Training Deep Networks from Zero to Hero: avoiding pitfalls and going beyond - Part III

Moacir Antonelli Ponti, Fernando Pereira dos Santos, Leo S. F. Ribeiro, Gabriel B. Cavallari

ponti@usp.br, fernando_persan@alumni.usp.br, leo.sampaio.ferraz.ribeiro@usp.br, gabriel.cavallari@usp.br

October, 2021

Topics

- **Transfer learning**
- Feature extraction
- Fine-tuning network
- Curriculum learning

https://github.com/maponti/trainingdeepnetworks

Transfer Learning

Definition:

"Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned" [Torrey and Shavlik, 2010].

Definition:

"Transfer learning and domain adaptation refer to the situation where what has been learned in one setting can be exploited to improve generalization in another setting" [Goodfellow et al., 2016].

Transfer Learning

$$D = \{X, P(X)\}, where \begin{cases} X = \{x_1, ..., x_n\} : \textit{feature space} \\ P(X) : \textit{probabilistic distribution} \end{cases} \quad (1)$$

$$T = \{Y, f(.)\}, where \begin{cases} Y = \{y_1, .., y_m\} : \textit{label space} \\ f(.) : \textit{prediction function} \end{cases} \quad (2)$$

$f(.)$ models $P(y|x)$ for all $y \in Y$ and $x \in X$

Given a source domain D_s and a learning task T_s, a target domain D_t and a learning task T_t, TL aims to improve the function learning of the target prediction in D_t using the knowledge in D_s and T_s, where $D_s \neq D_t$ **or** $T_s \neq T_t$.

Transfer learning using deep networks

- Regular approach for deep networks:
  - design its topology;
  - define its training strategies;
  - randomly initializing all parameters;
  - train from scratch.

- Download models pre-trained using a large datasets, such as ImageNet, and fine-tune the model;

- Download models pre-trained using a large datasets, such as ImageNet, and use as feature extractor.

Topics

- Transfer learning
- **Feature extraction**
- Fine-tuning network
- Curriculum learning

Feature extraction for an external classifier

- choose the desired pre-trained network;
- choose the desired extraction layer;
- choose the desired classifier.

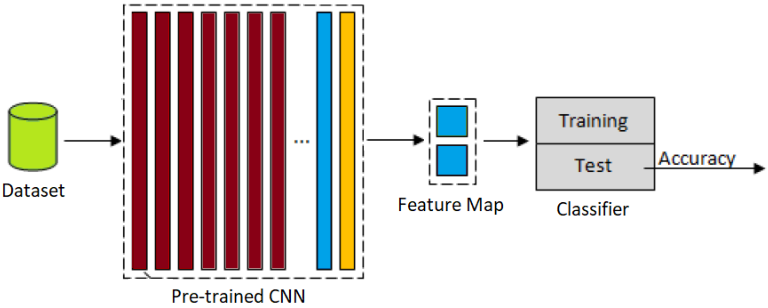

Figure 1: Feature extraction for an external classifier/detector.
(**Code: FeatureExtraction**)

Feature extraction and dimensionality reduction for an external classifier

- choose the desired pre-trained network;
- choose the desired extraction layer;
- **choose the dimensionality reduction technique**;
- choose the desired classifier.

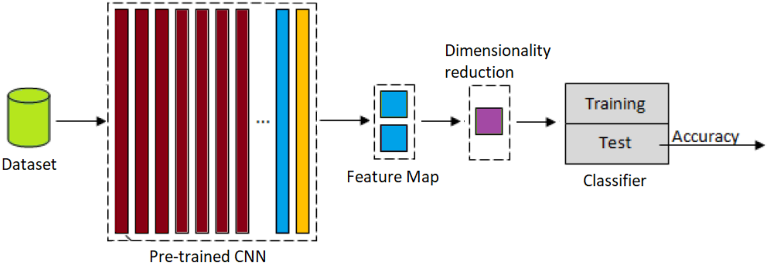

Figure 2: Feature extraction and dimensionality reduction for an external classifier/detector.

