An Introduction to Neural Machine Translation

ATA59 Carola F. Berger
TURN OFF
2-WAY RADIO
AND
CELL PHONE
Disclaimer 2

After this presentation, you will hopefully understand how neural MT works, but you will not understand what it does.
After this presentation, you will hopefully understand how neural MT works, but you will not understand what it does.

~100 million parameters
Outline

- Brief recap: previous MT approaches
- How do neural networks work?
- How do words get into and out of a neural network?
MT Approaches

Rules based:
Basically just grammar + dictionary

Statistical MT:
Chop sentences up into n-grams (sequences of n words) or phrases.
Training of the engine: Calculate frequency = probability in source and target language.
Translation after training: Chop source sentences into n-grams or phrases, apply previously calculated probabilities. 1-gram SMT = ?
MT Approaches

- **Rules based:**
  Basically just grammar + dictionary

- **Statistical MT:**
  Chop sentences up into n-grams (sequences of n words) or phrases.
  Training of the engine: Calculate frequency = probability in source and target language.
  Translation after training: Chop source sentences into n-grams or phrases, apply previously calculated probabilities.
  1-gram SMT = dictionary

Pros: Output is deterministic. No words missing.
Cons: Context!

- **Neural MT – this presentation**
Statistical MT

From G. M. de Buy Wenninger, K. Sima’an, PBML No. 101, April 2014, pp. 43
How Do Neural Networks Work?

A (not so) brief recap of last year’s presentation at ATA58. See also handouts as PDF in the app or on my website (see references).
Biological Neuron

Bruce Blaus, https://commons.wikimedia.org/wiki/File:Blausen_0657_MultipolarNeuron.png
Artificial Neuron
ARTIFICIAL NEURON

inputs

\[ x_1 \]

\[ x_2 \]

output

\[ w_1 \]

\[ w_2 \]
Artificial Neuron - Perceptron

If blob > 3 => output 1, else output 0
Artificial Neuron - Perceptron

If blob > 3 => output 1, else output 0
Artificial Neuron - Perceptron

If blob > 3 => output 1, else output 0
If blob > 3 => output 1, else output 0
Artificial Neuron - Perceptron

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

![Graph showing data points and output values](image-url)
Artificial Neuron - Perceptron
Artificial Neuron - Perceptron
Artificial Neuron - Perceptron
Artificial Neuron - Perceptron
Artificial Neuron - Perceptron
Artificial Neuron

inputs

$\mathbf{x}_1$

$w_1$

$w_2$

output
Neural Network

Human brain:


Carola F. Berger, An Introduction to NMT, ATA59
ARTIFICIAL NEURAL NETWORK

Adapted from: Cburnett, https://commons.wikimedia.org/wiki/File:Artificial_neural_network.svg
Artificial Neural Network

Training:

Adapted from: Cburnett, https://commons.wikimedia.org/wiki/File:Artificial_neural_network.svg

Carola F. Berger, An Introduction to NMT, ATA59
Artificial Neural Network

Training:

Feed in training data

Adapt weights ("arrows") according to difference between desired output and actual output, e.g. by backpropagation

Adapted from: Cburnett, https://commons.wikimedia.org/wiki/File:Artificial_neural_network.svg
Neural Net Example – Digit Recognition

Sample input data

```
8 9 3 1 4 5 9 0 3 3
5 3 7 6 7 5 8 8 5 3
8 9 8 5 7 2 0 9 8 4
4 6 6 5 0 3 9 6 8 9
8 / 9 3 5 9 3 3 2 7
8 5 / 3 9 8 0 4 7
9 8 8 1 5 6 5 9 4 9
6 5 0 0 2 7 4 8 3 /
4 5 2 2 1 2 4 8 /
4 6 9 2 2 7 6 0 8 5
```
Neural Net Example – Digit Recognition

Sample input (20x20 pixels)
Neural Net Example – Digit Recognition

Sample input (20x20 pixels)
Neural Net Example – Digit Recognition

Weights

400x25 dimensional weights

25x10 dimensional weights

Input Layer
20x20 pixels
(400 flat)

Hidden Layer
25 units

Output Layer
10 labels
digits 0-9
Neural Net Example – Digit Recognition

Weights to hidden units – “feature” extraction
Neural Net Example — Digit Recognition

Weights to hidden units
Neural Net Example – Digit Recognition

Weights to hidden units
Neural Net Example — Digit Recognition

Hidden units to output

- 400x25 dimensional weights
- 25x10 dimensional weights

Input Layer: 20x20 pixels (400 flat)
Hidden Layer: 25 units
Output Layer: 10 labels, digits 0-9
Neural Net Example – Digit Recognition

