

Image & AI Reading Group

Dynamic Neural Networks: A Survey

Yizeng Han*, Gao Huang*, *Member, IEEE*, Shiji Song, *Senior Member, IEEE*, Le Yang, Honghui Wang, and Yulin Wang

Abstract—Dynamic neural network is an emerging research topic in deep learning. Compared to static models which have fixed computational graphs and parameters at the inference stage, dynamic networks can adapt their structures or parameters to different inputs, leading to notable advantages in terms of accuracy, computational efficiency, adaptiveness, etc. In this survey, we comprehensively review this rapidly developing area by dividing dynamic networks into three main categories: 1) *instance-wise* dynamic models that process each instance with data-dependent architectures or parameters; 2) *spatial-wise* dynamic networks that conduct adaptive computation with respect to different spatial locations of image data and 3) *temporal-wise* dynamic networks that perform adaptive inference along the temporal dimension for sequential data such as video. We also discuss some open problems of dynamic networks, e.g., architecture design, decision-making, and application.

Karol Desnos, Ph.D.

Associate Professor

IETR-VAADER, Rennes, France

My Research Interests:

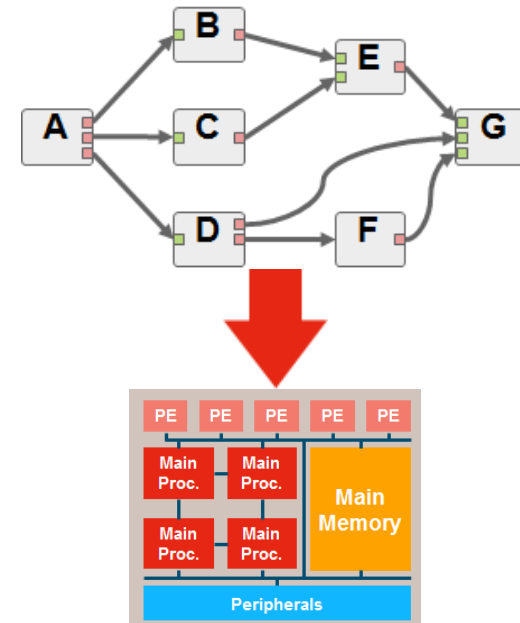
- **Model-Based Computer Aided Design**

- Dataflow models of computations
- Adaptive cyber-physical systems
- Multi-optimized: Energy, Memory, Latency, QoS, ...

- **Embeddable Reinforcement Learning**



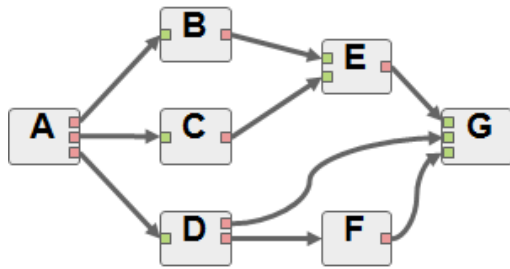
- **Image/Video Coding for Machines**



Why today's paper

Dynamic Neural Networks: A Survey

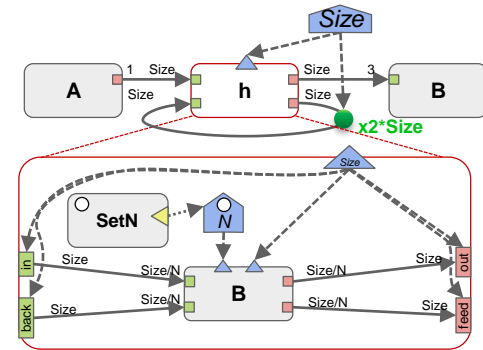
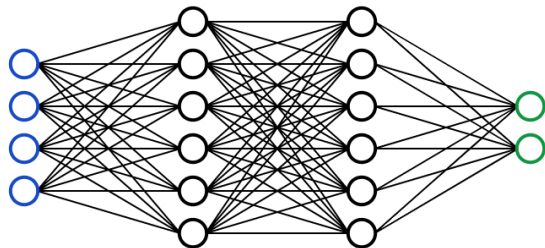
- Echo dataflow research



Synchronous Dataflow

≈

Deep Neural Network



Reconfigurable/Dynamic Dataflow

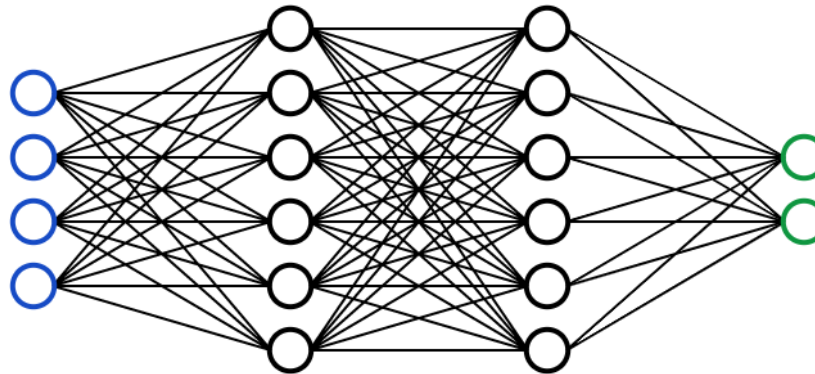
?

Dynamic Neural Network

?

(Static) Neural Network

- Computations on all input samples are identical
 - Fixed network architecture and data-path
 - All input samples are processed by all neurons
 - Learned weights are fixed by training



$$y = \mathcal{F}(x, \Theta)$$

x : input

$\mathcal{F}(\cdot, \Theta)$: CNN Function parameterized by Θ

y : output

Θ : Learnable Parameter

(Static) Neural Network

- Computations on all input samples are identical

$$y = \mathcal{F}(x, \Theta)$$

x : input $\mathcal{F}(\cdot, \Theta)$: CNN Function parameterized by Θ
 y : output Θ : Learnable Parameter

Dynamic Neural Network

- Computations are adaptively modified for input samples

$$y_i = \mathcal{F}_{(x_i)}(x_i, \Theta_{(x_i)})$$

x_i : input sample $\mathcal{F}_{(x_i)}(\cdot, \Theta)$: CNN Function parameterized by Θ and x_i
 y_i : output sample $\Theta_{(x_i)}$: Learnable Parameter parameterized by x_i

Warning: These equations are not sound, they only give an intuition of what's happening.



