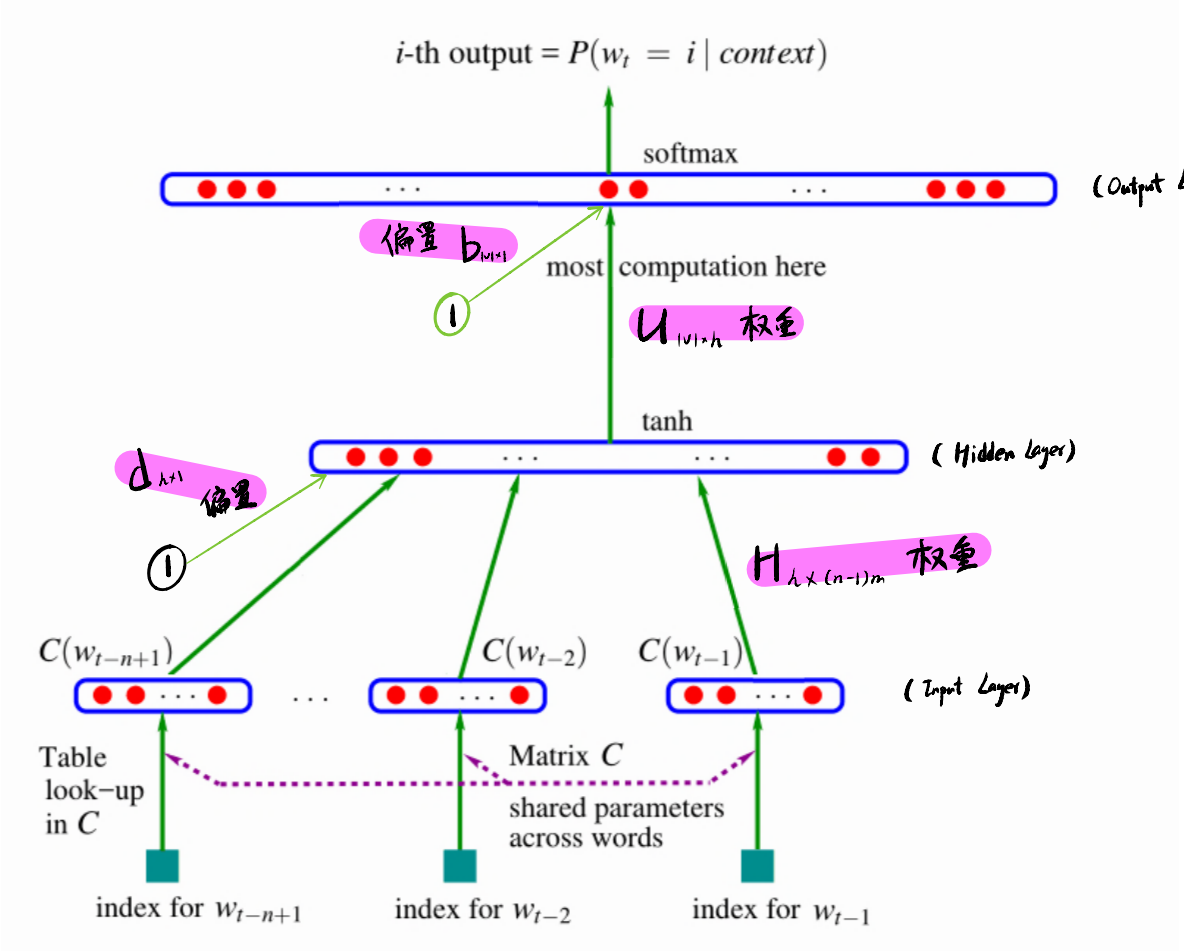
神经网络语言模型（Neural Network Language Model，NNLM）

— 2003，Bengio，《A Neural Probabilistic Language Model》
开山之作。奠定了 Deep Learning 在 NLP 领域的应用基础。

— NNLM 模型的本质，是基于神经网络训练的 N-Gram 语言模型

一个词与其他词之间的 N-Gram 关系强弱，体现在最终生成的词向量的值中

— 模型详解



步骤：

① 随机初始化字典中的每个单词的词向量 $\vec{w}_i$，一共有 $|V|$ 个单词
所有单词的词向量构成矩阵 C

$$C = \begin{pmatrix} \vec{w}_1 \\ \vec{w}_2 \\ \vec{w}_3 \\ \vdots \\ \vec{w}_{|V|} \end{pmatrix}_{|V|\times m} \longrightarrow (0.1 \quad 0.09 \quad 0.72 \quad 1.4 \cdots 0.7)_{1\times m}$$

② 将包 n-1 个词的词向量按照其在 N-Gram 中的顺序进行拼接，得到向量 x

$$x_{\underbrace{(n-1)m}_{\text{单词个数}}\times 1} = (w_{t-n+1}, w_{t-n+2}, \cdots, w_{t-2}, w_{t-1})$$

每个单词的词向量维数

※ 对向量拼接时的顺序，论文中给出的图示与公式并不一致。
是 $(w_{t-n+1} \to w_{t-1})$ 还是 $(w_{t-1} \to w_{t-n+1})$，理论上都可以。

③ 将向量 x 代入 Hidden Layer 的激活函数 tanh

Hidden Layer → Oupt Layer 的偏置

Hidden Layer → Output Layer 的权重

Input Layer → Hidden Layer 的权重

$$y_{|V|\times 1} = b_{|V|\times 1} + U_{|V|\times h}\tanh\left(d_{h\times 1} + H_{h\times(n-1)m}\, x_{(n-1)m\times 1}\right)$$

(n-1)个单词的词向量拼接所得

Input Layer → Hidden Layer 的偏置

给定 $w_{t-n+1}, w_{t-n+2}, \cdots, w_{t-2}, w_{t-1}$，预测 w_t

字典中的每一个词都有可能，但可能性不同

（所以 y 是由 $|V|$ 个值构成的向量）

使它的值最大

④ 使用 Softmax 构建损失函数

从向量 $y_{|V|\times 1}$ 从取单词 w_t 对应的值

$$\max_{\theta} P(w_t \mid w_{t-n+1}, w_{t-n+2}, \cdots, w_{t-2}, w_{t-1}) = \max_{\theta} \frac{e^{y_{w_t}}}{\sum_{i}^{|V|} e^{y_{w_i}}}$$

从向量 $y_{|V|\times 1}$ 从取单词 w_i 对应的值

遍历字典中所有单词

找到合适的参数 θ，使得表达式的值最大

θ 是一组模型参数，具体为上文的 C, H, d, U, b

⑤ 通过随机梯度上升（Stomastic Gradient Ascent）迭代求得最佳参数值

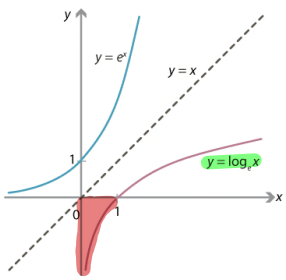
因为是梯度上升

因为这里得到的概率值 P 可能会非常小，使用 log 函数将不同 N-Gram 间的概率差异放大

$$\theta \longleftarrow \theta + \varepsilon \frac{\partial \log P(w_t \mid w_{t-n+1}, w_{t-n+2}, \cdots, w_{t-2}, w_{t-1})}{\partial \theta}$$

小到计算机无法处理，产生 underflow 错误

学习率



注意：我根据自己的理解，对论文中的图示进行了简化，对公式进行了修改。
可能有误。