Deep Learning

Tianhang Zheng
https://tianzheng4.github.io

Deep Learning

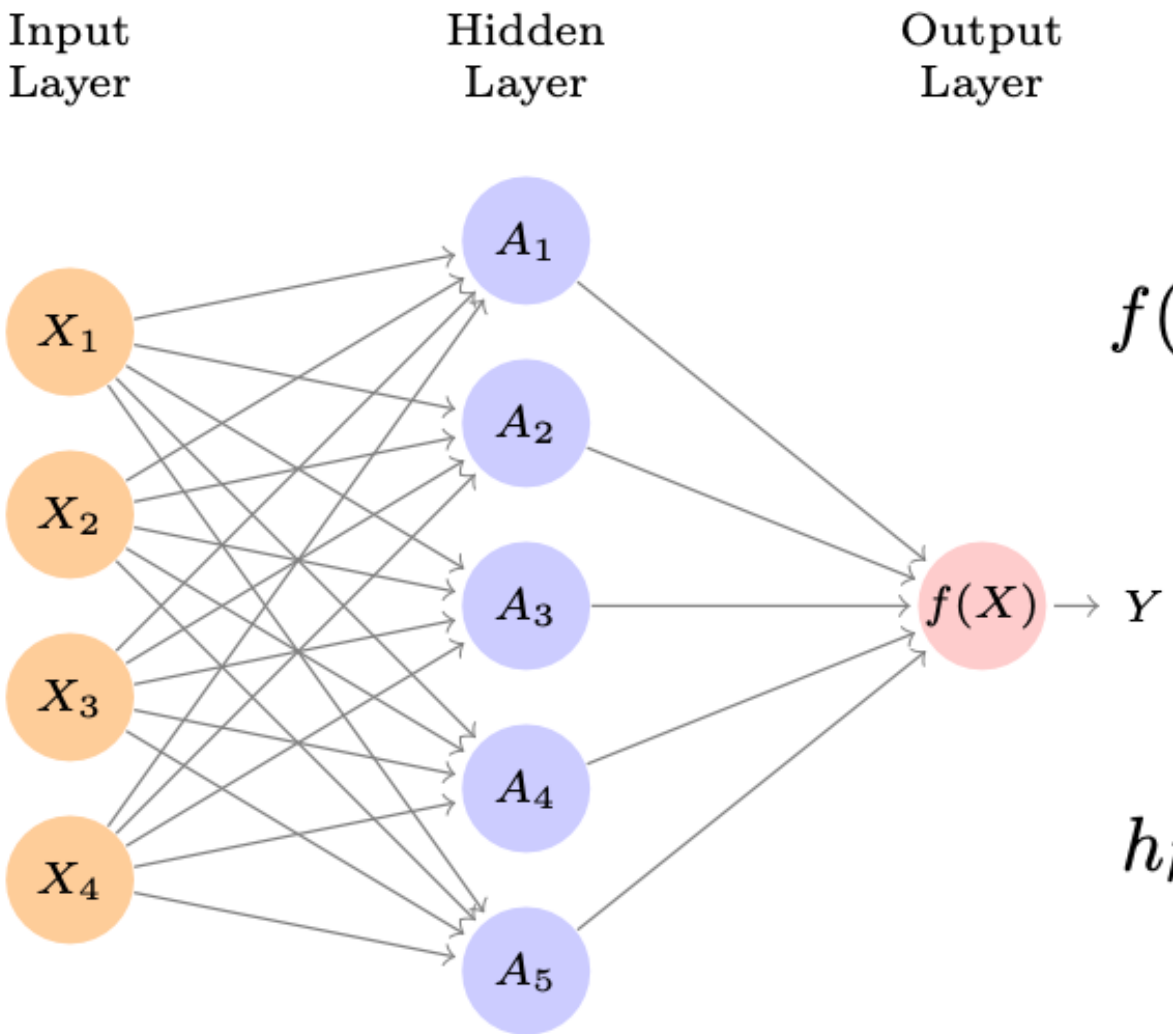
People begin to study neural networks starting from 1980

Deep Learning

Deep learning becomes very popular starting from 2012

ImageNet classification with deep convolutional neural networks (NeurIPS 2012) shows that deep neural networks can outperform traditional machine learning techniques by a large margin!

