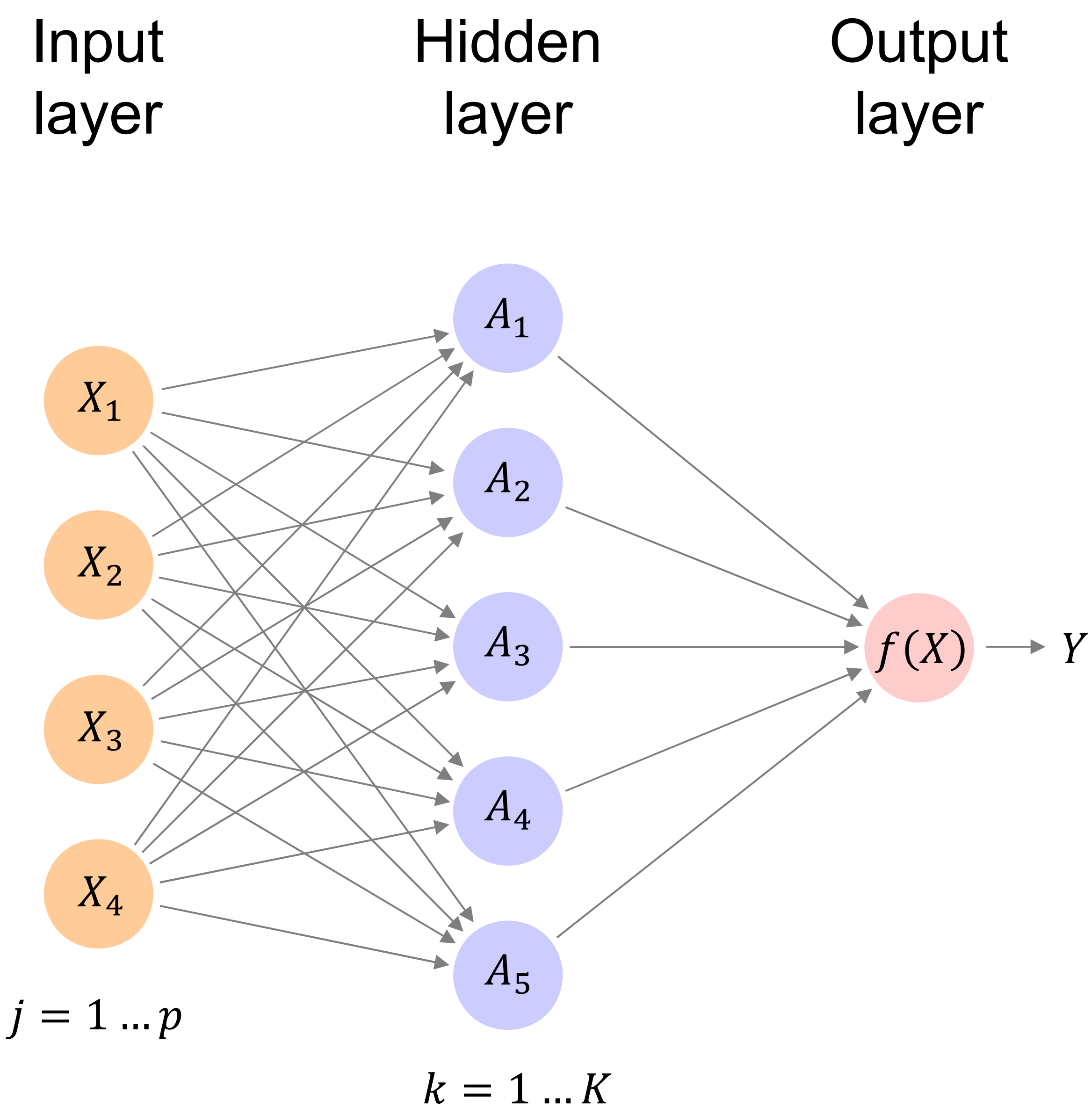
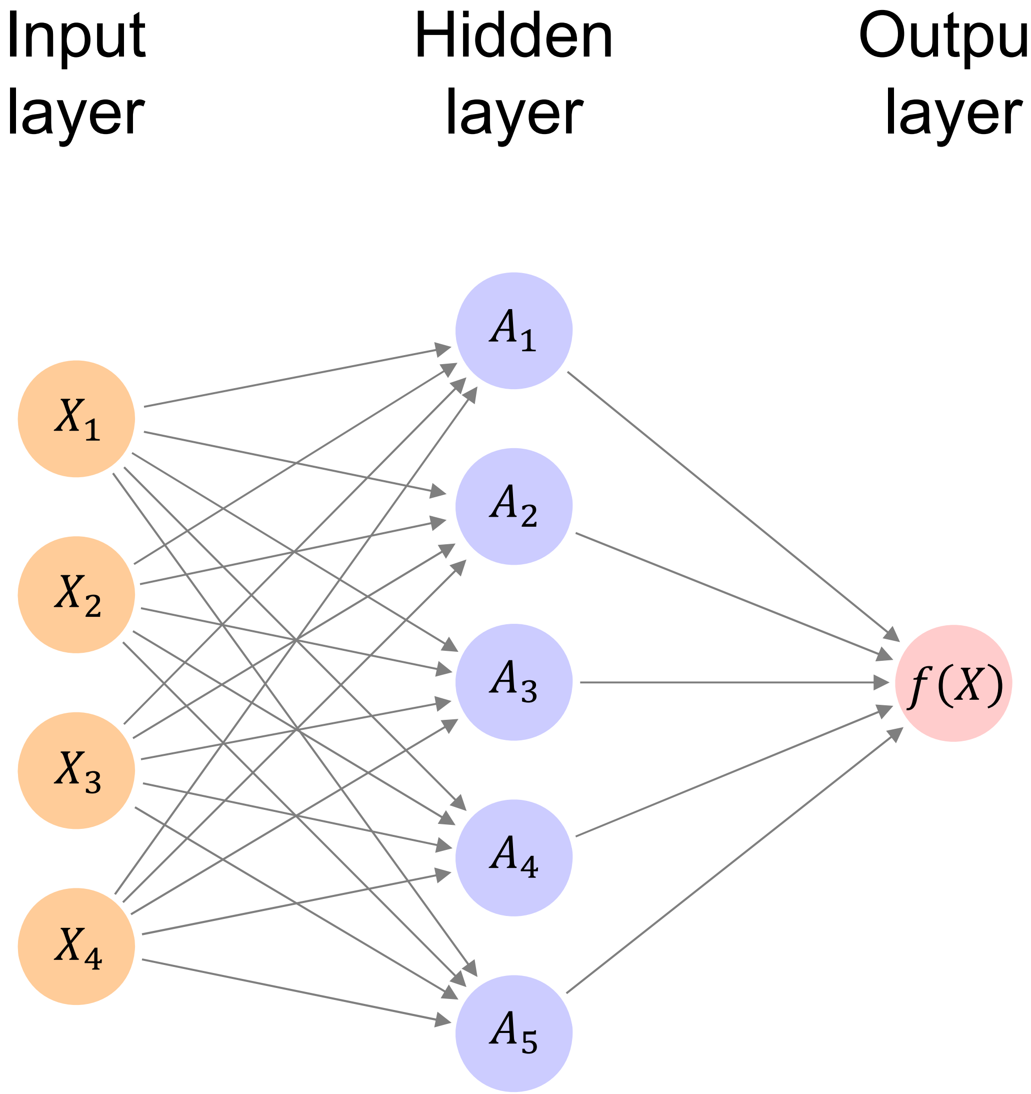
Today

