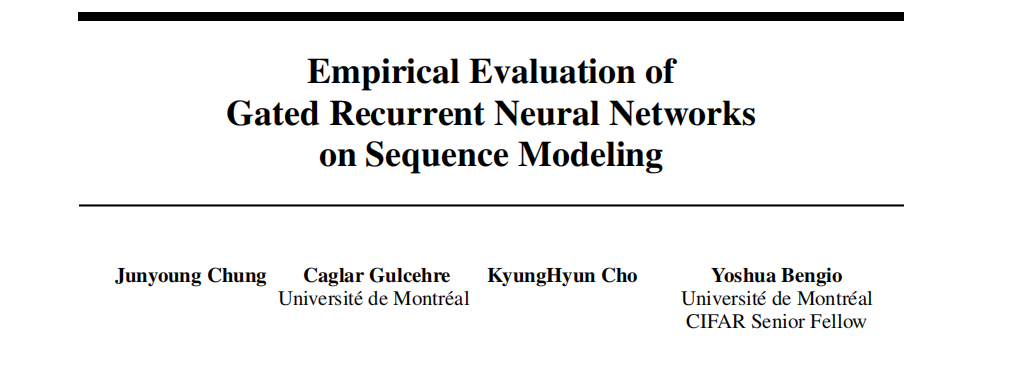
本周阅读了两篇文章，分别是：

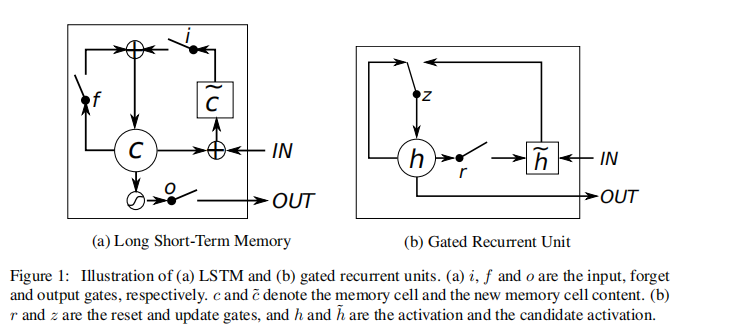
1. **Empirical Evaluation of Gated Recurrent Neural Networks**

**on Sequence Modeling**



首先是引进GRU模块的原因：

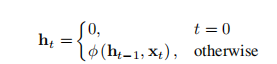
GRU模块被引进的原因是为了解决RNN模型在捕捉长期依赖时遇到的梯度消失或爆炸问题。GRU模块通过引入门控机制，使得模型能够更好地捕捉长期依赖，从而提高模型的性能。



该文章主要做评估了长短期记忆（LSTM）单元、门控循环单元（GRU）以及更传统的tanh单元在序列建模任务上的表现并得出结论：在某些数据集上，通过对所有模型使用固定数量的参数，GRU在CPU时间收敛性、参数更新和泛化能力方面都能胜过LSTM单元。

文章大致介绍了RNN的设计思想，在这里简要介绍：

RNN是一种扩展了传统前馈神经网络的模型，能够处理变长序列的输入。它通过在每个时间步上使用一个循环隐藏状态来处理变长序列，该隐藏状态的激活依赖于前一个时间步的状态。具体而言，对于一个序列 ，RNN更新其循环隐藏状态。通常，循环隐藏状态的更新可以用下式表示：



一个生成式的 RNN 输出给定当前状态下一序列元素的概率分布，这个生成模型能够通过使用特殊的输出符号表示序列的结束来捕获变长序列的分布。序列的概率可以分解为：



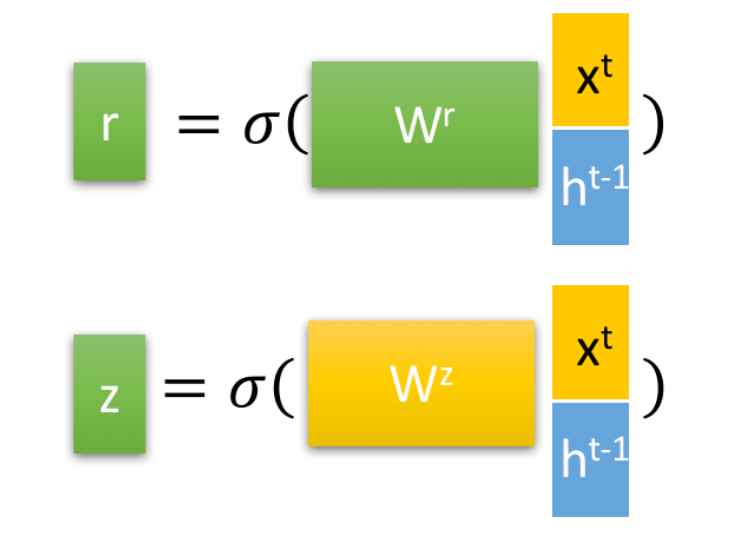
其中最后一个元素是特殊的序列结束值。我们使用如下方式对每个条件概率分布进行建模：



