Assignment 3, option 2

DD2424 Deep Learning in Data Science

Anna Canal Garcia
annacg@kth.se

May 25, 2018

1 Introduction

This assignment aims at training a two layer ConvNet to predict the language of a surname from its spelling. In order to do the task slightly harder the accented letters are removed and replaced by their non-accented version.

The two layer ConvNet uses mini-batch gradient descent method applied to a cost function that computes the cross-entropy loss. In order to speed up the training we will add a momentum term in the update step. Accuracy, loss function and confusion matrix will be calculated on the validation set.

Moreover, during the back-prop training we will have to compensate the data since it contains unbalanced classes.

2 Method

The mathematical details of the 2-layer ConvNet are as follows:

$$\boldsymbol{x}_i^{(1)} = max(0, \boldsymbol{X} * \boldsymbol{F}_{1i}) \tag{1}$$

$$\boldsymbol{X}^{(1)} = \left(x_1^{(1)}, x_2^{(1)}, \cdots, x_{n2}^{(1)}\right) \tag{2}$$

$$\boldsymbol{x}_i^{(2)} = max(0, \boldsymbol{X}^{(1)} * \boldsymbol{F}_{2i}) \tag{3}$$

$$\boldsymbol{X}^{(2)} = \left(x_1^{(2)}, x_2^{(2)}, \cdots, x_{n2}^{(2)}\right) \tag{4}$$

$$s = \boldsymbol{W} vec(\boldsymbol{X}^{(2)}) \tag{5}$$

$$\boldsymbol{p} = SOFTMAX(\mathbf{s}) = \frac{exp(\mathbf{s})}{\mathbf{1}^T exp(\mathbf{s})} \tag{6}$$

And in order to speed up training momentum is added in the update step:

$$\boldsymbol{v}_t = \rho \boldsymbol{v}_{t-1} + \eta g \tag{7}$$

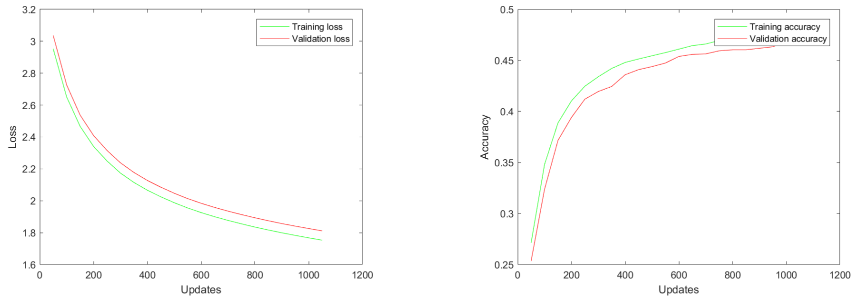
$$\theta_t = \theta_{t-1} - \boldsymbol{v}_t \tag{8}$$

In order to make the back-propagation efficient we will compute the convolutions as a matrix multiplications. We will compute MF and MX matrices. MF is based on the entries in the convolutional filter and performs all the convolutions at a given layer. MX is based on the entries in the input vector and performs all the convolutions at a given layer.

3 Results

1. **Analytic gradient computations check.**

   In order to test that the gradients computations are correct, the difference between them and the numerical is calculated. The difference of weight gradients is of 4.9135e-10, but in the case of the filters, the difference in the gradients is much bigger. Nevertheless, although if the gradients of the filters seemed to be incorrect, I trained the convolutional neural network with a batch of 100 samples, without momentum, learning rate 0.005, 1200 updates, and a network with 2 layers of 10 filters each and width of 5 and it resulted that the loss function was being reduced and the accuracy increased, as it can be seen in Fig.1.



(a) Loss function

(b) Accuracy

Figure 1: Loss function and accuracy with unbalanced data and $\eta = 0.005$, updates = 1200, n_batch = 100.

