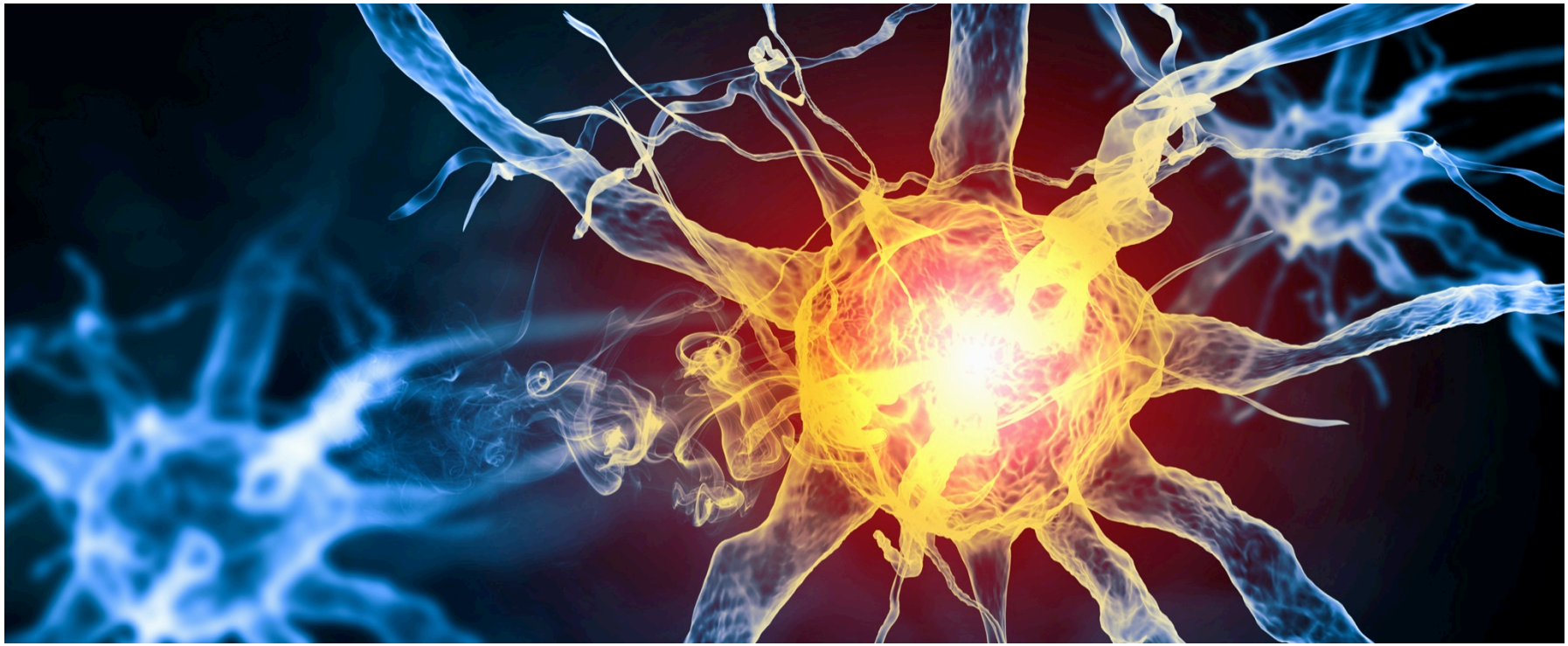
Zurich University
of Applied Sciences

Learning End-to-End

Thilo Stadelmann & Thierry Musy

Datalab Seminar, 24. June 2015, Winterthur



datalab
dlab.zhaw.ch

Zurich Universities of Applied Sciences and Arts

DISCLAIMER

Zurich University of Applied Sciences

zh aw

Today's talk is about the basic ideas of a single, inspiring, industry-proven paper from the nineties…

Agenda

Challenge
The bigger picture of ML – Sequential Supervised Learning as an example where simplistic paradigms fail

Proposed Solution
Global learning – Graph Transformer Networks – Example: Digit string recognition with heuristic oversegmentation – Advantage of discriminative training

Use Case
Error propagation through multiple modules – Check reading application

Conclusions
Summary – Outlook to other projects – Remarks

Zurich University
of Applied Sciences

zh
aw

CHALLENGE: SEQUENTIAL SUPERVISED LEARNING

Machine Learning

Wikipedia on «Learning», 2015:

> «...the act of **acquiring** new, **or modifying** and reinforcing, existing **knowledge**, behaviors, **skills**, **values**, or preferences and **may involve synthesizing** different types of information.»

A. Samuel, 1959:

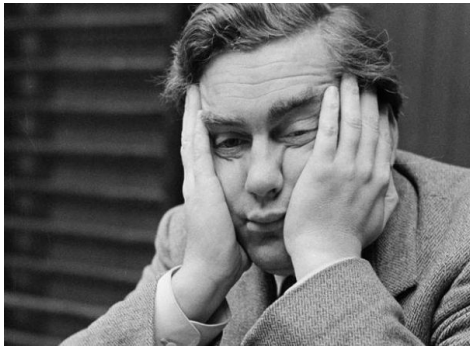
"do something"

> «…gives computers the **ability** to learn **without being explicitly programmed**.»

T.M. Mitchell, 1997:

> «…if its **performance** at tasks in T, as measured by P, **improves with experience** E.»

➔ In practice: Fitting parameters of a function to a set of data.
(data usually handcrafted, function chosen heuristically)



A landmark work in the direction of «learning»

LeCun et al., “Gradient-Based Learning Applied to Document Recognition”, 1998

PROC. OF THE IEEE, NOVEMBER 1998 1

Gradient-Based Learning Applied to Document Recognition -> LeNet 5

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

Abstract— **Multilayer Neural Networks trained with the backpropagation algorithm constitute the best example of a successful Gradient-Based Learning technique. Given an appropriate network architecture, Gradient-Based Learning algorithms can be used to synthesize a complex decision surface that can classify high-dimensional patterns such as handwritten characters, with minimal preprocessing. This paper reviews various methods applied to handwritten character recognition and compares them on a standard handwritten digit recognition task. Convolutional Neural Networks, that are specifically designed to deal with the variability of 2D shapes, are shown to outperform all other techniques.**

Real-life document recognition systems are composed of multiple modules including field extraction, segmentation, recognition, and language modeling. A new learning paradigm, called Graph Transformer Networks (GTN), allows such multi-module systems to be trained globally using Gradient-Based methods so as to minimize an overall performance measure.

Two systems for on-line handwriting recognition are described. Experiments demonstrate the advantage of global training, and the flexibility of Graph Transformer Networks.

A Graph Transformer Network for reading bank check is also described. It uses Convolutional Neural Network character recognizers combined with global training techniques to provides record accuracy on business and personal checks. It is deployed commercially and reads several million checks per day.

Keywords— **Neural Networks, OCR, Document Recognition, Machine Learning, Gradient-Based Learning, Convolutional Neural Networks, Graph Transformer Networks, Finite State Transducers.**

NOMENCLATURE

- GT Graph transformer.
- GTN Graph transformer network.
- HMM Hidden Markov model.
- HOS Heuristic oversegmentation.
- K-NN K-nearest neighbor.
- NN Neural network.
- OCR Optical character recognition.
- PCA Principal component analysis.
- RBF Radial basis function.
- RS-SVM Reduced-set support vector method.
- SDNN Space displacement neural network.
- SVM Support vector method.
- TDNN Time delay neural network.
- V-SVM Virtual support vector method.

The authors are with the Speech and Image Processing Services Research Laboratory, AT&T Labs-Research, 100 Schulz Drive Red Bank, NJ 07701. E-mail: {yann,leonb,yoshua,haffner}@research.att.com. Yoshua Bengio is also with the Département d'Informatique et de Recherche Opérationelle, Université de Montréal, C.P. 6128 Succ. Centre-Ville, 2920 Chemin de la Tour, Montréal, Québec, Canada H3C 3J7.

