1. 机器学习分类

   有监督学习：有明确的label，图像识别、垃圾邮件识别、股票预测

   无监督：聚类、降维

   半监督学习：有一部分数据有标签 文本分类、医学影像分析

   弱监督学习：不好的/笼统的标签 视频事件检测、社交媒体情感分析

2. 线性模型

   最小二乘解的**推导** $\hat{\beta} = (A^T A)^{-1} A^T Y$

   不可逆：提前给分布作为先验信息，利用最大后验估计。原因：信息不够

   先验：岭回归、脊回归

   计算：梯度下降：找到梯度并进行学习率的调整。学习率过大可能会跳过点，过小收敛速度过慢

   回归：sigmoid和softmax（多分类），损失函数用**交叉熵，形式记一下**

   线性回归问题：异或问题。（两层已被证明可以）

3. 分类器

   逻辑回归

   K-**近邻分类器具体执行过程**：选m个种子

4. 神经网络基础：激活函数（对常见的知道有哪些）：为输入提供非线性性、多层感知机：相比于单层有更好的表达能力、**损失函数**：交叉熵、均方误差、平均绝对误差、Lp范数，知道是什么、优化：自适应学习率与**反向传播**，**正则化：**validation阶段和test阶段设置应该一样

5. 卷积神经网络 **计算**

   空间上的权重共享，稀疏链接：上一层并不连接到下一层全部神经元。池化：max mean 空间金字塔。**反卷积**卷积计算，**对异常值outlier敏感**?

   空洞卷积，反卷积

6. 卷积输出尺寸$o = \lfloor \frac{i+2p-k}{s} \rfloor + 1$，i输入尺寸（input），p填充（padding，向外填充，因此总尺寸是i+2p），k卷积核尺寸（kernel size，需要扣除），s步长（stride）

7. **卷积神经网络感受野**：当前层的感受野

   $RF_{i+1} = RF_i + (k_{i+1} - 1) \times S_i$,其中$RF_0 = 1, S_i = \prod_{j=0}^{i} s_j, s_0 = 1$

