Economics 2355: OCR Architecture

Melissa Dell

Harvard University

February 2021

Outline

OCR v Object Detection

- My general takeaway from object detection is that - at least for the (rather complex) tasks I am interested in - the problem is essentially solved
- Not 100%, it would be great to need fewer labels and there will be further advancements, but the SOTA currently is good enough to produce very accurate information for my downstream research
- I'm not going to think: "I can't write that paper because the object detection problem is too hard" (even though it's still a lot of work)
- The literature is amazing to read, and consists of many of the most cited papers in deep learning

OCR v. Object Detection

- The vibe around OCR is different. Many computer scientists see OCR - and documents - as an essentially solved problem, because there was OCR that [sort of, barely, very narrowly] worked decades ago. Many people seem to view documents as solved (or perhaps just boring)
- But it is anything but a solved problem, and OCR problems still derail many, many promising projects
- This may be starting to change: a lot of interest in industry and increased interest in the deep learning community (albeit more towards OCR “in the wild” - i.e. in natural images)

OCR in the Wild


```
[([[189, 75], [469, 75], [469, 165], [189, 165]], '愚园路', 0.3754989504814148),
 ([[86, 80], [134, 80], [134, 128], [86, 128]], '西', 0.40452659130096436),
 ([[517, 81], [565, 81], [565, 123], [517, 123]], '东', 0.9989598989486694),
 ([[78, 126], [136, 126], [136, 156], [78, 156]], '315', 0.8125889301300049),
 ([[514, 126], [574, 126], [574, 156], [514, 156]], '309', 0.4971577227115631),
 ([[226, 170], [414, 170], [414, 220], [226, 220]], 'Yuyuan Rd.', 0.8261902332305908),
 ([[79, 173], [125, 173], [125, 213], [79, 213]], 'W', 0.9848111271858215),
 ([[529, 173], [569, 173], [569, 213], [529, 213]], 'E', 0.8405593633651733)]
```


```
[([[71, 49], [489, 49], [489, 159], [71, 159]], 'ポ<捨て禁止!', 0.6339447498321533),
 ([[95, 149], [461, 149], [461, 235], [95, 235]],
  'NOLITTER',
  0.32493865489959717),
 ([[80, 232], [475, 232], [475, 288], [80, 288]],
  '清潔できれいな港区を',
  0.9784268140792847),
 ([[109, 289], [437, 289], [437, 333], [109, 333]],
  '港 区 MINATO CITY',
  0.18788912892341614)]
```


```
[([[129, 79], [292, 79], [292, 183], [129, 183]], '서울', 0.9718757271766663),
 ([[368, 101], [531, 101], [531, 201], [368, 201]], '평양', 0.9701956510543823),
 ([[159, 176], [258, 176], [258, 232], [159, 232]],
  'Seoul',
  0.8239474892616272),
 ([[342, 189], [539, 189], [539, 262], [342, 262]],
  'Pyeongyang',
  0.3528004288673401),
 ([[186, 276], [289, 276], [289, 333], [186, 333]],
  '56Km',
  0.6299744844436646),
 ([[344, 288], [461, 288], [461, 344], [344, 344]],
  '205Km',
  0.3810771405696869)]
```

https://github.com/JaidedAI/EasyOCR

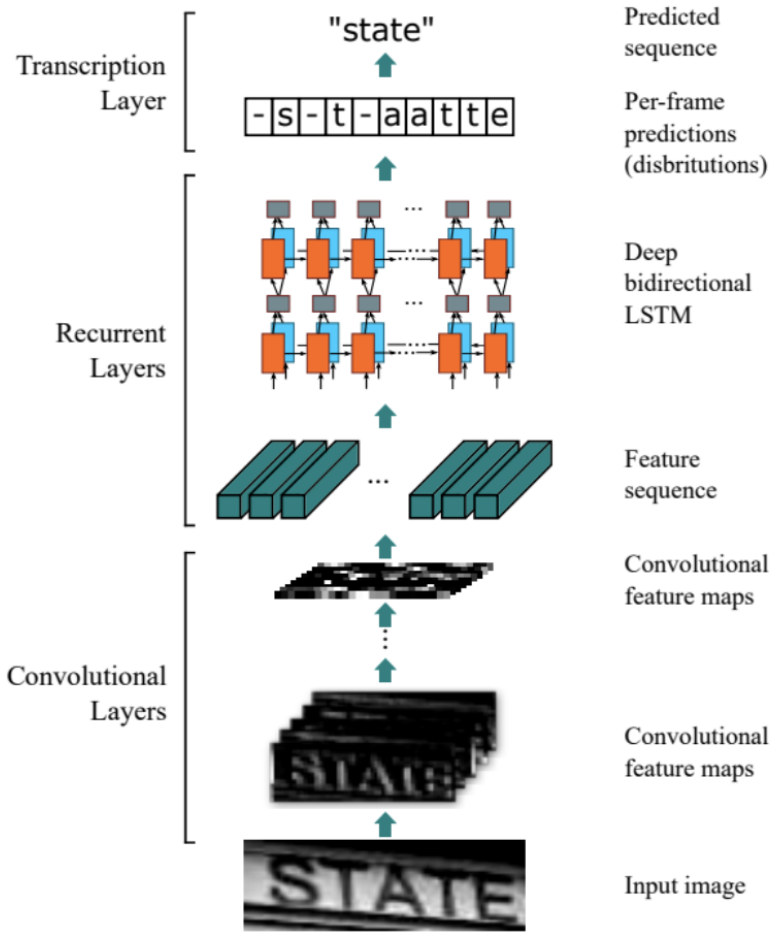
OCR Architecture

The current dominant paradigm in OCR has three components:

- A CNN features extractor (i.e. ResNet)
- A recurrent neural network (i.e. bidirectional LSTM)
- Connectionist Temporal Classification (CTC) Loss

Stack these architectures like legos

OCR Architecture: Convolutional Recurrent Neural Network



Shi et al., 2015

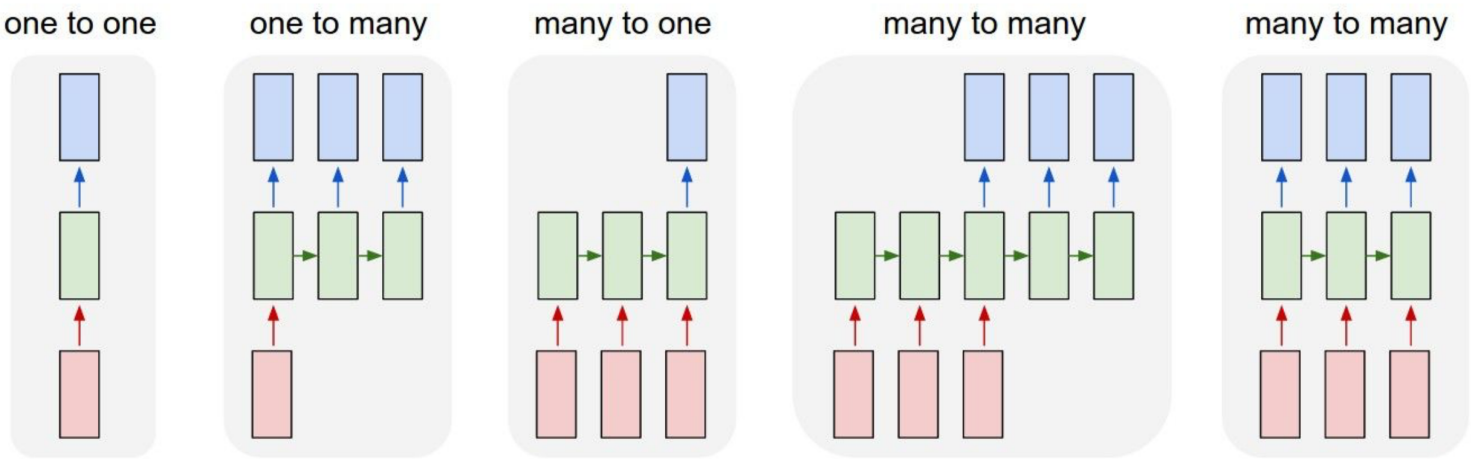
OCR Architecture

- We need to understand RNNs and CTC for this to make sense
- RNNs and CTC are much more broadly applicable and are important to understand even if OCR is not a primary interest
- Image captioning likewise involves stacking a CNN and RNN to generate arbitrary length sequences, and the architecture of speech-to-text is similar to OCR (use a frequency based features extractor on audio rather than a CNN on input images)