I. INTRODUCTION

Over the last several years, machine learning techniques, particularly when applied to neural networks, have played an increasingly important role in the design of pattern recognition systems. In fact, it could be argued that the availability of learning techniques has been a crucial factor in the recent success of pattern recognition applications such as continuous speech recognition and handwriting recognition.

The main message of this paper is that better pattern recognition systems can be built by relying more on automatic learning, and less on hand-designed heuristics. This is made possible by recent progress in machine learning and computer technology. Using character recognition as a case study, we show that hand-crafted feature extraction can be advantageously replaced by carefully designed learning machines that operate directly on pixel images. Using document understanding as a case study, we show that the traditional way of building recognition systems by manually integrating individually designed modules can be replaced by a unified and well-principled design paradigm, called *Graph Transformer Networks*, that allows training all the modules to optimize a global performance criterion.

Since the early days of pattern recognition it has been known that the variability and richness of natural data, be it speech, glyphs, or other types of patterns, make it almost impossible to build an accurate recognition system entirely by hand. Consequently, most pattern recognition systems are built using a combination of automatic learning techniques and hand-crafted algorithms. The usual method of recognizing individual patterns consists in dividing the system into two main modules shown in figure 1. The first module, called the feature extractor, transforms the input patterns so that they can be represented by low-dimensional vectors or short strings of symbols that (a) can be easily matched or compared, and (b) are relatively invariant with respect to transformations and distortions of the input patterns that do not change their nature. The feature extractor contains most of the prior knowledge and is rather specific to the task. It is also the focus of most of the design effort, because it is often entirely hand-crafted. The classifier, on the other hand, is often general-purpose and trainable. One of the main problems with this approach is that the recognition accuracy is largely determined by the ability of the designer to come up with an appropriate set of features. This turns out to be a daunting task which, unfortunately, must be redone for each new problem. A large amount of the pattern recognition literature is devoted to describing and comparing the relative

Outline

- Gradient-Based ML ✓
- Convolutional Neural Networks ✓
- Comparison with other Methods
- Multi-Module Systems & Graph Transformer Networks
- Multiple Object Recognition & Heuristic Oversegmentation
- Space Displacement Neural Networks
- GTN’s as General Transducers
- On-Line Handwriting Recognition System
- Check Reading System

Standard and Sequential Supervised Learning

Supervised Learning

Incanter Dataset

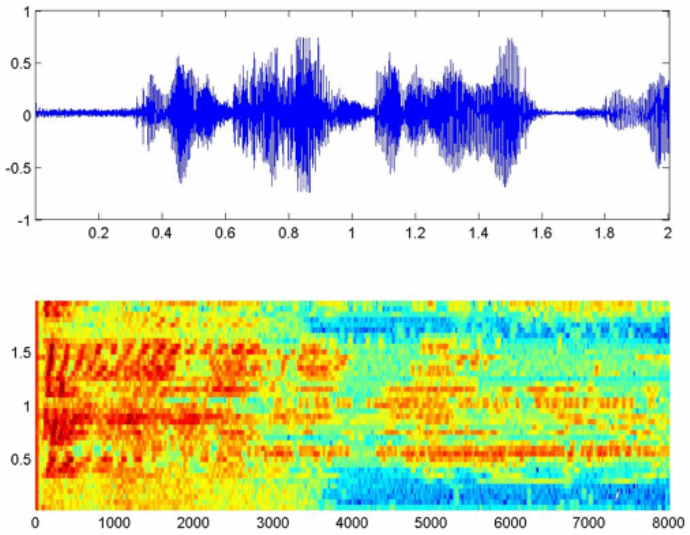
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa
5.4	3.4	1.7	0.2	setosa
5.1	3.7	1.5	0.4	setosa

feature vectors labels

Typical assumption on data:

- i.i.d.
- Surrounding tasks deemed simple(r)

Sequential Supervised Learning



Typical assumptions on data:

- Sequence information matters
- Overall task has many challenging components (e.g., segmentation → recognition → sequence assembly)

Approaches to classifying sequential data

Zurich University of Applied Sciences

zh aw

«A bird in the hand…» approach

- Train standard classifier, extend it using a sliding window and post-processing (e.g., smoothing)

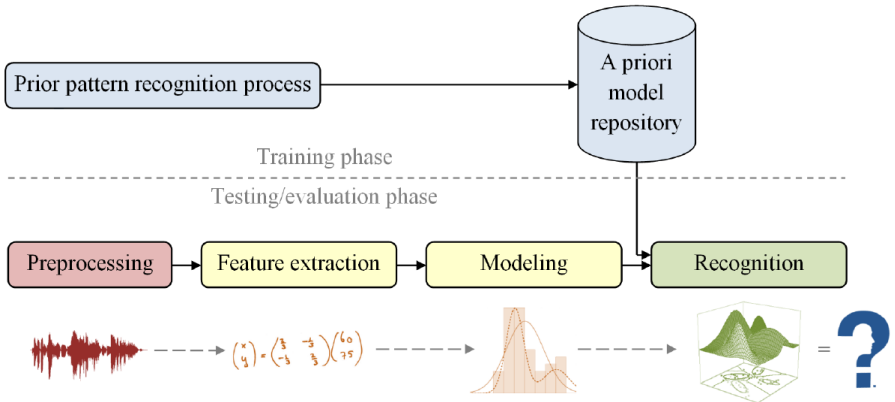
Direct modeling approach

- Train a generative (statistical) model of the sequence generation process (e.g., HMM)

«…two in the bush» approach

- Build a unified pattern recognition processing chain, optimize it globally with a unique criterion

See also: T.G. Dietterich, «Machine Learning for Sequential Data – A Review», 2002



