知能プログラミング演習 I

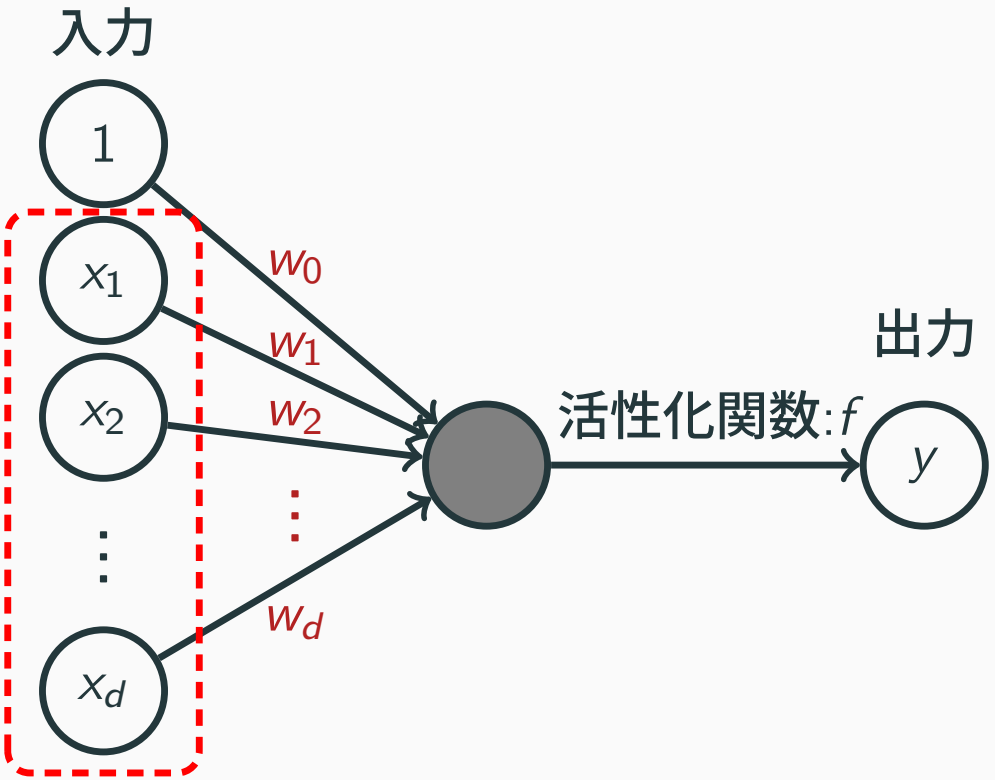
第 2 回: 3 層/深層ニューラルネットワーク

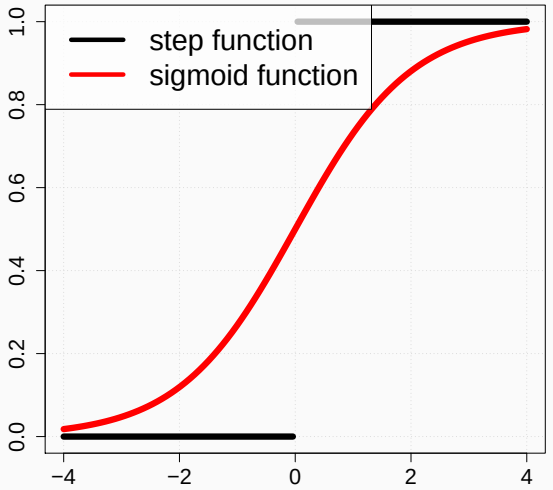
講義担当教員: 稲津 佑
スライド作成者: 稲津 佑，烏山 昌幸，田中 剛平，梅津 佑太

今日の講義内容

1. 3 層ニューラルネットワークの学習
2. 深層ニューラルネットワークの学習

前回の復習: パーセプトロン





活性化関数

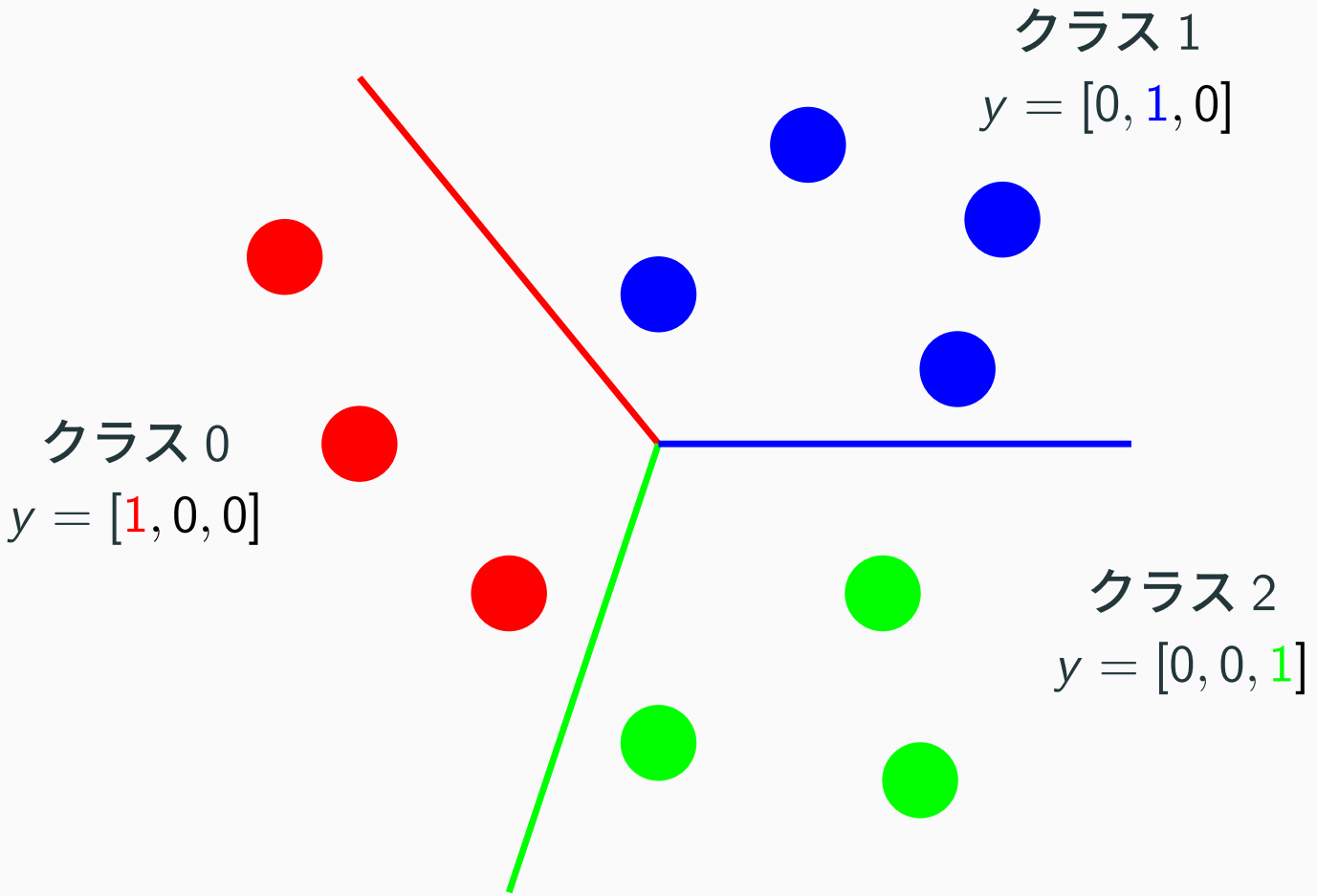
活性化関数 f を用いて

$$
\begin{aligned}
y &\approx f\left(w_0 + w_1 x_1 + \cdots + w_d x_d\right) \\
&= f\left(\boldsymbol{w}^\top \boldsymbol{x}\right)
\end{aligned}
$$

によって入出力関係をモデル化し，**確率的勾配降下法**で重みを学習

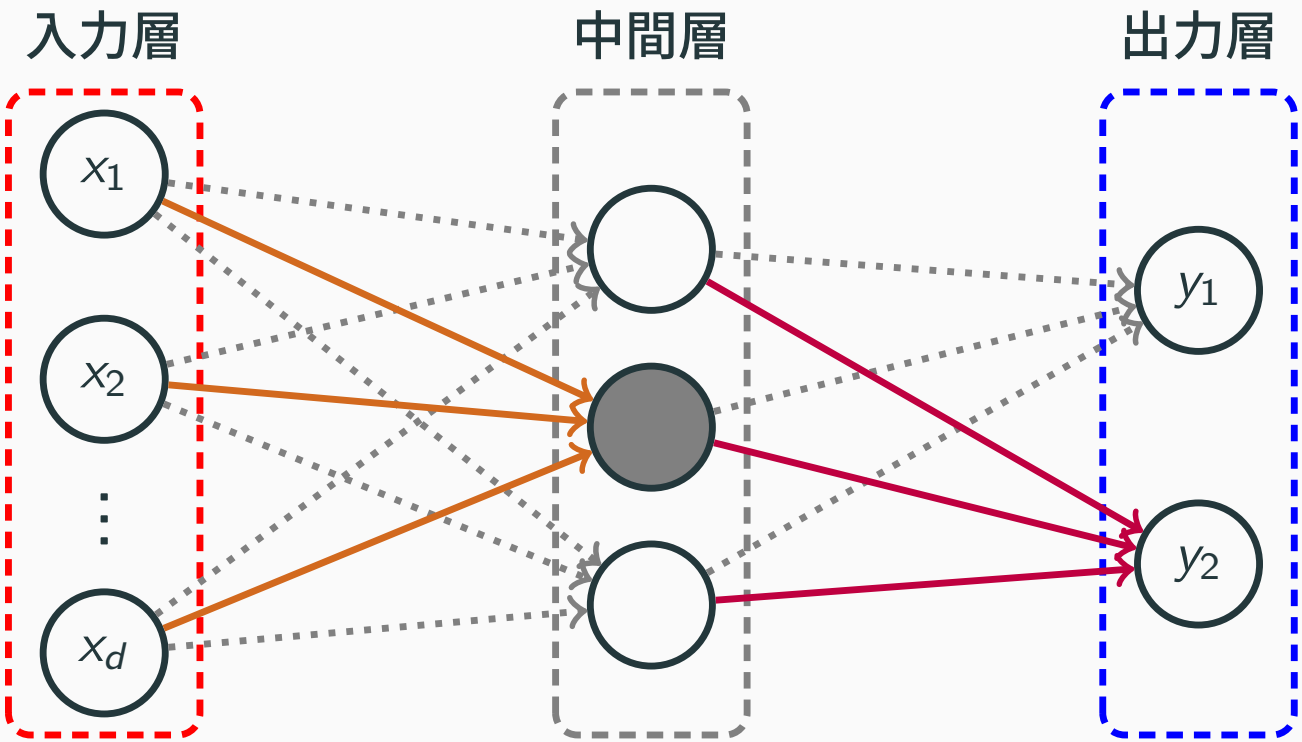
ただし，$\boldsymbol{x} = (1, x_1, \ldots, x_d)^\top, \boldsymbol{w} = (w_0, \ldots, w_d)^\top$

