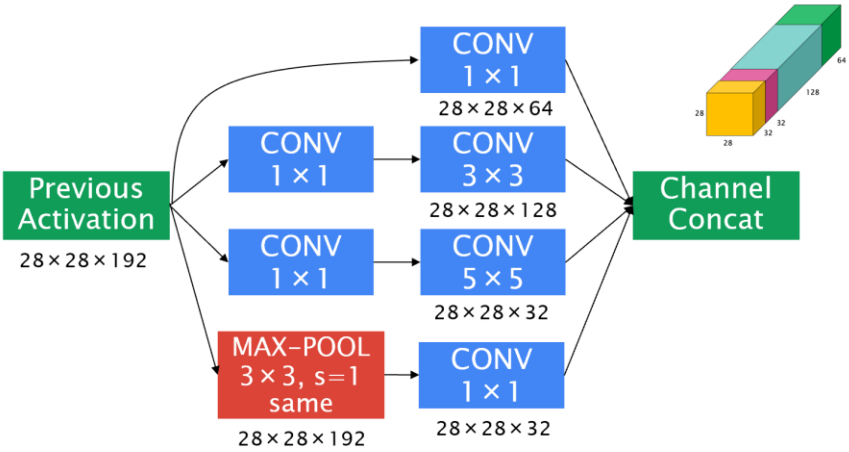
Defining advanced models

The goal of this set of exercises is to reproduce common deep learning architectures by using Pytorch. It is not requested to train each network, but to define the model and visualize it to verify that the architecture is reproduced correctly, including the input and output shape.

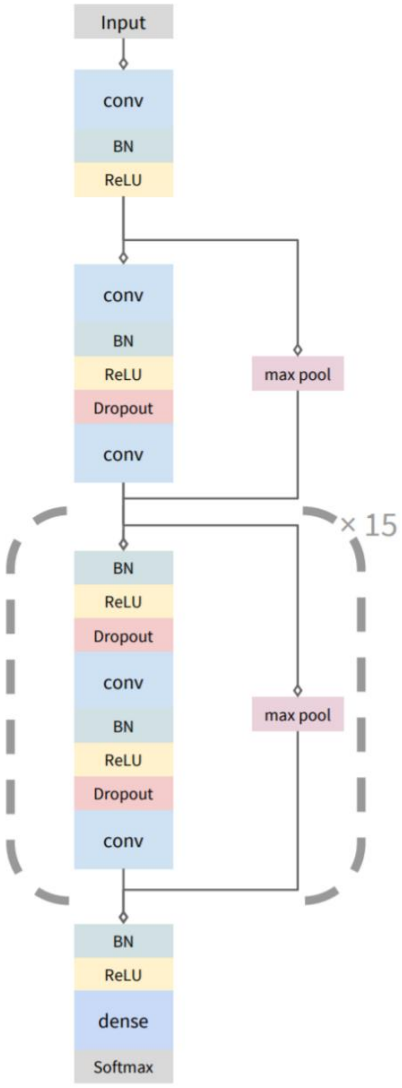
Exercise 1

Create a function with implements a single Inception module with the architecture depicted below.



Exercise 2

Convolutional neural networks can be applied to one-dimensional as well as two- or three-dimensional input. Let us implement a residual neural network that was proposed for ECG interpretation[1]. The network is trained on 30 second long signals sampled at 200Hz. The architecture is depicted in the figure to the right.



The convolutional layers all have a filter length of 16 and have 64. Every alternate residual block subsamples its inputs by a factor of 2, thus the original input is ultimately subsampled by a factor of 2^8. When a residual block subsamples the input, the corresponding shortcut connections also subsample their input using a Max Pooling operation with the same subsample factor.

The final fully connected layer and softmax activation produce a distribution over the 14 output classes for each time-step.

