机器学习笔记

Leoeon

2018 年 9 月 10 日

Contents

I ANN 8

1 CNN（卷积神经网络）[1] 9

1.1 卷积层 9

1.1.1 卷积分组 9

1.1.1.0.1 NIN[2] 9

1.1.1.0.2 Inception[3] [4] [5] [6] 9

1.1.1.0.3 MobileNet[7] 10

1.1.2 实现 10

1.1.2.0.4 Caffe 10

1.2 池化层 11

1.2.1 [8] 11

1.2.2 SPP（空间金字塔池化）[9] 11

1.3 全连接层 11

2 RNN 12

2.1 RNN 12

2.2 LSTM 12

2.2.1 LSTM 12

2.2.2 peephole connection 13

2.2.3 coupled记忆门与输入门 13

2.2.4 GRU(Gated Recurrent Unit) 13

2.3 Higher Order RNN[10] 13

3 AE（自编码） 15

3.1 AE 15

3.1.1 AE 15

3.1.2 稀疏性限制 16

3.1.3 Denoising Auto-Encoder[11] 16

3.2 多层AE[12] 16

4 Hopfield 17

4.1 Hopfield 17

4.1.1 DHNN（离散时间） 18

4.1.2 CHNN（连续时间） 18
4.2 Ising模型 18
4.3 RBM（受限波尔兹曼机） 19
4.3.1 对比散度训练法 19
4.3.2 二项分布 20
4.3.3 多项分布 20
4.4 DBN（深度信念网络） 20

5 other 21
5.1 Highway Network 21
5.1.1 Highway Network[13] [14] 21
5.1.2 Deep Residual Network[15] 21
5.1.3 DenseNet[16] 21
5.1.4 Dual Path Network[17] 21
5.2 Dropout 22
5.2.1 Dropout 22
5.2.2 SpatialDropout 22
5.2.3 DropConnect 22
5.3 Normalization[4] [18] 22
5.4 attention 23
5.5 随机权重网络 23
5.5.1 ELM（超限学习机） 23
5.5.2 ESN（回声状态网络） 23
5.6 capsule network[19] [20] 24
5.6.1 耦合系数c_{ij} 24
5.6.1.1 Dynamic Routing 24
5.6.1.2 EM Routing 24
5.7 图像分割 24
5.7.1 R-CNN[21] [22] [23] 24
5.7.1.1 基本思路 24
5.7.1.2 RoI Pooling 25
5.7.1.3 RPN 25
5.7.1.4 训练 25
5.7.2 YOLO[24] [25] [26] 25
5.7.2.1 25
5.7.2.2 训练 25
5.7.3 分割掩码[27] 25
5.8 梯度下降 26
5.8.1 SGD 26
5.8.2 Momentum 26
5.8.3 Nesterov Momentum 26
5.8.4 退火 26

5.8.5 Adagrad 26
5.8.6 Adadelta 26
5.9 激活函数 26

II 生成模型 28

6 显式概率分布 30
6.1 EM 30
6.1.1 GMM(高斯混合模型) 31
6.1.2 K-means 31
6.2 变分推断 32
6.2.1 平均场 32
6.2.1.1 目标 32
6.2.1.2 平均场假设 32
6.3 VAE 33
6.3.1 经典VAE[28] 33
6.3.1.1 目标 33
6.3.1.2 隐变量z假设 33
6.3.1.3 VAE 33
6.4 FVBN（Fully visible belief nets） 33
6.4.1 PixelRNN 34
6.4.2 ICA 34

7 隐式概率分布 35
7.1 GAN 35
7.1.1 传统GAN[29] [30] [31] 35
7.1.1.1 35
7.1.1.2 35
7.1.1.3 36
7.1.1.4 36
7.1.1.5 DCGAN[32] 36
7.1.1.6 CGAN[33] 36
7.1.1.7 InfoGAN[34] 36
7.1.2 GLS-GAN[35] 36
7.1.2.1 WGAN[36] 37
7.1.2.2 LS-GAN 38

III other 39

8 广义线性模型GLM 40
8.1 指数族分布 40
8.1.1 多项式分布 40

8.1.1.1 多项式分布 . . . 40
8.1.1.2 多项式分布 . . . 41
8.1.1.3 二项分布 . . . 41
8.1.1.4 伯努利分布(logistic回归) . . . 41
8.1.2 正态分布 . . . 42
8.1.2.1 二元正态分布 . . . 42
8.1.2.2 线性最小二乘法 . . . 42
8.1.3 其他例子 . . . 42
8.2 SVM . . . 43
8.2.0.1 核函数[37] . . . 43
8.2.0.2 松弛变量 . . . 44

9 流型学习 **45**
9.1 PCA（主成分分析） . . . 45
9.1.1 法一 . . . 45
9.1.2 法二 . . . 46
9.1.3 Kernel PCA . . . 46
9.1.4 白化 . . . 46
9.2 MDS（多维度尺度变换） . . . 47
9.3 isomap . . . 47
9.4 LLE . . . 47
9.5 LDA（线性判别分析） . . . 47

10 强化学习 **49**
10.1 强化学习 . . . 49
10.1.1 基本概念 . . . 49
10.1.2 已知模型 . . . 49
10.1.3 未知模型 . . . 50
10.1.3.1 蒙特卡洛法 . . . 50
10.1.3.2 时差学习 . . . 50
10.2 DQN（深度强化学习） . . . 50
10.2.0.3 Experience Replay . . . 50
10.2.0.4 Target Q . . . 51
10.2.0.5 Double DQN . . . 51
10.2.0.6 Prioritised replay . . . 51

11 决策树 **52**
11.1 单决策树 . . . 52
11.1.1 ID3 . . . 52
11.1.1.1 定义 . . . 52
11.1.1.2 算法 . . . 52
11.1.2 C4.5 . . . 52
11.1.2.1 定义 . . . 52

11.1.2.2 算法 . . . 53
11.1.3 最小二乘回归树 . . . 53
11.1.4 Cart分类树 . . . 53
11.2 Boosting . . . 53
11.2.1 随机森林 . . . 53
11.2.2 AdaBoost . . . 53
11.2.2.1 原理 . . . 53
11.2.2.2 具体算法 . . . 54
11.2.3 GBDT . . . 54

12 NLP **55**
12.1 隐含语义分析 . . . 55
12.1.1 PLSA . . . 55
12.1.2 LDA . . . 55
12.1.2.1 Dirichelet分布与多项分布 . . . 55
12.1.2.2 单主题 . . . 56
12.1.2.3 多主题 . . . 56
12.1.3 LFM(Latent factor model) . . . 57
12.2 统计语言模型 . . . 57
12.2.1 N-gram . . . 57
12.2.2 CBOW . . . 57
12.2.3 Skip-Gram . . . 58
12.2.4 隔词 . . . 58
12.3 词向量 . . . 58
12.3.1 One-hot Representation . . . 58
12.3.2 Distributed Representation . . . 58
12.3.2.1 Softmax . . . 58
12.3.2.2 Softmax . . . 58
12.3.2.3 Softmax的矩阵分解形式 . . . 59
12.3.2.4 Negative-Sampling [38] [39] . . . 59
12.3.2.5 Negative-Sampling的矩阵分解形式 [40] . . . 59
12.4 NMT（Neural Machine Translation） . . . 59
12.4.1 RNN . . . 59
12.4.2 seq2seq . . . 59
12.4.3 attention . . . 60

13 图结构学习 **61**
13.1 结点表示 . . . 61
13.1.1 GNN [41] . . . 61
13.1.2 DeepWalk [42] . . . 61
13.1.3 GraRep [43] . . . 61
13.1.4 类比CNN [44] . . . 62

13.2 图表示 62
13.2.1 62
13.2.2 Graph Kernel[45] 62

Part I

ANN

Chapter 1

CNN（卷积神经网络）[1]

1.1 卷积层

提取特征。

第l层、第k_l个卷积核：M^{l,k_l}*N^{l,k_l}的卷积核与输入图层每M^{l,k_l}*N^{l,k_l}的框点乘。框之间可能有重叠，依框边长$M^{l,k_l}N^{l,k_l}$与跨步$u_l v_l$决定。

$$x_{m,n}^{l,k_l} = f^{l,k_l}\left(\sum_{i,j,k_{l-1}} w_{i,j,k_{l-1}}^{l,k_l} x_{mu_l+i,nv_l+j,k_{l-1}}^{l-1} + b^{l,k_l}\right) \quad (i \in \pm\frac{M^{l,k_l}-1}{2}, j \in \pm\frac{N^{l,k_l}-1}{2})$$

1.1.1 卷积分组

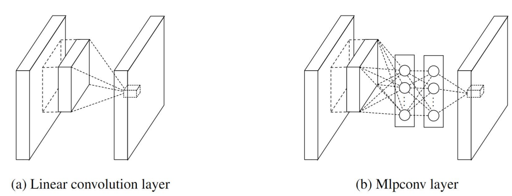

Figure 1.1: NIN

1.1.1.0.1 NIN[2] 将每一个卷积层替换为一个卷积层+多层全连接层。等价于在每一个卷积层后面加上多层1*1卷积层

1.1.1.0.2 Inception[3] [4] [5] [6]

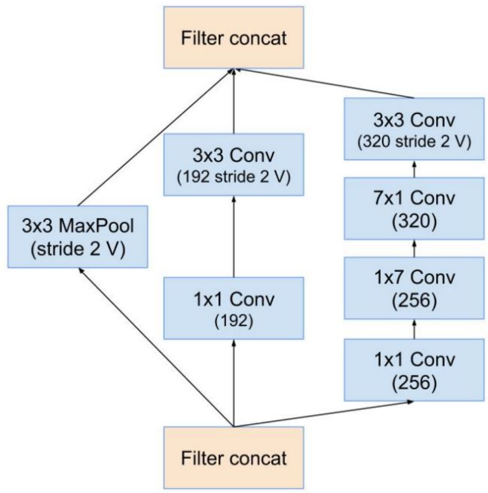

Figure 1.2: Inception

1.1.1.0.3 MobileNet[7] 假定$\boxed{W}^{k_l}_{k_{l-1}} \simeq u^{k_l}_{k_{l-1}} \boxed{V}_{k_{l-1}}$，即

$$\begin{aligned} \begin{bmatrix} \vdots \\ \boxed{X^l}^{k_l} \\ \vdots \end{bmatrix} &= \begin{bmatrix} & \vdots & \\ \cdots & \boxed{W}^{k_l}_{k_{l-1}} & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ \boxed{X^{l-1}}_{k_{l-1}} \\ \vdots \end{bmatrix} \\ &\simeq \begin{bmatrix} & \vdots & \\ \cdots & u^{k_l}_{k_{l-1}} & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ \boxed{V}_{k_{l-1}} \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} \vdots \\ \boxed{X^{l-1}}_{k_{l-1}} \\ \vdots \end{bmatrix} \end{aligned}$$

将空间相关与通道相关耦合的$\boxed{W}^{k_l}_{k_{l-1}}$，解耦成通道相关的$u^{k_l}_{k_{l-1}}$ 与空间相关的单通道$\boxed{V}_{k_{l-1}}$

假定$\boxed{W}^{k_l}_{k_{l-1}} \simeq \boxed{V}_{k_l} u^{k_l}_{k_{l-1}}$，即

$$\begin{aligned} \begin{bmatrix} \vdots \\ \boxed{X^l}^{k_l} \\ \vdots \end{bmatrix} &= \begin{bmatrix} & \vdots & \\ \cdots & \boxed{W}^{k_l}_{k_{l-1}} & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ \boxed{X^{l-1}}_{k_{l-1}} \\ \vdots \end{bmatrix} \\ &\simeq \begin{bmatrix} \vdots \\ \boxed{V}_{k_l} \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} & \vdots & \\ \cdots & u^{k_l}_{k_{l-1}} & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ \boxed{X^{l-1}}_{k_{l-1}} \\ \vdots \end{bmatrix} \end{aligned}$$

将空间相关与通道相关耦合的$\boxed{W}^{k_l}_{k_{l-1}}$，解耦成空间相关的单通道$\boxed{V}_{k_l}$ 与通道相关的$u^{k_l}_{k_{l-1}}$

1.1.2 实现

1.1.2.0.4 Caffe 输出= 权重* 输入

- 输入：输入通道数*输入单通道长度

- 权重：输出通道数*(输入通道数*感受野)
- 输出：输出通道数*输出单通道长度

1. 输入重排为：(输入通道数*感受野)*输出单通道长度
2. 输出= 权重*输入重排

1.2 池化层

平移对称性。

第l层：输入图层每$N^{l,k}*N^{l,k}$的框选出代表值。框之间不重叠

$$x_{m,n}^{l,k} = \mathrm{pool}(x_{i,j}^{l-1,k}) \quad (i,j \in m,n \pm \frac{N^{l,k}-1}{2})$$

$$x_{m,n}^{l,k} = \left(\sum_{i,j,k_{l-1}} |x_{mu_l+i,nv_l+j}^{l-1,k}|^p \right)^{\frac{1}{p}} \quad (i \in \pm\frac{M^{l,k_l}-1}{2}, j \in \pm\frac{N^{l,k_l}-1}{2})$$

1.2.1 [8]

当层数足够深时，将“卷积层+池化层”替换为“含跨步的卷积层”能得到大致相同的效果

1.2.2 SPP（空间金字塔池化）[9]

不同尺寸的输入，经过池化后输出固定尺寸的大小。框的尺寸随输入尺寸的变化而变化

1.3 全连接层

Chapter 2

RNN

2.1 RNN

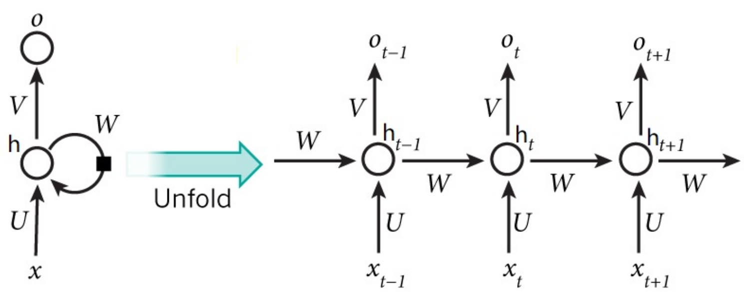

Figure 2.1: RNN

输入单元$\{\cdots, x_{t-1}, x_t, x_{t+1}, \cdots\}$，隐藏单元$\{\cdots, h_{t-1}, h_t, h_{t+1}, \cdots\}$，输出单元$\{\cdots, y_{t-1}, y_t, y_{t+1}, \cdots\}$。

$$h_t = f(Ux_t + Wh_{t-1})$$

$$o_t = softmax(Vh_t)$$

2.2 LSTM

2.2.1 LSTM

$$\begin{aligned}
\text{记忆门}_t &= \sigma(W_f * [\text{输出}_{t-1}, \text{输入}_t]) \\
\text{输入门}_t &= \sigma(W_i * [\text{输出}_{t-1}, \text{输入}_t]) \\
\text{备选}_t &= tanh(W_c * [\text{输出}_{t-1}, \text{输入}_t]) \\
\text{状态}_t &= \text{记忆门}_t * \text{状态}_{t-1} + \text{输入门}_t * \text{备选}_t \\
\text{输出门}_t &= \sigma(W_o * [\text{输出}_{t-1}, \text{输入}_t]) \\
\text{输出}_t &= \text{输出门} * tanh(\text{状态}_t)
\end{aligned}$$

