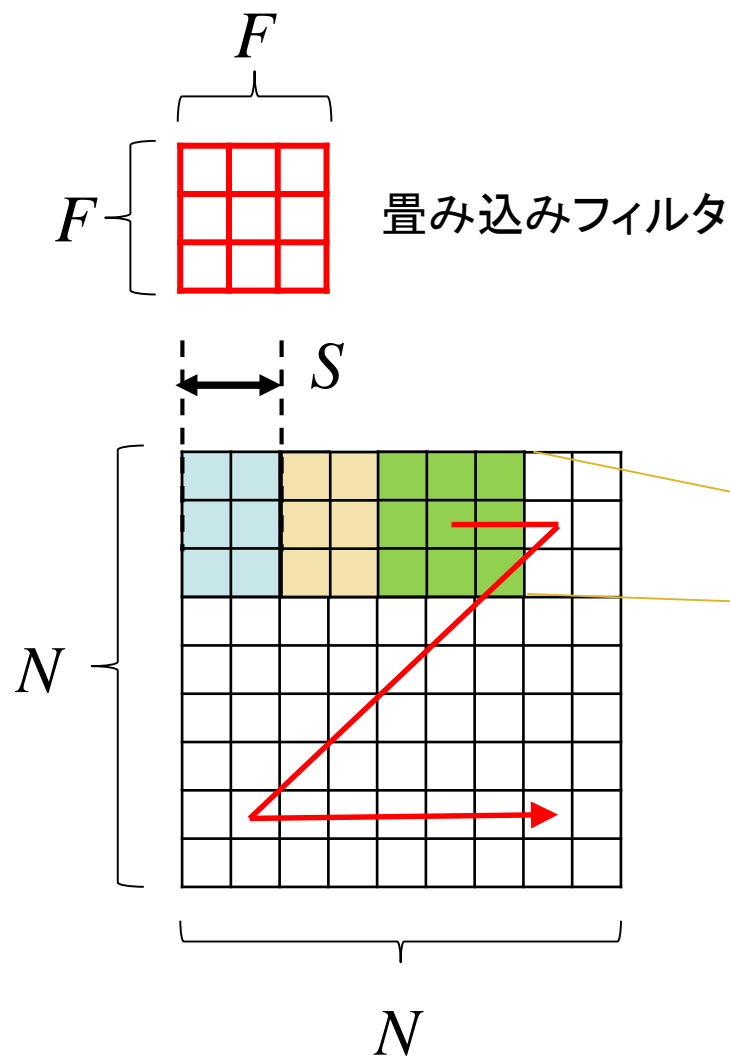
畳み込み層

- ▶ (まず簡単のため) 入力一層、フィルター一つの場合



畳み込み層

- ▶ (まず簡単のため) 入力一層、フィルタ一つの場合



注: 実際は入力層を除き、
 $S=1$ とする場合が多い
(つまり畳み込み層で解像度は落ちない)

$$(N - F) / S + 1$$

もう少し詳しく

<http://cs231n.github.io/convolutional-networks/> を改変

Zero-padding
フィルタがはみ出す分をゼロ埋め
 $(F-1)/2$

Input Volume (+pad 1) (3x7x7)

$X[0, :, :, :]$

0	0	0	0	0	0	0
0	0	1	0	2	0	0
0	1	1	0	1	2	0
0	1	1	1	0	2	0
0	2	1	0	1	2	0
0	2	2	2	1	1	0
0	0	0	0	0	0	0

$X[1, :, :, :]$

0	0	0	0	0	0	0
0	1	0	1	2	1	0
0	1	2	2	1	2	0
0	2	1	1	0	0	0
0	1	1	0	2	1	0
0	2	2	2	1	2	0
0	0	0	0	0	0	0

$X[2, :, :, :]$

0	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	2	1	2	2	0
0	1	2	1	2	2	0
0	2	1	1	2	1	0
0	0	0	2	2	0	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

$W[0, 0, :, :, :]$

-1	-1	1
-1	0	1
0	0	0

$W[0, 1, :, :, :]$

1	0	1
-1	0	1
-1	0	0

$W[0, 2, :, :, :]$

1	-1	-1
1	1	-1
-1	-1	0

Bias b_0 (1x1x1)

$b[0]$

1

Filter W1 (3x3x3)

$W[1, 0, :, :, :]$

1	0	1
0	0	1
0	0	-1

$W[1, 1, :, :, :]$

-1	1	-1
-1	1	0
-1	0	-1

$W[1, 2, :, :, :]$

1	1	-1
0	-1	1
-1	0	1

Bias b_1 (1x1x1)

$b[1]$

0

Output Volume (2x3x3)

$u[0, :, :, :]$

3	-1	-8
0	-1	-2
2	0	1

$u[1, :, :, :]$

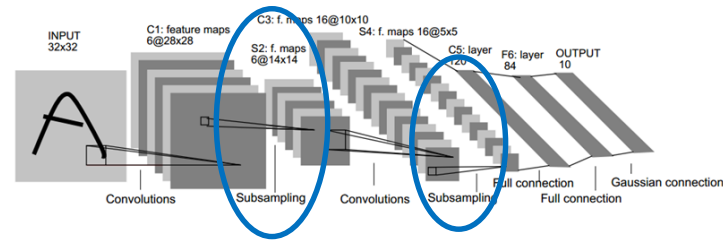
0	-1	-4
1	0	0
6	0	4



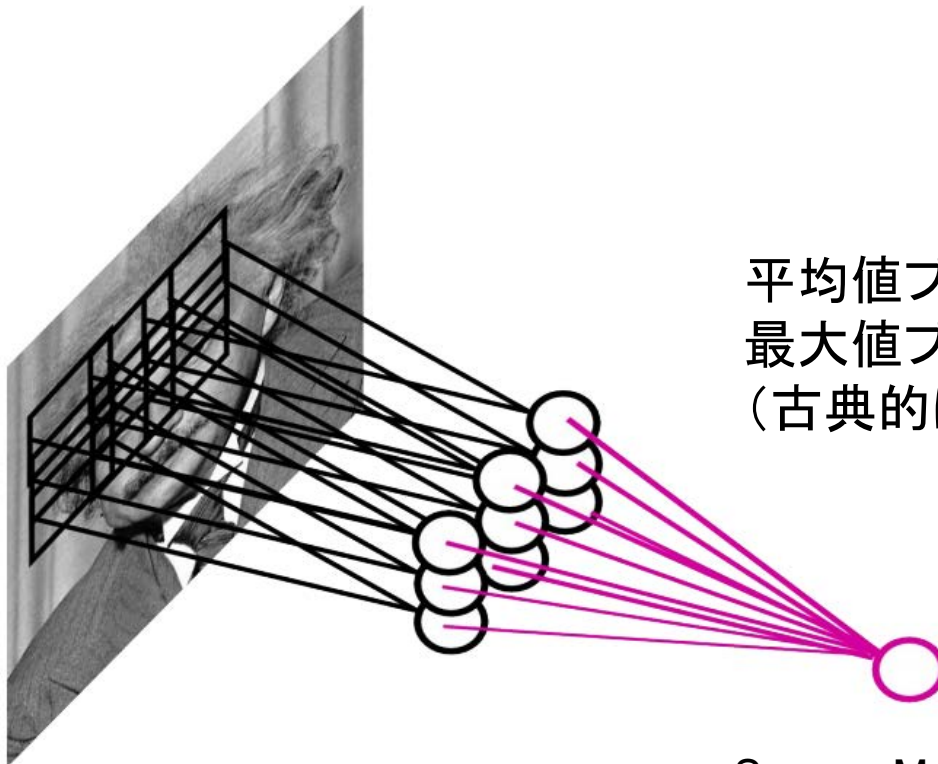
このあと活性化関数がかかる

入力: チャンネル数 3
サイズ N=7 (padding込)
フィルタ: 出力チャンネル数 2
サイズ F=3
stride S=2

プーリング層



- 一定領域内の畳み込みフィルタの反応をまとめる
 - 領域内での平行移動不変性を獲得

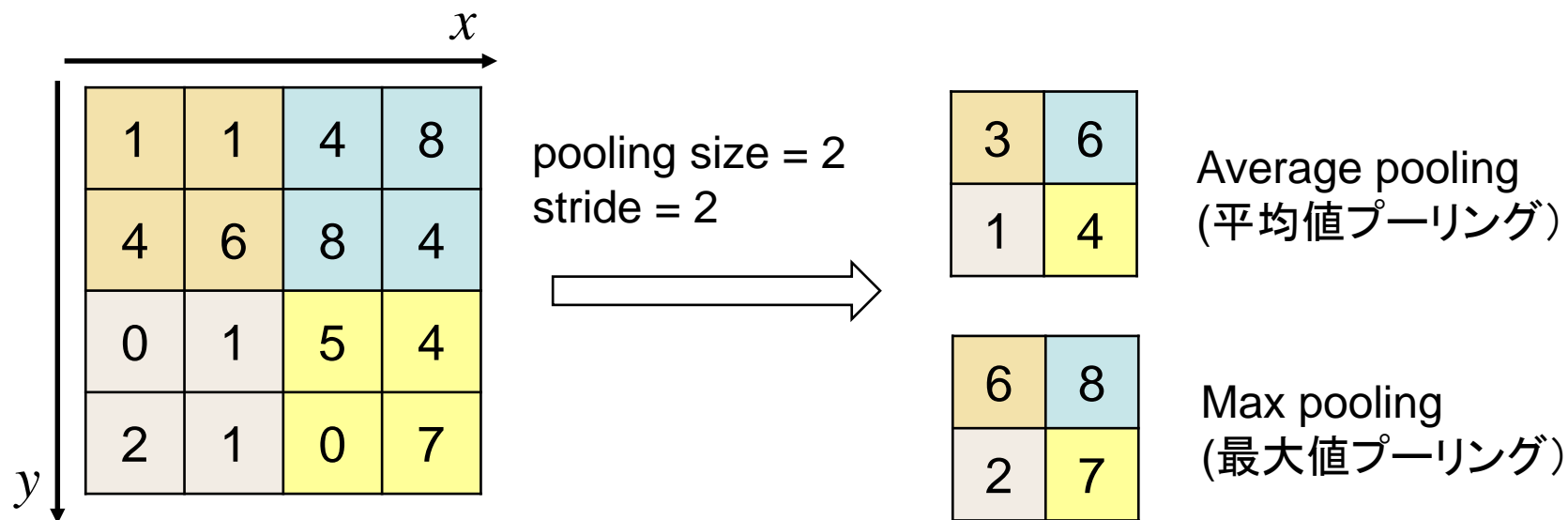


平均値プーリング、
最大値プーリングなど
(古典的には単純なサンプリング)

Source: M. Ranzato, CVPR'14 tutorial slides

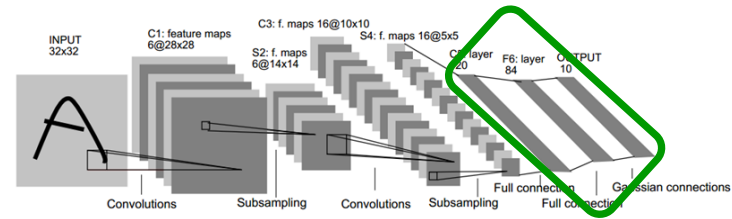
プーリング層

- ▶ Average pooling: 局所領域の平均値をとる
- ▶ Max pooling: 局所領域の最大値をとる

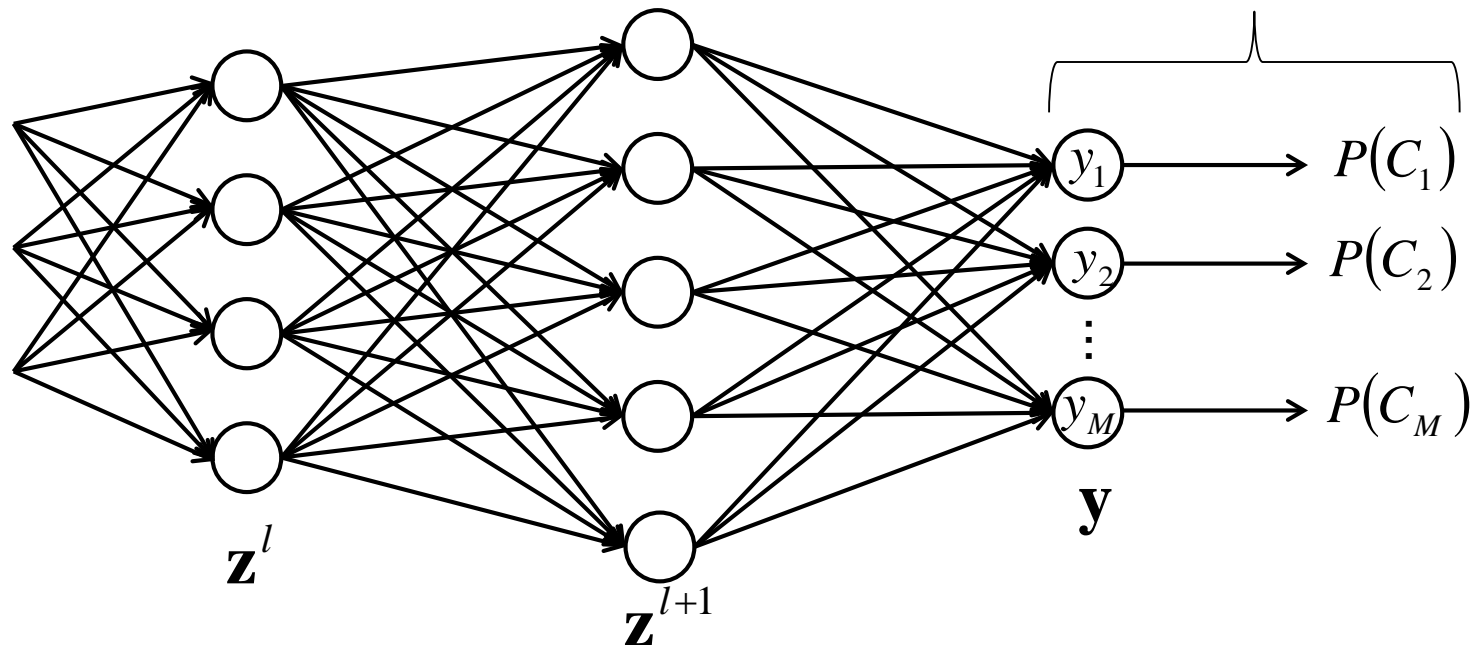


- ▶ 他にも Lp pooling, stochastic pooling などいろいろ

全結合層・出力層



- 要するにただの多層パーセプトロンです
(前回資料参照)



$$\text{Softmax} \quad P(C_i) = \frac{\exp(y_i)}{\sum_j^M \exp(y_j)}$$

Backprop: 畳み込み層

- ▶ 全受容野での誤差を束ねて更新

$z_{k,i}^l = h(u_{k,i}^l)$: k 番目の需要野の i 番目の出力値 (第 l 層)

$\delta_{k,i}^l$: k 番目の需要野の i 番目の誤差値 (第 l 層)

