

Deep Learning

Episode 5

Beating around image captioning

Maxim Borisyak, Mikhail Usvyatsov, Andrey Ustyuzhanin



Yandex
Data Factory

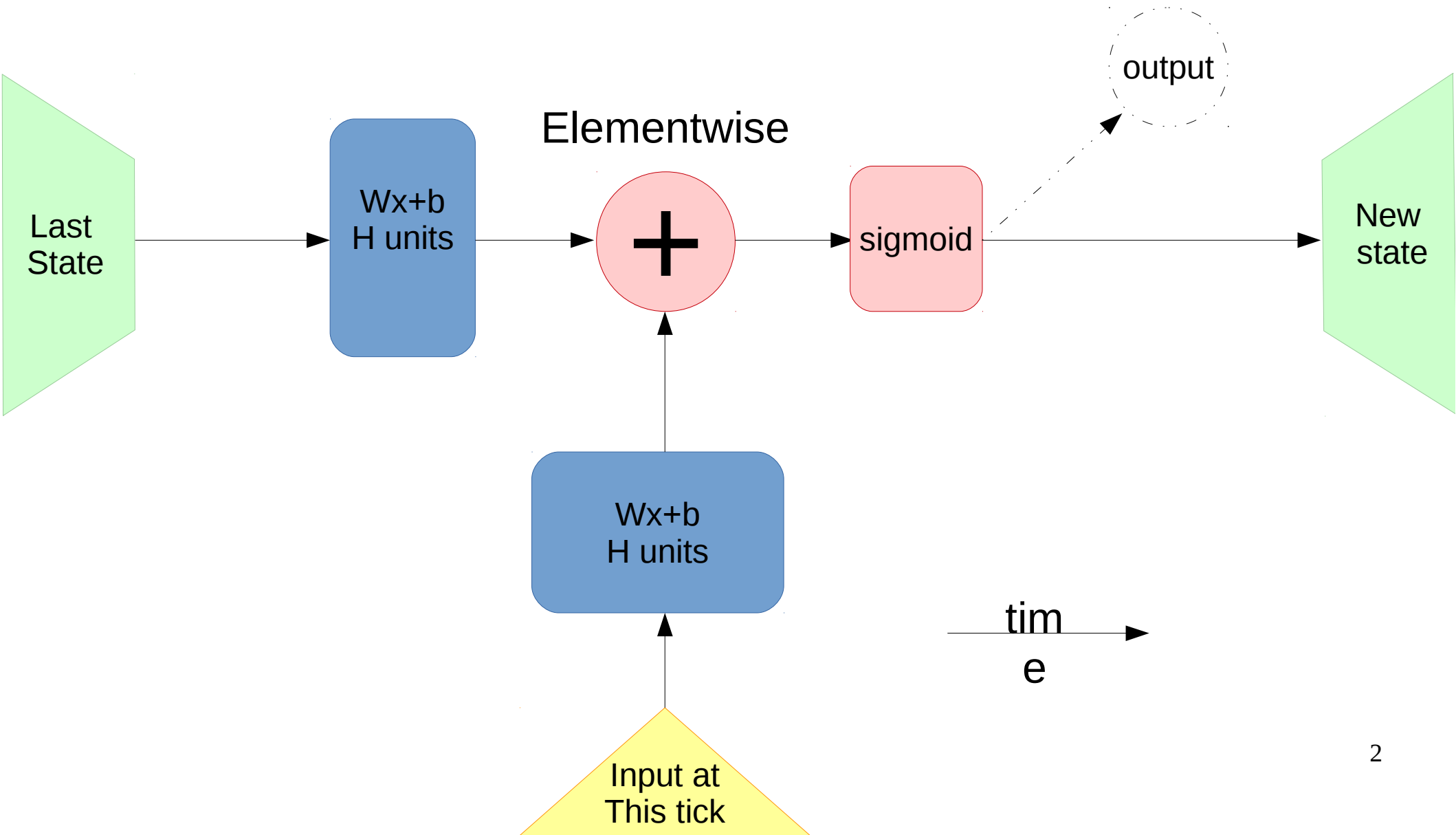
LAMBDA



British Hedgehog
Preservation Society

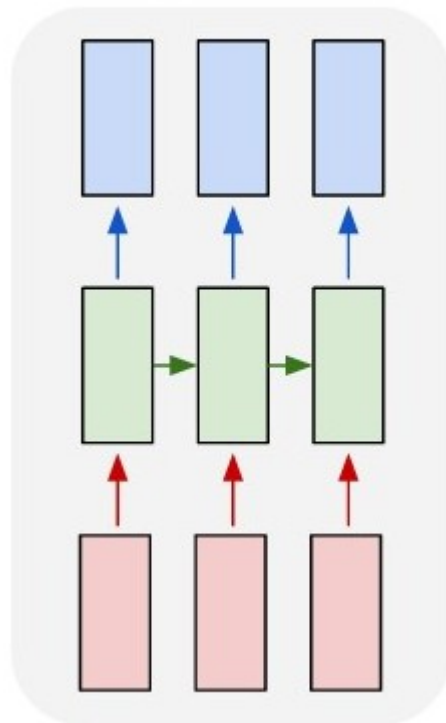


Recap: rnn step



Recurrent Architectures: regular

many to many



- Read sequence
- Predict sequence of answers at each tick

Tasks:

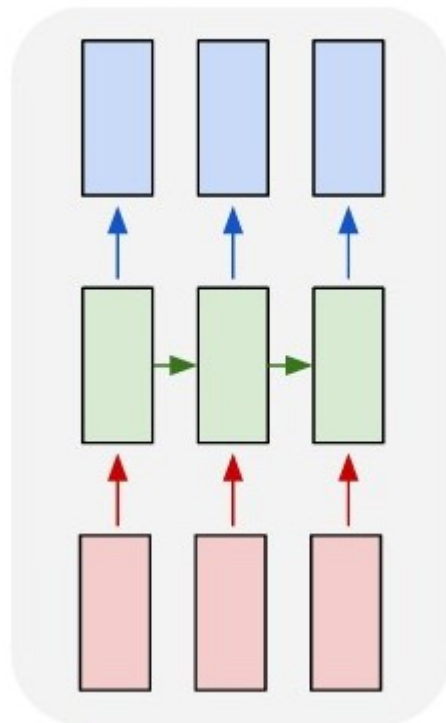
- Language model
- POS Tagging

How to implement?

