More EBM Applications: Trainable dissimilarity Metrics, Segmentation, Sequence Labeling

Yann LeCun (Courant Institute, NYU)

Sumit Chopra, Raia Hadsell, Feng Ning (NYU)

Leon Bottou (NEC), Yoshua Bengio (Montreal), Patrick Haffner (AT&T)

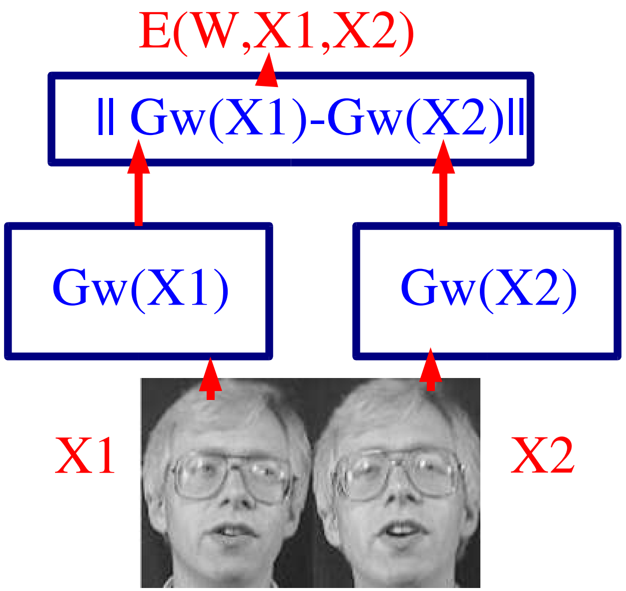
http://yann.lecun.com

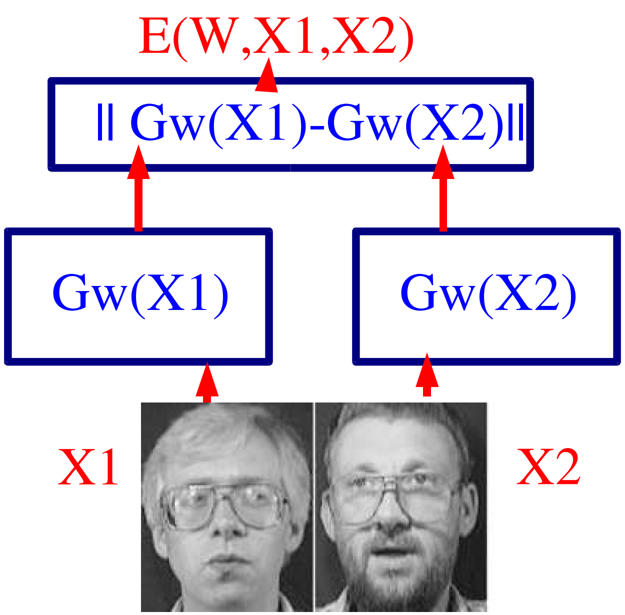
http://www.cs.nyu.edu/~yann

Learning an Invariant Dissimilarity Metric with EBMs

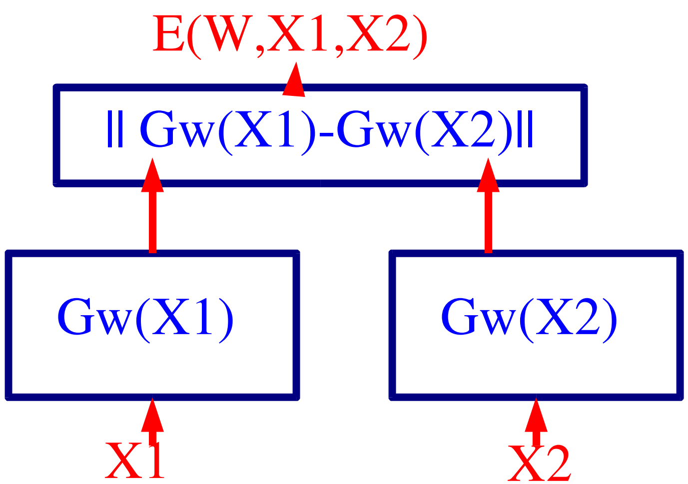
[Chopra, Hadsell, LeCun CVPR 2005]

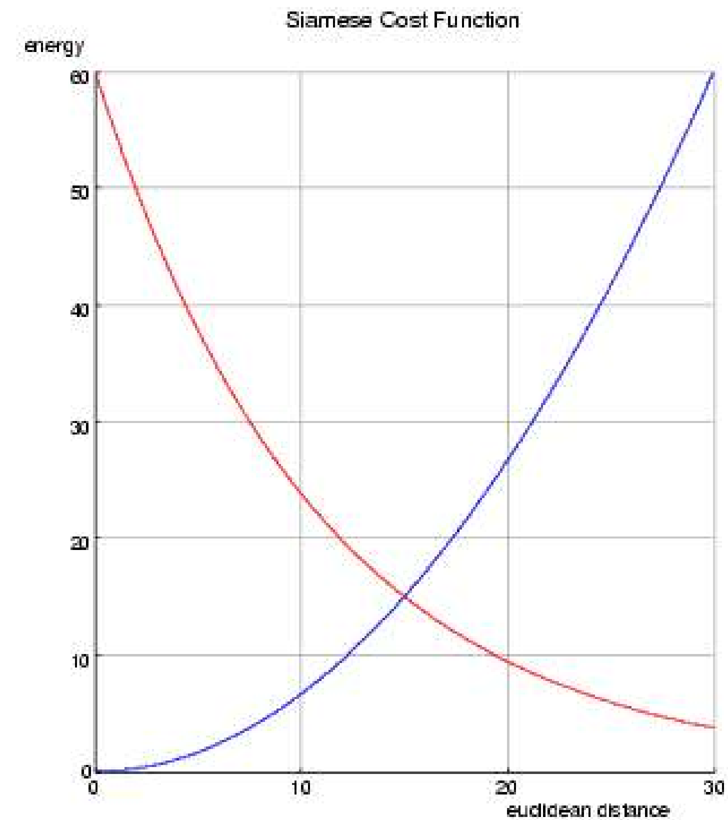
- Training a **parameterized, invariant dissimilarity metric** may be a solution to the **many-category problem**.
- Find a mapping **Gw(X)** such that the Euclidean distance **||Gw(X1)- Gw(X2)||** reflects the “semantic” distance between X1 and X2.
- Once trained, a trainable dissimilarity metric can be used to classify **new categories using a very small number of training samples** (used as prototypes).
- This is an example where probabilistic models are too constraining, because we would have to limit ourselves to models that can be normalized over the space of input pairs.
- With EBMs, we can put what we want in the box (e.g. A convolutional net).
- **Siamese Architecture**
- **Application:** face verification/recognition





Learning an Invariant Dissimilarity Metric with EBMs





- **Siamese models:** distance between the outputs of two identical copies of a model.
- E(W,X1,X2) = ||Gw(X1)-Gw(X2)||
- If X1 and X2 are from the **same category**, train the two copies of the model to produce **similar outputs**
- If X1 and X2 are from **different categories**, train the two copies of the model to produce **different outputs**
- Loss function: square-exponential loss:

$$L(W,Y,X_1,X_2) = (1-Y)\cdot\frac{2}{R}(\|G_w(X_1)-G_w(X_2)\|)^2 + Y\cdot 2Re^{-\frac{K}{R}||G_w(X_1)-G_w(X_2)||}$$

Face Verification datasets: AT&T/ORL

- The AT&T/ORL dataset
- Total subjects: **40**. Images per subject: **10**. Total images: **400**.
- Images had a moderate degree of variation in pose, lighting, expression and head position.
- Images from **35** subjects were used for training. Images from **5** remaining subjects for testing.
- Training set was taken from: **3500** genuine and **119000** impostor pairs.
- Test set was taken from: **500** genuine and **2000** impostor pairs.
- http://www.uk.research.att.com/facedatabase.html

