

# Layers in deep learning

*Jessica Tressou*

*19 février 2019*

From Keras Cheat sheet: [https://ugoproto.github.io/ugo\\_r\\_doc/keras.pdf](https://ugoproto.github.io/ugo_r_doc/keras.pdf)

## CORE LAYERS

- `layer_input`: précise les inputs mais peut aussi être intégrée dans la 1ère couche cachée
- `layer_dense`: ajoute d'une couche dont chaque neurone est lié à tous les noeuds de la couche précédente (densely connected neuron network layer). Nombre de neurones à définir (`units=`), en option la fonction d'activation peut être précisée directement (`activation=""`)
- `layer_activation`: précise la fonction d'activation à appliquer à l'output (peut être intégrée directement dans la description de la couche cachée, un paramètre de `layer_dense`)
- `layer_dropout` permet de régulariser en coupant une fraction des connections aléatoirement (paramètre: `rate` entre 0 et 1)
- `layer_reshape` pour modifier le format de l'output de la couche vers un certain format
- `layer_permute` permute les dimensions de l'input selon un pattern donné
- `layer_repeat_vector` répète l'input un certain nombre de fois
- `layer_lambda` change  $x$  en  $f(x)$  en précisant  $f$
- `layer_activity_regularization`
- `layer_masking`
- `layer_flatten`

## CONVOLUTIONAL LAYERS

- 1d temporal
- 2d or 2d\_transpose spatial
- 3d or 3d\_transpose spatial volume
- Long short-term memory (LSTM) = un recurrent neural network
- depthwise separable
- upsampling 1d, 2d, 3d
- zero padding 1d, 2d, 3d -> adding zeros on the borders (left and right for 1d, top, bottom left and right for 2d, ...)
- cropping 1d, 2d, 3d (deletes or removes a certain number of units, like cropping of an image)

## POOLING LAYERS

- max or average: takes the max or the average over a set of `pool_size` units (1d, 2d, 3d)
- global pooling: permet de régulariser/limiter l'overfitting

## ACTIVATION LAYERS

-> permet de définir la fonction d'activation à appliquer à l'output - linéaire - Logistic Sigmoid - Softmax - Seuil - ReLU = rectified linear unit = la fonction d'activation la plus utilisée :  $\phi(x) = \max(0, x)$  - Softplus  $\phi(x) = \log(1 + e^x)$  l'approximation lisse de ReLU, sa dérivée est la fonction logistique  $\phi'(x) = \frac{1}{1+e^{-x}}$

## **DROPOUT LAYERS**

-> le dropout est une méthode de régularisation qui coupe une fraction des connections (fixe à 0 une partie des poids) aléatoirement lors de la phase d'apprentissage. Cela permet de limiter l'overfitting.

cf . Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958. <http://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf>

## **RECURRENT LAYERS**

## **LOCALLY CONNECTED LAYERS**

-> similar to convolutional layers but weights are not shared (different filters for each patch) - 1d or 2d

## **Un exemple de construction (Deepsense)**

Yao, S., Hu, S., Zhao, Y., Zhang, A., & Abdelzaher, T. (2017, April). Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In Proceedings of the 26th International Conference on World Wide Web (pp. 351-360). International World Wide Web Conferences Steering Committee