Utilizando redes profundas para processamento de imagens - um caso prático

Fernando Pereira dos Santos

fernando_persan@alumni.usp.br
fernando.santos@birdie.ai
https://www.linkedin.com/in/fernando-persan/
https://github.com/fernandopersan/blueedtech

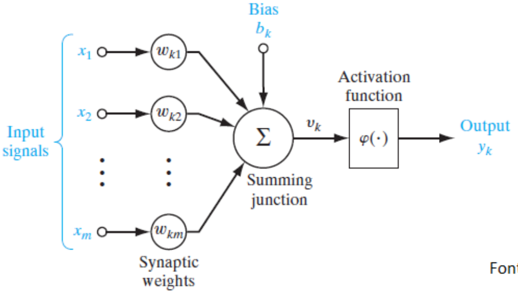
May, 2022

Topics

- **Deep Learning and CNNs**
- Feature extraction using pre-trained networks
- Fine-tuning network
- Unsupervised and semi-supervised approaches

Basics of Neural Networks

- the processing is carried out through units that store a numerical value, **neurons**, and their connections;
- each neuron k is composed of an input $X_k = \{x_1, x_1, ..., x_m\}$ and a single output y_k;
- each input value is associated with a weight $w_k = \{w_{k1}, w_{k2}, ..., w_{km}\}$ that indicates the strength of the connection;
- the cross product $w_k x_k$ is related to a value b_k called the bias that potentiates or not this product;
- an activation function Φ is responsible for interpreting the result processed in the neuron.



$$y_k = \Phi(\sum_{j=1}^{m} \omega_{kj}\, x_j + b_k)$$

Fonte: Neural Networks and Learning Machines (Haykin)

Figure 1: Basic NN structure.

Convolutional Neural Network

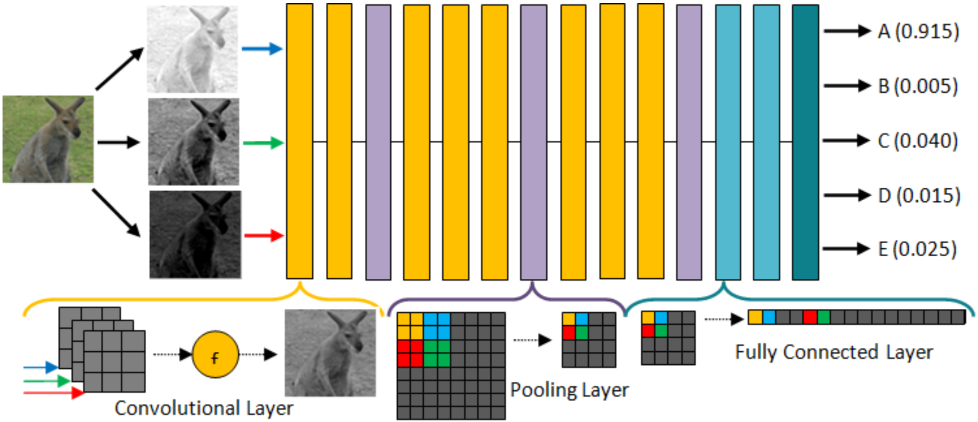


Figure 2: Generic CNN structure.

Convolution Neural Networks - Properties

$$f(x) = f_L(\ldots f_2(f_1(x_1, W_1), W_2), \ldots W_L) \quad (1)$$

- hierarchical structure;
- each layer provides a distinct feature map;
- low-level (colors and shapes) and high-level features (texture and semantics);
- end-layers are receptive fields of previous layer;
- great flexibility and high levels of cross-domain;

** **Low-level and top-level layers threshold is still uncertain**, with several heuristics prevailing to determine the optimal layer for each problem.