- See last lecture

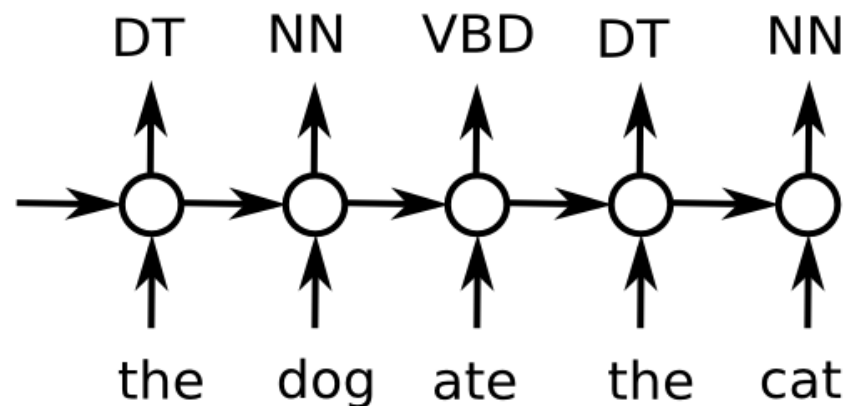
Recurrent Architectures: regular

many to many



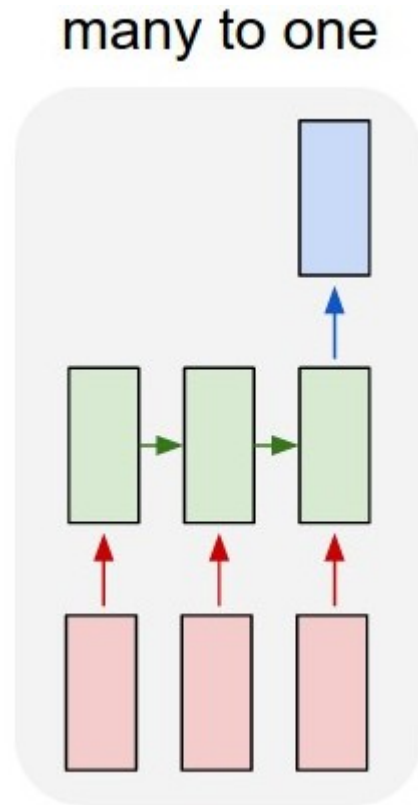
- Read sequence
- Predict sequence of answers at each tick

POS tagging



Why RNN?

Recurrent Architectures: Encoder



Encoder

- Read sequence
- Predict once

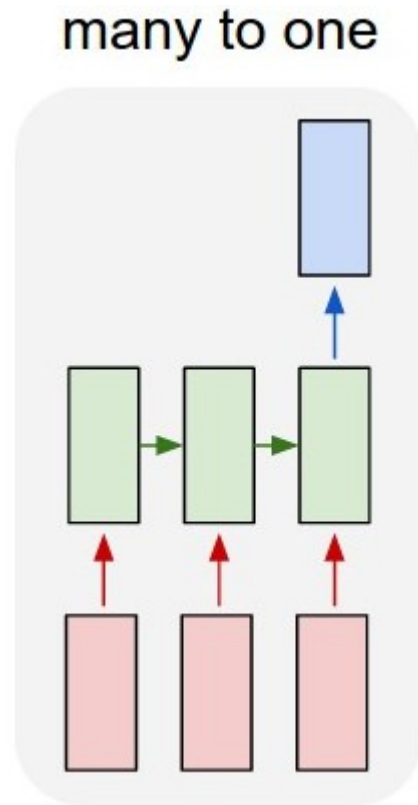
Tasks:

- ?!

How to implement?

- ?!

Recurrent Architectures: Encoder



Encoder

- Read sequence
- Predict once

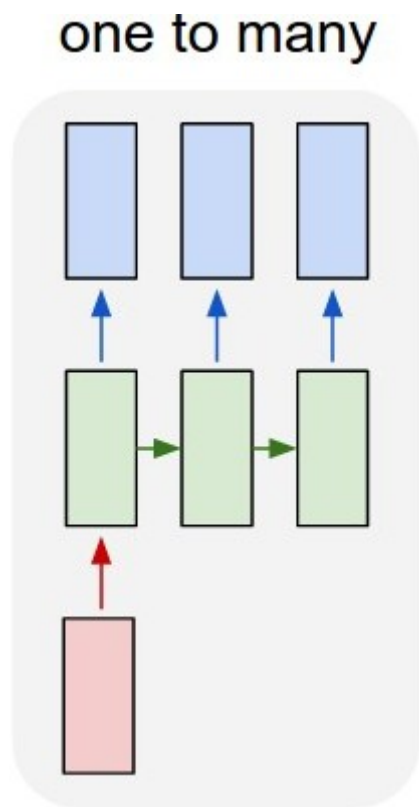
Tasks:

- Sentiment analysis
- Detect age by status
- Week3 homework
- Any text analysis

How to implement?

- Take last/max/mean over time

Recurrent Architectures: Decoder



Decoder

- Take one state
- Generate sequence

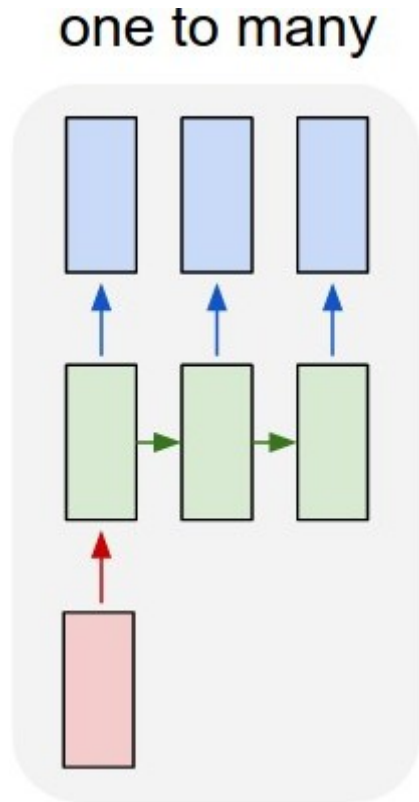
Tasks:

- Generate tweet given age & interests
- Generate molecule given desired properties

How to implement?

