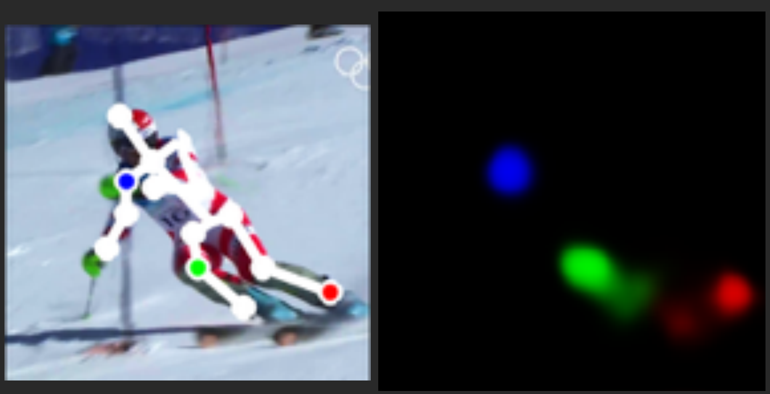
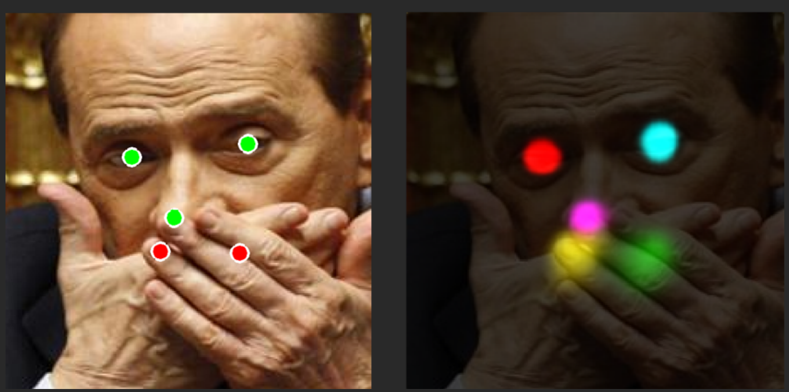
Augmenting Feedforward Models with Top-Down Feedback

Deva Ramanan
CMU



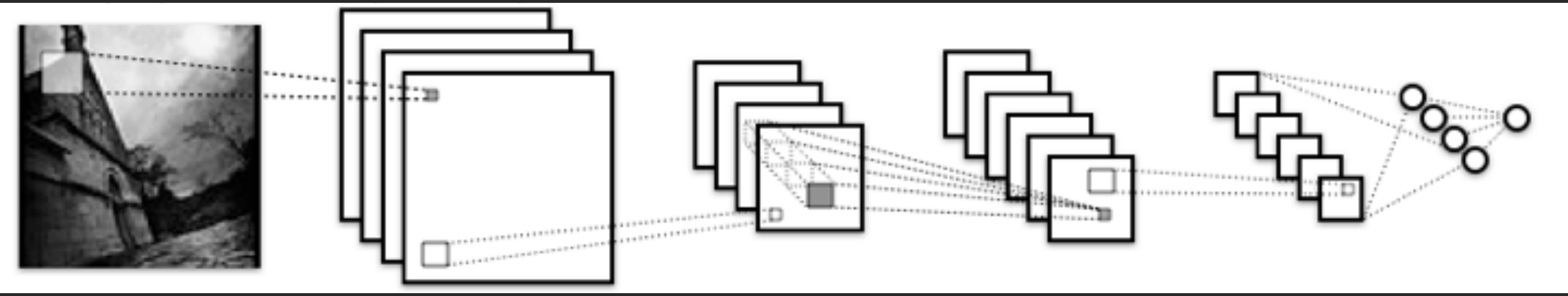


First author

Stole his talk

Peiyun Hu

Contemporary vision

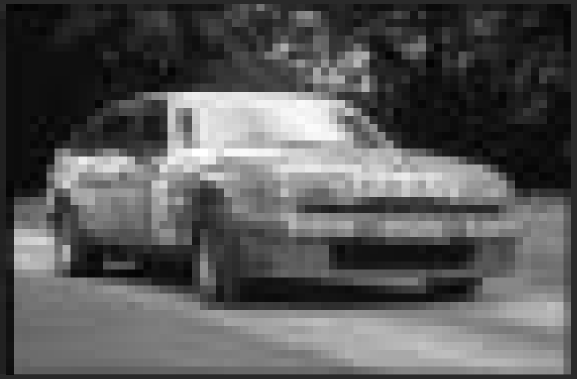


The impact of feedforward hierarchies has been undeniable

Some (of my personal) inspiration from human vision

(see Bruno's fanstastic talk for a proper description)

Some inspiration from human vision

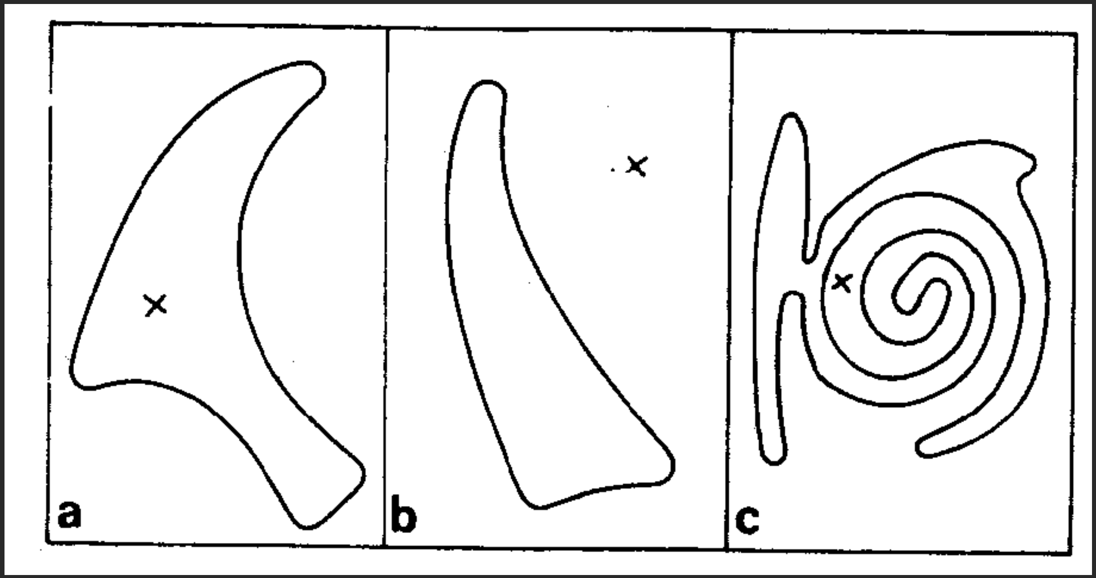


People can distinguish high-level concepts (animal/transport) in under 150ms (Thorpe)

Appears to suggest feed-forward computations suffice (or at least dominate)

Task-driven feedback

"Is 'X' inside the closed curve?"



50 ms 50 ms 500 ms

"Visual routines" Ullman 84

Some tasks appear to require purposeful examination

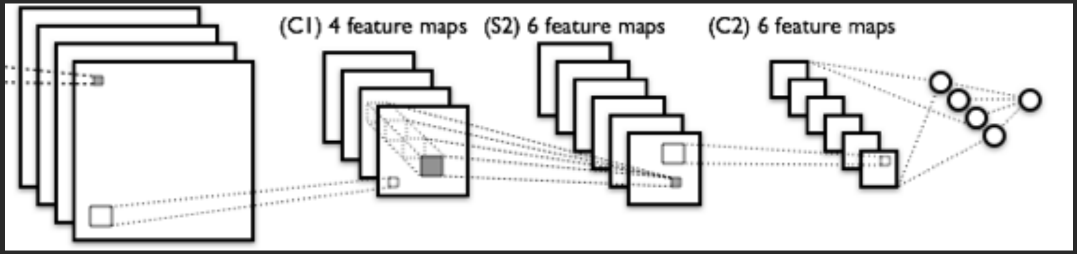
Task-driven feedback

Relation to visual question answering (pointed out by Russakovsky)

"How many slices of pizza are there?"

