DL

人工智慧是我们想要达成的目标，而机器学习是想要达成目标的手段，希望机器通过学习方式，他跟人一样聪明。而深度学习和机器学习有什么关系呢？深度学习就是机器学习的其中一种方法。

introduction

人工智慧是我们想要达成的目标，而机器学习是想要达成目标的手段，希望机器通过学习方式，他跟人一样聪明。而深度学习和机器学习有什么关系呢？深度学习就是机器学习的其中一种方法。

Regression和Classification的差别就是我们要机器输出的东西的类型是不一样。在Regression中机器输出的是一个数值，在Classification里面机器输出的是类别。假设Classification问题分成两种，一种叫做二分类输出的是是或否（Yes or No）；另一类叫做多分类（Multi-class），在Multi-class中是让机器做一个选择题，等于是给他数个选项，每个选项都是一个类别，让他从数个类别里选择正确的类别。

Regression 回归问题

输出一个标量；

分为训练阶段、测试阶段

$(y-y\cdot)^2$ // 真实减去预测括号平方；
主流模式是有监督的学习；

解决overfitting的办法？

打印资料上

unsupervised（无监督）learn wordEmbedding：

为什么wordEmbedding不需要DL？

1. shawdow line 可以训练非常快，而且效果可以

2. WordEmbedding 其实是其他深度学习的超参数，超参数训练可能就不需要深度学习。
   ### 共享参数：
   共享参数之后不管history多长，参数都不会增加。十多个history才效果比较好。
   同一个词汇在不同的位置都能得到一样的wordEmbedding；因为参数共享。

每个单词的意思可以从上下文读出来，

方法一：count base

如果两个单词经常一起出现，那么他们的词向量的值应该很相近。
两个词汇的词向量做内积应该和两个词汇在一篇相同的文章里面出现的次数成正比。

方法二 prediction base

将一段文本中前面n个词汇独热编码后的词向量输入到一个神经网络中，使其输出的各个维度中对应于文本中下一个词汇的那个维度的几率最大时，那么隐藏层对应的参数就是我们要的词向量。
除此之外也可以使用中间的词汇去预测上下文，或者使用上下文预测中间的词汇来求词汇对于的词向量。

为什么语义相近的词汇通常有相似的嵌入表示?

因为通常语义相近的词汇很大可能也有相同或者相似的上下文，也就是在神经网络中有相同或相似的输出，使得神经网络通过其拟合能力修改模型的参数使得这两个词汇有相似的隐藏层。也就是有相似的嵌入表示。
作业题题目答案

CNN

convolutional neural(神经元） network

- 参数公用，减少，模式在不同的地方
-
- subsampling（二次抽样）不会改变对象结果

matrix里面参数是学出来的。//不知道学到了什么模式的

彩色的图片，RGB 3 x 3x3的matrix
参数个数 3x3x3 27个参数。

convolution就是 Fully Connected的简化版，为什么呢?
部分神经元的相连，weight就是matrix的值。参数变少了。 使用同一个marix ，neural 共享了参数，大幅度减少了参数。

Max Pooling（池化
avarage pooling

第三步：flatten
拉直，然后dnn。

看学到了什么

- 第一个filter的参数；
- 输入某图片使得某层的filter某几个神经元输出最大。
- 输出的种类对图片的某个像素做微分，输出值的绝对值越大，说明这个像素对辨别是不是这个类别起的作用越大

热力图：

- 把image其中一小部分遮挡住，看遮掉哪个地方，machine辨识不出这个图片的类别
  分析全连接层。

deep dream：
让每一层学到的值；大的更大，小的更小

最后一个layer，
看到的图片也不是一个 1 2 3.。的图片也就是dl不仅仅是模拟数据，是有一定的举一反三的能力的

阿尔法GO用到的cnn：
19x19的image
cnn的好处。考虑image的特性：
围棋也有很小的模式。很多都是在5x5的pattern
alpha go does not use max pooling。。

