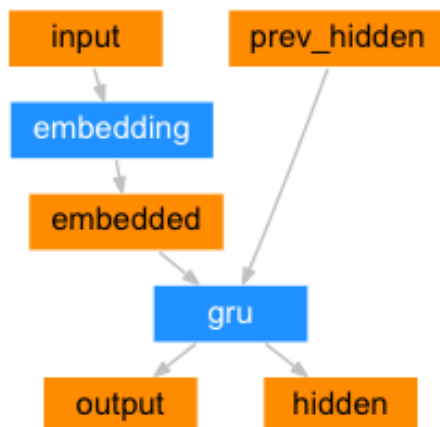


# Seq2Seq model with attention

谈一点感想： 什么？ decoder还有除了encoder产生的context以外的输入？？

读完sequence to sequence model 之后没有对代码进行操作（后来经过查询github发现我已经写过tensorflow版本的attention NN了），以至于一直未能真正理解Seq2Seq model中decoder的结构。

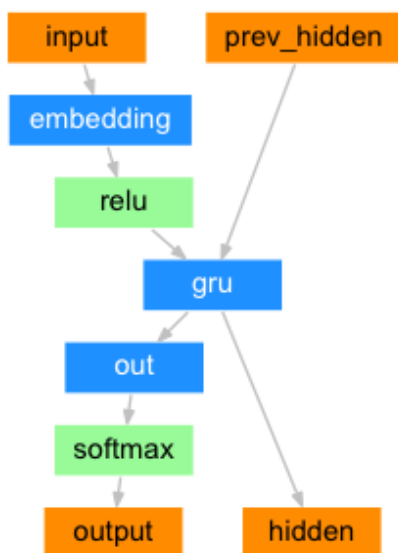
Encoder的结构先回忆一下：



encoder 核心就是GRU（也可能是LSTM），把input通过embedding(machine translation的话)，和hidden state一起输入到GRU。

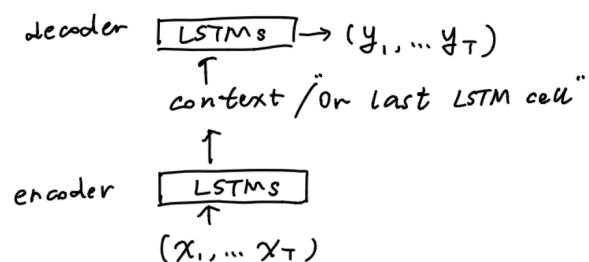
input在机器翻译中就是翻译前的sequence。

Decoder 不包含attention的结构如下：



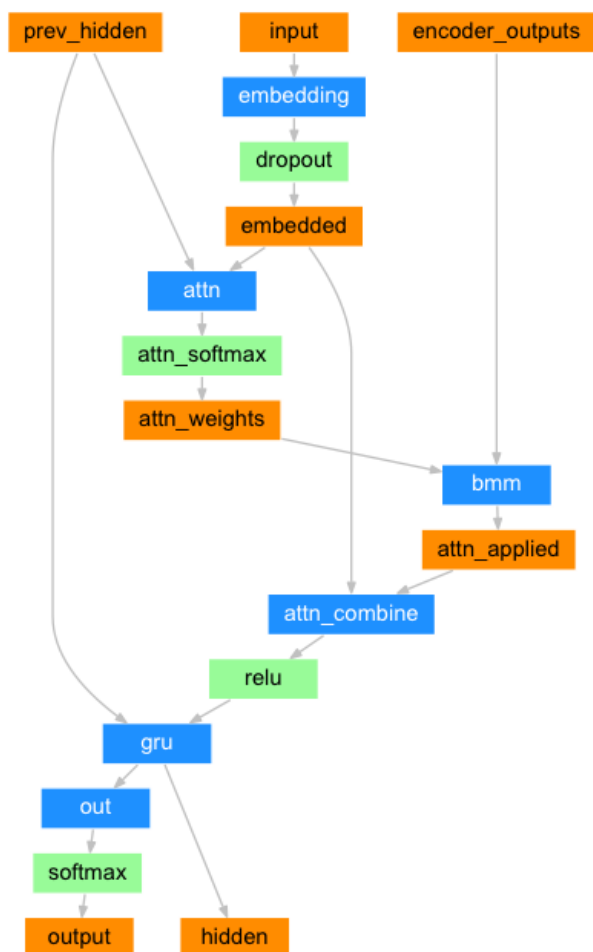
最简单的Decoder 形式中，prev\_hidden就是Encoder的最后一个hidden输出。

但是除此之外还有一个input。input到底是什么东西？按照通常理解的如下图的Seq2Seq机制，decoder似乎应该只有一个input就是encoder产生的context呀？



实际这里decoder的输入是配合GRU的作为GRU的输入，因为GRU的结构需要GRU(input, hidden)两个输入，于是就把机器翻译的真实结果(truth)作为input。翻译中会对一个sequence包裹上开始和结束标志(<SOS>和<EOS>)，于是decoder的第一个input就是<SOS>。这里的input还要讨论两种情况，是否是“教师强制”(teacher forcing)，如果是教师强制，那么input就是严格的预期的翻译后的结果，如果不是，那就可以是GRU自己产生的output。可以在训练的过程中随机选择是否教师强制。

来看一下带attention的Seq2Seq model结构图：



Decoder依旧需要input，与最简单的decoder区别是input也要参与到attention的计算中，这里的prev\_hidden不再是encoder的输出，而是decoder自己的hidden state，encoder\_output可以直接从context中提取，但是prev\_hidden要每次取上次decoder输出的hidden（如果第一次执行就先initial hidden state，通常初始化为全0）。

感想：温故而知新的感觉并没有那么好，因为这个模型结构看了不止一遍，甚至还跟着tensorflow的tutorial写过一次代码，但是decoder的输入输出过一阵子还是不明白。:(((