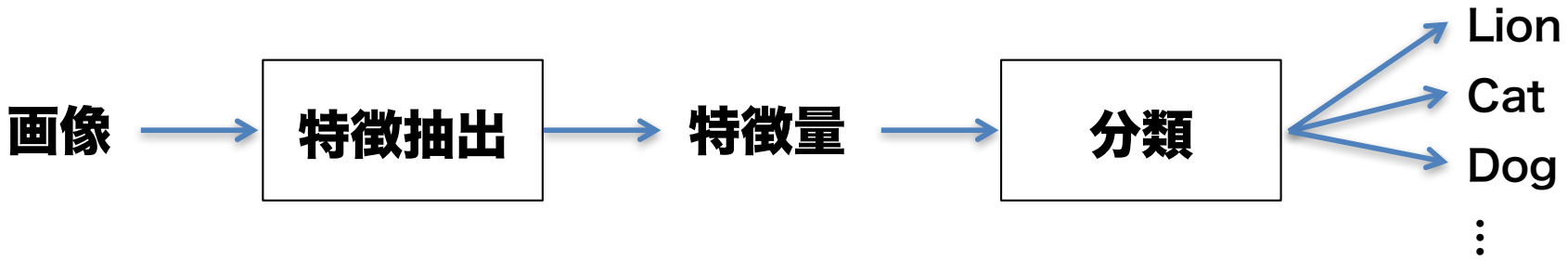
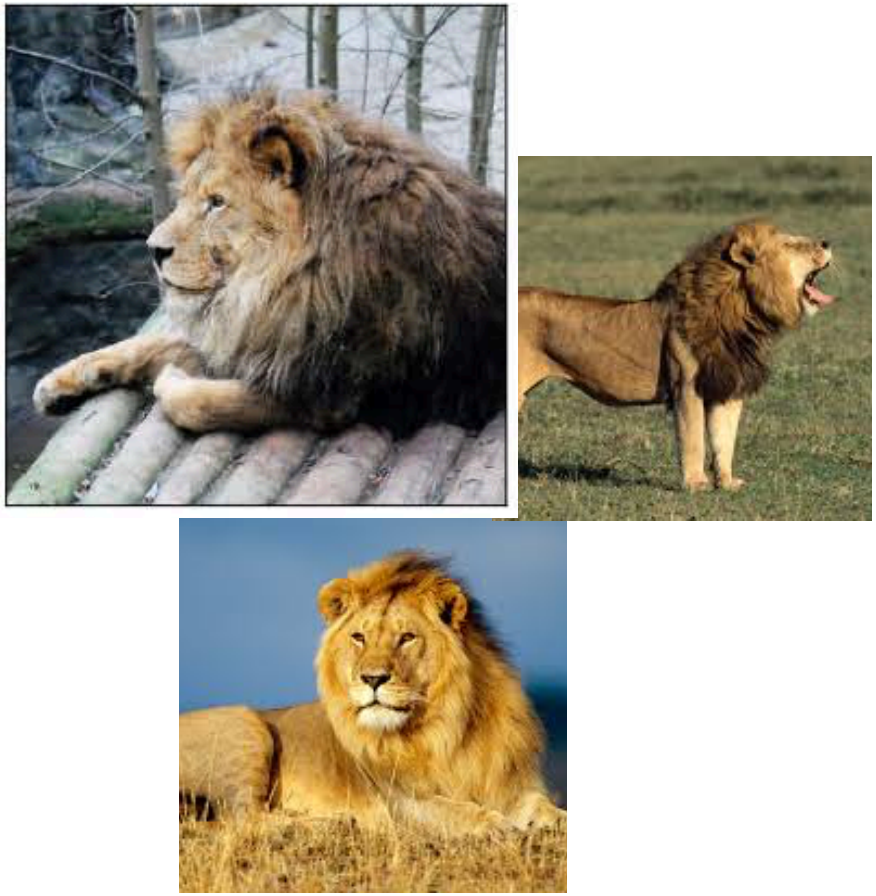
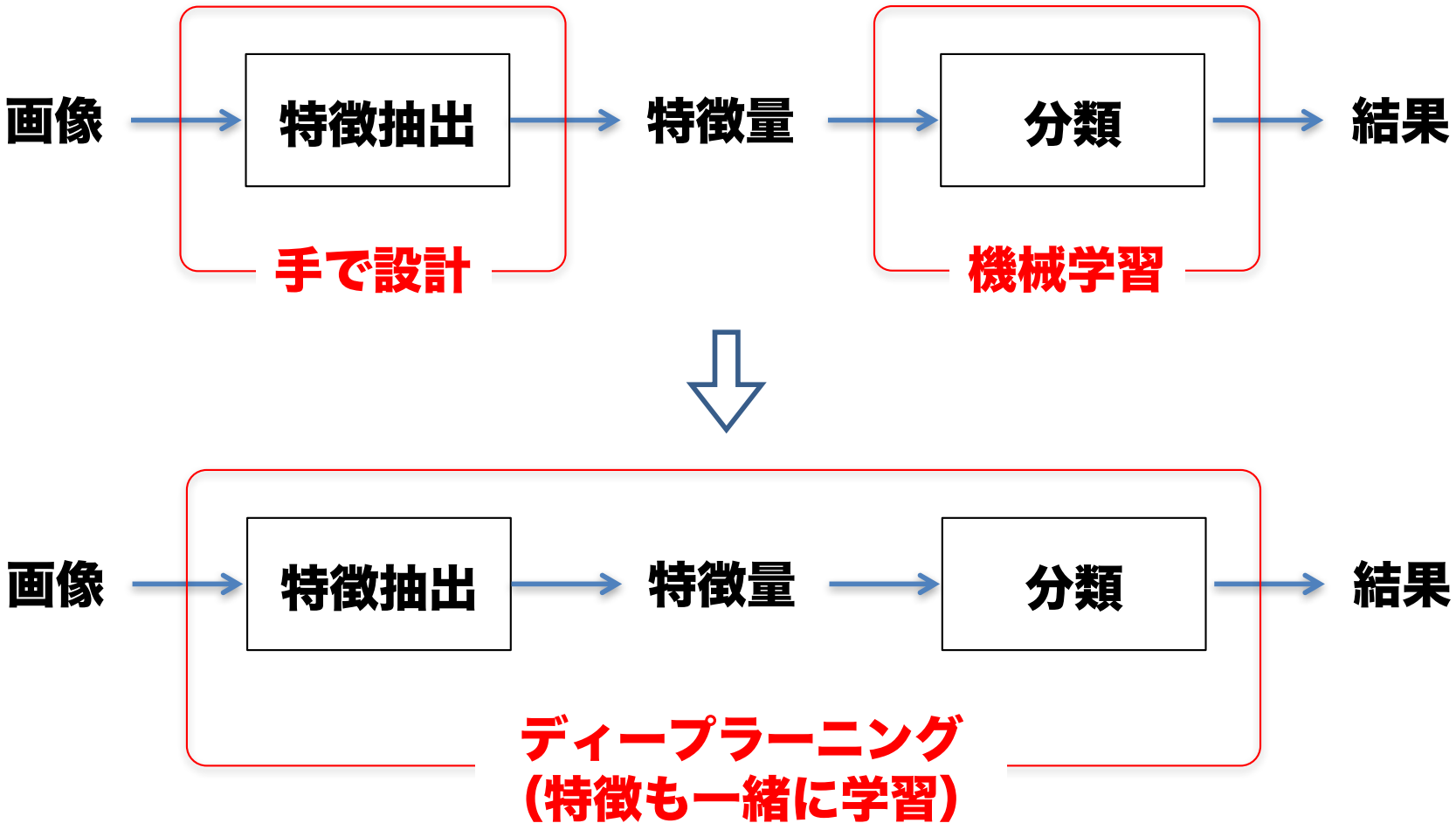
コンピュータビジョン
画像認識-II

問題の難しさ

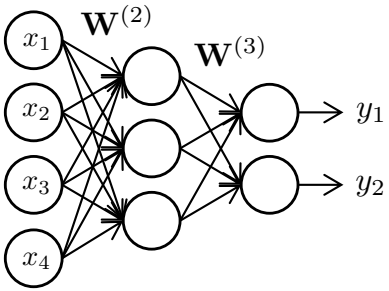




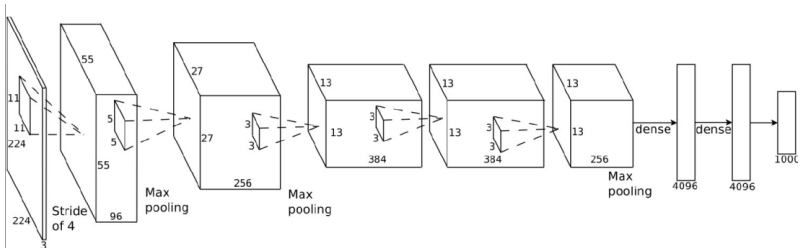
特徴の設計から特徴の学習へ



様々なニューラルネットワーク



①②フィードフォワードネット



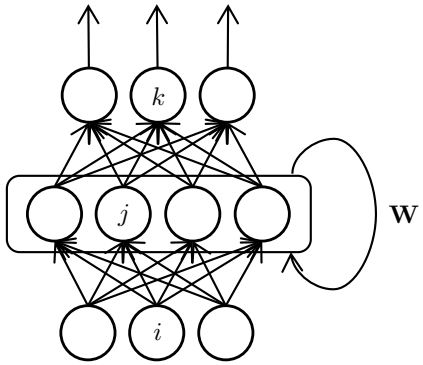
④たたみこみネット(CNN)

教師あり

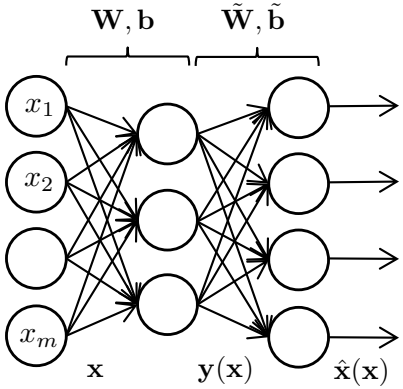
静的

教師なし

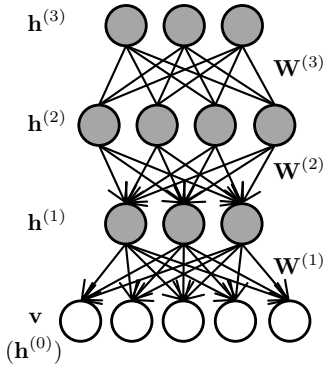
動的



⑤リカレントネット
(RNN)



③オートエンコーダ
(自己符号化器)



⑥ボルツマンマシン

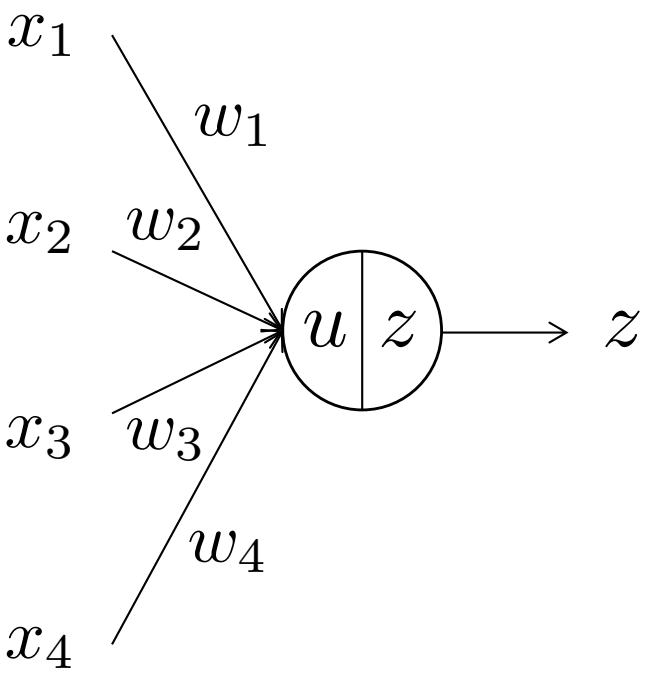
決定論的

確率的

ユニットの働きと活性化関数

$$u = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + b$$

$$z = f(u)$$



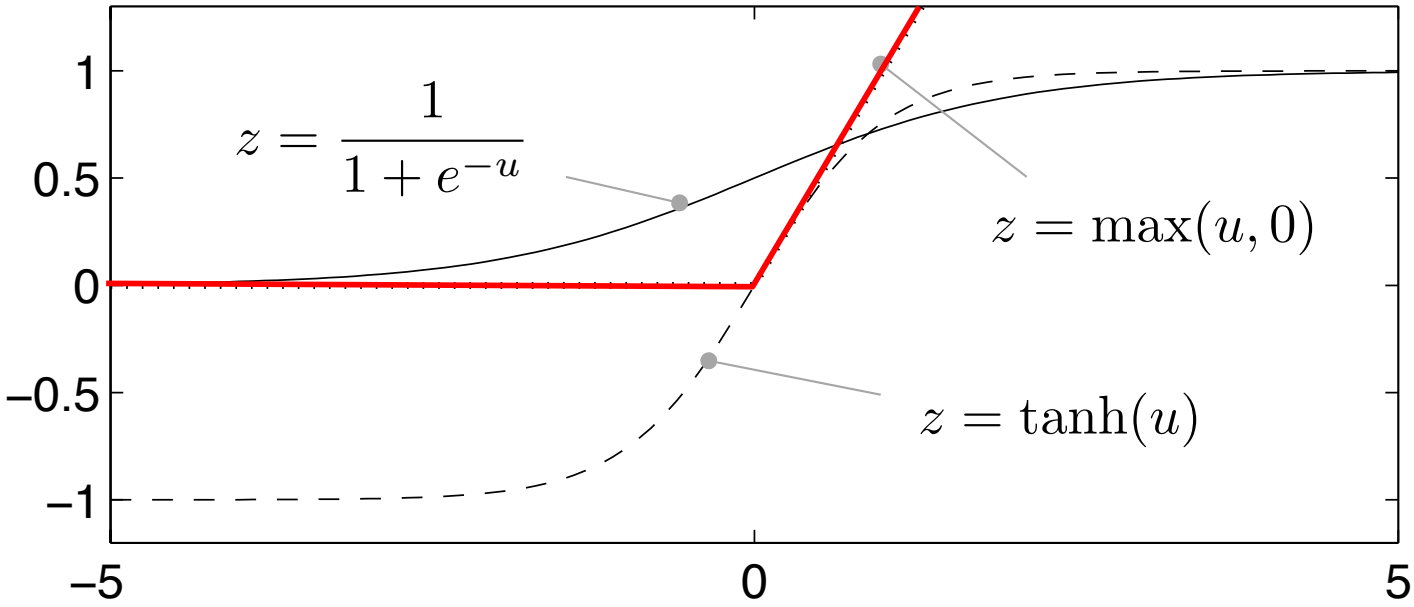
ユニットの働きと活性化関数

$$u = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + b$$

$$z = f(u)$$

ReLU: Rectified Linear Unit

pReLU: parametric -- [He+15]
Maxout [Goodfellow+13]