多クラス分類



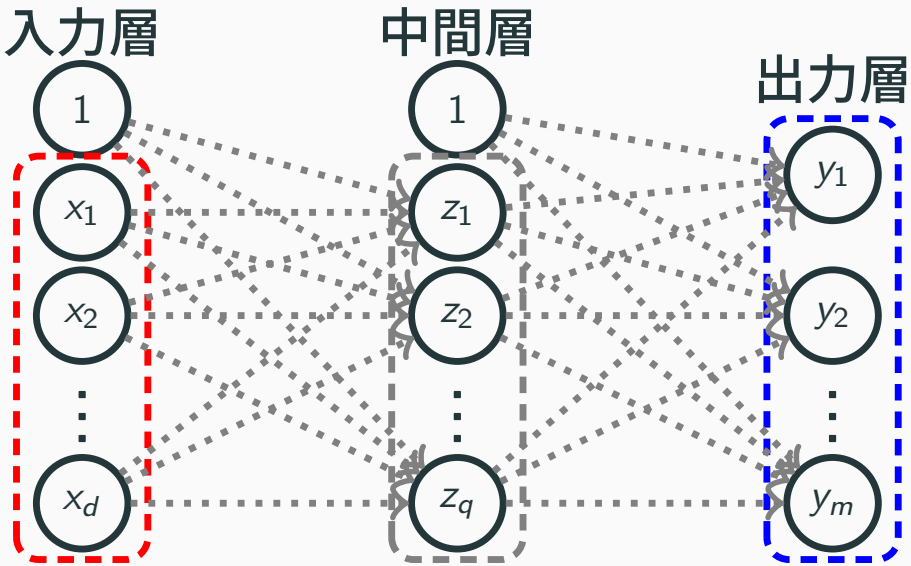
- 入力 ●, ●, ● をうまく分離する境界 (判別境界) を求める問題
- 各入力には正解ラベルがあり, ベクトルで表現する (1-of-K 表記)
- 例) 数値認識 0 から 9 までの 10 クラス分類

3 層ニューラルネットワーク



- パーセプトロンを組み合わせることで複雑なモデルを記述
  - ✓ すべての矢線には, 学習すべき重みがかかっている
  - ✓ 多クラス分類などの出力が多次元のモデルも学習可能
  - ✓ 順方向にネットワークが流れることを順伝播と呼ぶ

モデル



- 活性化関数を f (入力層 $\to$ 中間層), g (中間層 $\to$ 出力層) とする

$$z_j = f(w_{j0} + w_{j1}x_1 + \cdots + w_{jd}x_d) = f(\boldsymbol{w}_j{}^\top \boldsymbol{x}), \quad j = 1, 2, \ldots, q$$

$$y_k \approx g(v_{k0} + v_{k1}z_1 + \cdots + v_{kq}z_q) = g(\boldsymbol{v}_k{}^\top \boldsymbol{z}), \quad k = 1, 2, \ldots, m$$

ただし，$\boldsymbol{z} = (1, z_1, \ldots, z_q)^\top$ かつ

$\boldsymbol{w}_j = (w_{j0}, w_{j1}, \ldots, w_{jd})^\top$ # 入力層から中間層のユニット z_j への重み

$\boldsymbol{v}_k = (v_{k0}, v_{k1}, \ldots, v_{kq})^\top$ # 中間層から出力層のユニット y_k への重み

- 中間層にも入力に依存しない定数項があることに注意

重みの推定

- 後ほど定義する誤差関数 $E(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_q, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_m) = E(W, V)$ をできるだけ小さくするようにパラメータ W と V を学習

$$W = \begin{bmatrix} \boldsymbol{w}_1^\top \\ \boldsymbol{w}_2^\top \\ \vdots \\ \boldsymbol{w}_q^\top \end{bmatrix} = \underbrace{\begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1d} \\ w_{20} & w_{21} & \cdots & w_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ w_{q0} & w_{q1} & \cdots & w_{qd} \end{bmatrix}}_{\text{入力次元数} + 1} \Bigg\} \text{定数項除く中間層のユニット数}$$

$$V = \begin{bmatrix} \boldsymbol{v}_1^\top \\ \boldsymbol{v}_2^\top \\ \vdots \\ \boldsymbol{v}_m^\top \end{bmatrix} = \underbrace{\begin{bmatrix} v_{10} & v_{11} & \cdots & v_{1q} \\ v_{20} & v_{21} & \cdots & v_{2q} \\ \vdots & \vdots & \vdots & \vdots \\ v_{m0} & v_{m1} & \cdots & v_{mq} \end{bmatrix}}_{\text{中間層のユニット数} + 1} \Bigg\} \text{出力層のユニット数}$$

活性化関数の表記

以降では以下の省略記法を使用

- 中間層の活性化関数 f はベクトルに適用した場合，各要素に適用したものと解釈

$$\boldsymbol{z} = f(W\boldsymbol{x}) = \begin{bmatrix} f(\boldsymbol{w}_1^\top \boldsymbol{x}) \\ \vdots \\ f(\boldsymbol{w}_q^\top \boldsymbol{x}) \end{bmatrix}$$

- 一方，出力層の活性化関数 g は便宜上，m 次元ベクトルを受け取って m 次元ベクトルを返す関数として定義するが（後述），ある次元のみに適用した場合は対応する次元のスカラー出力を返すとみなす

$$g(V\boldsymbol{z}) = \begin{bmatrix} g(\boldsymbol{v}_1^\top \boldsymbol{z}) \\ \vdots \\ g(\boldsymbol{v}_m^\top \boldsymbol{z}) \end{bmatrix}$$

活性化関数の例

- 中間層でよく使われる活性化関数
  - ✓ ReLU (Rectified Linear Unit): $f(x) = \max\{0, x\}$
  - ✓ シグモイド関数: $f(x) = 1/(1 + e^{-x})$
  - ✓ ハイパボリックタンジェント: $f(x) = \tanh x = (e^{x} - e^{-x})/(e^{x} + e^{-x})$

- 出力層でよく使われる活性化関数 (便宜上こちらでは, m 次元入力 $\boldsymbol{x} \in \mathbb{R}^m$ から m 次元出力を返す関数として定義)
  - ✓ 恒等写像: 回帰問題で用いられる

$$g(\boldsymbol{x}) = \boldsymbol{x}$$

  - ✓ ソフトマックス関数: 多クラス分類問題で用いられるもので, 出力されたベクトルは, x の所属確率を表す（$[0, 1]$ 内に正規化され，足して１になる）

$$g(\boldsymbol{x}) = \frac{1}{\sum_{k=1}^{m} e^{x_k}} \begin{bmatrix} e^{x_1} \\ \vdots \\ e^{x_m} \end{bmatrix}$$

活性化関数の微分の例

- ReLU: $f(x) = \max\{0, x\}$

$$f'(x) = \begin{cases} 1 & x > 0 \\ 0 & \text{その他} \end{cases}$$

(厳密には $x = 0$ で微分不可だが, 0 とみなす)

- シグモイド関数: $f(x) = 1/(1 + e^{-x})$

$$f'(x) = f(x)(1 - f(x))$$

(前回，導出)

- ハイパボリックタンジェント: $f(x) = \tanh x$

$$f'(x) = 1 - f(x)^2$$

誤差関数の例

m 次元出力の教師データ (正解のラベルや観測した実数値) $y_1, \ldots, y_m$ とモデルの出力 $g(\boldsymbol{v}_1^\top \boldsymbol{z}), \ldots, g(\boldsymbol{v}_m^\top \boldsymbol{z})$ に対して,