Outline

Cases We've Covered So Far

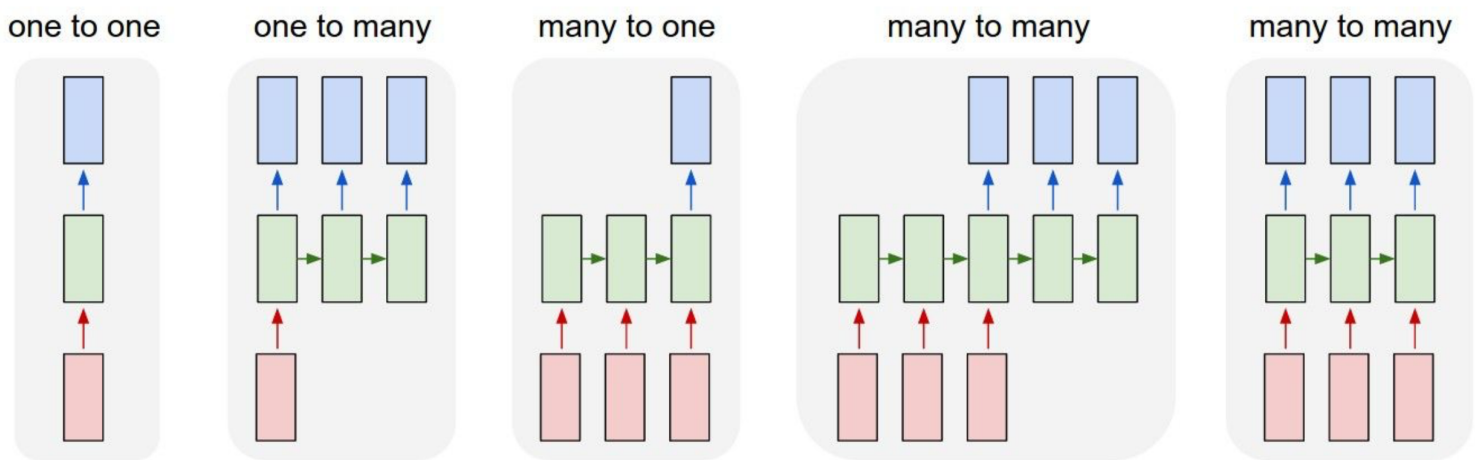
So far we have considered one-to-one network architectures, that consist of an input vector, hidden layers, and a fixed size output vector. RNNs let us operate over sequences (as input, output, or both):



Stanford CS231n Lecture Notes

Cases We've Covered So Far

Image captioning and OCR are one-to-many: the input is an image and output a variable length sequence of words. In vision, the many input cases come from video. In NLP, sentiment analysis is many (sequence of words) to one (the sentiment). Machine translation is many to many.



Stanford CS231n Lecture Notes

RNNs

A RNN is fed an input vector at every time step and also maintains an internal state. It can modify that state as a function of what is receives at each time step.

$$h_t = f_w(h_{t-1}, x_t)$$

- h_t is the state (history)
- x_t is the input vector at each time step
- f is a recurrence function and W are its parameters - the same function is applied at every t. This is what allows us to use an RNN with **arbitrary length sequences**, regardless of sequence length, we use the same function at every step.

Produce an output y_t based on the RNN state

Plain Vanilla RNN

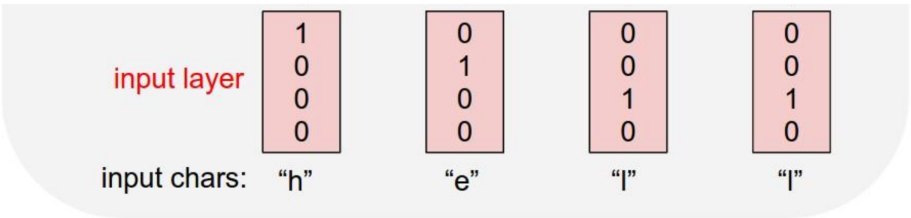
$$h_t = tanh(W_{hh}h_{t-1} + W_{xh}x_t) \quad (1)$$

$$y_t = W_{hy}h_t \quad (2)$$

Project, add, and squish the sum of the previous history and input vector. Then use another matrix projection on that hidden state to produce the output vector.

A Toy Example from Andrej Karpathy

Suppose we estimate a character level langauge model, for a four letter language [h, e, l, o]. We want to predict a distribution at each step for what letter should come next. Start with one-hot encoding of the word “hello”.

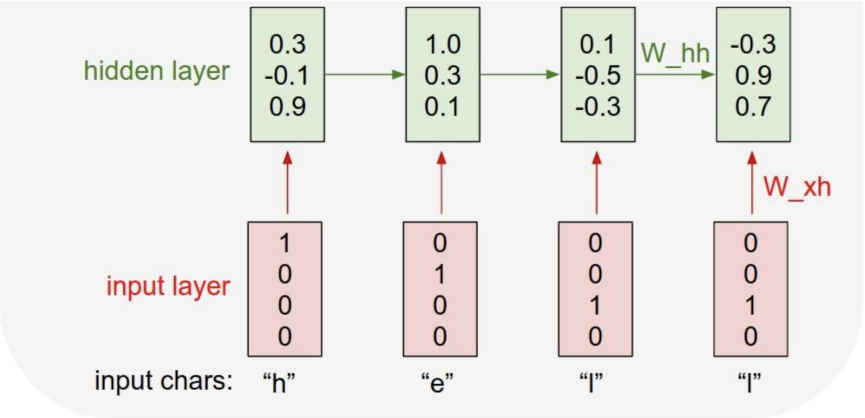


Stanford CS231n Lecture Notes

A Toy Example from Andrej Karpathy

Use recurrence formula to compute hidden state at every time step:

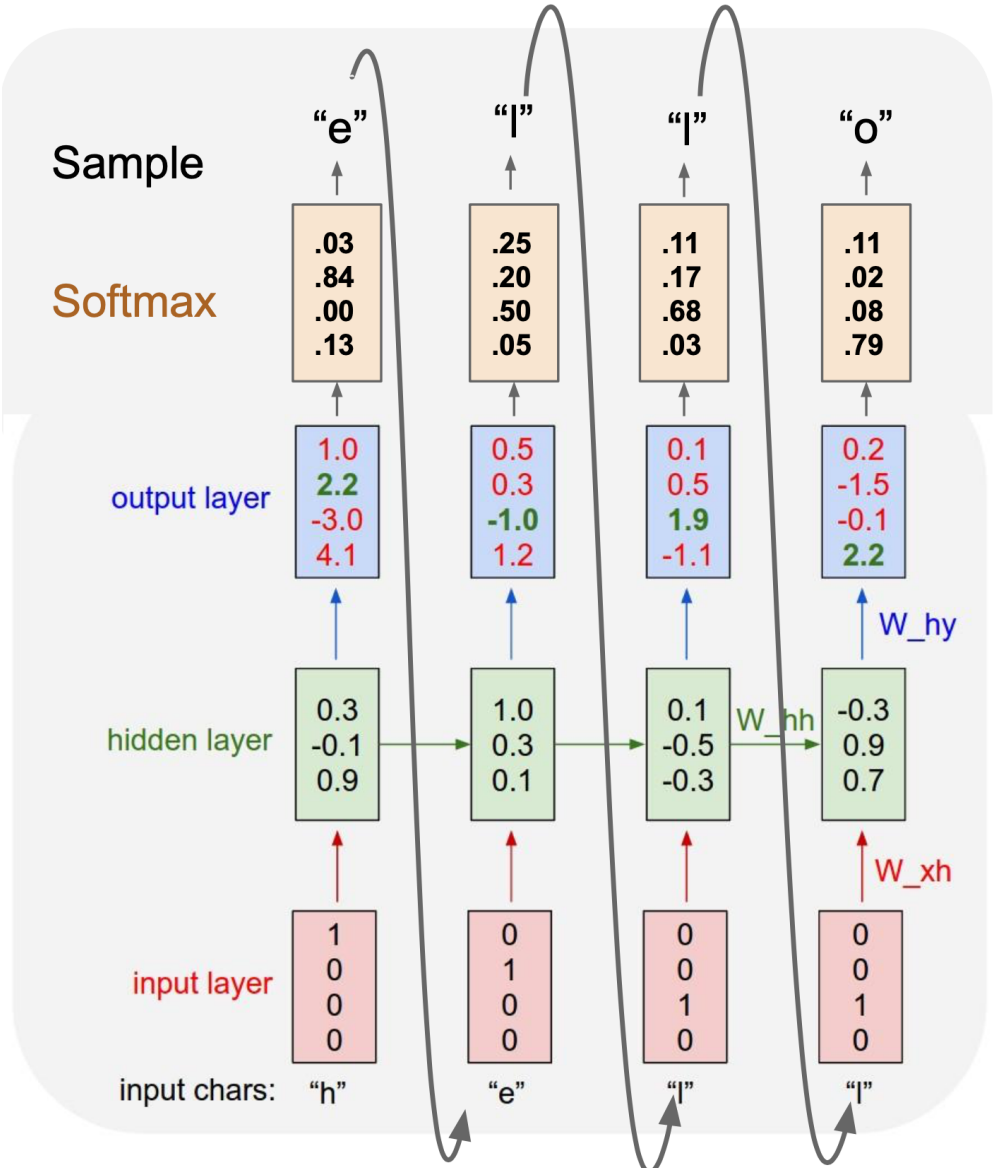
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$



