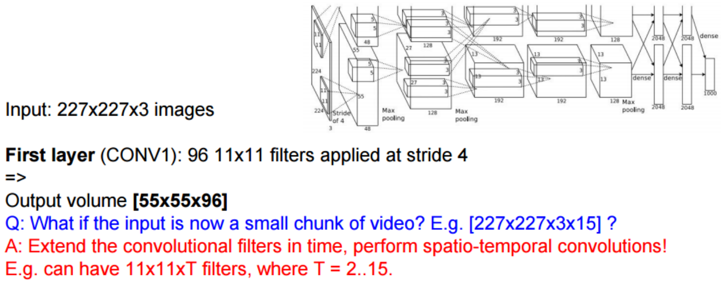
Chapter 1

RNN ConvNet

Local motion


Stride of 4
Max pooling
Max pooling
Max pooling
dense
dense
dense

Input: 227x227x3 images

First layer (CONV1): 96 11x11 filters applied at stride 4
=>
Output volume **[55x55x96]**
Q: What if the input is now a small chunk of video? E.g. [227x227x3x15] ?
A: Extend the convolutional filters in time, perform spatio-temporal convolutions!
E.g. can have 11x11xT filters, where T = 2..15.

FIGURE 1.1: Getting there

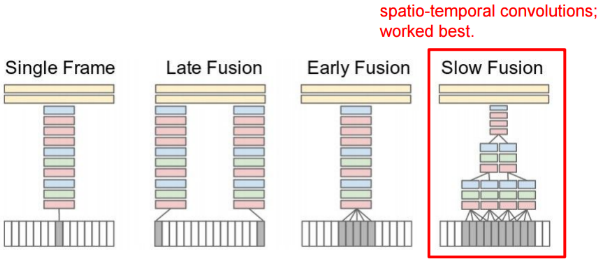


FIGURE 1.2: Large-scale Video Classification with Convolutional Neural Networks, Karpathy et al., 2014]

Global motion

Delving Deeper into Convolutional Networks for Learning Video Representations, Ballas et al., 2016

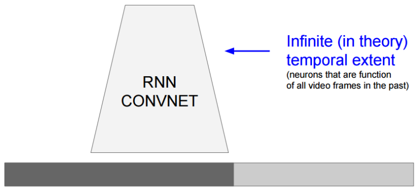
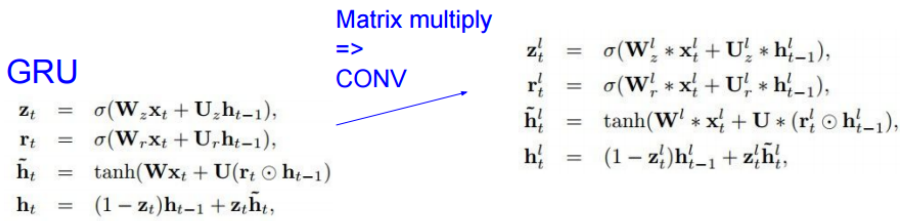


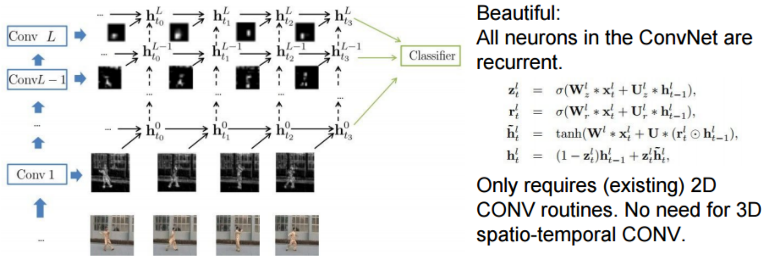
FIGURE 1.3: Large-scale Video Classification with Convolutional Neural Networks, Karpathy et al., 2014]


