University of Zurich[UZH]

Institut für Computerlinguistik
Text Crunching Center
Dr. Manfred Klenner

Maschinelles Lernen in der Computerlinguistik
Exercise 5

November 23, 2020

Submission Instructions :

1. Upload the solutions in pdf file naming **ex5_name1_name2.pdf** on olat
2. Please provide detailed methodology of you solution rather than writing only the final answer

K-fold cross-validated paired t test

Task 1. K-fold cross-validated paired t-test procedure is a common method for comparing the performance of two models/classifier. In this task, we will compare the performance of Logistic Regression and Decision Tree models. We have a labelled dataset D and we evaluated the performance of these models using 10-fold cross validation on D. The performance accuracy values of Logistic Regression and Decision Tree are given in table . Now using paired t-test, we would like to test our Null hypothesis given below:
Null Hypothesis: The Logistic Regression and Decision Tree models have equal performance on Dataset D: [3+1=4 points]

df↓ / P→	0.90	0.95	0.975	0.99	0.995	0.999	0.9995
∞	1.282	1.645	1.960	2.326	2.576	3.091	3.291
1	3.078	6.314	12.706	31.821	63.656	318.289	636.578
2	1.886	2.920	4.303	6.965	9.925	22.328	31.600
3	1.638	2.353	3.182	4.541	5.841	10.214	12.924
4	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	1.476	2.015	2.571	3.365	4.032	5.894	6.869
6	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	1.372	1.812	2.228	2.764	3.169	4.144	4.587

Figure 1: Critical value of t for the Student's t Distribution for given degree of freedom (df) and confidence level (P= $\Pr(X \leq x)$)

Model	Accuracy values from cross validation
Logistic Regression	$[0.95, 0.89, .92, 0.98, 0.88, 0.975, 0.90, 0.98, 0.98, 0.94]$
Decision Tree	$[0.85, 0.95, .88, 0.91, 0.94, 0.92, 0.955, 0.89, 0.82, 0.85]$

1. Find t value of our paired t-test.

2. Test if our Null hypothesis can be rejected or not for confidence levels of 95% and 99% respectively.
(Hint: Use the table given in Figure 1 to find t critical)

Train BERT Using Hugging face

Task 2. In this text classification task, you have to detect fake news (classifying an article as REAL or FAKE) using BERT Model from HuggingFace library. You must use a pre-trained BERT model and fine-tune it with respect to the given task.
To simplify the task, you have been given the skeleton code which loads , preprocess and tokenize the data. It also creates dataloaders for training, validation and test sets. In order to complete the code, you have to complete the #TODO part in the skeleton code. [2+2+1+2=7 points]

Dataset: News article is given with their fake/real label
Data file: Link
Skeleton Code : Link

1. #TODO1 : Implement the **NewsClassifier** class using pre-trained BERT model. (Both Initialization and forward method needs to be completed)

2. #TODO2 : Initialize all hyperparameters required for training. It includes loss function,learning rate, epochs, optimizer etc.

3. #TODO3 : Write the training loop to fine-tune the **NewsClassifier**. Hint: train_epoch and eval_model functions are already given in the code. You just have to call them to train the network.

4. Evaluate the model on test set after training. If classification accuracy on test set is 90−95%, you will get 1 Mark and for accuracy >95%, 2 marks will be given.

Note: Since BERT is a heavy model with lots of parameters, you can use google colab to train the network on GPU (Donot forget to change runtime to GPU in colab settings)

Pooling Strategies

Task 3. There are various pooling strategies used to reduce the signal resolution and lower the computational resources required to train the complete network. For more information, use `max_pool2d` and `avg_pool2d`. For this question assume that all the required libraries are installed. [1+2=3points]

- Write down the shape of tensor `x1, x2, x3, x4` after each pooling operation listed below

```
input = torch.randn(10, 30, 60).   #First dim(size 10) is batch dim
x1 = F.max_pool2d(input, kernal_size=(2,2))
x2 = F.max_pool2d(input, kernal_size=(3,1))
x3 = F.avg_pool2d(input, kernal_size=(1,3))
x4 = F.avg_pool2d(input, kernal_size=(2,3))

```

- Let's assume we have tensor `Input`

$$\texttt{Input} = \begin{bmatrix} 2 & 1 & 1 & 2 \\ 0 & 0 & 2 & 0 \\ 0 & 2 & 2 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix}$$

  Write down the output tensors `output_max` (after `max_pool2d`) and `output_avg` (after `avg_pool2d`) operation, both with `kernal_size=(2,2)` and `stride=(2,2)`

Hyperparameter tuning and Regularization

Task 4. In this task, we are going to use the same dataset which we used in Exercise 3. The code skeleton is already given in the jupyter notebook and you have to complete the code by filling the #TODO part in the skeleton code. Comments have also been provided with the #TODO as helpers.

Task Description: We'll use a public dataset from the BBC comprised of 2225 articles, each labeled under one of 5 categories: business, entertainment, politics, sport or tech. The dataset is broken into 1780 records for training and 445 for testing. The goal will be to build a system that can accurately classify previously unseen news articles (i.e Test set) into the right category.

Cleaned Dataset and Skeleton Code: Link

Reference links for LayerNorm, Dropout, Adam Optimizer

Evaluation Metrics : Test set Accuracy

Marking Scheme:

1. There are 6 #TODO parts and first 4 #TODO will get 0.5 points each and last 2 #TODO will get 0.25 points each. (3 points)
2. You will get 1 point if complete code runs without any error.
3. Points Based on Test Accuracy:
   2 points for accuracy $\geq 95\%$
   1 point for $90\% \leq$ accuracy $< 95\%$
   0 point for accuracy $< 90\%$