신경망 네트워크와 수학적 기반

Least squares data fitting

Setup

- we believe a scalar y and an n-vector x are related by *model*

$$y \approx f(x)$$

- x is called the *independent variable*
- y is called the *outcome* or *response variable*
- $f : \mathbf{R}^n \to \mathbf{R}$ gives the relation between x and y
- often x is a feature vector, and y is something we want to predict
- we don't know f, which gives the 'true' relationship between x and y

Least squares data fitting
Data

- we are given some *data* also called *observations*, *examples*, *samples*, or *measurements*

$$x^{(1)}, \ldots, x^{(N)}, \qquad y^{(1)}, \ldots, y^{(N)}$$

- $x^{(i)}, y^{(i)}$ is i-th *data pair*
- $x_j^{(i)}$ is the j-th component of i-th data point $x^{(i)}$

Least squares data fitting
Model

- choose model $\hat{f} : \mathbf{R}^n \to \mathbf{R}$, a guess or *approximation* of f
- *linear in the parameters* model form:

$$\hat{f}(x) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x)$$

- $f_i : \mathbf{R}^n \to \mathbf{R}$ are *basis functions* that we choose
- θ_i are *model parameters* that we choose
- $\hat{y}^{(i)} = \hat{f}(x^{(i)})$ is (the model's) *prediction* of $y^{(i)}$
- we'd like $\hat{y}^{(i)} \approx y^{(i)}$, *i.e.*, model is consistent with observed data

Least squares data fitting

- *prediction error* or *residual* is $r_i = y^{(i)} - \hat{y}^{(i)}$
- *least squares data fitting*: choose model parameters θ_i to minimize RMS prediction error on data set

$$\left(\frac{(r^{(1)})^2 + \cdots + (r^{(N)})^2}{N}\right)^{1/2}$$

- this can be formulated (and solved) as a least squares problem

Least squares data fitting

- express $y^{(i)}$, $\hat{y}^{(i)}$, and $r^{(i)}$ as N-vectors
  - $y^{\mathrm{d}} = (y^{(1)}, \ldots, y^{(N)})$ is vector of outcomes
  - $\hat{y}^{\mathrm{d}} = (\hat{y}^{(1)}, \ldots, \hat{y}^{(N)})$ is vector of predictions
  - $r^{\mathrm{d}} = (r^{(1)}, \ldots, r^{(N)})$ is vector of residuals
- $\mathrm{rms}(r^{\mathrm{d}})$ is *RMS prediction error*
- define $N \times p$ matrix A with elements $A_{ij} = f_j(x^{(i)})$, so $\hat{y}^{\mathrm{d}} = A\theta$
- least squares data fitting: choose θ to minimize

$$\|r^{\mathrm{d}}\|^2 = \|y^{\mathrm{d}} - \hat{y}^{\mathrm{d}}\|^2 = \|y^{\mathrm{d}} - A\theta\|^2 = \|A\theta - y^{\mathrm{d}}\|^2$$

- $\hat{\theta} = (A^TA)^{-1}A^Ty$ (if columns of A are linearly independent)
- $\|A\hat{\theta} - y\|^2/N$ is *minimum mean-square (fitting) error*

Fitting a constant model

- simplest possible model: $p = 1$, $f_1(x) = 1$, so model $\hat{f}(x) = \theta_1$ is a constant
- A = 1, so

$$\hat{\theta}_1 = (\mathbf{1}^T\mathbf{1})^{-1}\mathbf{1}^T y^{\mathrm{d}} = (1/N)\mathbf{1}^T y^{\mathrm{d}} = \mathbf{avg}(y^{\mathrm{d}})$$

- the mean of $y^{(1)}, \ldots, y^{(N)}$ is the least squares fit by a constant
- MSE is $\mathrm{std}(y^{\mathrm{d}})^2$; RMS error is $\mathrm{std}(y^{\mathrm{d}})$

Straight-line fit

- $p = 2$, with $f_1(x) = 1$, $f_2(x) = x$
- model has form $\hat{f}(x) = \theta_1 + \theta_2 x$
- matrix A has form

$$A = \begin{bmatrix} 1 & x^{(1)} \\ 1 & x^{(2)} \\ \vdots & \vdots \\ 1 & x^{(N)} \end{bmatrix}$$

