词汇的星空

本次作业要求使用word2vec技术来实现一个简易的翻译模型。

思路和模型

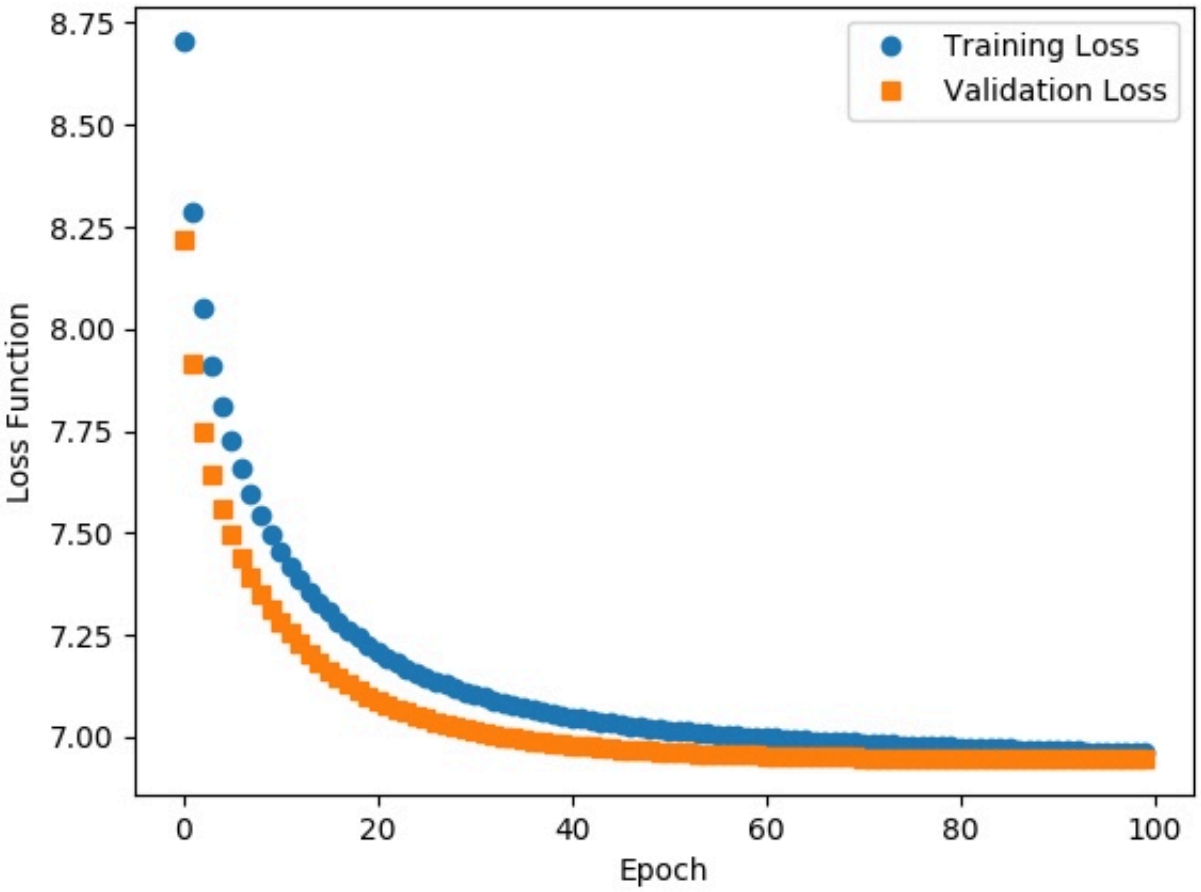
本次作业的大体思路是先使用英文的词向量来作为神经网络的输入，中文的词向量来作为输出。
最后比较输出的词向量与真实的词典中的结果做匹配计算，分别考虑全词匹配和单字匹配。