- ?!

Recurrent Architectures: Decoder



Decoder

- Take one state
- Generate sequence

How to implement?

- First hidden state (instead of zeros)
- Input at first tick
- Additional input at each tick

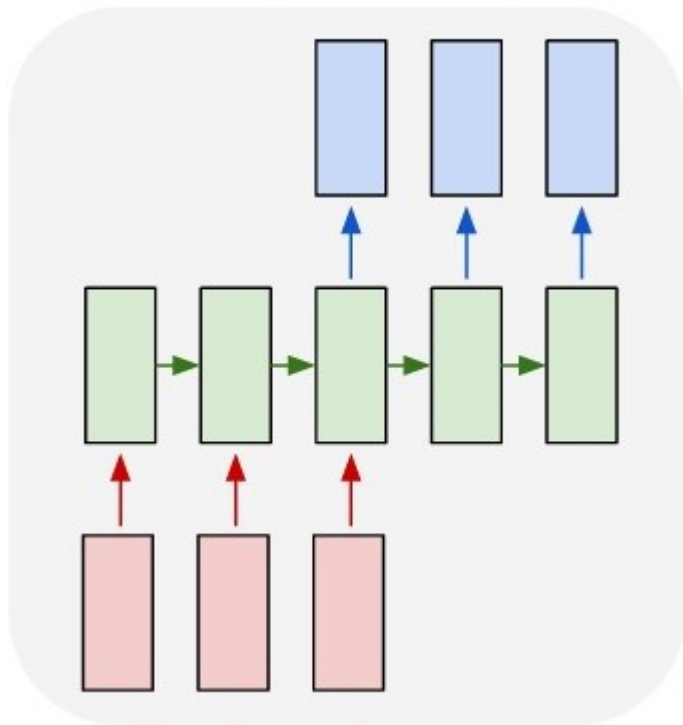
Seq2seq

How do we convert sequence to sequence of different kind/without time synchronization?

Example: Machine translation

Seq2seq

many to many

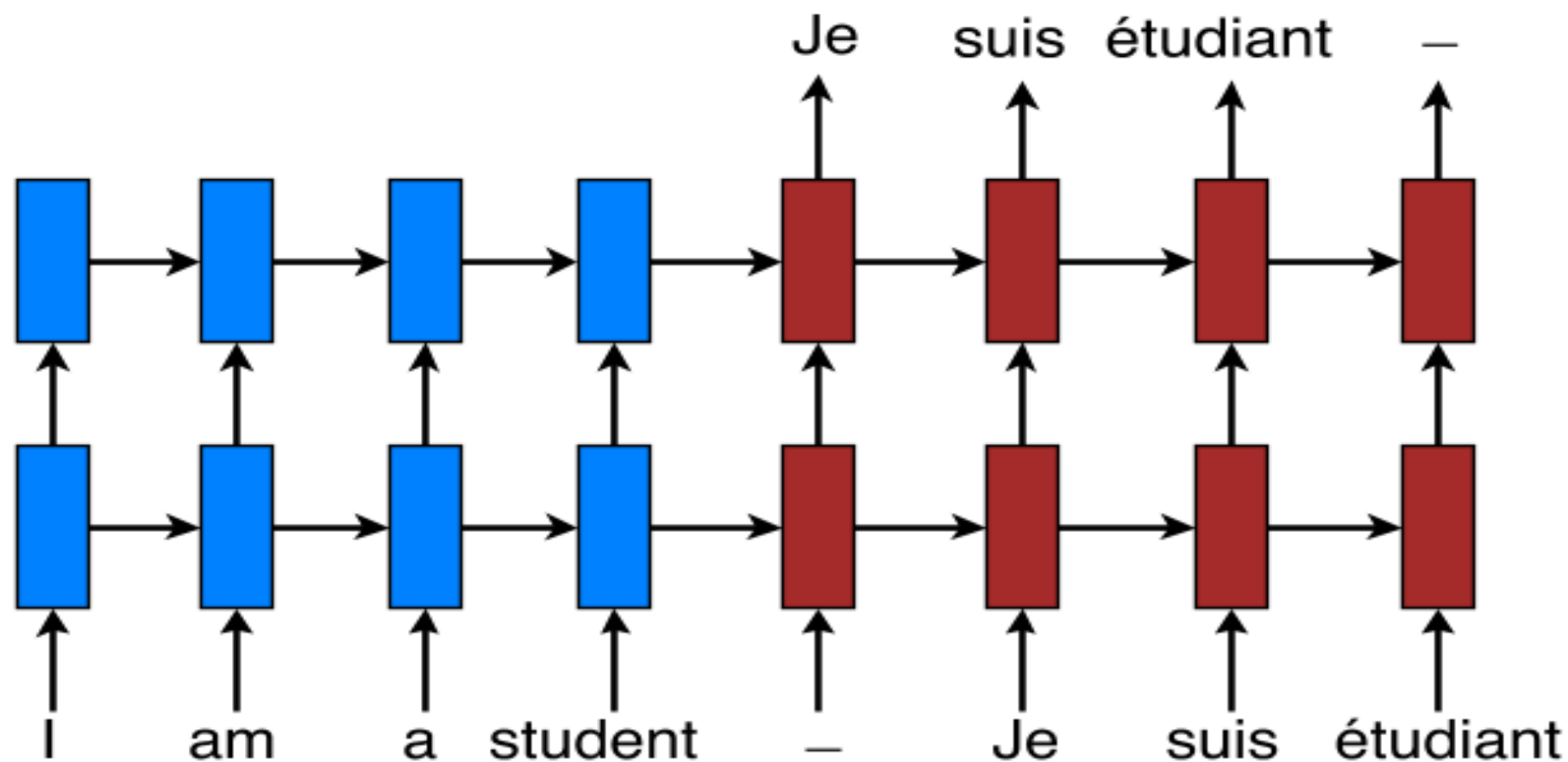


Idea:

