Sequence Transduction with Recurrent Neural Networks

Alex Graves

Representation Learning Workshop
ICML 2012
What is Sequence Transduction?

- Any task where input sequences are transformed into output sequences

- Practical examples: speech recognition, text-to-speech, machine translation, protein secondary structure prediction…

- Not so practical: Turing machines, human intelligence…

- Want a single framework able to handle as many kinds of sequence as possible
Recurrent Neural Networks

• RNNs can in principle learn any measurable sequence-to-sequence mapping to arbitrary accuracy (Hammer, 2000)

• They also work in practice: state-of-the-art results in handwriting recognition (Graves et al., 2008), text generation (Sutskever et al., 2011) and language modelling (Mikolov et al., 2010)

• Main strength is the robustness and flexibility of their internal representation of past events (a.k.a. memory)

• So they must be great for sequence transduction, right? Just train them to match input sequences to target sequences…
Variable Output Length

• For many sequence transduction you don’t know in advance how long the output will be

“Πάντα ρέι καὶ οὐδὲν μένει”
→ “All flows, nothing stays”
or “Everything flows; nothing remains”
or “Everything gives way and nothing stays fixed”

• This is a problem for standard RNNs, because the training targets have to be pre-aligned with the inputs
Structured Prediction

- Standard RNNs just map from inputs to outputs
- This ignores a valuable source of information: the outputs so far (language modelling / structured prediction)
- Can solve both problems by using two RNNs: one to model input-output dependencies (transcription) and another to model output-output dependencies (prediction)
- Each output is therefore conditioned on the whole input sequence and all previous outputs
Probabilistic Model

- Input sequence: \( \mathbf{x} = (x_1, x_2, \ldots, x_T) \in \mathcal{X}^* \)
- Target sequence: \( \mathbf{y} = (y_1, y_2, \ldots, y_U) \in \mathcal{Y}^* \)
- Bidirectional transcription network: \( a(t, \mathbf{x}), 1 \leq t \leq T \)
- Unidirectional prediction network: \( b(\mathbf{y}_{1:u}), 0 \leq u \leq U \)
- Output distribution:
  \[
  \Pr(v|t, u) = \frac{f(v, a(t, \mathbf{x}), b(\mathbf{y}_{1:u}))}{Z}, \quad v \in \mathcal{Y} \cup \emptyset
  \]
  For discrete targets, \( f \) is just the softmax of \( a + b \)

- To sample: start at \( t=0, u=0 \); if \( v \) is \( \emptyset \), output nothing and increment \( t \), else output something and increment \( u \)
Output Lattice

- All paths from bottom left to top right (like the red one) are ways of outputting “DOG” given “HUND”
- Multiply the transition probabilities to get the path probability
- Sum over all possible paths to get the total probability of “DOG”
- Can do this efficiently with a forward-backward algorithm
- Can draw a similar lattice for any string, hence have a distribution over sequences of all lengths

\[
\begin{align*}
\Pr(\text{red}) &= \Pr(D|1, 0) \Pr(O|1, 1) \Pr(\varnothing|1, 2) \\
&\quad \Pr(\varnothing|2, 2) \Pr(G|3, 2) \Pr(\varnothing|3, 3) \Pr(\varnothing|4, 3)
\end{align*}
\]
Speech Recognition
Handwriting Recognition

In other ways, too, life for the ordinary
Training and Testing

- The whole system is differentiable, so the log-likelihood of the targets can be maximised with gradient descent.

- Testing (decoding in HMM speak) is trickier: need a search algorithm (e.g. beam search) to look for the most probable output sequence.
Results

• For sequence labelling problems (e.g. speech and handwriting recognition) can compare the transducer with CTC RNNs (essentially the transcription network without the prediction network)

• CTC is state-of-the-art in offline handwriting recognition, so any improvement here is interesting

• Transducer gives better label error rate (e.g. TIMIT phoneme error rate reduced from 25.5% to 23.2%)

• Can do lexicon/language model free word recognition with the transducer (unlike CTC)

• But so far CTC with an external language model gives better word error rates
Nous avons découvert le mois dernier seulement que malgré la signature de l'union sociale il y a un an et malgré le fait que le gouvernement du Canada a accru immédiatement les transferts, en février 1999, de 3,5 milliards de dollars auxquels viendront s'ajouter 8 autres milliards de dollars, les provinces, notamment le Québec, ont pris cet argent pour, comme il l'a dit, se mettre plus de fonds dans les poches.

We found only last month that notwithstanding the social union we signed one year ago and notwithstanding the fact that the Government of Canada increased the transfers by $3.5 billion immediately last February 1999, with an additional $8 billion to follow, the provinces and in particular the province of Quebec took the additional money, as he said mettre plus de fonds dans les poches.

We have discovered last year only that the signature of the social union there is a year and the fact that the Government of Canada has been immediately transferred in February 1999, $3.5 billion with addition other billions, the provinces, including Quebec, that money for the more funding in the pockets.
[SHOW DEMO]
Extensions and Future Work

• Pre-training the prediction net (e.g. on large text corpus) improves results

• Can minimise expected word/label error rate instead of log loss: predict samples instead of targets

• RNNs are very general predictive models, but for tasks with discrete outputs it might be better to use more conventional language models, or some mix of the two

• Word level targets?

• Would like to look at tasks with continuous outputs, e.g. text-to-speech