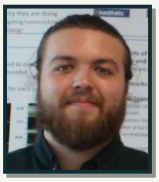
Margin-Mix: Semi-Supervised Learning for Face Expression Recognition

Corneliu Florea



Mihai Badea

Laura Florea

Andrei Racovițeanu

Constantin Vertan

The proposed method, called Margin Mix, addresses the problem of semi-supervised learning for face expression recognition. It has been developed by a team from Image Processing and Analysis Laboratory (LAPI) from Polytechnical University of Bucharest.

Problem and approach

Problem: **Semi-supervised learning**

- Some data has labels
- Most of the data is unlabeled

Approach

- Ask ***simultaneously*** for good predictions and discriminative embeddings
- Use embeddings clustering to self-label unlabeled data while aiming to low density area separation

Our method uses a deep convolutional network in a standard case of semi-supervised learning. We recall that semi-supervised learning assumes that some data has labels, while the rest, drawn from the same distribution, does not have labels.

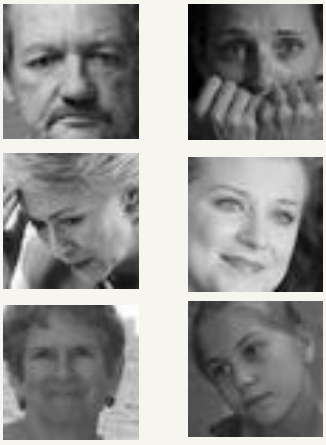
The solution is built upon several ideas.
The first idea refers to the fact that a convolutional network can simultaneously provide predictions and embeddings which are trained on the labelled data to be discriminative. Next, the network self-labels the unlabeled data using the relative distance to class centroids in the embedding space. It also refines embeddings such that it seeks to create low density areas, where the separation curves are placed.

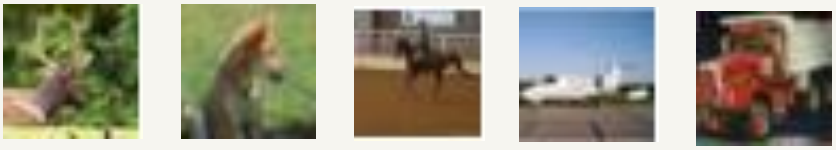
Task

Face expression recognition:

- Labels are: **neutral, anger, fear, disgust, happy, sad, surprise, contempt**

- Susskind showed that psychology student reached **89.2**% accuracy in a 6 expressions experiment
- Untrained user achieves 94% accuracy on CIFAR images on 10 class problem