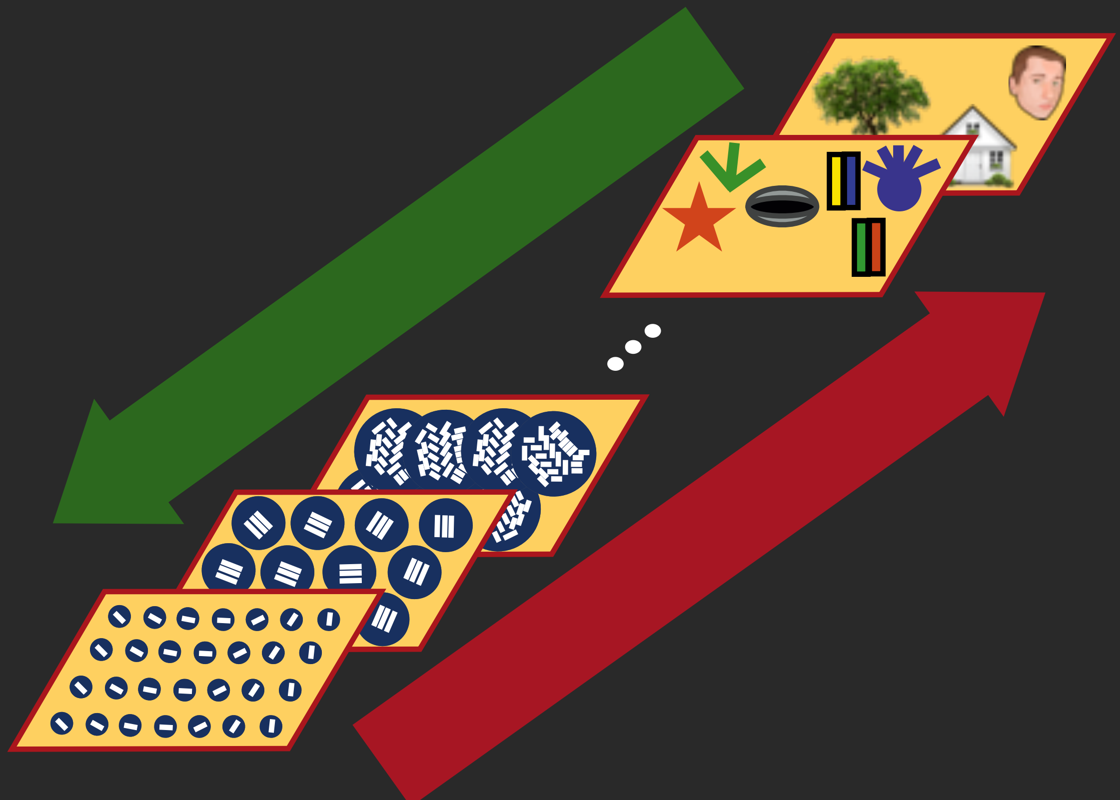
Reparse image with the knowledge that it contains a pizza

Pizza is present



A categorization of tasks

Hochstein & Ahissar 02



Vision at a glance (feedforward)

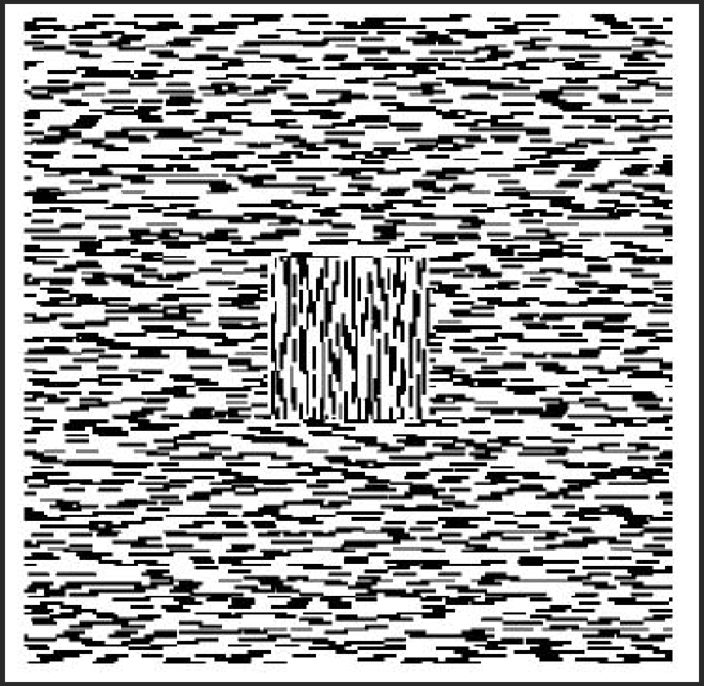
Rapid scene categorization

Vision with scrutiny (+feedback)

Fine-grained recognition

Spatial localization for manipulation

Feedback can occur quickly



V1 neurons tuned for vertical edges respond to figure-ground boundary after 50 ms
(seemingly after V2 activates)

Lee & Mumford 98, 03

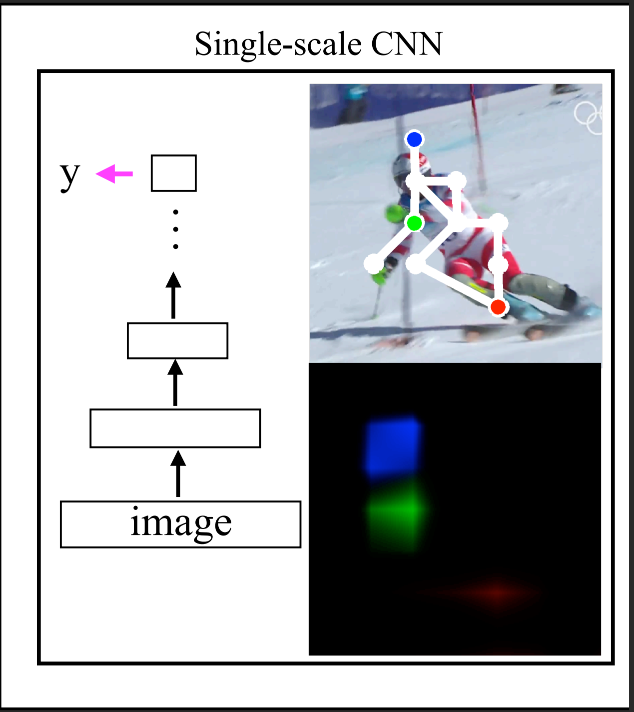
Some take-aways from human vision/neuroscience

1. Some visual tasks will benefit more from feedback

2. Feedback can happen quite quickly

3. Feedback is not limited to top-down attention

Preview of results



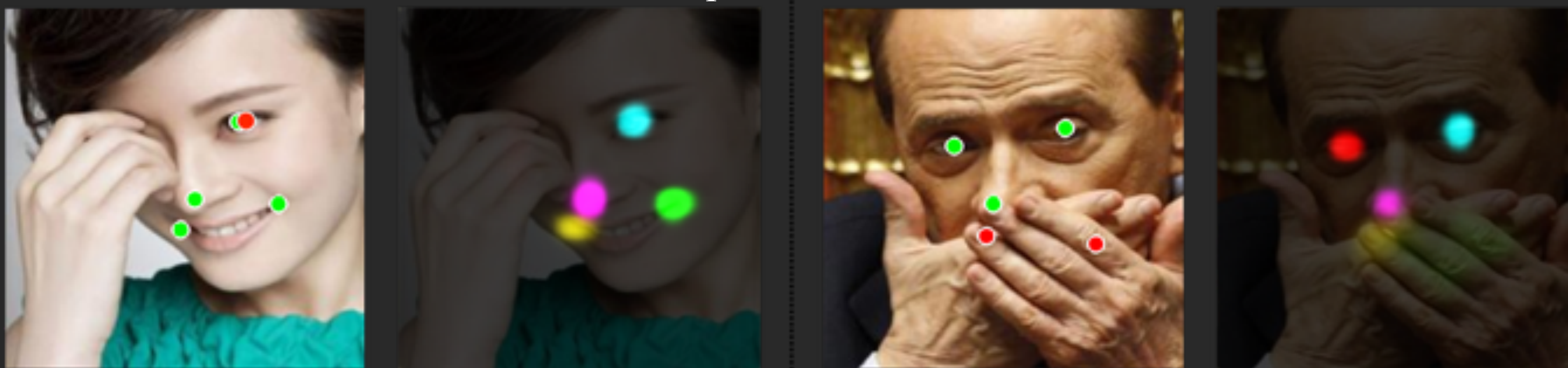
VGG

Preview of results

Max location

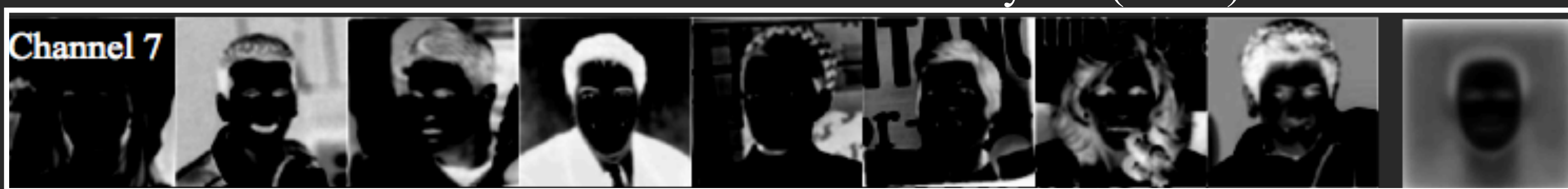
Predicted heatmap

Bottom-up



Preview of results

Feedforward activations from layer 1 (~1ms)