- Neural networks and deep learning
  - Single layer neural networks
  - Multi-layer neural networks
  - Convolutional neural networks
  - Recurrent neural networks

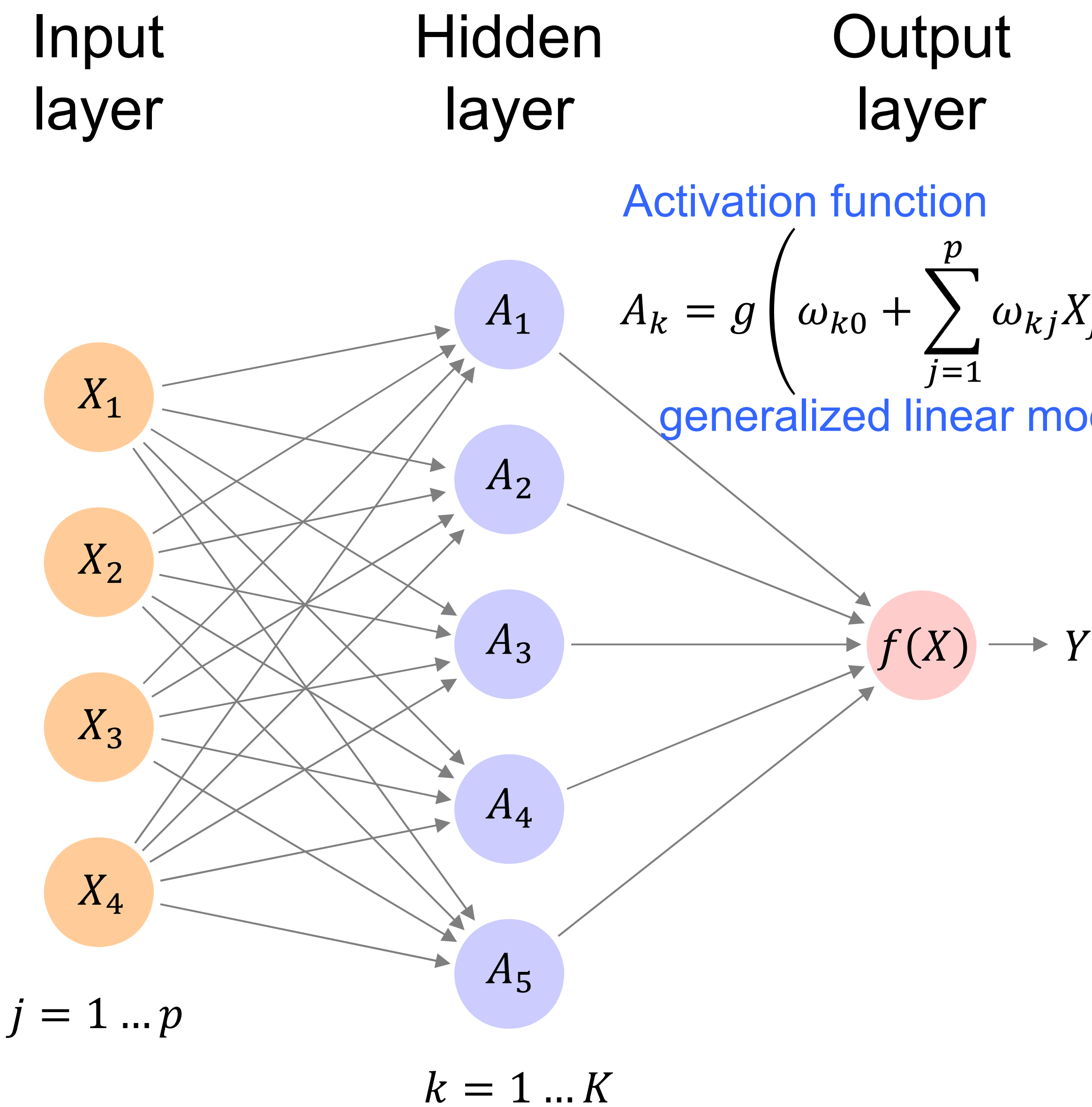
Single layer NN



Single layer NN



Single layer NN



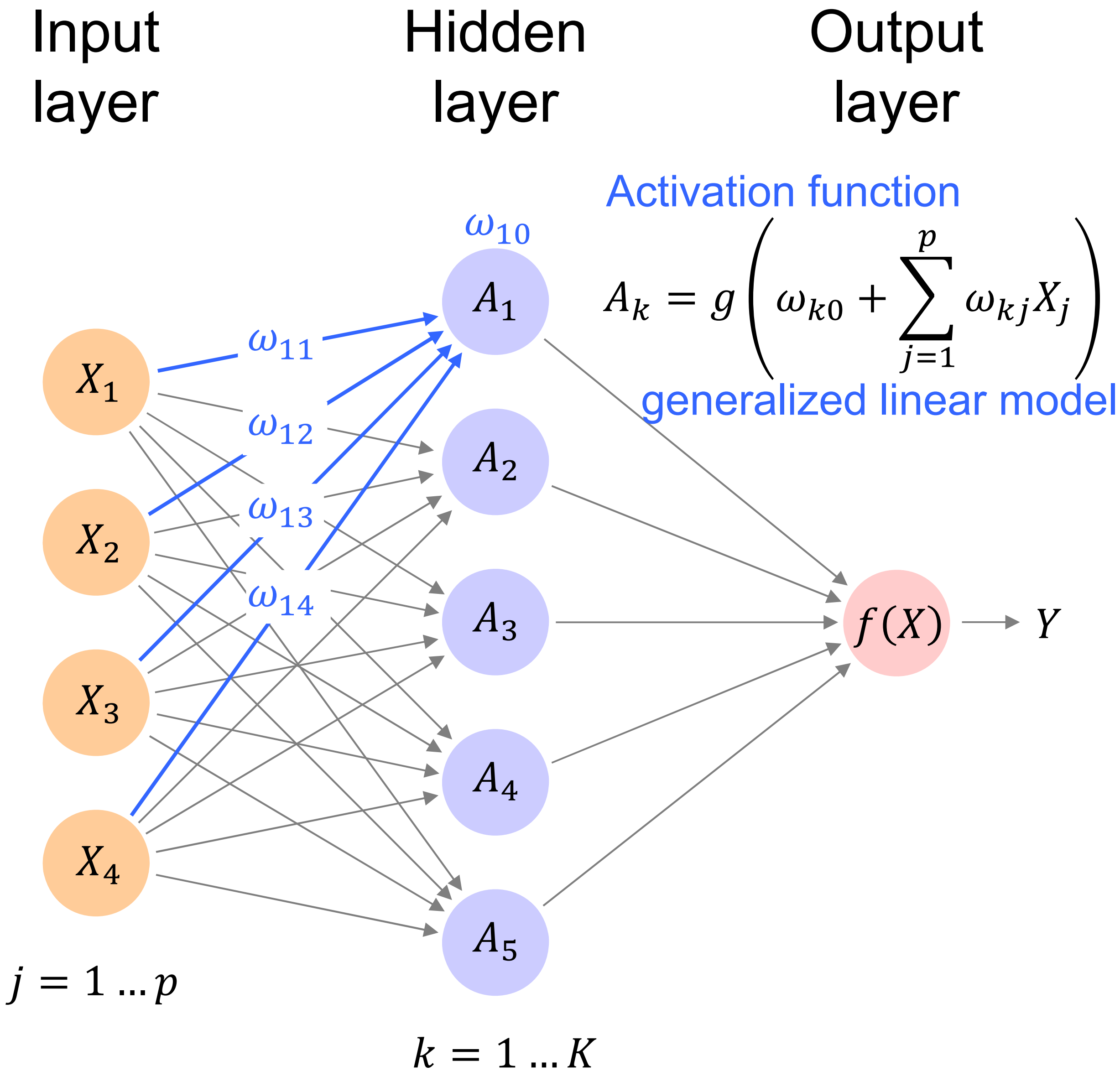