语音辨识也可以用cnn，

whyDeep

deepLearn：
其实在做模块化的工作。

模块化的好处是什么？

- 让模型变得简单了
- 把问题变简单了，train data 没有那么多，也可以把task做好。

语言上用模块化的概念：

只要隐藏层的神经元够多，一个隐藏层就可以拟合任意连续的函数。
（没有管效率）
减少了参数，比较少的参数，比较少的data。就可以达到同样的效果。
效率更高。
比较deep shadow network之间的效果，要参数差不多的情况。

End to End Learning

只给模型的输入、输出。不指明中间每一个函数怎么分工，让它自己去学到中间每一个生产线（函数）（层）应该做什么事情 叠一个很深的 neural network

非常像的输入，不同的输出
非常不同的输入，相同的输出。 简单的层，难以做到。
图片分开，
数字分开。
经过多层，层次变高，看起来聚类起来了，然后再来分开。

RNN（Recurrent Neural Network）

不同时刻储存在记忆单元里面的值不一样。

- 改变输入的顺序会改变输出

ELman Network：

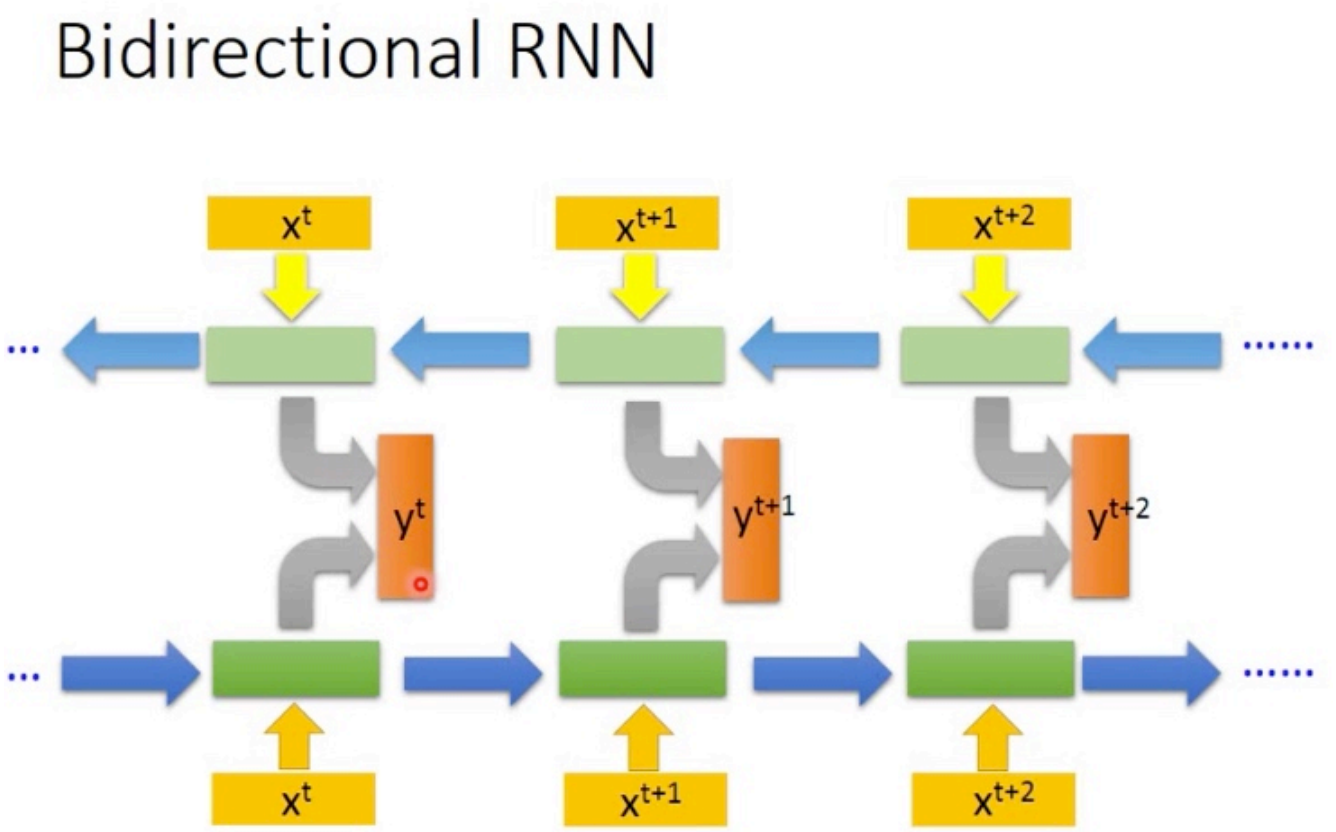
把隐藏层的输出存起来，下一个时间点再读出来

Jordan Network：

记忆里面存的是整个network output的值。把output的值下个时间点再读进来，因为output是有target的，所以可以比较清楚的知道存在menmory里面的是什么东西。

Bidirectional RNN

看了整个句子之后，才决定每个词汇的flat应该是什么，这样当然会比只看句子的一半得到的效果更好。
读取方向可以反过来，可以同时两边都读出结果出来，都接给一个output layer

