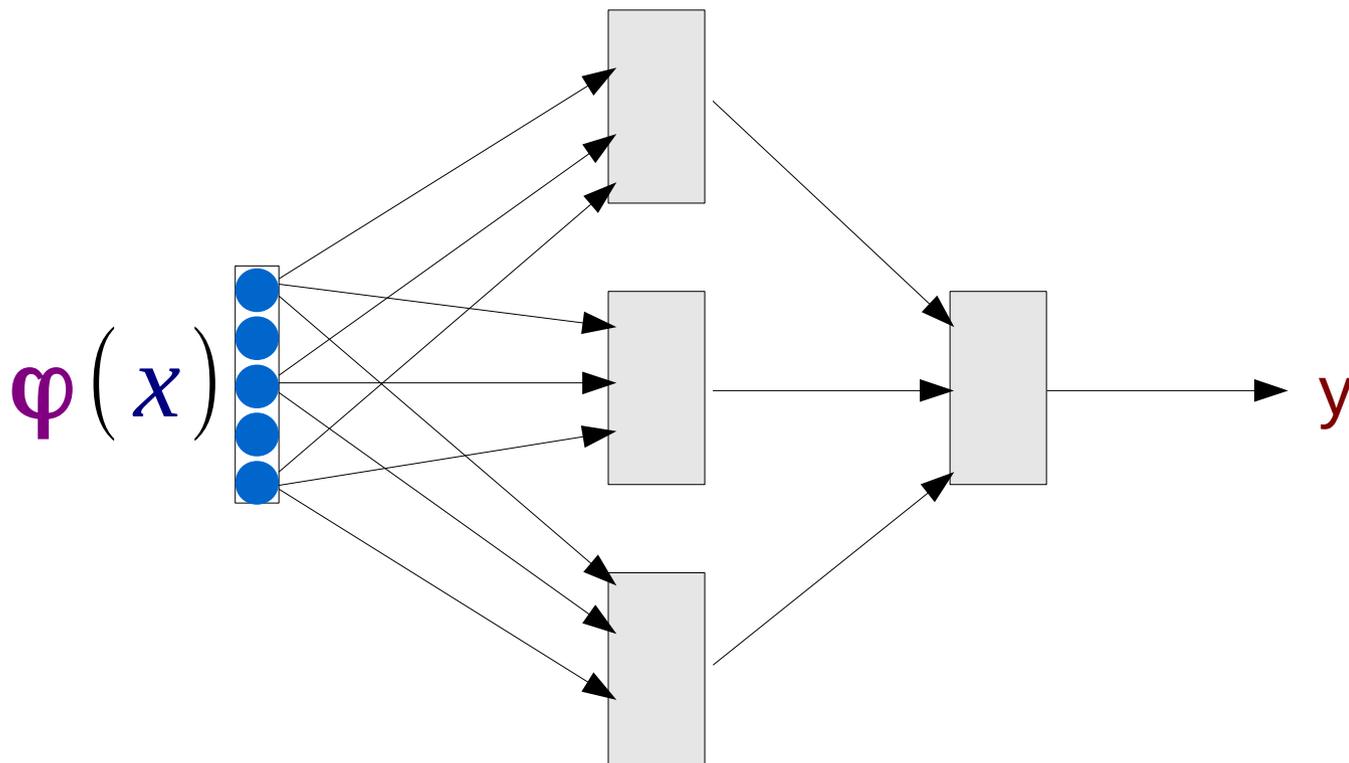


リカレントニューラルネット

Graham Neubig
奈良先端科学技術大学院大学 (NAIST)

フィードフォワード ニューラルネット

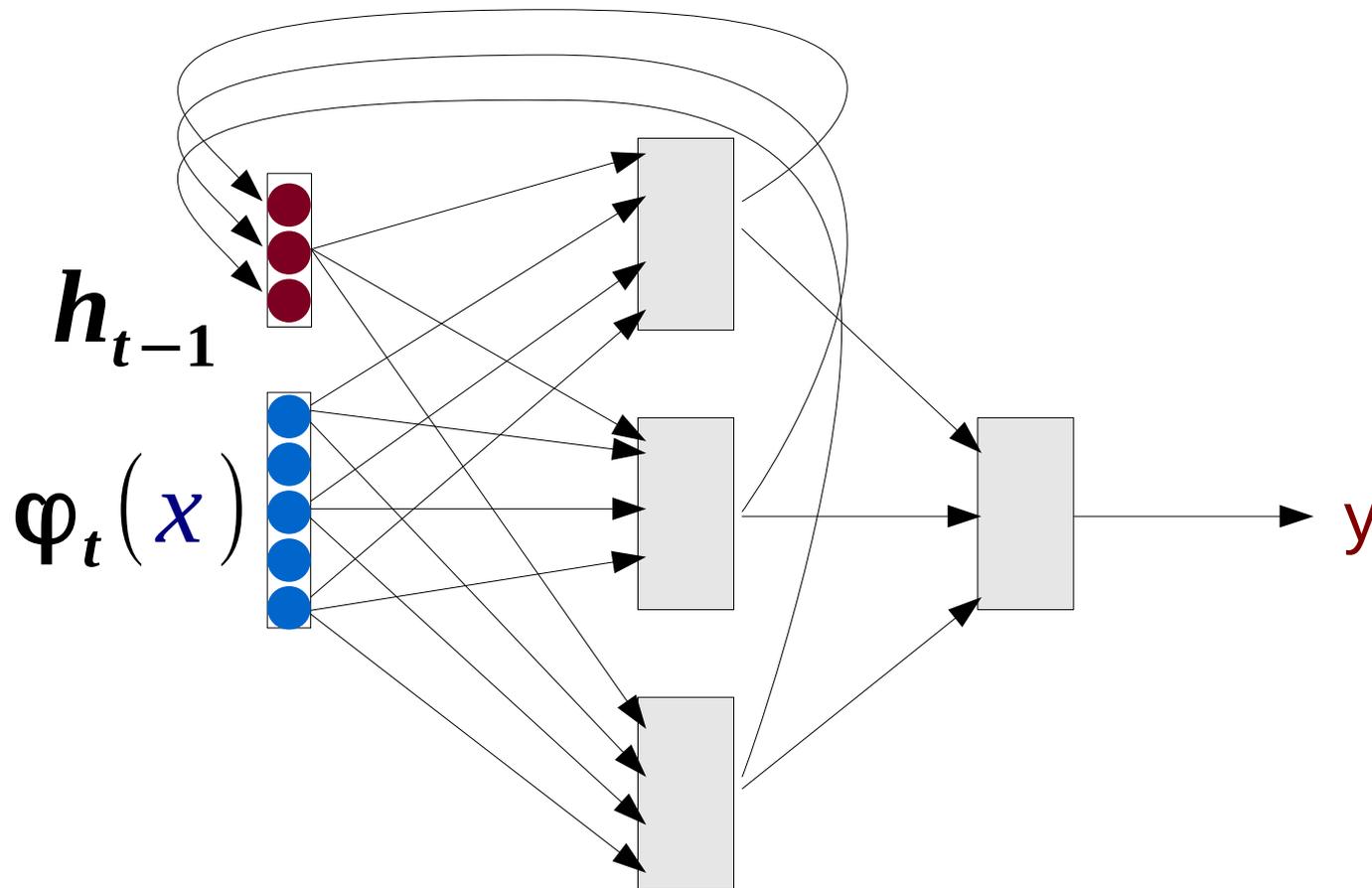
- すべての線が前方向のネット



- (有効非巡回グラフ DAG になっている)

