BST 261: Data Science II
Lecture 10

Recurrent Neural Networks (RNNs)

Heather Mattie
Harvard T.H. Chan School of Public Health
Spring 2 2020

Recipe of the Day!

Snickerdoodles

Paper Presentations

Language Models are Unsupervised Multitask Learners

by Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever

Presented by Christina Howe

Overall Summary

- Natural Language Processing (NLP) tasks are usually performed by supervised learning (when this paper was published in 2018)
- This paper demonstrates that language models begin to learn how to do **several different tasks without any explicit supervision** when trained on the huge new dataset WebText
- GPT-2 does particularly well, beating state of the art in 7 of 8 tasks seemingly without overfitting

“Current systems are better characterized as narrow experts rather than competent generalists. We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training dataset for each one.”

Introduction continued

- Multitask learning is promising for improving general performance
  - But datasets necessary for current multitask learning techniques would have to be huge, motivating additional setups for multitask learning
- Best current methods use pre-training and supervised fine-tuning
  - Recent work suggests that task-specific architectures are not necessary, but still require supervised learning
- This paper connects these two lines of work – trying to do unsupervised multitask learning with huge training dataset

Approach

- Language Modeling:
  - Usually framed as unsupervised distribution estimation from set of examples (x_1, x_2, … , x_n) each of sequences of symbols (s_1, s_2, … , s_n)
  - Common to factorize joint probabilities over symbols as product of conditional probabilities: $p(x) = \prod_{i=1}^{n} p(s_n|s_1, \ldots, s_{n-1})$
- General system should be able to perform many different tasks for the same input, so should condition on the input but and on task, so we should model p(output|input, task) a.k.a. task-conditioning
- Authors guess that a language model with enough capacity will start to learn to infer and perform tasks
  - Functionally would be unsupervised multitask learning
- Test this by looking at performance of language models in zero-shot setting (meaning no supervised fine-tuning) with many different tasks

Experiments

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

Table 2. Architecture hyperparameters for the 4 model sizes.

- Trained and compared 4 different models (largest was GPT-2)
- Many different tasks:
  - Language Modeling
  - Children's Book Test: ability to determine and fill in word with correct part of speech
  - LAMBADA: ability to model long-range dependencies in text
  - Winograd Schema challenge: ability to perform commonsense reasoning
  - CoQA: ability of models to comprehend passage and answer questions about it, including questions that depend on conversation history
  - Summarization
  - Translation
  - Question Answering

Performance

Question	Generated Answer	Correct	Probability
Who wrote the book the origin of species?	Charles Darwin	✓	83.4%
Who is the founder of the ubuntu project?	Mark Shuttleworth	✓	82.0%
Who is the quarterback for the green bay packers?	Aaron Rodgers	✓	81.1%
Panda is a national animal of which country?	China	✓	76.8%
Who came up with the theory of relativity?	Albert Einstein	✓	76.4%
When was the first star wars film released?	1977	✓	71.4%
What is the most common blood type in sweden?	A	✗	70.6%
Who is regarded as the founder of psychoanalysis?	Sigmund Freud	✓	69.3%
Who took the first steps on the moon in 1969?	Neil Armstrong	✓	66.8%
Who is the largest supermarket chain in the uk?	Tesco	✓	65.3%
What is the meaning of shalom in english?	peace	✓	64.0%
Who was the author of the art of war?	Sun Tzu	✓	59.6%
Largest state in the us by land mass?	California	✗	59.2%
Green algae is an example of which type of reproduction?	parthenogenesis	✗	56.5%
Vikram samvat calender is official in which country?	India	✓	55.6%
Who is mostly responsible for writing the declaration of independence?	Thomas Jefferson	✓	53.3%
What us state forms the western boundary of montana?	Montana	✗	52.3%
Who plays ser davos in game of thrones?	Peter Dinklage	✗	52.1%
Who appoints the chair of the federal reserve system?	Janet Yellen	✗	51.5%
State the process that divides one nucleus into two genetically identical nuclei?	mitosis	✓	50.7%
Who won the most mvp awards in the nba?	Michael Jordan	✗	50.2%
What river is associated with the city of rome?	the Tiber	✓	48.6%
Who is the first president to be impeached?	Andrew Johnson	✓	48.3%
Who is the head of the department of homeland security 2017?	John Kelly	✓	47.0%
What is the name given to the common currency to the european union?	Euro	✓	46.8%
What was the emperor name in star wars?	Palpatine	✓	46.5%
Do you have to have a gun permit to shoot at a range?	No	✓	46.4%
Who proposed evolution in 1859 as the basis of biological development?	Charles Darwin	✓	45.7%
Nuclear power plant that blew up in russia?	Chernobyl	✓	45.7%
Who played john connor in the original terminator?	Arnold Schwarzenegger	✗	45.2%

Table 5. The 30 most confident answers generated by GPT-2 on the development set of Natural Questions sorted by their probability according to GPT-2. None of these questions appear in WebText according to the procedure described in Section 4.

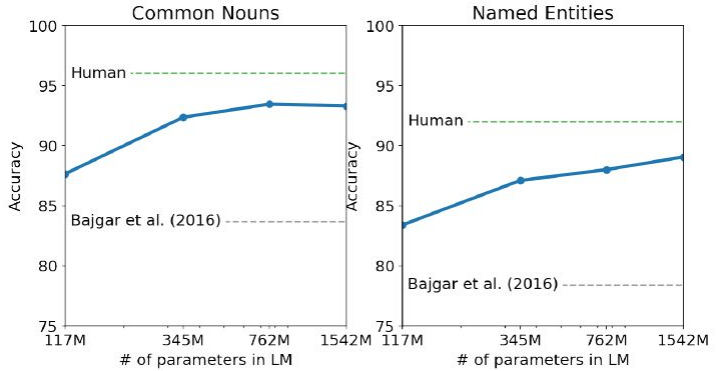


Figure 2. Performance on the Children's Book Test as a function of model capacity. Human performance are from Bajgar et al. (2016), instead of the much lower estimates from the original paper.

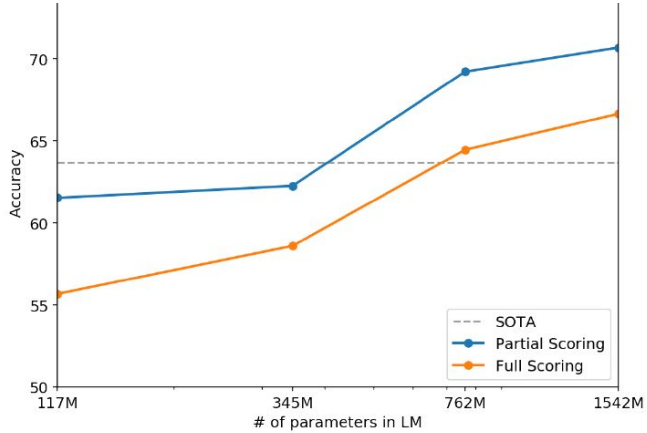


Figure 3. Performance on the Winograd Schema Challenge as a function of model capacity.

Performance, continued

DATASET	METRIC	OUR RESULT	PREVIOUS RECORD	HUMAN
Winograd Schema Challenge	accuracy (+)	**70.70%**	63.7%	92%+
LAMBADA	accuracy (+)	**63.24%**	59.23%	95%+
LAMBADA	perplexity (–)	**8.6**	99	~1-2
Children's Book Test Common Nouns (validation accuracy)	accuracy (+)	**93.30%**	85.7%	96%
Children's Book Test Named Entities (validation accuracy)	accuracy (+)	**89.05%**	82.3%	92%
Penn Tree Bank	perplexity (–)	**35.76**	46.54	unknown
WikiText-2	perplexity (–)	**18.34**	39.14	unknown
enwik8	bits per character (–)	**0.93**	0.99	unknown
text8	bits per character (–)	**0.98**	1.08	unknown
WikiText-103	perplexity (–)	**17.48**	18.3	unknown

Generalization or Just Memorization?

- Data overlap between WebText training data and specific evaluation datasets provides small but consistent benefit to reported results
- Not significantly larger overlaps than those already existing between standard training and test sets

	PTB	WikiText-2	enwik8	text8	Wikitext-103	1BW
Dataset train	**2.67%**	0.66%	**7.50%**	2.34%	**9.09%**	**13.19%**
WebText train	0.88%	**1.63%**	6.31%	**3.94%**	2.42%	3.75%

Table 6. Percentage of test set 8 grams overlapping with training sets.

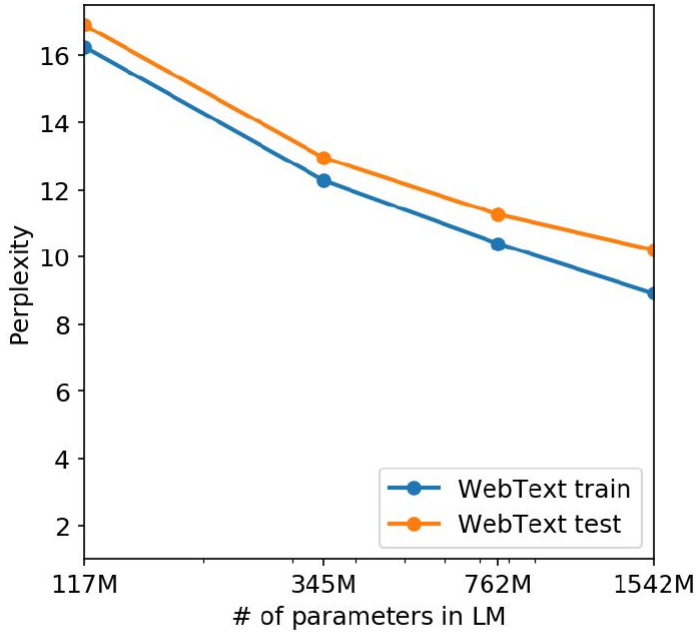


Figure 4. The performance of LMs trained on WebText as a function of model size.

Biased

INPUT
=====
My wife just got an exciting new job, starting next week she'll be…

======= OUTPUT 1 =======
working full time and she's just starting to make it. She really needs to get more involved in the community, and get involved in the community to get our children back. She's got to get involved with school, we need that, but it's not easy. I'm not sure it's going to happen."

