深度学习从mlp开始 //深度学习常用网络结构

//总结

mlp（前向传播网络）：

每一层（除输入外）都对上一层节点输出数据进行线性组合，再对其进行非线性变化（例如Sigmoid)

最后计算损失函数（MSE,CrossEntropyLoss），再用梯度下降法修正模型，直至梯度为0或循环结束。

而更新各层之间的系数可通过Hinton于1986年提出的BackPropagation算法。

而对于中间出现的过拟合现象，采用正则化的方式使各系数都较为平滑。

但该种模型过于简单，无法适应复杂的需求，因此人们提出了AE（auto encoder），包括Encoder与Decoder两部分

首先数据经过encoder压缩，再经过decoder解压。为了防止过拟合，在损失函数中加入了惩罚项

以及在输入中加入微小扰动，提高AE性能，引入KL散度。

CNN（卷积神经网络）：

由于mlp需要将数据展成一维向量（batch维不计），因此对于图片这种数据类型，尽管空间上像素相邻，

但展成向量后可能差距较远，因此会丢失许多相对位置信息。

受到神经科学的启发，LeCun与于1998年提出了卷积神经网络，由卷积层（离散卷积）与池化层构成，最后与一全连接层相连。

而池化层包括最大池化与平均池化。由于反向传播算法的特点，因此要求前者多储存最大元素的位置。

每个神经元只可见一部分数据且保持空间上的位置信息，越后层的神经元可见的范围越大，逐层提取特征。

CNN的反向传播算法如下：

最早提出的网络为LeNet，包括较大的卷积核，激活部分仍使用sigmoid函数。

由于sigmoid函数两侧的饱和区过大，当神经网络深度增加时，容易发生梯度消失现象。

因此效果并不显著，被后来的支持向量机（svm）超越，深度学习走进寒冬。

随后的AlexNet，激活层采用了relu函数，一定程度上解决了梯度消失问题，而且计算较sigmoid函数更简单

为了防止过拟合，开始在训练时采用dropout操作，使得最终期望不变，而测试时放弃dropout。

而且采用了数据增广操作。开启了GPU并行计算的先河

为了改进，VGGNet则选择用更小的卷积核3\*3代替原来的5\*5,7\*7，减少了参数量。

增加了神经网络的深度与各层的通道数。首次实验表明神经网络越深效果越好。

GoogLeNet则在中间设计自己的模块，分成多路卷积最终叠加，不仅有利于GPU并行计算。

还验证了网络的宽度同样能够提升模型的准确率。且广泛运用了1\*1卷积核，大幅减少了参数量。

看似上述的结论基本确立了今后的cnn的发展方向，即更深更宽。

然而人们实验发现，当层数增加到一定数目后，在训练集中的错误率不降反增，即发生了退化。

因此有人提出了BatchNormalization来解决，但仅解决了梯度消失的问题。

有人猜测是神经网络不擅长拟合恒等变换。

HeKaiMing则引入了残差网络（Resnet)，灵感或许来源于LSTM。他认为残差学习更为容易，而且有些层不一定真的需要学习东西。

因此将上层的输入直接与下层的输出相连，形成ShortCut，使得更深的网络成为可能，将其拓宽到了152层。

在ImageNet竞赛上也首次低过人类的错误率。

衔接

随后HuangG提出了DenseNet，脱离了加宽加深网络的思路，引入旁路。

在一个DenseNet block内实现全连接。

与上述ResNet简单在输出处相加不同，DenseNet则在输出处直接concat，大幅减少了参数的使用，更强调了底层数据。

同时为了控制通道数，引入过渡层。

尽管CNN具有一定的空间不变性（shift invariance），但当内部物体移动范围较大，经过旋转后，cnn的准确率会有所下降。

此时Spatial transformer被提出，包括Localization network, grid generator, sampler。

首先提取特征，学习需要的仿射变换后，再用后序对输出各坐标在输入中寻找对应像素（双线性插值），输入正常的cnn网络中。

提高了cnn的鲁棒性，具有更强的空间不变性。

RNN（循环神经网络）：

为了解决序列化。在nlp上的效果，应用。

对于语言序列，尽管CNN可以采用一维的卷积核来训练，但由于序列长度不定，上下文依赖关系不定等原因，CNN的表现往往不太好。

因此一种具有时序性的网络被首次提出。

每个单元包括此时序列的输入，还包括上个单元的隐藏信息，concat后学习出一个output（输入全连接层，softmax得到结果）与hidden（输入下一级）

测试时，将上个单元的输出作为下个单元的输入。然而，若一直用正式数据训练，模型在测试时经常表现不佳。

以上的训练方式叫teacher forcing，随后人们提出了Beam Search(quit greedy search)与Curriculum Learning(先易后难）

对RNN的反向传播算法进行分析，发现需要在时间维度上连乘，因此对于长序列来说，容易发生梯度消失现象，即无法利用过去长时间的信息。

此时LSTM被提出，引入记忆细胞与三个控制门，分别控制旧记忆的遗忘、记忆的补充以及形成下一个隐藏状态。

相当于创建了记忆的Shortcut，解决了梯度消失问题。

随后更少参数，计算更快的gru被提出。取消了记忆部分，直接对隐藏状态进行重置、输入操作，参数更少，运算更快。

中间为了适应文本的上下文有关性，分层RNN，双向RNN（无回路）都相继被提出，目的都是学习到上下文不同的语义（隐藏信息）

神经图灵机模仿图灵机在计算机内部开辟空间来模拟交互，建立输入纸带到输出纸带的映射，而读写头采用softmax结构。

修改采用先擦除再添加的方式。相较于LSTM在预测循环文本上的表现更优。

在翻译问题中，由于输入与输出并没有确定的字数关系，所以传统的RNN模型无法很好的表现。

此时Seq2Seq模型被提出，同样包含encoder与decoder两部分。将输入统一编码成一个hidden state

输入decoder，再由decoder将对应的翻译输出。在这个过程中同样存在着一些问题。

如果文本过长，一个hidden state很难体现全文各词之间的关系，但也是机器翻译的先驱。

后来的attention机制则在encoder输出hiddenstate之外还对每一个小单元进行输出，最后与decoder的输出做内积，经过softmax后得到结果。

卷积网络与循环网络结合可以生成许多东西。

//现代改进方法

Transformer

近几年提出了一种新的模型，用全attention的方式代替了原来的lstm或gru。

同样包括encoder与decoder部分，先将输入embedding层，加上位置信息Postitional embedding（公式）后输入encoder，encoder部分包括一个multi-head attention与前馈网络FFN，中间添加了残差操作以及normalization。

多头部分通过三个矩阵得到三个向量分别为Q,W,V，运用self-attention机制，最后输出K,V至decoder中。

Decoder层类似也是接受position embedding的输出，但为了防止过拟合，引入了masked multi-head attention，即每个词只能看到前面部分的文本。接着得到Q，与encoder的KV输入multihead attention中，其余与encoder一样。

可以并行化，更容易建立长久的依赖，

可解释性更强，<https://zhuanlan.zhihu.com/p/44121378>

但不是图灵完备的，不擅长做循环的。

Universal transformer

GPT无监督预训练加带监督的微调，将normalization放在attention前

Bert encoder 多语言更广泛 ，在输入中引入了mask

Vit Transformer cv nlp 将图片切片输入，position class embedding

不足：

1. 正确率仍不足

2. 环境能耗

3. 落地，移动端

4. 小样本学习

5. 缺乏逻辑推理能力

6. 多任务