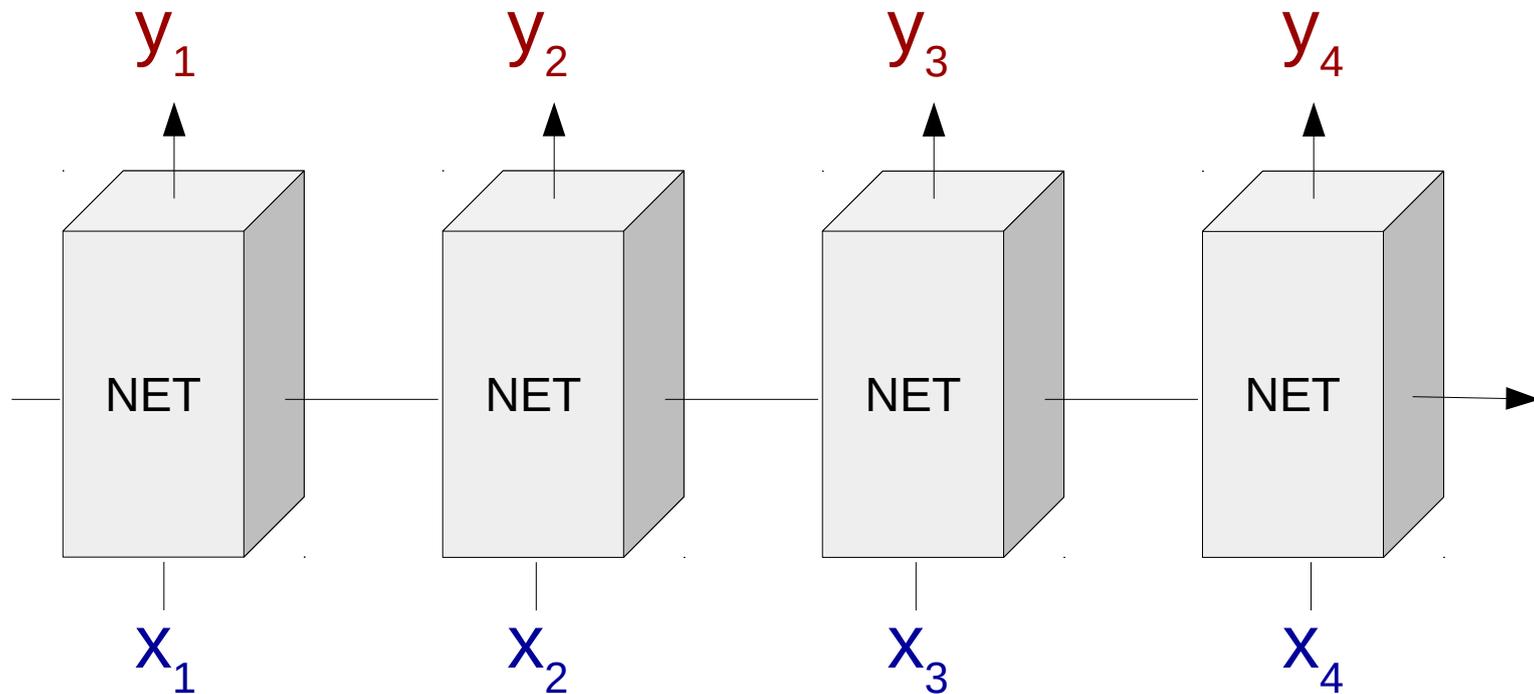
リカレントニューラルネット (RNN)

