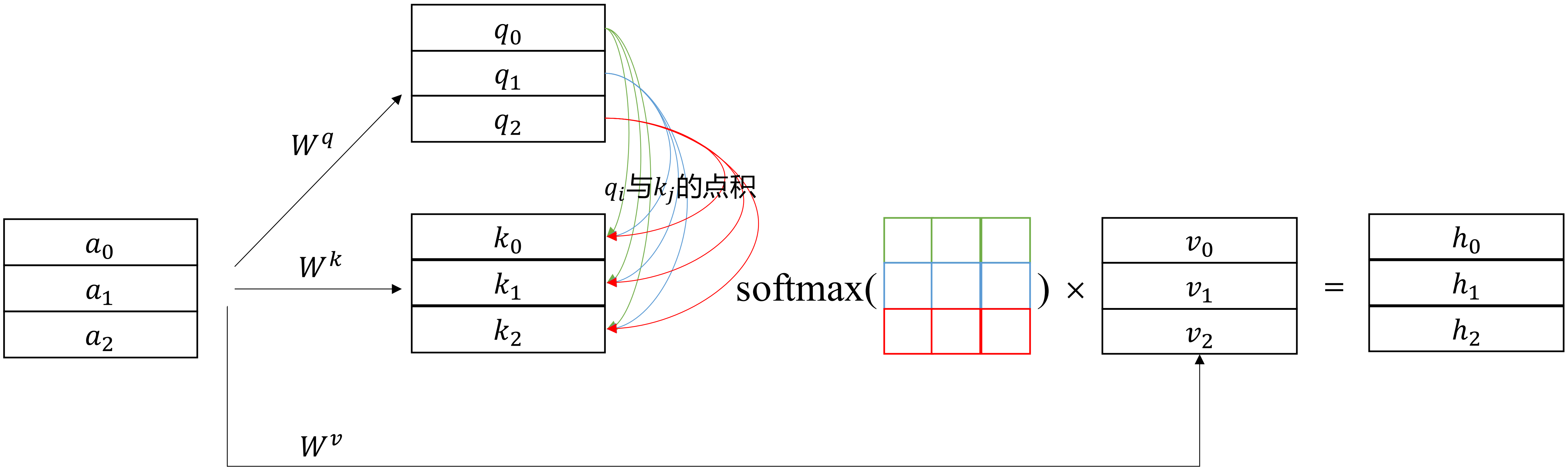
四川大学
SICHUAN UNIVERSITY

机器学习-第十三章 线性可分支持向量机

教师：胡俊杰 副教授

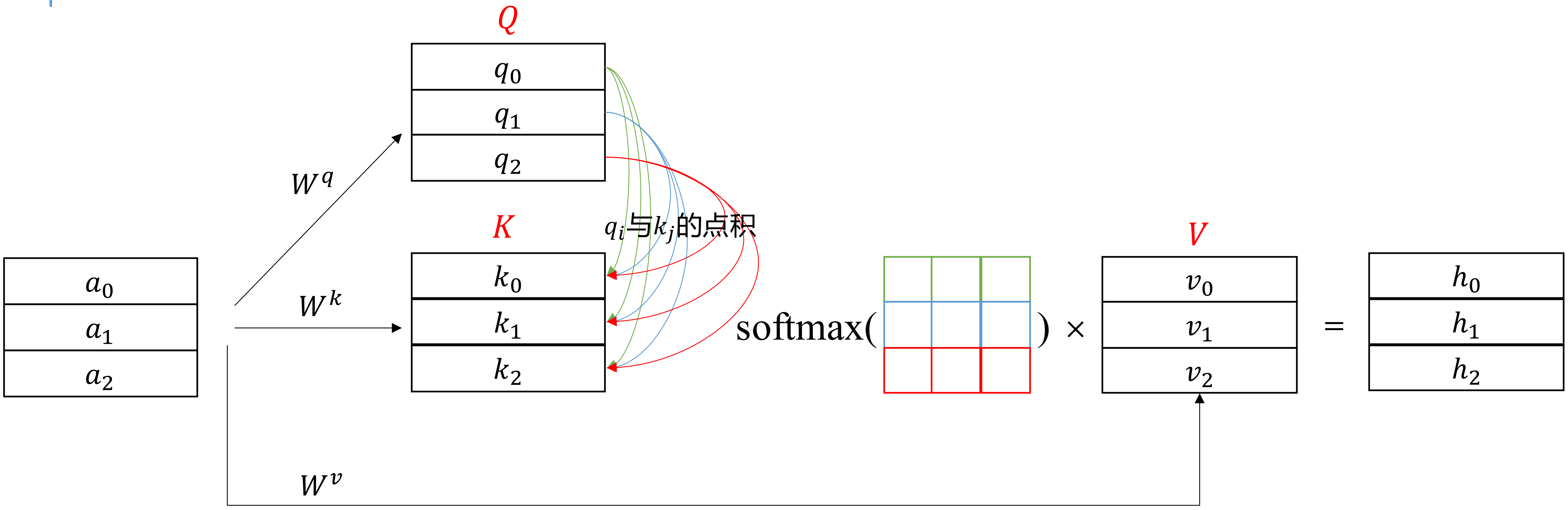
邮箱：hujunjie@scu.edu.cn

自注意力机制



- a_t为序列输入，$a_t \in \mathbb{R}^{1\times n}$
- W^q, W^k, W^v为可学习矩阵
- $W^q \in \mathbb{R}^{n\times m}, W^k \in \mathbb{R}^{n\times m}, W^v \in \mathbb{R}^{n\times d}$
- $q_t \in \mathbb{R}^{1\times m}, k_t \in \mathbb{R}^{1\times m}, v_t \in \mathbb{R}^{1\times d}$

自注意力机制

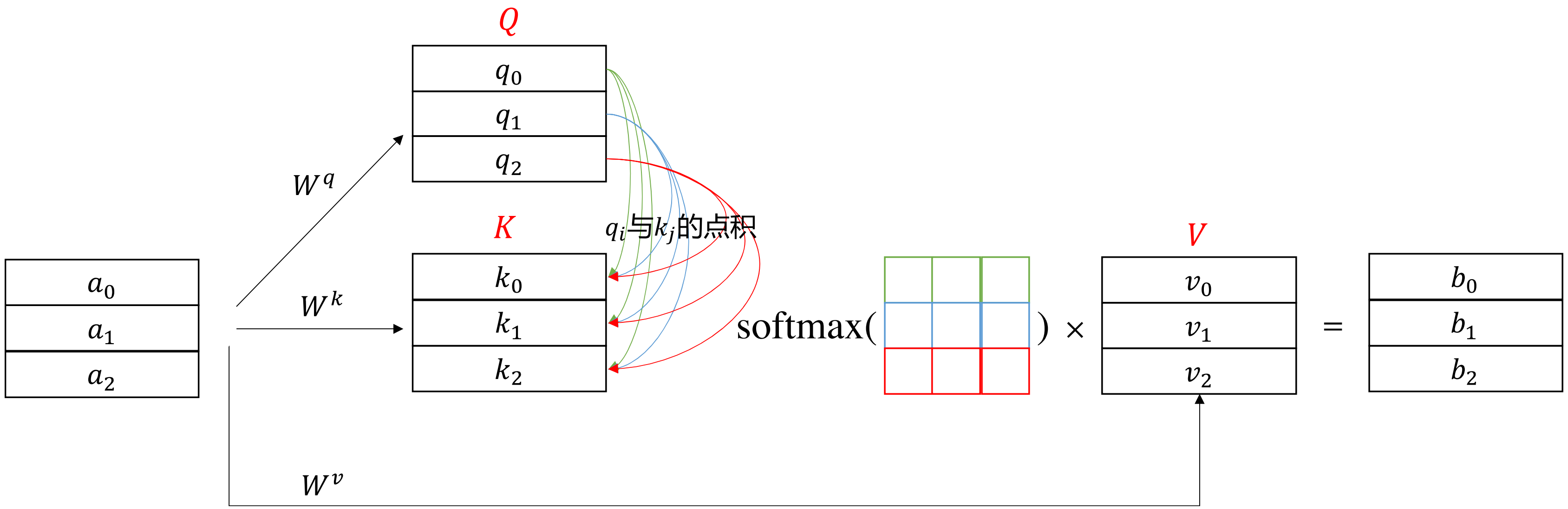


$$\text{Attention}(Q,K,V)=\text{softmax}\left(\frac{QK^T}{\sqrt{n}}\right)V$$

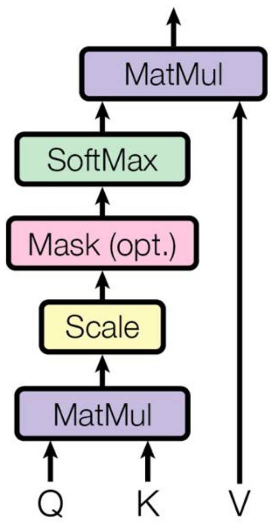
$$\text{softmax}(x_i)=\frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

- n代表a_t的维度， n 过大时， 将导致QK^T方差过大，softmax归一化后的数值分布差异将过大，影响计算的梯度强度

自注意力机制



Scaled Dot-Product Attention



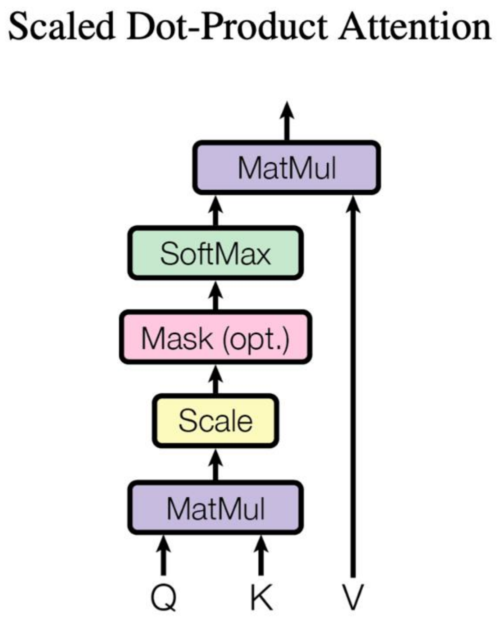
Attention is **all you need**

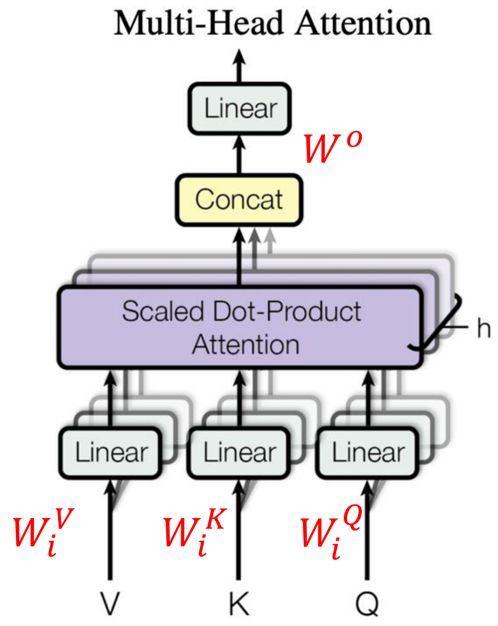
A Vaswani, N Shazeer, N Parmar… - Advances in neural …, 2017 - proceedings.neurips.cc

… to attend to **all** positions in the decoder up to and including that position. **We need** to prevent … **We** implement this inside of scaled dot-product **attention** by masking out (setting to −∞) …

