Two Strategies for Large-Scale Multi-label Classification on the YouTube-8M Dataset

Author: Dalei Li
Supervisor: Professor Luis Baumela
Technical University of Madrid

YouTube-8M Video Understanding Challenge

https://www.kaggle.com/c/youtube8m

YouTube-8M

- 7 million videos
  - 120 ~ 500 s
  - 1000+ reviews
- 4716 topics
  - Visually recognizable
- 3.4 topics per video

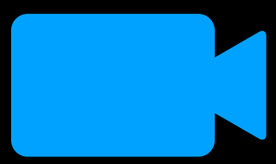
Electric guitar

Frame-level Dataset

- Extracted 1024-dimensional visual 📷 features using GoogLeNet (Inception-V3), per frame per second
- Extracted 128-dimensional audio 🎵 features from a CNN architecture for audio classification, per second

Video-level Dataset

- Average the visual and audio features over the frames of a video

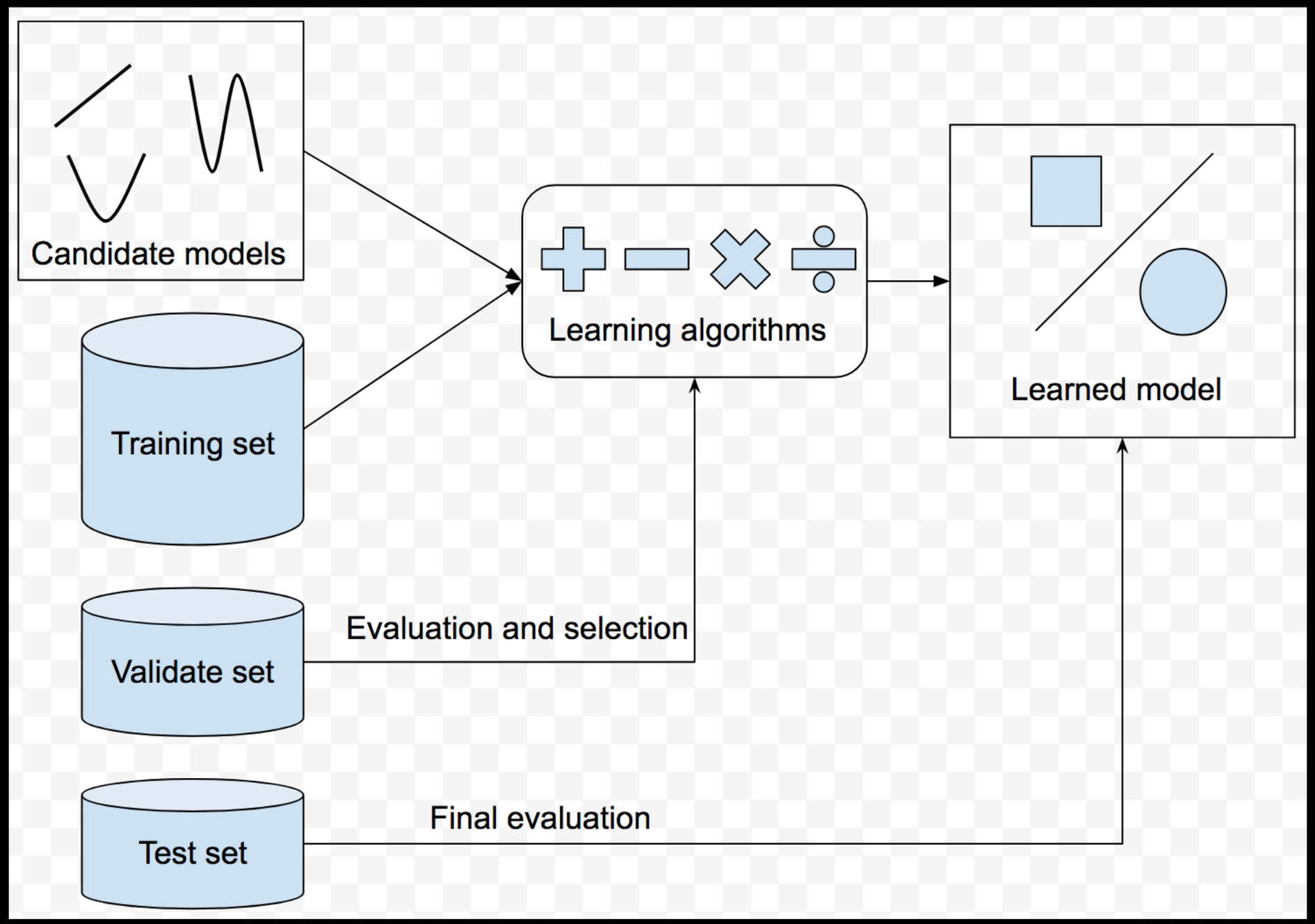


= (1024 + 128)-dimensional vector

Multi-label Classification

- Given the feature vector ***x*** of an instance (video) and a label (topic) *l* from **L**, predict its probability to have *l*

Architecture to Learn A Classification Model



Challenges

- Labels are not mutually exclusive
  - One-vs-rest strategy
- Huge number of instances
  - Streaming instances
  - Incremental learning

One-vs-rest Strategy

- Learn a binary classification model for each label l from $\mathbf{L}$, i.e., $P(l=1|\boldsymbol{x})$
- The instances that have label l are positive, the other instances are negative

Streaming Instances

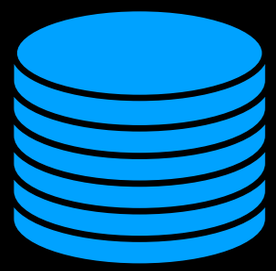
A

Truck

B

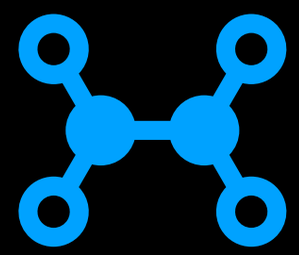
The goods can be split. The process can be applied to t instances one by one.

Training set



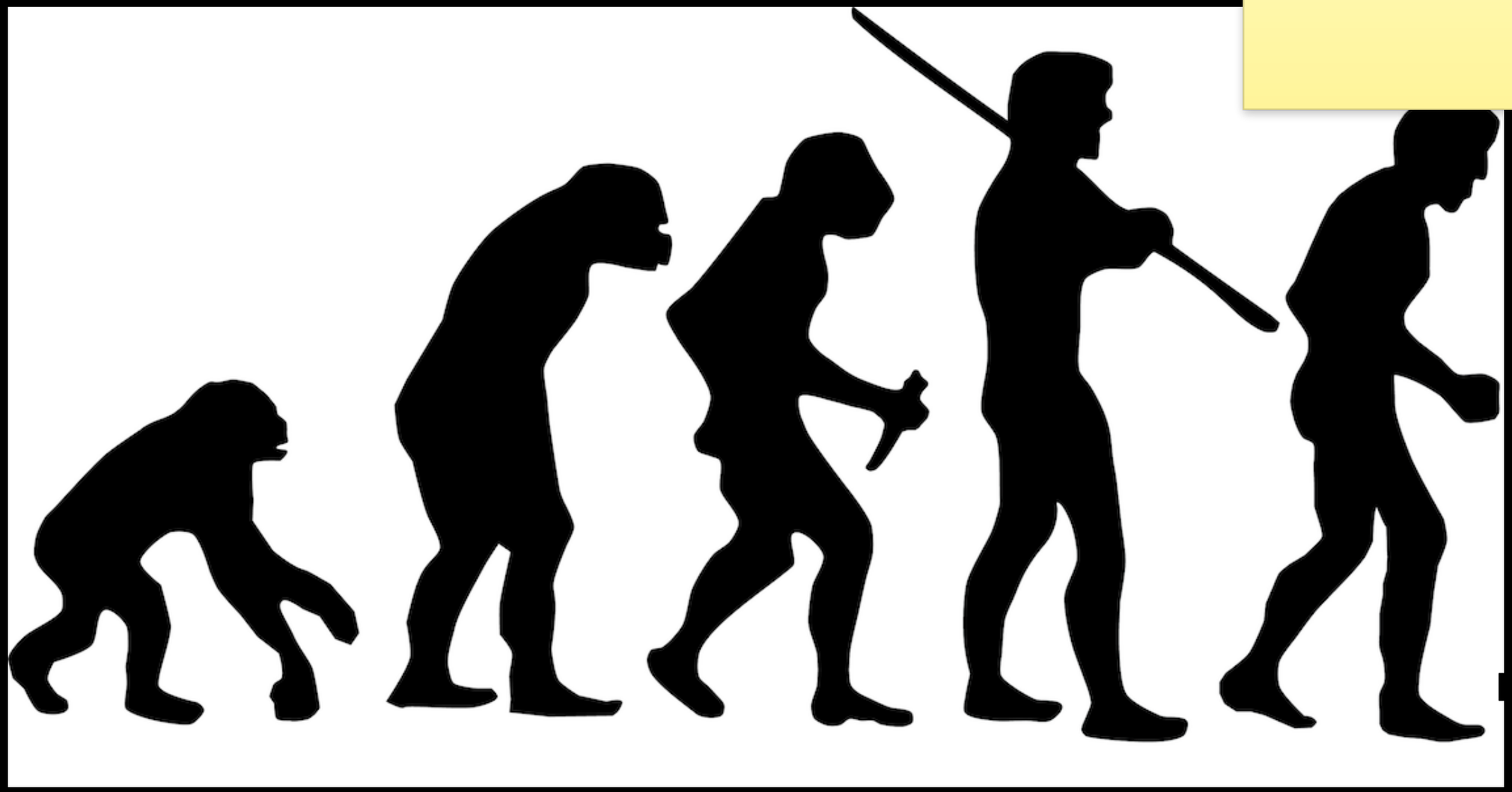
Process

Model



Incremental Learning

Improve the model from new instances (by learning new skills in human evolution).