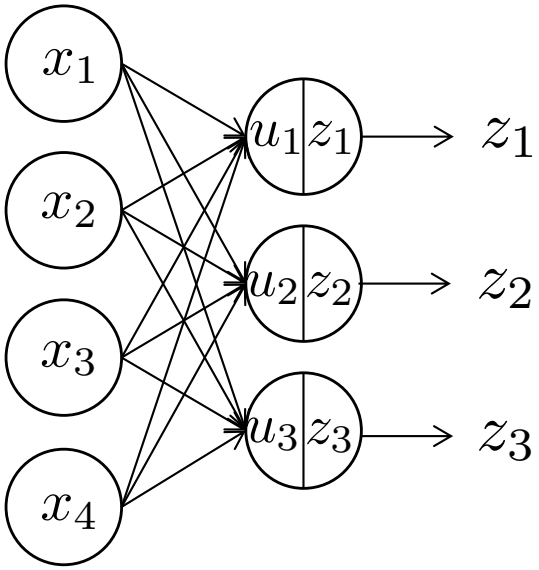
単層ネットワーク（全結合層）

$$u_j = \sum_{i=1}^{I} w_{ji} x_i + b_j$$

$$z_j = f(u_j)$$

$$\mathbf{u} = \mathbf{W}\mathbf{x} + \mathbf{b}$$

$$\mathbf{z} = \mathbf{f}(\mathbf{u})$$



$$\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_J \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_I \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_J \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} z_1 \\ \vdots \\ z_J \end{bmatrix},$$

$$\mathbf{W} = \begin{bmatrix} w_{11} & \cdots & w_{1I} \\ \vdots & \ddots & \vdots \\ w_{J1} & \cdots & w_{JI} \end{bmatrix}, \quad \mathbf{f}(\mathbf{u}) = \begin{bmatrix} f(u_1) \\ \vdots \\ f(u_J) \end{bmatrix}$$

多層ネットワーク

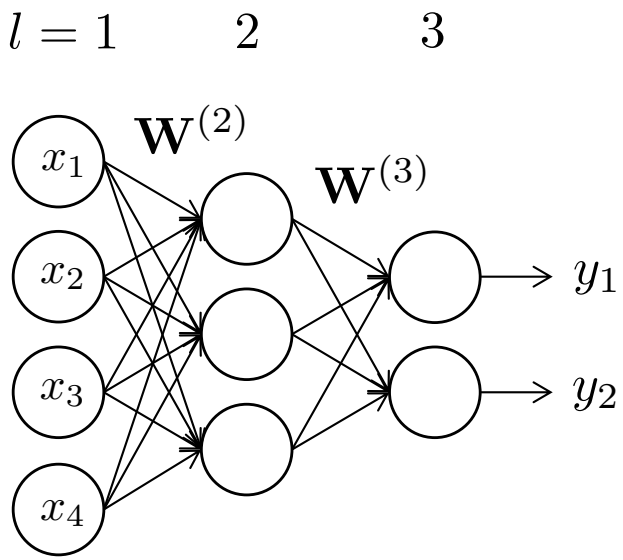
1層（入力層） $\mathbf{x} \equiv \mathbf{z}^{(1)}$

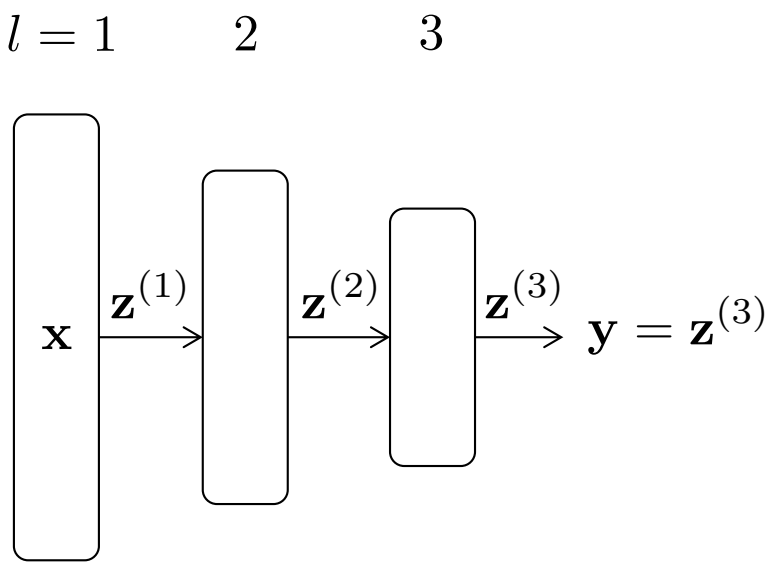
l → l+1 層への伝播

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

$$\mathbf{z}^{(l+1)} = \mathbf{f}\left(\mathbf{u}^{(l+1)}\right)$$

L層（出力層） $\mathbf{y} \equiv \mathbf{z}^{(L)}$





出力層の設計と誤差関数

クラス分類

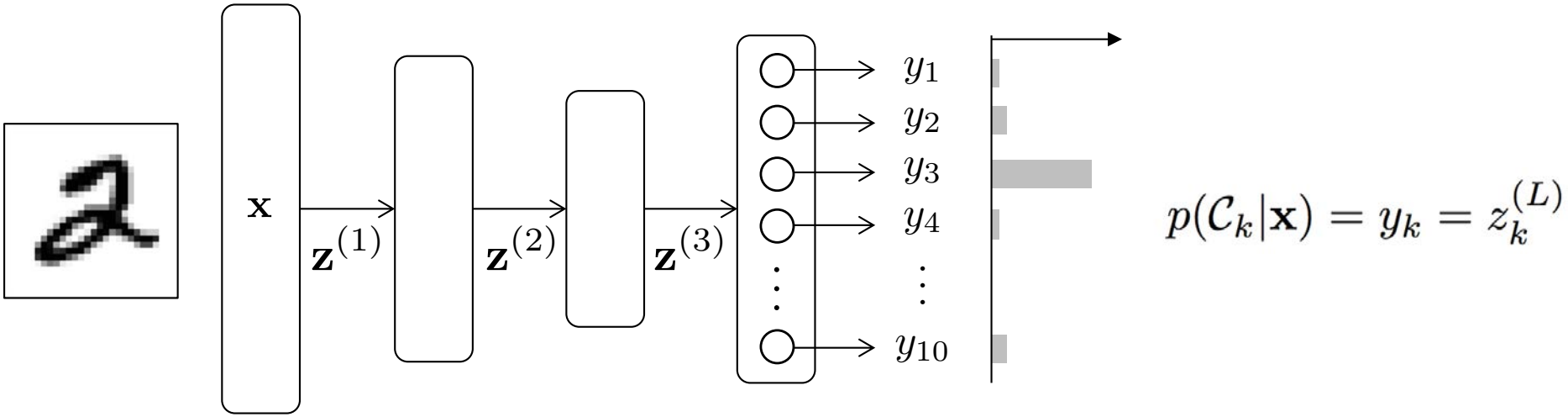
- クラス数と同数のユニット
- 活性化関数はソフトマックス
- 誤差は交差エントロピー

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} d_{nk} \log y_k(\mathbf{x}_n; \mathbf{w})$$

回帰

- 目標変数と同数のユニット
- 活性化関数は tanh や恒等写像
- 誤差は差の二乗和

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} \|\mathbf{d}_n - \mathbf{y}(\mathbf{x}_n; \mathbf{w})\|^2$$



$$p(\mathcal{C}_k|\mathbf{x}) = y_k = z_k^{(L)}$$

ソフトマックスと交差エントロピー

- 目標出力 d は正解クラスのみ１，それ以外は０をとるクラス数 K と同数の要素を持つベクトル（1-of-K符号化）

$$\mathbf{d} = [d_1, d_2, \ldots, d_K]$$

- 出力層にはソフトマックス活性化関数

$$y_k \equiv z_k^{(L)} = \frac{\exp(u_k^{(L)})}{\sum_{j=1}^{K} \exp(u_j^{(L)})} \quad \left(\sum_{k=1}^{K} y_k = 1 \right)$$

- 各ユニットの出力は各クラスの事後確率であると解釈
  - ユニットの総和は１
- ネットの出力と目標出力の差（誤差）：交差エントロピー

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} d_{nk} \log y_k(\mathbf{x}_n; \mathbf{w})$$

勾配降下法による学習

- E(w) の勾配方向にパラメータを繰り返し小修正

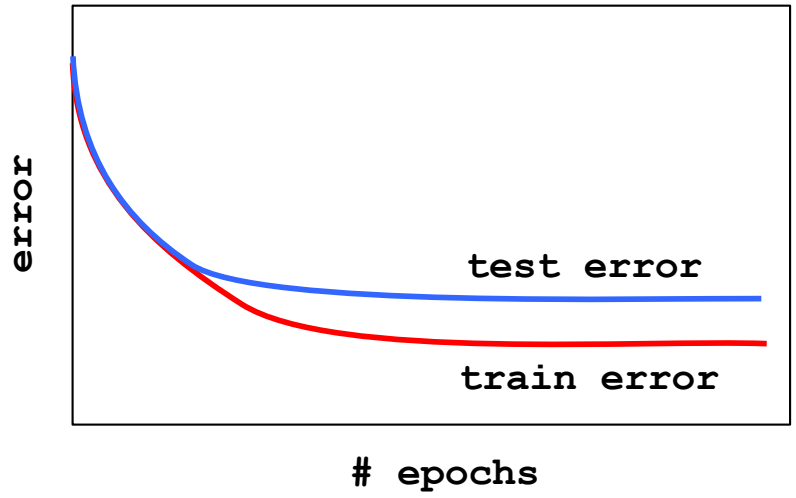
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \epsilon \nabla E$$

ε：学習係数（学習率）

$$\nabla E \equiv \frac{dE}{d\mathbf{w}} = \left[\frac{\partial E}{\partial w_1}, \cdots, \frac{\partial E}{\partial w_M} \right]^{\top}$$

パラメータ（重み）はランダムに初期化

- 学習に使っていないサンプルで汎化性能を予測評価



勾配の計算

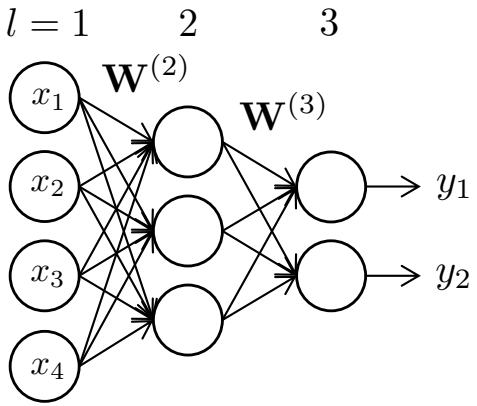
- 勾配の計算は大変

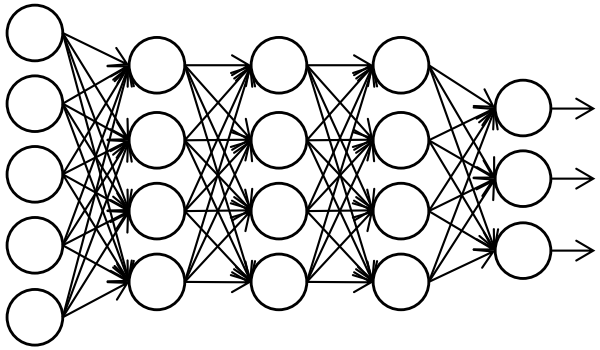
$$\frac{\partial E_n}{\partial w_{ji}^{(l)}} = (\mathbf{y}(\mathbf{x}_n) - \mathbf{d}_n)^\top \frac{\partial \mathbf{y}}{\partial w_{ji}^{(l)}}$$

２乗誤差の場合

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} \|\mathbf{d}_n - \mathbf{y}(\mathbf{x}_n; \mathbf{w})\|^2$$

$$\begin{aligned}
\mathbf{y}(\mathbf{x}) &= \mathbf{f}(\mathbf{u}^{(L)}) \\
&= \mathbf{f}(\mathbf{W}^{(L)}\mathbf{z}^{(L-1)} + \mathbf{b}^{(L)}) \\
&= \mathbf{f}(\mathbf{W}^{(L)}\mathbf{f}(\mathbf{W}^{(L-1)}\mathbf{z}^{(L-2)} + \mathbf{b}^{(L-1)}) + \mathbf{b}^{L}) \\
&= \mathbf{f}(\mathbf{W}^{(L)}\mathbf{f}(\mathbf{W}^{(L-1)}\mathbf{f}(\cdots \mathbf{f}(\mathbf{W}^{(l)}\mathbf{z}^{(l-1)} + \mathbf{b}^{(l)})\cdots)) + \mathbf{b}^{(L)})
\end{aligned}$$

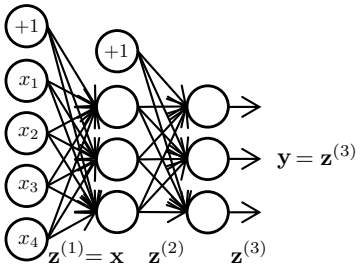


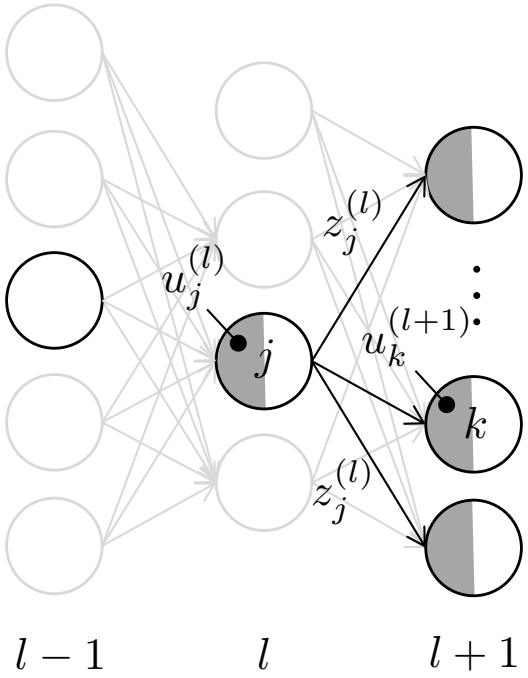


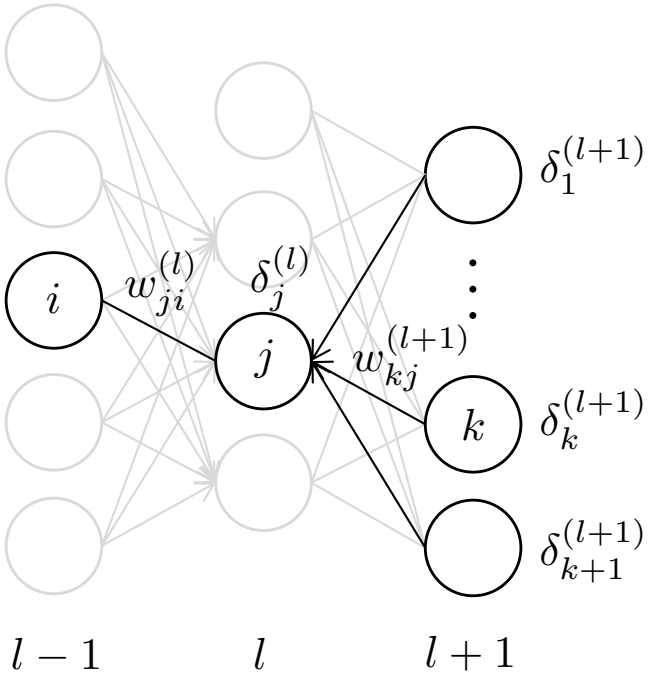
誤差逆伝播
(BP: Back Propagation)

”デルタ”を定義： $\delta_j^{(l)} \equiv \dfrac{\partial E_n}{\partial u_j^{(l)}}$ ⇒ 勾配は $\dfrac{\partial E_n}{\partial w_{ji}^{(l)}} = \delta_j^{(l)} z_i^{(l-1)}$

各層のデルタは逆伝播可能： $\delta_j^{(l)} = \sum_k \delta_k^{(l+1)} \left(w_{kj}^{(l+1)} f'(u_j^{(l)}) \right)$