Stanford CS231n Lecture Notes

A Toy Example from Andrej Karpathy

At every time step, have a softmax classifier over the next character



Another Character Level RNN Example (also from Karpathy)

- Apply a model analogous to the previous one on an actual corpus
- Sample a minibatch of 25 consecutive characters as the input, the target is offset by one character, since the objective is to predict what comes next
- Feed in characters as one hot vectors
- Essentially have 25 softmax classifiers for each batch; they all add up in backprop since using the same weights
- This model is just about sequences of characters. Doesn't know that words or higher level language concepts exist

Another Character Level RNN Example (also from Karpathy)

Initialize with some characters, and the RNN will give us the distribution over the next character in the sequence. Sample from the distribution, feed this into the next step, and continue to generate arbitrary text

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

Trained on Shakespeare's Work)

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

```
VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.
```

Stanford CS231n Lecture Notes

Trained on An Algebraic Topology Textbook)

For $\bigoplus_{n=1,\ldots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the compariciely in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, ?? and the fact that any U affine, see Morphisms, Lemma ??. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma ?? we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example ??. It may replace S by $X_{spaces,étale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma ??. Namely, by Lemma ?? we see that R is geometrically regular over S.

Lemma 0.1. *Assume (3) and (3) by the construction in the description.*

Suppose $X = \lim |X|$ *(by the formal open covering* X *and a single map* $\underline{Proj}_X(\mathcal{A}) = \mathrm{Spec}(B)$ *over* U *compatible with the complex*

$$Set(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X,\mathcal{O}_X}).$$

When in this case of to show that $\mathcal{Q} \to \mathcal{C}_{Z/X}$ *is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition ?? (without element is when the closed subschemes are catenary. If* T *is surjective we may assume that* T *is connected with residue fields of* S*. Moreover there exists a closed subspace* $Z \subset X$ *of* X *where* U *in* X' *is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem*

(1) f *is locally of finite type. Since* $S = \mathrm{Spec}(R)$ *and* $Y = \mathrm{Spec}(R)$.

Proof. This is form all sheaves of sheaves on X. But given a scheme U and a surjective étale morphism $U \to X$. Let $U \cap U = \coprod_{i=1,\ldots,n} U_i$ be the scheme X over S at the schemes $X_i \to X$ and $U = \lim_i X_i$. □

The following lemma surjective restrocomposes of this implies that $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{\mathcal{X},\ldots,0}$.

Lemma 0.2. *Let* X *be a locally Noetherian scheme over* S, $E = \mathcal{F}_{X/S}$. *Set* $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}'_n$. *Since* $\mathcal{I}^n \subset \mathcal{I}^n$ *are nonzero over* $i_0 \leq \mathfrak{p}$ *is a subset of* $\mathcal{J}_{n,0} \circ \overline{A}_2$ *works.*

Lemma 0.3. *In Situation ??. Hence we may assume* $\mathfrak{q}' = 0$.

Proof. We will use the property we see that $\mathfrak{p}$ is the mext functor (??). On the other hand, by Lemma ?? we see that

$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

where K is an F-algebra where δ_{n+1} is a scheme over S. □

Stanford CS231n Lecture Notes

Trained on An Algebraic Topology Textbook)

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram

$$\begin{array}{ccc} \mathcal{S} & \longrightarrow & \\ \downarrow & & \\ \xi & \longrightarrow & \mathcal{O}_{X'} \\ & & \uparrow \\ \text{gor}_s & & \\ & & \\ = \alpha' & \longrightarrow & \\ \updownarrow & & \\ = \alpha' & \longrightarrow & \alpha \end{array} \qquad \begin{array}{c} X \\ \downarrow \\ \end{array}$$

$$\text{Spec}(K_\psi) \qquad \text{Mor}_{Sets} \quad \text{d}(\mathcal{O}_{X_{fk}}, \mathcal{G})$$

is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \text{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \quad \text{-1}(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_t}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{s}})$$

is an isomorphism of covering of $\mathcal{O}_{X_i}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

Stanford CS231n Lecture Notes

Can do Way Better with Modern Language Models (GPT-2)

SYSTEM PROMPT (HUMAN-WRITTEN)	*A train carriage containing controlled nuclear materials was stolen in Cincinnati today. Its whereabouts are unknown.*
MODEL COMPLETION (MACHINE-WRITTEN, FIRST TRY)	The incident occurred on the downtown train line, which runs from Covington and Ashland stations. In an email to Ohio news outlets, the U.S. Department of Energy said it is working with the Federal Railroad Administration to find the thief. “The theft of this nuclear material will have significant negative consequences on public and environmental health, our workforce and the economy of our nation,” said Tom Hicks, the U.S. Energy Secretary, in a statement. “Our top priority is to secure the theft and ensure it doesn’t happen again.” The stolen material was taken from the University of Cincinnati’s Research Triangle Park nuclear research site, according to a news release from Department officials. The Nuclear Regulatory Commission did not immediately release any information. According to the release, the U.S. Department of Energy’s Office of Nuclear Material Safety and Security is leading that team’s investigation. “The safety of people, the environment and the nation’s nuclear stockpile is our highest priority,” Hicks said. “We will get to the bottom of this and make no excuses.

Open AI

Outline

RNNs in Practice

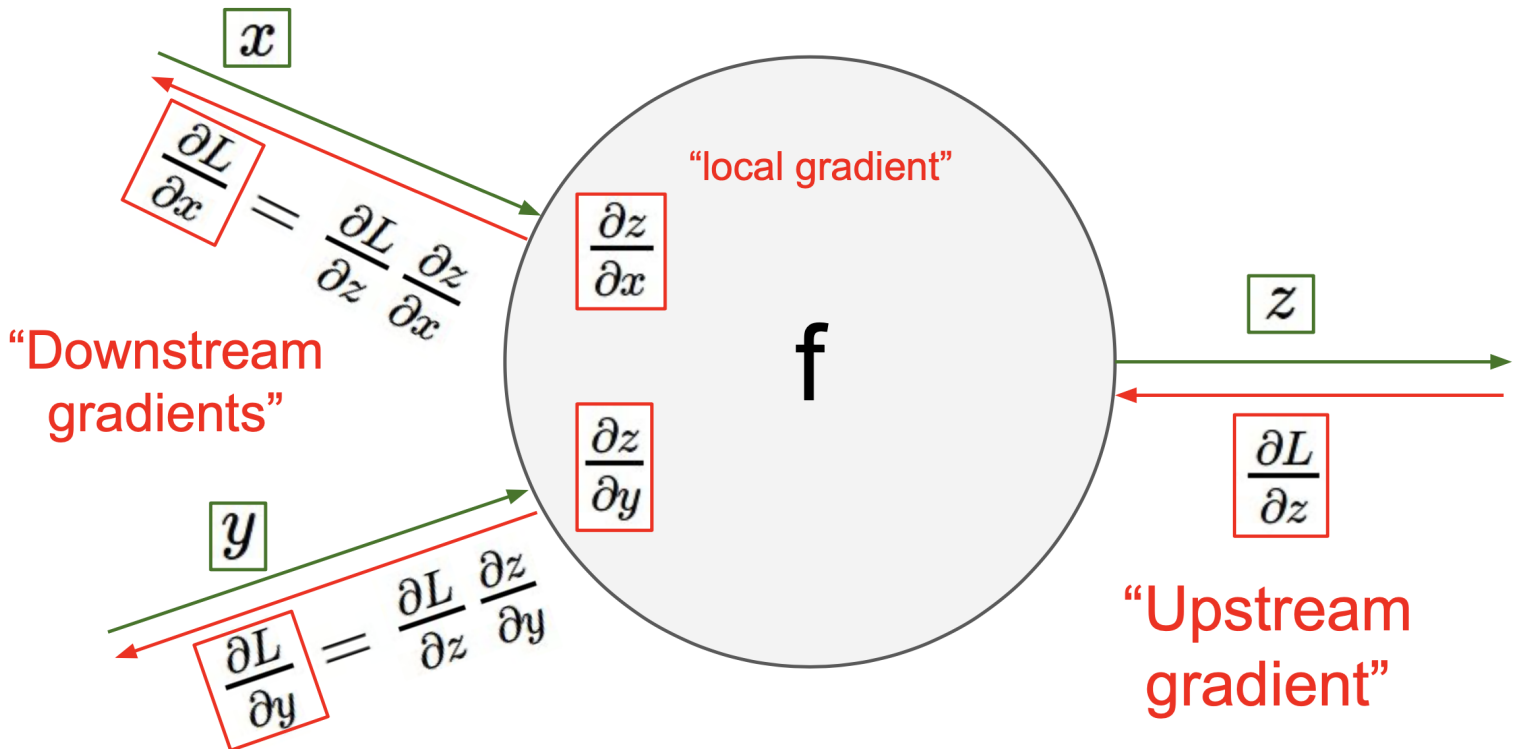
The RNN formulation that we saw earlier is not used in practice:

$$h_t = tanh\left(W\begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$

To understand why, we need to return to an earlier theme of the course: how gradients flow through networks

How Gradients Flow in Networks

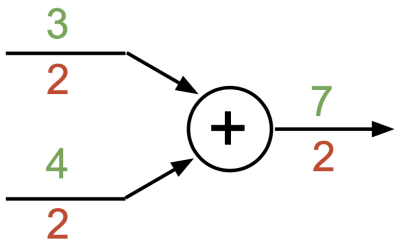
To backprop, we chain the local gradient with the upstream gradient



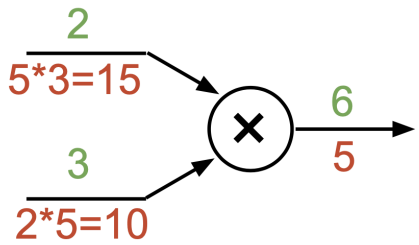
Stanford CS231n Lecture Notes

Computational Graphs

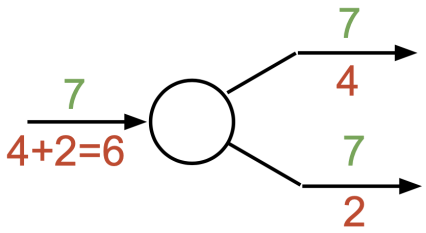
add gate: gradient distributor



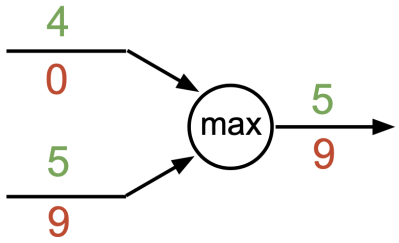
mul gate: "swap multiplier"



copy gate: gradient adder



max gate: gradient router



Stanford CS231n Lecture Notes

$$f(x) = x + y \rightarrow \frac{\partial f}{\partial x} = 1, \frac{\partial f}{\partial y} = 1$$

$$f(x) = ax \rightarrow \frac{\partial f}{\partial x} = a, \text{ etc.}$$

Gradient Flow in RNN

- Gradient flow complicates learning long-run dependencies (Bengio et al., 1994)
- Recall, what allows us to model arbitrary sequence lengths is using the same weight matrix at every step
- When we backprop to the earlier layers, the chain rule will entail matrix multiplication by W over and over again, since the network structure multiples by W at every time step in the forward pass
- If the largest singular value of W is > 1 the gradient will explode, if < 1 will vanish

Challenges of Learning Long-Term Dependencies

- We can avoid exploding gradients by clipping
- Unfortunately, vanishing gradients prevent us from training the network and necessitate a different architecture

RNNs in Practice

To solve this problem, we use LSTM (Hochreiter et al., 1997):

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

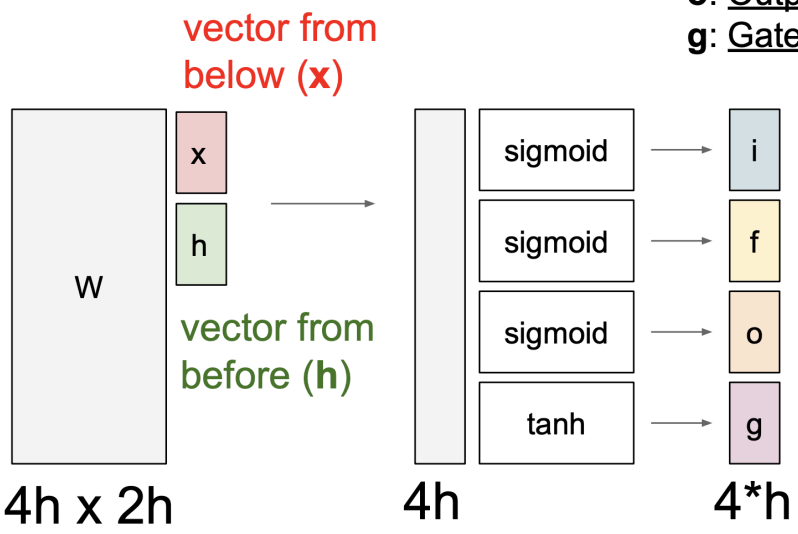
$$h_t = o \odot tanh(c_t)$$

$\odot$ is component wise multiplication for matrices

LSTMs

[Hochreiter et al., 1997]

i: Input gate, whether to write to cell
f: Forget gate, Whether to erase cell
o: Output gate, How much to reveal cell
g: Gate gate (?), How much to write to cell



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

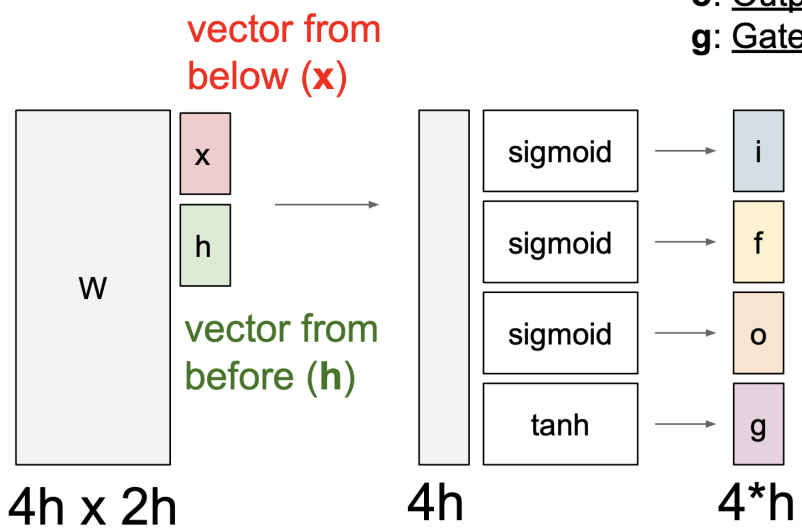
Stanford CS231n Lecture Notes

Think of these gates as being 0/1 (even though we squish them into sigmoids because we need differentiability)

LSTMs

[Hochreiter et al., 1997]

i: Input gate, whether to write to cell
f: Forget gate, Whether to erase cell
o: Output gate, How much to reveal cell
g: Gate gate (?), How much to write to cell



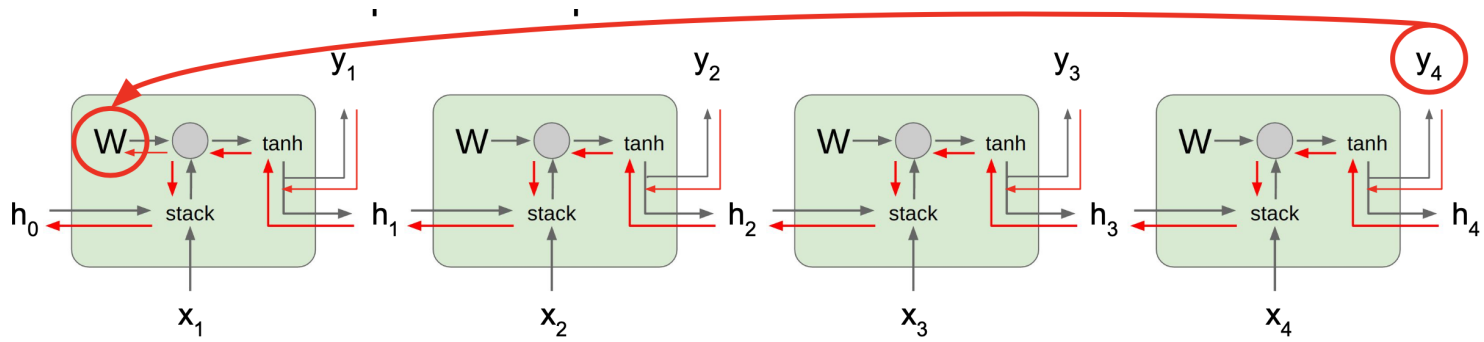
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