Why Dynamic Neural Network?

- **Accuracy++**
Larger parameter space > Better “representation power”
- **Efficiency++**
Selective computations > Less redundant & useless computat°
- **Adaptiveness++**
Tradeoff between Accuracy & Efficiency

And also

- Compatibility with latest advance in static NN design.
- Generality of tasks: NLP, Signal, image, video
- Interpretability: May be improved...

What this paper is about: How do Dynamic NN work?

Instance-wise

« Per-Sample »
dynamic constructs in
neural network
architectures

Spatial-wise

Finer-granularity
intra-sample
dynamic constructs

Temporal-wise

Finer-granularity
inter-sample
dynamic constructs

~~Inference~~

~~Training~~

Not in this digest

What this paper is (unfortunately) NOT about:

- Expected gains/loss in terms of accuracy, complexity*
- Navigation map to select the most appropriate technique.

*: Possibly because this is a complex issue, cf. later discussion.

What this paper is about: 236 references!

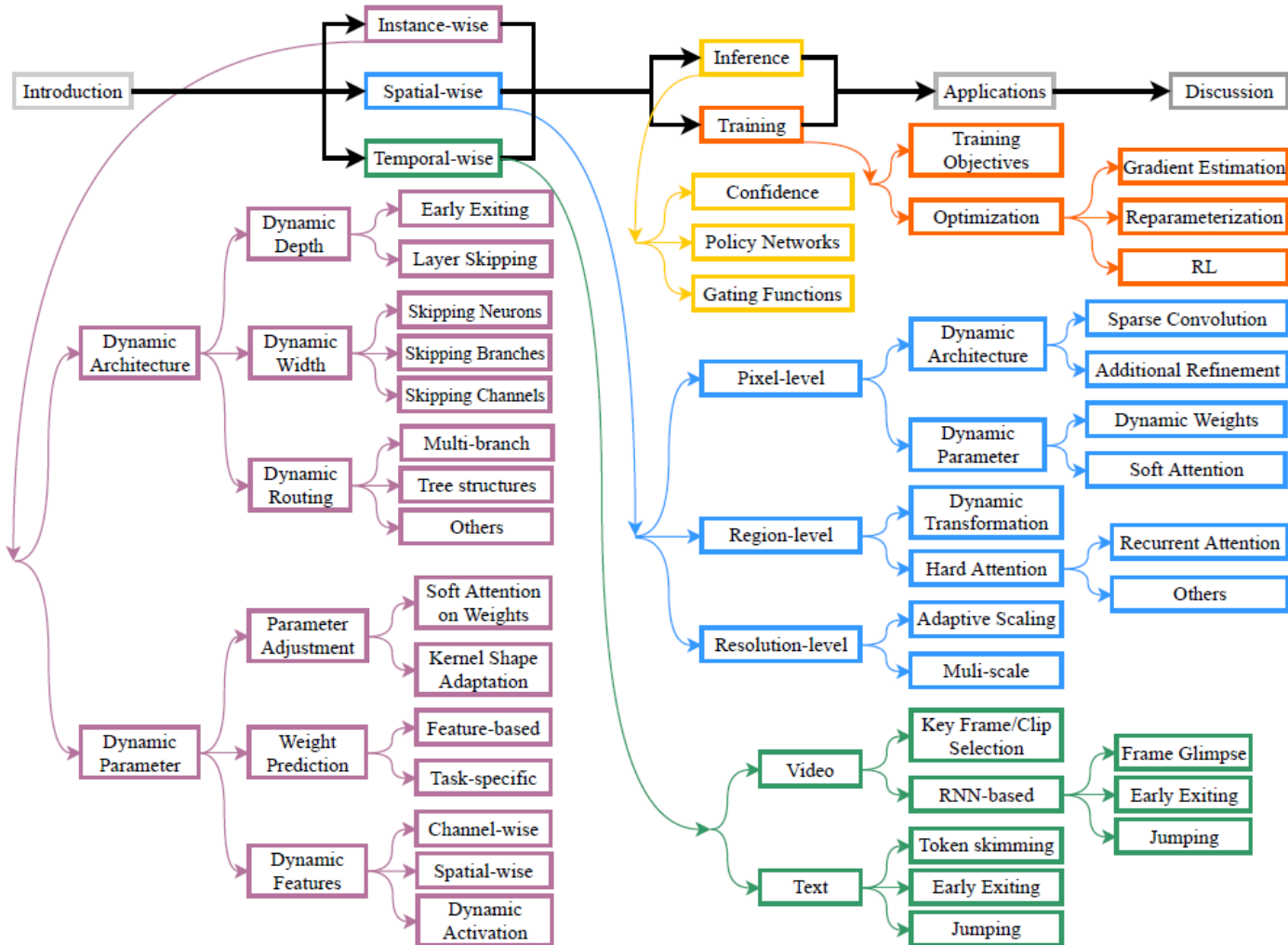


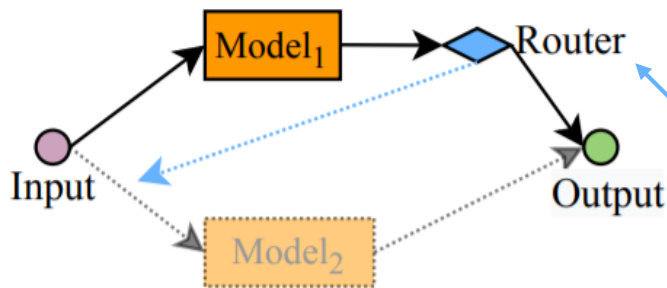
Fig. 1. Overview of the survey.