2.2.2 peephole connection

$$
\begin{aligned}
\text{记忆门}_t &= \sigma(W_f * [\text{状态}_{t-1}, \text{输出}_{t-1}, \text{输入}_t]) \\
\text{输入门}_t &= \sigma(W_i * [\text{状态}_{t-1}, \text{输出}_{t-1}, \text{输入}_t]) \\
\text{备选}_t &= tanh(W_c * [\text{输出}_{t-1}, \text{输入}_t]) \\
\text{状态}_t &= \text{记忆门}_t * \text{状态}_{t-1} + \text{输入门}_t * \text{备选}_t \\
\text{输出门}_t &= \sigma(W_o * [\text{状态}_t, \text{输出}_{t-1}, \text{输入}_t]) \\
\text{输出}_t &= \text{输出门} * tanh(\text{状态}_t)
\end{aligned}
$$

2.2.3 coupled记忆门与输入门

$$
\begin{aligned}
\text{记忆门}_t &= \sigma(W_f * [\text{输出}_{t-1}, \text{输入}_t]) \\
\text{输入门}_t &= 1 - \text{记忆门}_t \\
\text{备选}_t &= tanh(W_c * [\text{输出}_{t-1}, \text{输入}_t]) \\
\text{状态}_t &= \text{记忆门}_t * \text{状态}_{t-1} + \text{输入门}_t * \text{备选}_t \\
\text{输出门}_t &= \sigma(W_o * [\text{输出}_{t-1}, \text{输入}_t]) \\
\text{输出}_t &= \text{输出门} * tanh(\text{状态}_t)
\end{aligned}
$$

2.2.4 GRU(Gated Recurrent Unit)

$$
\begin{aligned}
\text{记忆门}_t &= 1 - \text{输入门}_t \\
\text{输入门}_t &= \sigma(W_i * [\text{状态}_{t-1}, \text{输出}_{t-1}, \text{输入}_t]) \\
\text{重置门}_t &= \sigma(W_r * [\text{状态}_{t-1}, \text{输出}_{t-1}, \text{输入}_t]) \\
\text{备选}_t &= tanh(W_c * [\text{重置门}_t * \text{输出}_{t-1}, \text{输入}_t]) \\
\text{状态}_t &= \text{记忆门}_t * \text{状态}_{t-1} + \text{输入门}_t * \text{备选}_t \\
\text{输出门}_t &= \text{无} \\
\text{输出}_t &= \text{状态}_t
\end{aligned}
$$

2.3 Higher Order RNN[10]

$$h_t = f(Ux_t + \sum_{\Delta t=1}^{T} W_{\Delta t} h_{t-\Delta t})$$

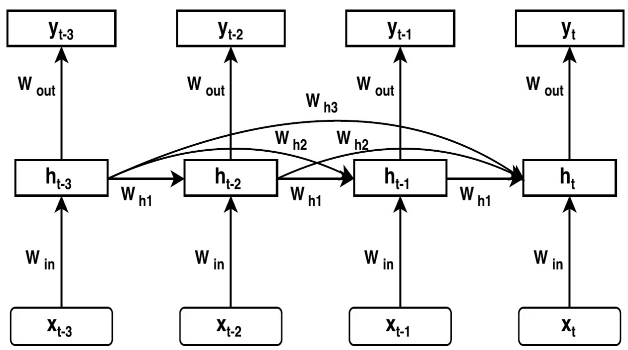

Figure 2.2: Higher Order RNN

Chapter 3

AE（自编码）

3.1 AE

3.1.1 AE

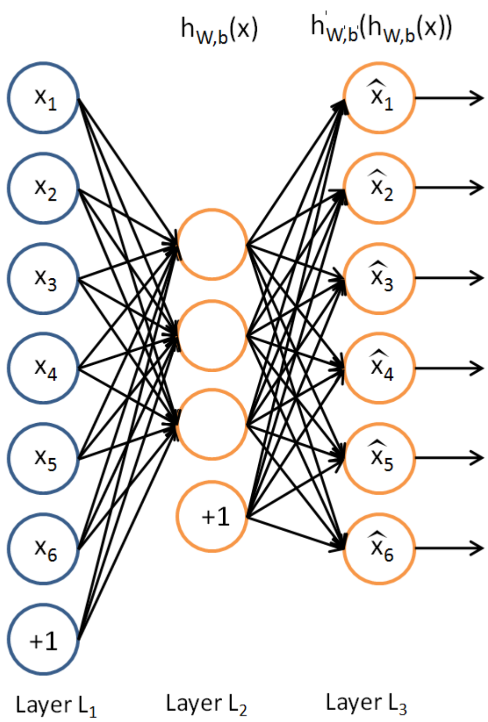

Figure 3.1: AE

一层隐藏层+一层输出层
无监督学习。输出层尽力还原输入层，则中间隐藏层为提取的特征。

- 隐藏层可取sigmoid函数
- 输出层取线性函数时，可取$L(x,\hat{x})=\frac{1}{2}||x-\hat{x}||$

- 输出层取sigmoid函数时，可取$L(x,\hat{x}) = -\sum_i (x_i \log \hat{x}_i + (1-x_i)\log(1-\hat{x}_i))$

3.1.2 稀疏性限制

限制神经元大部分时间$\vec{w} \cdot \check{x} < 0$

神经元j的激活度$\rho_j = \frac{1}{|S|}\sum_{\varsigma=1}^{|S|}[f(\vec{w}_j \cdot \check{x}_\varsigma)]$，期望接近于一个特定值$\rho$（譬如f为sigmoid函数时，可取$\rho = 0.05$）

在优化目标函数中加入惩罚因子$\sum_{j\in 隐藏层} KL(\rho||\rho_j)$。其中相对熵$KL(\rho||\rho_j) = \rho \ln \frac{\rho}{\rho_j} + (1-\rho)\ln\frac{1-\rho}{1-\rho_j}$

3.1.3 Denoising Auto-Encoder[11]

为提高鲁棒性，在自编码模型中，将输入x添加破坏变为y，经过自编码器得到$\hat{y}$，目标优化函数为$L(x,\hat{y})$。

可取$y = x +$ 高斯模型，或直接将x的某些分量随机为0得到y。

3.2 多层AE[12]

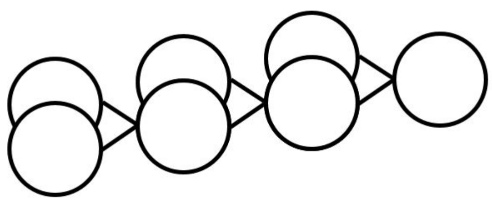

Figure 3.2: MulAE

逐层贪婪训练，每一层提取的特征作为下一层输入。（无监督训练完毕后再进行有监督训练为早期深度学习做法）

Chapter 4

Hopfield

4.1 Hopfield

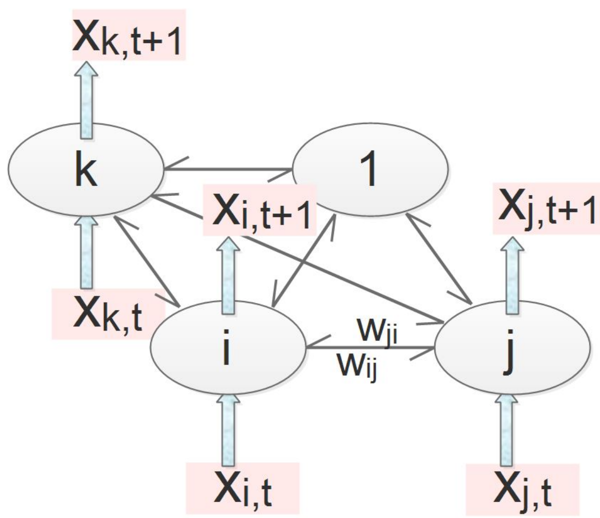

Figure 4.1: Hopfield

训练时：

$$E(\check{x}) \stackrel{def}{=} -\frac{1}{2}\begin{bmatrix} \cdots & x_i & \cdots \end{bmatrix}\begin{bmatrix} & \vdots & \\ \cdots & w_{ij} & \cdots \\ & \vdots & \end{bmatrix}\begin{bmatrix} \vdots \\ x_j \\ \vdots \end{bmatrix}$$

可取$\begin{cases} w_{ij} = w_{ji} \\ w_{ii} = 0 \end{cases}$，$x_i$只取双值。保证能量有最小值

优化方法：

1. 保持$\vec{w}$不变，改变$\check{x}$，至能量最小

2. 保持$\check{x}$不变，x_i、x_j值相同则增大w_{ij}，x_i、x_j值不同则减小w_{ij}
（譬如当x_i双值为1,-1时，$w_{ij} = \sum\limits_{\varsigma} x_{\varsigma i} x_{\varsigma j}$）

预测时：

4.1.1 DHNN（离散时间）

$$\begin{bmatrix} \vdots \\ x_{i,t+1} \\ \vdots \end{bmatrix} = f(\begin{bmatrix} & \vdots & \\ \cdots & w_{ij} & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ x_{j,t} \\ \vdots \end{bmatrix})$$

初始输入$\check{x}_0$进行迭代收敛至稳定点$\check{x}_T$，用$\check{x}_T$进行判别
可取

$$f(z) = \begin{cases} -1 & , \quad z < 0 \\ 1 & , \quad z \geq 0 \end{cases}$$

或

$$f(z) = \begin{cases} -1 & , \quad z < -1 \\ z & , \quad -1 \leq z \leq 1 \\ 1 & , \quad z > 1 \end{cases}$$

4.1.2 CHNN（连续时间）

$$E(\check{x}) \overset{def}{=} -\frac{1}{2} \begin{bmatrix} \cdots & x_i & \cdots \end{bmatrix} \begin{bmatrix} & \vdots & \\ \cdots & w_{ij} & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ x_j \\ \vdots \end{bmatrix} + \begin{bmatrix} \cdots & \frac{1}{R_i} & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ \int\limits_0^{x_i} f^{-1}(x')dx' \\ \vdots \end{bmatrix}$$

$$E(\check{x}(t)) \overset{def}{=} -\frac{1}{2} \begin{bmatrix} \cdots & x_i(t) & \cdots \end{bmatrix} \begin{bmatrix} & \vdots & \\ \cdots & w_{ij} & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ x_j(t) \\ \vdots \end{bmatrix} + \begin{bmatrix} \cdots & \frac{1}{R_i} & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ \int\limits_0^{x_i(t)} f^{-1}(x')dx' \\ \vdots \end{bmatrix}$$

可取

$$f(z) = \frac{1}{1 - e^{-z}}$$

4.2 Ising模型

$$w_{ij} = \begin{cases} J & \text{ij近邻} \\ H & \text{ij其中一个x为1} \\ 0 & \text{ij非近邻} \end{cases}$$

微观构型$\check{x}$的概率$p(\check{x}) = \frac{1}{Z} e^{-\frac{E(\check{x})}{kT}}$ （配分函数$Z = \sum\limits_{\check{x}} e^{-\frac{E(\check{x})}{kT}}$）

4.3 RBM（受限波尔兹曼机）

一层显层+一层隐层

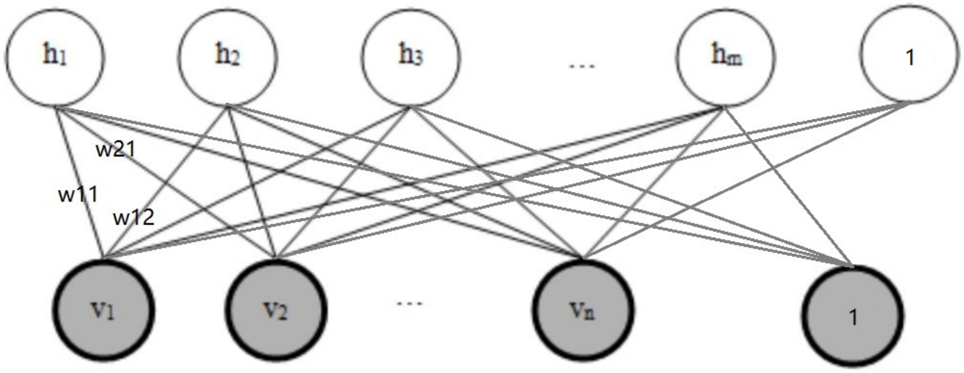

Figure 4.2: RBM

能量

$$E(\check{v},\check{h}) \overset{def}{=} -\begin{bmatrix} \cdots & v_i & \cdots \end{bmatrix} \begin{bmatrix} & \vdots & \\ \cdots & w_{ij} & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ h_j \\ \vdots \end{bmatrix}$$

概率

$$P(\check{v},\check{h}) \overset{def}{=} \frac{1}{Z}e^{-E(\check{v},\check{h})} = \frac{e^{-E(\check{v},\check{h})}}{\sum\limits_{\check{v}\check{h}} e^{-E(\check{v},\check{h})}}$$

自由能

$$F(\check{v}) \overset{def}{=} -ln\sum_{\check{h}} e^{-E(\check{v},\check{h})}$$

$$P(\check{v}) = \sum_{\check{h}} P(\check{v},\check{h}) = \frac{1}{Z}e^{-F(\check{v})}$$

优化目标

$$\text{argmax}_w \prod_{\check{v}_\varsigma} P(\check{v}) = \text{argmin}_w \sum_{\check{v}_\varsigma} F(\check{v})$$

即提取显层（训练样本）的特征藏于隐层参数中，最大概率还原显层

4.3.1 对比散度训练法

由每一个训练样本$\check{v}$ 求得$\check{h}$，再由$\check{h}$ 反求得$\check{v}'$。则$w_{ij} += \lambda(v_i - v_i')h_j$ 。循环训练至收敛($x_i = x_i'$)。改进：

1. 用多往返几次的$\check{v}'''$替代$\check{v}'$。
2. 用$p(v_i)$、$p(h_j)$替代v_i、h_j
3. 加正则项，对较大的权重w_{ij}进行惩罚
4. 用本次Δw_{ij}与多次前次Δw_{ij}线性加权

4.3.2 二项分布

假设h_j、v_i都只能取$\{0,1\}$

条件概率

$$P(h_j=1|\check{v})=\frac{1}{1+\exp{(-\sum\limits_i v_i w_{ij})}}$$

$$P(v_i=1|\check{v})=\frac{1}{1+\exp{(-\sum\limits_j w_{ij} h_j)}}$$

4.3.3 多项分布

假设h_j、v_i都只能取$(0,\cdots,0,1,0,\cdots,0)$其中之一

$$P(v_i^k=1|\check{h})=\frac{\exp{(\sum\limits_j w_{ij}^k h_j)}}{\sum\limits_{k=1}^K \exp{(\sum\limits_j w_{ij}^k h_j)}}$$

4.4 DBN（深度信念网络）

多层RBM组成，逐层训练，每层提取的特征作为下一层输入（即当前隐层作为下层隐层）。（训练完毕后再进行有监督训练为早期深度学习做法）

Chapter 5

other

5.1 Highway Network

5.1.1 Highway Network[13] [14]

将一层或多层由原本的$\check{y} = F(\check{x})$ 改为$\check{y} = F(\check{x}) * T(W_T\check{x}) + W_s\check{x} * C(W_C\check{x})$
（$*$表示按元素乘）
（W_s用于将$\check{x}$的维度转为与$F(W_F\check{x})$一致）

5.1.2 Deep Residual Network[15]

若某一多层网络可渐进估计某函数$H(\check{x})$，则等同可渐进估计$H(\check{x}) - \check{x}$
$W_T = T = W_C = C = 1$，即$\check{y} = F(\check{x}) + W_s\check{x}$

