CITIC Research Center, Universidade da Coruña

Can data placement be effective for Neural Networks classification tasks? Introducing the Orthogonal Loss

B. Cancela, V. Bolón-Canedo and A. Alonso-Betanzos

Introduction

Classic classification losses aim to separate the deep features between classes as much as possible:

- It does not matter where the deep features are placed, as long as they are clearly separated.

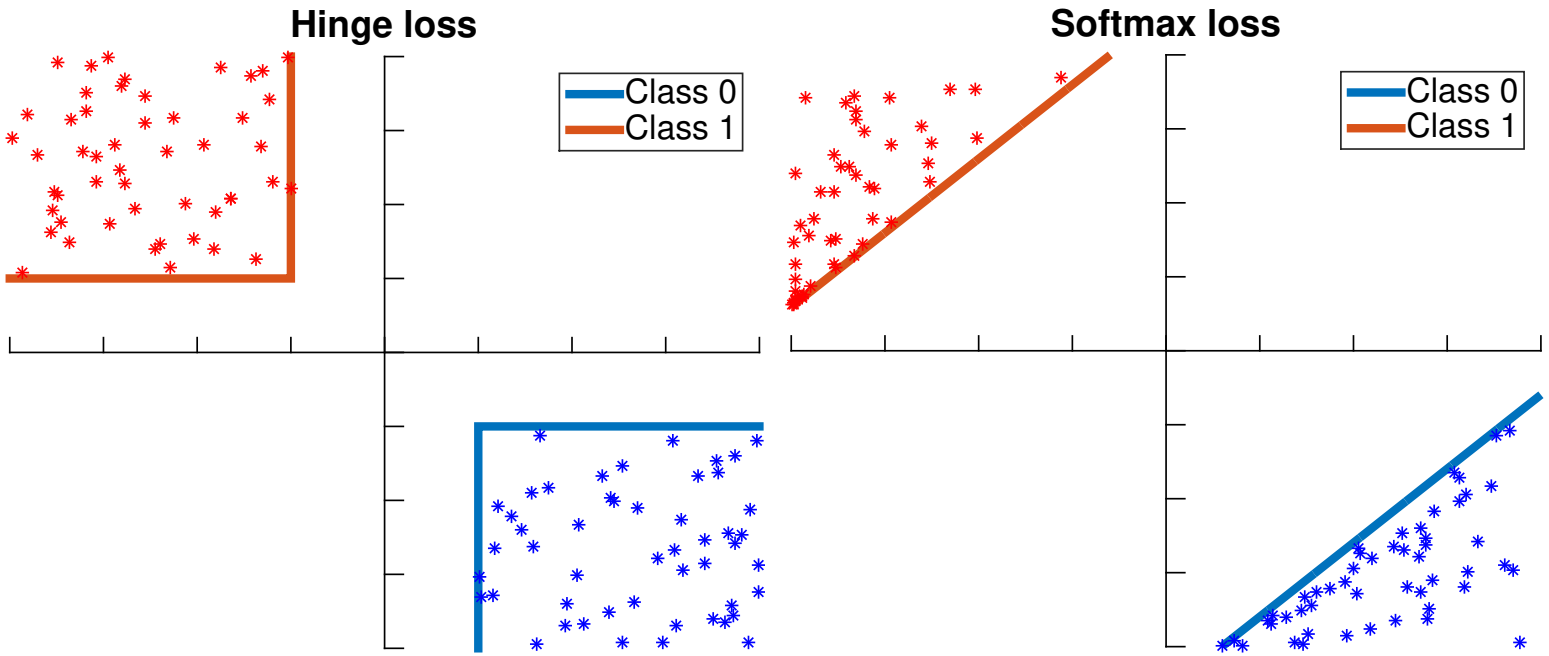
Furthermore, confidence scores do not provide any information about how confident is in the decision.

- For instance, an image is classified as a car with a 90% confidence, but it is a plane, which is not part of the training data.

Introduction

Imagine we have two features, two classes and a linear classifier which is the identity matrix.

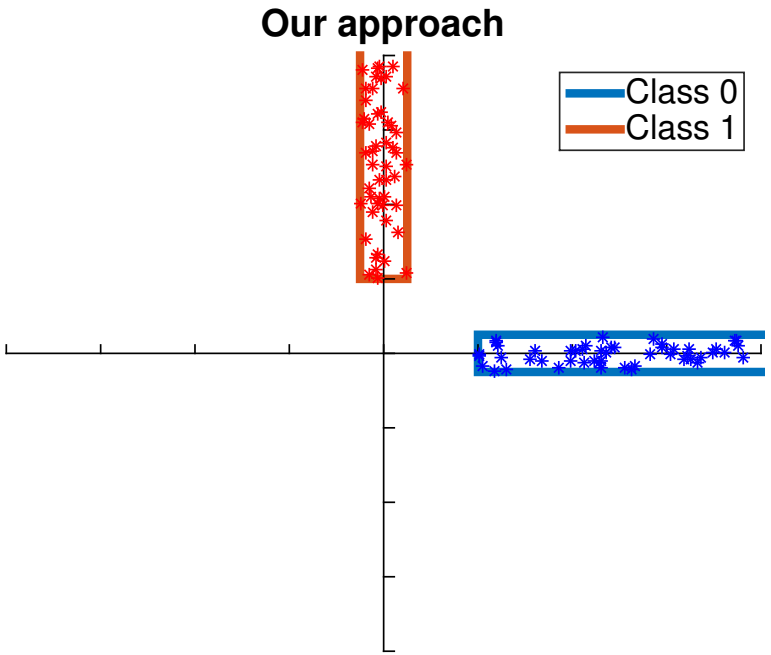
- Where the training data is placed for a loss value close to zero?