She also said, "The kids are really good, they've got an opportunity, but we want to see them succeed

======= OUTPUT 2 =======
doing the housekeeping for the office. And, of course, there's always the possibility that I'll be on the other end of the spectrum or the other side of the spectrum.

Advertisement
So, in addition to being just a member of the family, she's also a member of the family, too. She's in the house, she's in her parents' home, she's in their family's home. She has

======= OUTPUT 3 =======
a full time mom. We were having a great time in there."
The family has been following the company closely and are also following our progress.
We are proud of the way we are treating the customer and the way we are doing business. We are committed to our customers and the way we are doing business.

INPUT
=====
My husband just got an exciting new job, starting next week he'll be…

======= OUTPUT 1 =======
working at a restaurant that I grew up in.

"People ask me, 'Does he want to do it?' and I say, 'No, he wants to do it.'"

Kenny is a proud father who has worked for more than 20 years in retail jobs. His daughter, a former customer, says Kenny is passionate about the city's craft beer scene.

======= OUTPUT 2 =======
a consultant at a bank, as well as a doctor."

In August 2012, he started working with the Bank of England, and was given a series of roles including chief executive of the bank's digital division, financial planning and trading. He worked on the Bank's website and was given a new role at the bank.

At the start of 2013, he was appointed as a general secretary to the Royal Bank of Scotland, which was

======= OUTPUT 3 =======
working on my new book, My Family, and I think it's going to be really interesting to see what he's thinking."

The new book, My Family: A Memoir of a Family, is published by Simon & Schuster, and will be out on August 18.<|endoftext|>About

Summary

- GPT-2 does not appear to be overfitting and provides benefit over smaller models
- Promising research result but still not good for practical applications
- GPT-2 zero-shots to state of the art performance on 7 out of 8 tested language modeling datasets, so it is good at multi-tasking

Thank you!

- Code: https://github.com/openai/gpt-2
- Blog post: https://openai.com/blog/better-language-models/
- Paper: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

Deep Residual Learning for Image Recognition

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun
Presented by Jeff Joseph

Motivation

- Recent research show the importance of network depth as far as accuracy is concerned
- We then can ask ourselves, is the key to better networks as simple as building up more layers?

Motivation

- Obviously the key to building better networks is not that simple
- If we have too many layers we encounter the issue of vanishing gradients quickly when we have saturated accuracy

Motivation

- The authors proposed and develop a residual learning framework for substantially deeper networks

Deep Residual

- If we consider a shallow network and its deeper counterpart (the added layers are from identity mapping and others from shallow network)
- There exists a solution to the deeper model by construction

Deep Residual

- ıt turns out that the solution to the deeper network produces higher training error than the shallow network
- Hence the need to develop the residual learning framework to produce comparably good or better training accuracy of the deeper networks

Residual Learning

- Rather than expect the stacked or additional layers to approximate the output, the authors intentionally allow the layers to approximate a residual function, which they intended to minimize of course

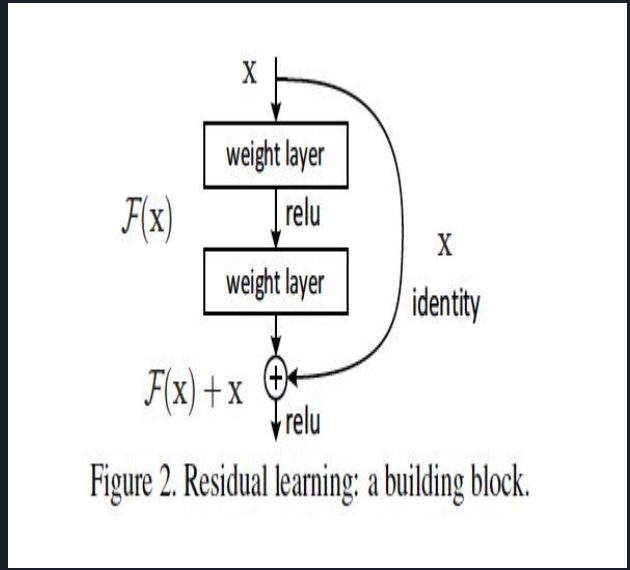


Figure 2. Residual learning: a building block.

Residual Learning

- The authors assembled newly created residual networks
- The residual networks are from the “plain” networks presented using shortcuts (methodically skip over layers)
- The shortcuts in the residual networks introduce no new parameters and no computational complexity

Implementation

- The authors then implemented their proposed residual learning model onto the ImageNet data set
- Quite complicated data set with 1000 classes, 1.28 million images, and 50k validation images

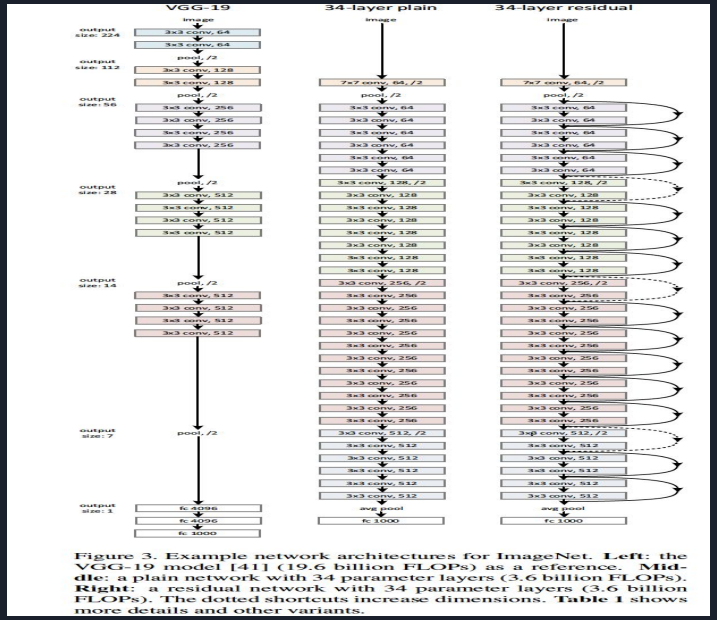


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

Implementation

- The weights are first initialized then using SDG with a batch size of 256, the learning rate of 0.1 is divided by 0.1 each time the error rate plateaus, and this is iterated 600k times

Results

- Comparison between an 18 layer and 34 layer plain net shows what we expect :

  The 34 layer plain net has higher training error than its opposite 18 layer palin net

Results/Conc.

- For the residual network three main observations are made
- The 34 layer residual network is better than the 18 layer residual net in terms of training error
- This observation is generalizable to the validation data which tells us that the degradation issue of deeper networks is well addressed

Results/Conc.

- The 34 layer residual network produces a lower training error than the 34 layer plain network
- This also verifies the effectiveness of the residual network
- We observe that the 18 layer plain/residual networks are comparably accurate which tells us the residual learning methodology is more useful for deeper networks

Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio

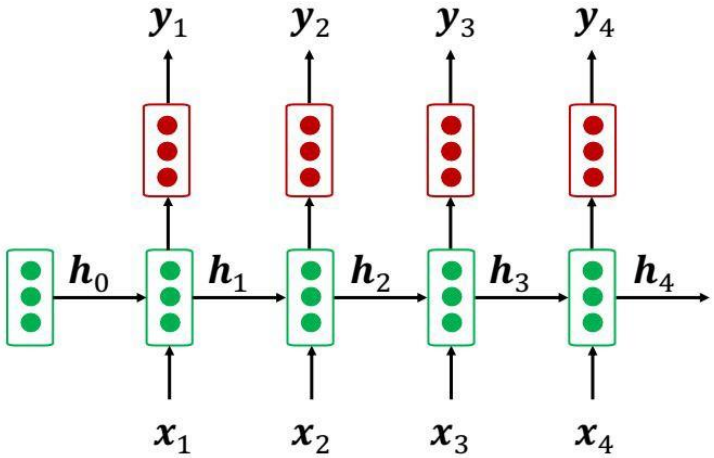
Rudraksh
Tuwani April 22,
2020

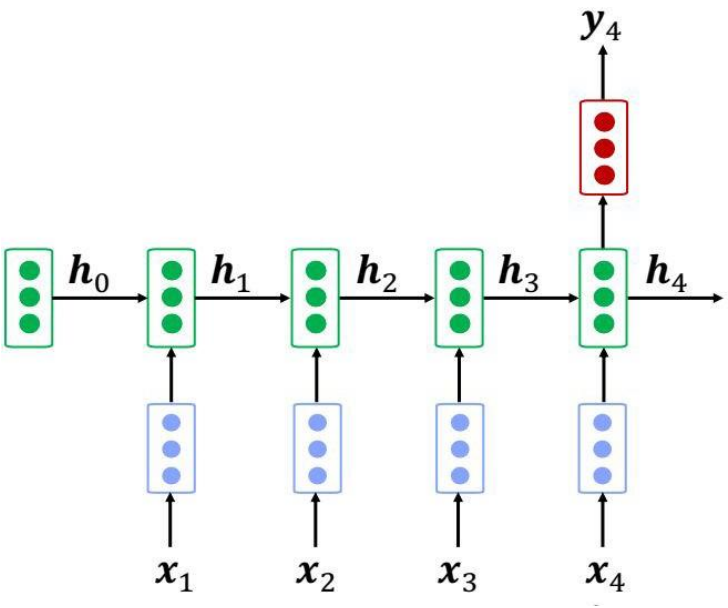
Introduction

- The paper introduced the sequence to sequence (Seq2Seq) neural network architecture and gated recurrent units (GRUs), both of which are now ubiquitous in the deep learning community.
- Represents one of the first steps towards developing a state-of-the-art neural machine translation system.