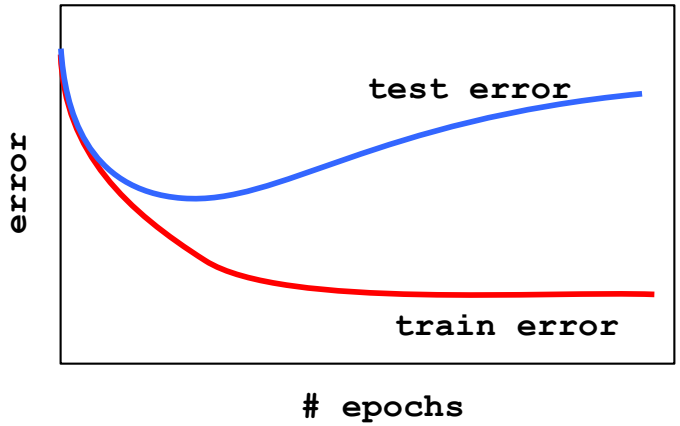




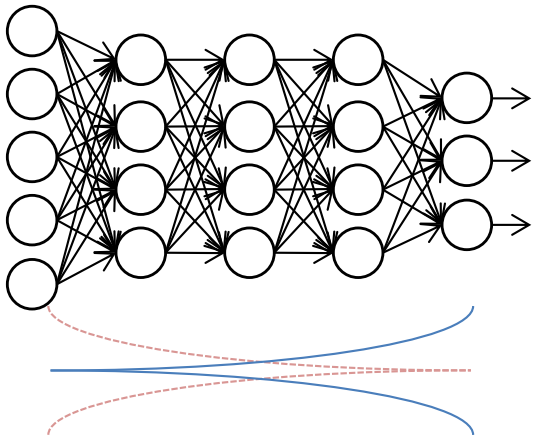


ディープネットの学習の困難さ

- 多層になると過適合（過学習）・ローカルミニマム



- 勾配消失（vanishing gradient）問題
  - デルタが急速に小さく，あるいは大きくなり制御できない



確率的勾配降下法
(SGD: Stochastic Gradient Descent)

バッチ学習

- 全サンプルについての誤差の総和を最小化

$$E(\mathbf{w}) = \sum_{n=1}^{N} E_n(\mathbf{w})$$

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \epsilon \nabla E$$

確率的勾配降下

- サンプル１個に関する誤差を最小化することを，サンプルをランダムに選びなおして繰り返す

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \epsilon \nabla E_n$$

毎回（tごとに）異なるEnを最小化していることに

ミニバッチとモメンタム

- ミニバッチ：数百個まで程度のサンプル集合ごとにパラメータ更新

$$E_t(\mathbf{w}) = \frac{1}{N_t} \sum_{n \in \mathcal{D}_t} E_n(\mathbf{w})$$

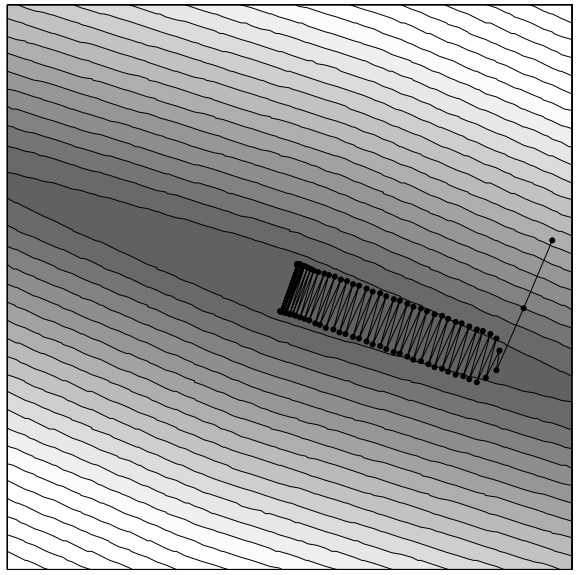
- モメン

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \epsilon \nabla E_t + \mu \Delta \mathbf{w}^{(t-1)}$$

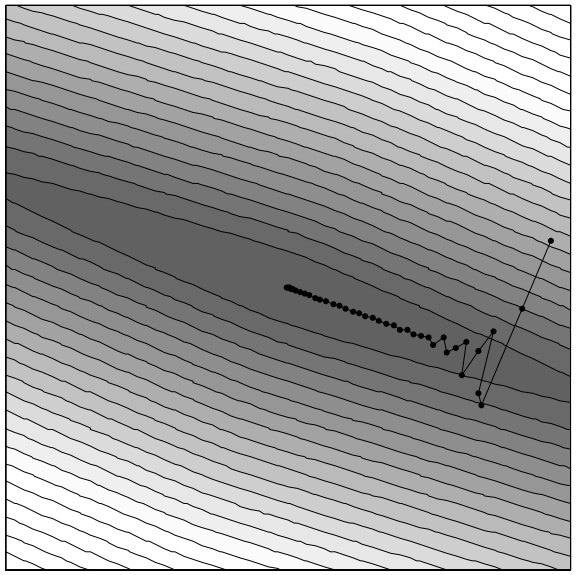
).5〜(前回修正量

$$\Delta \mathbf{w}^{(t-1)} \equiv \mathbf{w}^{(t-1)} - \mathbf{w}^{(t-2)}$$

加算



モメンタムなし



モメンタムあり

その他のトリック

- 重み減衰（Weight decay）
  - 際限なく重みが発散することを防ぐ・正則化の一種
  - λは普通ごく小さい値

$$E_t(\mathbf{w}) \equiv \frac{1}{N_t} \sum_{n \in \mathcal{D}_t} E_n(\mathbf{w}) + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \epsilon \left(\frac{1}{N_t} \sum \nabla E_n + \lambda \mathbf{w}^{(t)} \right)$$

- データの正規化（標準化）・白色化
- 重みの初期値（通常，ランダム）の決め方
- 学習係数の制御
  - 手動／自動で最初は大きく，徐々に小さく

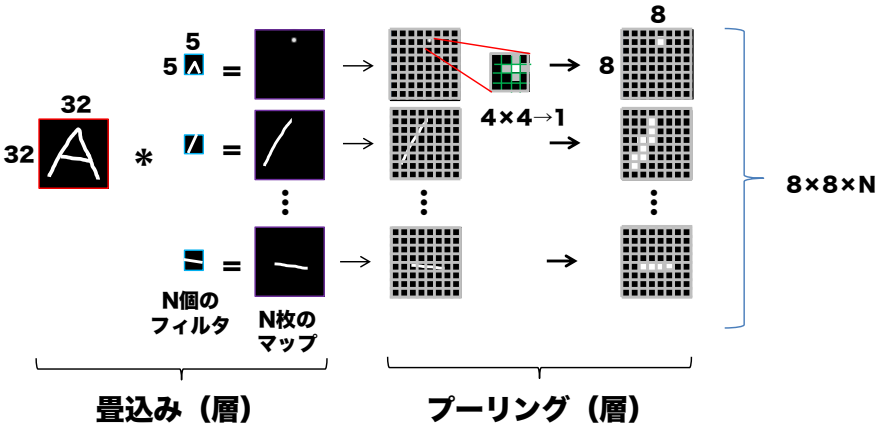
Batch normalization [Ioffe-Szegedy15]

“Xavier” initialization [Glorot-Bengio10]

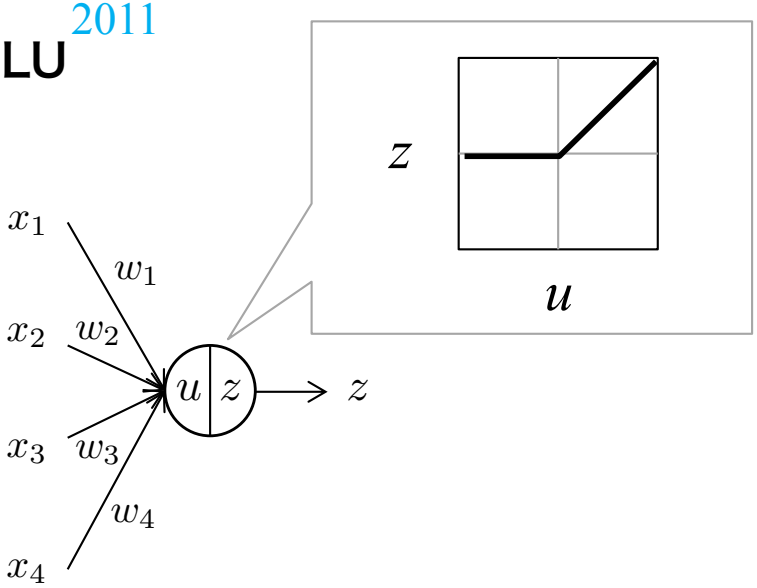
PReLU[He+15]

深層学習の中核技術

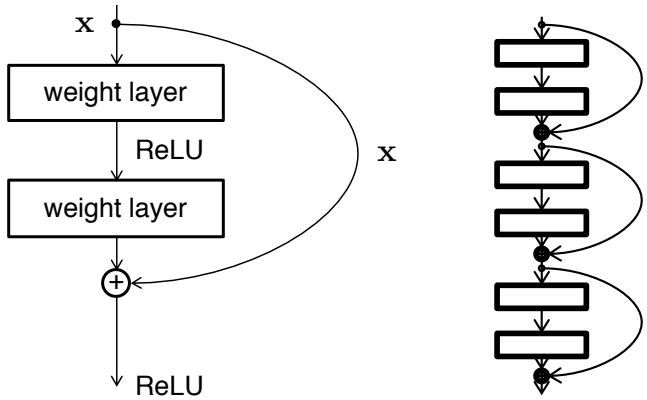
Convlution+Downsampling 1980-90



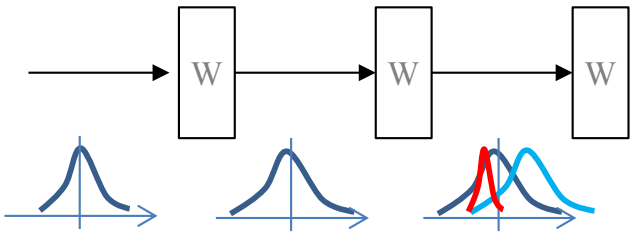
ReLU 2011



Skipconnection (ResNet) 2015



Batch Normalization 2015



CNN：畳込みニューラルネット
Convolutional Neural Network

- Neocognitronにルーツ [Fukushima80]
- LeNet：手書き文字認識への応用で成功 [LeCun+89]
  - Backpropagation Applied to Handwritten Zip Code Recognition, 1989
- 神経科学の知見が基礎
  - Hubel-Wiesel の単純細胞・複雑細胞
  - 局所受容野（local receptive field）

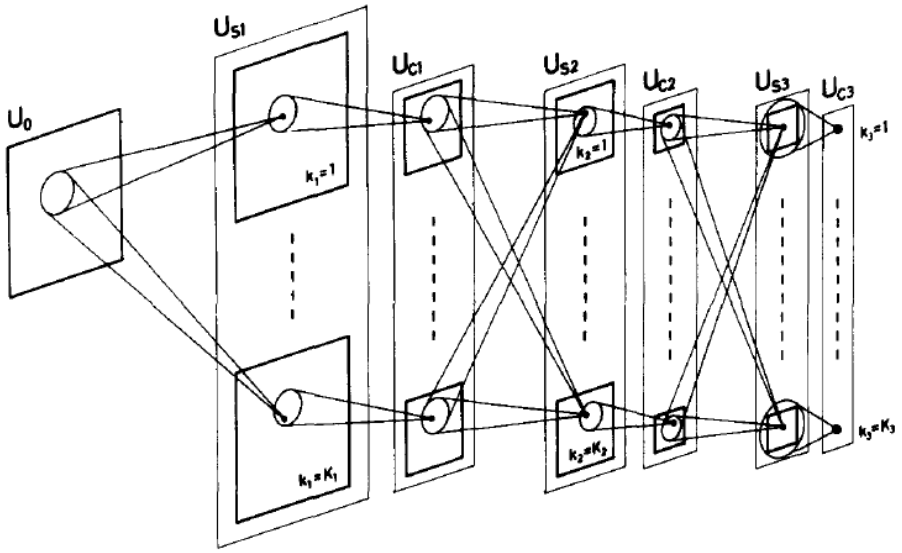


Fig 4 Schematic diagram illustrating the interconnections between layers in the neocognitron

[Fukushima+83]

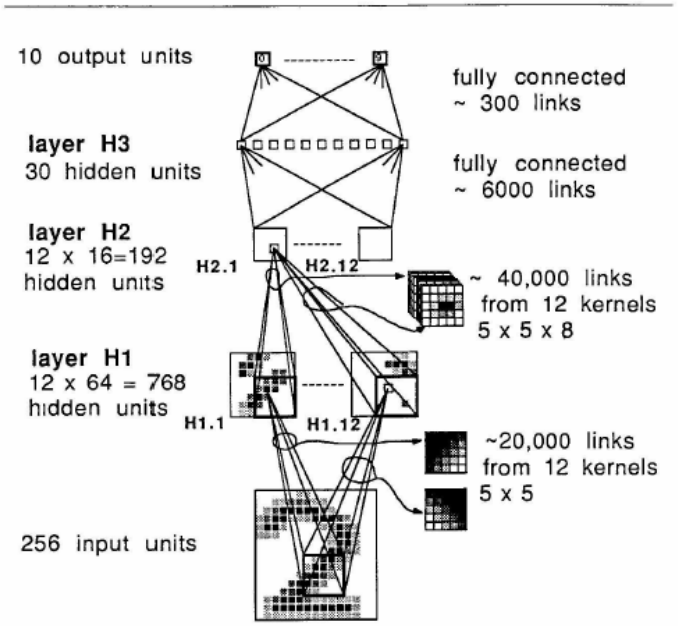
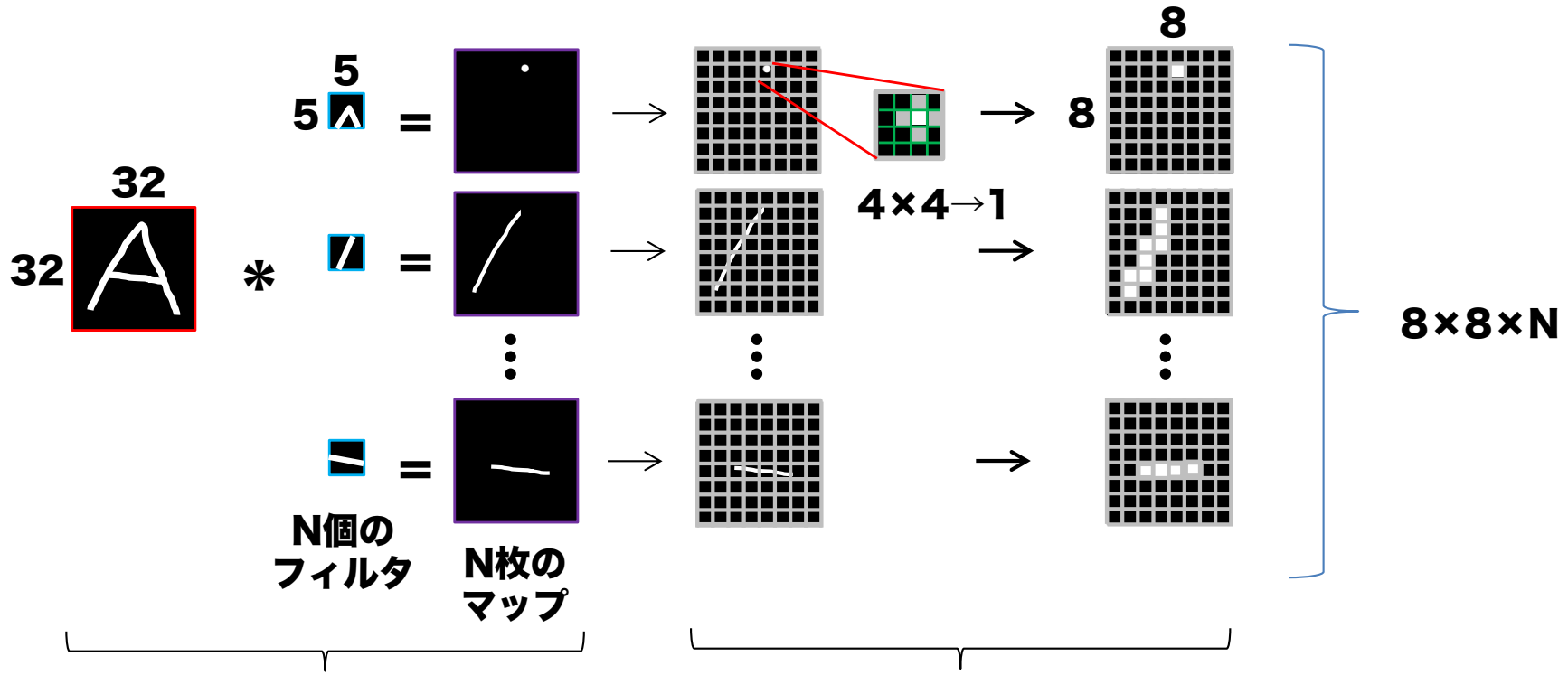
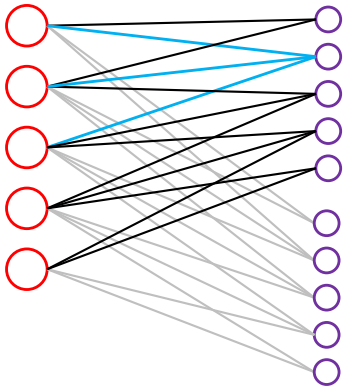


