BST 261: Data Science II
Lecture 16

Course Review

Heather Mattie
Harvard T.H. Chan School of Public Health
Spring 2 2021

Recipe of the Day!

Aperol Spritz Cocktail

Non-alcoholic Aperol Spritz

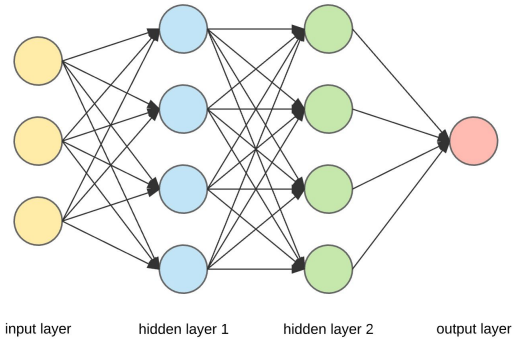
Course Review

What have we learned?

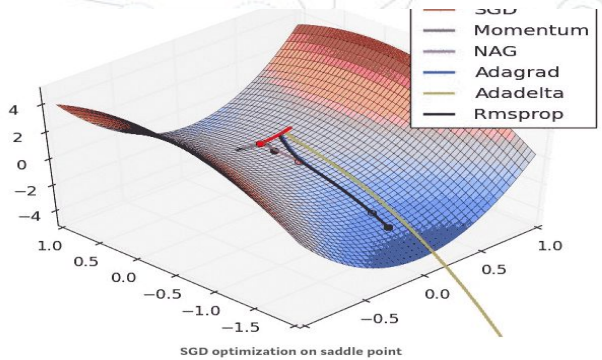
Quite a bit!

- Multilayer perceptrons → CNNs → RNNs → Advanced architectures

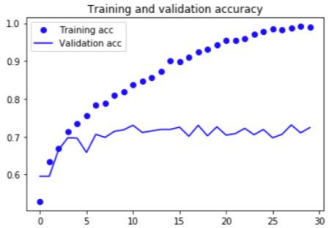
- Network architecture
  - Hidden units
  - Layers
  - Activation function
  - Loss functions
  - Optimization algorithms
  - Batch size

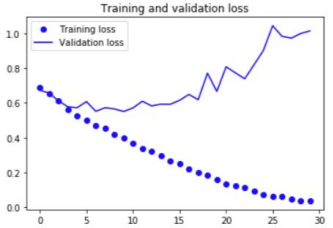


What have we learned?



- How networks learn
  - Gradient descent/ascent
  - Backpropagation
  - Forward pass and backward pass
  - Visualizing filters
  - Dense layer vs convolution layer vs RNN/LSTM/GRU layer

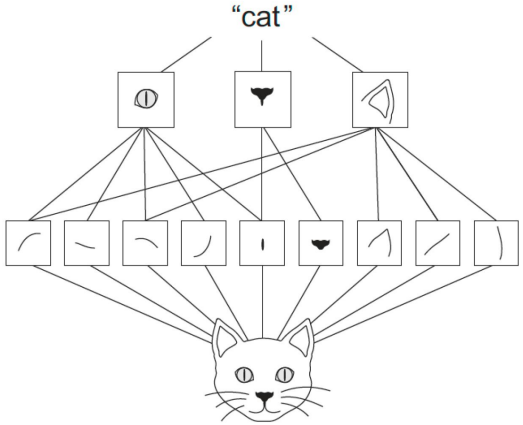


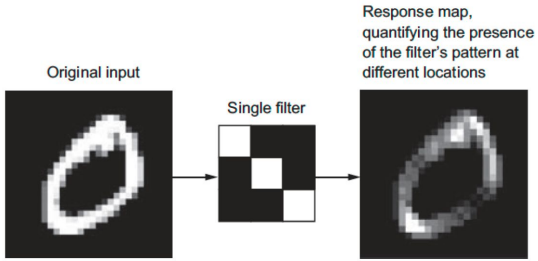


- Model performance
  - Underfitting vs overfitting
  - Regularization techniques
    - Dropout, L2 and L1 norms, network size
  - Bias/variance tradeoff

What have we learned?