Zurich University
of Applied Sciences

PROPOSED SOLUTION: GLOBAL LEARNING

Images sources: See references on last slide

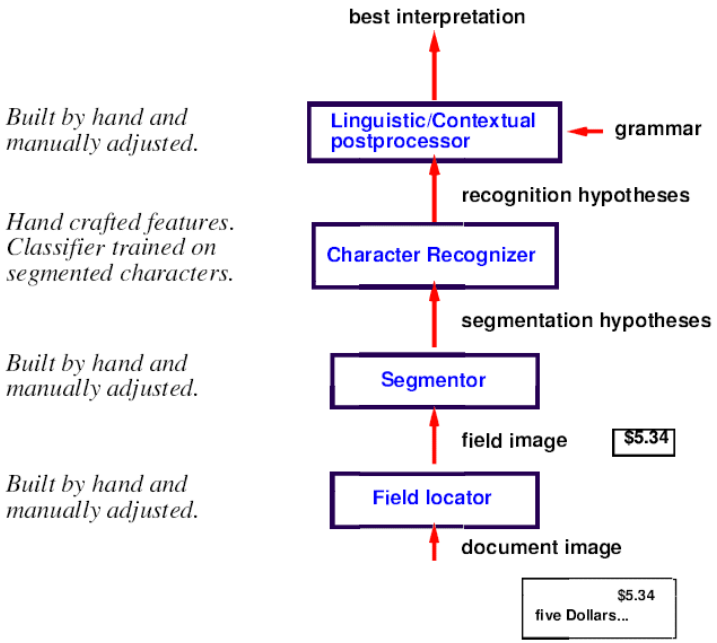
Example: Reading handwritten strings

Challenge

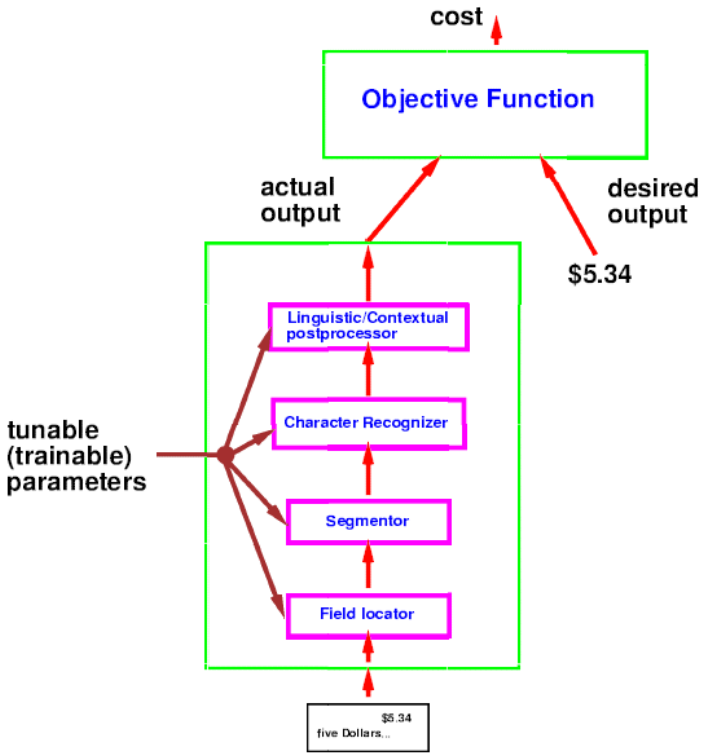
- Identify correct character string («342») on a piece of paper
- Therefore: Find correct segmentation & recognize individual characters

Global Learning
Learning end-to-end

What we know: Traditional pattern recognition system architecture

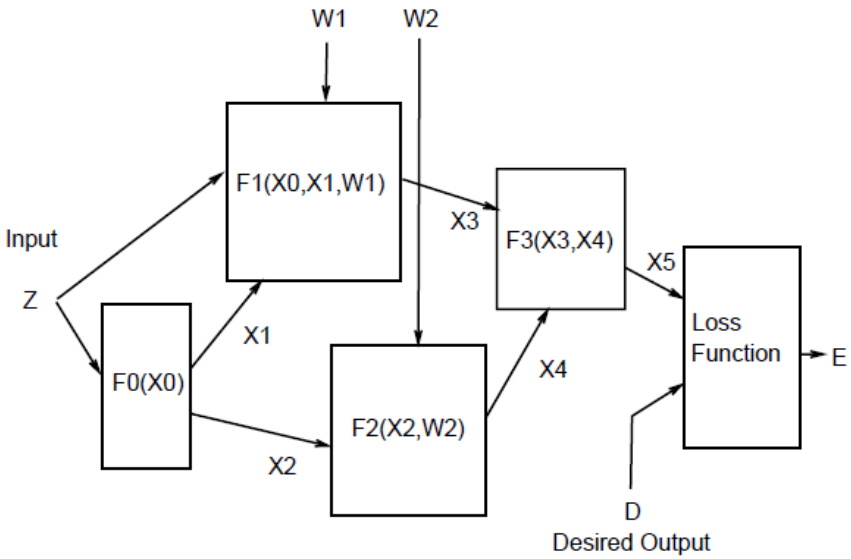


What we want: Train all parameters to optimize a global performance criterion



Foundation: Gradient-based learning

A trainable system composed of heterogeneous modules:



Backpropagation can be used if...

- cost (loss) function is differentiable w.r.t. parameters
- modules are differentiable w.r.t. parameters

➔ Gradient-based learning is the unifying concept behind many machine learning methods

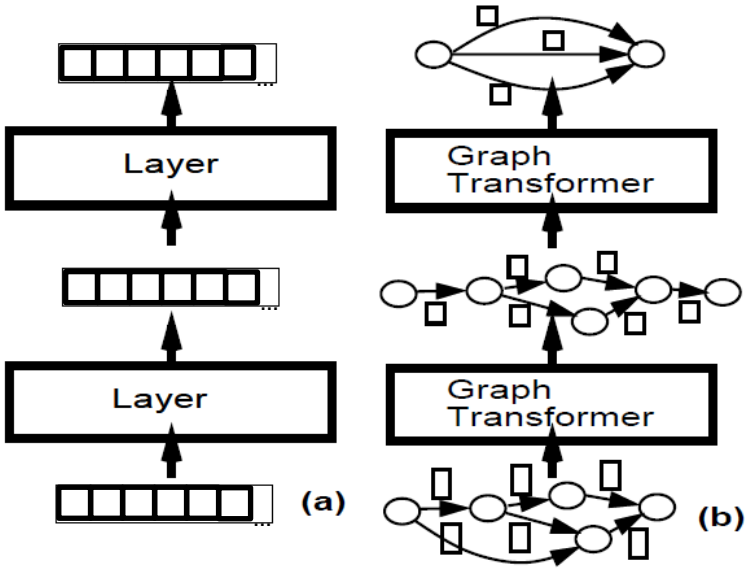
➔ Object-oriented design approach: Each module is a class with a `fprop()` and `bprop()` method

Graph Transformer Networks

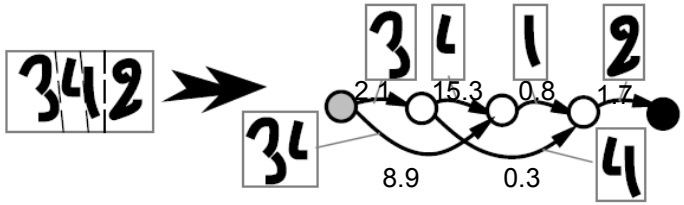
Network of pattern recognition modules that successively refine graph representations of the input

GTNs

- Operate on graphs (b) of the input instead fixed-size vectors (a)



- Graph: DAG with numerical information ("penalties") at the arcs



➔ GTN takes gradients w.r.t. module parameters and numerical data at input arcs

Example: Heuristic oversegmentation

…for reading handwritten strings

Loss Function: Average penalty (over all training data) of best (i.e., lowest penalty) correct path (i.e. associated with correct sequence)