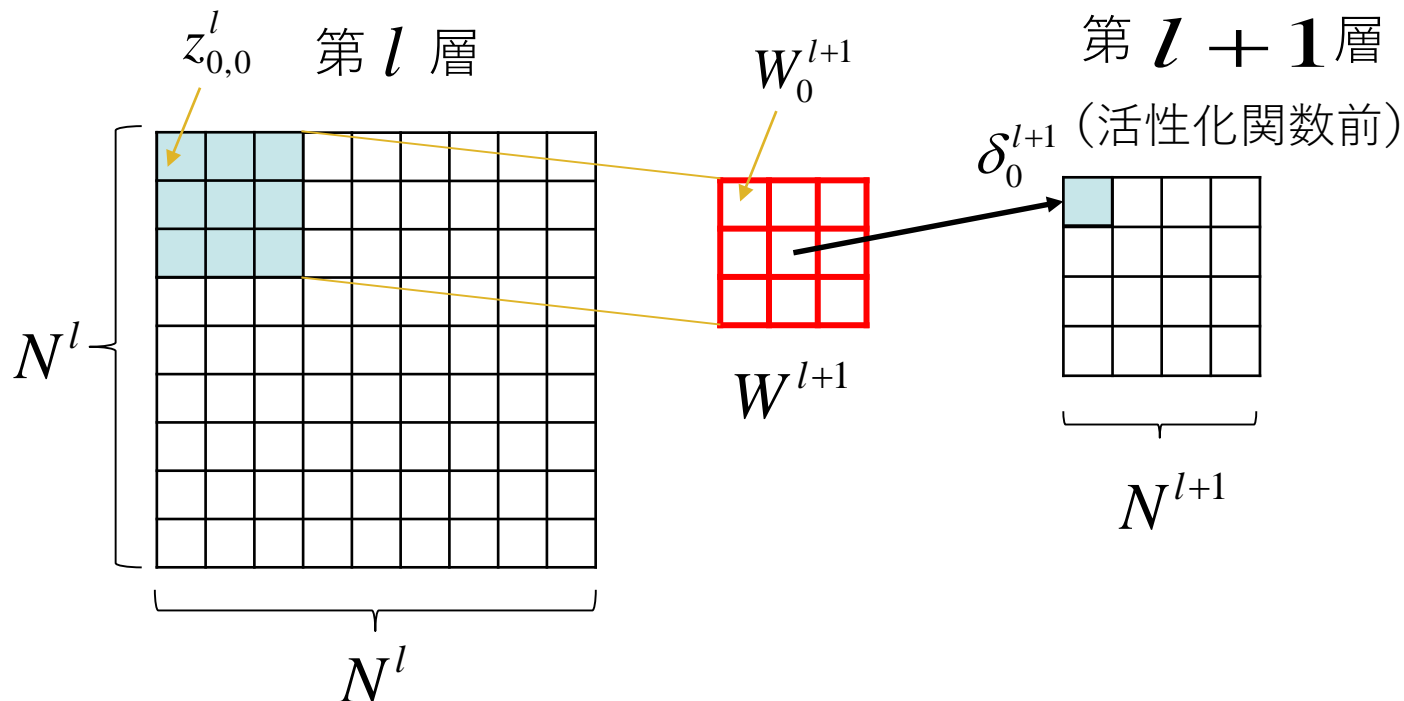
δ_k^{l+1} : 第 $l+1$ 層の対応する場所の誤差

W_i^{l+1} : フィルタの i 番目の係数

$$\frac{\partial J}{\partial W_i^{l+1}} = \sum_{k=0}^{(N^{l+1})^2 - 1} \delta_k^{l+1} z_{k,i}^l$$

$$\delta_{k,i}^l = h'(u_{k,i}^l) \sum_j \delta_j^{l+1} W_{i'}^{l+1}$$

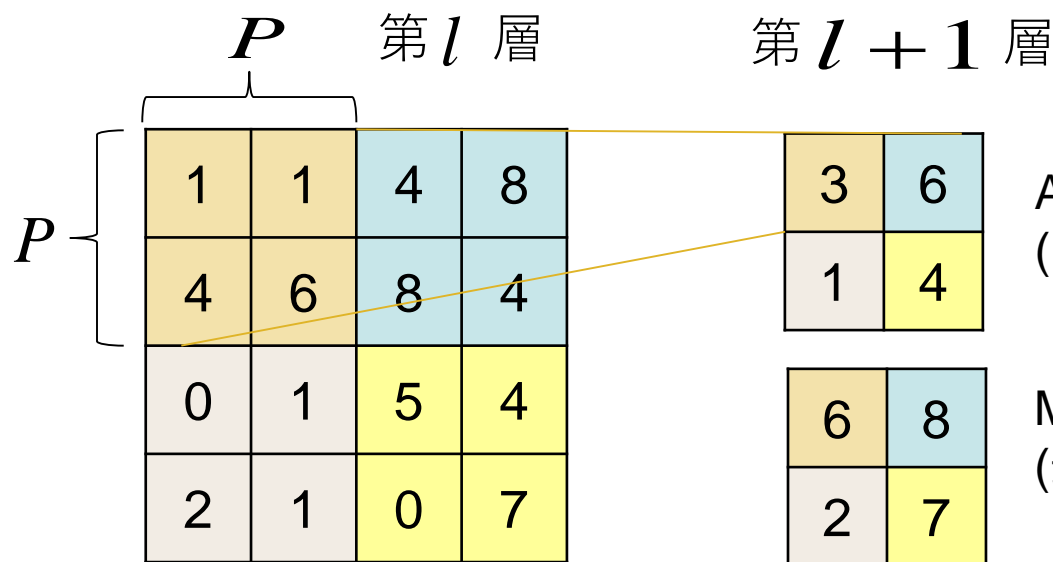
結合がある部分
のフィルタ係数



Backprop: プーリング層

- ▶ Average pooling の場合は簡単
- ▶ Max pooling の場合、feedforward時に選ばれたユニットにのみ誤差が伝播する (覚えておく必要がある)

$$\delta_{k,i}^l = \sum_i^{P^2} \delta_k^{l+1} \underline{W_i^{l+1}}$$



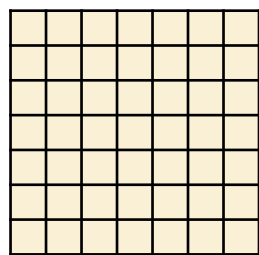
Average pooling (平均値プーリング) $\delta_{k,i}^l = \delta_k^{l+1} / P^2$

Max pooling (最大値プーリング)

$$\delta_{k,i}^l = \begin{cases} \delta_k^{l+1} & \text{if } i = \arg \max_j (z_{k,j}^l) \\ 0 & \text{otherwise.} \end{cases}$$

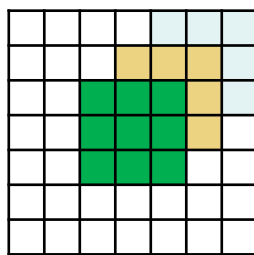
Deeper is better

- 7 x 7 の畳み込みは、3 x 3の畳み込み層を3つ積めば意味的に等価



$$7 \times 7 = 49$$

\doteq



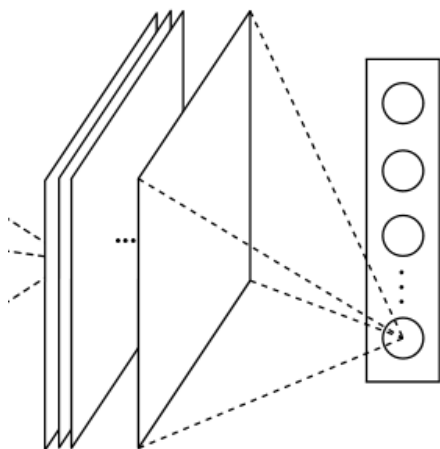
$$3 \times (3 \times 3) = 27$$

より少ないパラメータで、
より深い非線形性！

- ▶ 現在は、3 x 3や1 x 1の小さな畳み込み層をたくさん積むのが基本
 - 更に、3x3を3x1と1x3にばらす(factorization)することも…

全結合層はいらない？

- CNNのパラメータの大半は全結合層に集中
 - あくまで一層内の線形結合。非線形性は増えない。
 - ないよりはあった方がよいが、割に合わない
- 最近のCNNの多くは全結合層を持たない
 - Global average pooling：最終層の平均値プーリングをとり、そのままsoft maxへ入力



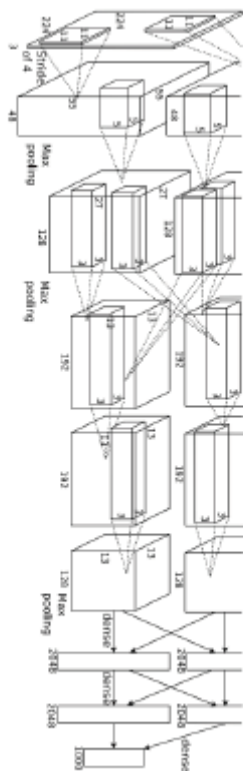
Min Lin et al., "Network In Network", In Proc. ICLR, 2014.

更に深く、広く…

2015 MSRA
(152層)

- 2012年以降劇的な向上が続いてきた

2012 AlexNet
(8層)



2014 VGG
(19層)



2014 GoogLeNet
(22層)



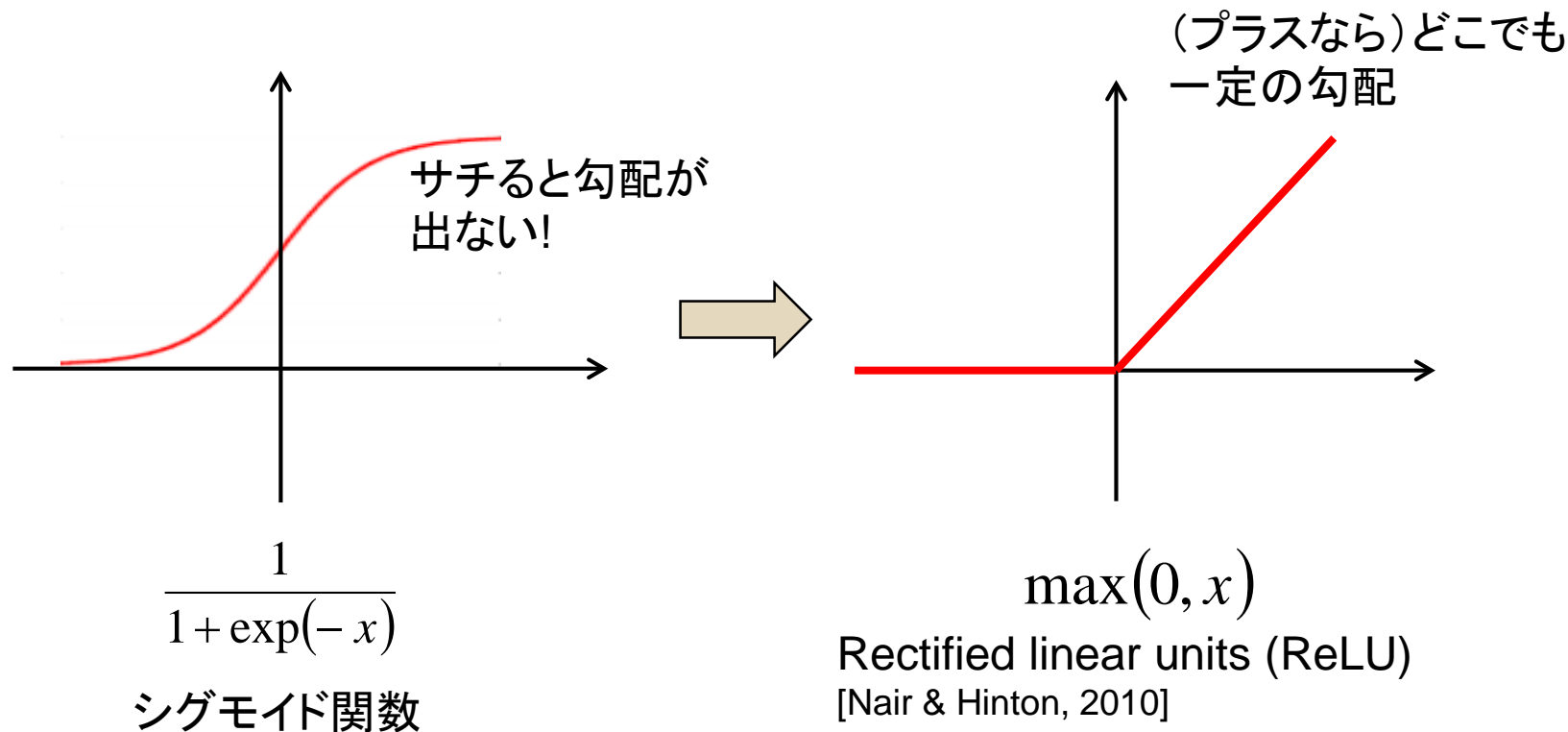
深いNNの学習を支える技術

- ネットワークの構成要素
 - 活性化関数：ReLU
 - 過学習抑制手法：Dropout
 - バッチ正規化
 - ネットワーク初期化法
 - Residual learning → 100~1000層
- 最適化手法
 - SGD、Adadelata、Adam、etc.
- 画像認識特有の工夫
 - 画像の前処理
 - データ拡張

} ~10層

} 20~30層

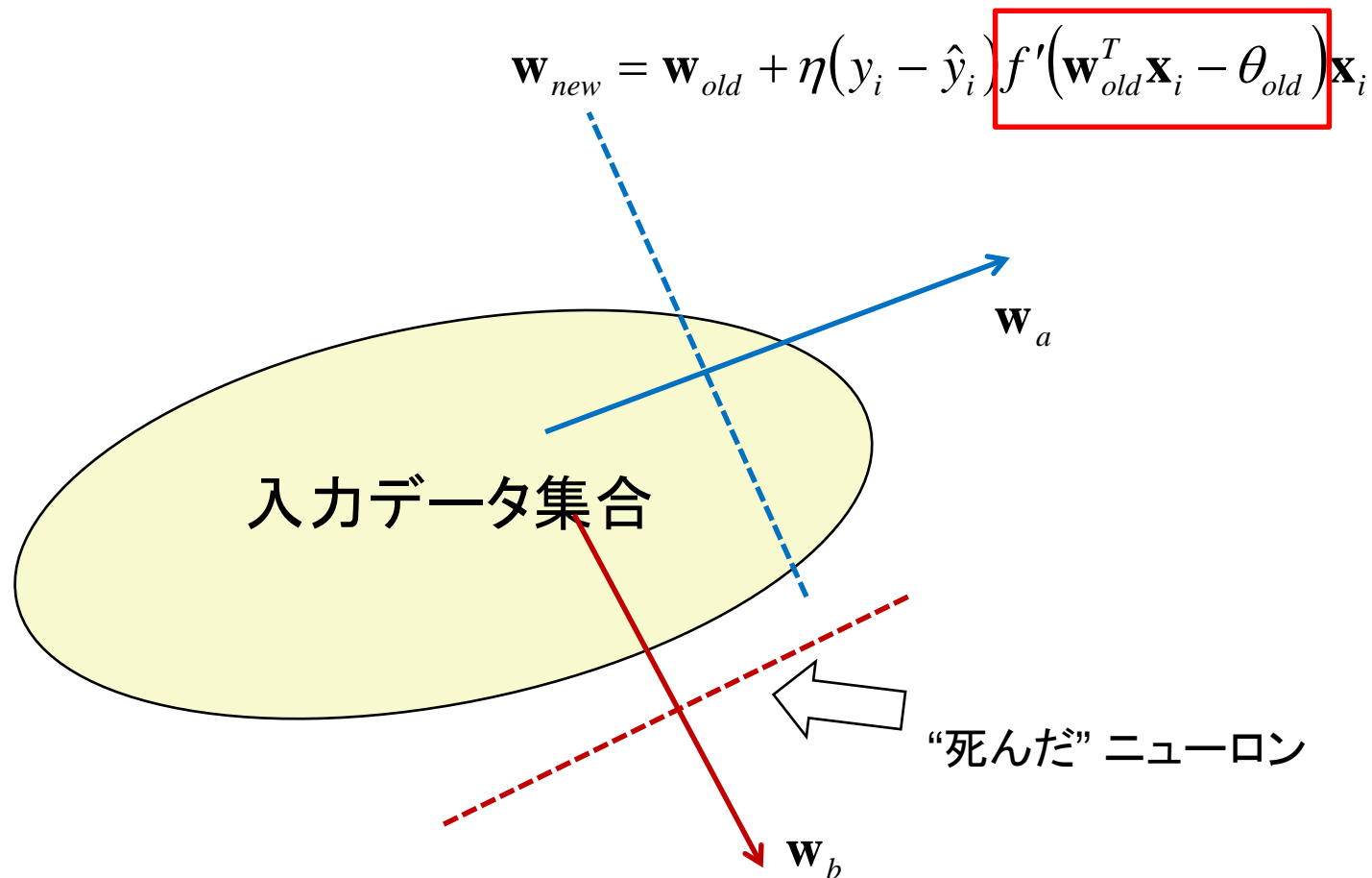
Rectified Linear Units (ReLU)



例) 単純パーセプトロン $\mathbf{w}_{new} = \mathbf{w}_{old} + \eta(y_i - \hat{y}_i) f'(\mathbf{w}_{old}^T \mathbf{x}_i - \theta_{old}) \mathbf{x}_i$
の更新式

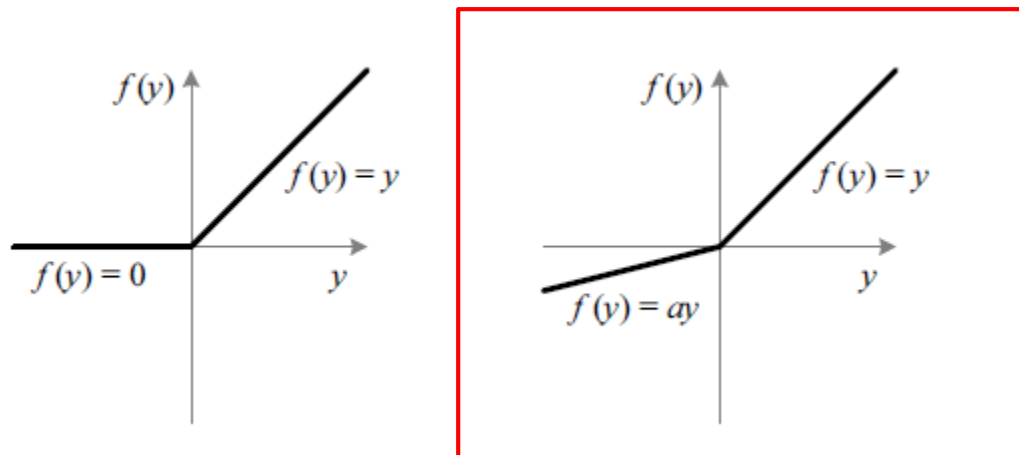
ReLUの弱点

- すべての入力データが負になると、常に微分がゼロとなる
 - パラメータが二度と更新されなくなる



Leaky ReLU

- 負の側にも少し勾配を与えたReLU



- MSR (2015)
 - PReLU: 負側の勾配係数もパラメータの一つとしてチューニング
 - ILSVRC'2014 のデータセットで 4.94% error

He et al., "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", arXiv preprint, 2015.