Single Layer Neural Network



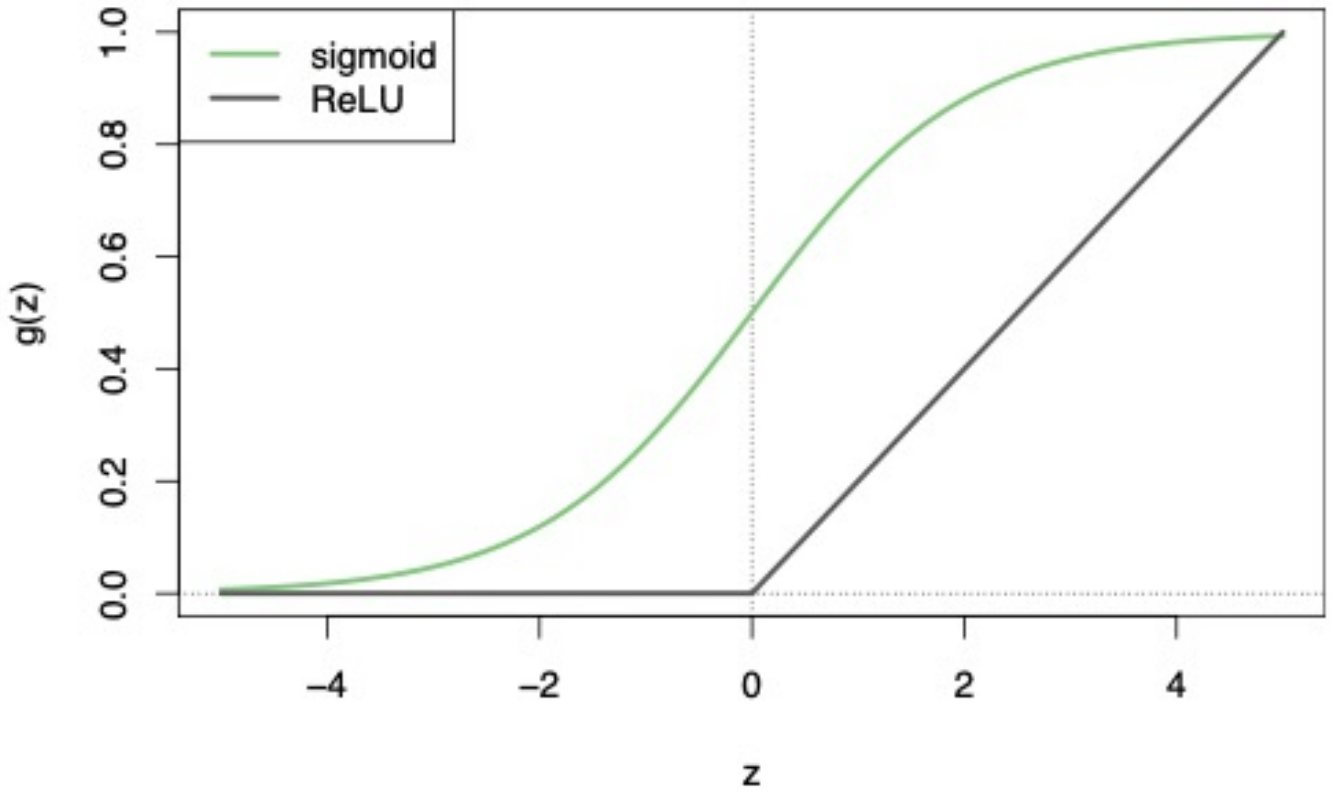
$$f(X) = \beta_0 + \sum_{k=1}^{K} \beta_k h_k(X)$$