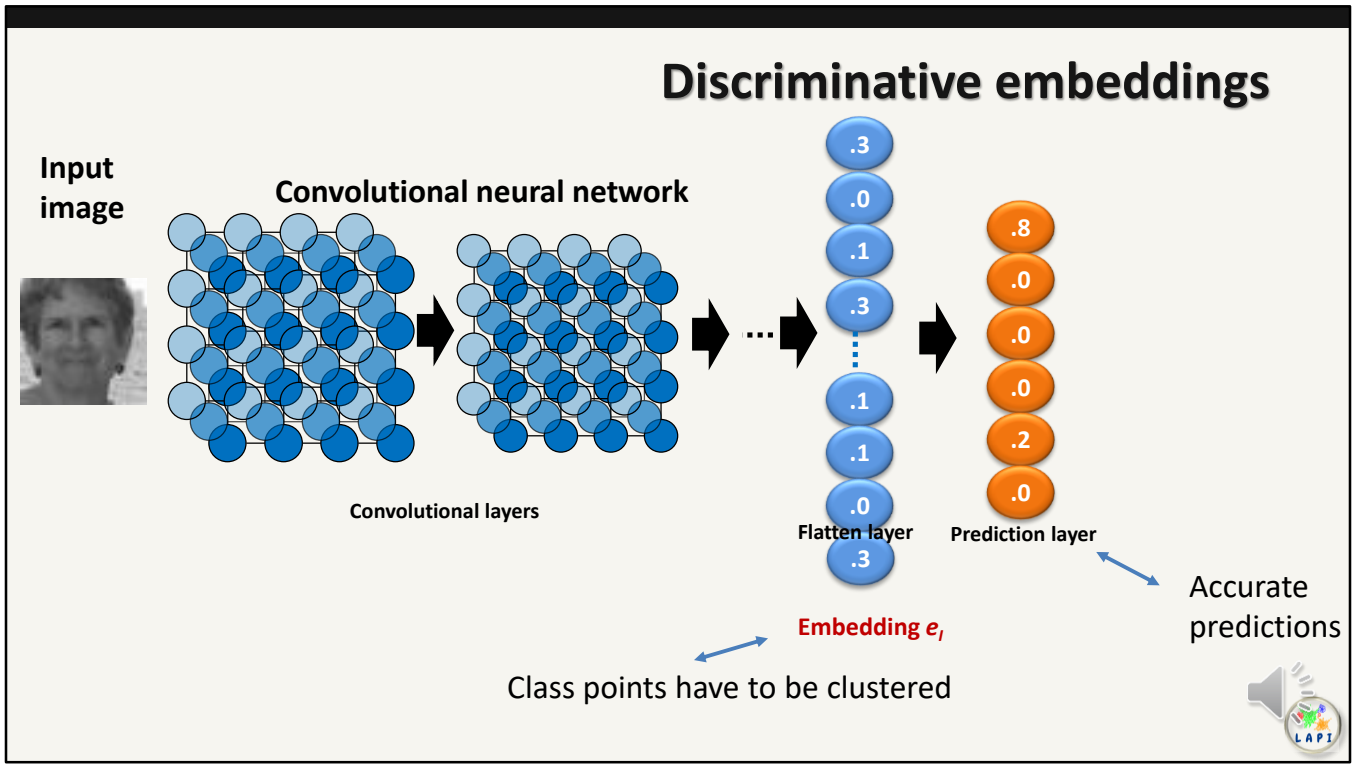




This method is built heaving in mind harder problems such as face expression recognition. Here, we try to associate one of the basic expression (listed with colored letters) to each image. To argue why we call it a harder problem let us do two things:

First, we kindly ask the viewer to try to identify the expression in the images from the right. Also try to identify the category of the CIFAR images from the bottom and compare the easiness to do these tasks.

Second, let us recall two attempts to manually classify data. An experiment recorded by Susskind showed that **trained** user reach almost 90% accuracy in recognizing 6 expressions while **untrained** users reached, the better performance of 94 % in recognizing CIFAR images in a 10-class experiment.



Now, let us go into some technical details, as we try to explain how our method works. Again, our method is built for deep learning.
Given a deep convolutional neural network and an input data, we take the associate embeddings from the last layer before the prediction one. As we use architectures from the residual network family, the embedding layer needs to be flatted so it will produce a vector.
Training the network assumes a dual task:

- provide accurate predictions and,
- construct embeddings that are discriminative, which means that the cluster data instances according to their classes.

Loss Function

$$L = LS + \lambda L_M$$

Total Loss — cross entropy

Margin Loss:

$$L_M = \sum_{i=1}^{N} \left(D(\hat{e}_i, c^C) - \frac{1}{C-1} \sum_{j=1, j \neq i}^{C} (\hat{e}_i, c^j) \right)$$

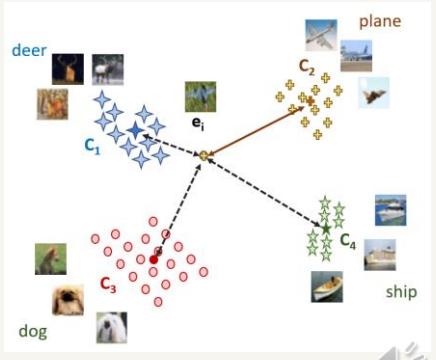
Where

Normalized embeddings $\hat{e}_i = \frac{e_i}{\|e_i\|}$

Centroids $c_i = \frac{\sum_{i=1}^{C} e_i \mu_i{}^C}{\sum_{i=1}^{C} \mu_i{}^C}, \mu_i{}^C = \begin{cases} 1, yi = C \\ 1, yi \neq C \end{cases}$

$D(x_1, x_2)$ - Euclidean distance

Behavior: The margin loss pushes points towards their centroid



The way to force the network to do these things in the same time is by the loss function used.
The total loss used is composed from the standard cross entropy loss and from the margin loss. The cross entropy seeks good predictions.
The margin loss is our proposal. It asks an embedding to be close to the centroid of its class and far from the centroids of other classes. This behavior is illustrated in the figure from right. PAUSE

Instead of the original embedding, we use normalized embeddings to prevent convergence toward no separation by scaling of the norms.
Class centroids are computed according to the standard fuzzy C-means algorithm.

