Ch 10.2: Multi-Layer Neural Nets

Lecture 31 - CMSE 381

Prof. Elizabeth Munch

Michigan State University
::
Dept of Computational Mathematics, Science & Engineering

Fri, Dec 1, 2023

Announcements

Last time:

- Single Layer Neural Nets

This lecture:

- Multi-layer Neural Nets
- Application to MNIST

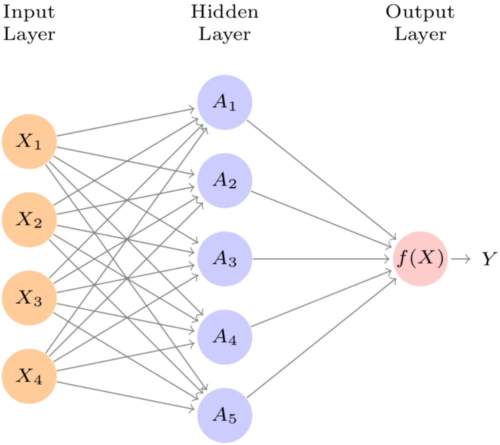
The end is near!

Lec #	Date			Reading	Homeworks
	Wed	Nov 29	**Midterm #3**		
32	Fri	Dec 1	Multi Layer NN	10.2	
33	Mon	Dec 4	CNN	10.3	
34	Wed	Dec 6	Unsupervised Learning & Clustering	12.1, 12.4	
35	Fri	Dec 8	Virtual: Project office hours		Project due

Section 1

Neural Nets

Feed Forward Neural Network: The cartoon



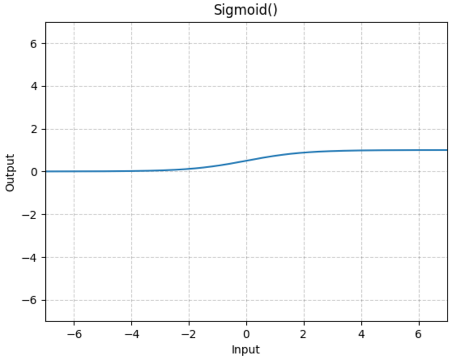
$$A_k = h_k(X) = g(w_{k0} + \textstyle\sum_{j=1}^{p} w_{kj} X_j),$$

$$f(X) = \beta_0 + \sum_{k=1}^{K} \beta_k A_k$$

Choices for activation function

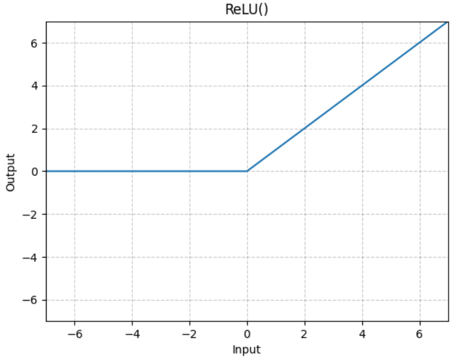
Sigmoid:

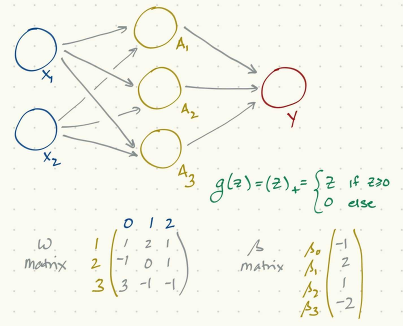
$$g(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$$



ReLU: Rectified linear unit

$$g(z) = (z)_+ = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{else.} \end{cases}$$





$$A_k = h_k(X) = g(w_{k0} + \textstyle\sum_{j=1}^{p} w_{kj} X_j),$$

$$A = g(\mathbf{W} \cdot \mathbf{X}) \qquad \mathbf{X}^T = (1 \ X_1 \ X_2 \ \cdots \ X_p)$$

$$f(X) = \beta_0 + \sum_{k=1}^{K} \beta_k A_k$$

$$Y = \beta \cdot \mathbf{A} \qquad \mathbf{A}^T = (1 \ A_1 \ A_2 \ \cdots \ A_K)$$