Xu et al., "Empirical Evaluation of Rectified Activations in Convolution", arXiv preprint, 2015.

ReLUの発展形

Table 3: Non-linearities tested.

Name	Formula	Year
none	$y = x$	-
sigmoid	$y = \frac{1}{1+e^{-x}}$	1986
tanh	$y = \frac{e^{2x}-1}{e^{2x}+1}$	1986
ReLU	$y = \max(x, 0)$	2010
(centered) SoftPlus	$y = \ln(e^x + 1) - \ln 2$	2011
LReLU	$y = \max(x, \alpha x), \alpha \approx 0.01$	2011
maxout	$y = \max(W_1x + b_1, W_2x + b_2)$	2013
APL	$y = \max(x, 0) + \sum_{s=1}^S a_i^s \max(0, -x + b_i^s)$	2014
VReLU	$y = \max(x, \alpha x), \alpha \in 0.1, 0.5$	2014
RReLU	$y = \max(x, \alpha x), \alpha = \text{random}(0.1, 0.5)$	2015
PReLU	$y = \max(x, \alpha x), \alpha \text{ is learnable}$	2015
ELU	$y = x, \text{ if } x \geq 0, \text{ else } \alpha(e^x - 1)$	2015

Mishkin et al., “Systematic evaluation of CNN advances on the ImageNet”, arXiv:1606.02228v1, 2016.

Dropout [Hinton, 2012]

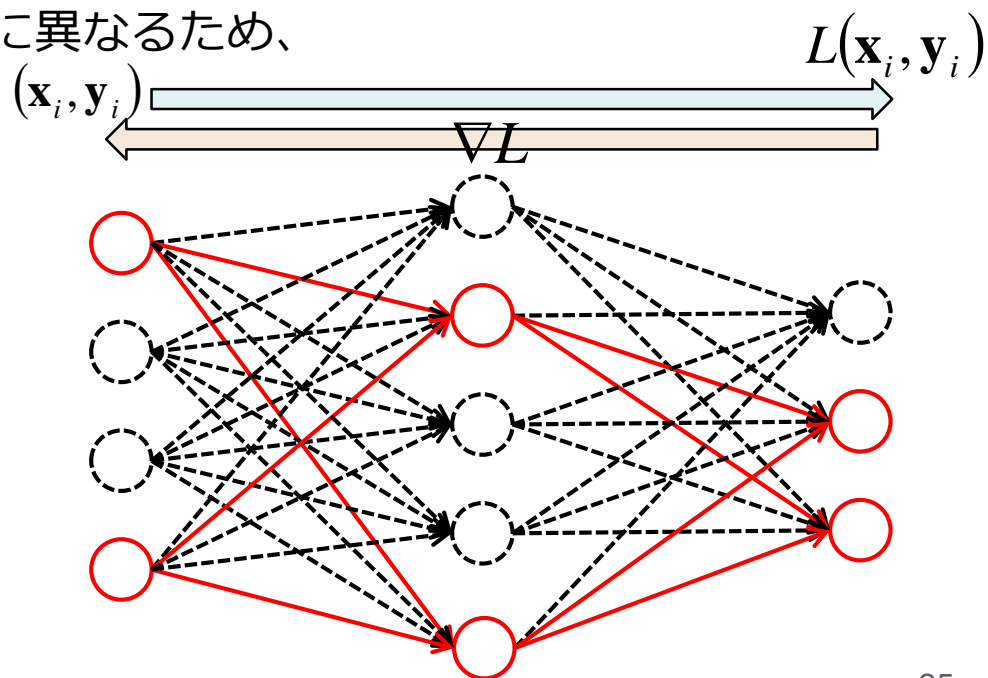
- 各訓練データのフィードバックの際に、一定確率(0.5)で中間ニューロン（ユニット）を無視
- テスト時は全ニューロンを使うが、結合重みを半分ににする

- 多数のネットワークを混ぜた構造

- 訓練データが各ニューロンごとに異なるため、バギングと同様の効果
(ただしパラメータは共有)

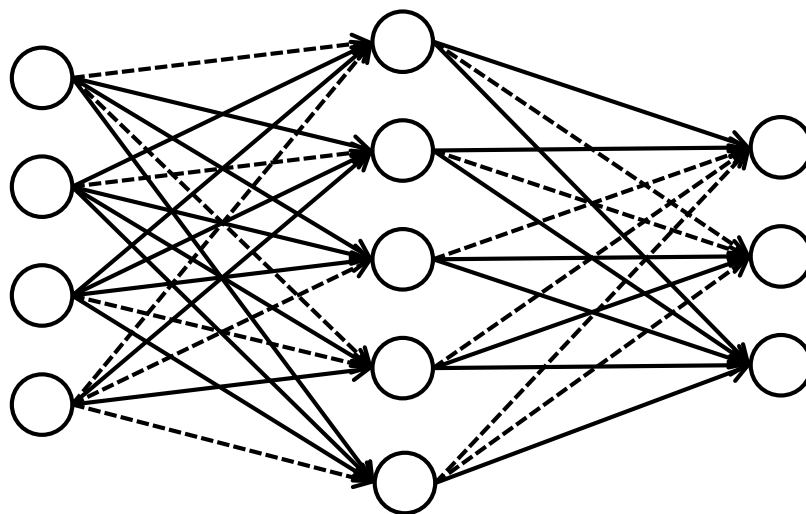
- L2正則化に近い効果

- [Wager et al., NIPS'13]



亜種

- Dropconnect [Wan et al., ICML'13]
 - ニューロンではなく、結合をランダムに落とす
 - Dropoutよりよいらしい？



- Standout [Ba et al., NIPS'13]
 - Dropoutで落とすニューロンをランダムでなく適応的に選択する

Batch normalization

- 各層で、ミニバッチごとに入力を正規化
 - 低層の変化に伴う入力の共変量シフトに追従
 - 学習を約14倍高速化、精度向上 (特に20層以上の多層モデルで効果を発揮)

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\dots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean}$$
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance}$$
$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$
$$y_i \leftarrow \underline{\gamma} \hat{x}_i + \underline{\beta} \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{ scale and shift}$$

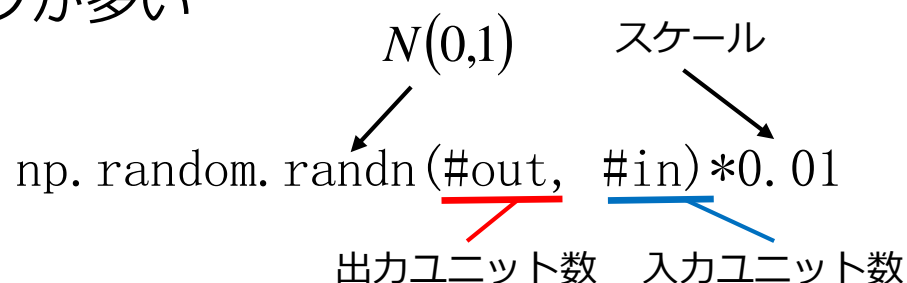
学習

ネットワーク重みの初期化

- 標準正規分布からのサンプリングが多い

- 基本形

- 小さいネットワークならOK
- スケールパラメータの調整が難しい



- Xavier Glorot and Yoshua Bengio (2010) : *Xavier initialization*

- tanh関数による活性が前提
- ReLUのような非対称な関数では前提が成立しない

$\text{np.random.randn}(\#out, \#in) / \text{np.sqrt}(\#in)$

- He et al., (2015) $\text{np.random.randn}(\#out, \#in) / \text{np.sqrt}(\#in/2)$

- 30層以上のネットワークではXavierと比較して顕著な優位性

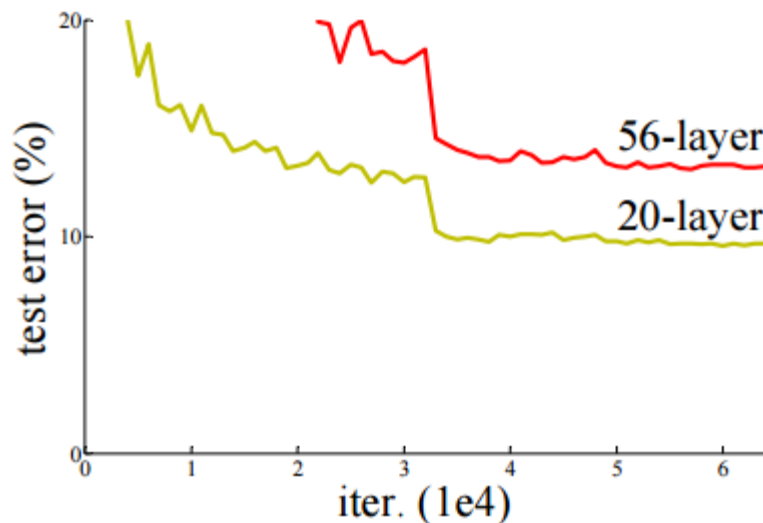
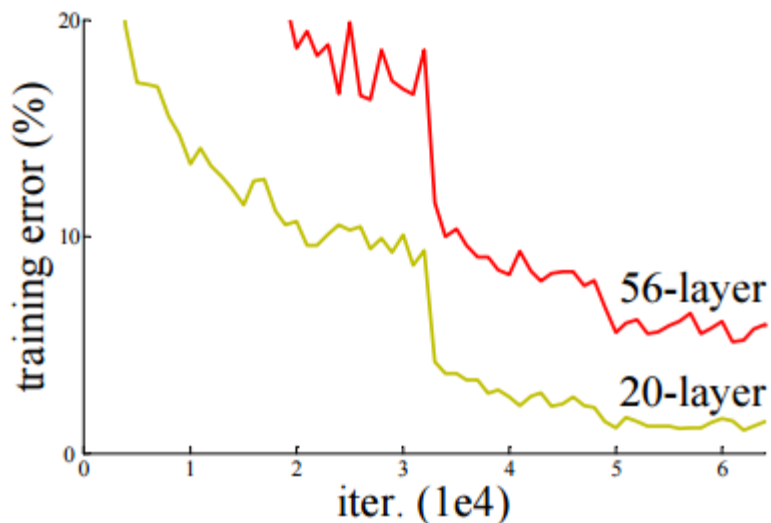
- その他も研究多数 (重要なトピックの一つ)

- C.f. Mishkin and Matas, "All you need is a good init", ICLR 2016.

クイズ

- ネットワークを大きくすればするほど…
 - 訓練誤差は、大きくなる？小さくなる？
 - テスト誤差は、大きくなる？小さくなる？

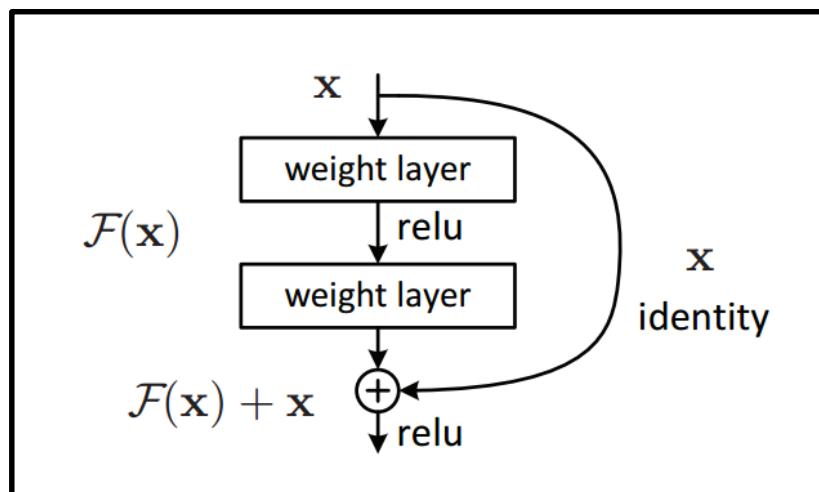
超多層ネットワークへ



He et al., "Deep Residual Learning for Image Recognition", arXiv preprint, 2015.

- 超多層(50層以上)になると, 訓練誤差もテスト誤差も大きくなる
= アンダーフィッティング
- 低層のパラメータがほとんど更新されないので, 結局学習が進まない

Deep Residual Learning (Hu et al., 2015)



- 低層の入力をバイパスする構造を入れる
- 低層のパラメータの更新速度を速める
- 様々な深さのネットワークのアンサンブル

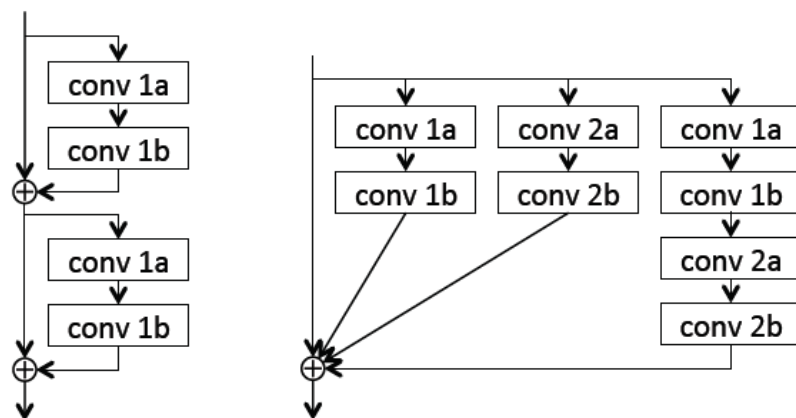


He et al., "Deep Residual Learning for Image Recognition", arXiv preprint, 2015.

Srivastava et al., "Highway Networks", ICML 2015 deep learning workshop, 2015.

ResNetを展開すると...