5.1.3 DenseNet[16]

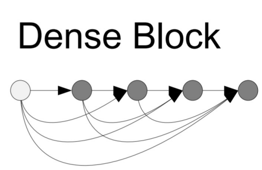

Figure 5.1: AE

采用拼接而非求和，发掘更多特征

5.1.4 Dual Path Network[17]

将Higher Order RNN 展开，可视为共享参数$f_{t,t'} = f_{\Delta t = t - t'}$ 的特例DenseNet：

$$h_t = F_t\left[\sum_{t'=0}^{t-1} f_{t,t'}(h_{t'})\right]$$

则可扩展为：

$$\begin{aligned} h_t &= F_t\left[\sum_{t'=0}^{t-1} f_{t,t'}(h_{t'}) + y_t\right] \\ y_t &\overset{def}{=} \sum_{t'=0}^{t-1} = y_{t-1}\phi_{t-1}(y_{t-1}) \end{aligned}$$

即在DenseNet 的基础上加入类似ResNet 的直接跨层连接

5.2 Dropout

使网络不过分依赖于某些特定神经元，在缺失某些特定信息时依然有效，提高健壮性

5.2.1 Dropout

- 训练过程中，每个神经元以一定概率p失活为0，若非0则其输出结果再除以p以恢复原大小
- 预测时不失活

5.2.2 SpatialDropout

5.2.3 DropConnect

每个权重w_{ij}以一定概率p失活为0

5.3 Normalization[4][18]

引入Normalization，让每层张量取值分布固定
每层张量大小= 尺寸L * 神经元个数C * batch大小N
将该张量分组

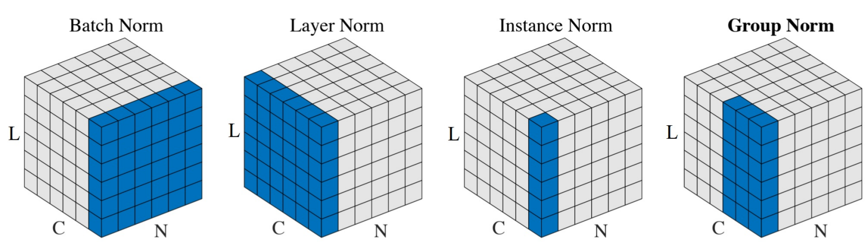

Figure 5.2: Normalization

1. 将x_i 化为均值0方差1：$x_i' = \frac{x_i - \mu_I}{\sqrt{\sigma_I^2 + \varepsilon}}$
2. 将x_i' 进行变换：$x_i'' = \gamma_I x_i' + \beta_I$ （γ_I、β_I作为参数在网络中迭代训练）

5.4 attention

由一个query 与一组$\{key_i, value_i\}$ 组成输出

$$\begin{aligned}\text{投影}c_i &= C(\text{key}_i, \text{query})\\ \text{输出output} &= O(\{c_i\}, \{\text{value}_i\})\end{aligned}$$

若函数O为加权求和，则可视为

$$|\text{output}\rangle = \sum_i |\text{value}_i\rangle \langle \text{key}_i|\text{query}\rangle$$

5.5 随机权重网络

5.5.1 ELM（超限学习机）

一层隐藏层+一层输出结点
训练样本集$\{(\vec{x}_\varsigma, \vec{y}_\varsigma)\}$。
隐藏层L个结点输出

$$\begin{bmatrix} & \vdots & \\ \cdots & f(\vec{W}_l^{12}\cdot\check{x}_\varsigma) & \cdots \\ & \vdots & \end{bmatrix}$$

输出层输出

$$\begin{bmatrix} \cdots & \vec{W}_l^{23} & \cdots \end{bmatrix}\begin{bmatrix} & \vdots & \\ \cdots & f(\vec{W}_l^{12}\cdot\check{x}_\varsigma) & \cdots \\ & \vdots & \end{bmatrix} = \begin{bmatrix} \cdots & \vec{y}_\varsigma & \cdots \end{bmatrix}$$

随机取$\vec{W}_l$，固定不变。则

$$\begin{bmatrix} \cdots & \vec{W}_l^{23} & \cdots \end{bmatrix} = \begin{bmatrix} \cdots & \vec{y}_\varsigma & \cdots \end{bmatrix}\begin{bmatrix} & \vdots & \\ \cdots & f(\vec{W}_l^{12}\cdot\check{x}_\varsigma) & \cdots \\ & \vdots & \end{bmatrix}^+$$

其中 $^+$为Moore-Penrose广义逆

5.5.2 ESN（回声状态网络）

将ELM扩展为时序网络，隐藏层

$$\begin{bmatrix} & \vdots & \\ \cdots & z_{l\varsigma}(t) & \cdots \\ & \vdots & \end{bmatrix} = \begin{bmatrix} & \vdots & \\ \cdots & f(\vec{W}_l^{12}\cdot\check{x}_\varsigma(t) + \vec{W}_l^{22}\cdot\vec{z}_\varsigma(t-1)) & \cdots \\ & \vdots & \end{bmatrix}$$

为保持稳定，需满足

- 矩阵W^{22} 的特征值绝对值必须小于1
- 矩阵W^{22} 为稀疏矩阵。稀疏度 $\overset{def}{=} \frac{\text{相互连接神经元数}}{\text{总神经元数}}$

5.6 capsule network[19][20]

将原本每个神经元替换为capsule，输出标量替换为输出张量。模长代表概率，方向代表属性。

一个capsule代表识别一类特征。当出现该类特征时该capsule都会给出高概率，而该类特征的不同具体实例体现在该张量其他自由度中。

	tradition	capsule
加权混合	$\begin{bmatrix} \vdots \\ z_j^{(t)} \\ \vdots \end{bmatrix} = \begin{bmatrix} & \vdots & \\ \cdots & w_{ij}^{(t)} & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ x_i^{(t)} \\ \vdots \end{bmatrix}$	$\begin{bmatrix} \vdots \\ \vec{z}_j^{(t)} \\ \vdots \end{bmatrix} = \begin{bmatrix} & \vdots & \\ \cdots & w_{ij}'^{(t)} & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ \vec{x}_i^{(t)} \\ \vdots \end{bmatrix}$
激活函数	$x_j^{(t+1)} = F(z_j^{(t)})$	$\vec{x}_j^{(t+1)} = F(\vec{z}_j^{(t)})$

$$w'_{ij} = w_{ij} c_{ij}$$

- w_{ij}作为参数优化
- c_{ij}为i与j间的耦合系数，表征了i对j的贡献

$F(\vec{z})$ 保持$\vec{z}$方向不变，模长单调缩放至$[0,1]$

损失函数：因输出模长代表概率，故最大化正确模长、最小化错误模长

5.6.1 耦合系数c_{ij}

5.6.1.1 Dynamic Routing

Listing 5.1: Dynamic Routing

```
for:
```

$$\begin{aligned} \vec{u}_{j|i}^{(t)} &= w_{ij}^{(t)} \vec{x}_i^{(t)} \\ b_{ij}^{(t)} &\mathrel{+}= \vec{u}_{j|i}^{(t)} \cdot \vec{x}_j^{(t+1)} \\ c_{ij}^{(t)} &= \frac{e^{b_{ij}}}{\sum\limits_j e^{b_{ij}}} \end{aligned}$$

5.6.1.2 EM Routing

5.7 图像分割

5.7.1 R-CNN[21][22][23]

5.7.1.1 基本思路

1. 每张图预框出一组子图
2. 判别每张子图是否有效，若有效则判别图像类别
3. 每张有效子图回归出更紧密的子图边框（中心、边长）

5.7.1.2 RoI Pooling

- 输入图X，经各卷积层，输出图X'；
- 输入图X的子图A，经各卷积层，输出亦为图X'的子图A'。

故“输入各子图”等价为“输入一张原图，最后一层再框分割”

5.7.1.3 RPN

因预框子图被推迟到最后一层特征层，不如改用另一个网络RPN进行预框子图。
RPN遍历各大小窗口作为预子图

1. 判别每个预子图是否有效
2. 每张有效子图回归出大致子图边框

有效预子图作为给R-CNN的预框子图，进行更精细识别
PRN与R-CNN交替更新参数，直至二者皆收敛。

5.7.1.4 训练

面积交并比IoU $\overset{def}{=} \frac{\bigcap_i S_i}{\bigcup_i S_i}$
回归子图边框的训练以预测子图与真实子图二者的IoU为标准
为避免重复，最后一步要用非极大值抑制NMS：每个真实子图仅保留对应最匹配的选取子图，与该最匹配子图IoU较大的有效子图都筛掉。

5.7.2 YOLO[24] [25] [26]

5.7.2.1

将全图划分为固定网格，每个目标由其中心所在网格负责识别
每个网格回归

- 若干个目标的边框（中心、边长）与置信度（$P(\text{有目标}) * \text{IoU}_{\text{预测}}^{\text{真实}}$）
- 该网格/该边框为每一类的概率$P(\text{类别}|\text{有目标})$

则可乘出每个目标的$P(\text{类别}) * \text{IoU}_{\text{预测}}^{\text{真实}}$

5.7.2.2 训练

以$1 - \text{IoU}$ 作为距离，对大量子图进行kmeans聚类，剔除同一目标的重复识别

5.7.3 分割掩码[27]

对于每个类，由识别出特征层上采样得浮点数掩码

5.8 梯度下降

$$\theta_t = \theta_{t-1} + \Delta\theta_t$$

5.8.1 SGD

$$\Delta\theta_t = -\eta\nabla_{\theta_{t-1}} f(\theta_{t-1})$$

5.8.2 Momentum

$$\Delta\theta_t = \mu\Delta\theta_{t-1} - \eta\nabla_{\theta_{t-1}} f(\theta_{t-1})$$

5.8.3 Nesterov Momentum

$$\Delta\theta_t = \mu\Delta\theta_{t-1} - \eta\nabla_{\theta_{t-1}} f(\theta_{t-1} + \mu\Delta\theta_{t-1})$$

5.8.4 退火

$$\Delta\theta_t = -\frac{\eta}{1+dt}\nabla_{\theta_{t-1}} f(\theta_{t-1})$$

5.8.5 Adagrad

$$\begin{array}{rcl} n_t & = & n_{t-1} + (\nabla_{\theta_{t-1}} f(\theta_{t-1}))^2 \\ \Delta\theta_t & = & -\frac{\eta}{\sqrt{n_t+\varepsilon}}\nabla_{\theta_{t-1}} f(\theta_{t-1}) \\ & = & -\frac{\eta}{\sqrt{\sum\limits_{i=1}^{t-1}(\nabla_{\theta_i} f(\theta_i))^2+\varepsilon}}\nabla_{\theta_{t-1}} f(\theta_{t-1}) \end{array}$$

5.8.6 Adadelta

$$\begin{array}{rcl} n_t & = & \nu n_{t-1} + (1-\nu)\nabla_{\theta_{t-1}} f(\theta_{t-1})^2 \\ \theta_t & = & \theta_{t-1} - \frac{\eta}{\sqrt{n_t+\varepsilon}}\nabla_{\theta_{t-1}} f(\theta_{t-1}) \end{array}$$

5.9 激活函数

- $\sigma(x) = \frac{1}{1+e^{-x}}$
- $\mathrm{Relu}(x) = \begin{cases} x & ,x > 0 \\ 0 & ,x < 0 \end{cases}$

- $\mathrm{LRelu}(x) = \begin{cases} x & ,x > 0 \\ \alpha x & ,x < 0 \end{cases}$ （α为固定常数）[46]
- $\mathrm{PRelu}(x) = \begin{cases} x & ,x > 0 \\ \alpha x & ,x < 0 \end{cases}$ （α为优化系数）[47]
- $\mathrm{ELU}(x) = \begin{cases} \alpha & ,x > 0 \\ \alpha(e^x - 1) & ,x < 0 \end{cases}$ （a为固定系数）[48]
- $\mathrm{Swish}(x) = \frac{x}{1+e^{-x}}$ [49]

Part II

生成模型

由真实$P_R(\check{x})$ 的数据分布$\{\check{x}_\varsigma\}$，估计$P_G(\check{x})$ 逼近$P_R(\check{x})$
[1]

[1]表征两概率分布$P(x)$与$Q(x)$间距离：

- 相对熵（Kullback－Leibler距离）

$$\begin{aligned} \mathrm{KL}(P||Q) &\stackrel{def}{=} \int P(x)\log\frac{P(x)}{Q(x)}dx \\ &= -H(P)-\int P(x)\log Q(x)dx \end{aligned}$$

  有

$$\begin{aligned} & \mathrm{KL}(P||Q) \\ = & -\int P(x)\log\frac{Q(x)}{P(x)}dx \\ = & -E_{P(x)}[\log\frac{Q(x)}{P(x)}] \\ \geq & -\log E_{P(x)}[\frac{Q(x)}{P(x)}] \\ = & -\log\int Q(x)dx \\ = & \ 0 \end{aligned}$$

  当$P(x)$与$Q(x)$越接近，$\mathrm{KL}(P(x)||Q(x))$越小

- Jensen－Shannon距离

$$\mathrm{JSD}_{\pi_1,\cdots,\pi_n}(P_1||\cdots||P_n) \stackrel{def}{=} H(\sum_i \pi_i P_i) - \sum_i \pi_i H(P_i) \geq 0$$

  π_i是P_i的权重

Chapter 6

显式概率分布

求出训练样本的显式概率分布$P(\check{x}|\vec{\theta})$

6.1 EM

观测变量x_ς，隐变量z_ς，参数θ

$$P(x_\varsigma|\theta) = \int dz_\varsigma P(x_\varsigma, z_\varsigma|\theta) = \int dz_\varsigma P(x_\varsigma|z_\varsigma, \theta) P(z_\varsigma|\theta)$$

$$P(\{x_\varsigma\}|\theta) = \prod_\varsigma P(x_\varsigma|\theta)$$

为求$\theta^* = \mathrm{argmax}_\theta P(\{x_\varsigma\}|\theta)$，迭代$\theta^{(t)}$

由

$$\begin{aligned} & \log P(\{x_\varsigma\}|\theta) \\ = & \sum_\varsigma \log P(x_\varsigma|\theta) \\ = & \sum_\varsigma \log P(x_\varsigma, z_\varsigma|\theta) - \sum_\varsigma \log P(z_\varsigma|x_\varsigma, \theta) \end{aligned}$$

以$P(z_\varsigma|x_\varsigma, \theta^{(t)})$为概率测度对求和中每项$\varsigma$求期望

$$\begin{aligned} & \log P(\{x_\varsigma\}|\theta) \\ = & \sum_\varsigma E_{P(z_\varsigma|x_\varsigma, \theta^{(t)})}[\log P(x_\varsigma, z_\varsigma|\theta)] - \sum_\varsigma E_{P(z_\varsigma|x_\varsigma, \theta^{(t)})}[\log P(z_\varsigma|x_\varsigma, \theta)] \\ \stackrel{def}{=} & Q(\theta, \theta^{(t)}) - H(\theta, \theta^{(t)}) \end{aligned}$$

因

$$H(\theta, \theta^{(t)}) - H(\theta^{(t)}, \theta^{(t)}) = -\sum_\varsigma \mathrm{KL}[p(z_\varsigma|x_\varsigma, \theta^{(t)})||p(z_\varsigma|x_\varsigma, \theta)] \leq 0$$