[1] https://arxiv.org/pdf/1707.01836.pdf

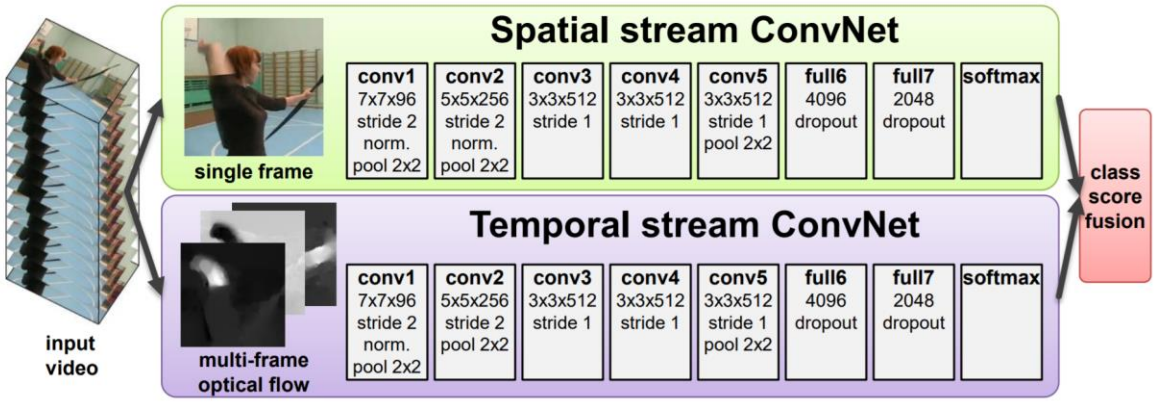
Exercise 3

Implement a two stream network for human action classification as proposed by Simonyan and Zisserman in Two-Stream Convolutional Networks for Action Recognition in Videos[2]. The task consists in classifying the frames in a video according to the action performed by the human. This particular architectures operates on a frame by frame basis.

Assuming that:

- The network takes as input one single frame and its corresponding optical flow
    - The optical flow[3] is already calculated and provided as additional input
    - The input size of both RGB and optical flow are 224 x 224
    - The optical flow for 2*L consecutive frames are encoded as 2*L grayscale input channels to the temporal stream branch
- The two branches have the same architecture but separate (not shared) parameters
- The number of possible actions is 100
- The class score fusion consists in taking the average of the two predictions

implement the architecture and calculate the total number of parameters.



[2] http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf
[3] The optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene.

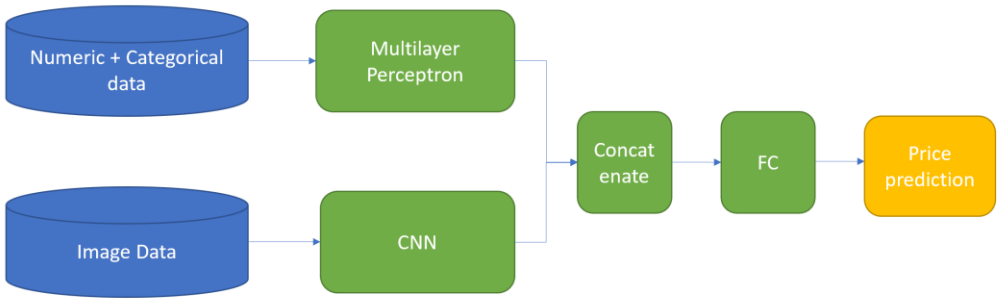
Exercise 4

Let us define a neural network to predict the price of the house starting from

- One or more images of the interior and exterior
- A set of categorical data

In particular, refer to the dataset described here to retrieve the list of input variables https://github.com/emanhamed/Houses-dataset

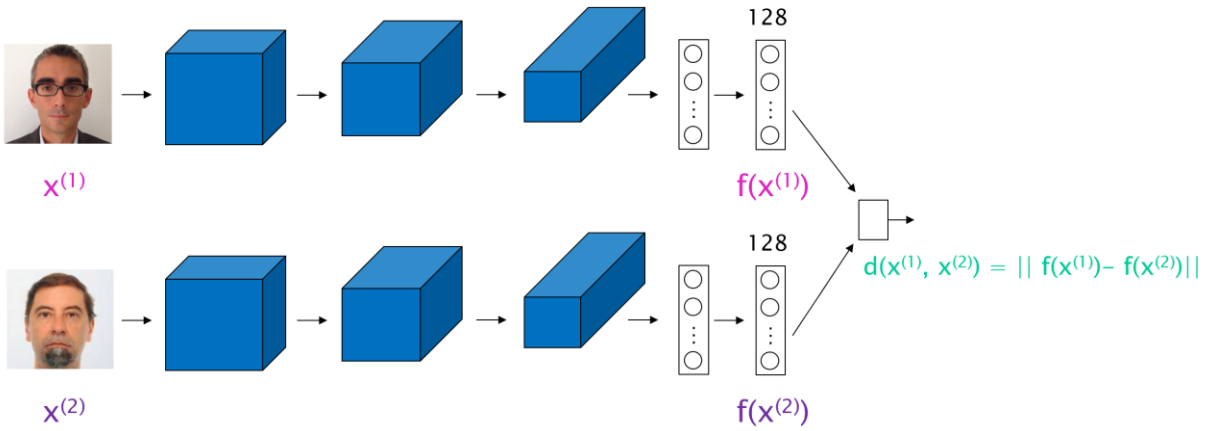
The high level architecture is depicted in the figure below. For extracting features from the image, use a CNN of your choice pre-trained on ImageNet.



Exercise 5 (optional)

1) Implement a simple Siamese network for face verification. The Siamese network must

- take as input two images
- convert them into a vector of fixed length using a set of convolutional and dense layers (hint: remember that both stream need to share the weights!)
- compute the distance



2) At inference time, only the part of the model that computes f(x) is needed. Extract the subnetwork from the Siamese model