- CNNs
  - Padding
  - Pooling
  - Strides
  - Filters
  - Translation invariance
  - Hierarchical learning
  - Lower layer representations vs higher layer representations
  - Data format (3D vs 4D tensors)
  - Object detection and localization
  - Face recognition
  - 1D CNN for sequential data
  - Landmark detection
  - Data augmentation
  - Neural style transfer

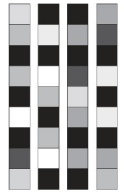




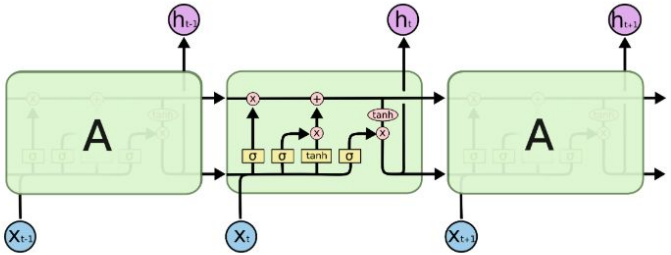
What have we learned?

- RNNs
  - Different types of sequential data
  - How RNNs preserve order in this type of data
  - SimpleRNN vs LSTM vs GRU layers
  - Tokens and tokenization
  - One-hot encoding and hashing
  - Word embeddings
  - Word2Vec and Glove
  - Time series data
  - Recurrent dropout
  - Text generation
  - Bidirectional recurrent layers

One-hot word vectors:
- Sparse
- High-dimensional
- Hardcoded



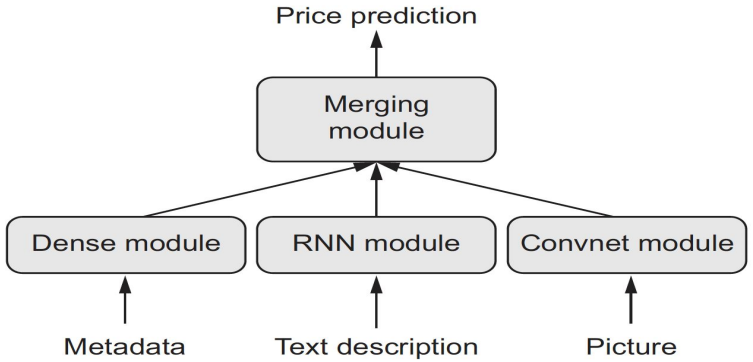
Word embeddings:
- Dense
- Lower-dimensional
- Learned from data



The repeating module in an LSTM contains four interacting layers.

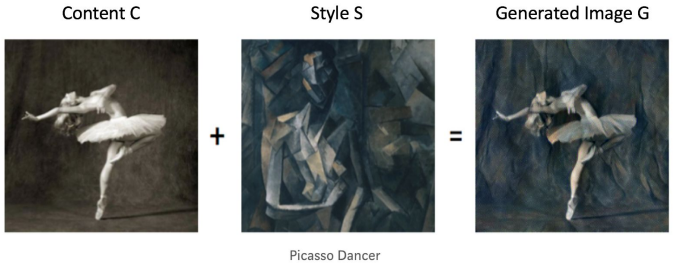
What have we learned?

- Advanced network architectures
  - One-to-many (multi-output/multi-head models)
  - Many-to-many
  - Many-to-one (multi-modal models)
  - Directed acyclic graphs
- Advanced architecture patterns
  - Batch normalization
  - Hyperparameter optimization
  - Model ensembling
- Implementation in Keras
  - Tensor (data) manipulation
  - Sequential model
  - MLP, CNN, RNN
  - Using GCP
  - Functional API

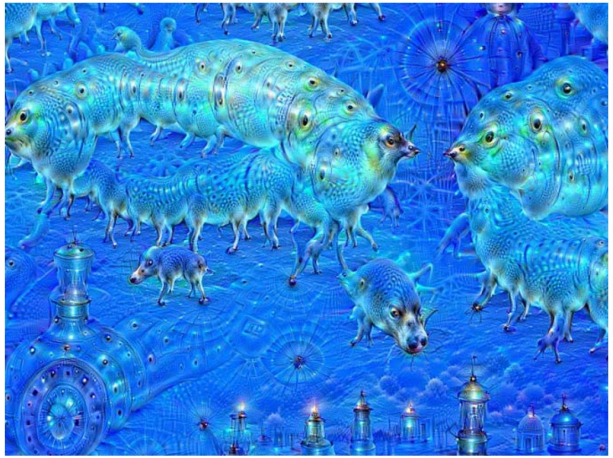


What have we learned?

