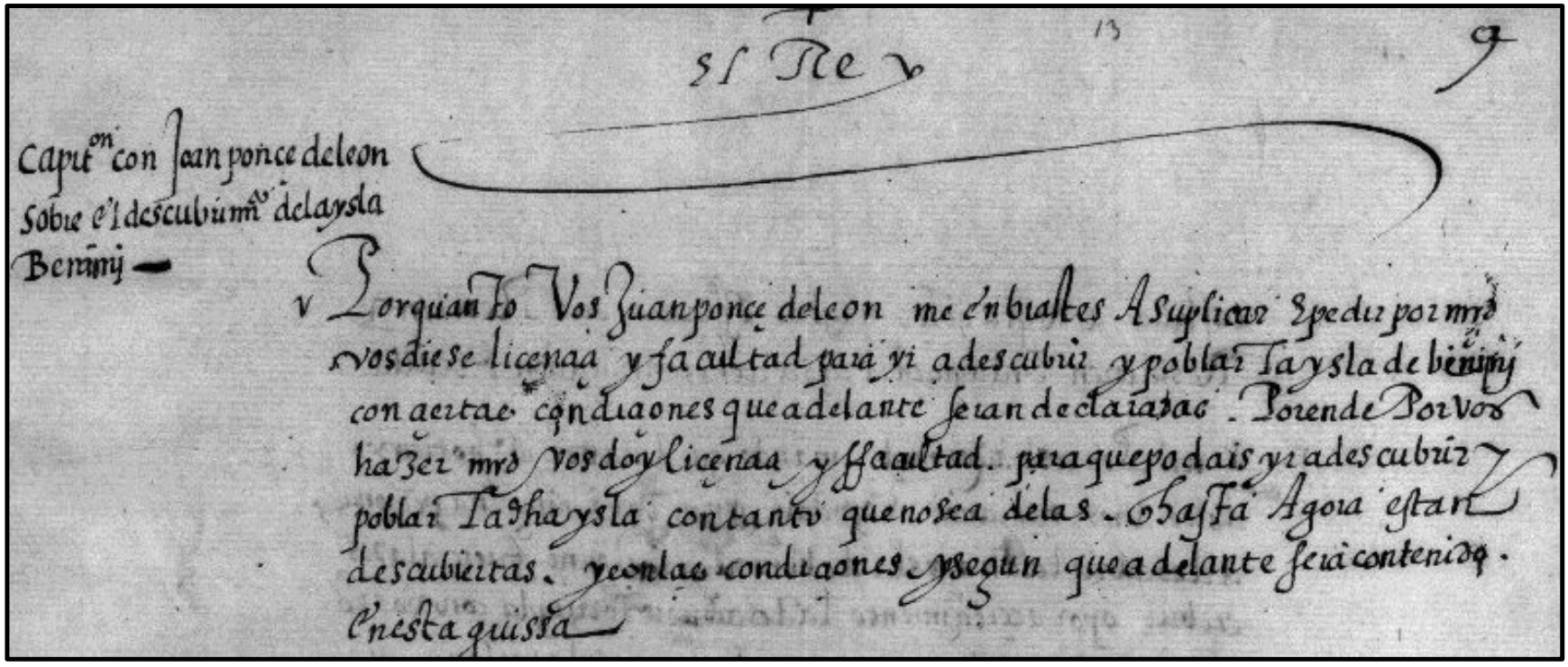
Deep-Learning Architectures for Handwritten Text Recognition

Concepts & Building Blocks

Pablo M. Olmos
olmos@tsc.uc3m.es

uc3m

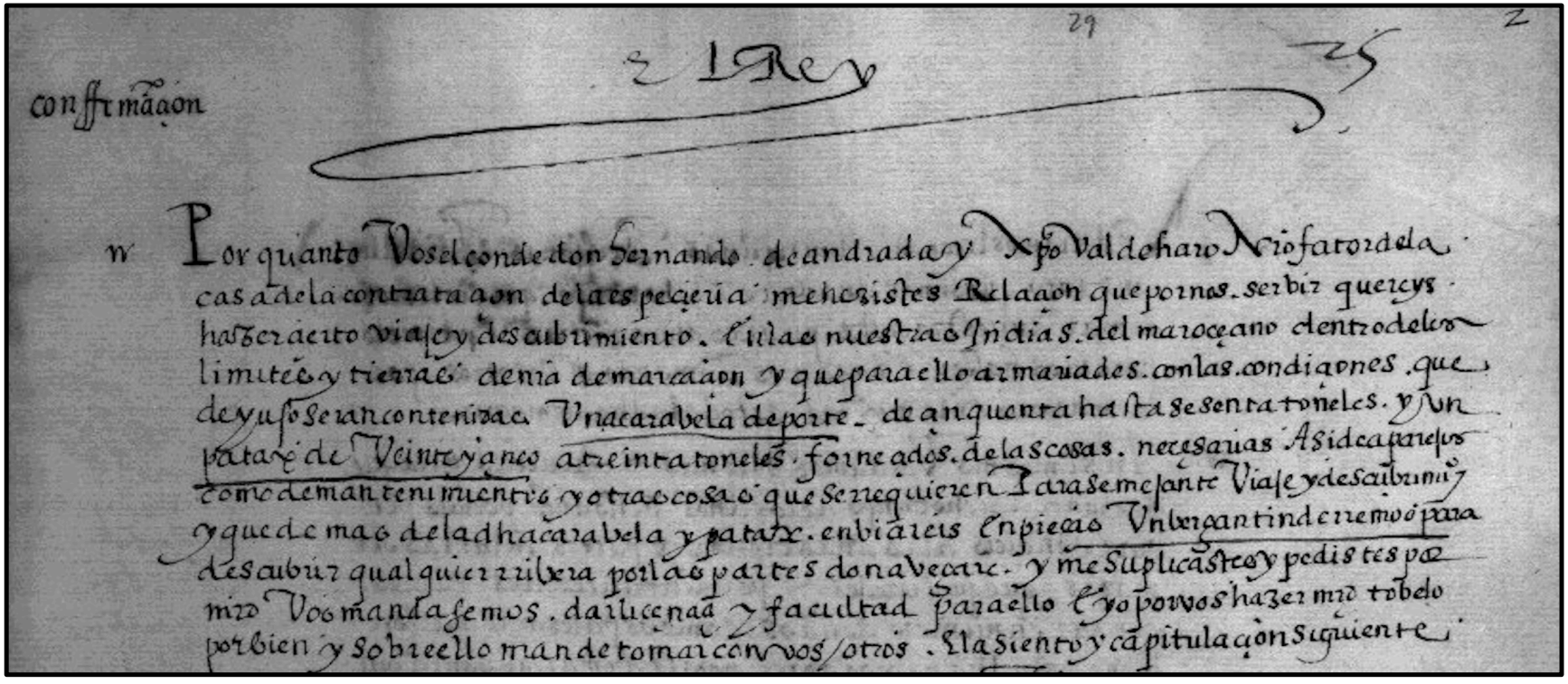
Index



CAPITULACION CON JUAN PONCE DE LEON SOBRE EL DESCUBRIMIENTO DE LA ISLA DE BENINY.

AÑO DE 1512 (1).