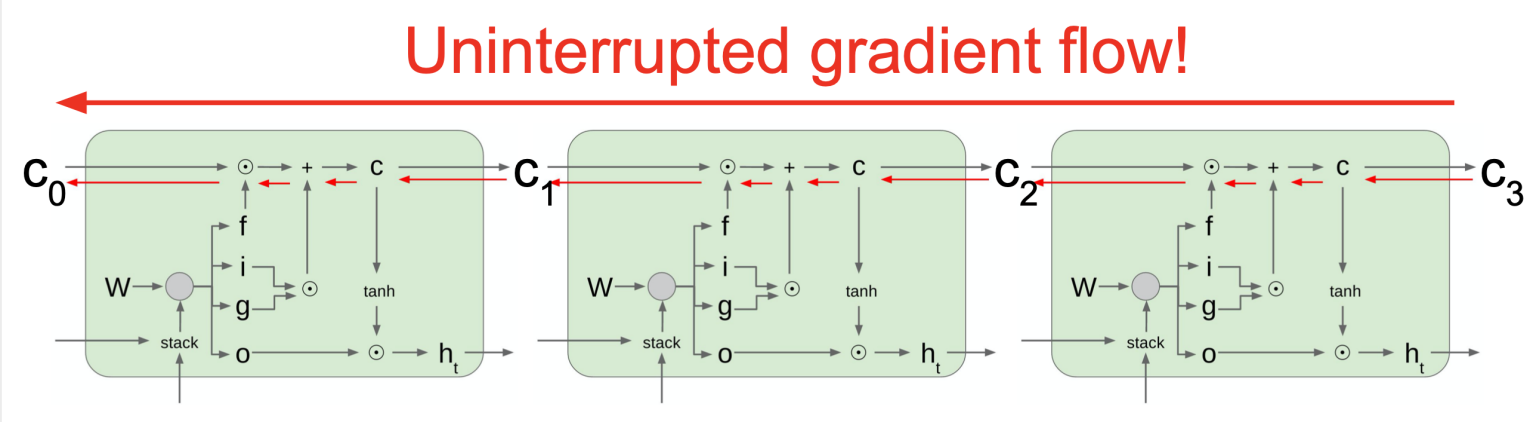
$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

Update the cell state by deciding whether to remember (forget gate), and deciding whether and how much (i and g) to write to a cell. Only allow part of cell value to enter the history (which determines output), in a learnable way (o).

Uninterrupted Gradient Flow

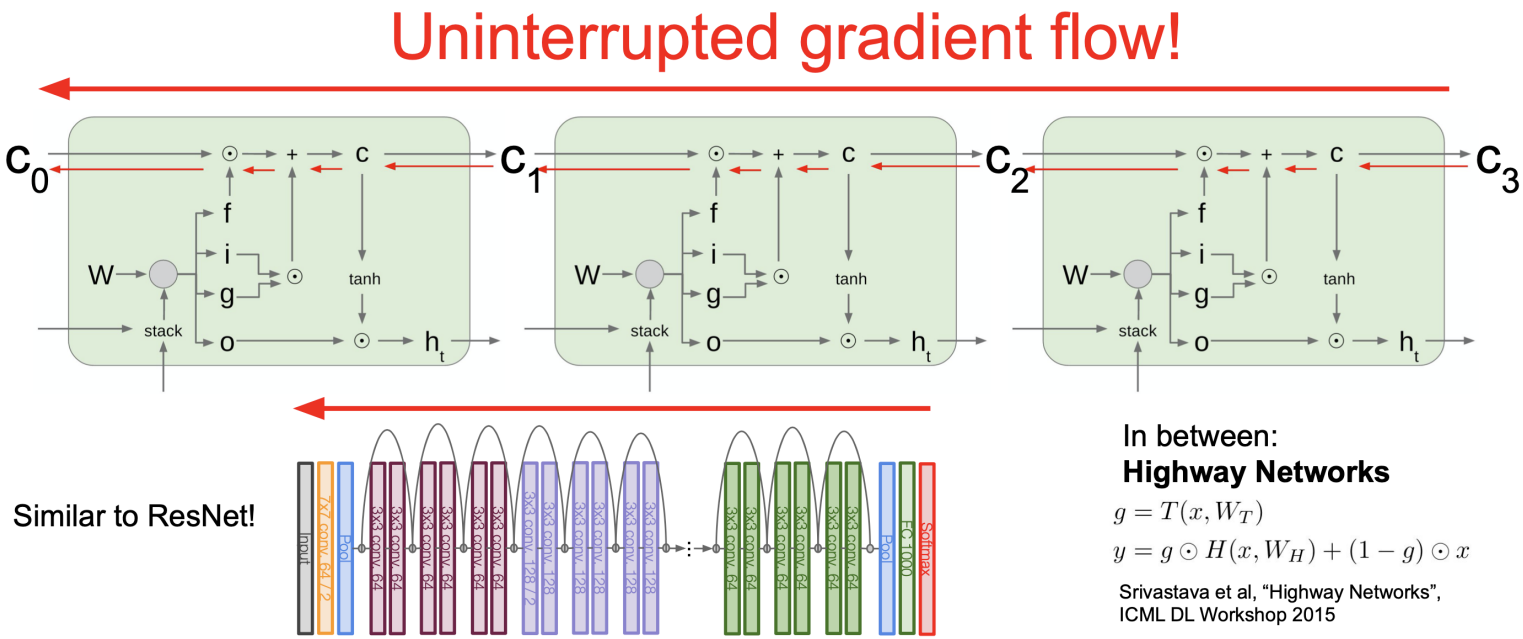




Stanford CS231n Lecture Notes

Provides a path for uninterrupted gradient flow because of the add gate (assuming you don't forget everything - avoid this by initializing the forget gate near 1).

Analogous to something we've seen before



Stanford CS231n Lecture Notes

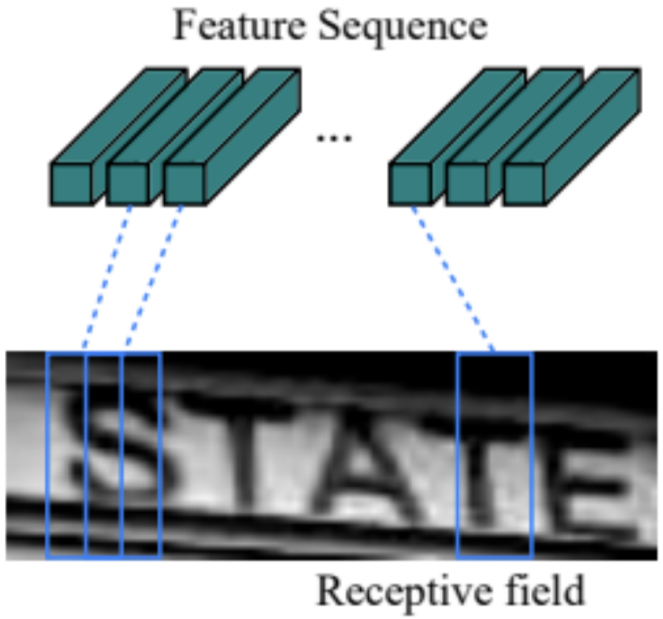
Skip connections in ResNet also act as add gates, allowing for unimpeded gradient flow.