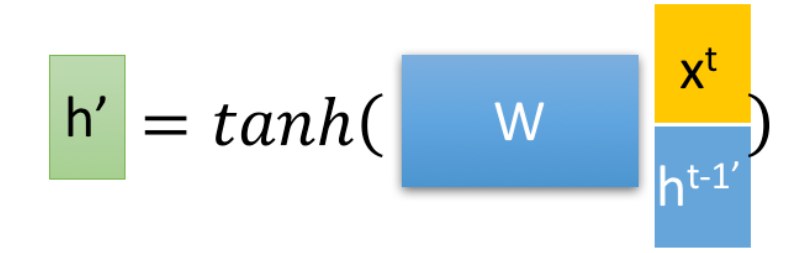
循环神经网络的训练存在一个挑战，即难以捕获长期依赖关系，因为梯度往往会消失（大多数情况下）或激增（很少，但影响严重）。为了克服这个问题采用了两种主要方法：一种方法是设计更好的学习算法，如简单的梯度裁剪、使用更复杂的二阶方法等。另一种方法是设计更复杂的激活函数，如门控单元。最早的尝试是长短期记忆（LSTM）单元，最近另一种叫做门控循环单元（GRU）的循环单元也被提出。

文章同时介绍了LSTM单元，关于GRU模块（后面也提到了区别）更新符号需要latex更加方便进行打印，同时下图可以较清晰的说明GRU模块的主要结构：

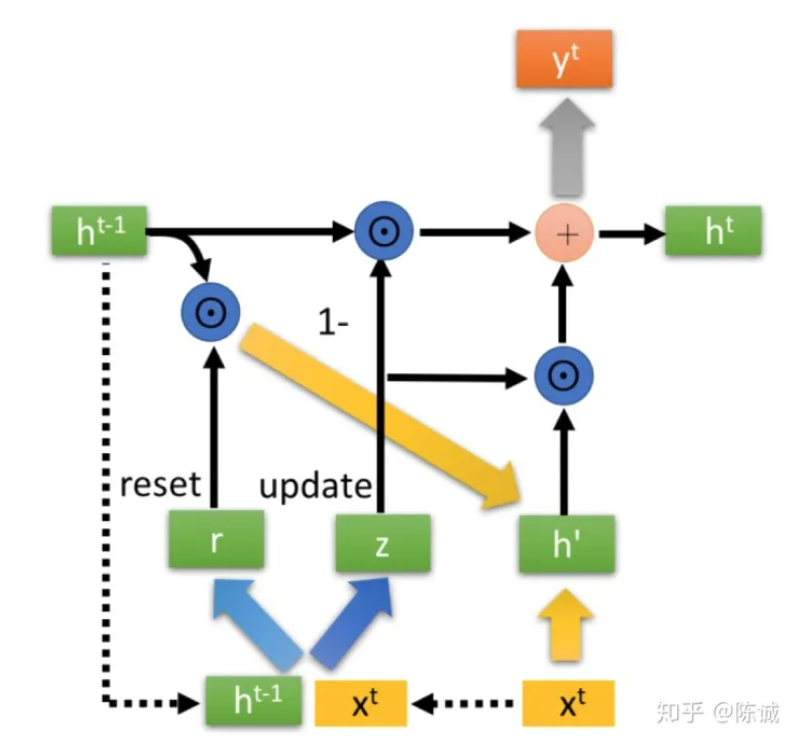
r（重置模块）、z（更新模块）门控：



隐藏状态更新：



在网上找到一张关于GRU模块的图：



而文章也指出，和LSMT的区别在于：



采用一种加权的方式进行的当前信息和已有信息的取舍。

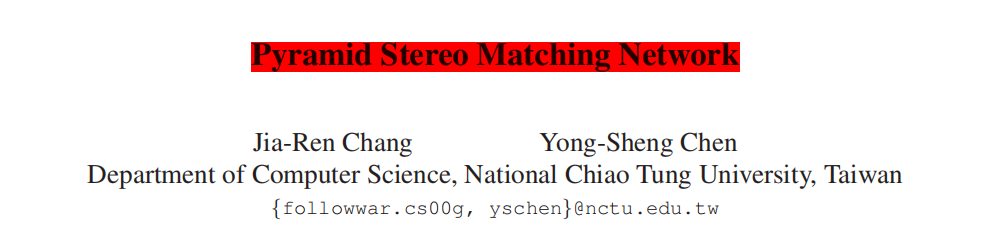
（记忆内容的受控暴露：LSTM通过输出门控制记忆内容的暴露，而GRU在没有控制的情况下完全暴露其内容）。同时，LSTM在计算新的记忆内容时没有单独的控制，而GRU在计算新的候选激活时控制了来自先前激活的信息流。