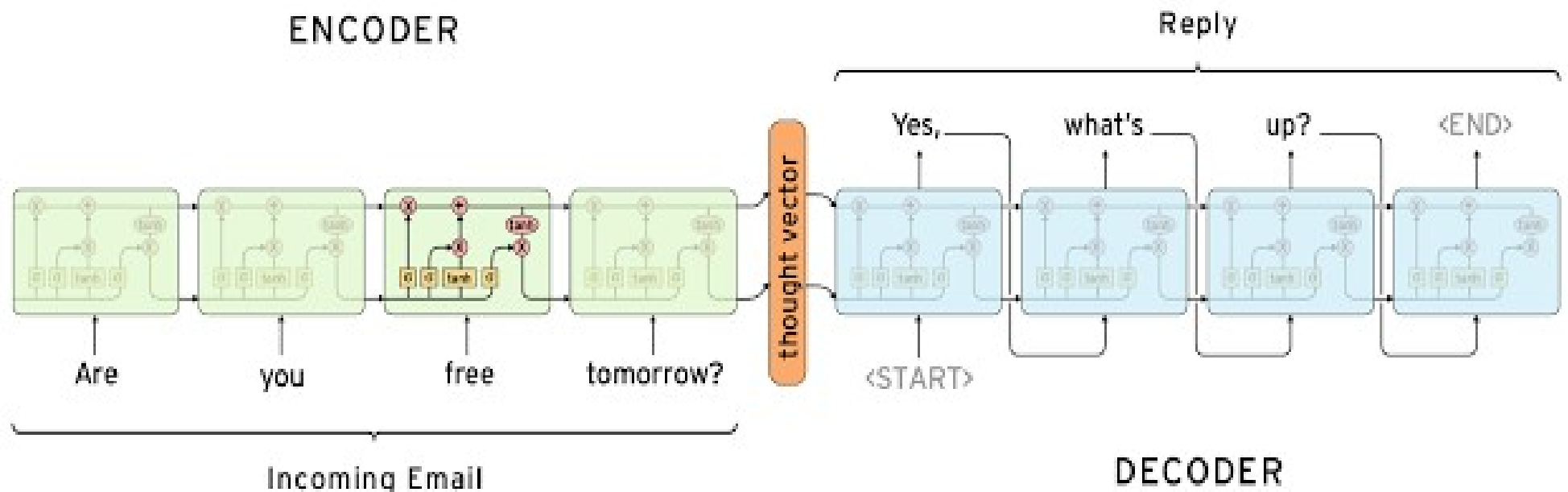
- first read (encode) the sequence
- then generate new one out of the encoded vector

How to implement that?

Seq2seq: Machine translation



Seq2seq: Conversation model



Exactly the same

MEMBER?

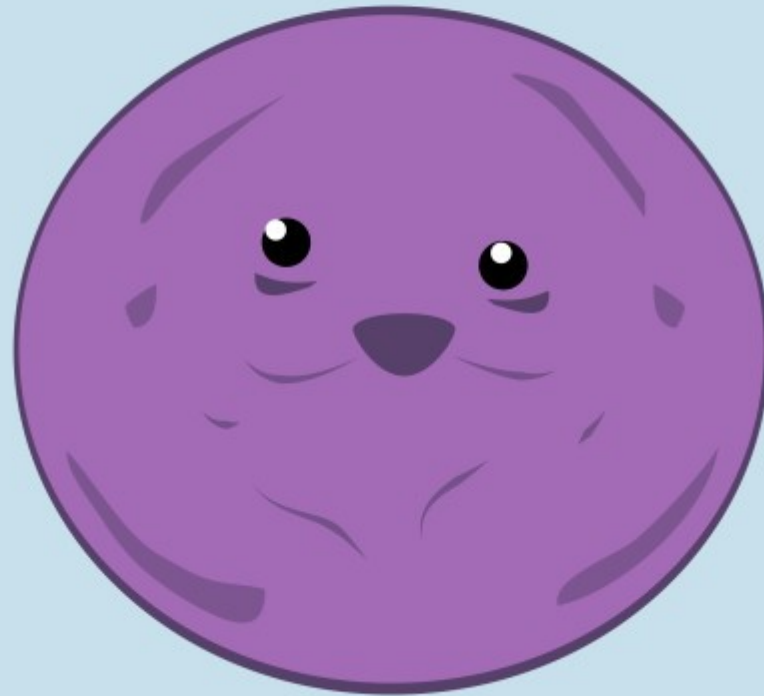


Image recognition



“Dog”

Image recognition



“Dog”

How do we solve that?

