

Ch 10.2: Multi-Layer Neural Nets

Lecture 30 - CMSE 381

Prof. Mengsen Zhang

Michigan State University

::

Dept of Computational Mathematics, Science & Engineering

Weds, April 9, 2025

Announcements

Last time:

- Single Layer Neural Nets

This lecture:

- Multi-layer Neural Nets
- Application to MNIST

Announcements:

- HW 9 Due Sunday 4/13
- Exam 3 4/21
- Project Due 4/25

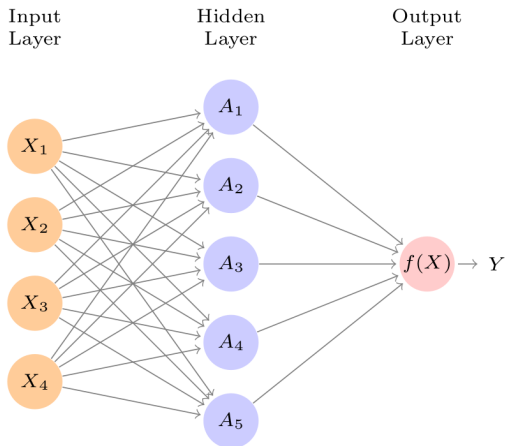
The end is near!

28	F	4/4	SVM	9.3, 9.4	HW #8 Due Sun 4/6
29	M	4/7	Single Layer NN	10.1	
30	W	4/9	Multi Layer NN	10.2	HW #9 Due Sun 4/13
31	F	4/11	CNN	10.3	
32	M	4/14	Unsupervised learning / clustering	12.1, 12.4	
33	W	4/16	Virtual: Project Office Hours		Project Due
	F	4/18	Review		
	M	4/21	Midterm #3		
	W	4/23			
	F	4/25			

Section 1

Neural Nets

Feed Forward Neural Network: The cartoon



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

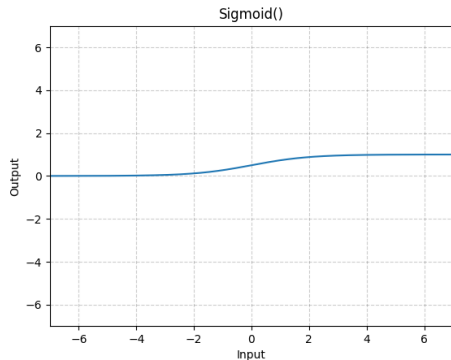
$$f(X) = \beta_0 + \sum_{k=1}^K \beta_k A_k$$

Test your understanding: [PollEv](#)

Choices for activation function

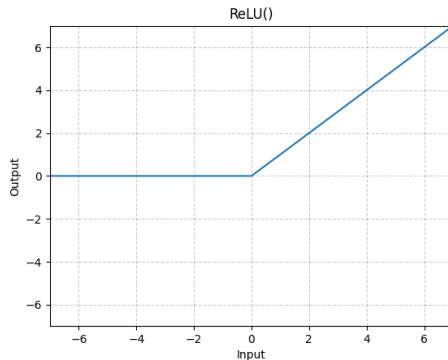
Sigmoid:

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

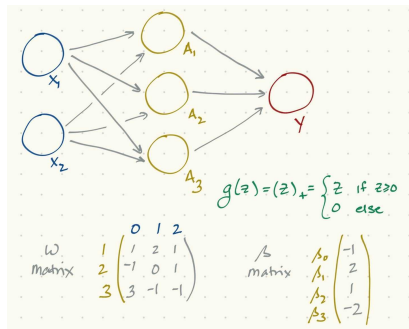


ReLU: Rectified linear unit

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



Matrix version



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

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



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

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

Training the model

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

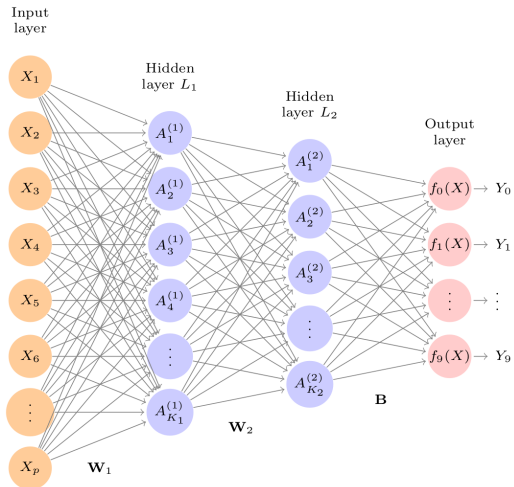
Chosen in advance:

Tuned by the model:

Section 2

Multilayer Neural Networks

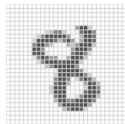
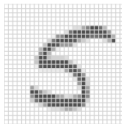
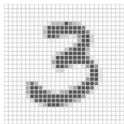
Multiple layers