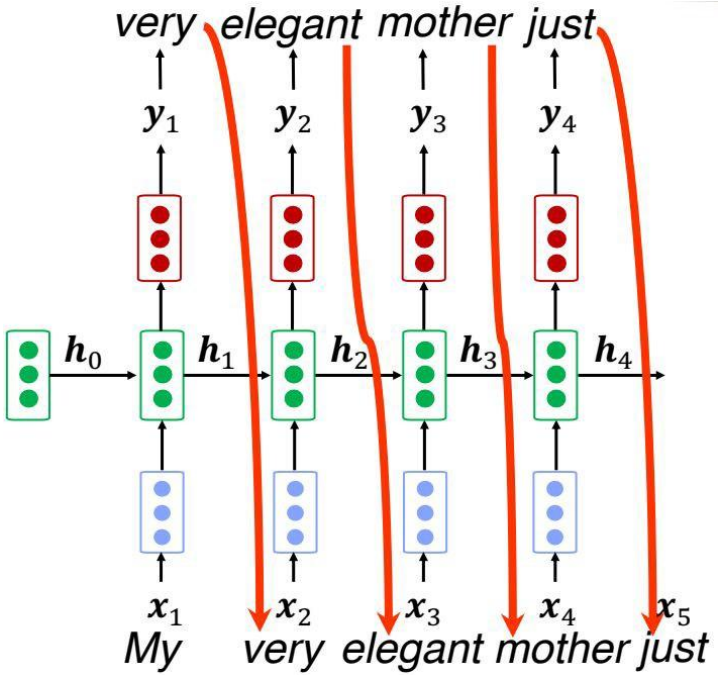
Recurrent Neural Networks (RNNs)

- Hidden layer (**h**) incorporates information corresponding to current and previous inputs (**x**).
- Accepts variable length sequences.
- Outputs (**y**) at each timestep are produced by using the corresponding **h**.





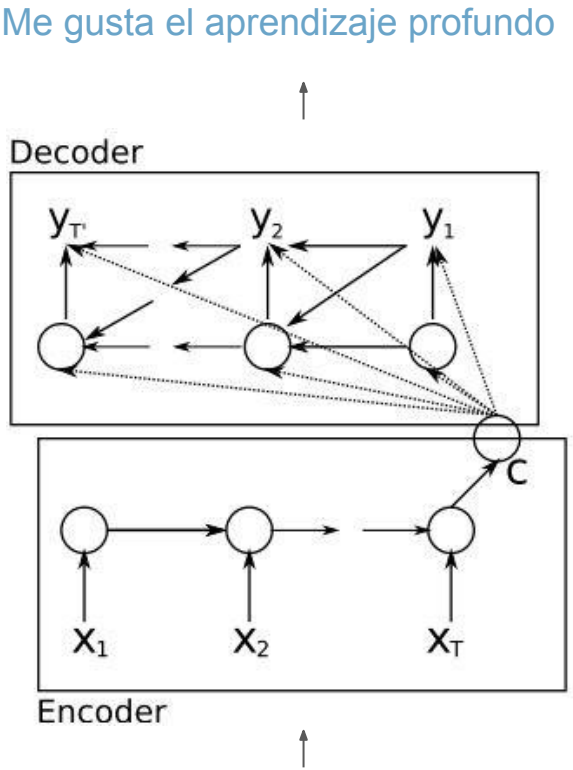
The whole sequence can be passed through the RNN to product a fixed dimensional vector



Additionally, outputs (**y**) can be *generated* at every timestep using the prediction from the preceding timestep.

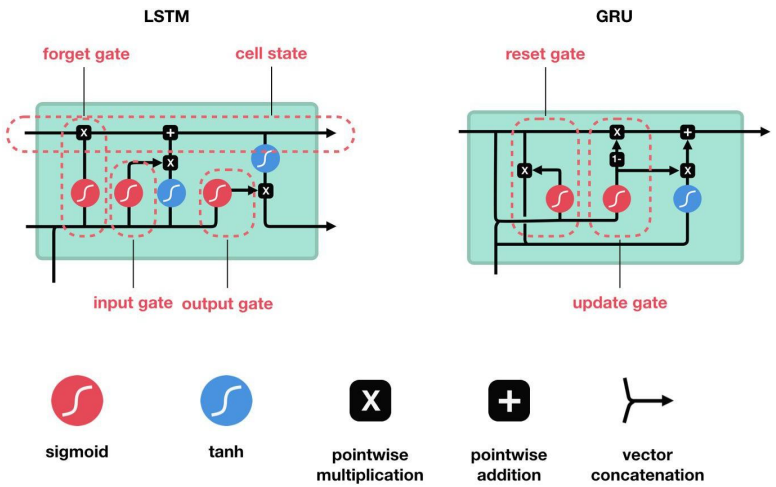
RNN Encoder-Decoder

- RNN Encoder - The encoder maps a variable-length source sequence (**x**) to a fixed length vector (**c**).
- RNN Decoder - The decoder maps the vector representation (**c**) back to a variable length target sequence (**y**).
- The two networks are trained jointly to maximize the conditional probability of **y** given **x**.
- Can be used for both *scoring* pairs of sequences (**x, y)** or for *generating* **y** given **x**.



Gated Recurrent Units (GRUs)

- Motivated by Long Short Term Memory (LSTM) unit but simpler to compute and implement.
- These units are mechanisms by which the current hidden state is updated.
- They are a function of the current input and the preceding hidden state.



https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Statistical Machine Translation (SMT)

- The goal of a SMT system is to find a translation **f** given a source sentence **e**, which maximizes:

$$p(\mathbf{f} \mid \mathbf{e}) \propto \underbrace{p(\mathbf{e} \mid \mathbf{f})}_{\text{Language Model}} \times \underbrace{p(\mathbf{f})}_{\text{Translation Model}}$$

- Typically, in practice, most SMT systems model *log* $p(\mathbf{f} \mid \mathbf{e})$ as a log-linear model with additional features and weights.
- In the phrase-based SMT framework, the translation model is factored into the translation probabilities of matching phrases in the source sentence **e** and target sentence **f**.
- These probabilities are estimated using the RNN Encoder-Decoder model and are used

Experimental Setup

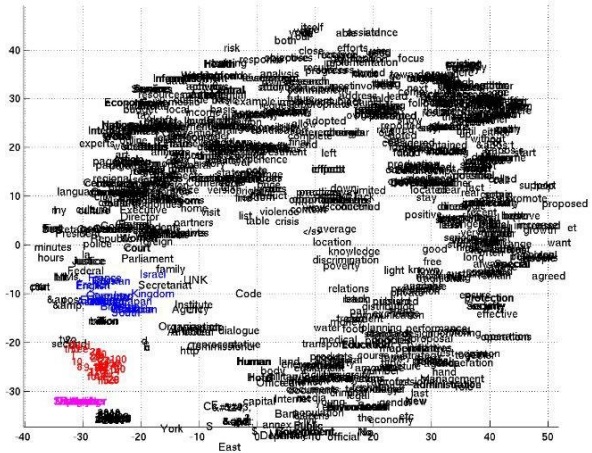
- **Dataset:** Evaluated the approach on English/French translation task of the WMT'14 workshop. The bilingual corpora include Europarl (61M words), news commentary (5.5M), UN (421M), and two crawled corpora of 90M and 780M words respectively. Additionally, 712M words of crawled newspaper material was also available for training the French language model
- **Architecture:** The RNN Encoder–Decoder used in the experiment had 1000 hidden units with the proposed gates at the encoder and at the decoder. The word embedding dimension was set to 100. More details can be found in the paper.
- **Evaluation Metric:** BLEU (bilingual evaluation understudy) was used to evaluate the quality of the SMT system. Higher is better.
- **Baseline Models:** The baseline model was built using Moses with default settings.

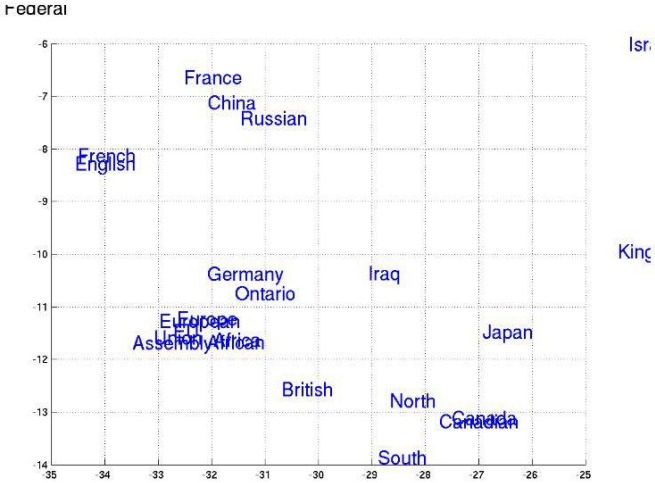
Experimental Results

- Additional features computed by the RNN Encoder-Decoder consistently improved over the baseline performance.
- Best performance achieved when both the phrase scores from the RNN Encoder-Decoder model were used in conjunction with CSML.

Models	BLEU	
	dev	test
Baseline	30.64	33.30
RNN	31.20	33.87
CSLM + RNN	31.48	34.64
CSLM + RNN + WP	31.50	34.54

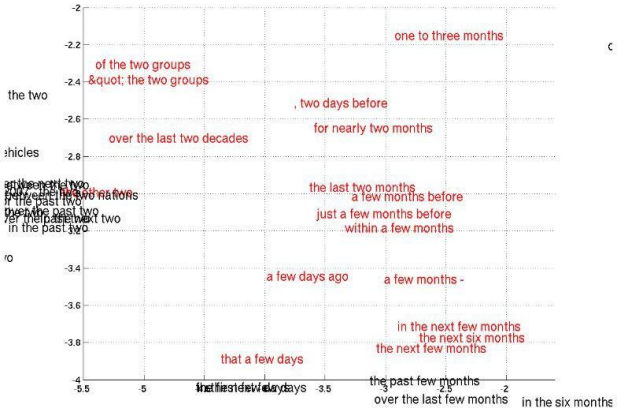
Qualitative Evaluation: Word Representations

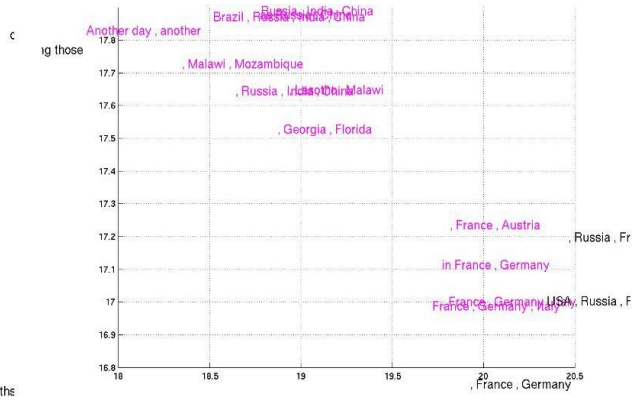


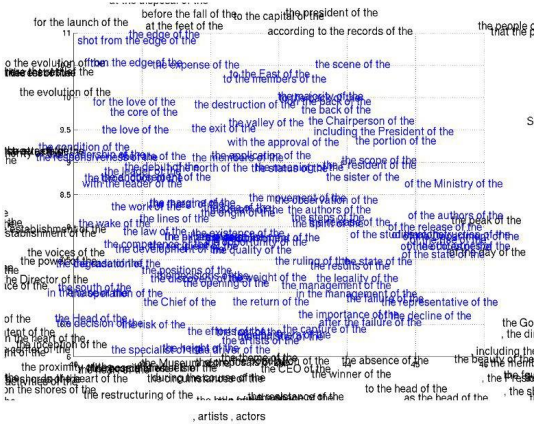


Semantically similar words get clustered together

Qualitative Evaluation: Phrase Representations







Visual inspection of the resulting projections revealed that the RNN Encoder–Decoder model captures both semantic and syntactic structures of the phrases.

