Agenda

- Brief Intro to Deep Learning
  - Neural Networks
- Word Representation Learning
  - Neural word representation
  - word2vec with Negative Sampling
  - Bias in word representation learning

---Break---

- **Recurrent Neural Networks**
- Attention Networks
- Document Classification with DL

Recurrent Neural Networks

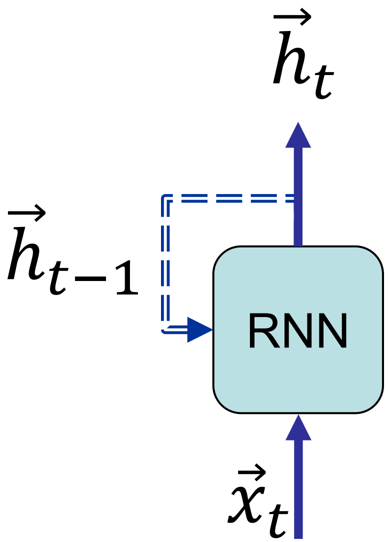
- Encodes/Embeds a sequence of entities (vectors) such as …
  - Sequence of word vectors
  - Time series

- … to a final composed vector as well as intermediary vectors on each time step

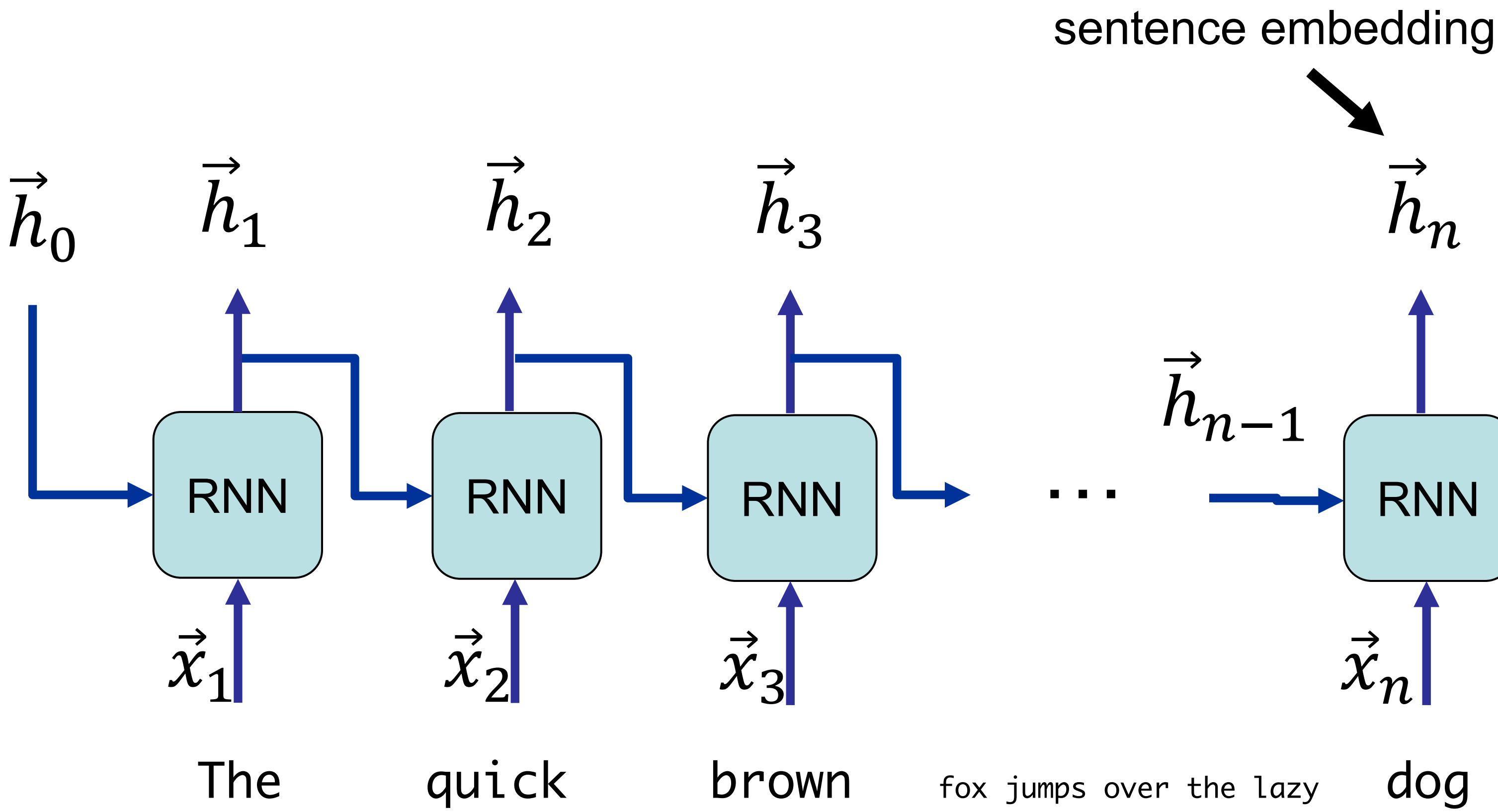
- The output is a function of input and the output of the previous time step