Figure 3 Log mean squared error (MSE) (top) and raw error rate (bottom) versus number of training passes

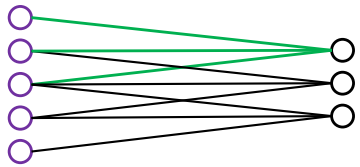
[LeCun+89]

CNNの二つの演算：畳込みとプーリング





- 共有重み
- 疎結合



- 固定配線（重み）
- 疎結合

畳込み
(convolution)

$$u_{ij} = \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i+p,j+q} h_{pq}$$

入力画像

77	80	82	78	70	82	82	140
83	78	80	83	82	77	94	151
87	82	81	80	74	75	112	152
87	87	85	77	66	99	151	167
84	79	77	78	76	107	162	160
86	72	70	72	81	151	166	151
78	72	73	73	107	166	170	148
76	76	77	84	147	180	168	142

$x_{i+p,j+q}$

$\otimes$

フィルタ

0.01	0.08	0.01
0.08	0.62	0.08
0.01	0.08	0.01

h_{pq}

$=$

出力（マップ）

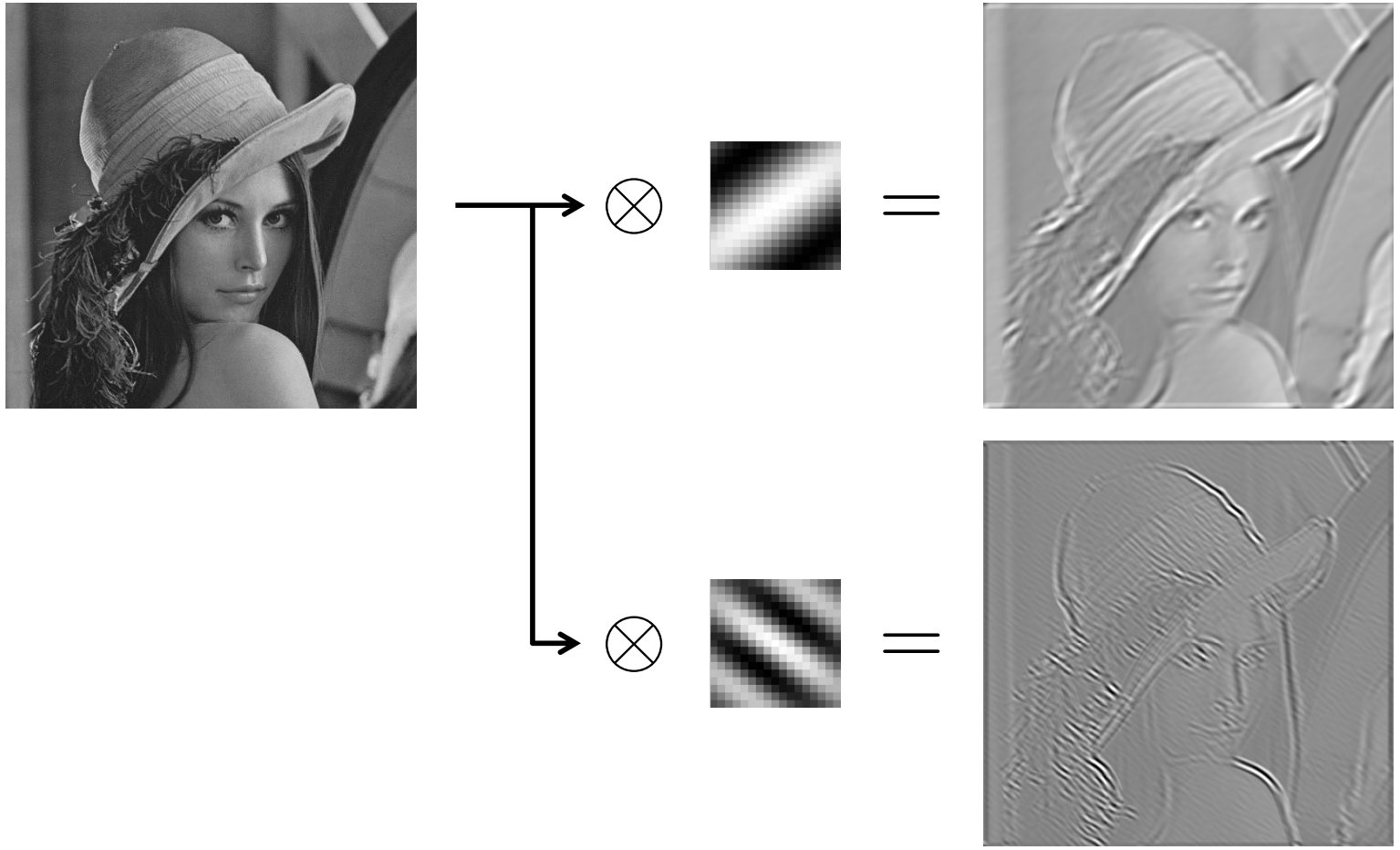
79	80	81	79	79	98
82	81	79	75	81	114
85	83	77	72	99	144
79	77	77	79	112	155
73	71	73	89	142	162
73	73	77	110	160	166

u_{ij}

畳込み
(convolution)

$$u_{ij} = \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i+p,j+q} h_{pq}$$

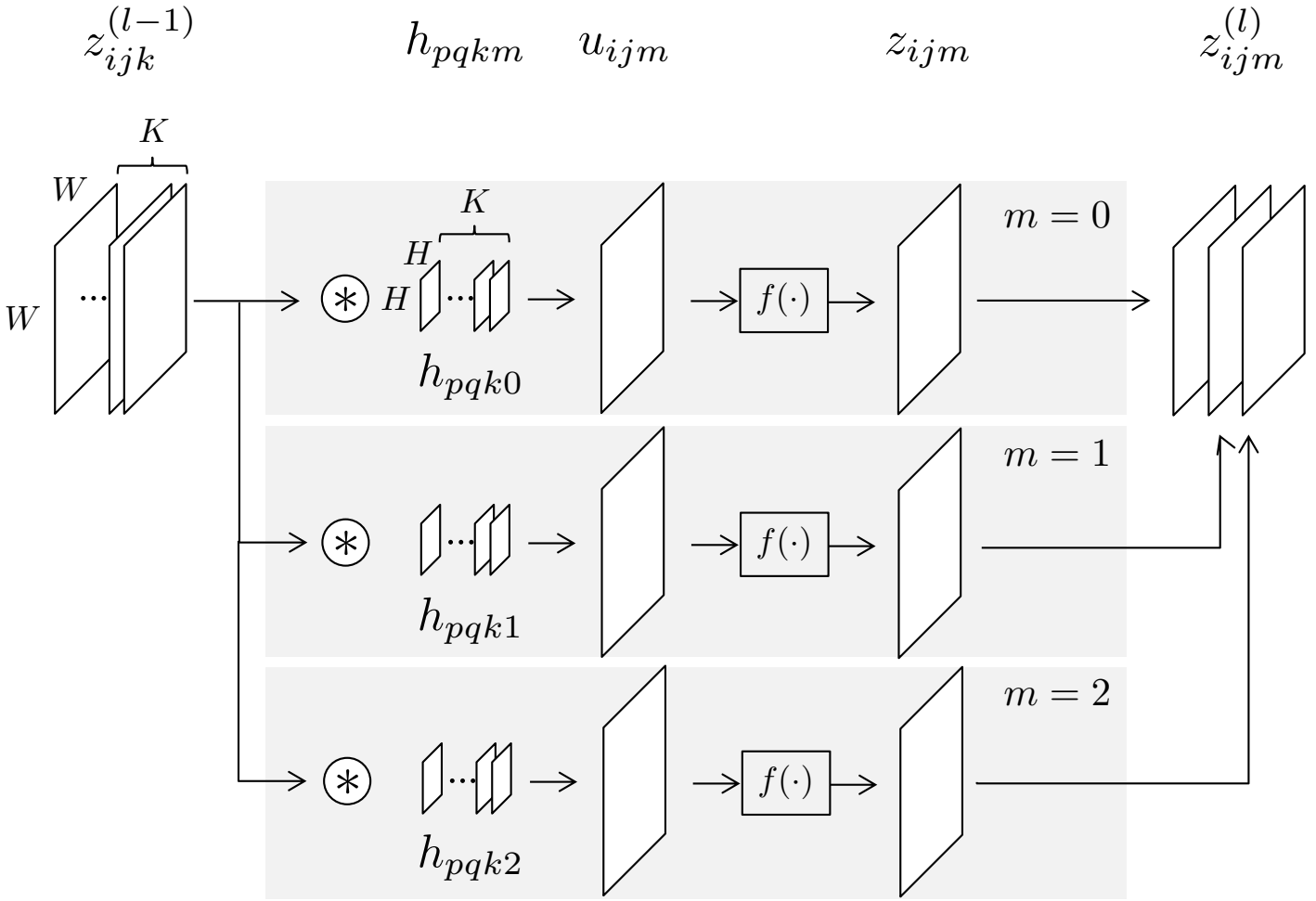
入力画像　　フィルタ　　出力（マップ）



$x_{i+p,j+q}$　　h_{pq}　　u_{ij}

畳込み

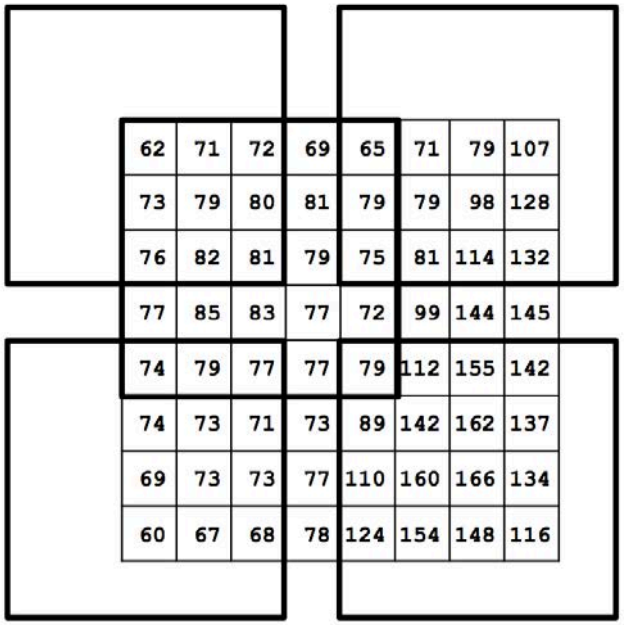
- 多チャネル入力・多チャネル出力



プーリング
(pooling)

- 小領域を代表する値を1つ選ぶ
  - 計算の構造自体は畳込みと類似
  - 小領域の間隔を空けることで，出力画像の解像度を小さくする

入力画像



→

出力画像

82	82	114	132
85	85	155	155
85	110	166	166
79	124	166	166

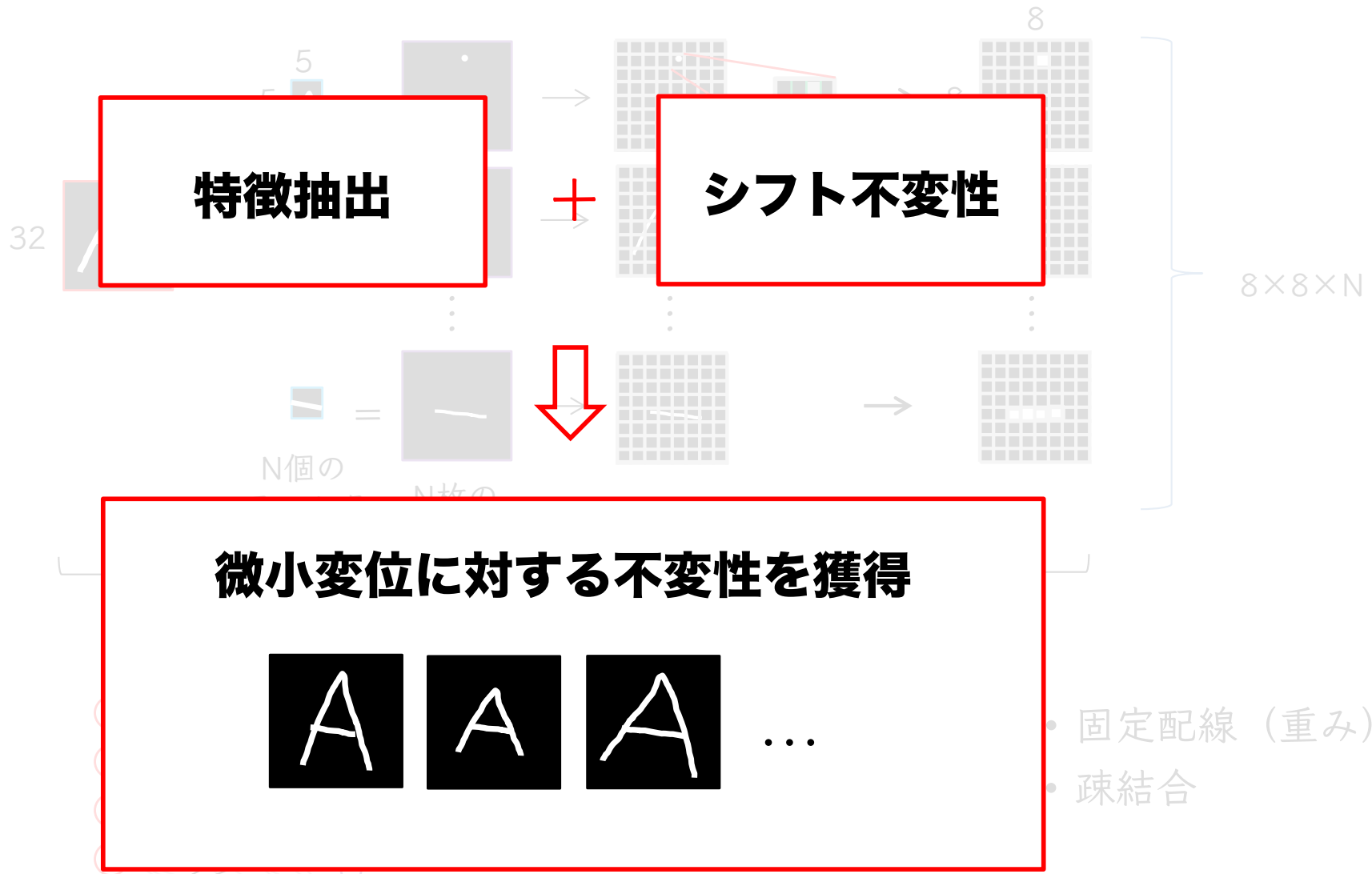
最大プーリング

$$u_{ijk} = \max_{(p,q)\in P_{ij}} z_{pqk}$$

平均プーリング

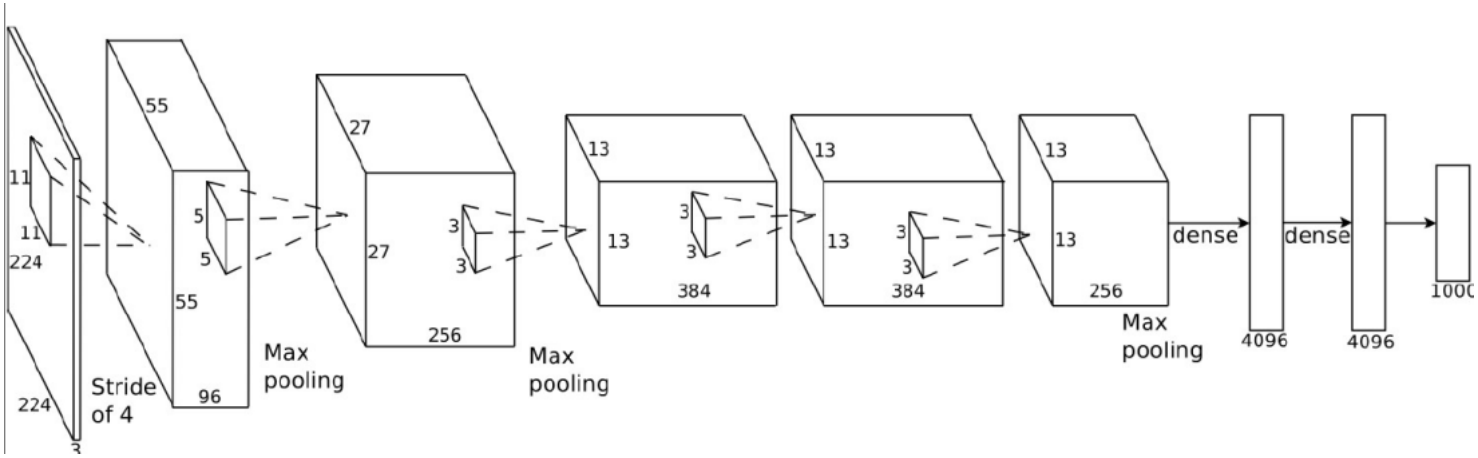
$$u_{ijk} = \frac{1}{H^2} \sum_{(p,q)\in P_{ij}} z_{pqk}$$