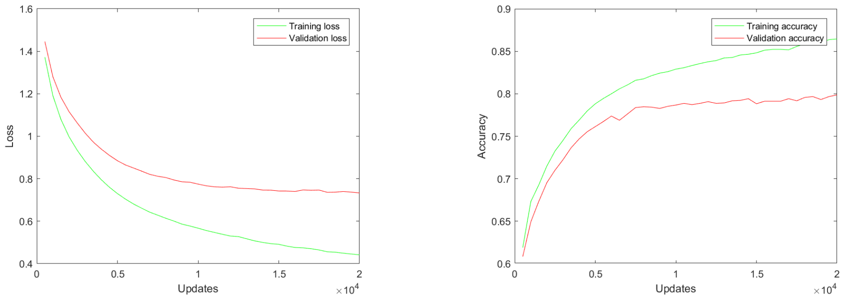
2. **Imbalanced dataset compensation**

   The training dataset contains unbalanced classes. In order to compensate that I chose the second option stated in the instructions: in each epoch of training I randomly sample the same number (which is the number of examples from the smallest class, 68) of examples from each class and this becomes the effective training set for this epoch.

3. **Validation loss with unbalanced dataset and balanced dataset**

   Network parameters: $k_1 = 5$, $k_2 = 3$, $n_1 = 20$ and $n_2 = 20$. Number of updates equal to 20000.

- Unbalanced dataset



(a) Loss function

(b) Accuracy

Figure 2: Loss function and accuracy with unbalanced data and $\rho = 0.9$, $\eta = 0.005$, updates = 20000, n_batch = 100.

As we can observe in Fig.2 with this network we achieved a final validation accuracy of 79.87% and the validation loss is reduced to 0.8. However, on Fig.3 we can observe that for the minority classes the precision (the percentage of all the examples predicted to belong to each class that are correctly and incorrectly classified) and recall (the percentage of all the examples belonging to each class that are correctly and incorrectly classified) are worst than with the classes that have more examples.

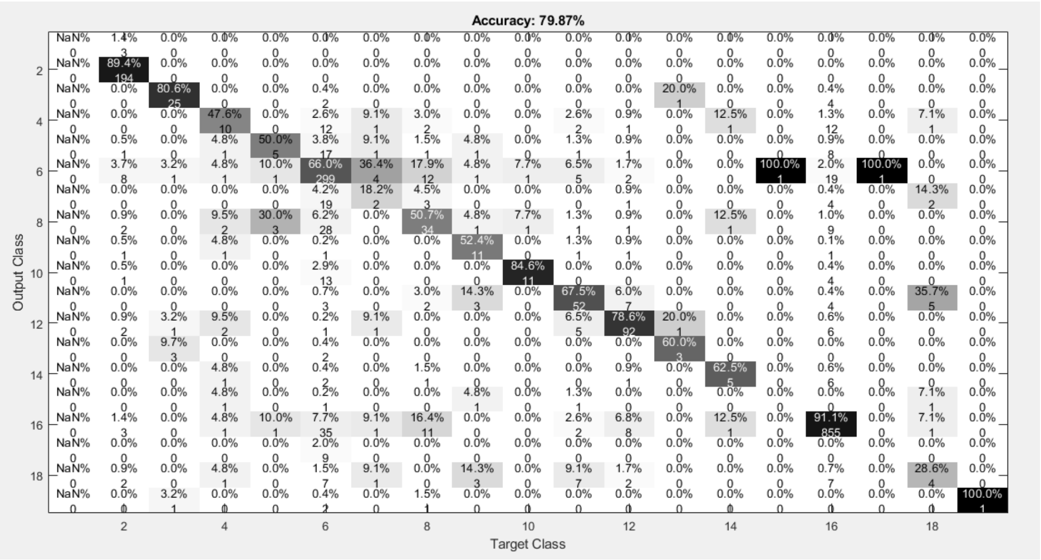
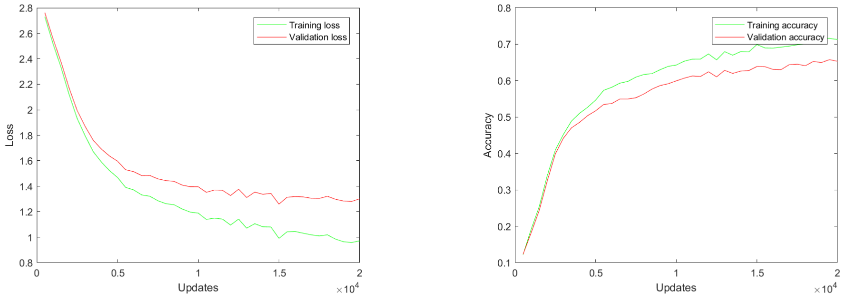


