机器学习导论
习题四

151242041, 王昊庭, hatsuyukiw@gmail.com

2017 年 5 月 17 日

1 [20pts] Reading Materials on CNN

卷积神经网络 (Convolution Neural Network, 简称 CNN) 是一类具有特殊结构的神经网络, 在深度学习的发展中具有里程碑式的意义。其中, Hinton 于 2012 年提出的AlexNet可以说是深度神经网络在计算机视觉问题上一次重大的突破。

关于 AlexNet 的具体技术细节总结在经典文章"ImageNet Classification with Deep Convolutional Neural Networks", by Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton in NIPS'12, 目前已逾万次引用。在这篇文章中, 它提出使用 ReLU 作为激活函数, 并创新性地使用 GPU 对运算进行加速。请仔细阅读该论文, 并回答下列问题 (请用 1-2 句话简要回答每个小问题, 中英文均可)。

(a) **[5pts]** Describe your understanding of how ReLU helps its success? And, how do the GPUs help out?

(b) **[5pts]** Using the average of predictions from several networks help reduce the error rates. Why?

(c) **[5pts]** Where is the dropout technique applied? How does it help? And what is the cost of using dropout?

(d) **[5pts]** How many parameters are there in AlexNet? Why the dataset size(1.2 million) is important for the success of AlexNet?

关于 CNN, 推荐阅读一份非常优秀的学习材料, 由南京大学计算机系吴建鑫教授[1]所编写的讲义 Introduction to Convolutional Neural Networks[2], 本题目为此讲义的 Exercise-5, 已获得吴建鑫老师授权使用。

Solution. (a) First, ReLU prevents saturation, as is common in other activations. Second, as the derivative of ReLU is either 0 or 1, it does not lead to gradient vanishing or exploding. GPU is superior in parallelization. As the authors point out, "current

[1] 吴建鑫教授主页链接为cs.nju.edu.cn/wujx

[2] 由此链接可访问讲义https://cs.nju.edu.cn/wujx/paper/CNN.pdf

GPUs are particularly well-suited to cross-GPU parallelization, as they are able to read from and write to one another's memory directly, without going through host machine memory."

(b) Ensemble methods (1) average out biases, and (2) reduce variance.

(c) The dropout layers are placed behind the first two FC layers. They reduce the chance of overfitting. The number of iterations needed to train the NN is doubled.

(d) There are 60 million parameters in AlexNet. The big dataset is essential to prevent overfitting.

2 [20pts] Kernel Functions

(1) 试通过定义证明以下函数都是一个合法的核函数：

(i) **[5pts]** 多项式核: $\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j)^d$;

(ii) **[10pts]** 高斯核：$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$, 其中 $\sigma > 0$.

(2) **[5pts]** 试证明 $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{1+e^{-\mathbf{x}_i^T \mathbf{x}_j}}$ 不是合法的核函数。

Proof. (1) (i) Polynomial kernel: $\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j)^d$.

First, κ is a symmetric function.

Define $\mathbf{K}$ as $k_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{K}_1$ as $k_{1ij} = \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$. $\mathbf{K}_1$ is the kernel matrix of the linear kernel, so it is positive semi-definite.

Since $\mathbf{K} = \mathbf{K}_1^d$, where the product of matrices are defined as the Hadamard product and the Hadamard product of two positive semi-definitive matrices is positive semi-definitive, $\mathbf{K}$ is positive semi-definitive.

Thus κ is a kernel function.

(ii) Gaussian kernel: $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$, where $\sigma > 0$.

First, κ is clearly a symmetric function.