- ℓ_2-誤差: 回帰問題 (g : 恒等写像)

$$E(W, V) = \sum_{j=1}^{m} (y_j - g(\boldsymbol{v}_j^\top \boldsymbol{z}))^2$$

- クロスエントロピー: 分類問題 (g : ソフトマックス関数, y は 1-of-K 表記 (「他クラス分類」のページ参照))

$$E(W, V) = -\sum_{j=1}^{m} y_j \log g(\boldsymbol{v}_j^\top \boldsymbol{z})$$

✓ 特に, $m = 2$ で $y \in \{0, 1\}$ の場合は, 整理すると前回のものに一致

勾配降下法

- **関数 $J(W)$ の行列微分**: $W = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_q)^\top \in \mathbb{R}^{q \times (d+1)}$ **としたとき**,

$$\frac{\partial J(W)}{\partial W} = \begin{bmatrix} \frac{\partial J(W)}{\partial w_{10}} & \cdots & \frac{\partial J(W)}{\partial w_{1d}} \\ \vdots & \ddots & \vdots \\ \frac{\partial J(W)}{\partial w_{q0}} & \cdots & \frac{\partial J(W)}{\partial w_{qd}} \end{bmatrix} = \begin{bmatrix} \frac{\partial J(W)}{\partial \boldsymbol{w}_1^\top} \\ \vdots \\ \frac{\partial J(W)}{\partial \boldsymbol{w}_q^\top} \end{bmatrix}$$

✓ **例**: $J(W) = \mathrm{tr}(W^\top W) = \sum_{i=1}^{p} \sum_{j=1}^{q} w_{ij}^2$ **ならば**

$$\frac{\partial J(W)}{\partial w_{ij}} = 2w_{ij} \quad \Rightarrow \quad \frac{\partial J(W)}{\partial W} = 2W$$

パラメータの更新規則

$$W^{(t+1)} = W^{(t)} - \eta_t \left. \frac{\partial E(W, V^{(t)})}{\partial W} \right|_{W=W^{(t)}}$$

$$V^{(t+1)} = V^{(t)} - \eta_t \left. \frac{\partial E(W^{(t)}, V)}{\partial V} \right|_{V=V^{(t)}}$$

誤差逆伝播法

微分を計算し整理 (導出は後で紹介) すると, 確率的勾配降下法の各ステップにおいて, パラメータは以下の通り更新すれば良い:

誤差逆伝播法の更新式

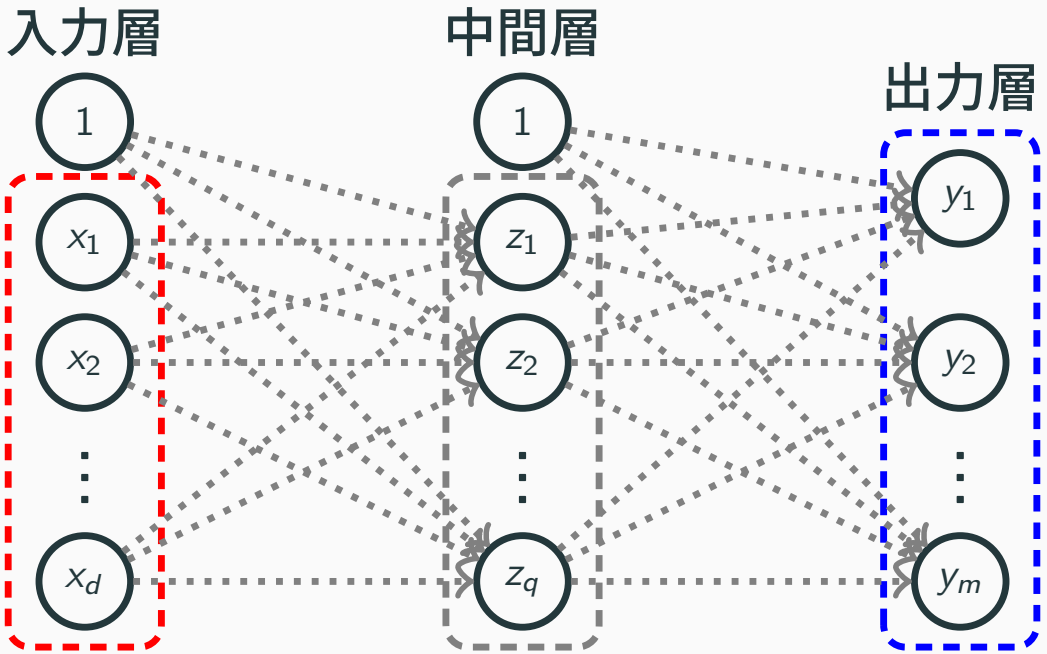
$$V^{(t+1)} = V^{(t)} - \eta_t \boldsymbol{\delta}_2 \boldsymbol{z}^{(t)\top}$$
$$W^{(t+1)} = W^{(t)} - \eta_t \boldsymbol{\delta}_1 \boldsymbol{x}^{\top}$$

ただし，
$$\boldsymbol{\delta}_2 = g(V^{(t)}\boldsymbol{z}^{(t)}) - \boldsymbol{y},$$
$$\boldsymbol{\delta}_1 = (\tilde{V}^{(t)\top}\boldsymbol{\delta}_2) \odot \nabla f(W^{(t)}\boldsymbol{x}) \quad \# \text{ 逆伝播の由来}$$

$$\nabla f(W^{(t)}\boldsymbol{x}) = \begin{bmatrix} f'(\boldsymbol{w}_1^{(t)\top}\boldsymbol{x}) \\ \vdots \\ f'(\boldsymbol{w}_q^{(t)\top}\boldsymbol{x}) \end{bmatrix}$$

であり，また，$\tilde{V}^{(t)}$ は V から一列目を除いた $m \times q$ 行列,
$\boldsymbol{v} \odot \boldsymbol{w} = (v_i w_i)_i$ は同じ長さのベクトル $\boldsymbol{v}, \boldsymbol{w}$ に対しては成分ごとの積 (アダマール積) を表す

誤差逆伝播法のイメージ



誤差逆伝播法

$$V^{(t+1)} = V^{(t)} - \eta_t \boldsymbol{\delta}_2 \boldsymbol{z}^{(t)\top}$$

$$W^{(t+1)} = W^{(t)} - \eta_t \boldsymbol{\delta}_1 \boldsymbol{x}^{\top}$$

ただし，

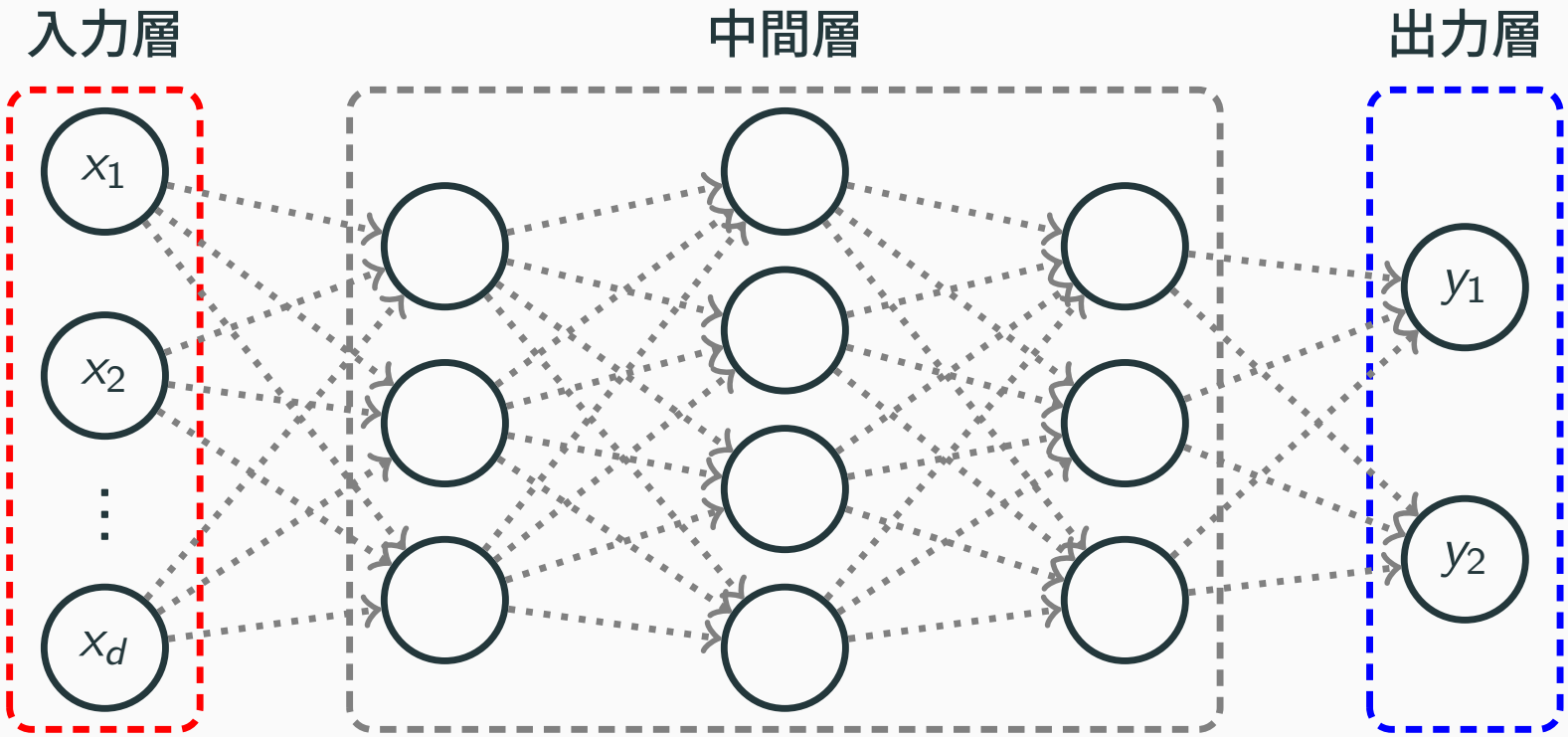
$$\boldsymbol{\delta}_2 = g(V^{(t)}\boldsymbol{z}^{(t)}) - \boldsymbol{y},$$

$$\boldsymbol{\delta}_1 = (\tilde{V}^{(t)\top}\boldsymbol{\delta}_2) \odot \nabla f(W^{(t)}\boldsymbol{x})$$