1. Dynamic Architectures

- Adapt data path to each sample by changing the network (1.1) depth or (1.2) width or by (1.3) routing data dynamically.

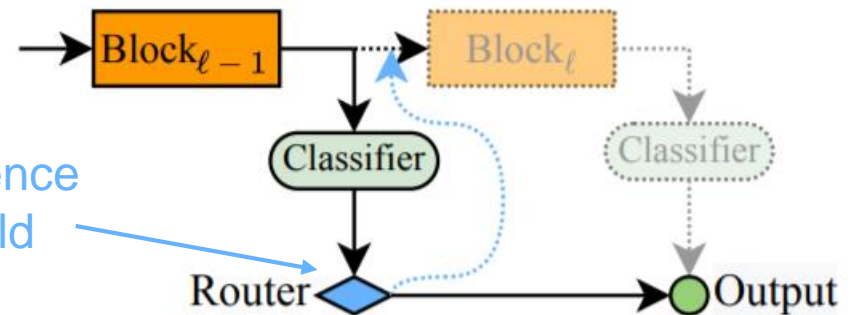
1.1. Dynamic Depth – 1.1.1 Early Exiting

- Stop computations early for “easy” samples.



a. Cascading networks

Confidence threshold



b. Intermediate classifier

☹ Redundancy between Model 1 and 2

☺ Less redundancy

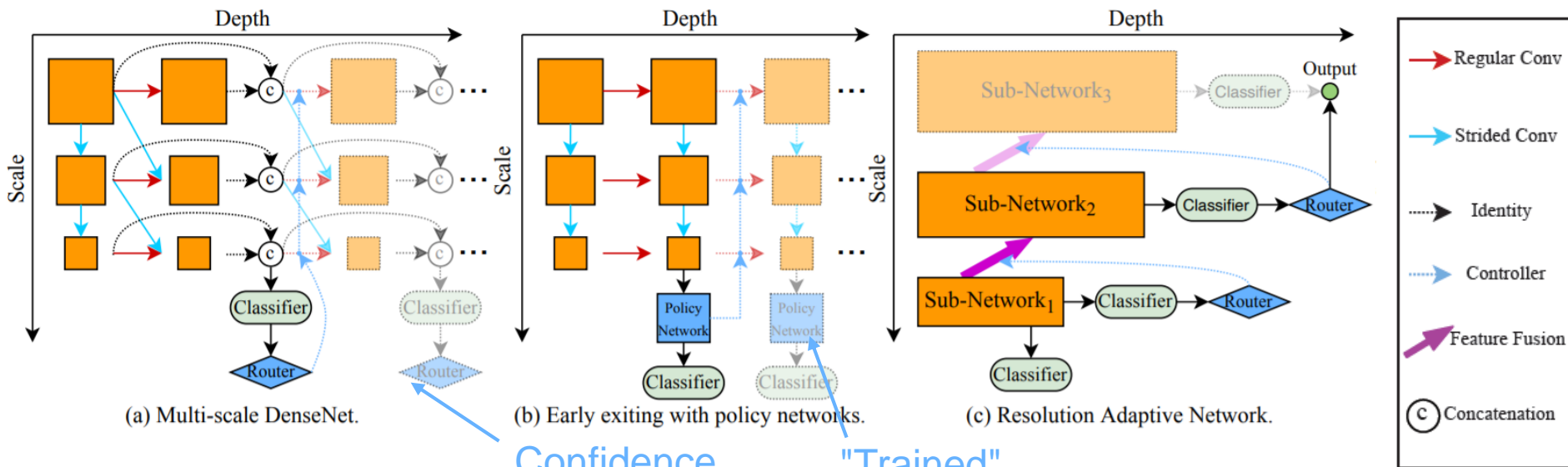
☹ Interference between classifiers at learning

☹ High res. features bad for classif

1. Dynamic Architectures

1.1. Dynamic Depth – 1.1.1 Early Exiting

- Stop computations early for “easy” samples.



c. Multi-scale networks

Confidence
threshold

"Trained"

😊 Better than both cascading and intermediate classif.

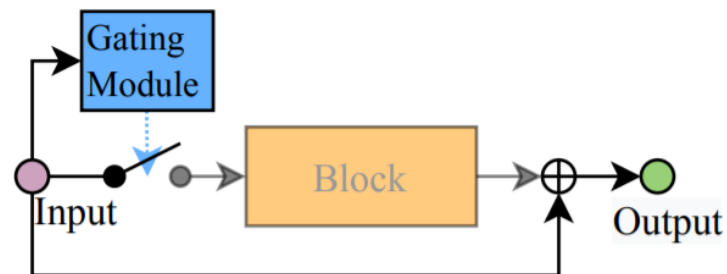
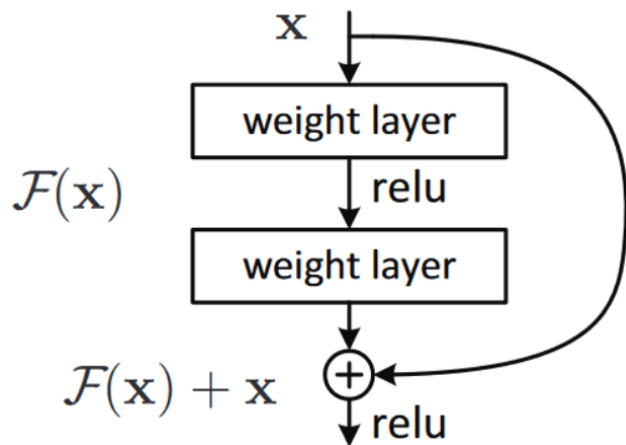
☹️ Touchy threshold tuning / training