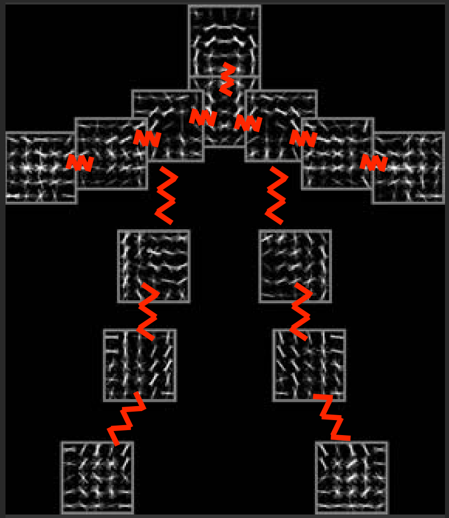


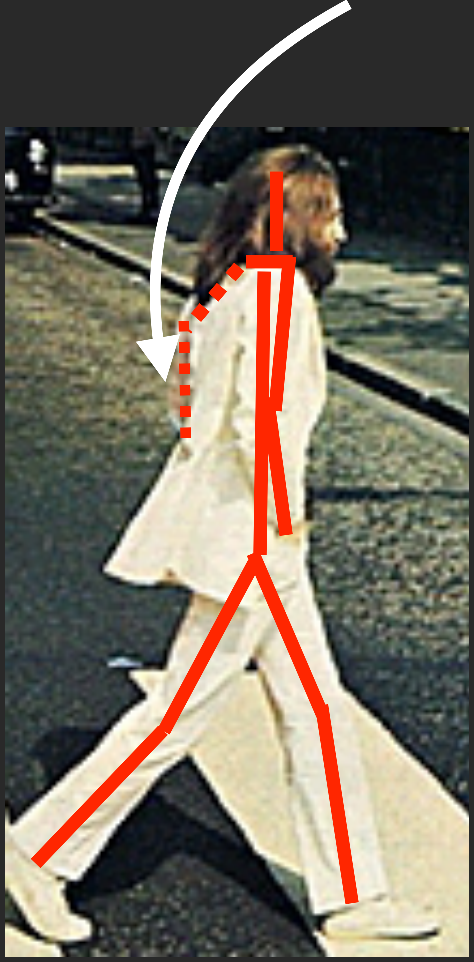
avg

Activations after feedback (~40ms)

Aside: probabilistic models already do this

Message-passing on "parts + structure models" naturally make use of top-down feedback

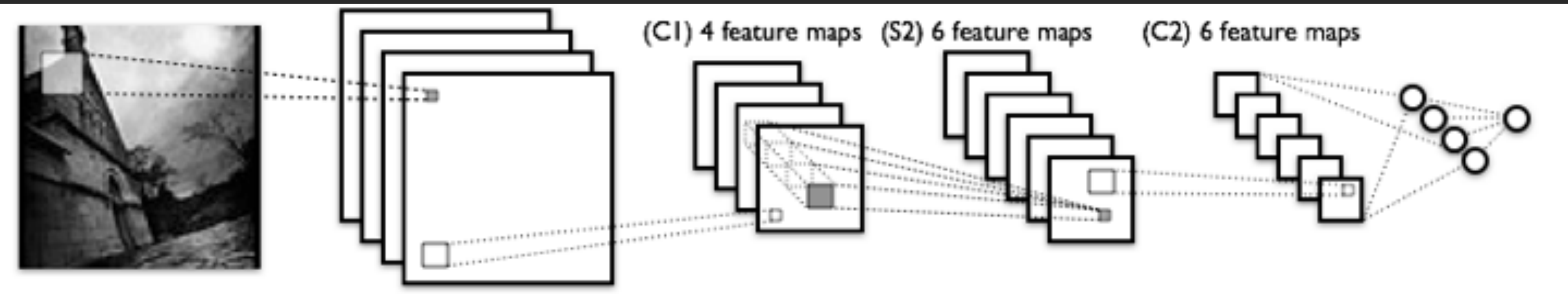




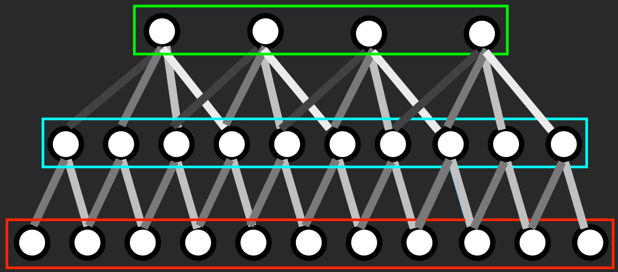
Confession: this was our original motivation

So how do we add feedback to deep models?

