VAADER
IETR research team

"AI FOR IMAGE" READING GROUP

Travel in the Deep Learning

MOUATH AOUAYEB 22/04/20201

This is me

Mouath Aouayeb, 25 ans

Ph.D. : Deep Learning for MiEs Analysis

INSA-VAADER
Kidiyo Kpalma
Waasim Hamidouche

CS-FAST
Renaud Seguier
Catherine Soladie

SUP'COM : Telecom Engineer, MIT
ENIT: Mastère en TICV
Univ. Paris Descartes:
Mastère en Math-Info

Deep Learning
Computer Vision
Traitement de Signal
Réseaux
Systèmes embarqués

Stage PFE: Détection des MiEs Faciales
18/02/2019

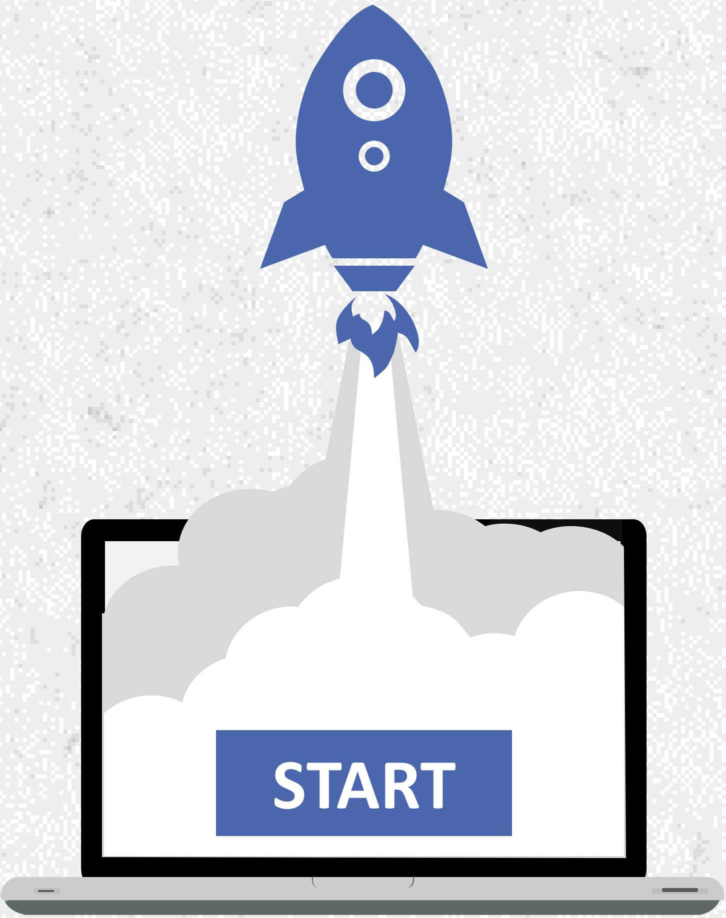
center of interest : new Tech, Sociology, Philosophy, History
Hobbies: Chess game, Hiking

What's on the menu Today ?

VAADER
IETR research team

Travel & DL

Analogy
Common points



DL Techniques

Losses, Layers,
Optimizers, ...

Vision Transformer

Smart CAR

MY LITTLE TRAVEL

Transport

Destination

DATA

AI

Deep Learning

LSTM

Fuel

Hitchhicking

Object Detection

Travel

MAP

Attention Model

CNN

Deep Learning

Travel & DL

VAADER IETR research team

Rennes

Paris
Nantes
Tunis
...

Car
Train
Smart Car
Autonomous Car

Fuel

Map

GPS
Voice recognition
Cameras , ...

Random

Classification
Object Detection
Tracking
...

Convolutional Neural Network (CNN)
RNN (LSTM)
Vision Transformer (ViT)
Reinforcment Learning