1. Dynamic Architectures

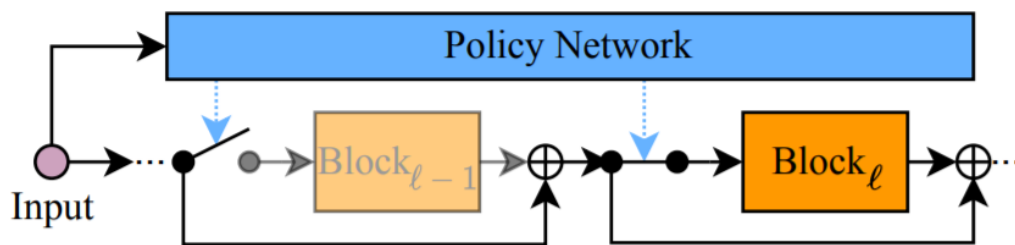
1.1. Dynamic Depth – 1.1.2 Layer Skipping

- Disable intermediate layers (requires “skip” connections)

Reminder:
Skip connection



a. Gating function for skipping a layer.
(also exists for dynamic bitwidth)

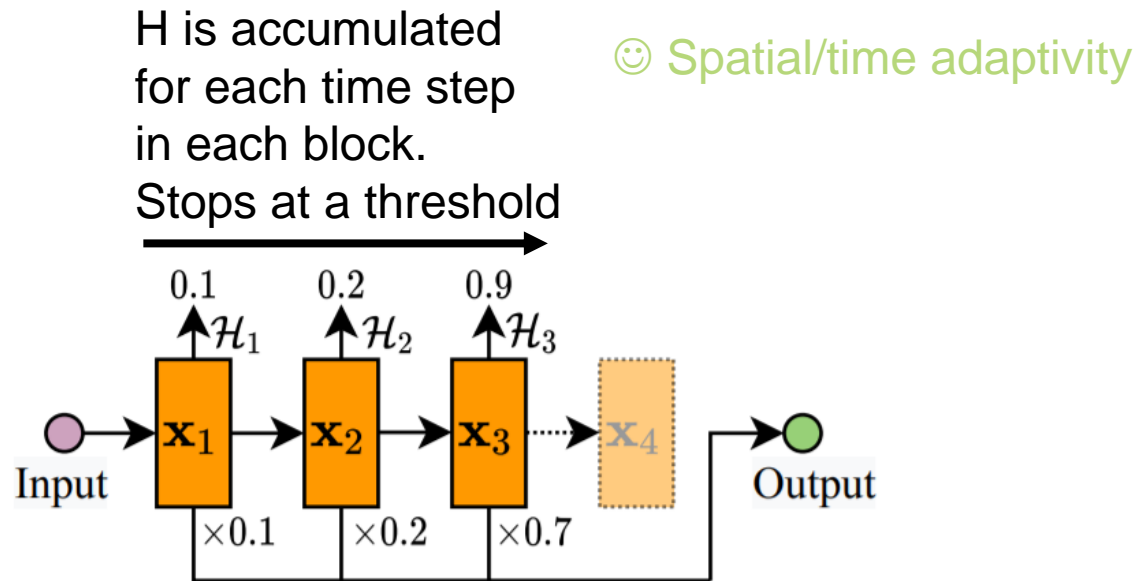


b. ML controlling more globally the activated layers

1. Dynamic Architectures

1.1. Dynamic Depth – 1.1.2 Layer Skipping

- Disable intermediate layers (requires “skip” connections)



c. Halting scores in a RNN stage

1. Dynamic Architectures

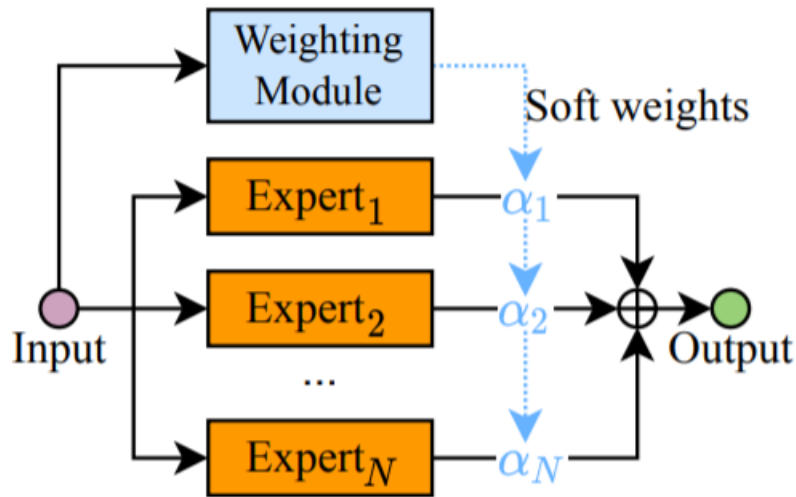
1.2. Dynamic Width – 1.2.1 of Fully connected layer

- Vague in the paper... (individual neuron activation, low rank approx.°)

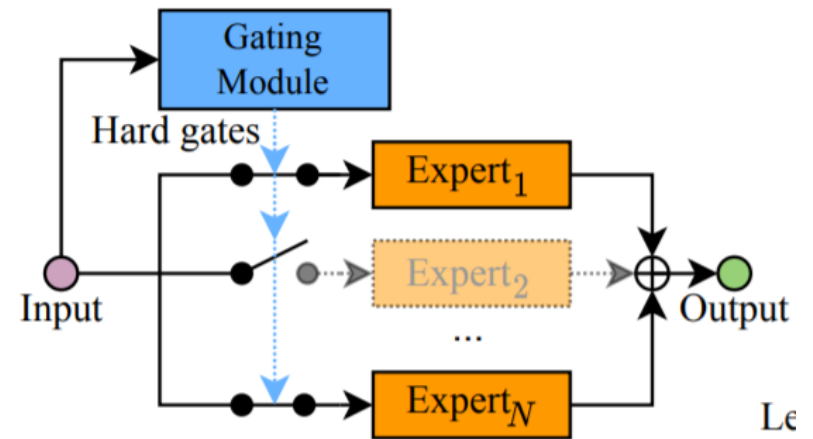
1. Dynamic Architectures

1.2. Dynamic Width – 1.2.2 Mixture of Experts

- “Expert” sub-networks run in parallel and fused.



