Regularization, Data Augmentation and Self-Supervised Learning

Efficient Deep Learning - Session 4

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

2023

2023-04-11

Regularization, DA and SSL

Regularization, Data Augmentation and Self-Supervised Learning

Efficient Deep Learning - Session 4

2023

Course organisation

Sessions

1. Introduction/Refresher on Deep Learning
2. Regularization, Data Augmentation,
3. Quantization,
4. Pruning,
5. Factorization,
6. Distillation and Self-Supervised Learning,
7. Embedded Software and Hardware for DL,
8. Final session.

└─Course organisation

Course organisation
Sessions
1 Introduction/Refresher on Deep Learning
2 Regularization, Data Augmentation,
3 Quantization,
4 Pruning,
5 Factorization,
6 Distillation and Self-Supervised Learning,
7 Embedded Software and Hardware for DL,
8 Final session.

Course organisation

Sessions

1. Introduction/Refresher on Deep Learning
2. Regularization, Data Augmentation,
3. Quantization,
4. Pruning,
5. Factorization,
6. Distillation and Self-Supervised Learning,
7. Embedded Software and Hardware for DL,
8. Final session.

Course organisation
Sessions
1 Introduction/Refresher on Deep Learning
2 Regularization, Data Augmentation,
3 Quantization,
4 Pruning,
5 Factorization,
6 Distillation and Self-Supervised Learning,
7 Embedded Software and Hardware for DL,
8 Final session.

Why this session ?

Regularization

Constrain the training for faster convergence and better generalization.

Data Augmentation (DA)

Help generalization by sampling training examples from a larger distribution using randomized transforms.

Self-supervised Learning (SSL)

Exploit DA and regularization tricks for learning representations, without labels

Significance

- In some (most?) cases, DA regularizes training and is needed.
- Large networks can't be trained without regularization.

Regularization prevents overfitting in neural networks, thus improve the predictions on data outside the training set. Regularization is any component of the model, training process or prediction procedure which is included to account for limitations of the training data (Early Stopping can also be seen as a form of regularization)

Why this session ?

Regularization

Constrain the training for faster convergence and better generalization.

Data Augmentation (DA)

Help generalization by sampling training examples from a larger distribution using randomized transforms.

Self-supervised Learning (SSL)

Exploit DA and regularization tricks for learning representations, without labels

Significance

- In some (most?) cases, DA regularizes training and is needed.
- Large networks can't be trained without regularization.

Why this session ?
Regularization
Constrain the training for faster convergence and better generalization.
Data Augmentation (DA)
Help generalization by sampling training examples from a larger distribution using randomized transforms.
Self-supervised Learning (SSL)
Exploit DA and regularization tricks for learning representations, without labels
Significance
In some (most?) cases, DA regularizes training and is needed.
Large networks can't be trained without regularization.

The best way to make a machine learning model generalize better is to train it on more data. Of course, in practice, the amount of data we have is limited. One way to get around this problem is to create fake data and add it to the training set. DA is a technique that increases the training set by creating new data points based on the original data during training

Why this session ?

Regularization

Constrain the training for faster convergence and better generalization.

Data Augmentation (DA)

Help generalization by sampling training examples from a larger distribution using randomized transforms.

Self-supervised Learning (SSL)

Exploit DA and regularization tricks for learning representations, without labels

Significance

- In some (most?) cases, DA regularizes training and is needed.
- Large networks can't be trained without regularization.

Why this session ?
Regularization
Constrain the training for faster convergence and better generalization.
Data Augmentation (DA)
Help generalization by sampling training examples from a larger distribution using randomized transforms.
Self-supervised Learning (SSL)
Exploit DA and regularization tricks for learning representations, without labels
Significance
In some (most?) cases, DA regularizes training and is needed.
Large networks can't be trained without regularization.

SSL in deep learning is a technique for learning data representation without the use of label for transfer learning or fine-tuning

Why this session ?