即

$$H(\theta^{(t+1)}, \theta^{(t)}) \leq H(\theta^{(t)}, \theta^{(t)})$$

则只需取$\theta^{(t+1)}$使得

$$Q(\theta^{(t+1)}, \theta^{(t)}) \geq Q(\theta^{(t)}, \theta^{(t)})$$

譬如

$$\theta^{(t+1)} = \mathrm{argmax}_\theta Q(\theta, \theta^{(t)})$$

即可满足

$$P(\{x_\varsigma\}|\theta^{(t+1)}) \geq P(\{x_\varsigma\}|\theta^{(t)})$$

从而使迭代逐步收敛至$\theta^* = \mathrm{argmax}_\theta P(\{x_\varsigma\}|\theta)$

6.1.1 GMM(高斯混合模型)

从K个正态分布中挑出一个，挑到第z_ς个概率为α_{z_ς}。第k个正态分布的概率$\phi(x_\varsigma|\mu_k\sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k}e^{-\frac{(x_\varsigma-\mu_k)^2}{2\sigma_k^2}}$

$$\begin{aligned} & P(x_\varsigma, z_\varsigma|\vec{\alpha}\vec{\mu}\vec{\sigma}) \\ = & P(x_\varsigma|z_\varsigma, \vec{\alpha}\vec{\mu}\vec{\sigma})P(z_\varsigma|\vec{\alpha}\vec{\mu}\vec{\sigma}) \\ = & \alpha_{z_\varsigma}\phi(x_\varsigma|\mu_{z_\varsigma}\sigma_{z_\varsigma}) \end{aligned}$$

$$\begin{aligned} & P(z_\varsigma|x_\varsigma, \vec{\alpha}^{(t)}\vec{\mu}^{(t)}\vec{\sigma}^{(t)}) \\ = & \frac{P(x_\varsigma, z_\varsigma|\vec{\alpha}^{(t)}\vec{\mu}^{(t)}\vec{\sigma}^{(t)})}{P(x_\varsigma|\vec{\alpha}^{(t)}\vec{\mu}^{(t)}\vec{\sigma}^{(t)})} \\ = & \frac{\alpha_{z_\varsigma}^{(t)}\phi(x_\varsigma|\mu_{z_\varsigma}^{(t)}\sigma_{z_\varsigma}^{(t)})}{\sum\limits_{k=1}^{K}\alpha_k^{(t)}\phi(x_\varsigma|\mu_k^{(t)}\sigma_k^{(t)})} \end{aligned}$$

$$\begin{aligned} & Q(\vec{\alpha}\vec{\mu}\vec{\sigma}|\vec{\alpha}^{(t)}\vec{\mu}^{(t)}\vec{\sigma}^{(t)}) \\ = & \sum_\varsigma E_{P(z_\varsigma|x_\varsigma,\vec{\alpha}^{(t)}\vec{\mu}^{(t)}\vec{\sigma}^{(t)})}[\log P(x_\varsigma, z_\varsigma|\vec{\alpha}\vec{\mu}\vec{\sigma})] \\ = & \sum_\varsigma \sum_{z_\varsigma=1}^{K} P(z_\varsigma|x_\varsigma, \vec{\alpha}^{(t)}\vec{\mu}^{(t)}\vec{\sigma}^{(t)}) \log P(x_\varsigma, z_\varsigma|\vec{\alpha}\vec{\mu}\vec{\sigma}) \end{aligned}$$

反复迭代

- $\mu_k^{(t+1)} = \frac{\sum\limits_\varsigma x_\varsigma P(k|x_\varsigma,\alpha_k^{(t)}\mu_k^{(t)}\sigma_k^{(t)})}{\sum\limits_\varsigma P(k|x_\varsigma,\alpha_k^{(t)}\mu_k^{(t)}\sigma_k^{(t)})}$
- $\sigma_k^{(t+1)} = \sqrt{\frac{\sum\limits_\varsigma (x_\varsigma-\mu_k)^2 P(k|x_\varsigma,\alpha_k^{(t)}\mu_k^{(t)}\sigma_k^{(t)})}{\sum\limits_\varsigma P(k|x_\varsigma,\alpha_k^{(t)}\mu_k^{(t)}\sigma_k^{(t)})}}$
- $\alpha_k^{(t+1)} = \frac{\sum\limits_\varsigma P(k|x_\varsigma,\alpha_k^{(t)}\mu_k^{(t)}\sigma_k^{(t)})}{\sum\limits_{k'}\sum\limits_\varsigma P(k'|x_\varsigma,\alpha_{k'}^{(t)}\mu_{k'}^{(t)}\sigma_{k'}^{(t)})}$

6.1.2 K-means

将训练数据集聚类为K类

$\alpha_k = \frac{1}{K}$，$\sigma_k = 1$

$$\begin{aligned} & P(x_\varsigma, z_\varsigma|\vec{\mu}) \\ = & P(x_\varsigma|z_\varsigma, \vec{\mu})P(z_\varsigma|\vec{\mu}) \\ = & \frac{1}{K}\phi(x_\varsigma|\mu_{z_\varsigma}) \end{aligned}$$

$$\begin{aligned} & P(z_\varsigma|x_\varsigma, \vec{\mu}^{(t)}) \\ = & \frac{P(x_\varsigma, z_\varsigma|\vec{\mu}^{(t)})}{P(x_\varsigma|\vec{\mu}^{(t)})} \\ = & \frac{\phi(x_\varsigma|\mu_{z_\varsigma}^{(t)})}{\sum\limits_{k=1}^{K}\phi(x_\varsigma|\mu_k^{(t)})} \end{aligned}$$

$$\begin{aligned}
& Q(\vec{\mu}|\vec{\mu}^{(t)}) \\
= & \sum_{\varsigma} E_{P(z_\varsigma|x_\varsigma,\vec{\mu}^{(t)})}[\log P(x_\varsigma, z_\varsigma|\vec{\mu})] \\
= & \sum_{\varsigma}\sum_{z_\varsigma=1}^{K} P(z_\varsigma|x_\varsigma,\vec{\mu}^{(t)}) \log P(x_\varsigma, z_\varsigma|\vec{\mu})
\end{aligned}$$

反复迭代

- $\mu_k^{(t+1)} = \frac{\sum_\varsigma x_\varsigma P(k|x_\varsigma,\mu_k^{(t)})}{\sum_\varsigma P(k|x_\varsigma,\mu_k^{(t)})}$

为简化计算，取$P(k|x_\varsigma,\mu_k^{(t)}) \simeq \delta($ x_ς 离$\mu_k^{(t)}$ 最近)，即将每个x_ς归到离它最近的$\mu_k^{(t)}$类上，则

- $\mu_k^{(t+1)} =$ 所有第k类的x_ς的平均值

6.2 变分推断

6.2.1 平均场

6.2.1.1 目标

已知$\{x\}$（即已知$p(x)$），为求$p(\vec{z}|x)$，用人造$q(\vec{z})$拟合之。

由

$$\begin{aligned}
& \log p(x) \\
= & \log \frac{p(x,\vec{z})}{p(\vec{z}|x)} \\
= & \log \frac{q(\vec{z})}{p(\vec{z}|x)} + \log p(x,\vec{z}) - \log q(\vec{z})
\end{aligned}$$

以$q(\vec{z})$为概率测度求期望

$$\begin{aligned}
& \log p(x) \\
= & \mathrm{KL}[q(\vec{z})||p(\vec{z}|x)] + E_{q(\vec{z})}[\log p(x,\vec{z})] - E_{q(\vec{z})}[\log q(\vec{z})]
\end{aligned}$$

当KL$[q(\vec{z})||p(\vec{z}|x)]$最小时，$q(\vec{z})$与$p(\vec{z}|x)$最接近，故只需使ELOB$[q(\vec{z})] \overset{def}{=} E_{q(\vec{z})}[\log p(x,\vec{z})] - E_{q(\vec{z})}[\log q(\vec{z})]$最大。

6.2.1.2 平均场假设

人造$q(\vec{z}) = \prod_{i=1}^N q_i(z_i)$

$$\begin{pmatrix}
\text{边缘分布} & & & & & & \\
p_j(z_j|q(\vec{z})) & = & E_{q(\vec{z})\neg q_j(z_j)}[p(z_1,\cdots,z_N)] & = & \prod_{i\neq j}\int dz_i q_i(z_i) & & p(z_1,\cdots,z_N) \\
\tilde{p}_j(z_j|q(\vec{z})) & \overset{def}{=} & \exp(E_{q(\vec{z})\neg q_j(z_j)}[\log p(z_1,\cdots,z_N)]) & = & \exp(\ \prod_{i\neq j}\int dz_i q_i(z_i) & \log & p(z_1,\cdots,z_N)\ \ )
\end{pmatrix}$$

$$E_{q(\vec{z})}[\log p(x,\vec{z})] = \int dz_j q_j(z_j) E_{q(\vec{z})\neg q_j(z_j)}[\log p(x,\vec{z})] = \int dz_j q_j(z_j) \log \tilde{p}_j(x, z_j|q(\vec{z}))$$

$$E_{q(\vec{z})}[\log q(\vec{z})] = \int d\vec{z} \prod_{i=1}^N q_i(z_i) \sum_{j=1}^N \log q_j(z_j) = \sum_{i=1}^N \int dz_i q_i(z_i) \log q_i(z_i) = \int dz_j q_j(z_j) \log q_j(z_j) + \mathrm{const}$$

则

$$\mathrm{ELOB}[q(\vec{z})] = -\mathrm{KL}[q_j(z_j)||\tilde{p}_j(x, z_j|q(\vec{z}))] + \mathrm{const}$$

$$\log p(x) = -\mathrm{KL}[q_j(z_j)||\tilde{p}_j(x, z_j|q(\vec{z}))] + \mathrm{KL}[q(\vec{z})||p(\vec{z}|x)] + \mathrm{const}$$

故反复迭代$q_j^{(t+1)}(z_j) = \tilde{p}_j(x, z_j|q^{(t)}(\vec{z}))$ 至收敛即可

6.3 VAE

6.3.1 经典VAE[28]

6.3.1.1 目标

已知$\{x\}$（即已知$p(x)$），为求$p(z|x)$，用人造$q(z|x)$拟合之。

$$\begin{aligned}
& \log p(x) \\
= \quad & \log p(x,z) - \log p(z|x) \\
= \quad & \log \tfrac{q(z|x)}{p(z|x)} - \log q(z|x) + \log p(x,z) \\
= \quad & \log \tfrac{q(z|x)}{p(z|x)} - \log \tfrac{q(z|x)}{p(z)} + \log p(x|z)
\end{aligned}$$

以$q(z|x)$为概率测度求期望

$$\begin{aligned}
& \log p(x) \\
= \quad & \mathrm{KL}[q(z|x)||p(z|x)] + E_{q(z|x)}[-\log q(z|x) + \log p(x,z)] \\
= \quad & \mathrm{KL}[q(z|x)||p(z|x)] - \mathrm{KL}[q(z|x)||p(z)] + E_{q(z|x)}[\log p(x|z)]
\end{aligned}$$

当$\mathrm{KL}[q(z|x)||p(z|x)]$最小时，$q(z|x)$与$p(z|x)$最接近，故只需使$-\mathrm{KL}[q(z|x)||p(z)] + E_{q(z|x)}[\log p(x|z)]$ 最大。

6.3.1.2 隐变量z假设

假设$q(z|x)$为：z由x生成，即$z = \tilde{z}(\epsilon, x)$，其中噪音$\epsilon \sim p(\epsilon)$。则

$$E_{q(z|x)}[f(z)] = E_{p(\epsilon)}[f(\tilde{z}(\epsilon, x))] = \frac{1}{L}\sum_{l=1}^{L} f(\tilde{z}(\epsilon_l, x)) \quad (\epsilon \sim p(\epsilon))$$

6.3.1.3 VAE

- encoder 网络：输入x_ς，输出$\tilde{z}(\epsilon, x_\varsigma)$的参数，和抽样的$\epsilon_l$一起生成$\tilde{z}(\epsilon_l, x_\varsigma)$
- decoder 网络：输入z_ς，输出$p(x|z_\varsigma)$的参数，按概率生成x'_ς

训练目标：使encoder输入x_ς与decoder输出x'_ς尽可能相同，正则项为$\mathrm{KL}[q(z|x)||p(z)]$

6.4 FVBN（Fully visible belief nets）

$$P(\vec{x}) = \prod P(x_i|x_1 \cdots x_{i-1})$$

6.4.1 PixelRNN

6.4.2 ICA

Chapter 7

隐式概率分布

7.1 GAN

$$\begin{cases} \min_D J^{(D)}(D,G) \\ \min_G J^{(G)}(D,G) \end{cases}$$

7.1.1 传统GAN[29] [30] [31]

7.1.1.1

$$\begin{aligned} J^{(D)}(D,G) &= -E_{x\sim P_{\text{data}}}[\log D(x)] - E_{z\sim P_{\text{noise}}}[\log(1-D(G(z)))] \\ &= -E_{x\sim P_{\text{data}}}[\log D(x)] - E_{x\sim P_G}[\log(1-D(x))] \end{aligned}$$

则

$$D^*(x) = \frac{P_{\text{data}}(x)}{P_{\text{data}}(x)+P_G(x)}$$

7.1.1.2

Minimax，零和博弈

$$\begin{aligned} J^{(G)}(D,G) &= -J^{(D)}(D,G) \\ &= E_{x\sim P_{\text{data}}}[\log D(x)] + E_{z\sim P_{\text{noise}}}[\log(1-D(G(z)))] \\ &= E_{z\sim P_{\text{noise}}}[\log(1-D(G(z)))] \end{aligned}$$

则

$$J^{(G)}(D^*,G) = 2\text{JSD}_{\frac{1}{2},\frac{1}{2}}(P_{\text{data}}||P_G) - 2\log 2$$

则

$$P^*_{\text{model}}(x) = P_{\text{data}}(x)$$

缺点：当$P_G(x)$ 与$P_{\text{data}}(x)$ 的支撑集是高维空间中的低维流型时，二者几乎处处不重叠的概率为1。此时$J^{(G)}(D^*,G)$梯度为零，无法学习G。

7.1.1.3

启发式，非饱和

$$J^{(G)}(D,G) = -E_{z\sim P_{\text{noise}}}[\log D(G(z))]$$

则

$$J^{(G)}(D^*,G) = \text{KL}(P_G||P_{\text{data}}) - 2\text{JSD}_{\frac{1}{2},\frac{1}{2}}(P_{\text{data}}||P_G) + E_{x\sim P_G}[\log D^*(x)] + 2\log 2$$

缺点：

- 两距离符号相反，相互冲突。
- KL距离不对称，导致在两种错误中，更宁可生成$\begin{cases} P_G = 0 \\ P_{\text{data}} = 1 \end{cases}$而避免$\begin{cases} P_G = 1 \\ P_{\text{data}} = 0 \end{cases}$

7.1.1.4

极大似然

$$J^{(G)}(D,G) = -E_{z\sim P_{\text{noise}}}[\frac{1}{1-D(G(z))}]$$

7.1.1.5 DCGAN[32]

具体网络实现

7.1.1.6 CGAN[33]

额外条件信息y，$D(x)$、$G(z)$变为$D(x|y)$、$G(z|y)$，将y与x、y与z一同作为D、G的输入

7.1.1.7 InfoGAN[34]

[1]

将z分为可解释的c与不可解释的白噪声z'。
要尽量使c占z的重要性增大，即使$I(c;G(z',c))$ 尽量大。则可在$J^{(G)}(D,G)$ 中加入正则项$-\lambda I(c;G(z',c))$