二つの演算：畳込みとプーリング



CNN：畳込みニューラルネット

- Convolutional Neural Network
  - 畳込み層とプーリング層の交互反復＋全結合層を持つフィードフォワードネット
  - 重みをランダムに初期化し，教師あり学習

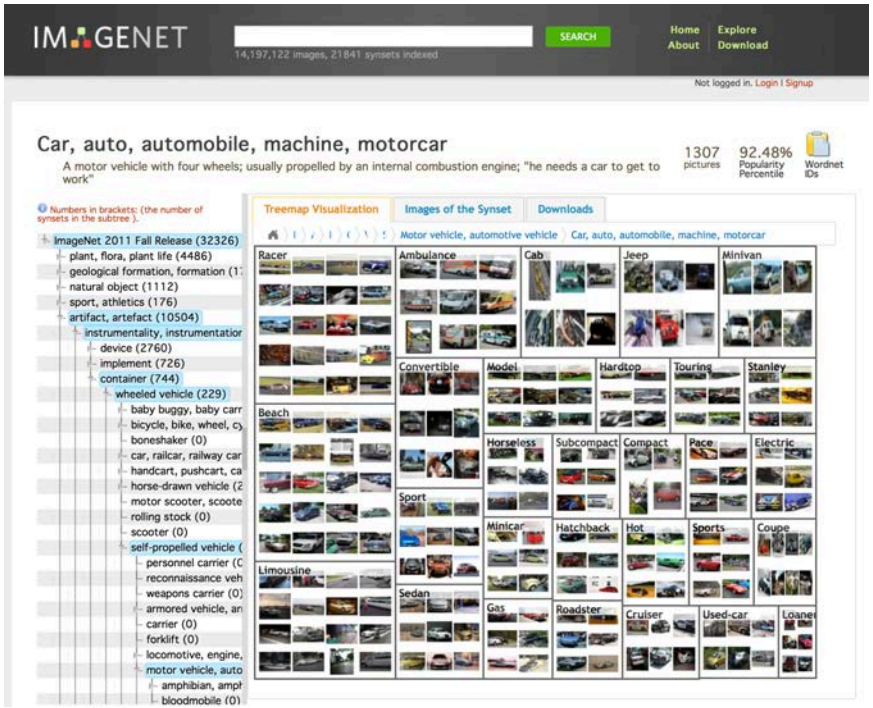


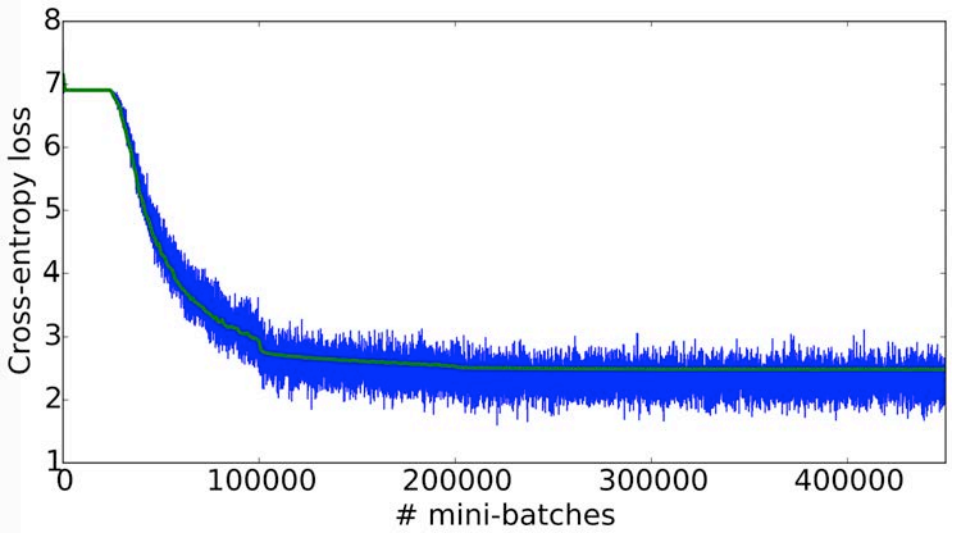
ILSVRC12のCNN [Krizhevsky+12]

“Alexnet”

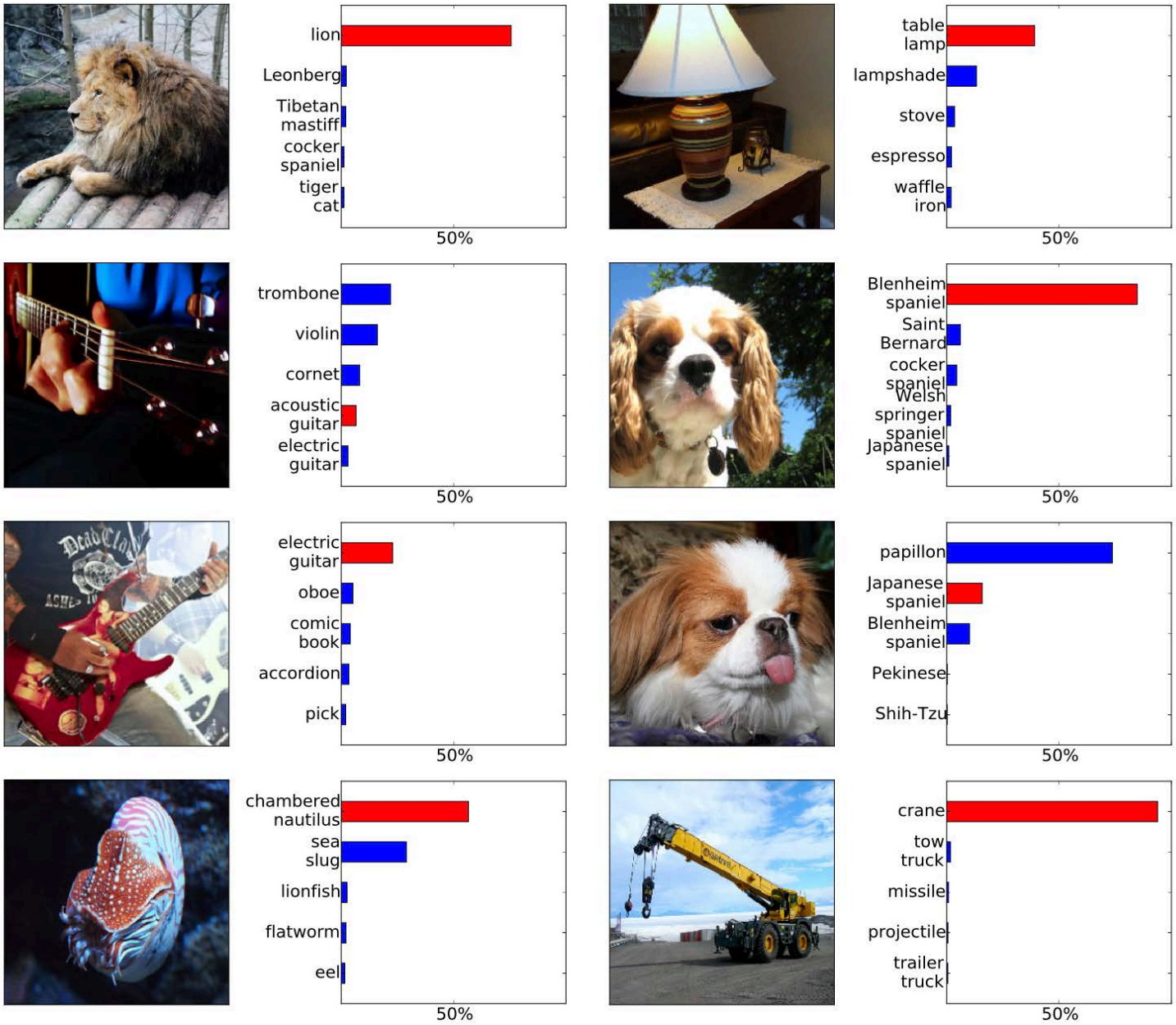
CNNで1000クラス物体認識を学習

- 100万枚を超える学習サンプル（画像・クラスラベル）
  - 1000の物体カテゴリ・1000枚／カテゴリ
- 最新のGPUを用いて数日〜数週間かけて学習
- 人を超える認識精度
  (He+, Delving deep into rectifier, 2015)



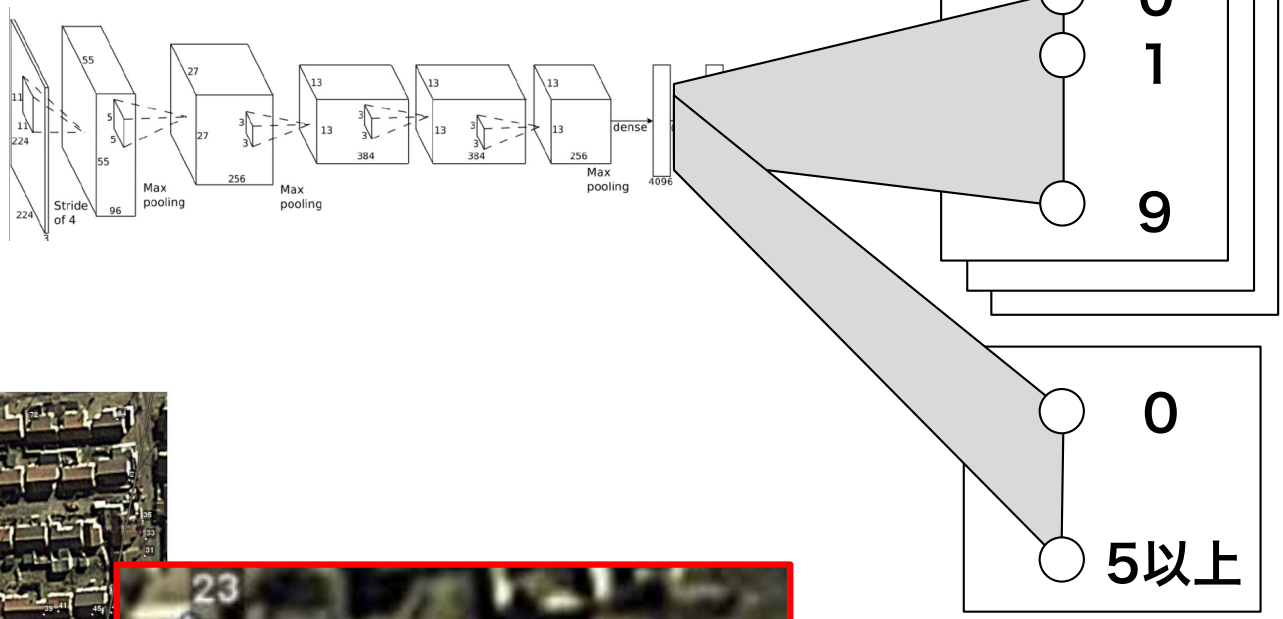


实例


lion
Leonberg
Tibetan mastiff
cocker spaniel
tiger cat
50%
table lamp
lampshade
stove
espresso
waffle iron
50%
trombone
violin
cornet
acoustic guitar
electric guitar
50%
Blenheim spaniel
Saint Bernard
cocker spaniel
Welsh springer spaniel
Japanese spaniel
50%
electric guitar
oboe
comic book
accordion
pick
50%
papillon
Japanese spaniel
Blenheim spaniel
Pekinese
Shih-Tzu
50%
chambered nautilus
sea slug
lionfish
flatworm
eel
50%
crane
tow truck
missile
projectile
trailer truck
50%