a. Soft fusion



b. Hard gating

☺ Accuracy++
 ☹ Computations--

☺ Accuracy+
 ☺ Computations++

1. Dynamic Architectures

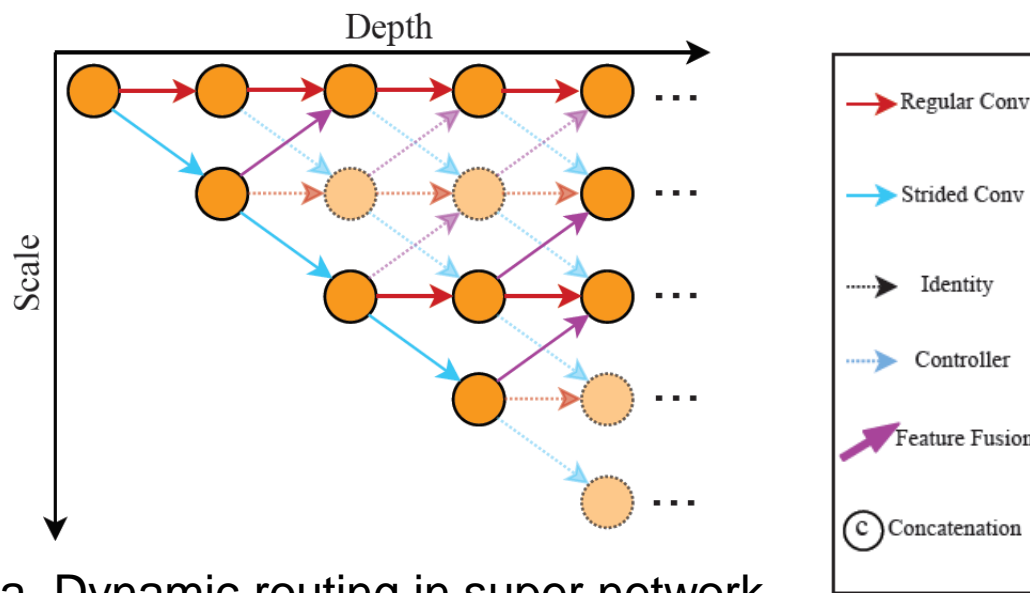
1.2. Dynamic Width – 1.2.3 Dynamic channel pruning

- Selective deactivation of channel during computations.
- a) Multi-stage archi
Similar to early exiting: activate secondary channels when confidence criteria is not met.
 - ☺ Adaptive accuracy
 - ☺ Adaptive computations
 - ☹ Highy variable latency
(full inference needed when confidence criteria is not met)
- b) Dynamic pruning with gating function
Deactivate channels individually (e.g. using markov decision process)
 - ☺ Adaptive accuracy (but less than mult stage)
 - ☺ Adaptive computations

1. Dynamic Architectures

1.3. Dynamic Routing

- Dynamic data path in a “super-network”
 - Path selection: unique branch path never merged/fused
 - Neural tree: multiple branch paths never fused (i.e. multiple decisions)
 - Other



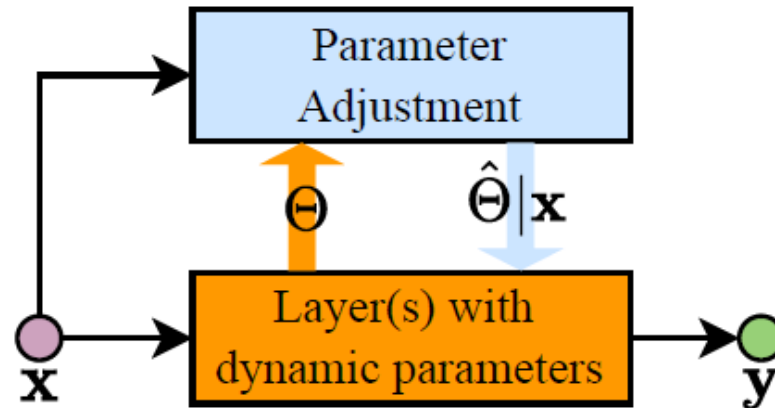
a. Dynamic routing in super network

😊 Adaptability++
 😊 Accuracy++
 😞 Complexity++

2. Dynamic Parameters

- Fixed Network Architecture
- Dynamically change model parameters for each samples, by
 - (2.1) Adjusting parameters,
 - (2.2) Generating parameters, or
 - (2.3) Rescaling features with soft attention.

2.1. Parameter Adjustment



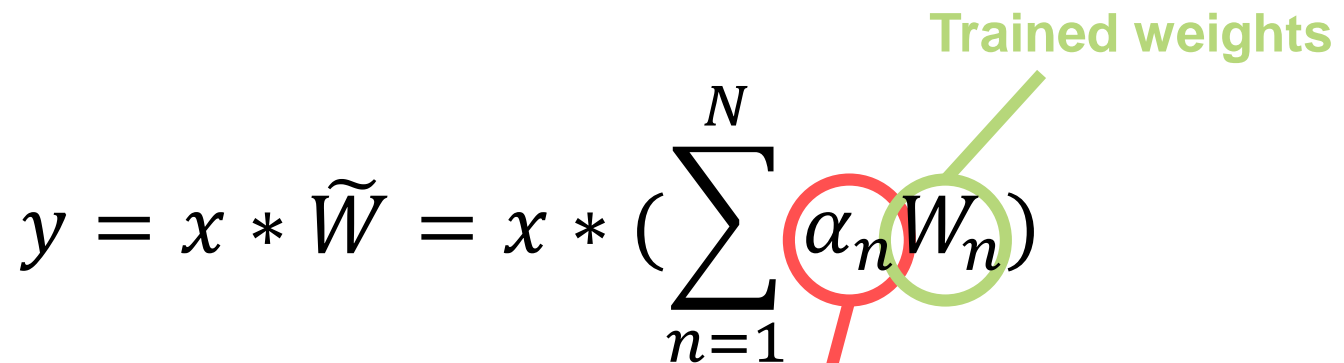
2. Dynamic Parameters

2.1. Parameter Adjustment - 2.1.1 Attention on weight

- Dynamically weight the different parameters before applying the convolution.