AT&T/ORL Dataset

Face Verification datasets: AR/Purdue dataset

- The AR/Purdue dataset
- Total subjects: **136**. Images per subject: **26**. Total images: **3536**.
- Each subject has 2 sets of 13 images taken 14 days apart.
- Images had very high degree of variation in pose, lighting, expression and position. Within each set of 13, there are 4 images with expression variation, 3 with lighting variation, 3 with dark sun glasses and lighting variation, and 3 with face obscuring scarfs and lighting variation.
- Images from **96** subjects were used for training. The remaining **40** subjects were used for testing.
- Training set drawn from: **64896** genuine and **6165120** impostor pairs.
- Test set drawn from: **27040** genuine and **1054560** impostor pairs.
- http://rv11.ecn.purdue.edu/aleix/aleix_face_DB.html

Face Verification dataset: AR/Purdue

Dataset for Verification

- tested on AT&T and AR/Purdue
- AT&T dataset

Number of subjects:	5
Images/subject:	10
Images/Model:	5
Total test size:	5000
Number of Genuine:	500
Number of Impostors:	4500

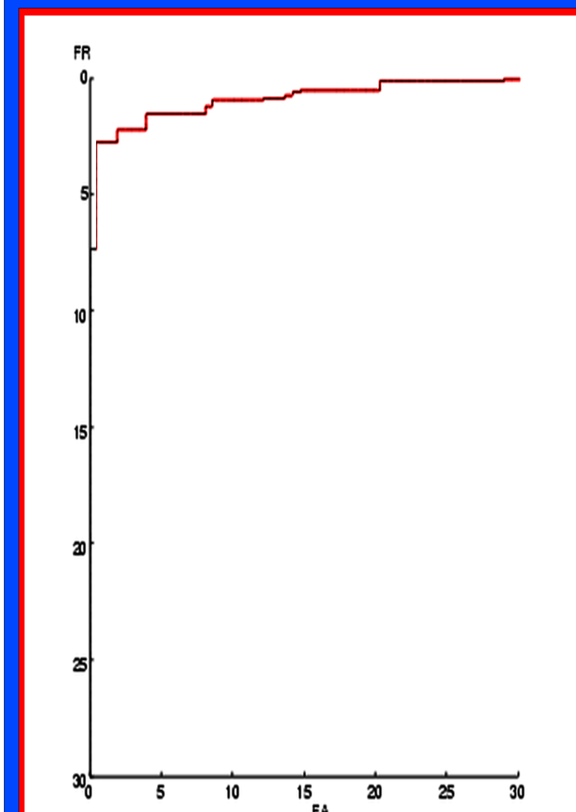
- Purdue/AR dataset

Number of subjects:	40
Images/subject:	26
Images/Model:	13
Total test size:	5000
Number of Genuine:	500
Number of Impostors:	4500

Verification Results

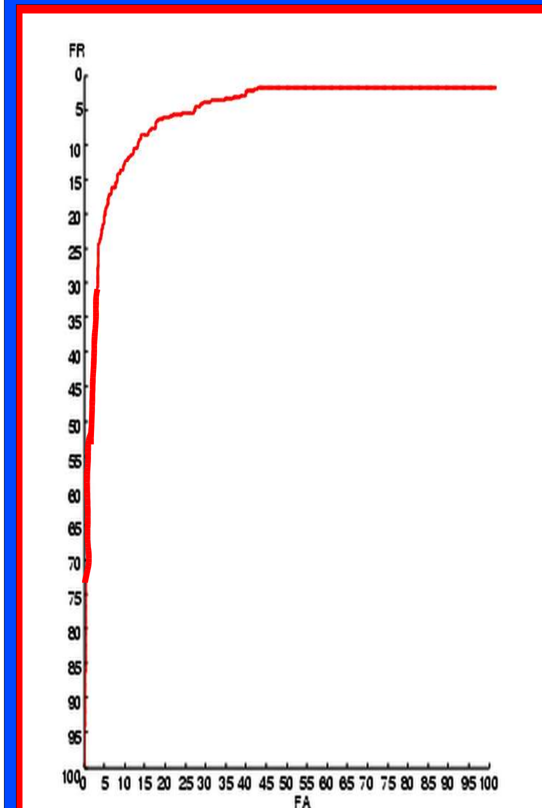
- The AT&T dataset

False Accept	False Reject
10.00%	0.00%
7.50%	1.00%
5.00%	1.00%



- The AR/Purdue dataset

False Accept	False Reject
10.00%	11.00%
7.50%	14.60%
5.00%	19.00%



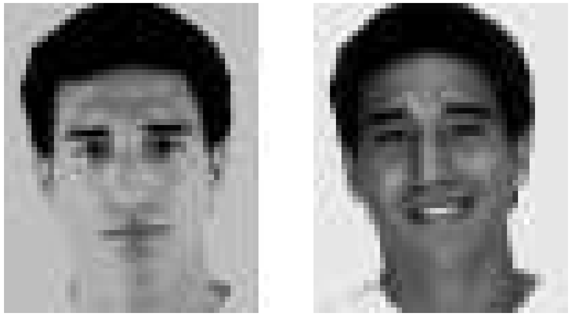
Internal state for genuine and impostor pairs

Classification Examples

Example: Correctly classified genuine pairs

energy: 0.3159

energy: 0.0043

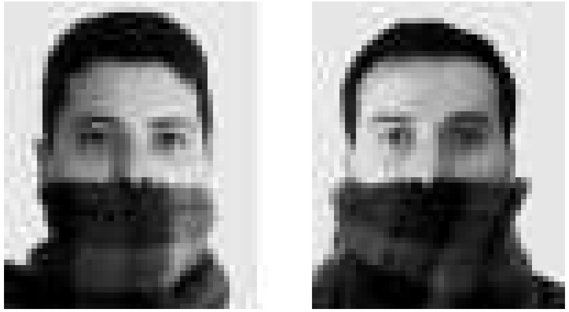


energy: 0.0046

Example: Correctly classified impostor pairs

energy: 20.1259

energy: 32.7897



energy: 5.7186

Example: Mis-classified pairs

energy: 10.3209

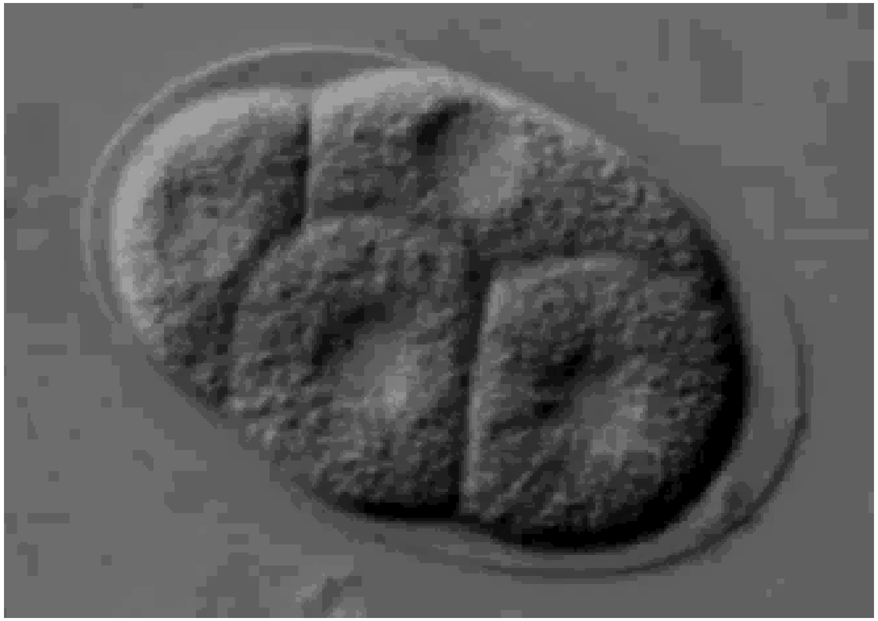
energy: 2.8243

C. Elegans Embryo Phenotyping

[Ning, Delhome, LeCun, Piano, Bottou, Barbano

IEEE Trans. Image Processing 2005 (in press)]

