Gradient-Based Learning Applied to Document Recognition

Douglas Hohensee

COS 598b

Pattern Recognition, in General

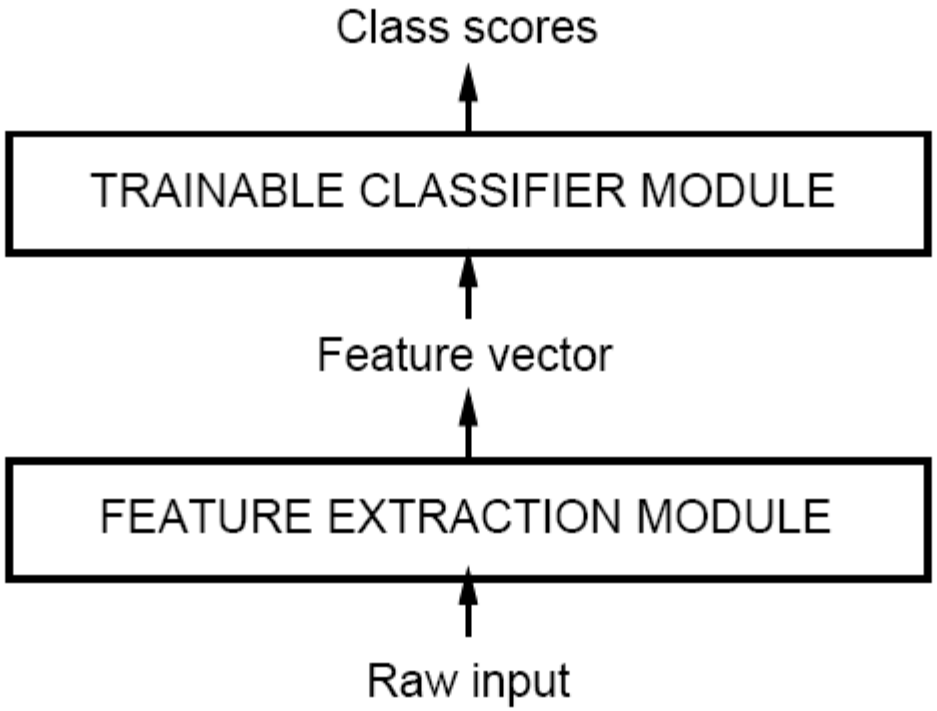


Fig. 1. Traditional pattern recognition is performed with two modules: a fixed feature extractor, and a trainable classifier.

Feature extraction

- Represent input with low-dimensional vectors
- Tends to be hand-crafted
- Success of algorithm depends on the chosen class of features

Gradient-Based Learning

- Classifiers calculate some function

$$Y^p = F(Z^p, W)$$

Z^p = the p-th input pattern
W = adjustable model parameters
Y^p = class label

- Loss Function:

$$E^p = D(D^p, F(Z^p, W))$$

  - Quantify discrepancy between D^p and Y^p

Gradient-Based Learning

- Theoretical performance limits ([3],[4],[5])]
- As # training examples increases,

$$E_{test} - E_{train} = k(h/P)^{\alpha} \qquad (1)$$

P = # of training samples

h = “effective capacity” ([6],[7])

0.5 <= α <= 1.0

k = constant

Ex: Structural Risk Minimization: find min of $E_{train} + kH(W)$

Gradient-Based Learning

- Gradient-based minimization procedure

$$W_k = W_{k-1} - \epsilon \frac{\partial E(W)}{\partial W}. \qquad (2)$$

- In practice, local minima seem not to be a problem
- “somewhat of a theoretical mystery”

Handwriting Recognition

- Problem 1: recognize individual characters
- Problem 2: separate out characters from neighbors

- Heuristic Over-Segmentation
  - Generate a lot of potential cuts, and select the best combination of cuts based on scores for each candidate character
  - But: Is half of ‘4’ a ‘1’? Is half of ‘8’ a ‘3’?

Handwriting Recognition

- Again, most systems = multiple modules
  - Field locator
  - Field segmenter
  - Recognizer
  - Contextual post-processor

  - Usually, each module trained separately!

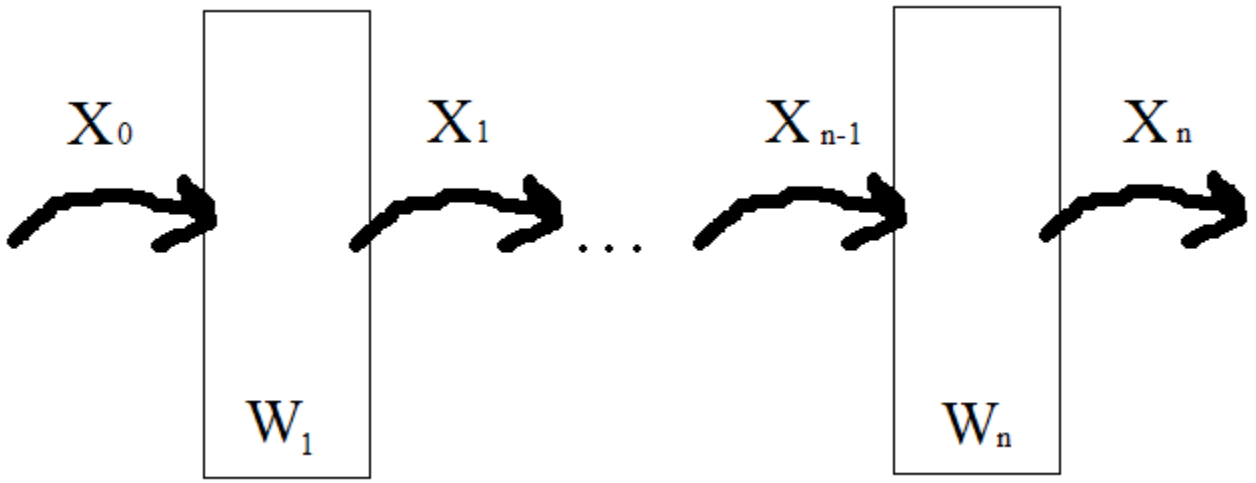
Globally Trainable System

- Start with a set of modules:

$$X_n = F_n(W_n, X_{n-1})$$

X_n = output of n-th module

W_n = parameters of n-th module



Globally Trainable System

if $\frac{\partial E^p}{\partial X_n}$ *is known, then*

$$\begin{aligned} \frac{\partial E^p}{\partial W_n} &= \frac{\partial F}{\partial W}(W_n, X_{n-1})\frac{\partial E^p}{\partial X_n} \\ \frac{\partial E^p}{\partial X_{n-1}} &= \frac{\partial F}{\partial X}(W_n, X_{n-1})\frac{\partial E^p}{\partial X_n} \end{aligned} \qquad (4)$$

Back-Propagation

- Calculate error in each output neuron.
- For each neuron, calculate local error.
- Adjust weights of each neuron to lessen local error.
- Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
- Repeat the steps above on the neurons at the previous level, using each one's "blame" as its error.

