Universität Augsburg
Fakultät für Angewandte Informatik

Deep Learning: Tutorial 8 - Autoencoders

Manuel Milling, Alice Baird, Thomas Wiest

11.12.2019

Gefördert vom Bundesministerium für Bildung und Forschung
LMU Ludwig-Maximilians-Universität München

Agenda

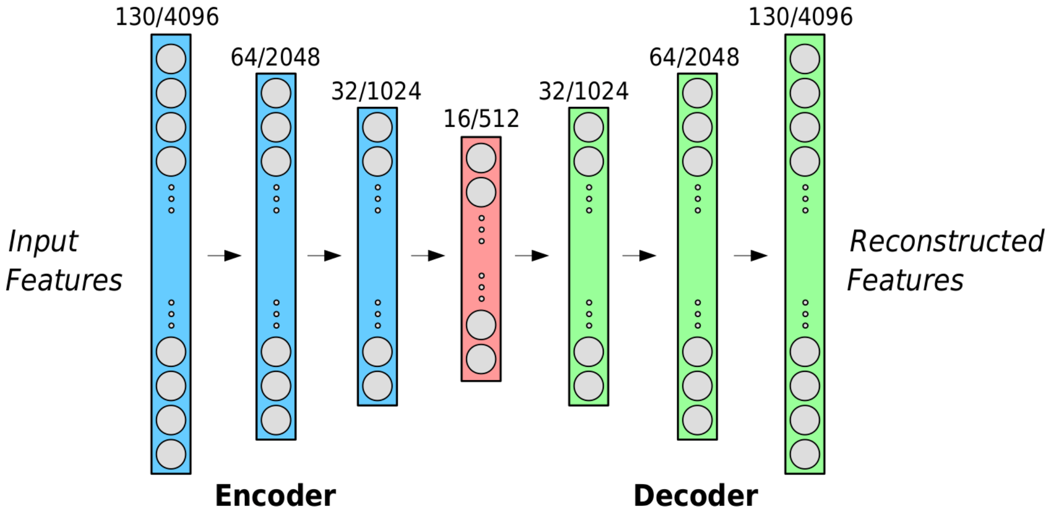
1. Autoencoders
2. Encoder - Decoder
3. Convolutional Layers in Autoencoders
4. Transpose Convolution

Autoencoders (AE)

- Autoencoders are a special kind of neural networks
- They can be assigned to the unsupervised learning technique (though it is sometimes stated that they self-supervised)
  - Therefore, AEs do not require labelled data
- The Goal of an AE is to learn a representation of data
- The method to do so consists of two steps
  - Firstly, compress the data to a smaller representation
  - Secondly, try to reconstruct the original data from the compressed data
- Applications
  - Learned representations can be used e.g., for classification tasks
  - AE can learn to reconstruct noise-less data from noisy data
  - Variational AEs are quite populer for data generation
- However, AEs do not necessarily have a very good performance on above-mentioned tasks

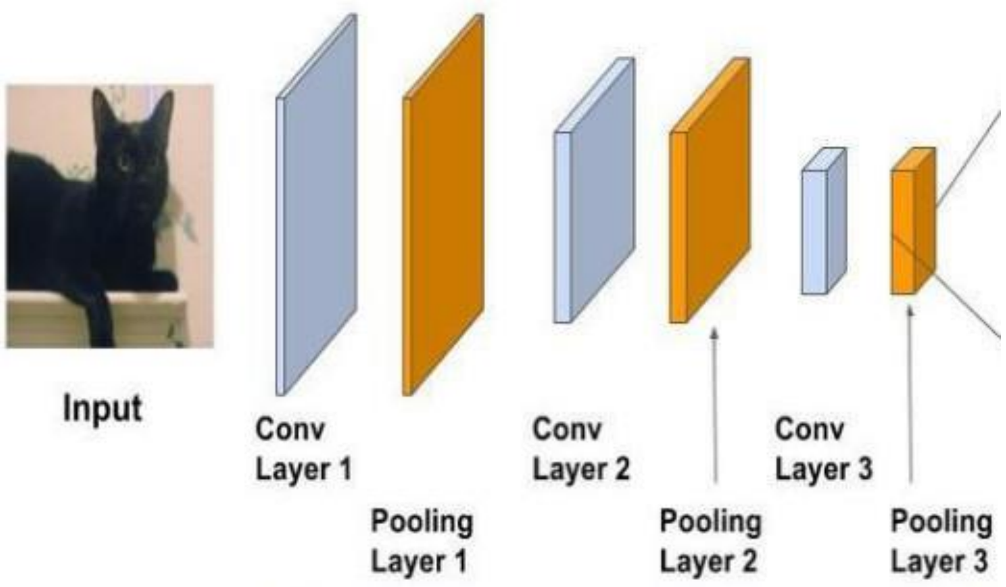
Encoder - Decoder

- An AE tries to reconstruct the data of the input using two components: an Encoder and a Decoder
- The Encoder is the part first part of the neural network, connecting the input to the representation layer
  - The representation layer should have a lower dimensionality/amount of features than the input layer (compression)
- The Decoder is the second part of the neural network connecting the representation layer to the output layer
  - The output layer has to be of the same dimension as the input, (input data = labels of output layer)
  - It is quite common to use a „symmetric“ architecture for encoder and decoder
- The AE is trained to minimise the reconstruction error (e.g., MSE between input and output layer) in the same manner as other neural networks



Convolutional Layers in AEs

- The simplest implementation of an AE consists of a set of fully connected layers, which decrease in size for the encoder and increase in size for the decoder
- For images however convolutional layers are much more helpful than fully connected layers.
- Regular convolutional layers are well suited for the encoder
- They generally decrease the height and width of the feature maps (even when increasing the number of feature maps)
- The decoder however has to transform the encoded representation back to the same „format“ as the original image



Transpose Convolutions

- Tranpose convolutions can be used to increase the width and height of the feature maps, which can not be done by traditional convolutions
- In regular convolutions the weights connect many features of an input feature map with one feature of an output feature map (many to one)
- In a transposed convolution one feature of an input feature map is connected to many features of an output feature map
- The transposed convolution can be interpreted as symmetric to the regular convolution, which is why the transposed convolution seems helpful for autoencoders.
- The figure on the right illustrates a transposed convolution with stride 2 for 2 feature maps (top-down)
- A down-top interpretations of the figure on the right illustrates a regular convolution with stride 2