Xiao Zhou

Mohammad Soleymani

CS535

20 March 2024

HW2

1. I used mean pooling for reducing the temporal dimension of the audio and visual sequences. Since the features are one dimensional, I used a two layer MLP model for the unimodal classification task. The parameter set I choose are:

   hidden_size1 = [512, 256, 128]

   learning_rates = [0.0005, 0.001, 0.005]

   batch_sizes = [128, 64, 32]

   After grid search the best parameter combination are:

	hidden _1	Batch Size	Learning Rate	F1
Visual	256	64	0.0005	0.582089552238
Audio	256	64	0.0005	0.578358208955
Text	256	64	0.0005	0.649253731343

2. ***After I finished the homework, the professor said what I did is actually subject dependent cross-validation since I didn't separate different speakers.*** I used class weights by assigning different weights to classes during training to help the model penalize misclassifications of the minority class more heavily. Specifically,

I used compute_class_weight from keras. The 15 confusion matrices results after adjusting weights are:

Visual

Fold: 0, F1: 0.5559701492537313

[[35 7 3 24]

[4 30 3 11]

[10 11 11 11]

[14 12 9 73]]

Fold: 1, F1: 0.5543071161048689

[[34 9 4 15]

[10 33 5 18]

[6 6 9 13]

[14 13 6 72]]

Fold: 2, F1: 0.5917602996254682

[[37 6 2 22]

[12 32 5 13]

[7 3 15 8]

[12 12 7 74]]

Fold: 3, F1: 0.5468164794007491

[[42 7 7 7]

[16 29 4 24]

[7 7 16 7]

[22 10 3 59]]

Fold: 4, F1: 0.4756554307116105

[[28 9 10 20]

[11 25 7 16]

[5 4 10 14]

[20 17 7 64]]

F1 for 5 folds: 0.5449101796407185

Audio

Fold: 0, F1: 0.5149253731343284

[[43 4 5 17]

[2 36 5 21]

[5 7 4 16]

[7 25 16 55]]

Fold: 1, F1: 0.5655430711610487

[[47 6 4 13]

[3 40 7 14]

[6 4 6 21]

[12 15 11 58]]

Fold: 2, F1: 0.5243445692883895

[[34 2 9 16]

[4 43 8 10]

[6 8 2 17]

[16 22 9 61]]

Fold: 3, F1: 0.5243445692883895

[[47 2 5 15]

[8 30 6 16]

[6 6 8 17]

[14 16 16 55]]

Fold: 4, F1: 0.5468164794007491

[[43 3 3 10]

[2 34 6 13]

[2 13 5 21]

[15 13 20 64]]

F1 for 5 folds: 0.5351796407185628

Text

Fold: 0, F1: 0.6156716417910447

[[47 3 0 11]

[6 42 6 11]

[5 5 11 14]

[15 16 11 65]]

Fold: 1, F1: 0.6367041198501873

[[42 4 3 14]

[7 45 7 14]

[6 6 13 6]

[10 15 5 70]]

Fold: 2, F1: 0.5805243445692884

[[48 10 3 18]

[6 26 4 9]

[3 3 16 18]

[13 19 6 65]]

Fold: 3, F1: 0.6067415730337079

[[44 3 3 15]

[7 35 4 13]

[3 5 13 16]

[12 16 8 70]]

Fold: 4, F1: 0.6367041198501873

[[46 3 5 6]

[7 46 3 10]

[4 7 13 13]

[15 8 16 65]]

F1 for 5 folds: 0.6152694610778443

3. Early Fusion F1: 0.746268656716418

[[62 5 2 9]

[1 36 3 7]

[3 4 17 5]

[12 11 6 85]]

I used the same MLP model for early fusion by concatenating all features together into a combined 1D feature with hidden_layer1 = 512, hidden_layer2 = 128, batch_size = 64, learning_rate = 0.001 and 100 epochs. The F1 slightly increased to 0.75 compared to 0.62, which is the highest F1 in the unimodal.

Late Fusion F1: 0.9776119402985075

[[77   0   0   1]

[0 55   0   1]

[0   1 26   0]

[1   0   2 104]]

I used 3 unimodel of audio, visual and text from the first problem and got the probability for each modality using softmax for each sample. Then I added up all possibilities from each modality and returned the label of the greatest combined probability. The F1 increased greatly to 0.98 compared to both unimodal and early fusion. Very few samples are misclassified according to the confusion matrix.

4. Late fusion performs best. Early fusion performs slightly better than unimodal since it adds more features into the modal, providing more information. However,

each dimension is treated the same although they are not in the same domain. Late fusion uses different modalities to provide complementary information about the emotions. Without getting conclusion from one modality first, it adds up the probabilities to get the final results, which also helps in reducing overfitting by preventing the model from relying too heavily on a single modality.