Por quanto vos Juan Ponce de Leon, Me embiastes a suplicar e pedir por merced, vos diese licencia y facultad para ir a descubrir y poblar las Islas de Beniny, con ciértas condiciones que adelante seran declaradas, por ende, por vos hacer merced, vos doy licencia y facultad para que podais ir á descubrir y poblar la dicha Isla, con tanto que no sea de las que hasta ahora estan descubiertas, y con las condiciones y segun que adelante será contenido en esta guissa.



conffrmaaon

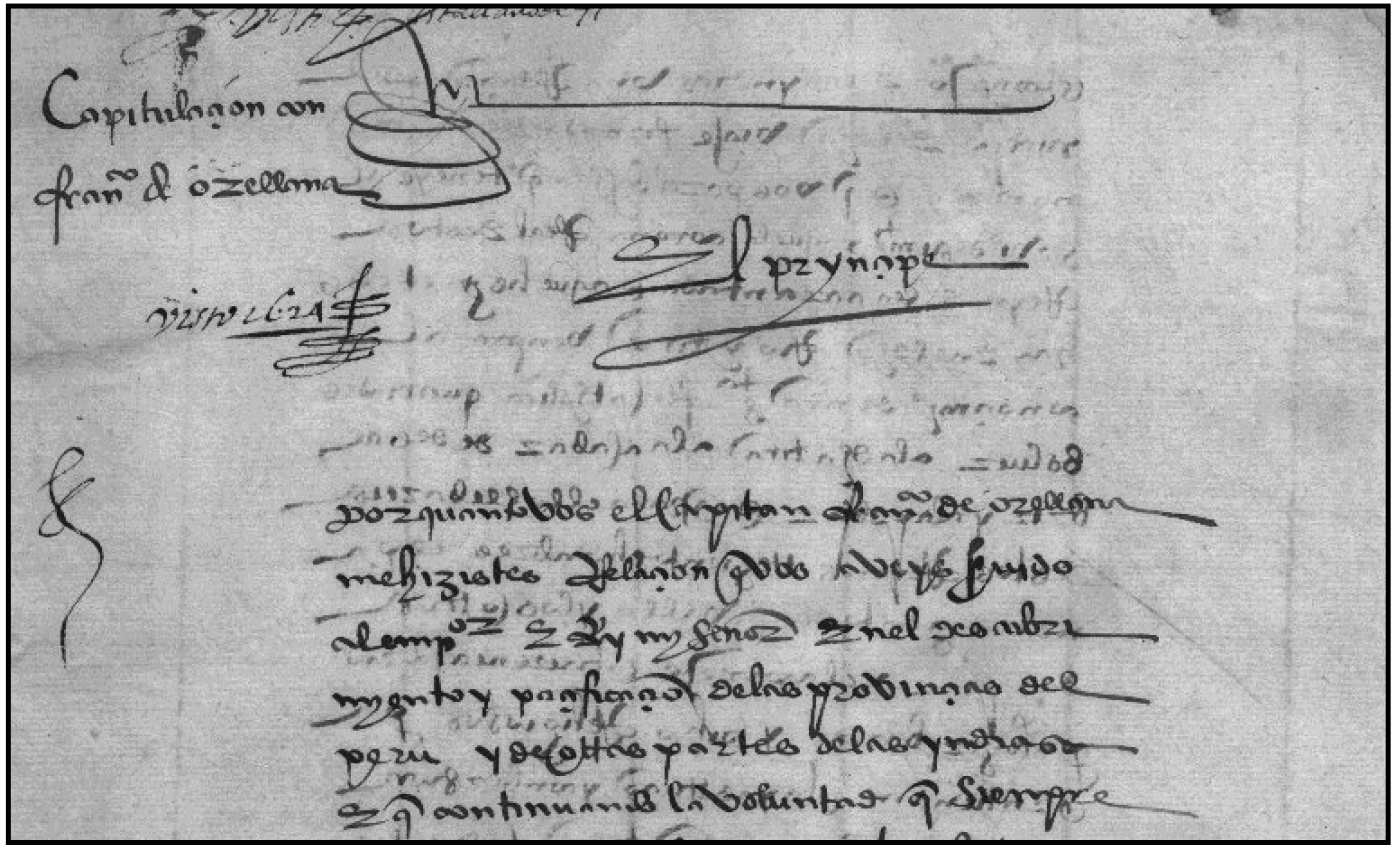
El Rey

Por quanto Vos el conde don Fernando de andrada y Xpo Valdeharo Nro factor dela casa dela contrataaon delas especeria mehezistes Relaaon que por nos. serbir quereys hazer cierto viaje y descubrimiento. En las nuestras Yndias. del mar oceano dentro delos limites y tierras de nra demarcaaon y que para ello armariades. con las condiaones. que de yuso seran contenidas. Una carabela de porte. de anquenta hasta sesenta toneles. y un patax de Veinte y anco a treinta toneles. fornecidos. delas cosas. necesarias Asi de capateles como de mantenimientos y otras cosas q̃ se requieren Para semejante Viaje y descubrimi y que de mas dela dha carabela y patax. enbiareis Un bergantin de remo para descubrir qualquier rribera por las partes donde vays. y me Suplicastes y pedistes por merced Vos mandasemos. dar licencia y facultad para ello Yo por vos hazer merced tobelo por bien y sobre ello mande tomar con vos/otros. El asiento y capitulaçon siguiente

Capitulacion que se tomó con Hernando de Andiada y Cristóbal de Haro para hacer varios descubrimientos.

Año de 1526 (1).

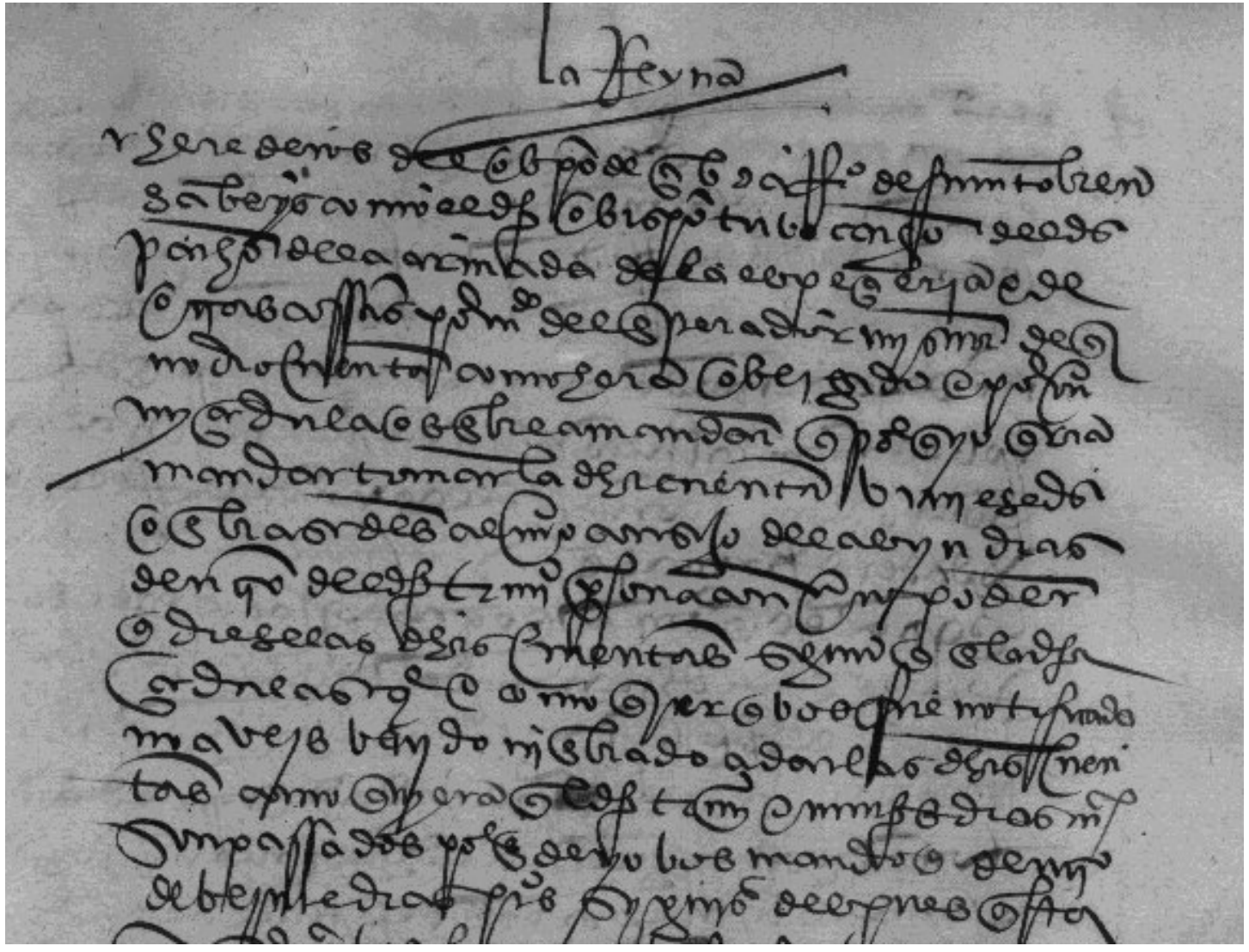
Por quanto vos el Conde Don Hernando de Andiada y Cristóbal de Haro, Nuestro factor de la casa de la contratacion de la especería, Me hicisteis relacion, que por Nos servir, quereis hacer cierto viaje y descubrimiento en las Nuestras Yndias del mar Occeano, dentro de los límites y tierras de Nuestra demarcacion; y què para ello, armariades con las condiciones que de suso seran contenidas, una caravela de porte de cinquenta hasta sesenta toneles y un patax de veinte y cinco á treinta toneles, fornecidos de las co-



CAPITULACION QUE SE TOMÓ CON FRANCISCO DE ORELLANA.