Models

- Streaming Instances
  - Multi-label k-nearest neighbor (k-NN)
  - Multi-label radial basis function (RBF) network
- Incremental Learning
  - One-vs-rest logistic regression
  - Multi-layer neural network

1. Multi-label k-NN

- One-vs-rest transforms to *m* independent binary classification problems

- Binary k-NN

  - Assume when an instance ***x*** has label *l* or not its *k* nearest neighbors have different characteristics

  - Given an unseen instance ***x*,** observe the characteristics of its *k* nearest neighbors to determine it has *l* or not

the instances that have the same label are usually more similar than the instances that have different labels. We assume the instances that are similar tend to have the same label. Similarity metric: Cosine similarity

Binary k-NN Idea

The assumption is formulated to

- Among the k nearest neighbors, the number C_l of instances that also have label l follow different distributions when an instance $\boldsymbol{x}$ has label l or not
- Compute the posterior probability $P(l=1|\boldsymbol{x})$ using these distributions

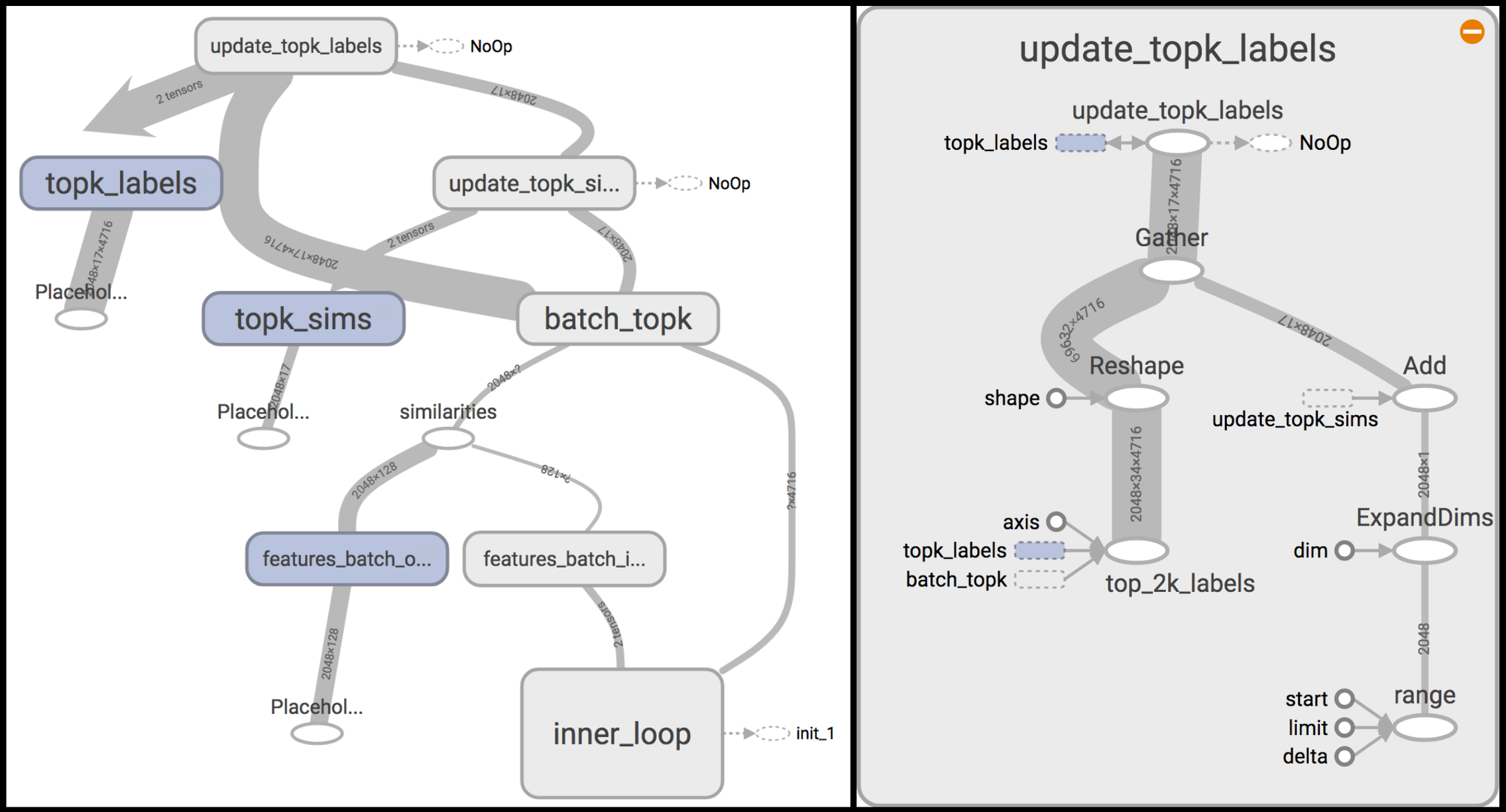
Binary k-NN Model

- Prior distribution - without looking at an instance $\boldsymbol{x}$, the probability distribution its label / follows
- Likelihood - knowing the label of $\boldsymbol{x}$, the distribution of the number C_l follows
- Posterior distribution - knowing the number C_l of $\boldsymbol{x}$, the distribution / follows

Implementation

- Prior distribution - while streaming the training instances into a sequence of batches, compute the sum of label and count
- Likelihood - compute C_l for each training instance with (without) label *l* and compute the proportion of possible each value, i.e., 0,…,*k*
- Posterior distribution - compute based on the prior distribution and likelihoods

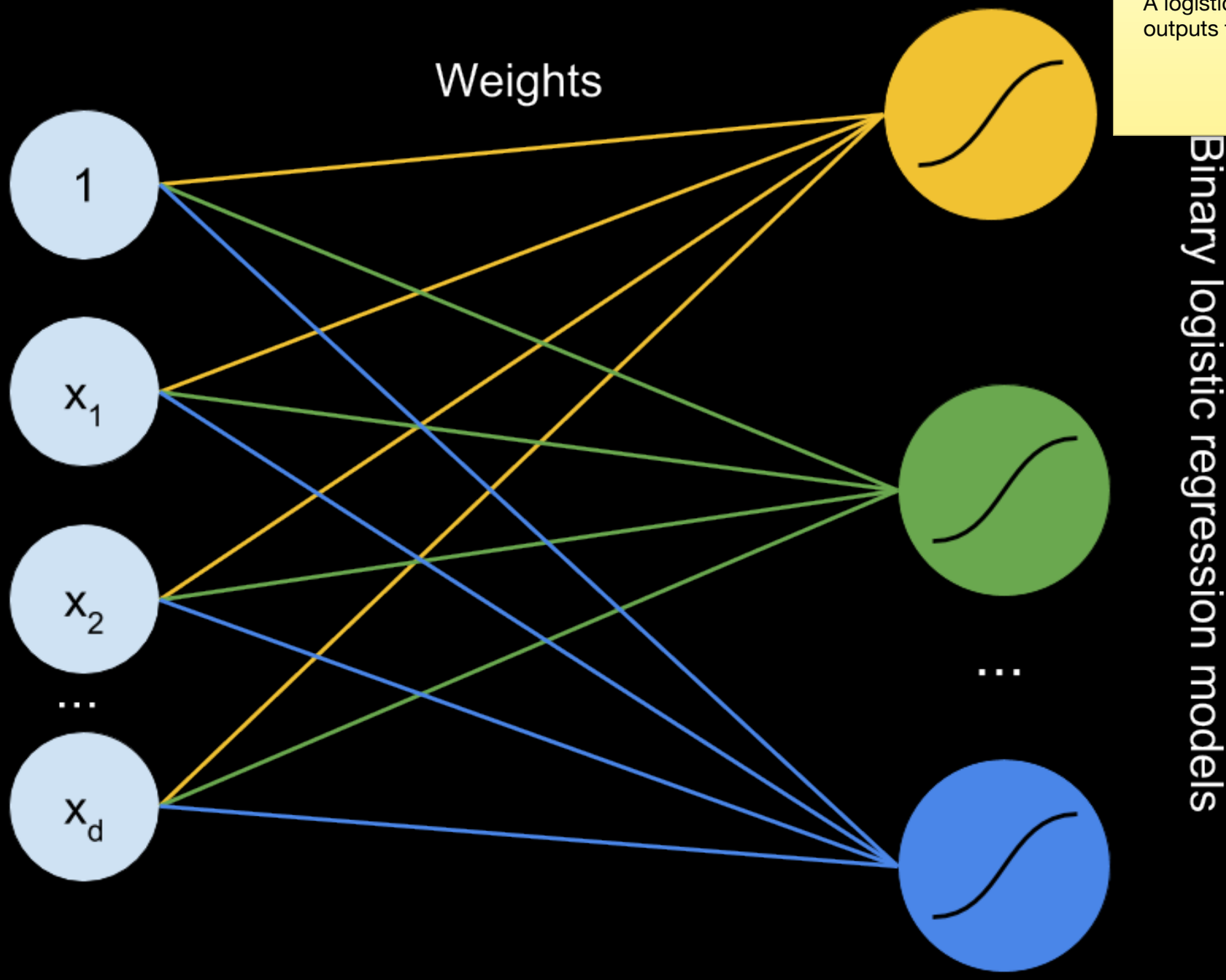
Implementation



The Tensorflow graph to find *k* nearest neighbors for a batch of training instances