DATA

Loss Function

Attention Model

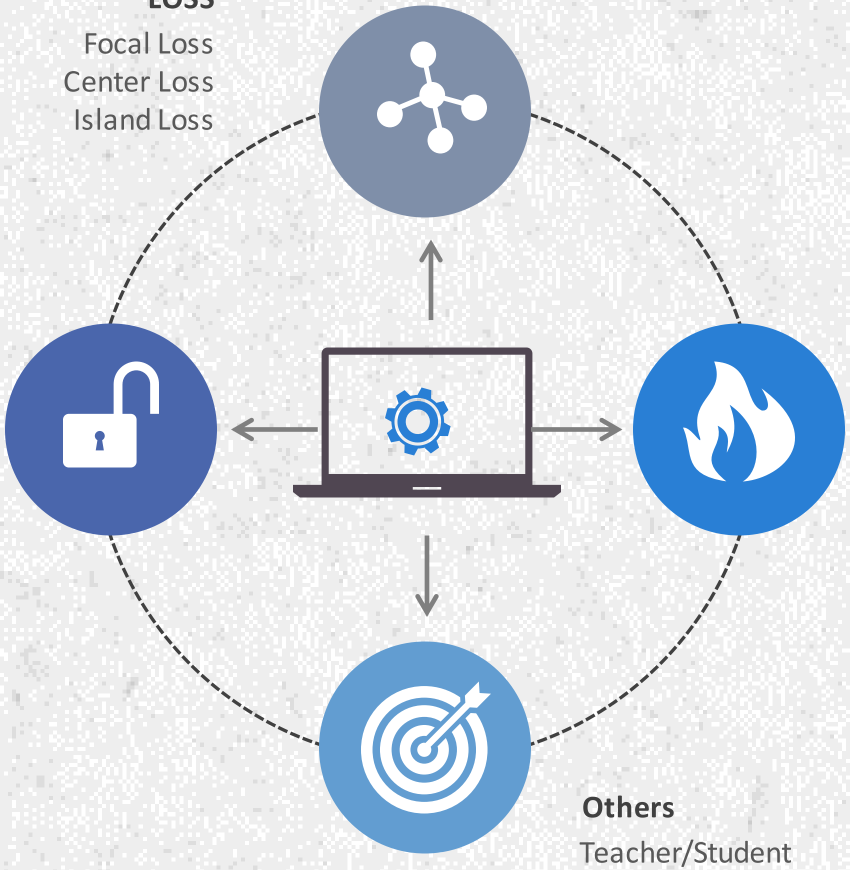
Travel & DL

Traffic signs	Optimizers
Vitesse	Learning Rate
Police	Scheduler
Hitchhiking	Fine-tuning
"Lost"	Overfitting
"Broken down car"	Underfitting
Wheels	Activation Function

DL Techniques

LOSS

Focal Loss
Center Loss
Island Loss



Layers

Squeeze and Excitation
Weight Normalisation
Global Normalisation

Optimizer/

AsamW/SGDW

Others

Teacher/Student
Distillation

Loss Function

[Categorical] Cross Entropy

$$\mathrm{CE}(p_t) = -\log(p_t)$$

[Categorical] Balanced Cross Entropy

$$\mathrm{CE}(p_t) = -\alpha_t \log(p_t).$$

- Imbalanced Quantity
- More focus on less frequent examples

Focal Loss [1]

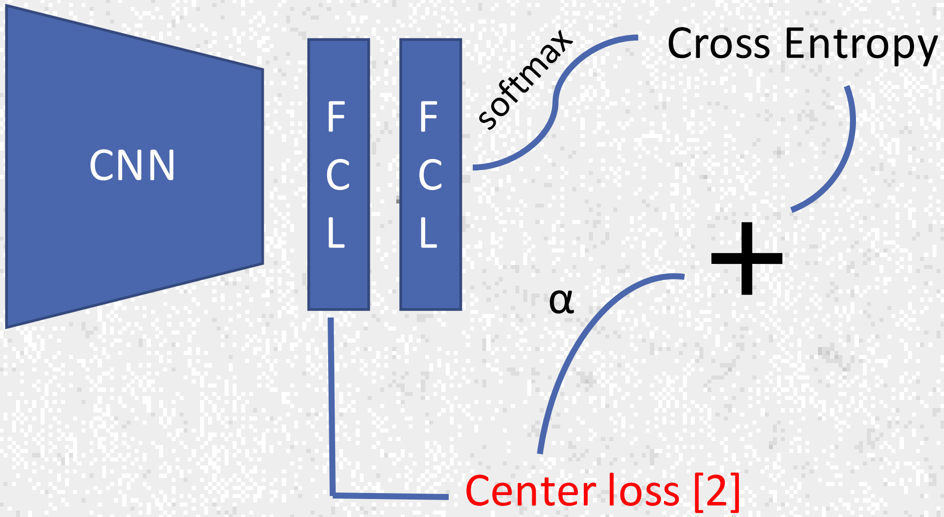
$$\mathrm{FL}(p_t) = -(1 - p_t)^\gamma \log(p_t).$$

