Karlsruhe Institute of Technology

Praktikum Smart Data Analytics

Übungsblatt 3

11.07.2022

Gruppe 4: Ola Sharfeldin, Xin Tian, Dmitrii Seletkov, Danilo Rosenthal, Dan Jia

Agenda

- Datasets
- Pre-training
- Self-training on Transformations
  - Binary Classification
  - Multi-class Classification
- Teacher and Student Model
  - Transfer Learning
  - Supervised Training
  - Fine-tuning

Datasets

Data sets

- Opportunity:
  - 18 classes
- Daphnet:
  - 2 classes
- PAMAP2:
  - 11 classes

Opportunity

```
Testing Accuracy: 0.89046097
validation accuracy: 0.89046097
f1_score_w 0.8893312580684116
f1_score_m 0.6012514897399348
confusion_matrix
[[8598   17   19   23   19   44   12   27   17    2   23    1    5    7
     4    6  196   31]
 [  13   49    0   23    0    0    0    0    0    0    0    0    0    0
     0    0    0    0]
 [   9    0   69    0   27    0    0    0    0    0    0    0    0    0
     0    0    0    3]
 [  22    7    0   38    3    0    0    0    0    0    0    0    0    0
     0    0    1    0]
 [   3    0    2    0   94    0    0    0    0    0    0    0    0    0
     0    0    0    0]
 [  66    0    1    0    0  159   14    5    2    0    0    1    0    0
     0    0    0    1]
 [  37    0    0    0    0    5  121    0    3    1    0    0    0    0
     0    0    0    2]
 [  36    0    0    0    0    1    0   79    1    1    1    0    1    0
     0    0    2    2]
 [  39    0    0    0    0    1    4    8   44    1    2    0    1    0
     3    0    0    1]
 [   9    0    0    0    0    0    0    1    0   21    5    0    0    1
     0    0    1    7]
 [   6    0    0    0    0    0    0    0    0    3   19    1    2    0
     0    0    0    0]
 [  10    0    0    0    0    0    0    3    0    5    1   13    6    3
     3    0    0    0]
 [   8    0    0    0    0    0    0    0    1    0    5    0   14    3
     7    0    0    0]
 [  15    0    0    0    0    0    0    0    0    0    0    1    0   43
     8    0    0    0]
 [  14    0    0    0    0    0    0    1    0    0    0    0    5   10
    31    0    0    0]
 [  55    0    0    1    0    0    0    0    0    0    0    0    0    0
     0   50    4    0]
 [ 128    2    0    0    0   11    0    0    0    0    0    0    0    0
     0    0  245    0]
 [  40    0    5    0    0    0    0    0    0    0    0    0    0    0
     0    0    0   68]]
```

Daphnet

```
Testing Accuracy: 0.89634144
validation accuracy: 0.89634144
f1_score_m 0.70743001966606919
confusion_matrix
[[3206   84]
 [ 307  175]]
```

PAMAP2

```
Testing Accuracy: 0.7420382
validation accuracy: 0.7420382
f1_score_w 0.7187668004530933
f1_score_m 0.69301085841678 76
confusion_matrix
[[56  0  1  1  0  1  0  0  0  0  0]
 [ 0 51  7  0  0  0  0  0  0  0  0]
 [ 0 20 40  0  0  0  0  0  0  0  1]
 [ 0  0  0 63  0  1  1  0  0  0  0]
 [ 0  0  0  1  0  0 19 36  0  0  0]
 [ 0  0  0  0  0 50  0  0  2  0  0]
 [ 0  0  0 12  0  0 55  0  0  0  0]
 [ 0  0  0  0  0  0  0 22 12  0  0]
 [ 0  0  0  0  0  0  3  1 22  2  0]
 [ 0  0  0  0  0  0  0  1 13 40  0]
 [ 0  0  0  0  0  0  0  0  0 27 67]]
```

Pre-training

Pre-training: Hypothesis

- Hypothesis: pretrain on another dataset and train the baseline enhances the results and convergence speed
- Works good in Computer Vision e.g. ImageNet [Huh16]