☆ Save 99 Cite Cited by 178597 Related articles All 73 versions ≫

自注意力机制



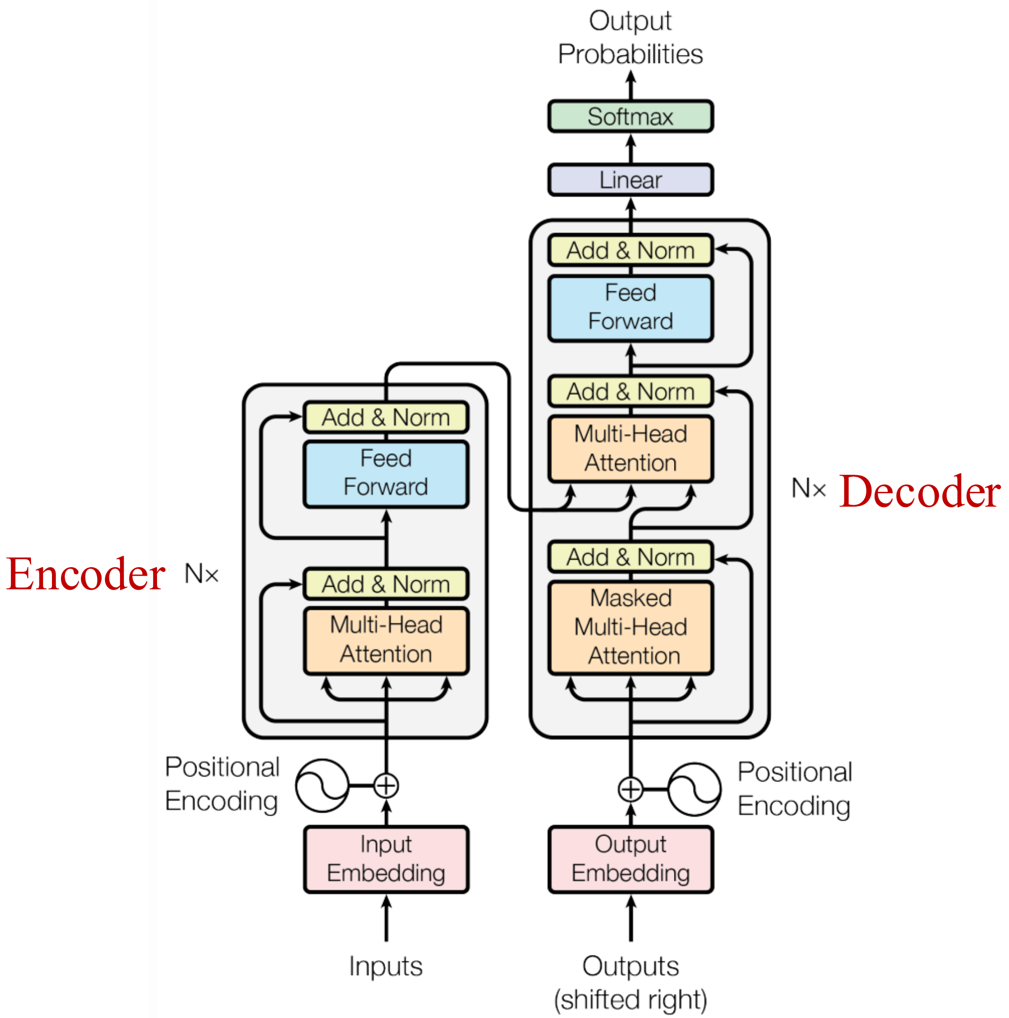


$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^o$$

$$\mathrm{head_i} = \mathrm{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right)$$

- h组并行的自注意力计算，增加特征的多样性（联想CNN的通道数）
- h组输出Concat后由W^o进行映射

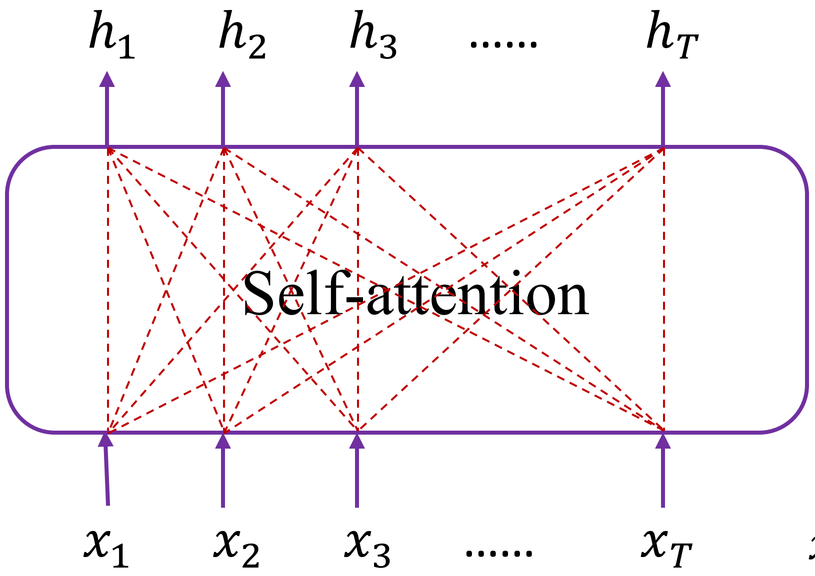
Encoder-decoder Architecture of Transformer



Goal of encoder: get contextualized representation for each token

Self-attention in encoder

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

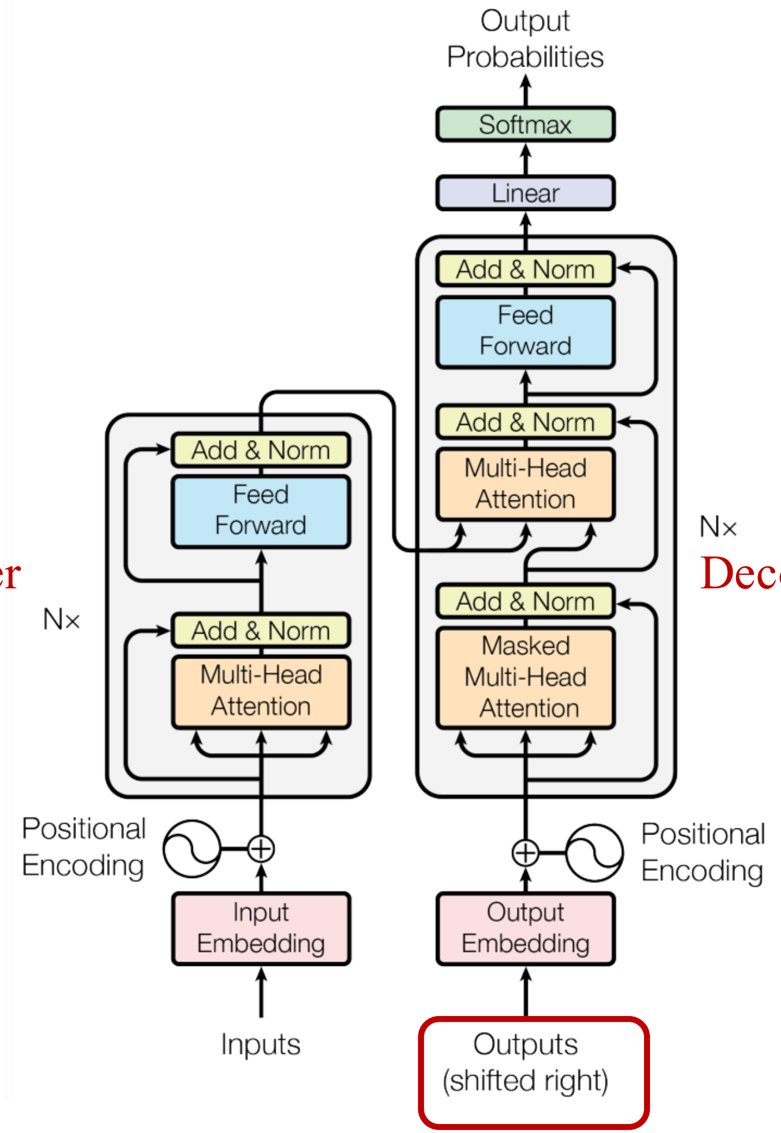


h_t: hidden representation of token t

h_t is related to every input token

x_t: embedding of token t

Encoder-decoder Architecture of Transformer



Encoder

Decoder

Goal of decoder: accomplish the next token prediction task autoregressively

Suppose the target sequence is [“A”, “B”, “C”]

input of decoder: [<SOS>, “A”, “B”, “C”]

label: [“A”, “B”, “C”, <EOS>]

Task of decoder:

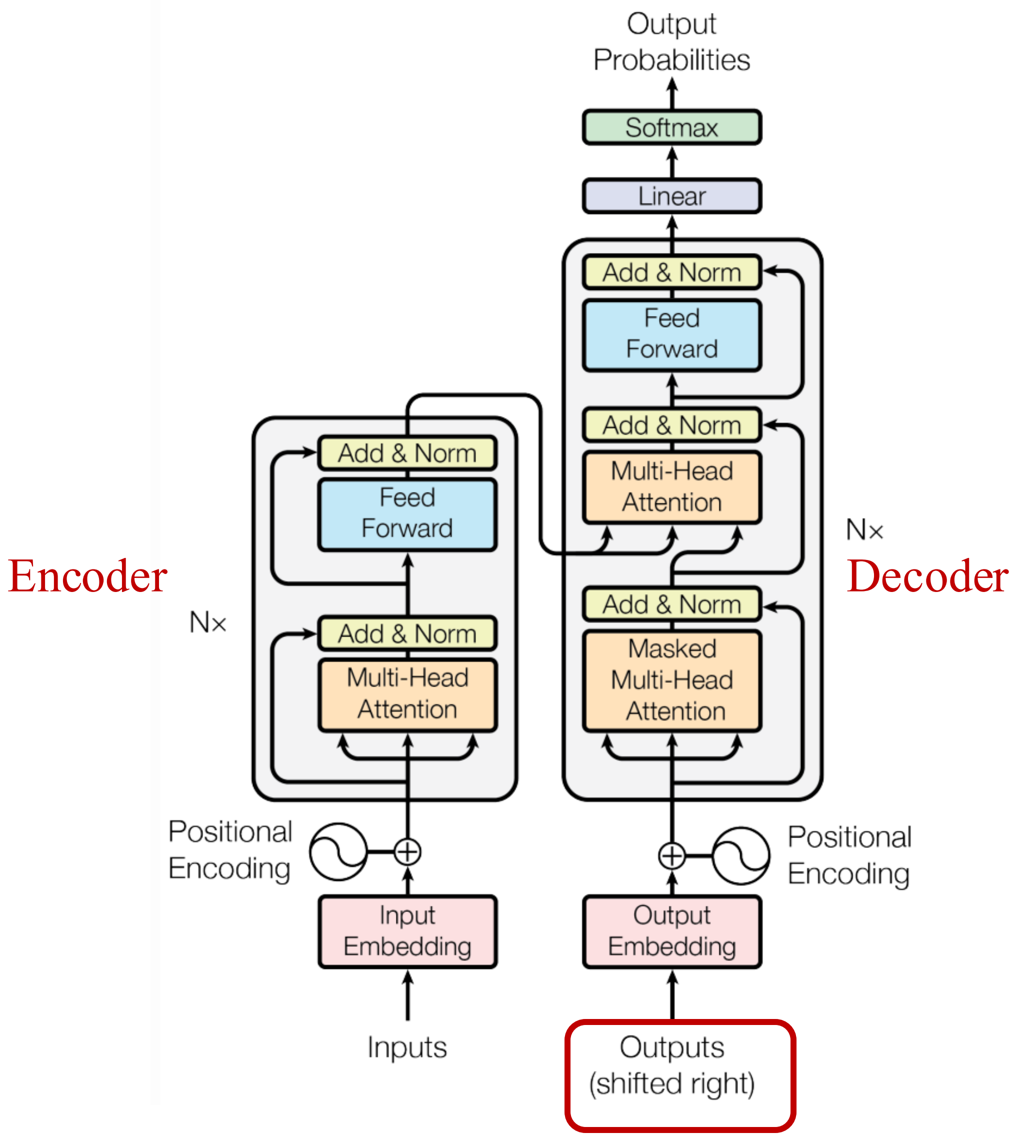
input [<SOS>], predict “A”