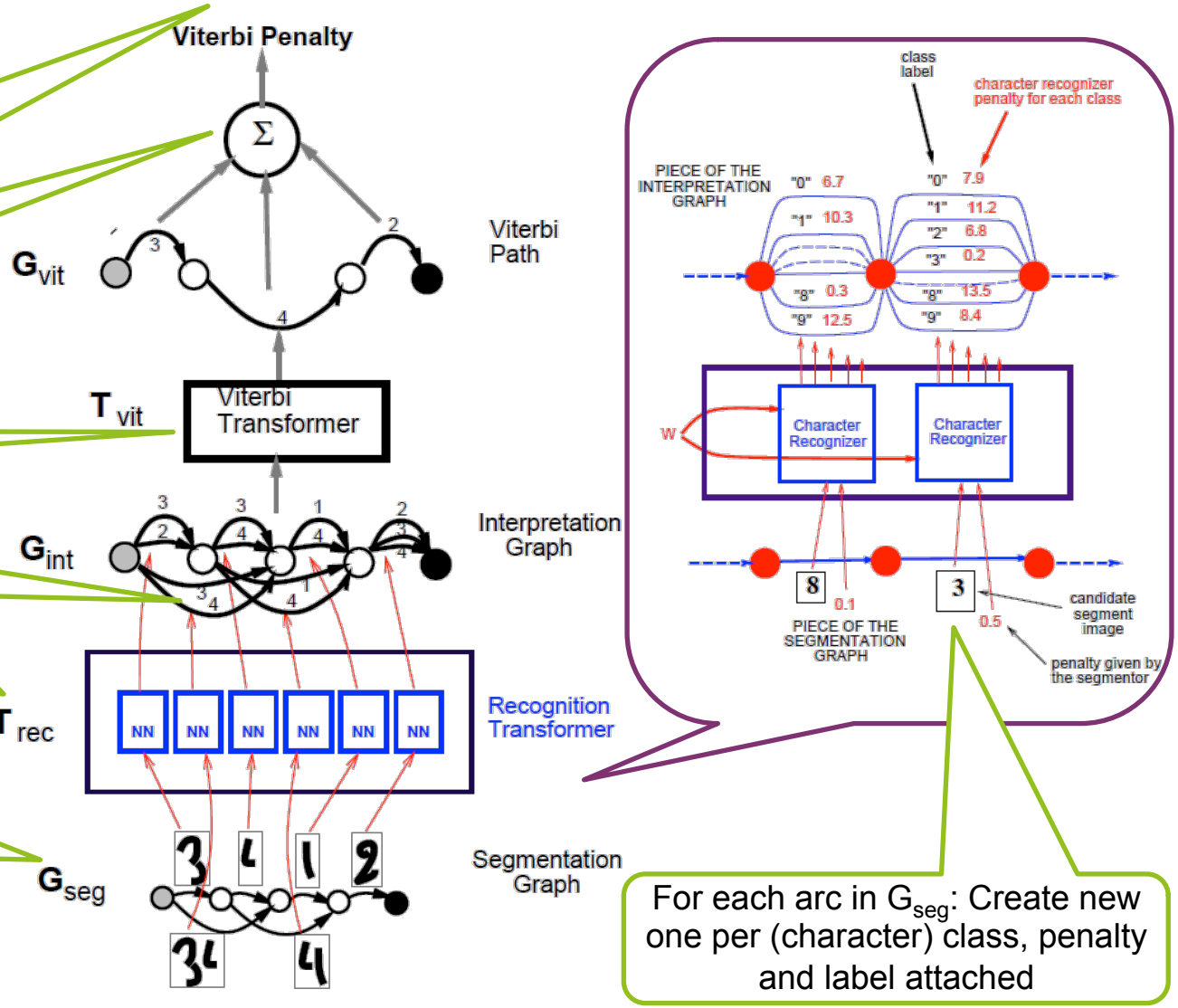
Sum up all penalties

Select path with least cum. penalty

Combined penalties
➔ handled via integrated training

Recognizes individual characters on single images (e.g. CNN)

Contains all possible segmentations (arcs: penalties & images)



For each arc in G_{seg}: Create new one per (character) class, penalty and label attached

Problems:
1. Trivial solution possible (Recognizer ignores input & sets all outputs to small values)
2. Penalty does not take competing answers into account (i.e., ignores training signals)

Solved: Discriminative training builds the class-"separating surfaces rather than modeling individual classes independently of each other" ➔ L=0 if best path is a correct path.

Zurich University of Applied Sciences

zh aw

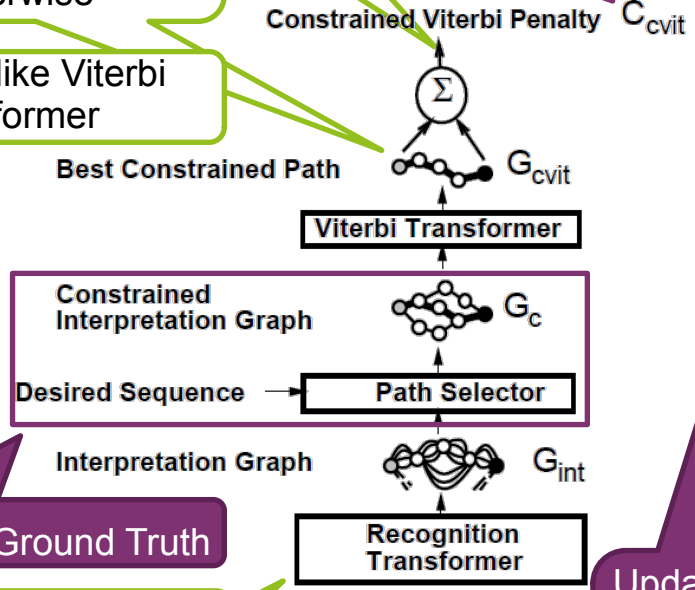
«Viterbi» training

Gradient w.r.t to loss function: 1

Loss function value: $L = C_{cvit}$

Gradient: 1 (for all arcs appearing in Viterbi path), 0 otherwise

Gradient: like Viterbi Transformer

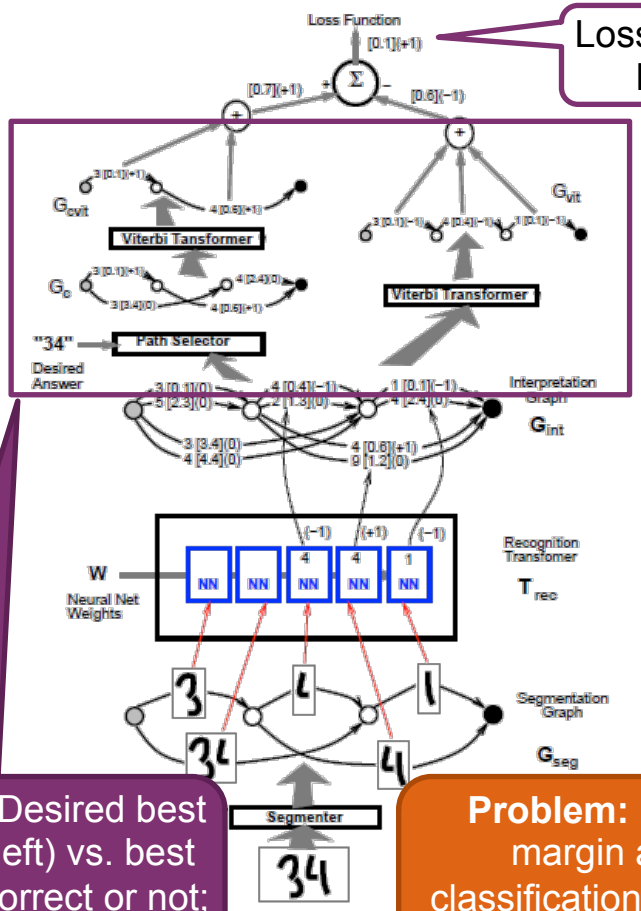


Update: Ground Truth

Gradient: Backprop for NN (one Instance per arc)

Discriminative training

Loss function value: $L = C_{cvit} - C_{vit}$



Update: Desired best output (left) vs. best output (correct or not; right)

Problem: Produces no margin as long as classification is correct (see paper for solution).

Remarks

Discriminative training

- Uses **all** available training **signals**
- Utilizes “penalties”, **not probabilities**
  ➔ **No** need for **normalization**
  ➔ Enforcing normalization is *“complex, inefficient, time consuming, ill-conditions the loss function”* [according to paper]
- Is the ***easiest/direct* way** to achieve the objective of classification (**as opposed to Generative training**, that solves the more complex density estimation task as an intermediary result)