2. Logistic Regression

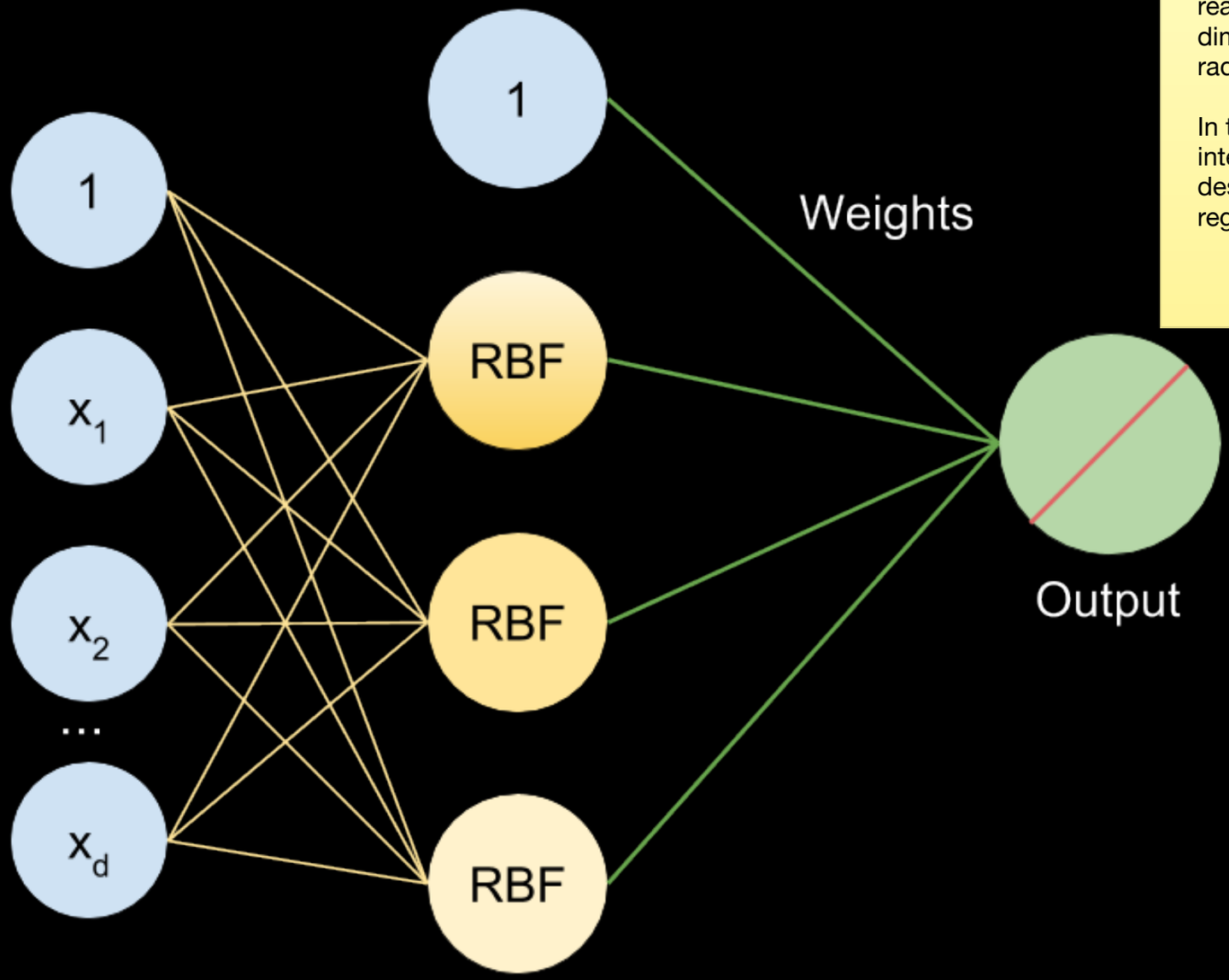
A logistic regression model naturally outputs the posterior probability.



3. Multi-label RBF Network

An RBF network was proposed to fit a real-valued function in a high-dimensional space with a group of radial basis functions.

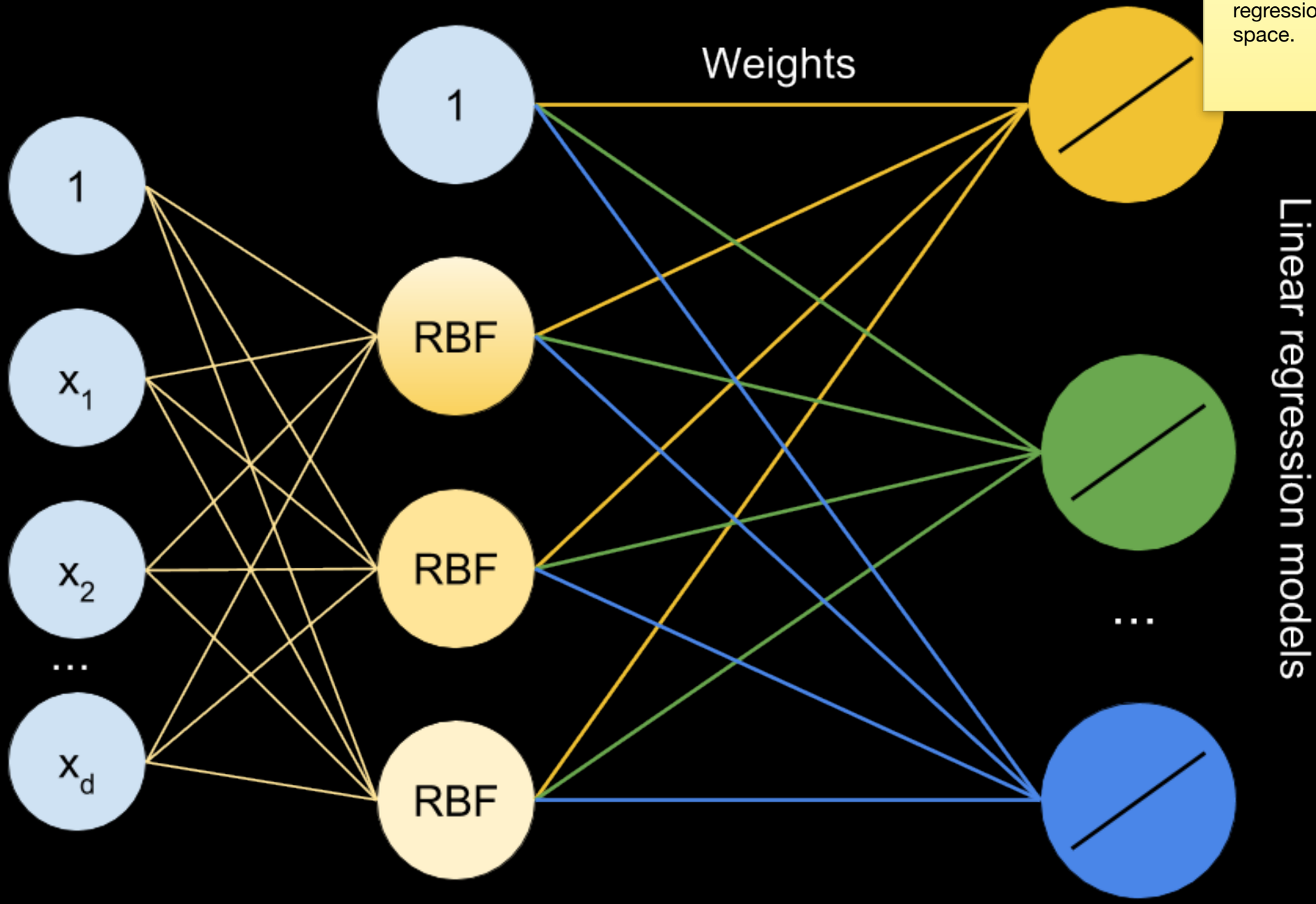
In the known data points, the interpolation value should equal the desired value. In essence, it is linear regression in the transformed space