input [<SOS>, “A”], predict “B”

input [<SOS>, “A”, “B”], predict “C”

input [<SOS>, “A”, “B”, “C”], predict <EOS>

Encoder-decoder Architecture of Transformer



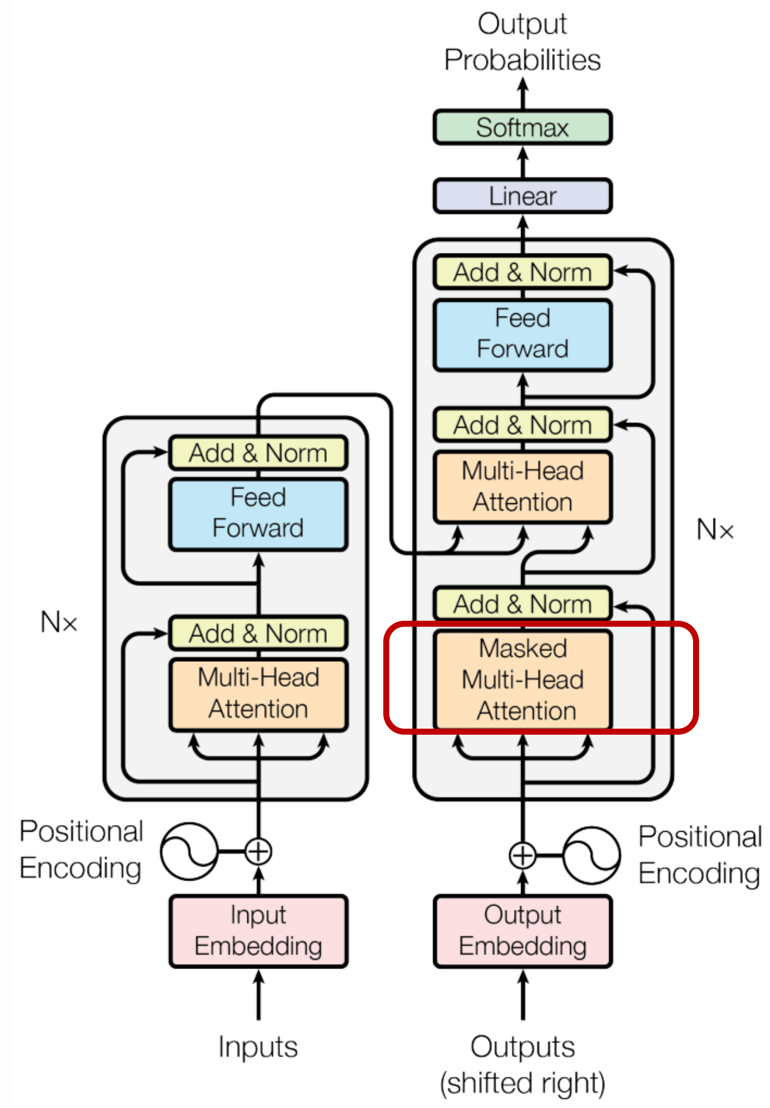
Next token prediction task in decoder part

input of decoder: $[< \mathrm{SOS} >, x_1, x_2, \dots, x_T]$ SOS: start of sentence

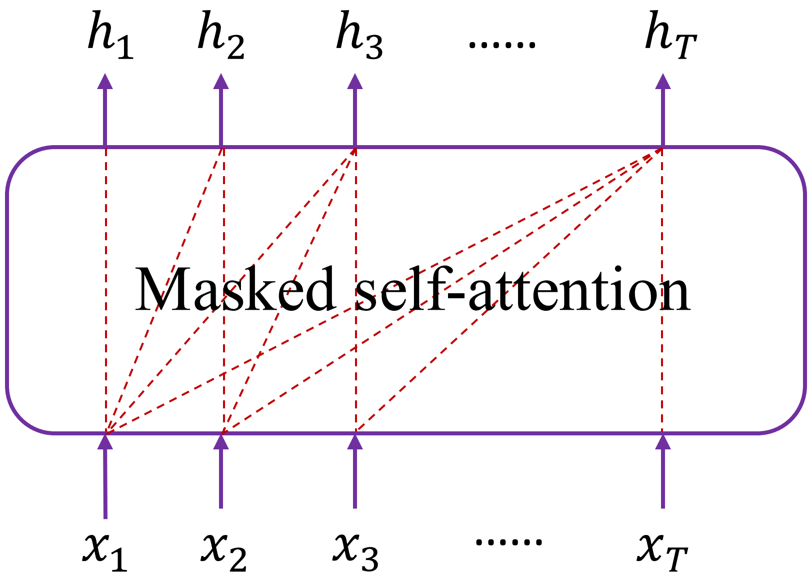
label: $[x_1, x_2, x_3, \dots, x_T, < \mathrm{EOS} >]$ EOS: end of sentence

- Vanilla implementation: Given previous $t-1$ tokens, predict the t-th token
- Efficient implementation: Given all T tokens as inputs, predict all the outputs simultaneously using masked self-attention
  - Benefits: Fully exploits parallel computatioin capabilities

Encoder-decoder Architecture of Transformer

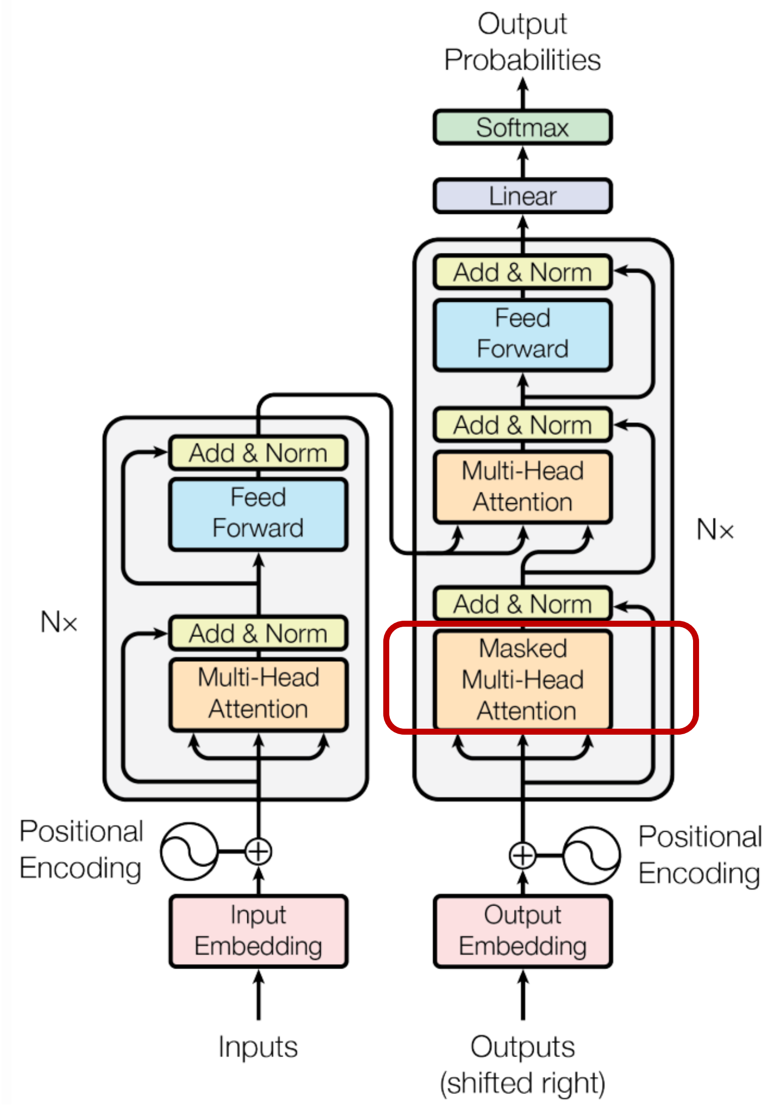


Masked self-attention in decoder



- h_t is only related to $[x_1, x_2, \dots, x_t]$

Encoder-decoder Architecture of Transformer



Efficient implementation of masked self-attention in decoder

$$\text{MaskedAttention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V$$

- M is causal mask with upper triangular part is $-\infty$, lower triangular part is 0, for example:

$$M = \begin{bmatrix} 0 & -\infty & -\infty & -\infty \\ 0 & 0 & -\infty & -\infty \\ 0 & 0 & 0 & -\infty \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Encoder-decoder Architecture of Transformer

$$\text{MaskedAttention}(Q,K,V) = \text{softmax}\left(\underbrace{\boxed{\frac{QK^T}{\sqrt{d_k}}}}_{S} + M\right)V \qquad M = \begin{bmatrix} 0 & -\infty & -\infty & -\infty \\ 0 & 0 & -\infty & -\infty \\ 0 & 0 & 0 & -\infty \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

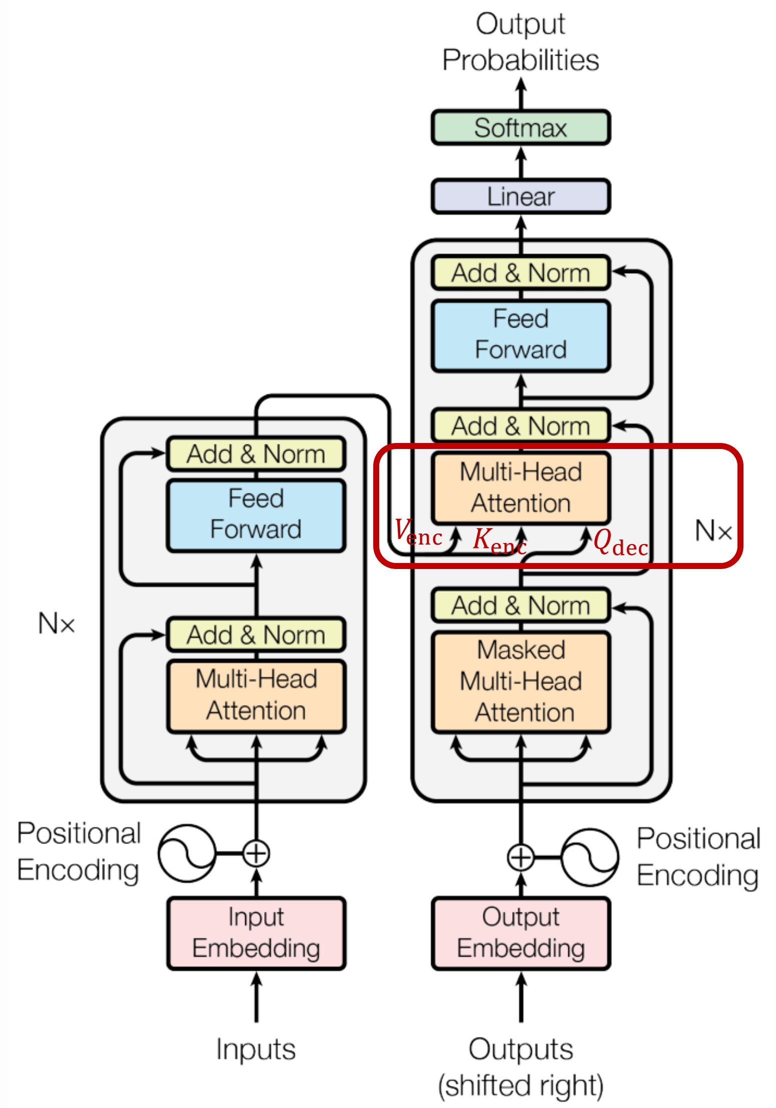
$$\boxed{\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \qquad e^{-\infty} = 0}$$

$$S = \frac{QK^T}{\sqrt{d_k}} = \begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ s_{21} & s_{22} & s_{23} & s_{24} \\ s_{31} & s_{32} & s_{33} & s_{34} \\ s_{41} & s_{42} & s_{43} & s_{44} \end{bmatrix} \qquad S + M = \begin{bmatrix} s_{11} & -\infty & -\infty & -\infty \\ s_{21} & s_{22} & -\infty & -\infty \\ s_{31} & s_{32} & s_{33} & -\infty \\ s_{41} & s_{42} & s_{43} & s_{44} \end{bmatrix} \qquad \text{softmax}(S+M) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ c_{21} & c_{22} & 0 & 0 \\ c_{31} & c_{32} & c_{33} & 0 \\ c_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix}$$

$$\text{softmax}(S+M)V = \begin{bmatrix} 1 & 0 & 0 & 0 \\ c_{21} & c_{22} & 0 & 0 \\ c_{31} & c_{32} & c_{33} & 0 \\ c_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} v_1 \\ c_{21}v_1 + c_{22}v_2 \\ c_{31}v_1 + c_{32}v_2 + c_{33}v_3 \\ c_{41}v_1 + c_{42}v_2 + c_{43}v_3 + c_{44}v_4 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}$$

