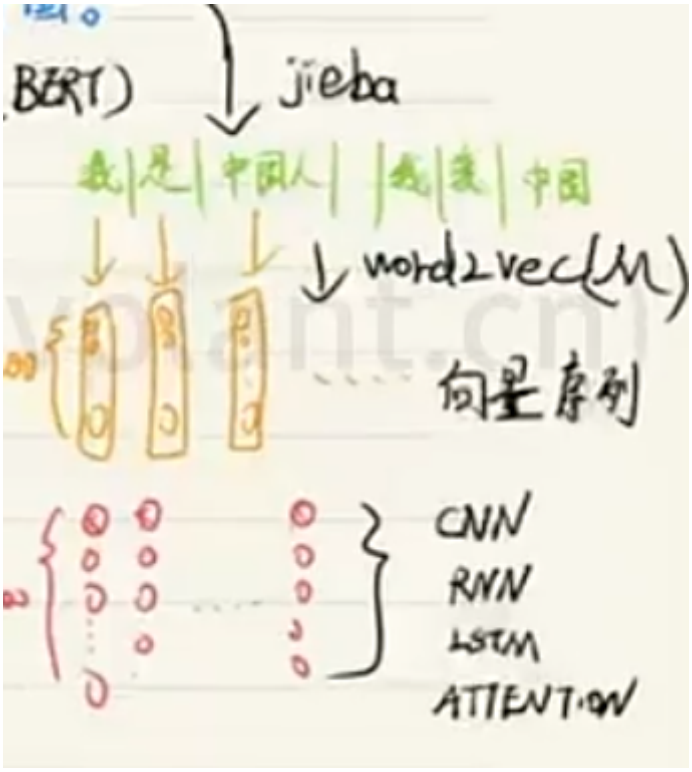
文本处理

1.传统方法



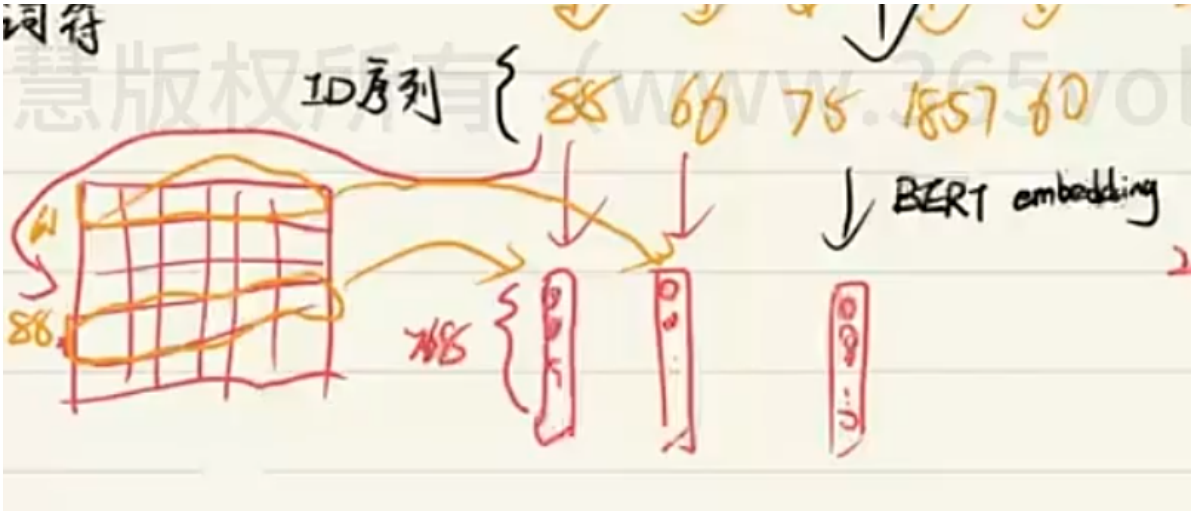
jieba分词器处理后，Word2vec处理，然后转换为向量序列使用CNN、RNN、LSTM等技术进行处理训练。