AÑO DE 1544 (1).

Por quanto vos el capitan Francisco de Orellana, Me hicistes rrelacion que vos habeis servido al Emperador y Rey Mi Señor, en el descubrimiento y pacificacion de las provincias del Pirú y otras partes de las Indias, y que continuando la voluntad que siempre habeis tenido de servir á Su Magestad, salistes de las provincias de Quito con Gonzalo Pizarro, al descubrimienio del valle de la canela, y que para ello empleastes en cien caballos y armas y herrage y otras cosas de rrescate,



There are 80 million of documents at the Archivo General de Indias de Sevilla. Only the 10% are digitalized and, **among them**, only the 10% of them are transcribed!

Researchers have to rely on the metadata provided for each document by the archivists. It contains very general information about the topic of the document.

Our (very) long-term goals

- Transcription of the millions of historic documents stored at historic archives.
- Provide search-by-content tools.
- Find latent structures in the corpus of documents, able to find related documents beyond what the current document metadata might suggest.

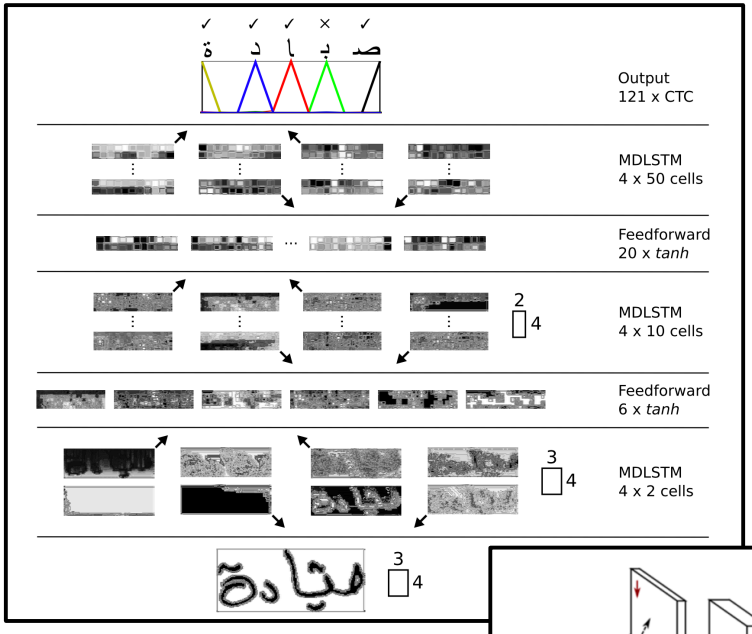
Current Status of the Project

- We are now implementing state-of-the-art techniques based on Deep-Learning (DL) for Modern Handwritten Text Recognition.
- **IAM Database** of (labelled) handwritten text.
- Investigate generalization to historical documents:
  - **Very different handwriting styles**. Almost each document has been written by a different person ...
  - Only a few pages of each document.
  - How much labeled data are enough?
  - ...

Today

Describe the basic elements of DL architectures for Handwritten Text Recognition:

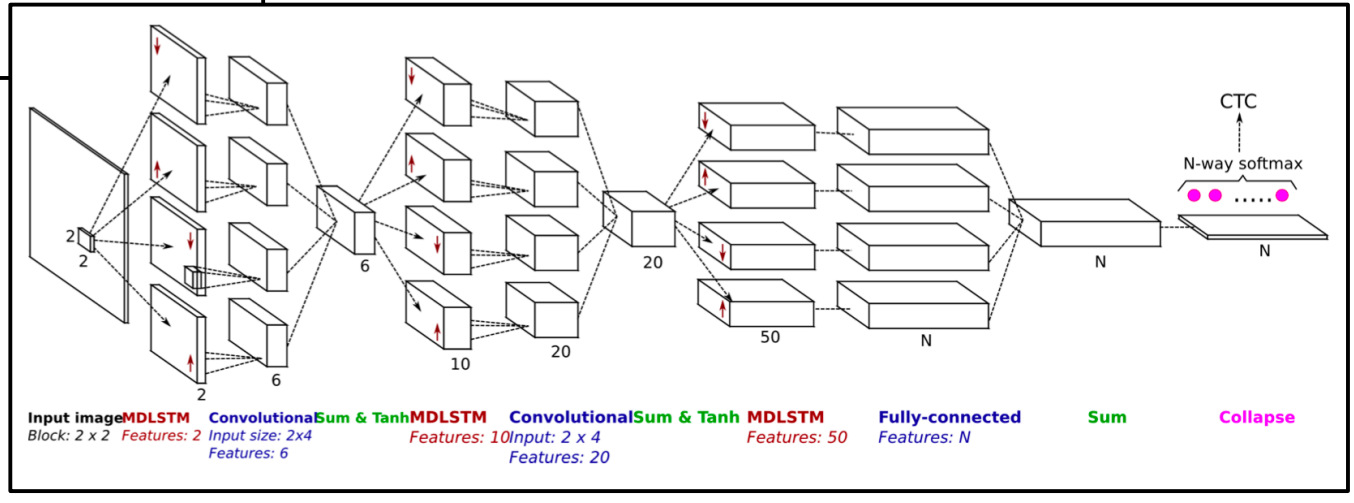
- Multi-Dimensional Long Short-Term Memory Recurrent Neural Networks.
- Connectionist Temporal Classification.



Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks

Alex Graves
TU Munich, Germany
graves@in.tum.de

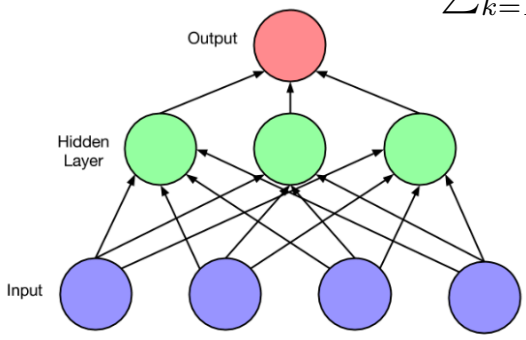
Jürgen Schmidhuber
IDSIA, Switzerland and TU Munich, Germany
juergen@idsia.ch

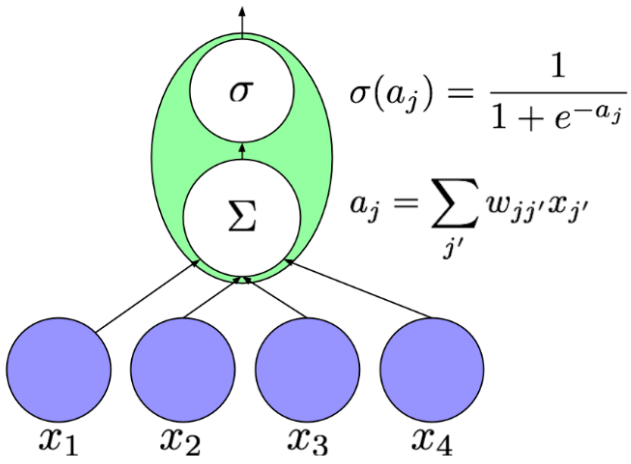


Index

Neural Networks

$$\hat{y}_k = \frac{\mathrm{e}^{a_k}}{\sum_{k=1}^{K} \mathrm{e}^{a'_k}}$$





$$\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} y_k^{(n)} \log(y_k^{(n)}) + (1 - y_k^{(n)}) \log(1 - y_k^{(n)})$$

Training: Back-propagation combined with Stochastic Gradient Descend

Figures: Lipton, Berkowitz (2015) *A Critical Review of Recurrent Neural Networks for Sequence Learning*