LSTMs: Additional Notes

- A paper called LSTM: A Search Space Odyssey (Greff et al., 2015) documents in detail that results are fairly robust to changing around the details of the LSTM architecture
- Today, it is much less common to use LSTMs for natural language processing. Most modern NLP uses the Transformer architecture, which operates over inputs in a sequence in parallel through an attention mechanism, rather than processing inputs sequentially
- While this has led to much better results in many NLP tasks, it hasn't been systematically incorporated into OCR in a way that improves performance. As we'll discuss next class, depending on what you are OCRing, it's not even obvious you want this sort of language model

Coming Back to OCR

We feed features sequences - produced by the CNN - into the LSTM and make predictions over them at each time step



Shi et al., 2015

LSTMs in OCR

- The contextual information from the sequence that the RNN captures will tend to make the OCR more stable and accurate than if we treated each character independently
  - Wide characters take several frames to describe
  - Easier, for example, to distinguish an “i” from and “l” in a comparison than independently
  - The RNN captures information about what tends to appear together
- Using a bidirectional LSTM allows us to consider contexts in both directions: just combine two LSTMs, one forward and one backward
- Multiple bidirectional LSTMs can be stacked, resulting in a deeper structure that allows higher levels of abstraction

RNNs in OCR

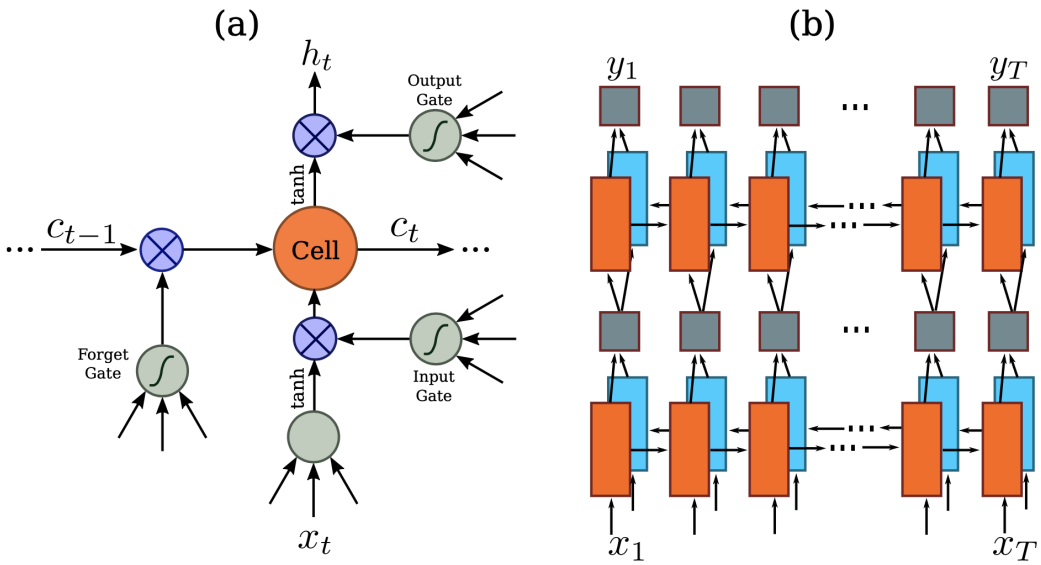
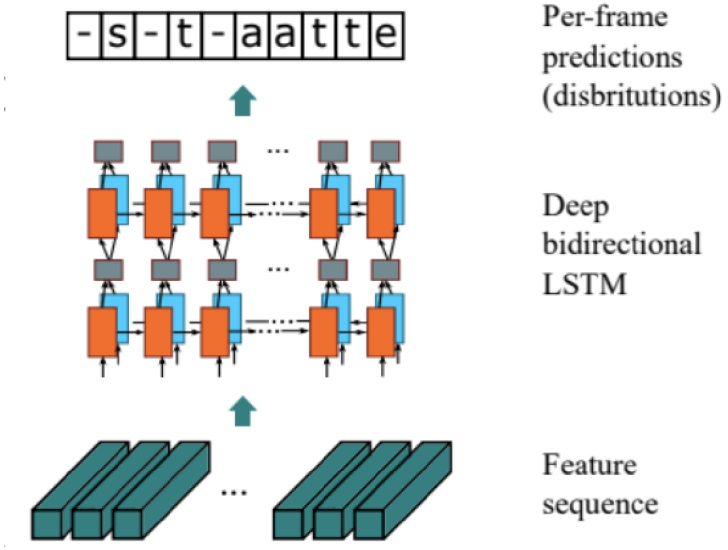


Figure 3. (a) The structure of a basic LSTM unit. An LSTM consists of a cell module and three gates, namely the input gate, the output gate and the forget gate. (b) The structure of deep bidirectional LSTM we use in our paper. Combining a forward (left to right) and a backward (right to left) LSTMs results in a bidirectional LSTM. Stacking multiple bidirectional LSTM results in a deep bidirectional LSTM.

Shi et al., 2015

Coming Back to OCR

The RNN outputs a character prediction for each features sequence



Shi et al., 2015

How do we compute the loss?

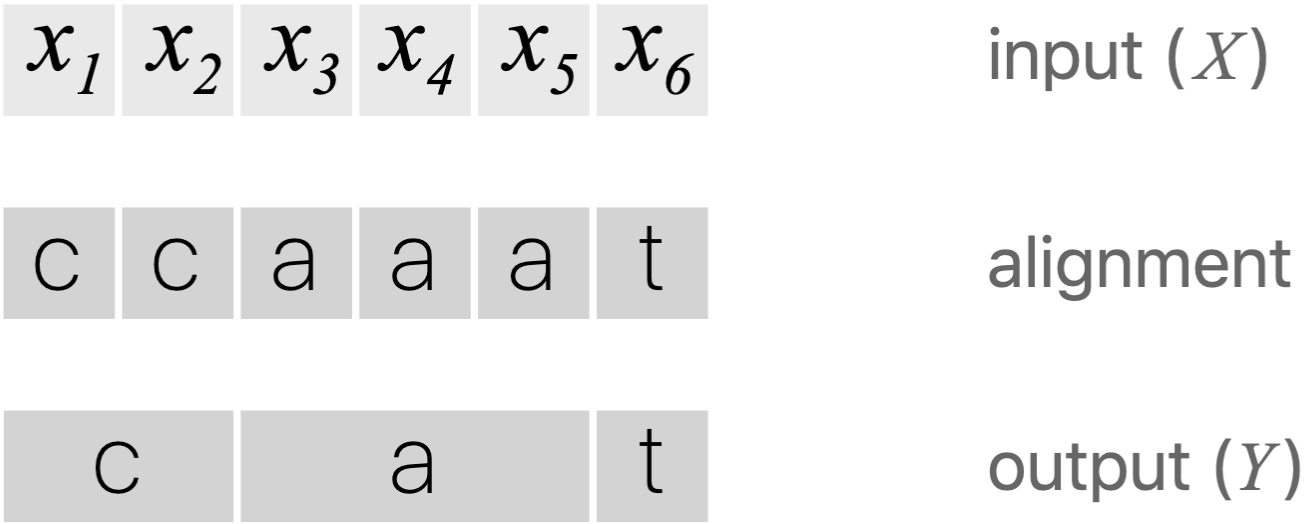
Outline

Connectionist Temporal Classification

- In OCR training data, we do not know the alignment between the input and the output. Annotators have typed what is on a page but do not label layouts of each character.
- The inputs and outputs can vary in length and the ratio between them is not constant.
- The same problem arises in speech recognition - where there are audio files and transcripts but not a precise alignment of where each syllable appears in the audio file.
- If one could get data augmentation to work well, presumably we would no longer have this problem with OCR.

Naive Approach

Assign an output character to each input step and collapse duplicates

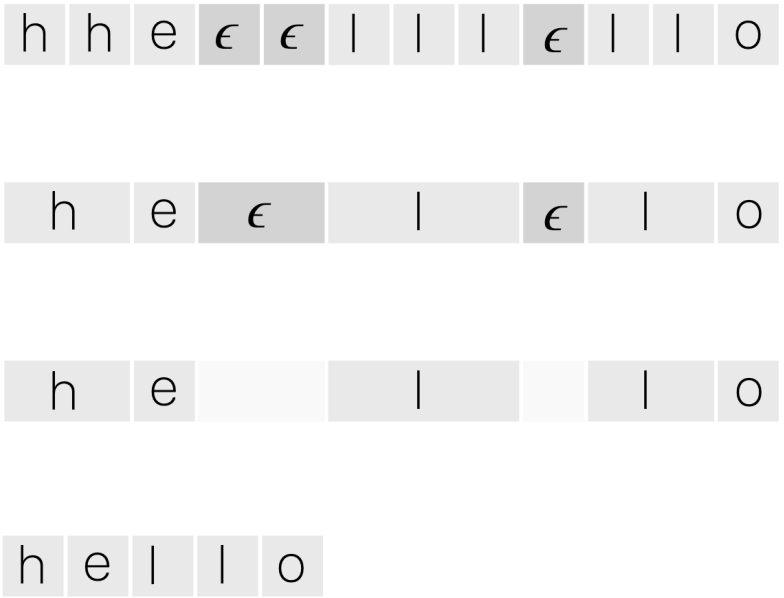


Hannun et al., 2017

Problem: lots of words have duplicate letters

CTC

CTC adds a blank token, called ϵ



First, merge repeat characters.