Activation function

$$A_k = g\left(\omega_{k0} + \sum_{j=1}^{p} \omega_{kj} X_j\right)$$

generalized linear model

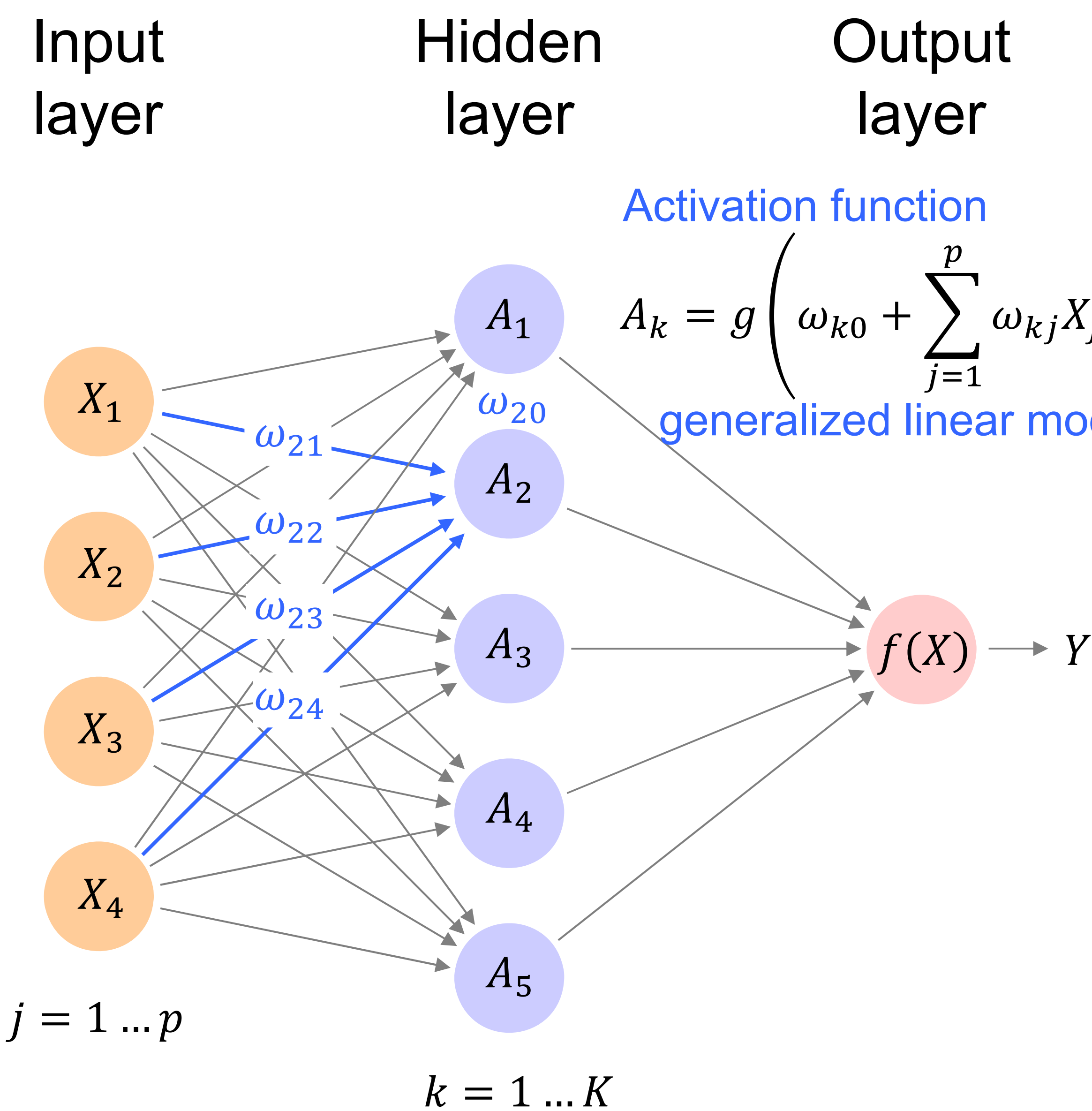
Transform from X to A

Single layer NN



Transform from X to A

Single layer NN



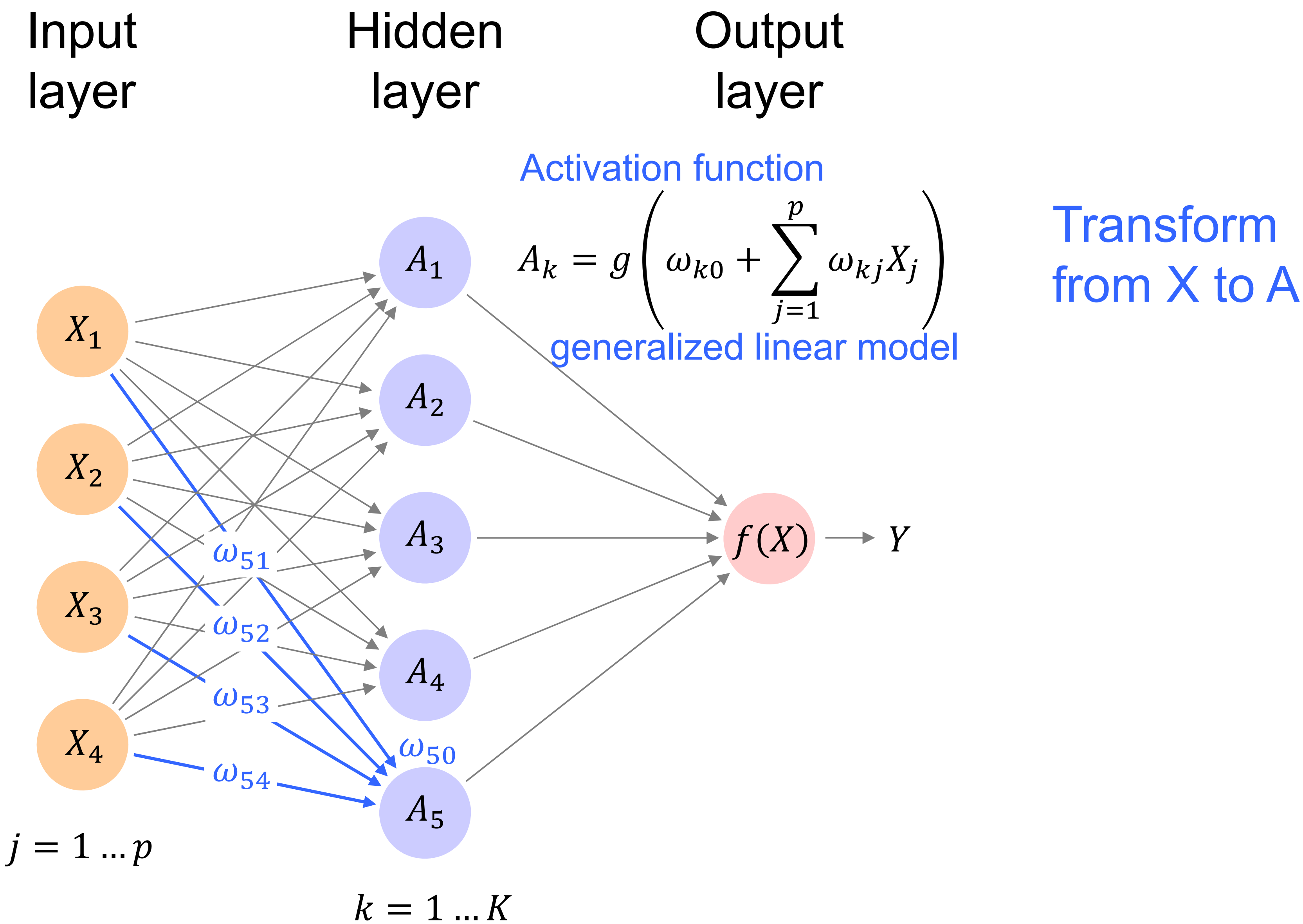