Inception

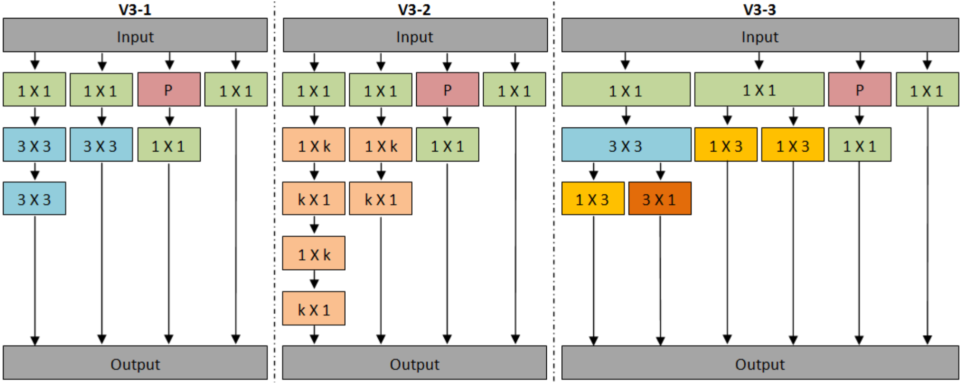


Figure 3: Inception Module. Three different modules were proposed: V3-1 incorporates the idea of small filters being more representative from VGGs; V3-2 applies factorization (orange blocks); and V3-3 increases space dimensionality by bank filters (also in orange blocks). Outputs performed feature map concatenation from each branch. All modules have in common two branches: pooling (*P*) with 1×1 convolution; and a single 1×1 convolution. Usually, *k* assumes 7.

ResNet

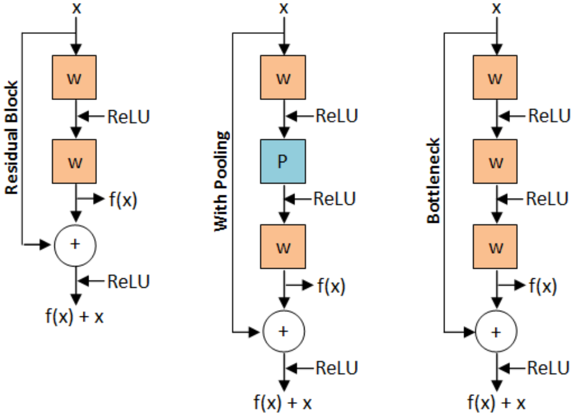


Figure 4: Residual Blocks. All three blocks perform sum of an input x with a data transformation $f(x)$, where w represents a convolutional layer and p defines a pooling layer. Bottleneck proposes to compress the input depth by a reduced number of filters and restore it before the sum.

Topics

- Deep Learning and CNNs
- **Feature extraction using pre-trained networks**
- Fine-tuning network
- Unsupervised and semi-supervised approaches

Feature extraction for an external classifier/detector

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

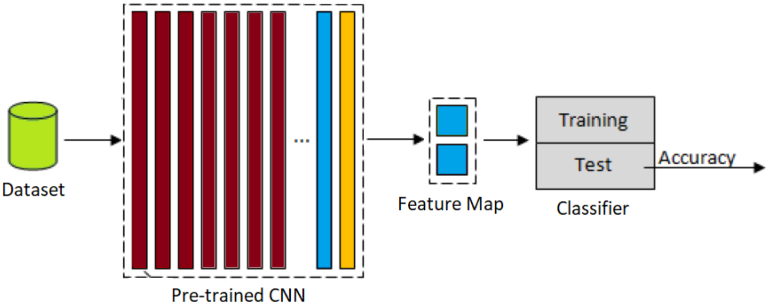


Figure 5: Feature extraction for an external classifier/detector. If necessary, apply cross-validation. (**Code: FeatureExtraction1**)

Feature extraction and dimensionality reduction for an external classifier/detector