Regularization

Constrain the training for faster convergence and better generalization.

Data Augmentation (DA)

Help generalization by sampling training examples from a larger distribution using randomized transforms.

Self-supervised Learning (SSL)

Exploit DA and regularization tricks for learning representations, without labels

Significance

- In some (most?) cases, DA regularizes training and is needed.
- Large networks can't be trained without regularization.

Why this session ?
Regularization
Constrain the training for faster convergence and better generalization.
Data Augmentation (DA)
Help generalization by sampling training examples from a larger distribution using randomized transforms.
Self-supervised Learning (SSL)
Exploit DA and regularization tricks for learning representations, without labels
Significance
In some (most?) cases, DA regularizes training and is needed.
Large networks can't be trained without regularization.

Weight Decay

An old idea (Krogh and Herz 1991): ℓ_2 penatly term is added to the loss, limits the growth of model weights.

Has been shown to increase generalization and suppresses irrelevant model weights.
Ressources :

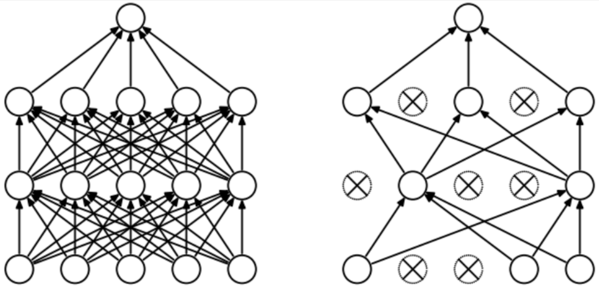
- https://proceedings.neurips.cc/paper/1991/file/8eefcfdf5990e441f0fb6f3fad709e21-Paper.pdf
- https://ja.d2l.ai/chapter_deep-learning-basics/weight-decay.html
- Readily available in pytorch (optimizer options)

Regularization
Weight Decay
An old idea (Krogh and Herz 1991): ℓ_2 penatly term is added to the loss, limits the growth of model weights.
Has been shown to increase generalization and suppresses irrelevant model weights.
Ressources :
https://proceedings.neurips.cc/paper/1991/file/8eefcfdf5990e441f0fb6f3fad709e21-Paper.pdf
https://ja.d2l.ai/chapter_deep-learning-basics/weight-decay.html
Readily available in pytorch (optimizer options)

Weight decay involves adding a term to the objective function that is proportional to the sum of the squares of the weights

Dropout

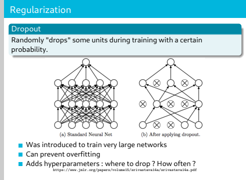
Randomly "drops" some units during training with a certain probability.



(a) Standard Neural Net (b) After applying dropout.

- Was introduced to train very large networks
- Can prevent overfitting
- Adds hyperparameters : where to drop ? How often ?

https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf



Dropout is a technique used only for training so that neurons will not learn redundant information of data, aswell as not relying on some specific features as they might be randomly dropped out. Applying dropout to a neural network amounts to sampling a “thinned” network from it. A neural net with n units, can be seen as a collection of 2^n possible thinned neural networks. So training a neural network with dropout can be seen as training a collection of 2^n thinned networks with extensive weight sharing, where each thinned network gets trained very rarely, if at all. At test time, the idea is to use a single neural net without dropout. The weights of this network are scaled-down versions of the trained weights. If a unit is retained with probability p during training, the outgoing weights of that unit are multiplied by p

Regularization

Batch Normalization (Ioffe & Szegedy, 2015)

Normalize feature distributions to the standard distribution by learning batch statistics.