Activation function

$$A_k = g\left(\omega_{k0} + \sum_{j=1}^{p} \omega_{kj} X_j\right)$$

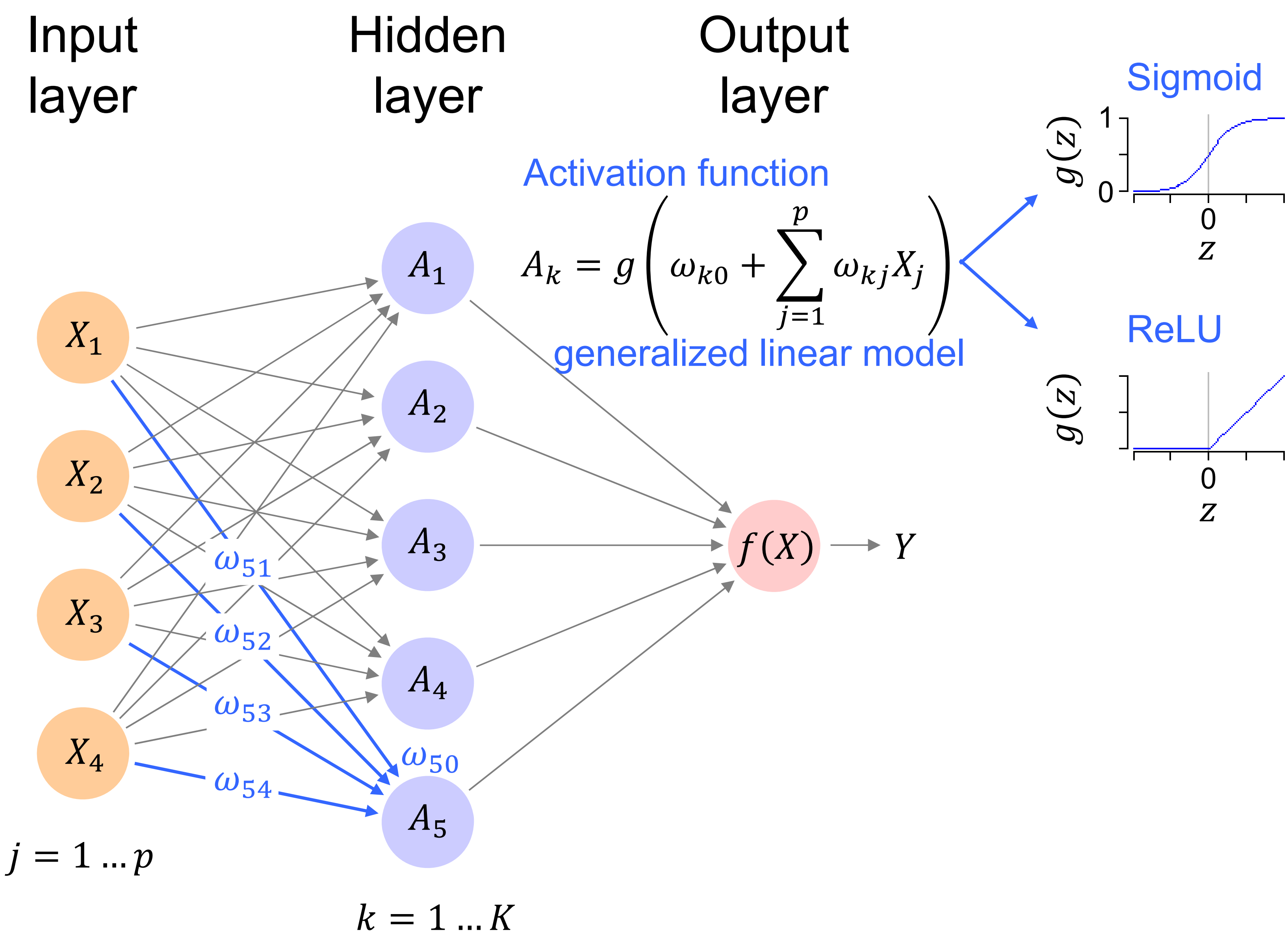
generalized linear model

Transform from X to A

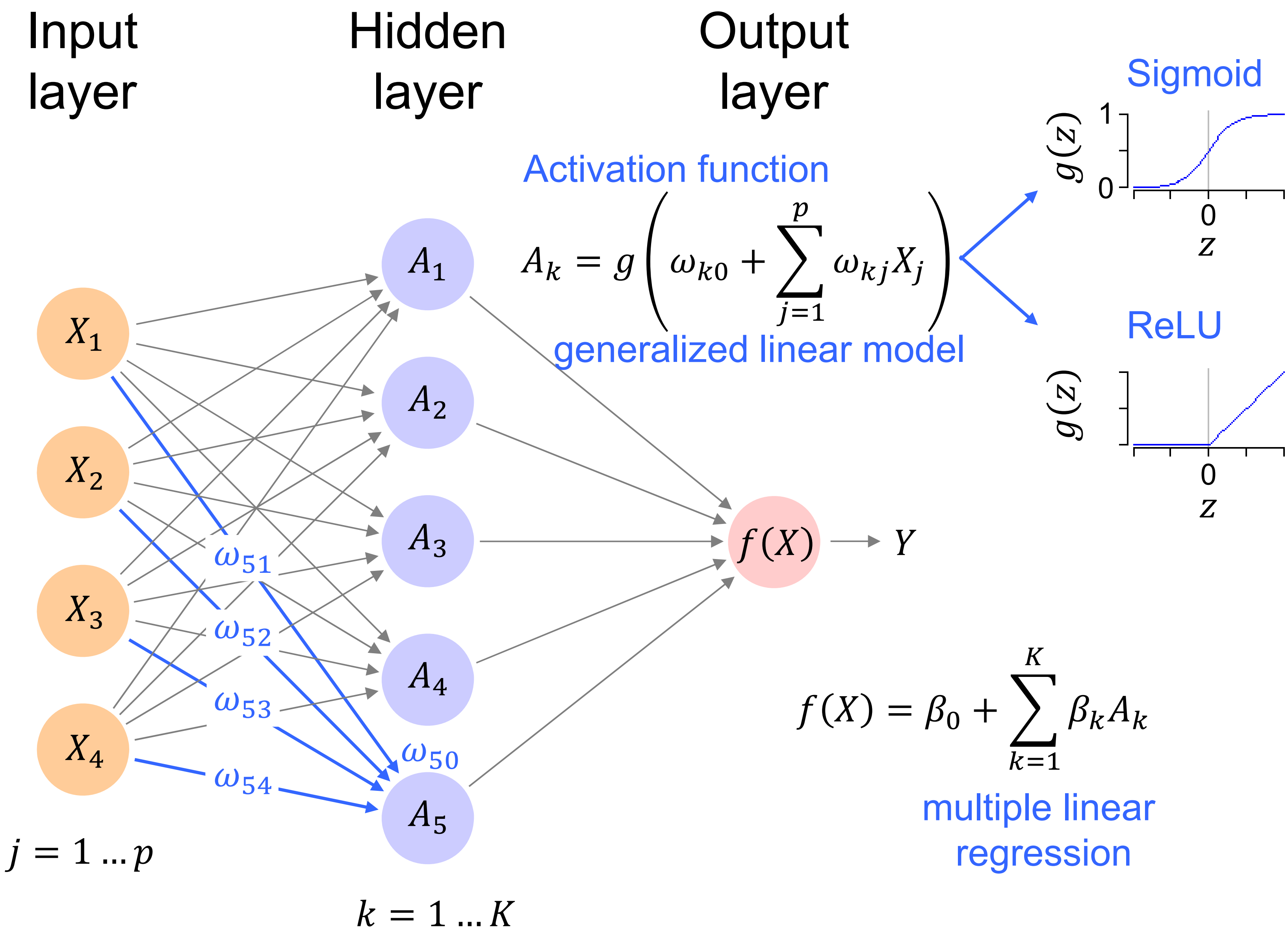
Single layer NN



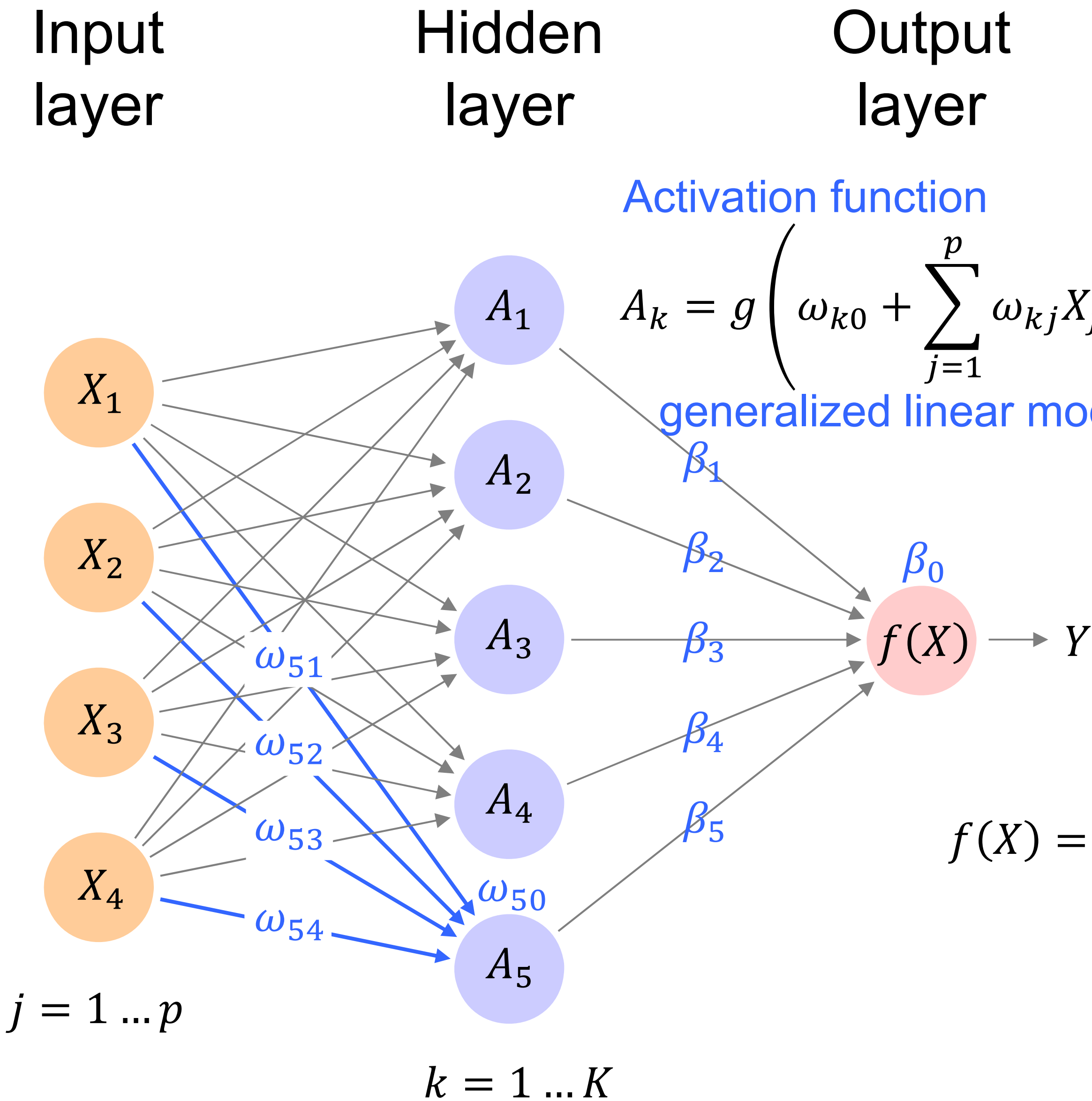
Single layer NN



