Homework 2 Report **Ricardo Rei 78047**
Deep Structured Learning (IST, Fall 2018)
Prof. Andre Martins
TAs: Vlad Niculae and Erick Fonseca Due Date: 31/10/18

Question 1

Exercise 1

After tuning the number of epochs, learning rate and regularization constant my best configuration is:

- 10 epochs
- 0.001 learning rate
- 0 weight decay (No regularization)

From Figure 1 we can observe that the model achieves 0.75% accuracy in both dev and train sets after 10 epochs. More than 10 epochs did not improve the results for the dev set and because of that I decided that 10 epochs was enough.

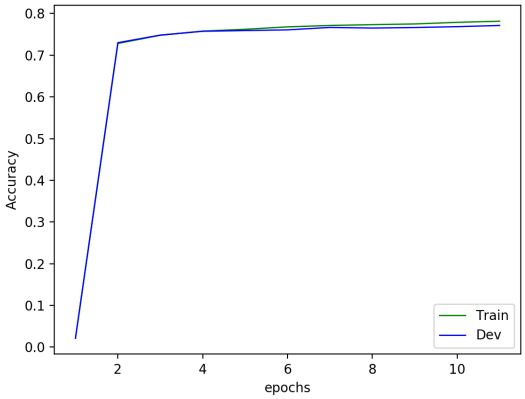

Figure 1: Train and dev accuracy for the Logistic Regression model with a learning rate of 0.001 and Adam optimizer

Exercise 2

With the following hyper-parameters:

- 20 epochs
- 0.001 learning rate
- 0 weight decay
- Sigmoid activations
- Adam optimizer

- **264 hidden layers**

I was able to achieve 92.6% in the train set, 87.2% accuracy in the dev set and 86.3% in the test set (Figure 2). The same hyper-parameters but only with 128 hidden layers achieves 81.7% dev accuracy and 86.4% rain accuracy (Figure 3).

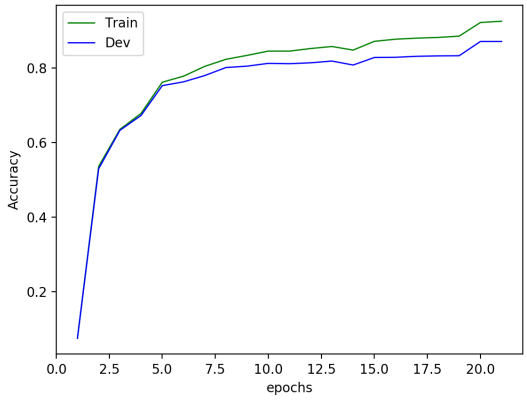

Figure 2: Train and dev accuracy using 264 hidden layers, 0.001 learning rate, no regularization and sigmoid activation's with Adam optimizer

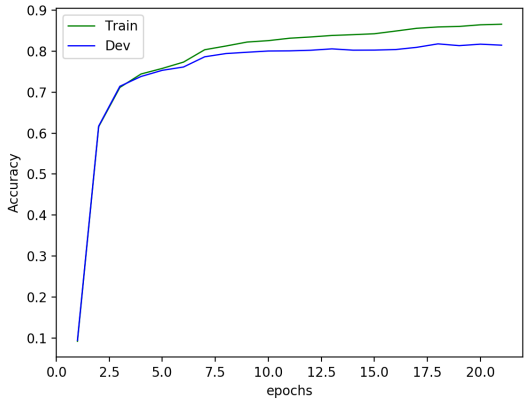

Figure 3: Train and dev accuracy using a learning rate of 0.001, no regularization and sigmoid activation's with Adam optimizer. This model achieved 86.4% and 81.7% train and dev accuracy respectively.

I also experimented with SGD optimizer and ReLU activations, the results are presented in Figure 4 and Figure 5 respectively.