Training the model

Choose parameters by minimizing RSS, $\sum_{i=1}^{n}(y_i - f(x_i))^2$ (or other loss function)

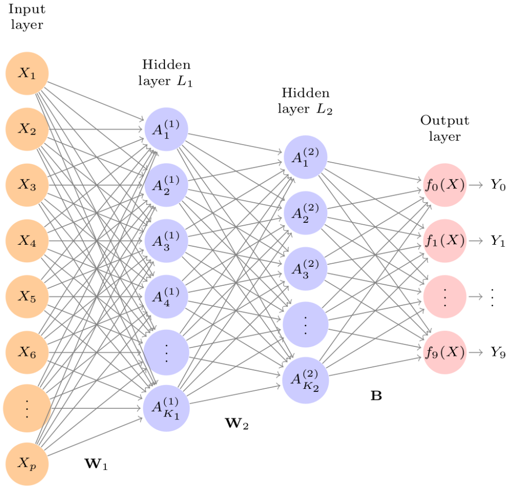
Chosen in advance:

Tuned by the model:

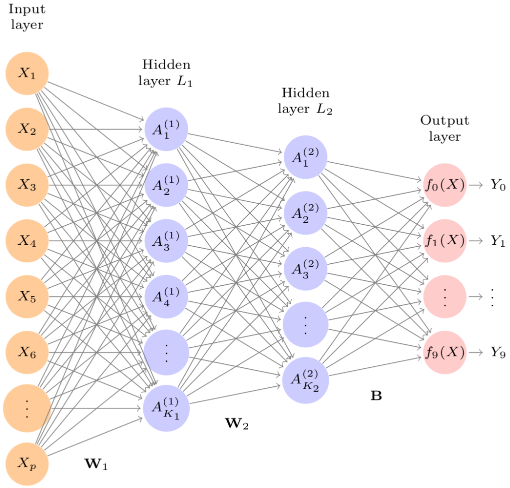
Section 2

Multilayer Neural Networks

Multiple layers



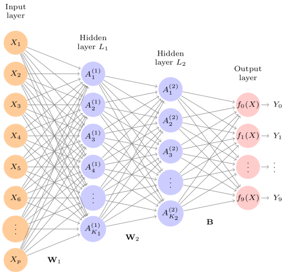
Hidden layers



$$
\begin{aligned}
A_k^{(1)} &= h_k^{(1)}(X) \\
&= g(w_{k0}^{(1)} + \textstyle\sum_{j=1}^{p} w_{kj}^{(1)} X_j)
\end{aligned}
$$

$$
\begin{aligned}
A_\ell^{(2)} &= h_\ell^{(2)}(X) \\
&= g(w_{\ell 0}^{(2)} + \textstyle\sum_{k=1}^{K_1} w_{\ell k}^{(2)} A_k^{(1)})
\end{aligned}
$$

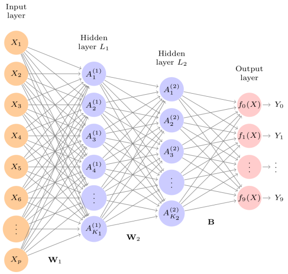
More on that architecture



$$
\begin{aligned}
A_k^{(1)} &= h_k^{(1)}(X) \\
&= g(w_{k0}^{(1)} + \textstyle\sum_{j=1}^{p} w_{kj}^{(1)} X_j)
\end{aligned}
$$

$$
\begin{aligned}
A_\ell^{(2)} &= h_\ell^{(2)}(X) \\
&= g(w_{\ell 0}^{(2)} + \textstyle\sum_{k=1}^{K_1} w_{\ell k}^{(2)} A_k^{(1)})
\end{aligned}
$$