GRU
Matrix multiply
=>
CONV

$$\begin{aligned} \mathbf{z}_t &= \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1}), \\ \mathbf{r}_t &= \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1}), \\ \tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}\mathbf{x}_t + \mathbf{U}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) \\ \mathbf{h}_t &= (1-\mathbf{z}_t)\mathbf{h}_{t-1} + \mathbf{z}_t\tilde{\mathbf{h}}_t, \end{aligned}$$

$$\begin{aligned} \mathbf{z}_t^l &= \sigma(\mathbf{W}_z^l * \mathbf{x}_t^l + \mathbf{U}_z^l * \mathbf{h}_{t-1}^l), \\ \mathbf{r}_t^l &= \sigma(\mathbf{W}_r^l * \mathbf{x}_t^l + \mathbf{U}_r^l * \mathbf{h}_{t-1}^l), \\ \tilde{\mathbf{h}}_t^l &= \tanh(\mathbf{W}^l * \mathbf{x}_t^l + \mathbf{U} * (\mathbf{r}_t^l \odot \mathbf{h}_{t-1}^l), \\ \mathbf{h}_t^l &= (1-\mathbf{z}_t^l)\mathbf{h}_{t-1}^l + \mathbf{z}_t^l\tilde{\mathbf{h}}_t^l, \end{aligned}$$

FIGURE 1.4: The idea is to modify GRU adding convolutions. The Gated Recurrent Unit (GRU) is a simplified version of an LSTM unit with fewer parameters. Just like an LSTM cell, it uses a gating mechanism to allow RNNs to efficiently learn long-range dependency by preventing the vanishing gradient problem. The GRU consists of a reset and update gate that determine which part of the old memory to keep vs. update with new values at the current time step.


Conv L
ConvL − 1
Conv 1
Classifier
$\mathbf{h}_{t_0}^L$ $\mathbf{h}_{t_1}^L$ $\mathbf{h}_{t_2}^L$ $\mathbf{h}_{t_3}^L$
$\mathbf{h}_{t_0}^{L-1}$ $\mathbf{h}_{t_1}^{L-1}$ $\mathbf{h}_{t_2}^{L-1}$ $\mathbf{h}_{t_3}^{L-1}$
$\mathbf{h}_{t_0}^0$ $\mathbf{h}_{t_1}^0$ $\mathbf{h}_{t_2}^0$ $\mathbf{h}_{t_3}^0$

Beautiful:
All neurons in the ConvNet are recurrent.

$$\begin{aligned} \mathbf{z}_t^l &= \sigma(\mathbf{W}_z^l * \mathbf{x}_t^l + \mathbf{U}_z^l * \mathbf{h}_{t-1}^l), \\ \mathbf{r}_t^l &= \sigma(\mathbf{W}_r^l * \mathbf{x}_t^l + \mathbf{U}_r^l * \mathbf{h}_{t-1}^l), \\ \tilde{\mathbf{h}}_t^l &= \tanh(\mathbf{W}^l * \mathbf{x}_t^l + \mathbf{U} * (\mathbf{r}_t^l \odot \mathbf{h}_{t-1}^l), \\ \mathbf{h}_t^l &= (1-\mathbf{z}_t^l)\mathbf{h}_{t-1}^l + \mathbf{z}_t^l\tilde{\mathbf{h}}_t^l, \end{aligned}$$

Only requires (existing) 2D CONV routines. No need for 3D spatio-temporal CONV.

FIGURE 1.5: Original paper diagram

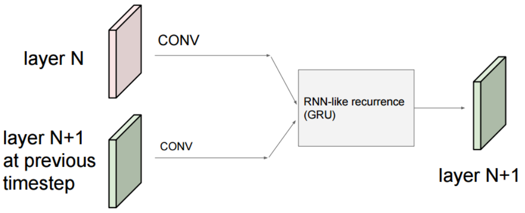


FIGURE 1.6: A more clear diagram

Summary

1. You think you need a Spatio-Temporal Fancy VideoConvNet
2. STOP. Do you really?
3. Okay fine: do you want to model:
    - local motion? (use 3D CONV), or
    - global motion? (use LSTM).

4. Try out using Optical Flow in a second stream (can workbetter sometimes)

5. Try out GRU-RCN! (imo best model)