- ノードの一部の出力が入力として戻ってくる



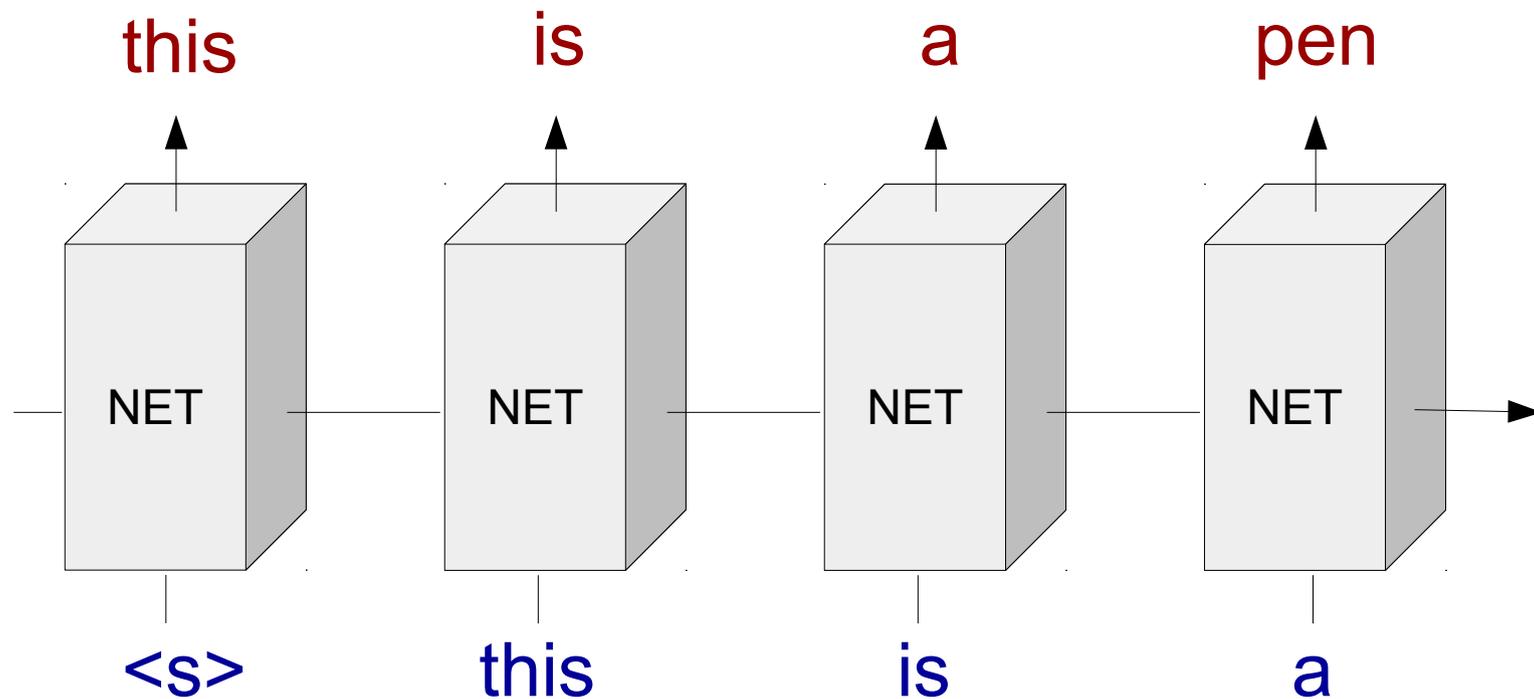
- 理由：「記憶」が可能

系列モデルとしての RNN



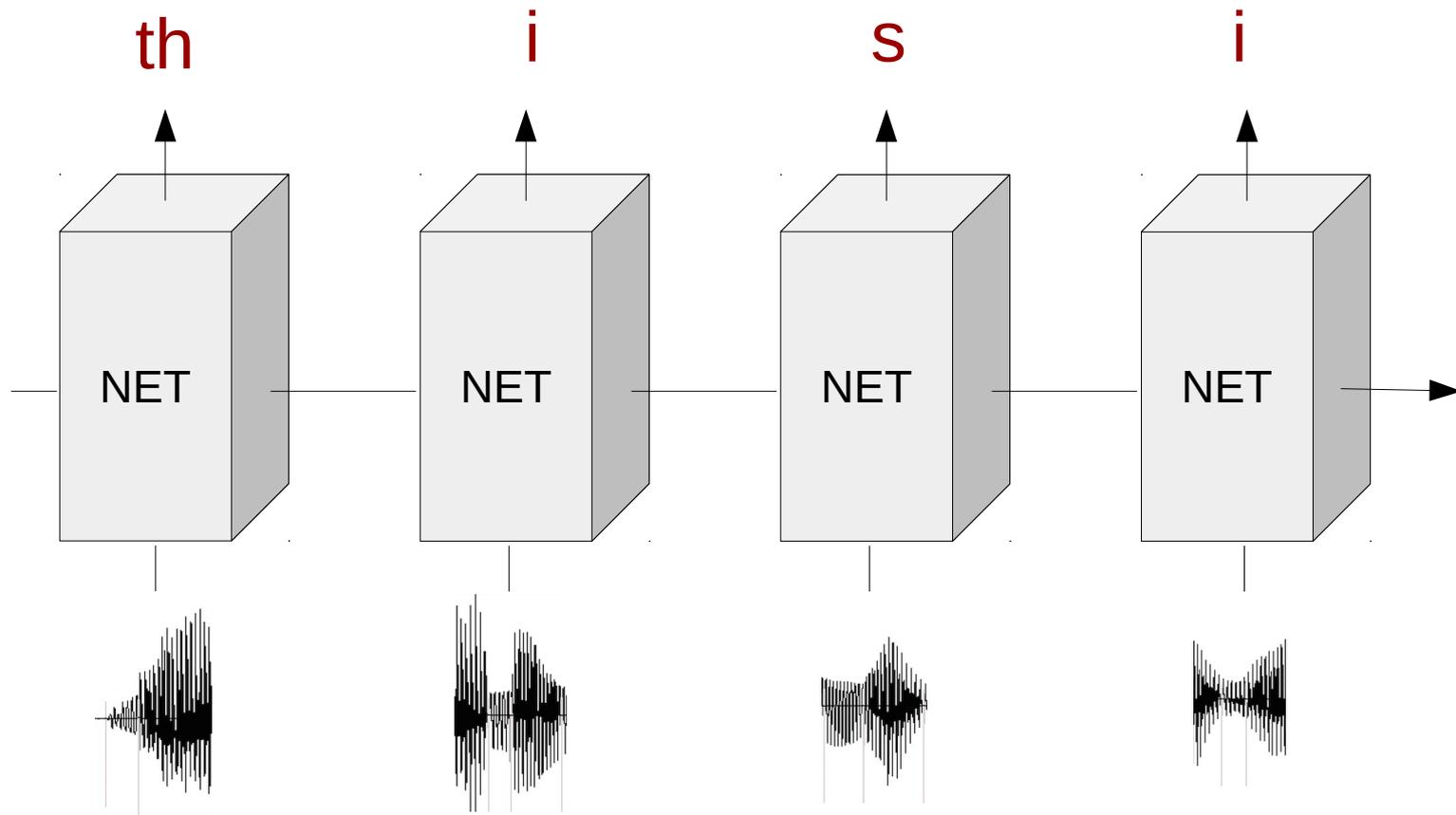
例：テキストの次の単語の予測

[Mikolov+11]