Single layer NN



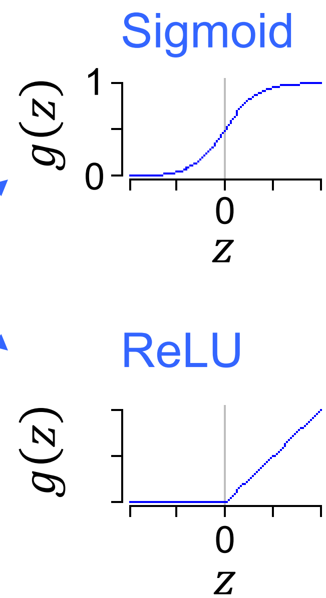
Single layer NN



Activation function

$$A_k = g\left(\omega_{k0} + \sum_{j=1}^{p} \omega_{kj} X_j\right)$$

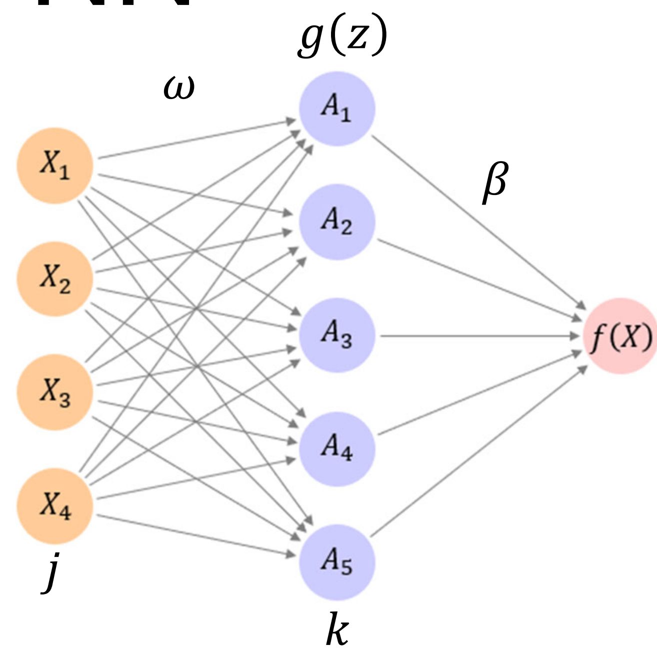
generalized linear model



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

multiple linear regression

Single layer NN



Model algorithm

define $g(z)$

load x_j

set K

set parameters: ω_{kj}, β_k

for each activation unit k in 1:K

    calculate linear predictor: $z_k = \omega_{k0} + \sum_j \omega_{kj} x_j$

    calculate nonlinear activation: $A_k = g(z_k)$

calculate linear model: $f(x) = \beta_0 + \sum_k \beta_k A_k$

return $f(x)$

Training algorithm

Loss function (SSQ)

$$R(\theta) = \frac{1}{2}\sum_{i=1}^{n}(y_i - f_\theta(x_i))^2$$

$$\theta = \begin{matrix} \beta_0 \\ \beta_1 \\ \vdots \\ \omega_{10} \\ \omega_{11} \\ \vdots \end{matrix}$$