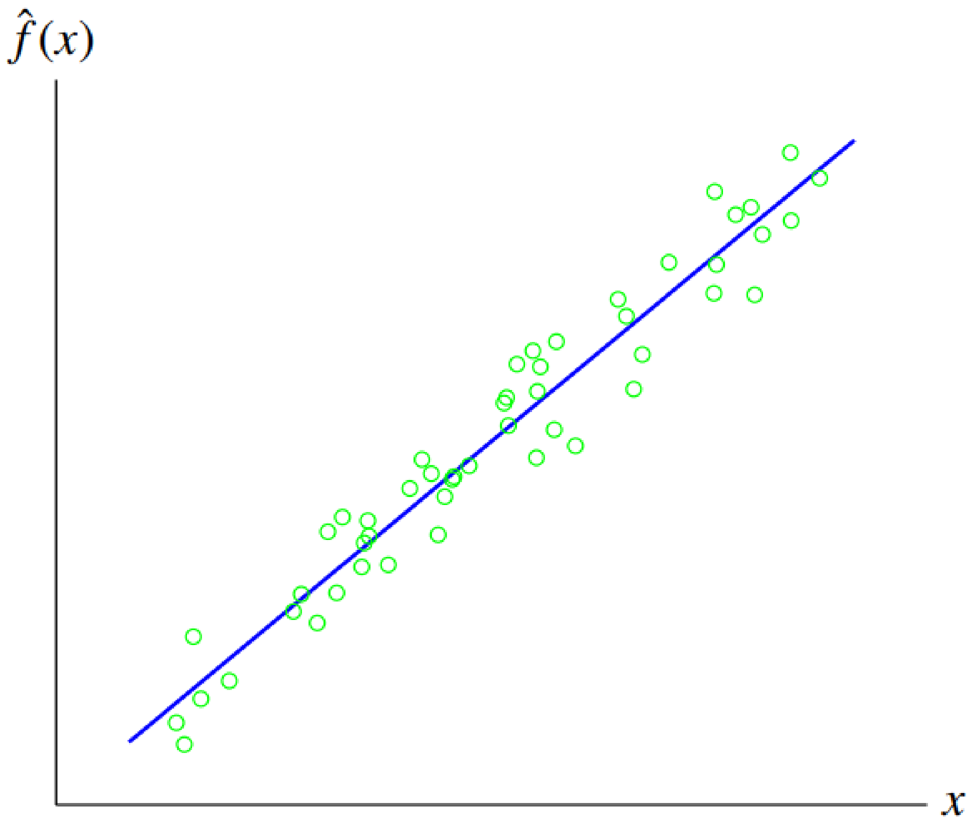
- can work out $\hat{\theta}_1$ and $\hat{\theta}_2$ explicitly:

$$\hat{f}(x) = \mathbf{avg}(y^{\mathrm{d}}) + \rho \frac{\mathbf{std}(y^{\mathrm{d}})}{\mathbf{std}(x^{\mathrm{d}})}(x - \mathbf{avg}(x^{\mathrm{d}}))$$

where $x^{\mathrm{d}} = (x^{(1)}, \ldots, x^{(N)})$

Least squares data fitting

Example



Least squares data fitting

Polynomial fit

- $f_i(x) = x^{i-1}$, $i = 1, \dots, p$
- **model is a polynomial of degree less than *p***

$$\hat{f}(x) = \theta_1 + \theta_2 x + \cdots + \theta_p x^{p-1}$$

(here x^i means scalar x to i-th power; $x^{(i)}$ is i-th data point)

- **A is Vandermonde matrix**

$$A = \begin{bmatrix} 1 & x^{(1)} & \cdots & (x^{(1)})^{p-1} \\ 1 & x^{(2)} & \cdots & (x^{(2)})^{p-1} \\ \vdots & \vdots & & \vdots \\ 1 & x^{(N)} & \cdots & (x^{(N)})^{p-1} \end{bmatrix}$$

Least squares data fitting
Example

$N = 100$ data points