List of possible GT modules

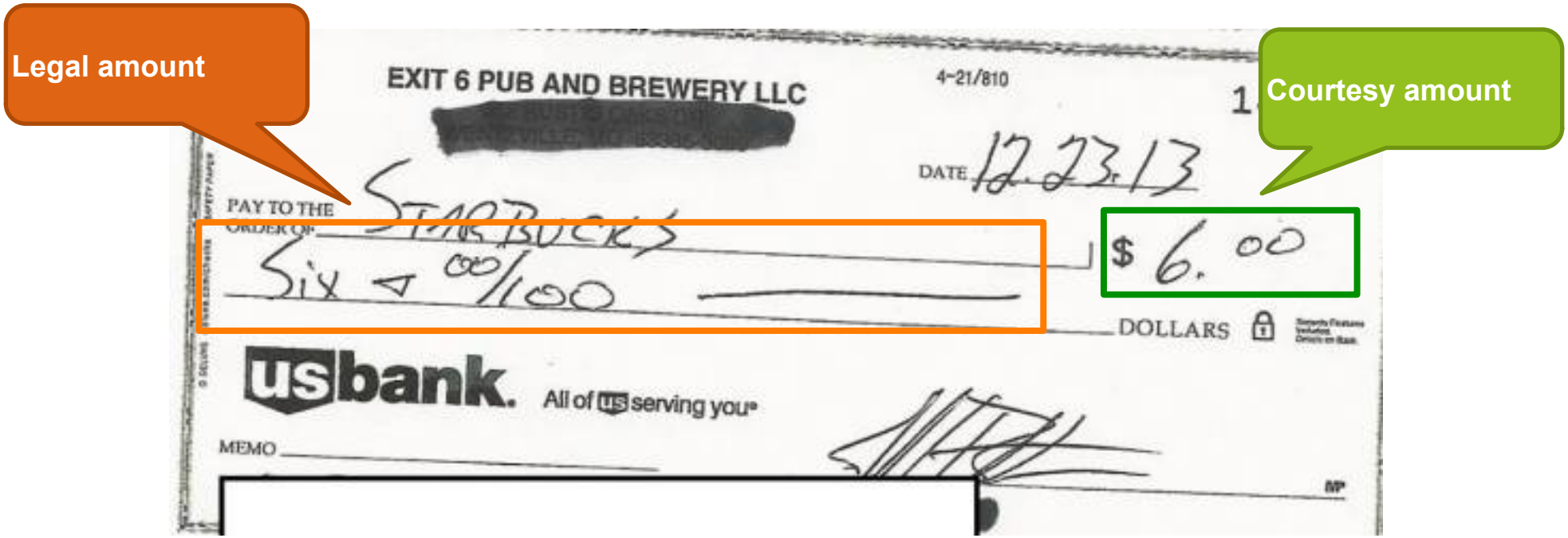
- All **building blocks of (C)NNs** (layers, nonlinearity etc.)
- **Multiplexer** (though not differentiable w.r.t. to switching input)
  → can be used to dynamically rewire GTN architecture per input
- **min**-function (though not differentiable everywhere)
- **Loss** function

Zurich University
of Applied Sciences

zh
aw

EXAMPLE: CHECK READER APPLICATION

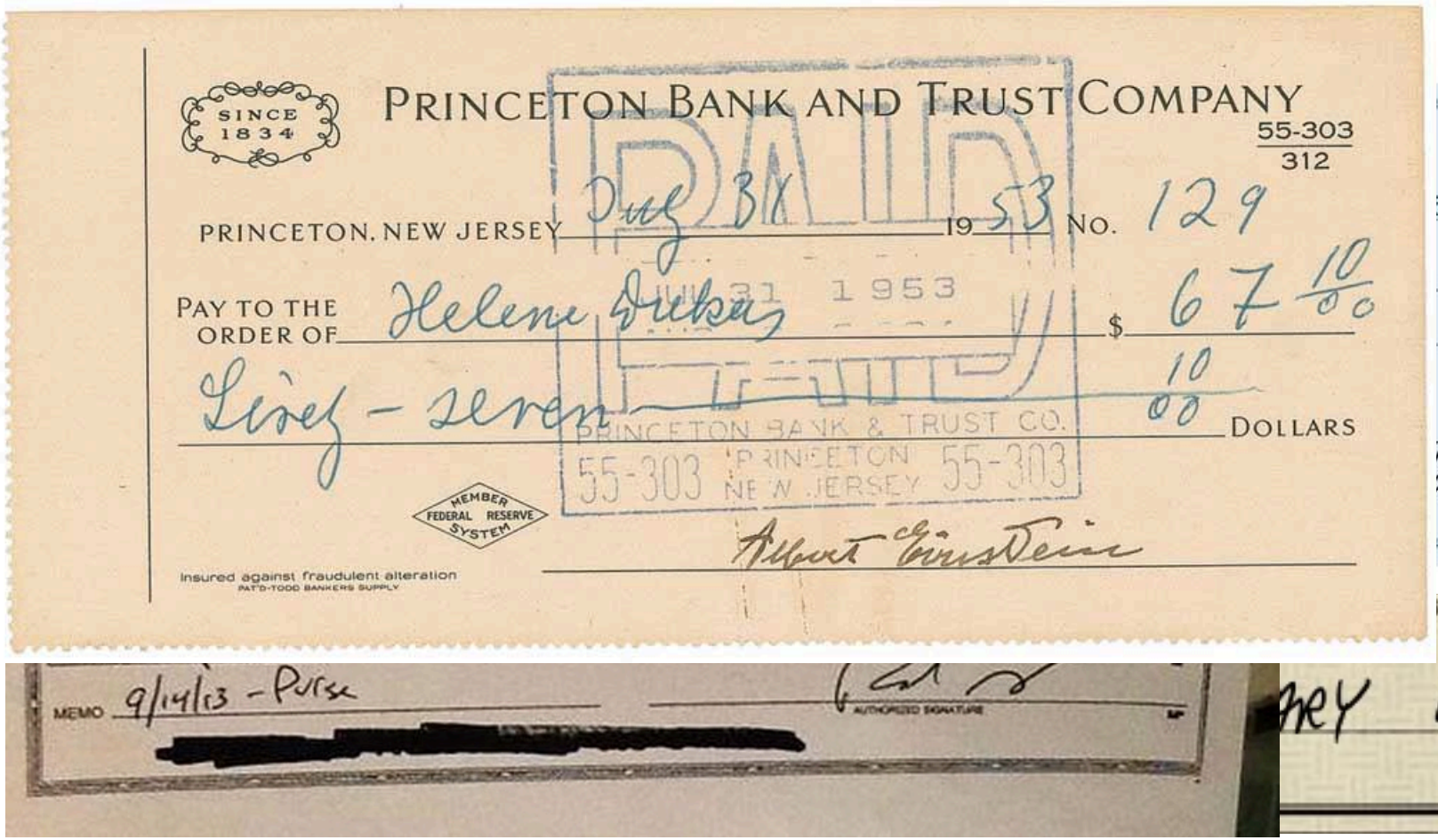
What is the amount?