Exercise 3

- With 2 hidden layers of size 128 each, the achieved results were: 83.8% train accuracy and 80.1% dev accuracy (Figure 6).
- With 3 hidden layers of size 128 each, the achieved results were: 80.6 % train accuracy and 78.7% dev accuracy (Figure 7

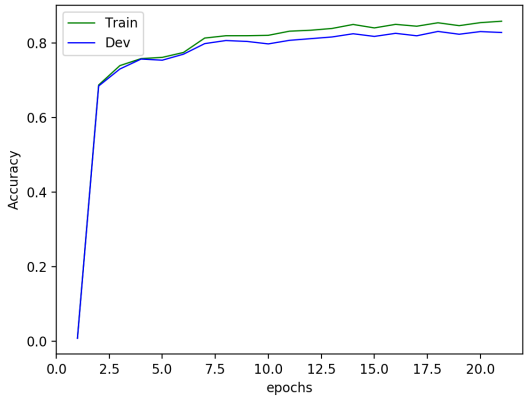

Figure 4: Train and dev accuracy using a learning rate of 0.008, no regularization and sigmoid activation's with SGD optimizer. This model achieved 85.8% and 82.8% train and dev accuracy respectively.

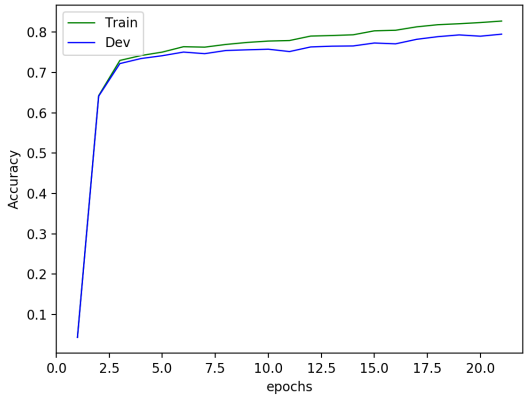

Figure 5: Train and dev accuracy using a learning rate of 0.001, no regularization and ReLU activation's with Adam optimizer.

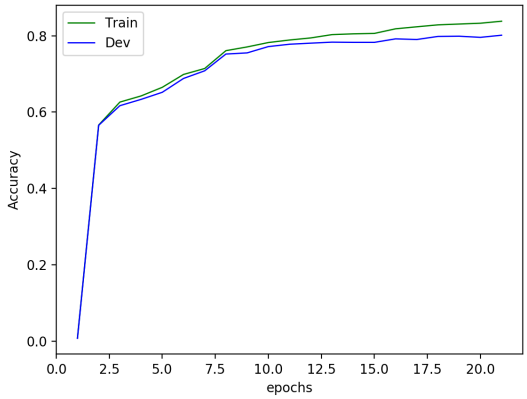

Figure 6: Train and dev accuracy for a deep feed forward neural network with 2 hidden layers of size 128.

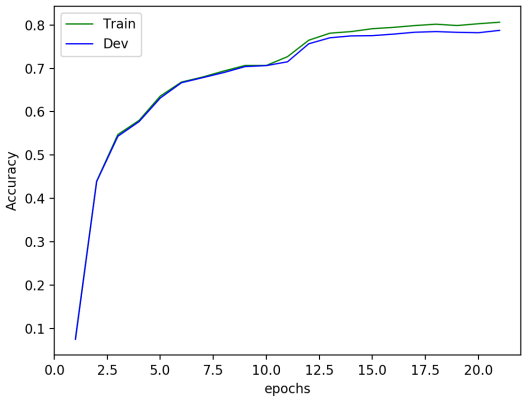

Figure 7: Train and dev accuracy for a deep feed forward neural network with 3 hidden layers of size 128.

Question 2

Exercise 1

a)

Knowing Johns activities, to compute the most likely weather for the past week we just need to run the Viterbi algorithm. The viterbi algorithm given the observations in table 3 returns the following path: Rainy - Rainy - Rainy - Sunny - Sunny - Sunny - Sunny

b)

Part I: If I have any observation we can only look into the Transition matrix, thus, the HMM is nothing more than a Markov Chain. Since you want to predict the weather between October 7 and 15 only knowing that it was Rainy on October 7 we need to compute: $P^t\gamma$ where $\in \{1, 2, ..., 7\}$ and γ is the initial distribution.

For October 8 our previsions would be:

$$\begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.5 \end{bmatrix}$$

Normalizing the result gives us:

$$\begin{bmatrix} 0.125 \\ 0.25 \\ 0.625 \end{bmatrix}$$

Since we win 1 euro if we guess correctly we have an expected reward of 0.625 cents and an expected cost of 0.375 cents which gives 0.25 cents profit for October 8.