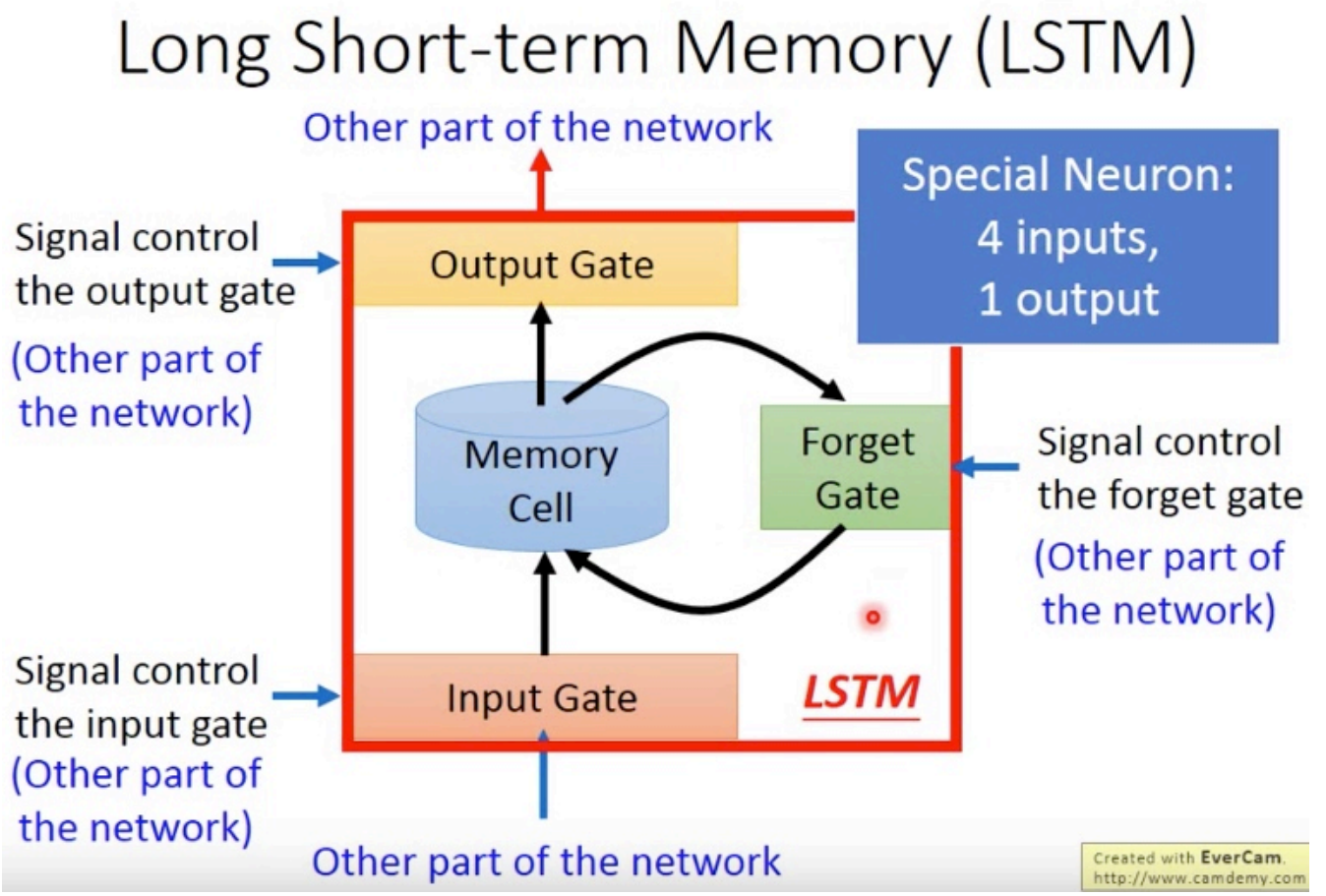


LSTM

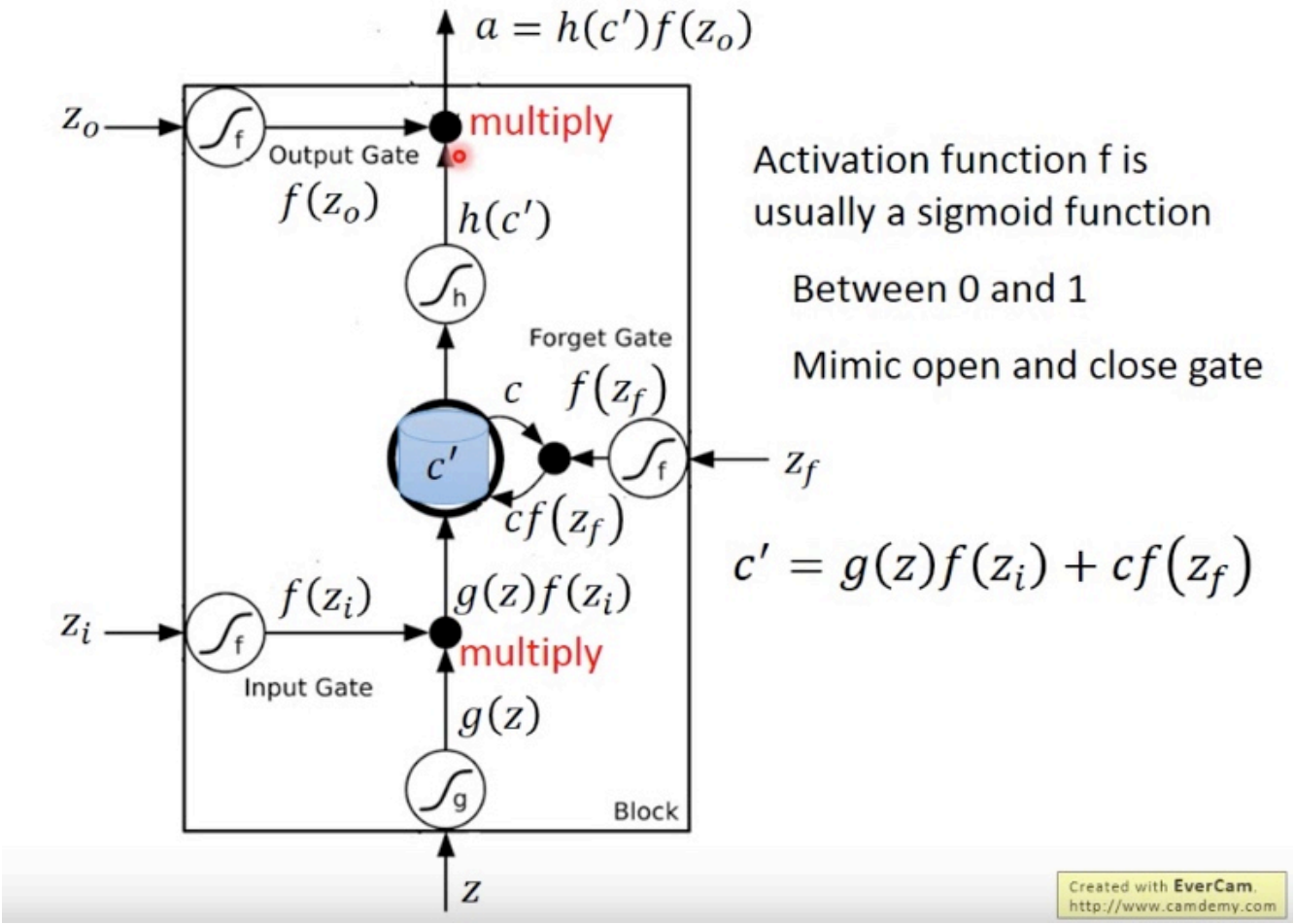
long Short-term Memory
对于一个Memory：
有一个input的闸门
neural network自己学到什么时候打开这个input 开关。
有一个output gate被打开的时候，才被读出