- Network:
  - Conv Input Layer - different for different datasets
  - Conv Hidden Layer (1, 3, 128, 128) - always the same
  - FC Output Layer - different for different datasets
- Pipeline
  - Train Net 1 on one dataset (Opportunity), save Conv Hidden layer 1
  - Train Net 2 on another dataset, but initialized with Conv Hidden layer 1
  - Train many times to obtain mean and variance

Pre-training: Results And Analysis

- PAMAP2 dataset, epochs=**50**, lr=0.0005, num_experiments=5

Approach	Test Accuracy	F1 score weighted	F1 score mean
Baseline	0.724 (0.035)	0.683 (0.046)	0.680 (0.044)
Pretrained on OPP	**0.811 (0.069)**	**0.802 (0.073)**	**0.774 (0.085)**

- PAMAP2 dataset, epochs=**100**, lr=0.0001, num_experiments=5

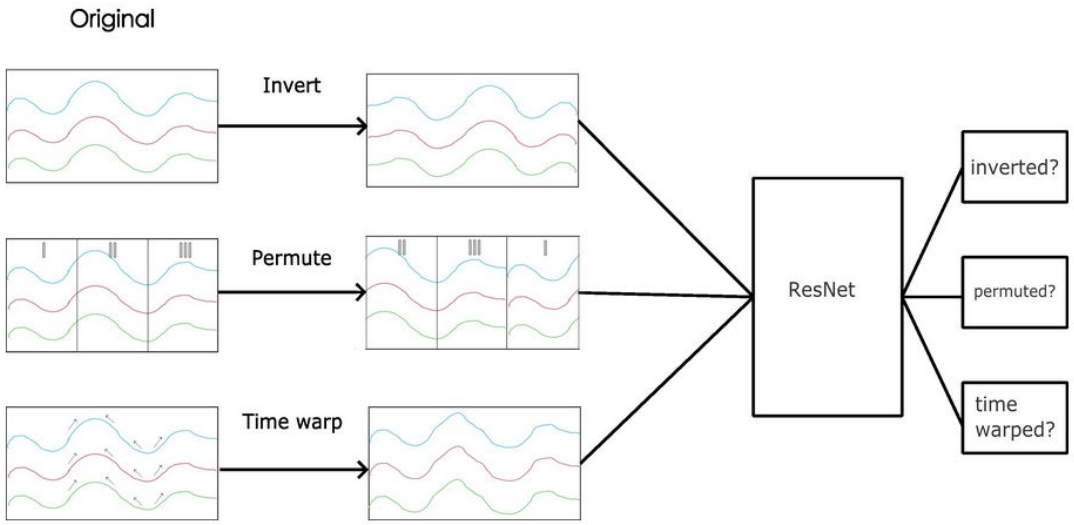
Approach	Test Accuracy	F1 score weighted	F1 score mean
Baseline	**0.908 (0.005)**	**0.906 (0.005)**	**0.894 (0.006)**
Pretrained on OPP	0.893 (0.021)	0.891 (0.022)	0.879 (0.023)

- Analysis
  - Pre-training enhances the convergence speed
  - Pre-training does not enhance the final results of the training
    - The network is too small
    - The num_epochs is large

Self-training on Transformations

Hypothesis

- Hypothesis: self-training on pretext task (transformation) enhances results
- Based on [Yuan22]
- Two different approaches:
  - Binary Classification, whether the transformation is applied
  - Multi-class Classification, which transformation is applied

