注意力机制与外部记忆

神经元来存储信息的容量和神经元的数量以及网络的复杂度成正比。

一方面是注意力，通过自上而下的信息选择机制来过滤掉大量的无关信息；另一方面是引入额外的外部记忆，优化神经网络的记忆结构来提高神经网络存储信息的容量。

注意力

解决信息超载问题的主要手段的一种资源分配方案，将计算资源分配给更重要的任务。

认知神经学中的注意力

一种是自上而下的有意识的注意力，称为聚焦式（focus）注意力。聚焦式注意力是指有预定目的、依赖任务的、主动有意识地聚焦于某一对象的注意力；另一种是自下而上的无意识的注意力，称为基于显著性（saliency-based）的注意力。基于显著性的注意力是由外界刺激驱动的注意，不需要主动干预，也和任务无关。

人工神经网络中的注意力机制

将最大汇聚（max pooling）、门控（gating）机制来近似地模型。看作是自下而上的基于显著性的注意力机制。除此之外，自上而下的会聚式注意力也是一种有效的信息选择方式。

用$\mathbf{X}=[x_1,x_2,\cdots,x_N]$表示 N 个输入信息，为了节省计算资源，不需要将所有的 N 个输入信息都输入到神经网络进行计算，只需要从 X 中选择一些和任务相关的信息输入给神经网络。

一是在所有输入信息上计算注意力分布，二是根据注意力分布来计算输入信息的加权平均。

注意力分布 给定一个和任务相关的查询向量 q，我们用注意力变量 z∈[1,N]查询向量 q 可以是动态生成来表示被选择信息的索引位置，即 z=i 表示选择了第 i 个输入信息。为了方便的，也可以是可学习的参数。计算，我们采用一种“软性”的信息选择机制，首先计算在给定 q 和 X 下，选择第 i 个输入信息的概率

$$\begin{aligned}\alpha_i &= p(z=i|X,\mathbf{q}) \\ &= \text{softmax}\left(s(\mathbf{x}_i,\mathbf{q})\right) \\ &= \frac{\exp\left(s(\mathbf{x}_i,\mathbf{q})\right)}{\sum_{j=1}^{N}\exp\left(s(\mathbf{x}_j,\mathbf{q})\right)}\end{aligned}$$

加性模型 $$s(\mathbf{x}_i,\mathbf{q})=\mathbf{v}^{\mathrm{T}}\tanh(W\mathbf{x}_i+U\mathbf{q}), \tag{8.2}$$

点积模型 $$s(\mathbf{x}_i,\mathbf{q})=\mathbf{x}_i^{\mathrm{T}}\mathbf{q}, \tag{8.3}$$

缩放点积模型 $$s(\mathbf{x}_i,\mathbf{q})=\frac{\mathbf{x}_i^{\mathrm{T}}\mathbf{q}}{\sqrt{d}}, \tag{8.4}$$

双线性模型 $$s(\mathbf{x}_i,\mathbf{q})=\mathbf{x}_i^{\mathrm{T}}W\mathbf{q}, \tag{8.5}$$

加性模型和点积模型的复杂度差不多，但是点积模型在实现上可以更好地利用矩阵乘积，从而计算效率更高。但当输入信息的维度 d 比较高，点积模型的值通常有比较大方差，从而导致 softmax 函数的梯度会比较小。因此，缩放点积模型可以较好地解决这个问题。双线性模型可以看做是一种泛化的点积模型。相比点积模型，双线性模型在计算相似度时引入了非对称性。

加权平均 注意力分布α_i可以解释为在给定任务相关的查询 q 时，第 i 个信息受关注的程度。我们采用一种“软性”的信息选择机制对输入信息进行汇总，

$$\begin{aligned}\mathbf{att}(X,\mathbf{q}) &= \sum_{i=1}^{N}\alpha_i\mathbf{x}_i,\\ &= \mathbb{E}_{z\sim p(z|X,\mathbf{q})}[\mathbf{x}]\end{aligned}$$

注意力机制的变体

硬性注意力（Hard Attention）

（1）一种是选取最高概率的输入信息，即

$$\mathbf{att}(X,\mathbf{q}) = \mathbf{x}_j, \tag{8.8}$$

其中j为概率最大的输入信息的下标，即$j = \arg\max_{i=1}^{N}\alpha_i$。

（2）另一种硬性注意力可以通过在注意力分布式上随机采样的方式实现。

只关注到某一个位置上的信息。硬性注意力的一个缺点是基于最大采样或随机采样的方式来选择信息，最终的损失函数与注意力分布之间的函数关系不可导，因此无法使用在反向传播算法进行训练。为了使用反向传播算法，一般使用软性注意力来代替硬性注意力。