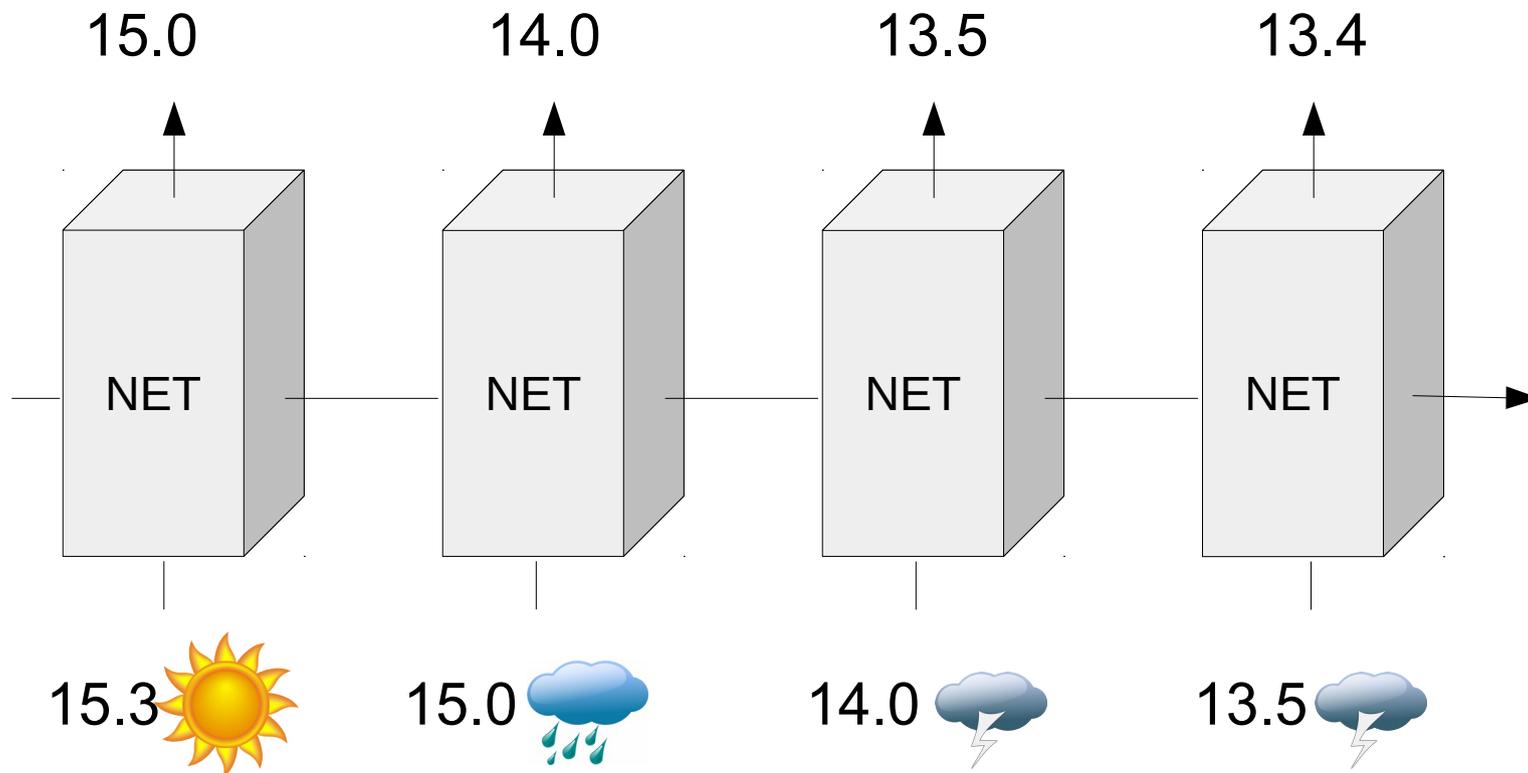
- テキスト生成が可能
- テキストの自然性を判断することが可能 (言語モデル)

例：音声信号から音素の予測 [Robinson+ 96]

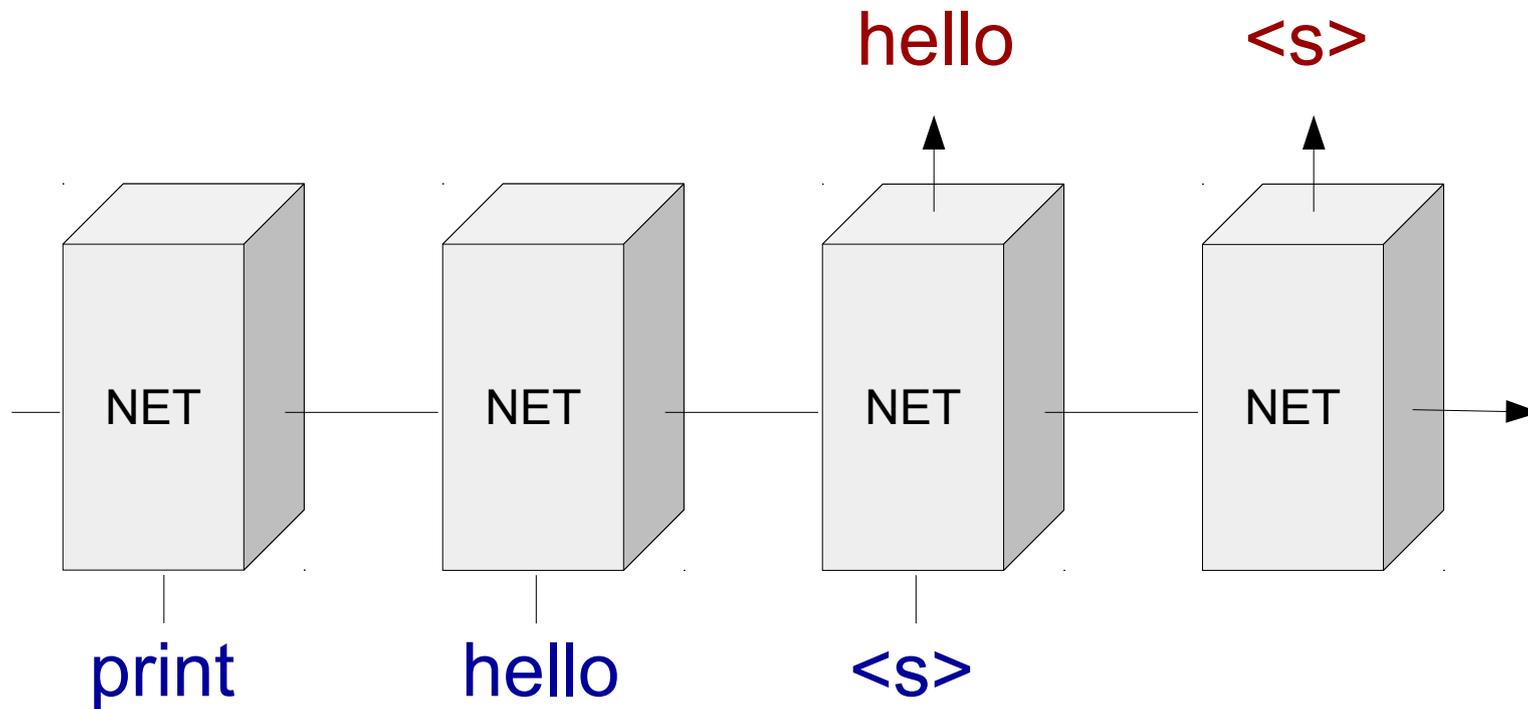


例：天気の予測

[Maqsood+ 04]



例：プログラムの実行 [Zaremba+14]



- (もっと複雑なプログラムも可能)

ほかにも...