forget gate ：

决定什么时候把memroy里面的值重置掉。
打开的时候，不重置。



每一个部分起作用表达式：



BPTT

一种反向传播算法的升级版本用于rnn
cliping

循环神经网络在训练过程中为什么容易出现梯度消失或梯度爆炸问题

因为RNN在训练的时候，将记忆单元里面的值在很长的不同的时间点反复使用，造成参数变化很小的时候，会对梯度造成很大的影响。

LSTM为什么能解决 gradient vanish的问题。not gradient explode

//rnn每个时间点，memory都会被完全覆盖掉。

因为在LSTM网络中，只要输入能够影响记忆单元（memory），那么这个影响会一直存在于记忆单元中。而不是像rnn每次记忆单元的值被替换。除非遗忘门被关闭，导致

记忆单元的值被重置，而大部分时间遗忘门是不会被关闭的。从而解决了梯度消失的问题。

GRU Gated Recurrent Unit 只有两个gete more simple than LSTM

旧的不去，新的不来。
也就是input gate 打开的时候，forget gate 关闭，也就是加入全部的input的话，就清空了记忆。（但是可以部分呀）

还有其他的：

- Clockwise RNN
- Structurally Constrained Recurrent Network （SCRN）
  ### RNN的应用：
- 评论分析
- 文章找关键词汇
- 语音识别
- 语言翻译
- 英文的声音翻译成中文的文字
- input句子得到其文法结构
- 语音搜索
- 语音的word embedding
- 聊天机器人

attention based model

1. 阅读理解
2. 看图回答问题
3. 语音，听力回答问题

loss function 是对模型像学出什么的期待。

神经网络欠缺的能力

- 记不住之前学到过的知识
- 靠边的权重去记忆，每次学习更新权重，参数更新了，以往的学习到的知识反倒不会了

单个神经元是什么？和线性模型、分类模型的关系

单个神经元的作用是做二分类。
在激活函数之前，都是一个线性回归的模型，加上一个非线性的激活函数。如果是个sigmod函数起到二分类的作用
relu函数：小于0输出为0 大于零直接输出。

图卷积模型与cnn rnn dnn 的关系

CNNs是encode了空间相关性的DNN，RNNs是encode进了时间相关性的DNN
GCN图卷积的核心思想是利用『边的信息』对『节点信息』进行『聚合』从而生成新的『节点表示』。
图卷积神经网络具有卷积神经网络的以下性质：
1、局部参数共享，算子是适用于每个节点（圆圈代表算子），处处共享。

2、感受域正比于层数，最开始的时候，每个节点包含了直接邻居的信息，再计算第二层时就能把邻居的邻居的信息包含进来，这样参与运算的信息就更多更充分。层数越多，感受域就更广，参与运算的信息就更多。