   对于包含空洞卷积的网络，将对应层juan的卷积核尺寸k替换成等效尺寸k'即可。

8. 神经网络应用：

   1. 常见的神经网络

   - AlexNet：引入ReLU，dropout，数据增强
   - VGG：小尺寸卷积核
   - ResNet：残差神经网络

   2. 效果评估：精确率=TP/（TP+FP）判断是对的里面有多少对的

   召回率=TP/（TP+FN）所有事实正例里判断为正的比例

   true false：是否预测正确

   positive negative：预测值是正还是负

   IoU: intersection over union=$\frac{area-of-overlap}{area-of-union}$

9. 算法流程，目标检测算法

   R-CNN 框一些框进行初步分类，再每个框CNN

   SPP-net：先提前做一次CNN+SPP

   Fast R-CNN SPP-net基础上pooling，尺寸保持一致

   Faster R-CNN 用RPN替代selective search

   YOLO：目标检测问题变成一个回归问题

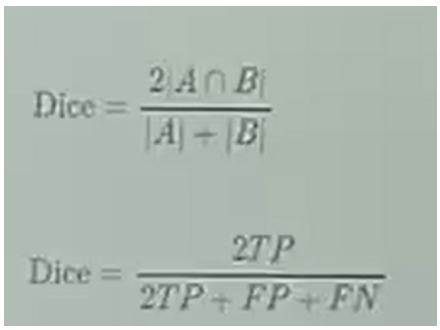
10. 图像分割算法：

    语义分割

    实例分隔（进一步区分同一类别中的不同实例）

11. 图像分割算法：输出与原始图片大小尺寸相同的图片：使用反卷积/上采样。FCN SegNet PSPNet，跳跃连接

    dice系数（背一下）



$$\text{Dice} = \frac{2|A \cap B|}{|A| + |B|}$$

$$\text{Dice} = \frac{2TP}{2TP + FP + FN}$$

    实例分隔：Mask R-CNN ROI Align：更精细

12. 人脸识别：

    人脸识别（多分类），人脸验证（二分类）

    close-set（训练好后不再接受新数据，被识别的人脸在训练集中，分类）/open-set（不在，特征提取） 闭集开集

13. 姿态估计：自顶向下：先检测每个人 的位置和边界框

    自底向上：先检测底

---

arthals 启动：

机器学习

1. 机器学习：模型的**表现**得到提升，在某些**任务**上，基于**经验** （三要素）
2. 分类：
   1. 有监督学习：提供标签
      - 分类：预测离散值，逻辑回归（二分类）
      - 回归：连续，线性回归
   2. 无监督学习：
      - 聚类：数据分组。K-均值聚类，
      - 密度估计：估计数据概率分布，用于异常检测。（可有参可午餐）
      - 降维：减少数据复杂性，保留重要特征，可用于数据可视化。主成分分析，自编码器，词嵌入（将单词映射到低维空间，用于自然语言处理）
   3. 半监督学习：标记了部分数据

4. 弱监督学习：标记数据不全/不可靠/模糊
5. 强化学习：与环境交互

3. 损失函数
    1. 0/1损失函数 二元分类
    2. 平方损失函数 回归
    3. 负对数似然（$loss(f(X)) = -\log(P_f(X))$） 密度估计

机器学习：找到预测规则f最小化损失函数的期望值

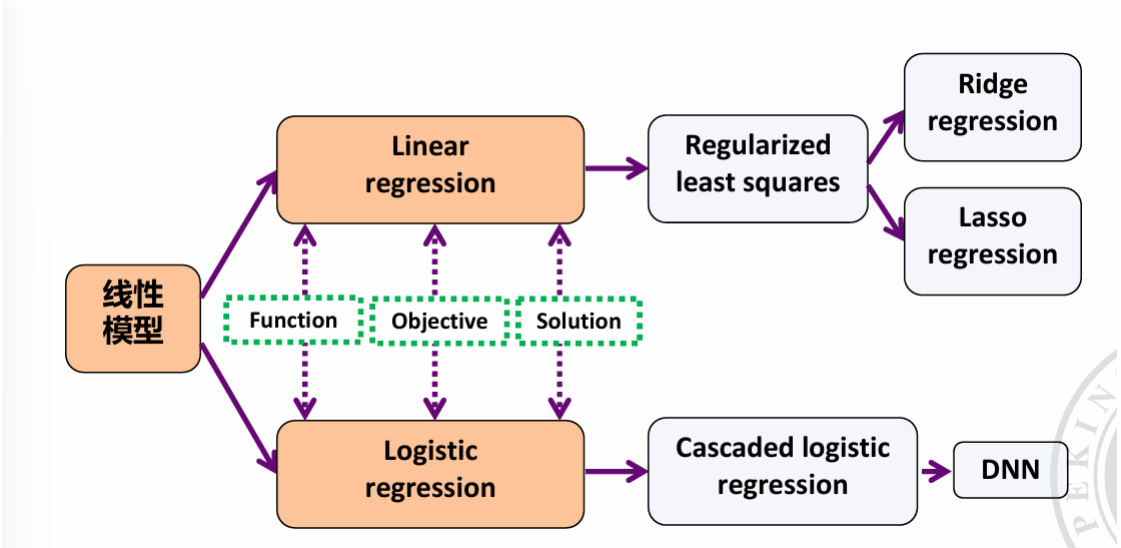
4. 有参数学习：函数形式已知，参数数量固定。学习完成后预测新数据时不再需要原始训练数据

- **线性回归**：$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$，其中 $\beta_0, \beta_1, \ldots, \beta_n$ 是需要学习的参数。
- **逻辑回归**：$y = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n)}}$，其中 $\beta_0, \beta_1, \ldots, \beta_n$ 是需要学习的参数。
- **词嵌入**：将单词映射到低维空间，通过学习词向量的参数。
- **卷积神经网络**：通过学习卷积核的参数，提取图像特征。