The effective capacity of NNs is sufficient for memorizing the entire data set

UNDERSTANDING DEEP LEARNING REQUIRES RETHINKING GENERALIZATION

Chiyuan Zhang*
Massachusetts Institute of Technology
chiyuan@mit.edu

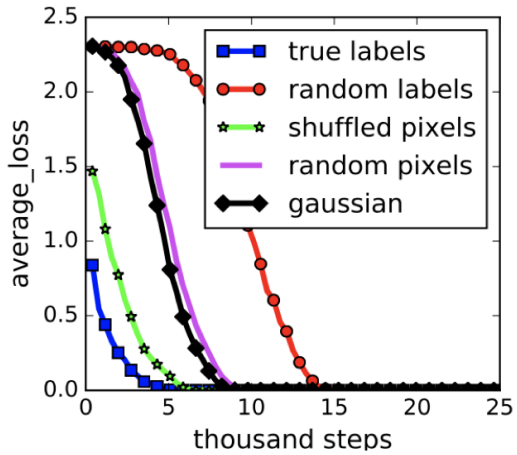
Samy Bengio
Google Brain
bengio@google.com

Moritz Hardt
Google Brain
mrtz@google.com

Benjamin Recht†
University of California, Berkeley
brecht@berkeley.edu

Oriol Vinyals
Google DeepMind
vinyals@google.com

"Deep NNs easily fit random labels"



Fitting random labels and random pixels on CIFAR10

Table 1: The training and test accuracy (in percentage) of various models on the CIFAR10 dataset. Performance with and without data augmentation and weight decay are compared. The results of fitting random labels are also included.

model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402	yes	yes	100.0	89.05
		yes	no	100.0	89.31
		no	yes	100.0	86.03
		no	no	100.0	85.75
(fitting random labels)		no	no	100.0	9.78
Inception w/o BatchNorm	1,649,402	no	yes	100.0	83.00
		no	no	100.0	82.00
(fitting random labels)		no	no	100.0	10.12
Alexnet	1,387,786	yes	yes	99.90	81.22
		yes	no	99.82	79.66
		no	yes	100.0	77.36
		no	no	100.0	76.07
(fitting random labels)		no	no	99.82	9.86
MLP 3x512	1,735,178	no	yes	100.0	53.35
		no	no	100.0	52.39
(fitting random labels)		no	no	100.0	10.48
MLP 1x512	1,209,866	no	yes	99.80	50.39
		no	no	100.0	50.51
(fitting random labels)		no	no	99.34	10.61

"even with all of the regularizers turned off, all of the models still generalize very well"

"It is unlikely that the regularizers are the fundamental reason for generalization, as the networks continue to perform well after all the regularizers removed. "

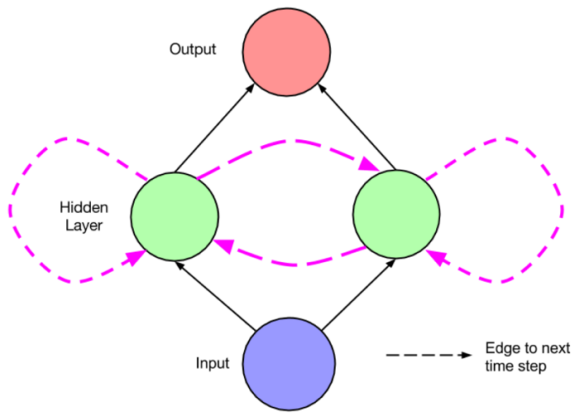
Exploiting temporal correlation: Recurrent NNs

- Recurrent neural networks (RNNs) are connectionist models with the ability to selectively pass information across sequence steps, while processing sequential data one element at a time.
- Feedforward neural networks augmented by the inclusion of edges that span adjacent time steps, introducing a notion of time to the model.

$$\boldsymbol{h}^{(t)} = \sigma(W^{\mathrm{hx}}\boldsymbol{x}^{(t)} + W^{\mathrm{hh}}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_h)$$

$$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}(W^{\mathrm{yh}}\boldsymbol{h}^{(t)} + \boldsymbol{b}_y)$$

$$\hat{\boldsymbol{y}}^{(t)} = P(\mathrm{label}|\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t)})$$



Figures: Lipton, Berkowitz (2015) *A Critical Review of Recurrent Neural Networks for Sequence Learning*

Why not Markov models?

- Generative models, EM-based training.
- Markov model approaches are limited because their states must be drawn from a modestly sized discrete state space $\mathcal{S}$
- Standard operations become infeasible with an HMM when the set of possible hidden states grows large.

- In contrast, RNNs can effectively handle a number of distinct states grows exponentially with the number of nodes in the layer.
- Discriminative supervised models, SGD training.

Bi-directional RNNs

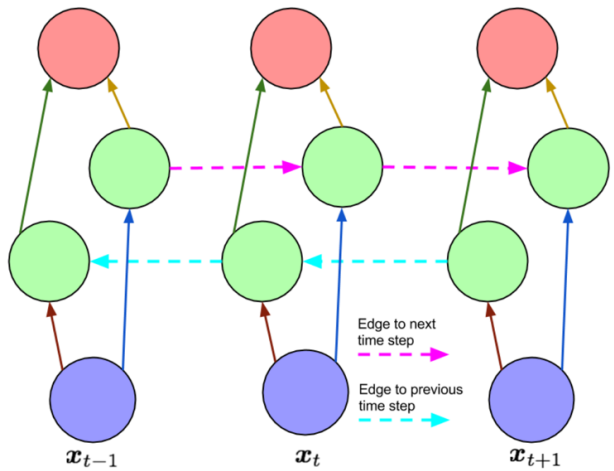
- There are two layers of hidden nodes. Both hidden layers are connected to input and output.
- The first layer has recurrent connections from the past time steps while in the second the direction of recurrent of connections is flipped.

$$\boldsymbol{h}^{(t)} = \sigma(W^{\mathrm{hx}}\boldsymbol{x}^{(t)} + W^{\mathrm{hh}}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_h)$$

$$\boldsymbol{z}^{(t)} = \sigma(W^{\mathrm{zx}}\boldsymbol{x}^{(t)} + W^{\mathrm{zz}}\boldsymbol{z}^{(t+1)} + \boldsymbol{b}_z)$$

$$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}(W^{\mathrm{yh}}\boldsymbol{h}^{(t)} + W^{\mathrm{yz}}\boldsymbol{z}^{(t)} + \boldsymbol{b}_y)$$

$$\hat{\boldsymbol{y}}^{(t)} = P(\mathrm{label}|\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(T)})$$



Figures: Lipton, Berkowitz (2015) *A Critical Review of Recurrent Neural Networks for Sequence Learning*

Long Short-Term Memory Cells

- RNNs are limited by their temporal expressiveness, which is manifested by the *vanishing gradient problem*. Don't have the ability to learn long-range dependencies.
- This problem is alleviated by LSTMs, which show superior ability to learn long-range dependencies.

$$\boldsymbol{g}^{(t)} = \phi(W^{\mathrm{gx}}\boldsymbol{x}^{(t)} + W^{\mathrm{gh}}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_g)$$

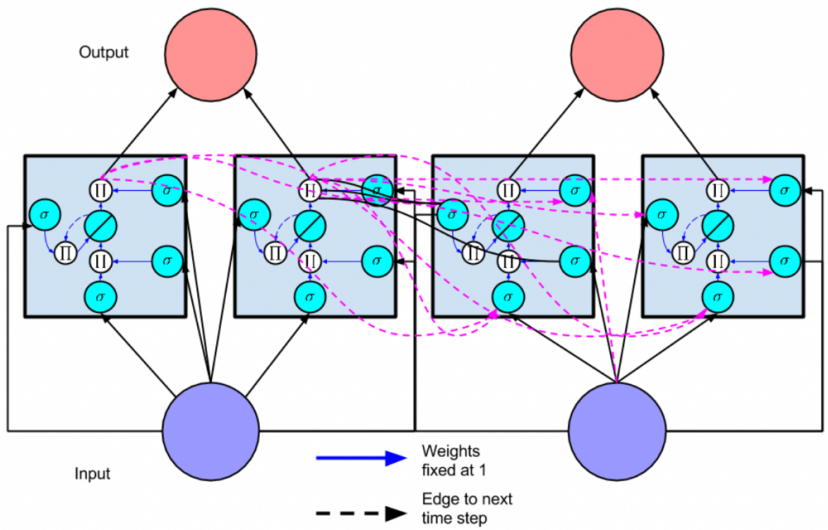
$$\boldsymbol{i}^{(t)} = \sigma(W^{\mathrm{ix}}\boldsymbol{x}^{(t)} + W^{\mathrm{ih}}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_i)$$

$$\boldsymbol{f}^{(t)} = \sigma(W^{\mathrm{fx}}\boldsymbol{x}^{(t)} + W^{\mathrm{fh}}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_f)$$

$$\boldsymbol{o}^{(t)} = \sigma(W^{\mathrm{ox}}\boldsymbol{x}^{(t)} + W^{\mathrm{oh}}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_o)$$

$$\boldsymbol{s}^{(t)} = \boldsymbol{g}^{(t)} \odot \boldsymbol{i}^{(i)} + \boldsymbol{s}^{(t-1)} \odot \boldsymbol{f}^{(t)}$$

$$\boldsymbol{h}^{(t)} = \phi(\boldsymbol{s}^{(t)}) \odot \boldsymbol{o}^{(t)}.$$

