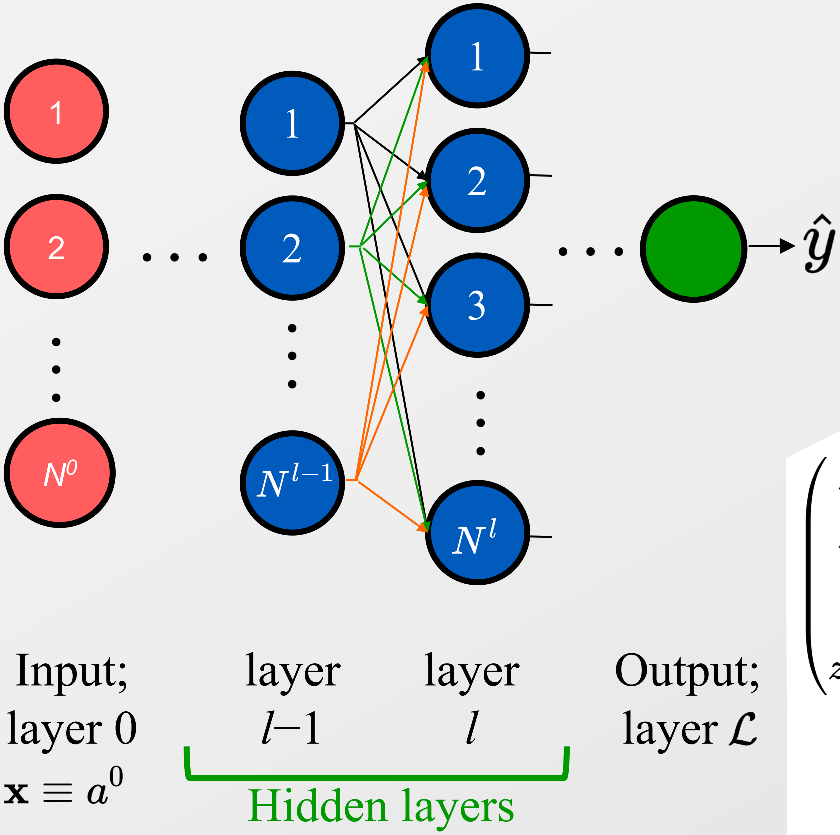
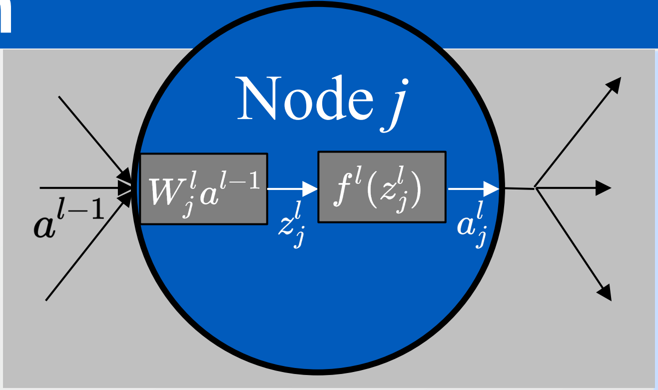
Lecture 22
Neural Networks 2

Backpropagation; case studies

Prof. David A. Kofke
CE 500 – Modeling Potential-Energy Surfaces
Department of Chemical & Biological Engineering
University at Buffalo

Here is a summary of the neural-network structure and notation





$$\begin{pmatrix} a_1^l \\ a_2^l \\ \vdots \\ a_{N^l}^l \\ 1 \end{pmatrix} = \begin{pmatrix} f^l(z_1^l) \\ f^l(z_2^l) \\ \vdots \\ f^l(z_{N^l}^l) \\ 1 \end{pmatrix}$$

activation function

$$a^l = f^l(z^l) = f^l \circ z^l$$

$$\begin{pmatrix} z_1^l \\ z_2^l \\ \vdots \\ z_{N^l}^l \end{pmatrix} = \begin{array}{c} \begin{array}{cccc} & & & b \end{array} \\ \begin{pmatrix} w_{11}^l & w_{12}^l & \cdots & w_{1(N^{l-1}+1)} \\ w_{21}^l & w_{22}^l & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ w_{N^l1}^l & \cdots & \cdots & w_{N^l(N^{l-1}+1)}^l \end{pmatrix} \end{array} \begin{pmatrix} a_1^{l-1} \\ a_2^{l-1} \\ \vdots \\ a_{N^{l-1}}^{l-1} \\ 1 \end{pmatrix}$$

$$\boxed{z^l} = \boxed{W^l}\boxed{a^{l-1}}$$

Weight indexes w_{jk}^l

Layer number

Number of node receiving input in layer l

Number of node providing output from layer l-1

Derivatives of loss function with respect to parameters are needed for learning

$$\theta(\text{new}) = \theta(\text{old}) - \gamma \frac{\partial L}{\partial \theta}$$