- **循环神经网络**：通过学习循环层的参数，提取序列数据的特征。
- **深度神经网络**：通过学习多层网络的参数，提取复杂的特征。

无参数学习：最近邻分类器。不对函数形式做出严格假设，参数数量可能随着数据量的增加而增加。

线性模型

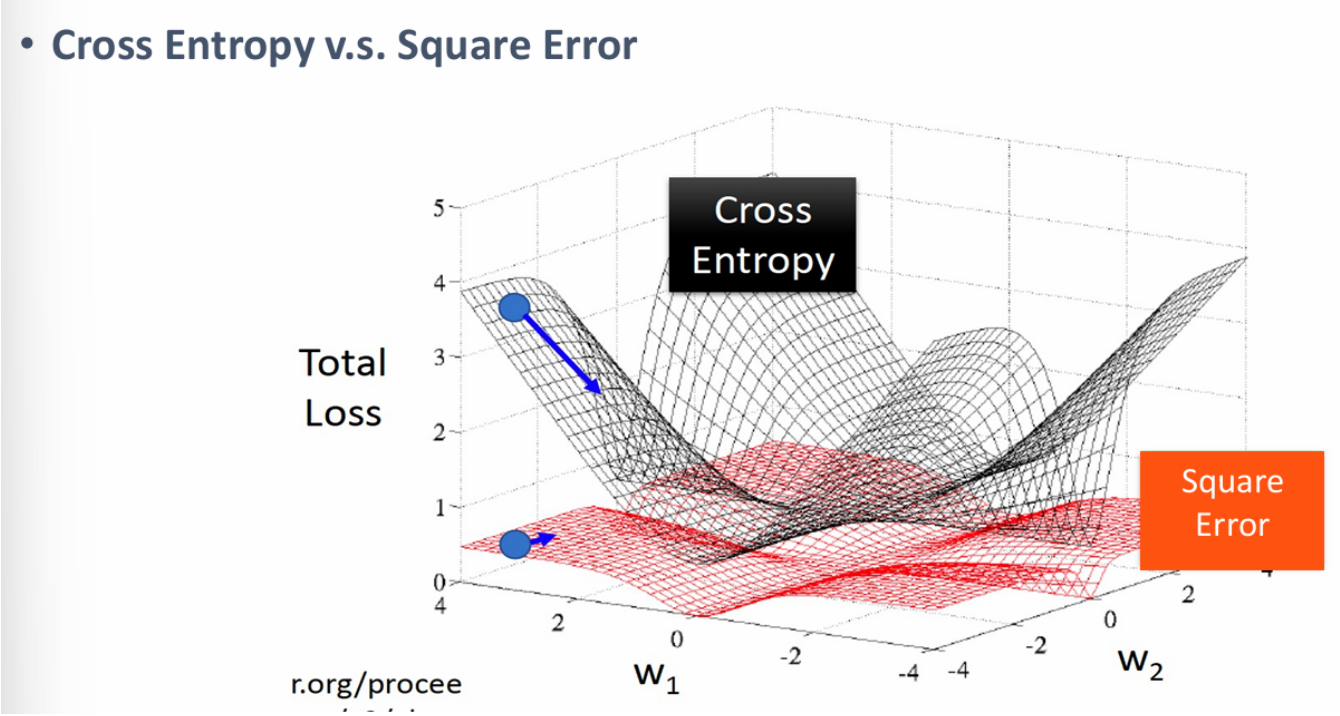
1. 最小二乘估计 $\hat{\beta} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{Y}$
2. A不可逆（样本数量 < 特征维度）--梯度下降法，学习率
3. ridge regression L2范数，lasso regression L1范数，先验分布的选择
4. 