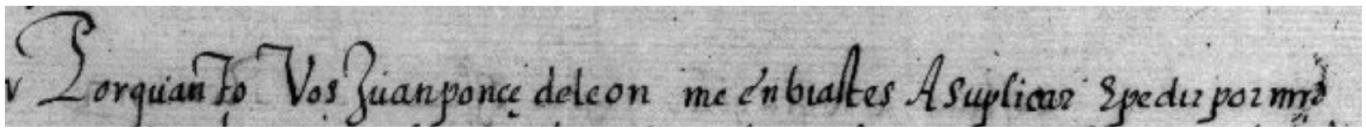


Figures: Lipton, Berkowitz (2015) *A Critical Review of Recurrent Neural Networks for Sequence Learning*

Index

Labelling alignment is a problem with RNNs

- Labelling unsegmented sequence data is a ubiquitous problem in real-world sequence learning.
- RNNs can only be trained to make a series of independent label classifications.
- Training data must be pre-segmented.
- Since the RNN network only outputs local classifications, a post-processing stage is required to give the final label sequence.



Porquanto Vos Juan ponce de leon me enbiastes A suplicar e pedir por merced

Hybrid HMM-RNN models

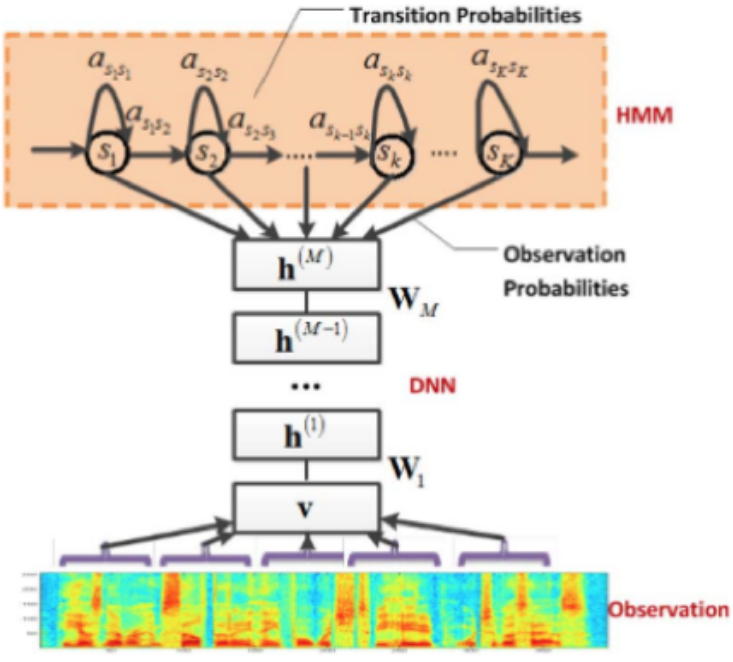
- HMM models the sequential structure of the data, able to segment localized classifications provided by the RNNs into a temporal classification of the entire label sequence.
- In HMMs, training is generative, even though sequence labelling is a discriminative task.
- Iterative approach, where the alignment provided by the HMM is used to successively retrain the neural network.

Markovian Models for Sequential Data

Yoshua Bengio †
Dept. Informatique et Recherche Opérationnelle
Université de Montréal, Montreal, Qc H3C-3J7, Canada

Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition

George E. Dahl, Dong Yu, *Senior Member, IEEE*, Li Deng, *Fellow, IEEE*, and Alex Acero, *Fellow, IEEE*

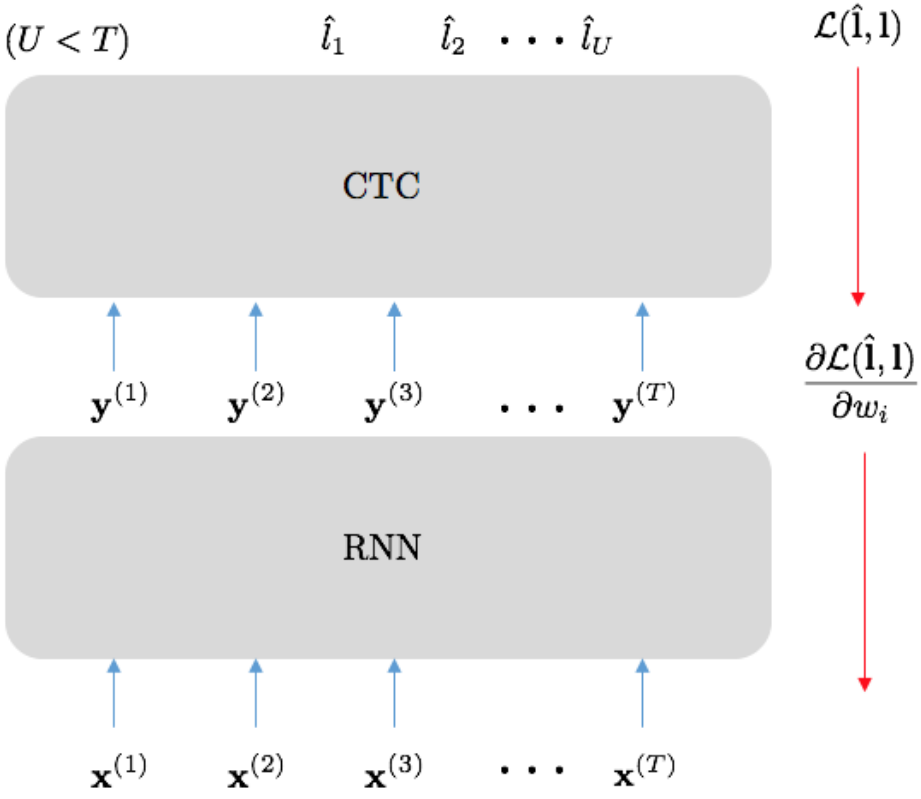


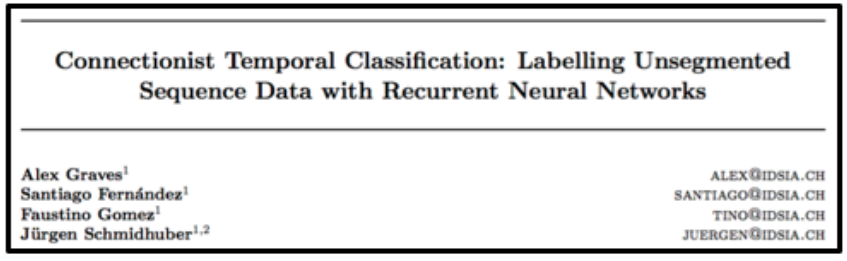
Algorithmic 1 Main Steps to Train CD-DNN-HMMs

1) Train a best tied-state CD-GMM-HMM system where state tying is determined based on the data-driven decision tree. Denote the CD-GMM-HMM *gmm-hmm*.
2) Parse *gmm-hmm* and give each senone name an ordered $senoneid$ starting from 0. The $senoneid$ will be served as the training label for DNN fine-tuning.
3) Parse *gmm-hmm* and generate a mapping from each physical tri-phone state (e.g., b-ah+t.s2) to the corresponding $senoneid$. Denote this mapping $state2id$.
4) Convert *gmm-hmm* to the corresponding CD-DNN-HMM dnn–$hmm1$ by borrowing the tri-phone and senone structure as well as the transition probabilities from gmm–hmm.
5) Pre-train each layer in the DNN bottom-up layer by layer and call the result *ptdnn*.
6) Use gmm–hmm to generate a state-level alignment on the training set. Denote the alignment $align$–raw.
7) Convert $align$–raw to $align$ where each physical tri-phone state is converted to $senoneid$.
8) Use the $senoneid$ associated with each frame in $align$ to fine-tune the DBN using back-propagation or other approaches, starting from $ptdnn$. Denote the DBN dnn.
9) Estimate the prior probability $p(s_i) = n(s_i)/n$, where $n(s_i)$ is the number of frames associated with senone s_i in $align$ and n is the total number of frames.
10) Re-estimate the transition probabilities using dnn and dnn–$hmm1$ to maximize the likelihood of observing the features. Denote the new CD-DNN-HMM dnn–$hmm2$.
11) Exit if no recognition accuracy improvement is observed in the development set; Otherwise use

Connectionist Temporal Classification (CTC)

- Pure discriminative methods tend to give better results for classification tasks, because they focus entirely on finding the correct labels.
- CTC models all aspects of the sequence with a single neural network.
- It also does not require pre-segmented training data, or external post-processing to extract the label sequence from the network outputs.