Summary

- Proposed a new neural network architecture, called an RNN Encoder–Decoder that is able to learn the mapping from an arbitrary length sequence to another sequence of arbitrary length.
- A new hidden unit called Gated Recurrent Units (GRUs) to improve the memory capacity and the ease of training
- Empirically evaluated the performance of the model on task on phrase based statistical machine translation (SMT)
- Qualitative analysis of the RNN Encoder-Decoder model revealed that the phrase representation learnt by the model preserves both syntactic and semantic structure.

BRIDGENETS: STUDENT-TEACHER TRANSFER LEARNING BASED ON RECURSIVE NEURAL NETWORKS AND ITS APPLICATION TO DISTANT SPEECH RECOGNITION

Jaeyoung Kim, Mostafa El-Khamy, Jungwon Lee

https://arxiv.org/abs/1710.10224

Introduction

- Distance speech recognition (DSR) is to recognize human speeches in the presence of noise, reverberation and interference caused mainly by the large distance between speakers and microphones.

- Traditional front-end approaches interconnect multiple independent components such as speech enhancer, acoustic speech detector and many other blocks before a speech recognition module.

Introduction

- End-to-end methods are proposed by optimizing multiple components in the unified framework.
    1. Multi-task denoising jointly optimizes denoising and recognition sub-networks using synchronized clean data.
    2. Knowledge distillation (KD) transfers the generalization ability of a bigger teacher network to a typically much smaller student network. Generalized distillation (GD) extends distillation methods by training a teacher network with separate clean data.
- BridgeNet provides multiple hints from a teacher network.
- The proposed recursive architecture can interactively refine recognition and denoising performance.

Network Description

BridgeNet uses a collection of triplets as training data: (x_t^*, x_t, y_t). x_t^* is the enhanced or less noisy data. x_t and y_t are noisy data and their labels

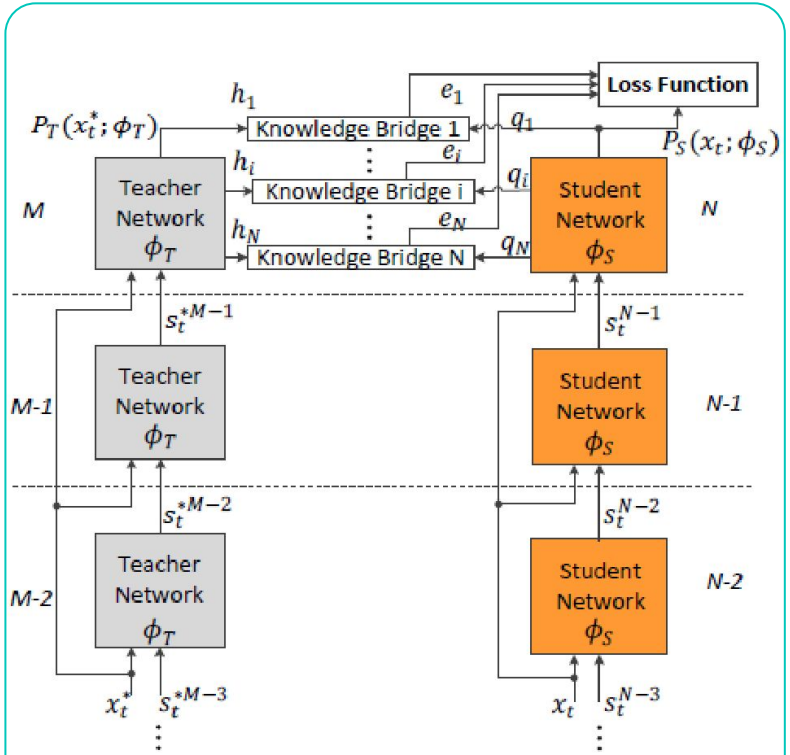


Fig. 1. Conceptual Diagram of BridgeNet

Network Description

An error measure e_i of how feature representation q_i from a student network agrees with the hint h_i is computed at the knowledge bridge as a MSE loss,

$$e_i(\phi_S) = \sum_{t=1}^{L} \|h_i(x_t^*) - q_i(x_t; \phi_S)\|^2$$

Where $\phi_S$is the learnable parameters of a student network.

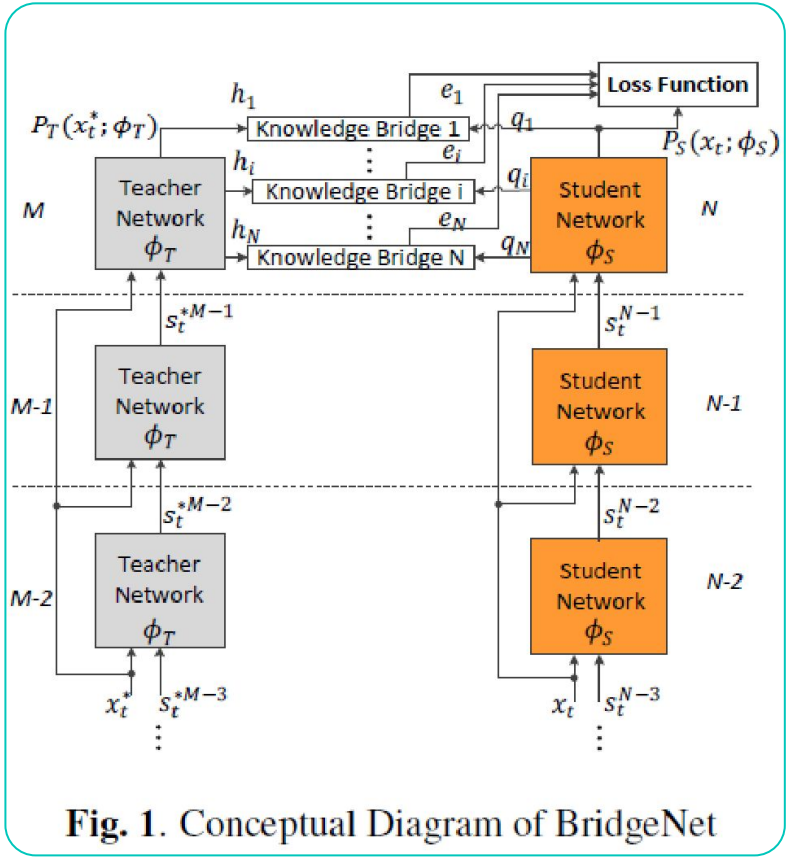


Fig. 1. Conceptual Diagram of BridgeNet

Network Description

$$L(\phi_S) = \sum_{i=1}^{N} \alpha_i e_i(\phi_S)$$

Where α_i is a predetermined weighting factor for e_i

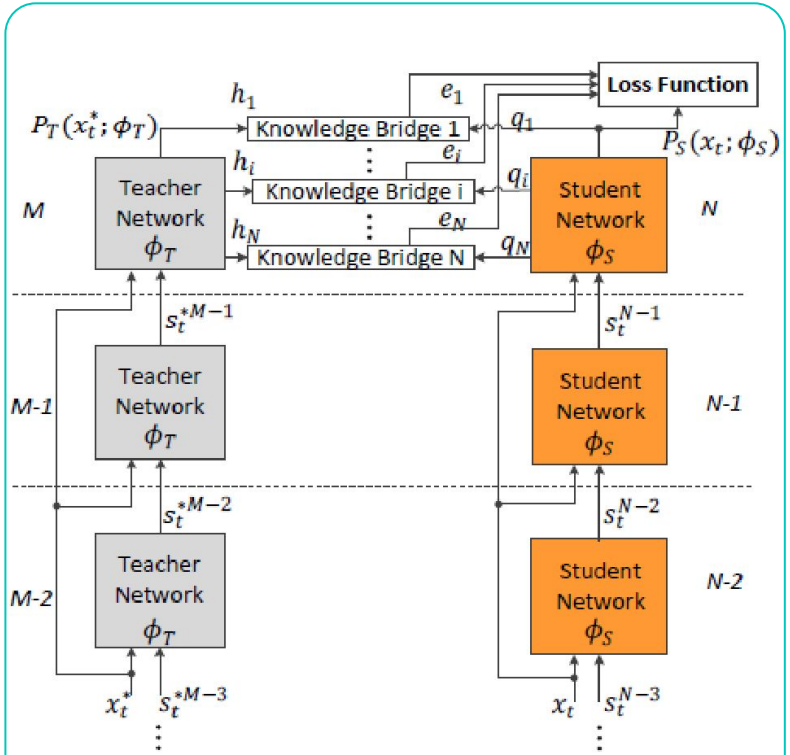


Fig. 1. Conceptual Diagram of BridgeNet

Recursive Architecture

It is composed of four sub-blocks: I and F take acoustic features and feedback states as their input, M merges I and F outputs and L produces recognized phone states.