这里最基本的神经网络模型是使用三层Linear，各层含有的神经元数量分别是100，30和200，其中隐含层输入使用了Tanh作为激励函数。

在实验中，除了这个模型之外，我们还继续尝试使用更深的网络，更多的隐含层单元，以及其他的激励函数。

实验结果

首先我们来看看默认的模型训练和测试结果：



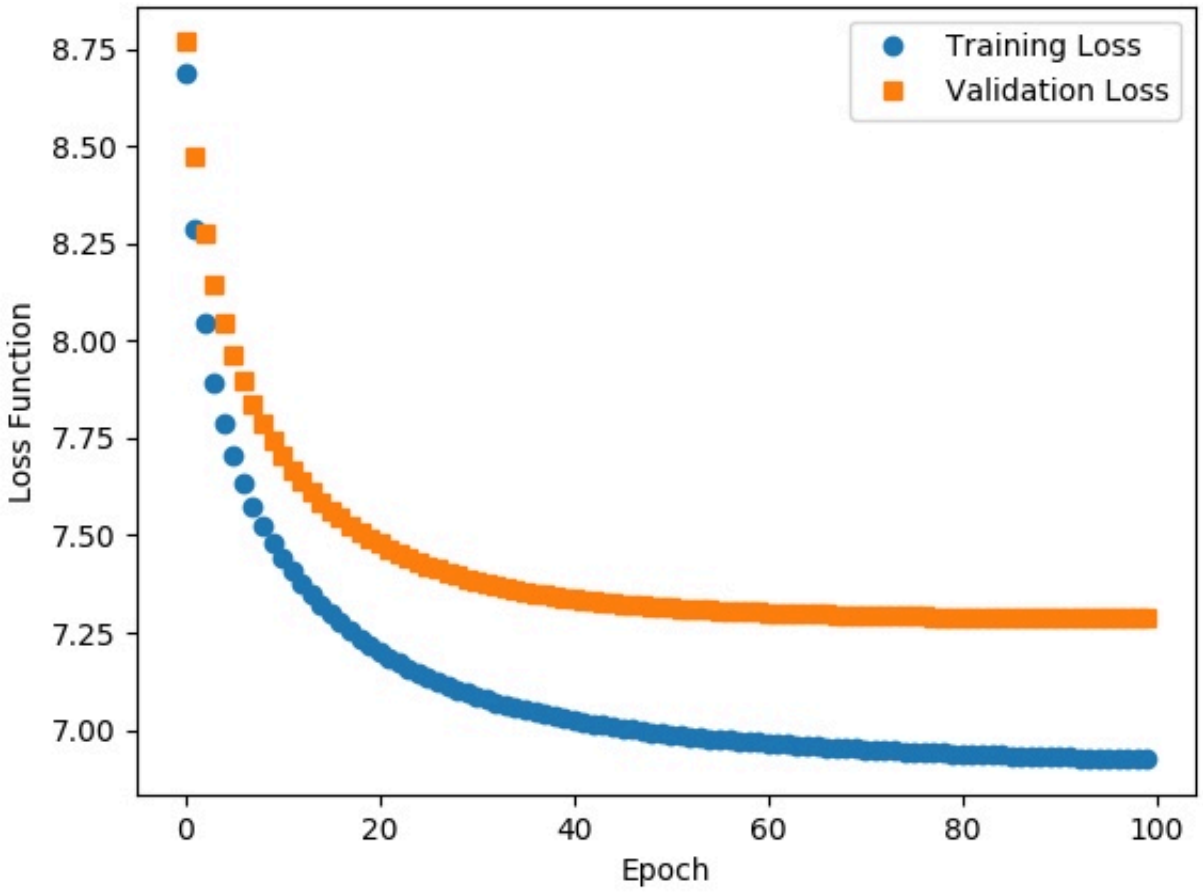
我们可以看到，Training loss 和 Validation loss 在 100 个 epoch 的时候两者已经非常接近了，我们继续看看当时的loss具体数值：

