

Đồ án môn học

Học Máy

Phân loại văn bản tin tức

Phan Ngọc Lân
Nguyễn Duy Mạnh

Giảng viên hướng dẫn: TS. Nguyễn Nhật Quang

Mở đầu

- Bài toán: Phân loại các văn bản tin tức vào các chủ đề có sẵn.
- Một văn bản gồm tiêu đề và nội dung.
- VD: Chủ đề = (Sports, World, Business, Sci/Tech)

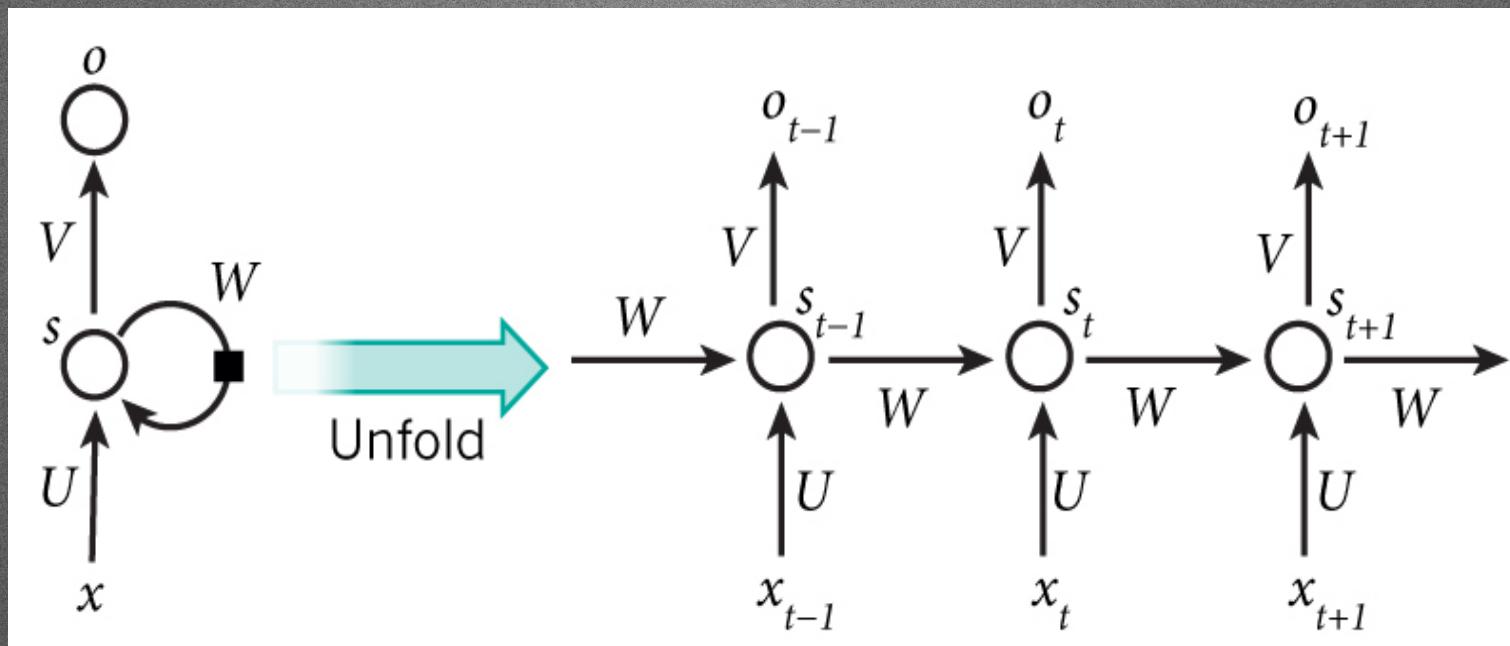
Input	Output
<p>Tiêu đề : Fears for T N pension after talks</p> <p>Nội dung : Unions representing workers at Turner Newall say they are 'disappointed' after talks with stricken parent firm Federal Mogul.</p>	Business

Giải quyết

- Không sử dụng học máy: sử dụng các bộ phân tích (parser), các kỹ thuật xử lý ngôn ngữ tự nhiên.
- Sử dụng học máy:
 - Naive Bayes
 - SVM
 - Mạng neuron
- Đồ án sử dụng mạng neuron hồi quy (Recurrent Neural Network).

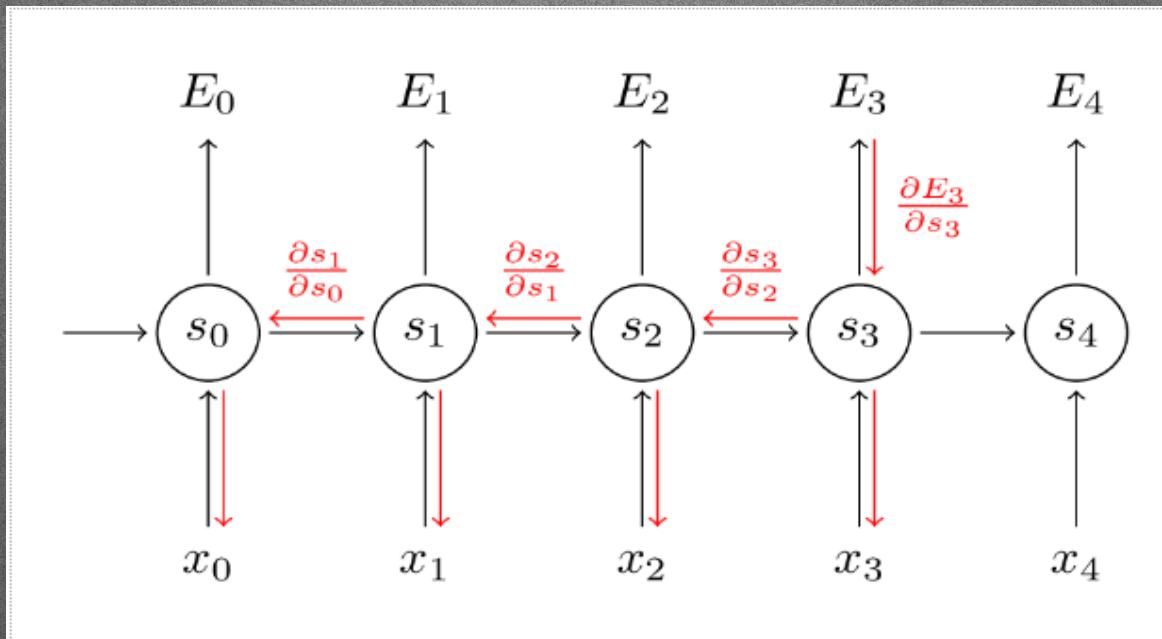
Recurrent Neural Network (RNN)