Each block can be any type of network. i_t^n, f_t^n, m_t^n, and s_t^n represents output for the corresponding sub-blocks. n indicates the recursion number. l_{init} is a zero vector used as input for the zero recursion

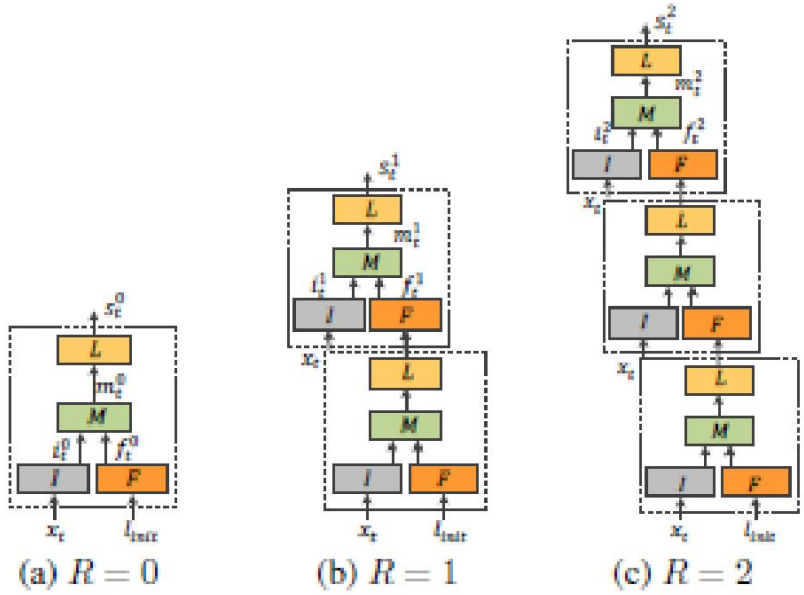


Fig. 2. Unrolling of a Recursive Network: R is the number of recursions. (a), (b) and (c) show how a recursive network is unrolled in the depth direction. The blocks with the same color share the same weights.

Recursive Architecture

The advantage of this sub-block division enables a network to recurse with heterogeneous input and output types.

The same input x_t is applied to the network for each recursion. This repeated input acts as a global shortcut path that is critical to train a deep architecture.

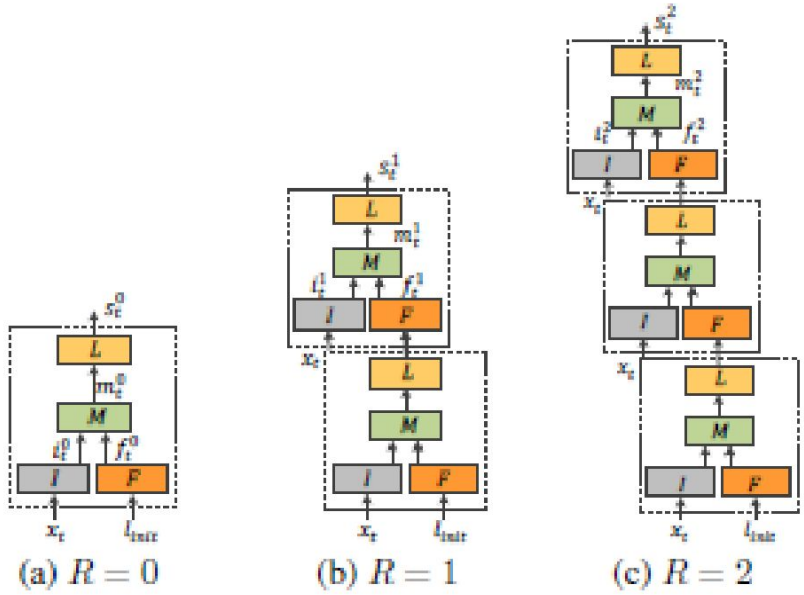


Fig. 2. Unrolling of a Recursive Network: R is the number of recursions. (a), (b) and (c) show how a recursive network is unrolled in the depth direction. The blocks with the same color share the same weights.

Recursive Architecture

The proposed recursive network can be formulated as follows:

$$m_t^n = g\left(W_1 \cdot i_t^n(x_t) + W_2 \cdot f_t^n\left(s_t^{n-1}\right) + b\right)$$

W_1, W_2, and b are the internal parameters of M.

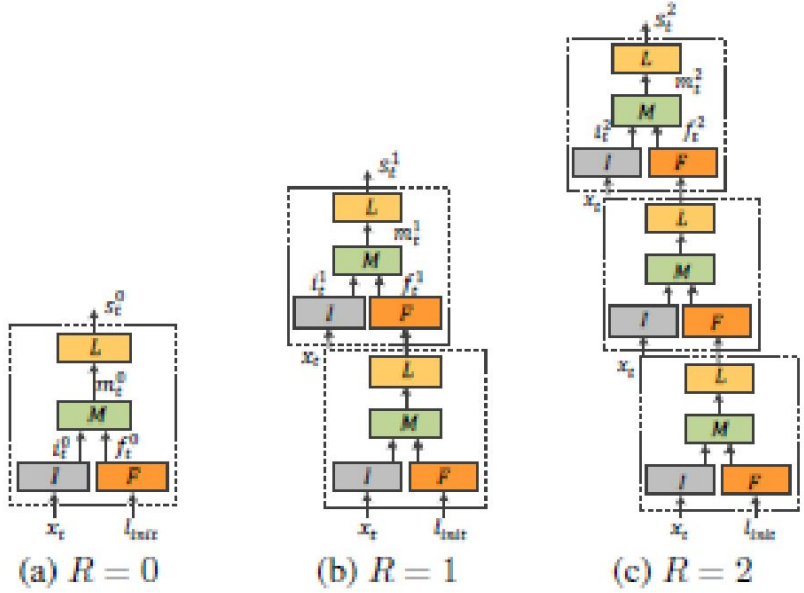


Fig. 2. Unrolling of a Recursive Network: R is the number of recursions. (a), (b) and (c) show how a recursive network is unrolled in the depth direction. The blocks with the same color share the same weights.

Recursive Architecture

CNN layer (I)
First LSTM layers (F)
Second LSTM layers (L)
Dimension reduction layers (M)

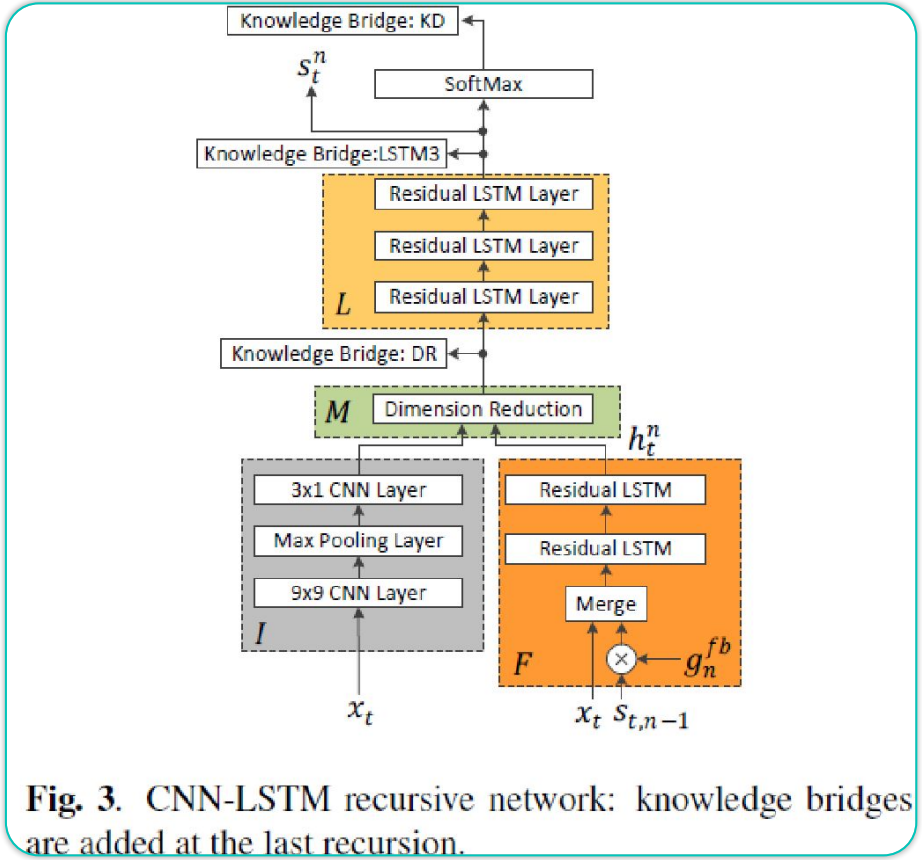


Fig. 3. CNN-LSTM recursive network: knowledge bridges are added at the last recursion.

Recursive Architecture

Feedback phones are processed in F controlled by a gate network, g_n^{fb}.

$$g_n^{fb} = \sigma(\omega_x x_t + \omega_s s_t^{n-1} + \omega_h h_{t-1}^n)$$

where s_t^{n-1} is a feedback state from the $(n-1)^{th}$ recursion, h_{t-1}^n is the output of F at the n^{th} recursion and ω_x, ω_s, and ω_h are weights to be learned.

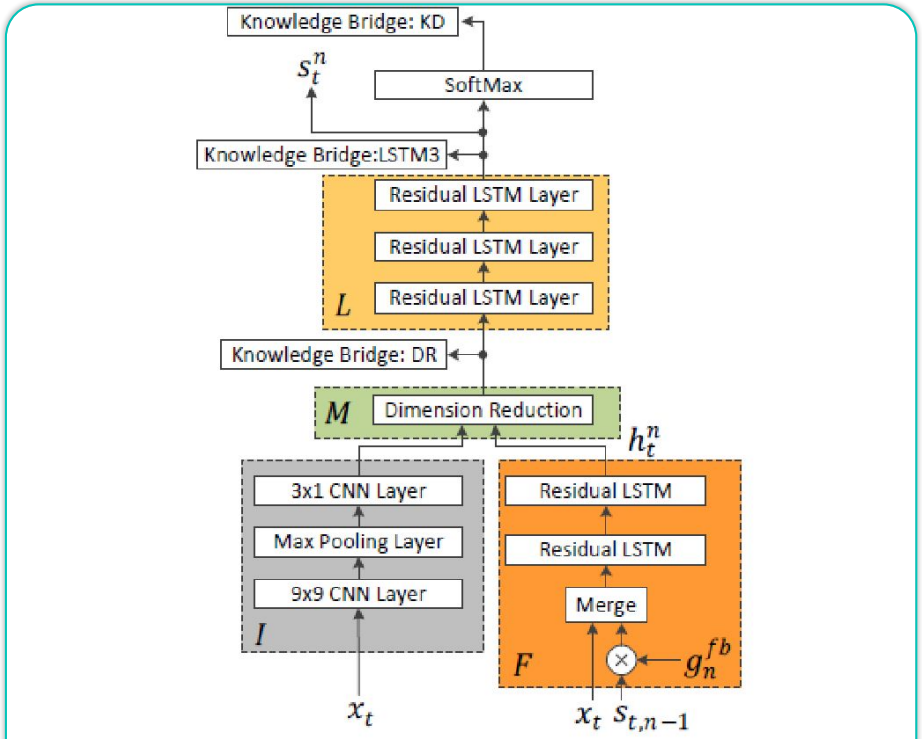


Fig. 3. CNN-LSTM recursive network: knowledge bridges are added at the last recursion.