7.1.2 GLS-GAN[35]

函数$C(a)$满足

$$\begin{cases} C(a) \geq a \\ C(a) = a \quad , \quad \forall a \geq 0 \end{cases}$$

则

$$\begin{cases} J^{(D)}(D,G) = E_{x\sim P_{\text{data}},z\sim P_{\text{noise}}}[C(-D(x)+D(G(z))+\Delta(x,G(z)))] \\ J^{(G)}(D,G) = -E_{z\sim P_{\text{noise}}}[D(G(z))] \end{cases}$$

[1] 互信息$I(X;Y) \overset{def}{=} H(X)+H(Y)-H(X,Y) = -\int p(x)\log p(x)dx - \int p(y)\log p(y)dy + \int p(x,y)\log p(x,y)dxdy$。$I(X;Y)$越大X、Y越相关；$I(X;Y)=0$时X、Y完全相互独立

证明 7.1 假设P_{data}支撑集为紧致集且*Lipschitz*连续[2]，$D(x)\le 0$，则到达纳什均衡时

$$\int |P_{data}(x)-P_{G^*}(x)|dx=0$$

证明：

因

$$\begin{array}{rcl} & J^{(G)}(D^*,G^*) & = & -E_{x\sim P_{G^*}}[D^*(x)] \\ \le & J^{(G)}(D^*,G\sim P_{data}) & = & -E_{x\sim P_{data}}[D^*(x)] \end{array}$$

则

$$\begin{aligned} J^{(D)}(D^*,G^*) &= E_{x\sim P_{data},z\sim P_{noise}}[C(-D^*(x)+D^*(G^*(z))+\Delta(x,G^*(z)))] \\ &\ge E_{x\sim P_{data},z\sim P_{noise}}[-D^*(x)+D^*(G^*(z))+\Delta(x,G^*(z))] \\ &= -E_{x\sim P_{data}}[D^*(x)]+E_{z\sim P_{noise}}[D^*(G^*(z))]+E_{x\sim P_{data},z\sim P_{noise}}[\Delta(x,G^*(z))] \\ &\ge E_{x\sim P_{data},z\sim P_{noise}}[\Delta(x,G^*(z))] \end{aligned}$$

若$||D||_L<1$，则

$$D(x)-D(G(z))\le \Delta(x,G(z))$$

则

$$C(-D(x)+D(G(z))+\Delta(x,G(z)))=-D(x)+D(G(z))+\Delta(x,G(z))$$

则

$$J^{(D)}(D,G)=-E_{x\sim P_{data}}[D(x)]+E_{z\sim P_{noise}}[D(G(z))]+E_{x\sim P_{data},z\sim P_{noise}}[\Delta(x,G(z))]$$

若$D(x)=\alpha\min\{0,P_{data}(x)-P_{G^*}(x)\}$，则

$$J^{(D)}(D,G^*)=-\int (P_{data}(x)-P_{G^*}(x))^2 I\{P_{data}(x)<P_{G^*}(x)\}dx+E_{x\sim P_{data},z\sim P_{noise}}[\Delta(x,G(z))]$$

若满足$P_{data}(x)<P_{G^*}(x)$ 的测度不为*0*，则

$$E_{x\sim P_{data},z\sim P_{noise}}[\Delta(x,G(z))]J^{(D)}(D^*,G^*)\le J^{(D)}(D,G^*)<E_{x\sim P_{data},z\sim P_{noise}}[\Delta(x,G(z))]$$

矛盾。故$P_{data}(x)=P_{G^*}(x)$ 几乎处处成立。

7.1.2.1 WGAN[36]

Wasserstein距离[3]

取$C(a)=a$，则

$$\begin{cases} J^{(D)}(D,G)=-E_{x\sim P_{\text{data}}}[D(x)]+E_{z\sim P_{\text{noise}}}[D(G(z))] \\ J^{(G)}(D,G)=-E_{z\sim P_{\text{noise}}}[D(G(z))] \end{cases}$$

---

[2] $||f||_L<K$（Lipschitz连续）：$\forall x_1,\forall x_2$，有$|f(x_1)-f(x_2)|\le K\Delta(x_1,x_2)$

[3] Wasserstein距离：

$$\begin{aligned} W(P_1||P_2) &\overset{def}{=} \inf_{P(x_1,x_2)\sim\prod[P_1(x_1),P_2(x_2)]} E_{(x_1,x_2)\sim P(x_1,x_2)}[||x_1-x_2||] \\ &= \frac{1}{K}\sup_{||f||_L<K}[E_{x\sim P_1(x)}[f(x)]-E_{x\sim P_2(x)}[f(x)]] \end{aligned}$$

其中

- $\prod[P_1(x_1),P_2(x_2)]$：边缘分布为$P_1(x_1),P_2(x_2)$的联合分布$P(x_1,x_2)$的集合
- $||f||_L<K$（Lipschitz连续）：$\forall x_1,\forall x_2$，有$|f(x_1)-f(x_2)|\le K\Delta(x_1,x_2)$

7.1.2.2 LS-GAN

要求

$$D(x) \geq D(G(z)) + \Delta(x, G(z))$$

松弛为

$$D(x) + \xi_{x,z} \geq D(G(z)) + \Delta(x, G(z))$$

则

$$\begin{cases} J^{(D)}(D,G) & = & -E_{x\sim P_{\text{data}}}[D(x)] + \lambda E_{x\sim P_{\text{data}}, z\sim P_{\text{noise}}}[\xi_{x,z}] \\ s.t. & & D(x) + \xi_{x,z} \geq D(G(z)) + \Delta(x, G(z)) \\ & & \xi_{x,z} \geq 0 \\ J^{(G)}(D,G) & = & -E_{z\sim P_{\text{noise}}}[D(G(z))] \end{cases}$$

等价于

$$\begin{cases} J^{(D)}(D,G) & = & -E_{x\sim P_{\text{data}}}[D(x)] + \lambda E_{x\sim P_{\text{data}}, z\sim P_{\text{noise}}}[\max\{0, -D(x) + D(G(z)) + \Delta(x, G(z))\}] \\ J^{(G)}(D,G) & = & -E_{z\sim P_{\text{noise}}}[D(G(z))] \end{cases}$$

等价于取$C(a) = \max\{0, a\}$

假设P_{data}支撑集为紧致集且Lipschitz连续[4]，$D(x) \leq 0$，则到达纳什均衡时

$$\int |P_{\text{data}}(x) - P_{G^*}(x)| dx \leq \frac{2}{\lambda}$$

[4] $||f||_L < K$（Lipschitz连续）：$\forall x_1, \forall x_2$，有$|f(x_1) - f(x_2)| \leq K\Delta(x_1, x_2)$

Part III

other

Chapter 8

广义线性模型GLM

$\vec{x}$ 以$p(\vec{y}|\vec{x})$ 概率映射到$\vec{y}$上。

已知或假设$\vec{y}$服从某种形式已知的分布$p(\vec{y}|\theta_1,\cdots,\theta_K)$。

将$\vec{x}$扩充为$\check{x}$，$\check{x}$中的每一分量为1、$\vec{x}$分量一次项、$\vec{x}$分量高次项等等。

将$\check{x}$投影到一组基$\{\vec{w}_1,\cdots,\vec{w}_K\}$ 上$\{\vec{w}_1\cdot\check{x},\cdots,\vec{w}_K\cdot\check{x}\}$上，即提取特征。

则只需找到一组满足$\{[-\infty,+\infty]\to$ 值域$[\theta_k]\}$的映射$\{\theta_k=f_k(\vec{w}_k\cdot\check{x})\}$。

训练样本集$\{(\vec{x}_\varsigma,\vec{y}_\varsigma)\}$，由$L=\prod_\varsigma p(\vec{y}_\varsigma|\vec{\theta}_\varsigma)$，求出$\vec{w}^*=\operatorname{argmax}_{\vec{w}}L$

8.1 指数族分布

为求映射$\{\theta_k=f_k(\vec{w}_k\cdot\check{x})\}$，若$p(\vec{y}|\theta_1,\cdots,\theta_K)$ 为指数族分布

$$C(\vec{\theta})\cdot H(\vec{y})\cdot e^{\sum_k Q_k(\theta_k)\cdot T_k(\vec{y})}=H(\vec{y})\cdot e^{\sum_k \eta_k\cdot T_k(\vec{y})-b(\vec{\eta})}$$

可取某个函数$h_k(\vec{w}_k\cdot\check{x})=E_y[T_k(\vec{y})]\left(=\frac{\partial b(\eta)}{\partial\eta_k}\right)$，代入得$f_k$

自然联系函数：若将h_k 形式取为$\frac{\partial b}{\partial\eta_k}$，则可得：$\vec{w}_k\cdot\check{x}=\eta_k=Q_k(\theta_k)$

8.1.1 多项式分布

8.1.1.1 多项式分布

$$\begin{array}{rcll}
p(y_1,\cdots,y_{K-1}|\theta_1,\cdots,\theta_{K-1}) & = & \frac{n!}{\prod_{k=1}^K y_k!}\prod_{k=1}^K\theta_k^{y_k} & (\sum_{k=1}^K\theta_k=1,\sum_{k=1}^K y_k=n)\\
 & = & \frac{n!}{\prod_{k=1}^K y_k!}\cdot e^{\sum_{k=1}^K\ln\theta_k\cdot y_k} & (\sum_{k=1}^K\theta_k=1,\sum_{k=1}^K y_k=n)\\
 & = & \theta_K^n\cdot\frac{n!}{\prod_{k=1}^K y_k!}\cdot e^{\sum_{k=1}^{K-1}\ln\frac{\theta_k}{\theta_K}\cdot y_k} & (\sum_{k=1}^K\theta_k=1)
\end{array}$$

得：

$$\theta_k=\begin{cases}\frac{e^{z_k}}{1+\sum_{j=1}^{K-1}e^{z_j}} & ,\quad k=1,\cdots,K-1\\ \frac{1}{1+\sum_{j=1}^{K-1}e^{z_j}} & ,\quad k=K\end{cases}$$

或：

$$\theta_k=\frac{e^{z_k}}{\sum_{j=1}^K e^{z_j}}\quad,\quad k=1,\cdots,K$$

8.1.1.2 多项式分布

（多项式分布特例：$n=1$）

$$\begin{array}{rcll} p(y_1,\cdots,y_{K-1}|\theta_1,\cdots,\theta_{K-1}) & = & \prod_{k=1}^{K}\theta_k^{y_k} & (\sum_{k=1}^{K}\theta_k=1,\sum_{k=1}^{K}y_k=1) \\ & = & e^{\sum_{k=1}^{K}\ln\theta_k\cdot y_k} & (\sum_{k=1}^{K}\theta_k=1,\sum_{k=1}^{K}y_k=1) \\ & = & \theta_K\cdot e^{\sum_{k=1}^{K-1}\ln\frac{\theta_k}{\theta_K}\cdot y_k} & (\sum_{k=1}^{K}\theta_k=1) \end{array}$$

得：

$$\theta_k=\begin{cases} \frac{e^{z_k}}{1+\sum_{j=1}^{K-1}e^{z_j}} & , \quad k=1,\cdots,K-1 \\ \frac{1}{1+\sum_{j=1}^{K-1}e^{z_j}} & , \quad k=K \end{cases}$$

或：

$$\theta_k=\frac{e^{z_k}}{\sum_{j=1}^{K}e^{z_j}} \quad , \quad k=1,\cdots,K$$

8.1.1.3 二项分布

（多项式分布特例：$K=2$）

$$\begin{array}{rcl} p(y|\theta) & = & \binom{n}{y}\theta^y(1-\theta)^{n-y} \\ & = & (1-\theta)^n\cdot\binom{n}{y}\cdot e^{\ln\frac{\theta}{1-\theta}\cdot y} \\ & & (y=0,\cdots,n) \end{array}$$

得：

$$\theta=\frac{1}{1+e^{-z}}$$

8.1.1.4 伯努利分布(logistic回归)

（多项式分布特例：$K=2$，$n=1$）

$$\begin{array}{rcl} p(y|\theta) & = & \theta^y(1-\theta)^{1-y} \\ & = & (1-\theta)\cdot e^{\ln\frac{\theta}{1-\theta}\cdot y} \\ & & (y=0,1) \end{array}$$

得：

$$\theta=\frac{1}{1+e^{-z}}$$

即：

$$\begin{cases} +\infty & \rightarrow & 1 \\ 0 & \rightarrow & \frac{1}{2} \\ -\infty & \rightarrow & 0 \end{cases}$$

则：

$$L=\prod_{\varsigma}\theta_{\varsigma}^{y_{\varsigma}}(1-\theta_{\varsigma})^{1-y_{\varsigma}}$$

为$\mathrm{argmax}_{\vec{w}}L$，令$\frac{\partial}{\partial w_k}\log L=0$，得$\sum_{\varsigma}(y_{\varsigma}-\theta_{\varsigma})x_{\varsigma k}=0$
由此求出$\vec{w}$

8.1.2 正态分布

8.1.2.1 二元正态分布

$$
\begin{aligned}
p(y|\mu,\sigma) &= \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(y-\mu)^2}{2\sigma^2}} \\
&= \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{\mu^2}{2\sigma^2}} \cdot e^{\frac{\mu}{\sigma^2}\cdot y-\frac{1}{2\sigma^2}\cdot y^2}
\end{aligned}
$$

8.1.2.2 线性最小二乘法

（二元正态分布特例）
$p(y|\mu)$ 服从高斯分布$\frac{1}{\sqrt{2\pi}}e^{-\frac{(y-\mu)^2}{2}}$，映射$z=\mu$
则$\mathrm{argmax}_{\vec{w}}L$ 有：$\mathrm{argmin}_{\vec{w}}||\vec{w}\cdot\check{x}-y||_2$
令$\check{X}=\begin{bmatrix} & \vdots & \\ \cdots & \check{x}_{\varsigma k} & \cdots \\ & \vdots & \end{bmatrix}$，$Y=\begin{bmatrix}\vdots\\ y_\varsigma\\ \vdots\end{bmatrix}$，$\vec{w}=\begin{bmatrix}\vdots\\ w_k\\ \vdots\end{bmatrix}$，则$\check{X}^T\check{X}\vec{w}=\check{X}^TY$

8.1.3 其他例子

泊松分布：

$$
\begin{aligned}
p(y|\lambda) &= \frac{\lambda^y}{y!}e^{-\lambda} \\
&= e^{-\lambda}\cdot\frac{1}{y!}\cdot e^{\ln\lambda\cdot y} \\
& (y=0,1,2,\cdots)
\end{aligned}
$$

几何分布：

$$
\begin{aligned}
p(y|\theta) &= (1-\theta)^{y-1}\theta \\
&= \frac{\theta}{1-\theta}\cdot e^{\ln(1-\theta)\cdot y} \\
& (y=0,1,2,\cdots)
\end{aligned}
$$

指数分布：

$$
\begin{aligned}
p(y|\lambda,\mu) &= \lambda e^{-\lambda(y-\mu)} \\
&= \lambda e^{\lambda\mu}\cdot e^{-\lambda\cdot y} \\
& (y>\mu)
\end{aligned}
$$

幂分布：

$$
\begin{aligned}
p(y|\theta) &= \theta y^{\theta-1} \\
&= \theta\cdot\frac{1}{y}\cdot e^{\theta\cdot\ln y} \\
& (0<y<1)
\end{aligned}
$$