GCN模型同样具备深度学习的三种性质：

1、层级结构（特征一层一层抽取，一层比一层更抽象，更高级）；

2、非线性变换 （增加模型的表达能力）；

3、端对端训练（不需要再去定义任何规则，只需要给图的节点一个标记，让模型自己学习，融合特征信息和结构信息。）

GCN四个特征：

1、GCN 是对卷积神经网络在 graph domain 上的自然推广。

2、它能同时对节点特征信息与结构信息进行端对端学习，是目前对图数据学习任务的最佳选择。

3、图卷积适用性极广，适用于任意拓扑结构的节点与图。

4、在节点分类与边预测等任务上，在公开数据集上效果要远远优于其他方法。

	LSTM	DNN	CNN	RNN	GCN
			图像处		

应用场景		语音，图像，自然语言处理	理中的检测或分类、目标检测、语音识别	时间序列数据的首选神经网络，主要用在自然语言处理，语音识别等	图的结点数据
优化器					
优点		学习速度快，相对shadow网络	共享参数，参数变少，训练更加不容易过拟合	可以考虑样本之间时序的关系，对于时序信号有更好的效果	对于拓扑结构的节点与图有很好的效果 |
缺点		参数数量的膨胀，极易陷入局部最优	深度模型容易出现梯度消散问题	很容易出现梯度消失，梯度爆炸的问题	Transductive learning的方式，需要把所有节点都参与训练才能得到node embedding，无法快速得到新node的embedding。
适合的数据	语音数据，有时序关系的数据		局部模式的数据，图，	序列，具有上下文关系的数据	图具有拓扑关系的数据。

DNN(深度神经网络)

神经网络是基于感知机的扩展，而DNN可以理解为有很多隐藏层的神经网络。多层神经网络和深度神经网络DNN其实也是指的一个东西，DNN有时也叫做多层感知机

（Multi-Layer perceptron,MLP）。

DNN存在的局限：
参数数量膨胀。由于DNN采用的是全连接的形式，结构中的连接带来了数量级的权值参数，这不仅容易导致过拟合，也容易造成陷入局部最优。

局部最优。随着神经网络的加深，优化函数更容易陷入局部最优，且偏离真正的全局最优，对于有限的训练数据，性能甚至不如浅层网络。

梯度消失。使用sigmoid激活函数（传递函数），在BP反向传播梯度时，梯度会衰减，随着神经网络层数的增加，衰减累积下，到底层时梯度基本为0。

无法对时间序列上的变化进行建模。对于样本的时间顺序对于自然语言处理、语音识别、手写体识别等应用非常重要。

CNN(卷积神经网络)

主要针对DNN存在的参数数量膨胀问题，对于CNN，并不是所有的上下层神经元都能直接相连，而是通过“卷积核”作为中介。同一个卷积核在多有图像内是共享的，图像通过卷积操作仍能保留原先的位置关系。

CNN之所以适合图像识别，正式因为CNN模型限制参数个数并挖掘局部结构的这个特点。

RNN(循环神经网络)

针对CNN中无法对时间序列上的变化进行建模的局限，为了适应对时序数据的处理，出现了RNN。

在普通的全连接网络或者CNN中，每层神经元的信号只能向上一层传播，样本的处理在各个时刻独立（这种就是前馈神经网络）。而在RNN中，神经元的输出可以在下一个时间戳直接作用到自身。

（t+1）时刻网络的最终结果O(t+1)是该时刻输入和所有历史共同作用的结果，这就达到了对时间序列建模的目的。

存在的问题：RNN可以看成一个在时间上传递的神经网络，它的深度是时间的长度，而梯度消失的现象出现时间轴上。

LSTM(长短时记忆单元)

为了解决RNN中时间上的梯度消失，机器学习领域发展出了长短时记忆单元LSTM，通过门的开关实现时间上记忆功能，并防止梯度消失。

扩展

深度神经网络中的梯度不稳定性，前面层中的梯度或会消失，或会爆炸。前面层上的梯度是来自于后面层上梯度的乘乘积。当存在过多的层次时，就出现了内在本质上的不稳定场景，如梯度消失和梯度爆炸。