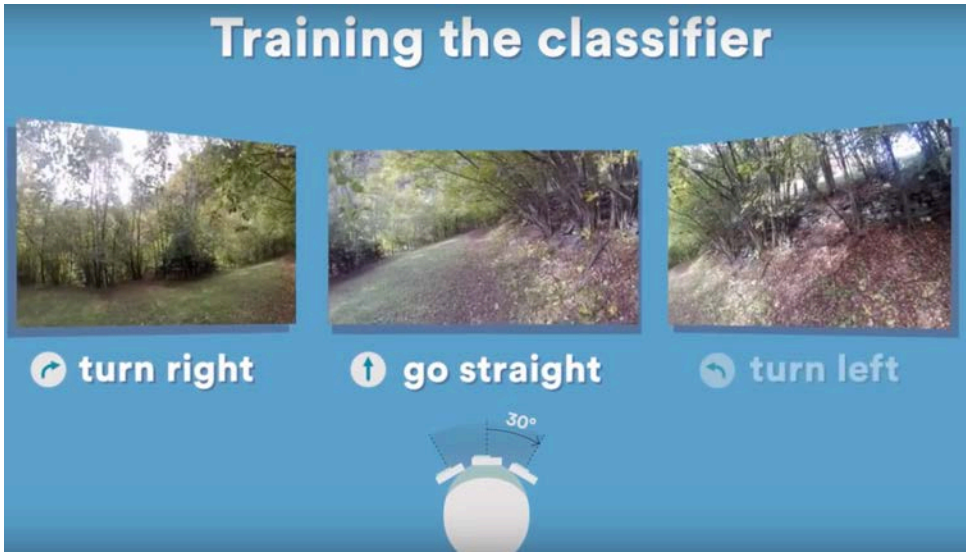
複数桁数の認識

Goodfellow+, Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks, 2013



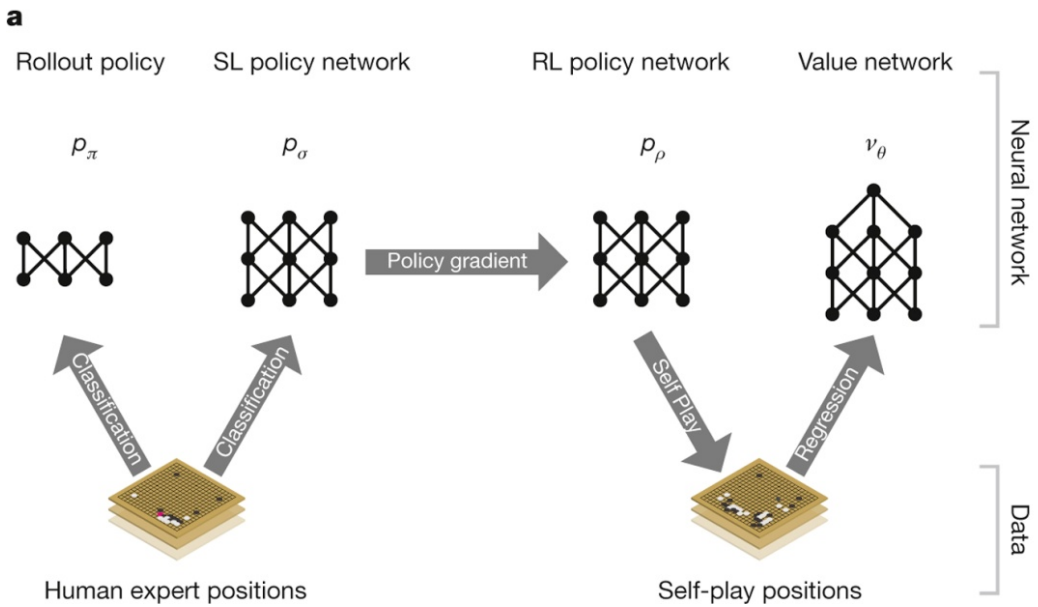
ドローン自律飛行

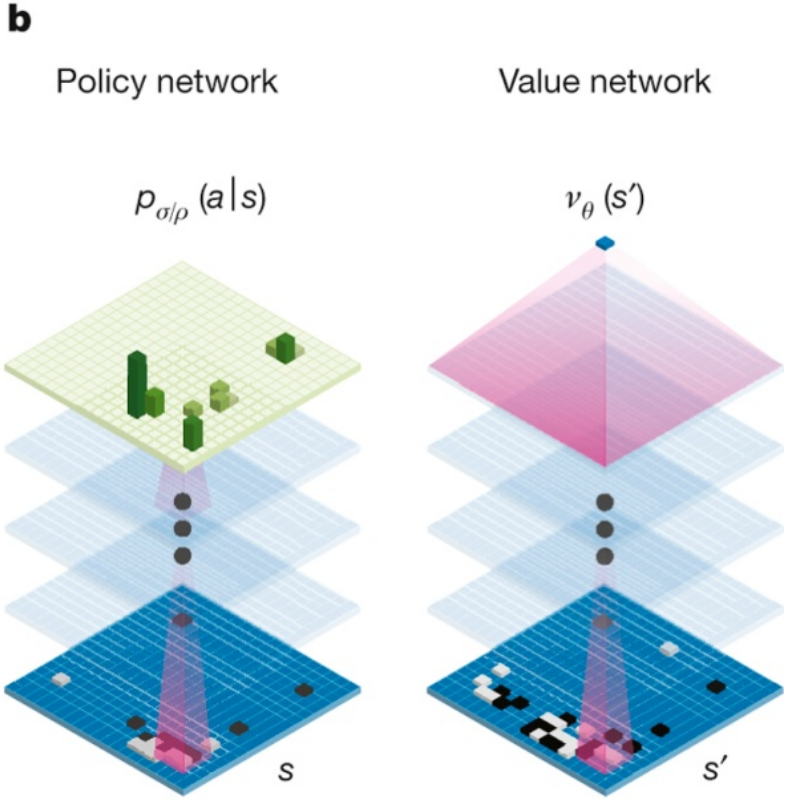
Giusti+, A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots, 2016



AlphaGo

D Silver *et al. Nature* **529,** 484–489 (2016) doi:10.1038/nature16961

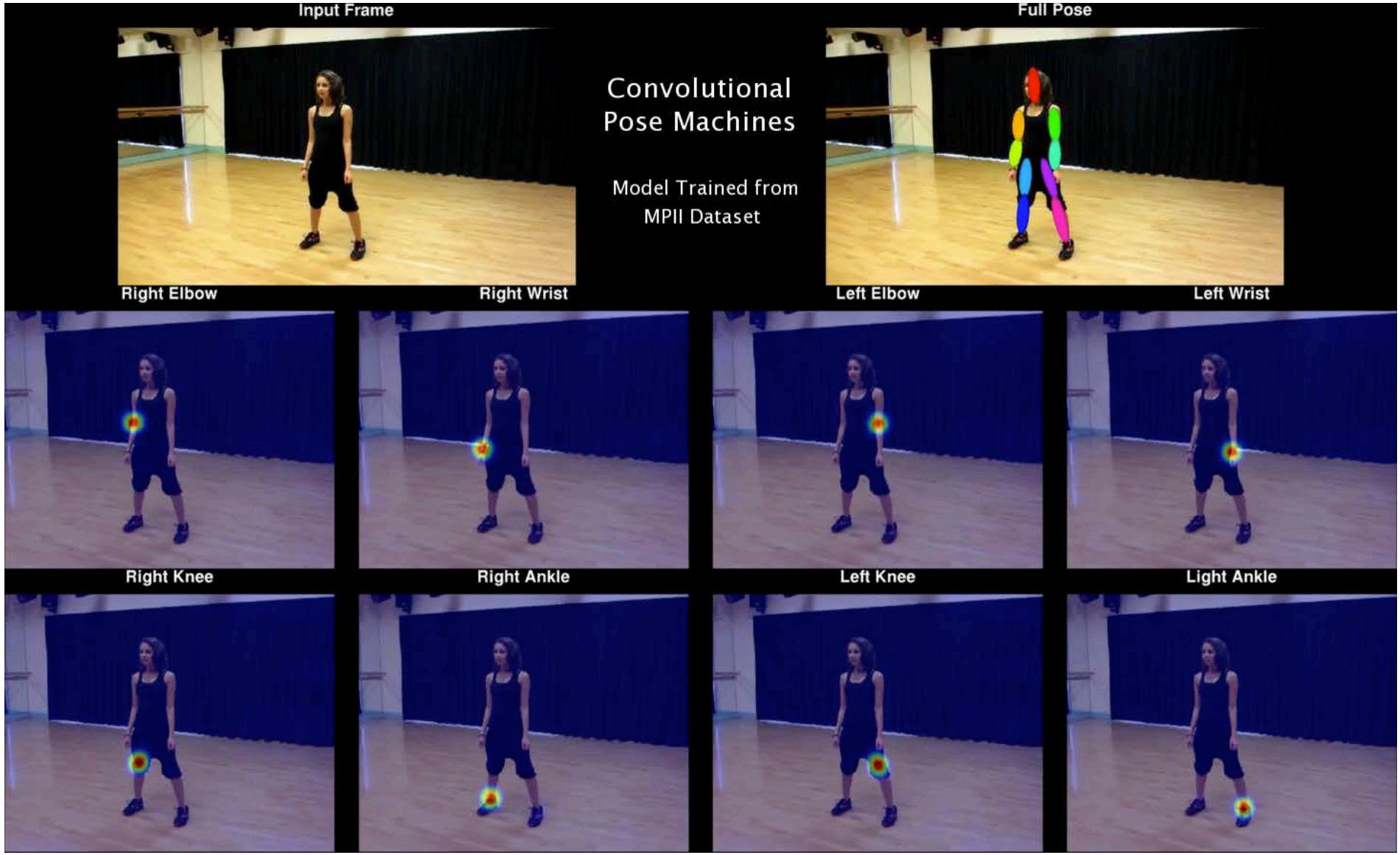




nature

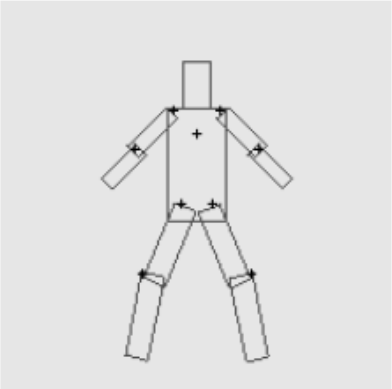
ポーズ推定：strucured prediction?パタン認識？

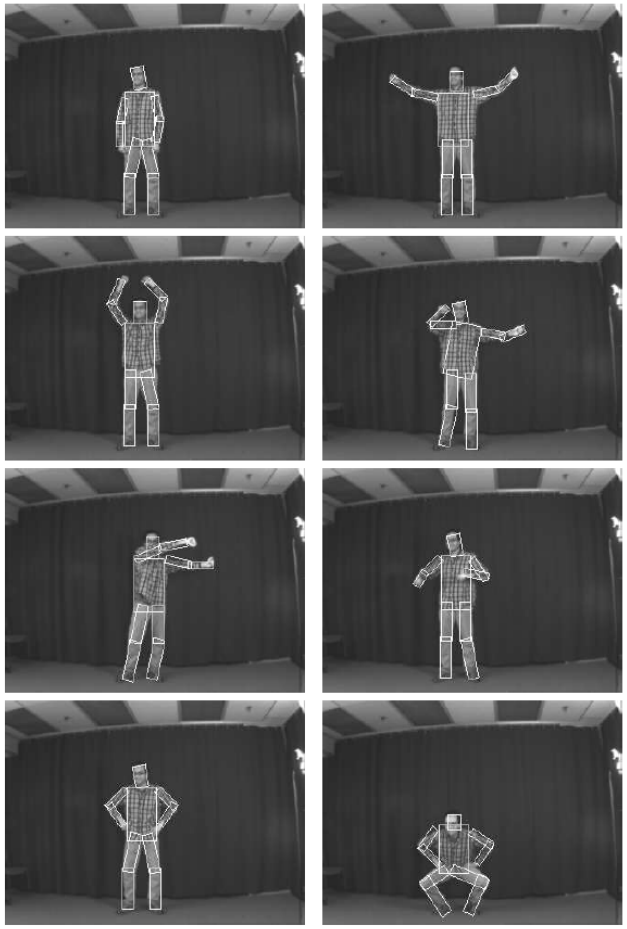
Shih-En Wei, Varun Ramakrishna, Takeo Kanade, Yaser Sheikh, Convolutional Pose Machines, 2016

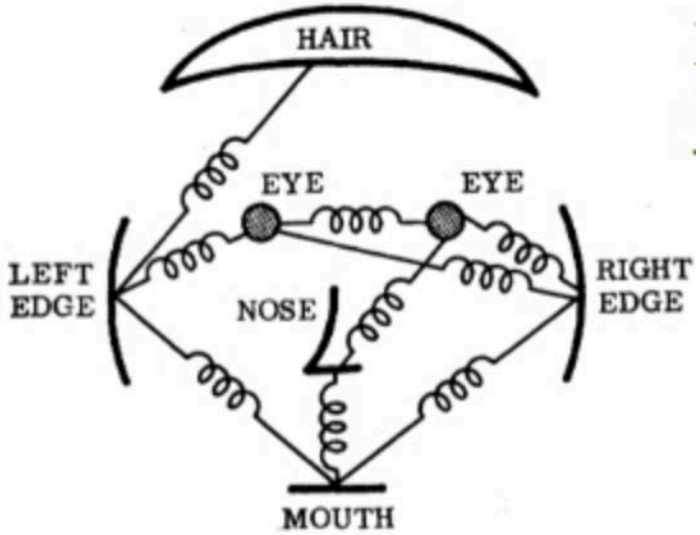


ポーズ推定問題の本質

- 伝統的には，典型的なstructured prediction
  - 各関節位置の推定が相互に影響を及ぼす
- 全体最適化の問題として定式化

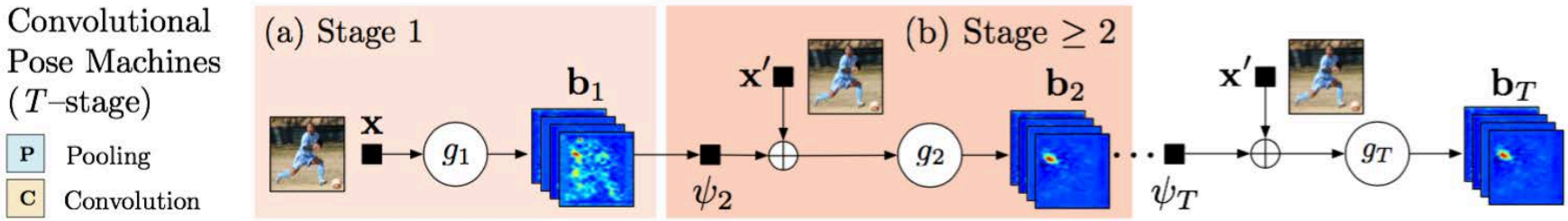


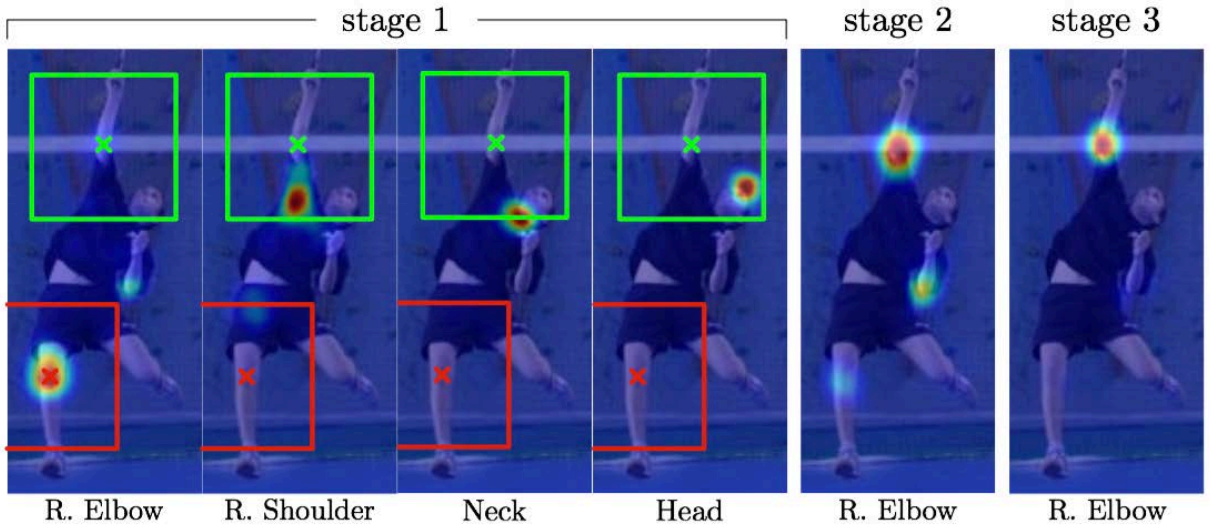


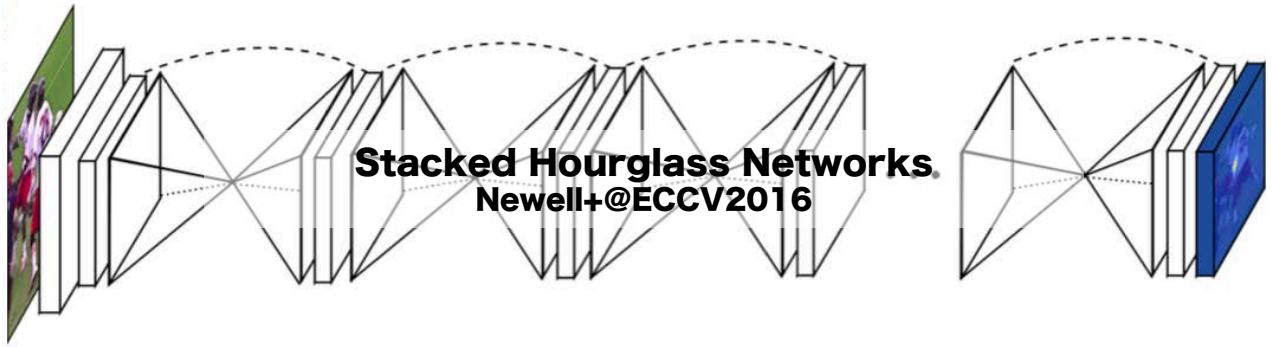


フォワード計算（の反復）でstructured prediction

Shih-En Wei, Varun Ramakrishna, Takeo Kanade, Yaser Sheikh, Convolutional Pose Machines, 2016