- 0轮，训练Loss: 8.70, 校验Loss: 8.22
- 99轮，训练Loss: 6.96, 校验Loss: 6.95

可以发现的确如此，第 100 个 epoch 的时候 ， 训练loss和校验的loss已经相差无几。最后让我们看看测试的匹配情况：

- 精确匹配率：0.10
- 一字匹配率：0.18

那么能不能有一些方法再提高一些呢，我们试图改一改激活函数吧。我们将Tanh改成了ReLU再次进行试验，得到训练和校验时的loss情况如下：



具体loss数值：

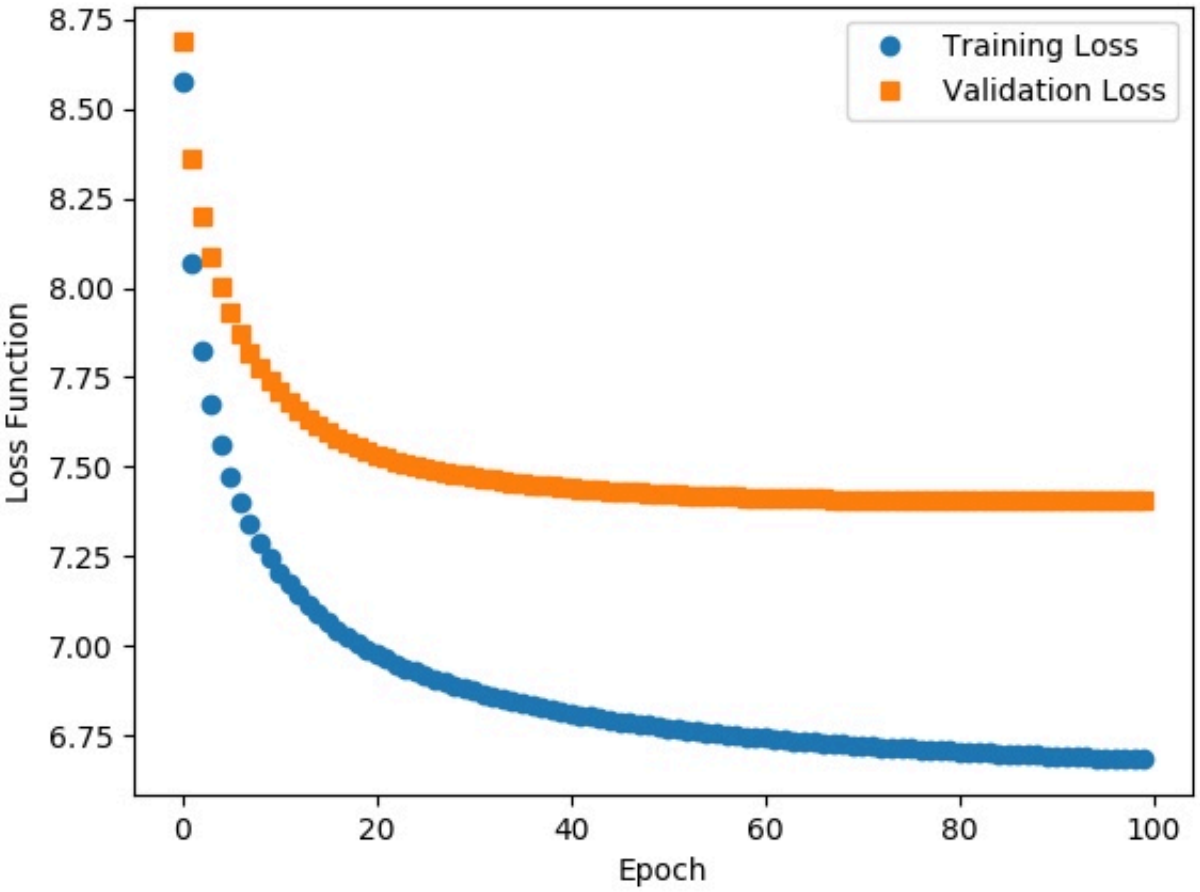
- 0轮，训练Loss: 8.69, 校验Loss: 8.77, 2017-11-12T10:50:50.213229+08:00
- 99轮，训练Loss: 6.92, 校验Loss: 7.29, 2017-11-12T10:57:37.192032+08:00
- 精确匹配率：0.09
- 一字匹配率：0.20

在将激活函数从Tanh改成了ReLU之后，从测试结果来看匹配率其实是不相上下，可以认为在这样的数值类型中，ReLU和Tanh两种常见的激活函数对结果的影响并不大。