- Advanced topics
  - Variational autoencoders (VAEs)
  - Generative adversarial networks (GANs)
  - Reinforcement learning (RL)
  - DeepDream
  - Neural style transfer
  - Text generation

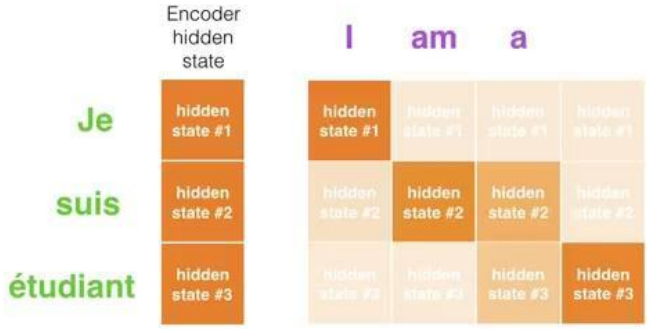


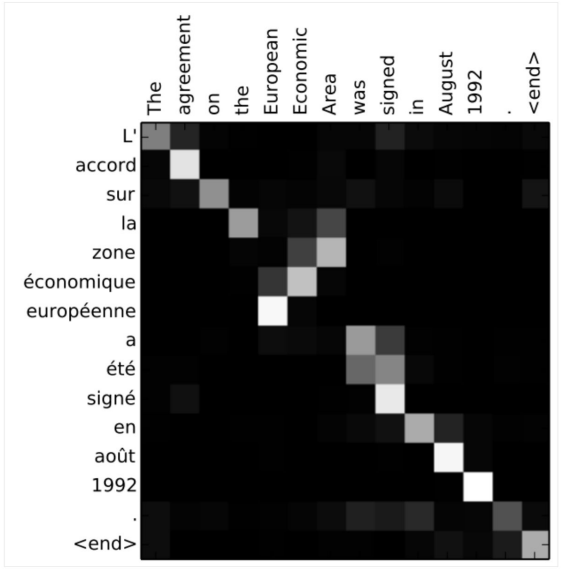
Picasso Dancer



Attention

- Scoring is done by the decoder at each time step
  - For each output word, scoring maps important / relevant words from the input sequence - higher weight means more relevance
  - This helps with the accuracy of the output prediction





You can see how the model paid attention correctly when outputing "European Economic Area". In French, the order of these words is reversed ("européenne économique zone") as compared to English. Every other word in the sentence is in similar order.

Transformers

- Solve all of the problems with classic RNNs
  - Allow for parallel computing
  - Use **attention**
    - Helps with loss of information problem

- [Attention is all you need paper]
  - December 2017
  - Huge breakthrough in NLP