键值对（key-value pair）

用$(K,V) = [(\mathbf{k}_1,\mathbf{v}_1),\cdots,(\mathbf{k}_N,\mathbf{v}_N)]$表示$N$个输入信息，给定任务相关的查询向量$\mathbf{q}$时，注意力函数为

$$\mathbf{att}\Big((K,V),\mathbf{q}\Big) = \sum_{i=1}^{N}\alpha_i\mathbf{v}_i, \tag{8.9}$$

$$= \sum_{i=1}^{N}\frac{\exp\Big(s(\mathbf{k}_i,\mathbf{q})\Big)}{\sum_j\exp\Big(s(\mathbf{k}_j,\mathbf{q})\Big)}\mathbf{v}_i, \tag{8.10}$$

多头注意力（Multi-Head Attention）

$$\mathbf{att}\Big((K,V),Q\Big) = \mathbf{att}\Big((K,V),\mathbf{q}_1\Big)\oplus\cdots\oplus\mathbf{att}\Big((K,V),\mathbf{q}_M\Big), \tag{8.11}$$

结构化注意力

注意力机制的应用

指针网络（Pointer Network）

一种序列到序列模型，和一般的序列到序列任务不同，这里的输出序列是输入序列的下标（索引）。

$$p(c_{1:m}|\mathbf{x}_{1:n}) = \prod_{i=1}^{m}p(c_i|c_{1:i-1},\mathbf{x}_{1:n}) \tag{8.12}$$

$$\approx \prod_{i=1}^{m}p(c_i|\mathbf{x}_{c_1},\cdots,\mathbf{x}_{c_{i-1}},\mathbf{x}_{1:n}), \tag{8.13}$$

$$p(c_i|c_{1:i-1},\mathbf{x}_{1:n}) = \mathrm{softmax}(s_{i,j}), \tag{8.14}$$

$$s_{i,j} = \mathbf{v}^{\mathrm{T}}\tanh(W\mathbf{x}_j + U\mathbf{h}_i), \forall j\in[1,n], \tag{8.15}$$

自注意力模型（Self-Attention Model）

使用神经网络来处理一个变长的向量序列时，我们通常可以使用卷积网络或

循环网络进行编码来得到一个相同长度的输出向量序列。

一种方法是增加网络的层数，通过一个深层网络来获取远距离的信息交互；另一种方法是使用全连接网络。全连接网络是一种非常直接的建模远距离依赖的模型，但是无法处理变长的输入序列。不同的输入长度，其连接权重的大小也是不同的。

$$Q = W_Q X \in \mathbb{R}^{d_3 \times N}, \tag{8.16}$$

$$K = W_K X \in \mathbb{R}^{d_3 \times N}, \tag{8.17}$$

$$V = W_V X \in \mathbb{R}^{d_2 \times N}, \tag{8.18}$$

其中 Q, K, V 分别为查询向量序列，键向量序列和值向量序列，W_Q, W_K, W_V 分别为可学习的参数矩阵。

$$\mathbf{h}_i = \mathbf{att}\Big((K, V), \mathbf{q}_i\Big) \tag{8.19}$$

$$= \sum_{j=1}^{N} \alpha_{ij} \mathbf{v}_j \tag{8.20}$$

$$= \sum_{j=1}^{N} \text{softmax}\left(s(\mathbf{k}_j, \mathbf{q}_i)\right) \mathbf{v}_j \tag{8.21}$$

如果使用缩放点积来作为注意力打分函数，输出向量序列可以写为

$$H = V \,\text{softmax}(\frac{K^{\mathrm{T}} Q}{\sqrt{d_3}}), \tag{8.22}$$

其中 softmax 为按列进行归一化的函数。

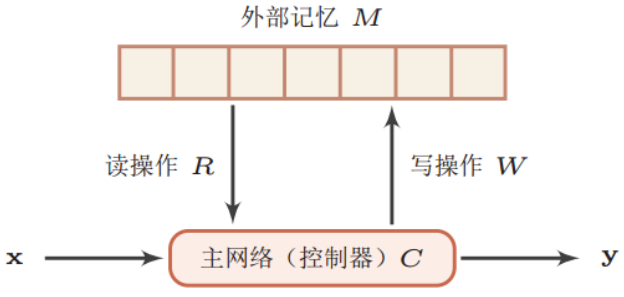
自注意力模型的权重是动态生成的，因此可以处理变长的信息序列。在单独使用时，自注意力模型一般需要加入位置编码信息来进行修正

