NLL Curves

January 21, 2016

1 NLL Curves

In [5]:
```
%matplotlib inline
import numpy as np
import scipy as sp
import matplotlib as mpl
import matplotlib.cm as cm
import matplotlib.pyplot as plt
import pandas as pd
pd.set_option('display.width', 500)
pd.set_option('display.max_columns', 100)
pd.set_option('display.notebook_repr_html', True)
import seaborn as sns
sns.set_style("whitegrid")
sns.set_context("poster")
```

In [6]:
```
with open('learning_rates.txt', 'r') as f:
    lines = f.readlines()
```

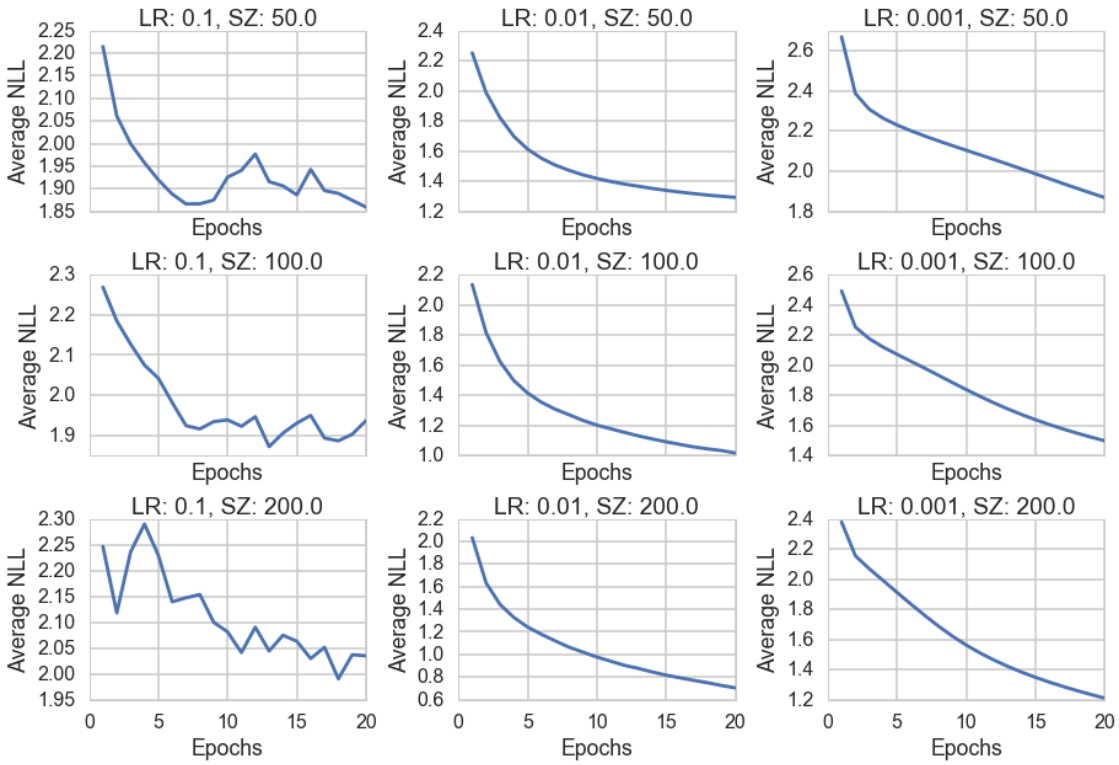
In [7]:
```
values = []
for l in lines:
    if "," in l:
        values.append(map(float, l.split(",")))
    else:
        values.append(float(l))

learning_rates = []
sizes = []
nlls = []
for idx, v in enumerate(values):
    if idx % 21 == 0:
        learning_rates.append(v[0])
        sizes.append(v[1])
    elif idx % 21 == 1:
        nlls.append(values[idx:idx+20])
    else:
        pass
```

1.1 Graph training error as a function of average NLL over epochs

1.1.1 LR = learning rate {0.00001, 0.0001, 0.001}

1.1.2 SZ = size of the hidden layer and the embedding size {50, 100, 200}

```
In [9]: f, ax = plt.subplots(3,3, sharex=True)
        X = range(1, 21)
        for i in range(len(nlls)):
            a = ax[i / 3][i % 3]
            a.plot(X, nlls[i])
            a.set_title("LR: %s, SZ: %s" % (learning_rates[i], sizes[i]))
            a.set_ylabel("Average NLL")
            a.set_xlabel("Epochs")
        plt.tight_layout()
```



1.1.3 Conclusion

A lower learning rate (< 1e-5) is optimal. Embedding sizes alter the steepness of the NLL curve, but after 20 epochs average NLL is near 2.6 in all cases.