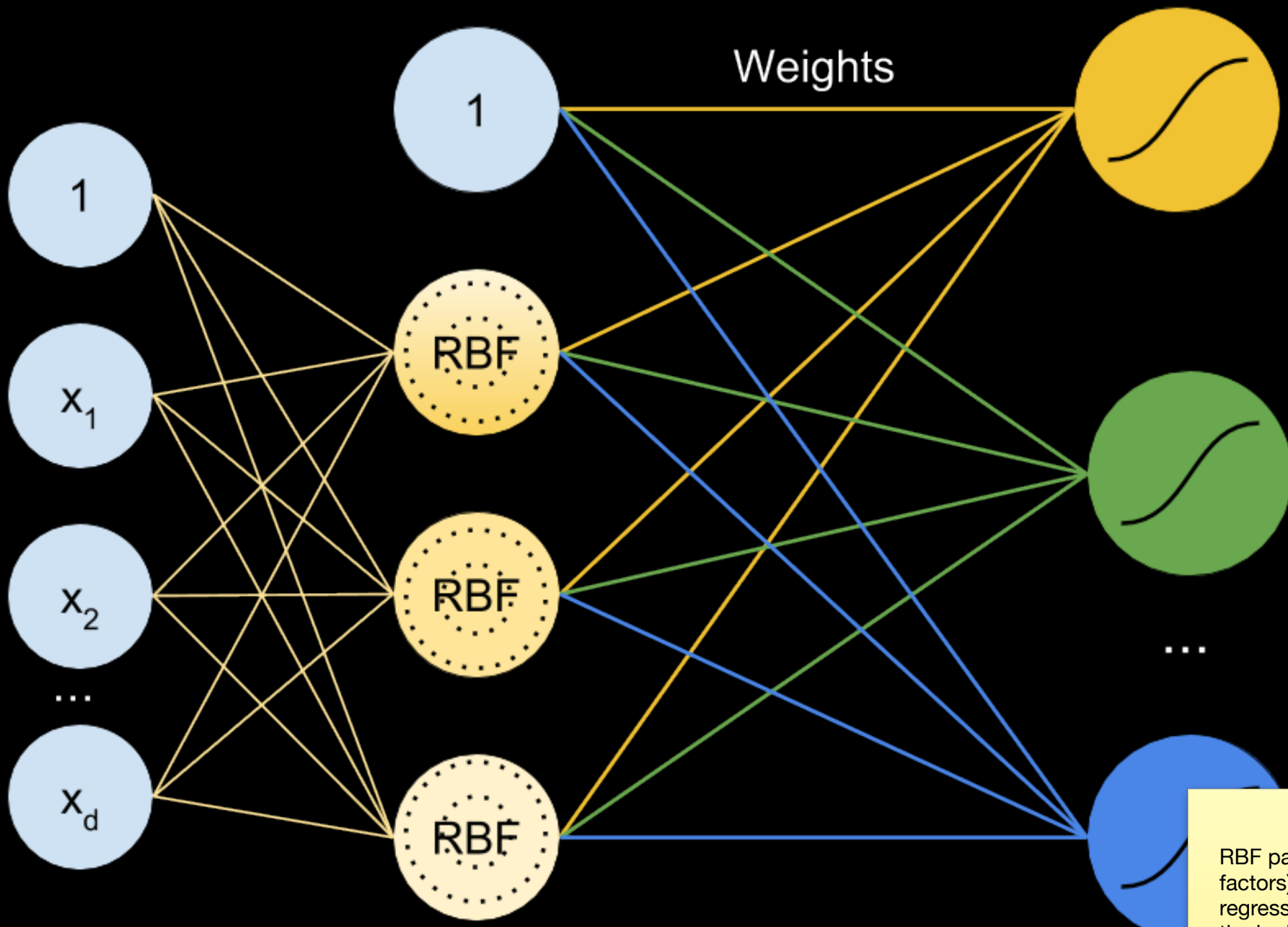
RBF network

Multi-label RBF Network

In essence, it is linear regression in the transformed space.



Three-phase Learning



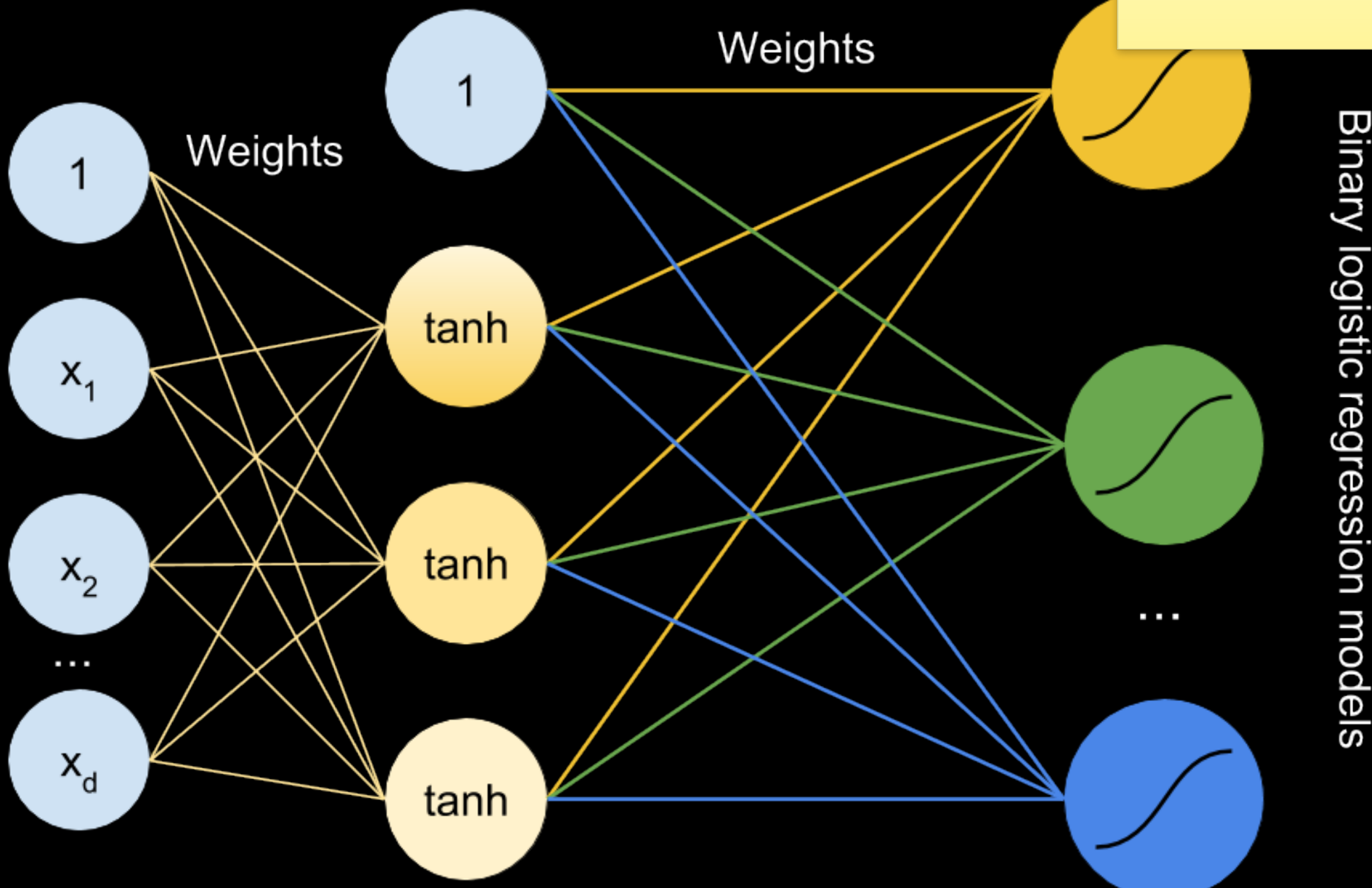
RBF parameters (centers and scaling factors) can be tunable. The linear regression result is used to initialize the logistic regression models.

Implementation

- Combine ML-RBF network proposed by Zhang et al. and three-phase learning by Schwenker et al. …
- The binary logistic regression models share the same group of centers
- Replace squared error in fine-tuning with cross entropy

4. Multi-layer Neural Network

RBF transformations are replaced with activation functions, such as sigmoid, rectified and tanh.



Experimental result

- Evaluation metric: global average precision
- ML-kNN
- Logistic regression
- Multi-label RBF network
- Multi-layer neural network

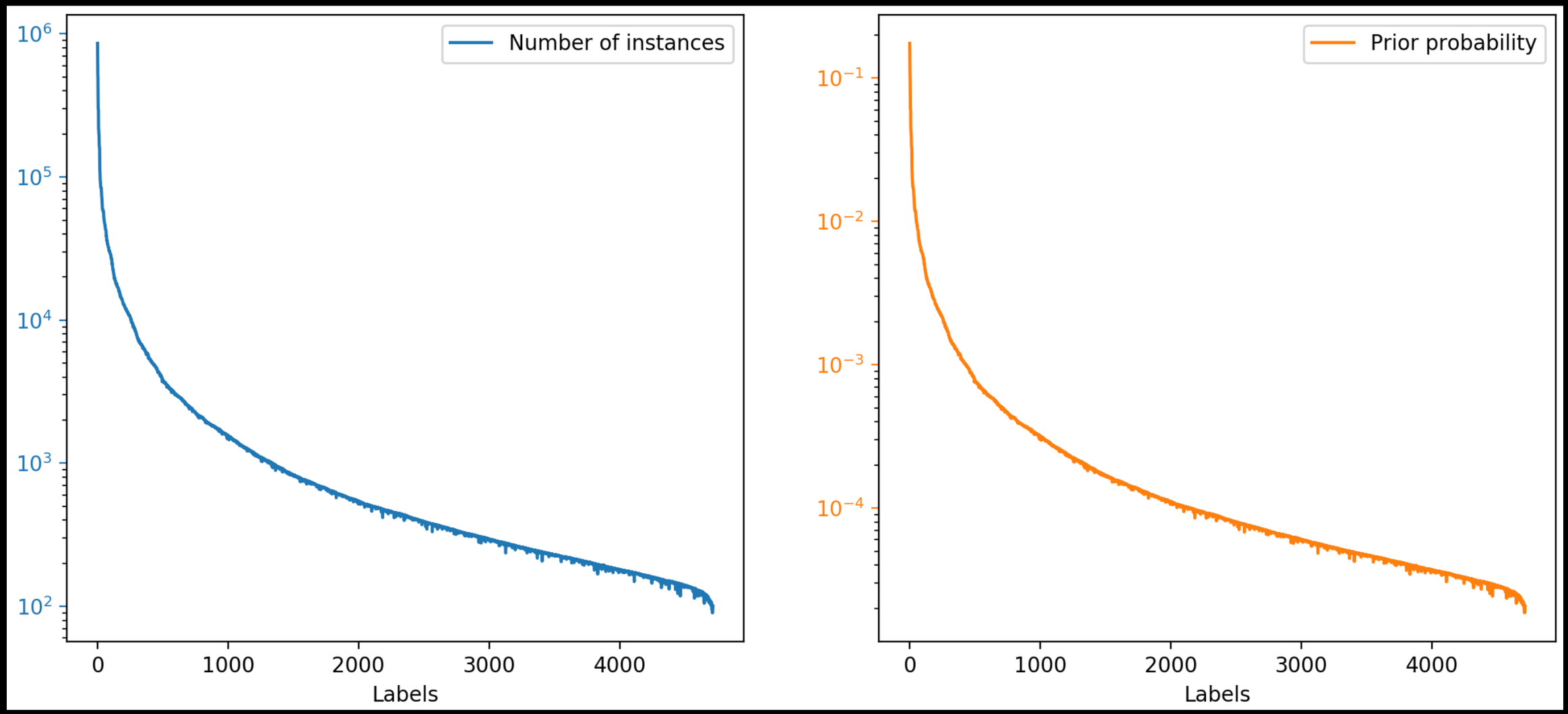
1. Global Average Precision

- Binary classification
  - Precision = true positive / (true positive + false positive)
  - Recall = true positive / (true positive + false negative)
- Average precision

$$\sum_{j=1}^{20} precision(j) \times \Delta recall(j)$$ **Over** $$\{(l_{\tau_i}, p_{\mathbf{x}}(l_{\tau_i}))\}_{i=1}^{20}$$