- Output $\vec{h}_t$ is also called hidden state

- With hidden state $\vec{h}$, the network access to a sort of memory from previous entities



RNN - unfolded

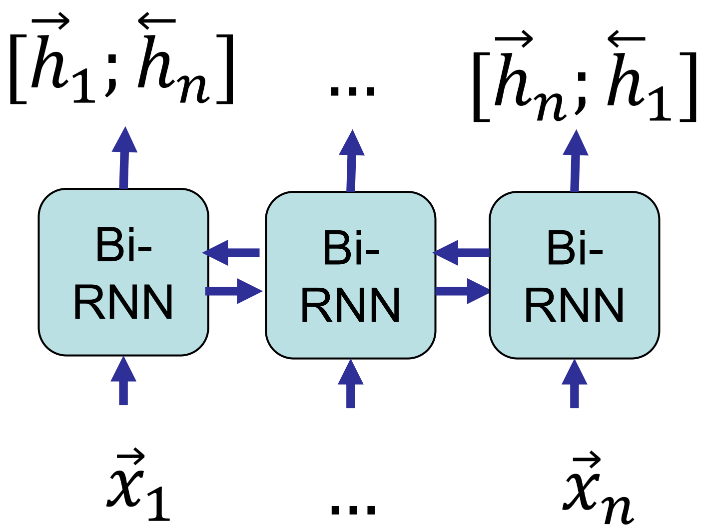


Types and Bidirectional

- Common types of RNN
  - Standard (Elman) RNN
  - Gated Recurrent Unit (GRU)
  - Long Short-Term Memory (LSTM)

Bidirectional RNN

- Reading the sequence from
  - Beginning to end $\vec{h}_1$ to $\vec{h}_n$
  - End to beginning $\overleftarrow{h}_n$ to $\overleftarrow{h}_1$
- Output at time step *t* is the concatenation of two hidden states $\vec{h}_t$ and $\overleftarrow{h}_{n-t}$



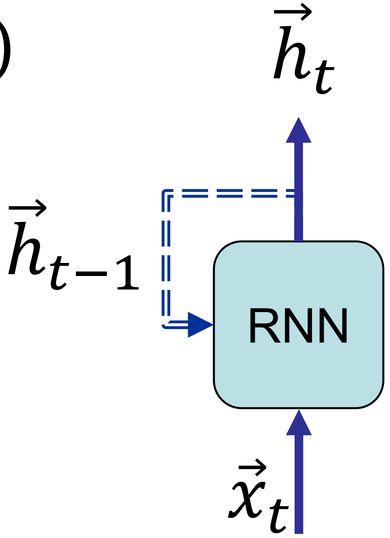
Standard RNN

- General form of the RNN function

$$\vec{h}_t = RNN(\vec{x}_t, \vec{h}_{t-1})$$

- First projects input $\vec{x}_t$ with parameter matrix W_{ih} , and previous hidden state $\vec{h}_{t-1}$ with parameter matrix W_{hh}, and then applies a non-linearity function on their sum:

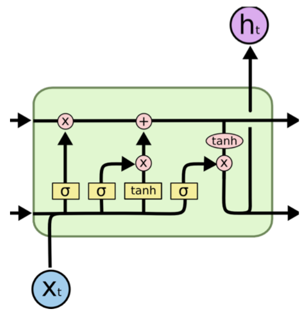
$$\vec{h}_t = \tanh(\vec{x}_t \cdot W_{ih} + \vec{h}_{t-1} \cdot W_{hh})$$

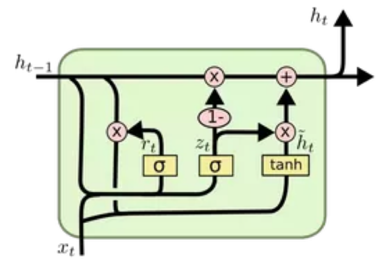


➢ Parameters are shown in red
➢ For simplicity biases are removed from the formulas

RNNs with Gates

- Two problems of Standard RNN:
  - Exploding gradient: approached with gradient clipping
  - Vanishing gradient: approached with gated RNNs such as GRU and LSTM by learning to "forget" the some parts of the memory





source

Gate Vector

- Commonly, a vector with values between 0 and 1, used for elementwise multiplication to an entity vector $\vec{v}$

$$\vec{g} \odot \vec{v}$$

- Acts as a gate on information flow of the entity vector

Gated Recurrent Unit (GRU)

- Calculate reset gate from input $\vec{x}_t$ and previous hidden state $\vec{h}_{t-1}$

$$\vec{r}_t = \sigma(\vec{x}_t \cdot W_{ir} + \vec{h}_{t-1} \cdot W_{hr})$$

- Calculate novel information vector $\vec{n}_t$ from input $\vec{x}_t$ and a “forgotten” part of previous hidden state $\vec{h}_{t-1}$

$$\vec{n}_t = tanh(\vec{x}_t \cdot W_{ih} + \vec{r}_t \odot (\vec{h}_{t-1} \cdot W_{hh}))$$

- Calculate update gate from input $\vec{x}_t$ and previous hidden state $\vec{h}_{t-1}$

$$\vec{z}_t = \sigma(\vec{x}_t \cdot W_{iz} + \vec{h}_{t-1} \cdot W_{hz})$$

- Finally output is composed of the novel information $\vec{n}_t$ and previous hidden state $\vec{h}_{t-1}$, decided by update gate

$$\vec{h}_t = (1 - \vec{z}_t) \odot \vec{n}_t + \vec{z}_t \odot \vec{h}_{t-1}$$

Agenda

- Brief Intro to Deep Learning
  - Neural Networks
- Word Representation Learning
  - Neural word representation
  - word2vec with Negative Sampling
  - Bias in word representation learning

---Break---

- Recurrent Neural Networks
- **Attention Networks**
- Document Classification with DL

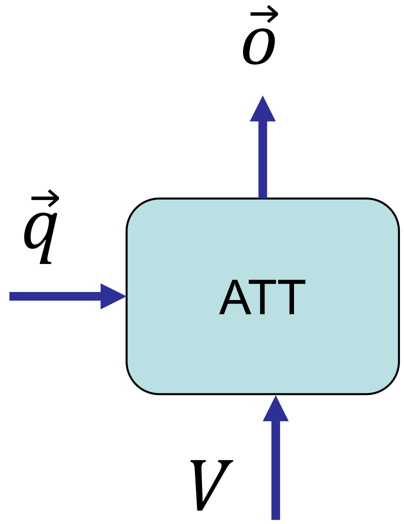
Attention Networks

- Encodes/Embeds a set of vectors to a composed vector

- Given a query vector $\vec{q}$ and a matrix of values V, an attention network "looks up" the query in the values, and produce output vector $\vec{o}$