- 機械翻訳 [Sutskever+ 14]
- 手書き文字の生成 [Graves+ 13]
- 画像から説明文の生成 [Karpathy+ 14]

復習：
フィードフォワードネットの逆伝搬

確率的勾配降下法 (stochastic gradient descent; SGD)

- ロジスティック回帰を含む確率モデルのための学習アルゴリズム

$w = 0$

for /iterations

for each labeled pair x, y in the data

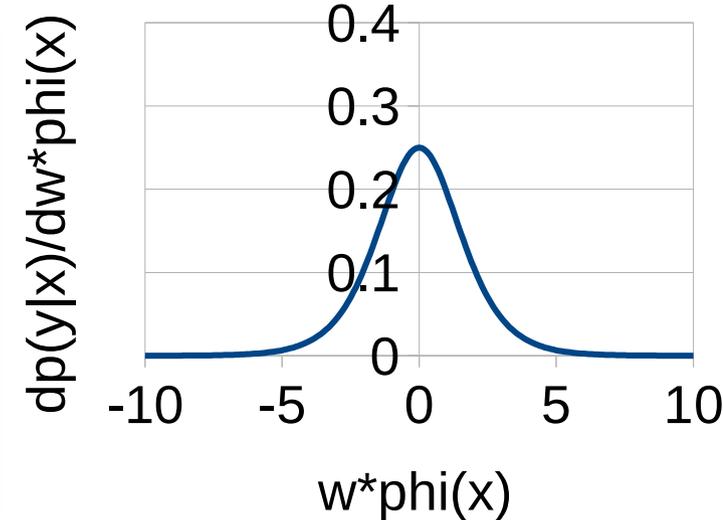
$w += \alpha * dP(y|x)/dw$

- つまり
 - 各学習例に対して勾配を計算
(y の確率が上がる方向)
 - その方向へ、学習率 α をかけた分だけ動かす

シグモイド関数の勾配

- 確率の微分

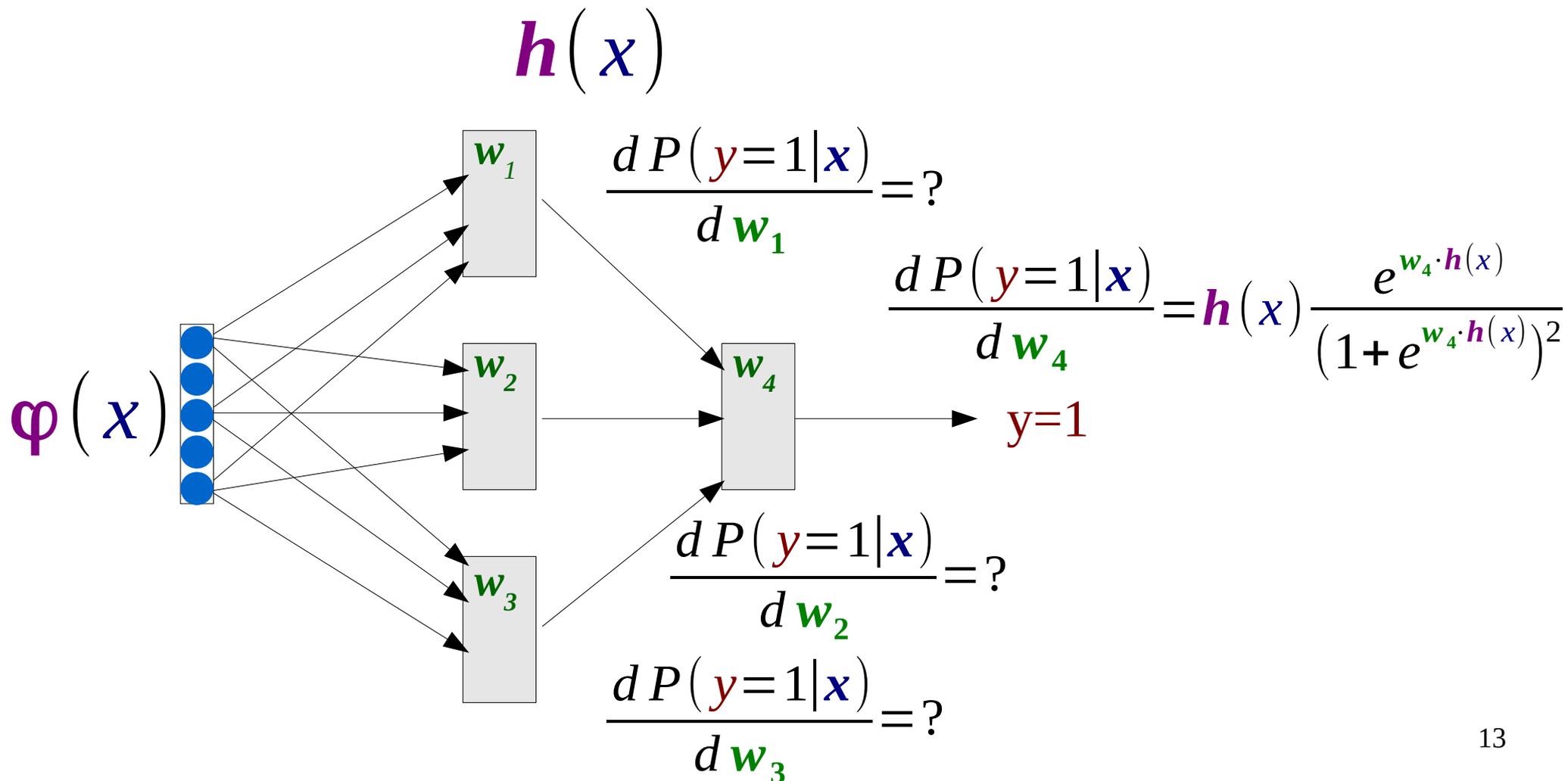
$$\begin{aligned} \frac{d}{d w} P(y=1|x) &= \frac{d}{d w} \frac{e^{w \cdot \varphi(x)}}{1+e^{w \cdot \varphi(x)}} \\ &= \varphi(x) \frac{e^{w \cdot \varphi(x)}}{(1+e^{w \cdot \varphi(x)})^2} \end{aligned}$$



$$\begin{aligned} \frac{d}{d w} P(y=-1|x) &= \frac{d}{d w} \left(1 - \frac{e^{w \cdot \varphi(x)}}{1+e^{w \cdot \varphi(x)}} \right) \\ &= -\varphi(x) \frac{e^{w \cdot \varphi(x)}}{(1+e^{w \cdot \varphi(x)})^2} \end{aligned}$$

学習：隠れ層の勾配が分からない

- 出力層のタグしか分からない



逆伝搬 (back propagation)