正则化最小二乘法：减少过拟合；级联逻辑回归：解决异或问题
5. 贝叶斯统计：最大化数据的似然函数和参数的先验概率的乘积

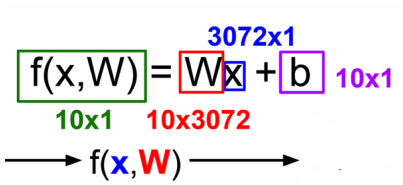
6. 交叉熵：度量两个概率分布之间的差异性信息，负对数，极小值点是f(x)=y

Cross entropy:

$$C(f(x^n),\hat{y}^n) = -\left[\hat{y}^n ln f(x^n) + (1-\hat{y}^n) ln\left(1-f(x^n)\right)\right]$$



7. logistic regression 的局限性：XOR异或问题，解决办法：级联模型--> deep neural networks (DNN)
8. 计算机视觉算法imagenet
9. 数据驱动的方法：
    - 收集图片和标签的数据集
    - 用机器学习算法训练分类器
    - 再测试图片上评估分类器
10. 分类器
    1. 线性分类器（有参方法）



    2. 多项式逻辑回归分类器：把原始线性分类器的分数变为概率形式
    3. 最近邻分类器（K-近邻，无参：从训练集的数据和标签开始，根据最相近的训练图片，预测测试图片的标签）

        超参数：K，distance metric，如何选择？超参数非常依赖于问题/数据集：尝试所有的取值；train，validation，test **better!**；数据划分，交叉验证，k折

        最后只在测试数据集上运行一次！
11. 聚类

    分割算法：把n个对象分割成K组（K给定）

    K-means算法：启发式算法，不断迭代更新聚类中心，将对象分配到离其最近的聚类中心所在的聚类中，逐步优化聚类结果，以逼近较优的 K 组分割。

    初始化中心：随机

聚类结果对初始化种子的选择非常敏感-->尝试不同的初始种子，且让他们相互远离

- *Input – Desired number of clusters,* ***k*** 输入想要聚成多少类
- *Initialize – the* ***k*** *cluster centers (randomly if necessary)* 初始化k个聚类的中心
- *Iterate –*

  1. Assign points to the nearest cluster centers 把每个点分配给距离最近的聚类中心

  2. Re-estimate the k cluster centers (aka the centroid or mean), by assuming the memberships found above are correct. 假设以上分配是正确的，重新估计k聚类中心

$$\vec{\mu}_k = \frac{1}{C_k} \sum_{i \in C_k} \vec{x}_i$$

- *Termination – If none of the objects changed membership in the last iteration, exit.*

  Otherwise go to 1.

聚类个数k的选择

聚类对outlier敏感

- 什么是机器学习，包含哪三个基本元素？
- 机器学习可以分为哪几类？
- 什么是线性模型？
- 什么是线性回归？
- 线性回归在什么情况下有Closed form的解？什么情况下没有？
- 什么是Ridge regression？什么是Lasso regression？他们的解有什么特点？
- 什么是广义的线性模型？

要点总结

- 什么是 logistic regression？它的三要素(function，objective，solution)是什么？
- 比较 logistic regression和linear regression的异同点。
- 线性图片分类器和多项式逻辑回归分类器的区别是什么？
- 什么是K近邻分类器？为什么说它是无参的机器学习算法？
- K近邻分类器他有哪些超参数？怎么选择超参数？
- 什么是聚类？什么是K-Means聚类？
- K-Means聚类的算法流程是什么？存在哪些问题？

神经网络基础

1. 单个神经元：输出与所有输入链接：全连接层（fully connected layer/dense layer）vector，bias，决策边界可以被偏置上下调整
2. 激活函数：为神经网络层的输出提供**非线性**
   - Sigmoid $f(z) = \sigma(z) = \frac{1}{1+e^{-z}}$ 输出可以表示概率
   - Tanh函数 $f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} = 1 - \frac{2}{1+e^{-2z}}$ 常用于回归任务，输出（-1，+1）
   - ReLU：特征选取和简化网络优化
   - Leaky ReLU 防止负输出为0导致丢失信息
   - Softmax 多分类：（计算涉及多个数值，其他只在一个数值上）

$$\boldsymbol{a}_i = f(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}$$

Softmax函数先对每个输出值作指数函数（exponential function）操作，然后归一化（normalise）每个值，以使得所有激活值之和为 1，代表100%的概率。