For every $n \in \mathbb{N}_+, \mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^n, a_1, \ldots, a_n \in \mathbb{R}$,

$$\begin{aligned}
\sum_{i,j=1}^{n} a_i a_j \kappa(\mathbf{x}_i, \mathbf{x}_j) &= \sum_{i,j=1}^{n} a_i a_j \exp(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j)}{2\sigma^2}) \\
&= \sum_{i,j=1}^{n} a_i a_j \sum_{k=0}^{\infty} \frac{(\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j)^k}{\sigma^k k!} \exp(-\frac{\|\mathbf{x}_i\|^2}{2\sigma^2}) \exp(-\frac{\|\mathbf{x}_j\|^2}{2\sigma^2}) \\
&= \sum_{k=0}^{\infty} \frac{1}{\sigma^k k!} \sum_{i,j=1}^{n} a_i \exp(-\frac{\|\mathbf{x}_i\|^2}{2\sigma^2}) a_j \exp(-\frac{\|\mathbf{x}_j\|^2}{2\sigma^2}) (\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j)^k \,.
\end{aligned}$$

The term above for each k is identical to the Mercer form of the polynomial kernel of degree k, so each term is non-negative.

Thus κ is a kernel function.

(2) Let $a_1 = a_2 = -1$, $\mathbf{x}_1 = (2), \mathbf{x}_2 = (1)$.

$$\begin{aligned}\sum_{i,j=1}^{n} a_i a_j \kappa(\mathbf{x}_i, \mathbf{x}_j) &= \begin{pmatrix} -1 & -1 \end{pmatrix} \begin{pmatrix} \frac{1}{1+e^{-4}} & \frac{1}{1+e^{-2}} \\ \frac{1}{1+e^{-2}} & \frac{1}{1+e^{-1}} \end{pmatrix} \begin{pmatrix} -1 \\ -1 \end{pmatrix} \\ &= -0.0579 \\ &< 0\,. \end{aligned}$$

Thus κ is not a kernel function.

□

3 [25pts] SVM with Weighted Penalty

考虑标准的 SVM 优化问题如下 (即课本公式 (6.35)),

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, i = 1, 2, \cdots, m. \end{aligned} \tag{3.1}$$

注意到，在(**??**)中，对于正例和负例，其在目标函数中分类错误的“惩罚”是相同的。在实际场景中，很多时候正例和负例错分的“惩罚”代价是不同的，比如考虑癌症诊断，将一个确实患有癌症的人误分类为健康人，以及将健康人误分类为患有癌症，产生的错误影响以及代价不应该认为是等同的。

现在，我们希望对负例分类错误的样本 (即 false positive) 施加 $k > 0$ 倍于正例中被分错的样本的“惩罚”。对于此类场景下,

(1) **[10pts]** 请给出相应的 SVM 优化问题;

(2) **[15pts]** 请给出相应的对偶问题，要求详细的推导步骤，尤其是如 KKT 条件等。

Solution. (1)

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1 - k_i\xi_i \\ & k_i\xi_i \geq 0, \quad i = 1, 2, \cdots, m \\ & k_i = \frac{1}{k} \quad \text{when } y_i = -1 \\ & k_i = 1 \quad \text{when } y_i = 1\,. \end{aligned} \tag{3.2}$$

(2) The Lagrangian function is

$$\begin{aligned} L(\mathbf{w}, b, \alpha, \xi, \mu) = & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \\ & + \sum_{i=1}^{m}\alpha_i(1 - k_i\xi_i - y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b)) - \sum_{i=1}^{m}\mu_i k_i\xi_i\,, \end{aligned}$$

where $\alpha_i \geq 0, \mu_i \geq 0$.

Let the partial derivatives equal to 0, and we come to the following,

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x_i}\,,$$
$$0 = \sum_{i=1}^{m} \alpha_i y_i\,,$$
$$C = k_i \alpha_i + k_i \mu_i\,.$$

Substitute the above equations into the Lagrangian function, and we get the dual problem

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j \\ \text{s.t.} \quad & \sum_{i=1}^{m} \alpha_i y_i = 0\,, \\ & 0 \leq k_i \alpha_i \leq C, i = 1, 2, \ldots, m\,. \end{aligned} \tag{3.3}$$

The KKT conditions are,

$$\begin{aligned} \alpha_i &\geq 0\,, \\ \mu_i &\geq 0\,, \\ -1 + k_i \xi_i + y_i(\mathbf{w}^{\mathrm{T}} \mathbf{x}_i + b) &\geq 0\,, \\ \alpha_i(-1 + k_i \xi_i + y_i(\mathbf{w}^{\mathrm{T}} \mathbf{x}_i + b)) &= 0\,, \\ \xi_i &\geq 0\,, \\ k_i \mu_i \xi_i &= 0\,. \end{aligned} \tag{3.4}$$