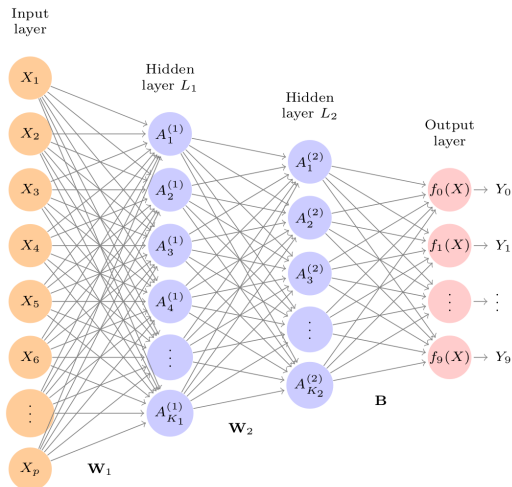
More typical for image classification

Example: MNIST

0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9



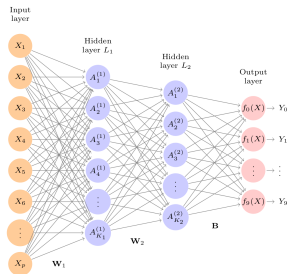
Hidden layers



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

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

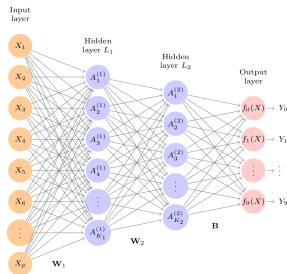
More on that architecture



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

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

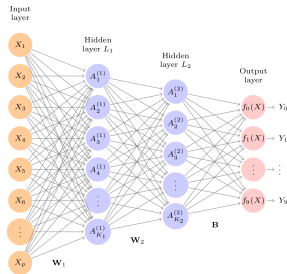
Matrix version: First layer



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

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

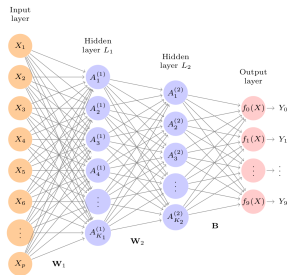
Matrix version: Second layer



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

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

Matrix version: Last layer, first step



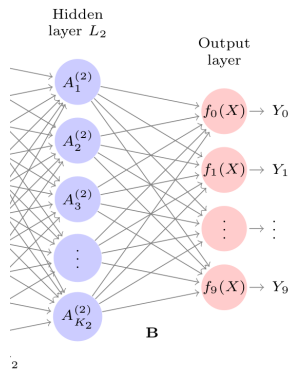
$$Z_m = \beta_{m0} + \sum_{\ell=1}^{K_2} \beta_{m\ell} h_{\ell}^{(2)}(X)$$

$$= \beta_{m0} + \sum_{\ell=1}^{K_2} \beta_{m\ell} A_{\ell}^{(2)},$$

$$\mathbf{Z} = \beta \cdot \mathbf{A}^{(2)}$$

$$\beta \text{ is } M \times (K_2 + 1) \text{ matrix} \quad (\mathbf{A}^{(2)})^T = (1 \ A_1^{(2)} \ A_2^{(2)} \ \dots \ A_{K_2}^{(2)})$$

The last column for classification: Softmax



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

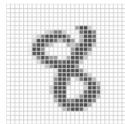
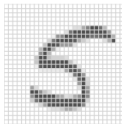
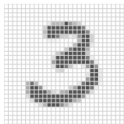
An example

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

$$Z = (1 \quad 3 \quad -1 \quad 2 \quad 5)$$

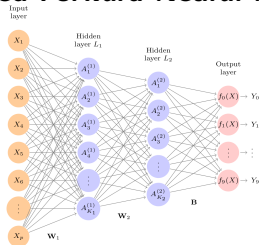
MNIST

0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9



Coding

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

	M	3/17	Midterm #2		Sun 3/16
21	W	3/19	Polynomial & Step Functions	7.1-7.2	
22	F	3/21	Step Functions; Basis functions; Start Splines	7.2-7.4	
23	M	3/24	Regression Splines	7.4	
24	W	3/26	Decision Trees	8.1	HW #6 Due Wed 3/26
25	F	3/28	Random Forests	8.2.1, 8.2.2	HW #7 Due Sun 3/30
26	M	3/31	Maximal Margin Classifier	9.1	
27	W	4/2	SVC	9.2	
28	F	4/4	SVM	9.3, 9.4	HW #8 Due Sun 4/6
29	M	4/7	Single Layer NN	10.1	
30	W	4/9	Multi Layer NN	10.2	
31	F	4/11	CNN	10.3	
32	M	4/14	Unsupervised learning / clustering	12.1, 12.4	HW #9 Due Sun 4/13
33	W	4/16	Virtual: Project Office Hours		
	F	4/18	Review		
	M	4/21	Midterm #3		
	W	4/23			
	F	4/25			Project Due

Q of the day: How do you tell if an artificial neural network is for classification or regression?
(if you only have the math)