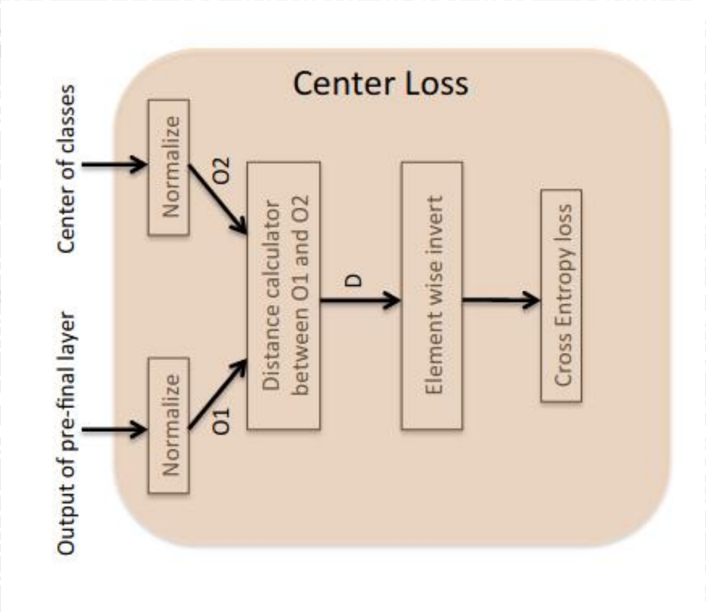
- Imbalanced Quality
- More focus on hard, misclassified examples

$$\mathrm{FL}(p_t) = -\alpha_t (1 - p_t)^\gamma \log(p_t).$$

[1] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2017 IEEE International Conference on Computer Vision (ICCV), 2017.
Code: https://github.com/facebookresearch/Detectron.

Loss Function





- Minimizes the embedding space distance of each point in a class to its center
- Bring together data-points belonging to the same class.

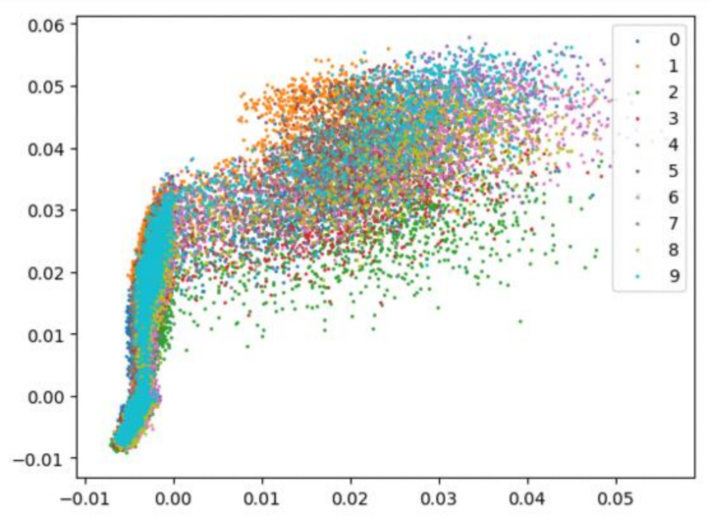
[2] Wen Y., Zhang K., Li Z., Qiao Y. (2016) A Discriminative Feature Learning Approach for Deep Face Recognition. In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016.

Code: https://github.com/KaiyangZhou/pytorch-center-loss

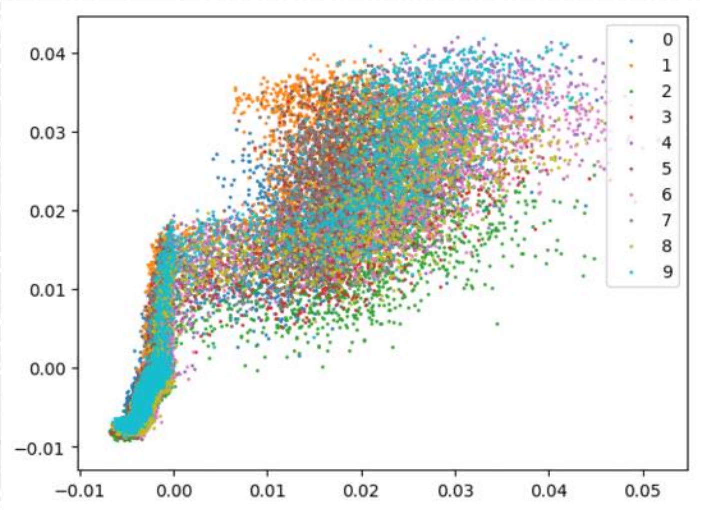
Loss Function

Visulaisation of the Feature Learning Process (t-SNE)

Without Center Loss



With Center Loss



[2] Wen Y., Zhang K., Li Z., Qiao Y. (2016) A Discriminative Feature Learning Approach for Deep Face Recognition. In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016.

