Published in Towards Data Science

Rani Horev Follow

Dec 27, 2018 · 6 min read

SlowFast Explained: Dual-mode CNN for Video Understanding

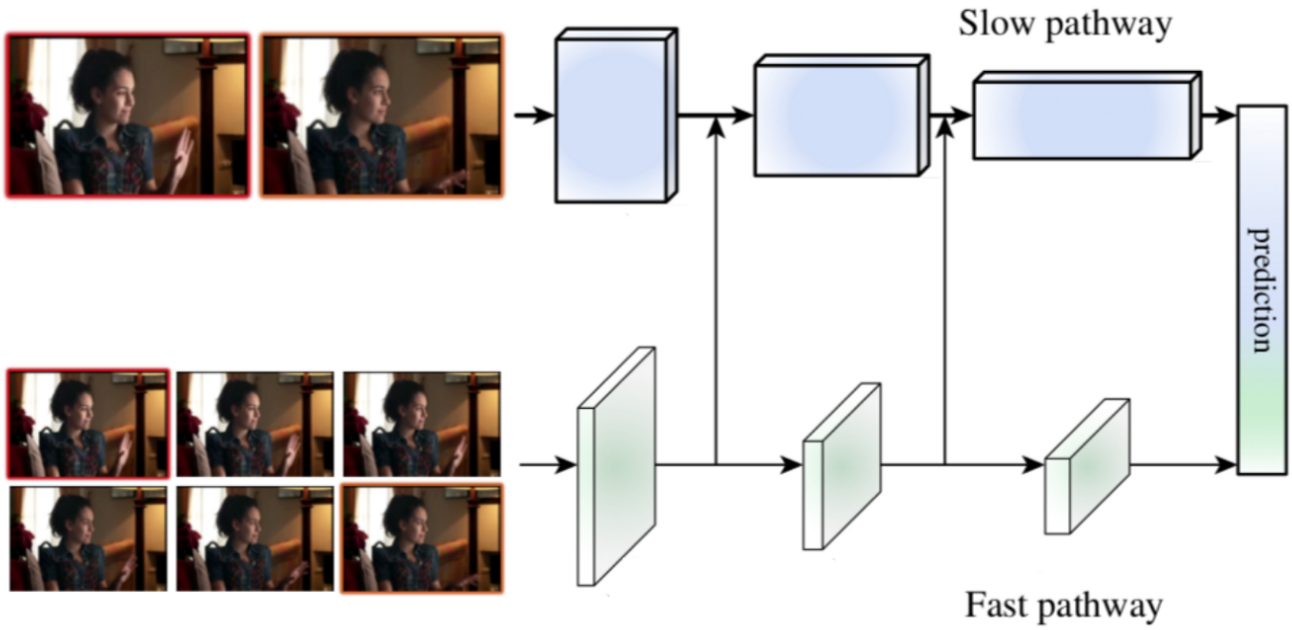
State-of-the-art deep learning architecture inspired by the visual mechanism of primates

Detecting objects in images and categorizing them is one of the more well-known Computer Vision tasks, popularized by the 2010 ImageNet dataset and challenge. While much progress has been achieved on ImageNet, a still vexing task is video understanding — analyzing a video segment and explaining what's happening inside of it. Despite some recent progress on solving video understanding, contemporary algorithms are still far from human-level results.

A new paper from Facebook AI Research, SlowFast, presents a novel method to analyze the contents of a video segment, achieving state-of-the-art results on two popular video understanding benchmarks — Kinetics-400 and AVA. At the heart of the method is the use of two parallel convolution neural networks (CNNs) on the same video segment — a Slow pathway and a Fast pathway.

The authors observe that frames in video scenes usually contain two distinct parts — static areas in the frame which don't change at all or change slowly, and dynamic areas which indicate something important that is currently going on. For instance, a video of a plane lifting off will include a relatively static airport with a dynamic object (the plane) moving quickly in the scene. In an everyday scenario of two people meeting, the

pathway) whose goal is to analyze the dynamic content of a video. The technique is partially inspired by the retinal ganglion in primates, in which 80% of the cells (P-cells) operate at low temporal frequency and recognize fine details, and ~20% of the cells (M-cells) operate at high temporal frequency and are responsive to swift changes. Similarly, in SlowFast the compute cost of the Slow pathway is 4x larger than that of the Fast pathway.