Training algorithm

Gradient descent

guess θ (typically random)

set ρ

do until $R(\theta)$ fails to decrease

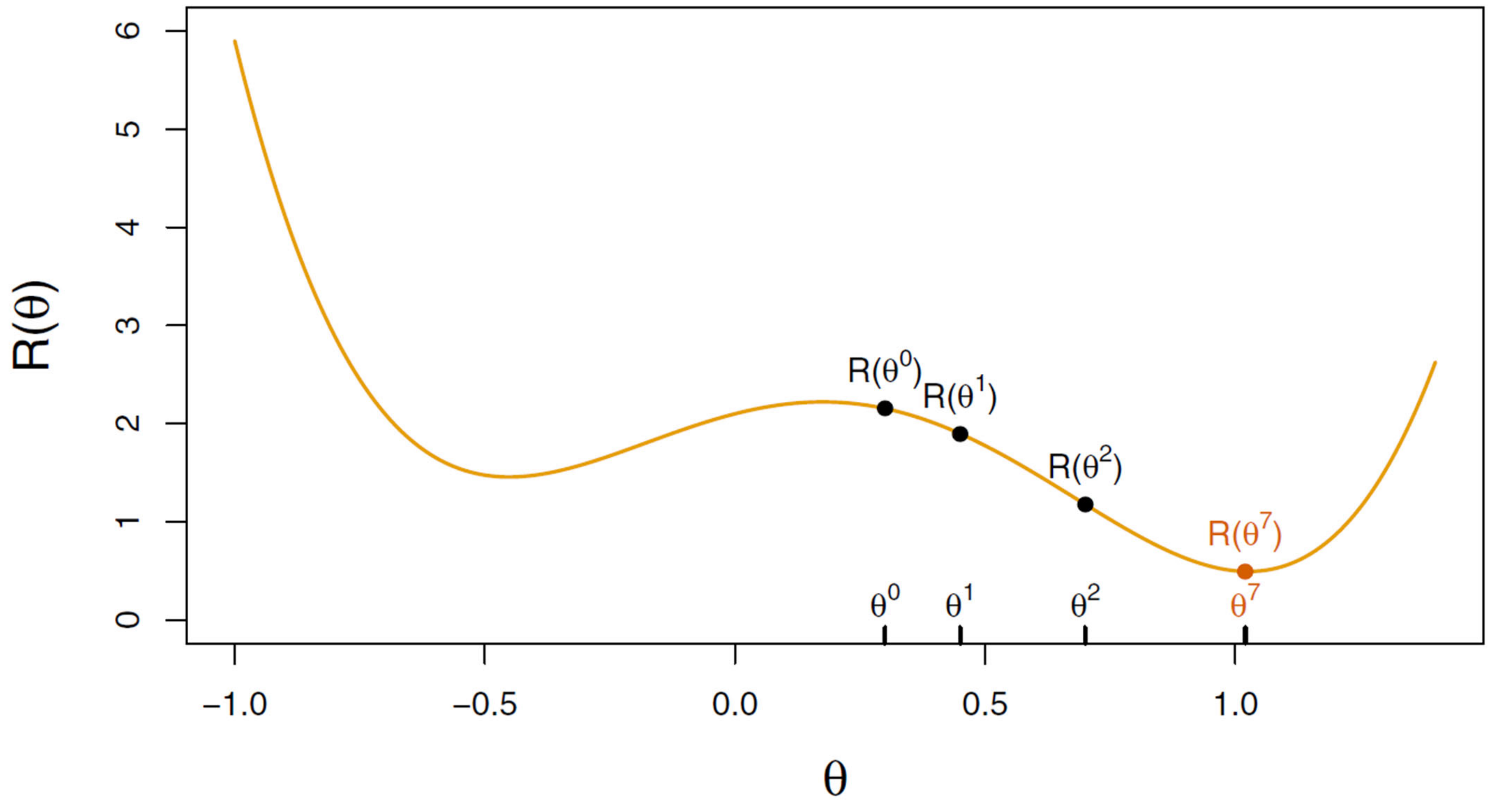
calculate derivative of $R(\theta)$: $\nabla R(\theta) = \frac{\delta R(\theta)}{\delta \theta}$

$\theta \leftarrow \theta - \rho \nabla R(\theta)$

Back propagation

Training algorithm

Gradient descent of loss surface (1D example)



Training algorithm

Stochastic gradient descent

guess θ (typically random)

set ρ

do until $R(\theta)$ fails to decrease

randomly sample the data

calculate derivative of $R(\theta)$: $\nabla R(\theta) = \frac{\delta R(\theta)}{\delta\theta}$

$\theta \leftarrow \theta - \rho\nabla R(\theta)$

Back propagation