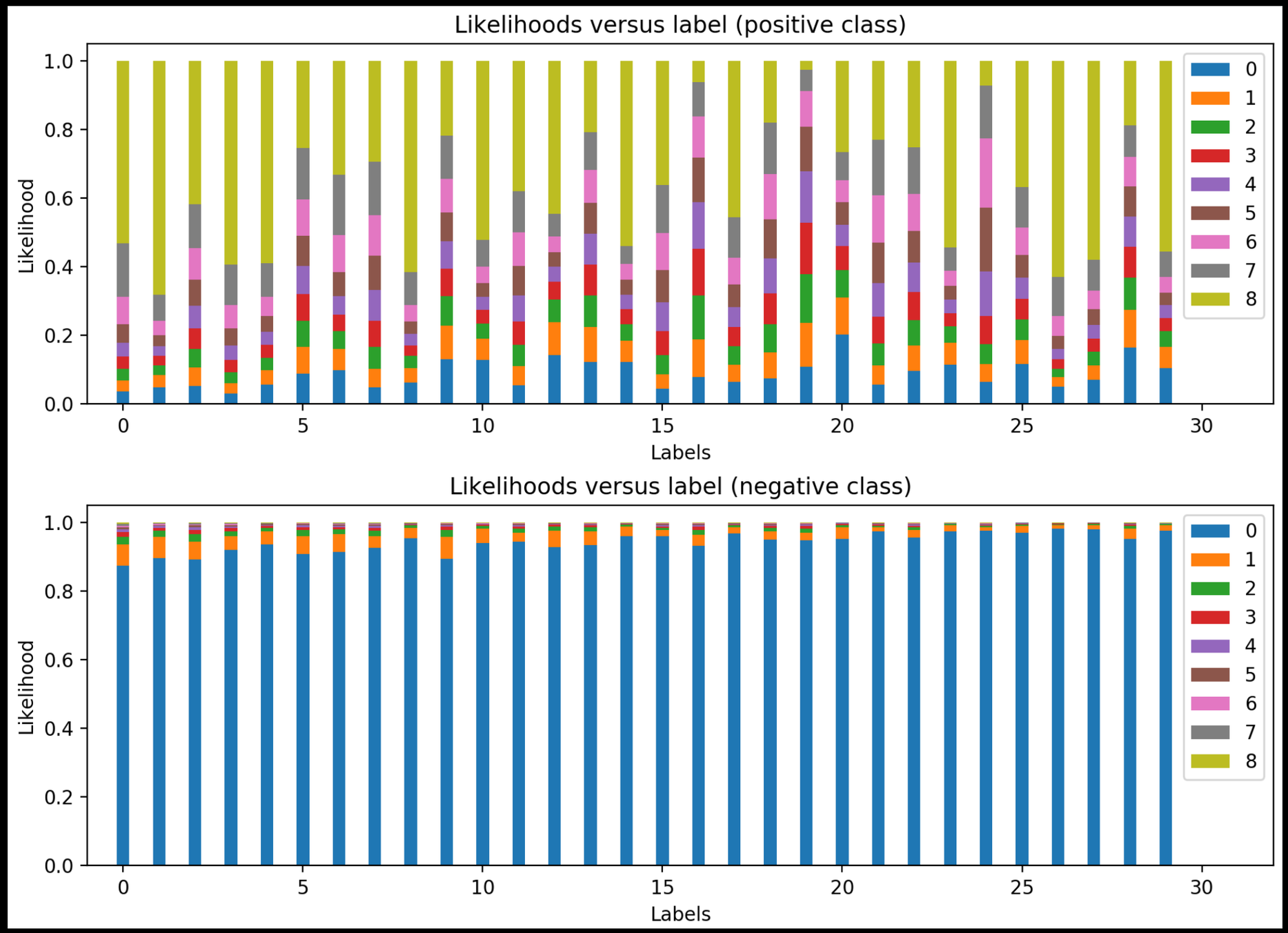
- Global average precision
  - Concatenate the predictions for the test set and sort them based on the posterior probability and compute the average precision

2. ML-kNN - model



Prior probability (smooth parameter s=1.0)