- 連鎖律を使って計算

$$\frac{dP(y=1|x)}{dw_1} = \frac{dP(y=1|x)}{dw_4 h(x)} \frac{dw_4 h(x)}{dh_1(x)} \frac{dh_1(x)}{dw_1}$$

$$\frac{e^{w_4 \cdot h(x)}}{(1 + e^{w_4 \cdot h(x)})^2}$$

↓

次の層の
エラー (δ_4)

$$w_{1,4}$$

↓

重み

$$1$$

↓

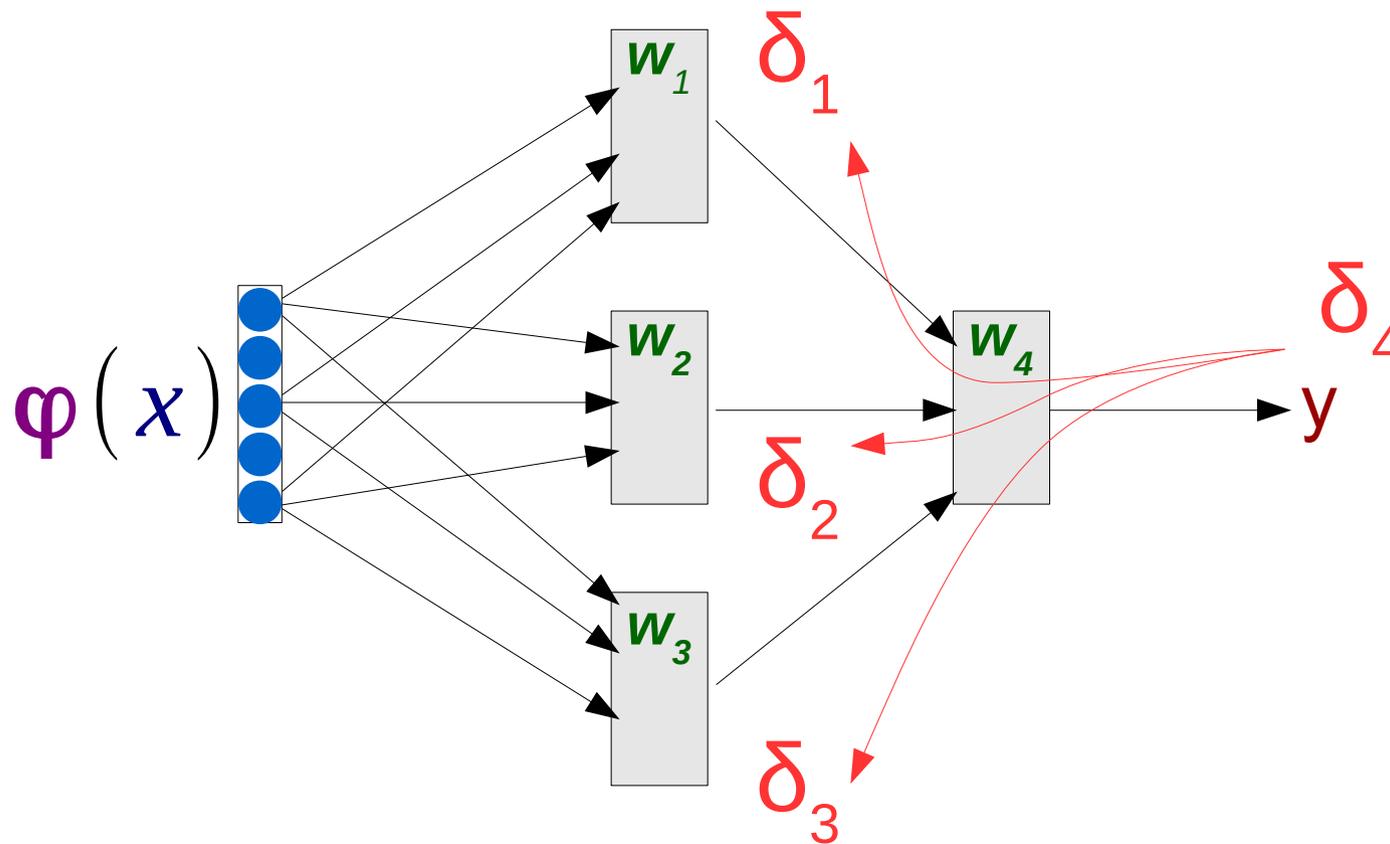
勾配

一般には
 i を次の層の
 j のエラーで計算

$$\frac{dP(y=1|x)}{dw_i} = \frac{dh_i(x)}{dw_i} \sum_j \delta_j w_{i,j}$$