那么可以继续尝试改变隐含层的神经元个数：

我们继续使用Tanh作为激活函数，将隐含层的神经元个数从30分别改成50，100和300。

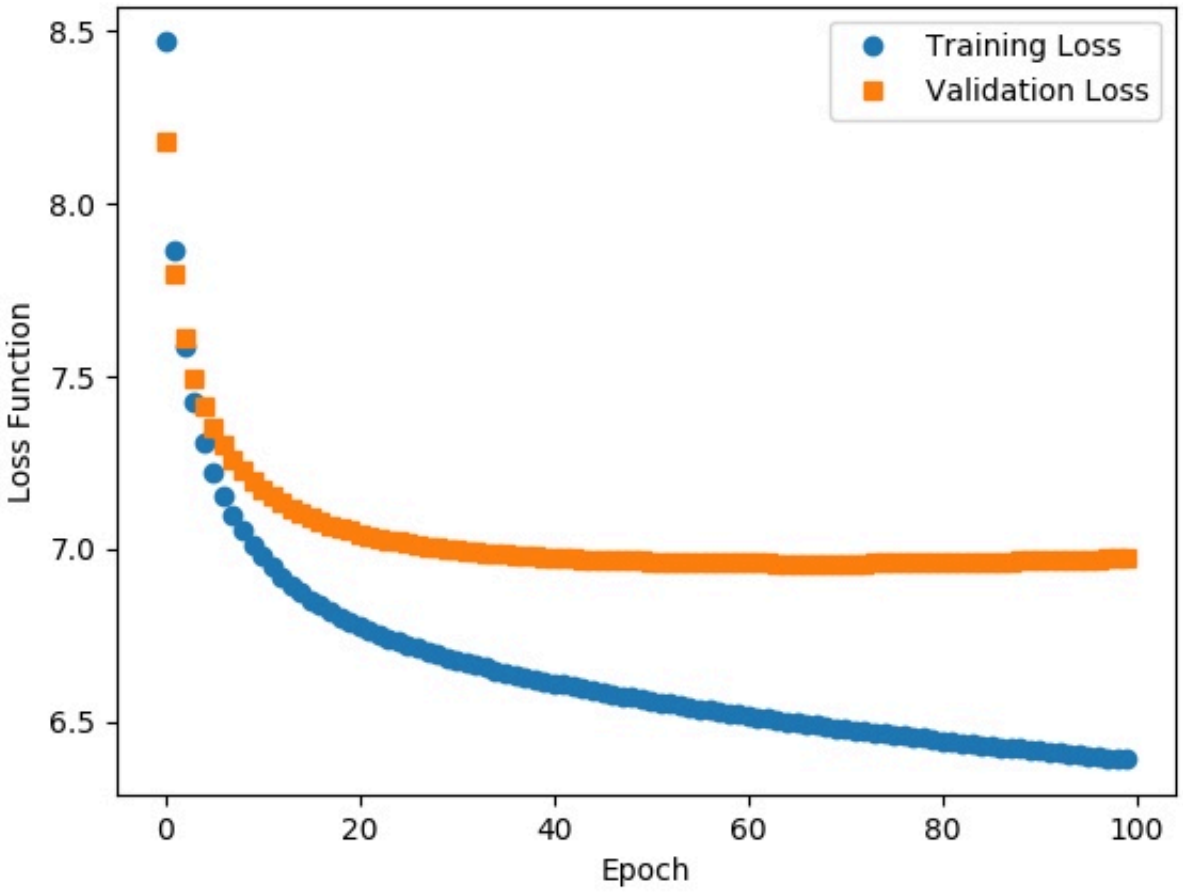
首先是隐含层使用50个神经元的：



具体的 Loss 情况如下：

- 0轮，训练Loss: 8.57, 校验Loss: 8.69, 2017-11-12T01:31:19.704521+08:00
- 99轮，训练Loss: 6.68, 校验Loss: 7.41, 2017-11-12T01:37:31.986376+08:00
- 精确匹配率：0.11
- 一字匹配率：0.22