Image recognition

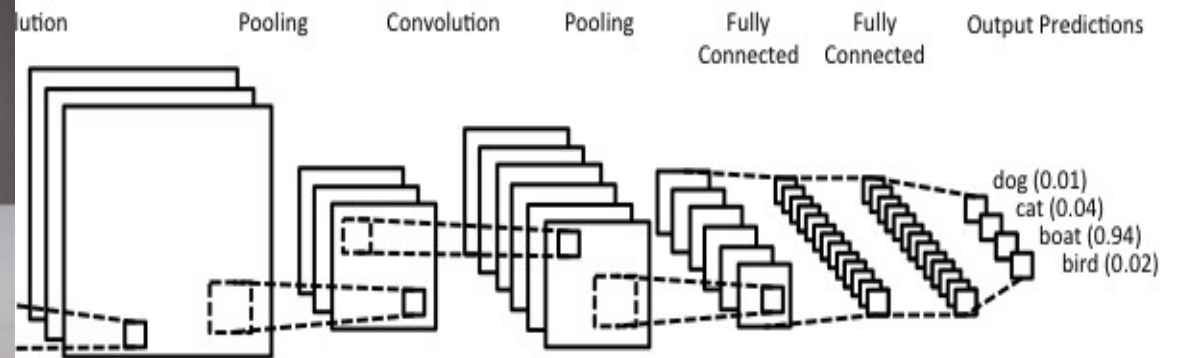
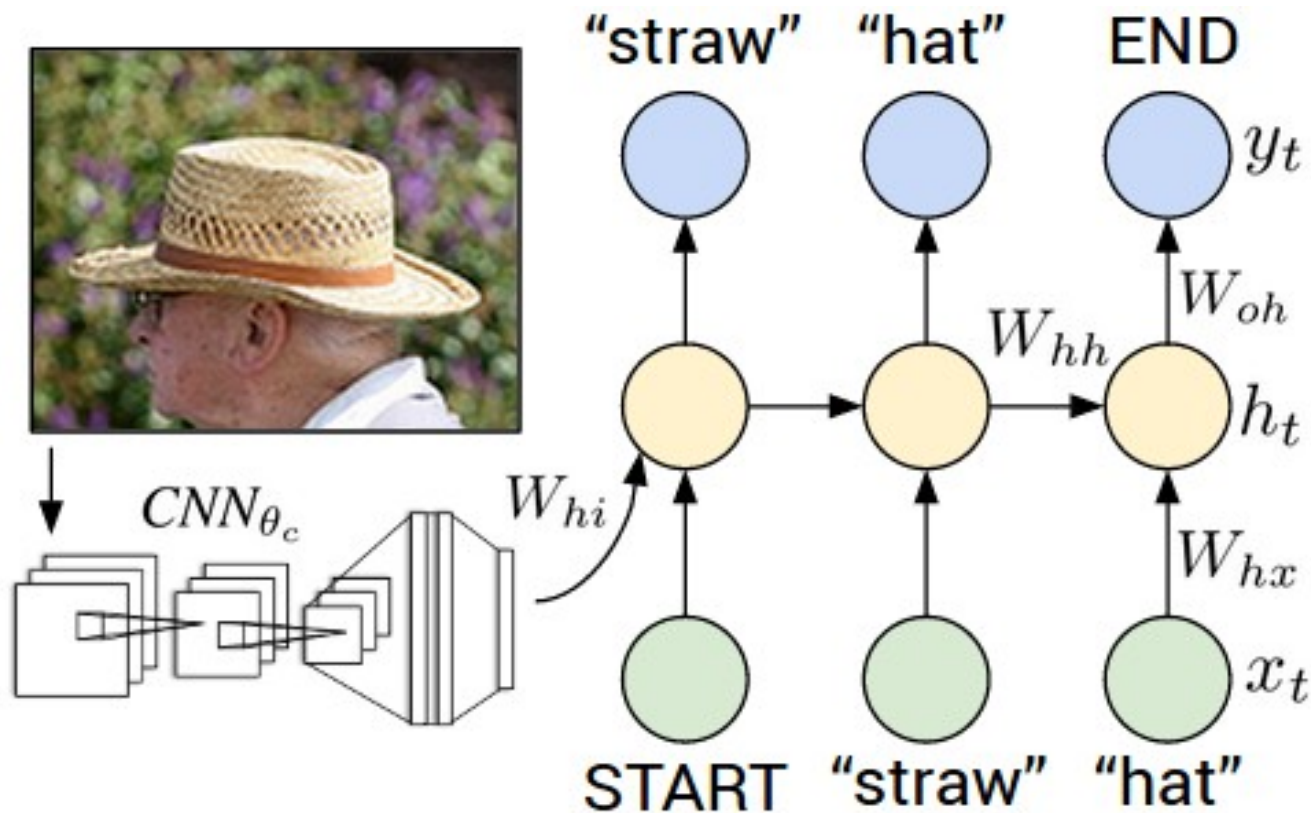


Image captioning



“A weird dog with
a blue collar is
standing on a
gray floor”

Task decomposition



Convolutional "encoder"

Recurrent "decoder"

Problem: we need a large CNN

Pre-training

Original network

- Requires 10^6 samples
- Trains for 1-2 weeks (GPU)

Typical problem

- 10^4 samples
- It must work in 3 days
or your boss kills you

How do you survive?