- v_t is row vector
- h_t is only related to tokens before t

Encoder-decoder Architecture of Transformer



Goal of cross attention: compute the relationship between h_t from decoder and contextualized representation from encoder

$$\text{softmax}(S + M)V = \begin{bmatrix} v_1 \\ c_{21}v_1 + c_{22}v_2 \\ c_{31}v_1 + c_{32}v_2 + c_{33}v_3 \\ c_{41}v_1 + c_{42}v_2 + c_{43}v_3 + c_{44}v_4 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} = H_{\text{dec}}$$

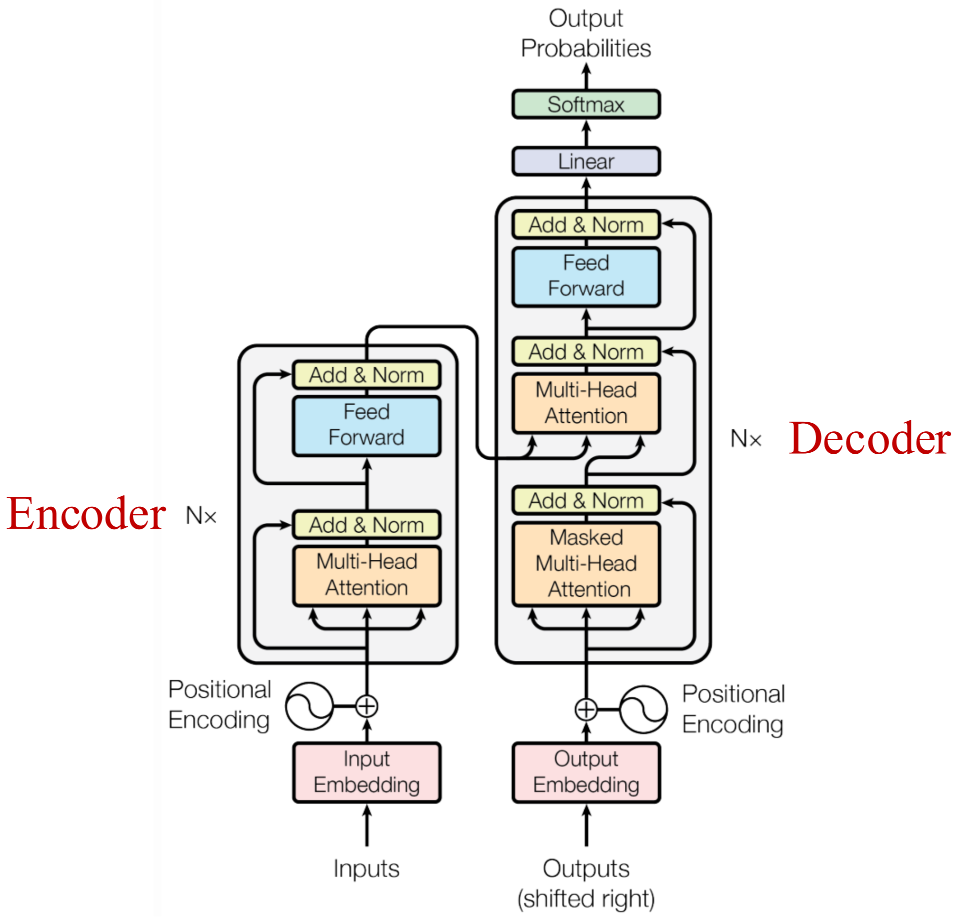
Query (from decoder): $Q_{\text{dec}} = H_{\text{dec}} W_Q$

Key (from encoder): $K_{\text{enc}} = H_{\text{enc}} W_K$

Value (from encoder): $V_{\text{enc}} = H_{\text{enc}} W_V$

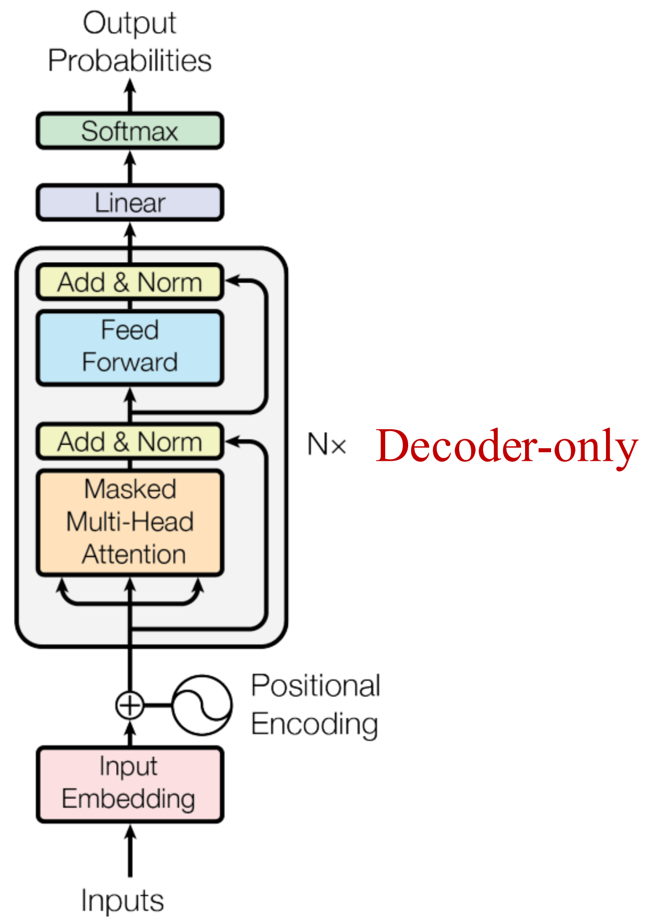
$$\text{CrossAttention}(Q_{\text{dec}}, K_{\text{enc}}, V_{\text{enc}}) = \text{softmax}\left(\frac{Q_{\text{dec}} K_{\text{enc}}^T}{\sqrt{d_k}}\right) V_{\text{enc}}$$