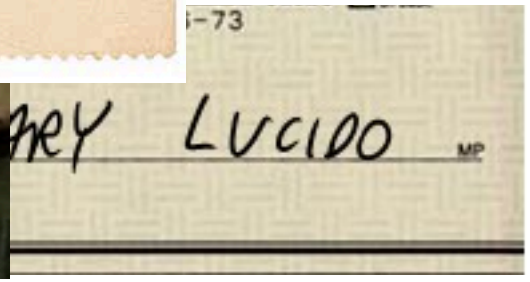


easy for human
but time consuming

interest in automating the process

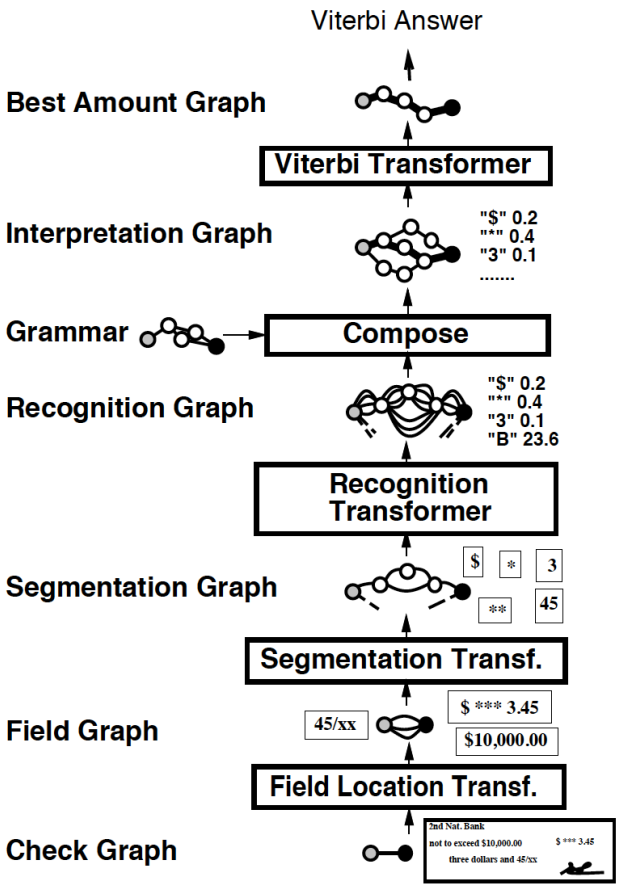
A Check Reading System





Goal: Successfully read the amount of $ 6.00

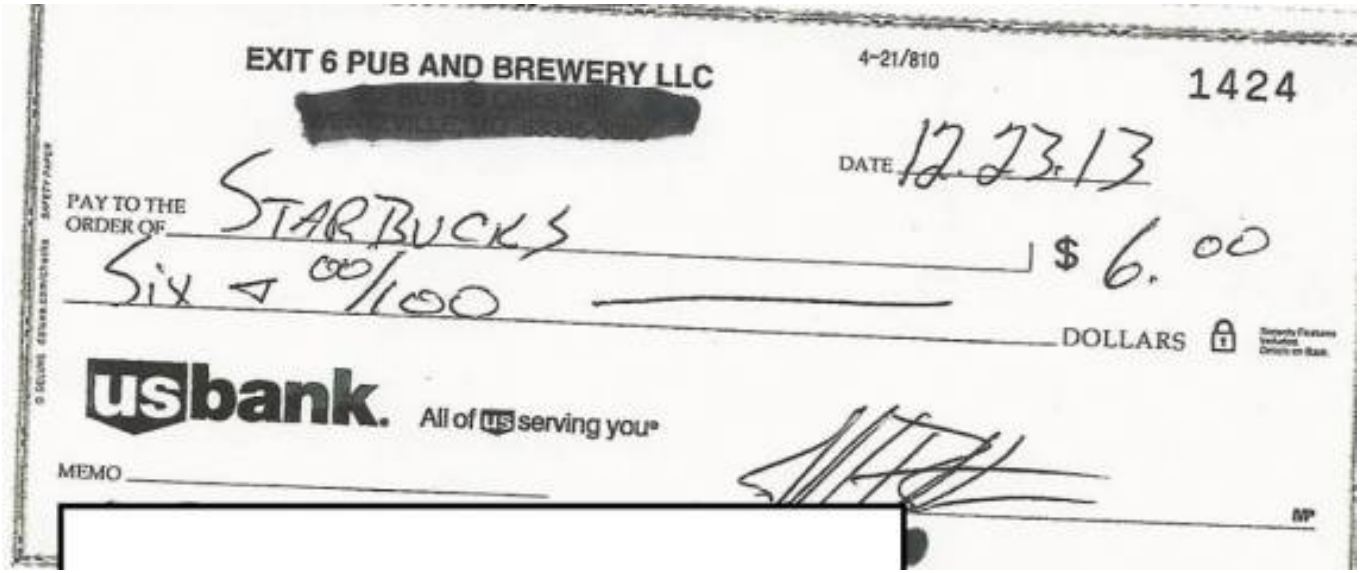
A Check Reading System

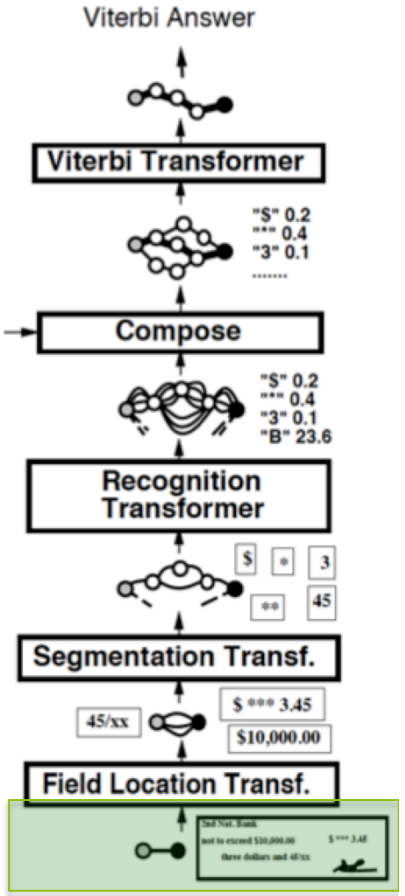


- NCR
- Reading millions of checks per day
- GTN-based
- Aimed performance:
  - 50% correct
  - 49% reject
  - 1% error

Check graph

- Input: trivial graph with a single arc that carries the image of the whole check



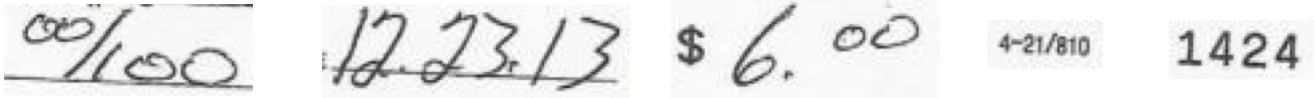


Field location

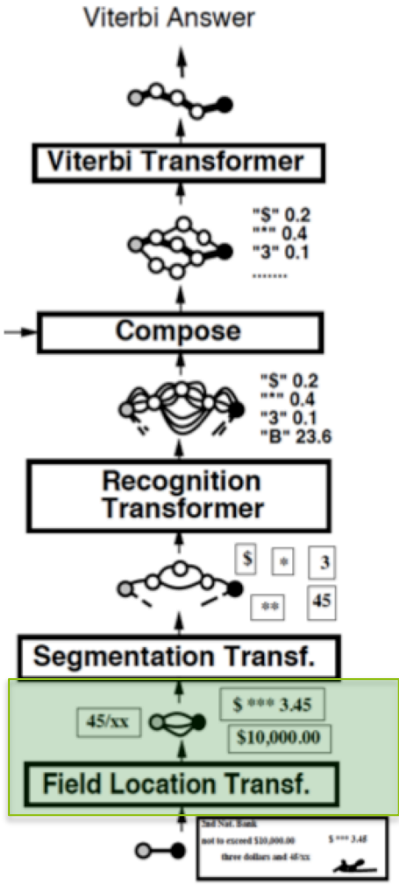
Zurich University of Applied Sciences

Transformer

- Performs classical image analysis (e.g. Connected component analysis, Ink density histogram, Layout analysis)
- Heuristically extract rectangluar zones which may content the amount.



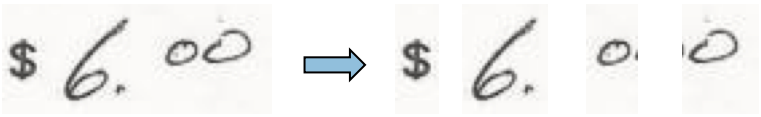
Field Graph

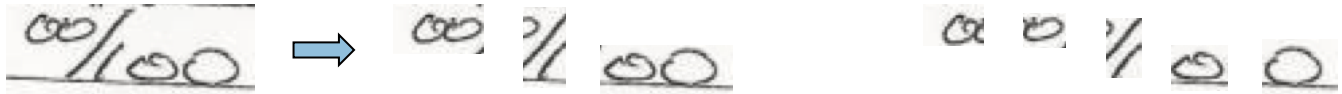


Segmentation

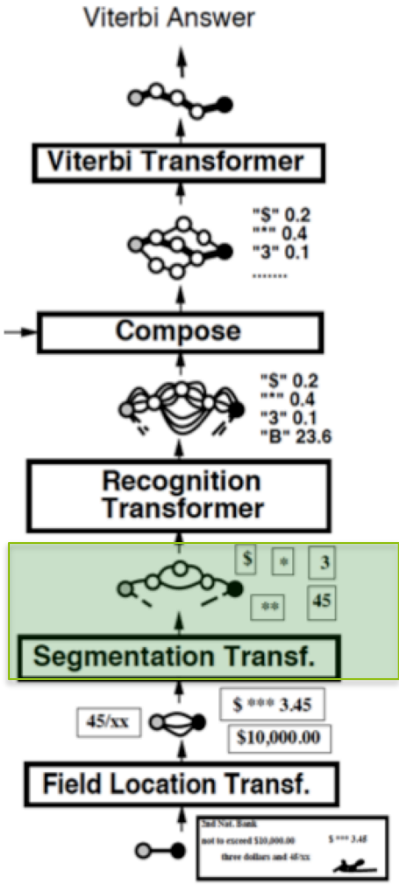
Transformer

- Performs heuristic image processing techniques (here hit an deflect) to find candidate cuts.