Recursive Architecture

A residual LSTM is used for F and L sub-blocks. It has a shortcut path between layers to avoid vanishing or exploding gradients commonly happening to deep networks.

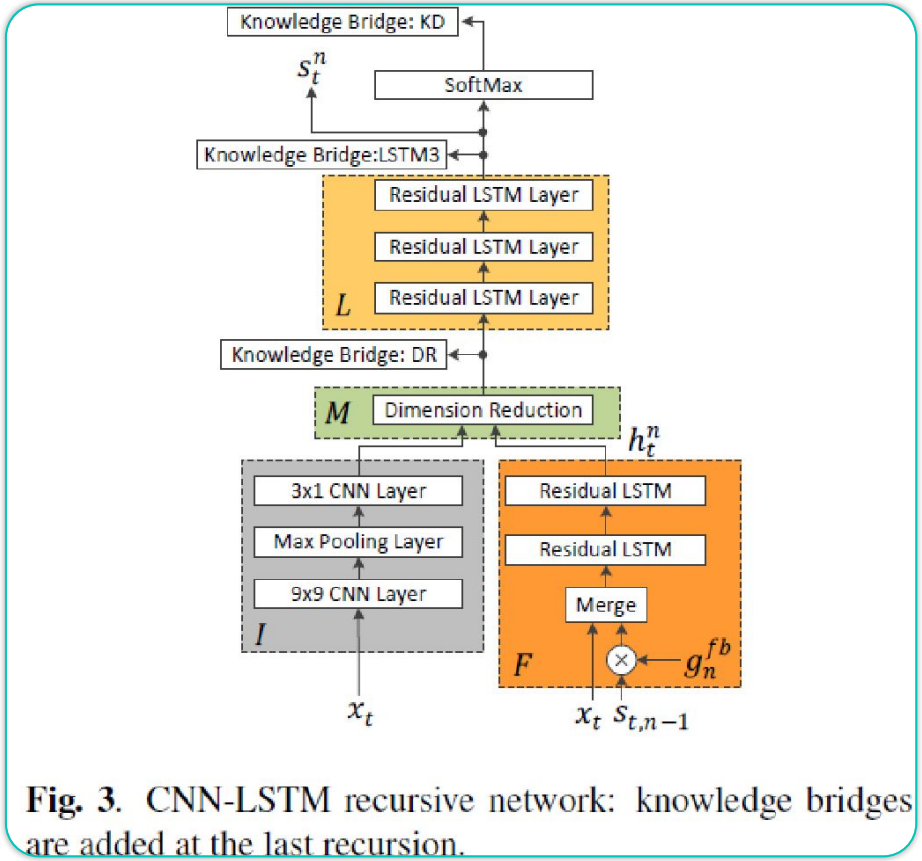


Fig. 3. CNN-LSTM recursive network: knowledge bridges are added at the last recursion.

Experimental Setup

- AMI corpus provides 100 hours meeting conversions recorded both by individual headset microphones (IHM) and single distant microphones (SDM). SDM can be improved as multiple distant microphone (MDM) by beamforming multiple SDM channels.

- Two types of word error rates (WER):
  - The all-speakers WER is to decode up to 4 concurrent speeches.
  - The main-speaker WER is to decode single main speaker at each time frame.

BridgeNet and Multi-Task Denosing on AMI

Table 1. Multi-Task Denoising on SDM: CNN-LSTM* was trained with a clean alignment from IHM. Other models used a noisy alignment from SDM

Acoustic Model	WER (all)	WER (main)
DNN	59.1%	50.5%
DNN, denoised	58.7%	50.2%
CNN-LSTM	50.4%	41.6%
CNN-LSTM, denoised	50.1%	41.4%
CNN-LSTM*	46.5%	37.7%
CNN-LSTM*, denoised	46.9%	38.2%

BridgeNet and Multi-Task Denosing on AMI

Table 2. BridgeNet: A teacher network is trained with IHM data and a student network is trained with SDM data. Rn means the network has n recursions. (e.g. baseline CNN-LSTM with R2 has two recursions)

Acoustic Model	WER (all)	WER (main)
CNN-LSTM (baseline), R0	46.5%	37.7%
KD, R0	44.8%	35.7%
KD+DR, R0	44.1%	35.3%
KD+DR+LSTM3, R0	44.0%	35.1%
CNN-LSTM, R2	45.8%	36.9%
KD, R1	43.7%	34.7%
KD+DR, R1	43.4%	34.7%
KD+DR+LSTM3, R1	42.6%	33.8%

Table 3. BridgeNet: A teacher network is trained with clean IHM data and a student network is trained with MDM data.

Acoustic Model	WER (all)	WER (main)
CNN-LSTM (Baseline), R0	43.4%	34.0%
KD, R0	42.8%	33.1%
KD+DR, R0	42.3%	32.5%
KD+DR+LSTM3, R0	41.8%	32.2%
CNN-LSTM, R2	43.0%	33.3%
KD, R1	40.4%	30.8%
KD+DR, R1	39.5%	29.9%
KD+DR+LSTM3, R1	39.3%	29.5%

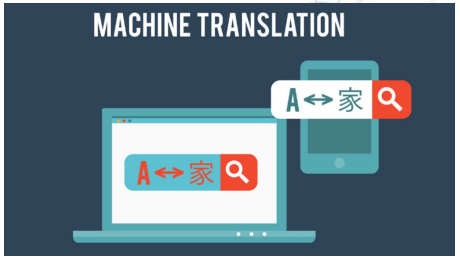
Recurrent Neural Networks (RNNs)

Neural Networks

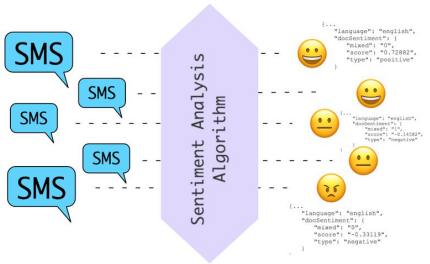
- So far we have seen:
  - Deep feedforward networks (MLPs)
    - Map a fixed length **vector** to a fixed length **scalar/vector**
    - Use case: classical machine learning
  - CNNS
    - Map a fixed length **matrix/tensor** to a fixed length **scalar/vector**
    - Use case: image recognition

- RNNs
  - Map a **sequence** of **matrices/tensors** to a **scalar/vector**
  - Map a **sequence** to a **sequence**
  - Use case: natural language processing (NLP)

NLP



- The challenge of language for computers:
  - Computers are built to process numbers
  - Language isn't easily represented by numbers
  - How can we represent human language in a computable fashion?
  - Applications: machine translation, text classification, information retrieval, sentiment analysis and many more
    - You already saw one example: classifying IMDb movie reviews as either positive or negative



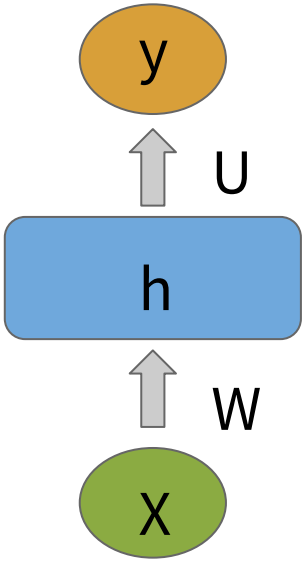
MLPs ⟶ RNNs

- RNNs are a natural extension of MLPs
- MLPs are “memoryless”, but often we need knowledge of the past sequence of events to predict the future

	Inputs	**Output**	**Probability**
MLP	X	y	$P(y\|X)$
RNN	$[x_1, x_2, x_3, \ldots, x_t]$	y	$P(y\|x_1, x_2, x_3, \ldots, x_t)$

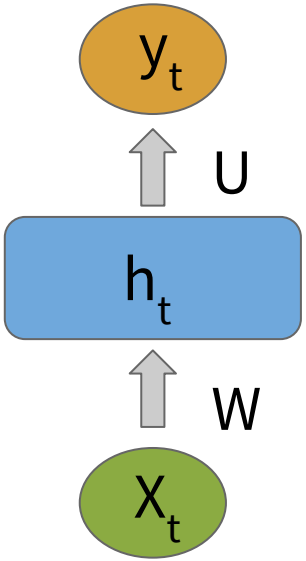
MLPs ⟶ RNNs

- Recall that the first hidden layer for an MLP is given by h = f(XW + b) where f() is the activation function and W is the weight matrix in the hidden layer, b is the bias term, and U is the weight matrix in the output layer



MLPs ⟶ RNNs

- RNNs add the concept of “state” to traditional neural networks
- To incorporate the notion of time we will index the hidden layer with t and feed it X_t:
  $h_t = f(X_t W + b)$