Auxiliary concepts

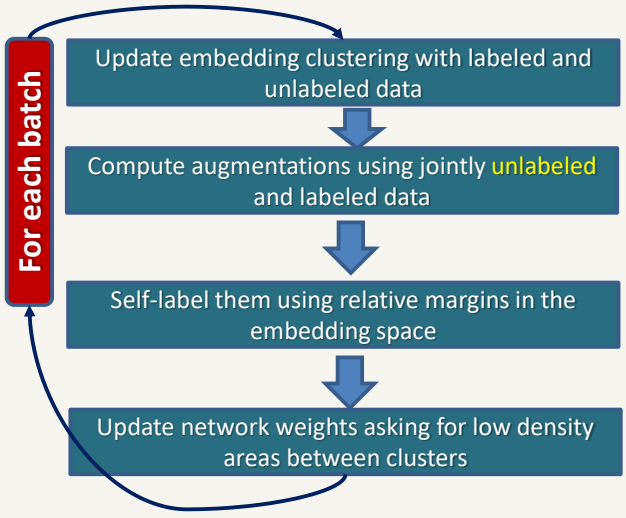
- Self-labeling - the network itself is used to provide labels for unlabeled data
- MixUp – augment each data instance by considering convex combinations between instances and predictions
- MixMatch – previous SSL solution, which creates new points by combining a labelled data and an unlabeled one

These ideas are integrated in the algorithm. This, in order to work, requires several other auxiliary concepts.
Our method relies on several ideas previously introduced such as:

- Self-labeling - the network itself is used to provide labels for unlabeled data. The training process, thus become an attempt to move away points from the border, thus creating spaces without data near the separation lines

- MixUp – augment each data instance by considering convex combinations between instances and predictions. This technique dramatically increases the population and thus permits much better sampling of the space. Intuitively, a learner has too much freedom to chose borders in a space with too few points. Increasing population size, we limit the arbitrary in leaner behavior.

Margin Mix algorithm



Algorithm 1: The MarginMix algorithm takes as input a batch of labeled data $\mathcal{X}$ and one without labels $\mathcal{U}$ and produces densely sampled input examples $\mathcal{X}'$ respectively self-labeled densely sample examples $\mathcal{U}'$. Self-labeling is based on clustering in the embedding space. The purpose is to adjust the weights of learner ψ

Data: : Batch of b labeled instances with embeddings and one–hot labels $\mathcal{X} = \{\dots,(\mathbf{x}_i,\mathbf{e}_i,\mathbf{y}_i),\dots\}$, $i = 1\dots b$, batch of b unlabeled instances $\mathcal{X}^u = \{\dots,(\mathbf{x}_i^u),\dots\}$, sharpening temperature T , number of augmentations N_{Aug}, β distribution parameter α for MixUp.