$$y = x * \tilde{W} = x * \left(\sum_{n=1}^N \alpha_n W_n \right)$$

Trained weights



- ☺ Model Capacity++
- ☺ Computations--
(compared to mixture of experts)
- ☹ Model size--

Depends on x
or even on « local » features
within x (for segmentation-aware CNN)

2. Dynamic Parameters

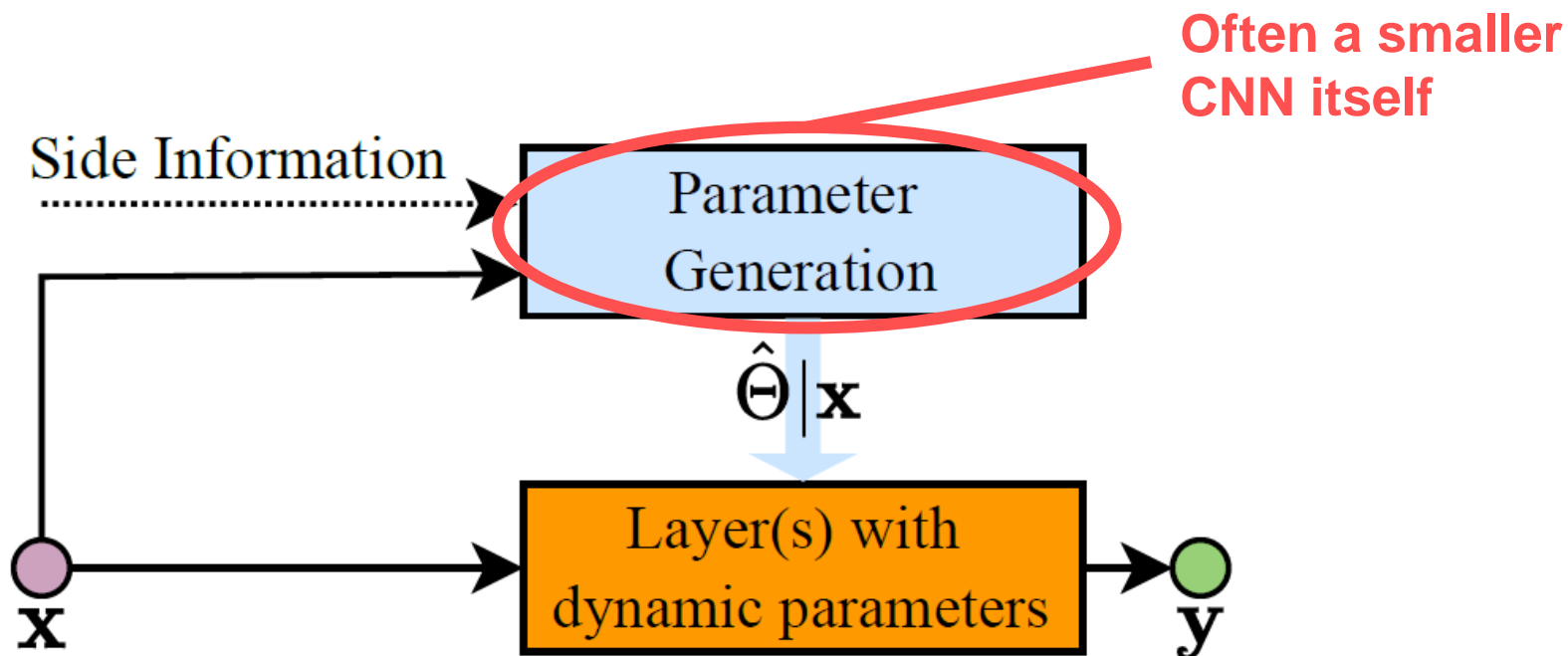
2.1. Parameter Adjustment - 2.1.1 Kernel Shape

- Reshape convolution Kernel and achieve “dynamic reception of fields”
- A bit vague in the paper...

2. Dynamic Parameters

2.2. Weight prediction

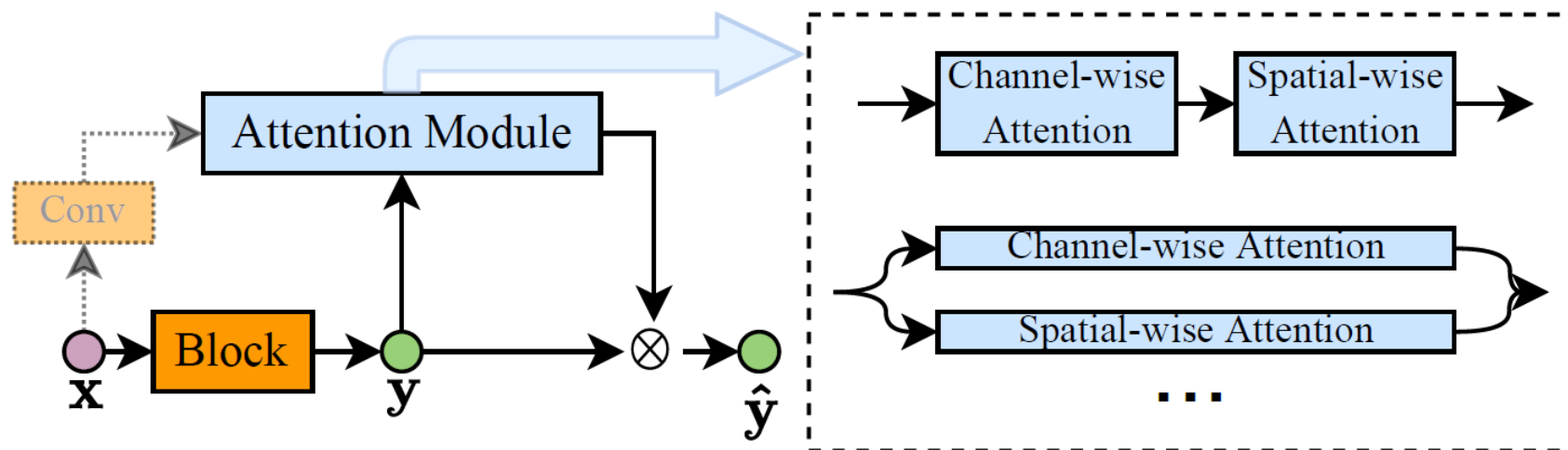
- Instance-wise generation of the parameters



