Deep Temporal Architectures

Andrew H. Fagg
GRU Layer Notes

Conventional wisdom: interchangeable with LSTM

- **activity_regularizer**: similar to kernel regularizers, but:
  - Sum abs activation (L1), or
  - Sum squared activation (L2)
  - Both: push latent representation to be more sparse
- **dropout**: prob of dropping input units
- **recurrent_dropout**: prob of dropping recurrent units
- **return_sequences**: output tensor includes time dimension
- **stateful** (Boolean): recurrent units keep state between examples
WaveNet
Stacking small convolutions to create large-scale filters

towardsdatascience.com
Implementation Notes

1-D convolution (we have done lots of 2-D conv so far)

- **kernel_size**: can be small
- **padding=”causal”**: kernel only “looks” at this time and before (it is not allowed to look ahead in time)
- **dilation_rate**:
  - 1 = use neighboring “pixels” from the input
  - 2 = use every other pixel
  - …
RNN Architectures

Image from: Andrej Karpathy
Basic Text Classification Architecture

- Text to 1-Hot encoding
- Embedding: compression of word-based encoding
- Bidirectional RNN: place beginning and ending of sentence on equal footing

www.tensorflow.org/tutorials/text/text_classification_rnn
Machine Translation

ENCODER

DECODER

<GO>

comment

allez

vous

?
Machine Translation

● Special control symbols: Start and End-of-Sentence
● During decoding
  ○ Output is a prob distribution over word possibilities
  ○ Must pick one
  ○ This one is then provided as input
Attention

So far:

● Encoder is a RNN
● Decoder has attention:
  ○ Weighted average of the encoder outputs
  ○ Attention mechanism allows the decoder to weigh certain words higher than others in making a decoding decision
  ○ Decoder does not rely on RNN to develop representation
Attention

Down Sides:

- Encoder is a RNN!
- The first words in the input do not have access to the last words
  - This context could be important in interpreting the first words
Transformers

“Attention is All You Need”

● Also use attention in the encoder
  ○ “Self attention”

● Dispense with RNNs entirely
  ○ No deep backpropagation of errors
  ○ Can do much of the computation in parallel
Transformers

New pieces:

● Attention in the encoder: first word can “see” the last one
● Multi-headed attention:
  ○ One word can “see” multiple words at once to decide how best to represent
● Positional encoding:
  ○ Replaces RNN
  ○ Allows us to still represent (relative) positions of words
Positional Embeddings

http://nlp.seas.harvard.edu/images/the-annotated-transformer_49_0.png
Positional Embeddings

- Each position: one vector
- Computing the difference between two positional encodings:
  - Linear operation
  - Difference is independent of t!
  - So: it doesn’t matter where the words are in the sentence as long as they have the same relative positions

https://kazemnejad.com/blog/transformer_architecture_positional_encoding/
Positional Embeddings

Benefits:

● Difference between two positions: linear computation + independent of location in the sentence!
● Positional inputs are bounded (+/-1)
● Better generalization to longer sequences than what the model has been trained on
Attention

- Q: Query
- K: Key
- V: Value

Scaled Dot-Product Attention

Attention(Q, K, V) = softmax(\(\frac{QK^T}{\sqrt{d_k}}\))V

Multi-Head Attention

MultiHead(Q, K, V) = Concat(head₁, ..., headₕ)W^O

where headᵢ = Attention(QWᵢ^Q, KWᵢ^K, VWᵢ^V)
Transformer Architecture

https://medium.com/@yacine.benaffane/transformer-self-attention-part-1-2664e10f080f
https://medium.com/@yacine.benaffane/transformer-self-attention-part-1-2664e10f080f
Masked Attention

● Don’t want the decoder to be able to “look ahead” at the answer
  ○ while available at time of training, it is not available during recall
● For future time steps, set attention alpha to zero
Evaluation Metric

BLEU: BiLingual Evaluation Understudy

- Counts number of matching N-grams between the translated sentence and the ground truth
- Easy and cost efficient to compute
- Not very sensitive to small changes in word/phrase orders
  - Which is what we want