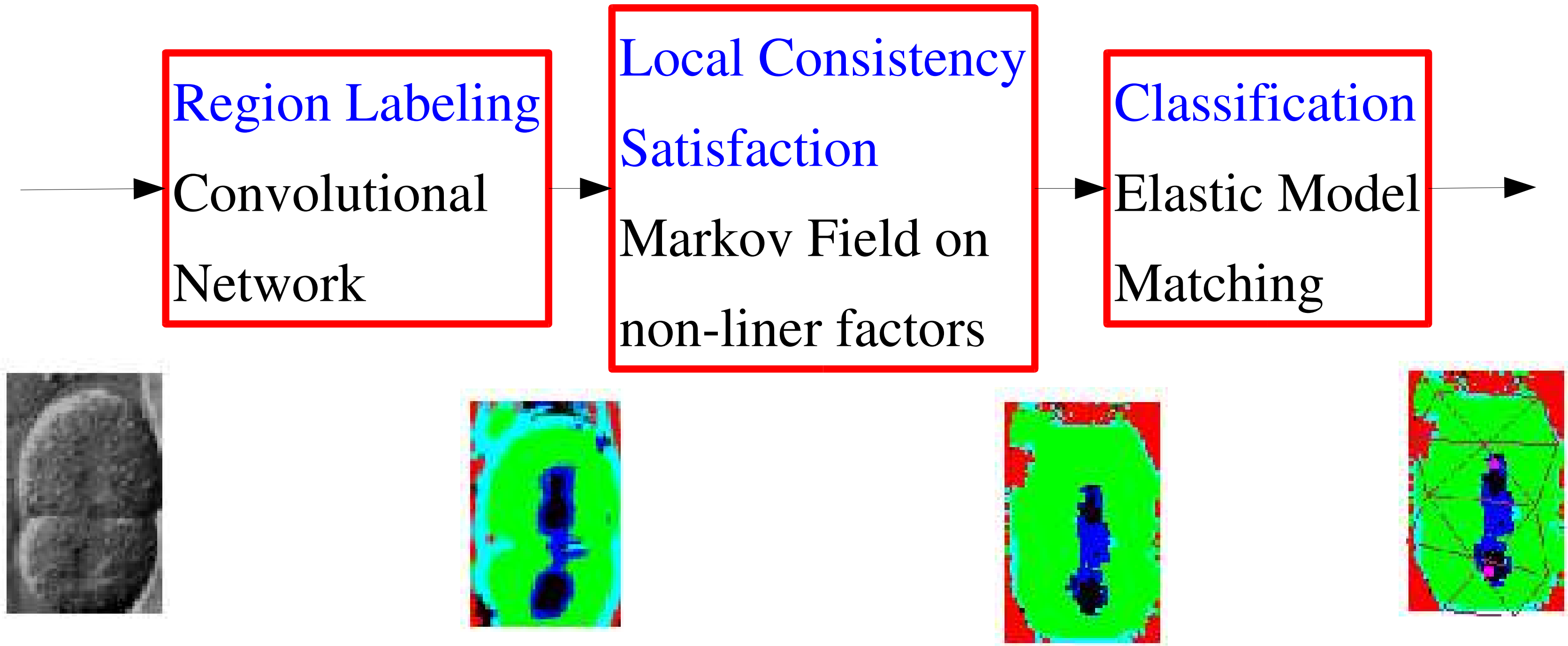
- **Analyzing results for Gene Knock-Out Experiments**
- **Automatically determining if a roundworm embryo is developing normally after a gene has been knocked out.**



Time-lapse movie

Architecture

- **Region Classification with a convolutional network**
- **Local Consistency with a Markov Field of non-linear factors**
- **Embryo classification with elastic model matching**



Region Labeling with a Convolutional Net

- Supervised training fromhand-labeled images
- 5 categories:
  - nucleus, nuclear membrane, cytoplasm, cell wall, external medium

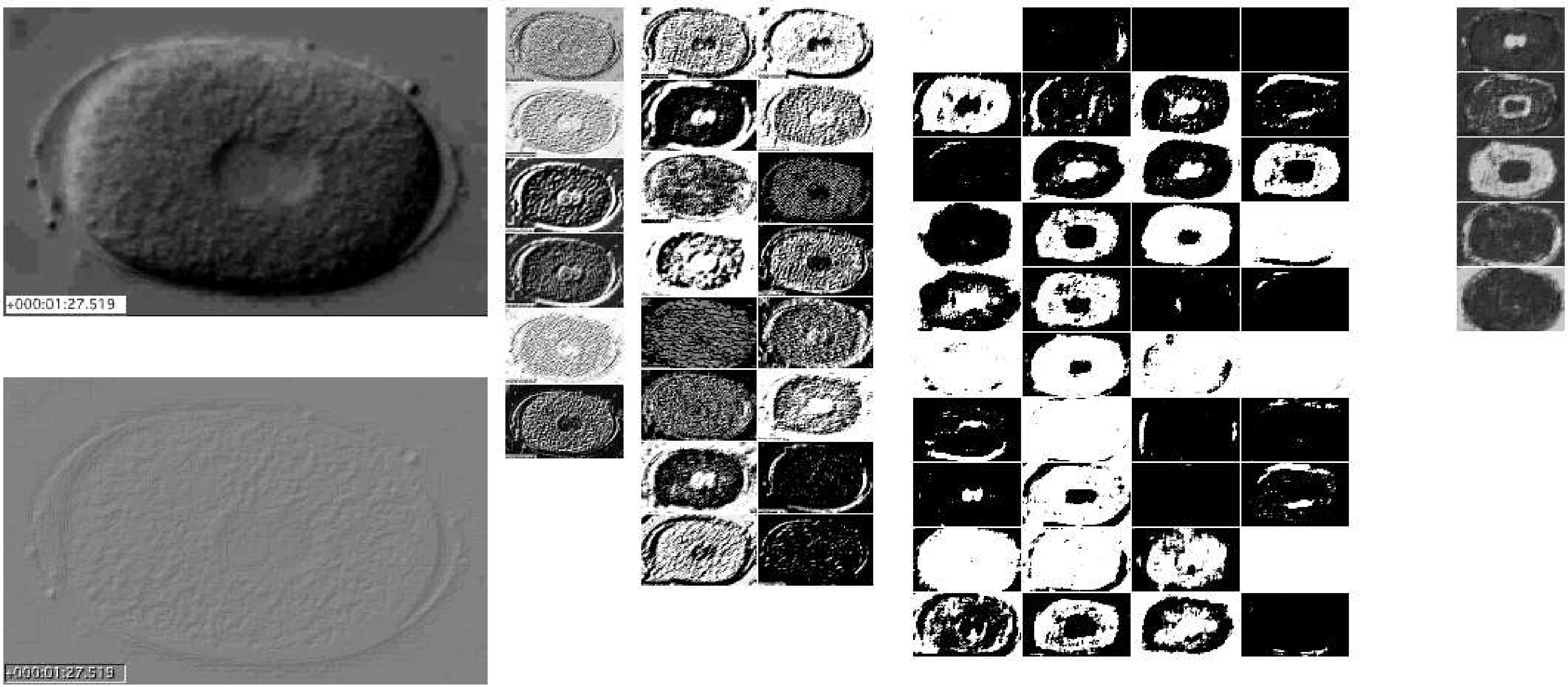
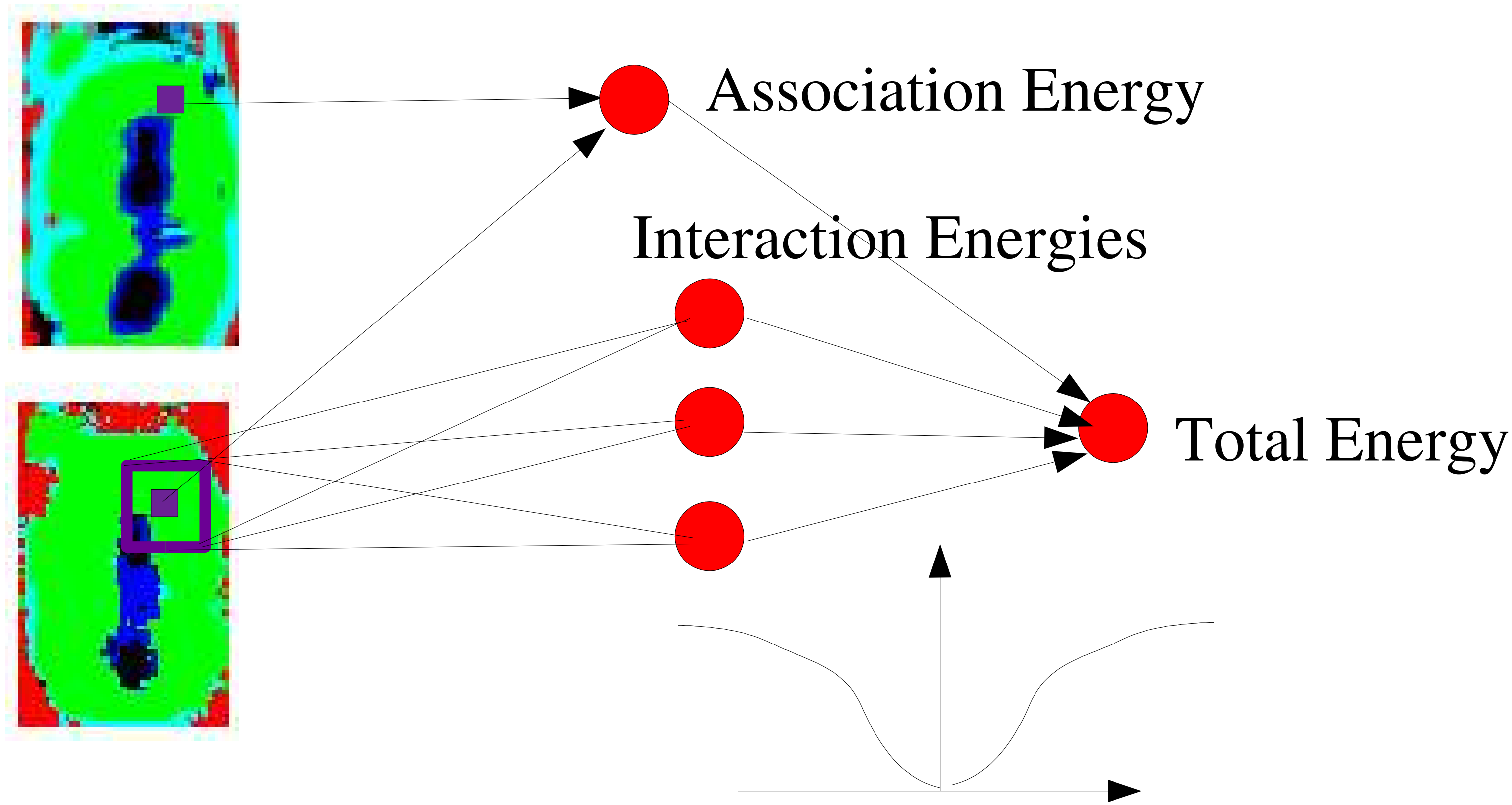