Hidden units to output

Input Layer
20x20 pixels (400 flat)

Hidden Layer
25 units

Output Layer
10 labels digits 0-9

400x25 dimensional weights

25x10 dimensional weights
Neural Net Example – Digit Recognition

Hidden units to output

400x25 dimensional weights

25x10 dimensional weights

Input Layer
20x20 pixels (400 flat)

Hidden Layer
25 units

Output Layer
10 labels
digits 0-9
Neural Net Example – Digit Recognition

Input         Internal convolution    Hidden       Output

Internal conv. 2

Input
Internal convolution
Hidden
Output

Carola F. Berger, An Introduction to NMT, ATA59 24
Neural Net Example — Digit Recognition

What happens with unknowns?
Klingon 6 [jav]
Neural Net Example – Digit Recognition

Klingon 6 [jav]

Input  Internal convolution  Hidden  Output

Internal conv. 2

Carola F. Berger, An Introduction to NMT, ATA59
Artificial Neural Network

Feed-forward neural net

Adapted from: Cburnett, https://commons.wikimedia.org/wiki/File:Artificial_neural_network.svg
How Do Words Get Into and Out of the Network?

Challenges for NMT:

- Input and output length not fixed, different sentence ordering in source and target languages
- Context
- Training metrics
How Do Words Get Into and Out of the Network?

Challenges for NMT:

- Input and output length not fixed, different sentence ordering in source and target languages => Use recurrent neural networks with attention or convolutional networks

- Context => document (or at least paragraph) level, not sentence level

- Training metrics
How Do Words Get Into and Out of the Network?

Recurrent neural net

Adapted from: Cburnett, https://commons.wikimedia.org/wiki/File:Artificial_neural_network.svg

Carola F. Berger, An Introduction to NMT, ATA59
How Do Words Get Into and Out of the Network?

Attention mechanism

(b) A person is standing on a beach with a surfboard.

How Do Words Get Into and Out of the Network?

Attention mechanism

How Do Words Get Into and Out of the Network?

**Source Text** ➔ **Encoder** ➔ **Hidden State** ➔ **Decoder** ➔ **Target Text**

RNN with attention

RNN with attention

tokenization (optional)
How Do Words Get Into and Out of the Network?

Source Text → Encoder → Hidden State → Decoder → Target Text

RNN with attention → RNN with attention
How Do Words Get Into and Out of the Network?

https://projector.tensorflow.org
How Do Words Get Into and Out of the Network?

Recall: SMT

From G. M. de Buy Wenninger, K. Sima’an, PBML No. 101, April 2014, pp. 43
Neural Nets - Recap

✓ Training = extraction of “features” (=patterns) from training data
Neural Nets - Recap

✓ Training = extraction of “features” (=patterns) from training data
✓ The more hidden layers and hidden units, the more parameters (possible overfitting!)
Neural Nets - Recap

✓ Training = extraction of “features” (=patterns) from training data
✓ The more hidden layers and hidden units, the more parameters (possible overfitting!)
✓ Beware: Garbage in -> worse garbage out!
Neural Nets - Recap

✓ Training = extraction of “features” (=patterns) from training data
✓ The more hidden layers and hidden units, the more parameters (possible overfitting!)
✓ Beware: Garbage in -> worse garbage out!
✓ ANNs work well for pattern recognition after training, including “context”
Neural Nets - Recap

- Training = extraction of “features” (=patterns) from training data
- The more hidden layers and hidden units, the more parameters (possible overfitting!)
- Beware: Garbage in -> worse garbage out!
- ANNs work well for pattern recognition after training, including “context”
- Completely unpredictable when confronted with new, hitherto unknown data
Training Data


Carola F. Berger, An Introduction to NMT, ATA59
Unpredictability

dog dog dog dog dog

dog dog dog dog
Unpredictability

dog dog dog dog dog dog dog

dog dog dog - reader email
**Unpredictability**

| dog  | dog  | dog  | dog  | dog  | dog  | dog  | dog  | dog  | dog  | dog  | dog  | dog  | dog  | dog  | dog  | dog |

Doomsday Clock is three minutes at twelve We are experiencing characters and a dramatic developments in the world

Carola F. Berger, An Introduction to NMT, ATA59
Doomsday Clock is three minutes at twelve. We are experiencing characters and dramatic developments in the world, which indicate that we are approaching the end times and Jesus' return.
References & Further Reading

- Slides at: https://www.CFBtranslations.com
- Handouts in app and also at https://www.CFBtranslations.com
- Google’s Tensorflow: https://www.tensorflow.org/
- Madly Ambiguous - game to illustrate how NMT deals with context: http://madlyambiguous.osu.edu