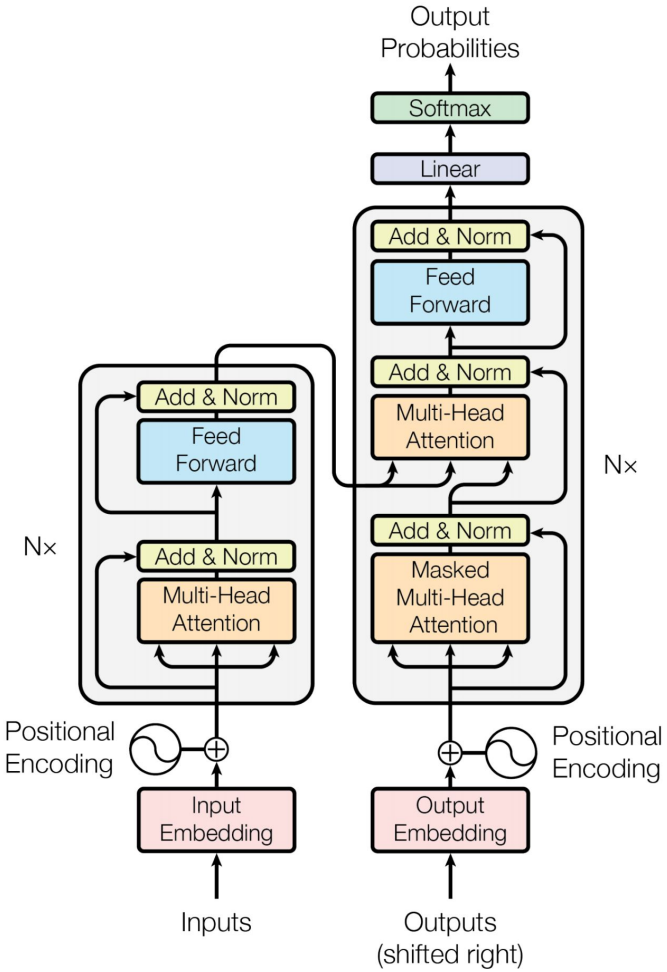


Figure 1: The Transformer - model architecture.

Transformers

- State of the art models
  - GPT-2, GPT-3
  - BERT
  - T5

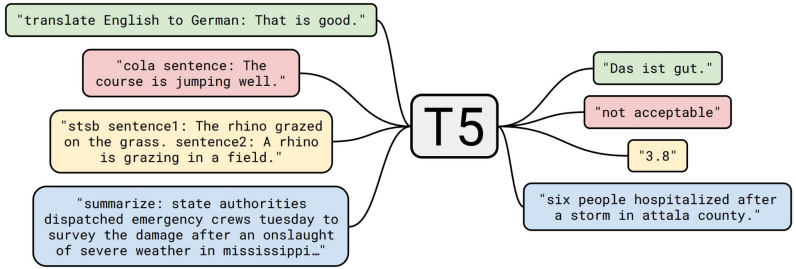
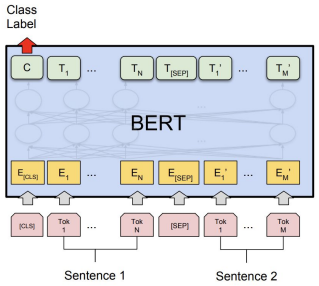
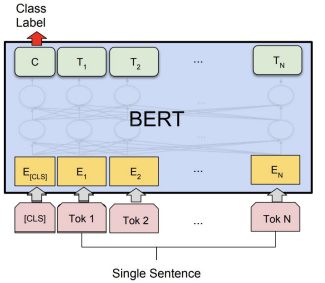


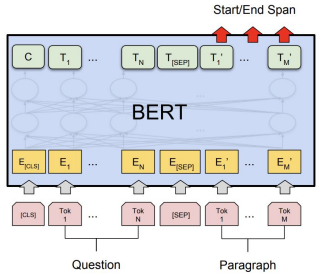
Figure 1: A diagram of our text-to-text framework. Every task we consider—including translation, question answering, and classification—is cast as feeding our model text as input and training it to generate some target text. This allows us to use the same model, loss function, hyperparameters, etc. across our diverse set of tasks. It also provides a standard testbed for the methods included in our empirical survey. "T5" refers to our model, which we dub the "**T**ext-**t**o-**T**ext **T**ransfer **T**ransformer".



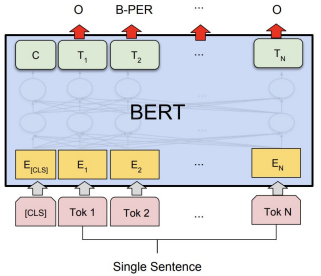
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA

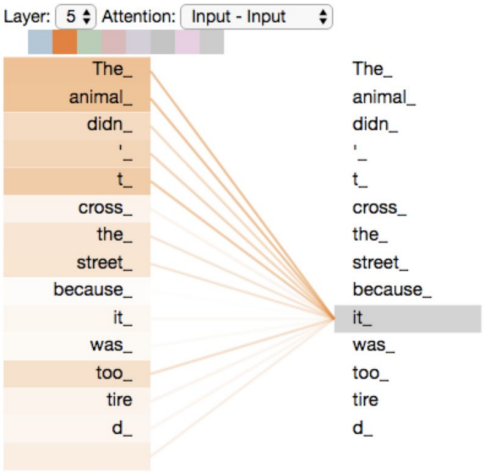


(c) Question Answering Tasks: SQuAD v1.1

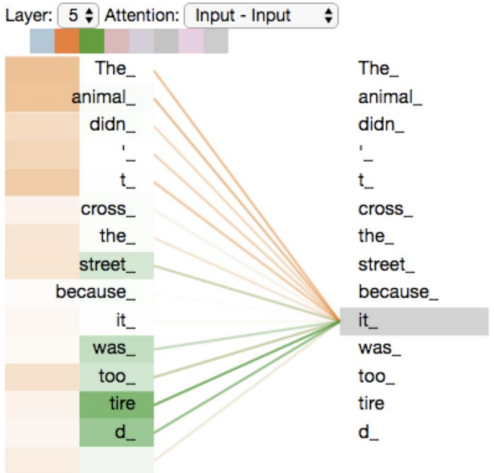


(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

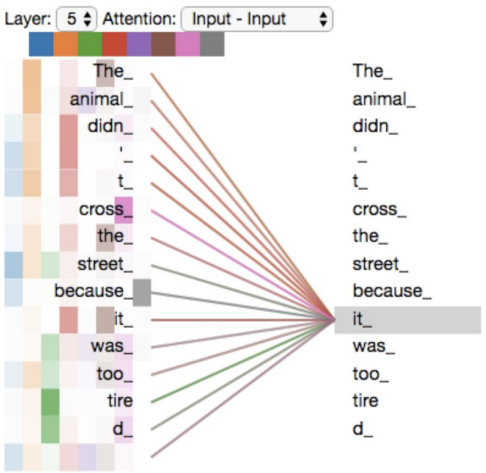
Multi-Head Attention



Self-attention



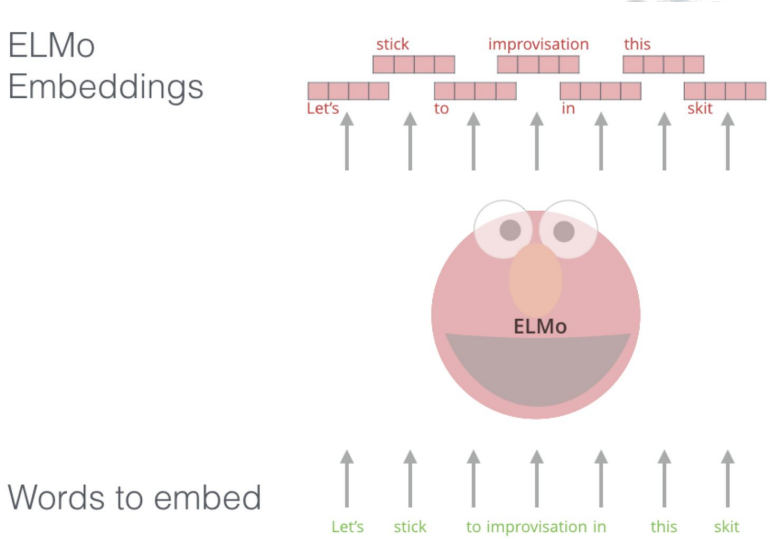
2-head Self-attention



8-head Self-attention

ELMo

- Recall: word embeddings are vector representations of words
  - We've used GloVe in class and in lab
  - Only 1 embedding for each word (fixed embeddings)
- Problem: the same word can have different meanings depending on the **context** in which it is used
- ELMo looks at an entire sequence before assigning each word it in an embedding
- Uses a bi-directional LSTM trained on a specific task



Bidirectional Encoder Representations Transformer (BERT)

- Has been described as the “marking the beginning of a new era in NLP”
  - Has broken several records for language-based tasks
  - Has been trained on massive datasets
  - Was made available as a pre-trained network
  - Now available in multiple sizes
- [Original paper]
- [GitHub repository]
- [Great post]