(wikipedia)

LeNet-5

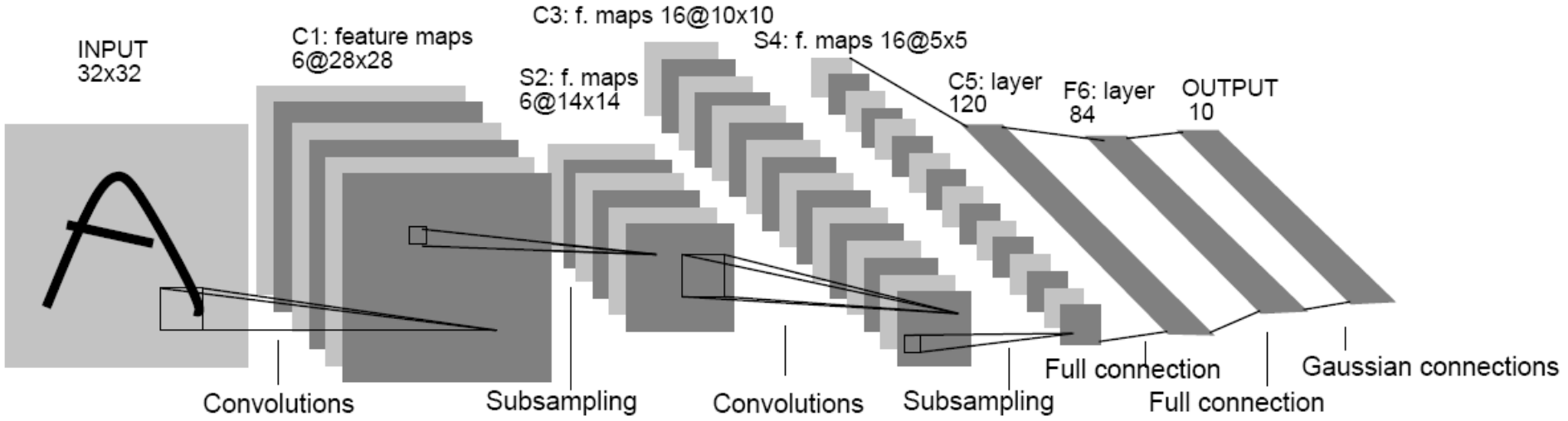
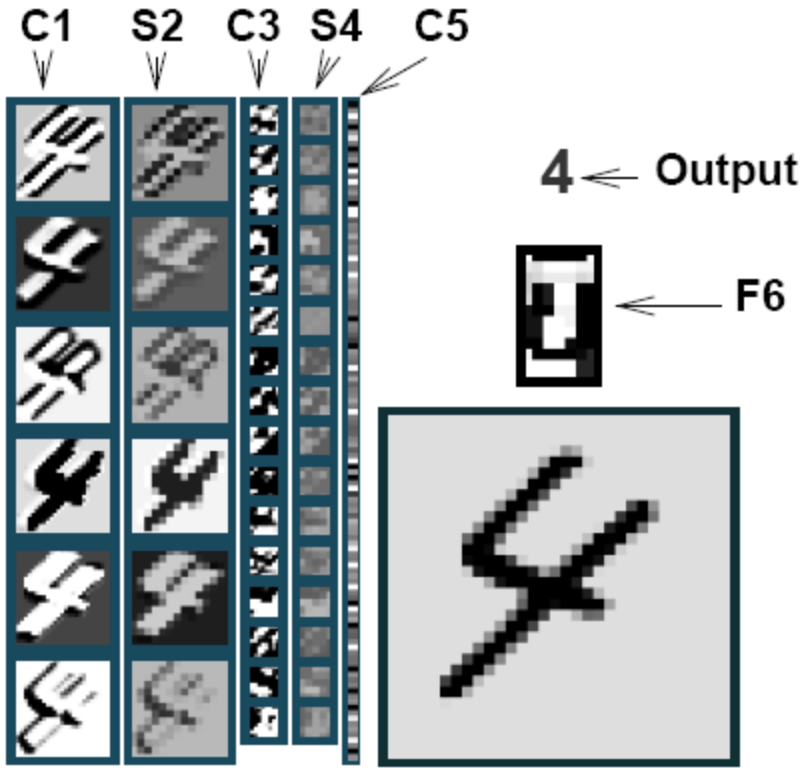


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

1. Local receptive fields
2. Shared weights
3. Sub-sampling

$f(a) = A \tanh(Sa)$

LeNet-5



LeNet-5

- RBF's can adapt
- But, E(W) has trivial solution, with all RBF identical and F6 const.

Fig. 3. Initial parameters of the output RBFs for recognizing the full ASCII set.

$$E(W) = \frac{1}{P}\sum_{p=1}^{P} y_{D^p}(Z^p, W) \tag{8}$$

$$E(W) = \frac{1}{P}\sum_{p=1}^{P}(y_{D^p}(Z^p, W) + \log(e^{-j} + \sum_i e^{-y_i(Z^p, W)})) \tag{9}$$

LeNet-5

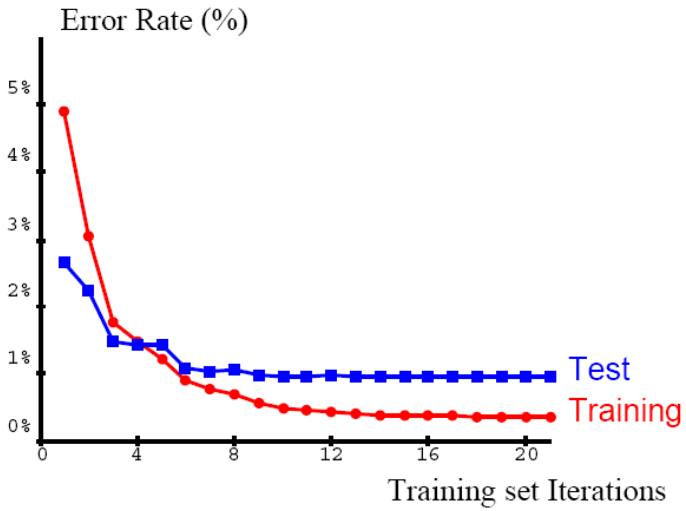


Fig. 5. Training and test error of LeNet-5 as a function of the number of passes through the 60,000 pattern training set (without distortions). The average training error is measured on-the-fly as training proceeds. This explains why the training error appears to be larger than the test error. Convergence is attained after 10 to 12 passes through the training set.

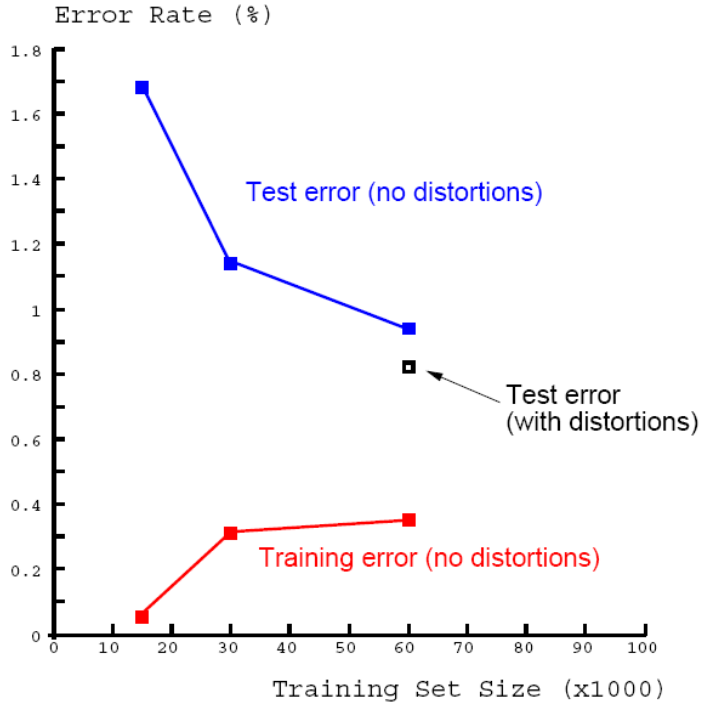


Fig. 6. Training and test errors of LeNet-5 achieved using training sets of various sizes. This graph suggests that a larger training set could improve the performance of LeNet-5. The hollow square show the test error when more training patterns are artificially generated using random distortions. The test patterns are not distorted.