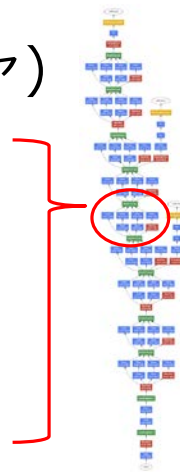
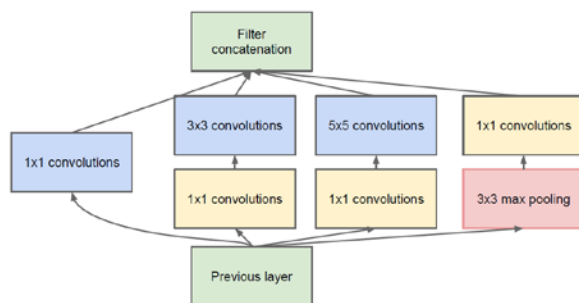
- ResNet構造は、さまざまな深さのサブネットワークのアンサンブルと等価（マルチパス）[Veit et al., 2016]



(a) 2 Residual Blocks (b) Unraveled Network of 2 Residual Blocks

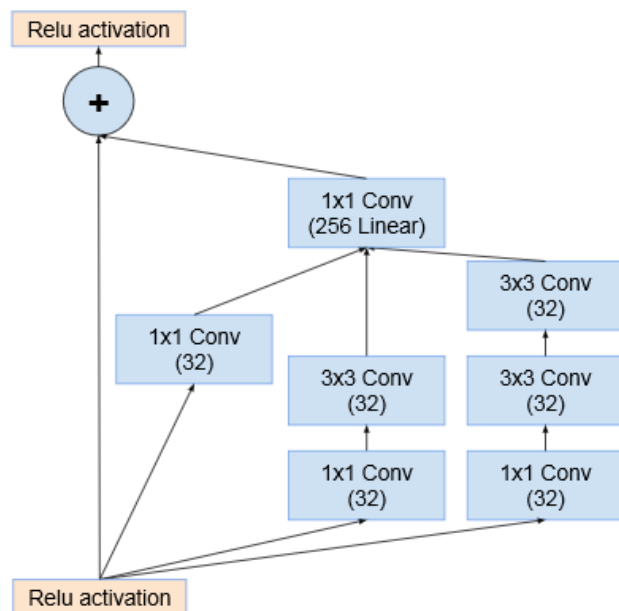
[武田, FIT'17より引用]

- 参考：GoogLeNet (Inceptionアーキテクチャ)

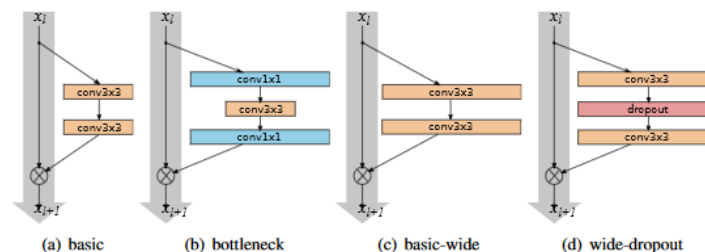


ResNet以降 (1)

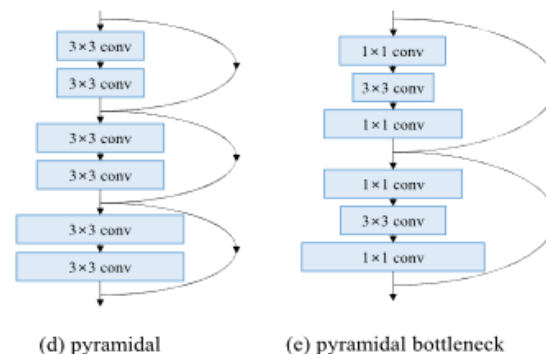
- 広さの拡張、マルチパス
- 広さと深さのトレードオフ



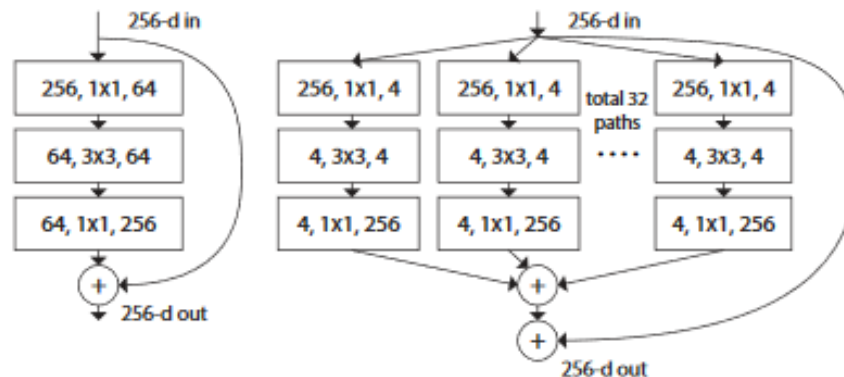
Inception ResNet
[Szegedy+, 2016]



Wide ResNet [Zagoruyko+, 2016]



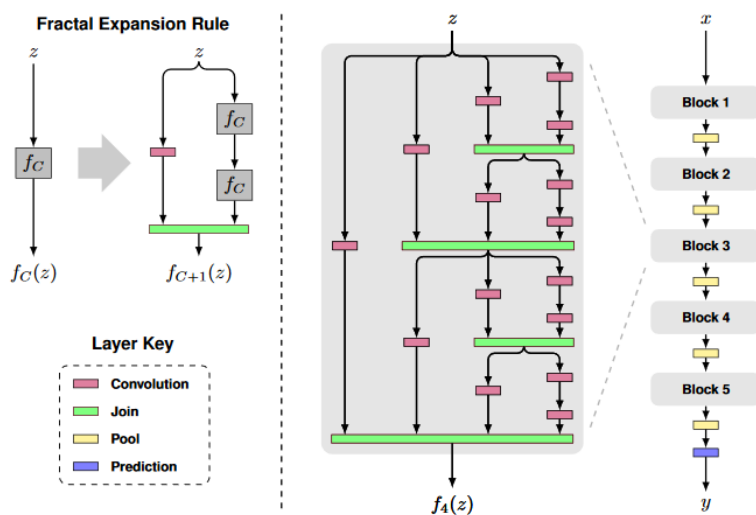
Pyramidal ResNet [Han+, 2016]



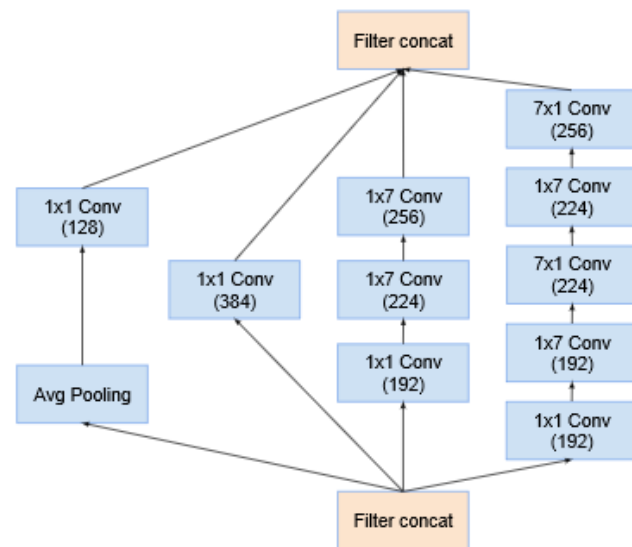
ResNeXt [Xie+, 2016]

ResNet以降 (2)

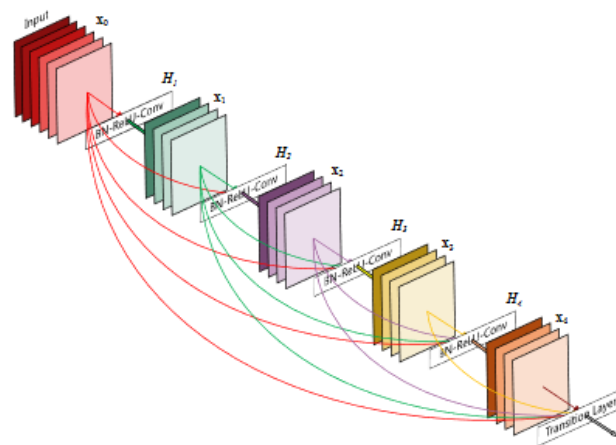
- ResNet亜種
 - Identity mappingでないskip connection
 - ネットワークつなぎ芸



FractalNet [Larsson+, 2017]



Inception-v4 [Szegedy+, 2016]



DenseNet [Huang+, 2016]

最適化手法の発達

- ミニバッチによるSGD

$$\mathbf{w} \leftarrow \mathbf{w} - \gamma \frac{\partial}{\partial \mathbf{w}} \left\{ \frac{1}{B} \sum_{i=1}^B L(\mathbf{x}_i, y_i) \right\}$$

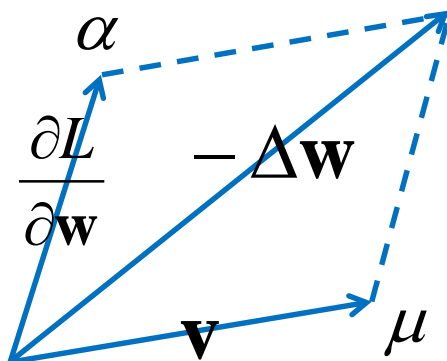
- バッチ内のデータの評価は並列化可能
 - 一般にSGDの並列化は難しいが、GPUの実装法まで含めて研究が進められている
Coates et al., “Deep learning with COTS HPC systems”, ICML’13
- 深層学習における損失関数は、しばしば鞍点(saddle point)やプラトー(plateau)が問題となる
 - どうやってそのような場所を抜けるか？

SGD + momentum

- 設定すべきハイパーパラメータ
 - 学習率、モメンタム、重み減衰率

$$\mathbf{v} \leftarrow \mu \mathbf{v} + \alpha \frac{\partial L}{\partial \mathbf{w}}$$

$$\mathbf{w} \leftarrow \mathbf{w} - \mathbf{v}$$



調整必須

```
train_net "lenet_t  
base_lr: 0.01  
momentum: 0.9  
w_decay: 0.000  
n_iter: 10000
```

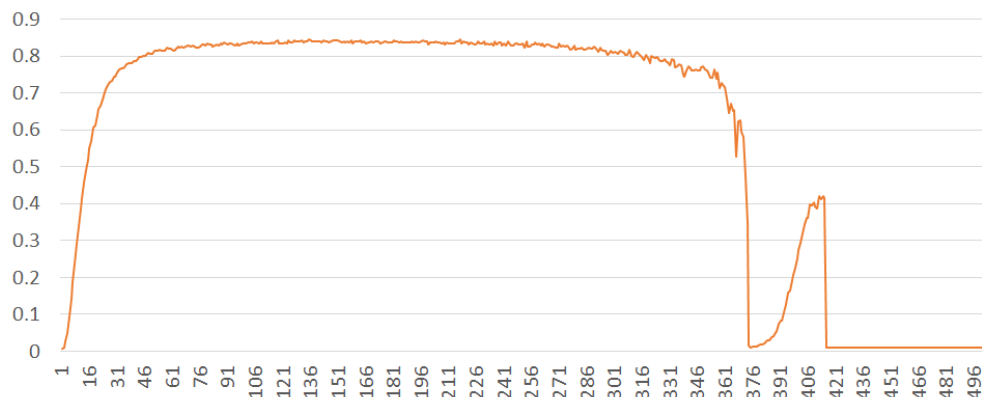
マジック
ナンバー？

学習率の設定が一番重要

- しかし一筋縄にはいかない…

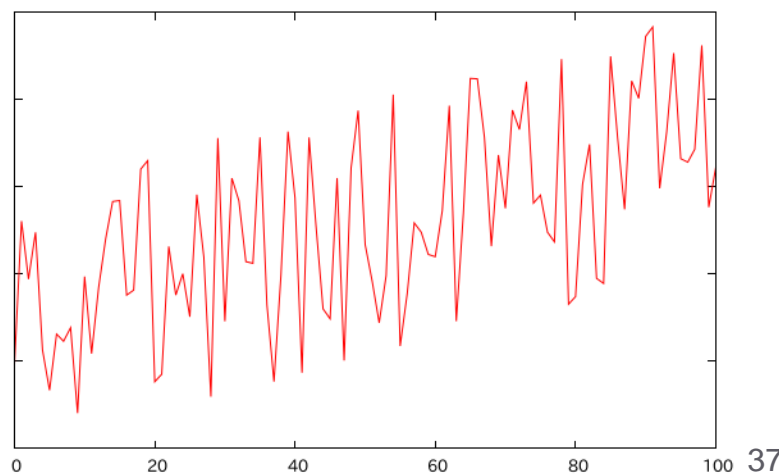
- 学習率が大きすぎる例

- すぐ頭打ちになる
- 途中で突然で破綻すること多い



- 学習率が小さすぎる例

- おおむね線形に見える場合
- 最終的にいいところまで行けるが、時間がかかりすぎる



学習率のスケジューリング

- 時間の経過 (=学習の進行) に伴い、学習率を小さくしていく操作
- 例) cuda-convnet チュートリアル (Krizhevsky)
 - 0.001 (150エポック) → 0.0001 (10エポック) → 0.00001 (10エポック)
 - 精度向上が頭打ちになったら下げしてみる？
- あるいは、単純に時間減衰させることもある
 - $1/t$, $\exp(-t)$ など

他の勾配降下手法

- 学習率を自動的に調整する手法も有効
(最適化の分野で盛んに研究されている)

- AdaGrad [Duchi+, 2011]

$$r \leftarrow r + \left| \frac{\partial L}{\partial \mathbf{w}} \right|^2 \quad \mathbf{w} \leftarrow \mathbf{w} - \frac{\alpha}{\sqrt{r + \varepsilon}} \frac{\partial L}{\partial \mathbf{w}}$$

- Adam [Kingma+, 2015]

$$\begin{aligned} r &\leftarrow \gamma r + (1 - \gamma) \left| \frac{\partial L}{\partial \mathbf{w}} \right|^2 & \mathbf{w} &\leftarrow \mathbf{w} - \frac{\alpha}{\sqrt{\frac{r}{1 - \gamma^t} + \varepsilon}} \frac{\mathbf{v}}{1 - \beta^t} \\ \mathbf{v} &\leftarrow \beta \mathbf{v} + (1 - \beta) \frac{\partial L}{\partial \mathbf{w}} \end{aligned}$$

- 他、AdaDelta、RMSProp等が有名
- 多くの問題で実用的によい結果を得るが、
しっかり学習率をスケジューリングされたSGDの方が優れているとされる

自然言語処理 (Natural language processing)

- 人間が使うような一般的な言語をコンピュータに処理させる（理解させる）ための技術の総称
- 言語： ある意味を持つシンボル（単語）の系列

government of the people, by the people, for the people

- 1. 単語をどう表現するか？
- 2. 系列（文、文章）の構造をどう表現するか？

伝統的には…

- 単語 → one-of-K (one-hot) 表現

government of the people, by the people, for the people

<i>governemnt</i>	$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \vdots \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \end{pmatrix}$
<i>of</i>										
<i>the</i>										
<i>people</i>										
<i>by</i>										
<i>for</i>										
\vdots										




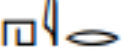

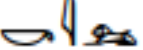

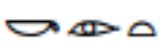




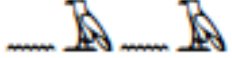
- 構造

- Bag of words (平均ベクトル)
- N-gram (連続するN語をカップリング)
- Markov model …など

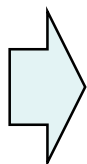
単語の表現とはどうあるべきだろうか？

A thought experiment: deciphering hieroglyphs

文章中での単語の共起を表した表

							
(knife)		51	20	84	0	3	0
(cat)		52	58	4	4	6	26
???		115	83	10	42	33	17
(boat)		59	39	23	4	0	0
(cup)		98	14	6	2	1	0
(pig)		12	17	3	2	9	27
(banana)		11	2	2	0	18	0






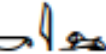







「??」の単語は何？



単語の表現とはどうあるべきだろうか？

- (cat) との類似度：共起語ベクトルの内積

Stefen Evert, “Distributional Semantic Models”,
NAACL 2010 Tutorial.








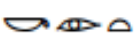


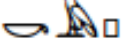
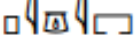

							
(knife)		51	20	84	0	3	0
(cat)		52	58	4	4	6	26
???		115	83	10	42	33	17
(boat)		59	39	23	4	0	0
(cup)		98	14	6	2	1	0
(pig)		12	17	3	2	9	27
(banana)		11	2	2	0	18	0

$$\text{sim}(\text{fish}, \text{cat}) = 0.961$$

単語の表現とはどうあるべきだろうか？

- (knife) との類似度




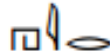

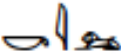

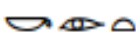


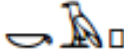
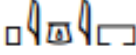
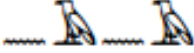
Stefen Evert, “Distributional Semantic Models”,
NAACL 2010 Tutorial.

