Advanced Topics in Neural Networks

MATH 270

Shizhe Zhang

Last updated: November 21, 2025

Contents

※ Lecture 1

1.1 Information

Instructor: Dr. Haiyan Huang Tu/Th 11:00am-12:29pm Lecture, 106 Stanley Office: 317 Evans

Office hour: Thursday 1-2pm 317 Evans

1.2 Introduction to Neural Networks

Supervised Learning Supervised learning: regression, classification

$$D = \{(x_i, y_i)\}_{i=1}^N, x_i \in \mathbb{R}^d, y_i \in \mathbb{R}^k$$

Try to find a function $f : \mathbb{R}^d \to \mathbb{R}^k$ such that $f(x_i) \approx y_i$ for all i.

Approach:

(1) Choose parametrization of functions f_θ $\theta \in \mathbb{R}^m$

(2) Optimize θ by gradient descent.

$$D \longrightarrow \mathcal{L}_D$$

Metaphor: $\theta \longrightarrow f_\theta$

Deep learning: f_θ is a neural network.

linear regression $\longrightarrow$ affine linear

Unsupervised Learning Goal is to learn the patterns and structure from data. e.g. clustering, dimensionality reduction, generative models.

Generative models: $D = \{(x_i)\}$, $\qquad x_i \overset{i.i.d.}{\sim} P$ for some prob distribution.

Want to learn P.

Example ***Generative models***

Chatgpt
P = distribution of natural language text.

1.3 Linear Regression

$$F(x) = xW + b, \qquad W \in \mathbb{R}^{d\times k}, b \in \mathbb{R}^k$$

W is the weight matrix, b is the bias vector. Simplification: $b = 0$

Need to quantify the error of $F_{W,b}$ on D. MAE,MSE.

$$\frac{1}{N}\sum_{i=1}^{N} |F_{W,b}(x_i) - y_i|_{1or2}$$

Let $Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}, X = \begin{pmatrix} ---x_1--- \\ ---x_2--- \\ \vdots \\ ---x_N--- \end{pmatrix}$

$$N * MSE = \sum_{i=1}^{N} |F_{W,b}(x_i) - y_i|_2^2 = \|XW - Y\|_F^2 = (Y - XW)^T(Y - XW)$$

$$\frac{\partial}{\partial W}\mathcal{L}_D(W) = -2X^T(Y - XW) = 0 \Rightarrow X^TXW = X^TY$$

$$W = (X^TX)^{-1}X^TY \quad \text{if } X^TX \text{ is invertible}$$

Beyond Linear Models Neural Networks: piecewise linear functions.

$$x \xrightarrow{g^*} Z \xrightarrow{h^*} a \to Z \to \cdots \to Z = y$$

g is affine linear, h is non-linear.

Theorem ***Universal Approximation Theorem***

NN can approximate any continuous f on compact set to any desired degree of accuracy as soon as $L \geq 2$.

A single neuron i in a layer l.

$$g_i^l(a^{l-1}) = z_i^l = \mathbf{a^{l-1}w_i^l} + b_i^l, \qquad a_i^l = h(z_i^l)$$

$$h_i^l(z_i^l) = ReLU(z_i^l) = \max(0, z_i^l) = a_i^l$$

From features to representations.

Example

NN to classify proofs as correct or incorrect.
Input: 'obvious', 'clear', 'well-known' neurons $\to$ 'handwavy' neuron

$\theta = \{\text{weights and biases of the network}\}$

Next: find loss function to measure error on D.

- Regression MSE:

$$\mathcal{L}_D(\theta) = \frac{1}{N}\sum_{i=1}^{N} |f_\theta(x_i) - y_i|^2$$

- Classification: cross-entropy loss

$$\mathcal{L}_D(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k} y_{i,j} \log f_\theta(x_i)_j$$

- NN will compute

$$Model\ output : F_\theta(x, y) = P(y|x), \qquad f_\theta(x) = \arg\max_y F_\theta(x, y)$$

$$\mathcal{L}_{(x_i, y_i)}(\theta) = -\log F_\theta(x_i, y_i)$$

Rmk:

$$\sum_y F_\theta(x, y) = 1, \qquad F_\theta(x, y) \geq 0$$

Example *Softmax*

$$F_\theta(x, y) = \frac{e^{\phi_\theta(x,y)}}{\sum_{y'} e^{\phi_\theta(x,y')}}$$

1.4 Train and Test

Split data into 3 parts: Training: 80%, Validation: 10%, Test: 10%

- Learnable parameters: W, b
- Hyperparameters: learning rate, number of layers, number of neurons per layer, batch size, ...

Tensorflow Playground for Visualization of NN

1.5 Backpropagation

$$\nabla \mathcal{L}_D(\theta) = \left(\frac{\partial \mathcal{L}_D}{\partial \theta_1}, \frac{\partial \mathcal{L}_D}{\partial \theta_2}, \ldots, \frac{\partial \mathcal{L}_D}{\partial \theta_m} \right)$$

$$\nabla_{z^l}(\mathcal{L}) = \nabla_{z^{l+1}}(\mathcal{L})\, a^l \ \mathrm{diag}(dh^l)$$