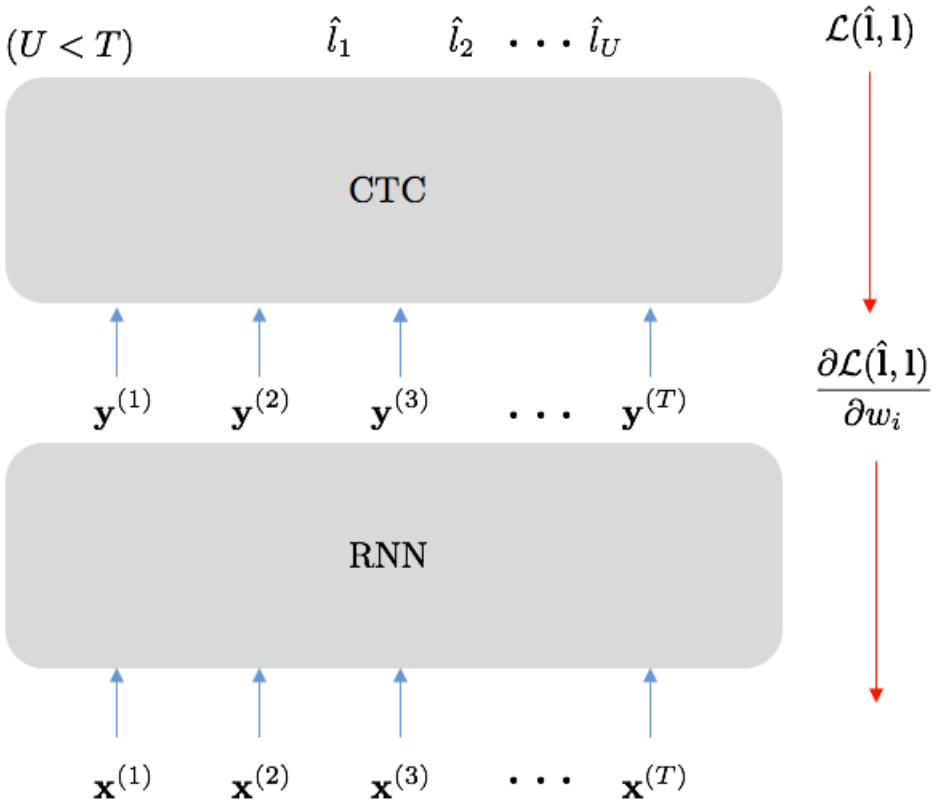


Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks

Alex Graves[1] ALEX@IDSIA.CH
Santiago Fernández[1] SANTIAGO@IDSIA.CH
Faustino Gomez[1] TINO@IDSIA.CH
Jürgen Schmidhuber[1,2] JUERGEN@IDSIA.CH

Notation

- $\mathcal{A}$ is the label alphabet. E.g., characters in the Latin alphabet.
- $\mathcal{A}' = \mathcal{A} \cup \{\text{blank}\}$
- $\boldsymbol{X} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(T)})$ is the input sequence
- $\boldsymbol{Y} = (\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(T)})$ is the RNN output, where $y_k^{(t)} \in [0, 1]$, $k = 1, \ldots, |\mathcal{A}'|$, and $\sum_{k=1}^{|\mathcal{A}'|} y_k^{(t)} = 1$ (softmax layer at the RNN output).
- $\hat{\boldsymbol{l}} = (\hat{l}_1, \hat{l}_2, \ldots, \hat{l}_U)$ is the out label sequence, $\hat{l}_j \in \mathcal{A}'$.



From Network Outputs to Labellings

- $y_k^{(t)} \rightarrow$ Probability of observing label k at time t.
- $\boldsymbol{Y} \rightarrow$ Probability distribution over all possible *paths* in the set $\tilde{\mathcal{A}}^T$:

$$p(\boldsymbol{\pi}|\boldsymbol{X}) = \prod_{i=1}^{T} y_{\pi_t}^{(t)} \quad \boldsymbol{\pi} \in \tilde{\mathcal{A}}^T$$

- A many-to-one map $\mathcal{F} : \tilde{\mathcal{A}}^T \rightarrow \tilde{\mathcal{A}}^{\leq T}$ is defined.
  - Remove repeated labels
  - Remove blanks
  - $\mathcal{F}(a - ab-) = \mathcal{F}(-aa - -abb) = aab$
  - Outputting a new label when the network switches from predicting no label to predicting a label, or from predicting one label to another.
- For any $\boldsymbol{l} \in \mathcal{A}^U$ with $U \leq T$ then

$$p(\boldsymbol{l}|\boldsymbol{X}) = \sum_{\pi \in \mathcal{F}^{-1}(\boldsymbol{l})} p(\pi|\boldsymbol{X})$$

From Network Outputs to Labellings

- $y_k^{(t)} \rightarrow$ Probability of observing label k at time t.
- $\boldsymbol{Y} \rightarrow$ Probability distribution over all possible *paths* in the set $\tilde{\mathcal{A}}^T$:

$$p(\boldsymbol{\pi}|\boldsymbol{X}) = \prod_{i=1}^{T} y_{\pi_t}^{(t)} \quad \boldsymbol{\pi} \in \tilde{\mathcal{A}}^T$$

- A many-to-one map $\mathcal{F} : \tilde{\mathcal{A}}^T \rightarrow \tilde{\mathcal{A}}^{\leq T}$ is defined.
  - Remove repeated labels
  - Remove blanks
  - $\mathcal{F}(a - ab-) = \mathcal{F}(-aa - -abb) = aab$
  - Outputting a new label when the network switches from predicting no label to predicting a label, or from predicting one label to another.
- For any $\boldsymbol{l} \in \mathcal{A}^U$ with $U \leq T$ then

$$p(\boldsymbol{l}|\boldsymbol{X}) = \sum_{\pi \in \mathcal{F}^{-1}(\boldsymbol{l})} p(\pi|\boldsymbol{X})$$

From Network Outputs to Labellings

- $y_k^{(t)} \rightarrow$ Probability of observing label k at time t.
- $\boldsymbol{Y} \rightarrow$ Probability distribution over all possible *paths* in the set $\tilde{\mathcal{A}}^T$:

$$p(\boldsymbol{\pi}|\boldsymbol{X}) = \prod_{i=1}^{T} y_{\pi_t}^{(t)} \quad \boldsymbol{\pi} \in \tilde{\mathcal{A}}^T$$

- A many-to-one map $\mathcal{F} : \tilde{\mathcal{A}}^T \rightarrow \tilde{\mathcal{A}}^{\leq T}$ is defined.
  - Remove repeated labels
  - Remove blanks
  - $\mathcal{F}(a - ab-) = \mathcal{F}(-aa - -abb) = aab$
  - Outputting a new label when the network switches from predicting no label to predicting a label, or from predicting one label to another.
- For any $\boldsymbol{l} \in \mathcal{A}^U$ with $U \leq T$ then

$$p(\boldsymbol{l}|\boldsymbol{X}) = \sum_{\boldsymbol{\pi} \in \mathcal{F}^{-1}(\boldsymbol{l})} p(\pi|\boldsymbol{X})$$

From Network Outputs to Labellings

$$p(\boldsymbol{l}|\boldsymbol{X}) = \sum_{\boldsymbol{\pi} \in \mathcal{F}^{-1}(\boldsymbol{l})} p(\pi|\boldsymbol{X}), \quad \boldsymbol{l} \in \mathcal{A}^{U}$$

- This collapse of different paths into the same label sequence $\boldsymbol{l}$ is what makes possible for CTC to use unsegmented data.
- $p(\boldsymbol{l}|\boldsymbol{X})$ can be efficiently computed using dynamic programming $\rightarrow$ **CTC Forward-Backward (CTC-FB) algorithm**.
- Further, the gradient of $-\log p(\boldsymbol{l}|\boldsymbol{X})$ w.r.t. y_k^t can be computed using **intermediate messages** computed during CTC-FB.