Research goal

Having an orthogonal classifier, is it possible to place the data over the classifier's autovectors, one per each class?

- Positive values for the direction of the correct class and zeros for the rest.



Research goal

Confidence scores do not provide any information about how confident is in the decision.

- For instance, an image is classified as a car with a 90% confidence, but it is a plane, which is not part of the training data.
- SOLUTION: A variant to the softmax function, aiming to provide a more accurate label confidence.

Requisites

Requisite 1: It should be a Neural Network with one hidden layer at list.

Requisite 2: The last layer will be fixed during the training procedure.

Requisite 3: The last layer will be initialized as an orthogonal matrix.

Exponential Orthogonal Loss

Forces the correct class to have the highest positive value.

- $\tilde{y}^{(i)}$: model prediction
- $y^{(i)}$: input label (one-hot encoding)

$$\mathcal{L}^{EOL}(\tilde{y}^{(i)}, y^{(i)}) = \sum \overbrace{y^{(i)}\ e^{R-\tilde{y}^{(i)}}}^{\mathcal{L}^{EOL}_{pos}} + \lambda \underbrace{(1-y^{(i)})\ (\tilde{y}^{(i)})^2}_{\mathcal{L}^{EOL}_{neg}}, \quad R > 0,$$

Trench Orthogonal Loss

Forces the correct class to have the fixed positive value R.

- $\tilde{y}^{(i)}$: model prediction
- $y^{(i)}$: input label (one-hot encoding)

$$\mathcal{L}^{TOL}(\tilde{y}^{(i)}, y^{(i)}) = \sum \overbrace{y^{(i)}\ (R - \tilde{y}^{(i)})^2}^{\mathcal{L}^{TOL}_{pos}} + \lambda\ \underbrace{(1 - y^{(i)})\ (\tilde{y}^{(i)})^2}_{\mathcal{L}^{TOL}_{neg}}, \quad R > 0,$$

Softmax variation

- $\tilde{y}^{(i)}$: model prediction
- Checks how good the projection is in each class.
- $\sum p \leq 1$

$$p(\tilde{y}^{(i)}) = \frac{\max(0, \tilde{y}^{(i)})}{\max(P, \sum \max(0, \tilde{y}^{(i)}))}, \quad P > 0.$$

Experimental Results

- Tested in 5 different datasets using two networks.
- The results are better (or comparable) than using the cross-entropy loss.

VGG	CE (fixed classifier)	CE	EOL	TOL
MNIST	99.57 ± 0.01	99.57 ± 0.04	99.57 ± 0.06	$\mathbf{99.64 \pm 0.01}$
Fashion-MNIST	94.26 ± 0.13	94.27 ± 0.07	$\mathbf{94.35 \pm 0.05}$	$\mathbf{94.45 \pm 0.14}$
CIFAR-10	91.70 ± 0.13	91.55 ± 0.17	$\mathbf{92.03 \pm 0.13}$	$\mathbf{91.82 \pm 0.12}$
CIFAR-100	66.38 ± 0.67	63.13 ± 0.80	$\mathbf{69.95 \pm 0.19}$	$\mathbf{69.90 \pm 0.34}$
STL-10	79.78 ± 0.56	78.27 ± 0.23	$\mathbf{81.39 \pm 1.08}$	$\mathbf{80.44 \pm 0.79}$