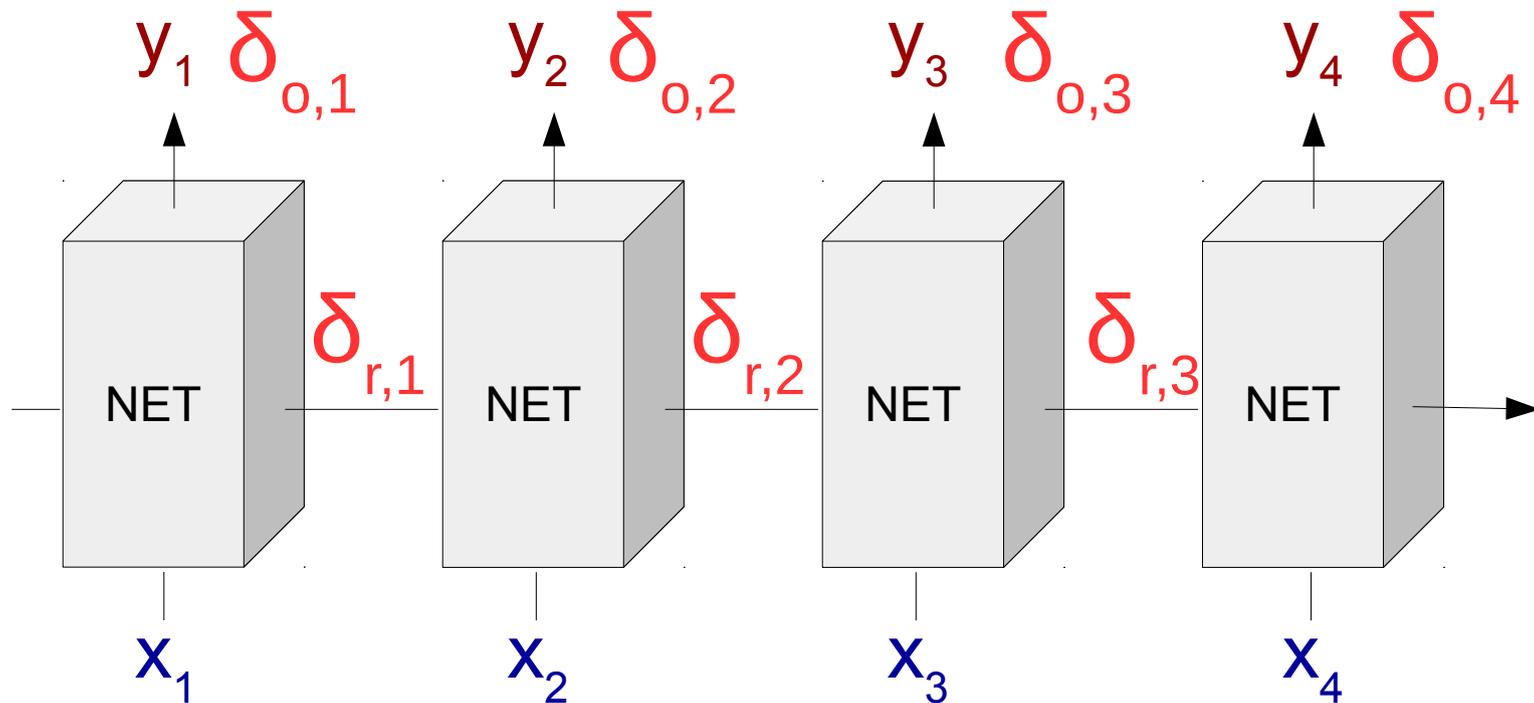
逆伝搬の概念図

- エラーを逆方向に伝搬



リカレントネットの逆伝搬

計算できるエラーは？

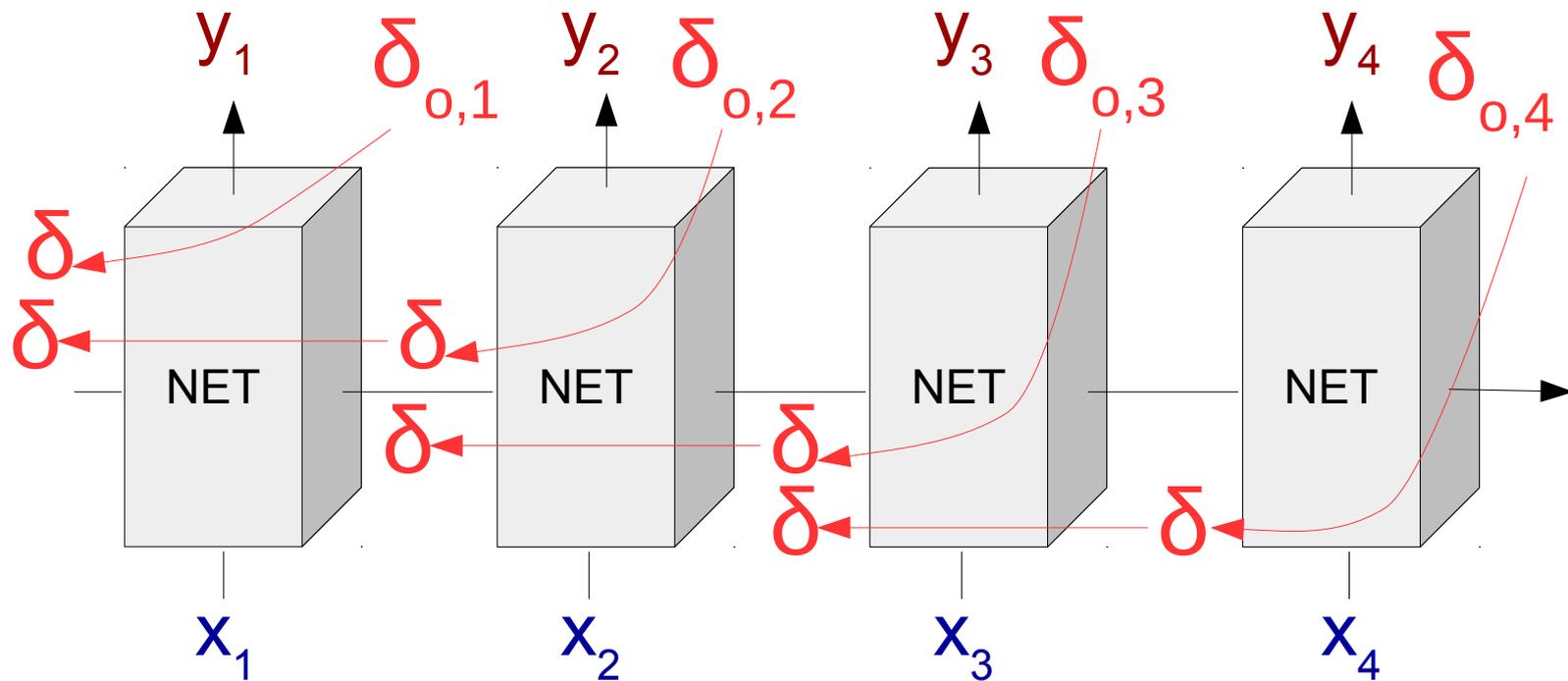


- 出力エラー δ_o は簡単に計算可能
- 次の時間からのリカレントエラー δ_r は逆伝搬

逆伝搬の方法

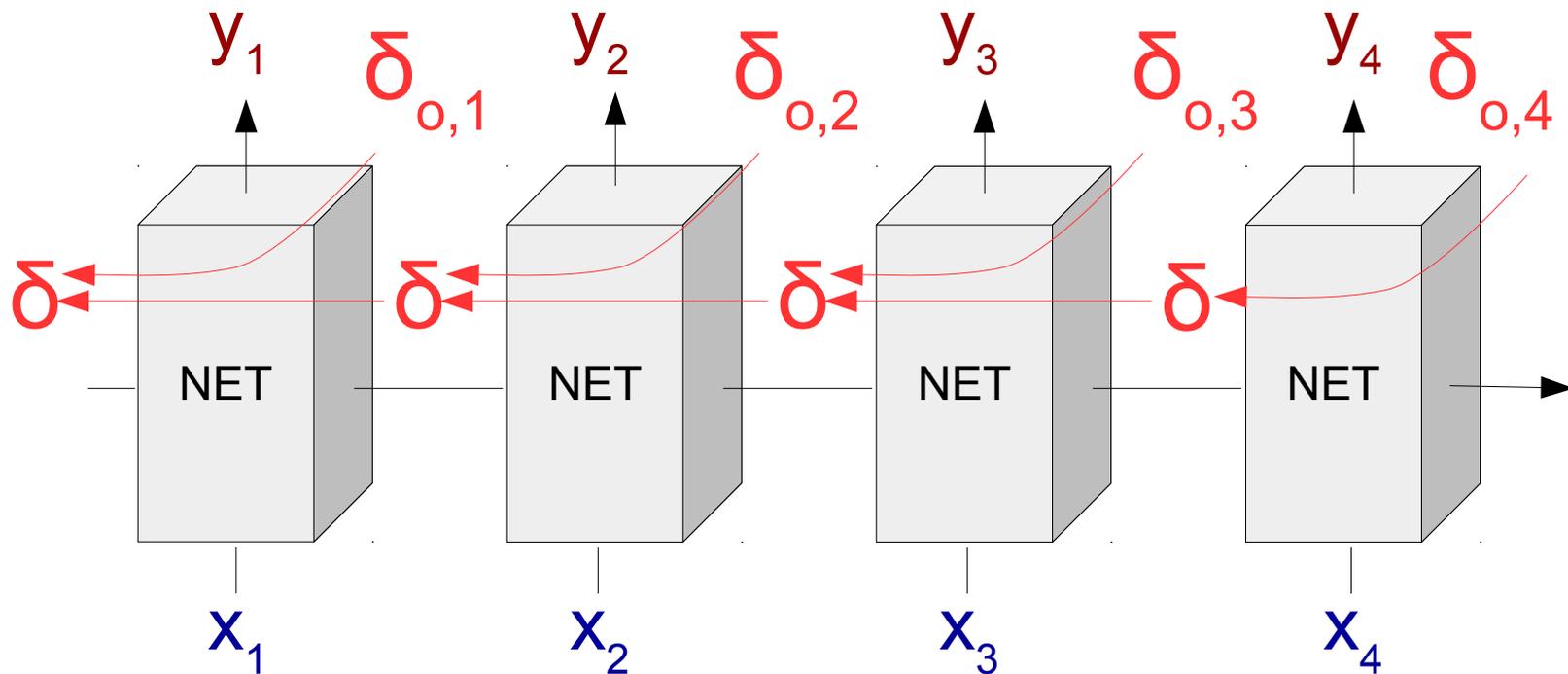
- 通時的逆伝播 (back propagation through time; BPTT)
 - 各 δ_0 に対して n 回 (時間) の逆伝搬を行う
- 完全勾配計算 (full gradient calculation)
 - 動的計画法に基づいて、系列全体の勾配を計算