- Consider a batch X
- Calculate $m = E(X)$ and $\sigma = Var(X)$
- Compute $\hat{X} = \frac{X - m}{\sigma} * \gamma + \beta$
- m and σ are continuously updated across batches using running statistics, and γ and β are learnable parameters (by default set to 1 and 0, respectively)

Notes

- Has been shown to accelerate training, increase generalization
- Can remove the need for DropOut
- Should be included by default after convolutions

http://proceedings.mlr.press/v37/ioffe15.pdf

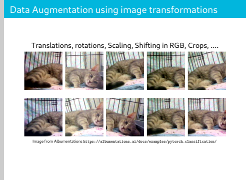
Regularization
Batch Normalization (Ioffe & Szegedy, 2015)
Normalize feature distributions to the standard distribution by learning batch statistics.
Consider a batch X
Calculate $m = E(X)$ and $\sigma = Var(X)$
Compute $\hat{X} = \frac{X - m}{\sigma} * \gamma + \beta$
m and σ are continuously updated across batches using running statistics, and γ and β are learnable parameters (by default set to 1 and 0, respectively)
Notes
Has been shown to accelerate training, increase generalization
Can remove the need for DropOut
Should be included by default after convolutions
http://proceedings.mlr.press/v37/ioffe15.pdf

SGD train- ing is complicated by the fact that the inputs to each layer are affected by the parameters of all preceding layers. The change in the distributions of layers' inputs presents a problem because the layers parameters need to continuously adapt to the new distribution. Batch Normalization serves to speed up convergence, and also allows the use of higher learning rates without risk of divergence.

Data Augmentation using image transformations

Translations, rotations, Scaling, Shifting in RGB, Crops, ...

Image from Albumentations `https://albumentations.ai/docs/examples/pytorch_classification/`



DA increases the amount of data that the model sees during training, it is only applied on the training set. Note that it's stochastic meaning that the model sees different augmented versions of the images in each epoch

Mixup, Cutout and Cutmix

Mixup

For a network F trained using Cross Entropy (CE),

- Sample x_i, x_j from the training data, associated to labels y_i, y_j.
- Defined mixed up data samples as $\tilde{x} = \lambda x_i + (1 - \lambda) x_j$
- $loss = \lambda CE(F(\tilde{x}), y_i) + (1 - \lambda) CE(F(\tilde{x}), y_j)$, where $\lambda \in [0, 1]$
- Train with backprop

Notes

- Has been shown to regularize training and achieves better generalization.
- Should be included most of the time when training classification networks !
- See Lab2.md for a proposed implementation

https://arxiv.org/pdf/1710.09412.pdf

Mixup, Cutout and Cutmix
Mixup
For a network F trained using Cross Entropy (CE),
Sample x_i, x_j from the training data, associated to labels y_i, y_j.
Defined mixed up data samples as $\tilde{x} = \lambda x_i + (1 - \lambda) x_j$
$loss = \lambda CE(F(\tilde{x}), y_i) + (1 - \lambda) CE(F(\tilde{x}), y_j)$, where $\lambda \in [0, 1]$
Train with backprop
Notes
Has been shown to regularize training and achieves better generalization.
Should be included most of the time when training classification networks !
See Lab2.md for a proposed implementation
https://arxiv.org/pdf/1710.09412.pdf

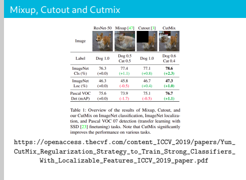
Mixup trains a neural network on convex combinations of pairs of examples and their labels. Mixup forces simple linear behavior in-between training samples, it is also robust against noisy labels

Mixup, Cutout and Cutmix

	ResNet-50	Mixup [47]	Cutout [3]	CutMix
Image				
Label	Dog 1.0	Dog 0.5 Cat 0.5	Dog 1.0	Dog 0.6 Cat 0.4
ImageNet Cls (%)	76.3 (+0.0)	77.4 (+1.1)	77.1 (+0.8)	**78.6 (+2.3)**
ImageNet Loc (%)	46.3 (+0.0)	45.8 (-0.5)	46.7 (+0.4)	**47.3 (+1.0)**
Pascal VOC Det (mAP)	75.6 (+0.0)	73.9 (-1.7)	75.1 (-0.5)	**76.7 (+1.1)**