							
(knife)		51	20	84	0	3	0
(cat)		52	58	4	4	6	26
???		115	83	10	42	33	17
(boat)		59	39	23	4	0	0
(cup)		98	14	6	2	1	0
(pig)		12	17	3	2	9	27
(banana)		11	2	2	0	18	0

$$\text{sim}(\text{cat icon}, \text{knife icon}) = 0.770$$

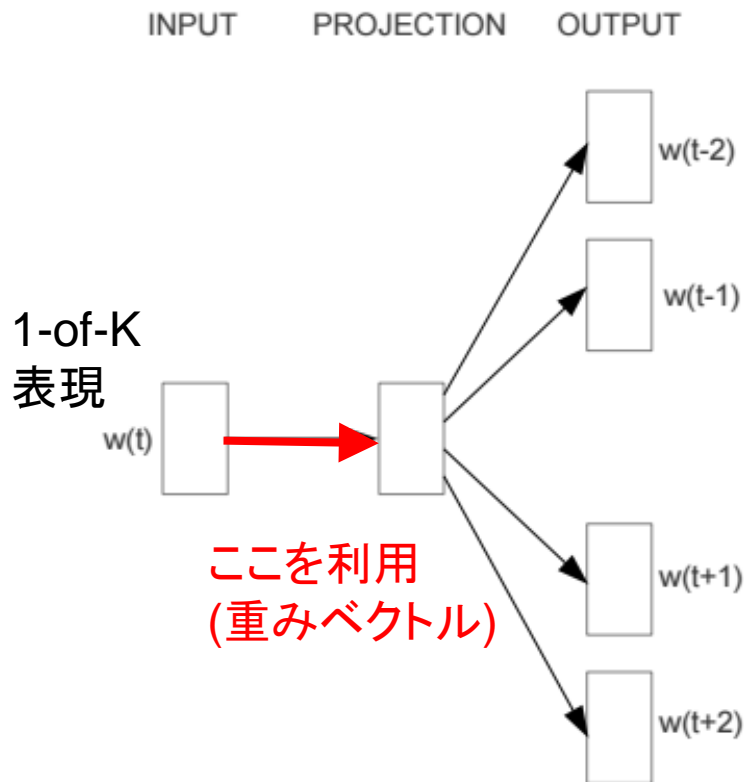
単語の表現とはどうあるべきだろうか？

Stefen Evert, “Distributional Semantic Models”,
NAACL 2010 Tutorial.

		get	see	use	hear	eat	kill
							
knife		51	20	84	0	3	0
cat		52	58	4	4	6	26
dog		115	83	10	42	33	17
boat		59	39	23	4	0	0
cup		98	14	6	2	1	0
pig		12	17	3	2	9	27
banana		11	2	2	0	18	0

おなじ文脈(コンテキスト)で表れる単語は近い意味を持つ(傾向がある)
→ ある単語と周辺の単語群の相関を捉えればよい(という仮説)

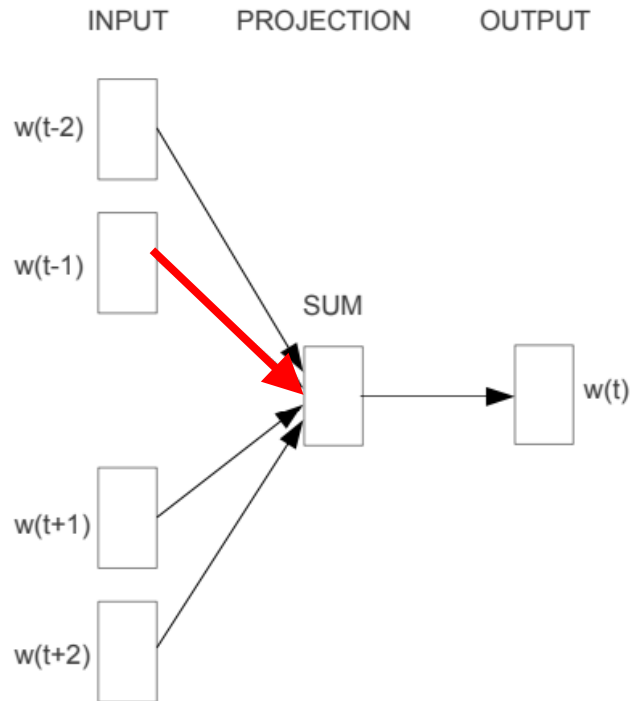
Skip-gram (word2vec) [Mikolov+, 2013]



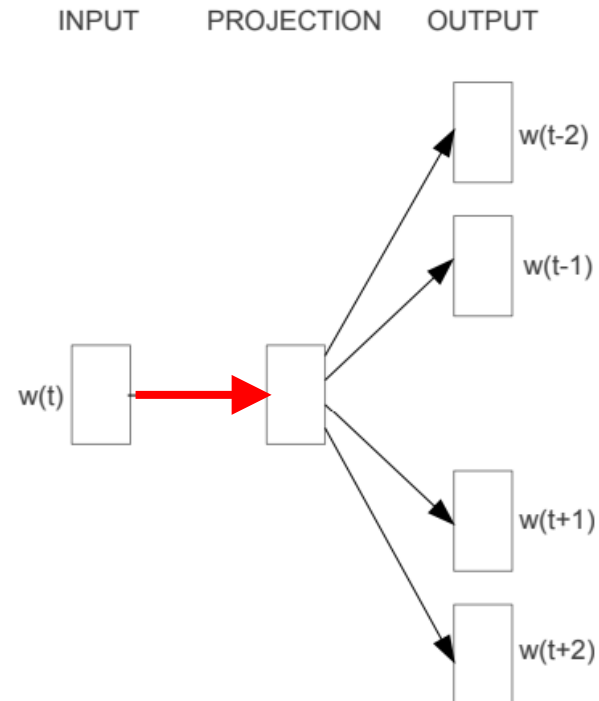
- ある単語から周辺の単語を予測するパーセプトロン
 - 別にディープではないが...
- 密で(比較的)低次元な埋め込みベクトルを獲得
- さまざまなNLPの性能を飛躍的に向上させた

Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space", ICLR 2013

Skip-gram と CBoW



CBoW



Skip-gram

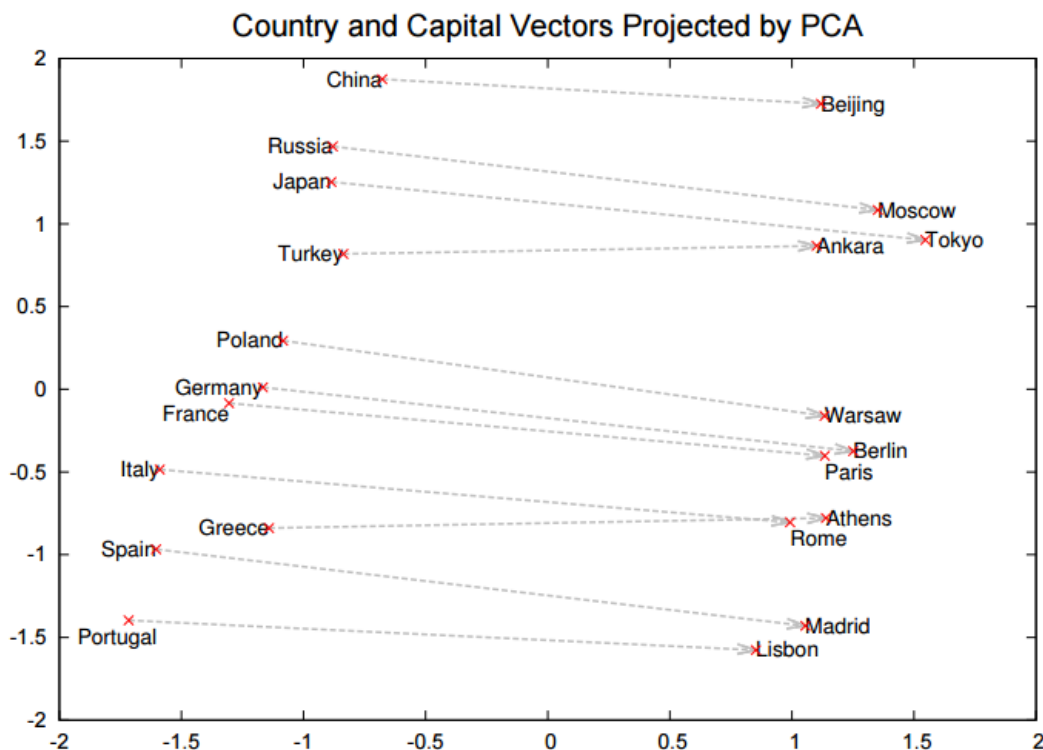
Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space", ICLR 2013

何がすごい？

Mikolov et al., “Distributed Representations of Words and Phrases and their Compositionality”, In Proc. of NIPS, 2014.

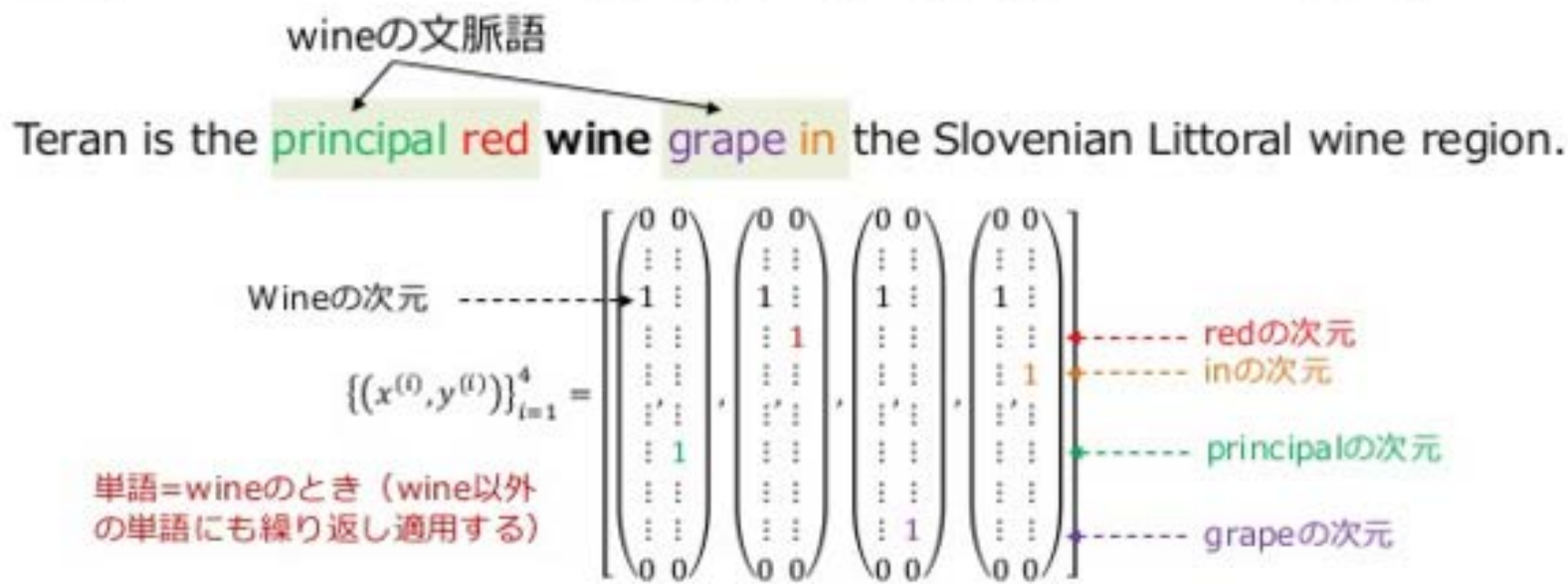
- 埋め込んだ空間は**線形**に意味的な構造を備えている
 - e.g. 国名→首都のベクトルが同じ方向・距離
 - これにより、 $\text{France} + (\text{Tokyo} - \text{Japan}) \div \text{Paris}$ のような演算が可能

※同様のアイデアは
実は古くから存在
(共起行列の分解)



正準相関分析による学習 (Stratos+ 2015)

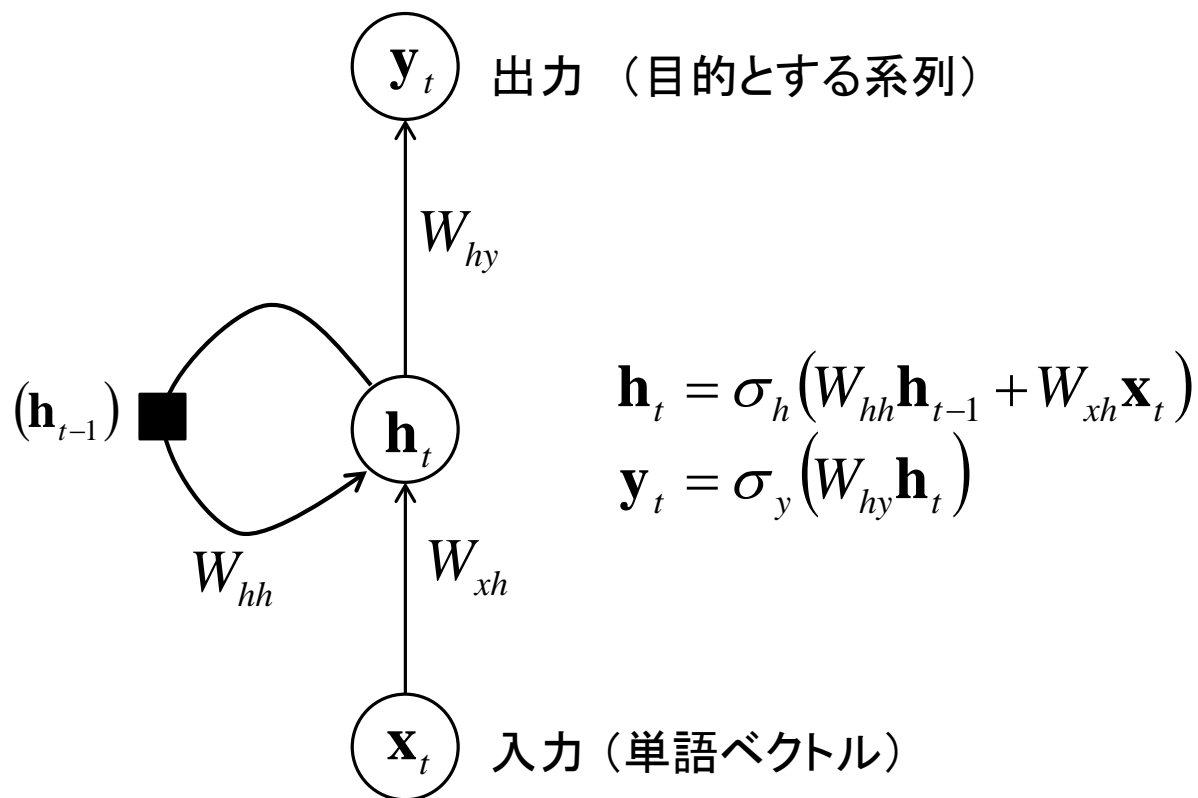
- 確率変数（ベクトル） X, Y を次のように定義
 - $X \in \mathbb{R}^n$: 単語の出現を表すone-hotベクトル
 - $Y \in \mathbb{R}^{n'}$: 文脈の出現を表すone-hotベクトル
- X, Y のサンプルの作成例（文脈幅 $h = 2$ の場合）



K. Stratos, M. Collins and D. Hsu, "Model-based Word Embeddings from Decompositions of Count Matrices", In Proc. of ACL, 2015.