CNNs

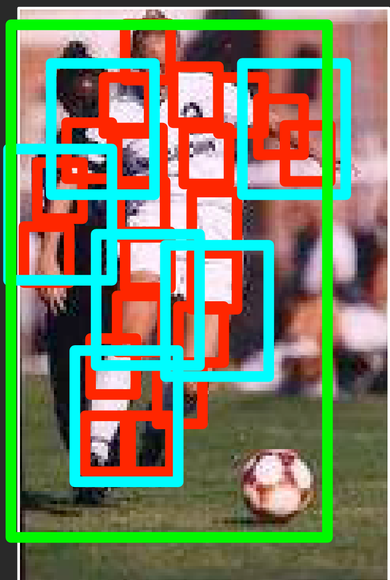


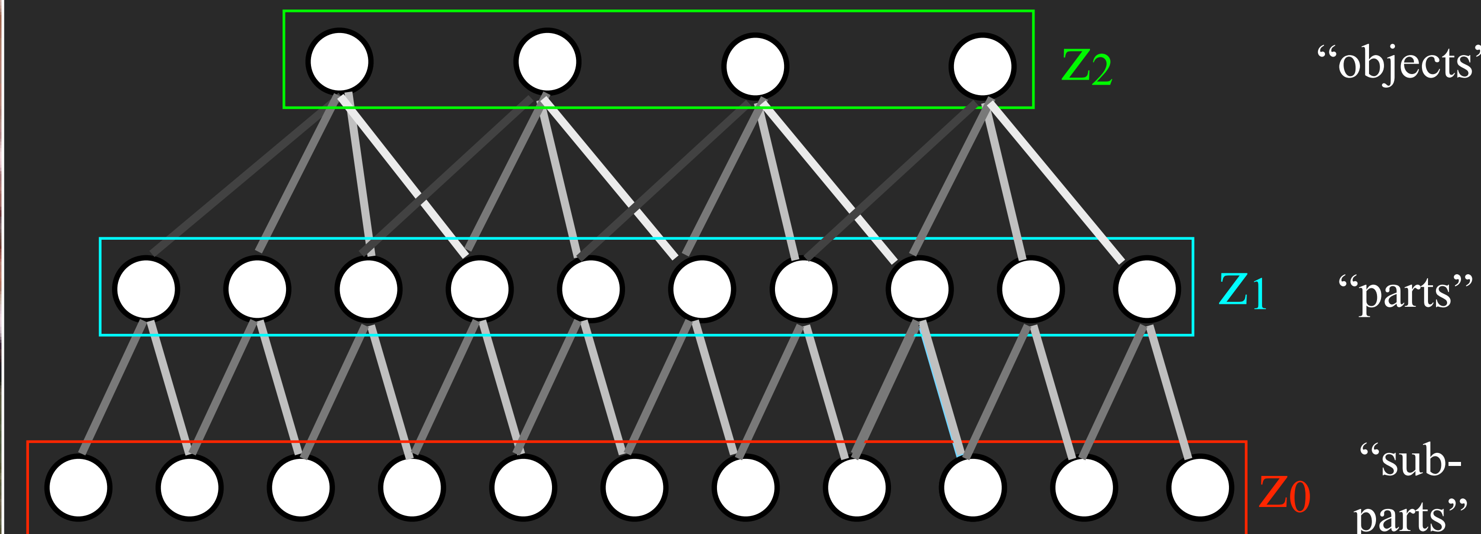
Boltzmann machines



(Convolutional) Boltzmann machines as deep latent-variable models

Salakhutdinov & Hinton 09
Le et al 09

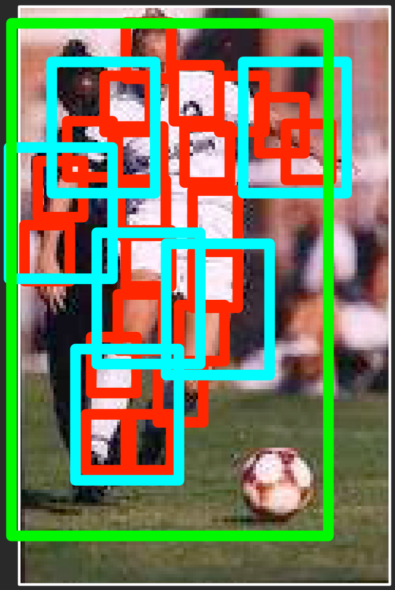


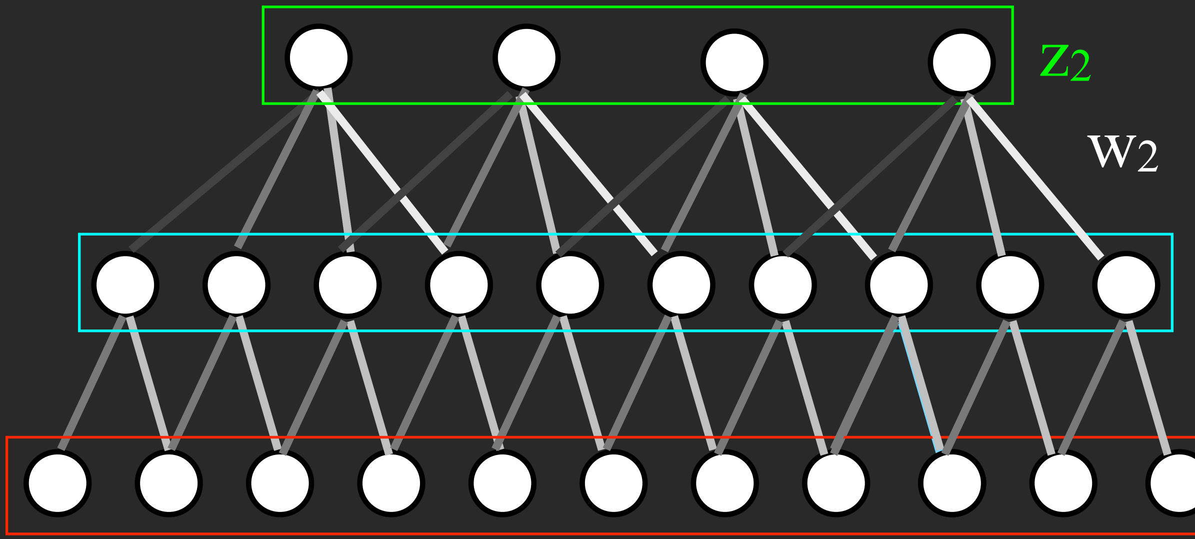


Binary latent variables: is there a (person, head, oriented edge) at a particular location?

$$P(z) \propto e^{S(z)} \quad \text{where} \quad S(z) = \frac{1}{2} z^T W z + b^T z$$

(Convolutional) Boltzmann machines as deep latent-variable models





Gibbs sampling:

$$z_i[u] \sim \text{sigmoid}(b_i + top_i[u] + bot_i[u])$$

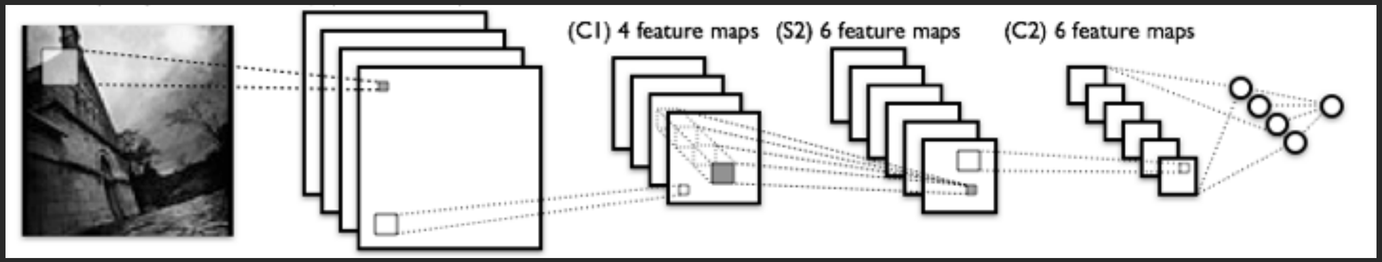
$$bot_i[u] = \sum_v w_i[v] z_{i-1}[u+v] \qquad \text{“convolution”} \quad w_1$$

$$top_i[u] = \sum_v w_{i+1}[v] z_{i+1}[u-v] \qquad \text{“deconvolution”} \quad w^T{}_2$$

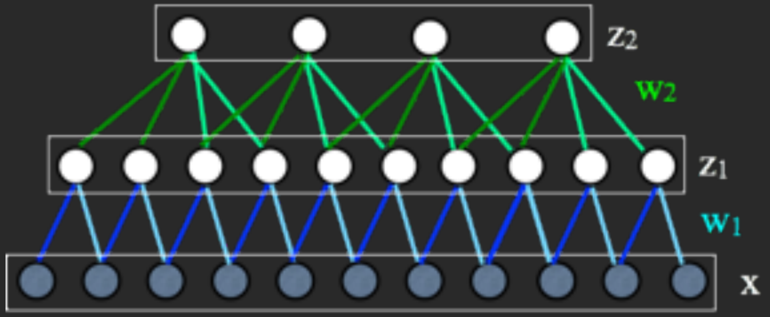
Arm detection *should* depend on low-level sub-parts and high-level objects found nearby

So why have practical results been dominated by CNNs?

CNNs



Boltzmann machines



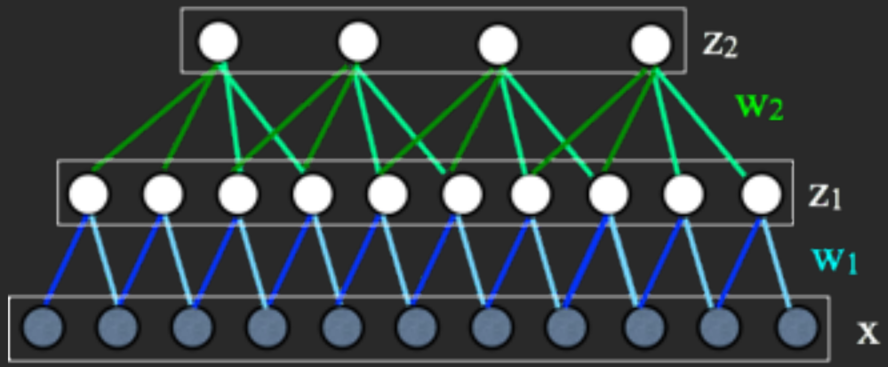
It seems that efficient inference (parallel computation) and learning (backprop) are key

Solution

Choose an alternative inference strategy that is more amenable to backprop: variational inference

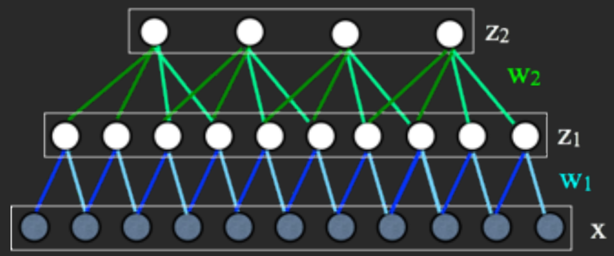
Mean-feild updates (Salakhutdinov & Hinton, Jorden et al, Jain, etc):

$$z_i[u] = \text{sigmoid}(b_i + bot_i[u] + top_i[u])$$



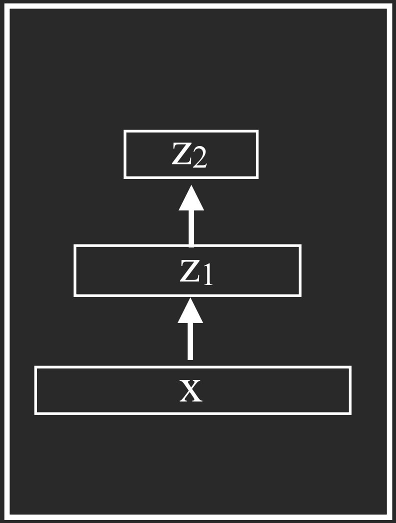
Implement sequence of inference updates with a neural net

cf. past work on "unrolling inference": Chen et al 15, Zheng et al 15, Goodfellow et al 13