Image Segmentation with Local Consistency Constraints

[Teh, Welling, Osindero, Hinton, 2001], [Kumar, Hebert 2003], [Zemel 2004]

- **Learn local consistency constraints with an Energy-Based Model so as to clean up images produced by the segmentor.**



C. Elegans Embryo Phenotyping

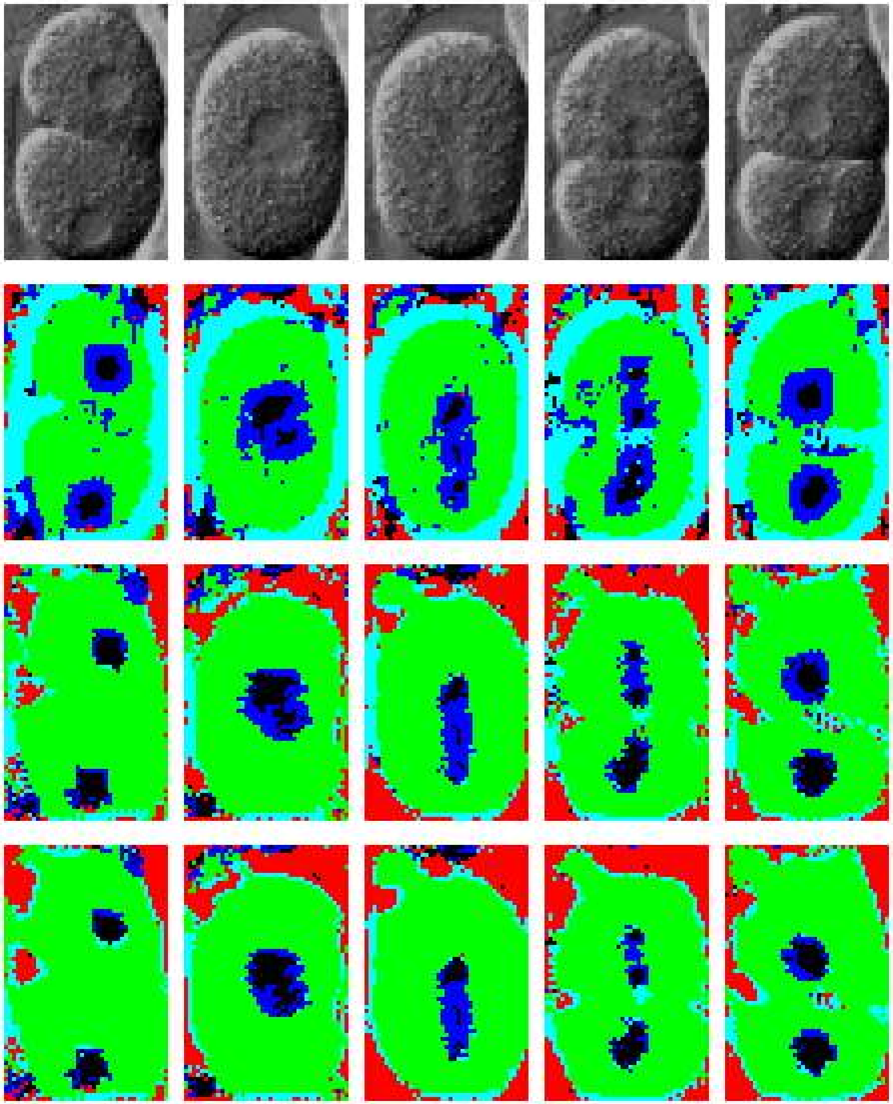
Analyzing results for Gene Knock-Out Experiments