3. 多层感知器（Multi-Layer Perception）

   将原有层的输出值当作特征值来学习，可以处理更复杂的输入数据，具有更好的表达能力

4. 损失函数：用来量化预测的输出和训练数据输出（ground truth）之间的误差（error，loss value），用来设定优化神经网络参数的目标（使误差尽可能小）
   1. 交叉熵损失：二分类常用，衡量模型预测概率分布于真实标签的概率分布之间的差异。适用于输出为概率值且目标是最小化分类错误的场景。
   2. MAE： $\mathcal{L}_{MAE} = \|\boldsymbol{y} - \boldsymbol{a}\|$ 用于回归问题

卷积神经网络

1. 图像处理：
   1. 图像增强：整体增强：直方图均一化；局部增强：平滑/锐化
   2. 边缘增强，prewitt算子

卷积：

1. 动机：空间上的权值共享（不同位置使用同一个卷积核，对应平移不变性。降低模型复杂度，降低过拟合风险），稀疏连接（每一层输出只依赖于一小部分输入，并非每一个神经元都与上一层的所有神经元相连。使模型对输入数据的局部变化更加敏感，更好地提取局部特征，提高泛化能力；大幅减少参数数量，加快学习速率），等变表示（一个模型的输出随着输入的某种变换而相应地、可预测地变换。例如平移不会改变模型特征表示的本质特征）
2. 每个输入channel都需要一个filter。
3. filter size，padding，strides
4. 卷积形状的计算：
   - 输入特征图尺寸（形状）为$height_{in} \times width_{in} \times channels_{in} \times n_filters$（宽 × 高 × 通道数× 卷积核数量），输出特征图尺寸为$height_{out} \times width_{out} \times n_filters$，卷积核形状 $K \times K \times channels_{in} \times channels_{out}$
   - $outpput_size = \lfloor \frac{input\_size + 2 \times padding - filter\_size}{stride} \rfloor + 1$
5. 感受野（Receptive field）:感受野指的是**输出特征图**上的一个元素（或者说像素）在原始输入图像上映射的区域大小

- 计算：$RF_{i+1} = RF_i + (k-1) \times S_i$
-
  - RF_i 是第 i 层的感受野
  - RF_{i+1} 是第 $i+1$ 层的感受野，也即当前层

  - k 是 **当前层的卷积核** 的大小
  - S_i 是之前所有层的步长的乘积（ **不包括本层** ），也即 $S_i = \prod_{j=1}^{i} stride_j$
  - 注意 **当前层的步长并不影响当前层的感受野，感受野和填补（padding）也没有关系**
- 空洞卷积有更大的感受野 参数：dilation rate（相邻元素间插入(d-1)个元素）

6. 3D卷积

7. 池化算法：目的：增强平移不变性

应用：

1.1 判别式任务

判别式任务关注于从给定的数据中识别或分类信息。常见的应用包括：

1.1.1 二维（2D）任务

- **分类（Classification）**：识别给定图像的类别。
- **检测（Detection）**：识别图像中的对象及其位置。
- **识别（Recognition）**：比如，人脸识别。
- **分割（Segmentation）**：将图像分成多个部分或对象。
- **检索（Retrieval）**：根据特定特征搜索相似图像。
- **语言处理（Language）**：例如，文本分类或情感分析。

1.1.2 三维（3D）任务

- **3D 建模（3D Modeling）**：从 2D 图像生成 3D 模型。
- **增强现实（Augmented Reality）**：在真实世界的视图中叠加计算机生成图像。
- **双目视觉（Binocular Vision）**：利用两个相机从不同角度捕捉图像，以模拟人的双眼视觉。