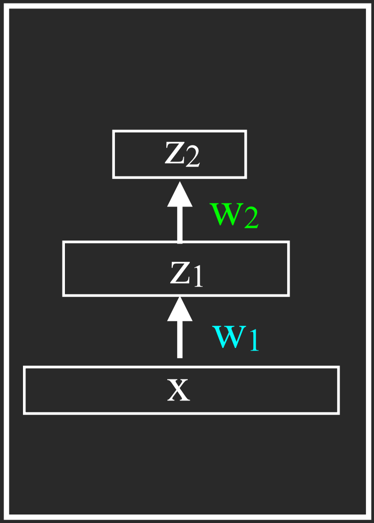


$$z_i[u] = \text{sigmoid}(b_i + bot_i[u] + top_i[u])$$

Bottom-up layerwise updates

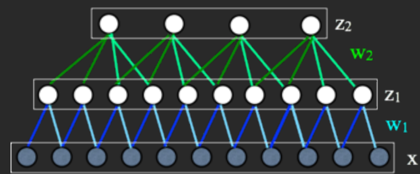


Feedforward CNN

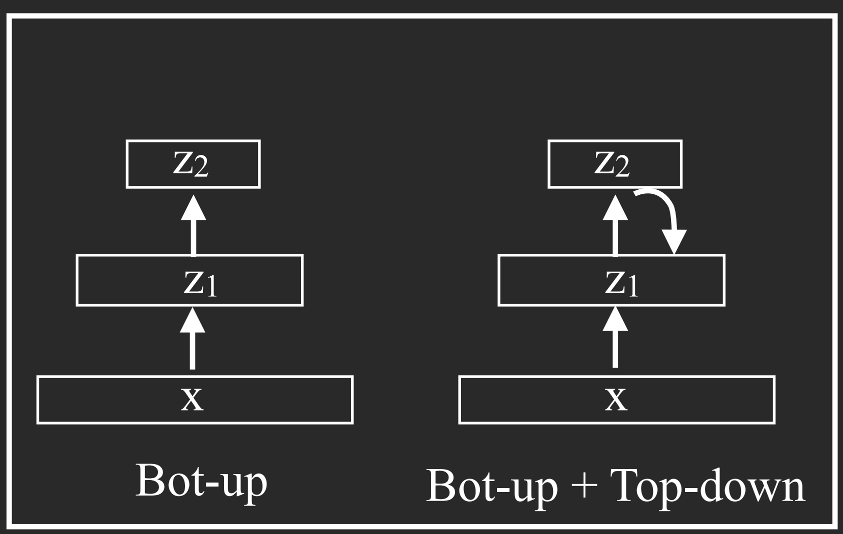


Use CNNS to learn to infer on Boltzmann machines

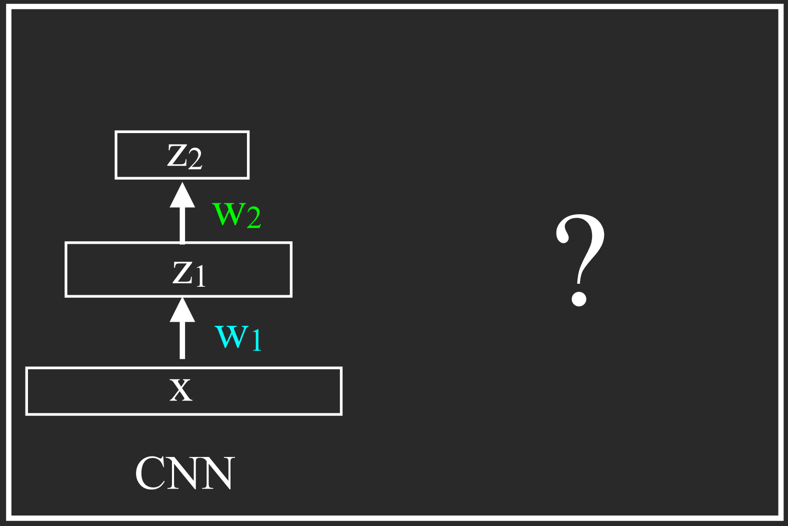
1. Use variational inference rather than Gibbs sampling (Salakhutdinov & Hinton)
2. Unroll sequence of mean-field updates into a neural net (Goodfellow et al)



Layerwise updates

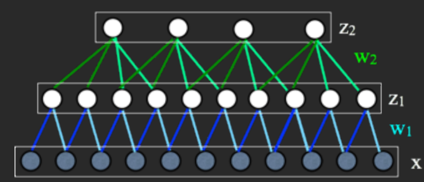


Neural net implementation

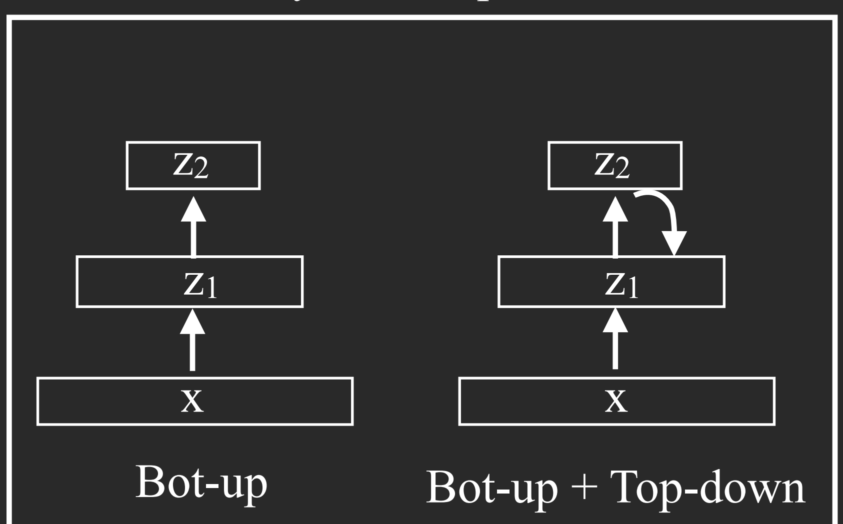


Use CNNS to learn to infer on Boltzmann machines

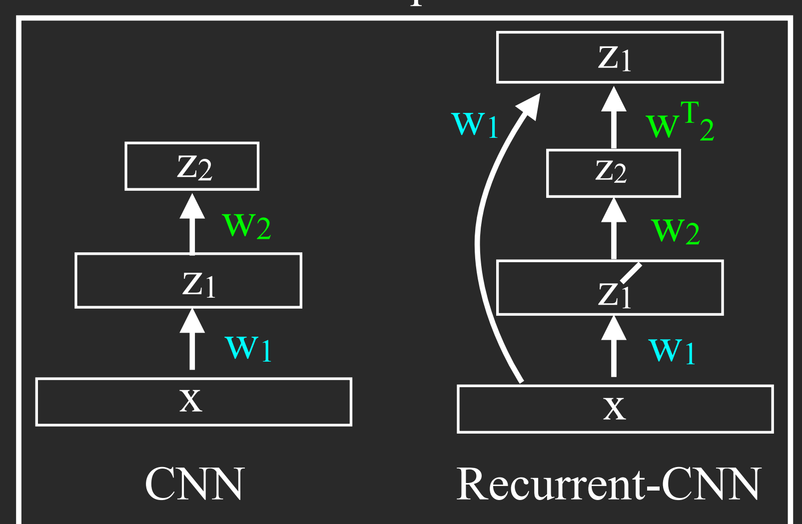
1. Use variational inference rather than Gibbs sampling (Salakhutdinov & Hinton)
2. Unroll sequence of mean-field updates into a recurrent neural net (Goodfellow et al)