Word2vec是一群用来产生词向量的相关模型。这些模型为浅层双层的神经网络，用来训练以重新建构语言学之词文本。网络以词表现，并且需猜测相邻位置的输入词，在word2vec中词袋模型假设下，词的顺序是不重要的。

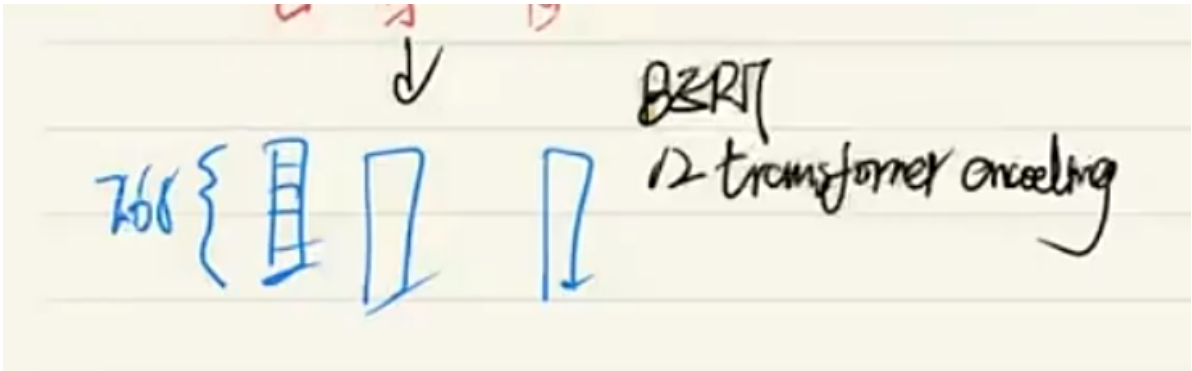
2.新方法

GPT、Bert等模型，以BERT为例：

使用模型对应分词器分词，然后查id表得到id序列。



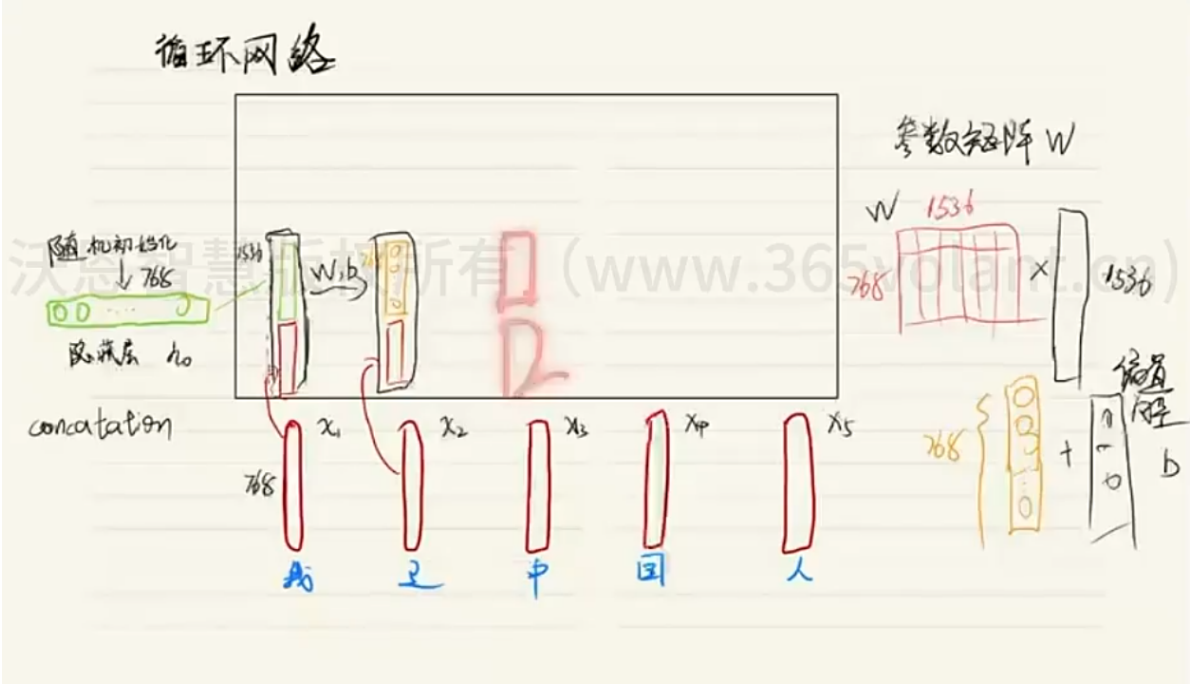
用id找表，得到一个矩阵（768维）



维度都不变