Code: https://github.com/KaiyangZhou/pytorch-center-loss

Loss Function

Center Loss [2]

$$\mathcal{L}_C = \frac{1}{2}\sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{c}_{y_i}\|^2$$

Island Loss [3]

$$\mathcal{L}_{IL} = \mathcal{L}_C + \lambda_1 \sum_{\mathbf{c}_j \in \mathcal{N}} \sum_{\substack{\mathbf{c}_k \in \mathcal{N} \\ \mathbf{c}_k \neq \mathbf{c}_j}} \left(\frac{\mathbf{c}_k \cdot \mathbf{c}_j}{\|\mathbf{c}_k\|_2 \|\mathbf{c}_j\|_2} + 1 \right)$$

RAN Loss [6]

GCCS Loss [5]

Sparse Center Loss [4]

[2] Wen Y., Zhang K., Li Z., Qiao Y. A Discriminative Feature Learning Approach for Deep Face Recognition. In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision ECCV 2016.

[3] J. Cai, Z. Meng, A.-S. Khan, Z. Li, J. O'Reilly, and Y. Tong, "Island loss for learning discrimi-native features in facial expression recognition," FG 2018

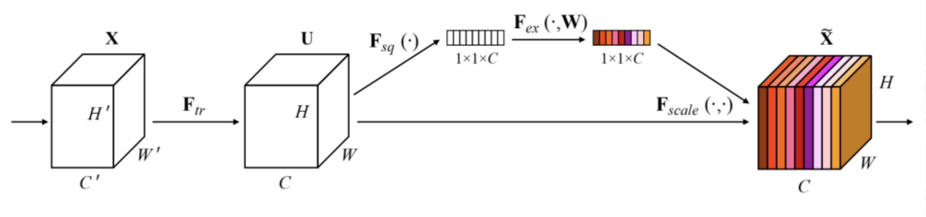
[4] A. H. Farzaneh and X. Qi, "Facial expression recognition in the wild via deep attentive center loss," inProceedings of the IEEE/CVF WACV, January 2021

[5] A. Ali, A. Migliorati, T. Bianchi, and E. Magli, "Beyond cross-entropy: learning highly separable feature distributions for robust and accurate classification,"ArXiv 2020

[6] K. Wang, X. Peng, J. Yang, D. Meng, and Y. Qiao, "Region attention networks for pose and 217occlusion robust facial expression recognition,"IEEE Transactions on Image Processing

Layers

Squeeze and Excitation [7]



- Channel Attention
- Fusion
- Optimize

=> Attention block & ~ small & rapid convergence

"The SE module can improve performance on both CNN and ViT,which means applying attention to channels benefits both CNN and ViT models." [8]

[7] J. Hu, L. Shen and G. Sun, "Squeeze-and-Excitation Networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018
[8] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, F. E. H. Tay, J. Feng, and S. Yan, "Tokens-to-token231vit: Training vision transformers from scratch on imagenet,"ArXiv 2021

Layers

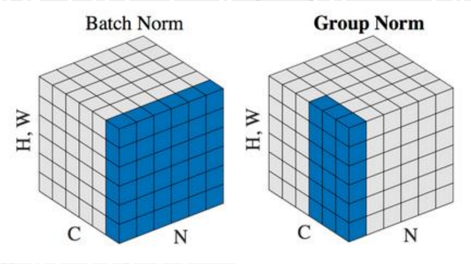
Weight Normalisation [9]

$$y = \phi(\mathbf{w} \cdot \mathbf{x} + b),$$

$$\mathbf{w} = \frac{g}{||\mathbf{v}||}\mathbf{v}$$

- Normalize weights of layer
- RNN, Reinforcement NN, GAN
- Speed-up convergence of Gradient Descent
- Improve the conditioning of the optimisation problem

Group Normalisation [10]