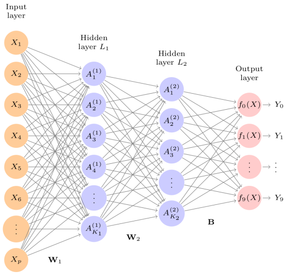
Matrix version: First layer



$$\begin{aligned} A_k^{(1)} &= h_k^{(1)}(X) \\ &= g(w_{k0}^{(1)} + \textstyle\sum_{j=1}^{p} w_{kj}^{(1)} X_j) \end{aligned}$$

$$A^{(1)} = g(\mathbf{W}^{(1)} \cdot \mathbf{X}) \qquad \mathbf{X}^T = (1 \; X_1 \; X_2 \; \cdots \; X_p)$$

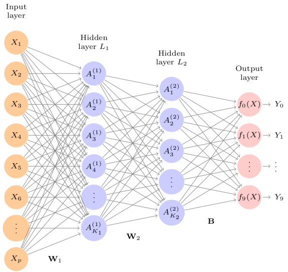
Matrix version: Second layer



$$\begin{aligned} A_\ell^{(2)} &= h_\ell^{(2)}(X) \\ &= g(w_{\ell 0}^{(2)} + \textstyle\sum_{k=1}^{K_1} w_{\ell k}^{(2)} A_k^{(1)}) \end{aligned}$$

$$A^{(2)} = g(\mathbf{W}^{(2)} \cdot \mathbf{A}) \qquad (\mathbf{A}^{(1)})^T = (1\ A_1^{(1)}\ A_2^{(1)}\ \cdots\ A_{K_1}^{(1)})$$

Matrix version: Last layer, first step

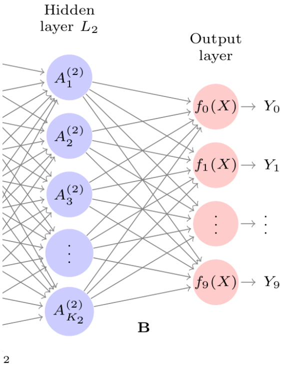


$$\begin{aligned} Z_m &= \beta_{m0} + \textstyle\sum_{\ell=1}^{K_2} \beta_{m\ell} h_\ell^{(2)}(X) \\ &= \beta_{m0} + \textstyle\sum_{\ell=1}^{K_2} \beta_{m\ell} A_\ell^{(2)}, \end{aligned}$$

$$\mathbf{Z} = \beta \cdot \mathbf{A}$$

β is $M \times (K_2 + 1)$ matrix $\qquad (\mathbf{A}^{(2)})^T = (1 \; A_1^{(2)} \; A_2^{(2)} \; \cdots \; A_{K_2}^{(2)})$

The last column for classification: Softmax

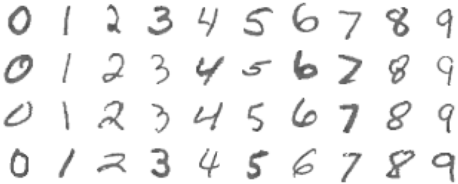


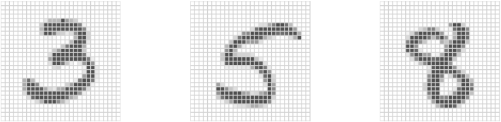
$$f_m(X) = \Pr(Y = m|X) = \frac{e^{Z_m}}{\sum_{\ell=0}^{9} e^{Z_\ell}},$$

An example

$$Z = \begin{pmatrix} 1 & 3 & -1 & 2 & 5 \end{pmatrix}$$

MNIST

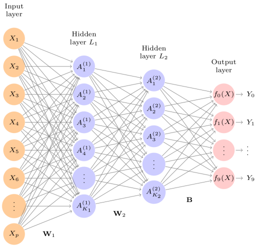




Coding

TL;DR

Feed Forward Neural Net



$$A_k = h_k(X) = g(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j),$$

- Combines input data using learned weights
- Linear combo of those to get output
- Sometimes softmax to get probability of classification

Next time

Lec #	Date			Reading	Homeworks
	Wed	Nov 29	**Midterm #3**		
32	Fri	Dec 1	Multi Layer NN	10.2	
33	Mon	Dec 4	CNN	10.3	
34	Wed	Dec 6	Unsupervised Learning & Clustering	12.1, 12.4	
35	Fri	Dec 8	Virtual: Project office hours		Project due