Decoder-only Architecture of Transformer



Encoder-decoder architecture

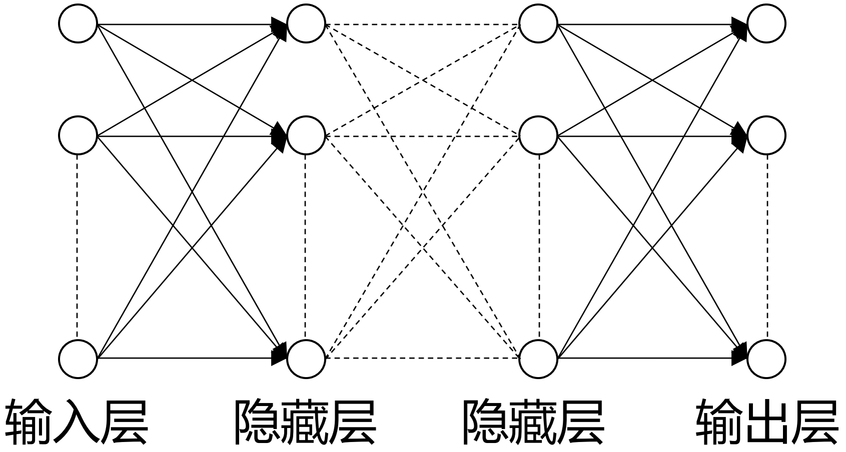
Example: vanilla Transformer

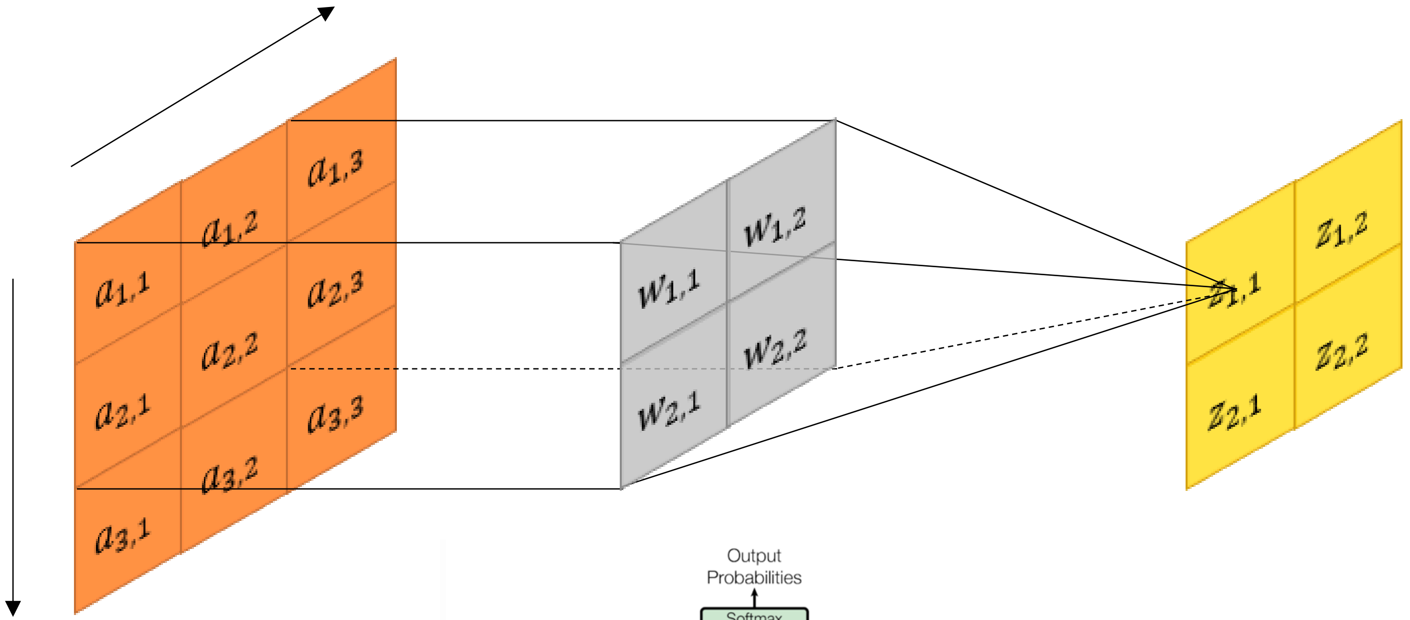


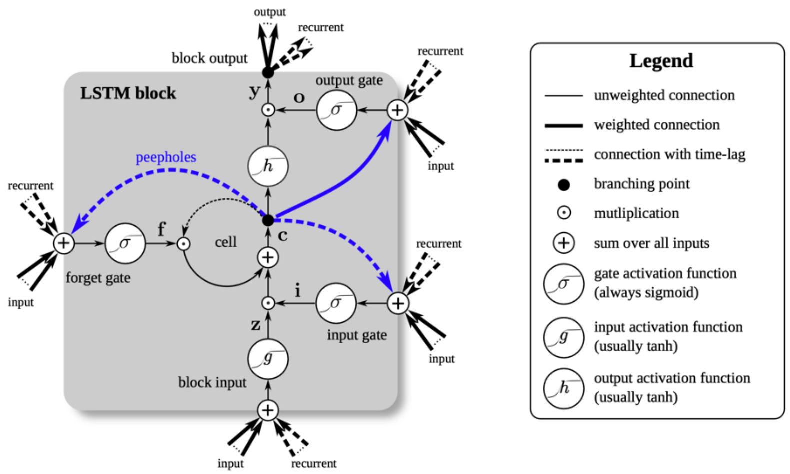
Decoder-only architecture

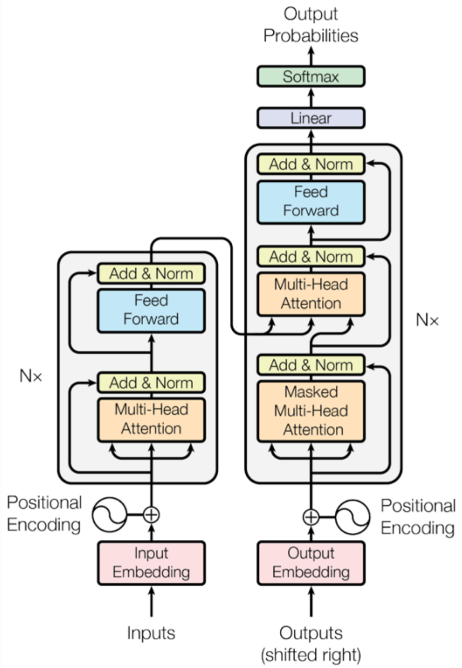
Example: GPT, LLaMA

神经网络回顾


输入层
隐藏层
隐藏层
输出层


$a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,1}$ $a_{3,2}$ $a_{3,3}$
$w_{1,1}$ $w_{1,2}$
$w_{2,1}$ $w_{2,2}$
$z_{1,1}$ $z_{1,2}$
$z_{2,1}$ $z_{2,2}$


output
recurrent
block output
LSTM block
y
o
output gate
peepholes
h
cell
c
f
forget gate
i
input gate
z
g
block input
input
recurrent
Legend
unweighted connection
weighted connection
connection with time-lag
branching point
mutliplication
sum over all inputs
gate activation function (always sigmoid)
input activation function (usually tanh)
output activation function (usually tanh)


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

本章目录

1.凸优化概述

■ 优化问题的一般形式

$$minimize\ f_0(x)$$

待优化的目标函数

$$subject\ to\ f_i(x) \boxed{\le} 0, i = 1,2,\ldots,m$$

$(s.t.)$

m个不等式约束

$$h_i(x) = 0, i = 1,2,\ldots,p$$

p个等式约束

1.凸优化概述

- 凸集（Convex set）：如果连接集合C中任意两点的线段都在C内，则C为凸集，即

  对于 $x_1, x_2 \in C$,且$0 \le \theta \le 1$

  $\theta x_1 + (1-\theta)x_2 \in C$

- 凸函数（Convex function）： $\boldsymbol{dom} f$是凸集，对于所有$x, y \in \boldsymbol{dom} f$,且$0 \le \theta \le 1$，以下不等式均成立

  $f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta)f(y)$

- 凸函数 vs 非凸函数

  简单 复杂

1.凸优化概述

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$$

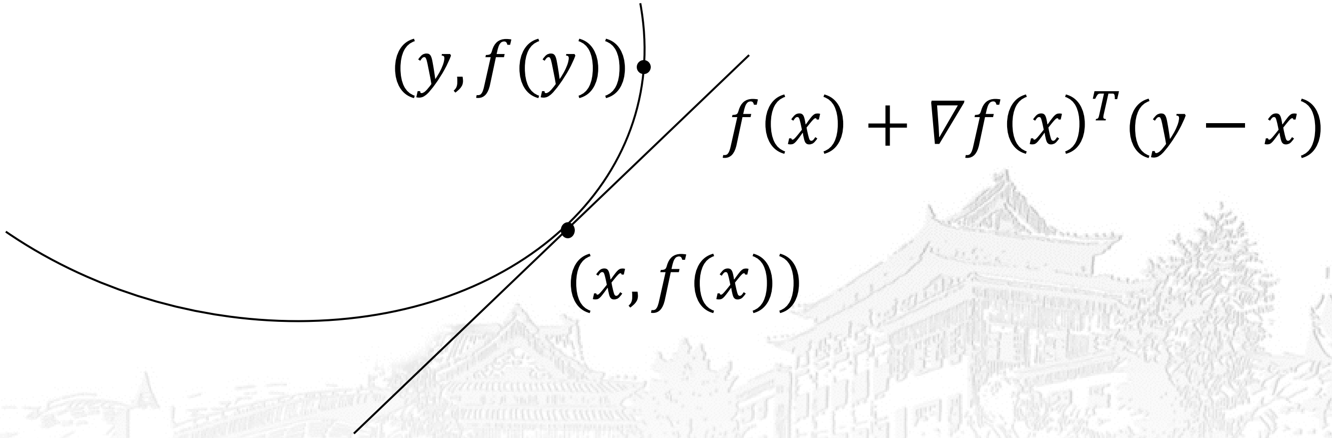
- 连接凸函数上任意两点的线段都在图像上方

1.凸优化概述

- 凸函数的一阶条件

f为可微函数，当且仅当$\boldsymbol{dom}f$是凸集，且以下不等式成立

$$f(y) \geq f(x) + \nabla f(x)^T(y-x)$$



- 凸函数的二阶条件

f二阶可微，当且仅当$\boldsymbol{dom}f$是凸集，且以下不等式成立

$$\nabla^2 f(x) \succcurlyeq 0$$

（Hessian矩阵正定或二阶导数大于等于0）

1.凸优化概述

$$minimize\ f_0(x)$$

$$s.t.\ f_i(x) \le 0, i = 1,2, \dots, m$$

$$h_i(x) = 0, i = 1,2, \dots, p$$

凸优化

- $\boldsymbol{domf}$是凸集， 目标函数$f_0(x)$和不等式约束$f_i(x)$为凸函数，等式约束$h_i(x)$ 为仿射函数，凸优化的目标在于找到全局最优解$x^* \in \boldsymbol{domf}$，使得对任意$x \in \boldsymbol{domf}$，$f(x^*) \le f(x)$均成立

- 可行解（feasible solution）：满足所有约束条件的解
- 最优解（optimal solution）：满足所有约束条件，且对任意$x \in \boldsymbol{domf}$，$f(x^*) \le f(x)$均成立

1.凸优化概述

$$minimize\ f_0(x)$$

$$s.t.\ f_i(x) \le 0, i = 1,2,\dots,m$$

$$h_i(x) = 0, i = 1,2,\dots,p$$

- 拉格朗日函数
  - 为每个约束指定一个拉格朗日乘子，以乘子为加权系数将约束增加到目标函数中

$$L(x,\lambda,v) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} v_i h_i(x)$$

$$L(x,\lambda,v): R^n \times R_+^m \times R^p \to R$$

$$x \in R^n \quad \lambda \in R_+^m, v \in R^p$$

1.凸优化概述

$$\begin{aligned} &minimize\ f_0(x) \\ &s.t.\ f_i(x) \le 0, i = 1,2,\dots,m \\ &\quad\ h_i(x) = 0, i = 1,2,\dots,p \end{aligned}$$

- 拉格朗日函数

$$L(x,\lambda,v) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} v_i h_i(x)$$

- 拉格朗日对偶函数
  - 对拉格朗日函数$L(x,\lambda,v)$中的x取下确界可定义拉格朗日对偶函数

$$g(\lambda,v) = \inf_{x\in R^n} L(x,\lambda,v) = \inf_{x\in R^n}\left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} v_i h_i(x)\right)$$

- 拉格朗日对偶函数 $g(\lambda,v): R_+^m \times R^p \to R$

inf (infimum): 下确界，数学分析中的概念，小于等于集合中的所有成员的最大实数

$$\inf\{x \in R: 0 < x < 1\} = 0$$

sup (supremum) : 上确界，大于等于集合中所有成员的最小实数

$$\sup\{x \in R: 0 < x < 1\} = 1$$

1.凸优化概述

■ 拉格朗日对偶函数

$$g(\lambda, v) = \inf_{x \in R^n} L(x, \lambda, v) = \inf_{x \in R^n} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} v_i h_i(x) \right)$$

- 拉格朗日对偶函数是凹函数，无论原问题是否为凸问题
- 拉格朗日对偶函数给出了原问题最优值的下界：$g(\lambda, v) \le p^*$， p^*：原问题（primal problem）的最优值（optimal value）

1.凸优化概述

- 拉格朗日对偶函数 $g(\lambda, v) = \inf_{x \in R^n} L(x, \lambda, v) = \inf_{x \in R^n} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} v_i h_i(x) \right)$

证明： $g(\lambda, v) \leq p^*$，其中p^*为原问题的最优值

假设$\tilde{x}$是原问题的可行解，即$f_i(\tilde{x}) \leq 0$，且$h_i(\tilde{x}) = 0$，对任意i均成立。由于$\lambda_i \geq 0$，则

$$\sum_{i=1}^{m} \lambda_i f_i(\tilde{x}) + \sum_{i=1}^{p} v_i h_i(\tilde{x}) \leq 0$$ 代入拉格朗日函数定义，可得

$$L(\tilde{x}, \lambda, v) = f_0(\tilde{x}) + \sum_{i=1}^{m} \lambda_i f_i(\tilde{x}) + \sum_{i=1}^{p} v_i h_i(\tilde{x}) \leq f_0(\tilde{x})$$ 且

$$g(\lambda, v) = \inf_{x \in R^n} L(x, \lambda, v) \leq L(\tilde{x}, \lambda, v) \leq f_0(\tilde{x})$$

对于任意可行解$\tilde{x}$， $g(\lambda, v) \leq f_0(\tilde{x})$都成立，因此$g(\lambda, v) \leq p^*$

1.凸优化概述

■ 拉格朗日对偶函数

$$g(\lambda, v) = \inf_{x \in R^n} L(x, \lambda, v) = \inf_{x \in R^n} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} v_i h_i(x) \right)$$

从拉格朗日对偶函数获得的下界中，哪个是最优的?