Original Images

Segmentation #1

Segmentation #2

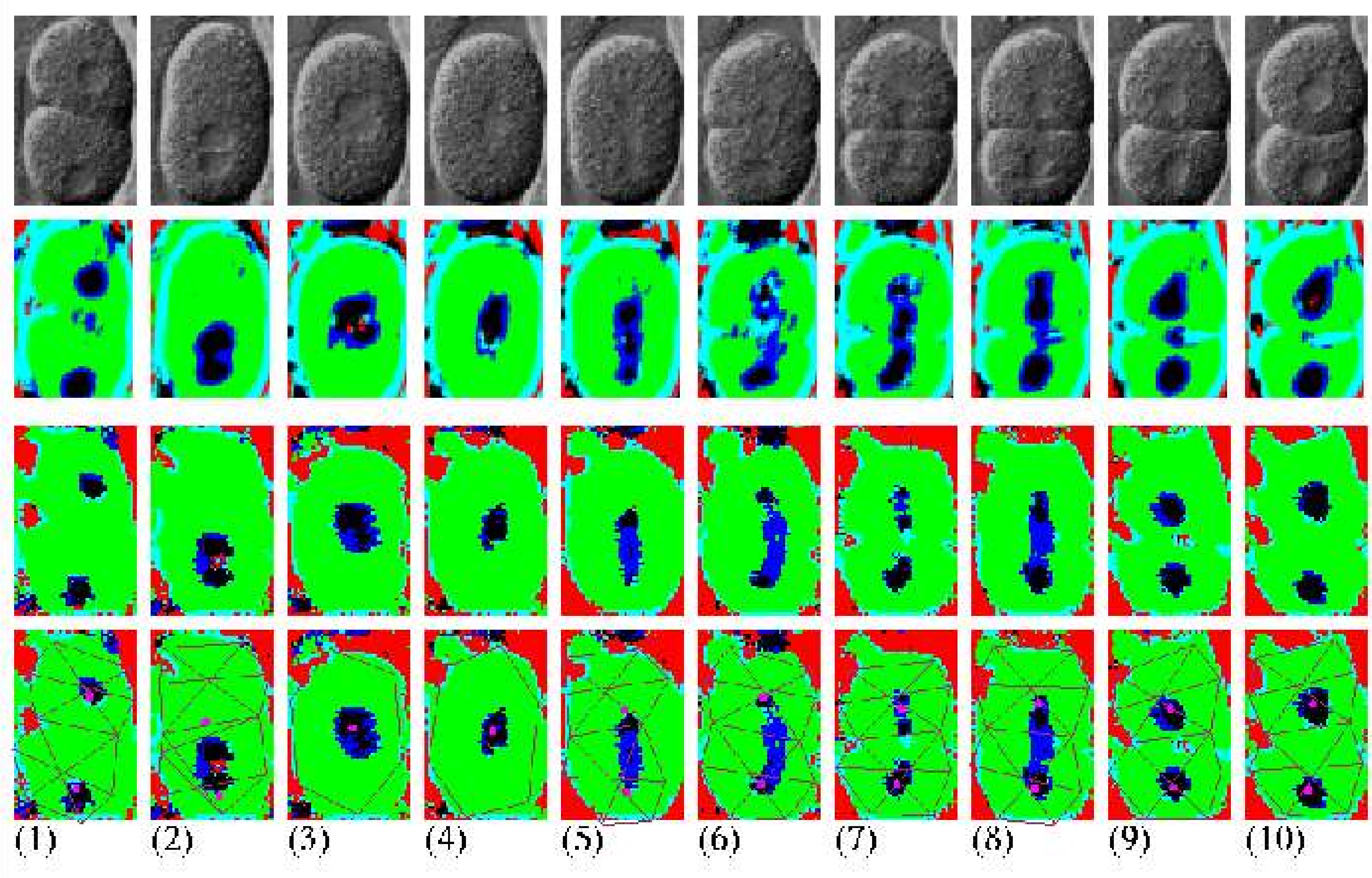
Non-Linear CRF Cleanup



(1) (2) (3) (4) (5)