1.2 生成式任务

生成式任务旨在基于已有的数据或模式生成新的数据实例。例如，根据一组图像生成新的图像（如图片修补、遮瑕），或根据一段文本生成相关的文本。

1. 目标检测：

    1. 效果评估：交并比 $IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union}$

Confusion		预测值	
		反例	正例
真实值	反例	TN(真反例)	FP(假正例)
	正例	FN(假反例)	TP(真正例)

混淆矩阵

精度率：$Precision = \frac{TP}{TP+FP}$

召回率：$Recall = \frac{TP}{TP+FN}$

- 真阳性（True Positive，TP）：预测为正类且实际上是正类的样本数。
- 假阳性（False Positive，FP）：预测为正类但实际上是负类的样本数。
- 真阴性（True Negative，TN）：预测为负类且实际上是负类的样本数。
- 假阴性（False Negative，FN）：预测为负类但实际上是正类的样本数。

P-R曲线 曲线下面积：AP值。mAP：每个类别的AP的平均值

2. 算法：

    - R-CNN: selective search (选择性搜索)：

步骤：

1. 从图像中提取约2000个候选区域
2. 将所有候选区域调整为给定大小
3. 每个候选区域输入VGG（深度卷积神经网络模型）提取特征并进行分类
4. 回归获得边界框的位置

局限性：

1. 选择性搜索慢
2. 调整候选区域尺寸可能导致宽高比例变化而影响分类准确性
3. 每个区域单独处理很耗时
4. 非端到端训练（模型不是做为一个整体训练），影响训练效率和效果

- 非极大值抑制 NMS：移除重叠边界框并保留最佳的一个。排序、比较、已知、迭代……得到一组没有重叠且置信度较高的检测框

  1. 首先，对每个类别的所有预测边界框进行排序，以便根据其分类得分或置信度进行排序。将得分最高的边界框作为参考框。
  2. 对于每个参考框，计算与其余边界框的重叠程度，即计算它们之间的 IoU（Intersection over Union）值。
  3. 如果某个边界框与参考框的 IoU 值大于预先设定的阈值（例如 0.5），则将其视为重叠，并从候选边界框列表中删除。
  4. 从列表中删除当前参考框，然后重复步骤 1-3，直到所有边界框都被处理。

  目标检测模型给出类别概率和存在置信度

- SPP Net（spatial pyramid pooling network 空间金字塔池化网络）

  步骤

  1. 全局特征提取，即将卷积层前置到selective search之前先完成一个全局的特征提取
  2. 候选区域选择：在特征图上选择
  3. 空间金字塔池化：每个候选区域应用SPP，可以输出固定大小的特征向量

  优点：使用全局特征，提高处理效率；保持了输入图像宽高比例

  局限性：

  - 选择性搜索，慢
  - 仍然不是端到端训练

- Fast R-CNN

  优点：使用RPN（区域提议网络）替换选择性搜索，速度更快；端到端训练，准确性提高

2. 图像分割：细粒度的分类
   - 语义分割
   - 实例分隔
3. 人脸识别
   - 步骤
     1. 检测图像中的人脸位置
     2. 人脸对齐到图像中心
     3. 对图像进行人脸身份识别
   - 分类