梯度爆炸（exploding gradient）：梯度爆炸就是由于初始化权值过大，前面层会比后面层变化的更快，就会导致权值越来越大，梯度爆炸的现象就发生了。

在深层网络或循环神经网络中，误差梯度可在更新中累积，变成非常大的梯度，然后导致网络权重的大幅更新，并因此使网络变得不稳定。在极端情况下，权重的值变得非常大，以至于溢出，导致 NaN 值。

网络层之间的梯度（值大于 1.0）重复相乘导致的指数级增长会产生梯度爆炸。

解决梯度爆炸的方法参考：详解梯度爆炸和梯度消失

梯度消失（vanishing gradient）：前面的层比后面的层梯度变化更小，故变化更慢，从而引起了梯度消失问题。

因为通常神经网络所用的激活函数是sigmoid函数，这个函数有个特点，就是能将负无穷到正无穷的数映射到0和1之间，并且对这个函数求导的结果是f′(x)=f(x)(1−f(x))。因此两个0到1之间的数相乘，得到的结果就会变得很小了。神经网络的反向传播是逐层对函数偏导相乘，因此当神经网络层数非常深的时候，最后一层产生的偏差就因为乘了很多的小于1的数而越来越小，最终就会变为0，从而导致层数比较浅的权重没有更新，这就是梯度消失。

因为sigmoid导数最大为1/4，故只有当abs(w)>4时梯度爆炸才可能出现。深度学习中最普遍发生的是梯度消失问题。

1 对比各种神经网络异同点

- 全连接DNN的结构里下层神经元和所有上层神经元都能够形成连接，带来的潜在问题是参数数量的膨胀，模型容易过拟合，参数消失，参数爆炸。及其陷入局部

最优

- 对于CNN来说，并不是所有上下层神经元都能直接相连，而是通过“卷积核”作为中介。同一个卷积核在所有图像内是共享的，图像通过卷积操作后仍然保留原先的位置关系。由于CNN模型限制参数了个数并挖掘了局部结构的这个特点。顺着同样的思路，利用语音语谱结构中的局部信息CNN照样能应用在语音识别中。cnn是局部的的连接，是一个稀疏的网络，dnn是前馈全连接神经网络

- RNN：CNN是刻画特征模拟的神经网络结构，无法对时间序列上的变化进行建模。然而，样本出现的时间顺序对于自然语言处理、语音识别、手写体识别等应用非常重要。对于这种需求就产生了循环神经网络RNN

- GCN:由于CNN无法处理Non Euclidean Structure的数据，又希望在这样的数据结构（拓扑图）上有效地提取空间特征来进行机器学习，在Non Euclidean Structure的数据上无法保持平移不变性。通俗理解就是在拓扑图中每个顶点的相邻顶点数目都可能不同，那么当然无法用一个同样尺寸的卷积核来进行卷积运算.所以GCN成为了研究的重点.广义上来讲任何数据在赋范空间内都可以建立拓扑关联，谱聚类就是应用了这样的思想。所以说拓扑连接是一种广义的数据结构，GCN有很大的应用空间。

2 为什么多个神经元连接起来能够起到如此强大的功能

单个。。。
因为就是涉及到 对空间的分割。
单个vs 深层学习
对空间的分割
每一直线分割出一个等价类。编码映射诱导了对空间的分割。
其实就是：分类的能力变强
很多神经元去分割，把分类分的越来越多类别，越来越细。

3 是否各种神经网络是否就很完善了，是否存在缺点局限性

学出的是特定数据的流型，是目前问题。迁徙能力差
遗忘性
受学出的流型的限制。

神经网络有什么问题?

以什么样的组合，达到什么样的效果

图神经网络：

从图的结构：两层神经元之间的连接构成一个二分图：
从线性代数的角度，向量的运算相加；

核心思想：在不规整的图中，把邻居结点的信息汇聚。
主要思路：1.