LeNet-5

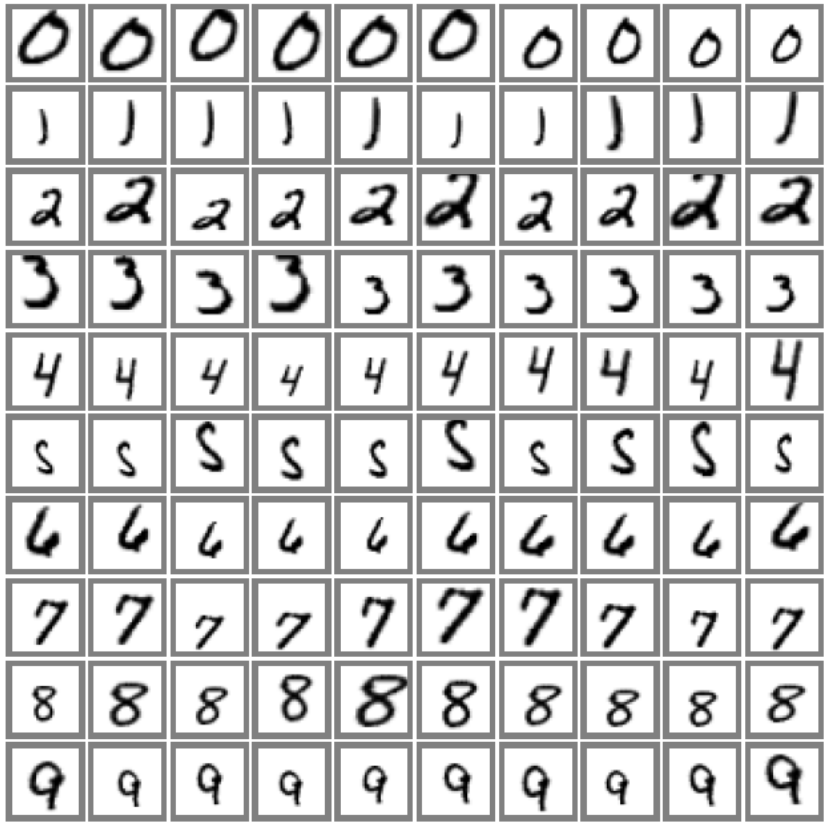


Fig. 7. Examples of distortions of ten training patterns.

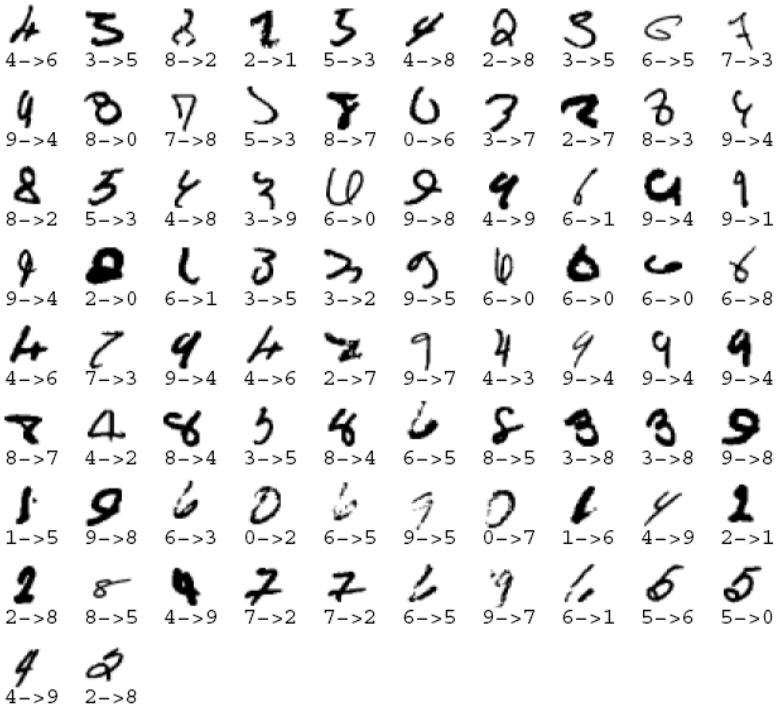


Fig. 8. The 82 test patterns misclassified by LeNet-5. Below each image is displayed the correct answers (left) and the network answer (right). These errors are mostly caused either by genuinely ambiguous patterns, or by digits written in a style that are underrepresented in the training set.