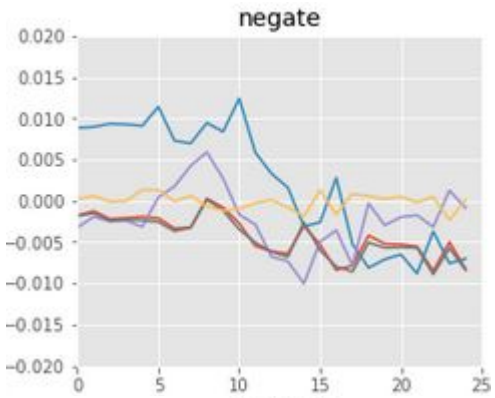


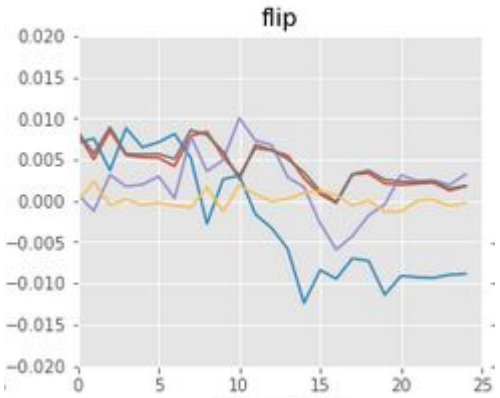
Multi-class classification self-training [Yuan22]

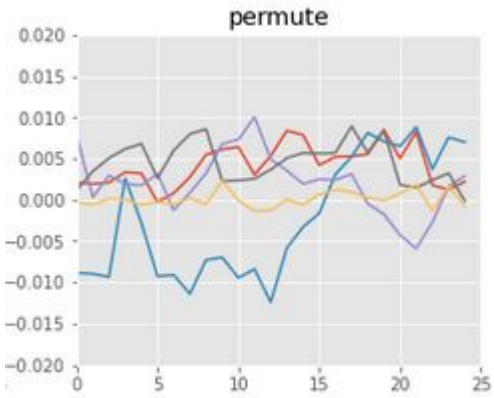
Transformations

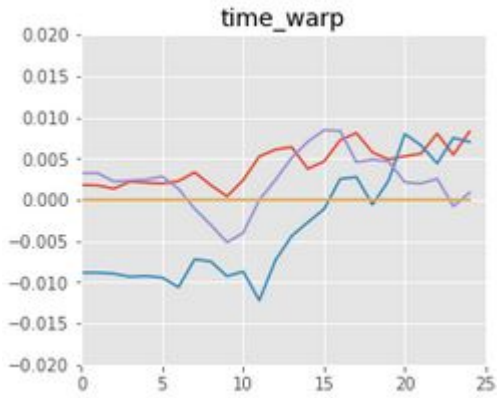
- Different transformations are possible:
  - flip, negate, permute, time warp
- Illustration on sample from PAMAP2

Original Sample









Transformations applied on original sample

Self-training: Results And Analysis

- PAMAP2 dataset, epochs=100, lr=0.0001, num_experiments=5

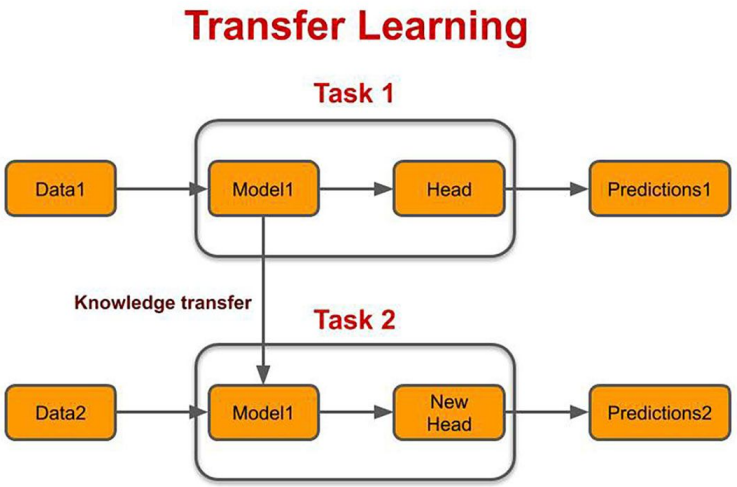
Approach	Flip	Permute	Time warp	Test Accuray	F1 Score weighted	F1 Score mean
Baseline	-	-	-	**0.908 (0.005)**	**0.906 (0.005)**	**0.894 (0.006)**
Binary Flip	**+**	-	-	0.819 (0.044)	0.807 (0.054)	0.793 (0.054)
Binary Permute	-	**+**	-	0.889 (0.009)	0.887 (0.009)	0.870 (0.013)
Binary Warp	-	-	**+**	0.900 (0,008)	0.899 (0.009)	0.885 (0.008)
MultiClass	**+**	**+**	**+**	0.809 (0.021)	0.791 (0.031)	0.772 (0.036)