C. Elegans Embryo Phenotyping

Analyzing results for Gene Knock-Out Experiments

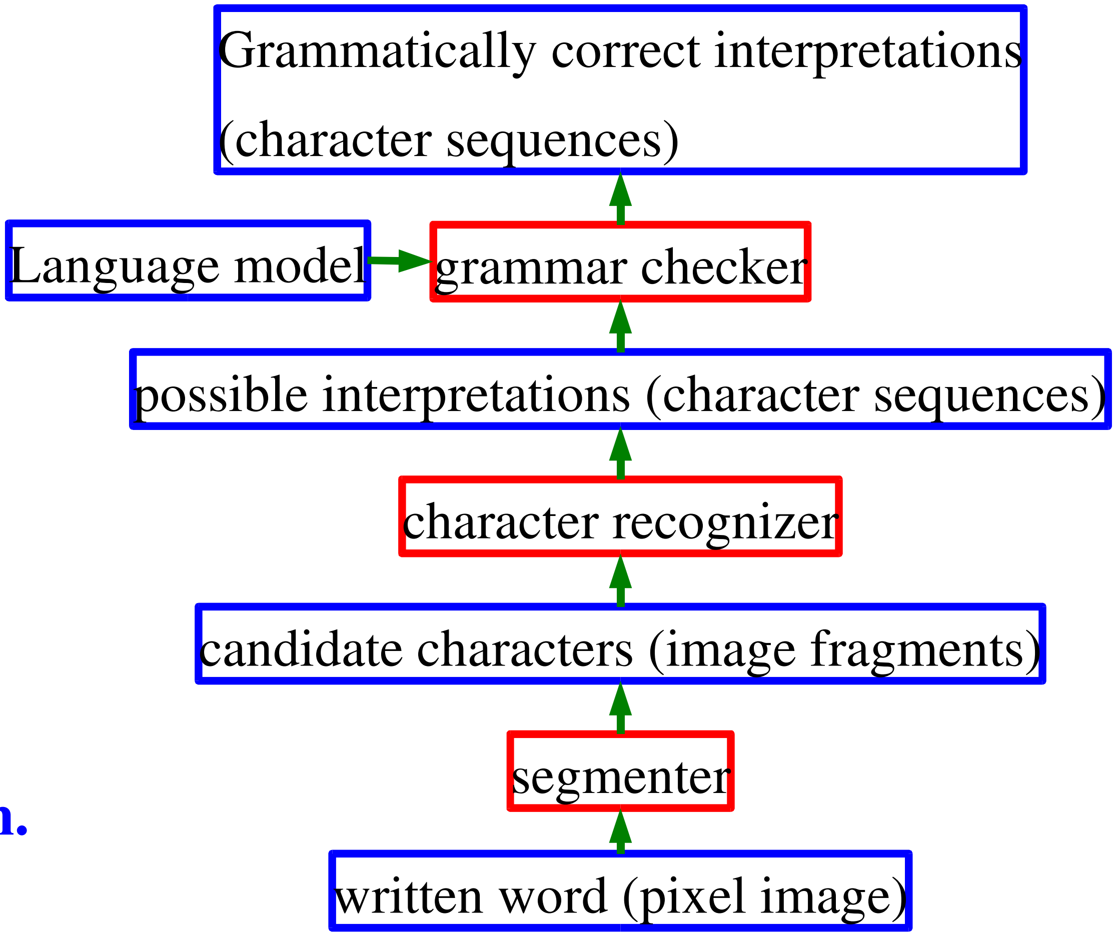


Sequence Labeling

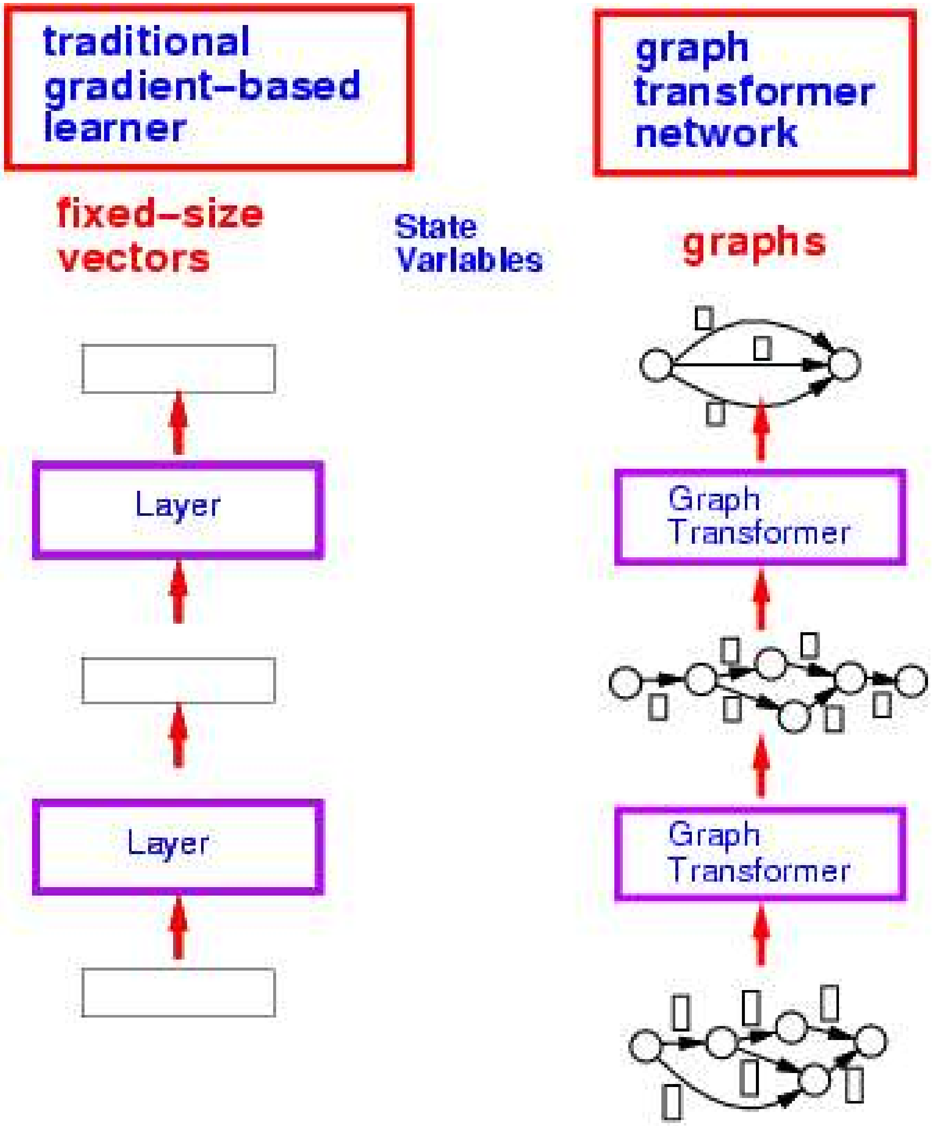
- **Many applications manipulate variable-length sequences, rather than fixed-size vectors or images.**
  - Speech Recognition, Handwriting Recognition, Natural Language Processing (parsing, tagging....), Biological Sequence Analysis......
- **What architectures can manipulate sequences?**
- **Alternative interpretations of sequences are best represented by directed graphs with values attached to the edges**
  - Each alternative segmentation and interpretation of a spoken sentence or a written word can be represented by a path in a lattice.
- **How do we build multi-layer modular systems that take graphs as inputs and produce graphs on output?**

End-to-End Training of a graph manipulating machine.

- Example: a handwriting recognition system.
- Each intermediate representation is a valued graph
- Each module is trainable
- The entire system is trained simultaneously so as to optimize a global loss function.



Using Graphs instead of Vectors.

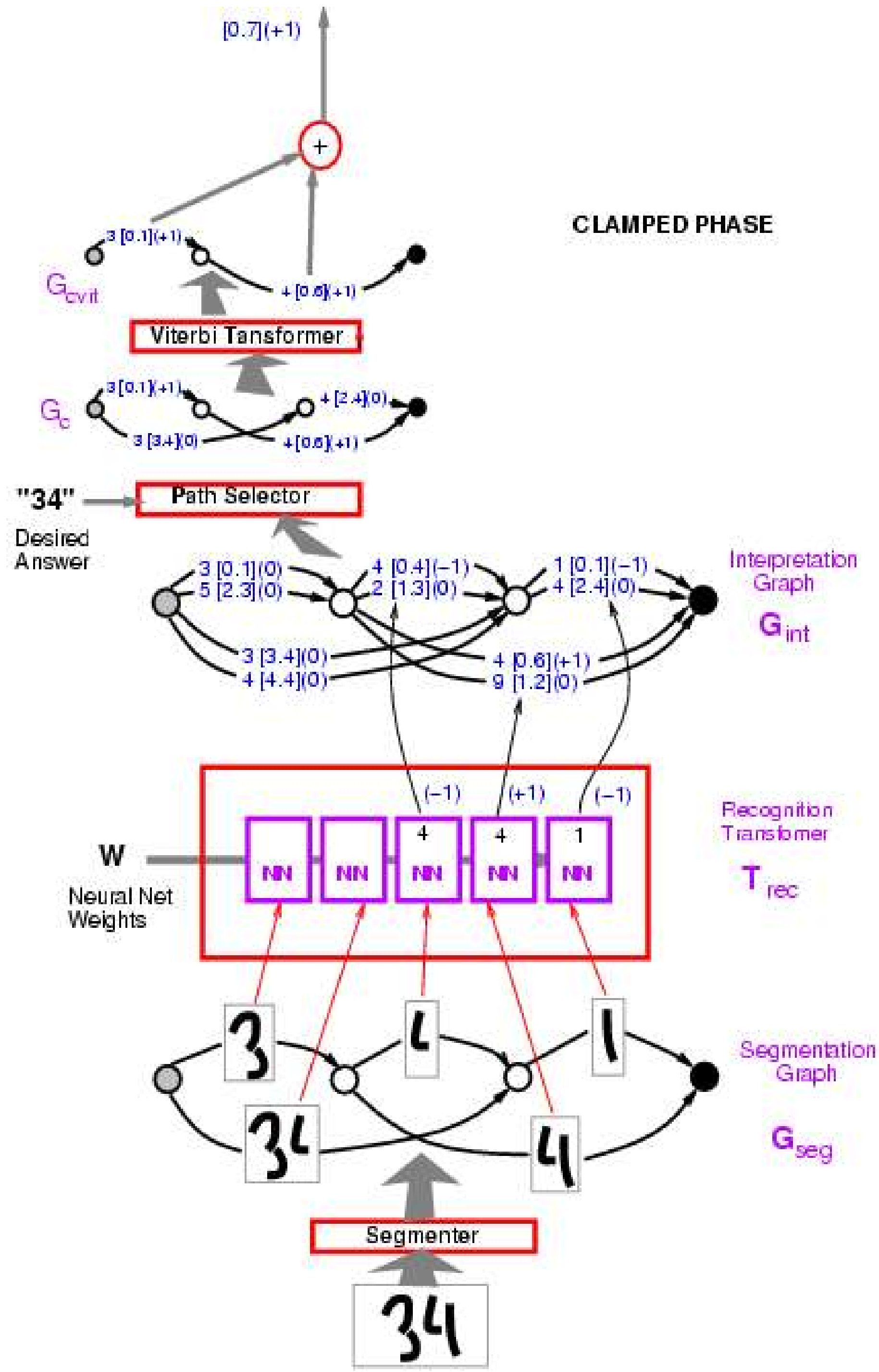


- Whereas traditional learning machines manipulate fixed-size vectors, Graph Transformer Networks manipulate graphs.

Graph Transformer Networks

- **Variables:**
  - X: input image
  - Z: path in the interpretation graph/segmentation
  - Y: sequence of labels on a path
- **Loss function: computing the energy of the desired answer:**