β分布：

$$
\begin{aligned}
p(y|a,b) &= \frac{1}{\beta(a,b)}y^{a-1}(1-y)^{b-1} \\
&= \frac{1}{\beta(a,b)}\cdot\frac{1}{y(1-y)}\cdot e^{a\cdot\ln y+b\cdot\ln(1-y)} \\
& (0<y<1)
\end{aligned}
$$

Γ分布：

$$
\begin{aligned}
p(y|\alpha,\lambda) &= \frac{\lambda^\alpha}{\Gamma(\alpha)}y^{\alpha-1}e^{-\lambda y} \\
&= \frac{\lambda^\alpha}{\Gamma(\alpha)}\cdot e^{-\lambda\cdot y+(\alpha-1)\cdot\ln y} \\
& (y>0)
\end{aligned}
$$

8.2 SVM

K=1

训练样本$\{(\vec{x}_\varsigma, y_\varsigma)\}(y_\varsigma = \pm 1)$

分类函数$f(\vec{x}) \overset{def}{=} (\vec{w} \cdot \check{x} + b)$

几何间隔$\gamma_\varsigma \overset{def}{=} (\frac{\vec{w}}{||\vec{w}||} \cdot \check{x}_\varsigma + \frac{b}{||\vec{w}||})y_\varsigma$

最大化训练样本集中最小的几何间隔

$$\max_{\vec{w},b} \min_{\varsigma} (\frac{\vec{w}}{||\vec{w}||} \cdot \check{x}_\varsigma + \frac{b}{||\vec{w}||})y_\varsigma$$

等价于

$$\begin{aligned} &\max_{\vec{w},b} \tilde{\gamma} \\ \text{s.t.} \quad &(\tfrac{\vec{w}}{||\vec{w}||} \cdot \check{x}_\varsigma + \tfrac{b}{||\vec{w}||})y_\varsigma \geq \tilde{\gamma} \end{aligned}$$

等价于

$$\begin{aligned} &\max_{\vec{w},b} \tfrac{1}{||\vec{w}||} \\ \text{s.t.} \quad &(\vec{w} \cdot \check{x}_\varsigma + b)y_\varsigma \geq 1 \end{aligned}$$

等价于$(\min\limits_{\vec{w}} \frac{1}{2}||\vec{w}||^2)$

$$\begin{aligned} &L(\vec{w}, \vec{c}) = \tfrac{1}{2}||\vec{w}||^2 - \sum_{\varsigma} c_\varsigma((\vec{w} \cdot \check{x}_\varsigma + b)y_\varsigma - 1) \\ &\min_{\vec{w},b} \max_{\vec{c}} L(\vec{w}, \vec{c}) \\ \text{s.t.} \quad &c_\varsigma \geq 0 \end{aligned}$$

（为取$\max\limits_{\vec{c}}$：当$(\vec{w} \cdot \check{x}_\varsigma + b)y_\varsigma > 1$时，$c_\varsigma = 0$。即仅有支持向量起作用）

等价于

$$\begin{aligned} &L(\vec{w}, \vec{c}) = \tfrac{1}{2}||\vec{w}||^2 - \sum_{\varsigma} c_\varsigma((\vec{w} \cdot \check{x}_\varsigma + b)y_\varsigma - 1) \\ &\max_{\vec{c}} \min_{\vec{w},b} L(\vec{w}, \vec{c}) \\ \text{s.t.} \quad &c_\varsigma \geq 0 \end{aligned}$$

等价于(由$\frac{\partial}{\partial w_i} L(\vec{w}, \vec{c}) = 0$ 得：$\vec{w} = \sum\limits_{\varsigma} c_\varsigma y_\varsigma \check{x}_\varsigma$；由$\frac{\partial}{\partial b} L(\vec{w}, \vec{c}) = 0$ 得：$0 = \sum\limits_{\varsigma} c_\varsigma y_\varsigma$)

$$\begin{aligned} &L(\vec{c}) = \sum_{\varsigma} c_\varsigma - \tfrac{1}{2} \sum_{\varsigma_1} \sum_{\varsigma_2} c_{\varsigma_1} c_{\varsigma_2} y_{\varsigma_1} y_{\varsigma_2} \check{x}_{\varsigma_1} \cdot \check{x}_{\varsigma_2} \\ &\max_{\vec{c}} L(\vec{c}) \\ \text{s.t.} \quad &c_\varsigma \geq 0 \\ &\sum_{\varsigma} c_\varsigma y_\varsigma = 0 \end{aligned}$$

（$f(\vec{x}) = \sum\limits_{\varsigma} c_\varsigma y_\varsigma \check{x}_\varsigma \cdot \check{x} + b$）

8.2.0.1 核函数[37]

因$\check{x}$可能因高维导致维度灾难，故可用核函数替代$\check{x}_1 \cdot \check{x}_2$

如：$(\vec{x}_1 \cdot \vec{x}_2 + R)^d$、$e^{-\frac{||\vec{x}_1 - \vec{x}_2||^2}{2\sigma^2}}$

8.2.0.2 松弛变量

$$\begin{aligned}
&\max_{\vec{w},b,\vec{\xi}} \frac{1}{||\vec{w}||} - \lambda||\vec{\xi}|| \\
\text{s.t.}\quad &(\vec{w}\cdot\check{x}_\varsigma + b)y_\varsigma \geq 1-\xi_\varsigma \\
&\xi_\varsigma \geq 0
\end{aligned}$$

等价于

$$\begin{aligned}
&\min_{\vec{w}} \frac{1}{2}||\vec{w}||^2 + \lambda||\vec{\xi}|| \\
\text{s.t.}\quad &(\vec{w}\cdot\check{x}_\varsigma + b)y_\varsigma \geq 1-\xi_\varsigma \\
&\xi_\varsigma \geq 0
\end{aligned}$$

等价于

$$\begin{aligned}
&L(\vec{w},\vec{c}) = \frac{1}{2}||\vec{w}||^2 - \sum_\varsigma c_\varsigma((\vec{w}\cdot\check{x}_\varsigma + b)y_\varsigma - (1-\xi_\varsigma)) + \lambda||\vec{\xi}|| - \sum_\varsigma d_\varsigma\xi_\varsigma \\
&\min_{\vec{w},b,\vec{\xi}}\max_{\vec{c},\vec{d}} L(\vec{w},\vec{c}) \\
\text{s.t.}\quad &c_\varsigma \geq 0 \\
&d_\varsigma \geq 0
\end{aligned}$$

等价于

$$\begin{aligned}
&L(\vec{w},\vec{c}) = \frac{1}{2}||\vec{w}||^2 - \sum_\varsigma c_\varsigma((\vec{w}\cdot\check{x}_\varsigma + b)y_\varsigma - (1-\xi_\varsigma)) + \lambda||\vec{\xi}|| - \sum_\varsigma d_\varsigma\xi_\varsigma \\
&\max_{\vec{c},\vec{d}}\min_{\vec{w},b,\vec{\xi}} L(\vec{w},\vec{c}) \\
\text{s.t.}\quad &c_\varsigma \geq 0 \\
&d_\varsigma \geq 0
\end{aligned}$$

等价于

(由$\frac{\partial}{\partial w_i}L(\vec{w},\vec{c}) = 0$ 、 $\frac{\partial}{\partial b}L(\vec{w},\vec{c}) = 0$ 、 $\frac{\partial}{\partial \xi_\varsigma}L(\vec{w},\vec{c}) = 0$ 得：$\vec{w} = \sum_\varsigma c_\varsigma y_\varsigma \check{x}_\varsigma$ 、 $0 = \sum_\varsigma c_\varsigma y_\varsigma$ 、 $\lambda - c_\varsigma = d_\varsigma$)

$$\begin{aligned}
&L(\vec{c}) = \sum_\varsigma c_\varsigma - \frac{1}{2}\sum_{\varsigma_1}\sum_{\varsigma_2} c_{\varsigma_1}c_{\varsigma_2}y_{\varsigma_1}y_{\varsigma_2}\check{x}_{\varsigma_1}\cdot\check{x}_{\varsigma_2} \\
&\max_{\vec{c}} L(\vec{c}) \\
\text{s.t.}\quad &0 \leq c_\varsigma \leq \lambda \\
&\sum_\varsigma c_\varsigma y_\varsigma = 0
\end{aligned}$$

（为取$\max\limits_{\vec{c}}$： $\begin{matrix} \text{当}(\vec{w}\cdot\check{x}_\varsigma + b)y_\varsigma > 1\text{时，} c_\varsigma = 0 \\ \text{当}(\vec{w}\cdot\check{x}_\varsigma + b)y_\varsigma < 1\text{时，} c_\varsigma = C \end{matrix}$ 。即仅有支持向量和离群向量起作用）

（$f(\vec{x}) = \sum_\varsigma c_\varsigma y_\varsigma \check{x}_\varsigma \cdot \check{x} + b$）

Chapter 9

流型学习

分布在低维流型上的样本$\{\vec{y_\varsigma}\}$ 被光滑嵌入f嵌入到高维空间中，在观察到高维空间中样本$\{\vec{x_\varsigma}\}$的条件下重构f与$\{\vec{y_\varsigma}\}$

9.1 PCA（主成分分析）

无监督学习。为找出最能代表训练样本$\{\vec{x_\varsigma}\}$的方向，即求

$$\forall ij \quad \begin{array}{rl} & \max\limits_{||w_i||=1} Var[\vec{w}_i X] \\ s.t. & Cov[\vec{w}_i X, \vec{w}_j X] = 0 \end{array}$$

令$\breve{x}'_\varsigma \overset{def}{=} \breve{x}_\varsigma - E[\breve{x}]$
假设SVD分解得：

$$\begin{bmatrix} \cdots & \breve{x}'_\varsigma & \cdots \end{bmatrix} = \begin{bmatrix} \cdots & \vec{w}_k & \cdots \end{bmatrix} \begin{bmatrix} \ddots & & \\ & \lambda_k & \\ & & \ddots \end{bmatrix} V^T$$

则将$\breve{x}'$投影到各$\vec{w}_k$方向上：

$$\vec{z}'^T = \breve{x}'^T \begin{bmatrix} \cdots & \vec{w}_k & \cdots \end{bmatrix}$$

只取特征值λ_k最大的若干个$\vec{w}_k$（即只取$\begin{bmatrix} \cdots & \vec{w}_k & \cdots \end{bmatrix}$的左半段矩阵）作为特征方向进行投影，能最大限度分离各$\breve{x}$

9.1.1 法一

矩阵

$$\begin{bmatrix} \cdots & \breve{x}'_\varsigma & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ \breve{x}'^T_\varsigma \\ \vdots \end{bmatrix} = \begin{bmatrix} \cdots & \vec{w}_k & \cdots \end{bmatrix} \begin{bmatrix} \ddots & & \\ & \lambda_k^2 & \\ & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \vec{w}_k^T \\ \vdots \end{bmatrix}$$

特征值与特征向量为$\{(\vec{w}, \lambda_k^2)\}$

9.1.2 法二

将$\check{x}'_\varsigma$投影到各$\vec{w}_k$方向上：

$$\begin{bmatrix} \vdots \\ \vec{z}_\varsigma'^T \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \check{x}_\varsigma'^T \\ \vdots \end{bmatrix} \begin{bmatrix} \cdots & \vec{w}_k & \cdots \end{bmatrix} = V \begin{bmatrix} \ddots & & \\ & \lambda_k & \\ & & \ddots \end{bmatrix}$$

则

$$\begin{bmatrix} \cdots & \vec{w}_k & \cdots \end{bmatrix} = \begin{bmatrix} \cdots & \check{x}_\varsigma & \cdots \end{bmatrix} V \begin{bmatrix} \ddots & & \\ & \frac{1}{\lambda_k} & \\ & & \ddots \end{bmatrix} = \begin{bmatrix} \cdots & \check{x}'_\varsigma & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ \vec{z}_\varsigma'^T \\ \vdots \end{bmatrix} \begin{bmatrix} \ddots & & \\ & \frac{1}{\lambda_k^2} & \\ & & \ddots \end{bmatrix}$$

矩阵

$$\begin{bmatrix} \vdots \\ \check{x}_\varsigma'^T \\ \vdots \end{bmatrix} \begin{bmatrix} \cdots & \check{x}'_\eta & \cdots \end{bmatrix} = V \begin{bmatrix} \ddots & & \\ & \lambda_k^2 & \\ & & \ddots \end{bmatrix} V^T = \begin{bmatrix} \vdots \\ \vec{z}_\varsigma'^T \\ \vdots \end{bmatrix} \begin{bmatrix} \cdots & \vec{z}'_\eta & \cdots \end{bmatrix}$$

的特征向量与特征值为$\{(\frac{1}{\lambda_k} \begin{bmatrix} \vdots \\ z'_{\varsigma k} \\ \vdots \end{bmatrix}, \lambda_k^2)\}$

则

$$\vec{z}'^T = \check{x}'^T \begin{bmatrix} \cdots & \vec{w}_k & \cdots \end{bmatrix} = \check{x}'^T \begin{bmatrix} \cdots & \check{x}'_\varsigma & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ \vec{z}_\varsigma'^T \\ \vdots \end{bmatrix} \begin{bmatrix} \ddots & & \\ & \frac{1}{\lambda_k^2} & \\ & & \ddots \end{bmatrix}$$

9.1.3 Kernel PCA

当$\check{x}$太高维甚至无穷维时，无法显式求出$\vec{w}_k$
则可在“法二”中，替换

$$\begin{bmatrix} \vdots \\ \check{x}_\varsigma'^T \\ \vdots \end{bmatrix} \begin{bmatrix} \cdots & \check{x}'_\eta & \cdots \end{bmatrix} = \begin{bmatrix} & \vdots & \\ \cdots & K(\vec{x}'_\varsigma, \vec{x}'_\eta) & \cdots \\ & \vdots & \end{bmatrix}$$

$$\check{x}'^T \begin{bmatrix} \cdots & \check{x}'_\varsigma & \cdots \end{bmatrix} = \begin{bmatrix} & \vdots & \\ \cdots & K(\vec{x}', \vec{x}'_\varsigma) & \cdots \\ & \vdots & \end{bmatrix}$$

9.1.4 白化

PCA后可进行白化，对$\vec{z}_k$ 方差归一，即$\frac{\vec{z}_k}{\lambda_k}$

9.2 MDS（多维度尺度变换）

无监督学习。已知任意两点间距$\delta_{\varsigma\eta}$。重构向量$\vec{x}_\varsigma$，使得各点间距为$\delta_{\varsigma\eta}$

即求

$$\min_{\vec{x}_1,\cdot,\vec{x}_{|S|}} \sum_{\varsigma\eta}(||\vec{x}_\varsigma - \vec{x}_\eta|| - \delta\varsigma\eta)^2$$

定义内积$t_{\varsigma\eta} \overset{def}{=} (\vec{x}_\varsigma - E[\vec{x}]) \cdot (\vec{x}_\eta - E[\vec{x}])$，距离$d_{\varsigma\eta} \overset{def}{=} ||\vec{x}_\varsigma - \vec{x}_\eta||$，则内积

$$t_{\varsigma\eta} = -\frac{1}{2}\left(d_{\varsigma\eta}^2 - \frac{1}{|S|}\sum_\mu d_{\varsigma\mu}^2 - \frac{1}{|S|}\sum_\nu d_{\nu\eta}^2 + \frac{1}{|S|^2}\sum_{\mu\nu} d_{\mu\nu}^2\right)$$

