**Differentiable model selection for ensemble learning**

Please cite as:

@inproceedings{

inproceedings,

author = {Kotary, James and Vito, Vincenzo and Fioretto, Ferdinando},

year = {2023},

month = {08},

pages = {1954-1962},

title = {Differentiable Model Selection for Ensemble Learning},

doi = {10.24963/ijcai.2023/217}

}

This paper contains the code associated with the paper Differentiable Model Selection for Ensemble Learning, James Kotary\*, Vincenzo Di Vito\*, Ferdinando Fioretto. In Proceedings of the International Joint Conference of Artificial Intelligence (IJCAI), 2023.

Full paper at <https://www.ijcai.org/proceedings/2023/0217.pdf>

**Usage**

Download the Sentiment Analysis dataset at <https://www.kaggle.com/datasets/msambare/fer2013>.

To train the ensemble’s base learner, run the file fer\_base\_learner.py

The following arguments can be set:

* --primary\_class, type=str, help = 'class(es) the base learner is mainly trained on.’

Ex: primary\_class=’0’ means that a high percentage of samples of the base learner training dataset belong to class ‘0’. You may want or not the base learner to be specialised on a subset of classes. This aspect is very influencing on the effectiveness of an ensemble learning algorithm.

* --model\_name, type=str, help='Model name for saving - will not save model if name is None')
* --lr, type=float,default=0.95, help='learning rate'
* --gamma, type=float, default=0.7, help='Learning rate step gamma (default: 0.7)'
* --batch\_size, type=int,default=64, help='batch size'
* --epochs', type=int, default=100, help = 'number of epochs'

To train the master ensemble, run the file fer\_e2eCEL.py. A variety of arguments can be set

* --c, type=int, help='number of models of the ensemble'
* --batch-size, type=int, 'help=input batch size for training'
* --test-batch-size, type=int, 'help=input batch size for testing',
* --epochs, type=int, 'help=number of epochs to train',
* --lr, type=float, help='learning rate',
* --gamma, type=float, help='Learning rate step gamma',
* --no-cuda, type=bool, help='disables CUDA training',
* --seed, type=int, help='random seed',
* --model\_name, help='Model name for saving - will not save model if name is None',
* --use\_softmax, type=bool, help='Apply softmax to the base learners combined predictions',
* --weight\_pred, type=bool, help='weight each base learner prediction by the corresponding model confidence',
* --injection, type=bool, help='Inject base learner confidence into the knapsack problem,
* --apply\_sum, type=bool, help='Apply sum to the 1/2 class soft predictions',
* --use\_cvxpy, type=bool, help='Use cvxpy layer to solve the knapsack problem',
* --softmax\_temperature, type=float, help='softmax temperature',
* --sched, type=bool, help='If false we do not use the scheduler'