外部记忆（External Memory）

结构化的外部记忆

引入结构化的记忆模块，将和任务相关的短期记忆保存在记忆中，需要时再进行读取。

结构化的外部记忆是带有地址的，即每个记忆片段都可以按地址读取和写入。按内容寻址通常使用注意力机制来进行。通过注意力机制可以实现一种“软性”的寻址方式，即计算一个在所有记忆片段上的分布，而不是一个单一的绝对地址。



可以将神经网络的参数和记忆容量的“分离”，即在少量增加网络参数的条件下可以大幅增加网络容量。注意力机制可以看做是一个接口，将信息的存储与计算分离。

典型的记忆网络

端到端记忆网络（End-To-End Memory Network， MemN2N）

一种可微的网络结构，可以多次从外部记忆中读取信息。在端到端记忆网络中，外部记忆单元是只读的。

给定一组需要存储的信息$m_{1:N} = \{m_1, \cdots, m_N\}$，首先将转换成两组记忆片段$A = [\mathbf{a}_1, \cdots, \mathbf{a}_N]$和$C = [\mathbf{c}_1, \cdots, \mathbf{c}_N]$，分别存放在两个外部记忆单元中，其中$A$用来进行寻址，$C$用来进行输出。

主网络根据输入 x 生成 q，并使用注意力机制来从外部记忆中读取相关信息 r，

$$\mathbf{r} = \sum_{i=1}^{N} \text{softmax}(\mathbf{a}_i^{\mathrm{T}}\mathbf{q})\mathbf{c}_i,$$

并产生输出

$$\mathbf{y} = f(\mathbf{q} + \mathbf{r})$$

多跳操作 为了实现更新复杂的计算，我们可以让主网络和外部记忆进行多轮交互。在第 k 轮交互中，主网络根据上次从外部记忆中读取的信息$\mathbf{r}^{(k-1)}$，产生新的查询向量

$$\mathbf{q}^{(k)} = \mathbf{r}^{(k-1)} + \mathbf{q}^{(k-1)}, \tag{8.27}$$

假设第k轮交互的外部记忆为$A^{(k)}$和$C^{(k)}$，主网络从外部记忆读取信息为

$$\mathbf{r}^{(k)} = \sum_{i=1}^{N} \text{softmax}((\mathbf{a}_i^{(k)})^{\mathrm{T}}\mathbf{q}^{(k)})\mathbf{c}_i^{(k)}. \tag{8.28}$$

在K轮交互后，用$\mathbf{y} = f(\mathbf{q}^{(K)} + \mathbf{r}^{(K)})$进行预测。这种多轮的交互方式也称为多跳（Multi-Hop）操作。多跳操作中的参数一般是共享的。为了简化起见，每轮交互的外部记忆也可以共享使用，比如$A^{(1)} = \cdots = A^{(K)}$和$C^{(1)} = \cdots = C^{(K)}$。

神经图灵机（Neural Turing machine， NTM）

在每个时刻t，控制器接受当前时刻的输入$\mathbf{x}_t$，上一时刻的输出$\mathbf{h}_{t-1}$和上一时刻从外部记忆中读取的信息$\mathbf{r}_{t-1}$，并产生输出$\mathbf{h}_t$，同时生成和读写外部记忆相关的三个向量：查询向量$\mathbf{q}_t$，删除向量$\mathbf{e}_t$和增加向量$\mathbf{a}_t$。然后对外部记忆$\mathcal{M}_t$进行读写操作，生成读向量$\mathbf{r}_t$，和新的外部记忆M_{t+1}。

读操作 在时刻t, 外部记忆的内容记为$M_t = [\mathbf{m}_{t,1}, \cdots, \mathbf{m}_{t,n}]$，读操作为从外部记忆$\mathcal{M}_t$中读取信息$\mathbf{r}_t \in \mathbb{R}^d$。

首先通过注意力机制来进行基于内容的寻找，即

$$\alpha_{t,i} = \text{softmax}(s(\mathbf{m}_{t,i}, \mathbf{q}_t)) \tag{8.29}$$

其中$\mathbf{q}_t$为控制器产生的查询向量，用来进行基于内容的寻址。$s(\cdot,\cdot)$为加性或乘性的打分函数。注意力分布$\alpha_{t,i}$是记忆片段$\mathbf{m}_{t,i}$对应的权重，并满足$\sum_{i=1}^{n} \alpha_{t,i} = 1$。