- General form of an attention network as a function

$$\vec{o} = ATT(\vec{q}, V)$$



➢ In general, query is also a matrix. Here it is assumed as a vector for simplicity

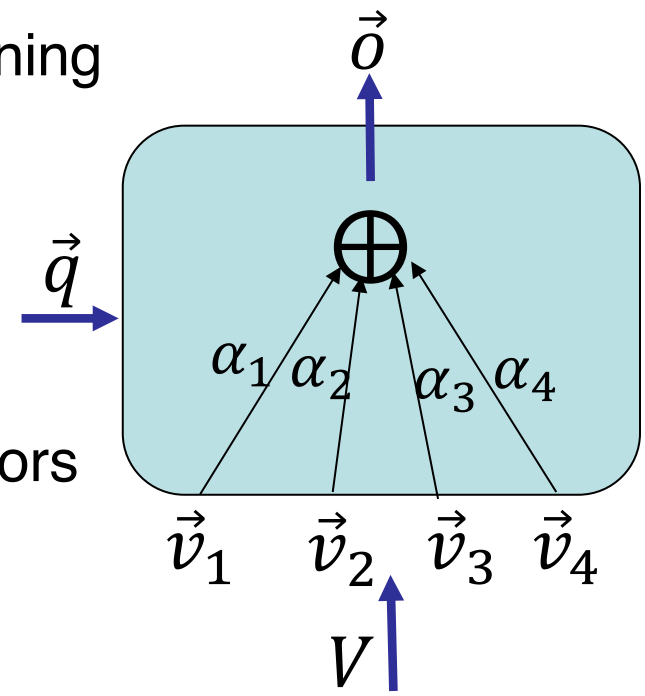
Attention Networks - details

- Given the query, an attention network learns to assign some amount of attention α_i on each value vector $\vec{v}_i$ using the attention function f

$$\alpha_i = f(\vec{q}, \vec{v}_i)$$

- where α is a probability distribution, meaning

$$\sum_{i=1}^{n} \alpha_i = 1$$



- Output is the weighted sum of value vectors

$$\vec{o} = \sum_{i=1}^{n} \alpha_i . \overrightarrow{v_i}$$

Method 1 - Scaled Dot-Product Attention

- First non-normalized attention $\tilde{a}_i$ is calculated by a simple dot product