Figure 3: Confusion matrix with unbalanced data and $\rho = 0.9$, $\eta = 0.005$, updates = 20000, n_batch = 100.

- Balanced dataset

Now the loss is reduced less than before, as we can observe in Fig.4

but the precision and recall on the minority classes is better. On the confusion matrix in Fig.5 we can observe than in the diagonal, the numbers that were smaller before now are increased, that means that the prediction with minority classes is better. With that network we will be able to generalize better than before.



(a) Loss function

(b) Accuracy

Figure 4: Loss function and accuracy with balanced data and $\rho = 0.9$, $\eta = 0.005$, updates $= 20000$, n_batch $= 100$.

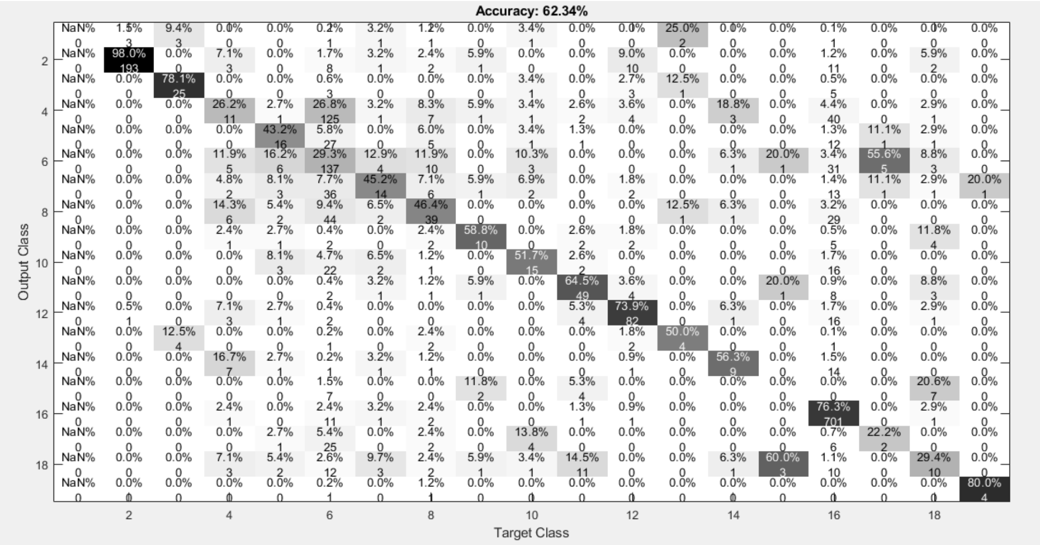


Figure 5: Confusion matrix with balanced data and $\rho = 0.9$, $\eta = 0.005$, updates $= 20000$, n_batch $= 100$.

4. **Best performing ConvNet**

Network parameters: $k_1 = 5$, $k_2 = 3$, $n_1 = 20$ and $n_2 = 40$.
Training parameters: $\eta = 0.005$, $\rho = 0.9$, $n\_epochs = 500$, $n\_batch = 100$ and $n\_updates = 6000$.