- 人脸识别

  close-set：分类问题，输入人脸在训练集中

  open-set：
- 人脸认证

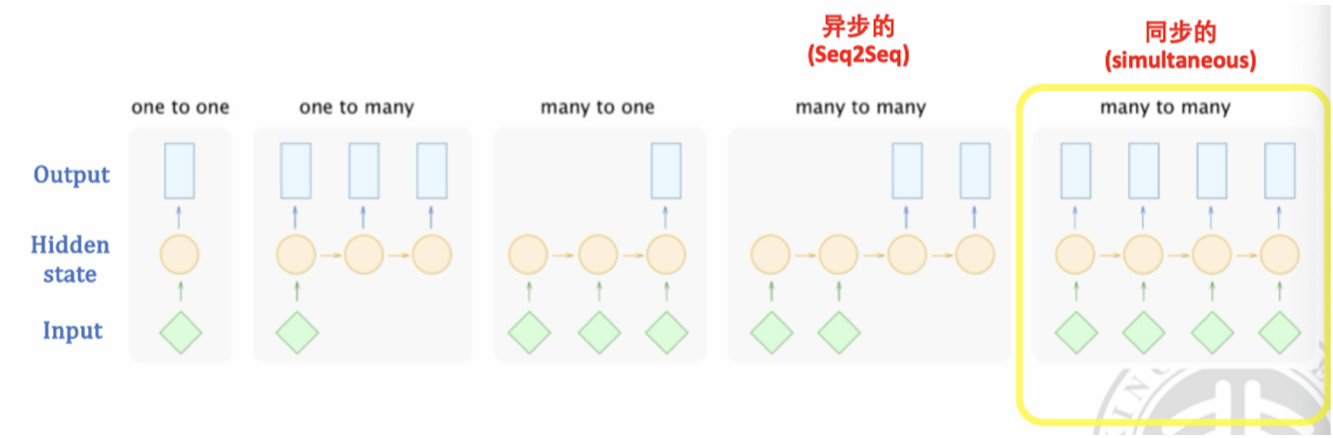
对抗神经网络

统计生成模型是**数据驱动**的方法

1. 动机：
2. 训练算法：采样，更新判别器D，更新生成器G

递归神经网络（recurrent neural network）

1. 专为处理时间序列数据设计的扩展深度学习架构
2. 词的表示：
    1. 一位有效编码向量（one-hot vector）：缺点：维度灾难、单词表示独立
    2. 词袋模型：用单词频率表示句子：缺点：维度灾难、丢失位置信息。**以上两种都丢失了语义信息。**
    3. 词嵌入（word embedding）：用一组浮点数向量来表示一个单词。优点：降维、语义、计算效率
    4. 习得词嵌入：自监督学习（自身生成伪标签），通过比较上下文来找到相似的单词
    5. Word2Vec：CBOW，SG
        - 噪声对比估计（NCE）（训练）：多个目标输出，每一个是预测是/否是上下文单词，因此是sigmoid函数而非多分类的softmax
3. 序列数据（sequential data）
    1. 时间步



运用递归神经网络（RNN）存储和处理时序信息

4. 朴素递归神经网络（Vanilla RNN）

局限性：长序列的信息会被遗忘

5. LSTM 长短期记忆网络

   添加门控机制控制信息流动

   运算过程：

- 计算遗忘门：决定从细胞状态中丢弃什么信息

$$f_t = sigmoid([h_{t-1}, x_t]W_f + b_f)$$

其中：

- f_t: 遗忘门（forget gate）的值
- $sigmoid$: sigmoid 激活函数，用来将值压缩到 0 和 1 之间
- $[h_{t-1}, x_t]$: 前一时刻的隐藏状态与当前时刻的输入数据 **拼接**
- W_f, b_f: 遗忘门的权重矩阵、偏置项

随后，遗忘门的输出值 f_t 会与前一时刻的细胞状态相乘，从而决定丢弃哪些信息：

$$C'_{t-1} = f_t \odot C_{t-1}$$

- 计算输入门：控制当前输入和过去记忆的结合程度
  - 一个候选信息向量，决定我们要在细胞状态中存储哪些新信息 $\tilde{C}_t = \tanh([h_{t-1}, x_t]W_C + b_C)$
    - $\tilde{C}_t$: 当前时刻的信息向量（information vector）
    - W_C, b_C: 信息向量的权重矩阵、偏置项
  - 一个输出向量，决定我们要把候选向量的哪些部分添加到细胞状态中