最后加一个全连接层（分类器）

循环网络（RNN）



参数有：隐藏层h_0和x_n对应长度，参数矩阵w，还有偏置向量b

在循环神经网络（Recurrent Neural Network，RNN）中，参数矩阵是通过模型的训练过程来学习得到的。这些参数矩阵包括输入到隐藏层的权重矩阵、隐藏层到隐藏层的权重矩阵和隐藏层到输出层的权重矩阵，以及对应的偏置向量。

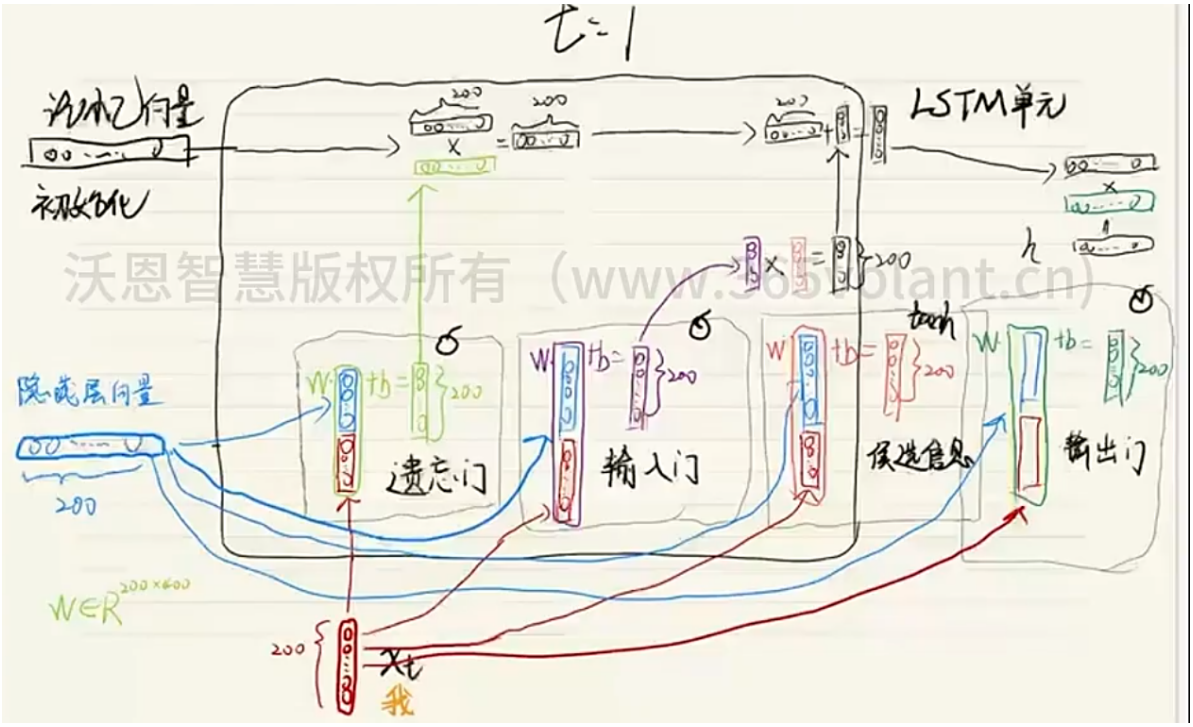
训练RNN的过程通常使用反向传播算法（Backpropagation）和梯度下降优化算法来更新参数。具体步骤如下：