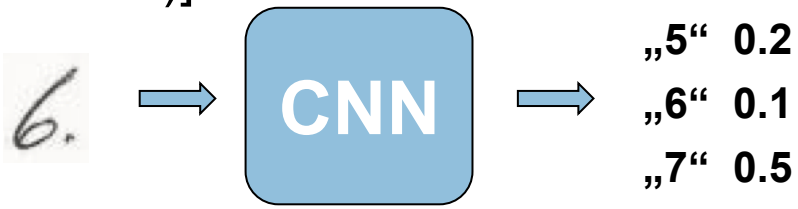
Field Graph



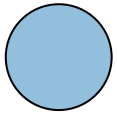
Recognition

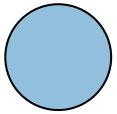
Transformer

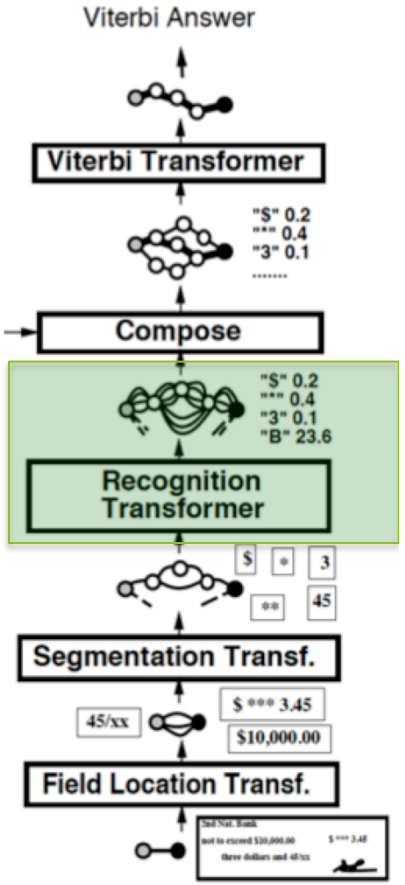
- Iterates over all arcs in the segmentation graph and runs a character recognizer [here: LeNet5 with 96 classes (ASCII: 95, rubbish: 1)]



Field Graph







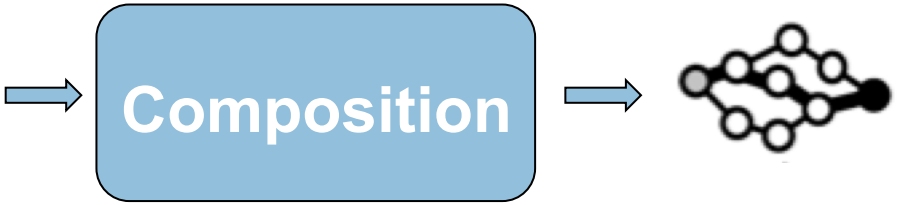
Composition

Zurich University of Applied Sciences

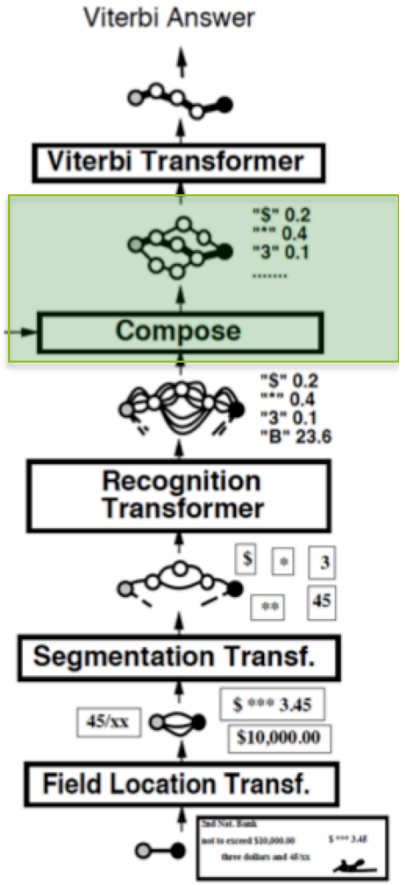
Transformer

- Takes two graphs, the recognition graph and a grammar graph, to find valid representation of check amounts.