后面是具体实验的内容：

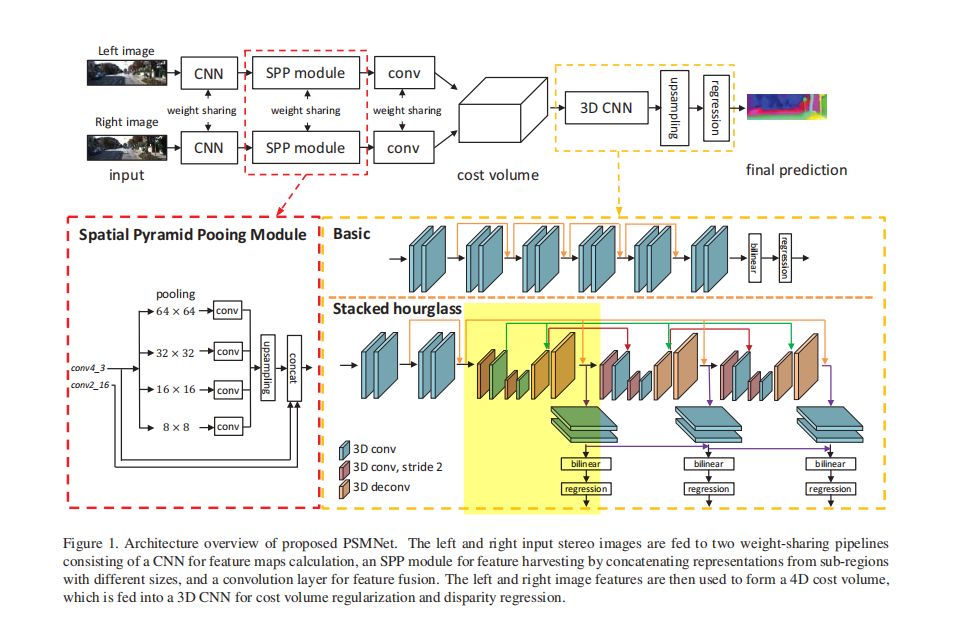
评估了LSTM、GRU和tanh单元在序列建模任务中的性能。

结果表明，门控单元（LSTM和GRU）相对传统的tanh单元具有更快的收敛速度和更好的最终解决方案。

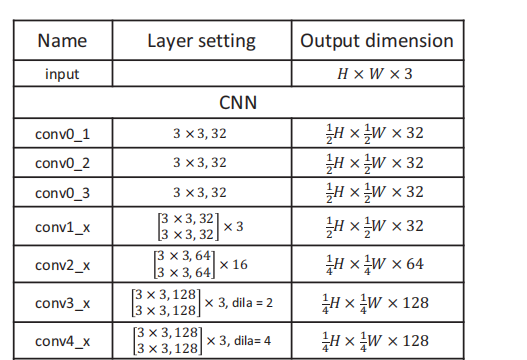
1. **Pyramid Stereo Matching Network**



主要介绍了一个比较经典的网络：

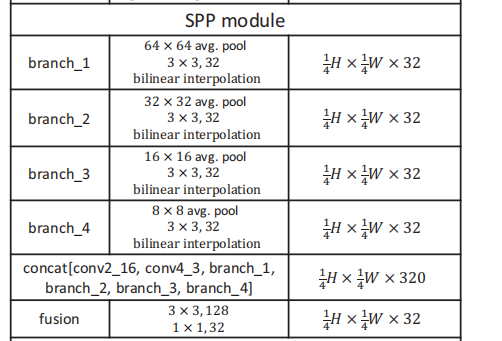


关于CNN：



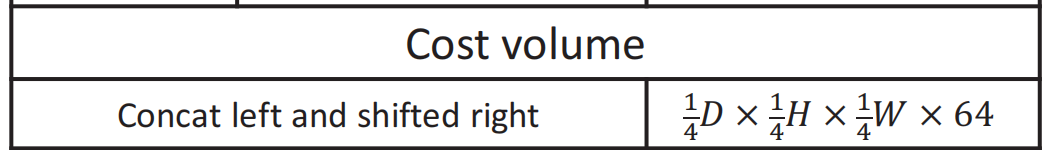
前3层是标准的卷积层，kernel\_size为3×3，channel数为32，其中conv0\_1的步长为2，将原始分辨率下采样到1/2。conv1\_x和conv2\_x采用了Resnet中经典的网络结构residual block，分别重复了3次和16次。其中，conv2\_x的通道数为64，conv2\_1的步长为2，将分辨率下采样到原始分辨率的1/4。conv3\_x和conv4\_x也采用了residual block，采用扩大卷积来进一步扩大感受野。输出特征图大小为输入图像大小的1/4×1/4，都重复了3次，channel数为128。需要注意的是，在residual block中，常规的卷积被替换为空洞卷积，膨胀率分别为2和4，用于在提升感受野的同时保持特征图分辨率不变。

关于SPP：



SSP使用了不同面积的平均池化层，覆盖面积越大，感受野越大，并且将不同层次得到的结构进行拼接，从而得到不同感受野，从局部到全局的特征。为了得到固定的特征，通常要求输入大小固定，使用不同核的最大池化层进行计算得到不同特征，通过将不同图片输入进行结合不同核得到不同结果，再将不同核得到结果拼接在一起，以满足特征多样性。通过设计池化层的大小，可以决定所有数据统一的输出大小，解决resize后会有信息丢失的问题。其通过尺寸不同的pooling可以获取位于不同全局层次的领域特征。