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

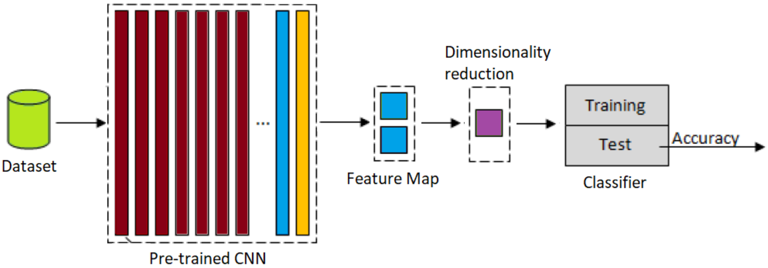


Figure 6: Feature extraction and dimensionality reduction for an external classifier/detector. If necessary, apply cross-validation. (**Code: FeatureExtraction2**)

Topics

- Deep Learning and CNNs
- Feature extraction using pre-trained networks
- **Fine-tuning network**
- Unsupervised and semi-supervised approaches

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.

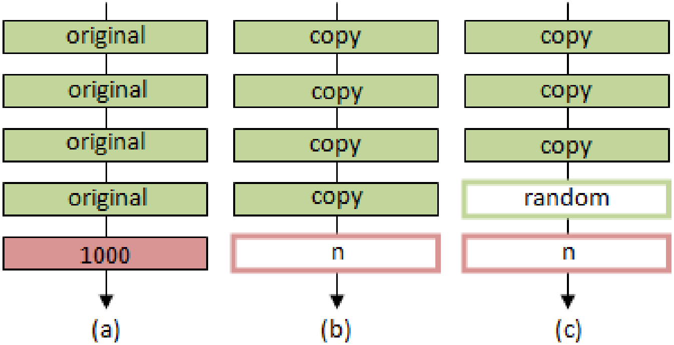


Figure 7: Fine-tuning: modifications in the structure, initialization of parameters, and new training.

Limited resolution

- ResNet50 with top: input of 224 X 224;
- ResNet50 without top: width and height should be no smaller than 197;

- Inception with top: input of 299 X 299;
- Inception without top: width and height should be no smaller than 139;

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

Large sample necessary

- sample representativeness;
- depth of architecture network (fixed);
- complexity of the task (not easy to measure);

- to ensure representativity of the domain is required a large set of instances;

- Option 1: Data Augmentation;
- Option 2: Similar domain.

Overtraining, overfitting, and overparametrization

- defining accurately when to stop the training;

- **overtraining** provides to the network a memorization of data distribution, resulting in poor performance with test samples and avoids generalization to other similar domains (**overfitting**);
- **overparametrization** increases the computational costs;

Network training

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;

Loss Function (measures the progress of network training):

- Cross-entropy will express the penalty for predicting a label $\hat{y}$ in which should be y;
- Mean Square Error will express the reconstruction error;

Network training (Code: CNNFineTuning)

Optimization Algorithm (adjusts parameters and minimize the loss function):

- Stochastic Gradient Descent (SGD);
- Adaptive Moment Estimation (Adam);

Batch Size (defines the amount of instances loaded in memory):

- some say it should be as large as possible, occupying all memory;
- some say that small batchs allow greater precision in minimizing loss;
- some say that a batch size with 32 examples is ideal, independent of the task;

Number of Iterations (epochs):

- until the network converges measured by the loss function.

Topics

- Deep Learning and CNNs
- Feature extraction using pre-trained networks
- Fine-tuning network
- **Unsupervised and semi-supervised approaches**

AutoEncoders

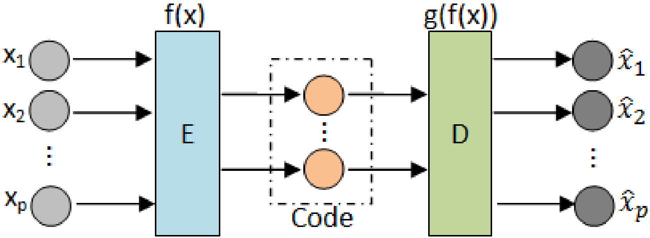


Figure 8: Generic Autoencoder: (E) encoder; (D) decoder.