Likelihood - first 30 labels

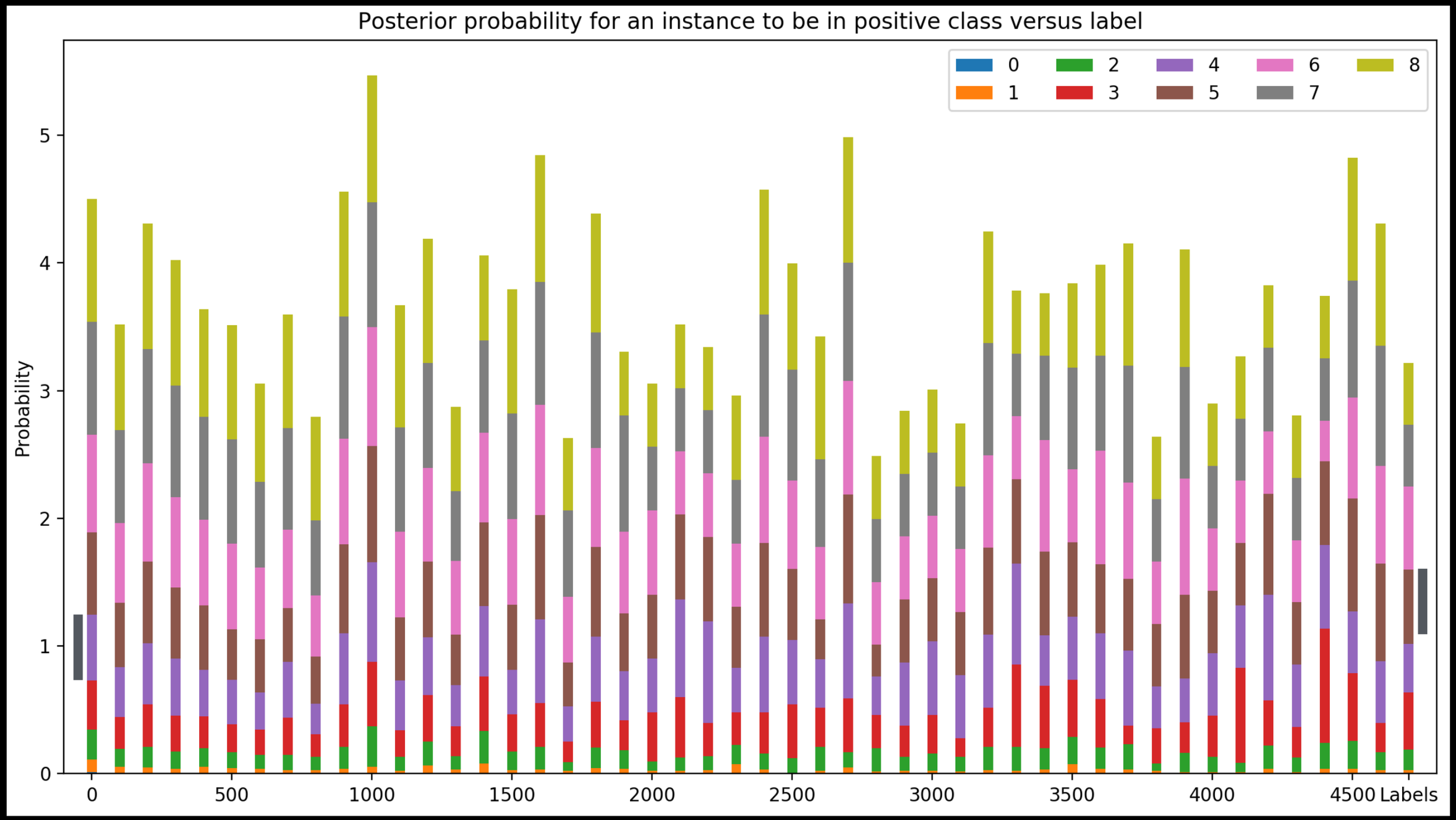


k=8, s=1.0

Likelihood - last 30 labels

k=8, s=1.0

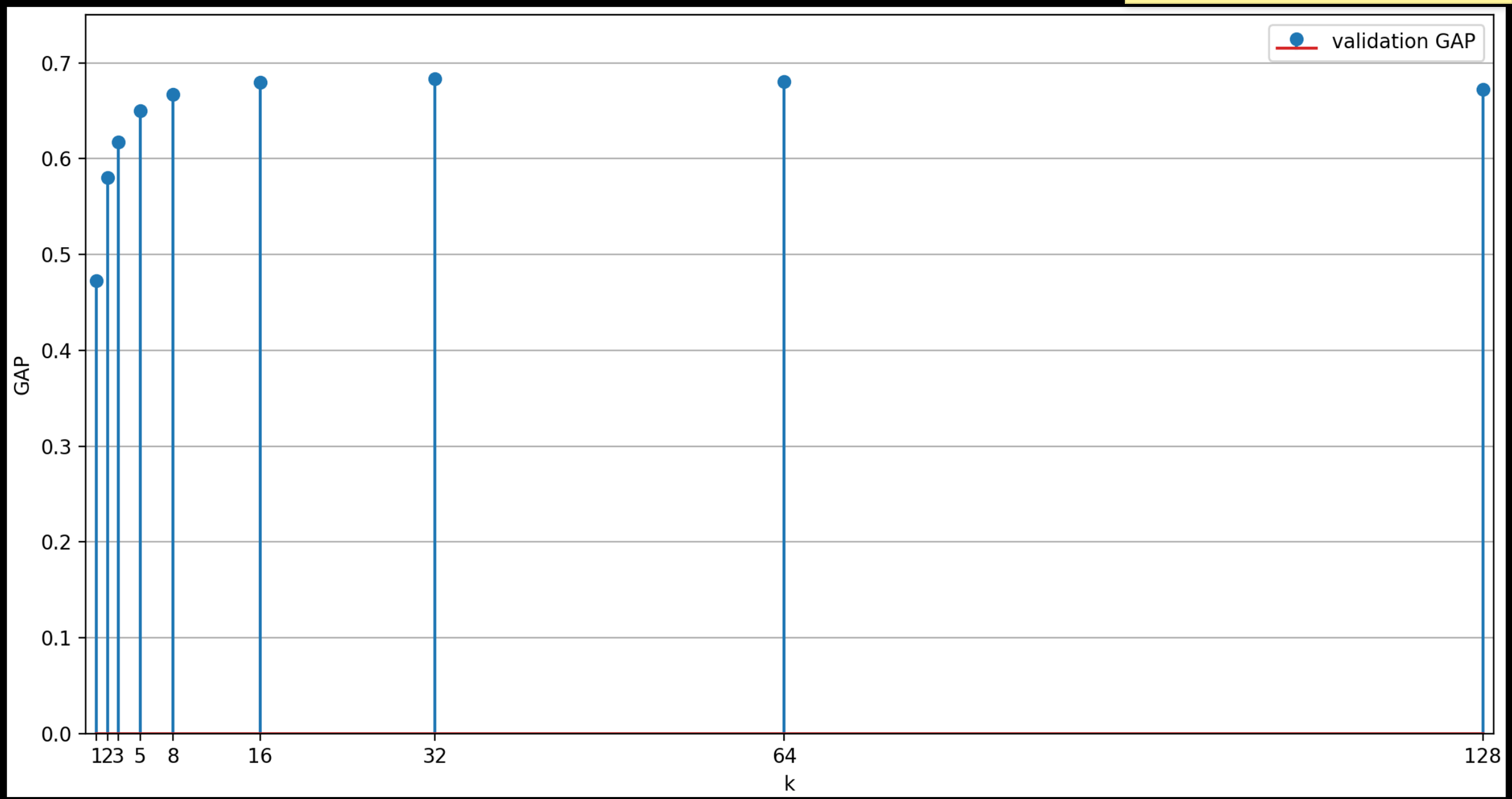
Posterior probability



k=8, s=1.0

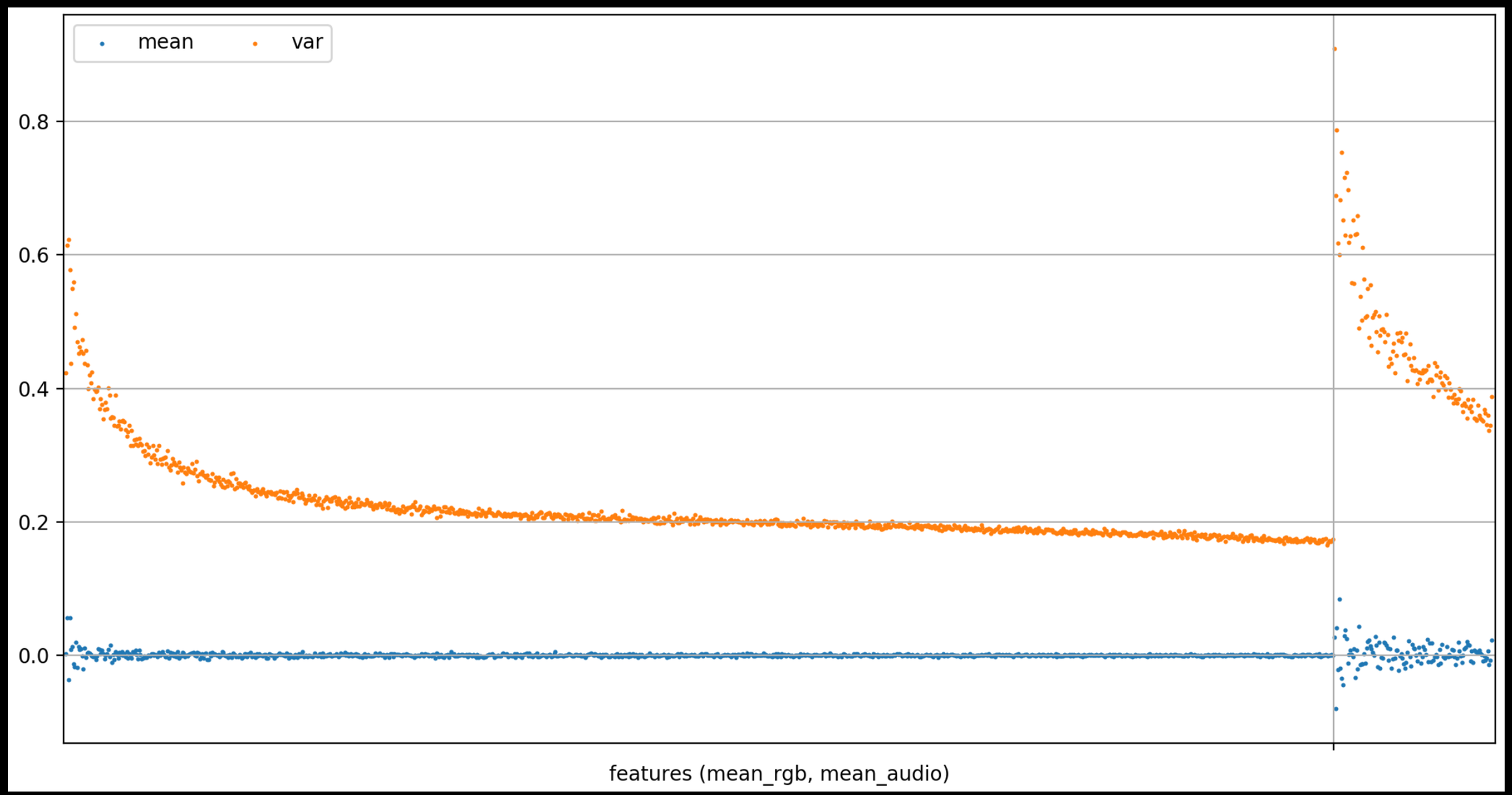
Tune k

0.4723, 0.5805, 0.6178, 0.6497, 0.6673, 0.6802, 0.6837, 0.6805, 0.6721.



Validation GAPs (s=1.0)

3. Logistic regression - standard scale



Features mean and variance in the training set

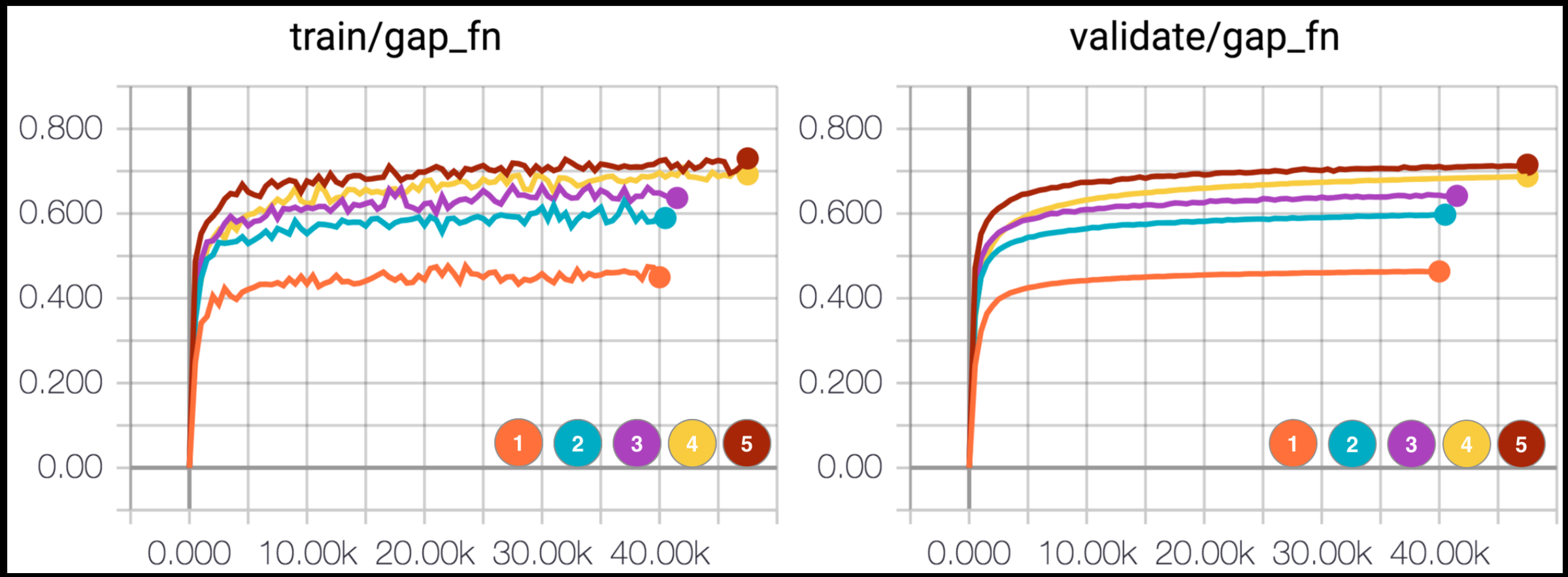
Result

Case	Initialization method	Standard scale	Misc	Best validation GAP
1	random	no	N/A	**0.6554**
2	random	yes	N/A	0.5882
3	linear regression	no	N/A	**0.6555**
4	linear regression	yes	N/A	0.5885
5	random	no	weighted classes	0.5560
6	random	no	instances weighting	0.6397