这些特征会依次经过kernel\_size=3×3, channel=32的卷积层，然后通过上采样（双线性插值）恢复到SSP的输入特征分辨率1/4 H×1/4 W，并与CNN模块在conv2\_16和conv4\_3的输出在channel上进行拼接，得到1/4 H×1/4 W×320(64+128+32\*4)的特征图，作为SSP的输出。SSP的输出特征图既包含了不同扩大倍数感受野的特征，也包含了具有不同尺度邻域信息的特征。

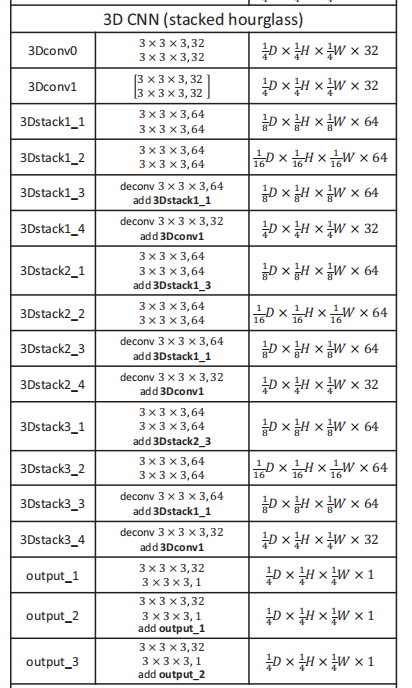
关于Cost Volume:



每个视差等级表示两个特征图之间的水平偏移量，即视差等级1对应着1个像素，视差等级2对应着2个像素，以左侧特征图为基准，将右侧特征图向左平移相应像素，共移动D次，得到D个左、右特征图的组合。将移动后的特征图在Channel上进行拼接，将右侧特征图多余的部分裁剪掉，空缺的部分补0。这样对于每一个等级的视差图，大小为1/4 H×1/4 W×64(2×32)。由于当前特征图的下采样倍数是1/4，视差等级也会进行相应比例的下采样，即一共有1/4D个大小为1/4 H×1/4 W×64的Cost Volume。可以将Cost Volume看作是一个4D Tensor，维度是1/4 H×1/4 W×1/4 D×64。

关于3D-CNN：

提出了两种用于成本体积正则化的3D CNN架构：基本架构和堆叠沙漏架构（不知道翻译是否准确）。



在基本架构中，网络简单地使用了残差块构建，包含12个3 × 3 × 3的卷积层。随后，通过双线性插值将成本体积上采样至大小为H × W × D。最后，使用回归方法计算出大小为H × W的视差图。为了获取更多上下文信息，采用了一个堆叠的沙漏（编码器-解码器）体系结构，由重复的自顶向下/自底向上处理以及中间监督组成。该堆叠沙漏架构包含三个主要的沙漏网络，每个网络生成一个视差图。因此，堆叠沙漏结构有三个输出和相应的损失函数（Loss 1、Loss 2、Loss 3），损失函数在第六步中进行描述。在训练阶段，总损失计算为这三个损失的加权和，3D CNN学习使用堆叠的多个沙漏网络并结合中间监督来规范成本量。