次の話題

- 深層ニューラルネットワークの学習

ディープニューラルネットワーク



- 中間層を増やす (深くする) ことで, より複雑なモデルを表現
  - ✓ 深層学習では, 入力層と出力層をつなぐ矢線上のすべての重みを学習
  - ✓ 層が深ければ深いほどパラメータ数は増加
    - 上のダイアグラムで $d = 1024$ とした場合, パラメータ数は $1024 \times 3 + 3 \times 4 + 4 \times 3 + 3 \times 2 = 3102$ 個

実社会における深層学習

2016年10月18日 07時00分 公開

自動運転技術：

ディープニューラルネットワークで自動運転向け画像認識、デンソーと東芝が共同開発

デンソーと東芝は、自動運転や高度運転支援の画像認識システムに向けた人工知能技術を共同開発する。東芝は人工知能技術を搭載した画像認識プロセッサーを2018年にサンプル出荷する目標だ。デンソーは2020年以降の実用化を目指す。

[齊藤由希，MONOist]

印刷／PDF　通知　見る　ツイート　52 シェア　52 いいね！　0 B! Bookmark　9 Pocket　G+

デンソーと東芝は2016年10月17日、自動運転や高度運転支援の画像認識システム向けの人工知能（AI）技術を共同開発すると発表した。両社で車載用プロセッサーに実装可能なDNN（Deep Neural Network、ディープニューラルネットワーク）を開発し、GPUを搭載するシステムよりも低消費電力な画像処理を実現する。

http://monoist.atmarkit.co.jp/mn/articles/1610/18/news021.html

実社会における深層学習

Talk

製造

デンソー、人とロボットの協働に向け生成AIを使った自律型ロボット制御技術を開発

Microsoft Azureの「Azure OpenAI Service」と「Azure AI Speech」でロボットの実行基盤を整備

2024年2月14日

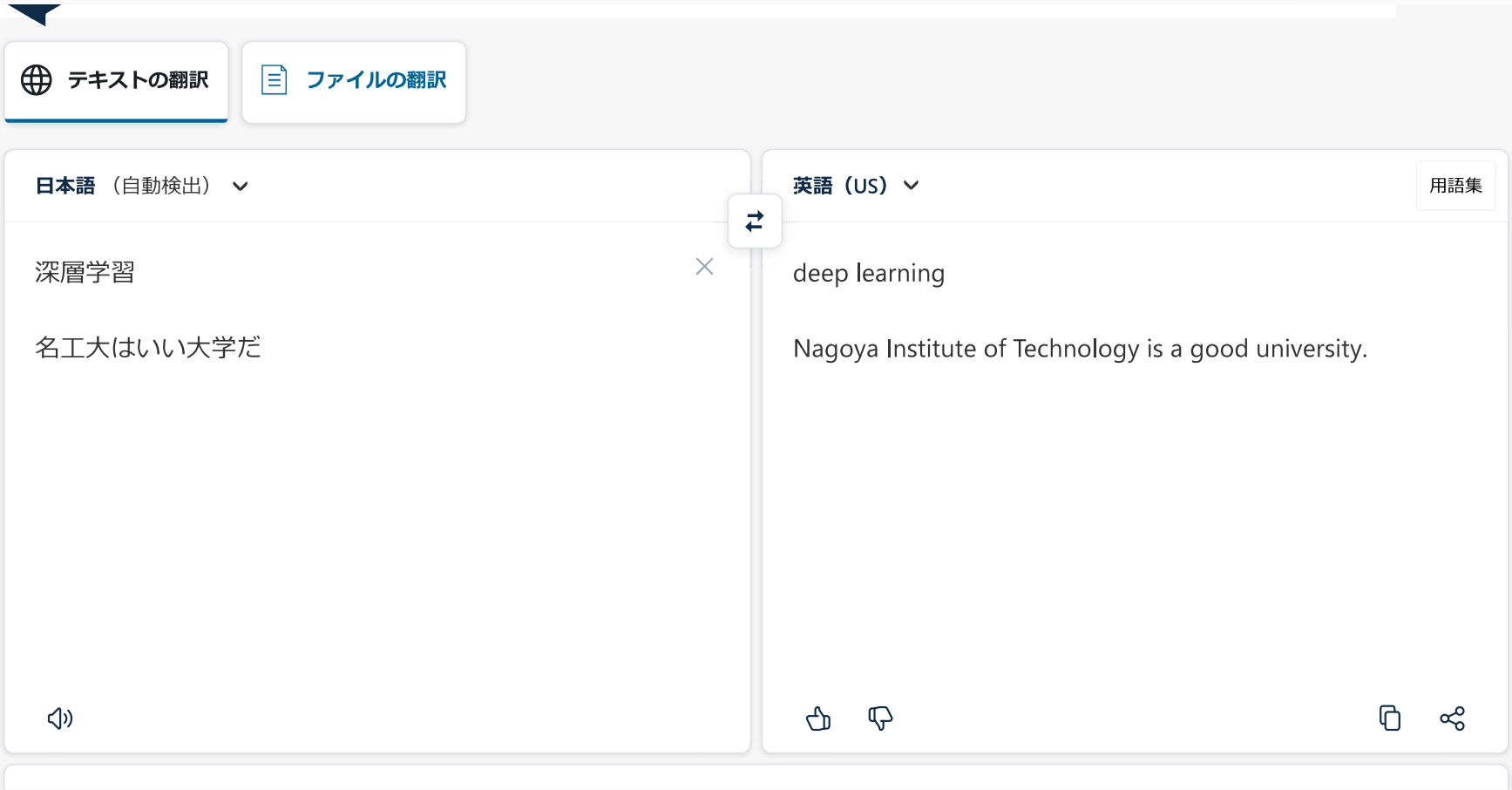
Sponsored

提供：日本マイクロソフト

「100年に1度」と言われる大変革が進む自動車業界。その中で自動車部品サプライヤー国内最大手のデンソーがモビリティ（移動）を軸にした社会的価値の創造に向けた事業ドメインの拡大を図っている。その一環として、Microsoft Azureの「Azure OpenAI Service」が提供する生成AI技術を使い、自然言語で指示できる自律型ロボッ制御技術を開発した。中核メンバーに、生成AI技術を使ったロボット制御技術の開発の狙いや開発した仕組み、将来構想などを聞いた。（文中敬称略）

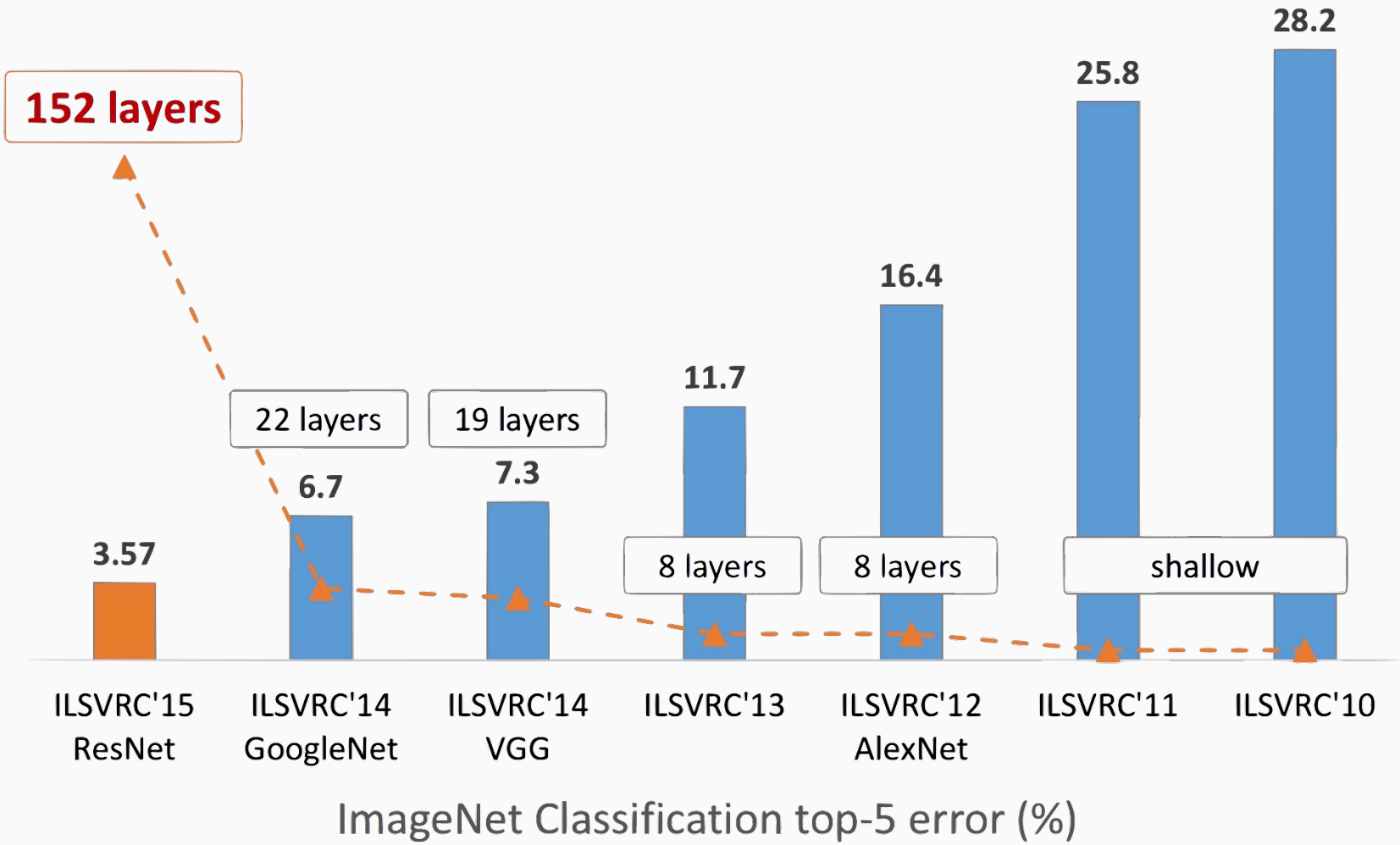
https://dcross.impress.co.jp/docs/talk/003540.html