LeNet-5

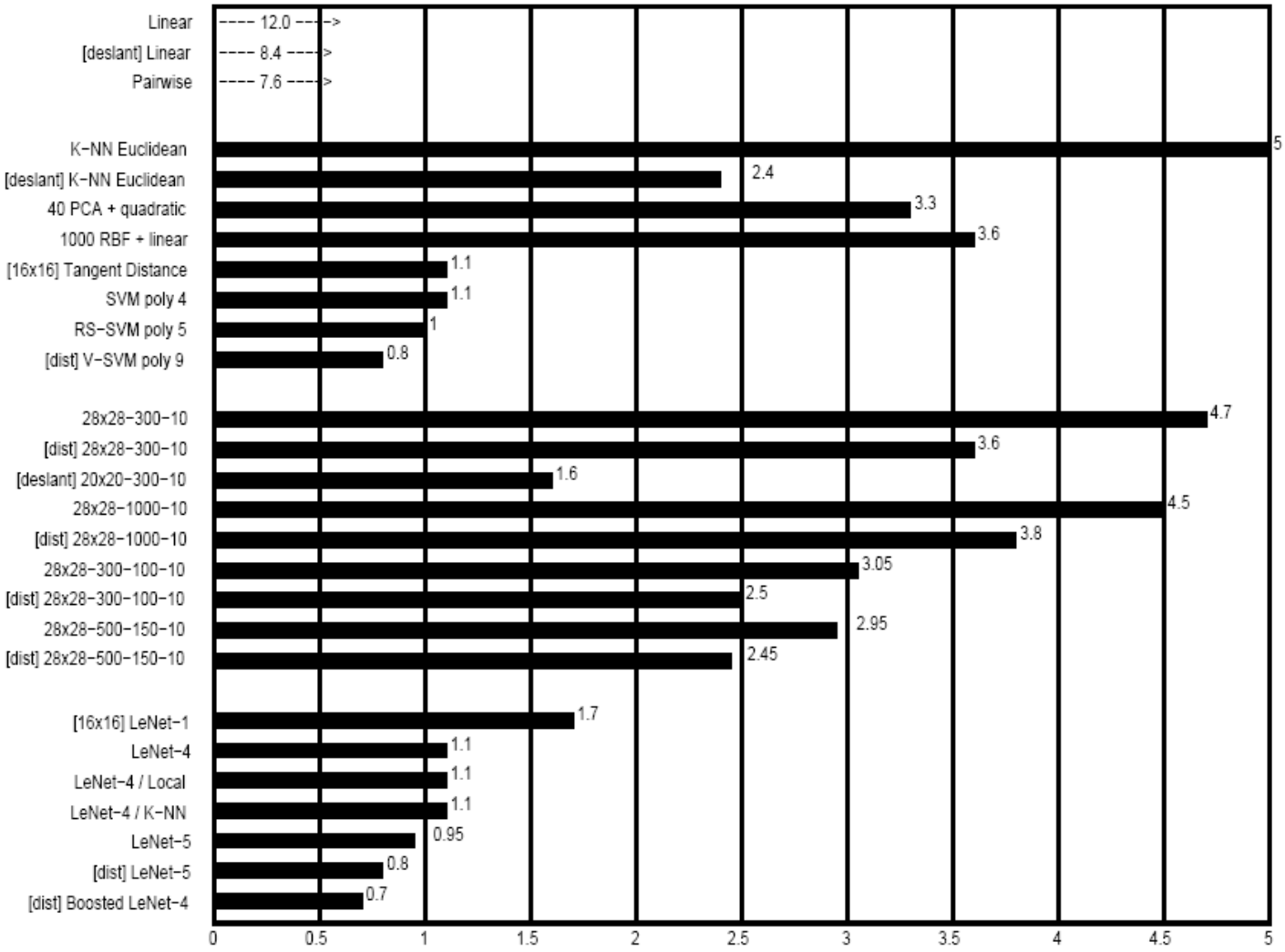


Fig. 9. Error rate on the test set (%) for various classification methods. [deslant] indicates that the classifier was trained and tested on the deslanted version of the database. [dist] indicates that the training set was augmented with artificially distorted examples. [16x16] indicates that the system used the 16x16 pixel images. The uncertainty in the quoted error rates is about 0.1%.

Limitations of convolutional networks

- State information passed between modules is fixed-size

- Can’t deal with variable-sized input, e.g.
  - Continuous speech recognition
  - Handwritten word recognition

Multi-module Systems

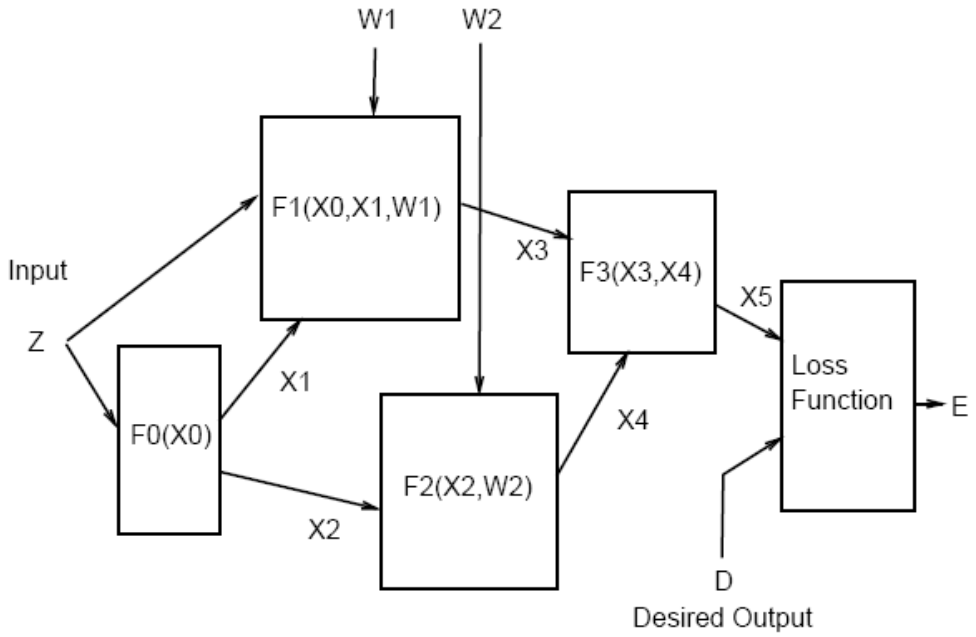


Fig. 14. A trainable system composed of heterogeneous modules.

Graph Transformer Networks (GTN)

Example:

Acoustic vectors
-> phonemes lattice
-> words lattice
-> single sequence of words (result)

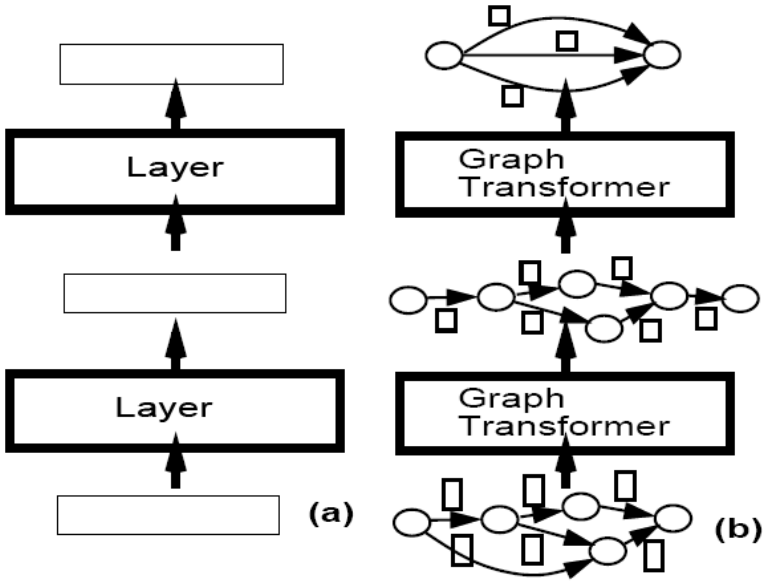


Fig. 15. Traditional neural networks, and multi-module systems communicate fixed-size vectors between layer. Multi-Layer Graph Transformer Networks are composed of trainable modules that operate on and produce graphs whose arcs carry numerical information.

GTN: Word Segmentation

- Candidate cuts at
  - Minima in vertical projection profile
  - Minima of the distance between upper & lower contours

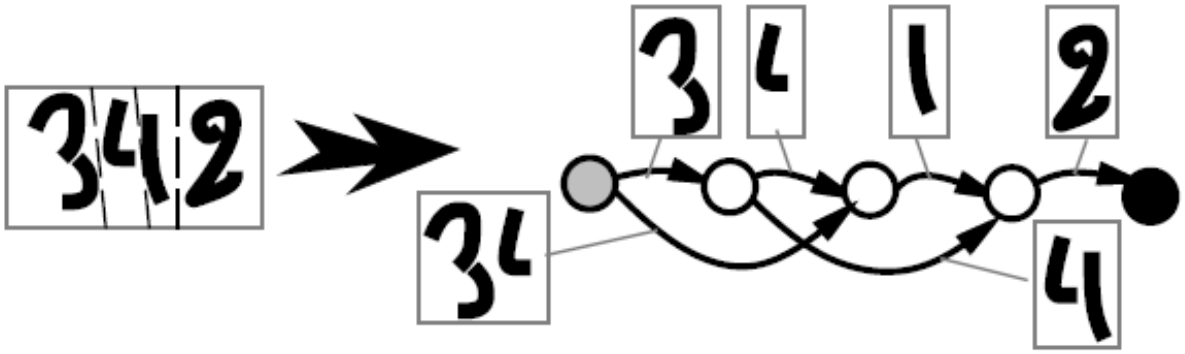


Fig. 16. Building a segmentation graph with Heuristic Over-Segmentation.