Comparison among different logistic regression models

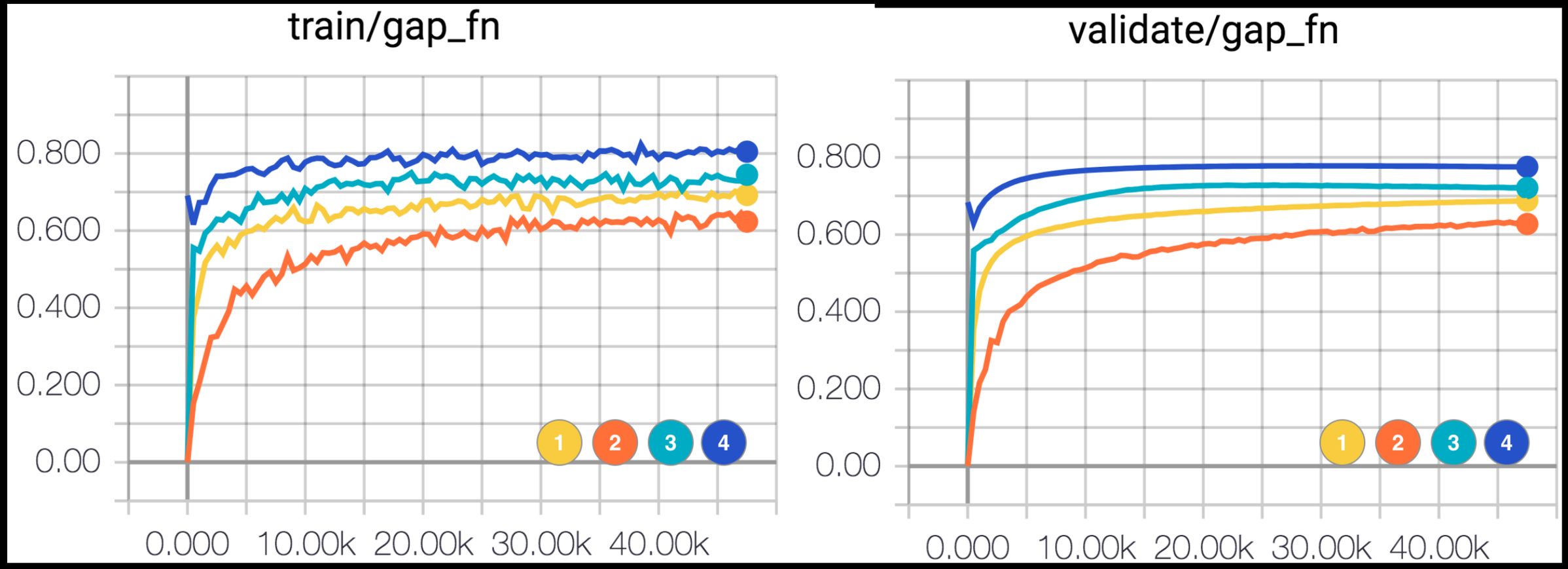
4. Multi-label RBF Network

GAPs are 0.46, 0.60, 0.64, 0.68, 0.71, respectively.



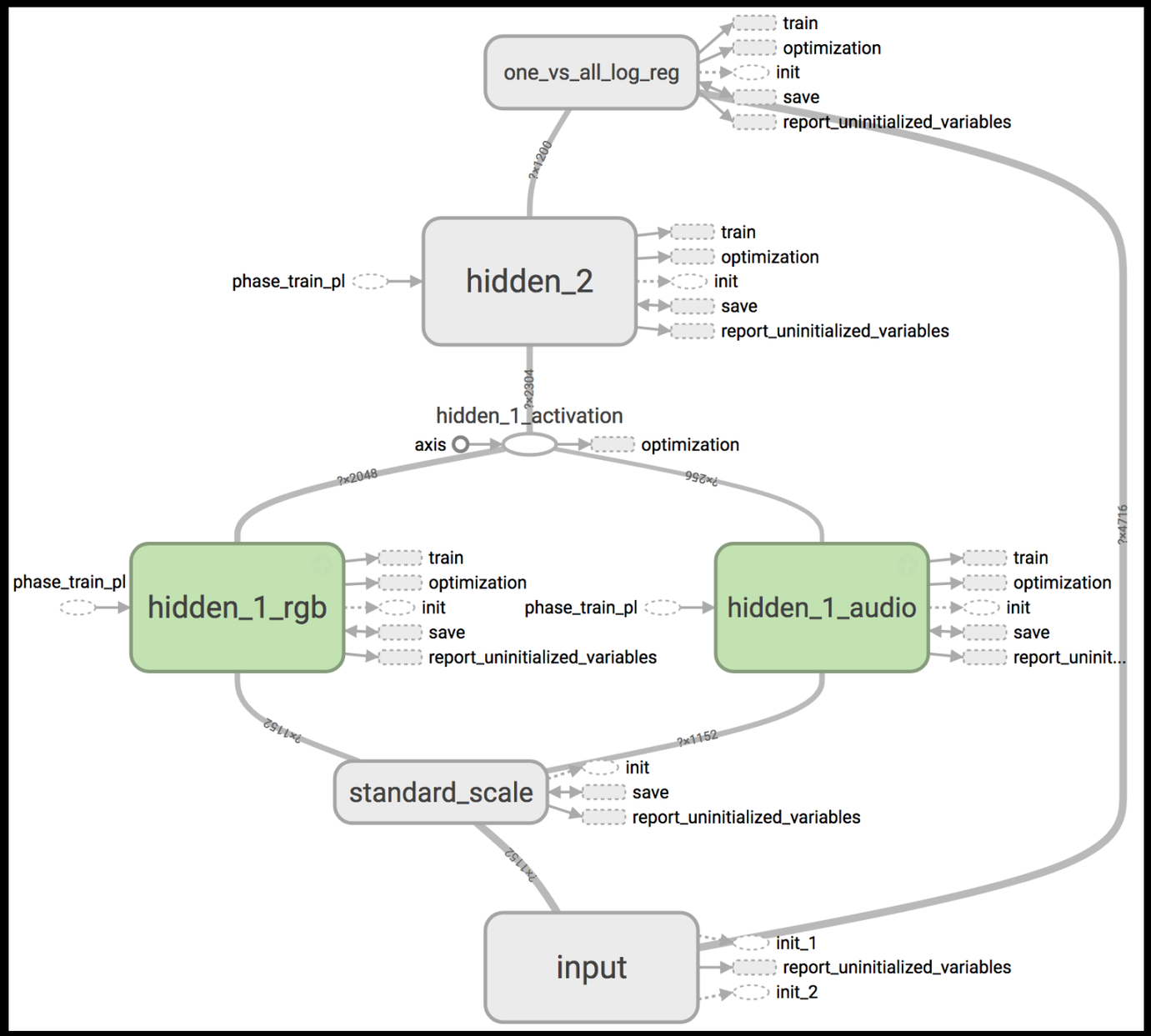
Training and validation GAPs versus training steps (proportion of centers 1e-5, 5e-5, 1e-4, 1e-3, 5e-3)