继续增加，在隐含层使用100个神经元。

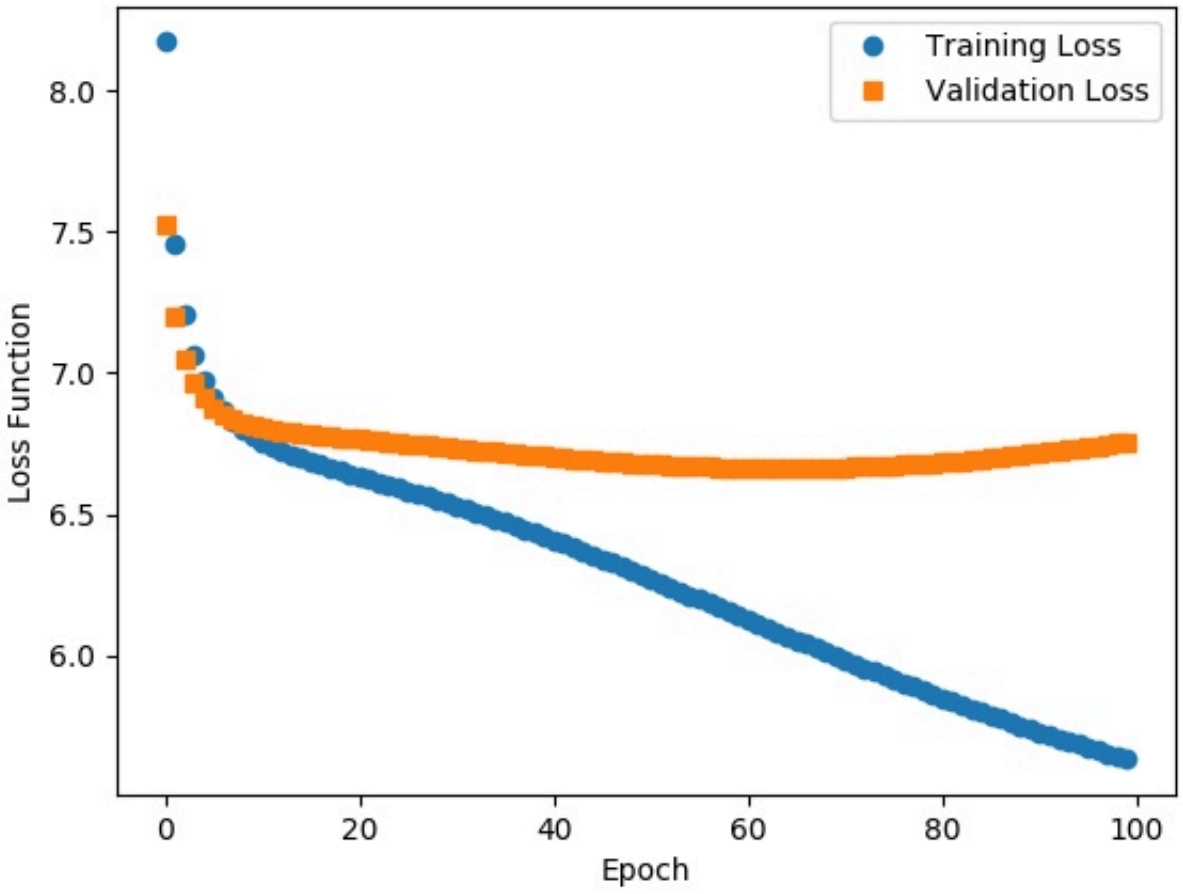


- 0轮，训练Loss: 8.47, 校验Loss: 8.18, 2017-11-12T01:41:50.216397+08:00
- 99轮，训练Loss: 6.39, 校验Loss: 6.97, 2017-11-12T01:48:17.482972+08:00

我们可以看到其实在最后几个epoch的时候，虽然训练loss依然在持续下降，但是校验loss已经开始有增大的趋势，表明已经开始有过拟合的现象发生。我们将实验过程中校验最低时候的模型取出做测试，得到精确匹配和一字匹配的结果：

- 精确匹配率：0.18
- 一字匹配率：0.32

最后，我们继续加大神经元的数量，我们在隐含层使用300个神经元。



果不其然，更多的隐层神经元会使得模型的捕捉能力大大增强，更快地进入了过拟合状态。下面是训练过程中的Loss情况，其中大约在第80个epoch的时候，校验Loss降到了最低点。