GTN: Recognition Transformer

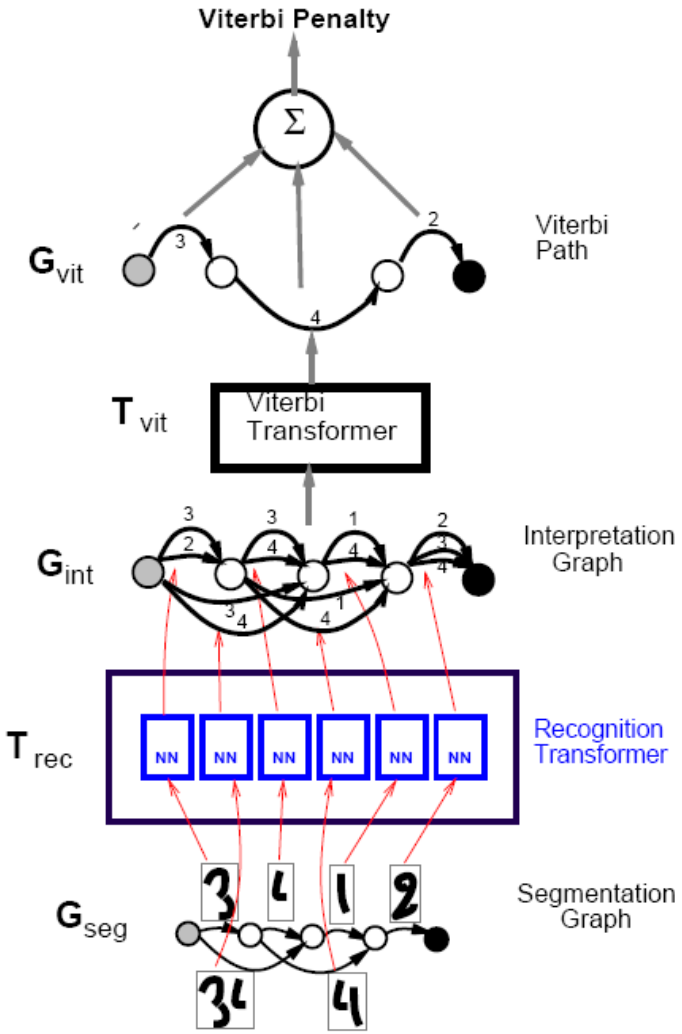


Fig. 17. Recognizing a character string with a GTN. For readability, only the arcs with low penalties are shown.

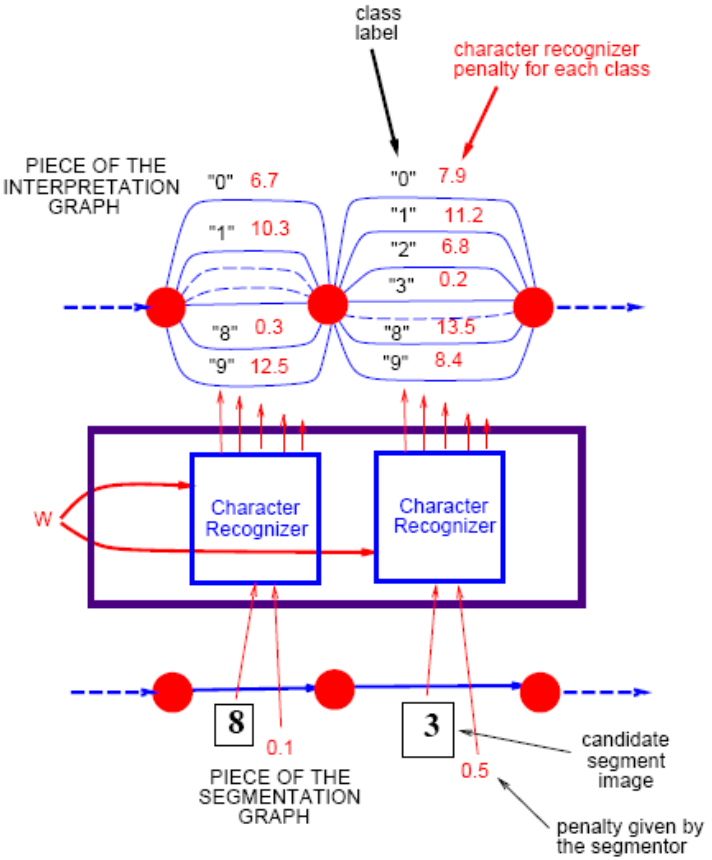


Fig. 18. The recognition transformer refines each arc of the segmentation arc into a set of arcs in the interpretation graph, one per character class, with attached penalties and labels.

GTN: Viterbi algorithm

$$v_n = \min_{i \in U_n} (c_i + v_{s_i}). \qquad (10)$$

v_n: viterbi penalty

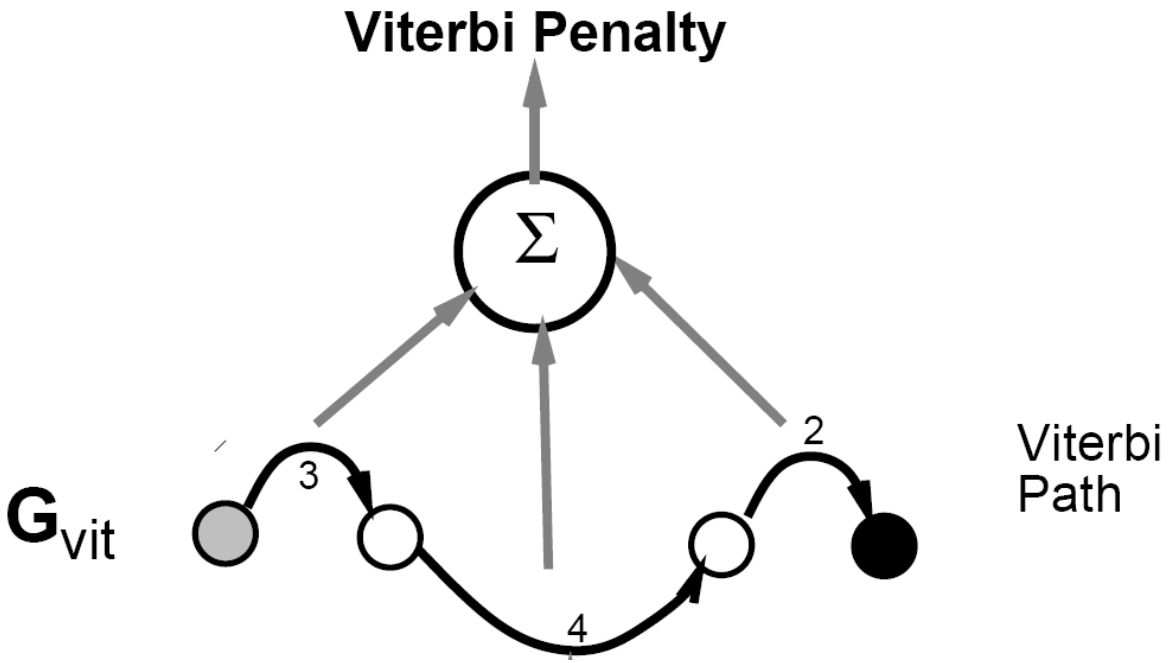
$v_{start} = 0$

c_i = penalty association w/arc i

U_n: {set of arcs w/destination n}

GTN: Viterbi training

- Minimize:
  - Penalty of the path of *the correct sequence*



Space Displacement Neural Network (SDNN)