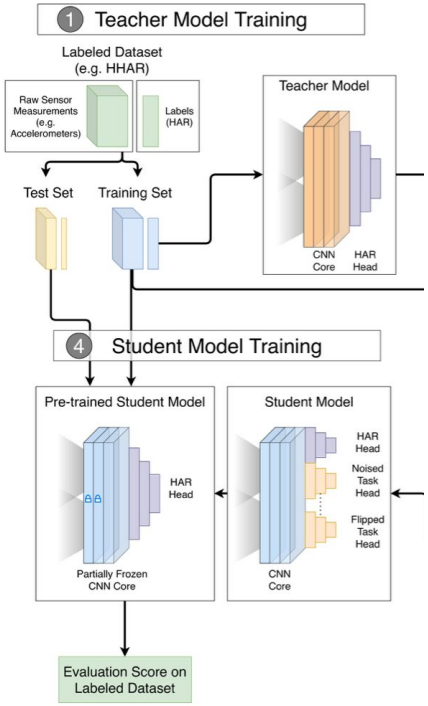
- Analysis:
  - No visible advantage of self-training
  - Possible reasons:
    - Too small network

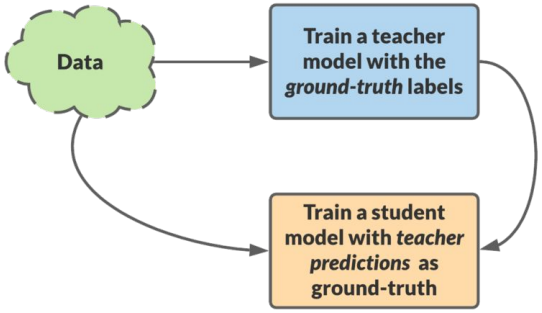
Teacher and Student Model

Transfer Learning

- Train a model with a large dataset, extract *the optimal parameters (weights)* and then store them as ***a good experience***.
- Take full advantage of old knowledge rather than train a new model from zero when we get ***a new similar task***.