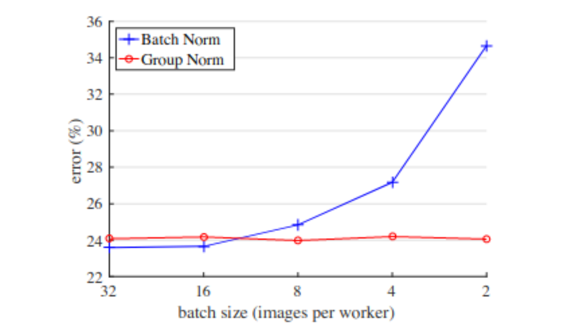
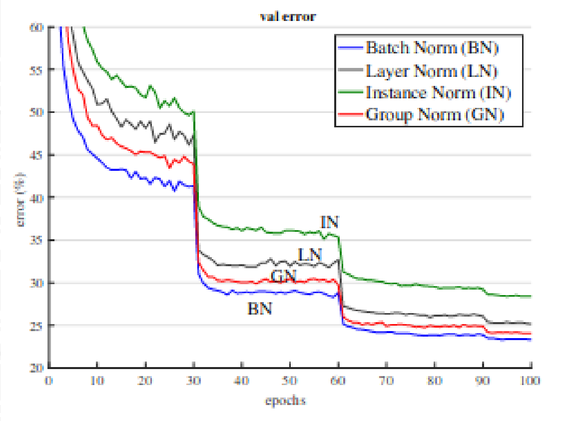


Figure 1. **ImageNet classification error *vs.* batch sizes.** This is a ResNet-50 model trained in the ImageNet training set using 8 workers (GPUs), evaluated in the validation set.



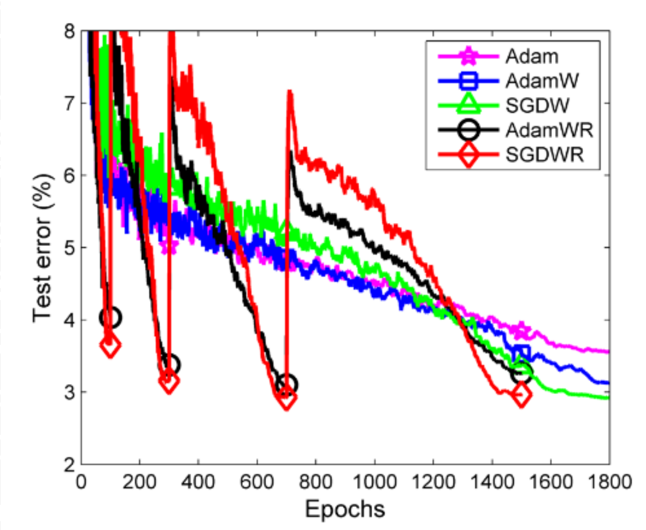
[9] Tim Salimans and Diederik P. Kingma. 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. (*NIPS'16*) 2016
Code: tensorflowAddons ou https://github.com/openai/weightnorm
[10] Wu, Y., He, K. Group Normalization. *Int J Comput Vis* **128,** 742–755 (2020).
https://openaccess.thecvf.com/content_ECCV_2018/papers/Yuxin_Wu_Group_Normalization_ECCV_2018_paper.pdf
Code: https://github.com/facebookresearch/Detectron/blob/master/projects/GN

Optimizer

AdamW/SGDW [11]

$$\boldsymbol{\theta}_{t+1} = (1-\lambda)\boldsymbol{\theta}_t - \alpha \nabla f_t(\boldsymbol{\theta}_t),$$

- Decouple the rate λ and the learning rate α.
- Decay the weights simultaneously with the update of θt .