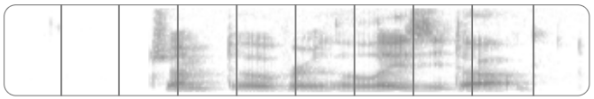
Then, remove any ϵ tokens.

The remaining characters are the output.

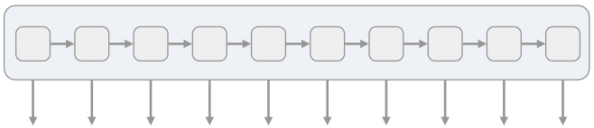
Hannun et al., 2017

Mapping from input to output is many to one.

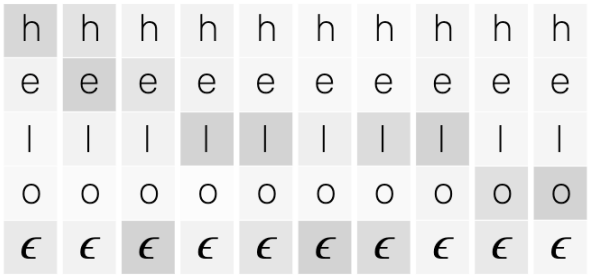
CTC Illustrated



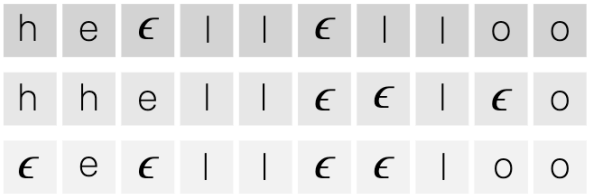
We start with an input sequence, like a spectrogram of audio.



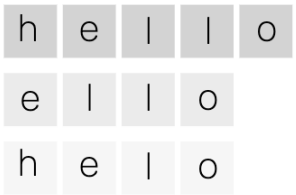
The input is fed into an RNN, for example.



The network gives $p_t(a \mid X)$, a distribution over the outputs {h, e, l, o, ϵ} for each input step.



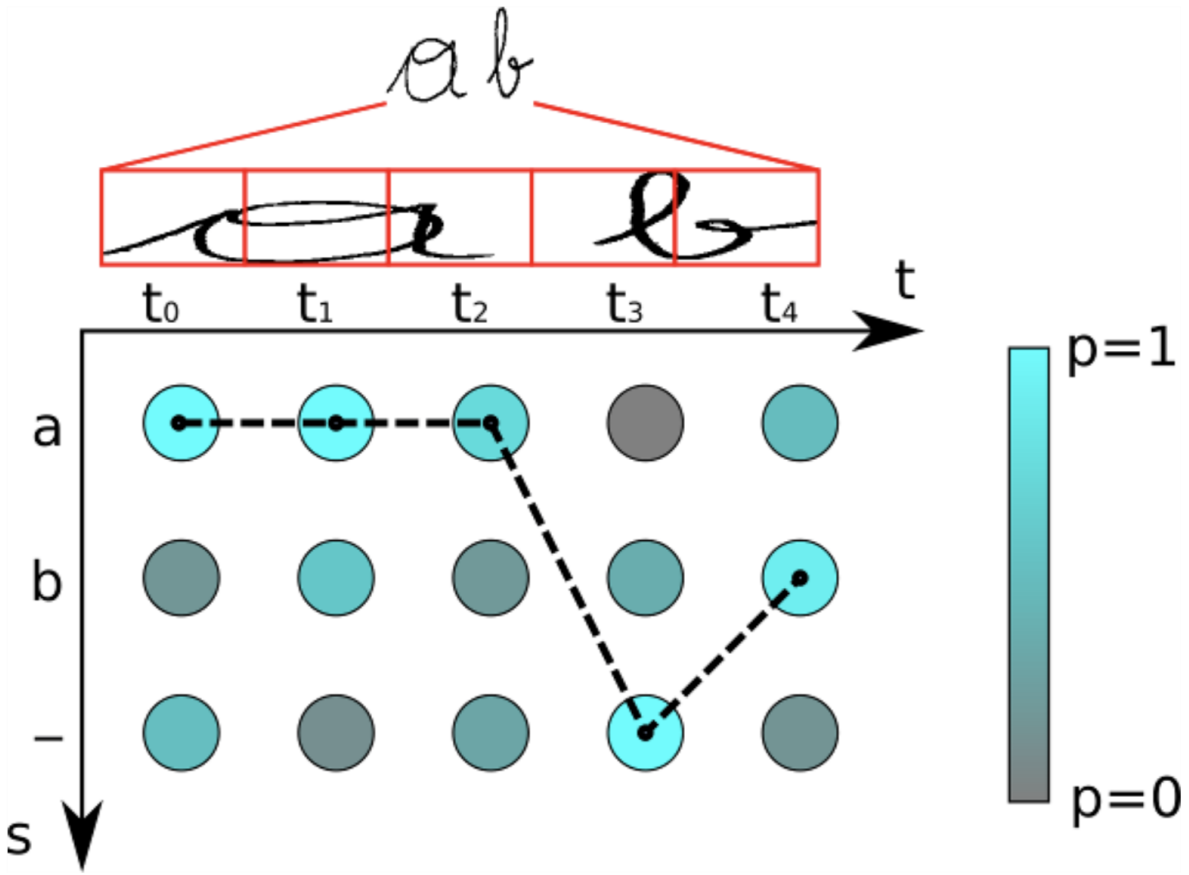
With the per time-step output distribution, we compute the probability of different sequences



By marginalizing over alignments, we get a distribution over outputs.

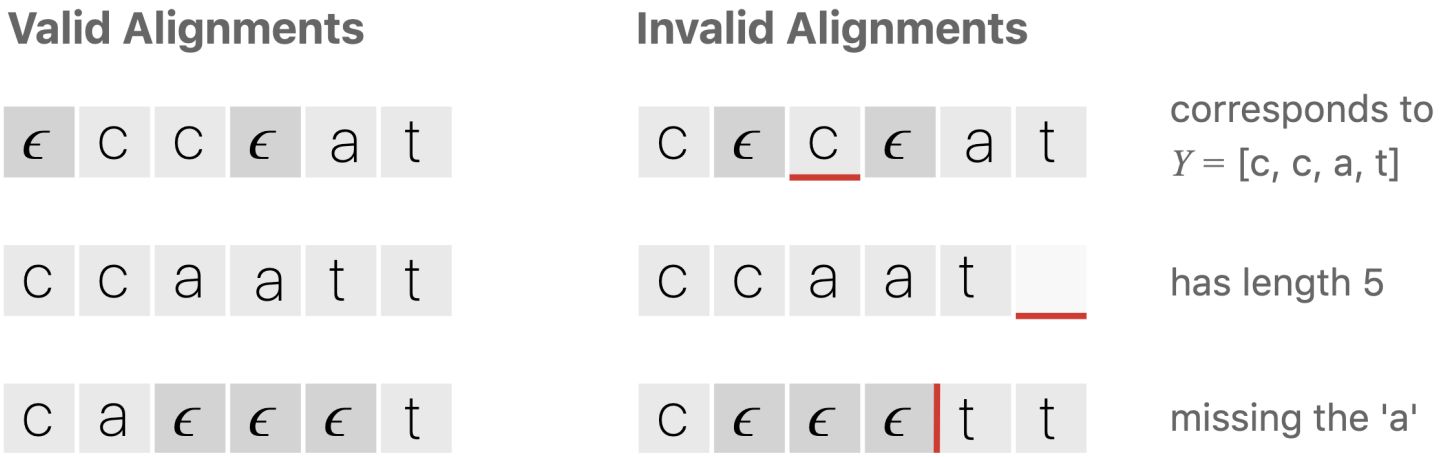
Hannun et al., 2017

CTC Illustrated



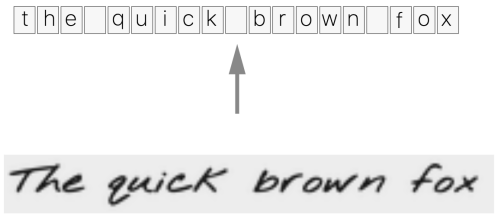
Hannun et al., 2017

Valid and Invalid Alignments

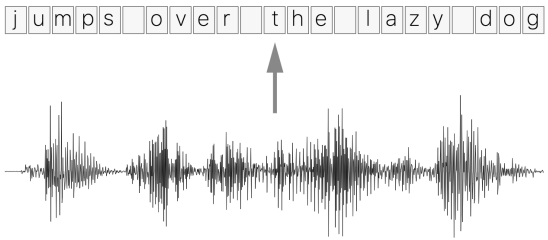


Hannun et al., 2017

Analogous for OCR and Text-to-Speech



Handwriting recognition: The input can be (x, y) coordinates of a pen stroke or pixels in an image.