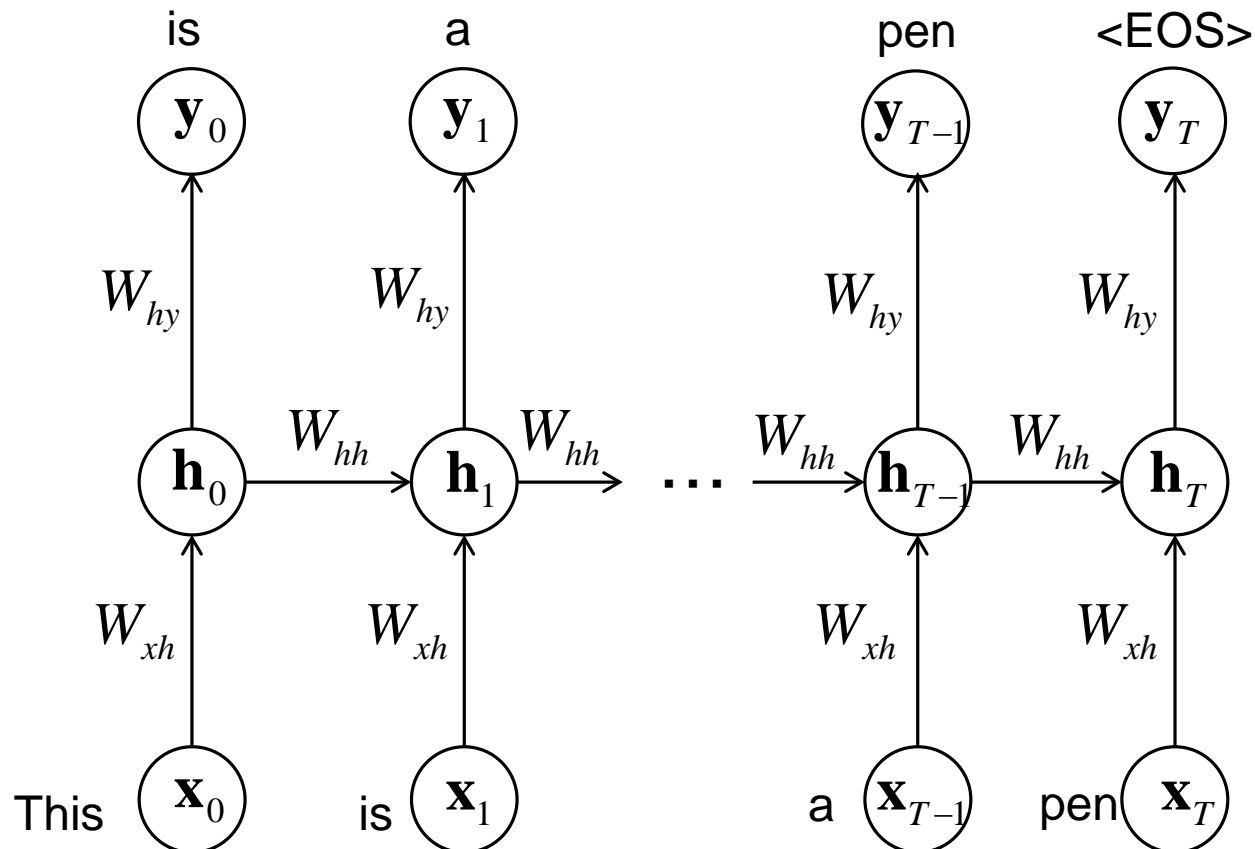
Recurrent Neural Network (RNN)

- 自分の一個前の隠れ状態を再入力するネットワーク
- 隠れ状態は、入力系列の記憶を全て保持した分散表現となる
- 理論的には、任意のタイムスケールでの入出力依存関係を表現可能



時間方向に展開してみると…

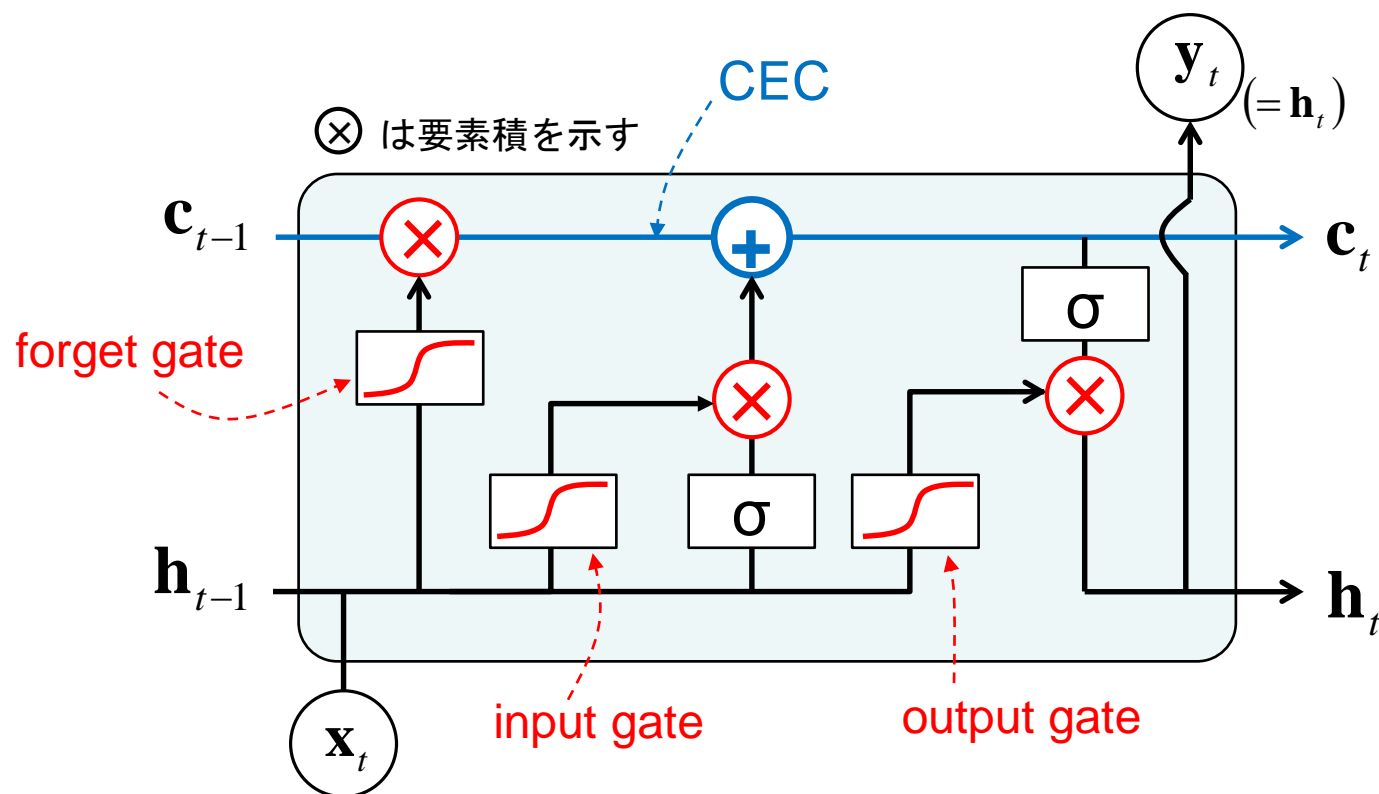
- 静的な(深い)ネットワークとして書ける
 - 普通のパーセプトロンと同様、誤差逆伝播による学習が可能
- 他の深層モデル同様、誤差消失により実際には遠い依存関係の学習が困難であったが、LSTM [Hochreiter+, 1997] により大幅な進展



Long short-term memory (LSTM)

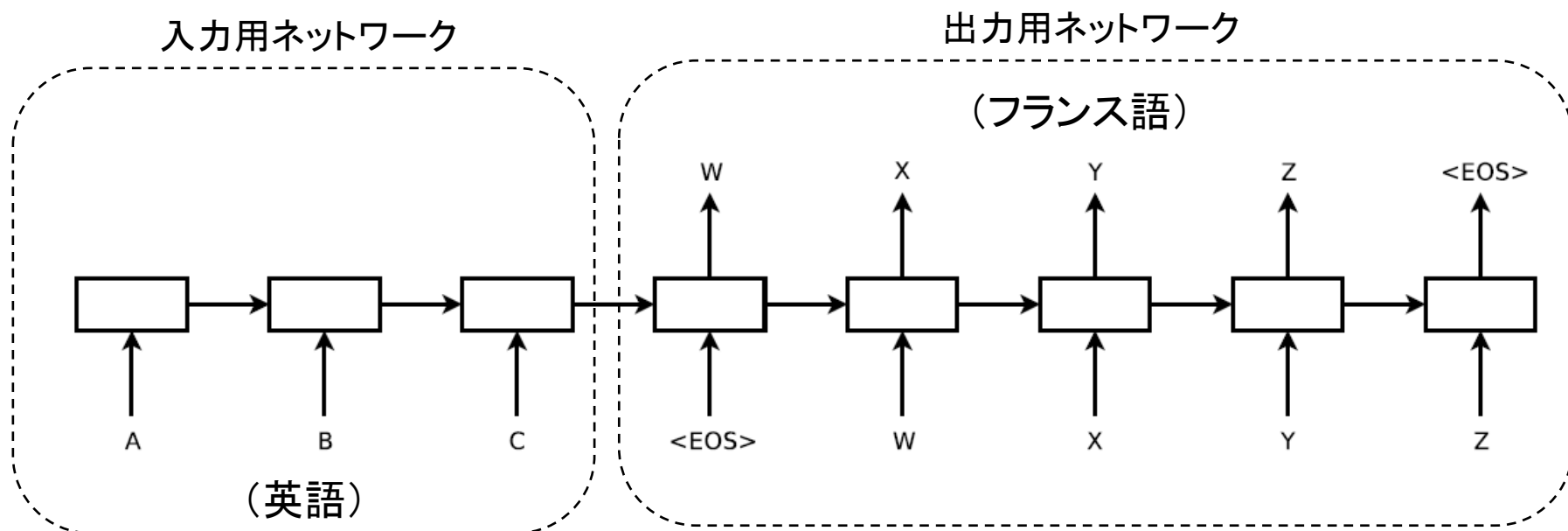
[Hochreiter+, 1997]

- エラーを記憶させる素子を明示的に用意
 - **Constant Error Carousel (CEC)**
 - 基本的には入力をどんどん足していきだけ → 入力系列の“記憶”
 - エラーはCEC内に留まり続けるので、誤差消失・爆発問題を回避できる
- ゲートにより、**選択的に記憶の追加・取り出し・消去を行う**
 - 重要な時系列パターンに対して注意を向ける効果



RNNを用いた機械翻訳

- Sequence to sequence [Sutskever+, NIPS'14]
 - 二つのRNN (LSTM) を接続し、英語・フランス語単語列の入出力関係を学習
 - 損失：出力の各ステップの cross entropy loss の平均



ちなみに...

- 最近は畳み込みネットもよくNLPに用いられる！

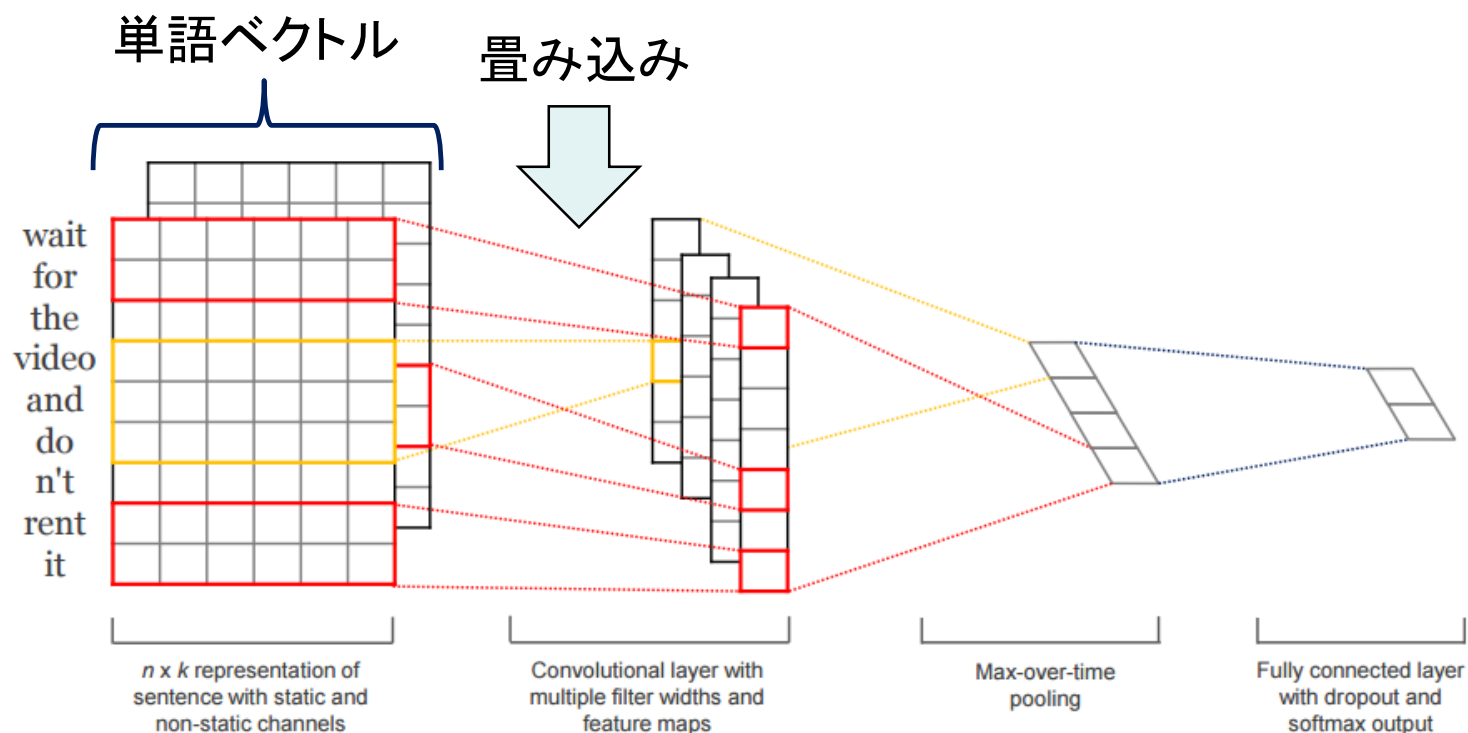
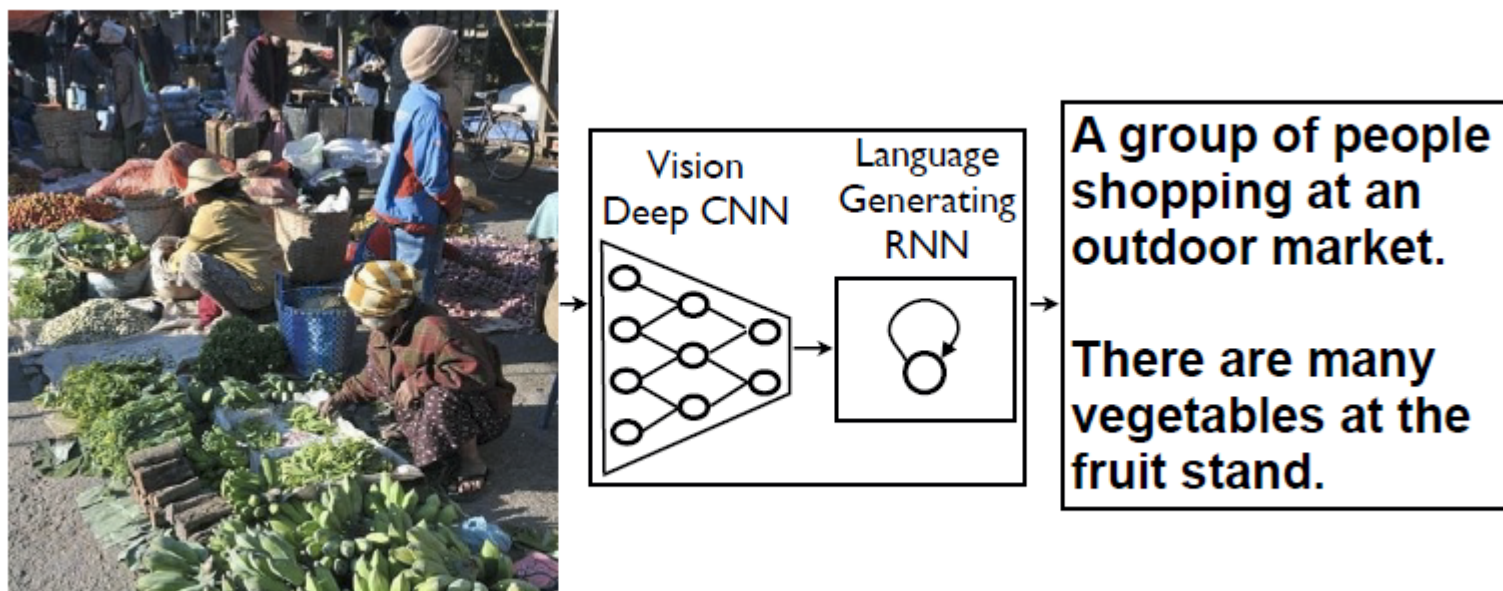


Figure 1: Model architecture with two channels for an example sentence.