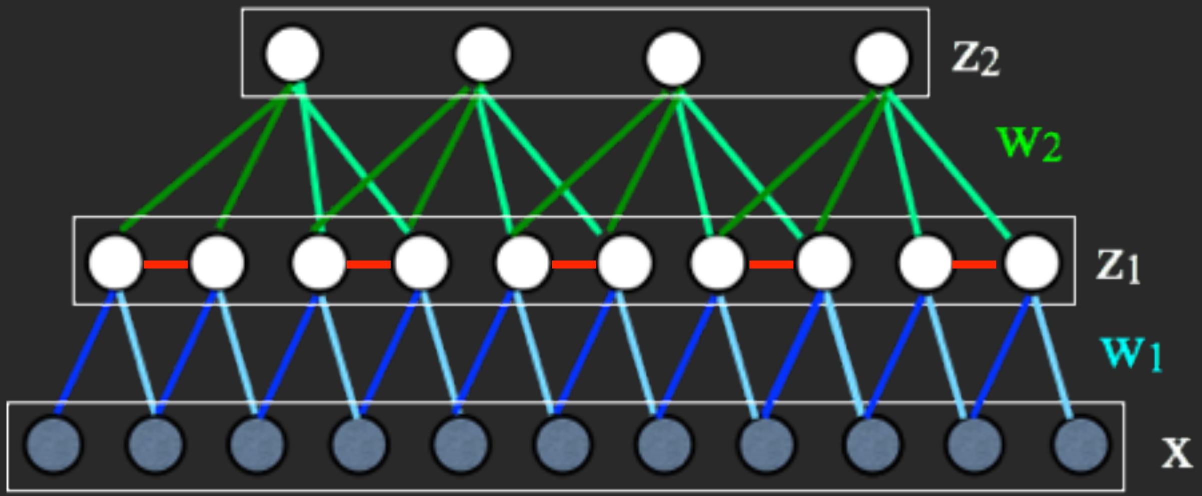
Layerwise updates



Neural net implementation

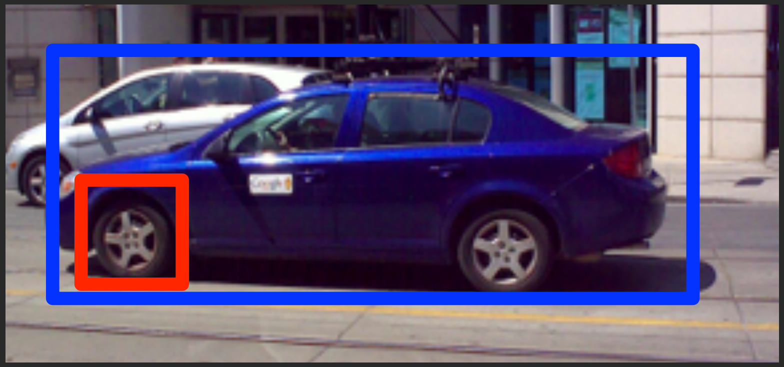


Top-down localization



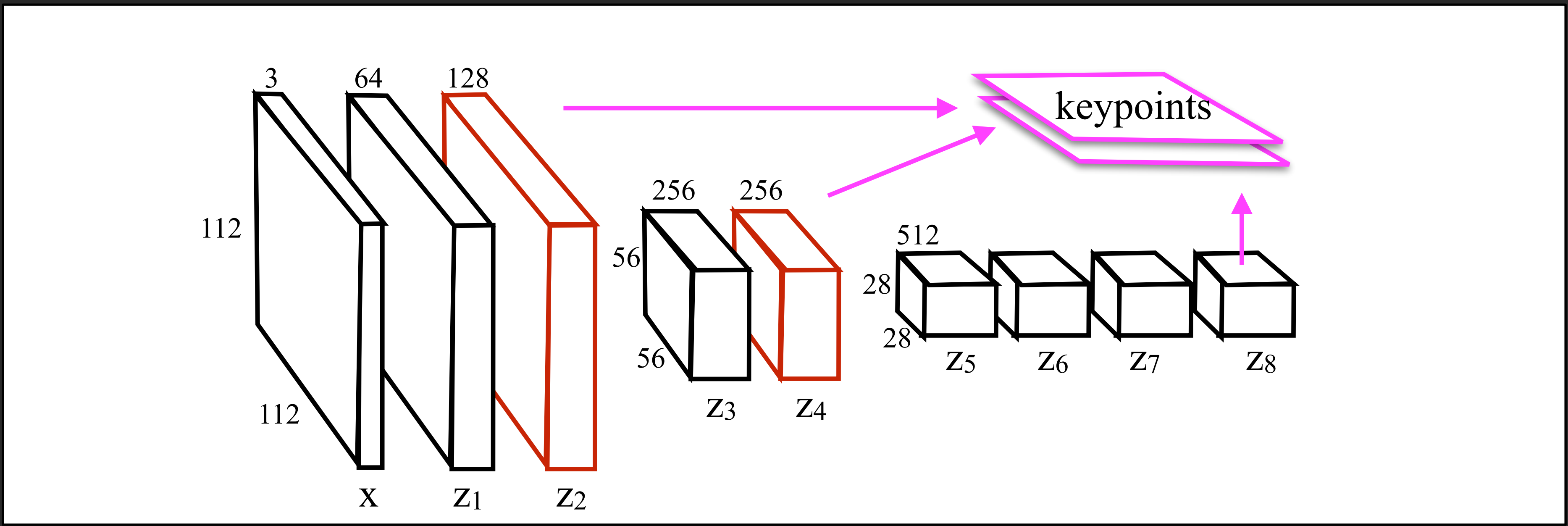
1. Model "max-pooling" using lateral inhibition connections (red edges)

2. Above model allows for top-down localization
   e.g., a car "object" can influence the activation and location of a wheel "part"



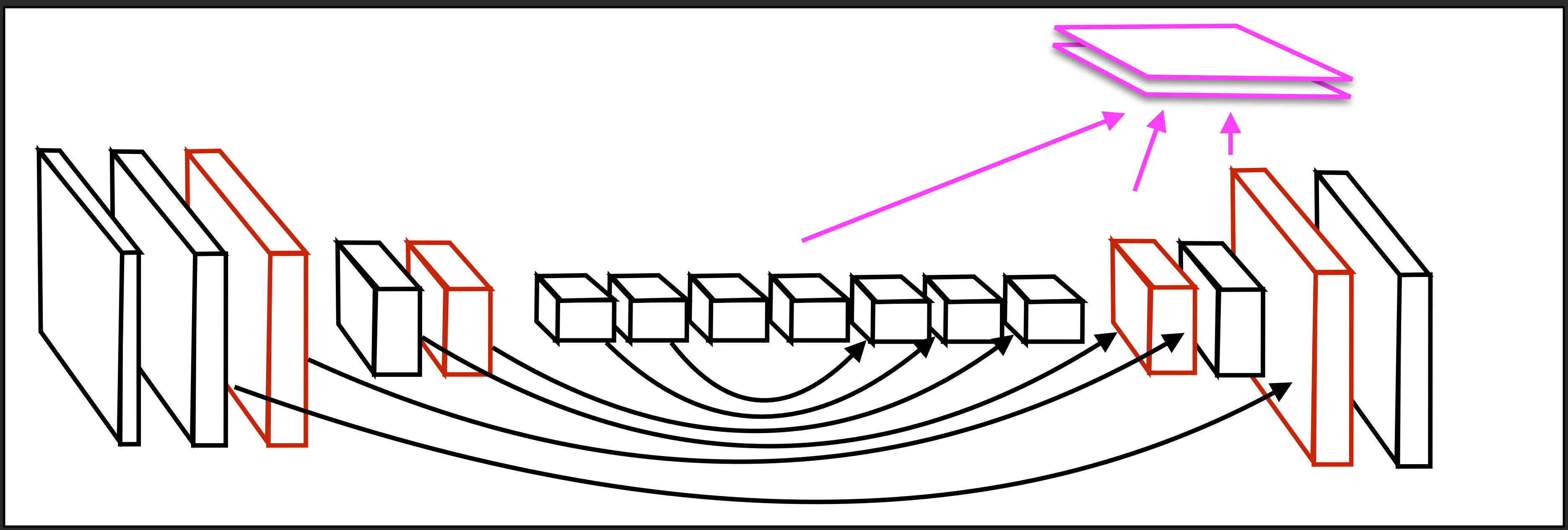
Train unrolled model with backprop

Bottom-up pass



Train unrolled model with backprop

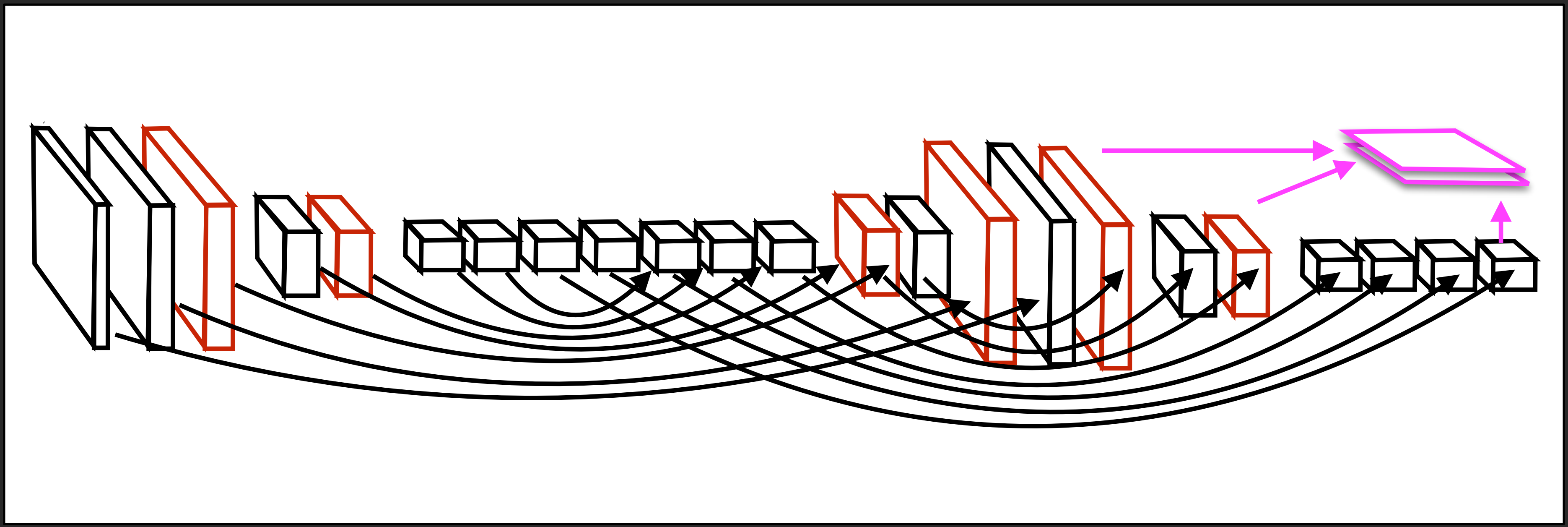
Bottom-up + top-down pass



(cf similar architectures: Autoencoders, DeConvNets,
U-Nets, Hourglass Nets, Ladder Networks)

Train unrolled model with backprop

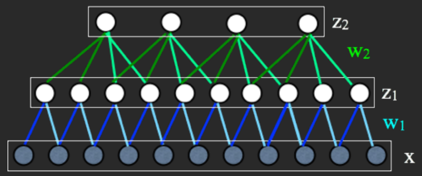
Bottom-up + top-down pass + bottom-up +….