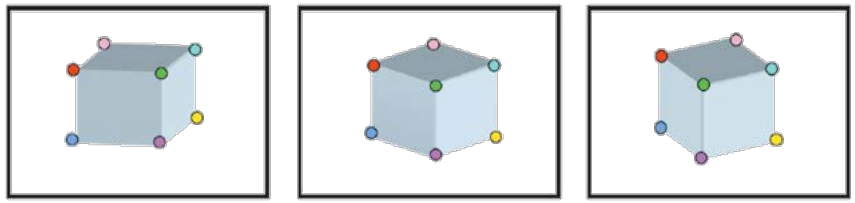


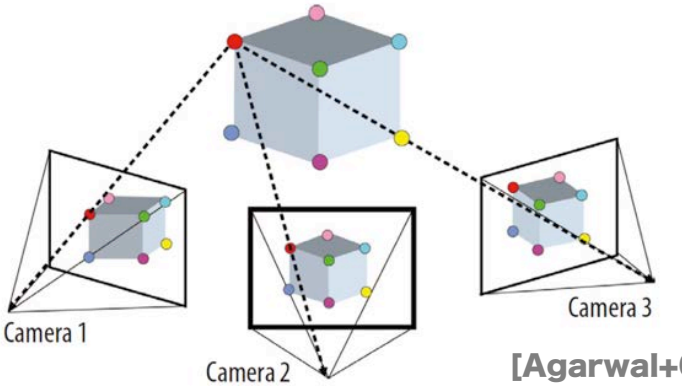




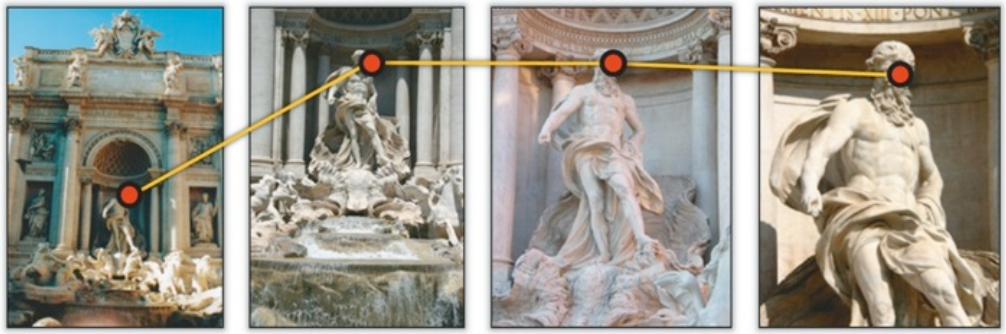
SfM：多視点画像→３次元再構成

structure-from-motion





[Agarwal+09]



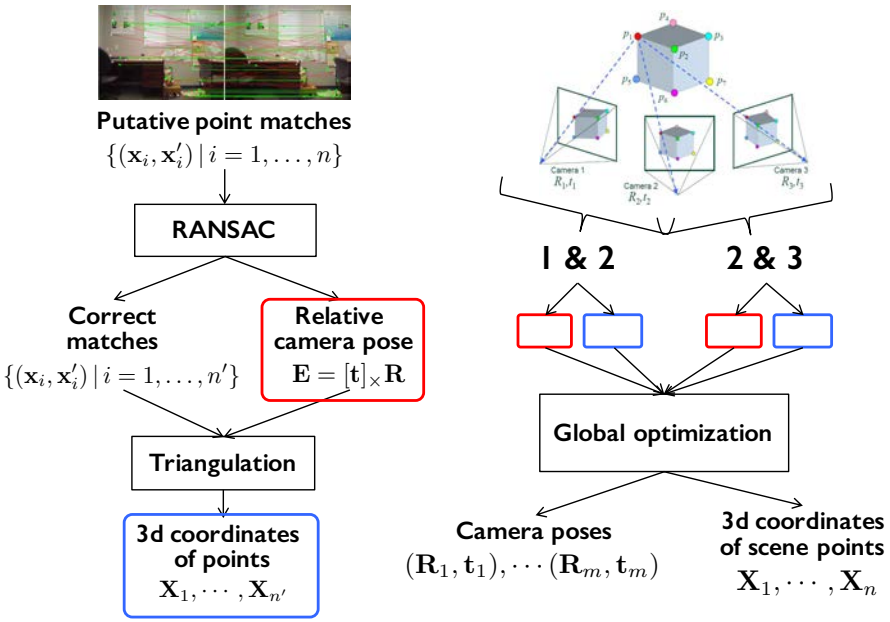
[Agarwal+09]

同じ点がどこに写っているか？

$$\underbrace{\begin{bmatrix} \mathbf{x}_{11} & \mathbf{x}_{12} & \cdots & \mathbf{x}_{1n} \\ \mathbf{x}_{21} & \mathbf{x}_{22} & \cdots & \mathbf{x}_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{x}_{m1} & \mathbf{x}_{m2} & \cdots & \mathbf{x}_{mn} \end{bmatrix}}_{n} \quad \underbrace{\begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_m \end{bmatrix}}_{4} \quad \underbrace{\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots \mathbf{X}_n \end{bmatrix}}_{n}$$

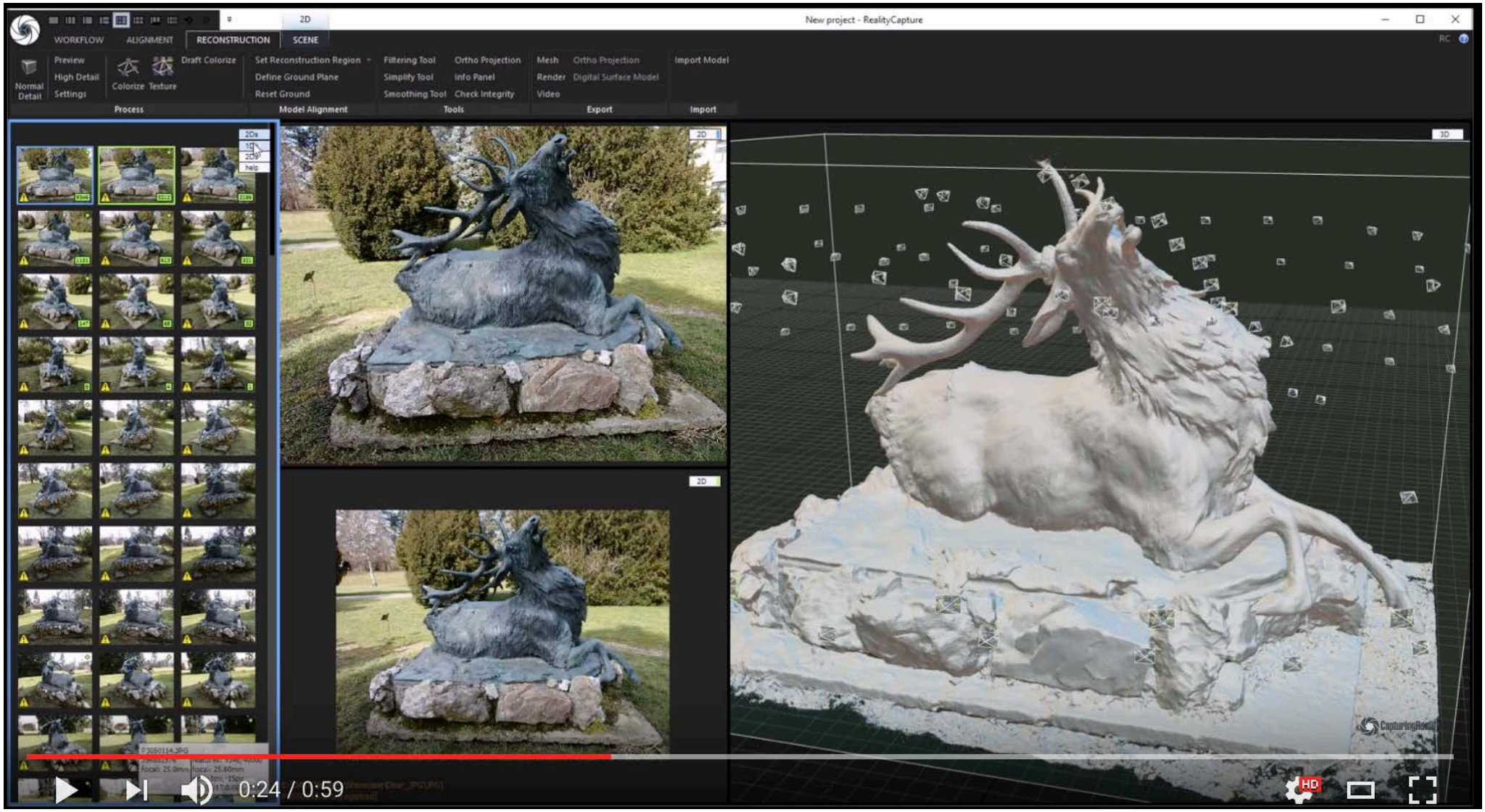
m

点の画像座標　**カメラの姿勢**　**点の空間座標**



処理のフロー

SfM：多視点画像→３次元再構成



RealityCapture　www.capturingreality.com

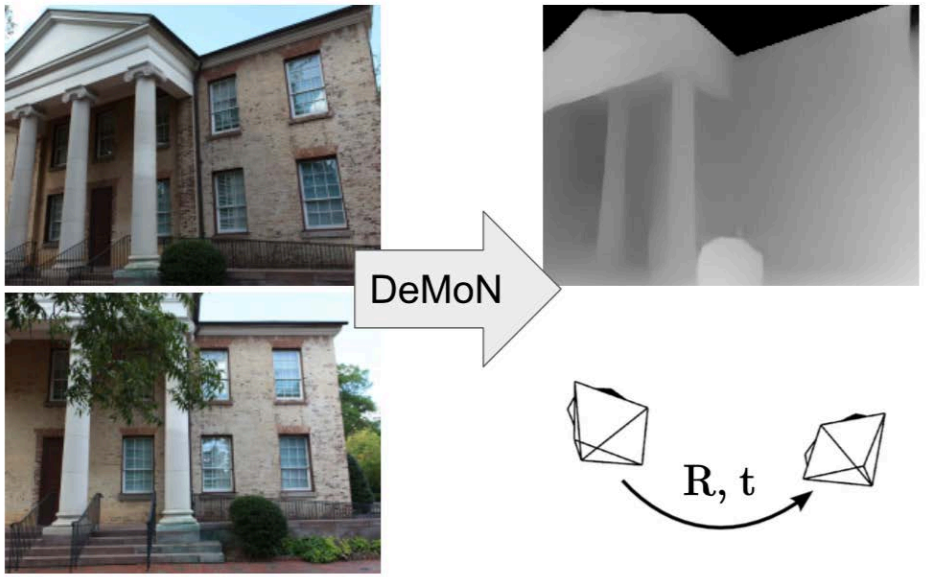
CNNでSfM（+MVS）が解ける

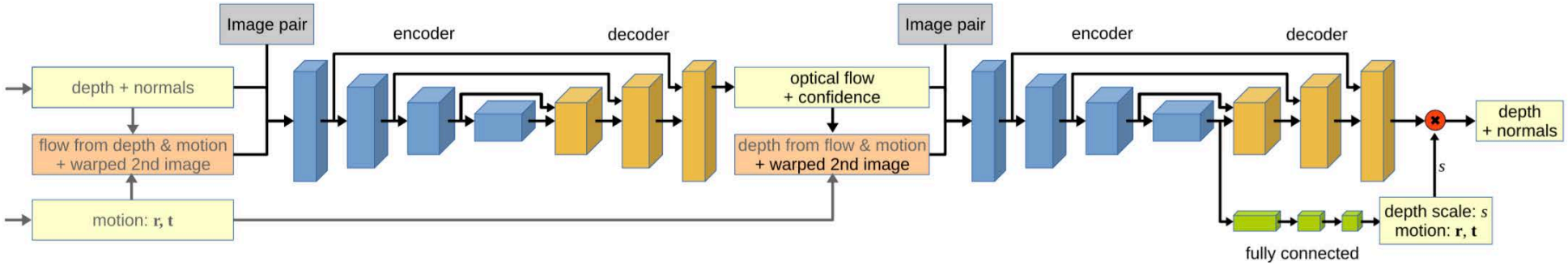
Ummenhofer+, DeMoN: Depth and Motion Network for Learning Monocular Stereo, CVPR2017

- CNNで2-view SFMを実行

		Depth			Motion	
	Method	L1-inv	sc-inv	L1-rel	rot	trans
MVS	Base-Oracle	**0.019**	**0.197**	**0.105**	0	0
	Base-SIFT	0.056	0.309	0.361	21.180	60.516
	Base-FF	0.055	0.308	0.322	**4.834**	17.252
	DeMoN	0.047	0.202	0.305	5.156	**14.447**
Scenes11	Base-Oracle	0.023	0.618	0.349	0	0
	Base-SIFT	0.051	0.900	1.027	6.179	56.650
	Base-FF	0.038	0.793	0.776	1.309	19.425
	DeMoN	**0.019**	**0.315**	**0.248**	**0.809**	**8.918**
RGB-D	Base-Oracle	**0.026**	0.398	0.336	0	0
	Base-SIFT	0.050	0.577	0.703	12.010	56.021
	Base-FF	0.045	0.548	0.613	4.709	46.058
	DeMoN	0.028	**0.130**	**0.212**	**2.641**	**20.585**
Sun3D	Base-oracle	0.020	0.241	0.220	0	0
	Base-SIFT	0.029	0.290	0.286	7.702	41.825
	Base-FF	0.029	0.284	0.297	3.681	33.301
	DeMoN	**0.019**	**0.114**	**0.172**	**1.801**	**18.811**
NYUv2	Base-oracle	-	-	-	-	-
	Base-SIFT	-	-	-	-	-
	Base-FF	-	-	-	-	-
	DeMoN	-	-	-	-	-

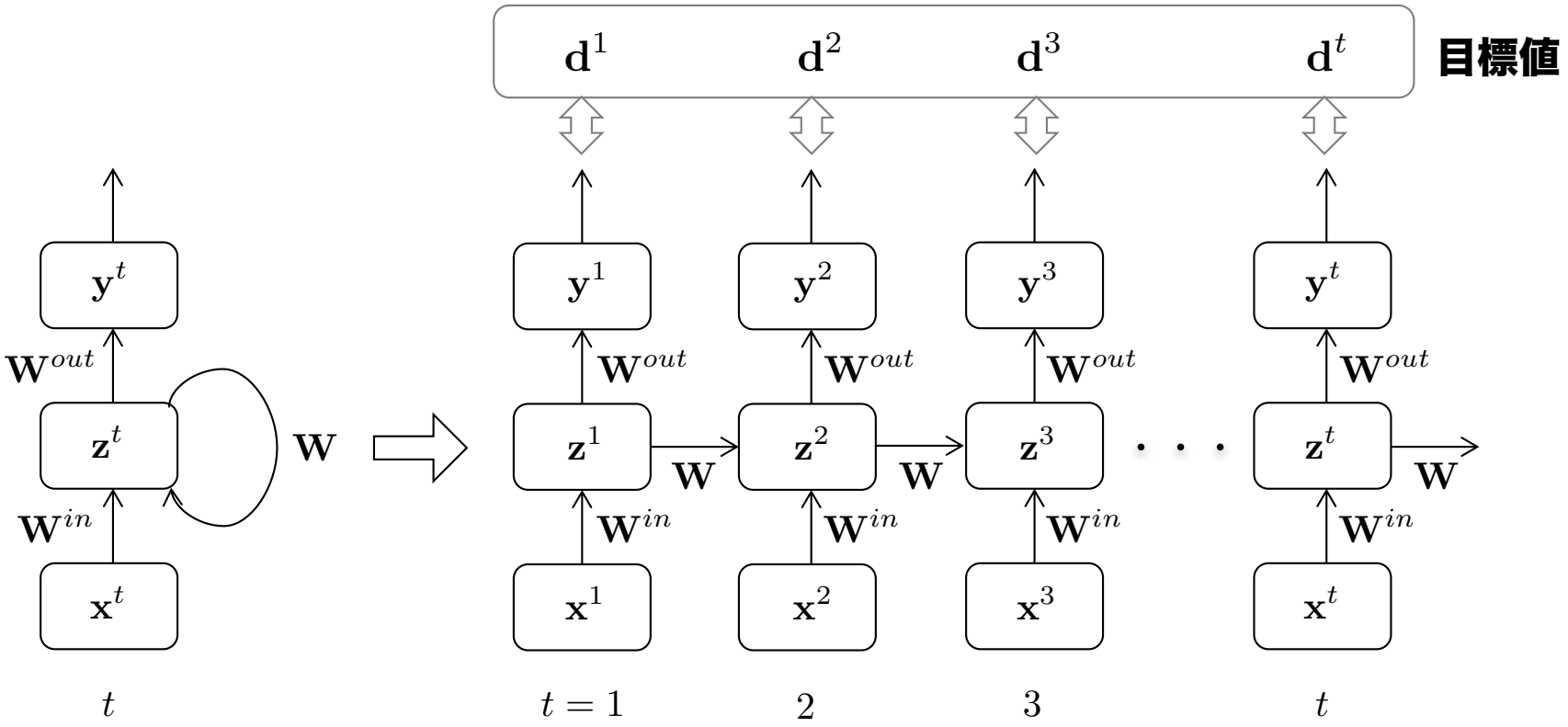
	Depth
Method	sc-inv
Liu indoor	0.260
Liu outdoor	0.341
Eigen VGG	0.225
DeMoN	**0.203**
Liu indoor	0.816
Liu outdoor	0.814
Eigen VGG	0.763
DeMoN	**0.303**
Liu indoor	0.338
Liu outdoor	0.428
Eigen VGG	0.272
DeMoN	**0.134**
Liu indoor	0.214
Liu outdoor	0.401
Eigen VGG	0.175
DeMoN	**0.126**
Liu indoor	0.210
Liu outdoor	0.421
Eigen VGG	**0.148**
DeMoN	0.180