画像説明文生成（≡画像から言語への翻訳）

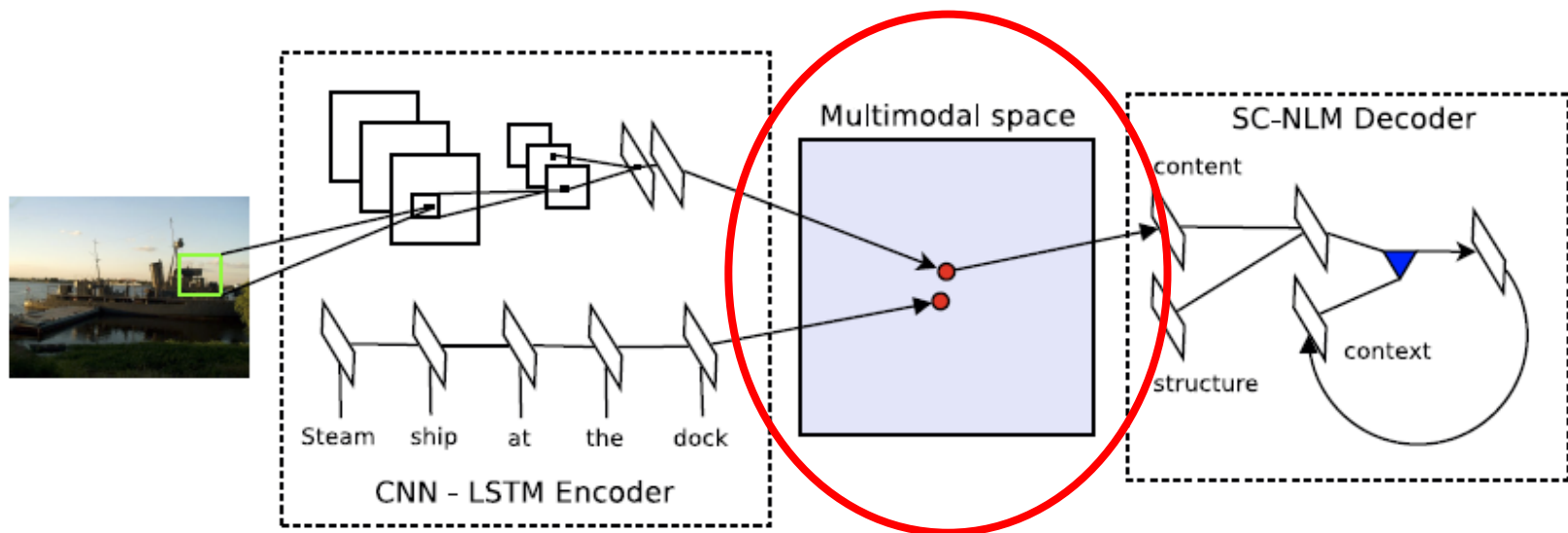
- CNN (画像側)の出力をRNN(言語側)へ接続
 - RNN側の誤差をCNN側までフィードバック



O. Vinyals et al., "Show and Tell: A Neural Image Caption Generator", In Proc. CVPR, 2015.

画像とテキストのマルチモーダル分散表現

- ▶ 共通の上位レイヤ(潜在空間)へマッピング [Kiros et al., 2014]
 - 異なるモダリティ間での“演算”が可能



R. Kiros et al., "Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models", TACL, 2015.

Nearest images



- blue + red =



- blue + yellow =



- yellow + red =



- white + red =

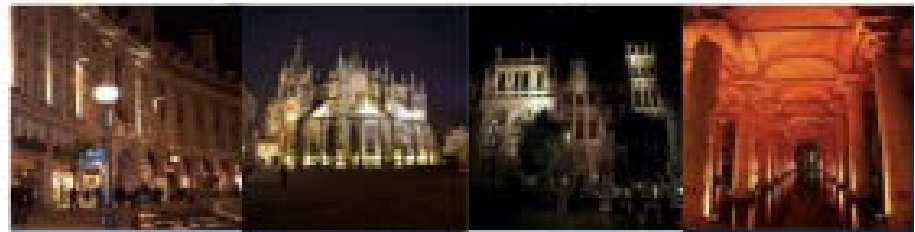


[Kiros et al., 2014]

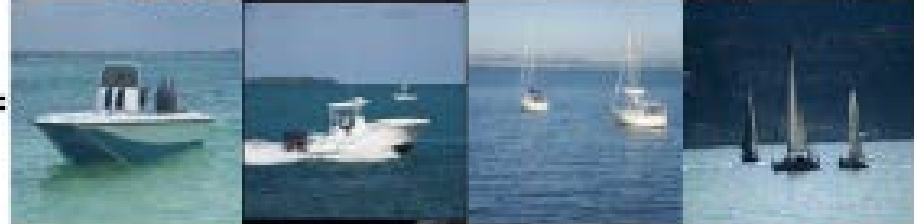
Nearest images



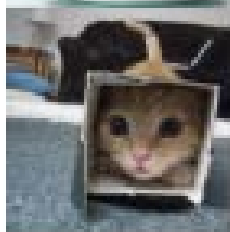
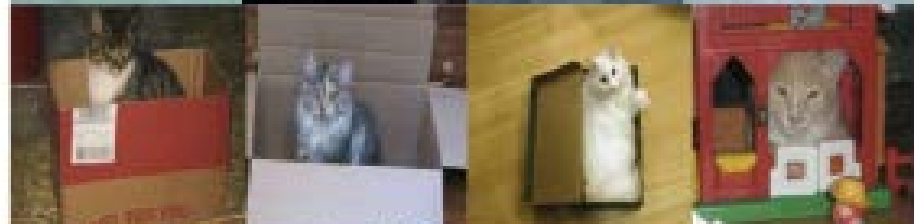
- day + night =



- flying + sailing =



- bowl + box =



- box + bowl =



[Kiros et al., 2014]




画像内容に対するQ&A

- LSTMを用いた質問入力と回答の対応関係学習

H. Gao et al., “Are You Talking to a Machine? Dataset and Methods for Multilingual Image Question Answering”, 2015.

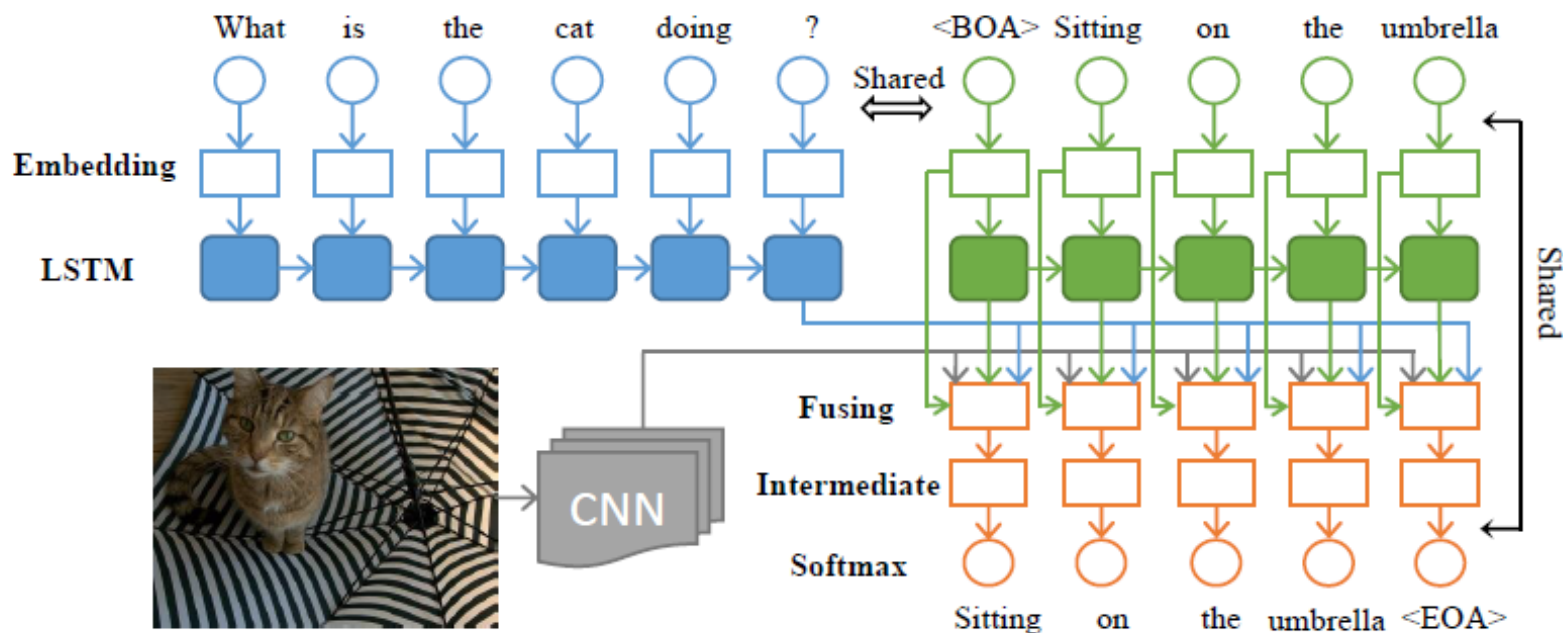
Image					
Question	公共汽车是什么颜色的? What is the color of the bus?	黄色的是什么? What is there in yellow?	草地上除了人以外还有什么动物? What is there on the grass, except the person?	猫咪在哪里? Where is the kitty?	观察一下说出食物里任意一种蔬菜的名字? Please look carefully and tell me what is the name of the vegetables in the plate?
Answer	公共汽车是红色的。 The bus is red.	香蕉。 Bananas.	羊。 Sheep.	在椅子上。 On the chair.	西兰花。 Broccoli.

M. Ren et al., “Image Question Answering: A Visual Semantic Embedding Model and a New Dataset”, 2015.

		
CQ5429: What do two women hold with a picture on it? Ground truth: cake VIS+LSTM-2: cake (0.5611) VIS+BOW: laptop (0.1443) LSTM: umbrellas (0.1567) BOW: phones (0.1447)	CQ24952: What is the black and white cat wearing? Ground truth: hat VIS+LSTM-2: hat (0.6349) LSTM: tie (0.5821)	CQ25218: Where are the ripe bananas sitting? Ground truth: basket VIS+LSTM-2: basket (0.4965) LSTM: bowl (0.6415)
		CQ25218a: What are in the basket? Ground truth: bananas VIS+LSTM-2: bananas (0.6443) LSTM: bears (0.0956)

画像内容に対するQ&A

- NNを使った機械翻訳モデルの応用
- 質問文に加え、CNN対象画像の特徴抽出を行い、回答文生成のRNNへ入力



H. Gao et al., "Are You Talking to a Machine? Dataset and Methods for Multilingual Image Question Answering", 2015.

その他深層学習に関する話題（ごく一部）

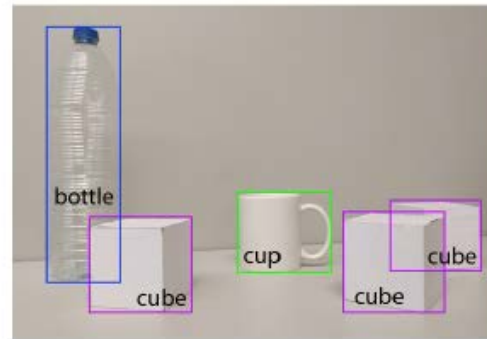
- 高度な画像認識タスクの実現
- 強化学習との融合
- 深層学習による生成モデル
- 敵対的入力

さらに高度な画像認識

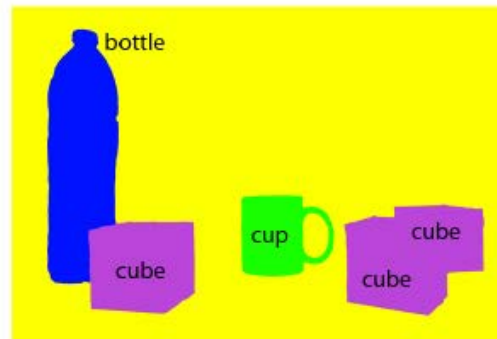
- 物体検出、セグメンテーション



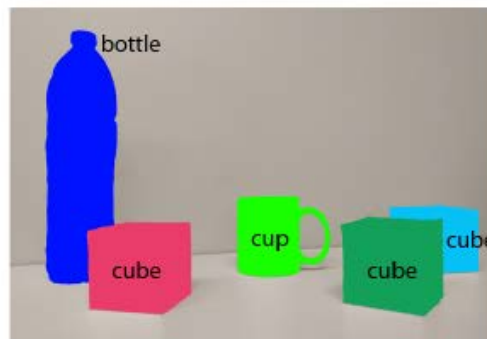
(a) Image classification



(b) Object localization



(c) Semantic segmentation



(d) Instance segmentation

[Garcia et al., 2017]

Mask-RCNN [He et al., ICCV'17]

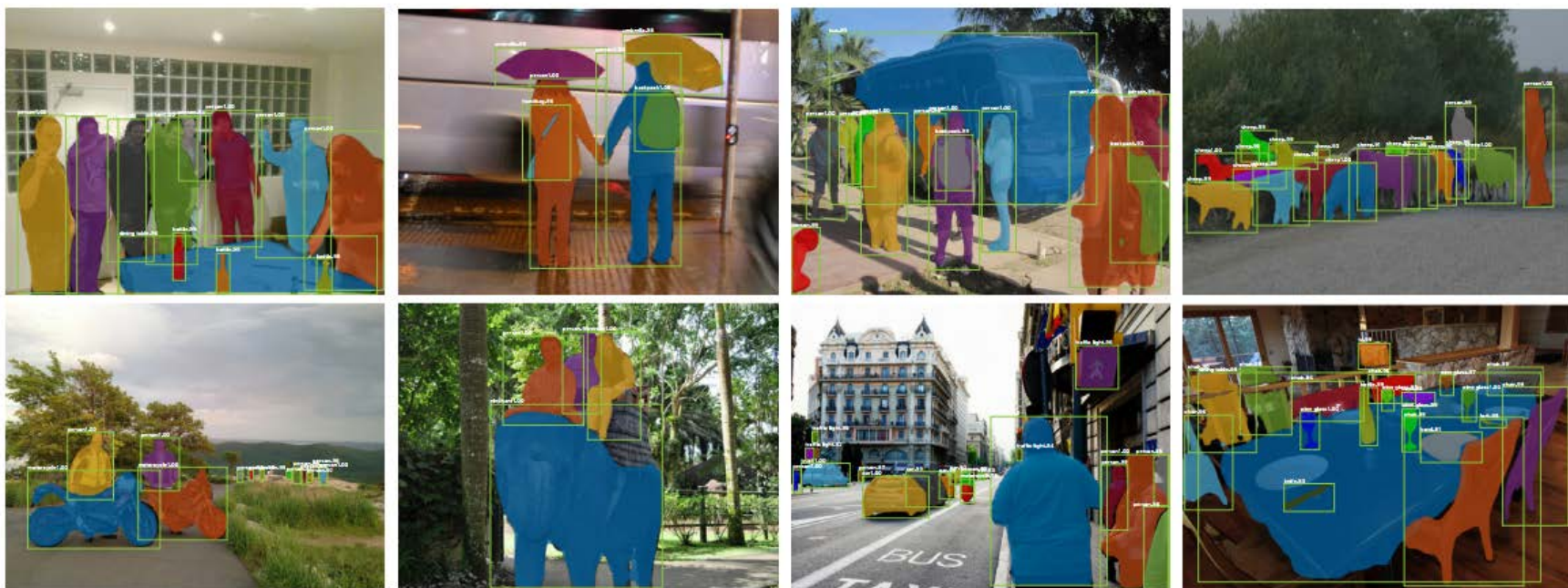
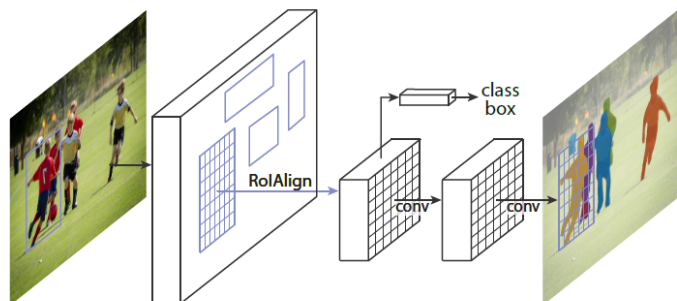
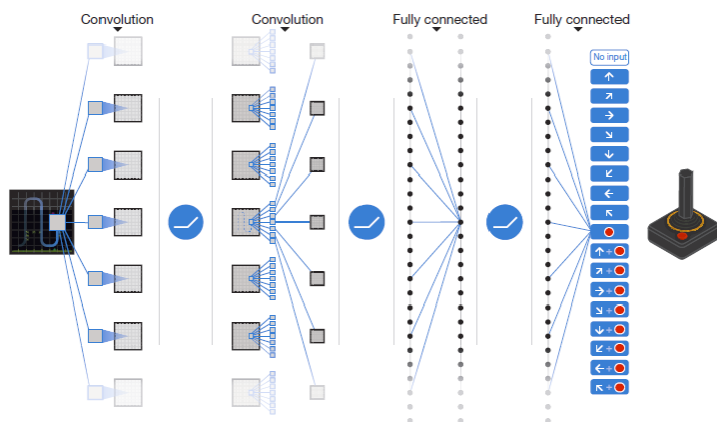


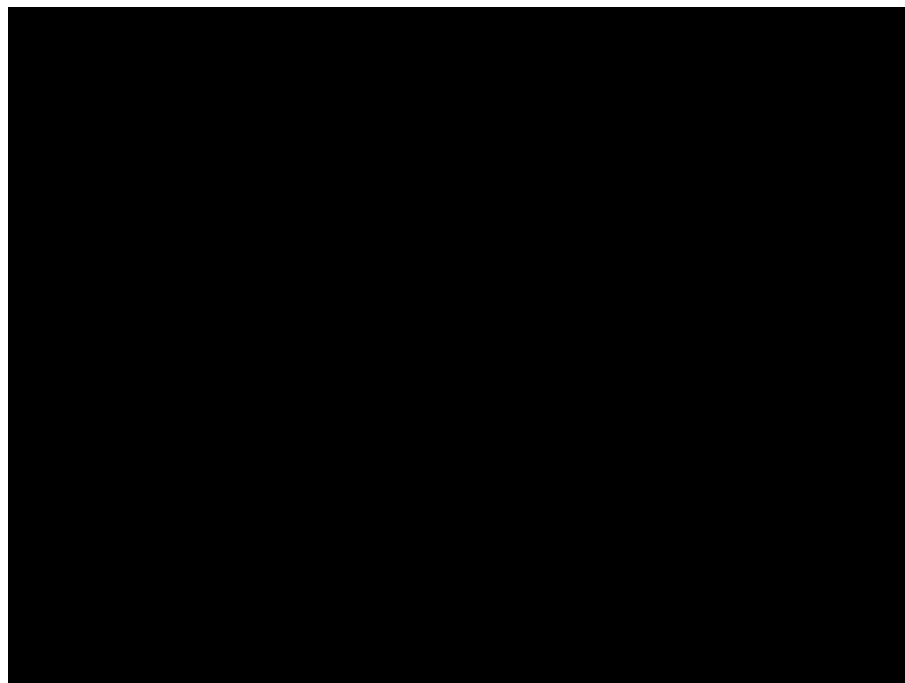
Figure 2. **Mask R-CNN** results on the COCO test set. These results are based on ResNet-101 [19], achieving a *mask* AP of 35.7 and running at 5 fps. Masks are shown in color, and bounding box, category, and confidences are also shown.