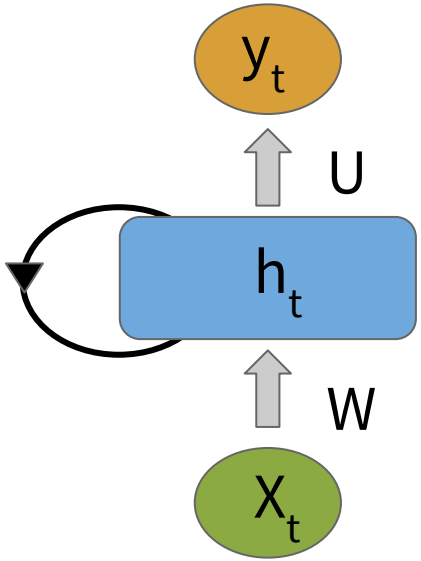
MLPs ⟶ RNNs

- To incorporate information from the previous state we will make the following modification:

$h_t = f(X_t W + b) \longrightarrow h_t = f(X_t W + h_{t-1} U + b)$

Input at time t

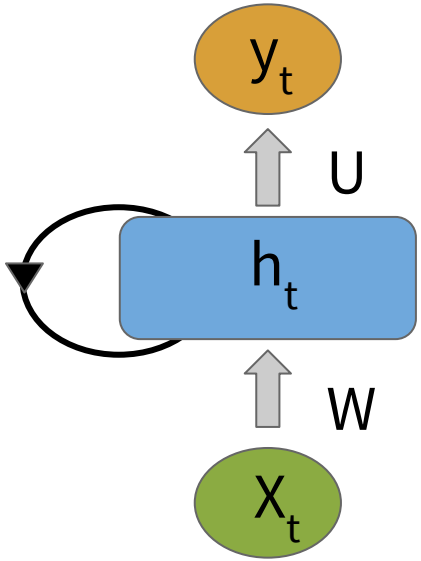
Hidden state from previous time point



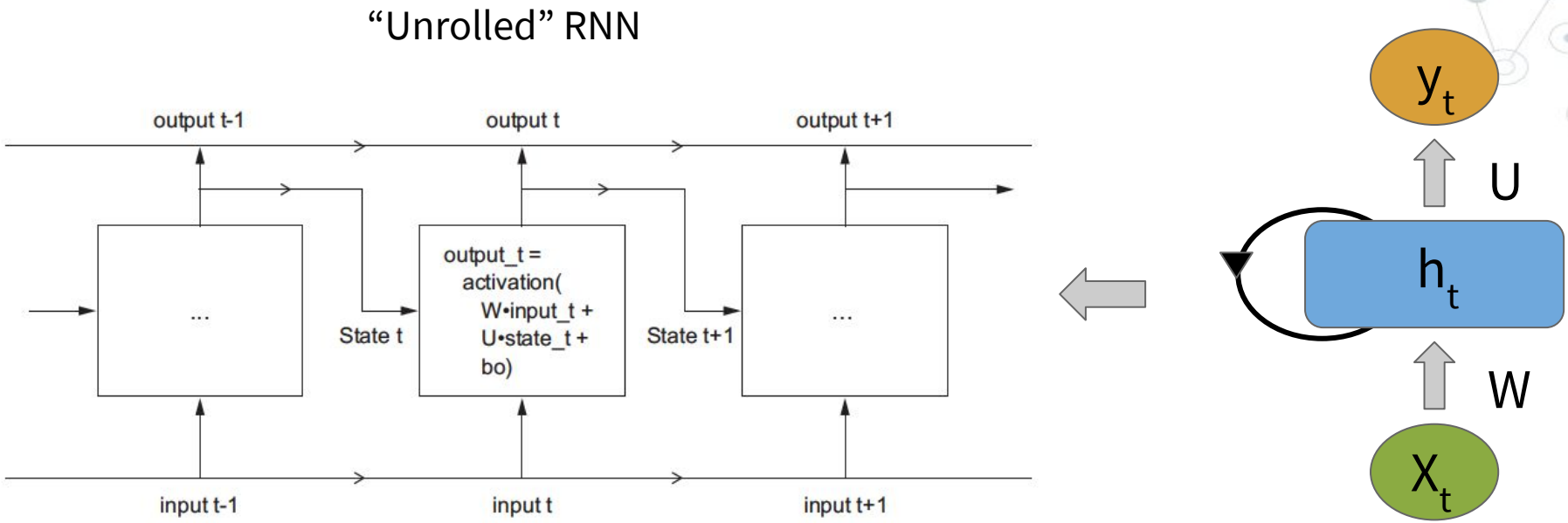
- This is equivalent to connecting the hidden state to itself

RNN Backprop

- How do we backprop through something with a loop?
- Have to backprop through depth and time
- This is similar to what we saw with MLPs, but we aren't going to go through it here

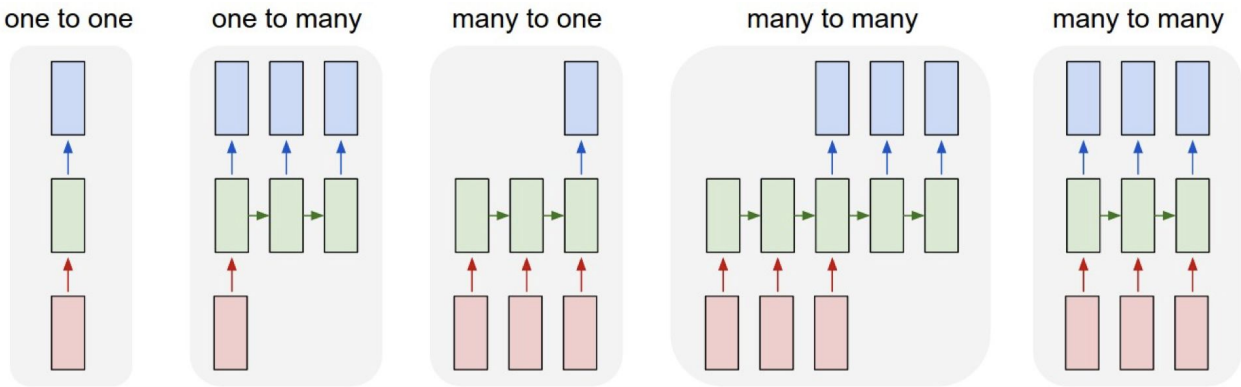


RNNs


“Unrolled” RNN
output t-1
output t
output t+1
...
State t
output_t =
activation(
W•input_t +
U•state_t +
bo)
State t+1
...
input t-1
input t
input t+1
y_t
U
h_t
W
X_t