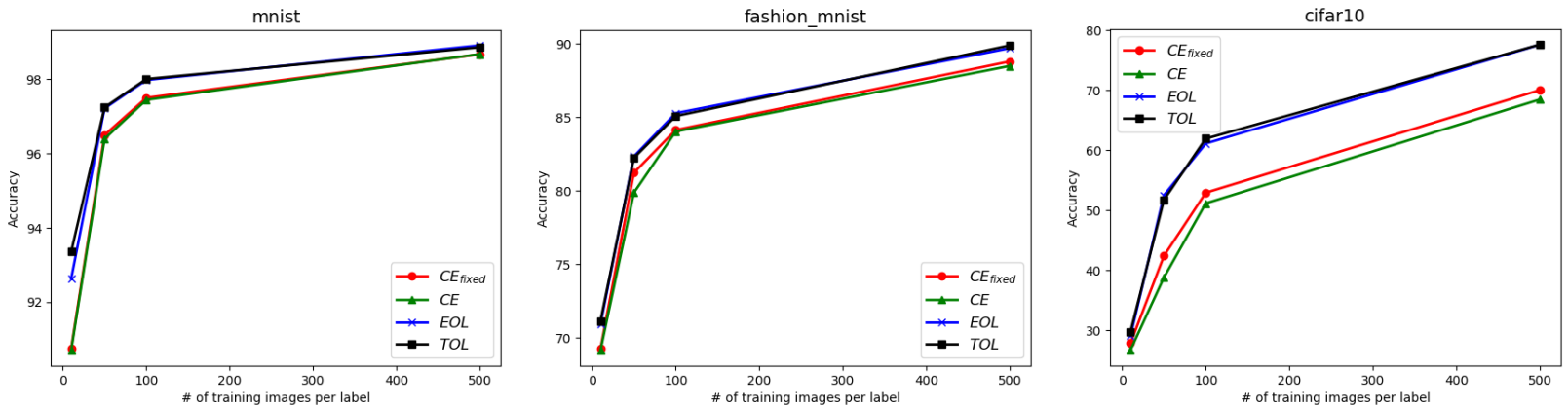
Experimental Results

- Tested in 5 different datasets using two networks.
- The results are better (or comparable) than using the cross-entropy loss.

WRN-16-4	CE (fixed classifier)	CE	EOL	TOL
MNIST	99.67 ± 0.02	99.64 ± 0.02	99.64 ± 0.02	99.66 ± 0.01
Fashion-MNIST	95.31 ± 0.10	95.26 ± 0.14	$\mathbf{95.47 \pm 0.13}$	95.21 ± 0.18
CIFAR-10	94.59 ± 0.13	94.59 ± 0.17	$\mathbf{94.88 \pm 0.13}$	94.75 ± 0.10
CIFAR-100	74.95 ± 0.09	74.70 ± 0.24	74.92 ± 0.09	$\mathbf{75.39 \pm 0.18}$
STL-10	86.19 ± 0.35	85.70 ± 0.21	$\mathbf{86.65 \pm 0.34}$	$\mathbf{86.71 \pm 0.28}$

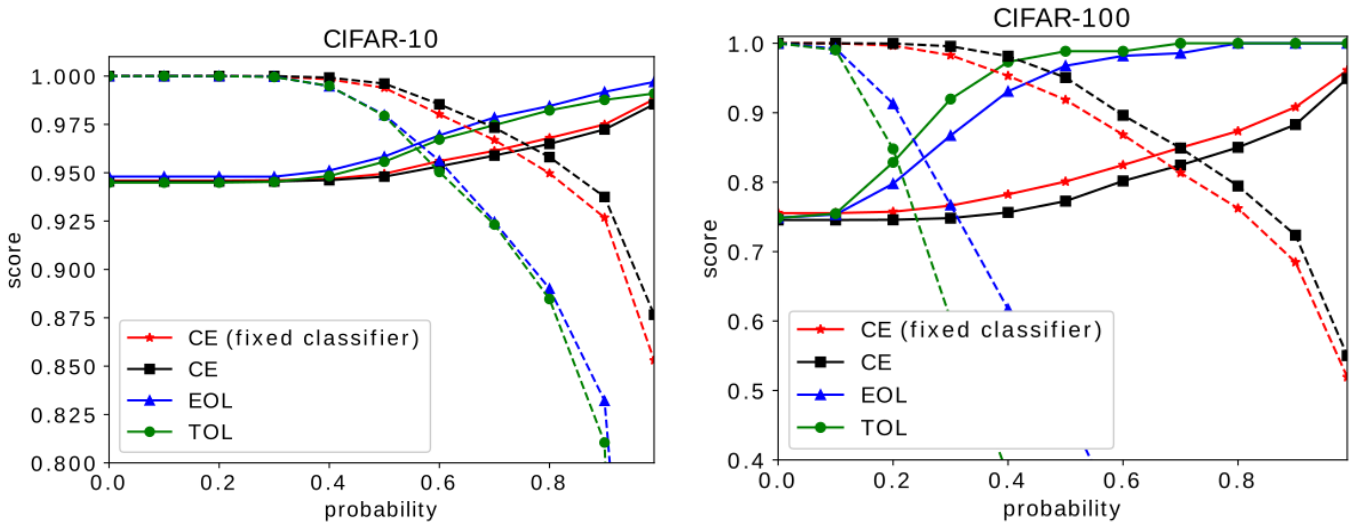
Using a reduced training data version

Both EOL and TOL obtain a better accuracy when using a subset of the training data.



Effect of the softmax variant

The number of errors is reduced as our variant output increases its confidence.



Conclusions

- Deep features can be effectively placed without losing performance (sometimes even better).
- It obtains better results when the number of training samples is low.
- The softmax variant provides a better degree of certainty in the probability outputs.
- Deep features placed in orthogonal projections have interesting properties.

CITIC Research Center, Universidade da Coruña

Can data placement be effective for Neural Networks classification tasks? Introducing the Orthogonal Loss

B. Cancela, V. Bolón-Canedo and A. Alonso-Betanzos