2. Dynamic Parameters

2.3. Dynamic Features – 2.3.1/2 Channel/Spatial-Wise

- Works on the convolution output, instead of convolution parameters themselves.



- ☺ « Simpler » than modifying weights
- ☺ Mathematically equivalent
- ☹ **Computations+**
(soft attention weights existing computations)

2. Dynamic Parameters

2.3. Dynamic Features – 2.3.3 Dynamic Activation

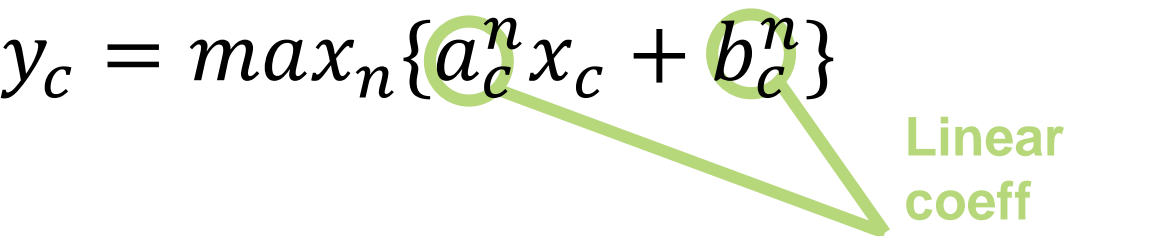
- Change the activation function
(instead of applying attention mechanisms before it)

Classic ReLU $y_c = \max(x_c, 0)$



Channel
index

Dynamic ReLU $y_c = \max_n \{ a_c^n x_c + b_c^n \}$



Linear
coeff
computed
from x

- ☺ Simple modification of CNN
- ☺ Good results for vision

Back to the beginning

What this paper is about: How do Dynamic NN work?

Instance-wise

« Per-Sample »
dynamic constructs in
neural network
architectures



Depth

Width

Param.

Spatial-wise

Finer-granularity
intra-sample
dynamic constructs



Pixel-wise

Region-level

Resolution-level

Temporal-wise

Finer-granularity
inter-sample
dynamic constructs

☺ Computations--
☹ Not GPU friendly

(hard/soft attention on salient regions)

(Dynamic multi-scale witchcraft)

Back to the beginning

What this paper is about: How do Dynamic NN work?

Instance-wise

« Per-Sample »
dynamic constructs in
neural network
architectures



Depth
Width
Param.