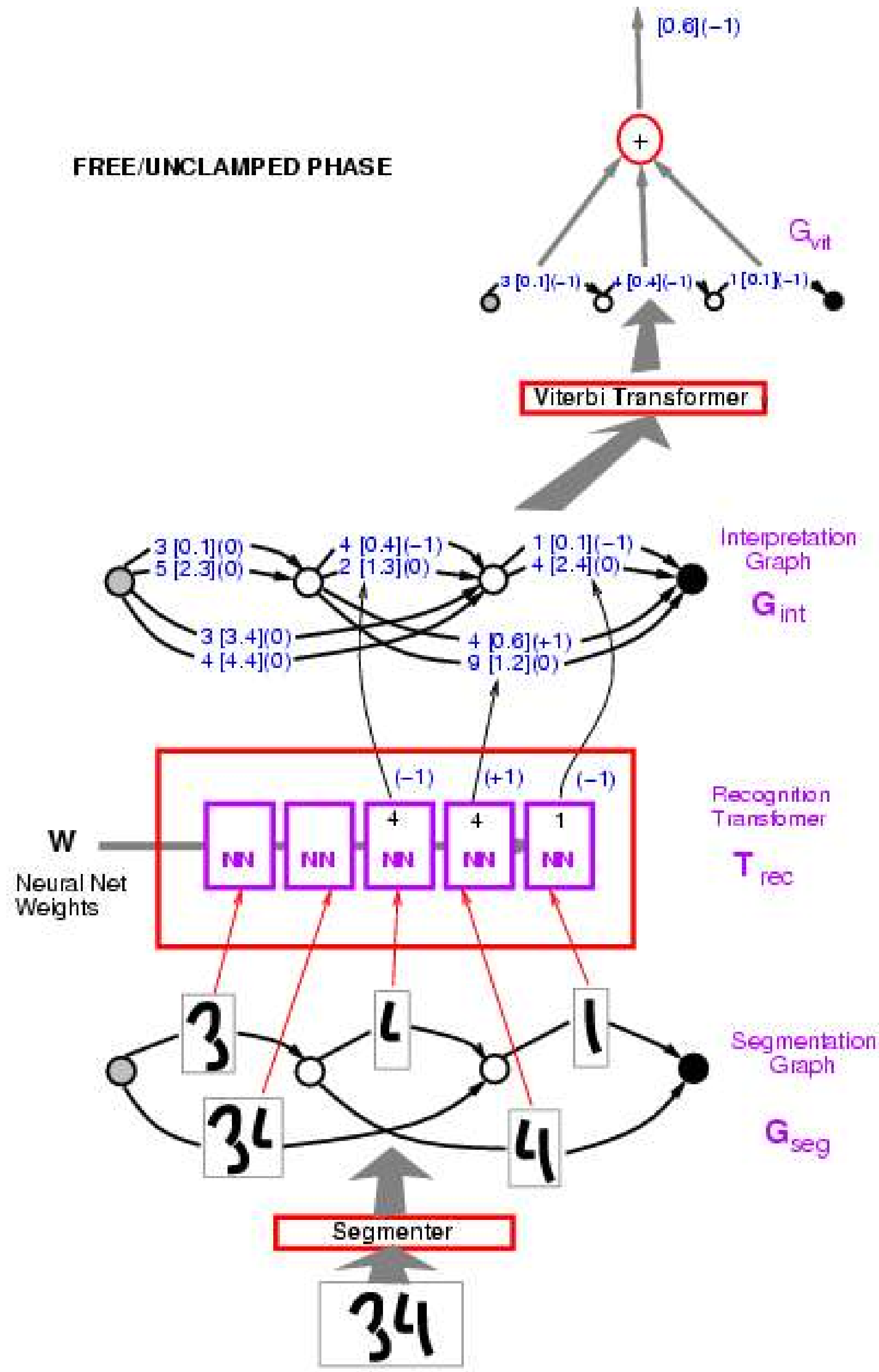
$$E(W, Y, X)$$



Graph Transformer Networks

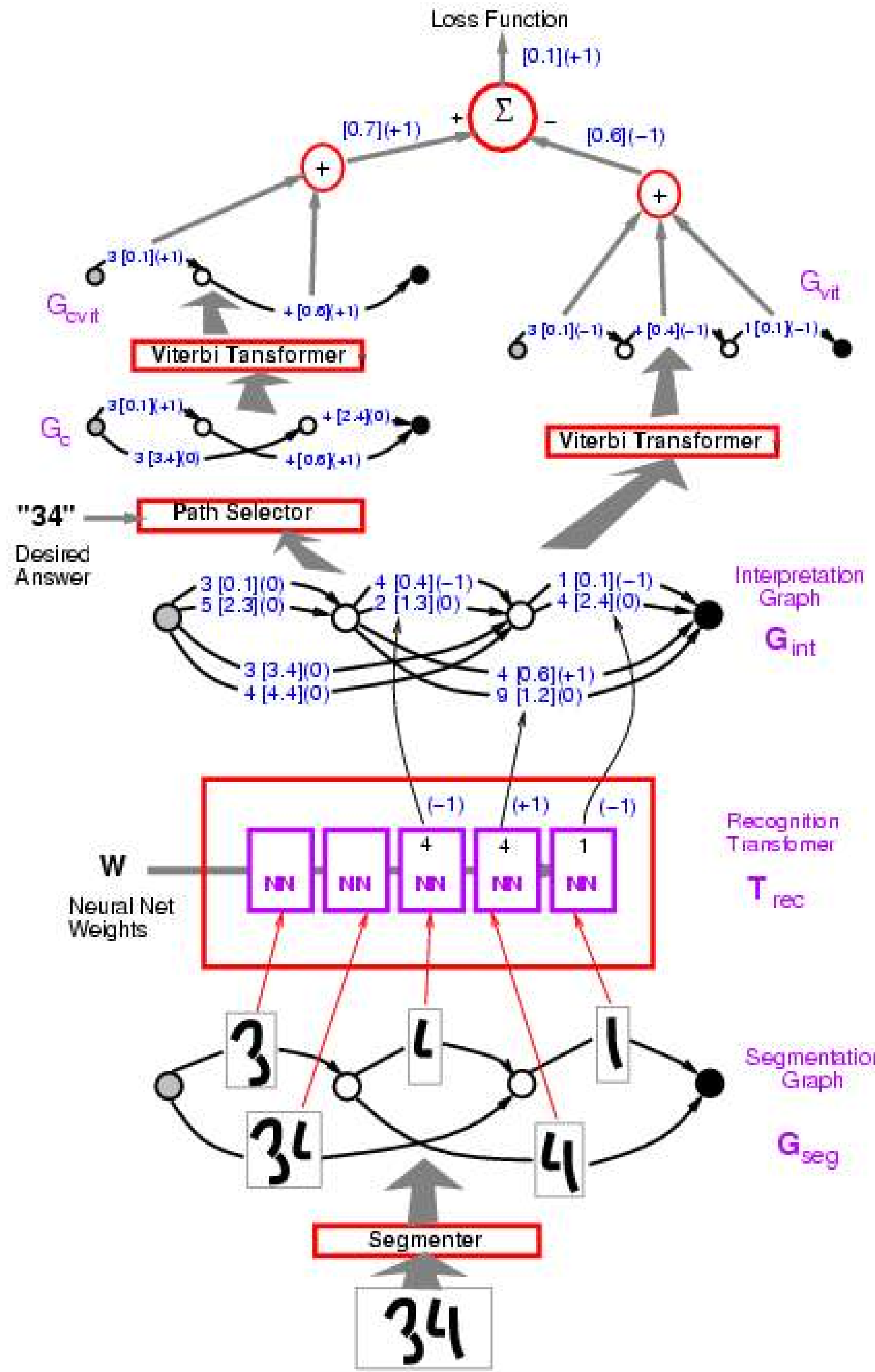
- **Variables:**
  - X: input image
  - Z: path in the interpretation graph/segmentation
  - Y: sequence of labels on a path
- **Loss function: computing the constrastive term:**