当$g(\lambda, v) = -\infty$，则其提供的下界无实际意义

■ 拉格朗日对偶问题

$$\max_{\lambda \geq 0, v} g(\lambda, v) = \max_{\lambda \geq 0, v} \inf_{x \in R^n} L(x, \lambda, v)$$

- 假设拉格朗日对偶问题（dual problem）的最优值为d^*

1.凸优化概述

■ 弱对偶

$$d^* \leq p^*$$

- 拉格朗日对偶函数给出了原问题最优值的下界：$g(\lambda, v) \leq p^*$
- 拉格朗日对偶问题： $\max_{\lambda \geq 0, v} g(\lambda, v)$

■ 强对偶

$$d^* = p^*$$

$$minimize\ f_0(x)$$
$$s.t.\ f_i(x) \leq 0, i = 1,2,\ldots,m$$
$$h_i(x) = 0, i = 1,2,\ldots,p$$

- 如果$f_0(x),\ldots,f_m(x)$为凸函数，通常情况下强对偶成立
- 强对偶成立的一般条件：Slater条件

■ 对偶间隙

$$p^* - d^*$$

1.凸优化概述

$$minimize\ f_0(x)$$

$$s.t.\ f_i(x) \le 0, i = 1,2,\dots,m$$

$$h_i(x) = 0, i = 1,2,\dots,p$$

■ 拉格朗日函数

$$L(x,\lambda,v) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} v_i h_i(x)$$

■ KKT条件

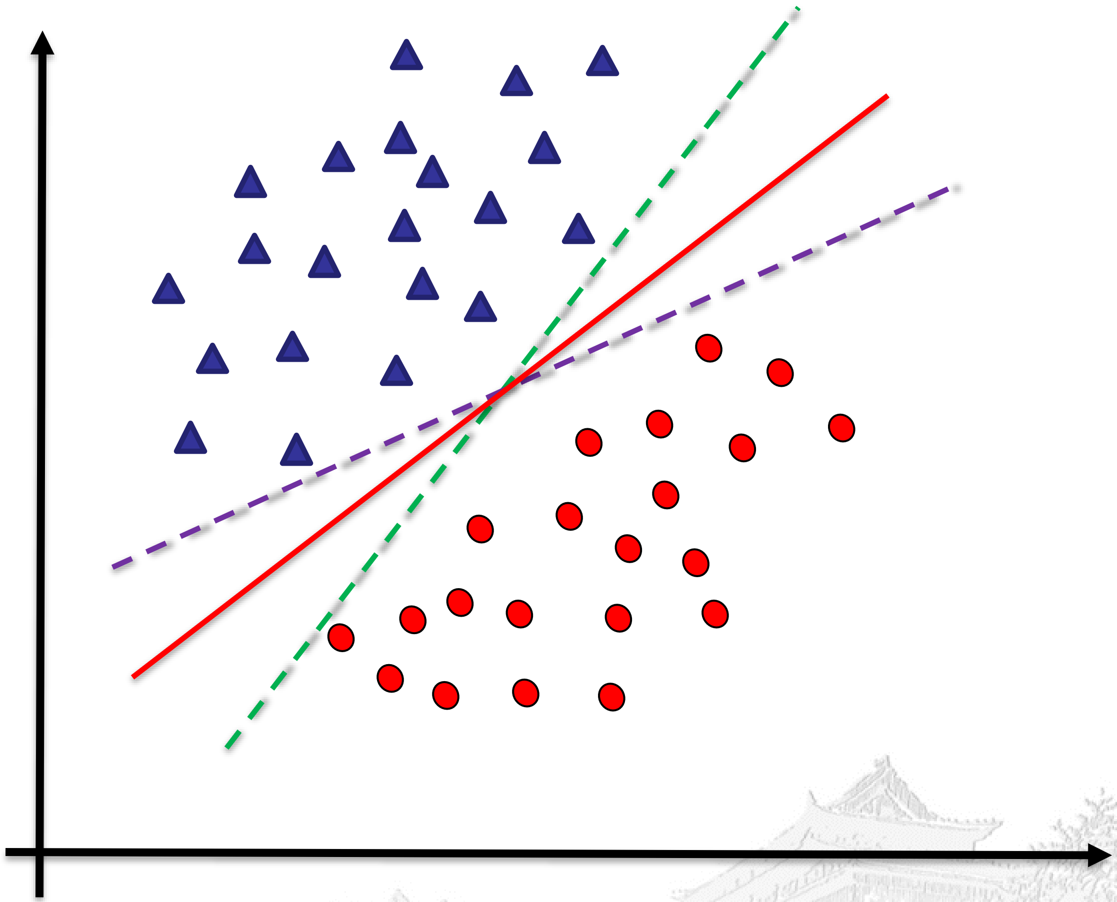
x^*和(λ^*, v^*)分别是原问题和对偶问题的最优解，且对偶间隙为0，则

$$\begin{cases} \nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^{p} v_i^* \nabla h_i(x^*) = 0 & \text{稳定性条件} \\ f_i(x^*) \le 0, i = 1,\dots,m & \text{原始可行性条件} \\ h_i(x^*) = 0, i = 1,\dots,p & \text{原始可行性条件} \\ \lambda_i^* \ge 0, i = 1,\dots,m & \text{对偶可行性条件} \\ \lambda_i^* f_i(x^*) = 0, i = 1,\dots,m & \text{互补松弛条件} \end{cases}$$

- 当原问题为凸问题时，且不等式约束为凸函数，等式约束为仿射变换，则KKT条件为充要条件，且对偶间隙为0 。即满足KKT条件的解为最优解，最优解一定满足KKT条件

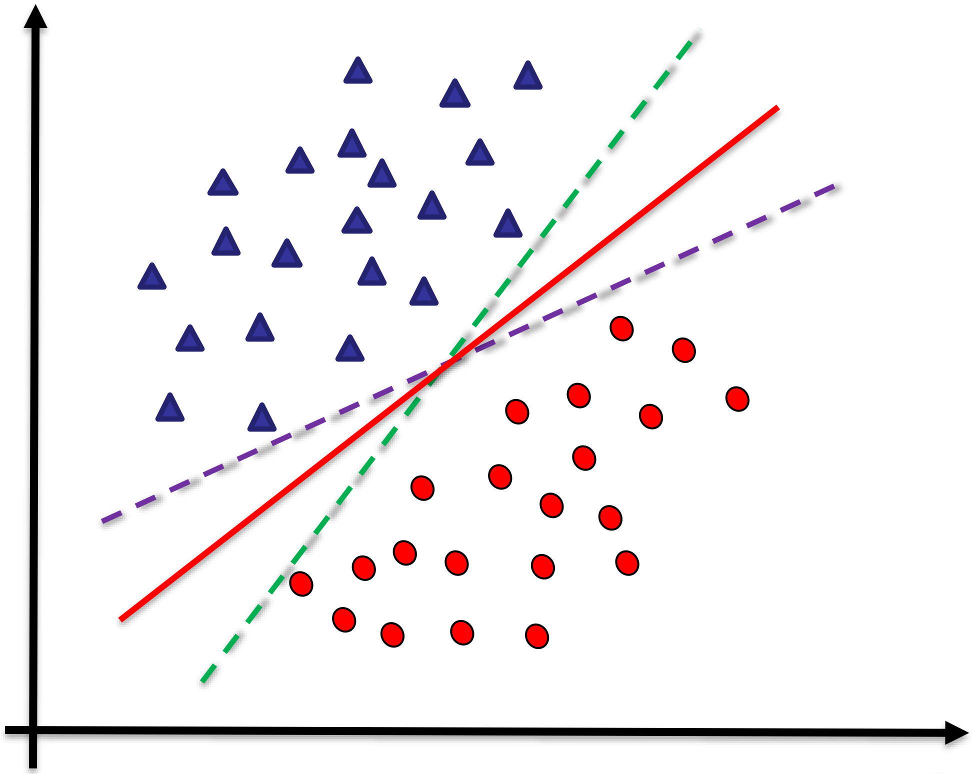
本章目录

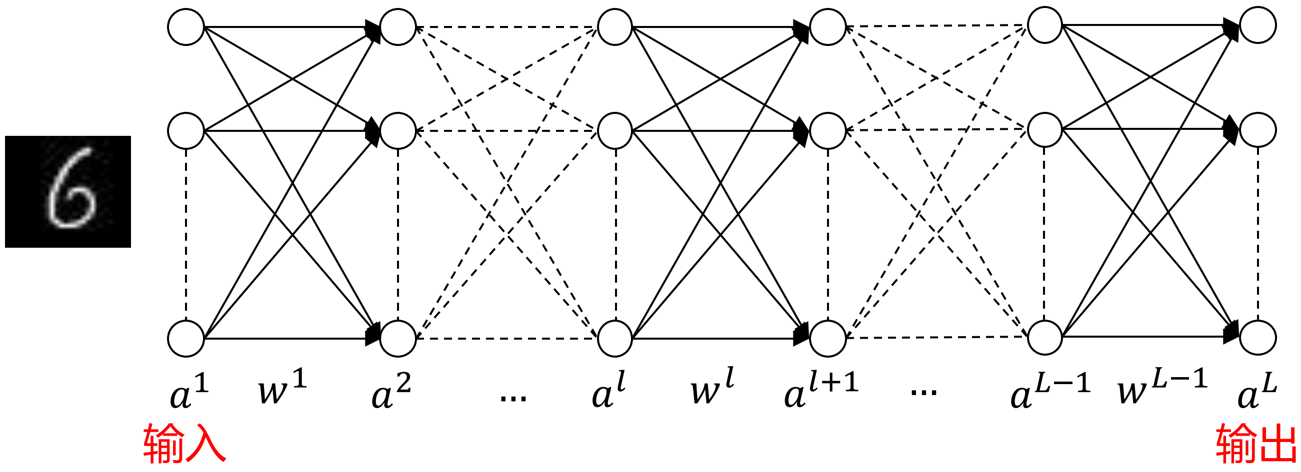
2.支持向量机概述



- 对于左侧二分类问题，选择哪一条决策超平面?

2.支持向量机概述



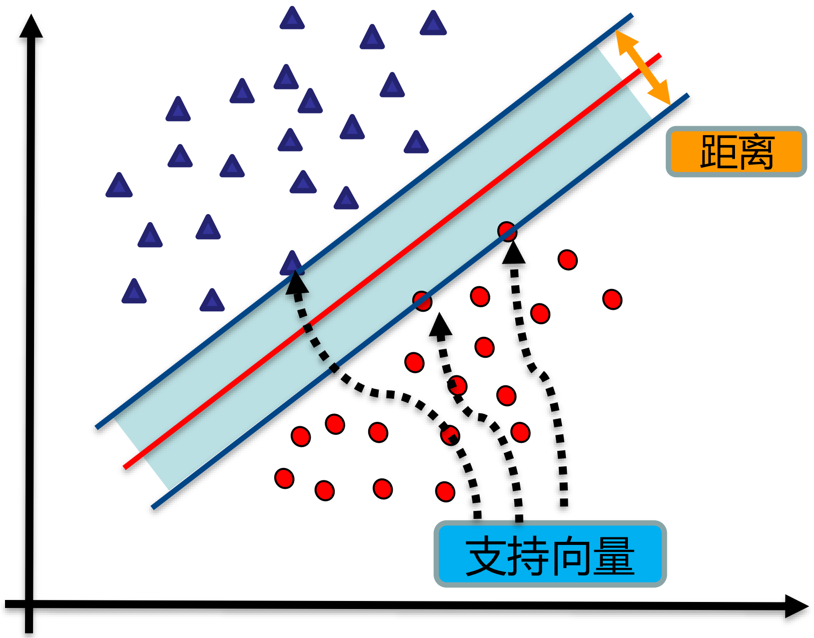


- 红色的决策线对输入扰动更加鲁棒（robust）
- 神经网络也存在鲁棒性问题，典型代表：对抗样本

2.支持向量机概述

支持向量机（Support Vector Machine，SVM）

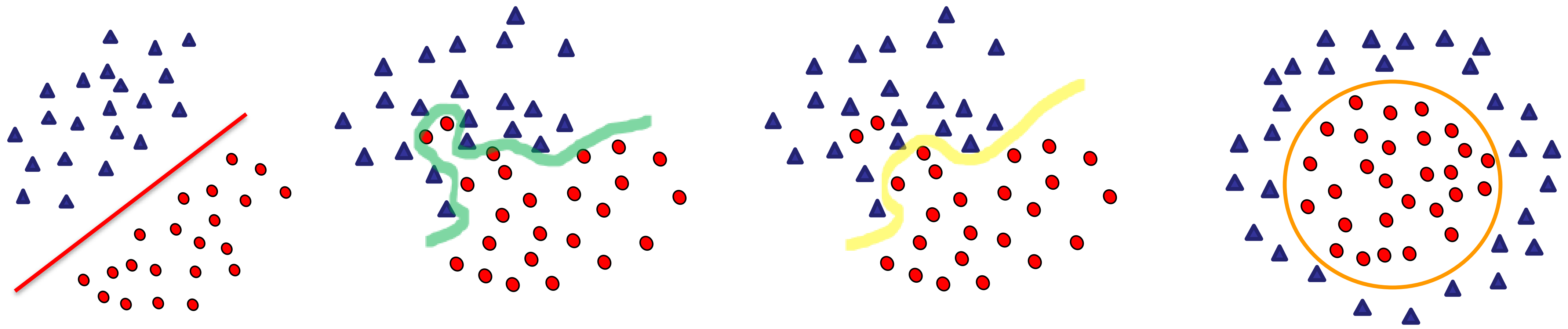
是一类以监督学习方式对数据进行二分类的分类模型



- SVM核心思想是寻找一个分类超平面，使得样本点与超平面的距离最大化
- SVM也被称为最大间隔分类器（Large Margin Classifier）
- 支持向量（Support Vector）：距离超平面最近的点