$$\tilde{a}_i = \frac{\vec{q}.\vec{v}_i}{\sqrt{d}}$$

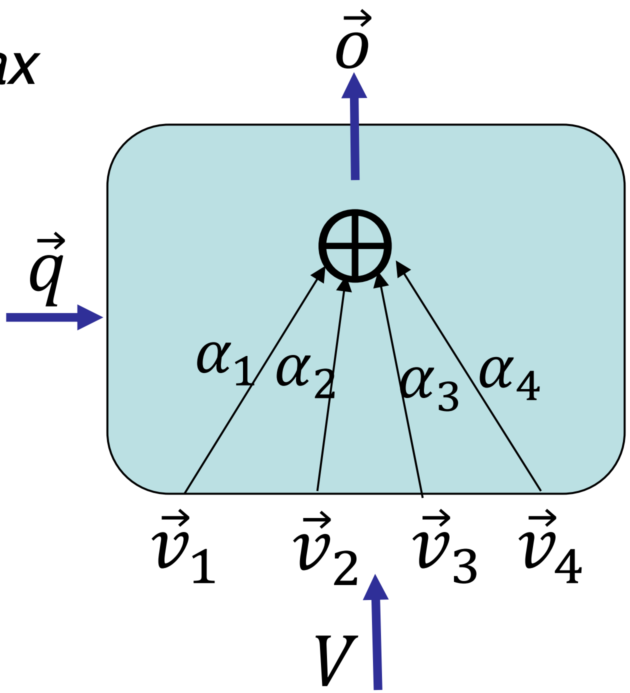
- where *d* is the dimension of vectors

- Attentions are then normalized with *softmax*

$$\alpha_i = softmax(\vec{\tilde{a}})_i$$

- As before, output is the weighted sum

$$\vec{o} = \sum_{i=1}^{n} \alpha_i.\overrightarrow{v_i}$$



Method 2 - Multi-Layer Perceptron Attention

- First non-normalized attention $\tilde{a}_i$ is calculated by a neural network

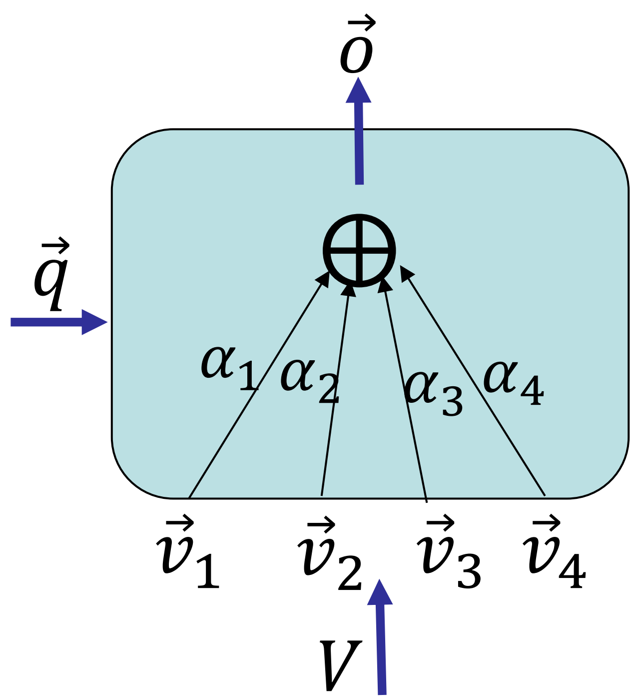
$$\tilde{a}_i = \vec{u} \cdot tanh(\vec{q} \cdot W_1 + \vec{v}_i \cdot W_2)$$

- As before, attentions are normalized with *softmax*

$$\alpha_i = softmax(\vec{\tilde{a}})_i$$

- and output is …

$$\vec{o} = \sum_{i=1}^{n} \alpha_i . \vec{v_i}$$



➢ Model parameters are shown in red

Agenda

- Brief Intro to Deep Learning
  - Neural Networks
- Word Representation Learning
  - Neural word representation
  - word2vec with Negative Sampling
  - Bias in word representation learning

---Break---

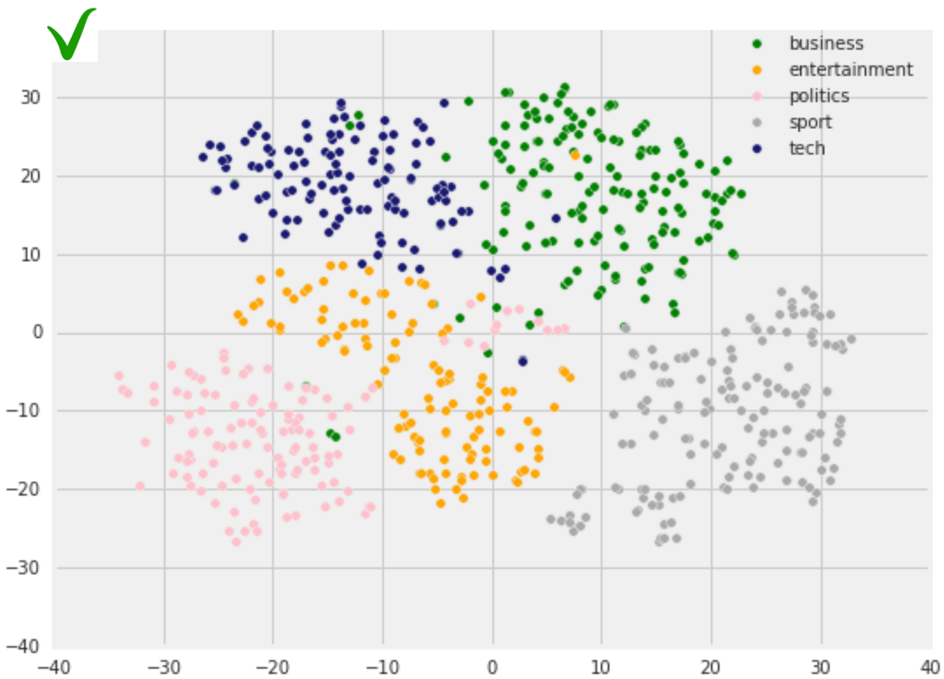
- Recurrent Neural Networks
- Attention Networks
- **Document Classification with DL**