$$E(W, \check{Y}, X)$$



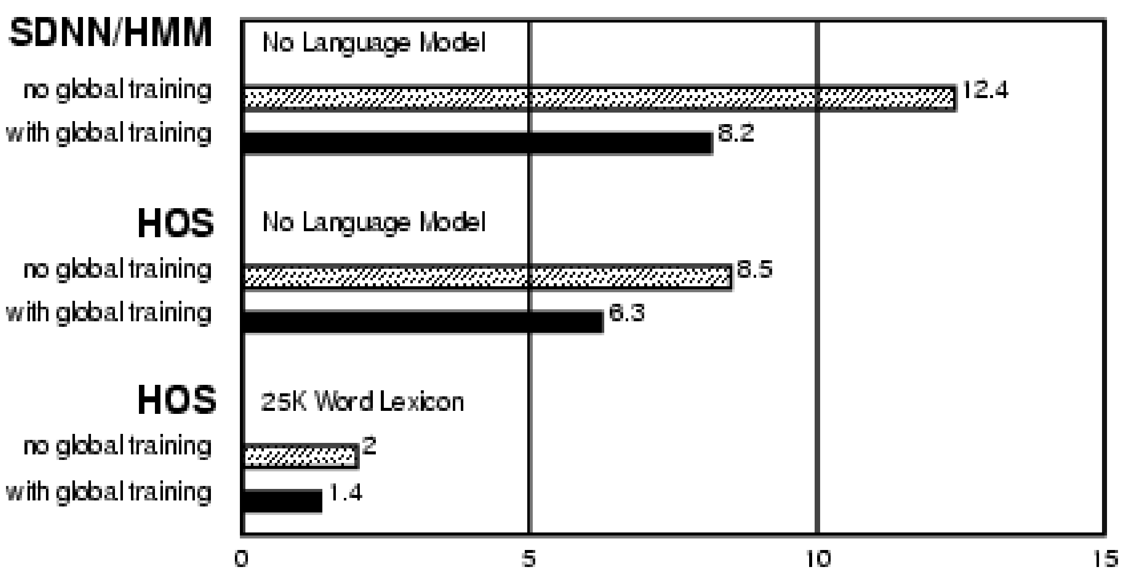
Graph Transformer Networks

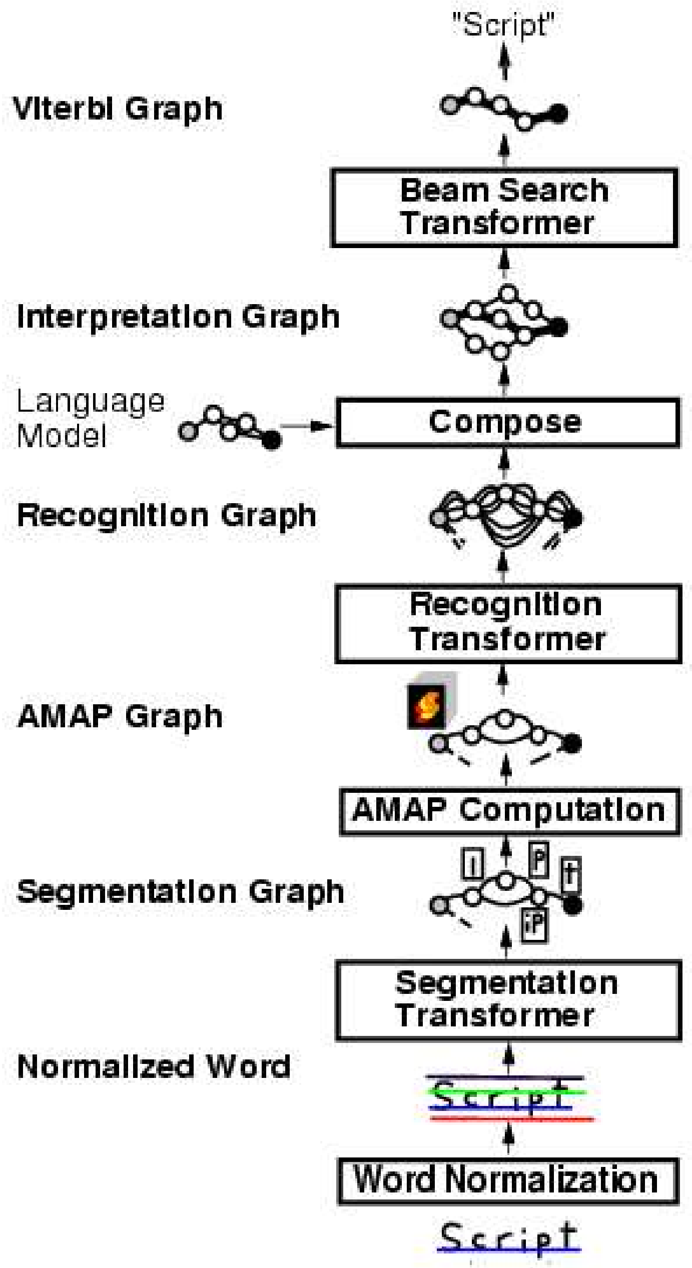
- **Example: Perceptron loss**
- **Loss = Energy of desired answer – Energy of best answer.**
  - (no margin)



Global Training Helps

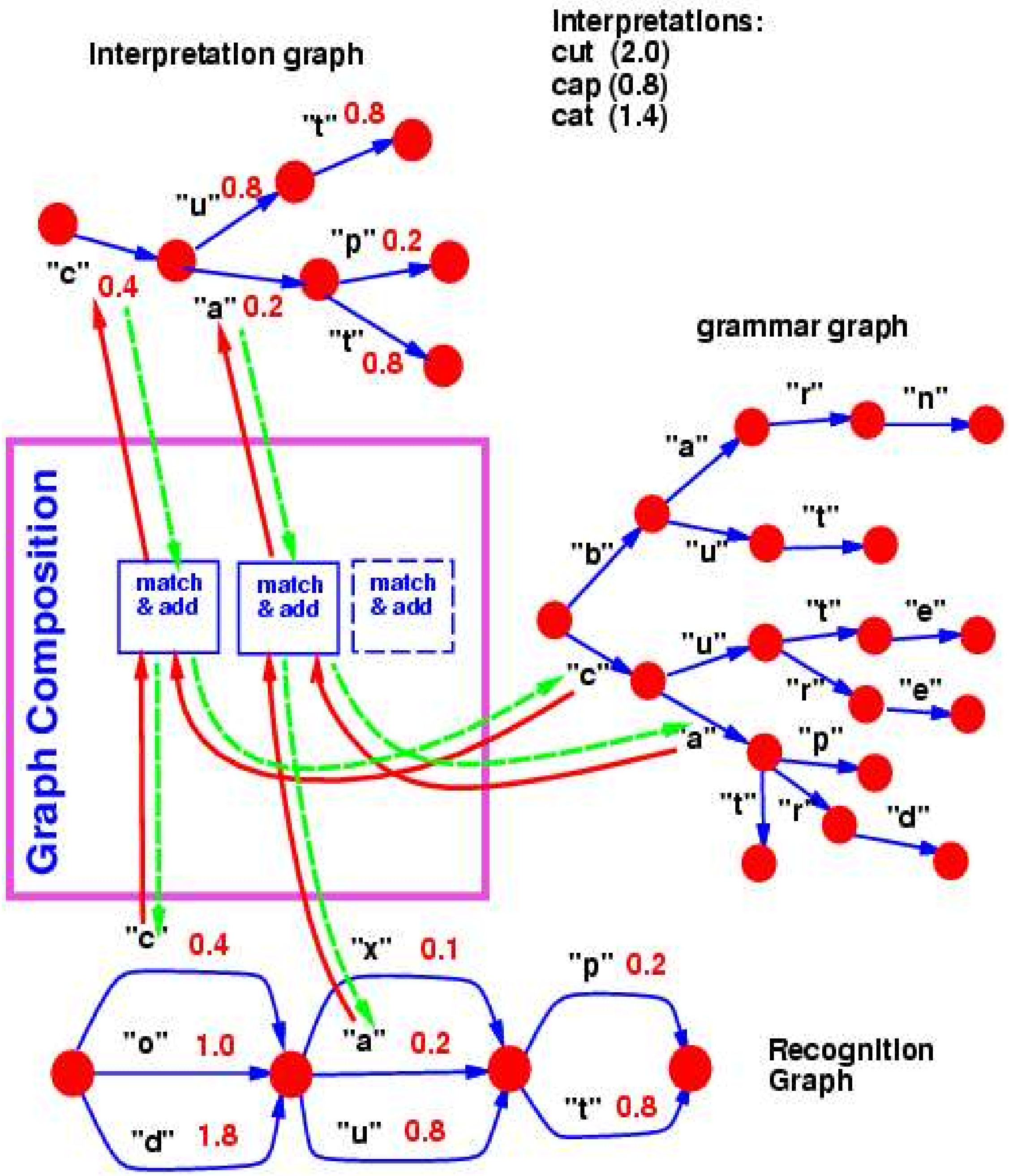
- **Pen-based handwriting recognition (for tablet computer)**
  - [Bengio&LeCun 1995]





Graph Composition, Transducers.

- The composition of two graphs can be computed, the same way the dot product between two vectors can be computed.
- General theory: semi-ring algebra on weighted finite-state transducers and acceptors.



Check Reader

- **Graph transformer network trained to read check amounts.**
- **Trained globally with Negative-Log-Likelihood loss.**
- **50% percent corrent, 49% reject, 1% error (detectable later in the process.**
- **Fielded in 1996**
- **Processes an estimated 10% of all the checks written in the US.**