网络结构是什么个事情，适应使用场景，几种网络类型适合处理的数据类型

考思路：
比如一种交通道路的图片数据，分别用dnn，cnn，GCN 图网络来处理：
交通图的来处理
使用cnn来处理。
使用GCN来处理，道路当成一个结点，道路的分叉口当作一个结点
使用dnn来处理：每个时刻的交通流量信息，写成一个向量。
每个路口的流量，对应一个维度。坏处：一维度的向量，损失了信息。
需要用到循环神经网络，时间的度量

循环神经网络RNN 与图网络 GCN的关系?

把RNN看出一个无限延展、线性的图。
RNN循环神经网络考察的是时间上的相关性
DNN各个维度特征之间的相关性
CNN局部空间之间的相关性
GCN基于图这种拓扑结构，邻居结点之间的，一跳两跳之间的相关性；可以叠加多层。考察的是图之间的连通性

从局部的角度看图网络就是：
消息传递，信息融合

单个神经元的作用就是个二分类

分类和线性回归的组合；单个神经元的作用比较弱；
单个神经元的作用实际上就是在空间上切了一刀，

神经网络中的神经元，单个神经元的作用实际上就是在空间上切了一刀。多个神经元，就可以分出一个很细的平面,分多刀，更细致。+deep
多个神经元就是会切很多刀，deep 只是让模型更好更快的学习出

- why Depp ？
-

为什么DL这么强?

①深度学习的成功是基于两条

- 数据本身的内在规律
- 深度学习能揭示并利用这些规律
- 自然数据背后隐藏着流型结构
- 深度学习方法可以提取这些游戏结构：
    - 并用神经网络来表达流型间的映射
    - 给出流型本身的参数坏和参数表示

    。。。看最后的ppt上

    ②数据学科中的基本定律可归结为：
- 流形分布定律
- 聚类分布定律

DL的不足，局限性：

分的越细，能力越强，分类种类越多，越容易受到攻击：

- 容易被攻击 why?
- 胞腔之间的距离更加相近，数据的微小改变，胞腔容易被改变成另外一个。就变成了另外一个类别。
- 即使图片产生了很小的差别也会造成很大的偏差
- 遗忘性
- 没有可迁移性
- 模型没有可解释性的局限性。
- 黑客找到一个方向的最小改变，使得胞腔的改变最大。使得模型是坏的
- 有目标的攻击，约束是变换越小越好

模型的可解释性：

线性模式是可解释的。w大，对模型的重要性大
决策树是可解释的。
但是目前的深度学习可解释性还很差。

无监督学习如何学习的呢?

- 监督学习是一种目的明确的训练方式，你知道得到的是什么；而无监督学习则是没有明确目的的训练方式，你无法提前知道结果是什么。
- 监督学习需要给数据打标签；而无监督学习不需要给数据打标签。
- 监督学习由于目标明确，所以可以衡量效果；而无监督学习几乎无法量化效果如何。

无监督学习是一种机器学习的训练方式，它本质上是一个统计手段，在没有标签的数据里可以发现潜在的一些结构的一种训练方式。

聚类、降维

聚类：简单说就是一种自动分类的方法，在监督学习中，你很清楚每一个分类是什么，但是聚类则不是，你并不清楚聚类后的几个分类每个代表什么意思。

降维：降维看上去很像压缩。这是为了在尽可能保存相关的结构的同时降低数据的复杂度。

降维的过程是编码，升维的多层是解码

自编器的编码encode和 decode解码过程可以不对称

无监督模型，输入数据，和输出数据之间差别越低越好。
目标是中间的code，编码

图片故意加噪声

迫使网络学 出去噪的能力

任意函数都能使用Relu函数去拟合

深蓝国际象棋
alphago 围棋

单个神经元的作用是什么?

对空间的剖分，越多越细，更容易受到攻击
多个神经元为什么有这么强的作用。分割更多
学习了流型，只能适应某种数据，
没有可迁移性
和深度+why deep关联起来，

各个网络之间的特点，适合的数据；