RNN＝ディープフィードフォワードネット

- 時間方向に展開 → フィードフォワードネットになる

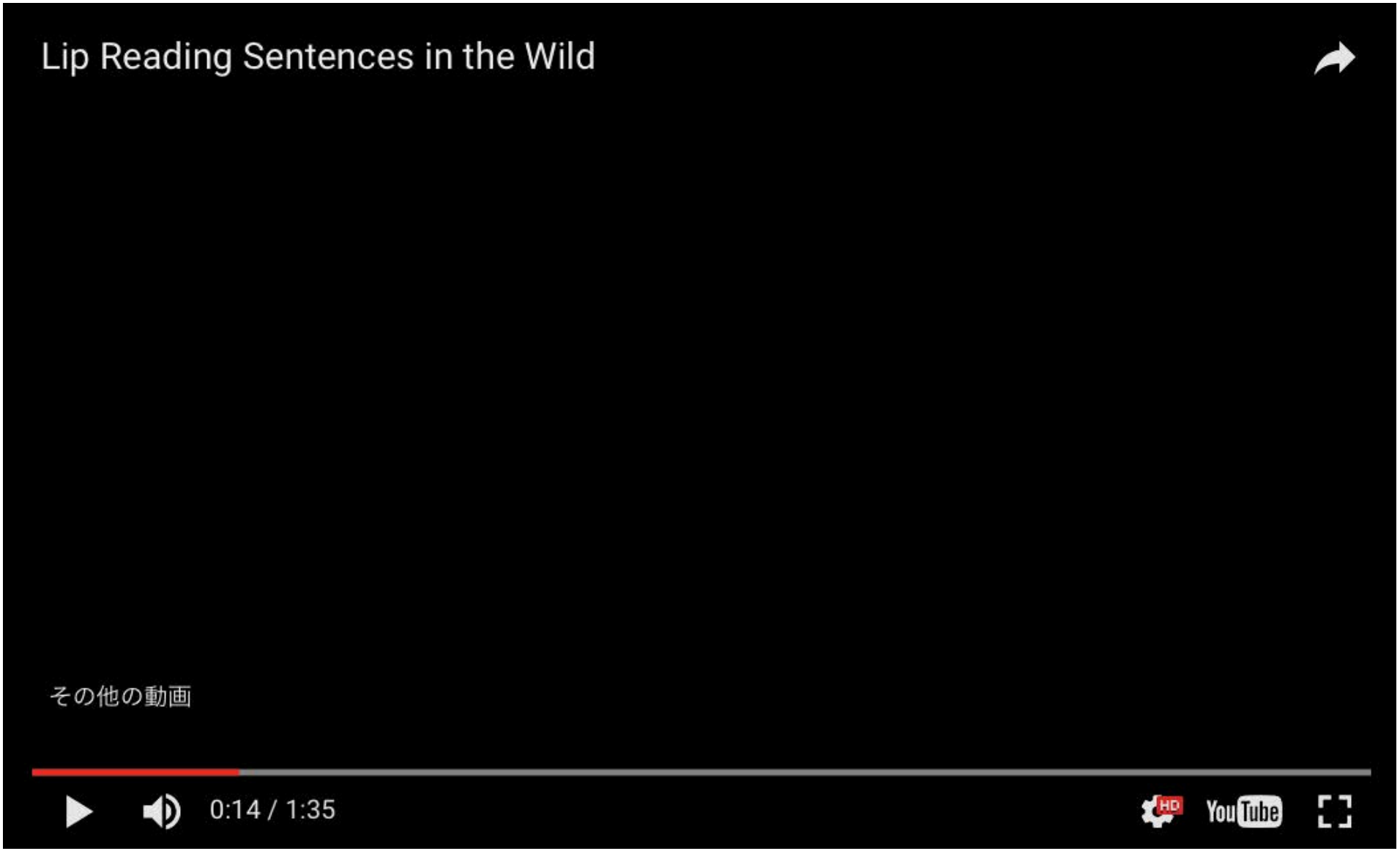


$$E(\mathbf{w}) = \sum_n \sum_t \sum_k d_{nk}^t \log y_k^t(\mathbf{x}_n; \mathbf{w})$$

（クラス分類の場合の誤差関数）

Lip reading

Chung+, Lip Reading Sentences in the Wild, arXiv, Nov. 2016



https://youtu.be/5aogzAUPilE

Lip reading

Chung+, Lip Reading Sentences in the Wild, arXiv, Nov. 2016

Lip reading

Chung+, Lip Reading Sentences in the Wild, arXiv, Nov. 2016

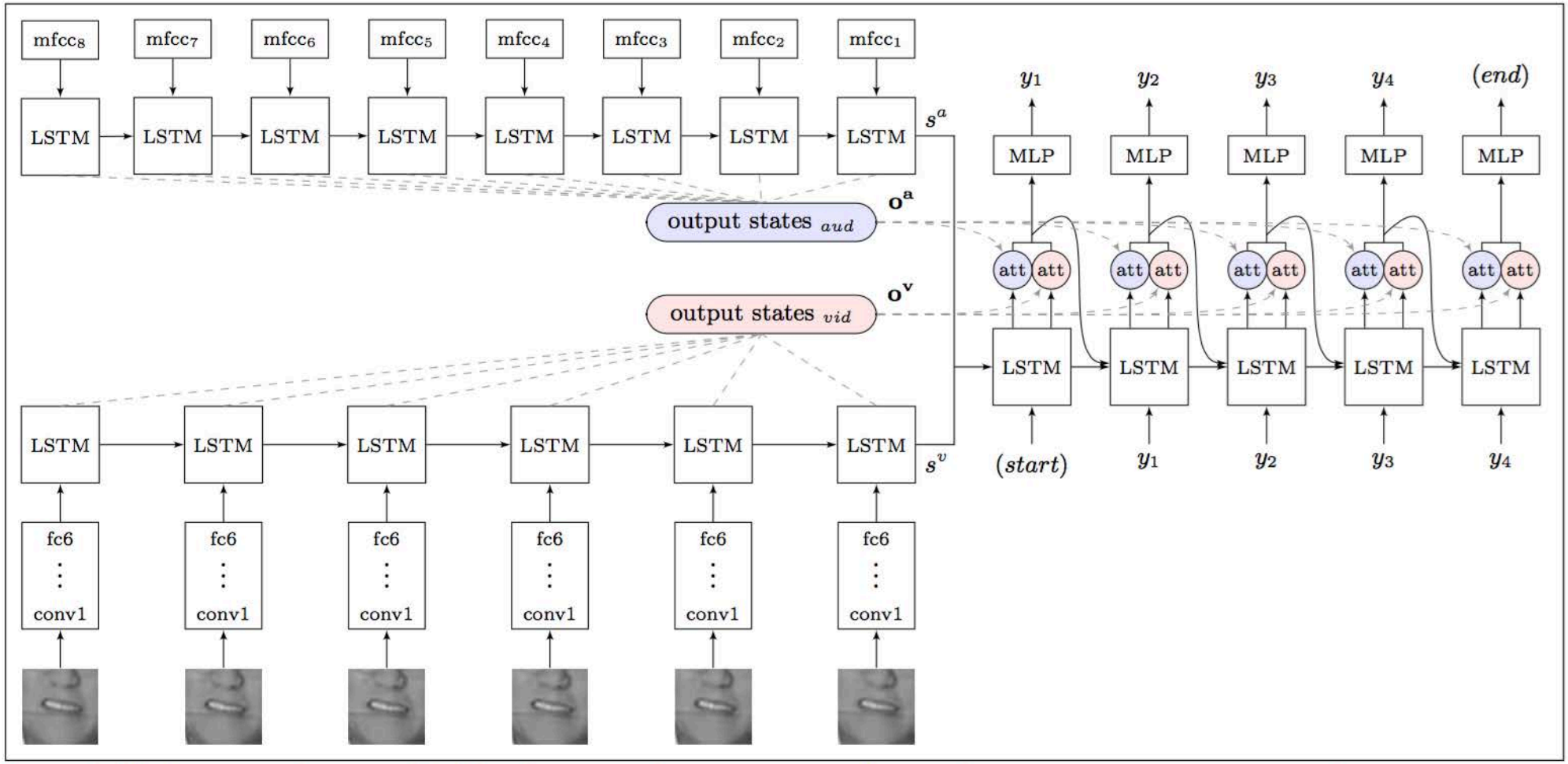
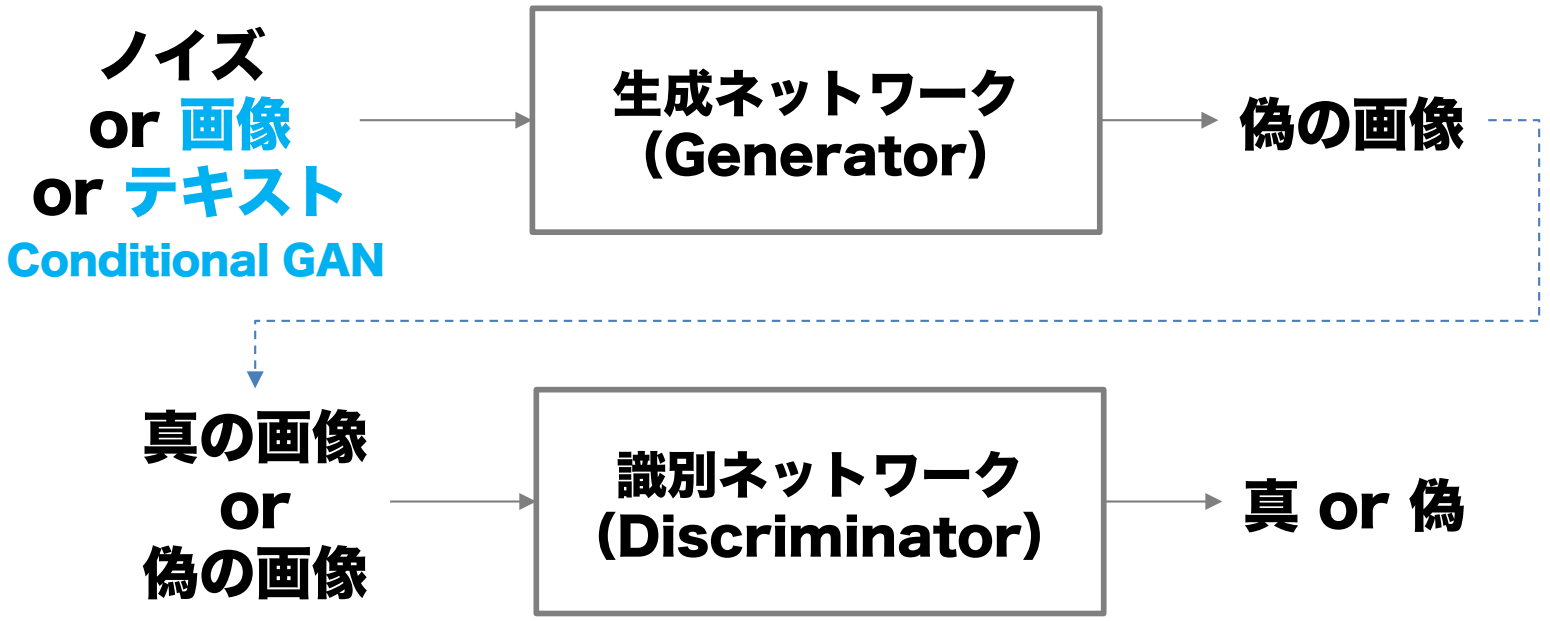


Figure 1. *Watch, Listen, Attend and Spell* architecture. At each time step, the decoder outputs a character y_i, as well as two attention vectors. The attention vectors are used to select the appropriate period of the input visual and audio sequences.

Generative Adversarial Network (GAN)

- 二つのネットワークを競争的(adversarial）に学習
  - Gはなるべく本物らしい画像を生成してDを騙すように，Dは本物と偽物を正確に見極めてGに騙されないように



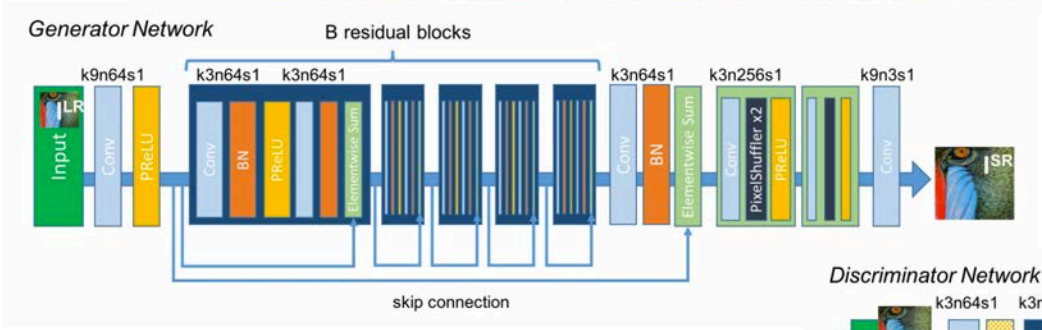
$$\min_G \max_D V(D,G)$$

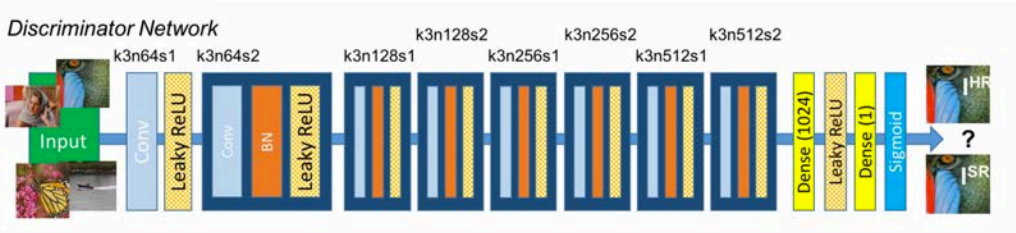
$$V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z))]$$

超解像

Ledig+, Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, arXiv 2016

- 真の画像と生成画像の差





GANによる画像変換

Isola+, Image-to-Image Translation with Conditional Adversarial Networks, arXiv 2016

- Encoder-Decoder形
  スキップコネクションあり
  Generator