通時的逆伝搬



- 1つの出力エラーだけを考慮
- n 回の計算だけを行う (ここでは $n=2$)

完全勾配計算

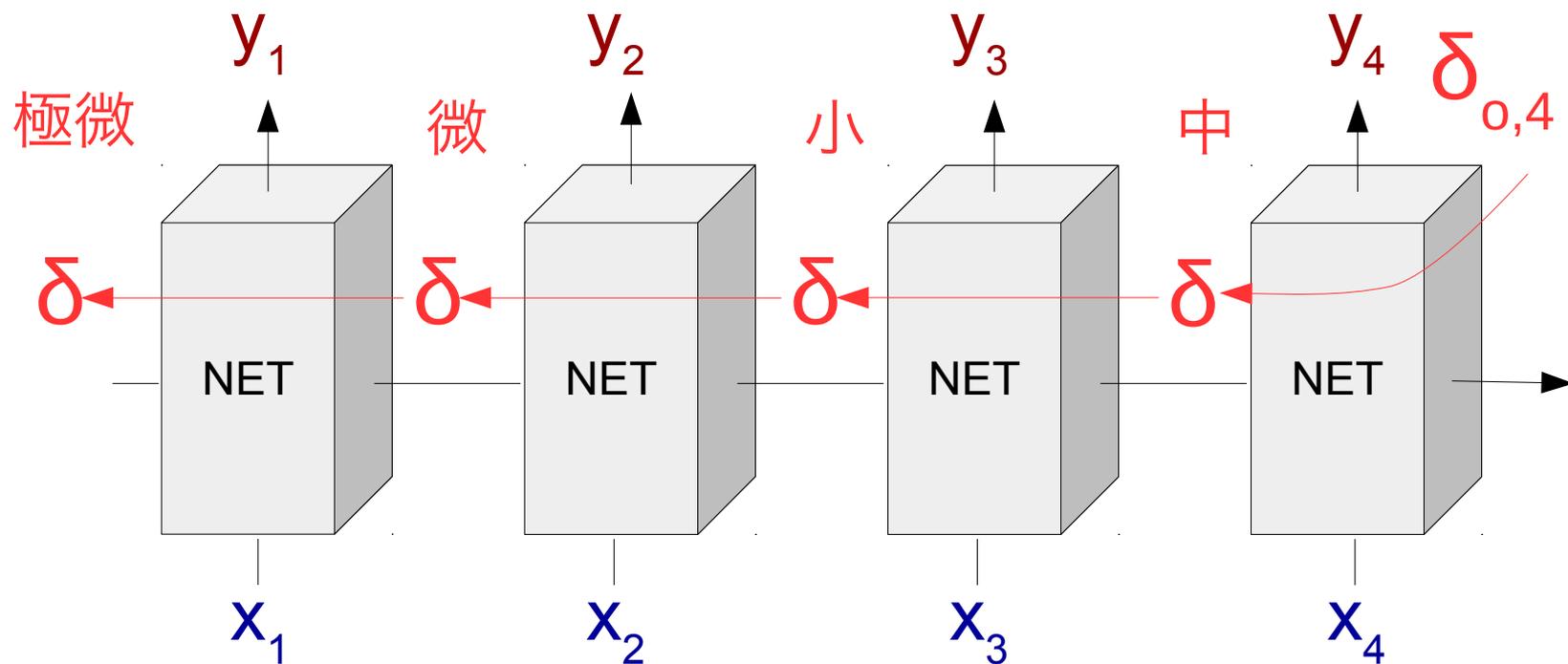


- まず系列のネット結果全体を計算
- 後ろからエラーを計算

BPTT? 完全勾配計算?

- 完全勾配計算
 - + 計算量が少ない、ステップ数の指定がない
 - - 系列全体の計算結果を保存するのが必要
- BPTT:
 - + メモリ量は系列の長さではなくステップ数に比例
 - - 計算量が多い、長距離の情報が捉えにくい

ニューラルネットにおける消える勾配



- 「Long Short Term Memory」などでこの問題に対処

参考文献

参考文献

- A. Graves. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850, 2013.
- A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. arXiv preprint arXiv:1412.2306, 2014.
- I. Maqsood, M. R. Khan, and A. Abraham. An ensemble of neural networks for weather forecasting. *Neural Computing & Applications*, 13(2):112–122, 2004.
- T. Mikolov, M. Karafit, L. Burget, J. Cernocky, and S. Khudanpur. Recurrent neural network based language model. In *Proc. 11th InterSpeech*, pages 1045–1048, 2010.
- T. Robinson, M. Hochberg, and S. Renals. The use of recurrent neural networks in continuous speech recognition. In *Automatic speech and speaker recognition*, pages 233–258. Springer, 1996.
- I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.
- W. Zaremba and I. Sutskever. Learning to execute. *CoRR*, abs/1410.4615, 2014.