$$f(x) = \Phi(Wx + b_e) \tag{2}$$

$$g(f(x)) = \Phi(Wf(x) + b_d) \tag{3}$$

Feature extraction from AutoEncoder

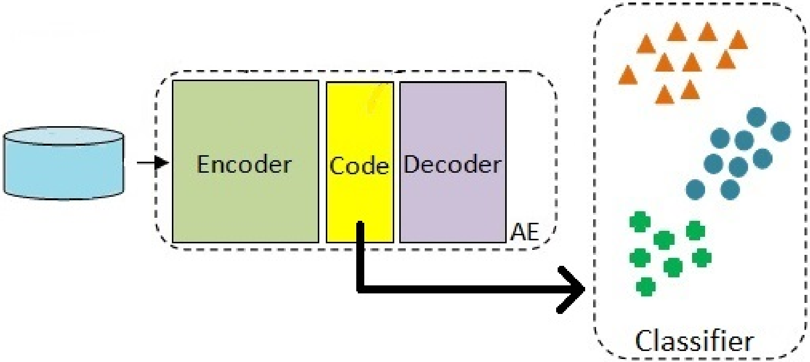


Figure 9: AE feature extraction.
(**Code: AutoEncoder**)

Feature extraction from AutoEncoder

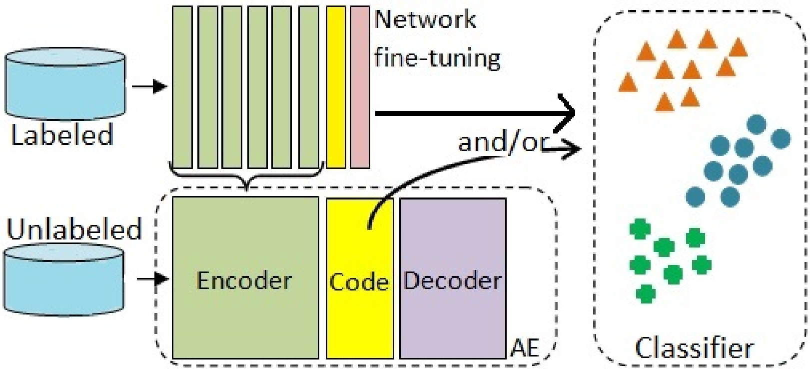


Figure 10: CNN fine-tuning using AE pre-trained encoder.

Final Considerations

- Deep network has been the state-of-the-art in many computer vision applications;
- Transfer learning is a widely exploited resource when looking for model generalization and in situations with low labeled data;
- Feature extraction using pre-trained networks allows generalization, post-processing, and low computational cost;
- Fine-tuning network is a transfer learning technique that provides excellent performance; however, it is necessary to model the network according to the task and the obtained data.

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: 30thSIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T 2017), 2017. p. 17–41.
- PONTI, M. A.; SANTOS, F. P.; RIBEIRO, L. S. F.; CAVALLARI, G. **Training deep networks from zero to hero: avoiding pitfalls and going beyond**. In: 34thSIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T 2021), 2021.
- HE, K.; ZHANG, X.; REN, S.; SUN, J. **Deep residual learning for image recognition**. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. p. 770–778.
- HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.;ANDREETTO, M.; ADAM, H. **Mobilenets: Efficient convolutional neural networks for mobile vision applications**. arXiv preprint arXiv:1704.04861, 2017.
- 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.
- SANTOS, F. P.; THUMÉ, G. S.; PONTI, M. A. **Data augmentation guidelines for cross-dataset transfer learning and pseudo labeling**. In: 34thSIBGRAPI Conference on Graphics, Patterns and Images, 2021.

Libraries documentation

- **Keras:** https://keras.io/api/
- **Sklearn:** https://scikit-learn.org/stable/
- **MatPlotLib:** https://matplotlib.org/
- **Colab:** https://colab.research.google.com/

Utilizando redes profundas para processamento de imagens - um caso prático

Fernando Pereira dos Santos

fernando_persan@alumni.usp.br
fernando.santos@birdie.ai
https://www.linkedin.com/in/fernando-persan/
https://github.com/fernandopersan/blueedtech

May, 2022