Pre-training

- 1. Train a network on large dataset

cifar X

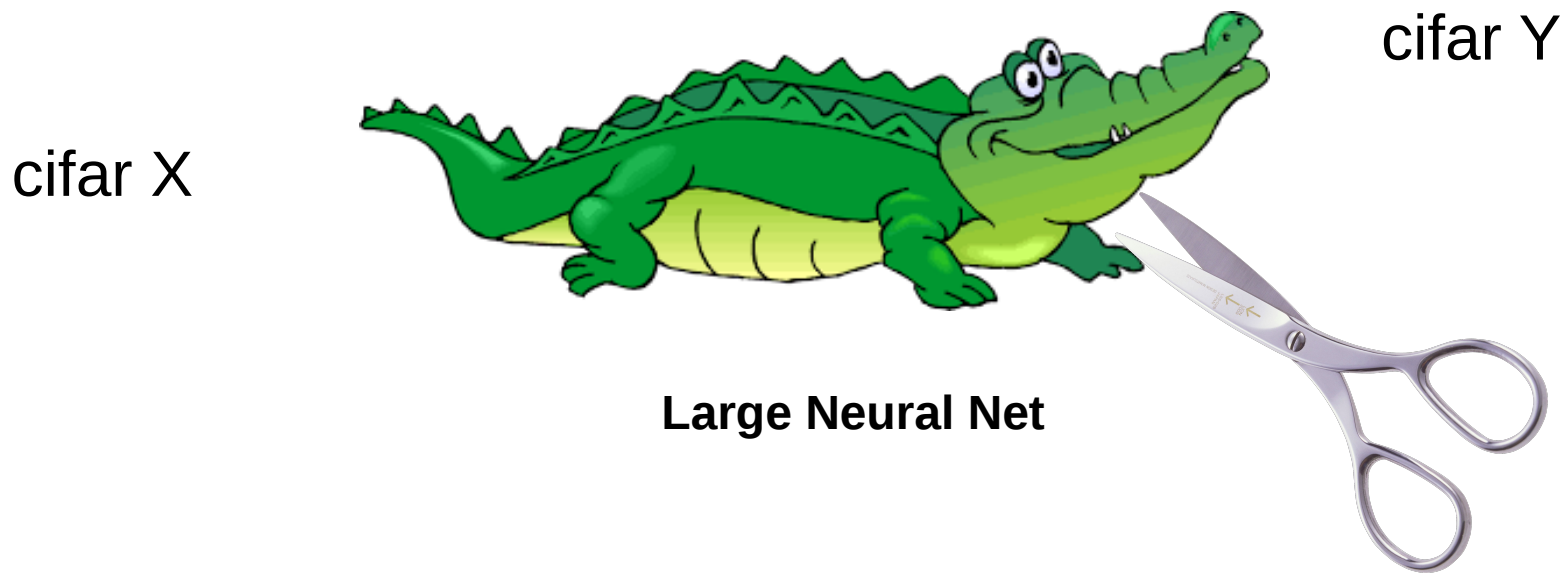


cifar Y

Large Neural Net

Pre-training

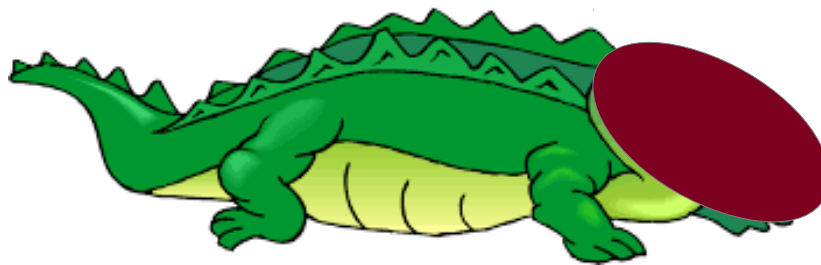
- 1. Train a network on large dataset
- 2. Take some intermediate layer



Pre-training

- 1. Train a network on large dataset
- 2. Take some intermediate layer

cifar X

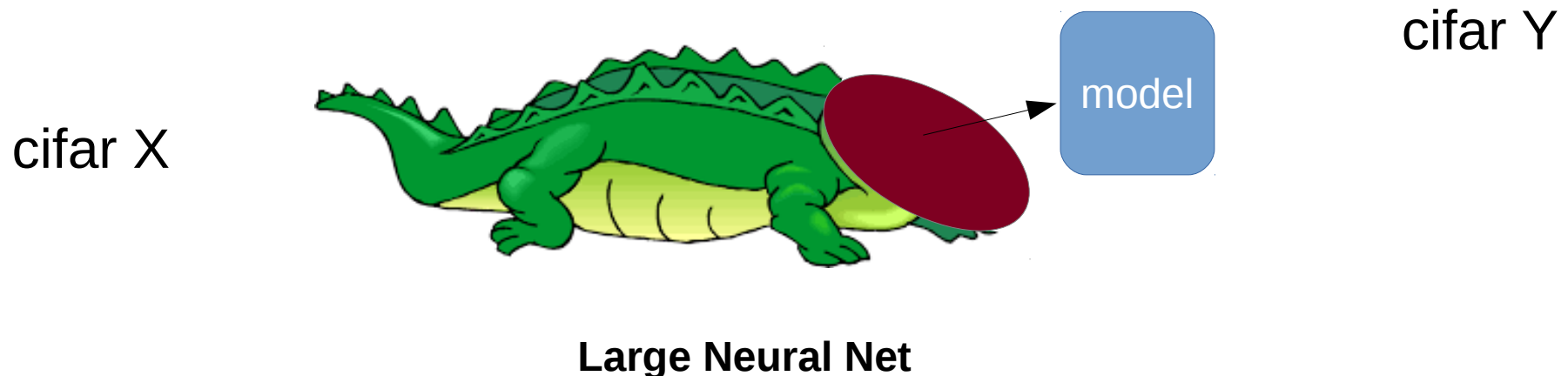


Large Neural Net

cifar Y

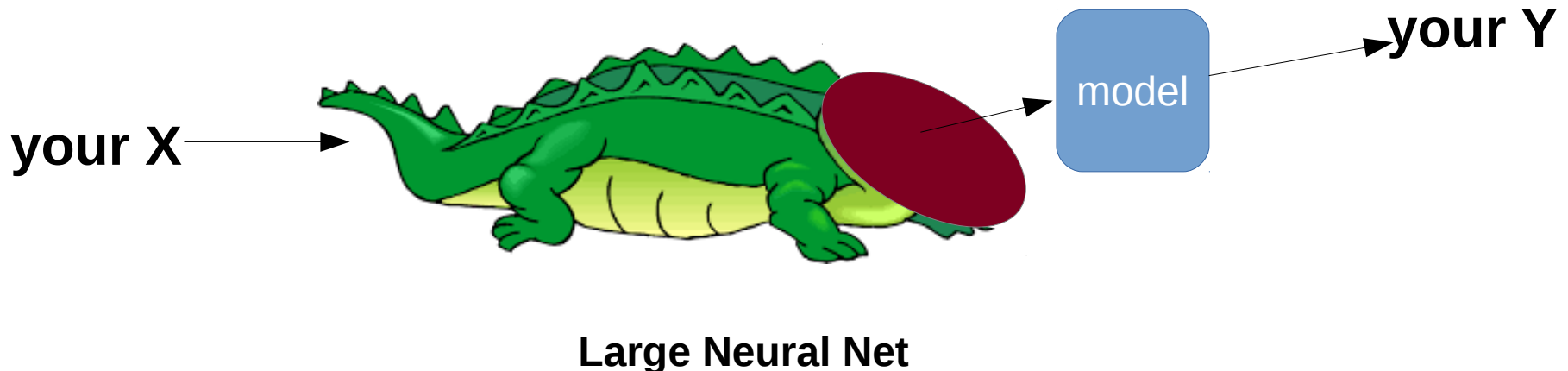
Pre-training

- 1. Train a network on large dataset
- 2. Take some intermediate layer
- 3. Build model on top of it



Pre-training

- 1. Train a network on large dataset
- 2. Take some intermediate layer
- 3. Build model on top of it
- 4. Train model for your objective