1. 初始化参数矩阵：在训练开始之前，需要随机初始化参数矩阵，通常使用较小的随机值。
2. 前向传播：将训练数据输入到RNN中，沿着时间序列进行前向传播，计算隐藏层和输出层的值。
3. 计算损失函数：将模型预测的输出与真实标签进行比较，计算损失函数（例如均方误差）来衡量模型的预测误差。
4. 反向传播：通过反向传播算法，从输出层向隐藏层传播误差，计算参数的梯度。反向传播算法根据链式法则来计算每一层的梯度，然后将梯度传递回每个时间步。
5. 参数更新：使用梯度下降优化算法（如随机梯度下降），根据梯度的方向和学习率来更新参数矩阵。梯度下降算法通过迭代的方式逐渐调整参数值，使得损失函数最小化。
6. 重复步骤2至步骤5：重复执行前向传播、损失计算、反向传播和参数更新的步骤，直到达到预定的训练轮数或达到收敛条件。

输出是一个矩阵（h_n组成的矩阵），根据后续任务选择后面不同处理

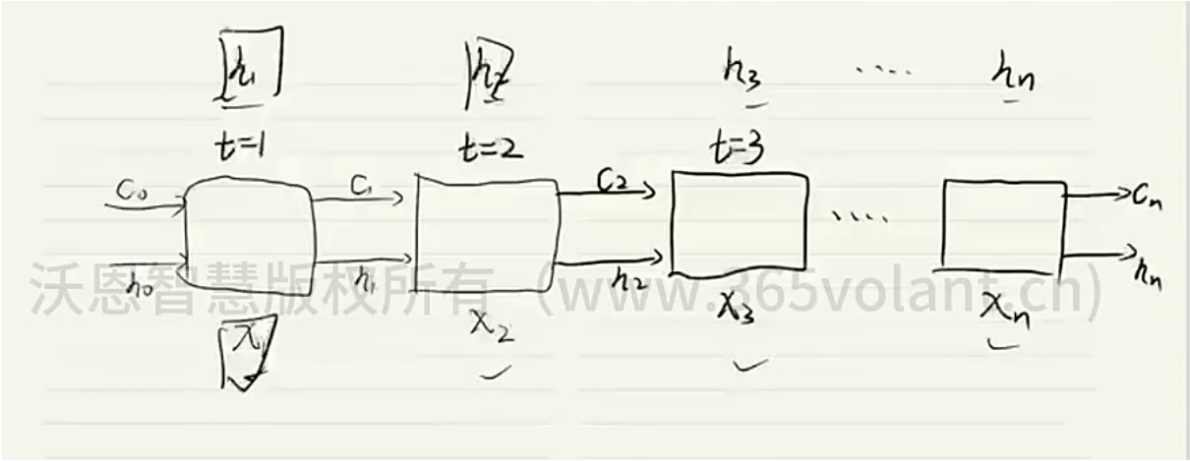
LSTM（长短期记忆网络）

有非常多的权重矩阵w



sigmoid函数是一种常用的激活函数，其平滑的特性使得它适用于表示概率或控制门的输出。在LSTM中，除了遗忘门，输入门（Input Gate）和输出门（Output Gate）也使用sigmoid函数来进行门控操作。此外，LSTM还使用双曲正切函数（tanh函数）来生成候选单元状态和输出值。

（图中×号表示对应位置的向量相乘），图中展示了下一时间使用的隐藏层向量是怎么生成的，没有与输出门向量对应位置向量相乘的向量则作为下一时间的**记忆向量**



h_1到h_n组成了一个新的矩阵，可以用其进行下一步计算，也可以只用h_n进行下一步计算