- Alternative to Heuristic Over-Segmentation

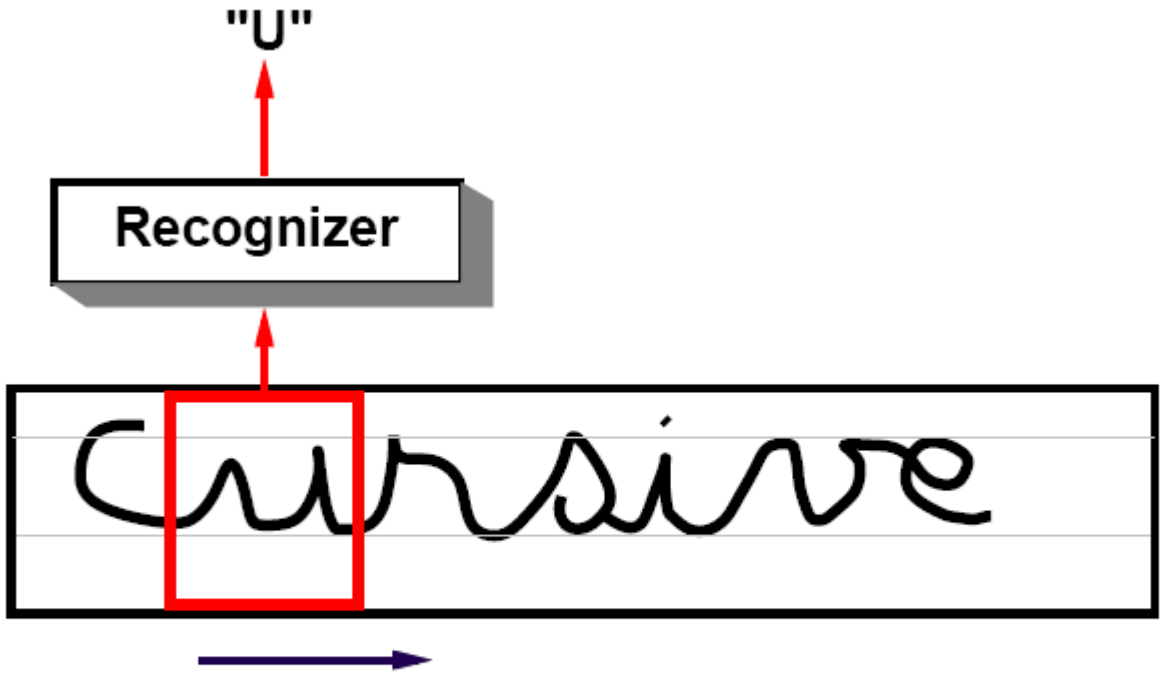


Fig. 22. Explicit segmentation can be avoided by sweeping a recognizer at every possible location in the input field.

SDNN

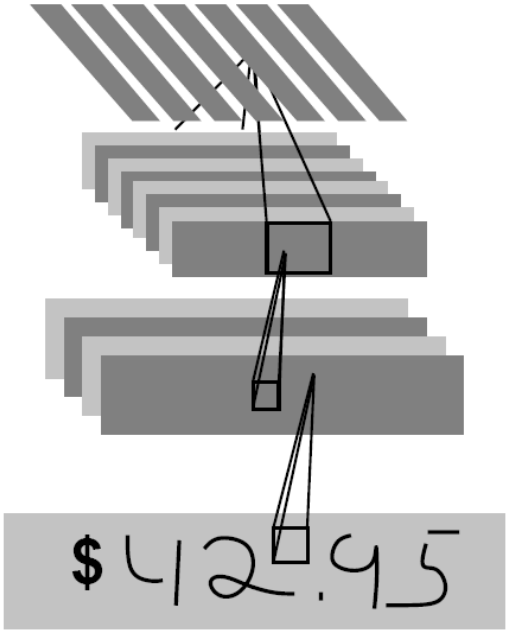


Fig. 23. A Space Displacement Neural Network is a convolutional network that has been replicated over a wide input field.

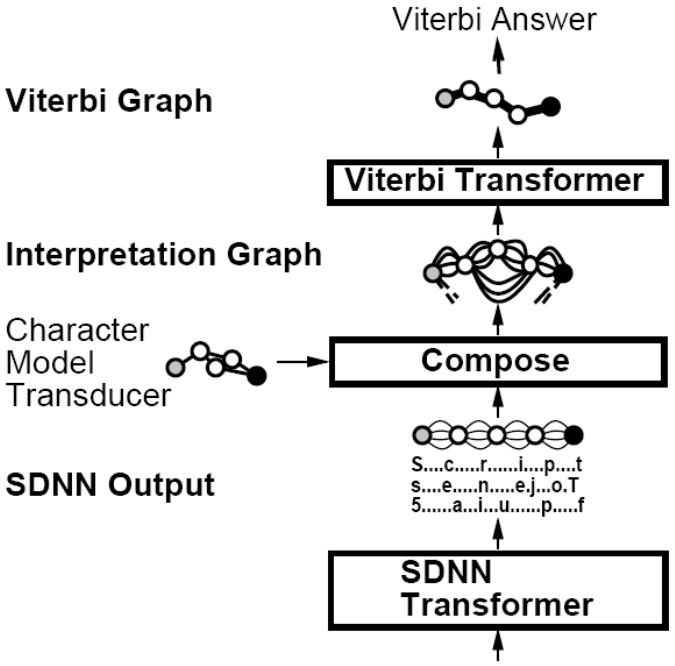


Fig. 24. A Graph Transformer pulls out the best interpretation from the output of the SDNN.

SDNN

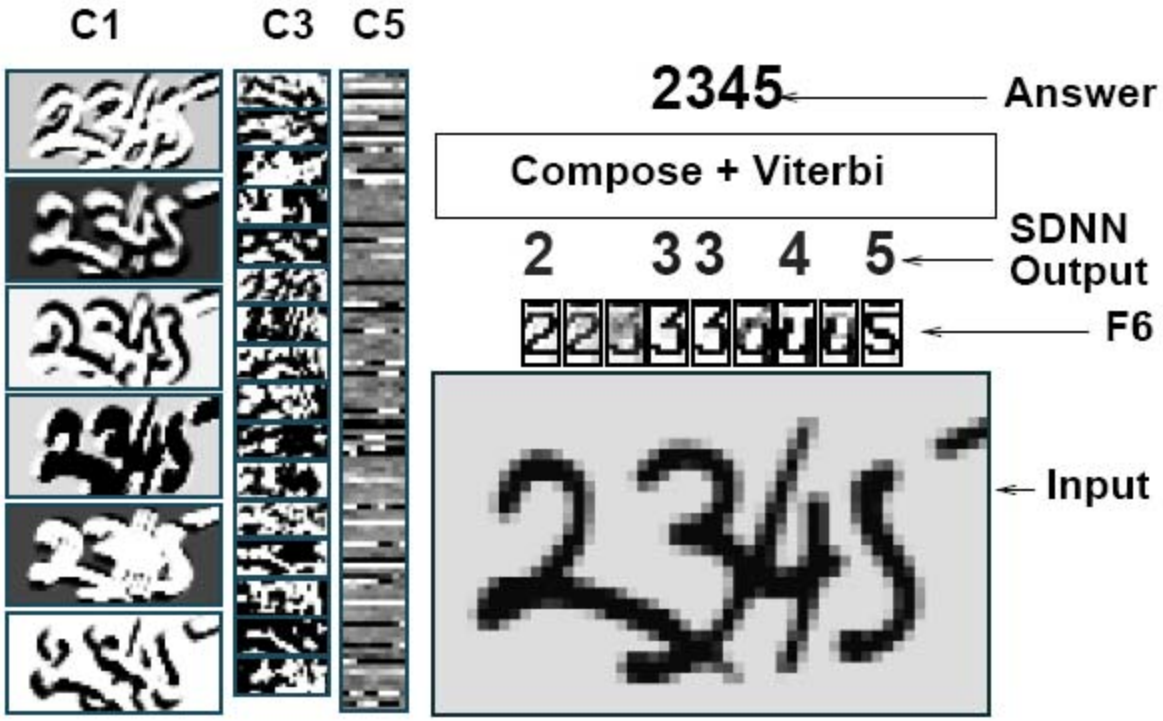


Fig. 25. An example of multiple character recognition with SDNN. With SDNN, no explicit segmentation is performed.