Spatial-wise

Finer-granularity
intra-sample
dynamic constructs



Pixel-wise
Region-level
Resolution-level

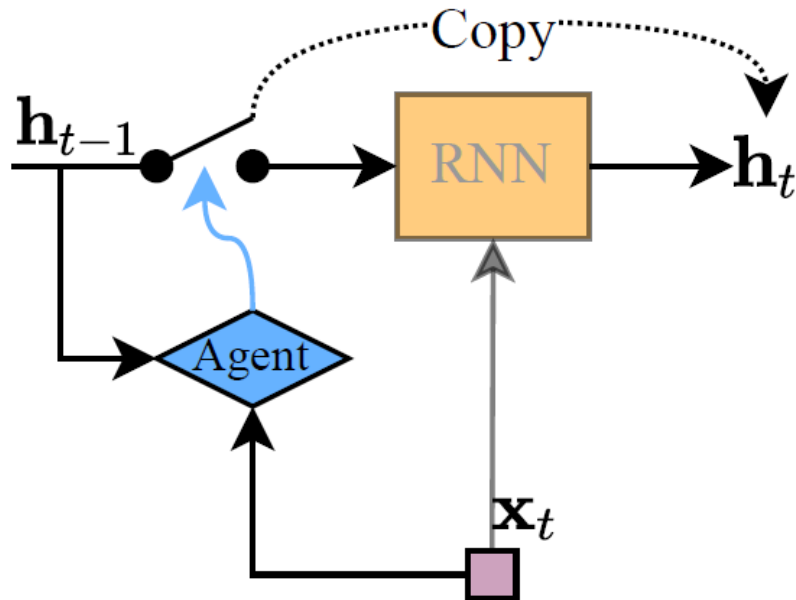
Temporal-wise

Finer-granularity
inter-sample
dynamic constructs

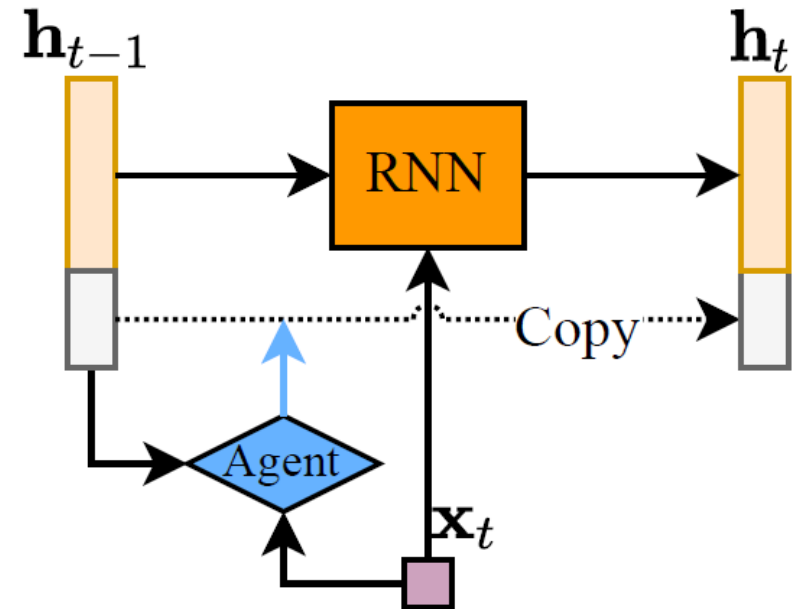


RNNs for text
RNNs for video

1. Dynamic Update of Hidden State

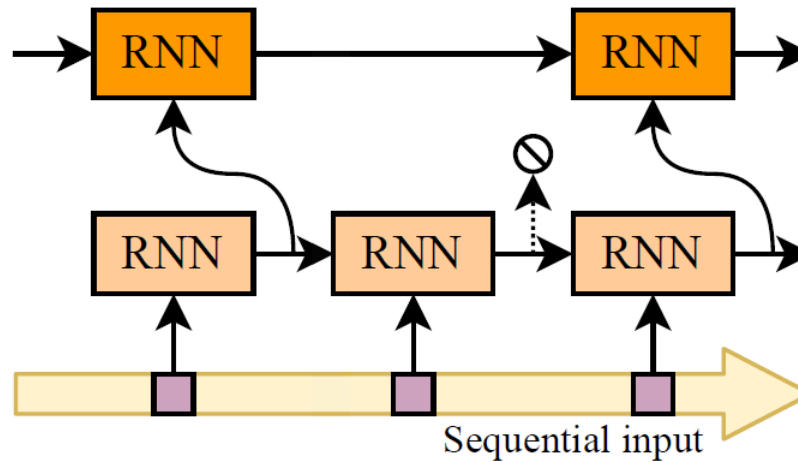


(a) Skip update of hidden state.

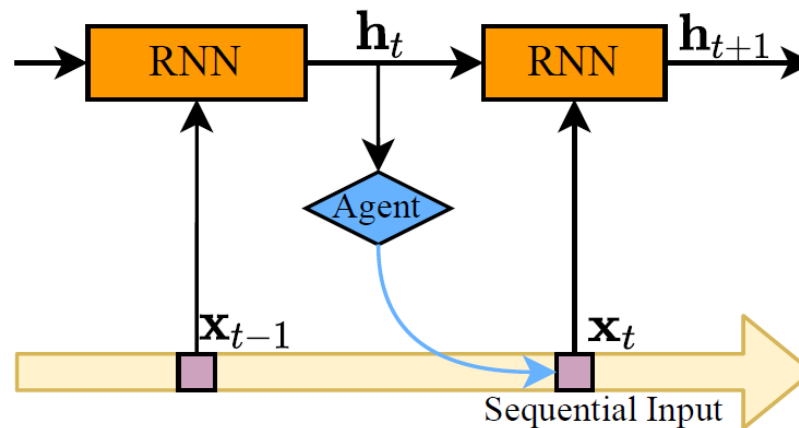


(b) Partial update of hidden state.

2. Multi-scale RNNs



3. Temporal Dynamic RNNs



Applications

TABLE 3

Applications of Dynamic Networks. For the type column, I, S, T stand for instance-wise, spatial-wise and temporal-wise respectively.

Fields	Data	Type	Subfields & references
Computer Vision	Image	I	Object detection (face [38], [181], [182], facial point [183], pedestrian [184], general [31], [185], [186], [187], [188]) Image segmentation [101], [189], Super resolution [190], Style transfer [191], Coarse-to-fine classification [192]
		I & S	Image segmentation [32], [119], [136], [138], [140], [144], [146], [193], [194], [195], [196], [197], Image-to-image translation [198], Object detection [105], [106], [137], [138], [153], Semantic image synthesis [199], [200], [201], Image denoising [202], Fine-grained classification [148], [150], [203], [204] Eye tracking [148], Super resolution [141], [143], [205]
		I & S & T	General classification [37], [149], [152], Multi-object classification [206], [207], Fine-grained classification [151]
	Video	I	Multi-task learning (human action recognition and frame prediction) [208]
		I & T	Classification (action recognition) [56], [166], [170], [172], [173], [175], [176], [177], [209], Semantic segmentation [210] Video face recognition [20], [171], Action detection [168], [169], Action spotting [167], [174]
		I & S & T	Frame interpolation [211], [212], Video super resolution [213], Video deblurring [214], [215], Action prediction [216]
Point Cloud	I & S	3D Shape classification and segmentation, 3D scene segmentation [217], 3D semantic scene completion [218]	
Natural Language Processing	Text	I	Neural language inference, Text classification, Paraphrase similarity matching, and Sentiment analysis [54], [55]
		I & T	Language modeling [11], [16], [111], [160], [162], Machine translation [16], [33], [34], Classification [59], [60], [164], Sentiment analysis [156], [158], [159], [161], [165], Question answering [33], [58], [158], [161], [163]
Cross-Field	Image & Text	I & S & T	Image captioning [120], [219], visual question answering [220]
Others	Document classification [146], Link prediction [221], Graph classification [113], Stereo confidence estimation [222], Recommendation systems [223]		

Open issues for research

- **Theoretical background for dynamic archi.**
- **Archi. Design for Dynamic Network**
(Most of dynamic network are currently derived from static archi.)
- **Applicability to more diverse tasks.**
(Most techniques tested on classif only...)
- **Efficient HW implementation (!!!)**
Computational gains do not translate well into perf. gains. (GPU ☹)
- **Robustness against Adversarial Attacks (!!!)**
- **Interpretability**

Thanks for enduring!