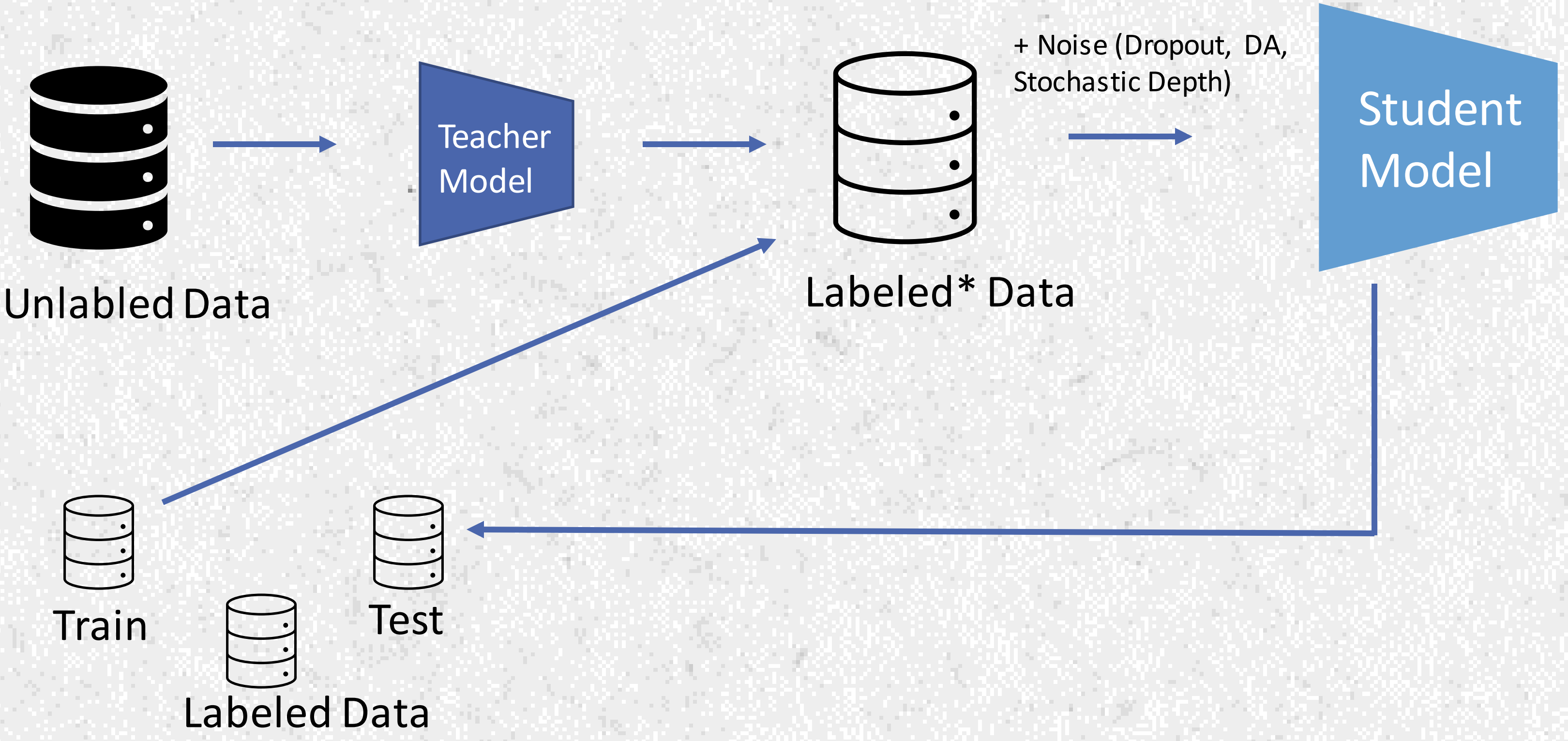
SGDW:

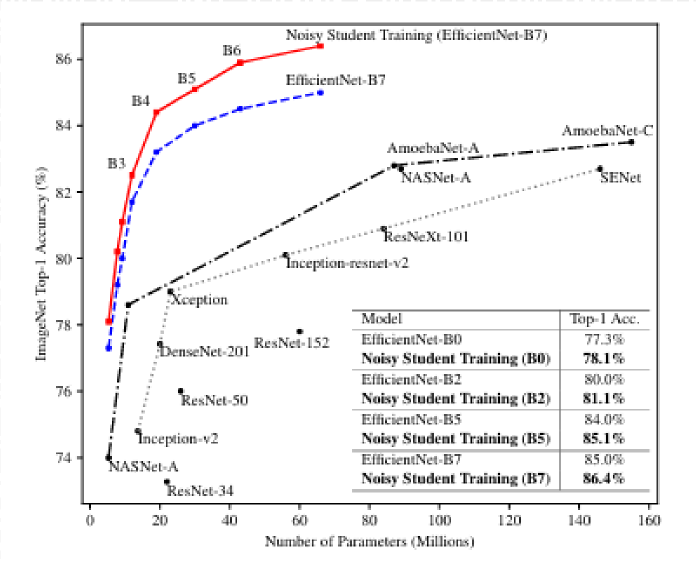
repeat
$t \leftarrow t+1$
$\nabla f_t(\boldsymbol{\theta}_{t-1}) \leftarrow \text{SelectBatch}(\boldsymbol{\theta}_{t-1})$
$\boldsymbol{g}_t \leftarrow \nabla f_t(\boldsymbol{\theta}_{t-1}) \; +\lambda\boldsymbol{\theta}_{t-1}$
$\eta_t \leftarrow \text{SetScheduleMultiplier}(t)$
$\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + \eta_t \alpha \boldsymbol{g}_t$
$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \boldsymbol{m}_t \; -\eta_t\lambda\boldsymbol{\theta}_{t-1}$