- Loss function

$$L(W^1, \ldots, W^{\mathcal{L}}) \equiv \sum_{\text{samples},i}^{n_{\text{batch}}} L_i(\hat{y}_i) \equiv \sum_i L_i \circ \hat{y}_i \equiv \sum_i (y_i - \hat{y}_i)^2 \equiv \sum_i \left(y_i - \hat{y}(\mathbf{x}_i; W^1, \ldots, W^{\mathcal{L}})\right)^2$$

θ is a generic symbol for a NN parameter. In our case, it is the W matrices

sample index, i

- Derivative matrix represents all derivatives for layer l

$$\frac{\partial L}{\partial W^l} \equiv \begin{pmatrix} \frac{\partial L}{\partial w^l_{11}} & \cdots & \frac{\partial L}{\partial w^l_{1(N^{l-1}+1)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial w^l_{N^l 1}} & \cdots & \frac{\partial L}{\partial w^l_{N^l(N^{l-1}+1)}} \end{pmatrix} = \sum_i \frac{\partial L_i}{\partial W^l} = \sum_i \frac{\partial L_i}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial W^l}$$

$$\frac{\partial L_i}{\partial \hat{y}_i} \equiv L_i' \circ \hat{y}_i = -2(y_i - \hat{y}_i)$$

ŷ is given as a series of nested operations

$$\hat{y} = W^{\mathcal{L}} f^{\mathcal{L}-1}(W^{\mathcal{L}-1} \dots f^2(W^2 f^1(W^1 \mathbf{x})))$$

define notation

$$\equiv W^{\mathcal{L}} f^{\mathcal{L}-1} \circ W^{\mathcal{L}-1} \dots f^2 \circ W^2 f^1 \circ W^1 \mathbf{x}$$

– Expression can end with any a or z

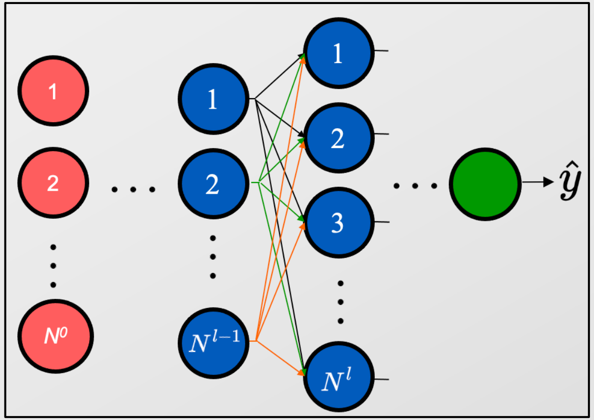
$$\hat{y}_i = \hat{y}(\mathbf{x}_i) = z^{\mathcal{L}}(\mathbf{x}_i)$$

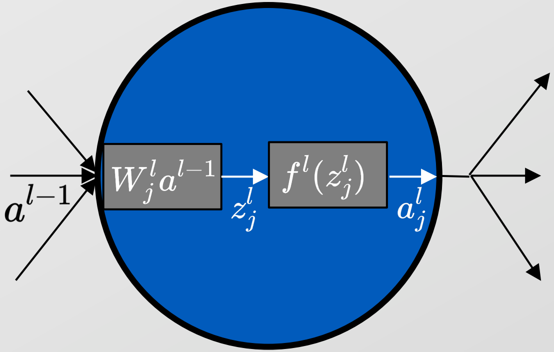
No activation function in output layer: $f^{\mathcal{L}}(z) = z$

$$\boxed{z^l = W^l a^{l-1}}$$

$$\boxed{\begin{aligned} a^l &= f^l(z^l) \\ &= f^l \circ z^l \end{aligned}}$$

$$\begin{aligned}
&= W^{\mathcal{L}} a^{\mathcal{L}-1}(\mathbf{x}_i) \\
&= W^{\mathcal{L}} f^{\mathcal{L}-1} \circ z^{\mathcal{L}-1}(\mathbf{x}_i) \\
&= W^{\mathcal{L}} f^{\mathcal{L}-1} \circ W^{\mathcal{L}-1} a^{\mathcal{L}-2}(\mathbf{x}_i) \\
&= W^{\mathcal{L}} f^{\mathcal{L}-1} \circ W^{\mathcal{L}-1} f^{\mathcal{L}-2} \circ \dots f^l \circ z^l(\mathbf{x}_i) \\
&= W^{\mathcal{L}} f^{\mathcal{L}-1} \circ W^{\mathcal{L}-1} f^{\mathcal{L}-2} \circ \dots f^l \circ W^l a^{l-1}(\mathbf{x}_i) \\
&= W^{\mathcal{L}} f^{\mathcal{L}-1} \circ W^{\mathcal{L}-1} f^{\mathcal{L}-2} \circ \dots f^l \circ W^l f^{l-1} \circ z^{l-1}(\mathbf{x}_i)
\end{aligned}$$





