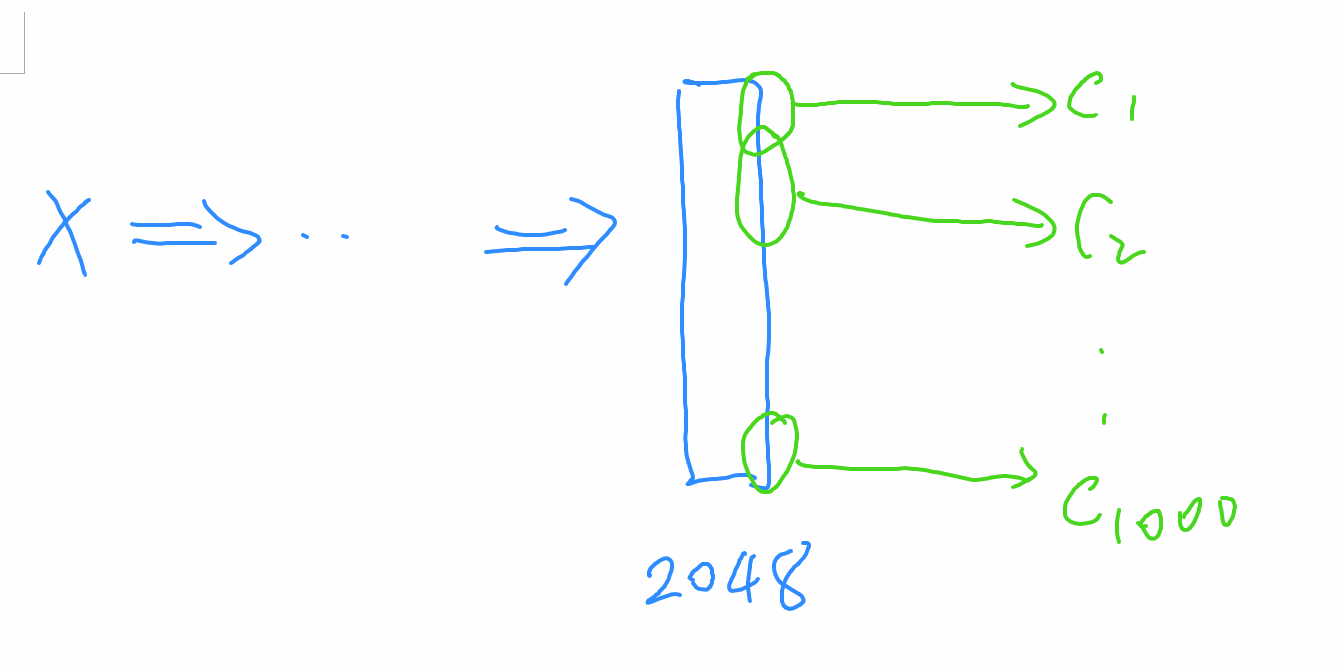
­

Each class depends heavily on a small number of features.

However, the model still depends on “all” of the features at test time. Our goal is to make the model depends only on the “important features”. But “important features” can only be defined wrt a given class.

GAO Han: Pick one class C1, and train a **binary classifier for C1** only. The iid-ood gap is significantly reduced.

Suggestion:

* Make a super-class, C\_dog (union of all dogs) [another person, C\_flower]
* Determine the important features for C\_dog
  + Determined by fine-tuned model: Take ResNet, remove the last layer, add a binary classifier, learn the W for the last layer in new model by fine-tuning while keep the backbone frozen.
* Build a binary classifier for C\_dog using IR.
  + Set the weight for unimportant features to 0 for both classes

New issue: Classes are unbalanced: much fewer dog images than non-dog images. Solution: Reweight dog-images proportionally.

* 10% dog images, 90% non-images
* Weight for each dog image would be 9, while the weight for each non-dog image is 1
* The rebalance at the batch-level or dataset-level? Start with dataset-level rebalance.
* The rebalance at the batch-level or dataset-level? Start with dataset-level rebalance. (Sample an equal number of non-dog images? Just do this)

• Need to handle class imbalance at test time also. (Sample an equal number of non-dog images? Just do this)

Define each training example is , feed it to the model to get the heatmap

We could get a “purified” input .

Feed into m and compute and activation for each feature unit .

Let , it is the contribution of regards to class .

Overall, for each class c, define

We could get a heatmap to see which group of feature units are more important for each class.