強化学習との融合（深層強化学習）

- Deep Q-learning [Mnih et al, NIPS'13, Nature'15]
 - DeepMind（Googleに買収されたベンチャー）の発表
 - 強化学習の報酬系に畳み込みネットワークを接続（生画像を入力）
 - アタリのクラシックゲームで人間を超える腕前



Mnih et al., "Human-Level Control Through Deep Reinforcement Learning", Nature, 518(7540):529-533, 2015.

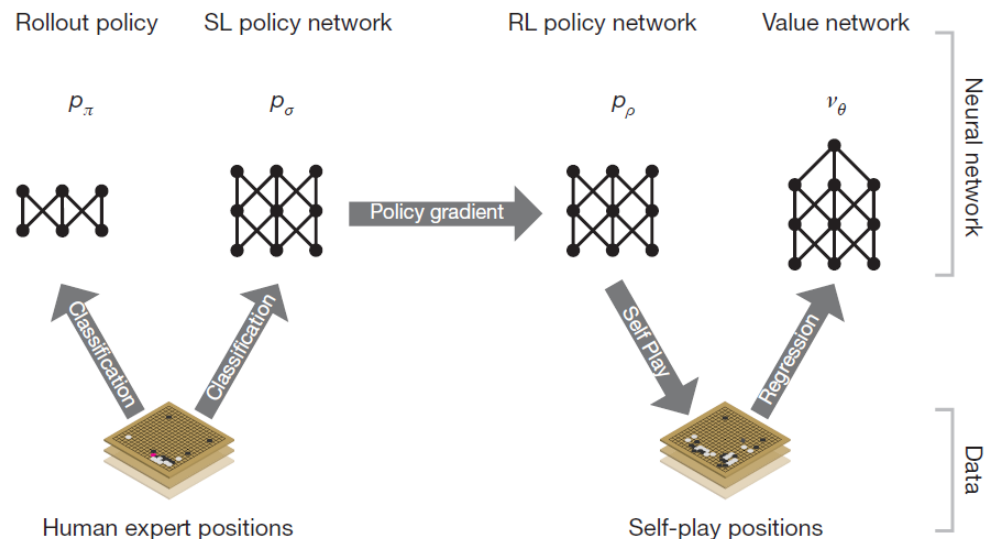


AlphaGo

- Google (DeepMind) のAI囲碁プログラミングが
韓国のトップ棋士に勝利 (2016年3月)
 - 最初は人間の棋譜を教師とするが、最後は自己対局を大量に行い学習する
※最新版(AlphaGoZero)では、ゼロから自己対局のみで更に強くなっている
 - 入力部に畳み込みニューラルネットワークを利用



<http://japan.cnet.com/news/service/35079593/>



D. Silver et al., "Mastering the Game of Go with Deep Neural Networks and Tree Search," Nature, Vol. 529, No. 7587, pp.484-489, 2016.

D. Silver et al., "Mastering the game of Go without human knowledge," Nature, Vol. 550, pp. 354–359, 2017.

深層学習による生成モデル

- Deep Boltzmann Machine
- Deep Belief Network
- Autoencoder
- Variational autoencoder (VAE)

- Generative Adversarial Network (GAN)
 - [Goodfellow+, 2014]
 - データ生成を行うネットワークと、生成されたデータと本物のデータを識別するネットワークを競わせるように学習

文章からの画像生成

- [Zhang et al, ICCV'17]

This bird has a yellow belly and tarsus, grey back, wings, and brown throat, nape with a black face



This bird is white with some black on its head and wings, and has a long orange beak



This flower has overlapping pink pointed petals surrounding a ring of short yellow filaments



Zhang et al., "StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks", In Proc. of IEEE ICCV, 2017.

画像スタイル変換

- Image-to-image translation
[Isola et al., 2016]



Isola et al., “Image-to-Image Translation with Conditional Adversarial Networks”, In Proc. IEEE CVPR, 2017.

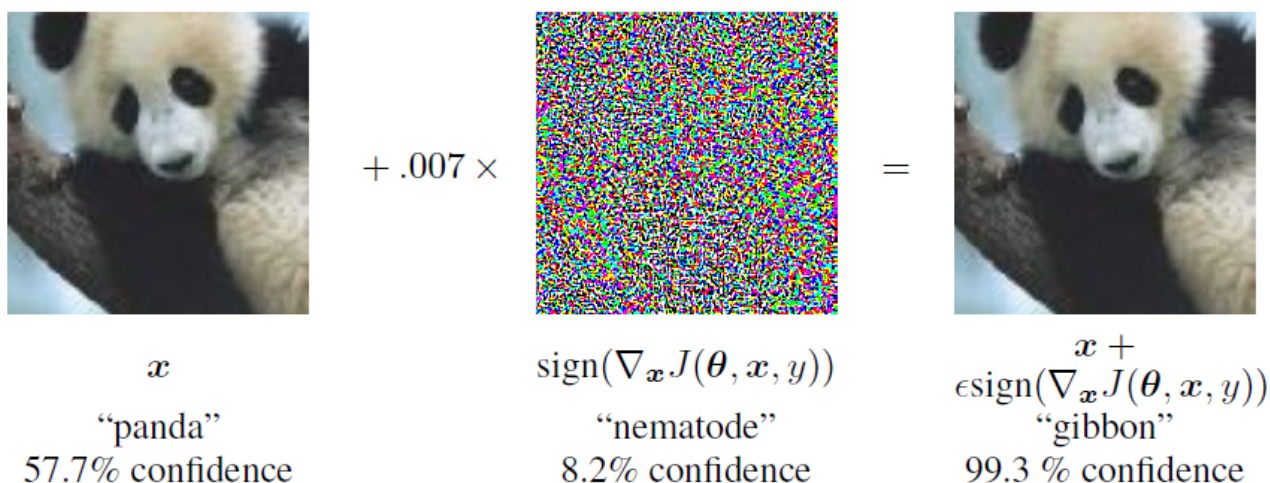
- Cycle GAN
[Zhu et al., 2017]



Zhu et al., “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”, In Proc. IEEE ICCV, 2017.

敵対的入力 (Adversarial Example)

- DNNは敵対的な入力により容易に騙される
 - 入力をワーストケースの方向へ少し変化させる

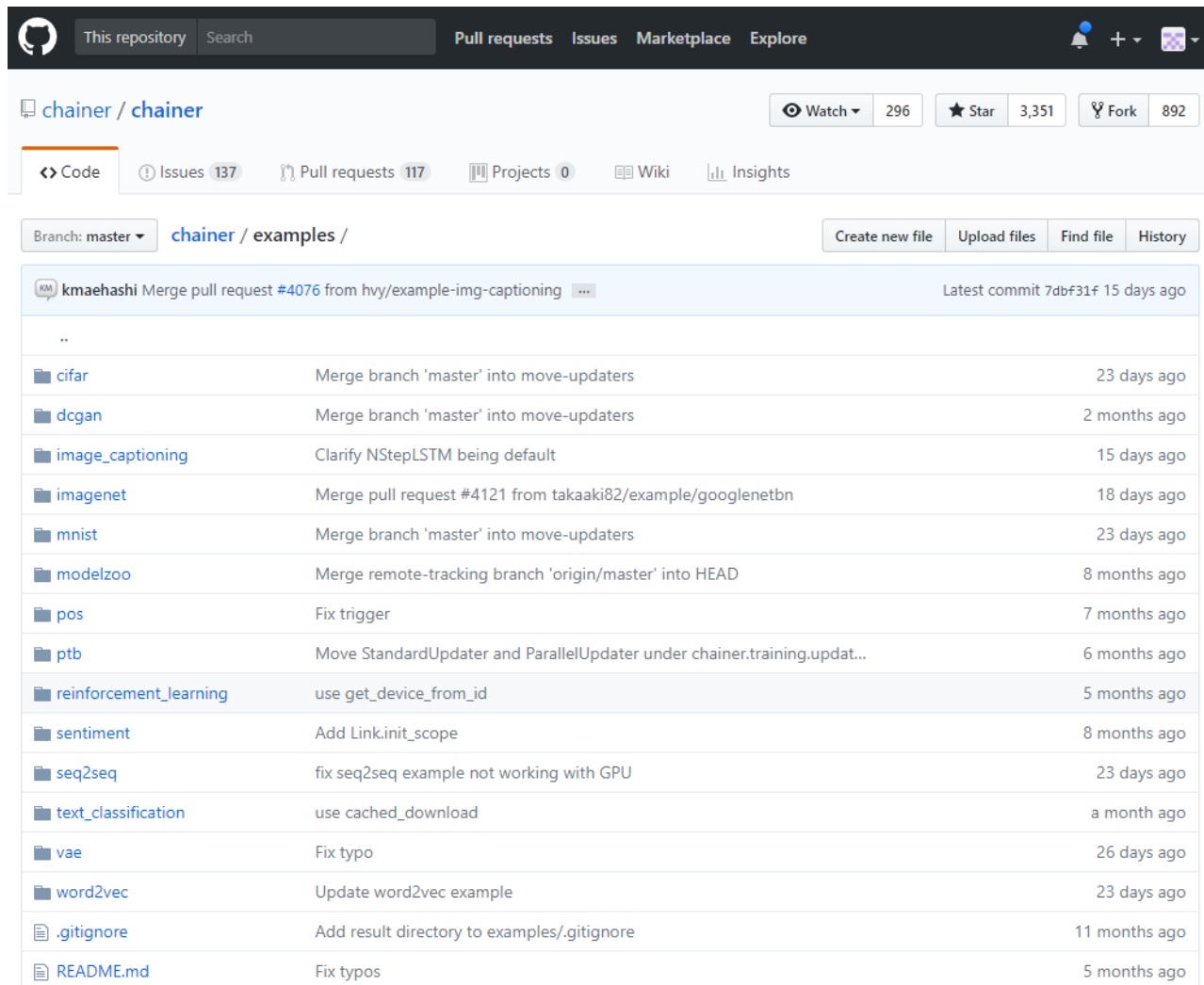


Goodfellow et al., “Explaining and harnessing adversarial examples”, In Proc. of ICLR, 2015.

- ▶ NO Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles [Lu et al., 2017/7/12]
- ▶ Synthesizing Robust Adversarial Example [Athalye et al., 2017/7/24]

Chainer 実装集@github (official)

- <https://github.com/chainer/chainer/tree/master/examples>



The screenshot shows the GitHub repository page for `chainer/chainer`. The repository has 296 watches, 3,351 stars, and 892 forks. The `examples` directory is selected, showing a list of subdirectories and their commit history. The latest commit is by `kmaehashi` merging pull request #4076 from `hvy/example-img-captioning`, dated 15 days ago.

Directory	Commit Message	Time Ago
..	..	
cifar	Merge branch 'master' into move-updaters	23 days ago
dcgan	Merge branch 'master' into move-updaters	2 months ago
image_captioning	Clarify NStepLSTM being default	15 days ago
imagenet	Merge pull request #4121 from takaaki82/example/googlenetbn	18 days ago
mnist	Merge branch 'master' into move-updaters	23 days ago
modelzoo	Merge remote-tracking branch 'origin/master' into HEAD	8 months ago
pos	Fix trigger	7 months ago
ptb	Move StandardUpdater and ParallelUpdater under chainer.training.updat...	6 months ago
reinforcement_learning	use get_device_from_id	5 months ago
sentiment	Add Link.init_scope	8 months ago
seq2seq	fix seq2seq example not working with GPU	23 days ago
text_classification	use cached_download	a month ago
vae	Fix typo	26 days ago
word2vec	Update word2vec example	23 days ago
.gitignore	Add result directory to examples/.gitignore	11 months ago
README.md	Fix typos	5 months ago

データサイエンス

~まとめ~

やってきたこと

- Python、R
- 統計基礎： 検定・推定
- 多変量解析・次元削減： PCA, FLDA, CCA等
- 回帰分析： 線形回帰、スパース線形回帰、一般化線形モデル
- クラスタリング： 階層的手法、k-means、EMアルゴリズム
- クラス分類(生成的)： k最近傍識別、ナイーブベイズ識別
- クラス分類(識別的)： 最小二乗識別、ロジスティック回帰、SVM
- アンサンブル学習： バギング、ブースティング、ランダムフォレスト
- 線形手法の非線形拡張： カーネル法、explicit embedding
- 深層学習： ニューラルネット、畳み込みネット、RNN

データサイエンス：私が好きな定義

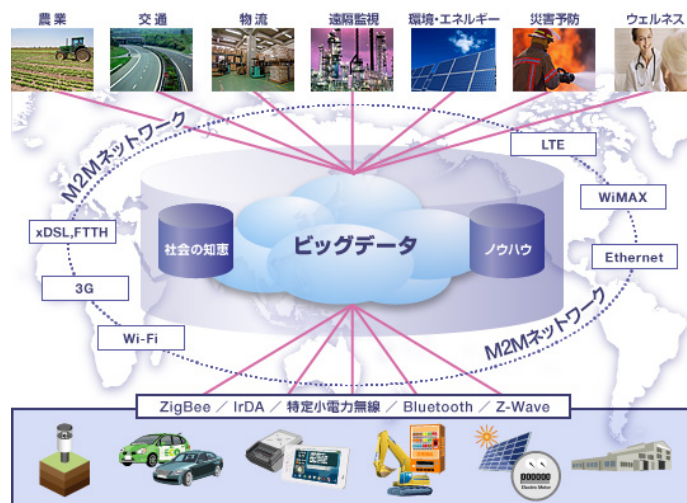
- データを収集し、分析に適した形に整え、**データにストーリーを語らせ、そのストーリーを他者に伝える**
 - Loukides, 「What is data science?」, 2010.
- ビジネス、システムの現状を理解した上で、データをもとに、**そもそも何をしたらよいのか**を提示する
- データ“**サイエンス**”と言われる所以

データマイニングのプロセス

- CRISP-DM (CRoss-Industry Standard Process for Data Mining)
 - Business understanding
 - Data understanding
 - Data preparation
 - Modeling
 - Evaluation
 - Deployment
- 実際はここでつまづくことが多い...
(研究、実務が本番)
- アカデミアでは(主に)ここを極める
(講義ではやはりここが中心)

最後に

- いろいろなチャンスがある面白い時代
 - 実世界と情報世界が密に結合
 - M2M, IoT, ロボティクス, ...
 - 第三次人工知能ブーム
 - 専門家として、本質は何かを常に冷静に考えること



http://jpn.nec.com/cloud/service/saas_common/m2m.html

- 躍らされず、ただ斜に構えず、上手にビッグデータ時代を楽しんでください！