The layer-l z derivative (δ^l) can be used to get the desired W^l derivatives

$$\begin{pmatrix} \frac{\partial L}{\partial w^l_{11}} & \cdots & \frac{\partial L}{\partial w^l_{1(N^{l-1}+1)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial w^l_{N^l 1}} & \cdots & \frac{\partial L}{\partial w^l_{N^l(N^{l-1}+1)}} \end{pmatrix} \leftarrow \frac{\partial L_i}{\partial W^l} = \sum_{j=1}^{N^l} \boxed{\frac{\partial L_i}{\partial z^l_j}} \frac{\partial z^l_j}{\partial W^l} \equiv \sum_{j=1}^{N^l} \boxed{\delta^l_j} \frac{\partial z^l_j}{\partial W^l}$$

define

$$\delta^l \equiv \begin{pmatrix} \delta^l_1 \\ \vdots \\ \delta^l_{N^l} \end{pmatrix}$$

$$\boxed{z^l = W^l a^{l-1}}$$

$$\begin{pmatrix} z^l_1 \\ z^l_2 \\ \vdots \\ z^l_{N^l} \end{pmatrix} = \begin{pmatrix} w^l_{11} & w^l_{12} & \cdots & w_{1(N^{l-1}+1)} \\ w^l_{21} & w^l_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ w^l_{N^l 1} & \cdots & \cdots & w^l_{N^l(N^{l-1}+1)} \end{pmatrix} \cdot \begin{pmatrix} a^{l-1}_1 \\ a^{l-1}_2 \\ \vdots \\ a^{l-1}_{N^{l-1}} \\ 1 \end{pmatrix}$$

Each z_j depends only on row *j* of *W*

$$\sum_j \delta^l_j \frac{\partial z^l_j}{\partial W^l} = \delta^l_1 \times \begin{pmatrix} a^{l-1}_1 & a^{l-1}_2 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} + \delta^l_2 \times \begin{pmatrix} 0 & 0 & \cdots & 0 \\ a^{l-1}_1 & a^{l-1}_2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} + \cdots + \delta^l_{N^l} \times \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a^{l-1}_1 & a^{l-1}_2 & \cdots & 1 \end{pmatrix}$$

same

$$= \begin{pmatrix} \delta^l_1 a^{l-1}_1 & \delta^l_1 a^{l-1}_2 & \cdots & \delta^l_1 \\ \delta^l_2 a^{l-1}_1 & \delta^l_2 a^{l-1}_2 & \cdots & \delta^l_2 \\ \vdots & \vdots & \ddots & \vdots \\ \delta^l_{N^l} a^{l-1}_1 & \delta^l_{N^l} a^{l-1}_2 & \cdots & \delta^l_{N_l} \end{pmatrix} = \begin{pmatrix} \delta^l_1 \\ \delta^l_2 \\ \vdots \\ \delta^l_{N_l} \end{pmatrix} \begin{pmatrix} a^{l-1}_1 & a^{l-1}_2 & \cdots & 1 \end{pmatrix} = \delta^l (a^{l-1})^{\mathrm{T}}$$

$$\boxed{\frac{\partial L_i}{\partial W^l} = \delta^l (a^{l-1})^{\mathrm{T}}}$$

Recursion can be used to compute δ^l

$$\hat{y}_i = W^{\mathcal{L}} f^{\mathcal{L}-1} \circ W^{\mathcal{L}-1} f^{\mathcal{L}-2} \circ \ldots f^l \circ \boxed{z^l(\mathbf{x}_i)}$$

$$= W^{\mathcal{L}} f^{\mathcal{L}-1} \circ W^{\mathcal{L}-1} f^{\mathcal{L}-2} \circ \ldots f^l \circ \boxed{W^l f^{l-1} \circ z^{l-1}(\mathbf{x}_i)}$$

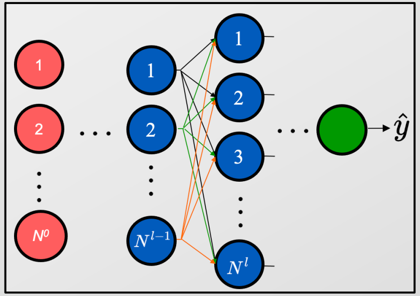
$$\delta^l \equiv \frac{\partial L_i}{\partial z^l} = (L_i' \circ \hat{y}_i) \frac{\partial \hat{y}_i}{\partial z^l}$$