4 [35pts] SVM in Practice - LIBSVM

支持向量机 (Support Vector Machine，简称 SVM) 是在工程和科研都非常常用的分类学习算法。有非常成熟的软件包实现了不同形式 SVM 的高效求解，这里比较著名且常用的如 LIBSVM[3]。

(1) **[20pts]** 调用库进行 SVM 的训练，但是用你自己编写的预测函数作出预测。

(2) **[10pts]** 借助我们提供的可视化代码，简要了解绘图工具的使用，通过可视化增进对 SVM 各项参数的理解。详细编程题指南请参见链接：http://lamda.nju.edu.cn/ml2017/PS4/ML4_programming.html.

(3) **[5pts]** 在完成上述实践任务之后，你对 SVM 及核函数技巧有什么新的认识吗？请简要谈谈。

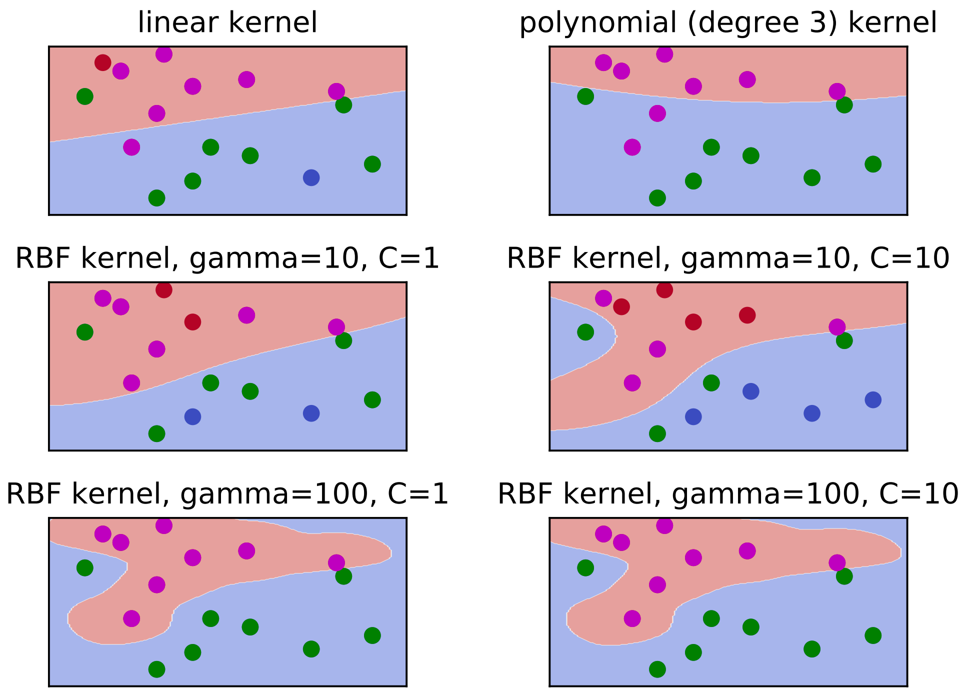
[3]LIBSVM 主页课参见链接：https://www.csie.ntu.edu.tw/~cjlin/libsvm/

Solution.

1. 对于 RBF kernel SVM, 超参数 γ 和 C 代表模型内秉的归纳假设. γ, C 越大, 单个训练样例的影响力越大, 决策边界越复杂; 反之, 则决策边界 (模型) 越简单.

2. Support vector 的数量远高于这个作者预想. 软 SVM 事实上是很软的.

3. The visualization is on the next page. Positive support vectors are colored green, negative support vectors magenta, positive non-support vectors red, and negative non-support vectors blue.


linear kernel
polynomial (degree 3) kernel
RBF kernel, gamma=10, C=1
RBF kernel, gamma=10, C=10
RBF kernel, gamma=100, C=1
RBF kernel, gamma=100, C=10