For October 9:

$$\begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}^2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.5 \end{bmatrix}$$

Normalizing the result gives us:

$$\begin{bmatrix} 0.222 \\ 0.319 \\ 0.458 \end{bmatrix}$$

For October 9 we have an expected reward of 0.458 cents and an expected cost of 0.542 cents which gives -0.084 cents profit for October 8 (negative profit). If we keep doing this the chance that we predict the correct weather will keep decreasing and eventually it will converge to 1/3 probability for each possible state, which gives us an expected reward of 0.333 cents and an expected cost of 0.666 cents (-0.333 cents profit).

We can conclude that without additional information this game is not profitable at all.

Part II: After observing John's activities and the weather in October 7 and October 15 we can use the forward-backward algorithm in order to make our predictions. The forward-backward algorithm allow us to compute, for each position i, the label $\hat{y}_i$ that maximizes this posterior probability:

$$\hat{y}_i = argmax_{y_i} P(y_i|x)$$

After running the forward-backward algorithm I get the following path: Rainy - Rainy - Rainy - Sunny - Sunny - Sunny - Sunny and for each timestep the following probabilities:

$$t_0 = \begin{bmatrix} 0.041 \\ 0.195 \\ 0.764 \end{bmatrix} \; t_1 = \begin{bmatrix} 0.063 \\ 0.207 \\ 0.730 \end{bmatrix} \; t_2 = \begin{bmatrix} 0.140 \\ 0.285 \\ 0.575 \end{bmatrix} \; t_3 = \begin{bmatrix} 0.469 \\ 0.444 \\ 0.087 \end{bmatrix} \; t_4 = \begin{bmatrix} 0.755 \\ 0.159 \\ 0.086 \end{bmatrix} \; t_5 = \begin{bmatrix} 0.614 \\ 0.241 \\ 0.146 \end{bmatrix} \; t_5 = \begin{bmatrix} 0.614 \\ 0.241 \\ 0.146 \end{bmatrix}$$

$$t_6 = \begin{bmatrix} 0.970 \\ 0.024 \\ 0.010 \end{bmatrix}$$

Taking all this probabilities and applying a similar reasoning as presented in the Part I we get:

t_0 expected profit $= 0.041 * -1 + 0.195 * -1 + 0.764 * 1 = 0.528$
t_1 expected profit $= 0.063 * -1 + 0.207 * -1 + 0.730 * 1 = 0.460$
t_2 expected profit $= 0.140 * -1 + 0.285 * -1 + 0.575 * 1 = 0.150$
t_3 expected profit $= 0.469 * 1 + 0.444 * -1 + 0.087 * -1 = -0.062$
t_4 expected profit $= 0.755 * 1 + 0.159 * -1 + 0.086 * -1 = 0.509$
t_5 expected profit $= 0.614 * 1 + 0.241 * -1 + 0.146 * -1 = 0.227$
t_6 expected profit $= 0.970 * 1 + 0.024 * -1 + 0.010 * -1 = 0.939$
$\sum$ of expected profits $= 2.751$ euros.
We can conclude that information makes us richer.

Exercise 2

HMM's assume that observations are independent. With that said, we cannot model probabilities that depend on two consecutive observations. If we want to exploit such inter-observation features we need to use CRFs.

Question 3

Exercise 1

Taking back my implementation from the perceptron and changing it to take advantage of the structure behind the OCR data I was able to achieve 97.23%, 88.89% and 88.02% train, dev and test accuracy respectively (Figure 8).

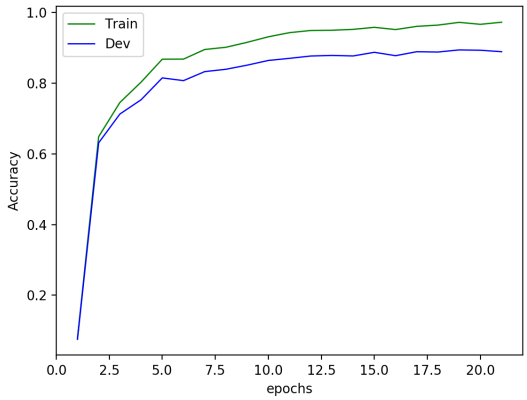

Figure 8: Train and dev accuracy with the structured-perceptron algorithm.