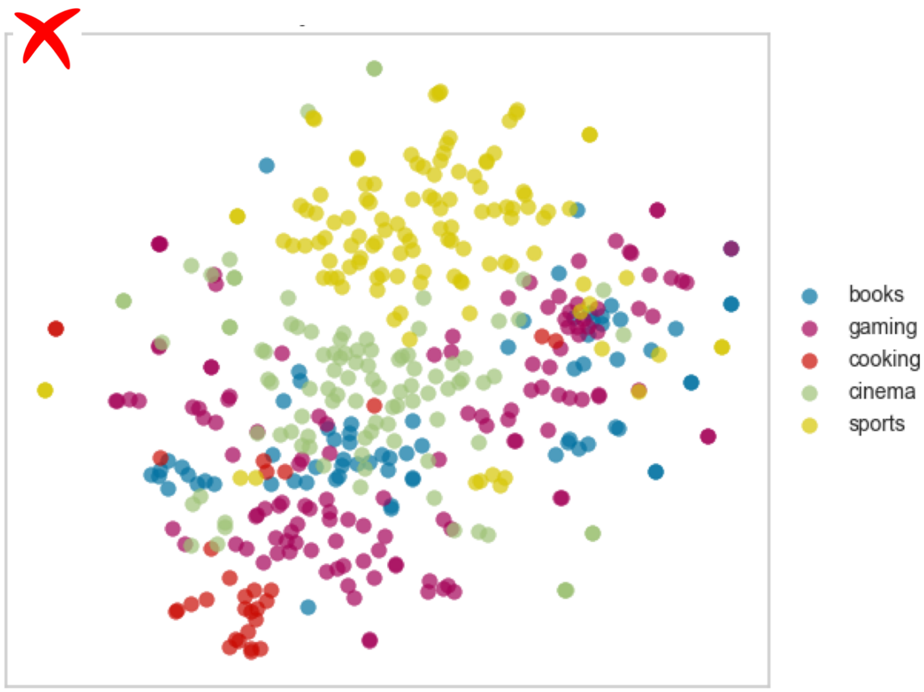
Document Classification (Recap)

- Create a document representation, e.g. using
  - TF-IDF → spare vectors
  - Principle Component Analysis (PCA) → dimensionality reduction
  - Latent Semantic Indexing (LSI) → semantic vectors
  - Latent Dirichlet Allocation (LDA) → topic-based vectors
  - Deep Learning
- Steps
  - Given document representations of training data, learn a classifier to predict the classes
  - Use the classification model to predict the classes of the test-set documents
  - Evaluate the predictions

Document Representation

- Document representation is the key!
- The classes can be more effectively predicted when document representations are linearly separable.