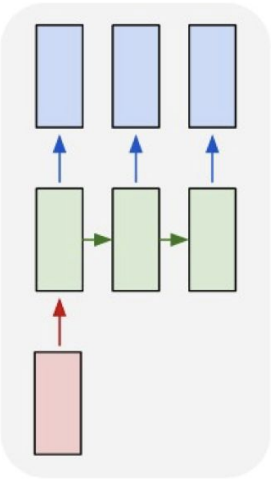
RNNs

- There are many ways to configure the input ⇨ output mapping

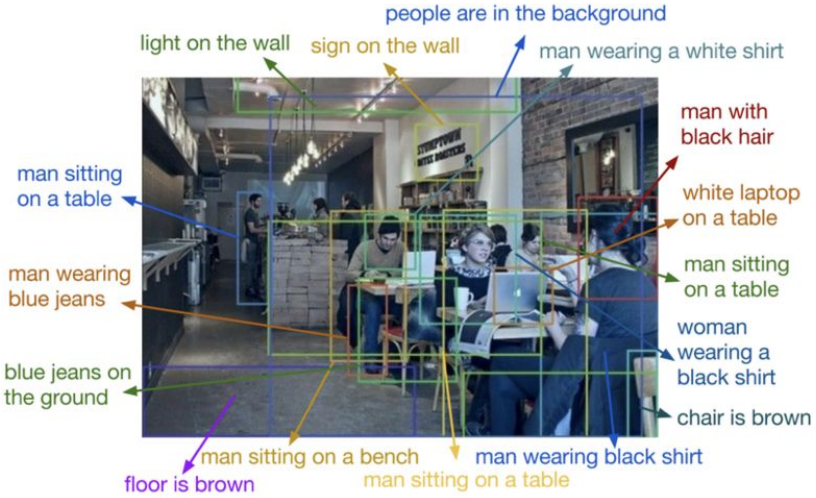


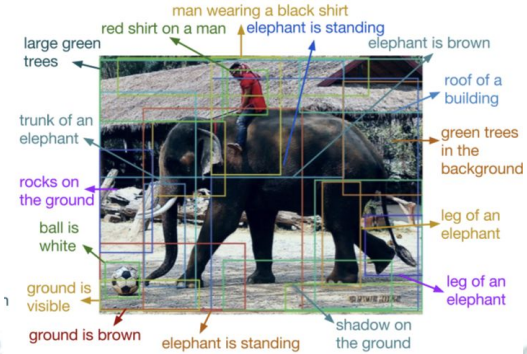
RNNs

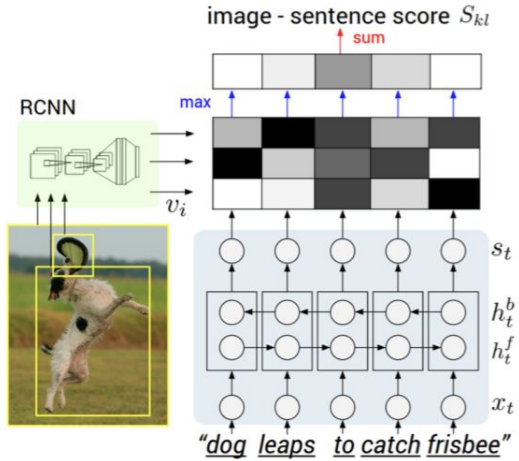
one to many



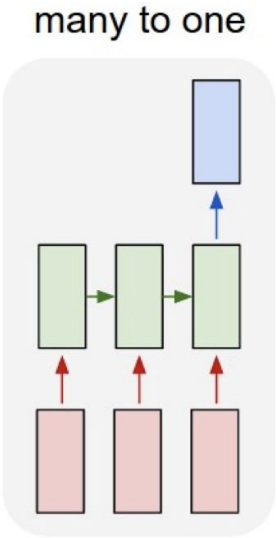
Ex: image captioning



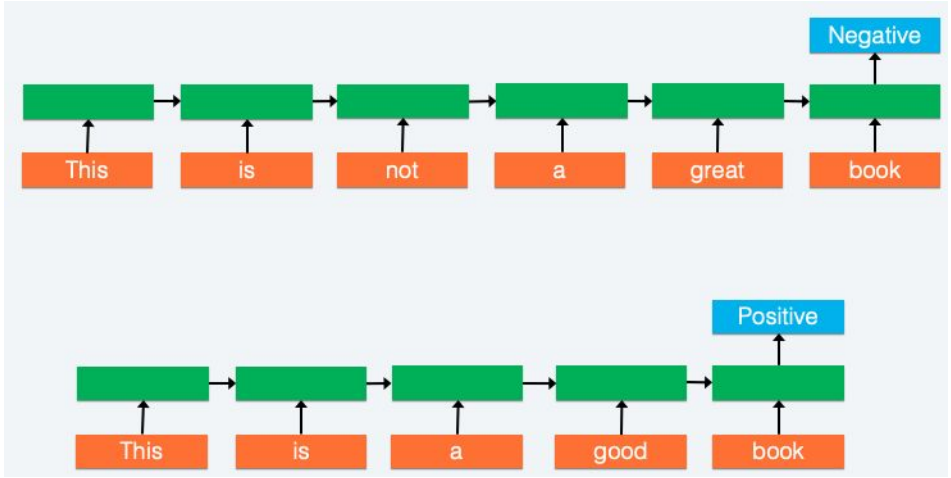




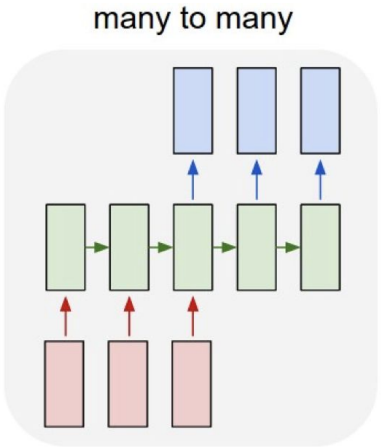
RNNs



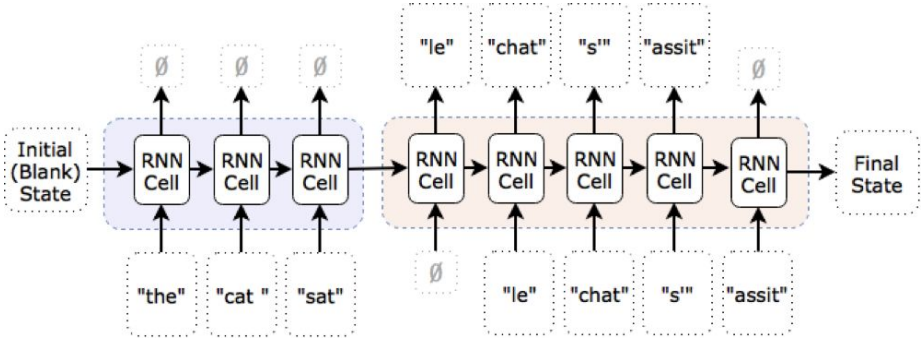
Ex:
Sentiment
Analysis

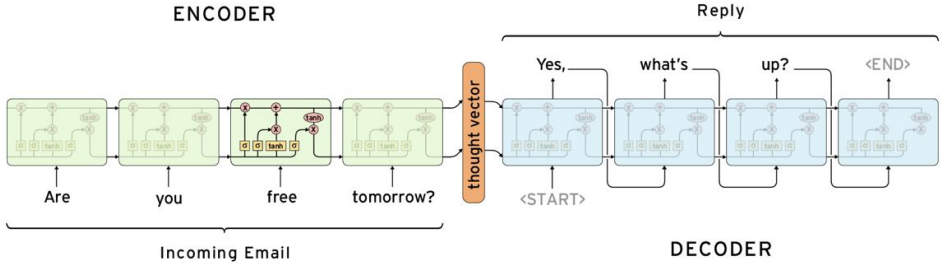


RNNs

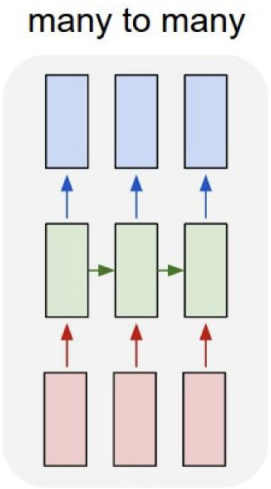


Ex:
Translation,
automated
response

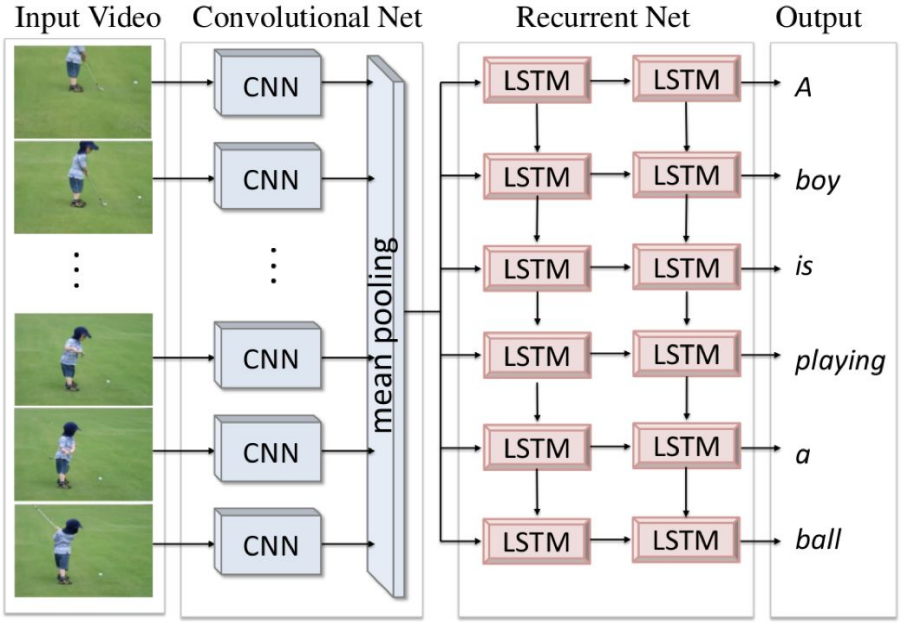




RNNs



Ex: frame by frame image captioning



RNNs

- High-level takeaways:
  - RNNs provide a way to handle **sequence** data where the order of events is important
  - Simple modification to MLP model
  - RNNs maintain a "**state**" that reflects current configuration of the "world"

RNNs

- High-level takeaways:
  - RNNs provide a natural way to “update” your beliefs about the world as new information arrives
  - Really **flexible** and can model many different scenarios that get weird/complicated quickly
  - CNNs = hard to understand but easy to implement; RNNs = easy to understand but hard to implement

Applications

- Document and time series classification e.g. identifying the topic of an article or the author of a book
- Time series comparisons e.g. estimating how closely related two documents are
- Sentiment analysis
- Time series forecasting e.g. predicting weather (something that needs major improvement for Boston...)
- Sequence-to-sequence learning e.g. decoding an English sentence into Turkish

Working with text data

Text Data

- Text data can be understood as either a sequence of characters or a sequence of words
  - Most common to work at the level of words
- Like all other neural networks, we can't simply input raw text - we must **vectorize the text**: transform it into numeric tensors
- We can do this in multiple ways:
  - Segment text into words, and transform each word into a vector
  - Segment text into characters and transform each character into a vector
  - Extract n-grams (overlapping groups of multiple consecutive words or characters) of words or characters, and transform each n-gram into a vector

Text Data

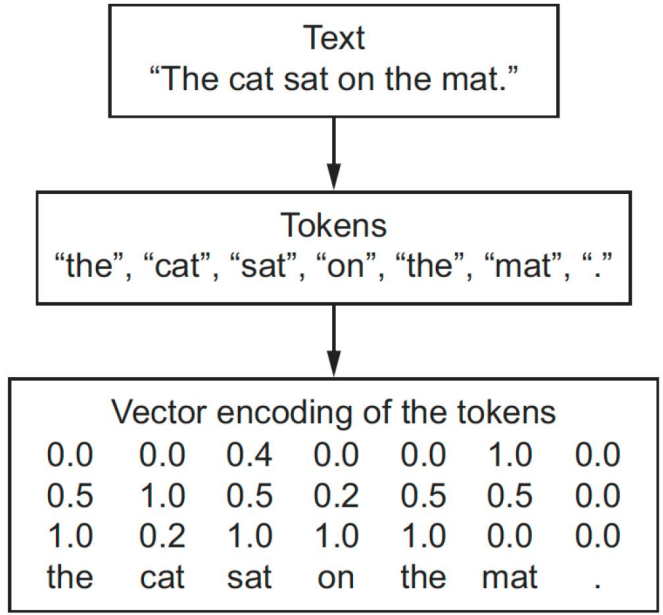
- The different units into which you break down text (words, characters, n-grams) are called **tokens**, and the action of breaking text into tokens is **tokenization**

- There are multiple ways to associate a vector with a token
  - One-hot encoding
  - Token embedding (or word embedding)

Tokenization

Tokens

Tokenization



N-grams

- Word **n-grams** are groups of N (or fewer) consecutive words that you can extract from a sentence. The same concept may also be applied to characters instead of words.

- For example, the sentence "**Data science rocks my socks off!**" can be decomposed into a set of 3-grams:
  - {**"Data", "Data science", "science", "science rocks", "Data science rocks", "rocks", "rocks my", "science rocks my", "my", "my socks", "socks", "rocks my socks", "off", "socks off", "my socks off"**}

N-grams

- This set is called a **bag of 3-grams**, which refers to the fact that it is a set of tokens, rather than a list or sequence: the tokens have no specific order
- This family of tokenization methods is called **bag-of-words**
- Order is not preserved, so the general structure of the sentence is lost
- Typically only used in shallow language-processing models
- Extracting n-grams is a form of feature engineering that deep learning models do automatically in another way

One-hot Encoding

- Most common and most basic way to turn a token into a vector
- We used this with the IMDB data set

1. Associate a unique integer index with every word

2. Then, turn the integer index i into a binary vector of size N (the size of the vocabulary, or number of words in the set)

- The vector is all 0s except for the ith entry, which is 1