Multi-layer feature extraction for an external classifier

- choose the desired pre-trained network(s);
- choose the desired extraction layers;
- **feature map fusion**;
- **< choose the dimensionality reduction technique >**;
- choose the desired classifier.

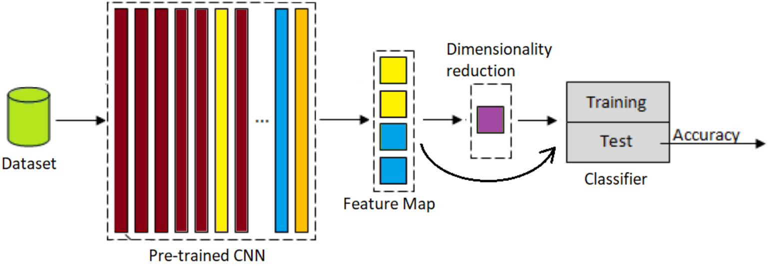

Figure 3: Multi-layer feature extraction for an external classifier/detector.

Topics

- Transfer learning
- Feature extraction
- **Fine-tuning network**
- Curriculum learning

Fine-tuning Network

Fine-tuning

It consists of reusing weights from pre-trained network with large datasets and refining the solution by training the model with the dataset of current task domain.

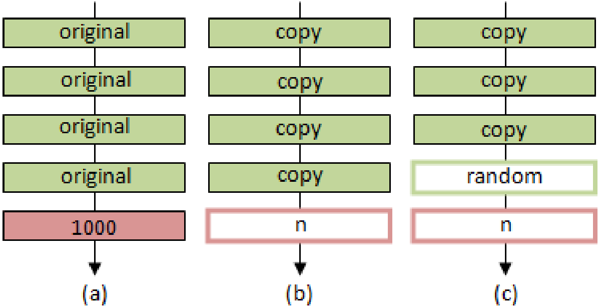

Figure 4: Fine-tuning: modifications in the structure, initialization of parameters, and new training.
(**Code: CNNFineTuning**)

Issues

- A preprocessing step must be applied to reduce or increase **resolutions**, impacting on loss of information or noisy addition.

- To ensure representativity of the domain is required a **large set of instances**:
  - Data Augmentation;
  - Similar domain;

- Weights Update:
  - propagation: weights from the previous training will be influenced by the new training;
  - frozen layers: only new layers are updated without changing the previous weights of the network.

Topics

- Transfer learning
- Feature extraction
- Fine-tuning network
- **Curriculum learning**

Curriculum learning

Curriculum learning

Curriculum learning is an advanced training strategy based on the concept of learning instances considered simpler before more difficult instances.

Figure 5: Different acquisition perspectives for contents of the same label: left) images that can be considered simpler for learning; b) images that can be considered more complex for learning, as they contain occlusions and multiple objects in the same scene.

Scoring function

- **Scoring function** is a metric to sort the training examples from the easiest to the most difficult.

- **Steps:**
  - assign the degree of difficulty to each instance;
  - sort the instances according to their degree of difficulty.

- **Transfer scoring function**:
  - extract features using the pre-trained CNN for the training set;
  - use the feature vector for prediction in an external classifier, getting the score for each instance.

- **Self-taught scoring function**:
  - the deep network is trained from scratch during few epochs;
  - the training set is predicted, getting the score for each instance.

Pacing function

- **Pacing function:** dictates the learning speed to incorporate more examples into the training set.
- When we consider a scoring function as a random measurement and a constant pacing function, we have the conventional training of the neural network.