CTC Forward-Backward (I)

- Given $\boldsymbol{l}$, we define $\boldsymbol{l}'$ with blanks added to the beginning and end and inserted every pair of labels.
- For $t = 1, \ldots, T$ and $u = 1 \ldots, 2U + 1$ we define the following two sets of paths and cumulated probabilities

$$V(t,u) = \left\{ \boldsymbol{\pi} \in \tilde{\mathcal{A}}^t : \mathcal{F}(\boldsymbol{\pi}) = \boldsymbol{l}_{\lfloor \frac{u}{2} \rfloor}, \pi_t = l'_u \right\} \rightarrow \alpha(t,u) = \sum_{\boldsymbol{\pi} \in V(t,u)} \prod_{i=1}^{t} y_{\pi_i^{(i)}}$$

$$W(t,u) = \left\{ \boldsymbol{\pi} \in \tilde{\mathcal{A}}^{T-t} : \mathcal{F}(\hat{\boldsymbol{\pi}} + \boldsymbol{\pi}) = \boldsymbol{l} \ \ \forall \hat{\boldsymbol{\pi}} \in V(t,u) \right\} \rightarrow \beta(t,u) = \sum_{\boldsymbol{\pi} \in W(t,u)} \prod_{i=t}^{T} y_{\pi_i^{(i)}}$$

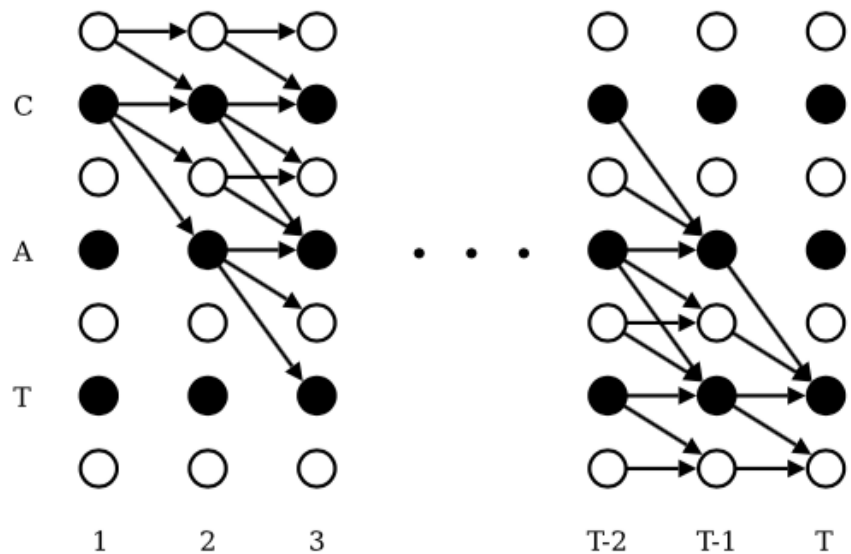


Figure: A.Graves et al. *Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks.* (2006)

CTC Forward-Backward (II)

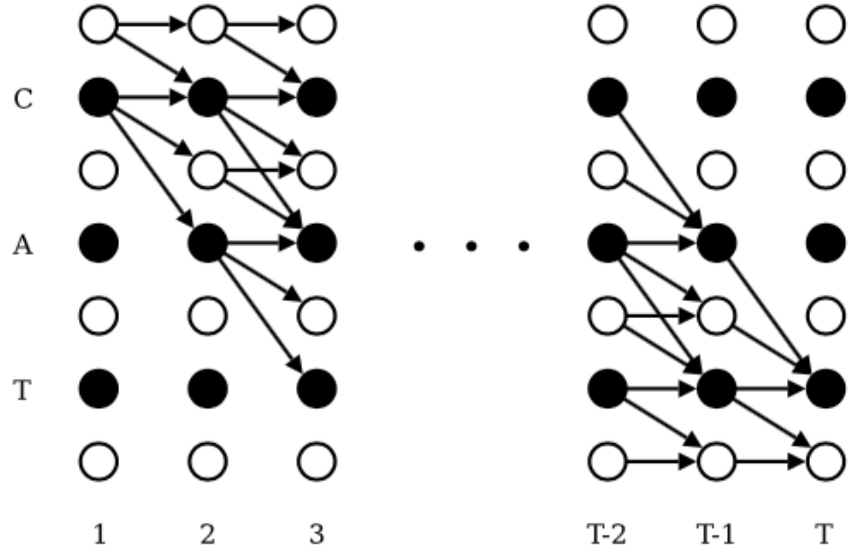


Figure: A.Graves et al. *Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks.* (2006)

$$\alpha(t,u) = f\Big(\alpha(t-1,u), \alpha(t-1,u-1), \alpha(t-1,u-2)\Big)$$

$$\beta(t,u) = g\Big(\beta(t+1,u), \beta(t+1,u+1), \alpha(t+1,u+2)\Big)$$

$$p(\boldsymbol{l}|\boldsymbol{X}) = \alpha(t,2U) + \alpha(t,2U+1)$$

$$\frac{-\log p(\boldsymbol{l}|\boldsymbol{X})}{\partial y_k^{(t)}} = -\frac{1}{p(\boldsymbol{l}|\boldsymbol{X}) y_k^{(t)}} \sum_{u:\boldsymbol{l}'_u=k} \alpha(t,u)\beta(t,u)$$

CTC decoding

Upon training, for a new input sequence $\boldsymbol{X}$, we would ideally compute

$$\boldsymbol{l}* = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l}|\boldsymbol{x})$$

Best Path Decoding (Trivial)

Let $\boldsymbol{\pi}^* = \arg\max_{\pi} p(\boldsymbol{\pi}|\boldsymbol{X})$, then

$$\boldsymbol{l}^* \approx \mathcal{F}(\boldsymbol{\pi}^*)$$

Prefix Search Decoding

Modified Forward/Backward algorithm to calculate the probabilities of successive extensions to labelling prefixes. However, the number of prefixes it must expand grows exponentially with the input sequence length.

Constrained Decoding

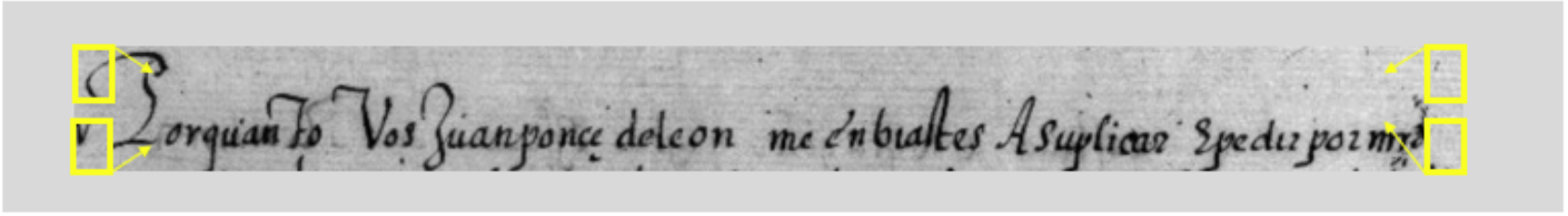
Constrain the output labellings according to some predefined grammar

Index

Multi-dimensional LSTMs

- Generalize bi-RNNs by providing recurrent connections along all spatio-temporal dimensions present in the data.
- Robust to local distortions along any combination of input dimensions.