根据注意力分布α_t，可以计算读向量（read vector）$\mathbf{r}_t$作为下一个时刻控制器的输入。

$$\mathbf{r}_t = \sum_{i=1}^{n} \alpha_i \mathbf{m}_{t,i}. \tag{8.30}$$

写操作 外部记忆的写操作可以分解为两个子操作：删除和增加。

首先，控制器产生删除向量（erase vector）$\mathbf{e}_t$和增加向量（add vector）$\mathbf{a}_t$，分别为要从外部记忆中删除的信息和要增加的信息。

删除操作是根据注意力分布来按比例地在每个记忆片段中删除$\mathbf{e}_t$，增加操作根据注意力分布来进行按比例地给每个记忆片段加入$\mathbf{a}_t$。

$$\mathbf{m}_{t+1,i} = \mathbf{m}_{t,i}(\mathbf{1} - \alpha_{t,i}\mathbf{e}_t) + \alpha_{t,i}\mathbf{a}_t, \forall i \in [1, n]. \tag{8.31}$$

通过写操作得到下一时刻的外部记忆$\mathcal{M}_{t+1}$。

基于神经动力学的联想记忆

联想记忆模型（Associative Memory Model）主要是通过神经网络的动态演化来进行联想，有两种应用场景：1）输入的模式和输出的模式在同一空间，这种模型叫做自联想记忆模型（Auto-Associative Model）。自联想模型可以通过前馈神经网络或者循环神经网络来实现，也经常称为自编码器（Auto-Encoder）；2）输入的模式和输出的模式不在同一空间，这种模型叫做异联想记忆模型（Hetero-Associative Model）。

Hopfield 网络

所有神经元都相互连接的不分层的神经网络。每个神经元既是输入单元，又是输出单元，没有隐藏神经元。一个神经元和自身没有反馈相连，不同神经元之间连接权重是对称的。

假设一个 Hopfield 网络有 m 个神经元，第 i 个神经元的更新规则为

$$s_i = \begin{cases} +1 & \text{if } \sum_{j=1}^{m} w_{ij}s_j + b_i \geq 0, \\ -1 & \text{otherwise}, \end{cases} \tag{8.32}$$

异步更新是每次更新一个神经元。神经元的更新顺序可以是随机或事先固定的。同步更新是指一次更新所有的神经元，需要有一个时钟来进行同步。

能量函数

每个不同的网络状态定义一个标量属性，称为“能量”。

$$E = -\frac{1}{2}\sum_{i,j} w_{ij}s_i s_j - \sum_i b_i s_i \tag{8.35}$$

$$= -\frac{1}{2}\mathbf{s}^{\mathrm{T}}W\mathbf{s} - \mathbf{b}^{\mathrm{T}}\mathbf{s}. \tag{8.36}$$

Hopfield 网络是稳定的，即能量函数经过多次迭代后会达到收敛状态（吸引点（Attractor））。权重对称是一个重要特征，因为它保证了能量函数在神经元激活时单调递减，而不对称的权重可能导致周期性震荡或者混乱。

联想记忆 Hopfield网络存在有限的吸引点（Attractor），即能量函数的局部最小点。每个吸引点$\mathbf{u}$都对应一个“管辖”区域$\mathcal{R}_{\mathbf{u}}$，如果输入向量$\mathbf{x}$落入这个区域，网络最终会收敛到$\mathbf{u}$。因此，吸引点可以看作是网络中存储的信息。将网络输入$\mathbf{x}$作为起始状态，随时间收敛到吸引点$\mathbf{u}$上的过程作为检索过程。即使输入向量$\mathbf{x}$是有部分信息或有噪声，只用其位于对应存储模式的“吸引”区域内，那么随着时间演化，网络最终会收敛到其对应的存储模式。因此，Hopfield

的检索是基于内容寻址的检索，具有联想记忆能力。

信息存储 信息存储是指将一组向量$x_1,\cdots,x_N$存储在网络中的过程。存储过程主要是调整神经元之间的连接权重，因此可以看做是一种学习过程。

如果两个神经元经常同时激活，则它们之间的连接加强；如果经常不同时激活，则连接消失。这种学习方式称为 Hebbian 法则。

使用联想记忆增加网络容量