a_j^{l-1} depends only on z_j^{l-1}

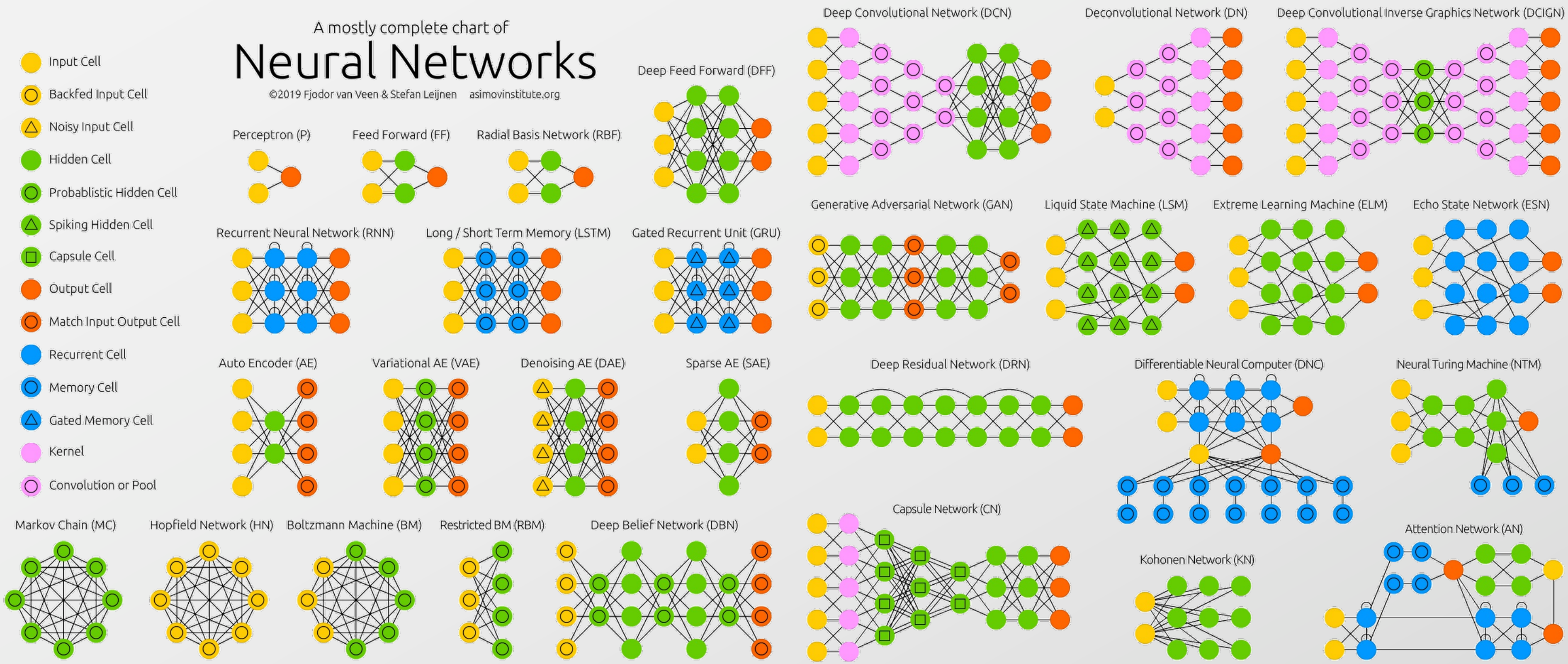
$$(\delta^{l-1})^{\mathrm{T}} = \boxed{(L_i' \circ \hat{y}_i) \left(\frac{\partial \hat{y}_i}{\partial z^l} \right)}^{\mathrm{T}} \boxed{\frac{\partial z^l}{\partial z^{l-1}}} = (\boxed{\delta^l})^{\mathrm{T}} \boxed{W^l \mathrm{diag}\left[(f^{l-1})' \circ z^{l-1} \right]}$$

$z^l = W^l f^{l-1} \circ z^{l-1}$

This is the basis for *backpropagation*. Compute derivatives for output layer, and sweep back to get derivatives for each lower layer in succession (l → l-1), to the first one



Feed-forward is just one of many types of neural networks that have been developed



https://www.asimovinstitute.org/neural-network-zoo/

Suggested Reading/Viewing

- Dral, Pavlo O; Kananenka, Alexei A; Ge, Fuchun; Xue, Bao-Xin, Chapter 8, Neural Networks. In *Quantum Chemistry in the Age of Machine Learning*.
  - https://doi.org/10.1016/B978-0-323-90049-2.00011-1
  - Posted on UBLearns
- Jinzhe Zeng, Liqun Cao, and Tong Zhu, Chapter 12, Neural Network Potentials. In *Quantum Chemistry in the Age of Machine Learning*.
  - Posted on UBLearns
- https://en.wikipedia.org/wiki/Backpropagation
- Using TensorFlow to solve regression problems, including the MPG example
  - https://www.youtube.com/watch?v=-vHQub0NXI4