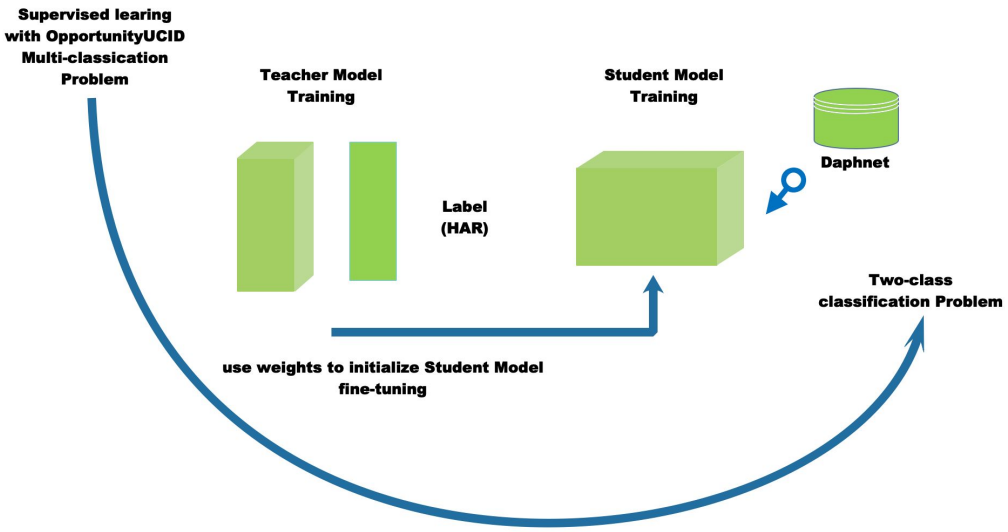


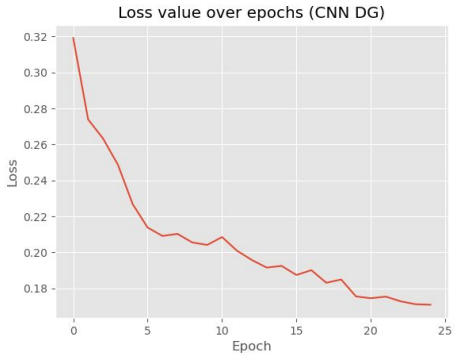


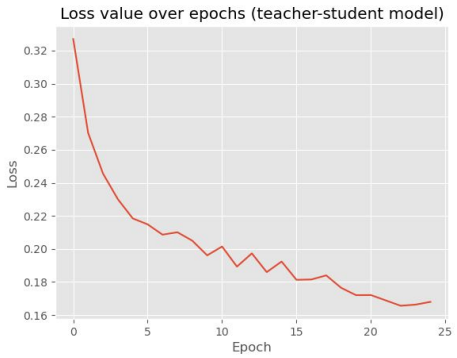
https://github.com/iantangc/SelfHAR [CHI IAN TANG, 2021]

How to improve performance with TL? Experiment Design

- Student Model: *baseline*
  a. generally, initialize this network with random weights.
  b. Instead of random weight, adopt the weights that is extracted from ***the teacher model*** to train ***this student model***
- Teacher Model
  a. Train *a teacher model* ahead of time.
  b. That is so-called pre-trained model.
  c. And this process is pre-training.

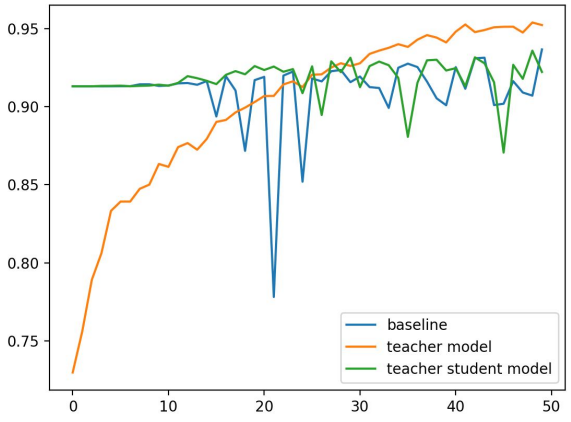


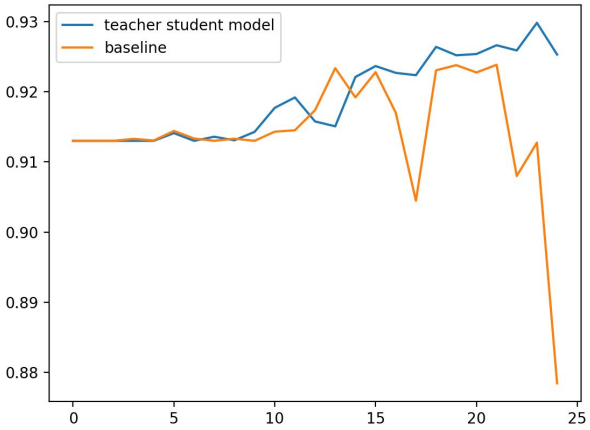




Experiment Results

Iteration	Metrics	Model		
		Baseline, Student Model (Daphnet)	Teacher Model (OpportunityUCID)	Teacher-Student Model (Daphnet)
50	Testing Accuracy	0.8971368	0.89246917	0.890509
	Validation Accuracy	0.8971368	0.89246917	0.890509
	f1 score	0.6856091	0.89249132	0.681746
25	Testing Accuracy	0.8894486	-	0.9135737
	Validation Accuracy	0.8894486	-	0.9135737
	f1 score	**0.7277423**	-	**0.7816385**





no significant improvement

- ❖ the scale of teacher model is small

Summary

Summary And Future Works

- Tested approaches:
  - Pre-training
  - Self-training on transformations
  - Teacher-Student models
- Pre-training allows faster convergence
- Self-training does not show better performance due to
  - Network Architecture
  - Hyperparameters
- Teacher-Student shows better performance
  - combine with large unlabeled dataset

Questions?