for $b = 1 : N_{batch}$ **do**
  Compute embeddings for labeled samples $\mathbf{e}_b = \psi(\mathbf{x}_b)$;
  Update centroids: $\mathbf{c}^c = \frac{\sum_{i=1}^{N} \mu_i^c \mathbf{e}_i}{\sum_{i=1}^{N} \mu_i^c}$; $\mu_i^c = \begin{cases} 1 \,, y_i = c \\ 0 \,, y_i \neq c \end{cases}$;
  $\tilde{\mathbf{x}_b} = \text{Augment}(\mathbf{x}_b)$ *# data augmentation to* $\mathbf{x}_b$;
  for $k = 1$ *to* N_{Aug} **do**
    $\tilde{\mathbf{x}_k^u} = \text{Augment}(\mathbf{x}_k^u) : \mathbf{x}' = \lambda \mathbf{x}_i^l + (1-\lambda)\mathbf{x}_j^u$ *# one of the k-th data augmentation to* $\mathbf{x}_k^u$;
    Self-label $\mathbf{x}'$ by : $y_i^c = \sum_{k=1}^{N_{aug}} y_k^c = \sum_{k=1}^{N_{aug}} \frac{1}{\sum_{j=1}^{C} \left(\frac{\|\mathbf{e}_i^k - \mathbf{c}^c\|_2}{\|\mathbf{e}_i^k - \mathbf{c}^j\|_2} \right)^2}$
  end
  Compute the average of all predictions across $\tilde{\mathbf{x}_k^u}$
end
Collect augmented labeled data: $\tilde{\mathcal{X}} = (\mathbf{x}_b, \mathbf{y}_b); b \in \{1,\dots,N_{batch}\}$;
Collect augmented unlabeled data with their self predicted labels: $\tilde{\mathcal{X}}^u = (\mathbf{x}_b^u, \mathbf{y}_b^u); b \in \{1,\dots,N_{batch}\}$;
Concatenate $\tilde{\mathcal{W}} = (\tilde{\mathcal{X}}, \tilde{\mathcal{X}}^u)$;
Use MixUp (convex combinations) for pairs of labeled and new data $\mathcal{X}' = MixUp(\tilde{\mathcal{X}}, \tilde{\mathcal{W}})$ and pairs of unlabeled and new data $\mathcal{X}'_u = MixUp(\tilde{\mathcal{X}}^u, \tilde{\mathcal{W}})$;
Compute total loss using $\mathcal{X}'$ $\mathcal{X}'_u$;
Update network weights;

On the right- hand of the slide, we present a detailed version of the Margin Mix. On the left-hand side of the slide, we show a schematic of the method.

Mainly it consist of four steps:

1. prepare for self labelling. This means that we need centroids that are properly spaced. We update the centroids using labelled and unlabeled data. In the first iteration the contribution of label data dominates, while afterwards the labeled and unlabeled data become even
2. Augment data using both labeled and unlabeled data. While labelled data are few, there are many unlabeled data so we can densely sample the data space
3. Use self-prediction based on distance to centroids to associate labels to new data. Points in the middle are having little influence, while points close to centroids do impact the border
4. Update the network weight such that the embedding favor good clustering and accurate predictions. Good clustering means that as few points as possible are left in the middle.

Evaluation

- Standard benchmarks: **error** on CIFAR-10, CIFAR100, SVHN, STL

	CIFAR -10			CIFAR -100	SVHN		STL	
labels	250	1000	4000	10000	1000	4000	1000	5000
MixMatch	11.80	7.75	6.24	28.88	3.27	2.89	10.18	5.59
MarginMix	10.76	8.33	6.17	29.12	3.35	3.33	9.85	5.80

- Classes are relatively separable
- Performance comparable with prior art, although errors are slightly larger

Now let us discuss the results.
First, we evaluated our method on several standard benchmarks. In these cases, it is assumed that a subset from the dataset has labels and the remainder doesn't have them. We would like to point that these problems are relatively easy, fact indicated by the high performance when considering the fully supervised problem.
Looking at the results, one may see that MarginMix performance is comparable with prior art, although errors are slightly larger. The lack of influence of the points in the middle probably it cost us

Evaluation

- Face expression recognition: Accuracy on FER+ and RAF-DB
  - More difficult problem
  - SOTA is lower for fewer classes than on standard benchmarks

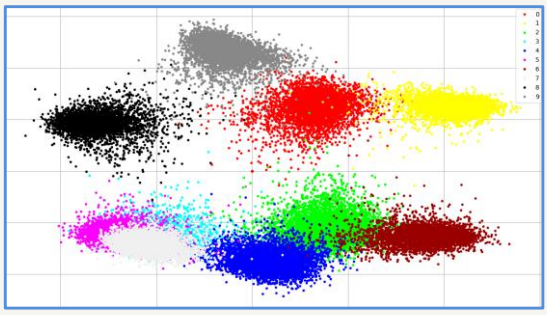
FER+				
labels	320	2000	4000	10000
MixMatch	45.6	58.3	70.9	71.2
MarginMix	50.7	60.8	75.2	81.2
RAF-DB				
labels	320	400	1000	4000
MixMatch	35.6	42.3	60.4	65.2
MarginMix	40.6	45.7	66.5	70.7

