COMP 521 - HW #7

Ilker Kesen, 57641

MODEL

I devised a neural network-based solution by taking advantage of commonly used deep learning methodologies. I used PyTorch for this assignment.

I applied cross-validation to select a robust model. Since we're dealing with imbalanced data where negative samples are majority, I did choose validation split rigorously: one positive example per negative example. I used 10k examples for validation split. 10k instances for validation split is good for this task: It's not less and not too much (the more training data, the way better usually), so it's enough. They were also some problematic models due to floating number calculations, I didn't include them to my ensemble zoo.

I also performed data preprocessing. Since some features have broad ranges, I normalized all of them (except one-hot features) by using standard normalization (and it helped).

Since we have imbalanced data, I used F1 metric instead of accuracy. If we'd have used accuracy, the scores would mislead us because we've got plenty of negative samples.

The neural network I developed for this task is an MLP with a single hidden layer. It uses relu as activation function and also includes a dropout layer which can be considered as a regularization method mimicking ensembling. In this method, some units are cancelled, the rest is used, and which units are going to be cancelled is being changed in every forward propagation.

I used RMSprop as optimization method which is superior to vanilla stochastic gradient descent. I trained all the ensembles for 50 epochs and stored and then used the models with best F1-scores. I used gradient clipping to prevent gradients from exploding.

In ensembling, I generated probabilities and did take their averages to generate final probabilities.