実社会における深層学習



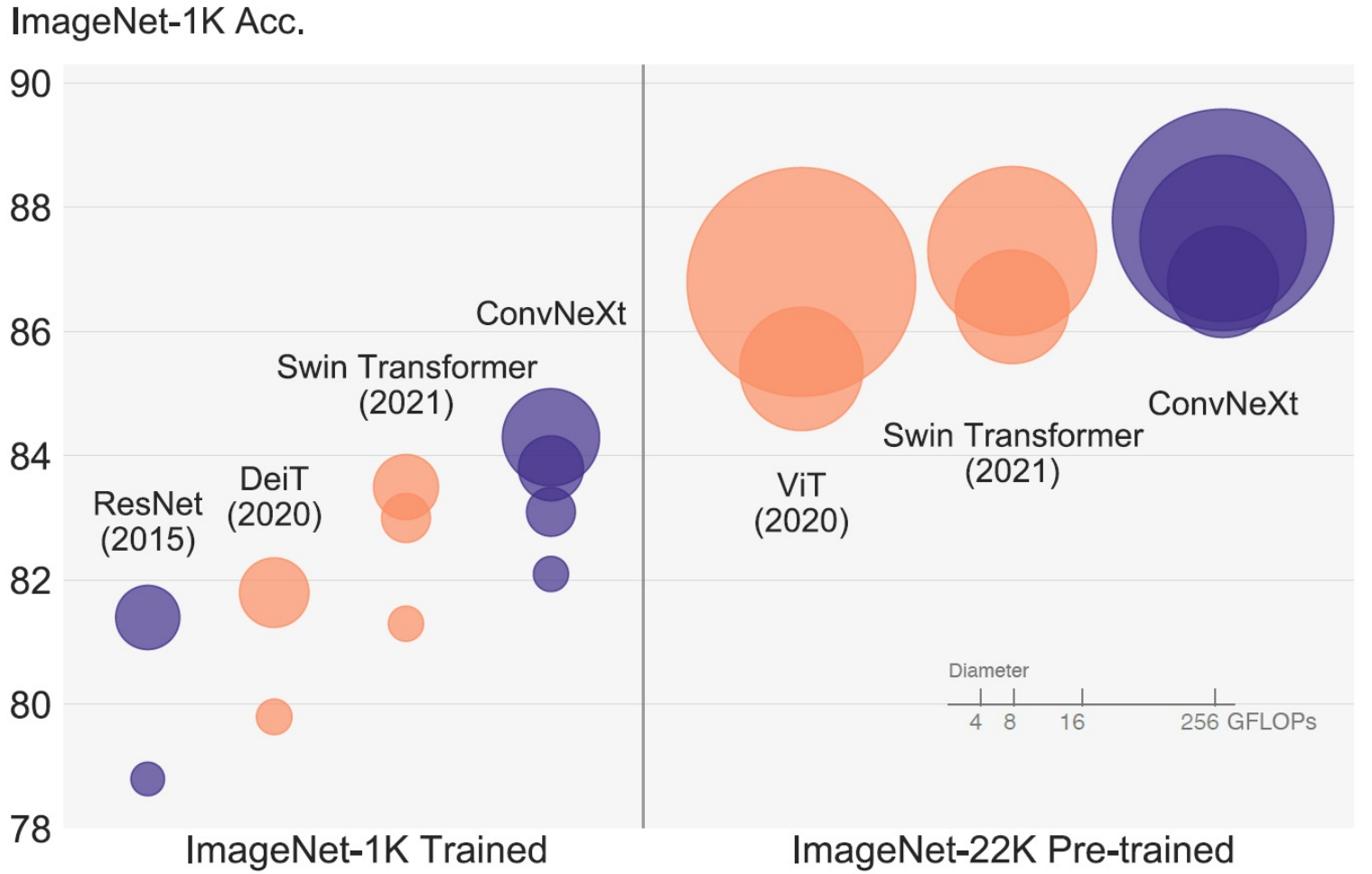
実社会における深層学習

研究としての深層学習



http://kaiminghe.com/ilsvrc15/ilsvrc2015_deep_residual_learning_kaiminghe.pdf

研究としての深層学習



ConvNeXt と ViT の性能比較

https://openaccess.thecvf.com/content/CVPR2022/papers/Liu_A_ConvNet_for_the_2020s_CVPR_2022_paper.pdf

(参考) なぜ層を深くするのか?

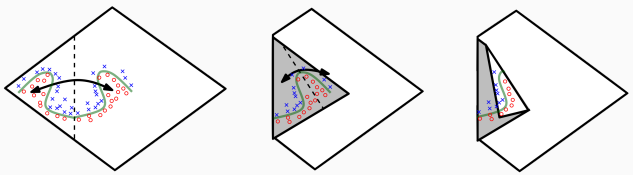
- 3 層ニューラルネットワークでも, 中間層のユニット数 $q \to \infty$ で, 緩い仮定のもと任意の関数を任意の精度で近似できる

  疑問: じゃあ, なんで深い方がいいの?

  答え: 深さに対して表現力が指数的に増大するから
- ニューラルネットの表現力: 領域を何個の多面体に分割できるか
  - ✓ q ($\geq d$): 中間層の幅 (共通), r: 層の数. ReLU による領域の分割数は

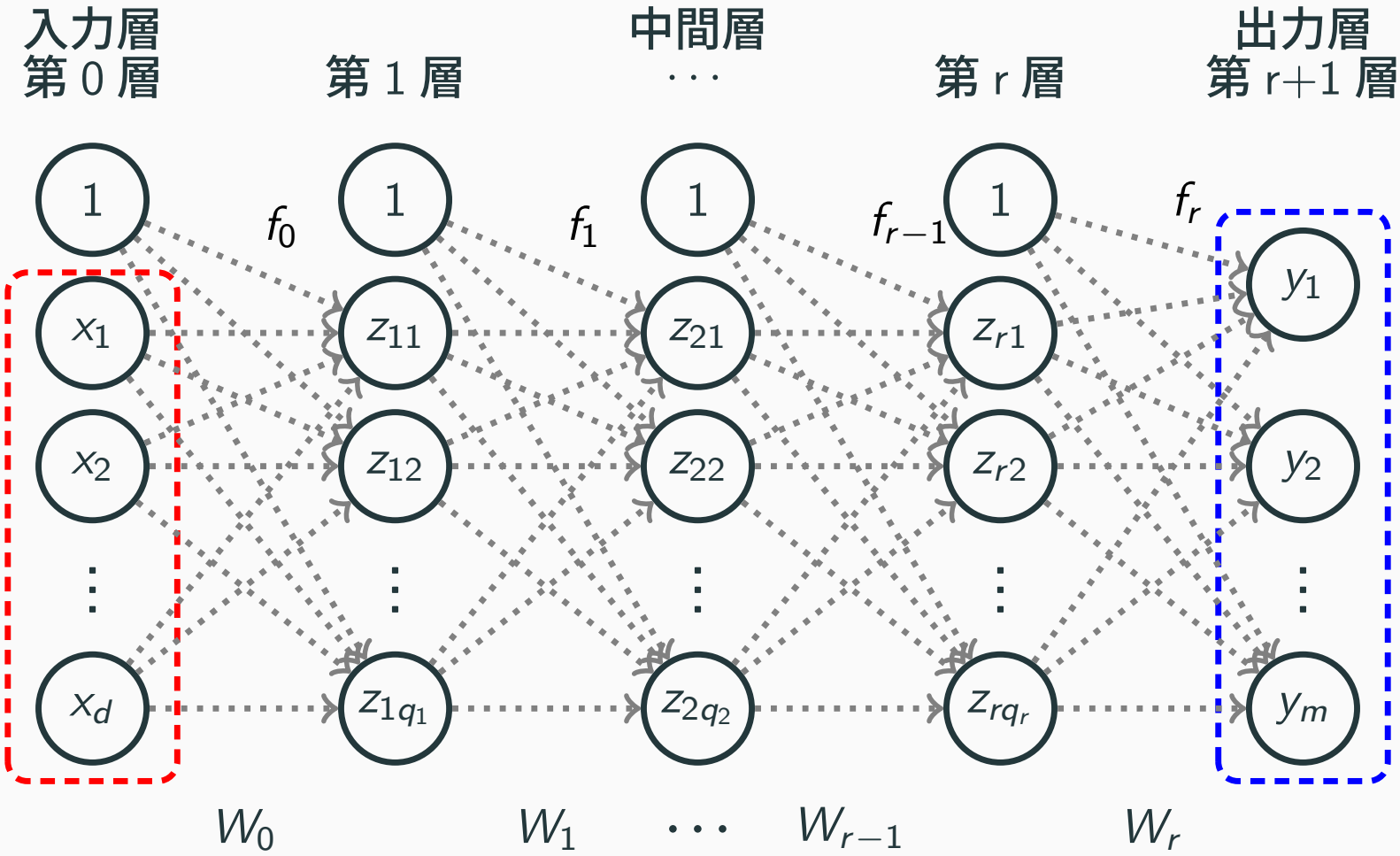
$$\Omega\left(\left(\frac{q}{d}\right)^{r-1}\sum_{j=0}^{d}\binom{q}{j}\right) \approx \Omega\left(\left(\frac{q}{d}\right)^{(r-1)d} q^d\right)$$