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

Activation Function

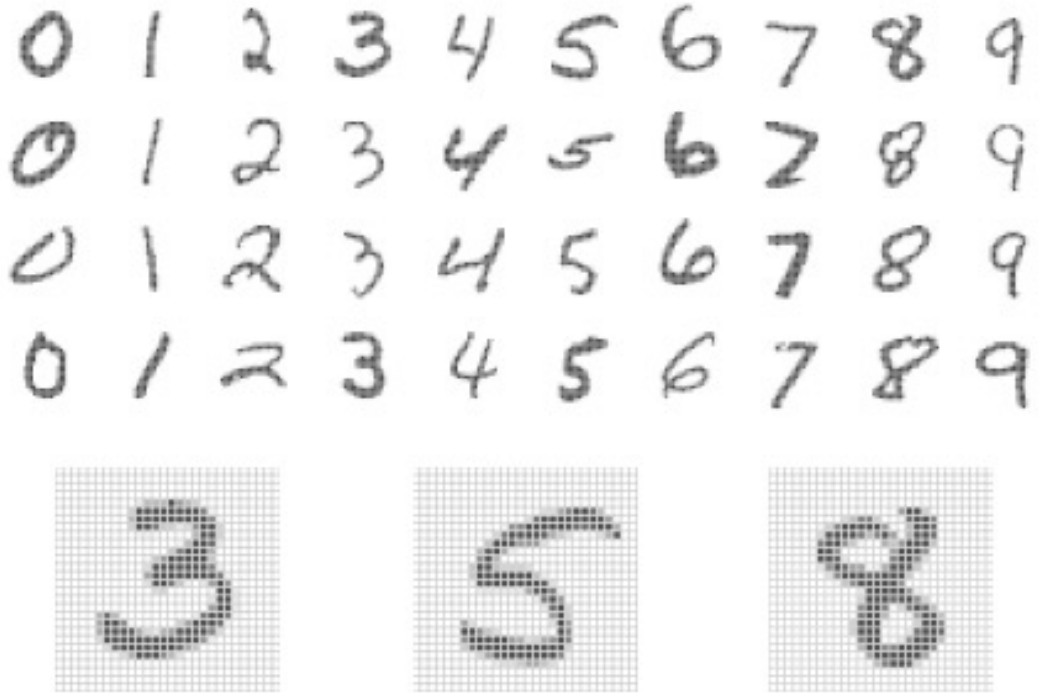
$$g(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j)$$

