

# AI Friendly Theme

Your Name (2025/01/01)

### 1. 教師あり学習 (Supervised Learning)

- 入力から出力への関数を学習
- 例：分類、回帰

### 2. 教師なし学習 (Unsupervised Learning)

- データの構造・パターンを発見
- 例：クラスタリング、次元削減

### 3. 強化学習 (Reinforcement Learning)

- 環境との相互作用から報酬を最大化する方策を学習
- 例：ロボット制御、ゲームAI

**教師あり学習:** 入力  $x \in \mathcal{X}$  から出力  $y \in \mathcal{Y}$  への関数  $f$  を学習するタスク

- 入力  $x$  は特徴量(features)または予測変数とも呼ばれる
- 出力  $y$  はラベル、ターゲット、または応答とも呼ばれる
- 経験  $E$  は入力-出力ペアの集合  $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$

## 引用の例: ノーフリーランチ定理

すべてのモデルは間違っているが、いくつかのモデルは有用である。  
— ジョージ・ボックス

あらゆる種類の問題に対して最適に機能する単一の最良モデルは存在しません

### 適切なモデルの選び方

1. ドメイン知識に基づく
2. 交差検証やベイズ法などのモデル選択技術を使用

教師あり学習の目標は、任意の入力に対してラベルを確実に予測するモデルを自動的に作成すること

### ■ 一般的な損失関数

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{n=1}^N \ell(y_n, f(x_n; \theta))$$

### ■ モデルトレーニングの目標: 経験的リスク最小化

$$\hat{\theta} = \operatorname{argmin}_{\theta} \mathcal{L}(\theta)$$

## 畳み込みと似た操作で、重みベクトルを反転させない

$$[w * x](i) = w_{-L}x_{i-L} + \cdots + w_{-1}x_{i-1} + w_0x_i + w_1x_{i+1} + \cdots + w_Lx_{i+L}$$

簡略化すると（負のインデックスを排除）：

$$[w \circledast x](i) = \sum_{u=0}^{L-1} w_u x_{i+u}$$

ディープラーニングの文献では通常、**畳み込み**は**相互相関**を意味します

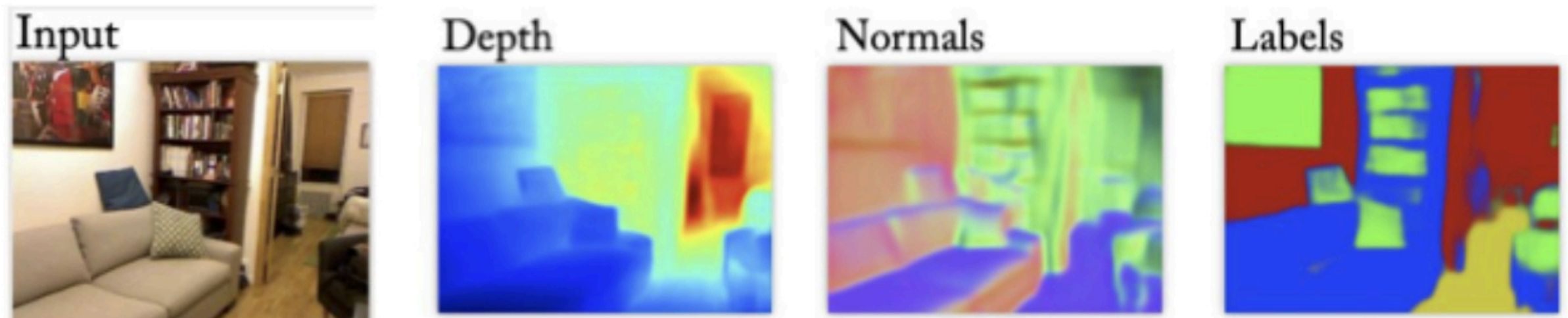


図14.31: マルチタスク密度予測問題の図解。[EF15]の図1より。Rob Fergus氏の許可を得て使用。

**分類問題:** 出力空間はクラス集合  $\mathcal{Y} = \{1, 2, \dots, C\}$

- クラス = 順序付けられていない相互排他的なラベル
- 二項分類:  $C = 2$ 、 $y \in \{0, 1\}$  または  $\{-1, +1\}$
- 多クラス分類:  $C > 2$



(a)



(b)



(c)

図1.2: アヤメの花の3種類（Setosa、Versicolor、Virginica）



## マックスプーリング

- 各領域内の最大値を出力
- エッジや特徴の存在を強調

## 平均プーリング

- 各領域内の平均値を出力
- 全体的な特徴を捉える傾向

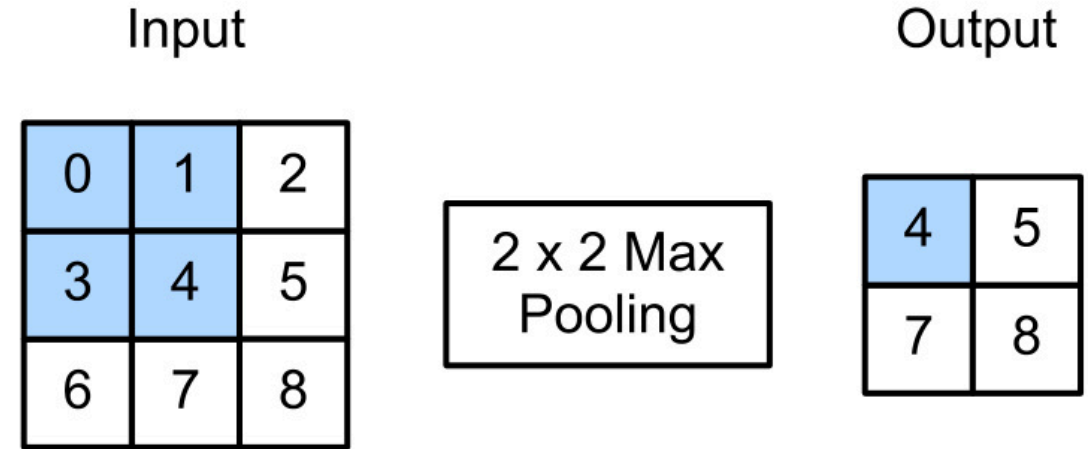


図14.12:  $2 \times 2$  フィルターとストライド1を使用したマックスプーリングの図解

# コードブロックの例

```
import numpy as np

def softmax(x):
    return np.exp(x) / np.sum(np.exp(x), axis=0)

x = np.array([1, 2, 3])
print(softmax(x))
```

インラインコードブロック `code` は背景が薄い青になります

**Table** の例です

列1	列2	列3
A1	B1	description of A1 and B1
A2	B2	description of A2 and B2
A3	B3	description of A3 and B3
A4	B4	description of A4 and B4
A5	B5	description of A5 and B5