SDNN

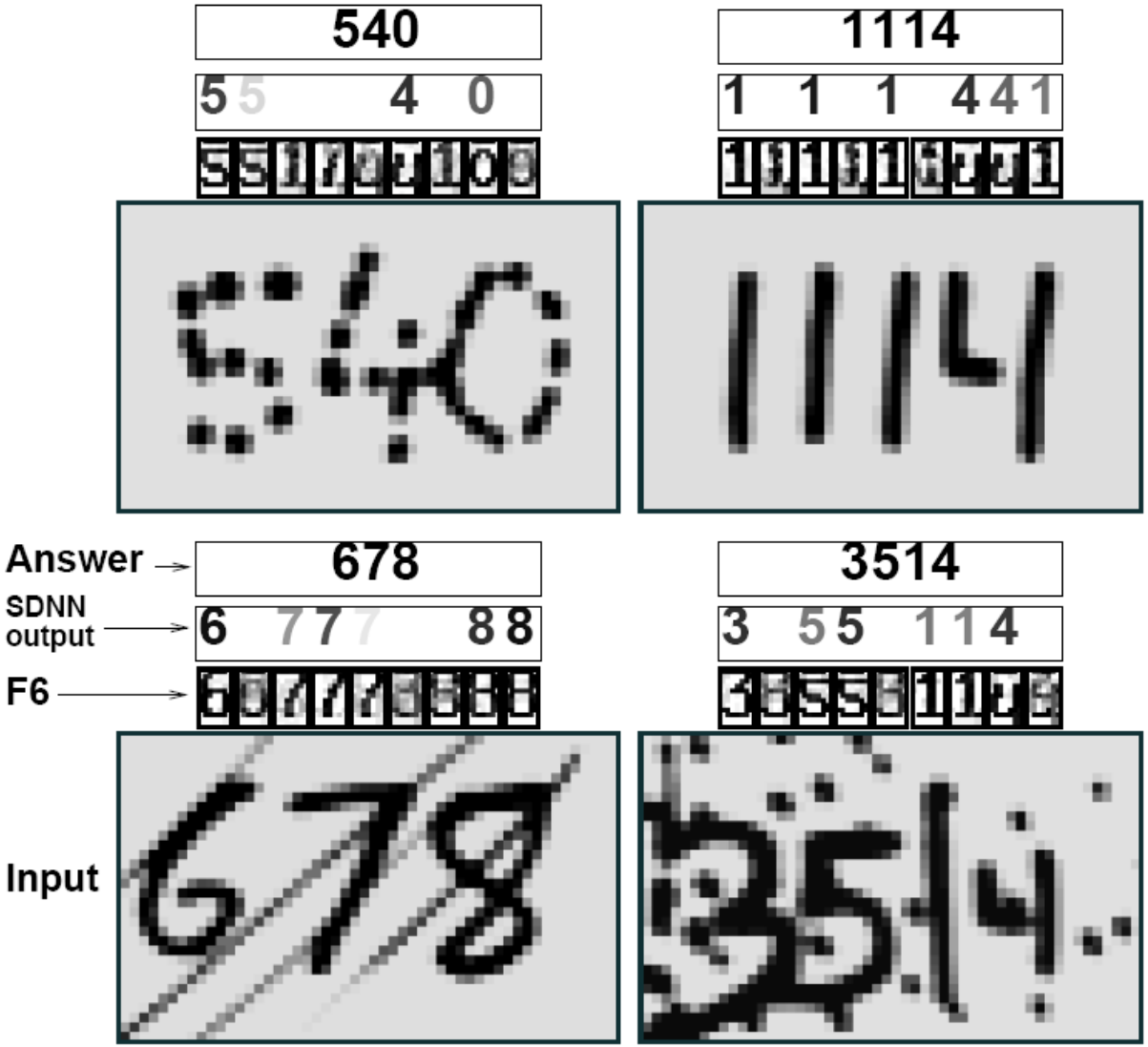


Fig. 26. An SDNN applied to a noisy image of digit string. The digits shown in the SDNN output represent the winning class labels, with a lighter grey level for high-penalty answers.

AMAP

- On-line handwriting recognition
- “annotated image”
- Each pixel is a 5-element feature vector
  - 4 features associated with 4 orientations of the pen trajectory
  - 1 feature associated with local curvature in area near pixel

AMAP

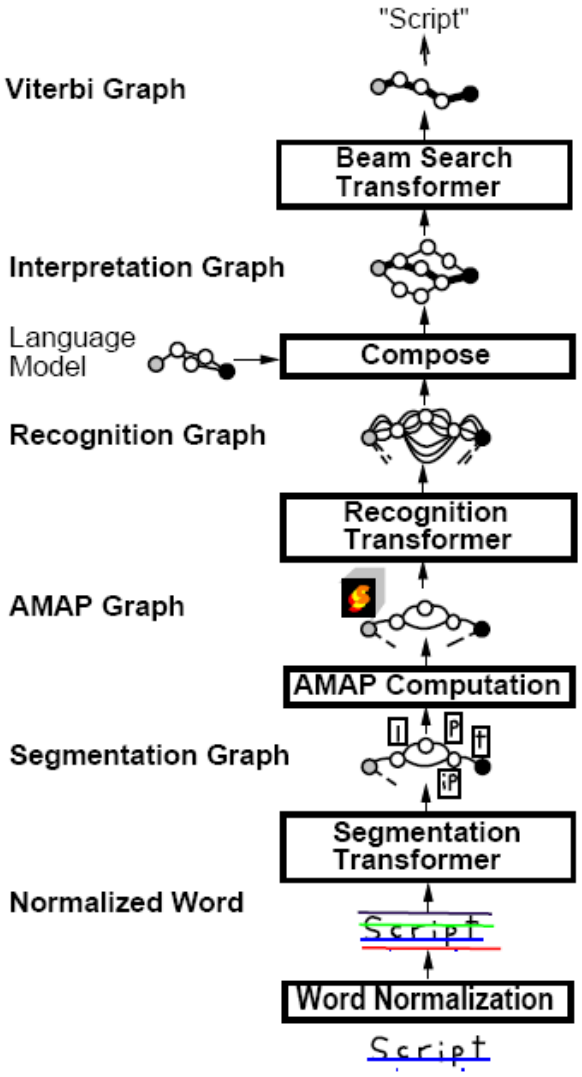


Fig. 30. An on-line handwriting recognition GTN based on heuristic over-segmentation