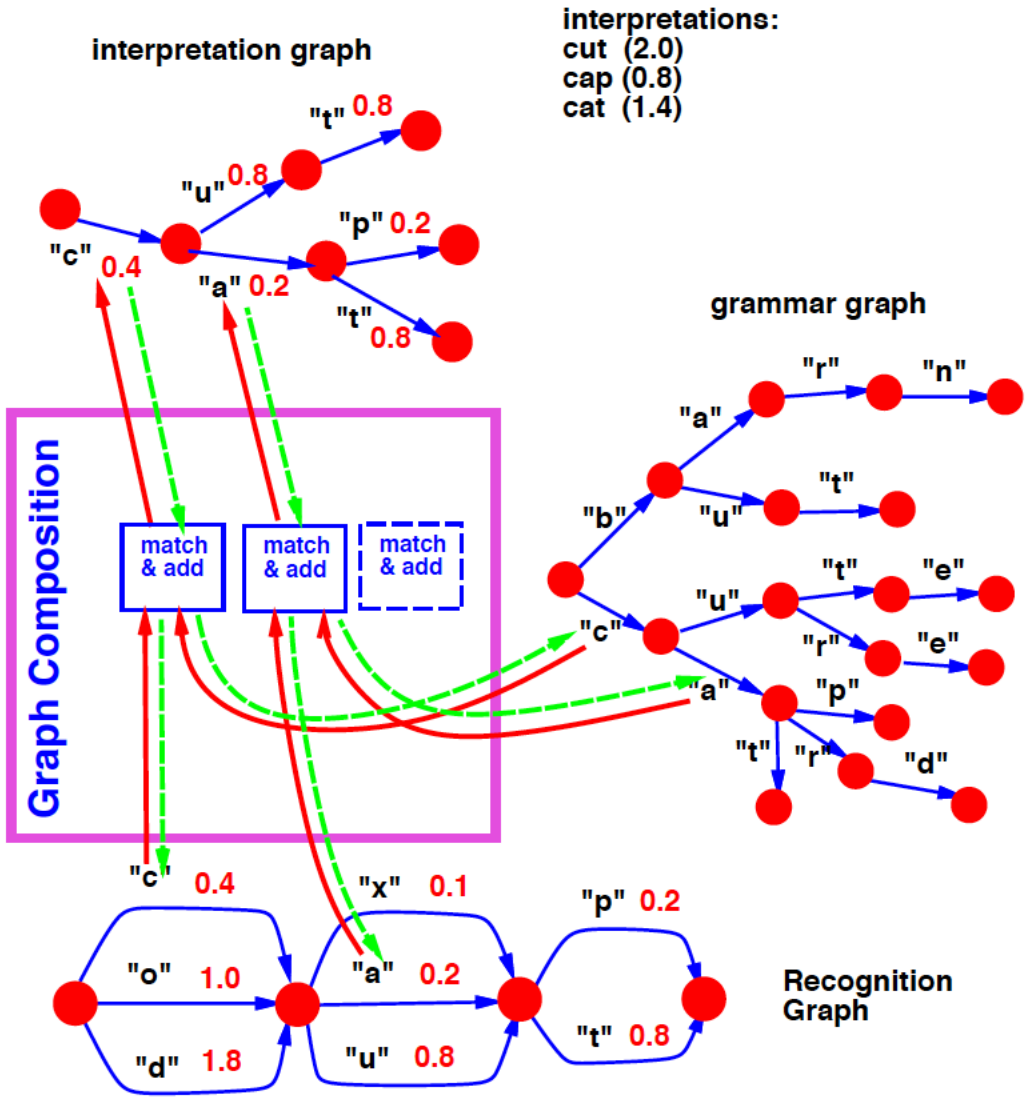
Graph with all well-formed amounts



Field Graph



Composition

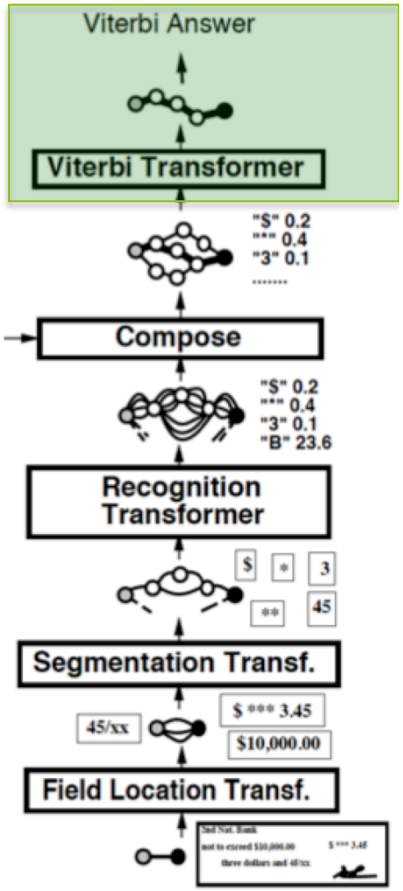


Viterbi

Transformer

- Selects the path with the lowest accumulated penalty.

Field Graph



Learning

Zurich University of Applied Sciences

zh aw

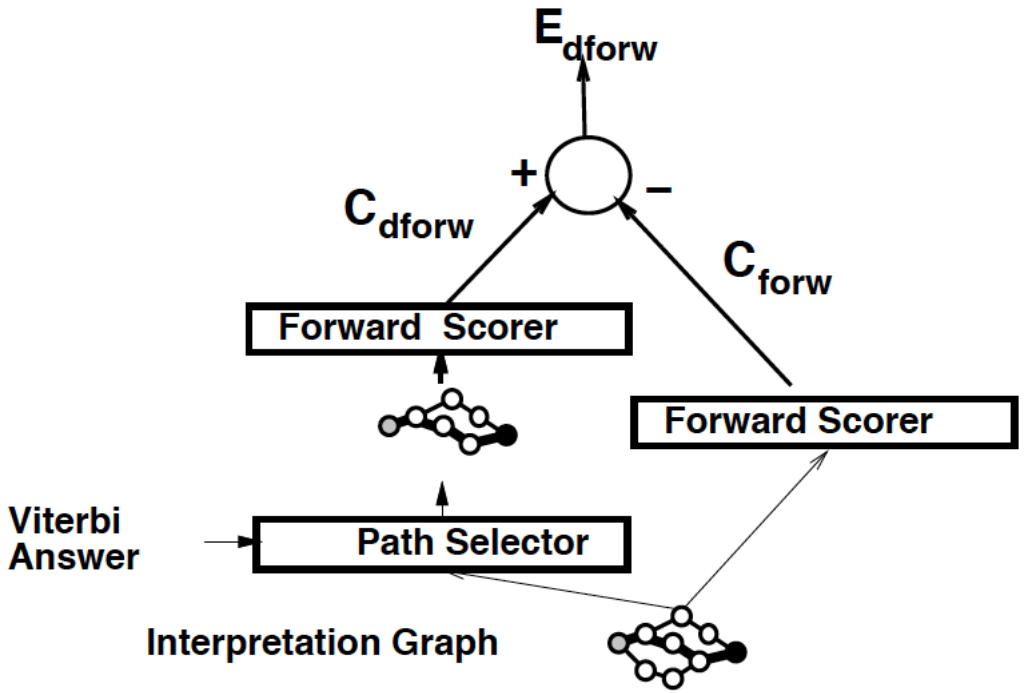
- Every transformer need to be differentiable
- Each stage of the system has tunable parameters
- Initialized with reasonable values
- Trained with whole checks labeled with the correct amount

- Loss function E minimized by **Discriminative Forward** criterion:

$$L = C_{cvit} - C_{vit}$$

Discriminative Forward

- Difference between:
  - Path with the lowest sum of penalties and the correct Seq
  - Path with the lowest sum of penalties and the allowed Seq



Zurich University
of Applied Sciences

REMARKS / OUTLOOK

Conclusions

Example: Learning to segment without intermediate labels


THE CAVALIER DAILY

Opponents Meet; College Election Details Planned

Gregory To Speak At Coalition Rally

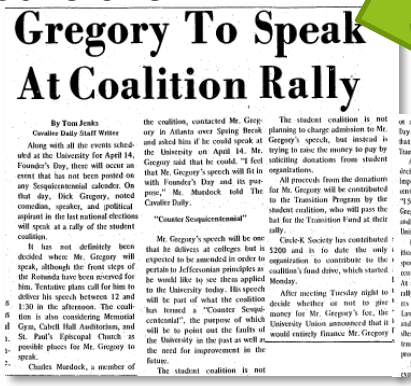
To Convince City

Fair Housing Supporters Plan Rally

Court Ruling Necessitates Draft Re-Evaluation

Council Calls Optional Classes

possible without crop marks?


Gregory To Speak At Coalition Rally

Less need for manual labeling

- Ground truth only needed for final result (not for every intermediate result like e.g. segmentation)

Early errors can be adjusted later due to…

- …unified training of all pattern recognition modules under one regime
- …postponing hard decisions until the very end

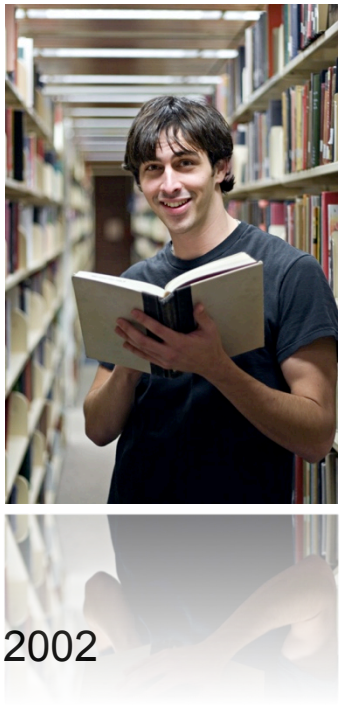
No call upon probability theory for modeling / justification

- Occam's razor: Choose easier discriminative model over generative one
  Vapnik: Don't solve a more complex problem than necessary
- No need for normalization when dealing with "penalties" instead of probabilities → no "other class" examples needed
- Less constrains on system architecture and module selection

Zurich University
of Applied Sciences

zh
aw

Further Reading



- Original short paper: Bottou, Bengio & LeCun, “Global Training of Document Processing Systems using Graph Transformer Networks”, 1997
  http://www.iro.umontreal.ca/~lisa/pointeurs/bottou-lecun-bengio-97.pdf
- Landmark long paper: LeCun, Bottou, Bengio & Haffner, “Gradient-Based Learning Applied to Document Recognition”, 1998
  http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf
- Slide set by the original authors: Bottou, “Graph Transformer Networks”, 2001
  http://leon.bottou.org/talks/gtn
- Overview: Dietterich, “Machine Learning for Sequential Data: A Review”, 2002
  http://eecs.oregonstate.edu/~tgd/publications/mlsd-ssspr.pdf
- Recent work: Collobert, “Deep Learning for Efficient Discriminative Parsing”, 2011
  http://ronan.collobert.com/pub/matos/2011_parsing_aistats.pdf