Two sample document representation sets, projected to two-dimensional spaces

source source

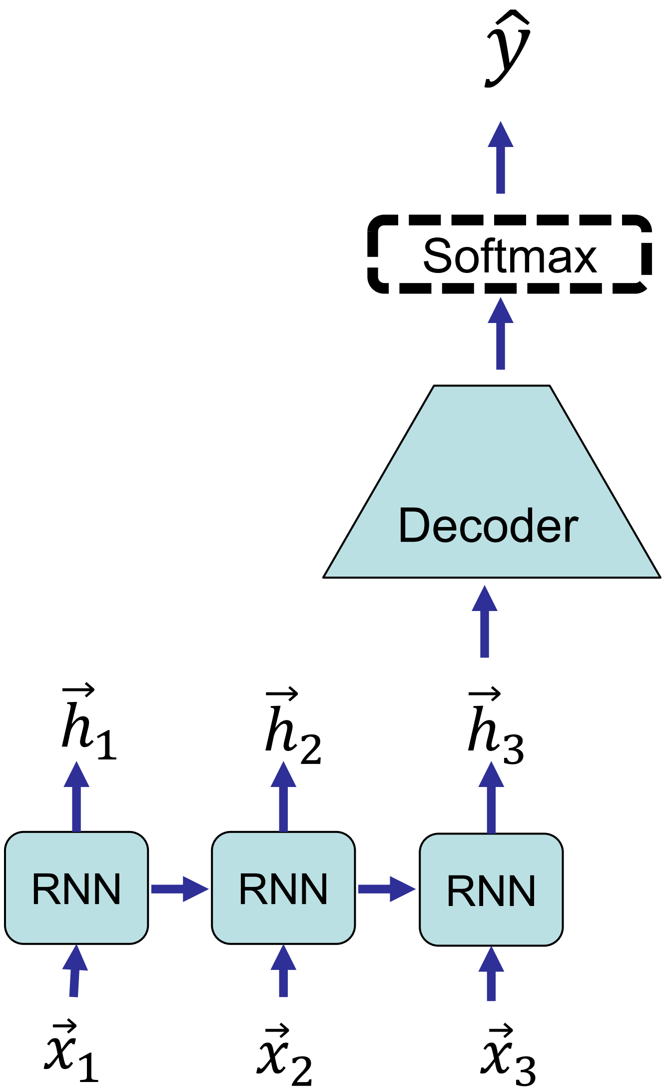
Document Classification with DL

RNNClassModel (practical session)

$\vec{x}$ word embedding

$\hat{y}$ probability distribution of predicted output

Decoder: linear projection to output classes



Document Classification with DL

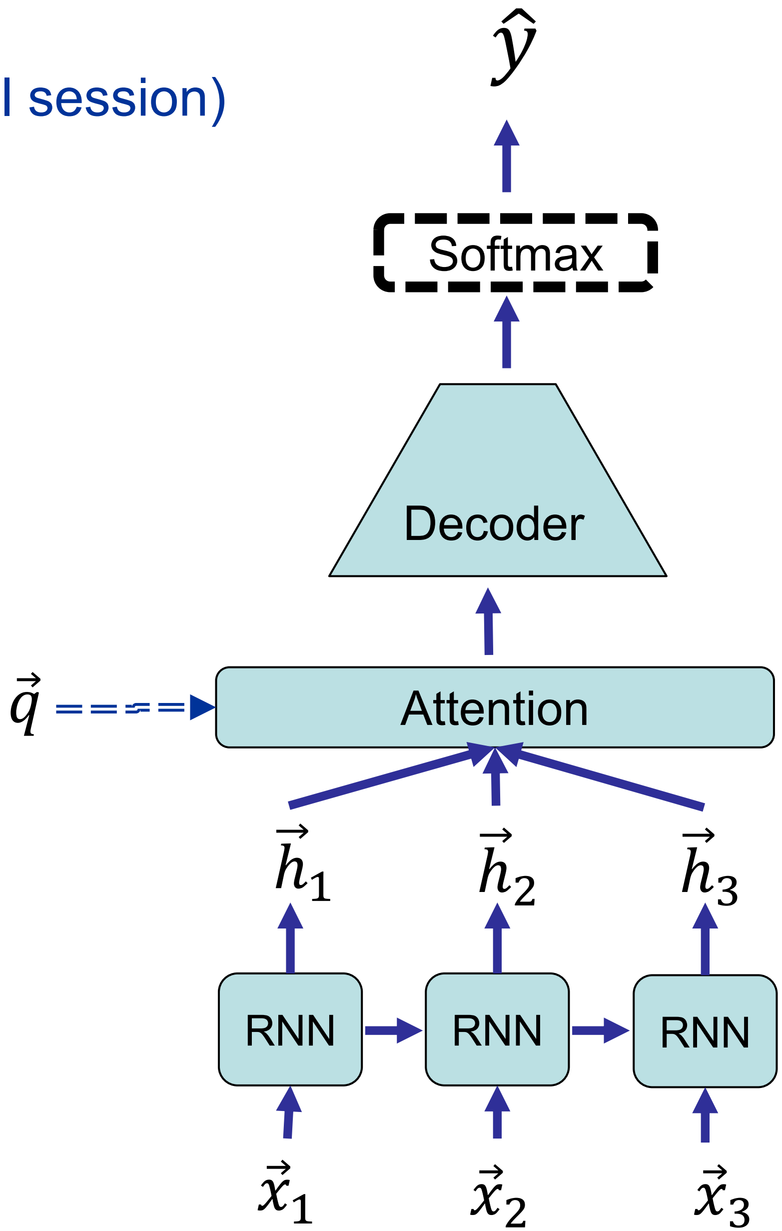
ATTClassModel (practical session)