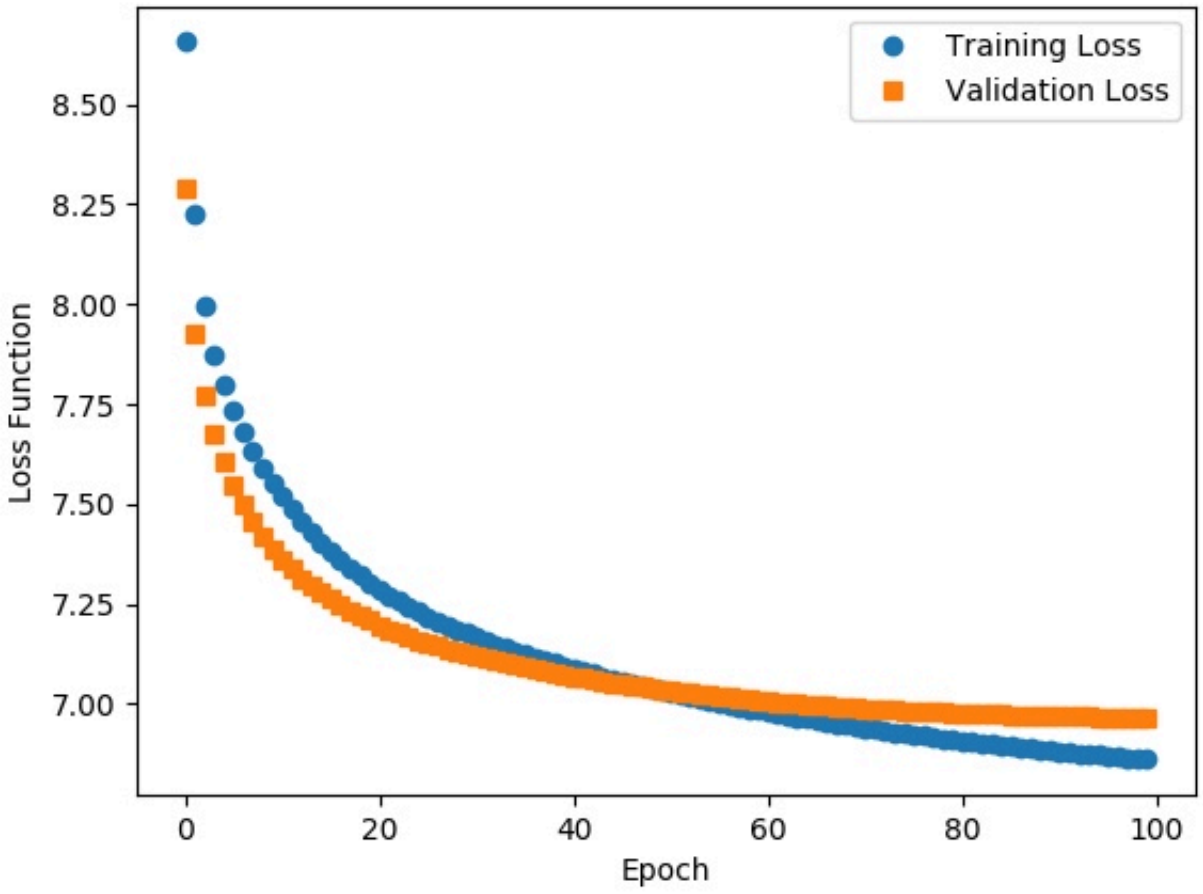
- 0轮，训练Loss: 8.17, 校验Loss: 7.53, 2017-11-12T01:59:15.320790+08:00
- 80轮，训练Loss: 5.85, 校验Loss: 6.68, 2017-11-12T02:04:25.084537+08:00
- 99轮，训练Loss: 5.63, 校验Loss: 6.76, 2017-11-12T02:05:37.470264+08:00

