Exercises Week 4
Natural Language Processing and Deep Learning

Jes Frellsen
IT University of Copenhagen

Fall 2018

We will make use of TensorFlow for all these exercises. You can install TensorFlow using either pip (`pip install tensorflow`) or Anaconda (`conda install tensorflow`).

Exercise 1

The purpose of the first exercise is to get more familiar with TensorFlow.

(a) Download Notebook 1 from the lecture.

(b) Change the code, such that you fit a 4th degree polynomial instead.

Exercise 2

In this exercise we will build a Convolutional Neural Network (CNN) for predicting the secondary structure of a protein based on its amino acid sequence. We will be inspired on the CNN implemented by Wang et al. (2016); however we will not include the conditional random field (CRF) extension described in this paper.

Dataset

We will make use of a dataset created by Zhou and Troyanskaya (2014). I have preprocessed the dataset, which can be downloaded from:

`https://www.dropbox.com/s/jpszfjp86j0apxp/cullpdb%2Bprofile_6133_ss3-preprocessed.npz?dl=0`

This is a compressed binary file in NumPy format that can be loaded using:

```
data = np.load('cullpdb+profile_6133_ss3-preprocessed.npz')
X_train = data['X_train']
y_train = data['y_train']
X_validation = data['X_validation']
y_validation = data['y_validation']
X_train = data['X_test']
y_train = data['y_test']
```

The data set is already split into a training set (5,600 proteins), a validation set (256 proteins) and a test set (272 proteins).

The input array (`X`) has three dimensions. This first dimension is the index of the protein. The second dimension is the amino acid position in the protein, which is always assumed to be 700. The last dimension contains a one-hot encoding of the 20 amino acid identity and evolutionary profile obtained from a multiple alignment. Both the one-hot encoding and the profile have an additional 21st character called 'X' which represent an unknown amino acid and a 22nd character called `'NoSeq'`, which is used as padding for proteins with less than 700 amino acids. The two last dimensions of the input array are illustrated below:

	A	C	E	D	G	F	I	H	K	M	L	N	Q	P	S	R	T	W	V	Y	X	-	
AA 1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
AA 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
AA 3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
⋮											⋮												... (1)
AA 698	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
AA 699	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	

	A	C	E	D	G	F	I	H	K	M	L	
	0.00	0.00	0.01	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
	0.67	0.00	0.08	1.00	0.01	0.01	0.04	0.00	0.27	0.01	0.01	
	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.82	0.23	0.14	
...						⋮						... (2)
	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	

	N	Q	P	S	R	T	W	V	Y	X	-	
	0.00	0.00	0.81	0.00	0.00	0.00	0.00	0.00	0.27	0.00	0	
	0.04	0.00	0.96	0.86	0.04	0.07	0.00	0.00	0.27	0.01	0	
	0.00	0.00	0.80	1.00	0.00	0.00	0.00	0.05	0.27	0.00	0	
...						⋮						(3)
	0	0	0	0	0	0	0	0	0	0	1	
	0	0	0	0	0	0	0	0	0	0	1	

Here (1) is the one-hot encoding of the 20 amino acids and the `'NoSeq'` (shown as -). The 11 first amino acids of the profile are shown in (2), and the remaining of the profile is shown in (3). In this example, the first amino acid (AA 1) is an Glutamic Acid (E), the second and the thirds are Arginine (R). The two last rows (AA 698 and AA 699) are encoded as `'NoSeq'` in both the one-hot encoding and the profile.

The output array (`y`) is also has three dimensions. This first dimension is again the index of the protein, and the second dimension is the amino acid position in the protein (assumed to of size 700). The last dimension contains a one-hot encoding of the secondary structure class: helix (`H`), sheet/extended (`E`), coil (`C`) and `'NoSeq'` (`-`). The

two last dimensions of the output array are illustrated below:

$$
\begin{array}{c}
 \\
\text{AA 1} \\
\text{AA 2} \\
\text{AA 3} \\
\vdots \\
\text{AA 698} \\
\text{AA 699}
\end{array}
\overbrace{\left\{\begin{bmatrix}
\text{H} & \text{E} & \text{C} & \text{-} \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
 & & \vdots & \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}\right.}^{}
\cdots \tag{4}
$$

In this example, the first amino acid (AA 1) is classified as coil, and the second and the third are in an helix. Again, the two last amino acids are (AA 698 and AA 699) are encoded as `'NoSeq'`.

Task 1: A sensible baseline

(a) Start by calculating the frequency of each secondary structure class in the dataset.

Task 2: A fully connected model for secondary structure prediction

Take a look at the fully connected model in Notebook 2.

(b) How many parameters does the model have?

(c) What is the accuracy on the test set?

(d) How does this accuracy compare to the class frequency?

1 Task 3: A CNN for secondary structure prediction

Now implement a CNN for predicting secondary structure. Use three convolutional layers, a kernel of width 11 and 40 channels. A convolutional layer can be implemented by:

```
# The input variable
X = tf.placeholder(tf.float32, [None, 700, 44], name="X")

# The first convolutional layer
filter1_width = 11
filter1_input_size = 44
filter1_channels = 40

W1 = tf.get_variable(name="W1",
       shape=[filter1_width,
              filter1_input_size,
```

```
                filter1_channels],
        initializer=tf.contrib.layers.xavier_initializer_conv2d())
b1 = tf.get_variable("b1",
        [filter1_channels],
        initializer=tf.random_normal_initializer())
conv1 = tf.nn.conv1d(value=X,
           filters=W1, stride=1, padding='SAME')
a1 = tf.nn.bias_add(conv1, b1)
z1 = tf.nn.relu(a1)
```

Note that padding is set to `'SAME'`, which means that zero padding is applied, such that the input and output has the same size.

(e) How many parameters does the model have?

(f) What is the accuracy on the test set?

(g) How does this accuracy compare to the class frequency?

References

S. Wang, J. Peng, J. Ma, and J. Xu. Protein secondary structure prediction using deep convolutional neural fields. *Scientific reports*, 6:18962, 2016.

J. Zhou and O. Troyanskaya. Deep supervised and convolutional generative stochastic network for protein secondary structure prediction. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 745–753, Bejing, China, 22–24 Jun 2014. PMLR.