Table 1: Overview of the results of Mixup, Cutout, and our CutMix on ImageNet classification, ImageNet localization, and Pascal VOC 07 detection (transfer learning with SSD [23] finetuning) tasks. Note that CutMix significantly improves the performance on various tasks.

https://openaccess.thecvf.com/content_ICCV_2019/papers/Yun_CutMix_Regularization_Strategy_to_Train_Strong_Classifiers_With_Localizable_Features_ICCV_2019_paper.pdf



Mixup boosts performance for classification (CLS) problems but degrades results for localization (LOCC) and objet detection (OO) problems. Cutout improves results for both CLS and LOC tasks but degrades performance for OO. CutMix improves results for all three tasks

Application to Self supervised Learning

Self-Supervised Learning

Learn representations of input samples without labels or annotations

How ?

Train encoders (e.g. ResNet) on pre-text tasks:

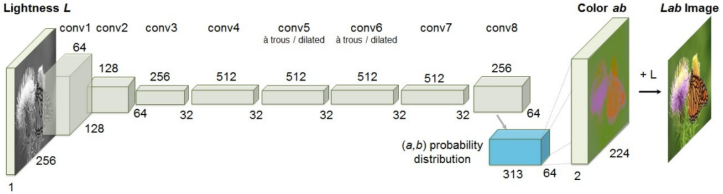
- Self-Prediction
- Contrastive Learning

Trained encoders are expected to learn general features that generalize to supervised tasks.

Application to Self supervised Learning
Self-Supervised Learning
Learn representations of input samples without labels or annotations
How ?
Train encoders (e.g. ResNet) on pre-text tasks:
Self-Prediction
Contrastive Learning
Trained encoders are expected to learn general features that generalize to supervised tasks.

SSL consists in learning a model that outputs useful representations of data without the use of label. This model can be used for transfer learning to other tasks. The model is trained in a supervised way with "labels" created from the data itself

Self-Prediction : Colorful Image Colorization

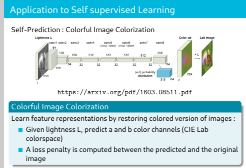


https://arxiv.org/pdf/1603.08511.pdf

Colorful Image Colorization

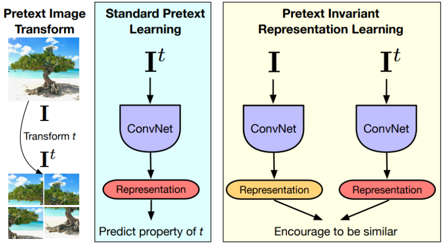
Learn feature representations by restoring colored version of images :

- Given lightness L, predict a and b color channels (CIE Lab colorspace)
- A loss penalty is computed between the predicted and the original image


Application to Self supervised Learning
Self-Prediction : Colorful Image Colorization
https://arxiv.org/pdf/1603.08511.pdf
Colorful Image Colorization
Learn feature representations by restoring colored version of images :
Given lightness L, predict a and b color channels (CIE Lab colorspace)
A loss penalty is computed between the predicted and the original image

Colorizing images is an example of self-prediction tasks that learns representation of data by predicting a part of the data (colors in this case)

Contrastive Learning : Pretext-Invariant Representations Learning (PIRL)

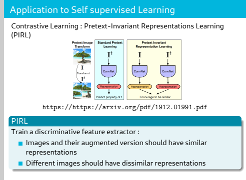


https://https://arxiv.org/pdf/1912.01991.pdf

PIRL

Train a discriminative feature extractor :

- Images and their augmented version should have similar representations
- Different images should have dissimilar representations



Contrastive learning is the state-of-the-art method for learning good representation of data, it consists in learning discriminative features by measuring similarities and dissimilarities between samples, it relies heavily on data augmentation