High-level illustration of the SlowFast network. (Source: SlowFast)

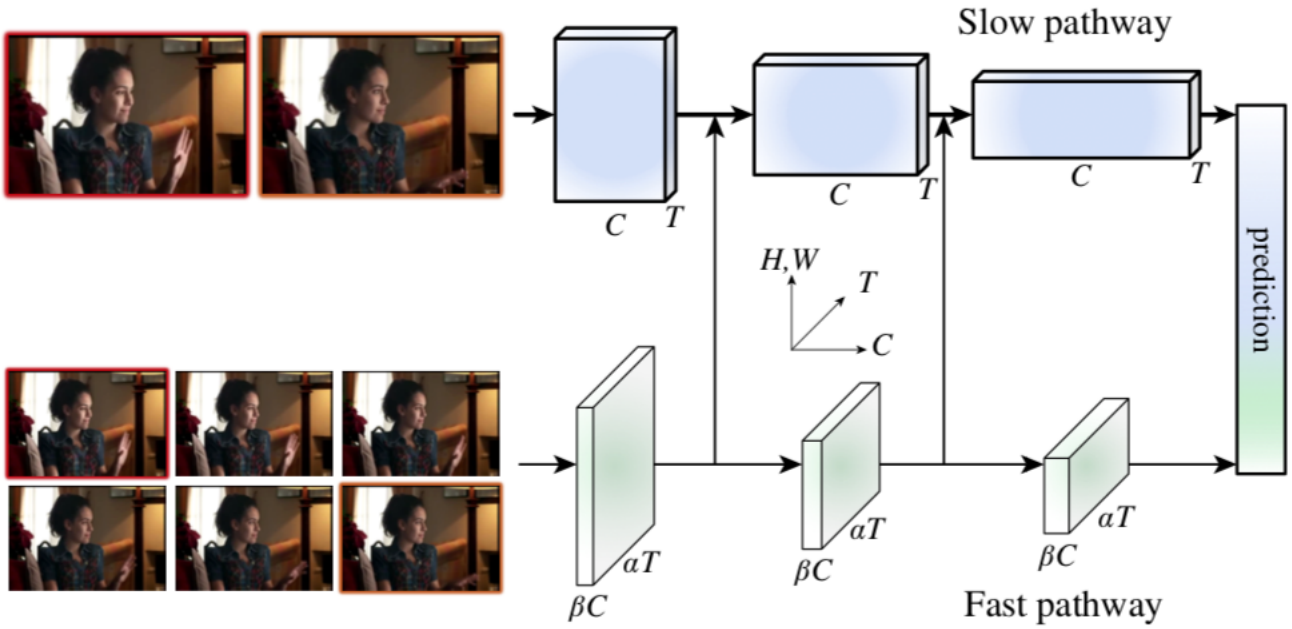
How SlowFast Works

Both the Slow and Fast pathways use a 3D ResNet model, capturing several frames at once and running 3D convolution operations on them.

The Slow pathway uses a large temporal stride (i.e. number of frames skipped per second) **τ**, typically set at 16, allowing for approximately 2 sampled frames per second. The Fast pathway uses a much smaller temporal stride τ/α, with α typically set at 8, allowing for 15 frames per second. The Fast pathway is kept lightweight by using a significantly smaller channel size (i.e. convolution width; number of filters used), typically set at ⅛ of the Slow channel size. The channel size of the Fast pathway is

stage	*Slow* pathway	*Fast* pathway	output sizes $T \times S^2$
raw clip	-	-	64×224^2
data layer	stride 16, 1^2	stride **2**, 1^2	*Slow* : 4×224^2 *Fast* : 32×224^2
conv_1	1×7^2, 64 stride 1, 2^2	5×7^2, 8 stride 1, 2^2	*Slow* : 4×112^2 *Fast* : 32×112^2
pool_1	1×3^2 max stride 1, 2^2	1×3^2 max stride 1, 2^2	*Slow* : 4×56^2 *Fast* : 32×56^2
res_2	$\begin{bmatrix} 1 \times 1^2, 64 \\ 1 \times 3^2, 64 \\ 1 \times 1^2, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 1^2, 8 \\ 1 \times 3^2, 8 \\ 1 \times 1^2, 32 \end{bmatrix} \times 3$	*Slow* : 4×56^2 *Fast* : 32×56^2
res_3	$\begin{bmatrix} 1 \times 1^2, 128 \\ 1 \times 3^2, 128 \\ 1 \times 1^2, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 1^2, 16 \\ 1 \times 3^2, 16 \\ 1 \times 1^2, 64 \end{bmatrix} \times 4$	*Slow* : 4×28^2 *Fast* : 32×28^2
res_4	$\begin{bmatrix} 3 \times 1^2, 256 \\ 1 \times 3^2, 256 \\ 1 \times 1^2, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 3 \times 1^2, 32 \\ 1 \times 3^2, 32 \\ 1 \times 1^2, 128 \end{bmatrix} \times 6$	*Slow* : 4×14^2 *Fast* : 32×14^2
res_5	$\begin{bmatrix} 3 \times 1^2, 512 \\ 1 \times 3^2, 512 \\ 1 \times 1^2, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 1^2, 64 \\ 1 \times 3^2, 64 \\ 1 \times 1^2, 256 \end{bmatrix} \times 3$	*Slow* : 4×7^2 *Fast* : 32×7^2
global average pool, concate, fc			# classes