Reusing features

- Chop off “head”
- use “neck” as feature extractor
- Train ANY classifier
 - even random forest will do



Large Neural Net

Fine-tuning

- Chop off “head”
- “freeze” body (consider constant)
- Build new neural network in it's place
- Train “head” only for several iterations
- Un-freeze body and train full network

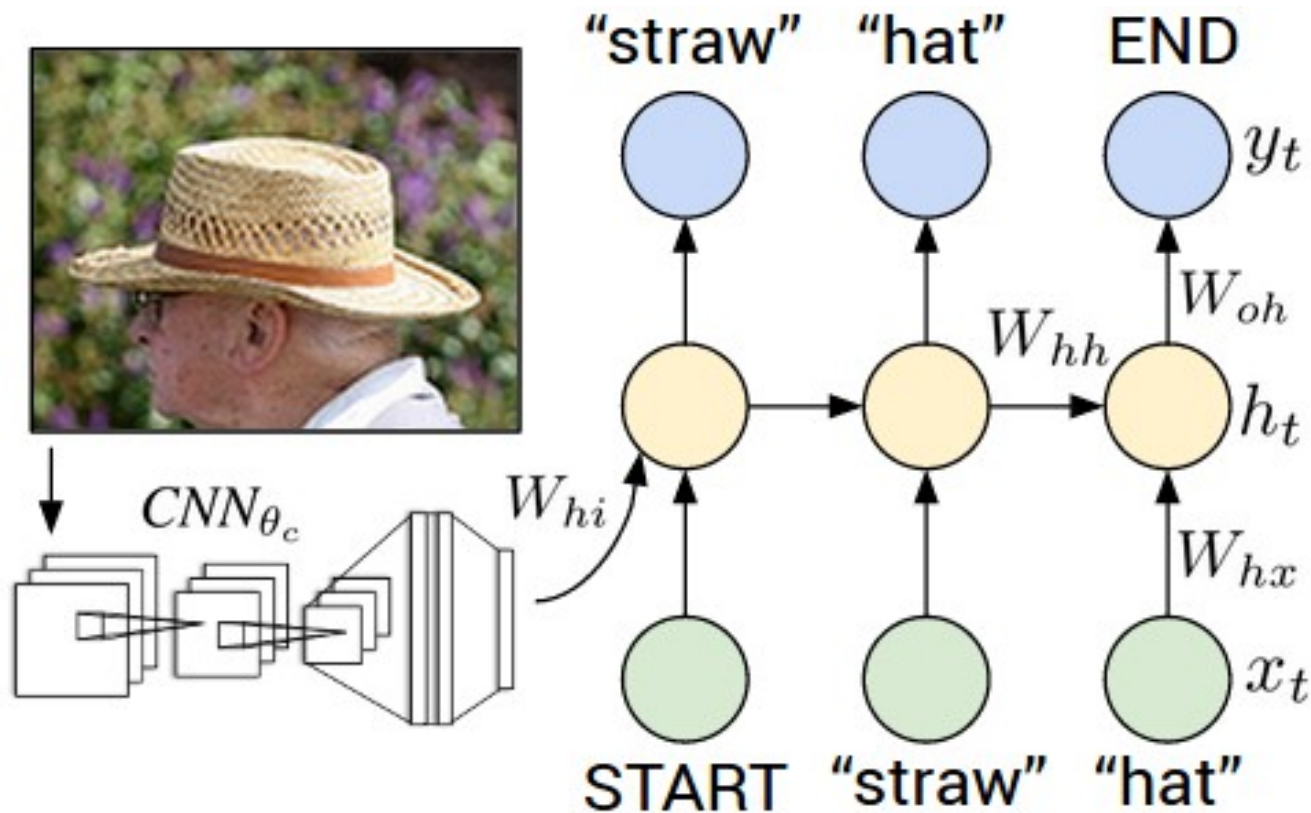


Large Neural Net

Model zoo

- Most deep learning frameworks have a library of pre-trained networks
 - e.g. VGG, GoogleNet, char-RNN
- You just need to download weights
- <https://github.com/Lasagne/Recipes/>