- 対称性の高い関数は, 多層にすることで得をする



Montúfar, et al., On the Number of Linear Regions of Deep Neural Networks, 2014

深層ニューラルネット (多層パーセプトロン)



- q_k $(k = 1, \ldots, r)$: k 層目のユニット数 (定数ユニット除いて)
- W_k $(k = 0, \ldots, r)$: k 層目の出力に対する重み行列（後述）
- f_k $(k = 0, \ldots, r)$: k 層目の出力を変換する活性化関数

パラメータ

- 最適化すべきパラメータ $W_k \in \mathbb{R}^{q_{k+1} \times (q_k+1)}$:

$$W_k = \underbrace{\left[\begin{array}{c} \boldsymbol{w}_{k1}^\top \\ \vdots \\ \boldsymbol{w}_{kq_{k+1}}^\top \end{array}\right]}_{k \text{ 番目の層のユニット数 } (q_k)+1} \Bigg\} \text{定数ユニット以外の } k+1 \text{ 番目の層のユニット数 } q_{k+1}$$

  ただし，$k = 0, 1, \ldots, r$，かつ $q_0 = d, q_{r+1} = m$ とする

- このとき，$\boldsymbol{z}_0' = \boldsymbol{x}$ かつ $\boldsymbol{z}_k = \begin{bmatrix} 1 \\ \boldsymbol{z}_k' \end{bmatrix}$ $(k = 0, \ldots, r)$ とすると，順伝播は以下のように表現できる [1]

$$\begin{aligned} \boldsymbol{z}_{k+1}' &= f_k(W_k \boldsymbol{z}_k), \qquad k = 0, \ldots, r-1, \\ \boldsymbol{y} &\approx f_r(W_r \boldsymbol{z}_r) \end{aligned}$$

[1] これまで同様，活性化関数 f_k をベクトルに適用した場合は，各要素に f_k を適用する

順伝播の流れ

$$
\begin{aligned}
z'_0 &\leftarrow x, & z_0 &\leftarrow \begin{bmatrix} 1 \\ z'_0 \end{bmatrix} \\
z'_1 &\leftarrow f_0(W_0 z_0), & z_1 &\leftarrow \begin{bmatrix} 1 \\ z'_1 \end{bmatrix} \\
z'_2 &\leftarrow f_1(W_1 z_1), & z_2 &\leftarrow \begin{bmatrix} 1 \\ z'_2 \end{bmatrix} \\
&\vdots & &\vdots \\
z'_r &\leftarrow f_{r-1}(W_{r-1} z_{r-1}), & z_r &\leftarrow \begin{bmatrix} 1 \\ z'_r \end{bmatrix}
\end{aligned}
$$

最後の出力： $f_r(W_r z_r)$

誤差関数

誤差関数はこれまでと同じ．
ただし，最適化するパラメータが多い $W_0, \ldots, W_r$

- 回帰問題: 最後の中間層から出力層への活性化関数を恒等写像 $f_r(W_r \boldsymbol{z}_r) = W_r \boldsymbol{z}_r$ とし, 誤差関数は最小二乗誤差を用いる:

$$E(W_0, \ldots, W_r) = \frac{1}{2}\|\boldsymbol{y} - W_r \boldsymbol{z}_r\|^2 = \frac{1}{2}\sum_{j=1}^{m}(y_j - \boldsymbol{w}_{rj}^\top \boldsymbol{z}_r)^2$$

- 分類問題: 最後の中間層から出力層への活性化関数をソフトマックス関数 $f_r(W_r \boldsymbol{z}_r) = e^{W_r \boldsymbol{z}_r}/(\sum_{i=1}^{m} e^{\boldsymbol{w}_{ri}^\top \boldsymbol{z}_r})$ とし [2], 誤差関数はクロスエントロピーを用いる:

$$E(W_0, \ldots, W_r) = -\sum_{j=1}^{m} y_j \log f_r(\boldsymbol{w}_{rj}^\top \boldsymbol{z}_r)$$

[2] $e^{W_r \boldsymbol{z}_r} = [e^{\boldsymbol{w}_{r1}^\top \boldsymbol{z}_r}, \ldots, e^{\boldsymbol{w}_r{}_m^\top \boldsymbol{z}_r}]^\top$

確率的勾配降下法

- 最適化もこれまでと同じく，学習率 η_t として，あるデータ $(\boldsymbol{y}, \boldsymbol{x})$ に対する誤差関数の値を微分して以下のように更新 (全ての層のパラメタを更新する):

パラメータの更新規則

$$W_k^{(t+1)} = W_k^{(t)} - \eta_t \frac{\partial E(W_0^{(t)}, \ldots, W_r^{(t)})}{\partial W_k}, k = 0, \ldots, r$$

- 層が深くても，三層パーセプトロンと同じ手順で誤差逆伝播を考えることが可能
- 以降では，最終層がソフトマックスである分類問題の場合を考える

W_k に関する微分（結論のみ，導出は最後に掲載）

- $k = r$ の場合

$$\frac{\partial E(W_0, \ldots, W_r)}{\partial W_r} = \boldsymbol{\delta}_r \boldsymbol{z}_r^\top$$

  ただし，$\boldsymbol{\delta}_r = f_r(W_r \boldsymbol{z}_r) - \boldsymbol{y}$

- $k = 0, \ldots, r-1$ の場合

$$\frac{\partial E(W_0, \ldots, W_r)}{\partial W_k} = \boldsymbol{\delta}_k \boldsymbol{z}_k^\top$$

  ただし，$\boldsymbol{\delta}_k = (\tilde{W}_{k+1}^\top \boldsymbol{\delta}_{k+1}) \odot \nabla f_k(W_k \boldsymbol{z}_k)$，また，$\tilde{W}_{k+1}$ は W_{k+1} の第 1 列 (定数項に対応する部分) を除いた $q_{r+1} \times q_r$ 次元の行列，$\odot$ はベクトルの要素毎の積

逆伝播の流れ

$$
\begin{aligned}
&\boldsymbol{\delta}_r \leftarrow f_r(W_r \boldsymbol{z}_r) - \boldsymbol{y}, \\
&\boldsymbol{\delta}_{r-1} \leftarrow (\tilde{W}_r^\top \boldsymbol{\delta}_r) \odot \nabla f_{r-1}(W_{r-1} \boldsymbol{z}_{r-1}) \\
&\boldsymbol{\delta}_{r-2} \leftarrow (\tilde{W}_{r-1}^\top \boldsymbol{\delta}_{r-1}) \odot \nabla f_{r-2}(W_{r-2} \boldsymbol{z}_{r-2}) \\
&\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
&\boldsymbol{\delta}_0 \leftarrow (\tilde{W}_1^\top \boldsymbol{\delta}_1) \odot \nabla f_0(W_0 \boldsymbol{z}_0)
\end{aligned}
$$

疑似コード（課題 2 と対応）

あるエポックでの誤差逆伝播法のアルゴリズムは以下の通り．ただし，η は適当な学習率．

1. ランダムにデータを読み込む: $(\boldsymbol{y}, \boldsymbol{x}) \in \{(\boldsymbol{y}_i, \boldsymbol{x}_i) \mid i = 1, \ldots, n\}$
2. 順伝播
   - 2.1 $\boldsymbol{z}_0 \leftarrow \begin{bmatrix} 1 \\ \boldsymbol{x} \end{bmatrix}$
   - 2.2 $\boldsymbol{z}'_{k+1} \leftarrow f_k(W_k \boldsymbol{z}_k),\ \boldsymbol{z}_{k+1} \leftarrow \begin{bmatrix} 1 \\ \boldsymbol{z}'_{k+1} \end{bmatrix},$
     $\boldsymbol{u}_{k+1} \leftarrow \nabla f_k(W_k \boldsymbol{z}_k)$ （を $k = 0, \ldots, r-1$ で実行）
   - 2.3 $\boldsymbol{g} \leftarrow f_r(W_r \boldsymbol{z}_r)$ # 最終層の出力
3. 逆伝播:
   - 3.1 $\boldsymbol{\delta}_r \leftarrow \boldsymbol{g} - \boldsymbol{y}$
   - 3.2 $\boldsymbol{\delta}_k \leftarrow (\tilde{W}_{k+1}^\top \boldsymbol{\delta}_{k+1}) \odot \boldsymbol{u}_{k+1}$ $(k = 0, 1, \ldots, r-1)$
     大きい k $(k = r-1)$ から小さい k の順番で計算する必要があることに注意
4. 重みの更新: $W_k \leftarrow W_k - \eta \boldsymbol{\delta}_k \boldsymbol{z}_k^\top$ $(k = 0, 1, \ldots, r)$

課題のための準備: 分類結果の評価

- 学習後のモデルの出力 (各クラスの所属確率) が最大のクラスにテストデータを分類
  - ✓ 例えば, ソフトマックス関数の出力が $[0.8, 0.04, 0.1, 0.06]$ なら, 予測結果はクラス 0
- 混同行列 (confusion matrix) による整理
  - ✓ 与えられたデータ集合に対して，予測されたクラスと実際のクラスのそれぞれの数を集計

		予測結果			
		0	**1**	**2**	**3**
実際のクラス	**0**	10	3	3	4
	1	2	8	5	5
	2	0	5	12	3
	3	3	2	3	12

- 対角成分は分類結果の正答数を表しており, 正答率は (正答数の合計)/(データ数) で評価する
  - ✓ 上の例なら正答率は $42/80 = 52.5\%$

まとめ

- 深層ニューラルネットワーク
  - 最も単純には，層を深くしたパーセプトロン
  - 表現能力が（中間層のユニットを増やすより効率的に）向上
  - (今日は触れないが) 単純に深くする以外にも色々な接続関係が研究されている
- 誤差逆伝播によるパラメータ更新
  - 導出は次ページ以降を参照

参考 1: 誤差関数の勾配の導出 I（3 層)

中間層から出力層への活性化関数をソフトマックス関数, 誤差関数としてクロスエントロピーを考える.