$\vec{x}$ word embedding

$\hat{y}$ probability distribution of predicted output

Decoder: linear projection to output classes

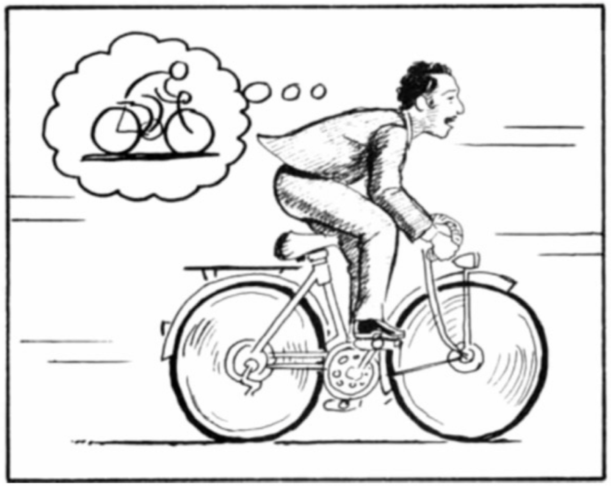
$\vec{q}$ attention query on words: *which word is informative?*



Challenges

- Semantics of language and the world

"The image of the world around us, which we carry in our head, is just a model. Nobody in his[/her] head imagines all the world, government or country. He[/She] has only selected concepts, and relationships between them, and uses those to represent the real system." Mental Model [7]



[8]

- Representation Learning
  - Abstract representation of spatial and temporal aspects of information
  - Various granularities → abstraction level
  - Task-specific, domain specific → transfer learning
  - Commonalities among languages → multilingual models

Challenges

- Understanding information contents
  - Information Retrieval
  - Summarization
  - Question/Answering

- Aspects of information tailoring and provision
  - Personalization vs. De-personalization
  - Controversy detection
  - Multiple points of view

Challenges

- Exploring aspects of society
  - Computational Social Science with NLP
    - Economy
    - Sociology
    - Psychology
- Ethics, fairness, and transparency
  - implications of the new technology on the society
  - Ownership of data and models, laws, etc.
  - Ethical bias in data and algorithms
  - Interpretability of the models

References

[1] Jurafsky, Dan, and James H. Martin. *Speech and language processing*. Vol. 3. London: Pearson, 2014.
[2] Forrester, Jay Wright. Counterintuitive behavior of so- cial systems, 1971. URL https://en.wikipedia. org/wiki/Mental_model. [Online; accessed 01- Nov-2017].
[3] Ha, David, and Jürgen Schmidhuber. "World Models." arXiv preprint arXiv:1803.10122 (2018)