One can model an infinitely deep model with a finite-number of passes
(by equivalence to mean-feild)

Seems like going deeper and adding skip connections (cf. residual nets) increases performance.
Proposal: let’s use structured probabilistic models as an underlying design principles

Crucial “detail”: sigmoidal vs rectified activations

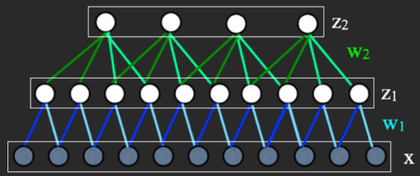


$$P(z) \propto e^{S(z)} \quad \text{where} \quad S(z) = \frac{1}{2} z^T W z + b^T z$$

$$\text{Boltzmann:} \quad z_i \in \{0, 1\}$$

Do binary variables suffice to pass info along abstraction layers?

Crucial "detail": sigmoidal vs rectified activations



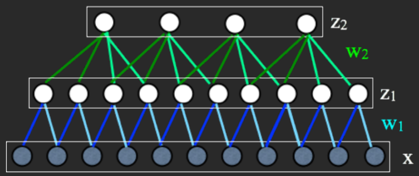
$$P(z) \propto e^{S(z)} \quad \text{where} \quad S(z) = \frac{1}{2} z^T W z + b^T z$$

$$\text{Boltzmann:} \quad z_i \in \{0, 1\}$$

$$\text{Gaussian:} \quad z_i \in R$$

Relax binary restriction:
model reduces to a Gaussian (with some caveats), implying features are linear functions of image

Crucial “detail”: sigmoidal vs rectified activations

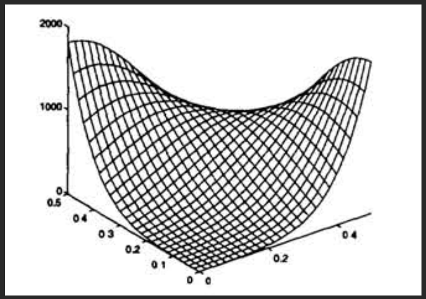


$$P(z) \propto e^{S(z)} \quad \text{where} \quad S(z) = \frac{1}{2} z^T W z + b^T z$$

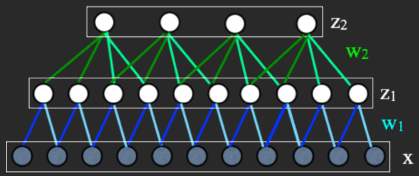
$$\text{Boltzmann:} \quad z_i \in \{0, 1\}$$

$$\text{Gaussian:} \quad z_i \in R$$

(Socci & Seung 98) $\quad \text{Rectified Gaussian:} \quad z_i \in R^+$



Deep Rectified Gaussians



$$P(z) \propto e^{S(z)} \quad \text{where} \quad S(z) = \frac{1}{2} z^T W z + b^T z$$

$$\text{Boltzmann:} \quad z_i \in \{0, 1\}$$

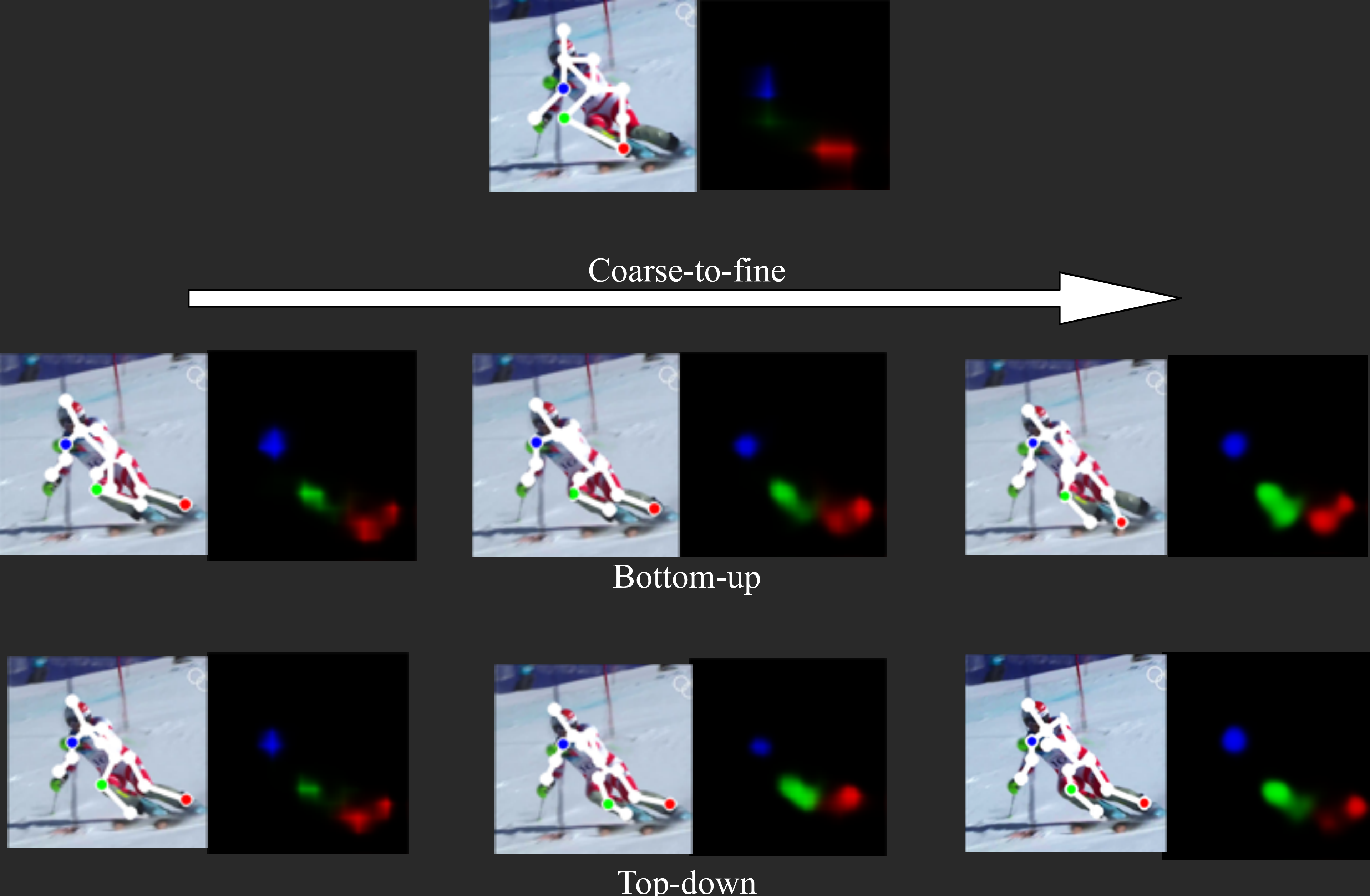
$$\text{Gaussian:} \quad z_i \in R$$

$$\text{Rectified Gaussian:} \quad z_i \in R^+$$

Hierarhically Rectified Gaussians (Hu & Ramanan 16; come see our CVPR poster!) pass continuous info between hierarchical layers, but produce nonlinear features