Centers selection and fine-tuning

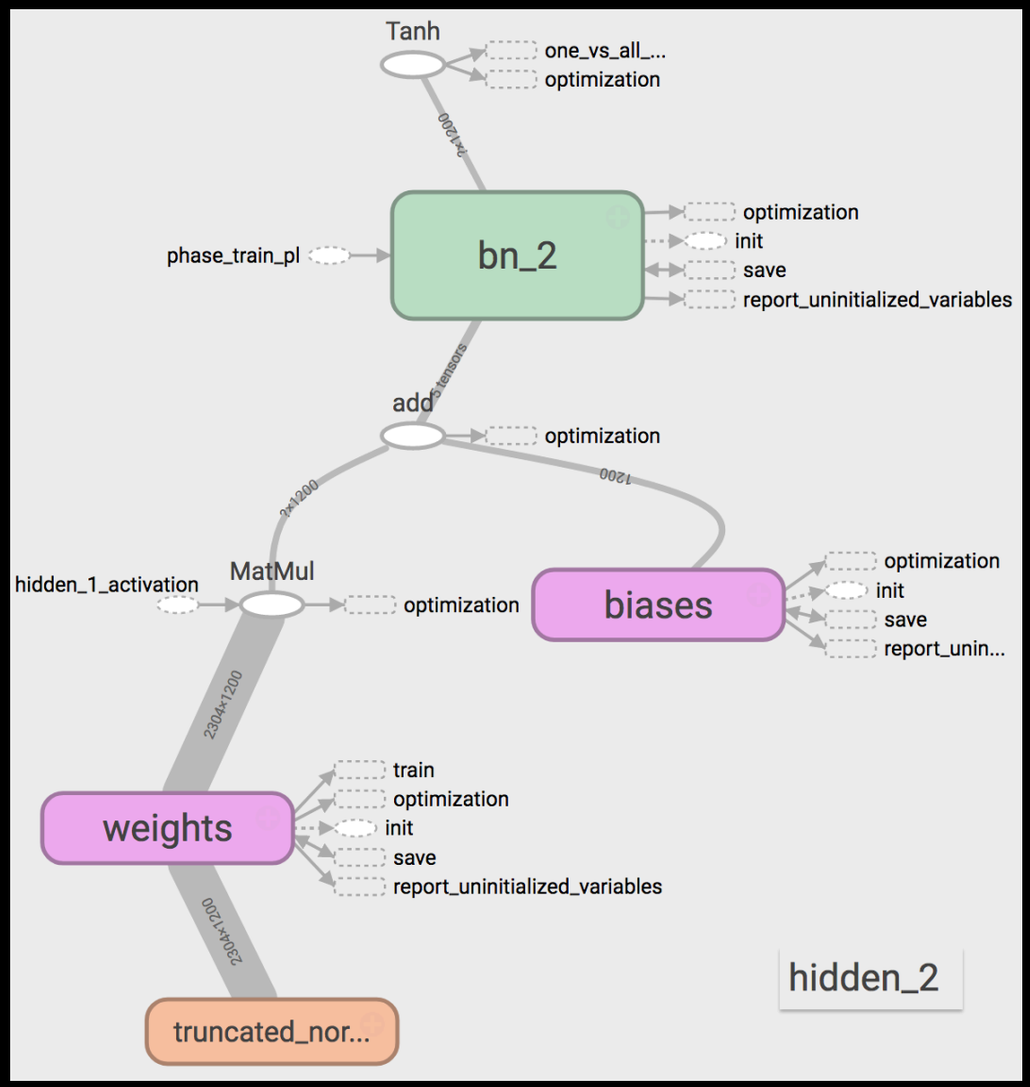


Case	#Centers	Mean distance	Weights initialization	Misc	Validation GAP
1	3753 (k-means)	0.4844	truncated normal	N/A	0.6827
2	4862 (random)	N/A	truncated normal	N/A	0.6243
3	3859 (k-means)	0.4839	truncated normal	standard scale	0.7235
4	3919 (k-means)	0.4832	linear regression	fine tuning	**0.7769**

5. Multi-layer Neural Network

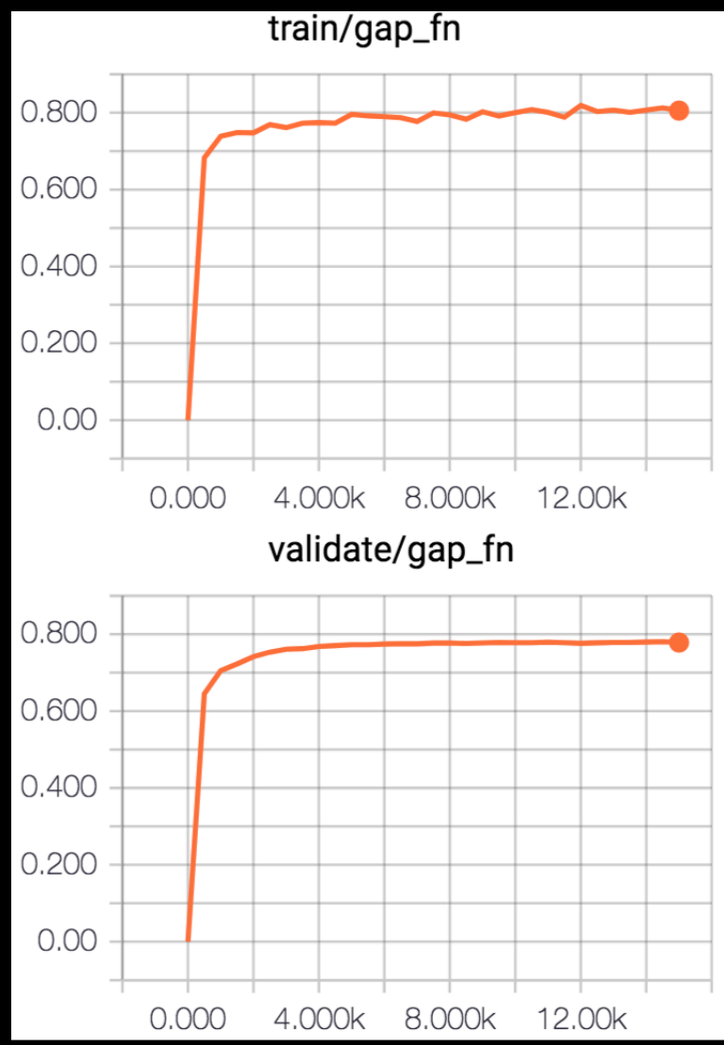


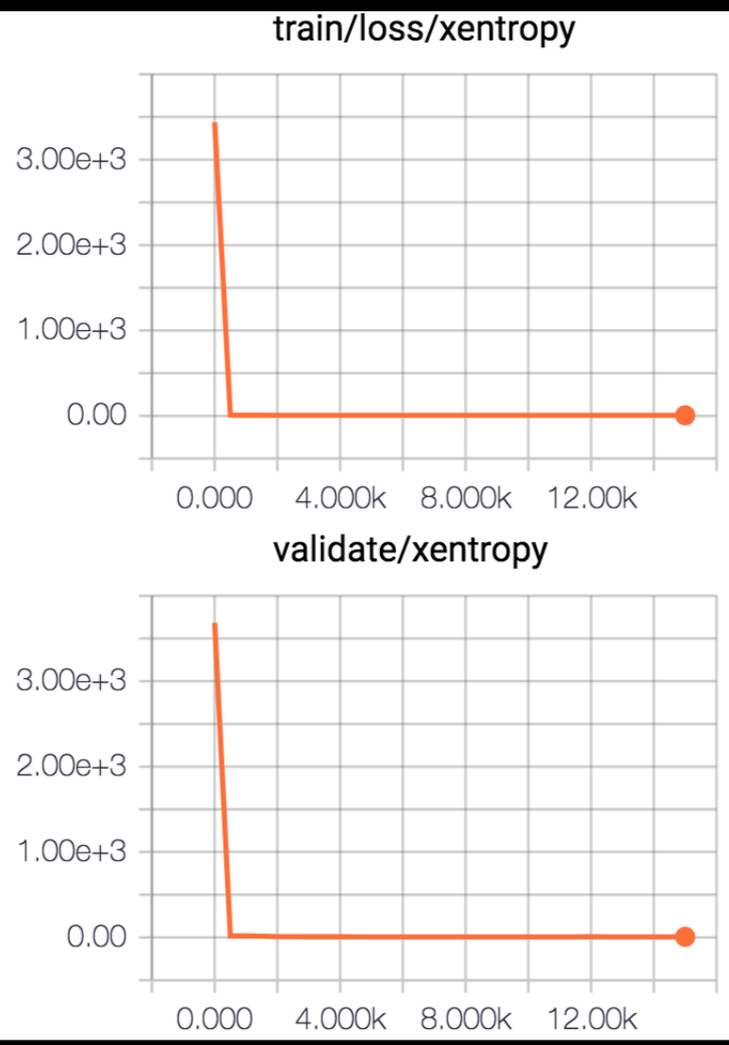
Architecture



Hidden_2 layer

Training and validation





Dropout and Bagging

- Applying dropout improves the performance slightly
- Bagging of six neural networks achieves the private GAP 0.801 (ranked at 111 / 650 teams)

Comparison with other methods

- Using frame-level data, 1.7 TB
- Concatenate raw features to hidden layer activations (residual network)
- Explore labels dependency
- Ensemble much more models and the same model at different training steps

Summary

- Implemented ML-kNN that supports tuning k (passing a list of *k*)
- Implemented k-means that supports removing empty or small clusters during each iteration and supports Cosine and Euclidean distance
- Implemented linear regression that does not require data to be shifted and supports tuning l_2 regularization (passing a list of l_2 reg. rates)
- Implemented one-vs-rest logistic regression that supports class weights and instance weights. More importantly, it supports any feature transformation implemented in Tensorflow
- Implemented stable standard scale transformation that supports any feature transformation
- Implemented multi-label RBF network that supports three-phase learning and experimented with the hyper-parameters, such as the number of centers and scaling factors
- Implemented multi-layer neural networks and experimented with architecture, batch normalization and dropout and bagging
- Finally, the implementations can be easily adapted to other large datasets

Future Work

- Make use of frame-level dataset
- Larger *k* in ML-kNN
- Exploit more centers and the centers from different clustering results in RBF network
- Exploit the validation set
- Explore other ensemble methods

Acknowledges

- Professor Luis Baumela, Marta Patiño
- Friend Xinye Fu, Xiaofeng Liu
- The algorithms authors
- Etc.

References

- [1] ML-kNN,Zhang, Min-Ling and Zhou, Zhi-Hua. “ML-KNN: A lazy learning approach to multi-label learning”. In: Pattern recognition 40.7 (2007), pp. 2038–2048.
- [2] Three-phase learning RBF network, Schwenker, Friedhelm, Kestler, Hans A, and Palm, Günther. “Three learning phases for radial-basis-function networks”. In: Neural networks 14.4 (2001), pp. 439–458.
- [3] Multi-label RBF network, Zhang, Min-Ling. “ML-RBF: RBF neural networks for multi-label learning”. In: Neural Processing Letters 29.2 (2009), pp. 61–74.
- [4] YouTube-8M dataset, Abu-El-Haija, Sami et al. “Youtube-8m: A large-scale video classification benchmark”. In: arXiv preprint arXiv:1609.08675 (2016).
- Open course Neural Networks for Machine Learning by Geoffrey Hinton, https://www.coursera.org/learn/neural-networks
- Open course Machine Learning by Hsuan-Tien Lin, https://www.csie.ntu.edu.tw/~htlin/course/ml15fall/
- Open course Data Mining by Jia Li, http://www.personal.psu.edu/jol2/course/stat557/material.html
- Etc.

Thank you!