2.支持向量机概述

硬间隔、软间隔和非线性 SVM



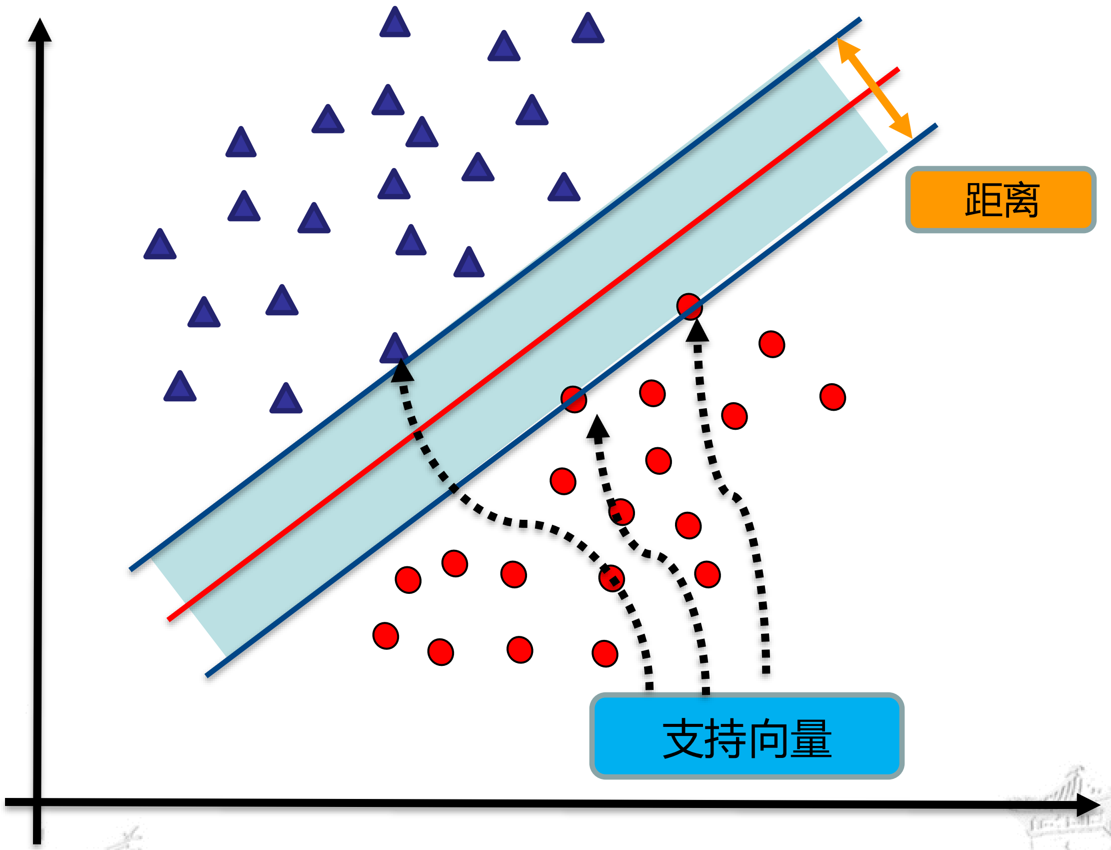
线性可分　　硬间隔　　软间隔　　线性不可分

假如数据是完全的线性可分的，那么学习到的模型可以称为硬间隔支持向量机。简而言之，硬间隔指的就是完全分类准确，不能存在分类错误的情况。软间隔，就是允许一定量的样本分类错误

2.支持向量机概述

算法思想

找到集合中的支持向量，用这些点构建一个超平面（称为决策面），使得支持向量到该超平面的距离最大



2.支持向量机概述

背景知识

任意超平面可以用下面这个线性方程来描述：

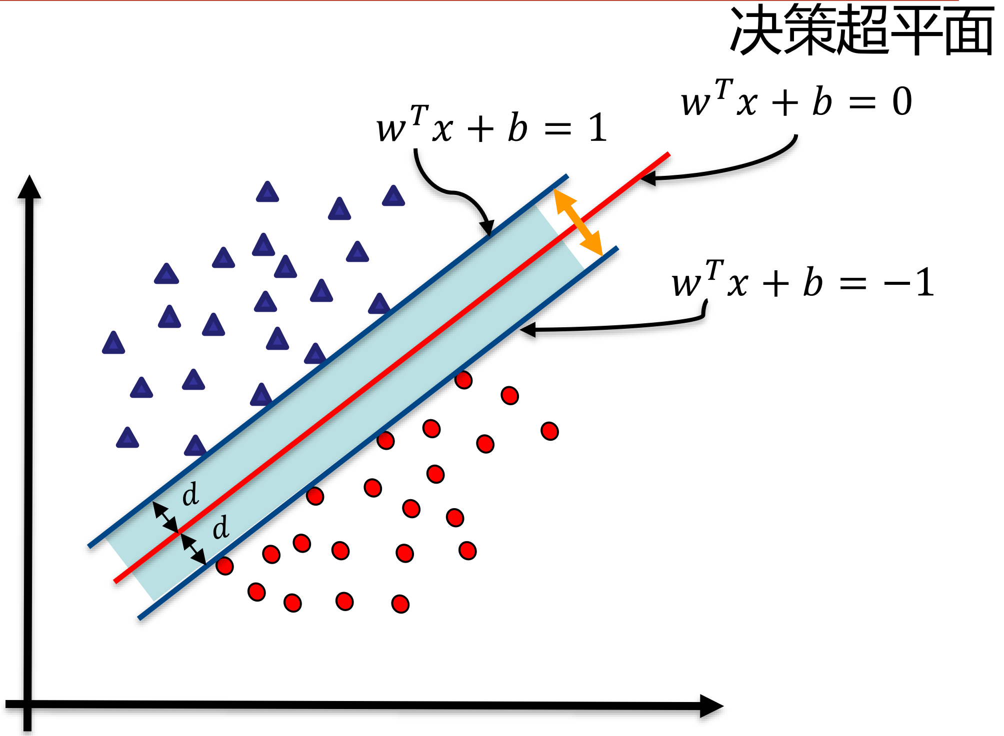
$$w^T x + b = 0$$

二维空间点 (x, y)到直线 $Ax + By + C = 0$的距离公式是：

$$\frac{|Ax + By + C|}{\sqrt{A^2 + B^2}}$$

扩展到 n 维空间后，点 $x = (x^{(1)}, x^{(2)} \dots x^{(\mathrm{n})})$ 到超平面 $w^T x + b = 0$ 的距离为：$\frac{|w^T x + b|}{||w||}$

其中 $||w|| = \sqrt{w_1^2 + \cdots w_n^2}$



如图所示，根据支持向量的定义，假设支持向量到决策超平面的距离为 d，其他点到决策超平面的距离大于 d

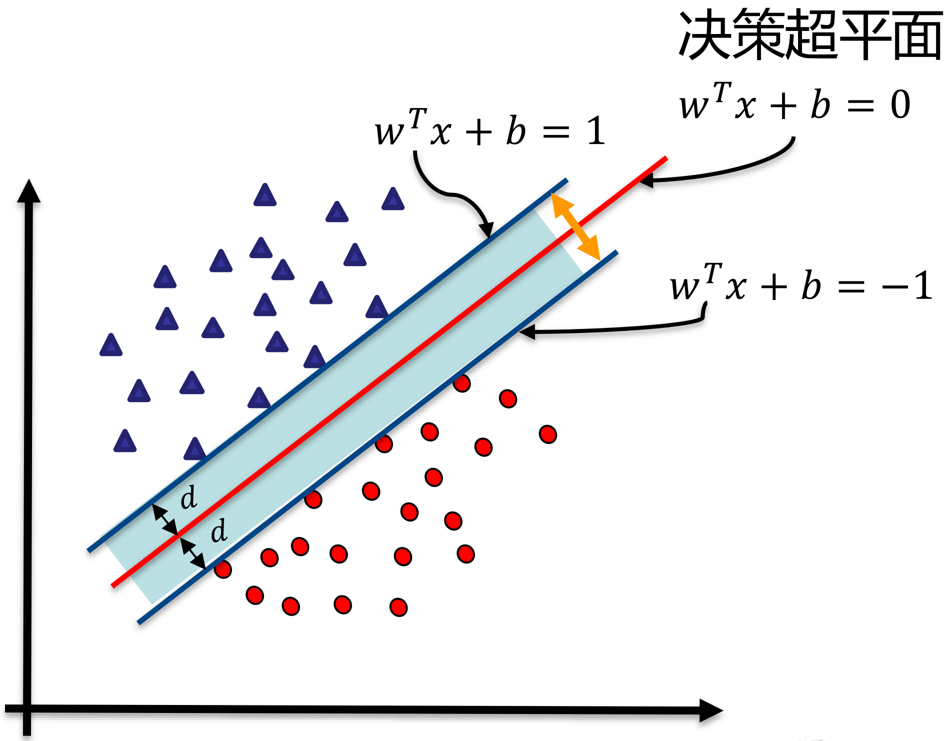
2.支持向量机概述

- 给定训练数据集$D=\{(x_1,y_1),(x_2,y_2),\dots,(x_N,y_N)\}$，其中

$x_i\in R^n, y_i\in\{+1,-1\}, i=1,2,\dots,N$

假设超平面(w,b)将线性可分数据集正确分类，即对于$(x_i,y_i)\in D$，若$y_i=+1$，则$w^Tx_i+b>0$；若$y_i=-1$，则$w^Tx_i+b<0$，令