[11] *Loshchilov, I. and F. Hutter. "Decoupled Weight Decay Regularization." ICLR (2019).*
Code: https://github.com/loshchil/AdamW-and-SGDW}

Others

Teacher Model/ Noisy Student [12]



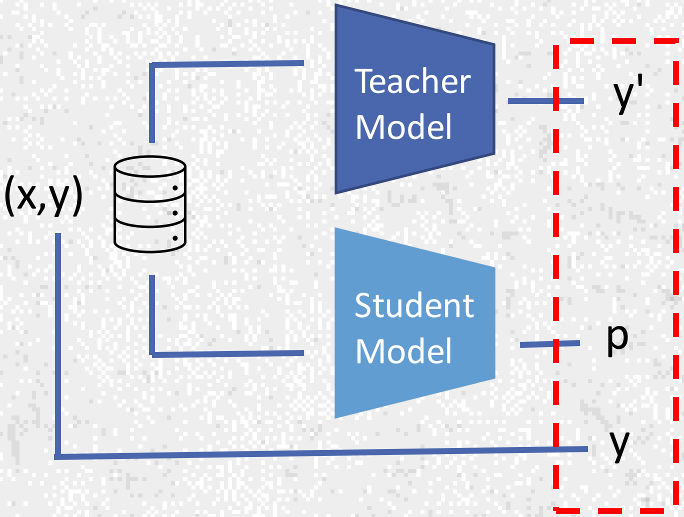


Model	Top-1 Acc.
EfficientNet-B0	77.3%
Noisy Student Training (B0)	**78.1%**
EfficientNet-B2	80.0%
Noisy Student Training (B2)	**81.1%**
EfficientNet-B5	84.0%
Noisy Student Training (B5)	**85.1%**
EfficientNet-B7	85.0%
Noisy Student Training (B7)	**86.4%**

[12] Q. Xie, M. -T. Luong, E. Hovy and Q. V. Le, "Self-Training With Noisy Student Improves ImageNet Classification," *CVPR*, 2020
Code: https://github.com/google-research/noisystudent

Others

Distillations:



Soft distillation [13] [14] : "minimizes the Kullback-Leibler divergence between the softmax of the teacher and the softmax of the student model."

$$\mathcal{L}_{\mathrm{global}} = (1-\lambda)\mathcal{L}_{\mathrm{CE}}(\psi(Z_{\mathrm{s}}), y) + \lambda\tau^2 \mathrm{KL}(\psi(Z_{\mathrm{s}}/\tau), \psi(Z_{\mathrm{t}}/\tau)).$$

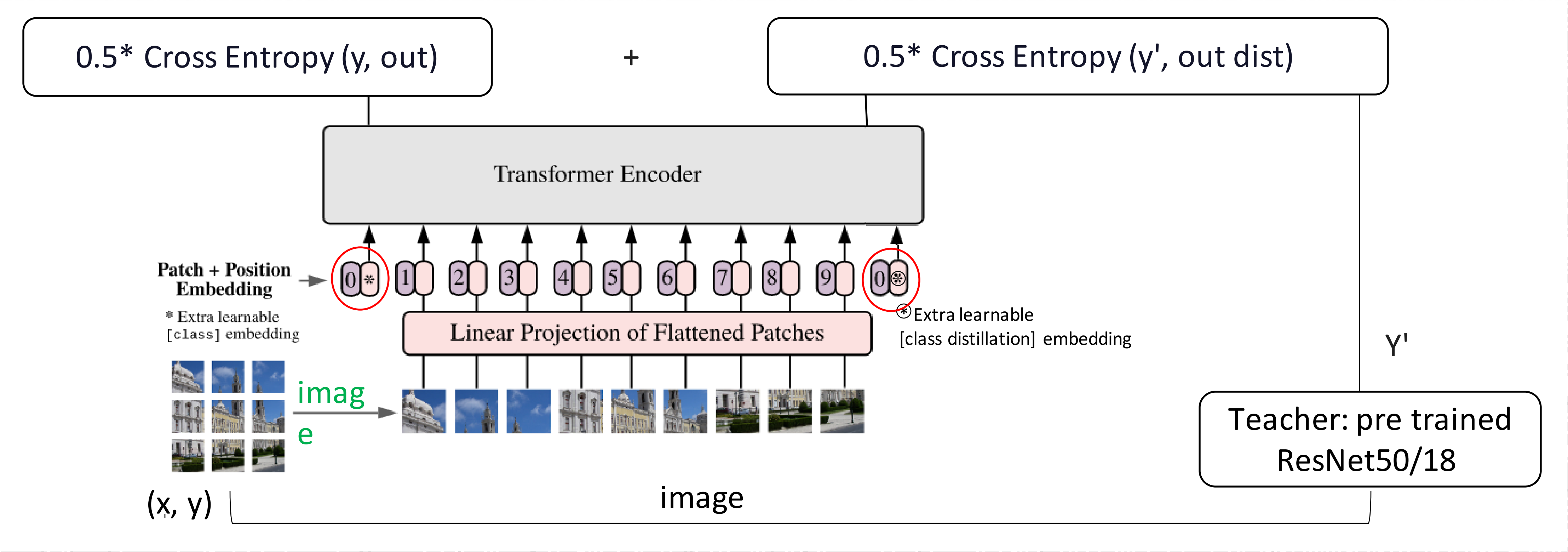
Hard Label distillation : "minimizes the Cross-Entropy between the softmax of the teacher and the softmax of the student model."