可完全用距离$d_{\varsigma\eta}$表示

分解

$$\begin{bmatrix} \vdots \\ \vec{x}_\varsigma'^T \\ \vdots \end{bmatrix} \begin{bmatrix} \cdots & \vec{x}_\eta' & \cdots \end{bmatrix} = \begin{bmatrix} & \vdots & \\ \cdots & t_{\varsigma\eta} & \cdots \\ & \vdots & \end{bmatrix} = U^T\Lambda U = (\Lambda^{\frac{1}{2}}U)^T(\Lambda^{\frac{1}{2}}U)$$

取特征值最大的k'个$\vec{u}_i$分量作为$\vec{x}$

9.3 isomap

无监督学习。映射过程中尽可能保持全局流形上测地线的距离

未知流型结构，用有限数据采样估计测地线：

构造邻接图$W_{ij} = \begin{cases} 1 & , \quad ||\vec{x}_i - \vec{x}_j|| < \varepsilon \\ 0 & , \quad ||\vec{x}_i - \vec{x}_j|| \geq \varepsilon \end{cases}$，则任意两点间测地线$d_M$用邻接图上的最短路径长度近似

用MDS 计算映射后的坐标$\vec{y}$，使得映射坐标下的欧氏距离与原来的测地线距离尽量相等：

$$\min_{\vec{y}} \sum_{i,j}(||\vec{y}_i - \vec{y}_j|| - d_M(\vec{x}_i, \vec{x}_j))^2$$

9.4 LLE

无监督学习。由流型在局部等价于欧几里得空间，尽可能保持流型局部线性关系

对任意点$\vec{x}_\varsigma$，只考虑其周围的点$\vec{x}_\eta$（记为$\eta \sim \varsigma$）：

1. 将高维坐标间关系反映到权重w中：$\mathrm{argmin}_w \sum_\varsigma ||\vec{x}_\varsigma - \sum_{\eta\sim\varsigma} w_{\varsigma\eta}\vec{x}_\eta||^2$
2. 将权重w反映到低维坐标$\vec{y}$中：$\mathrm{argmin}_{\vec{y}} \sum_\varsigma ||\vec{y}_\varsigma - \sum_{\eta\sim\varsigma} w_{\varsigma\eta}\vec{y}_\eta||^2$

9.5 LDA（线性判别分析）

监督学习，分类。使得投影后类内方差最小，类间方差最大

训练样本集$\{(\tilde{x}_\varsigma, y_\varsigma)\}$，其中$y_\varsigma$属于有限的离散值（分类问题）

- 整体散度$S_T \overset{def}{=} \sum_{\check{x}_\varsigma \in D} (\check{x}_\varsigma - \bar{\check{x}})(\check{x}_\varsigma - \bar{\check{x}})^T$
- 类内散度$S_W \overset{def}{=} \sum_i \sum_{\check{x}_\varsigma \in D_i} (\check{x}_\varsigma - \bar{\check{x}}_i)(\check{x}_\varsigma - \bar{\check{x}}_i)^T$
- 类间散度$S_B \overset{def}{=} S_T - S_W = \sum_i |S|_i (\bar{\check{x}}_i - \bar{\check{x}})(\bar{\check{x}}_i - \bar{\check{x}})^T$

投影到一维$\vec{w}$上，$z_\varsigma = \vec{w}^T \check{x}_\varsigma$
则目标函数为投影后的

$$\max_{\vec{w}} \frac{\vec{w}^T S_B \vec{w}}{\vec{w}^T S_W \vec{w}}$$

等价于（因$\vec{w}$的模长不重要）

$$\begin{aligned} &\max_{\vec{w}} \vec{w}^T S_B \vec{w} \\ s.t. \quad &\vec{w}^T S_W \vec{w} = 1 \end{aligned}$$

Chapter 10

强化学习

10.1 强化学习

10.1.1 基本概念

状态s，动作a

学习策略：当前s下采取a的概率$\pi(s,a)$

系统反馈：当前s_1下采取a后变为s_2的概率$P(s_1 \xrightarrow{a} s_2)$

奖励$R_{s,a}$，衰减因子γ

状态-动作价值

$$\begin{aligned} q_\pi(s,a) &\overset{def}{=} E_\pi[\textstyle\sum_{k=0}^{\infty}\gamma^k R_{k+t}|s_t=s,a_t=a] \\ &= E_\pi[R_t+\gamma q_\pi(s_{t+1},a_{t+1})|s_t=s,a_t=a] \\ &= R_{s,a}+\gamma\textstyle\sum_{s'\in S}P(s\xrightarrow{a}s')\sum_{a'\in A}\pi(s',a')q(s',a') \\ &= R_{s,a}+\gamma\textstyle\sum_{s'\in S}P(s\xrightarrow{a}s')v(s') \end{aligned}$$

状态价值

$$\begin{aligned} v_\pi(s) &\overset{def}{=} E_\pi[\textstyle\sum_{k=0}^{\infty}\gamma^k R_{k+t}|s_t=s] \\ &= E_\pi[R_t+\gamma v_\pi(s_{t+1})|s_t=s] \\ &= \textstyle\sum_{a\in A}\pi(s,a)\left(R_{s,a}+\gamma\sum_{s'\in S}P(s\xrightarrow{a}s')v(s')\right) \\ &= \textstyle\sum_{a\in A}\pi(s,a)q(s,a) \end{aligned}$$

10.1.2 已知模型

已知$R_{s,a}$，$P(s_1 \xrightarrow{a} s_2)$ 下的学习

状态价值：

$$v(s)=\sum_{a\in A}\pi(s,a)[R_{s,a}+\gamma\sum_{s'\in S}P(s\xrightarrow{a}s')v(s')]$$

或

$$v(s)=\max_a[R_{s,a}+\gamma\sum_{s'\in S}P(s\xrightarrow{a}s')v(s')]$$

迭代$v(s)$至收敛

更新策略：

$$\pi(s,a)=\begin{cases}1-\varepsilon & ,\quad (a=\text{argmax}_a q(s,a))\\ \frac{\varepsilon}{|A-1|} & ,\quad (a\neq\text{argmax}_a q(s,a))\end{cases}$$

循环以上两步至收敛

10.1.3 未知模型

$R_{s,a}$，$P(s_1 \xrightarrow{a} s_2)$ 未知，仅由环境反馈s、R下的学习：

10.1.3.1 蒙特卡洛法

由当前π生成序列：$s_1,a_1,R_1,\cdots,s_k,a_k,R_k$

每个时刻t，

$$\begin{array}{rll}\text{每次抽样} & \begin{cases}V(s_t)+=\sum_{i=0}^{k-t}\gamma^i R_{t+i}\\ ++N(s_t)\end{cases} & \begin{cases}Q(s_t,a)+=\sum_{i=0}^{k-t}\gamma^i R_{t+i}\\ ++N(s_t,a_t)\end{cases}\\ \text{最终} & v(s_t)=\frac{V(s_t)}{N(s_t)} & q(s_t,a_t)=\frac{Q(s_t,a_t)}{N(s_t,a_t)}\end{array}$$

10.1.3.2 时差学习

TD(0)

$$\begin{array}{lclcl}v(s) & = & \lambda(r+\gamma v(s')) & + & (1-\lambda)v(s)\\ q(s,a) & = & \lambda(r+\gamma q(s',a')) & + & (1-\lambda)q(s,a)\\ q(s,a) & = & \lambda(r+\gamma max_{a'}q(s',a')) & + & (1-\lambda)q(s,a)\end{array}$$

10.2 DQN（深度强化学习）

强化学习中，状态s为天文数字，无法构建完整的表$q(s,a)$。故设法用函数$q(s,a,\theta)$拟合$q(s,a)$，用神经网络表示该函数

用Q-learning，逼近

$$q(s,a,\theta)=r+\gamma\max_{a'}q(s',a',\theta)$$

则损失函数

$$L(\theta)\overset{def}{=}E[(r+\gamma\max_{a'}q(s',a',\theta)-q(s,a,\theta))^2]$$

10.2.0.3 Experience Replay

按策略π 生成序列$s_1,a_1,R_1,\cdots,s_k,a_k,R_k$，从中随机抽取若干个进行训练（避免按连续选取会有相干性）。

重复以上若干遍至训练出正确网络$q(s,a,w)$用以拟合$q(s,a)$。

10.2.0.4 Target Q

新旧两个网络。用旧网络进行计算，参数更新至新网络上，延迟一段时间后再将新网络参数复制回旧网络。避免相关性太大。

$$L(\theta) \stackrel{def}{=} E[(r + \gamma \max_{a'} q(s', a', \theta_{\text{old}}) - q(s, a, \theta_{\text{new}}))^2]$$

10.2.0.5 Double DQN

$$L(\theta) \stackrel{def}{=} E[(r + \gamma q(s', \text{argmax}_{a'} q(s, a, \theta_{\text{new}}), \theta_{\text{old}}) - q(s, a, \theta_{\text{new}}))^2]$$

10.2.0.6 Prioritised replay

从Experience Replay中抽取(s,a)进行训练时，抽样概率与$|r + \gamma \max_{a'} q(s', a', \theta_{\text{old}}) - q(s, a, \theta_{\text{new}})|$ 成正比。

Chapter 11

决策树

回归树：每个节点都有预测值。最小化均方差，使分到节点中的数据与预测值方差最小

分类树：最大熵

11.1 单决策树

11.1.1 ID3

11.1.1.1 定义

对集合G，属性A将其分为子集G_a（不同G_a有不同A值）

信息熵

$$S(G,A) \overset{def}{=} -\sum_a \frac{|G_a|}{|G|} \log \frac{|G_a|}{|G|}$$

信息增益

$$Gain(G,A) \overset{def}{=} S(G,\text{正反例}) - \sum_a \frac{|G_a|}{|G|} S(G_a,\text{正反例})$$

11.1.1.2 算法

树各节点为样本集合

对每个节点选取信息增益最大的属性A，该节点中样本对A的不同值生成不同子节点。

持续分类直至每个节点正反取值一致，或用光所有属性。

11.1.2 C4.5

11.1.2.1 定义

信息增益率

$$GainR(G,A) \overset{def}{=} \frac{Gain(G,A)}{S(G,A)}$$

11.1.2.2 算法

树各节点为样本集合

对每个节点选取信息增益率最大的属性A，该节点中样本对A的不同值生成不同子节点。

持续分类直至每个节点正反取值一致，或用光所有属性。

11.1.3 最小二乘回归树

空间D划分为多个区域D_s，寻找划分方式S

$$\min_S \left\{ \sum_s \sum_{(x_\varsigma, y_\varsigma) \in D_s} (y_\varsigma - \bar{y}_s)^2 \right\}$$

其中区域D_s的输出值$\bar{y}_s = \frac{1}{|D_s|} \sum_{(x_\varsigma, y_\varsigma) \in D_s} y_\varsigma$

依次递归划分区域

11.1.4 Cart分类树

空间D中，属于第k类的空间$D_k = D \bigcap C_k$，则基尼系数

$$Gini(D) \overset{def}{=} \sum_k \frac{|D_k|}{|D|} \left(1 - \frac{|D_k|}{|D|}\right) = 1 - \sum_k \left(\frac{|D_k|}{|D|}\right)^2$$

空间D划分为多个区域D_s，寻找划分方式S

$$\min_S \left\{ \sum_s \frac{|D_s|}{|D|} Gini(D_s) \right\}$$

依次递归划分区域

11.2 Boosting

11.2.1 随机森林

对每棵树，从A个总训练样本中有放回抽取a个作为其训练样本（可取a=A）。

对每个结点，从F个维度属性中不放回抽取f个作为其判断属性，从f个判断属性中找出最佳属性进行划分。

预测时用所有树共同决定分类。

11.2.2 AdaBoost

11.2.2.1 原理

多个弱分类器共同决定分类。

分类错误的训练样本权重加大，分类正确的训练样本权重减小。

训练完毕后，误差率大的弱分类器投票权重较小，误差率小的弱分类器投票权重较大。

11.2.2.2 具体算法

第t轮训练样本$(x_\varsigma, y_\varsigma)$ 的权重为$w_{t,\varsigma}$，构建弱分类器$f_t(x)$ 使分类误差率

$$\varepsilon_t = \sum_\varsigma w_{t,\varsigma} I(f_t(x_\varsigma) \neq y_\varsigma)$$

最小。

分类器f_t的重要程度

$$c_t = \frac{1}{2} \log \frac{1-\varepsilon_t}{\varepsilon_t}$$

更新

$$w_{t+1,\varsigma} = \begin{cases} \frac{1}{Z_t} w_{t,\varsigma} e^{-c_t} & , \quad y_\varsigma = f_t(x_\varsigma) \\ \frac{1}{Z_t} w_{t,\varsigma} e^{+c_t} & , \quad y_\varsigma \neq f_t(x_\varsigma) \end{cases}$$

最终强分类器$F(x) = \sum_t c_t f_t(x)$

11.2.3 GBDT

回归树

训练样本在第i棵树的输入值= 训练样本在第i-1棵树的输入值- 训练样本被第i-1棵树分类的预测值，即每一棵树学的是之前所有树结论和的残差

预测时依次经过所有树

Chapter 12

NLP

12.1 隐含语义分析

12.1.1 PLSA

第m篇文档属于第k个主题的概率为θ_{mk}，词v属于第k个主题的概率为ϕ_{vk}。则每个词的生成概率为

$$P(v|m) = \sum_k \phi_{vk}\theta_{mk}$$

第m篇文章的生成概率为

$$P(\vec{w}|m) = \prod_v \sum_k \phi_{vk}\theta_{mk}$$

12.1.2 LDA

12.1.2.1 Dirichelet分布与多项分布

$$\Gamma(x) = \int\limits_0^{+\infty} t^{x-1}e^{-t}dt$$

Dirichelet分布：

$$Dir(\vec{p}|\vec{\alpha}) = \frac{\Gamma(\sum\limits_k \alpha_k)}{\prod\limits_k \Gamma(\alpha_k)} \prod_k p_k^{\alpha_k - 1} \quad (\sum_k p_k = 1)$$

多项分布：

$$Mult(\vec{n}|\vec{p}) = \frac{(\sum\limits_k n_k)!}{\prod\limits_k (n_k!)} \prod_k p_k^{n_k} = \frac{\Gamma(\sum\limits_k n_k + 1)}{\prod\limits_k \Gamma(n_k + 1)} \prod_k p_k^{n_k}$$

则

$$\begin{cases} Dir(\vec{p}|\vec{\alpha}+\vec{n}) & = & Mult(\vec{n}|\vec{p}) & Dir(\vec{p}|\vec{\alpha}) \\ \text{后验分布} & = & \text{似然函数} & \text{先验分布} \\ P(\theta|x) & = & P(x|\theta) & P(\theta) \end{cases}$$

12.1.2.2 单主题

词袋模型，只与词频有关。

对每篇文章的词频，其概率为$Mult(\vec{n}|\vec{p})$。则可假设先验分布为$Dir(\vec{p}|\vec{\alpha})$。

因后验分布为

$$Dir(\vec{p}|\vec{\alpha}+\vec{n}) = \frac{1}{Z(\vec{\alpha}+\vec{n})}\prod_v p_v^{\alpha_v+n_v-1}$$

由训练样本词频n_v可估计得

$$\hat{p}_v = E_{Dir(\vec{p}|\vec{\alpha}+\vec{n})}[p_v] = \frac{\alpha_v+n_v}{\sum_{v'}(\alpha'_v+n'_v)}$$

预测文章概率为

$$P(\vec{n}|\vec{\alpha}) = \int P(\vec{n}|\vec{p})P(\vec{p}|\vec{\alpha})d\vec{p} = \frac{Z(\vec{\alpha}+\vec{n})}{Z(\vec{\alpha})}$$