- Classes are less separable
- Performance better than prior art

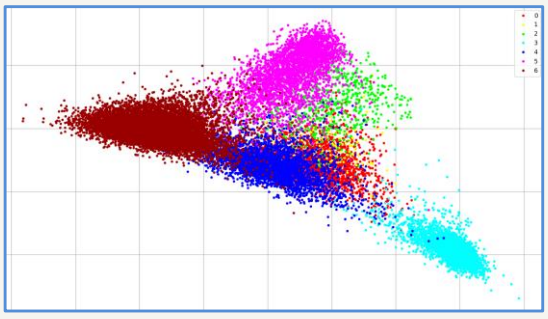
On the second part of the evaluation, we consider the problem of categorical face expressions recognition. Noting that on purely supervised learning the performance is lower, we strengthen our claim that this problem is harder.
In terms of semi-supervised learning, we applied the same procedure as in the case of standard benchmarks, which is to consider a nominated subset to have labels and the rest to be unlabeled.
In this case, our method performs noticeably better than prior art. The comparison is fair as long as we changed only the process of self-labelling and left everything else as from the previous solutions.

Harder vs Less Harder Problems

- Two dimensional embeddings



CIFAR -10

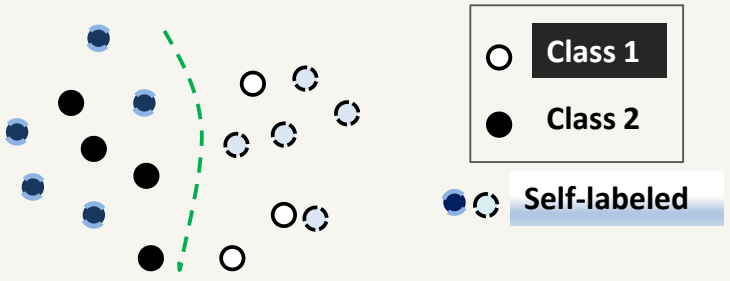


FER+

We also delved in the difficulty of the problems. This time we used statistical means
In these images, we considered a two dimensional, thus plottable, embedding. Classes corresponding to expressions are obviously less separable in FER+ database, than are the image categories in CIFAR-10. Thus we can strengthen our claim that face expression recognition is a harder problem

Intuition – Standard Approach

Easy problem



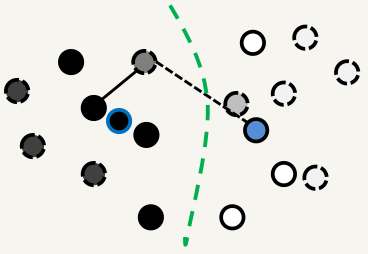
Semi-supervised learning with self labeling, first draws initial border, than using this border attaches pseudo-labels to unlabeled data and follows by slightly readjusting the border to place it equally distant to given points

In the following slides, we provide a graphical interpretation and intuition of what is happening in semi-supervised problems. First, we consider a easy problem, than a harder one. We compare our method with the standard MixMatch solutions.
We illustrate the behavior on a two class problems. Points drawn with solid colors are labelled, while ones with dashed lines and two colored are not.

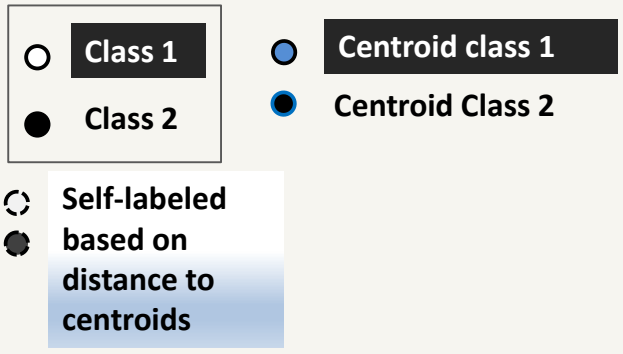
Semi-supervised learning with self labelling, first draws the initial borders. Then, using these borders attaches pseudo-labels to unlabeled data and follows by slightly readjusting the borders to place them equally distant to the given points.