- パラメータ $V = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m)^\top \in \mathbb{R}^{m \times (q+1)}$(中間層から出力層へのパラメータ) に関する誤差関数の微分は

$$\begin{aligned} \frac{\partial E(W,V)}{\partial \boldsymbol{v}_k} &= -\frac{\partial}{\partial \boldsymbol{v}_k} \sum_{i=1}^{m} y_i \log g(\boldsymbol{v}_i^\top \boldsymbol{z}) \\ &= (g(\boldsymbol{v}_k^\top \boldsymbol{z}) - y_k)\boldsymbol{z} \qquad \# \text{ 補足 1 参照} \end{aligned}$$

つまり, $g(V\boldsymbol{z}) = (g(\boldsymbol{v}_1^\top \boldsymbol{z}), \ldots, g(\boldsymbol{v}_m^\top \boldsymbol{z}))^\top$ と表記すれば,

$$\Rightarrow \frac{\partial E(W,V)}{\partial V} = \begin{bmatrix} \frac{\partial E(W,V)}{\partial \boldsymbol{v}_1^\top} \\ \vdots \\ \frac{\partial E(W,V)}{\partial \boldsymbol{v}_m^\top} \end{bmatrix} = (g(V\boldsymbol{z}) - \boldsymbol{y})\boldsymbol{z}^\top$$

参考 1: 誤差関数の勾配の導出 II（3 層)

$\boldsymbol{z} = f(W\boldsymbol{x})$ は入力層から中間層へのパラメータ W に依存するので,

$$
\begin{aligned}
\frac{\partial E(W,V)}{\partial \boldsymbol{w}_j} &= -\frac{\partial}{\partial \boldsymbol{w}_j}\sum_{i=1}^{m} y_i \log g(\boldsymbol{v}_i^\top \boldsymbol{z}) \\
&= -\sum_{i=1}^{m} y_i \frac{\partial \log g(\boldsymbol{v}_i^\top \boldsymbol{z})}{\partial \boldsymbol{w}_j} \\
&= -\sum_{i=1}^{m} y_i \sum_{i'=1}^{m} \frac{\partial \log g(\boldsymbol{v}_i^\top \boldsymbol{z})}{\partial \boldsymbol{v}_{i'}^\top \boldsymbol{z}} \frac{\partial \boldsymbol{v}_{i'}^\top \boldsymbol{z}}{\partial \boldsymbol{w}_j} \\
&= -\sum_{i'=1}^{m} \underbrace{\left(\sum_{i=1}^{m} y_i \frac{\partial \log g(\boldsymbol{v}_i^\top \boldsymbol{z})}{\partial \boldsymbol{v}_{i'}^\top \boldsymbol{z}} \right)}_{\text{補足 1 の後半 p21 と同じ形}} \frac{\partial \boldsymbol{v}_{i'}^\top \boldsymbol{z}}{\partial \boldsymbol{w}_j} \\
&= \sum_{i=1}^{m} (g(\boldsymbol{v}_i^\top \boldsymbol{z}) - y_i) \frac{\partial \boldsymbol{v}_i^\top \boldsymbol{z}}{\partial \boldsymbol{w}_j} \quad \# \text{ 上の行の } i \text{ は消えるので,} i' \text{ を } i \text{ に変更}
\end{aligned}
$$

参考 1: 誤差関数の勾配の導出 III（3 層)

ここで,

$$\begin{aligned}\frac{\partial \boldsymbol{v}_i^\top \boldsymbol{z}}{\partial \boldsymbol{w}_j} &= \frac{\partial}{\partial \boldsymbol{w}_j}\{v_{i0} + v_{i1} f(\boldsymbol{w}_1^\top \boldsymbol{x}) + \cdots + v_{iq} f(\boldsymbol{w}_q^\top \boldsymbol{x})\} \\ &= v_{ij} \frac{\partial f(\boldsymbol{w}_j^\top \boldsymbol{x})}{\partial \boldsymbol{w}_j} = v_{ij} \nabla f(\boldsymbol{w}_j^\top \boldsymbol{x}) \boldsymbol{x}\end{aligned}$$

より, $\tilde{v}_j = (v_{1j}, \ldots, v_{mj})^\top$ とすれば,

$$\frac{\partial E(W, V)}{\partial \boldsymbol{w}_j} = \sum_{i=1}^{m} (g(\boldsymbol{v}_i^\top \boldsymbol{z}) - y_i) v_{ij} \nabla f(\boldsymbol{w}_j^\top \boldsymbol{x}) \boldsymbol{x} = \underbrace{\tilde{v}_j^\top (g(V\boldsymbol{z}) - \boldsymbol{y}) \nabla f(\boldsymbol{w}_j^\top \boldsymbol{x})}_{\text{スカラー}} \boldsymbol{x}$$

参考 1: 誤差関数の勾配の導出 IV（3 層)

$\tilde{V} = (\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_q)$ を, V から 1 列目を取り除いた行列[3] として,

$$\frac{\partial E(W, V)}{\partial W} = \left[\left(\tilde{V}^\top (g(V\boldsymbol{z}) - \boldsymbol{y}) \right) \odot \nabla f(W\boldsymbol{x}) \right] \boldsymbol{x}^\top$$

となる. ただし, 同じ長さのベクトル $\boldsymbol{v}, \boldsymbol{w}$ に対して,

$$\boldsymbol{v} \odot \boldsymbol{w} = \left(v_i w_i \right)_i$$

は成分ごとの積 (アダマール積) を表す.

[3] V から切片項に対応する部分を取り除いたもの

補足 1) ソフトマックスの微分（3 層)

$\boldsymbol{u} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_m]^\top \boldsymbol{z}$ として

$$
\begin{aligned}
-\frac{\partial}{\partial v_{kj}} \sum_{i=1}^{m} y_i \log g(\boldsymbol{v}_i^\top \boldsymbol{z}) &= -\frac{\partial}{\partial \boldsymbol{u}^\top} \left(\sum_{i=1}^{m} y_i \log g(u_i) \right) \frac{\partial \boldsymbol{u}}{\partial v_{kj}} \\
&= -\frac{\partial}{\partial \boldsymbol{u}^\top} \left(\sum_{i=1}^{m} y_i \log g(u_i) \right) \boldsymbol{e}_k z_j \\
&= -\frac{\partial}{\partial u_k} \left(\sum_{i=1}^{m} y_i \log g(u_i) \right) z_j
\end{aligned}
$$

なので (ただし，$\boldsymbol{e}_k$ は k 番目の要素が 1 で他は 0 のベクトル),

$$
-\frac{\partial}{\partial \boldsymbol{v}_k} \sum_{i=1}^{m} y_i \log g(\boldsymbol{v}_i^\top \boldsymbol{z}) = -\frac{\partial}{\partial u_k} \left(\sum_{i=1}^{m} y_i \log g(u_i) \right) \boldsymbol{z}
$$

$$
\begin{aligned}
-\frac{\partial}{\partial u_k}\left(\sum_{i=1}^{m} y_i \log g(u_i)\right) &= -\sum_{i=1}^{m} \frac{y_i}{g(u_i)} \frac{\partial g(u_i)}{\partial u_k} \\
&= -\sum_{i=1}^{m} \frac{y_i}{g(u_i)} \frac{\partial}{\partial u_k}\left(\frac{\exp(u_i)}{\sum_{i'=1}^{m} \exp(u_{i'})}\right) \\
&= -\frac{y_k}{g(u_k)} \frac{\exp(u_k)(\sum_{i'=1}^{m} \exp(u_{i'})) - \exp(u_k)^2}{(\sum_{i'=1}^{m} \exp(u_{i'}))^2} \\
&\quad + \sum_{i \neq k} \frac{y_i}{g(u_i)} \frac{\exp(u_i)\exp(u_k)}{(\sum_{i'=1}^{m} \exp(u_{i'}))^2} \\
&= -\frac{y_k}{g(u_k)} \left(g(u_k) - g(u_k)^2\right) + \sum_{i \neq k} \frac{y_i}{g(u_i)} g(u_i) g(u_k) \\
&= -y_k + y_k g(u_k) + (\sum_{i \neq k} y_i) g(u_k) \\
&= g(u_k) - y_k \qquad \# \sum_{i=1}^{m} y_i = 1 \text{ より}
\end{aligned}
$$

参考 2: W_r に関する誤差関数の微分の導出 I（深層）

- 準備: ソフトマックス関数の第 j 成分を $\boldsymbol{w}_{rk}$ で微分する.

$$f_r(\boldsymbol{w}_{rj}^\top \boldsymbol{z}_r) = \frac{1}{\sum_{i=1}^{m} e^{\boldsymbol{w}_{ri}^\top \boldsymbol{z}_r}} e^{\boldsymbol{w}_{rj}^\top \boldsymbol{z}_r}$$