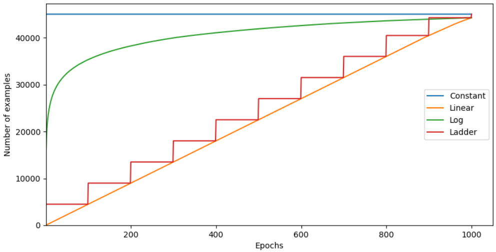

Figure 6: Pacing functions: Constant (conventional training), Linear, Logarithm, and Ladder (**Code: CurriculumLearning**)

Important considerations

- We have already dynamic functions that measure the degree of difficulty during network training;
- Keep the training set available balanced;
- Setting the learning rate wrongly can cause training performance degradation, being worse than conventional training.

Curriculum learning for tasks

- The easiest task is performed before the most difficult ones;
- Simpler loss functions can be used to start the training for a deep network and then alter to more complex loss functions;
- AutoEncoders can be use to initialize the weights before the supervised learning;
- Even when we are performing conventional fine-tuning, we are using the idea of curriculum learning.

AE to initialize weights

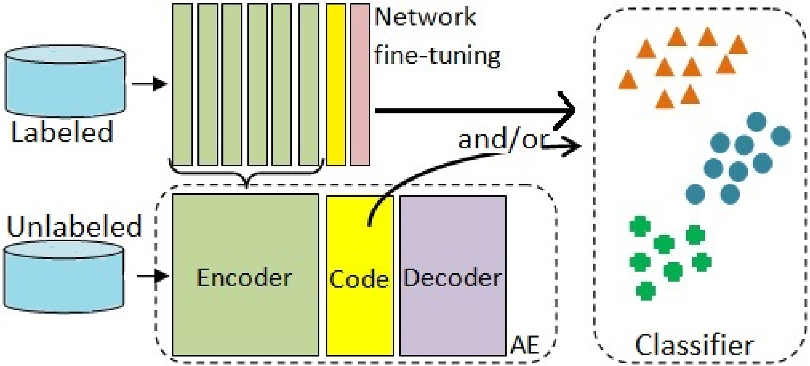

Figure 7: First, training an AE to weight the parameters; then, network fine-tuning to improve performance. (**Code: CurriculumTask**)

Complementary Readings

- PONTI, M.; RIBEIRO, L. S.; NAZARE, T. S.; BUI, T.; COLLOMOSSE, J. **Everything you wanted to know about deep learning for computer vision but were afraid to ask**. In: 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T 2017), 2017. p. 17–41.
- PAN, S. J.; YANG, Q.et al. **A survey on transfer learning**. IEEE Transactions on knowledge and data engineering, Institute of Electrical and Electronics Engineers, Inc., 345 E. 47 th St.NY NY 10017-2394 USA, v. 22, n. 10, p. 1345–1359, 2010.
- YOSINSKI, J.; CLUNE, J.; BENGIO, Y.; LIPSON, H. **How transferable are features in deep neural networks?** In: Advances in neural information processing systems, 2014. p.3320–3328.
- BENGIO, Y.; LOURADOUR, J.; COLLOBERT, R.; WESTON, J. **Curriculum learning**. In: 26th Annual International Conference on Machine Learning, 2009, pp. 41–48.
- HACOHEN, G.; WEINSHALL, D. **On the power of curriculum learningin training deep networks**. In: International Conference on Machine Learning, 2019, pp. 2535–2544.

Referencies I

Goodfellow, I. J., Bengio, Y., and Courville, A. (2016).
Deep learning.
MIT Press.

Torrey, L. and Shavlik, J. (2010).
Transfer learning.
In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pages 242–264. IGI Global.

Training Deep Networks from Zero to Hero: avoiding pitfalls and going beyond - Part III

Moacir Antonelli Ponti, Fernando Pereira dos Santos, Leo S. F. Ribeiro, Gabriel B. Cavallari

ponti@usp.br, fernando_persan@alumni.usp.br, leo.sampaio.ferraz.ribeiro@usp.br, gabriel.cavallari@usp.br

October, 2021