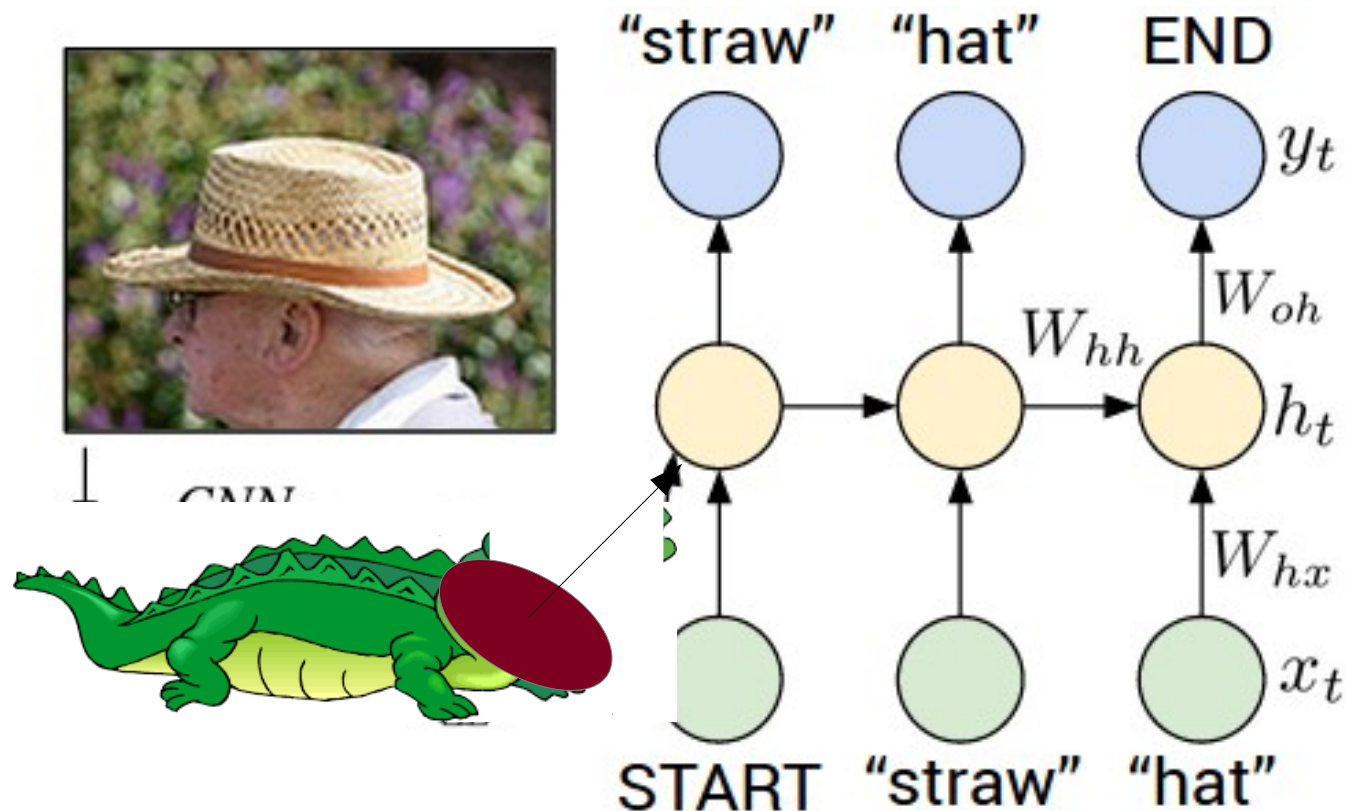
Task decomposition



Convolutional "encoder"

Recurrent "decoder"

Task decomposition



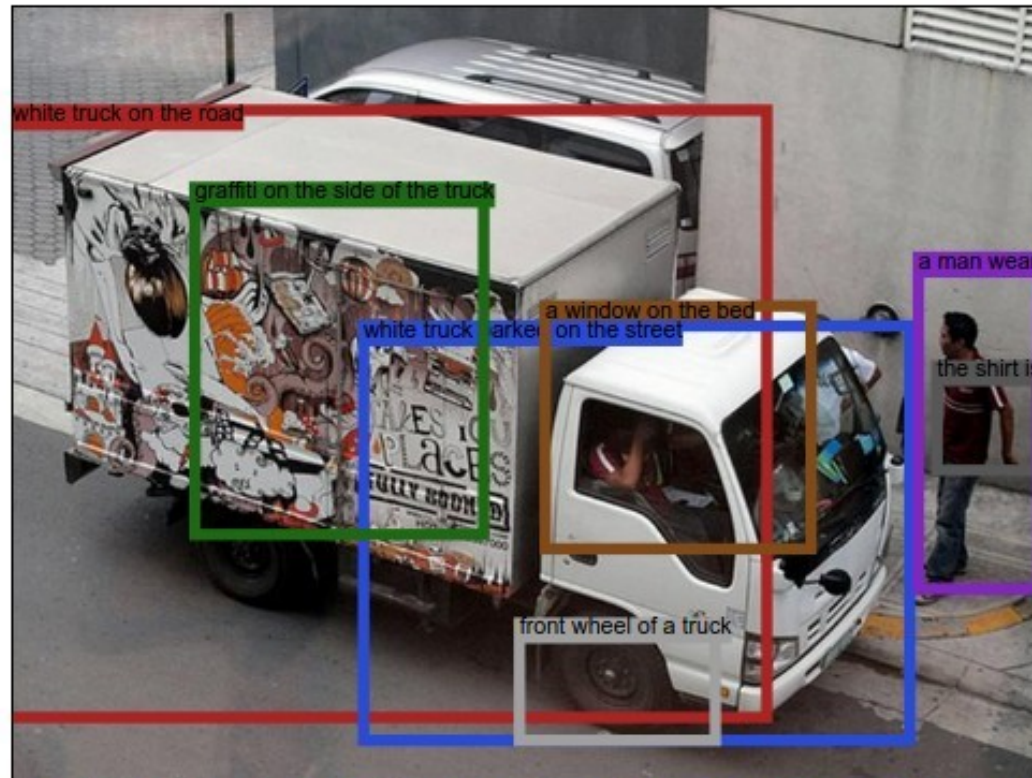
Convolutional "encoder"

Start with pre-trained

Recurrent "decoder"

Train from scratch

Captioning objects on image



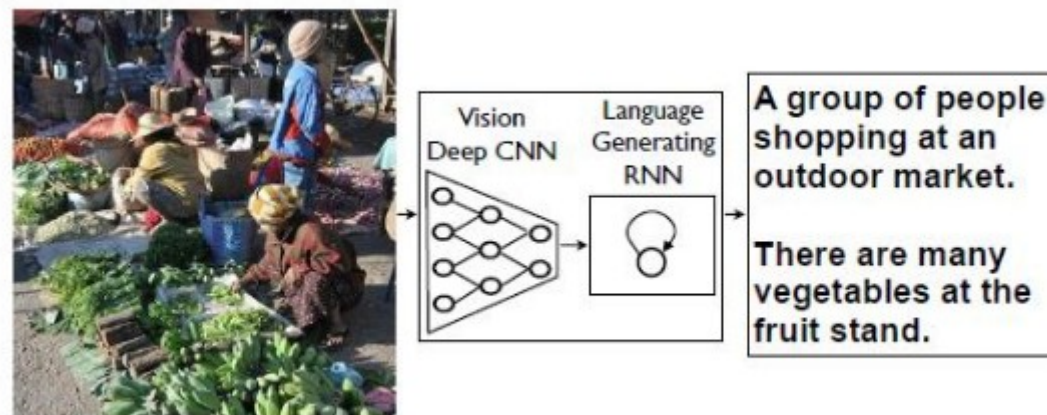
a white truck on the road. white truck parked on the street. the shirt is red. graffiti on the side of the truck. a window on the bed. a man wearing a black shirt. front wheel of a truck.

How could we approach this one?

Brace yourselves



Image captioning demos



- Demo - <http://stanford.io/2esMxOq>
- Upload your image - <http://bit.ly/2eAoueP>