より, 分数関数の微分と, $\frac{\partial e^{\boldsymbol{w}_{rj}^\top \boldsymbol{z}_r}}{\partial \boldsymbol{w}_{rk}} = e^{\boldsymbol{w}_{rk}^\top \boldsymbol{z}_r} \boldsymbol{z}_r \ (j = k) \ \& \ 0 \ (j \neq k)$ などを用いて,

$$j = k \Rightarrow \frac{\partial f_r(\boldsymbol{w}_{rk}^\top \boldsymbol{z}_r)}{\partial \boldsymbol{w}_{rk}} = (1 - f_r(\boldsymbol{w}_{rk}^\top \boldsymbol{z}_r)) f_r(\boldsymbol{w}_{rk}^\top \boldsymbol{z}_r) \boldsymbol{z}_r$$

$$j \neq k \Rightarrow \frac{\partial f_r(\boldsymbol{w}_{rj}^\top \boldsymbol{z}_r)}{\partial \boldsymbol{w}_{rk}} = -f_r(\boldsymbol{w}_{rj}^\top \boldsymbol{z}_r) f_r(\boldsymbol{w}_{rk}^\top \boldsymbol{z}_r) \boldsymbol{z}_r$$

参考 2: W_r に関する誤差関数の微分の導出 II（深層）

ソフトマックス関数の微分を用いて, 誤差関数 (クロスエントロピー) を $\boldsymbol{w}_{rk}$ で微分すると,

$$\begin{aligned}\frac{\partial E(W_0,\ldots,W_r)}{\partial \boldsymbol{w}_{rk}} &= -\sum_{j=1}^{m} y_j \frac{\partial \log f_r(\boldsymbol{w}_{rj}^\top \boldsymbol{z}_r)}{\partial \boldsymbol{w}_{rk}} \\ &= -\sum_{j=1}^{m} y_j \frac{1}{f_r(\boldsymbol{w}_{rj}^\top \boldsymbol{z}_r)} \frac{\partial f_r(\boldsymbol{w}_{rj}^\top \boldsymbol{z}_r)}{\partial \boldsymbol{w}_{rk}} = (f_r(\boldsymbol{w}_{rk}^\top \boldsymbol{z}_r) - y_k)\boldsymbol{z}_r\end{aligned}$$

となる [4]. $W_r = (\boldsymbol{w}_{r1},\ldots,\boldsymbol{w}_{rm})^\top$ であったから,

$$\begin{aligned}\frac{\partial E(W_0,\ldots,W_r)}{\partial W_r} &= \left(\frac{\partial E(W_0,\ldots,W_r)}{\partial \boldsymbol{w}_{r1}},\ldots,\frac{\partial E(W_0,\ldots,W_r)}{\partial \boldsymbol{w}_{rm}}\right)^\top \\ &= (f_r(W_r\boldsymbol{z}_r) - \boldsymbol{y})\boldsymbol{z}^\top\end{aligned}$$

[4] $\boldsymbol{y}$ は正解ラベルで 1, それ以外は 0 なので, $\boldsymbol{y}$ の要素をすべて足すと 1 になることに注意!

参考 2: W_{r-1} に関する誤差関数の微分 I（深層）

同様に, 誤差関数を $\boldsymbol{w}_{r-1,k}$ $(k = 1, \ldots, q_r)$ で微分すると,

$$\begin{aligned}
\frac{\partial E(W_0, \ldots, W_r)}{\partial \boldsymbol{w}_{r-1,k}} &= \sum_{j=1}^{m} \frac{\partial E(W_0, \ldots, W_r)}{\partial \boldsymbol{w}_{rj}^\top \boldsymbol{z}_r} \frac{\partial \boldsymbol{w}_{rj}^\top \boldsymbol{z}_r}{\partial \boldsymbol{w}_{r-1,k}} \\
&= \sum_{j=1}^{m} (f_r(\boldsymbol{w}_{rj}^\top \boldsymbol{z}_r) - y_j) \frac{\partial \boldsymbol{w}_{rj}^\top (1, f_{r-1}(W_{r-1}\boldsymbol{z}_{r-1})^\top)}{\partial \boldsymbol{w}_{r-1,k}} \\
&= \sum_{j=1}^{m} (f_r(\boldsymbol{w}_{rj}^\top \boldsymbol{z}_r) - y_j) w_{rjk} \nabla f_{r-1}(\boldsymbol{w}_{r-1,k}^\top \boldsymbol{z}_{r-1}) \boldsymbol{z}_{r-1}
\end{aligned}$$

ここで, $\tilde{\boldsymbol{w}}_{rk} = (w_{r1k}, \ldots, w_{rmk})^\top$ は W_r の k 列目なので,

$$\frac{\partial E(W_0, \ldots, W_r)}{\partial \boldsymbol{w}_{r-1,k}} = (f_r(W_r \boldsymbol{z}_r) - \boldsymbol{y})^\top \tilde{\boldsymbol{w}}_{rk} \nabla f_{r-1}(\boldsymbol{w}_{r-1,k}^\top \boldsymbol{z}_{r-1}) \boldsymbol{z}_{r-1}$$

参考 2: W_{r-1} に関する誤差関数の微分 II（深層）

したがって,

$$\begin{aligned}
\frac{\partial E(W_0,\ldots,W_r)}{\partial W_{r-1}} &= \left(\frac{\partial E(W_0,\ldots,W_r)}{\partial \boldsymbol{w}_{r-1,1}},\ldots,\frac{\partial E(W_0,\ldots,W_r)}{\partial \boldsymbol{w}_{r-1,q_r}}\right)^\top \\
&= \begin{pmatrix} \tilde{\boldsymbol{w}}_{r1}^\top (f_r(W_r\boldsymbol{z}_r)-\boldsymbol{y})\nabla f_{r-1}(\boldsymbol{w}_{r-1,1}^\top \boldsymbol{z}_{r-1})\boldsymbol{z}_{r-1}^\top \\ \vdots \\ \tilde{\boldsymbol{w}}_{rq_r}^\top (f_r(W_r\boldsymbol{z}_r)-\boldsymbol{y})\nabla f_{r-1}(\boldsymbol{w}_{r-1,q_r}^\top \boldsymbol{z}_{r-1})\boldsymbol{z}_{r-1}^\top \end{pmatrix} \\
&= \left[\tilde{W}_r^\top (f_r(W_r\boldsymbol{z}_r)-\boldsymbol{y}) \odot \nabla f_{r-1}(W_{r-1}\boldsymbol{z}_{r-1})\right]\boldsymbol{z}_{r-1}^\top
\end{aligned}$$

ただし, $\tilde{W}_r$ は W_r の第 1 列 (切片項に対応する部分) を除いた $q_{r+1}\times q_r$ 次元の行列.

参考 3: 一般の誤差関数に対する誤差逆伝播法（深層）

より一般に, $u_{kl} = \boldsymbol{w}_{kl}^\top \boldsymbol{z}_k$ $(l = 1, \ldots, q_{k+1})$ とすれば,

$$\begin{aligned}\frac{\partial E(W_0, \ldots, W_r)}{\partial u_{kl}} &= \sum_{j=1}^{q_{k+2}} \frac{\partial E(W_0, \ldots, W_r)}{\partial u_{k+1,j}} \frac{\partial u_{k+1,j}}{\partial u_{kl}} \\ &= \sum_{j=1}^{q_{k+2}} \frac{\partial E(W_0, \ldots, W_r)}{\partial u_{k+1,j}} w_{k+1,jl} \nabla f_k(u_{kl})\end{aligned}$$

なので, $\delta_{kl} = \partial E(W_0, \ldots, W_r)/\partial u_{kl}$ として, 漸化式

$$\delta_{kl} = \sum_{j=1}^{q_{k+2}} \delta_{k+1,j} w_{k+1,jl} \nabla f_k(u_{kl})$$

が得られる. このとき, $\boldsymbol{\delta}_k = (\delta_{k1}, \ldots, \delta_{kq_{k+1}})^\top$ は, W_{k+1} の第 1 列を除いた $q_{k+2} \times q_{k+1}$ 次元の行列を用いて, 以下のように書ける:

$$\boldsymbol{\delta}_k = \tilde{W}_{k+1}^\top \boldsymbol{\delta}_{k+1} \odot \nabla f_k(W_k \boldsymbol{z}_k), \qquad (k = 0, 1, \ldots, r-1)$$

参考 3: 一般の誤差関数に対する誤差逆伝播法（深層）

したがって, 第 k 層において, パラメータに関する微分は,

$$\frac{\partial E(W_0, \ldots, W_r)}{\partial \boldsymbol{w}_{kl}} = \frac{\partial E(W_0, \ldots, W_r)}{\partial u_{kl}} \frac{\partial u_{kl}}{\partial \boldsymbol{w}_{kl}} = \delta_{kl} \boldsymbol{z}_k$$

より,

$$\frac{\partial E(W_0, \ldots, W_r)}{\partial W_k} = \begin{pmatrix} \delta_{k1} \boldsymbol{z}_k^\top \\ \vdots \\ \delta_{kq_{k+1}} \boldsymbol{z}_k^\top \end{pmatrix} = \left\{ \tilde{W}_{k+1}^\top \boldsymbol{\delta}_{k+1} \odot \nabla f_k(W_k \boldsymbol{z}_k) \right\} \boldsymbol{z}_k^\top$$

したがって, パラメータの更新式は

$$\begin{aligned} W_k &\leftarrow W_k - \eta \left\{ \tilde{W}_{k+1}^\top \boldsymbol{\delta}_{k+1} \odot \nabla f_k(W_k \boldsymbol{z}_k) \right\} \boldsymbol{z}_k^\top \\ &= W_k - \eta \boldsymbol{\delta}_k \boldsymbol{z}_k^\top \end{aligned}$$