Intuition - MarginMix

Easy problem



Semi-supervised border
Border is preserved

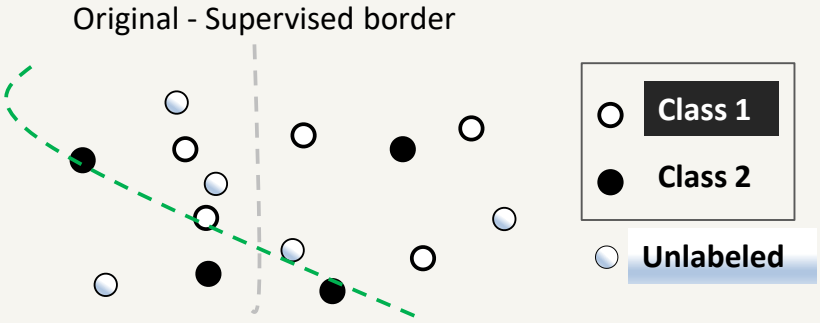


MarginMix solution first seeks centroids and self labels new points with respect to the distance to centroids. For an easy problem, the behavior is correct.

Our solution seeks first the centroids and self label new points with respect to the distance to centroids. The border results somehow given distance to centroids. For an easy problem, especially since there is a low-density area around the border, the behavior is correct.

Intuition – Standard Approach

Difficult problem

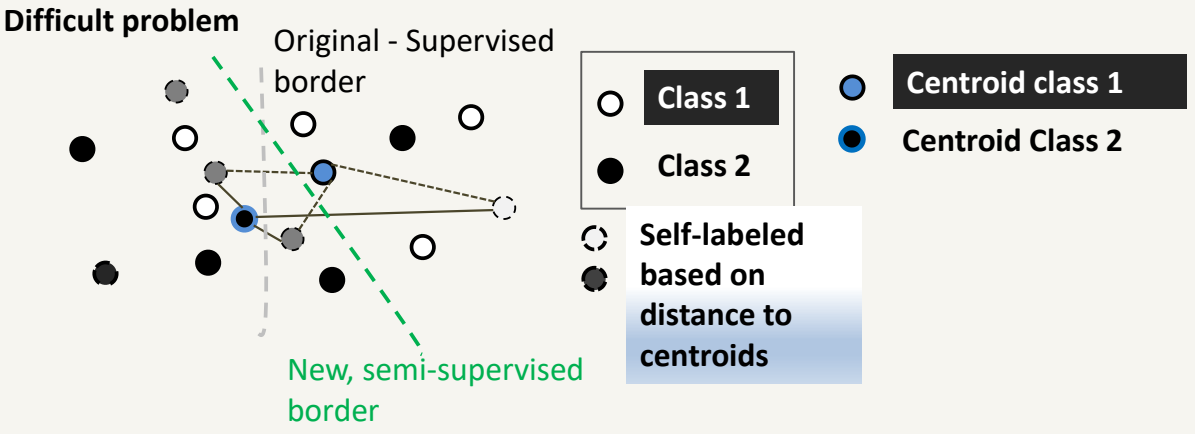


Border based on labelled data

For a harder problem, where points from different classes are mixed together, standard self-labelling may easily be degraded due to unstable initial borders.

For a harder problem, where points from different classes are mixed (because they are very similar), standard self-labelling, may easily be degraded due unstable initial borders. There is some arbitrary influence in establishing the position of the border due to difficult points.

Intuition - MarginMix



MarginMix, which is able to provide soft labeling, allows the new data (placed between clusters and respectively centroids), to have little influence on the new border. Thus the found border is no longer unstable and subject to specific point choice.

Our proposal, which is able to provide soft labelling, allows that new data, placed between clusters and respectively centroids, to have little influence on the new borders. Therefore the found borders are no longer unstable and subject to specific point choice. Again the idea is that points in the middle do almost nothing.

Conclusions

- On easy problems, there are few points between classes that if chosen in the border refinement process can confuse the learner
- On harder problems, there are many points capable of creating confusion. These points should be down-weighted. One way to achieve this is to use soft labeling
- We use the distance cluster centroids to soft-labels synthetical instances
- To have reliable clustering, we ask the network to create discriminative embeddings
- The resulting Margin-Mix Algorithm shows improved performance on harder problems such as face expression recognition

On easy problems, there are few points between classes that if chosen in the border refinement process can confuse the learner

On harder problems, there are many points capable of creating confusion. These points should be down-weighted. One way to achieve this is to use soft labeling

We use the distance cluster centroids to soft-labels synthetical instances

To have reliable clustering, we ask the network to create discriminative embeddings

The resulting Margin-Mix Algorithm shows improved performance on harder problems such as face expression recognition

THANK YOU FOR

YOUR ATTENTION!