12.1.2.3 多主题

第m篇文章的第n个单词w_{mn}属于第k个主题$(z_{mn}=k)$的概率为θ_k^m，第k个主题出现该词$(w_{mn}=v)$的概率为ϕ_v^k。

$$\begin{array}{ccccc}
\text{Doc} & \rightarrow & \text{Topic} & \rightarrow & \text{Word} \\
\text{m} & [\theta_k^m] & \text{k} & [\phi_v^k] & \text{v} \\
 & \uparrow & & \uparrow & \\
 & [\alpha_k] & & [\beta_v] &
\end{array}$$

第m篇文章，在主题分布$[\theta_k^m]$下，各主题词数$[n_k^m]$概率为$Mult([n_k^m]|[\theta_k^m])$。则可假设$[\theta_k^m]$先验分布为$Dir([\theta_k^m]|[\alpha_k])$，其后验分布为

$$Dir([\theta_k^m]|[\alpha_k]+[n_k^m]) = \frac{1}{Z([\alpha_k]+[n_k^m])}\prod_k(\theta_k^m)^{\alpha_k+n_k^m-1}$$

则

$$\begin{aligned}
P([n_k^m]|[\alpha_k]) &= \int P([n_k^m]|[\theta_k^m])\ P([\theta_k^m]|[\alpha_k])\ d[\theta_k^m] \\
&= \tfrac{Z([\alpha_k]+[n_k^m])}{Z([\alpha_k])}
\end{aligned}$$

对所有文章所有词，第k个主题，在词频分布$[\phi_v^k]$下，各词数$[n_v^k]$概率为$Mult([n_v^k]|[\phi_v^k])$。则可假设$[\phi_v^k]$先验分布为$Dir([\phi_v^k]|[\beta_v])$，其后验分布为

$$Dir([\phi_v^k]|[\beta_v]+[n_v^k]) = \frac{1}{Z([\beta_v]+[n_v^k])}\prod_v(\phi_v^k)^{\beta_v+n_v^k-1}$$

则

$$\begin{aligned}
P([n_v^k]|[\beta_v]) &= \int P([n_v^k]|[\phi_v^k])\ P([\phi_v^k]|[\beta_v])\ d[\phi_v^k] \\
&= \tfrac{Z([\beta_v]+[n_v^k])}{Z([\beta_v])}
\end{aligned}$$

12.1.3 LFM(Latent factor model)

已知$\begin{bmatrix} & \vdots & \\ \cdots & R_{ik} & \cdots \\ & \vdots & \end{bmatrix}$，找出隐含主题分类j

$$\begin{bmatrix} & \vdots & \\ \cdots & R_{ik} & \cdots \\ & \vdots & \end{bmatrix} = \begin{bmatrix} & \vdots & \\ \cdots & P_{ij} & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} & \vdots & \\ \cdots & Q_{jk} & \cdots \\ & \vdots & \end{bmatrix}$$

$$\min_{PQ} \sum_{ik} (R_{ik} - \sum_j P_{ij} Q_{jk})^2 + \lambda_P ||P|| + \lambda_Q ||Q||$$

$$\begin{cases} P_{i'j'} += \eta_P \left(\left(\left(\begin{bmatrix} \cdots & R_{i'k} & \cdots \end{bmatrix} - \begin{bmatrix} \cdots & P_{i'j} & \cdots \end{bmatrix} \begin{bmatrix} & \vdots & \\ \cdots & Q_{jk} & \cdots \\ & \vdots & \end{bmatrix} \right) \right) \begin{bmatrix} \vdots \\ Q^T_{kj'} \\ \vdots \end{bmatrix} - \lambda_P P_{i'j'} \right) \\ Q_{j'k'} += \eta_Q \left(\begin{bmatrix} \cdots & P^T_{j'i} & \cdots \end{bmatrix} \left(\begin{bmatrix} \vdots \\ R_{ik'} \\ \vdots \end{bmatrix} - \begin{bmatrix} & \vdots & \\ \cdots & P_{ij} & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ Q_{jk'} \\ \vdots \end{bmatrix} \right) - \lambda_Q Q_{j'k'} \right) \end{cases}$$

12.2 统计语言模型

一段语句的概率

$$p(w_1, \cdots, w_T) = \prod_{t=1}^{T} p(w_t | w_1, \cdots, w_T)$$

或

$$p(w_1, \cdots, w_T) = \prod_{t=1}^{T} p(w_1, \cdots, w_T | w_t)$$

12.2.1 N-gram

假设每个词出现概率仅与它之前的n-1个词有关

$$p(w_t | w_1, \cdots, w_T) \simeq p(w_t | w_{t-n+1}, \cdots, w_{t-1})$$

12.2.2 CBOW

假设每个词出现概率仅与它前后的2n个词有关

$$p(w_t | w_1, \cdots, w_T) \simeq p(w_t | w_{t-n}, \cdots, w_{t+n})$$

12.2.3 Skip-Gram

假设每个词出现概率仅与它前后的2n个词有关

$$p(w_1, \cdots, w_T|w_t) \simeq p(w_t|w_{t-n}, \cdots, w_{t+n})$$

12.2.4 隔词

以上各种模型可不限于紧邻前后，跳过一些词的情况亦可，用于扩展词组和提取远距离信息。可对远距离的词组乘以衰减系数

12.3 词向量

将每个词或者连续几个词表示为坐标空间中的一个点

12.3.1 One-hot Representation

每个词表示为一个向量$(0, \cdots, 0, 1, 0, \cdots, 0)$，向量长度为字典大小。
实践中用Hash表给每个词分配一个编号

12.3.2 Distributed Representation

linear bag-of-words contexts
每个词w表示为一个低维实数向量$\vec{w}$ （亦可身为中心词向量$\vec{w}'$与身为周围词向量$\vec{w}''$不同）

12.3.2.1 Softmax

用周围词表示中心词，最大化给定中心词时周围词概率

$$\begin{aligned} & \operatorname{argmax} \prod_{t=1}^{T} \prod_{-n\leq j\leq n, j\neq 0} P(w_{t+j}|w_t) \\ = & \operatorname{argmax} \frac{1}{T} \sum_{t=1}^{T} \sum_{-n\leq j\leq n, j\neq 0} \log P(w_{t+j}|w_t) \end{aligned}$$

其中

$$P(w_O|w_I) = \frac{e^{\vec{w}_O^T \cdot \vec{w}_I}}{\sum\limits_{\vec{w}\in W} e^{\vec{w}^T \cdot \vec{w}_I}}$$

将$\vec{w}$作为输入，经神经网络输出L。同时训练神经元参数与$\vec{w}$的取值

12.3.2.2 Softmax

语料库D中出现的词对(w_1, w_2)出现概率较大

$$\begin{aligned} & \operatorname{argmax} \prod_{(w_1,w_2)\in D} P(w_1, w_2 \in D) \\ = & \operatorname{argmax} \sum_{w_1\in W_1} \sum_{w_2\in W_2} P(w_1, w_2) \log \frac{e^{\vec{w}_1^T \vec{w}_2}}{\sum\limits_{w_2'\in W_2} e^{\vec{w}_1^T \vec{w}_2'}} \end{aligned}$$

其中

$$P((w_1,w_2)\in D)=\frac{e^{\vec{w}_1^T\vec{w}_2}}{\sum\limits_{w_2'\in W_2}e^{\vec{w}_1^T\vec{w}_2'}}$$

12.3.2.3 Softmax的矩阵分解形式

令$\frac{\partial L}{\partial \vec{w}_1^T\vec{w}_2}=0$，解得

$$\begin{aligned}\vec{w}_1^T\vec{w}_2 &= \log\frac{P(w_1,w_2)}{P(w_1)}+\log\sum_{w_2'\in W_2}e^{\vec{w}_1^T\vec{w}_2'}\\ &= \log P(w_2|w_1)+\log\sum_{w_2'\in W_2}e^{\vec{w}_1^T\vec{w}_2'}\end{aligned}$$

12.3.2.4 Negative-Sampling[38] [39]

语料库D中出现的词对(w_1,w_2)出现概率较大，随机产生$\tilde{D}$的词对(w_1,w_2)出现概率较小

$$\begin{aligned}&\operatorname{argmax}\prod_{(w_1,w_2)\in D}P((w_1,w_2)\in D)\prod_{(w_1,w_2)\in\tilde{D}}(1-P((w_1,w_2)\in D))\\ =\;&\operatorname{argmax}\left[\sum_{w_1\in W_1}\sum_{w_2\in W_2}P(w_1,w_2)\log\sigma(\vec{w}_1^T\vec{w}_2)+\lambda\sum_{w_1\in W_1}\sum_{w_2\in W_2}P(w_1)P(w_2)\log\sigma(-\vec{w}_1^T\vec{w}_2)\right]\end{aligned}$$

其中

$$P((w_1,w_2)\in D)=\sigma(\vec{w}_1^T\vec{w}_2)=\frac{1}{1+e^{-\vec{w}_1^T\cdot\vec{w}_2}}$$

12.3.2.5 Negative-Sampling的矩阵分解形式[40]

令$\frac{\partial L}{\partial \vec{w}_1^T\vec{w}_2}=0$，解得

$$\vec{w}_1^T\vec{w}_2=\log\frac{P(w_1,w_2)}{P(w_1)P(w_2)}-\log\lambda$$

12.4 NMT（Neural Machine Translation）

12.4.1 RNN

输入 x_t
特征 $h_t=H(x_t,h_{t-1})$
输出 $y_t=Y(h_t)$

12.4.2 seq2seq

编码输入 x_t
编码特征 $h_t=H(x_t,h_{t-1})$
译码特征 $h'_{t'}=H'(x'_{t'},h'_{t'-1})$
译码输出 $y'_{t'}=Y'(h'_{t'})$
编码译码连接 $\begin{cases}h'_0=h_T\\ x'_{t'}=0\end{cases}$ 或$\{x'_{t'}=h_T$

12.4.3 attention

译码输入 $x'_{t'} = X'_{t'}(\{h_t\}, \{a_{tt'}\})$

注意力 $a_{tt'} = A(h_t, h'_{t'-1})$

Chapter 13

图结构学习

13.1 结点表示

计算出结点w的特征向量$\vec{w}$，用以表征w及周围结点的平均场

13.1.1 GNN[41]

反复迭代$\vec{w}^{(t)}$，直至收敛

$$\vec{w}^{(t)} = f(\vec{c}^{(t-1)}) \quad (\forall c\text{在}w\text{近邻})$$

13.1.2 DeepWalk[42]

节点i到j的权重$\tilde{a}_{ij}$，归一化得转移概率$a_{ij} = \frac{\tilde{a}_{ij}}{\sum\limits_{j'} \tilde{a}_{ij'}}$。即转移概率矩阵$A = [a_{ij}]$，k步转移矩阵$A^k$。

随机游走出序列，将该序列视为NLP中的语句，序列中的词对即为$(w_1, w_2) \in D$。

当序列长度为K时，$[P(w_2|w_1)] = \sum\limits_{k=1}^{K} A^k$

13.1.3 GraRep[43]

节点i到j的权重$\tilde{a}_{ij}$，归一化得转移概率$a_{ij} = \frac{\tilde{a}_{ij}}{\sum\limits_{j'} \tilde{a}_{ij'}}$。即转移概率矩阵$A = [a_{ij}]$，k步转移矩阵$A^k$。

$$\begin{aligned}
L_k &\overset{def}{=} \sum_{w,c} L_k(w,c) \\
L_k(w,c) &\overset{def}{=} P_k(c|w) \log \sigma(\vec{w}^T \vec{c}) + \lambda P_k(c) \log \sigma(-\vec{w}^T \vec{c}) \\
&= A^k_{wc} \log \sigma(\vec{w}^T \vec{c}) + \frac{\lambda}{|V|} \sum_{w'} A^k_{w'c} \log \sigma(-\vec{w}^T \vec{c}) \\
&\quad \left(P_k(c) = \sum_{w'} P_k(c|w') P_0(w'), \text{假定} P_0(w') = \frac{1}{|V|} \right)
\end{aligned}$$

令$\frac{\partial L_k}{\partial \vec{w}^T \vec{c}} = 0$，得：

$$\vec{w}^T \vec{c} = \log \frac{A^k_{wc}}{\sum\limits_{w'} A^k_{w'c}} - \log \frac{\lambda}{|V|} \overset{def}{=} y_{wc}$$

对$[y_{wc}]$做SVD分解，则筛选前d个即为$\vec{w}^T$与$\vec{c}$的d维特征向量

13.1.4 类比CNN[44]

按某种顺序对节点排序 $\Longleftrightarrow$ 图片隐含上下左右的顺序

每个节点周围最近的节点作为卷积核输入 $\Longleftrightarrow$ 中心像素周围的像素作为卷积核输入

按排序从中挑出固定数目中心节点 $\Longleftrightarrow$ 不同分辨率的图片，缩放成固定大小输入

13.2 图表示

13.2.1

$$\vec{z}_G = \sum_{i \in G} \vec{z}_i$$

13.2.2 Graph Kernel[45]

图$G^1 = (V^1, E^1)$与图$G^2 = (V^2, E^2)$的直积$G_\times$定义为：$V^\times \overset{def}{=} \{(v_i^1 v_r^2) : v_i^1 \in V_1, v_r^2 \in V_2\}$，$E^\times \overset{def}{=} \{(v_i^1 v_r^2) \to (v_1^2 v_r^2) : v_i^1 \to v_j^1 \in E^1, v_r^2 \to v_s^2 \in E^2\}$

对邻接矩阵$\tilde{A}$和概率转移矩阵A都有矩阵直积：$\tilde{A}^\times = \tilde{A}^1 \otimes \tilde{A}^2$、$A^\times = A^1 \otimes A^2$

Bibliography

[1] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, and Gang Wang. Recent advances in convolutional neural networks. *arXiv preprint arXiv:1512.07108*, 2015.

[2] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[5] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015.

[6] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016.

[7] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[8] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *european conference on computer vision*, pages 346–361. Springer, 2014.

[10] Rohollah Soltani and Hui Jiang. Higher order recurrent neural networks. *arXiv preprint arXiv:1605.00064*, 2016.

[11] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[12] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.

[13] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[14] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385, 2015.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[16] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.

[17] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In *Advances in Neural Information Processing Systems*, pages 4470–4478, 2017.

[18] Yuxin Wu and Kaiming He. Group normalization. *arXiv preprint arXiv:1803.08494*, 2018.

[19] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017.

[20] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. 2018.

[21] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[22] Ross Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[24] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[25] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017.

[26] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[27] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[28] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[29] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[30] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[31] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *NIPS 2016 Workshop on Adversarial Training. In review for ICLR*, volume 2016, 2017.

[32] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[33] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[34] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.

[35] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, and Zhen Wang. Least squares generative adversarial networks. *arXiv preprint ArXiv:1611.04076*, 2016.

[36] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[37] Hal Daumé III. From zero to reproducing kernel hilbert spaces in twelve pages or less, 2004.

[38] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[39] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[40] O. Levy and Y. Goldberg. Neural word embedding as implicit matrix factorization. *Advances in Neural Information Processing Systems*, 3:2177–2185, 2014.

[41] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[42] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[43] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 891–900. ACM, 2015.

[44] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International Conference on Machine Learning*, pages 2014–2023, 2016.

[45] S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11(Apr):1201–1242, 2010.

[46] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, 2013.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[48] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[49] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Swish: a self-gated activation function. *arXiv preprint arXiv:1710.05941*, 2017.