$$i_t = sigmoid([h_{t-1}, x_t]W_i + b_i)$$

- i_t: 输入门（input gate）的输出值
- W_i, b_i: 输入门的权重矩阵、偏置项

- 计算新的细胞状态：根据遗忘门和输入门的结果更新 $C_t = f_t \odot C'_{t-1} + i_t \odot \tilde{C}_t$

- 计算输出门和新的隐藏状态

输出门决定了 **下一时刻的隐藏状态**，隐藏状态包含了过去信息，并通过输出门过滤：

$$o_t = sigmoid([h_{t-1}, x_t]W_o + b_o)$$

新的隐藏状态由 **输出门和新的细胞状态** 决定：

$$h_t = o_t \odot \tanh(C_t)$$

其中：

- o_t: 输出门（output gate）的值
- W_o, b_o: 输出门的权重矩阵、偏置项
- h_t: 当前时刻的隐藏状态

注意事项：

- 门控都是sigmoid函数（输出在0，1之间）
- 向向量输入信息时用tanh：输出范围-1，1，可以使信息的变化范围更大，反向传播时更易传递

6. LSTM变体，如门控循环单元GRU，减少了计算成本和内存使用

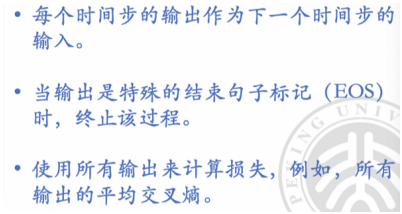
7. 时间序列的应用：

- many-to-one:

  句子情感分类：分类任务，用最后一个输出计算，隐藏向量顶部堆叠一个全连接层和softmax

- one-to-many：

图片描述：

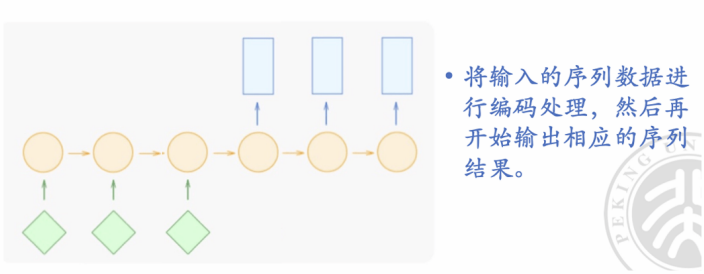


- many-to-many(同步):

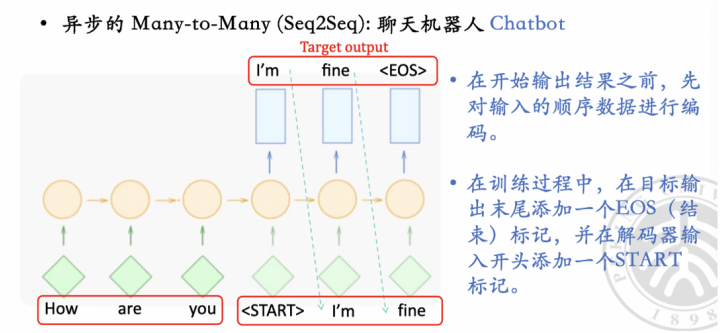
  交通计数：训练时需要预先定义一个序列长度来计算损失，输入数据一个接一个进行处理。

  文本生成/语言建模：每一步的输出作为其下一步的输入

- many-to-many(异步):



聊天机器人：



往年题经验：

1. 跳跃连接可以有效缓解梯度消失问题
2. 判别器D的损失函数旨在最大化真实数据被识别为真实数据的概率，同时最小化生成数据被识别为真实数据的概率。而生成器G的损失函数则旨在最大化生成数据被识别为真实数据的概率。因此损失函数形如

$$\mathcal{L}_D = -\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1 - D(G(\boldsymbol{z}))]$$

$$\mathcal{L}_G = -\mathbb{E}_{\boldsymbol{z}\sim p_z}[\log D(G(\boldsymbol{z}))]$$