$$\begin{cases} w^Tx_i+b\geq +1, y_i=+1 \\ w^Tx_i+b\leq -1, y_i=-1 \end{cases}$$



- 将以上两个方程合并，可得： $y_i(w^Tx_i+b)\geq 1$

- $y_i(w^Tx_i+b)\geq 1$为约束条件，即要求决策超平面(w,b)将所有样本分类正确

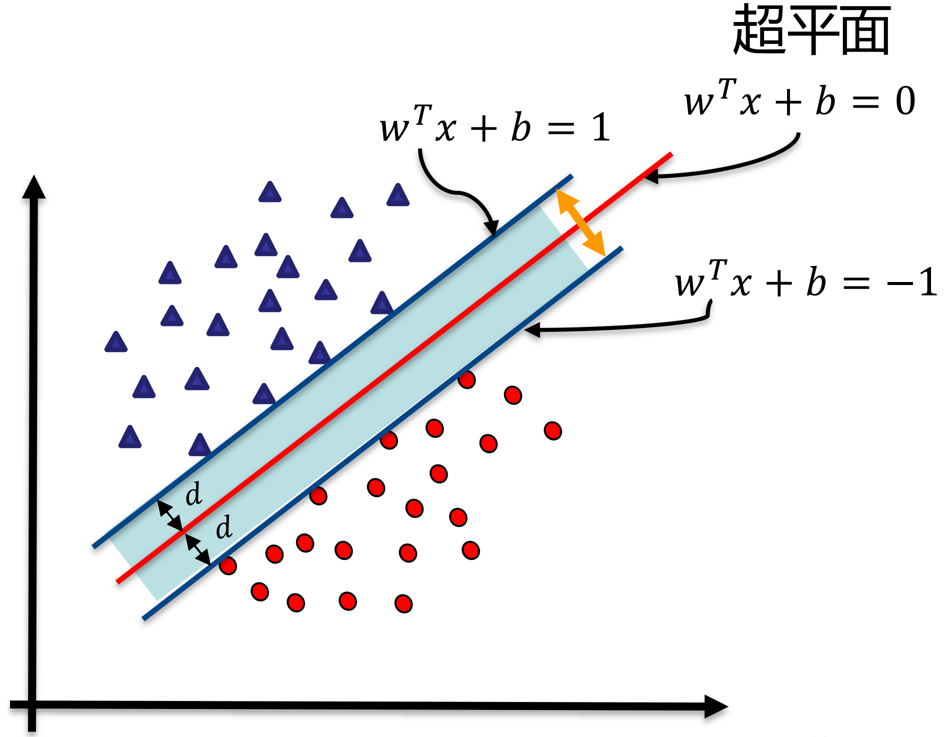
2.支持向量机概述

■ 支持向量到超平面的距离可以写为：$d=\frac{|w^Tx+b|}{||w||}=\frac{1}{||w||}$

■ 两类（正类和负类）支持向量到超平面的距离之和为$\gamma=\frac{2}{||w||}$，也被称为“间隔”，即优化目标

$$\max_{w,b}\frac{2}{||w||}$$

$$s.t.\,y_i(w^Tx_i+b)\geq 1, i=1,2,\dots,N$$



2.支持向量机概述

$$\max_{w,b} \frac{2}{||w||}$$

$$s.t.\ y_i(w^T x_i + b) \geq 1, i = 1,2, \dots, N$$

$$\min_{w,b} \frac{1}{2} ||w||^{\boxed{2}}$$ 方便后续求导

$$s.t.\ y_i(w^T x_i + b) \geq 1, i = 1,2, \dots, N$$

- 即支持向量机（SVM）的基本形式

本章目录

3.支持向量机求解

$$\min_{w,b} \frac{1}{2} \|w\|^2$$

$$s.t.\ y_i(w^T x_i + b) \geq 1, i = 1,2, \dots, N$$

请写出以上优化问题对应的拉格朗日函数

$$minimize\ f_0(x)$$

$$s.t.\ f_i(x) \leq 0, i = 1,2, \dots, m$$

$$h_i(x) = 0, i = 1,2, \dots, p$$

- 拉格朗日函数

$$L(x, \lambda, v) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} v_i h_i(x)$$

$$L(x, \lambda, v): R^n \times R_+^m \times R^p \to R$$

$$x \in R^n \quad \lambda \in R_+^m, v \in R^p$$

3.支持向量机求解

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

$$s.t.\, y_i(w^T x_i + b) \geq 1, i = 1,2,\dots,N$$

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{N} \alpha_i\left(1 - y_i(w^T x_i + b)\right)$$

$$= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i y_i(w^T x_i + b) + \sum_{i=1}^{N} \alpha_i$$

- α_i为拉格朗日乘子，与教材记号保持一致

$$minimize\ f_0(x)$$

$$s.t.\ f_i(x) \leq 0, i = 1,2,\dots,m$$

$$h_i(x) = 0, i = 1,2,\dots,p$$

- 拉格朗日函数

$$L(x,\lambda,v) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} v_i h_i(x)$$

$$L(x,\lambda,v): R^n \times R_+^m \times R^p \to R$$

$$x \in R^n \qquad \lambda \in R_+^m, v \in R^p$$

3.支持向量机求解

原优化问题：
$$\min_{w,b} \frac{1}{2}\|w\|^2$$
$$s.t.\ y_i(w^T x_i + b) \geq 1, i = 1,2,\ldots,N$$

拉格朗日函数：$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i y_i (w^T x_i + b) + \sum_{i=1}^{N} \alpha_i$

- 线性可分SVM满足强对偶条件，即$d^* = p^*$，原问题最优值等于对偶问题最优值
- 通过解对偶问题，得到最优解$\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_N^*)$，得到原问题的最优解$(w^*, b^*)$

3.支持向量机求解

构造拉格朗日对偶函数：

$$\min_{w,b} L(w,b,\alpha) \qquad L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{N}\alpha_i y_i(w^T x_i+b)+\sum_{i=1}^{N}\alpha_i$$

$$\frac{\partial}{\partial w}L(w,b,\alpha)=?$$

$$\frac{\partial}{\partial b}L(w,b,\alpha)=?$$

提示：$\|w\|^2=w^T w$

3.支持向量机求解

求 $\min_{w,b} L(w,b,\alpha)$ $\qquad L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{N}\alpha_i y_i(w^T x_i+b)+\sum_{i=1}^{N}\alpha_i$

$$\frac{\partial}{\partial w}L(w,b,\alpha)=w-\sum_{i=1}^{N}\alpha_i\, y_i x_i=0 \quad\Rightarrow\quad w=\sum_{i=1}^{N}\alpha_i\, y_i x_i$$

$$\frac{\partial}{\partial b}L(w,b,\alpha)=-\sum_{i=1}^{N}\alpha_i\, y_i=0 \quad\Rightarrow\quad \sum_{i=1}^{N}\alpha_i\, y_i=0$$

3.支持向量机求解

$$\boxed{w=\sum_{i=1}^{N}\alpha_i\, y_i x_i \qquad \sum_{i=1}^{N}\alpha_i\, y_i=0}$$

代入 $L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{N}\alpha_i y_i(w^T x_i+b)+\sum_{i=1}^{N}\alpha_i$

$$\min_{w,b} L(w,b,\alpha)=\frac{1}{2}w^T w-\sum_{i=1}^{N}\alpha_i y_i(w^T x_i+b)+\sum_{i=1}^{N}\alpha_i$$

$$=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(x_i\cdot x_j\right)-\sum_{i=1}^{N}\alpha_i y_i\left(\left(\sum_{j=1}^{N}\alpha_j\, y_j x_j\right)\cdot x_i+b\right)+\sum_{i=1}^{N}\alpha_i$$

$$=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(x_i\cdot x_j\right)+\sum_{i=1}^{N}\alpha_i$$

3.支持向量机求解

构建拉格朗日对偶问题

求 $\min\limits_{w,b} L(w,b,\alpha)$ 对α的极大

$$\max_{\alpha}\min_{w,b} L(w,b,\alpha)=\max_{\alpha}\ -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(x_i\cdot x_j\right)+\sum_{i=1}^{N}\alpha_i$$

$$s.t.\sum_{i=1}^{N}\alpha_i\,y_i=0$$

$$\alpha_i\geq 0, i=1,2,\dots,N$$

最小化

$$\min_{\alpha}\ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(x_i\cdot x_j\right)-\sum_{i=1}^{N}\alpha_i$$

3.支持向量机求解

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i$$

$$s.t. \sum_{i=1}^{N}\alpha_i y_i = 0 \qquad \alpha_i \geq 0, i = 1,2,\dots,N$$

设$\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)$为拉格朗日对偶问题最优解，则原始问题最优解$w^*$和$b^*$如下

$$w^* = \sum_{i=1}^{N}\alpha_i^* y_i x_i$$

$$\boxed{b^* = ?}$$

■ KKT条件

x^*和(λ^*, v^*)分别是原问题和对偶问题的最优解，且强对偶成立，则

$$\begin{cases} \nabla f_0(x^*) + \sum_{i=1}^{m}\lambda_i^*\nabla f_i(x^*) + \sum_{i=1}^{p} v_i^*\nabla h_i(x^*) = 0 \\ f_i(x^*) \leq 0, i = 1,\dots,m \\ h_i(x^*) = 0, i = 1,\dots,p \\ \lambda_i^* \geq 0, i = 1,\dots,m \\ \boxed{\lambda_i^* f_i(x^*) = 0, i = 1,\dots,m} \end{cases}$$

$$\alpha_i^*(y_i(w^T x_i^* + b^*) - 1) = 0$$

3.支持向量机求解

- KKT条件

x^*和(λ^*, v^*)分别是原问题和对偶问题的最优解，且强对偶成立，则

$$\begin{cases} \nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^{p} v_i^* \nabla h_i(x^*) = 0 \\ f_i(x^*) \le 0, i = 1, \dots, m \\ h_i(x^*) = 0, i = 1, \dots, p \\ \lambda_i^* \ge 0, i = 1, \dots, m \\ \boxed{\lambda_i^* f_i(x^*) = 0, i = 1, \dots, m} \end{cases}$$

$$\alpha_i^*(y_i(w^T x_i^* + b^*) - 1) = 0$$

至少存在一个$\alpha_j^* > 0$，（反证法：若α^* 均为0，则$w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i = 0$，然而$w^* = 0$ 不是原始优化问题的解），因此

$$y_j\left(w^{*T} x_j^* + b^*\right) - 1 = 0$$

将$w^* = \sum_{i=1}^{N} \alpha_i y_i x_i$代入上式，并利用$y_j^2 = 1$可得

$$b^* = y_j - \sum_{i=1}^{N} \alpha_i^* y_i (x_i \cdot x_j)$$

3.支持向量机求解

线性可分支持向量机学习算法

第1步：根据原始优化问题，写出拉格朗日函数

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{N}\alpha_i y_i(w^T x_i+b)+\sum_{i=1}^{N}\alpha_i$$

第2步：求$\min\limits_{w,b} L(w,b,\alpha)$，并代入$L(w,b,\alpha)$

$$\min_{w,b} L(w,b,\alpha)=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(x_i\cdot x_j\right)+\sum_{i=1}^{N}\alpha_i$$

第3步：求解拉格朗日对偶问题，即

$$\max_{\alpha}\min_{w,b} L(w,b,\alpha)=\max_{\alpha}\ -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(x_i\cdot x_j\right)+\sum_{i=1}^{N}\alpha_i$$

$$s.t.\sum_{i=1}^{N}\alpha_i\, y_i=0$$

$$\alpha_i\geq 0, i=1,2,\dots,N$$

求得最优解$\alpha^*=(\alpha_1^*,\alpha_2^*,\dots,\alpha_N^*)$

第4步：根据KKT条件可得原优化问题最优解w^*和b^*

$$w^*=\sum_{i=1}^{N}\alpha_i\, y_i x_i \qquad b^*=y_j-\sum_{i=1}^{N}\alpha_i^* y_i(x_i\cdot x_j)$$

第5步：构建决策超平面以及分类决策函数

决策超平面：$w^{*T}x+b^*=0$

决策函数：$f(x)=sign(w^{*T}x+b^*)$

3.支持向量机求解

$$w^* = \sum_{i=1}^{N} \alpha_i \, y_i x_i \qquad b^* = y_j - \sum_{i=1}^{N} \alpha_i^* y_i (x_i \cdot x_j)$$

决策超平面：$w^{*T}x + b^* = 0$

决策函数：$f(x) = sign(w^{*T}x + b^*)$

- 如何高效地求α_i^*，SMO（Sequential Minimal Optimization, 序列最小优化算法）

谢 谢!