$$\mathcal{L}_{\mathrm{global}}^{\mathrm{hardDistill}} = \frac{1}{2}\mathcal{L}_{\mathrm{CE}}(\psi(Z_s), y) + \frac{1}{2}\mathcal{L}_{\mathrm{CE}}(\psi(Z_s), y_{\mathrm{t}})$$

[13] Geoffrey E. Hinton, Oriol Vinyals, and J. Dean. Distilling the knowledge in a neuralnetwork.arXiv preprint arXiv:1503.02531, 2015.
[14] Longhui Wei, An Xiao, Lingxi Xie, Xin Chen, Xiaopeng Zhang, and Qi Tian. Cir-cumventing outliers of autoaugment with knowledge distillation.ECCV, 2020.
Code: https://github.com/yoshitomo-matsubara/torchdistill

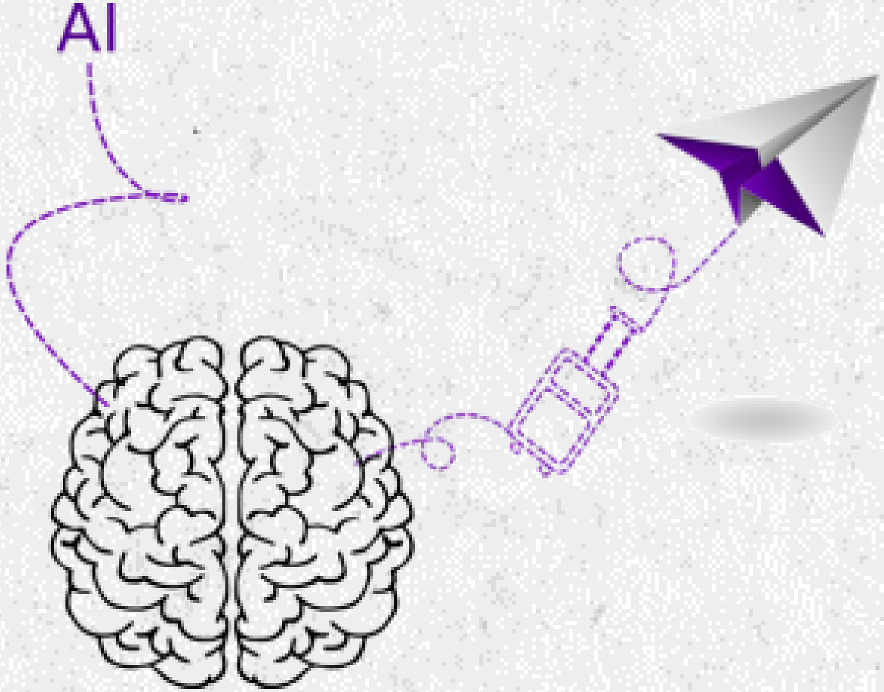
Others

Distillations: Distillation through attention, DeiT [15]



[15] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablay-rolles, and H. J´egou. Training data-efficient image trans-formers & distillation through attention.arXiv, 2020
code: https://github.com/facebookresearch/deit

THANK YOU



TRAVEL IN THE DEEP LEARNING

MOUATH