Porquanto Vos Juan ponce de leon me enbiastes A suplicar e pedir por merced



$$\boldsymbol{h}_{i,j}^{\mathrm{NW}} = \mathrm{LSTM}\left(\boldsymbol{x}_{i,j}, \boldsymbol{h}_{i,j-1}^{\mathrm{NW}}, \boldsymbol{h}_{i-1,j}^{\mathrm{NW}}, \boldsymbol{h}_{i-1,j\pm1}^{\mathrm{NW}}\right)$$

$$\boldsymbol{h}_{i,j}^{\mathrm{NE}} = \mathrm{LSTM}\left(\boldsymbol{x}_{i,j}, \boldsymbol{h}_{i,j+1}^{\mathrm{NE}}, \boldsymbol{h}_{i-1,j}^{\mathrm{NE}}, \boldsymbol{h}_{i-1,j\pm1}^{\mathrm{NE}}\right)$$

$$\boldsymbol{h}_{i,j}^{\mathrm{SW}} = \mathrm{LSTM}\left(\boldsymbol{x}_{i,j}, \boldsymbol{h}_{i,j+1}^{\mathrm{SW}}, \boldsymbol{h}_{i+1,j}^{\mathrm{SW}}, \boldsymbol{h}_{i+1,j\pm1}^{\mathrm{SW}}\right)$$

$$\boldsymbol{h}_{i,j}^{\mathrm{NE}} = \mathrm{LSTM}\left(\boldsymbol{x}_{i,j}, \boldsymbol{h}_{i,j+1}^{\mathrm{NE}}, \boldsymbol{h}_{i+1,j}^{\mathrm{NE}}, \boldsymbol{h}_{i+1,j\pm1}^{\mathrm{NE}}\right)$$

Network Hierarchy

- Subsampling layers, alternating between MLTSMs and feedfoward/convolutional layers.
- The number of features computed by these layers increases as the size of the feature maps decrease.
- At the top of this network, there is one feature map for each label.
- A collapsing layer sums the features over the vertical axis, yielding a sequence of prediction vectors, effectively delaying the 2D to 1D transformation just before the character predictions, normalized with a softmax activation.

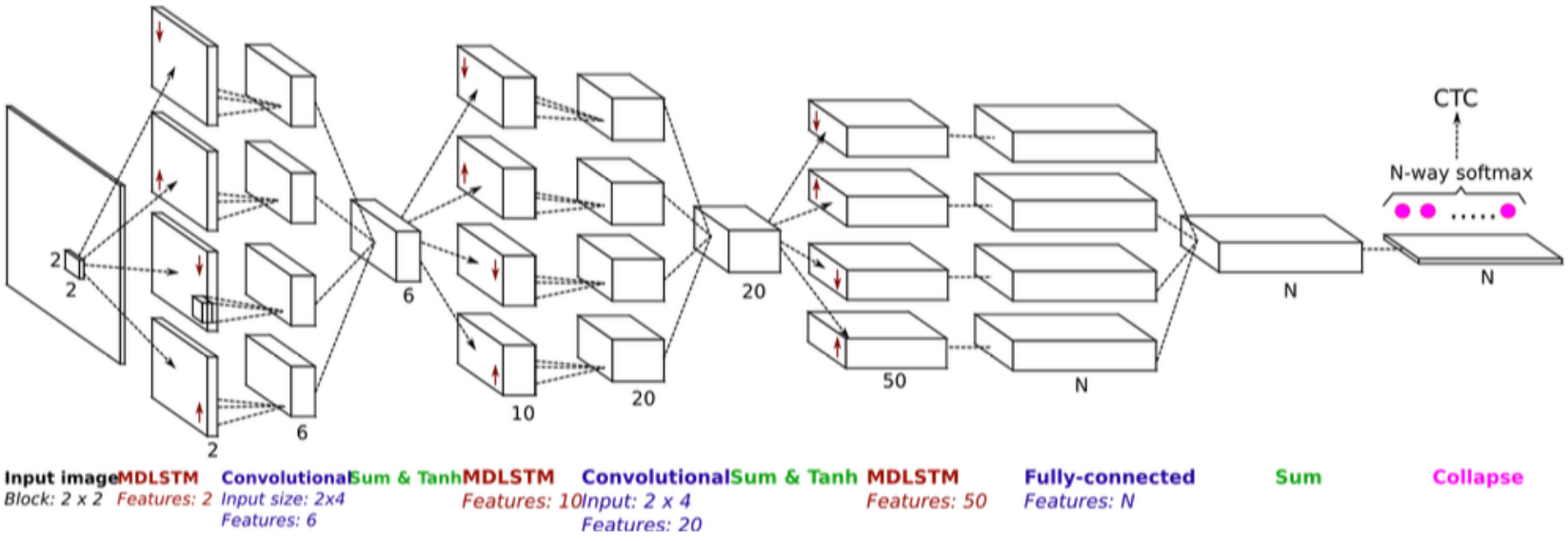


Figure: Pham et al. *Dropout improves recurrent neural networks for handwriting recognition (2014)*

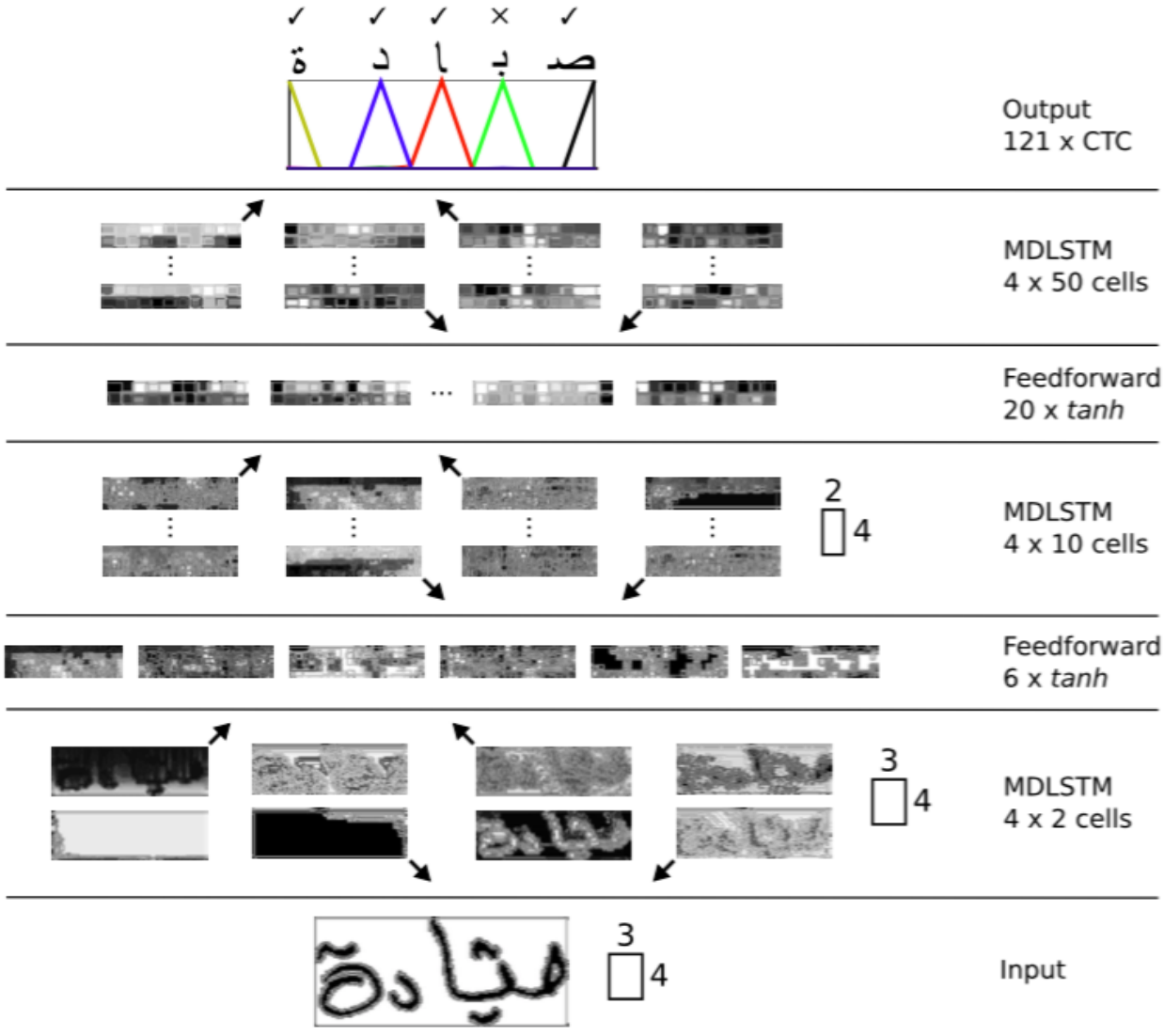


Figure: A.Graves and J.Schmidhuber. *Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks.* (2008)