我们用第80个epoch时候的模型，做测试：

- 精确匹配率：0.13
- 一字匹配率：0.30

三组不同的隐含层神经元数量的实验表明，更多的隐层神经元其实并不能让模型变得更好。那么如果网络的结构发生变化呢？比如是否可以尝试更多层的隐层？

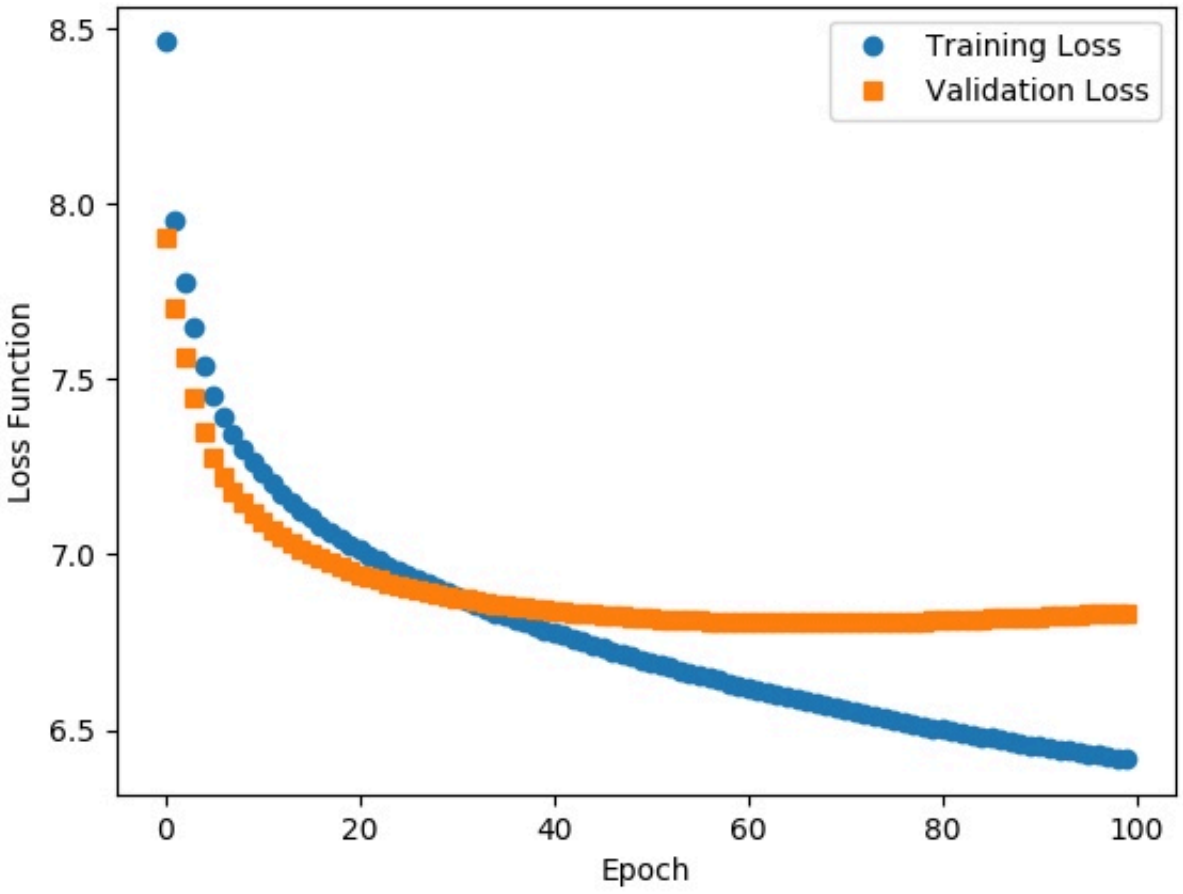
下面首先我们将使用三层的网络，但是依然是全连接层。首先是 100-30-50-200 的网络结构，中间两层使用Tanh作为激活函数。



相较之前的实验，我们发现校验Loss在初始阶段下降得更快了，一度比训练Loss还要低，大约在50个epoch个时候两者产生交点。下面是具体的Loss数值：

- 0轮，训练Loss: 8.46, 校验Loss: 7.91, 2017-11-12T11:30:10.138245+08:00
- 99轮，训练Loss: 6.42, 校验Loss: 6.84, 2017-11-12T11:38:38.983010+08:00
- 精确匹配率：0.09
- 一字匹配率：0.18

下面我们在隐层使用更多的神经元个数，分别是 100-50-100-200



我们发现和前面的情形有些类似，都是在初始阶段校验Loss就下降得非常厉害，大约在30个epoch的时候发生交叉，随后大约在70个epoch时发生过拟合现象，校验Loss开始上升。

- 0轮，训练Loss: 8.66, 校验Loss: 8.29, 2017-11-12T11:40:04.555003+08:00
- 70轮，训练Loss: 6.56, 校验Loss: 6.81, 2017-11-12T11:36:09.917833+08:00
- 99轮，训练Loss: 6.86, 校验Loss: 6.97, 2017-11-12T11:48:53.568733+08:00
- 精确匹配率：0.12
- 一字匹配率：0.20

总结

1. 使用ReLU来替换Tanh在本实验中没有提升
2. 使用更多的隐含层单元是可以让模型具有更强的捕捉能力，但是会提早进入过拟合状态
3. 使用更深层网络结构在本例中几乎没有提升，但是却可以让模型训练时候的泛化能力更好