MAP updates:

$$z_i[u] = \max(0, b_i + top_i[u] + bot_i[u])$$


Coarse-to-fine
Bottom-up
Top-down

Simultaneous localization + visibility prediction

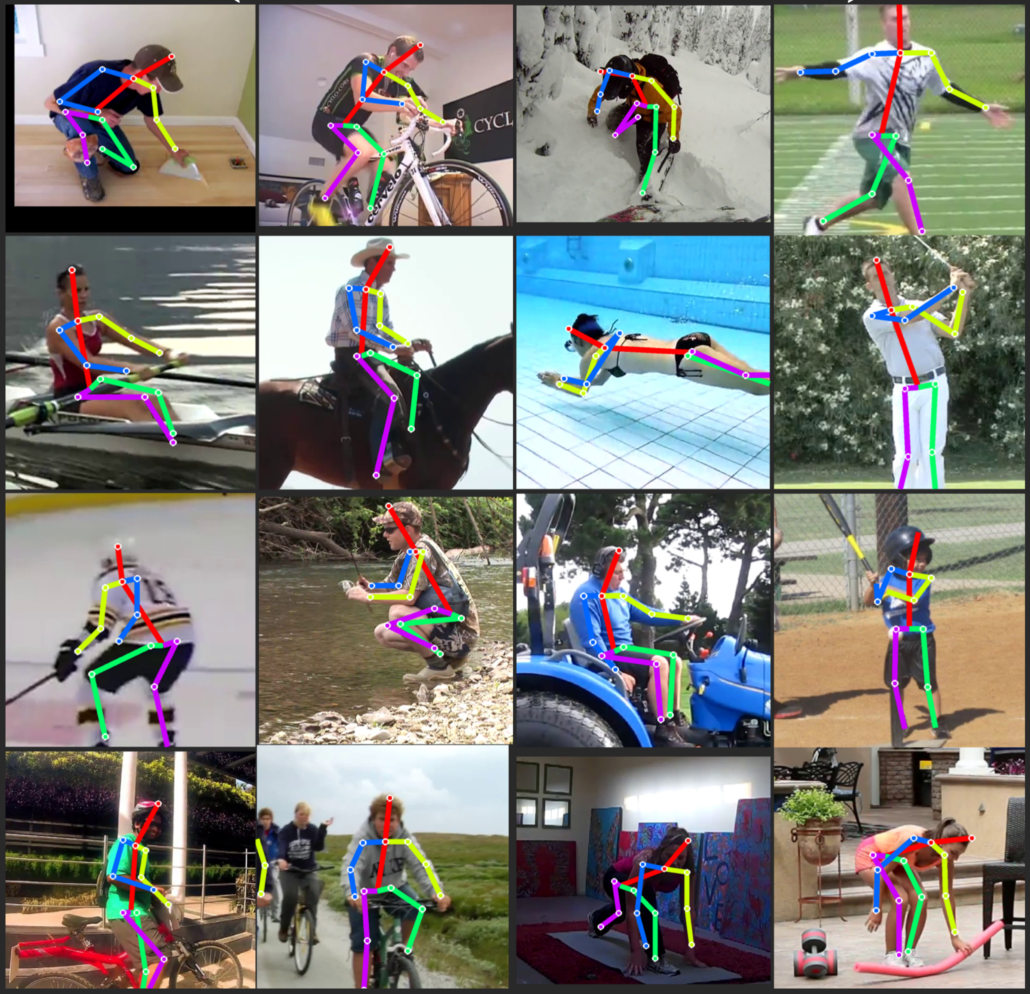
Caltech Occluded Faces occluded-point localization error (% of eye-eye distance)

Bottom-up: 21.26 %

Top-down: 15.3 %

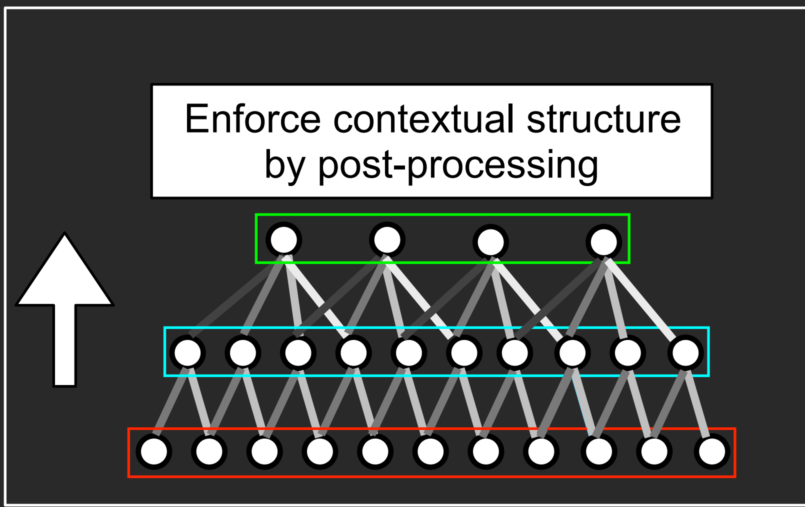
Improvement comes “for free” (no increase in # of parameters)

Human pose estimation (MPII dataset)

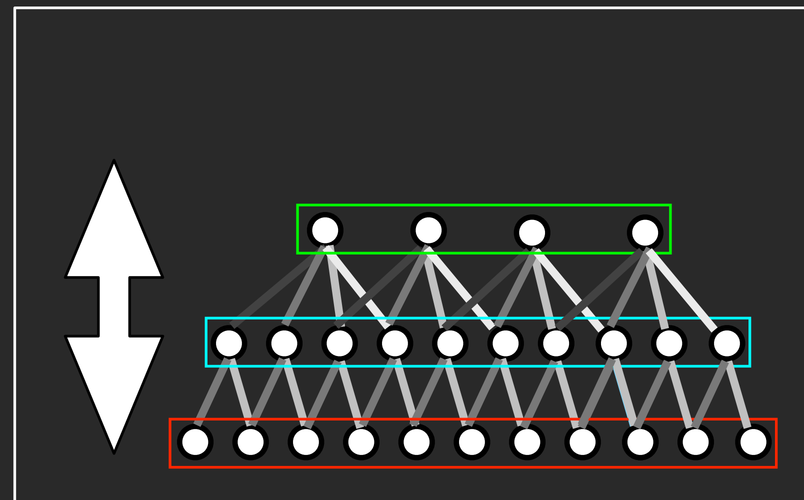


State-of-the-art (for a fleeting moment)

Take-aways (1)



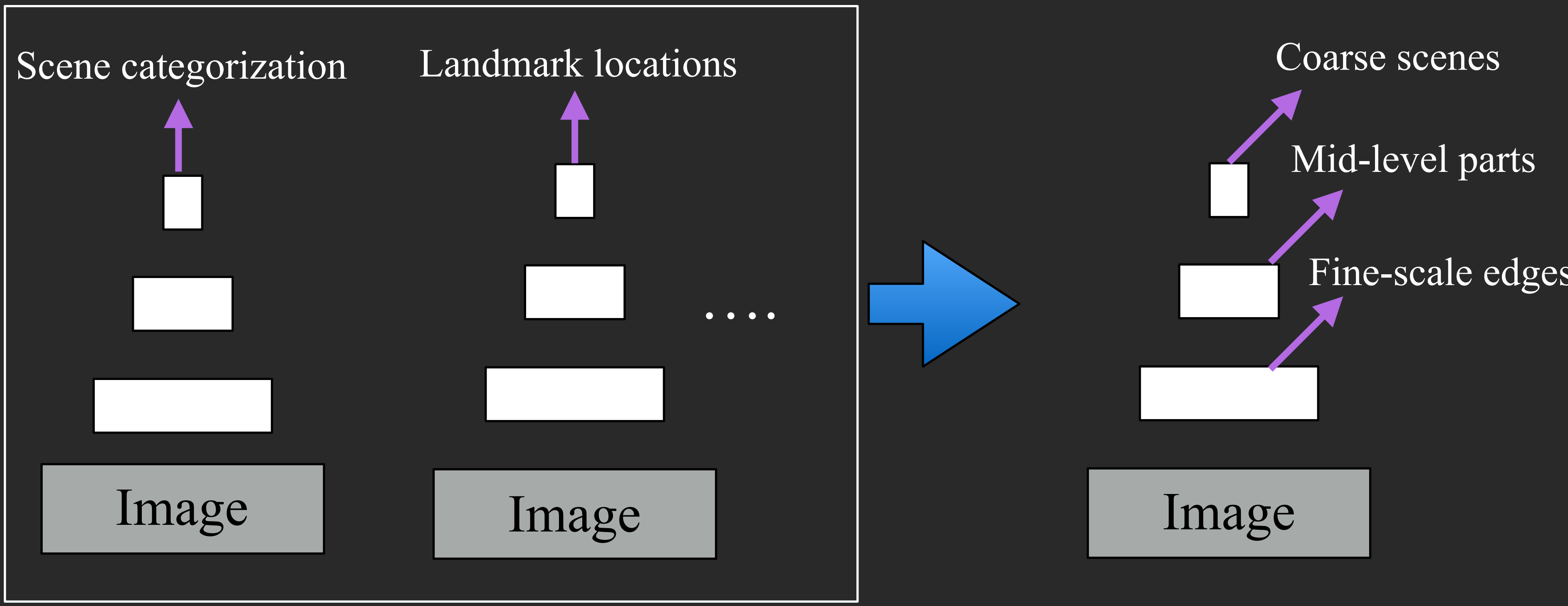
versus



- CNNs can be viewed as inference machines (if we untie their hands)
- Blurs distinction between learning and inference (backprop as feedback?)

Take-aways (2)

Rather than training and storing hundreds of task-specific models, learn+store universal feature extractor for both vision-at-a-glance and with-scrutiny tasks



Thanks!

Please visit poster in workshop and main conference