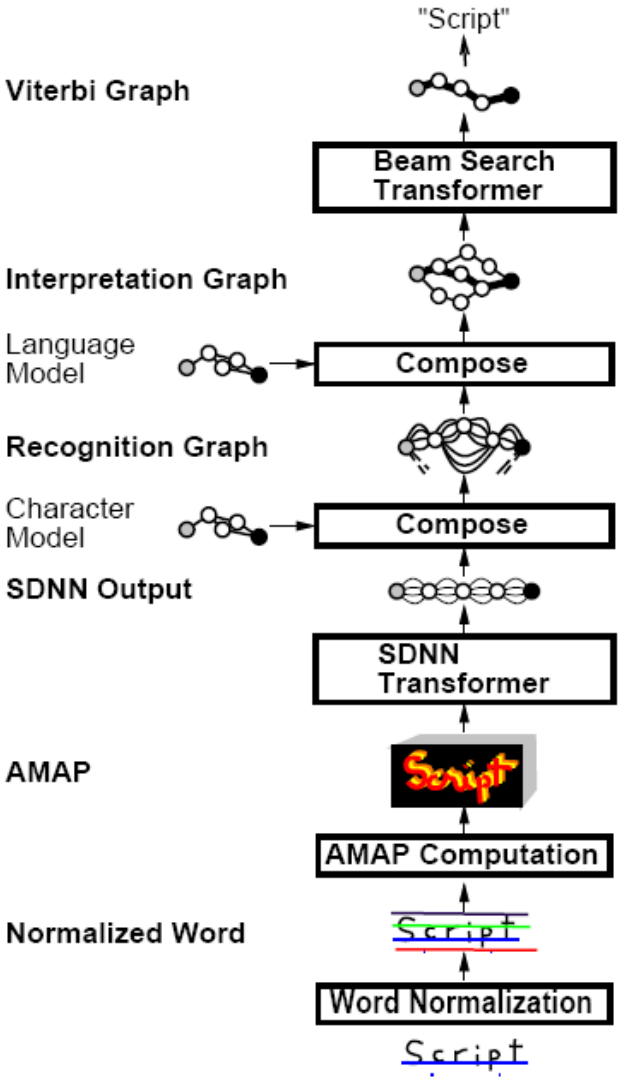


Fig. 31. An on-line handwriting recognition GTN based on Space-Displacement Neural Network

AMAP

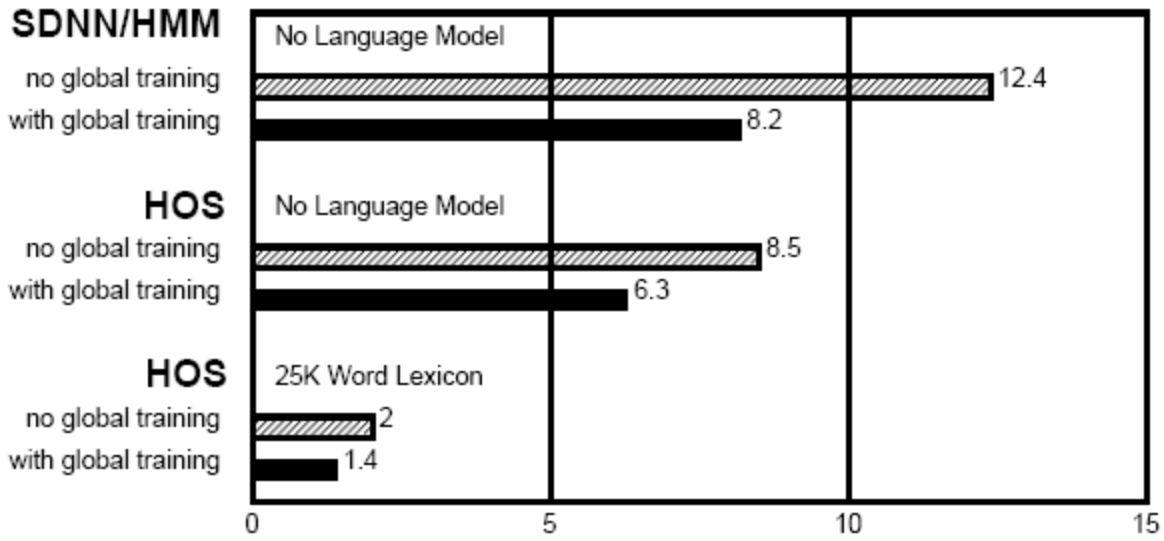


Fig. 32. Comparative results (character error rates) showing the improvement brought by global training on the SDNN/HMM hybrid, and on the Heuristic Over-Segmentation system (HOS), without and with a 25461 words dictionary.

GTN with Bank Checks

- Multi-module system
- “50% correct /
  49% reject /
  1% error” criterion
- Reject low-confidence outputs

- Results:
  82/17/1

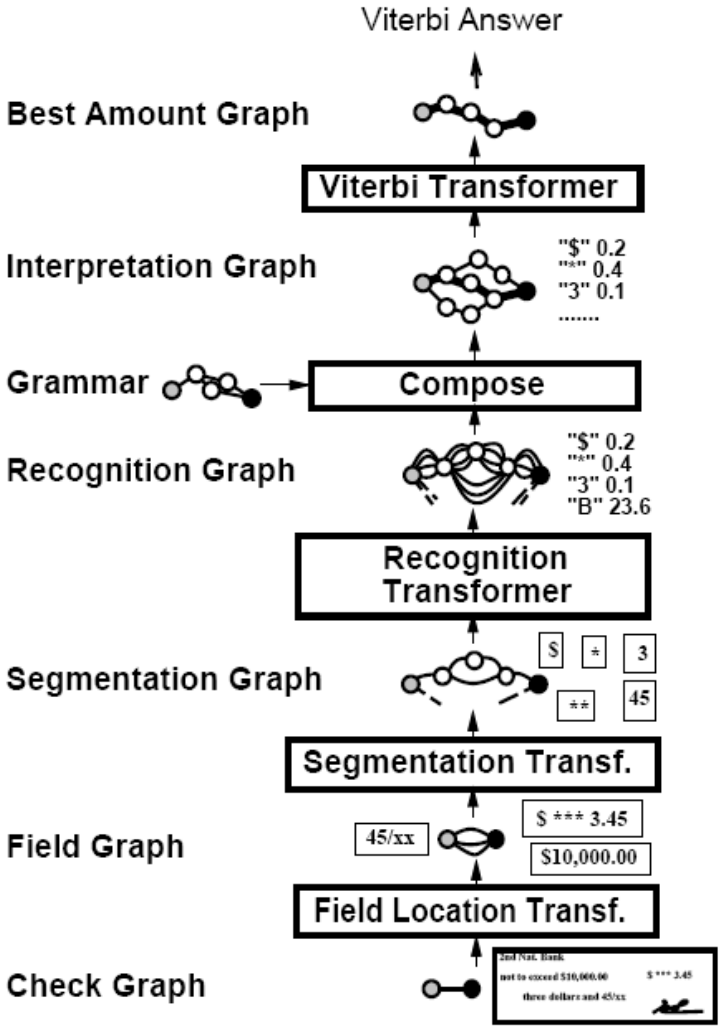


Fig. 33. A complete check amount reader implemented as a single cascade of Graph Transformer modules. Successive graph transformations progressively extract higher level information.

Conclusion: global training = good

- Segmentation and recognition modules shouldn't learn independently
- It's easier to make datasets for globally trained systems
- Globally trained systems perform better