Appendix

Pre-training: Results

- PAMAP2 dataset, epochs=50, lr=0.0005, num_experiments=5

```
Test accuracy. Mean (std) = 0.724 (0.035)
Val accuracy. Mean (std) = 0.724 (0.035)
f1 score weighted. Mean (std) = 0.683 (0.046)
f1 score mean. Mean (std) = 0.680 (0.044)
```

Baseline after 50 epochs

```
Test accuracy. Mean (std) = 0.811 (0.069)
Val accuracy. Mean (std) = 0.811 (0.069)
f1 score weighted. Mean (std) = 0.802 (0.073)
f1 score mean. Mean (std) = 0.774 (0.085)
```

Pretrained on Opportunity after 50 epochs

- PAMAP2 dataset, epochs=100, lr=0.0001, num_experiments=5

```
Test accuracy. Mean (std) = 0.908 (0.005)
Val accuracy. Mean (std) = 0.908 (0.005)
f1 score weighted. Mean (std) = 0.906 (0.005)
f1 score mean. Mean (std) = 0.894 (0.006)
```

Baseline after 100 epochs

```
Test accuracy. Mean (std) = 0.893 (0.021)
Val accuracy. Mean (std) = 0.893 (0.021)
f1 score weighted. Mean (std) = 0.891 (0.022)
f1 score mean. Mean (std) = 0.879 (0.023)
```

Pretrained on Opportunity after 100 epochs

- Analysis
  - Pre-training enhances the convergence speed
  - Pre-training does not enhance the final results of the training
    - The network is too small
    - The num_epochs is large

Results

- PAMAP2 dataset, epochs=100, lr=0.0001, num_experiments=5

```
Test accuracy. Mean (std) = 0.908 (0.005)
Val accuracy. Mean (std) = 0.908 (0.005)
f1 score weighted. Mean (std) = 0.906 (0.005)
f1 score mean. Mean (std) = 0.894 (0.006)
```

Baseline after 100 epochs

```
Test accuracy. Mean (std) = 0.869 (0.010)
Val accuracy. Mean (std) = 0.869 (0.010)
f1 score weighted. Mean (std) = 0.866 (0.012)
f1 score mean. Mean (std) = 0.850 (0.016)
```

Self-trained on negate

```
Test accuracy. Mean (std) = 0.900 (0.008)
Val accuracy. Mean (std) = 0.900 (0.008)
f1 score weighted. Mean (std) = 0.899 (0.009)
f1 score mean. Mean (std) = 0.885 (0.008)
```

Self-trained on time warp

```
Test accuracy. Mean (std) = 0.889 (0.009)
Val accuracy. Mean (std) = 0.889 (0.009)
f1 score weighted. Mean (std) = 0.887 (0.009)
f1 score mean. Mean (std) = 0.870 (0.013)
```

Self-trained on permute

```
Test accuracy. Mean (std) = 0.819 (0.044)
Val accuracy. Mean (std) = 0.819 (0.044)
f1 score weighted. Mean (std) = 0.807 (0.054)
f1 score mean. Mean (std) = 0.793 (0.054)
```

Self-trained on time flip

```
Test accuracy. Mean (std) = 0.809 (0.021)
Val accuracy. Mean (std) = 0.809 (0.021)
f1 score weighted. Mean (std) = 0.791 (0.031)
f1 score mean. Mean (std) = 0.772 (0.036)
```

Self-trained on flip, permute, time warp

Results

```
Test accuracy. Mean (std) = 0.724 (0.035)
Val accuracy. Mean (std) = 0.724 (0.035)
f1 score weighted. Mean (std) = 0.683 (0.046)
f1 score mean. Mean (std) = 0.680 (0.044)
```

Baseline after 50 epochs

```
Test accuracy. Mean (std) = 0.746 (0.043)
Val accuracy. Mean (std) = 0.746 (0.043)
f1 score weighted. Mean (std) = 0.719 (0.055)
f1 score mean. Mean (std) = 0.708 (0.060)
```

Self-trained on permute after 50 epochs