An example instantiation of the SlowFast network. The dimensions of kernels are denoted by {T×S², C} for temporal (T), spatial (S), and channel © sizes. Strides are denoted as {temporal stride, spatial stride ^ 2}. The speed ratio (frame skipping rate) is α = 8 and the channel ratio is 1/β = 1/8. τ is 16. The green colors mark higher temporal resolution, and orange colors mark fewer channels, for the Fast pathway. The lower temporal resolution of the Fast pathway can be observed in the data layer row while the smaller channel size can be observed in the conv1 row and afterward in the residual stages. Residual blocks are shown by brackets. The backbone is ResNet-50. (Image & Description from SlowFast)



Lateral Connections

As shown in the visual illustration, data from the Fast pathway is fed into the Slow pathway via lateral connections throughout the network, allowing the Slow pathway to become aware of the results from the Fast pathway. The shape of a single data sample is different between the two pathways (Fast is $\{\alpha T, S^2, \beta C\}$ while Slow is $\{T, S^2, \alpha\beta C\}$), requiring SlowFast to perform data transformation on the results of the Fast pathway, which is then fused into the Slow pathway by summation or concatenation.

The paper suggests three techniques for data transformation, with the third one proving in practice to be the most effective:

1. Time-to-channel: Reshaping and transposing $\{\alpha T, S^2, \beta C\}$ into $\{T , S^2, \alpha\beta C\}$, meaning packing all α frames into the channels of one frame.
2. Time-strided sampling: Simply sampling one out of every α frames, so $\{\alpha T , S^2, \beta C\}$ becomes $\{T , S^2, \beta C\}$.
3. Time-strided convolution: Performing a 3D convolution of a 5×12 kernel with $2\beta C$ output channels and stride = α.

Interestingly, the researchers found that bidirectional lateral connections, i.e. also feeding the Slow pathway into the Fast pathway, do not improve performance.

Combining the pathways

At the end of each pathway, SlowFast performs Global Average Pooling, a standard operation intended to reduce dimensionality. It then concatenates the results of the two pathways and inserts the concatenated result into a fully connected classification layer which uses Softmax to classify which action is taking place in the image.

Datasets

SlowFast was tested on two major datasets — Kinetics-400, created by DeepMind, and AVA, created by Google. While both datasets include annotations for video scenes, they differ slightly:

- AVA includes 430 15-minute annotated YouTube videos, with 80 atomic visual actions. Each action is both described and located within a bounding box.

Results

SlowFast achieves state-of-the-art results on both datasets. In Kinetics-400 it surpasses the best top-1 score by 5.1% (79.0% vs 73.9%) and the best top-5 score by 2.7% (93.6% vs 90.9%). It also achieves state-of-the-art results on the new Kinetics-600 dataset, which is similar to the Kinetics-400 dataset but with 600 categories of human actions, each represented in at least 600 videos.

For AVA testing, the SlowFast researchers first used a version of the Faster R-CNN object detection algorithm, combined with an off-the-shelf person detector, providing a set of regions-of-interest. They then pre-trained the SlowFast network on the Kinetics dataset, and finally ran it on the regions-of-interest. The result was 28.3 mAP (median average precision) a dramatic improvement on the AVA state-of-the-art of 21.9 mAP. It's worth noting that the compared results also pre-trained on Kinetics-400 and Kinetics-600, providing no special advantage to SlowFast vs previous results.

Interestingly, the paper compares the results of the Slow-only and Fast-only networks to the combined network. In Kinetics-400, Slow-only achieves a top-1 result of 72.6% and a 90.3% top-5 score while Fast-only achieves a top-1 result of 51.7% and a top-5 result of 78.5%.

	Top-1 result	Top-5 result
Slow-Only	72.6%	90.3%
Fast-Only	51.7%	78.5%
Previous State-of-the-art	73.9%	90.9%
SlowFast	79.0%	93.6%

Compute

SlowFast is lighter in compute compared to standard ResNet implementations, requiring 20.9 GFLOPs to reach convergence in the Slow network and 4.9 GFLOPs in the Fast network, compared to 28.1 to 44.9 GFLOPs in common 3D ResNet-50 baselines on the same dataset.

Implementation Details

SlowFast is implemented in PyTorch and will be open-sourced.

Conclusion

SlowFast presents a novel and interesting approach to video understanding, taking advantage of the intuitive structure of real-world scenes and getting some inspiration from biological mechanisms. The paper shows that further optimizations of the model, such as using a deeper ResNet or applying additional established Computer Vision techniques, can achieve even better results and further our ability to use software to understand real-world situations.

To stay updated with the latest Deep Learning research, subscribe to my newsletter on LyrnAI