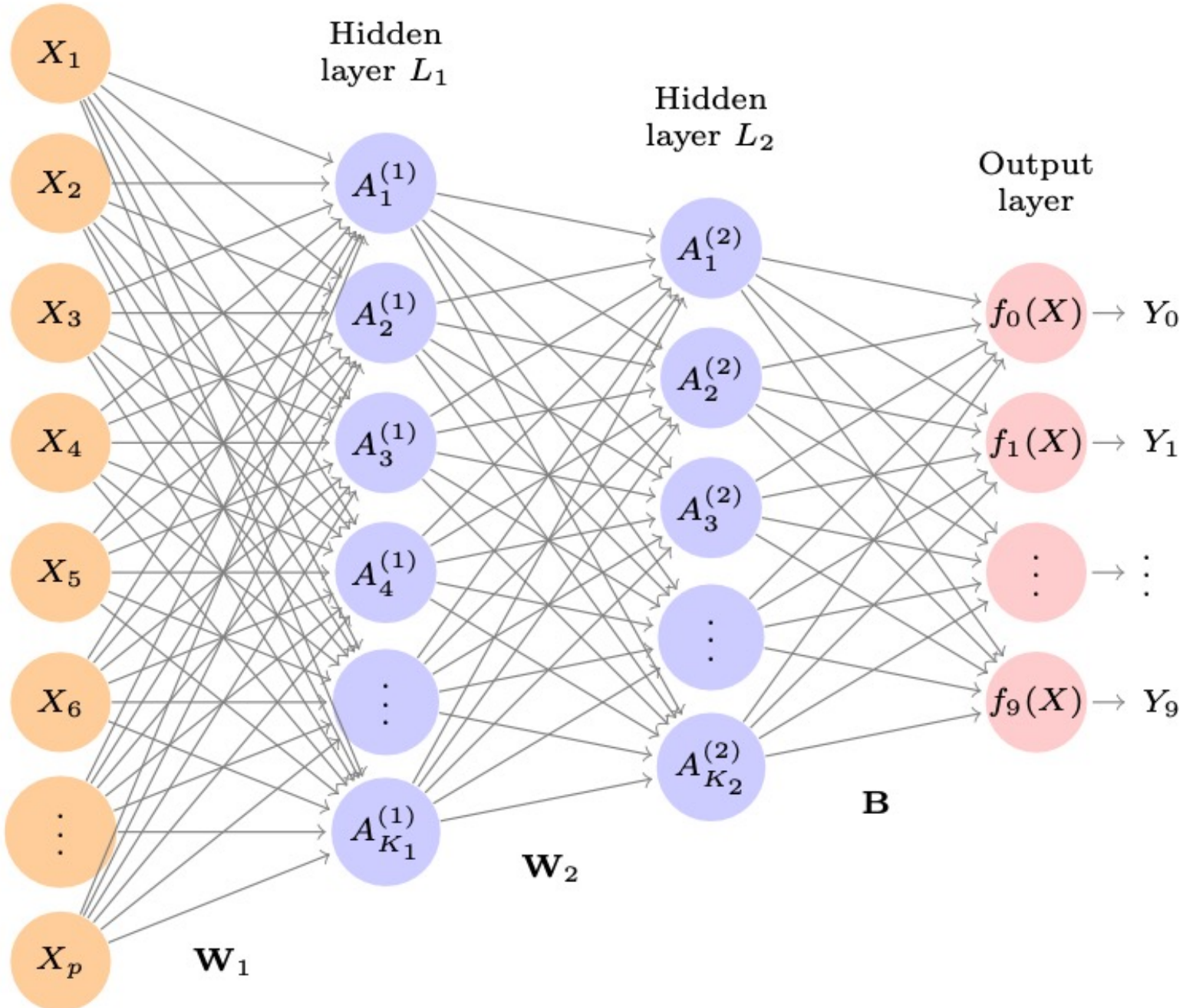
Activation functions in hidden layers are typically nonlinear, otherwise the model collapses to a linear model.



Example: MNIST Digits

Build a two-layer network with 256 units at first layer, 128 units at second layer, and 10 units at output layer.




X_1
X_2
X_3
X_4
X_5
X_6
X_p
Hidden layer L_1
$A_1^{(1)}$
$A_2^{(1)}$
$A_3^{(1)}$
$A_4^{(1)}$
$A_{K_1}^{(1)}$
Hidden layer L_2
$A_1^{(2)}$
$A_2^{(2)}$
$A_3^{(2)}$
$A_{K_2}^{(2)}$
Output layer
$f_0(X) \to Y_0$
$f_1(X) \to Y_1$
$f_9(X) \to Y_9$
$\mathbf{W}_1$
$\mathbf{W}_2$
$\mathbf{B}$

Softmax Output

Let $Z_m = \beta_{m0} + \sum_{\ell=1}^{K_2} \beta_{m\ell} A_\ell^{(2)}$, $m = 0, 1, \ldots, 9$ be 10 linear combinations of activations at second layer.

Softmax output:

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

Cross-Entropy Loss

Negative multinomial log-likelihood:

$$-\sum_{i=1}^{n}\sum_{m=0}^{9} y_{im}\log(f_m(x_i))$$

y_{im} is 1 if true class for observation i is m, else 0

Why Neural Network?

Method	Test Error
Neural Network + Ridge Regularization	2.3%
Neural Network + Dropout Regularization	1.8%
Multinomial Logistic Regression	7.2%
Linear Discriminant Analysis	12.7%