There is an increment of the number of filters with respect to the previous network in order to have better results. In order to have less computational time, this last test is done for 6000 updates, but we can check that the loss is similar than the previous one at 6000 updates. The final validation accuracy of the network is 57.10%.

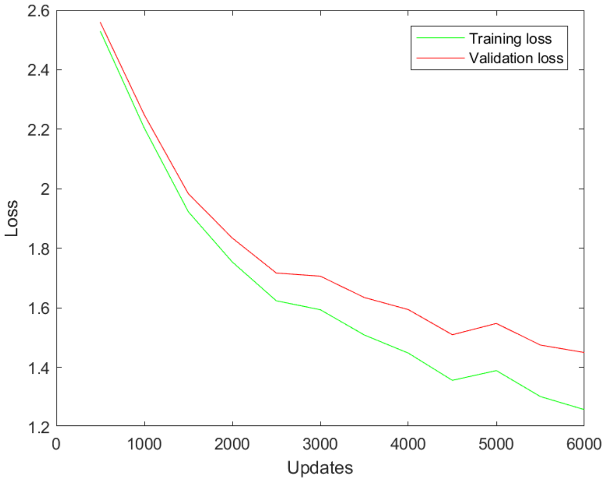


Figure 6: Confusion matrix with balanced data and $\rho = 0.9$, $\eta = 0.005$, updates = 6000, n_batch = 100.

5. **Probability vector output by your best network when applied to the surnames of 5 friends**

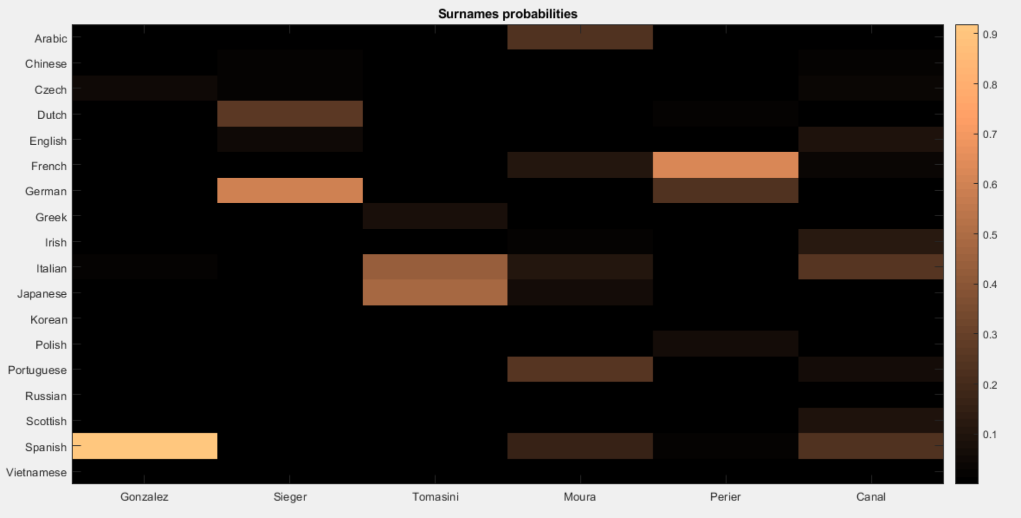


Figure 7: Probability vector output for 6 surnames

Finally, with the best network from previous question, the surnames of 5 friends and my surname are predicted. The surnames and its labels are the following: Gonzalez (spanish), Sieger (german), Tomasini (italian), Perier (french), Gogoulou (greek) and Canal (spanish). In Fig.7 you can see in a light color the higher accuracies. For Gonzalez (91.88%), Sieger (59.97%) and Perier (63.13%) the prediction is really good. For Tomasini the accuracy achieved is 44.2% and in the case of my surname, Canal, is of 24.3% and for Moura is 24.67%. As more distinguished is the surname, better prediction. Surnames like Moura and Canal, which are short and include consecutive letters that could be typical from different languages are difficult to predict. The accuracy of all the surnames is 66.67%.