Speech recognition: The input can be a spectrogram or some other frequency based feature extractor.

Hannun et al., 2017

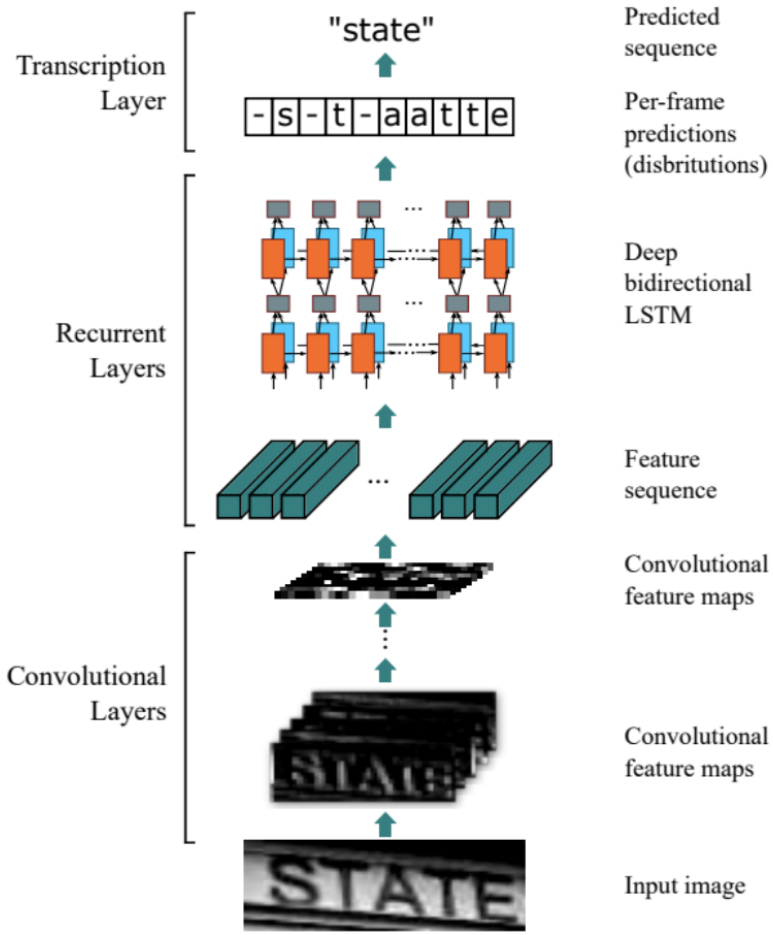
CTC Loss

- Usually the CTC loss is used in combination with an RNN that estimates per time step probabilities, as above
- CTC loss could be very expensive to compute because there can be a very large number of alignments; see the paper for how a dynamic programming algorithm can be used to compute the loss much more quickly by merging alignments that have reached the same output at the same step

Outline

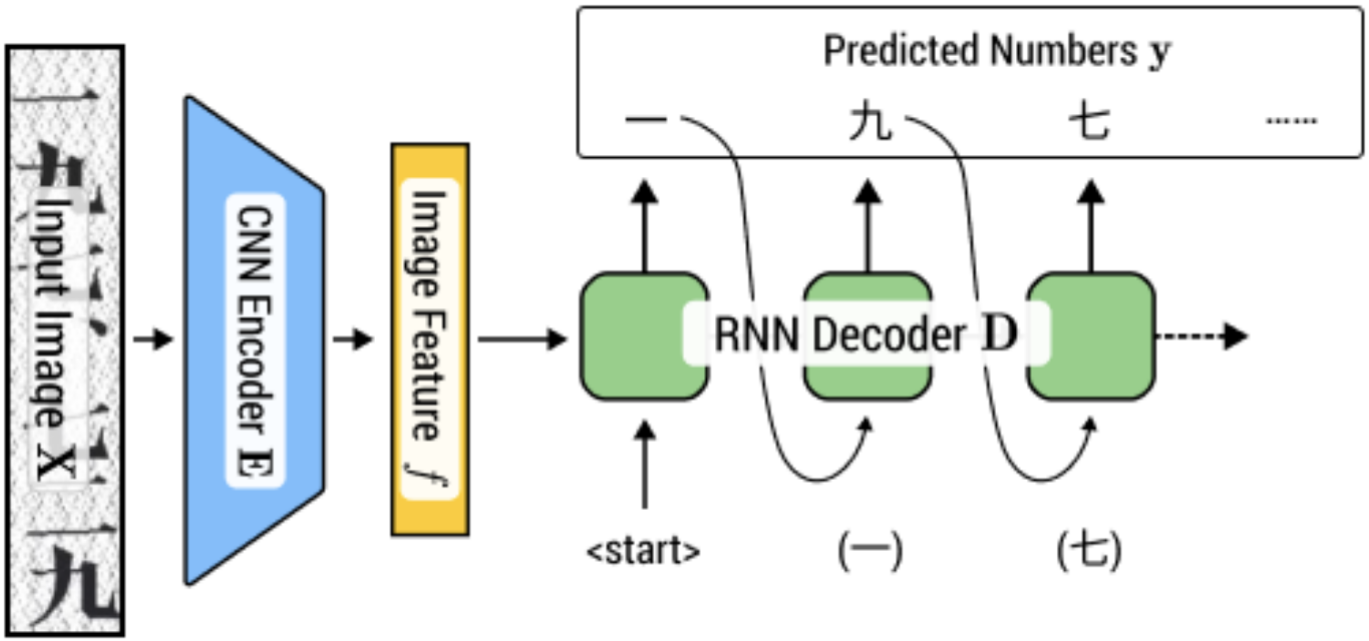
Putting it Together

OCR stacks a CNN, RNN, and CTC loss together for an end-to-end solution.



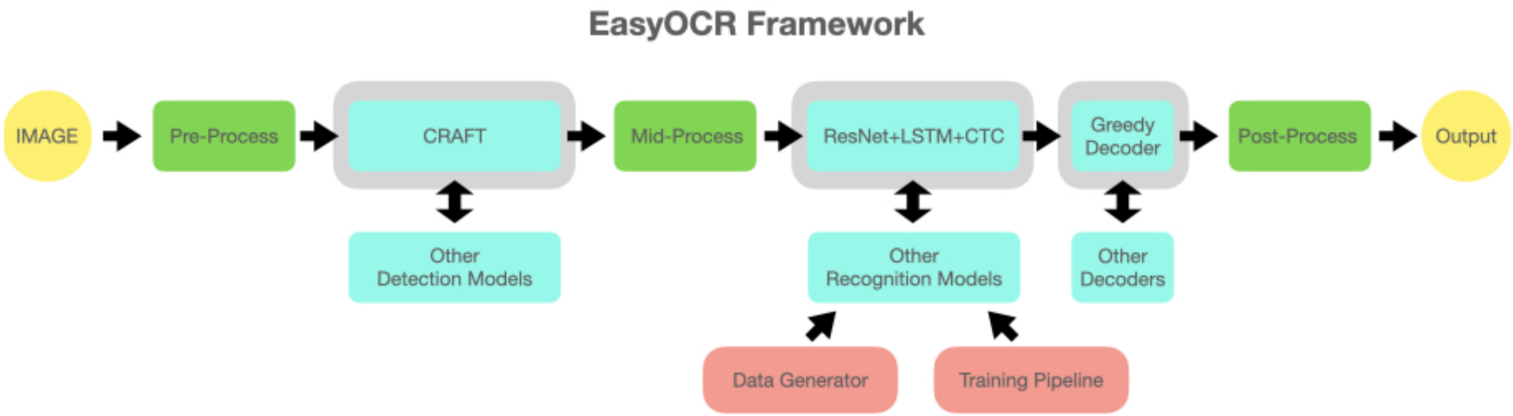
Shi et al., 2015 "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition"

Our Custom Numbers OCR Pipeline



Easy OCR

Open-source package primarily for OCR in the wild



We'll discuss OCR in practice next class.