- Cách tiếp cận: Coi input là 1 chuỗi các bước (time-step) có liên hệ với nhau. Mỗi bước lần lượt đóng góp vào output.



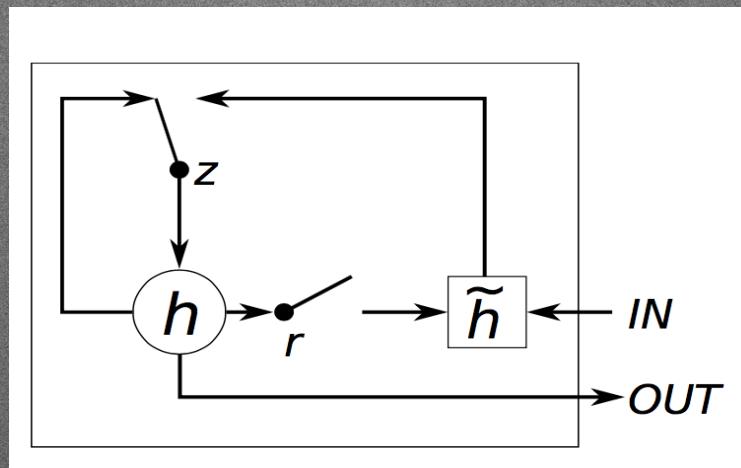
Back-propagation Through Time (BPTT)

- Một phiên bản chỉnh sửa của thuật toán lan truyền ngược.
- “Trải” mỗi neuron RNN thành 1 mạng con, sau đó áp dụng quy tắc chuỗi đạo hàm.



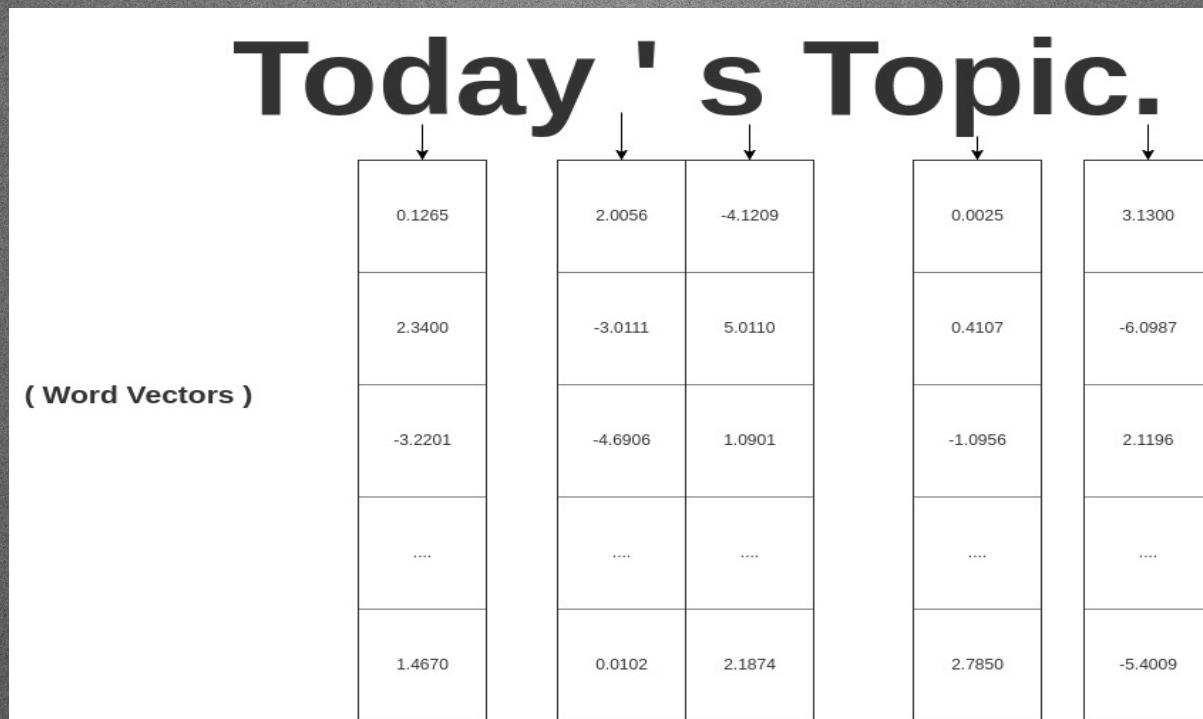
Gated Recurrent Unit (GRU)

- Một trong 2 biến thể phổ biến của RNN. Được đề xuất năm 2014.
- Sử dụng 2 “cổng” update và reset để chọn lọc trạng thái (ngữ cảnh) được lưu giữ qua mỗi time-step.



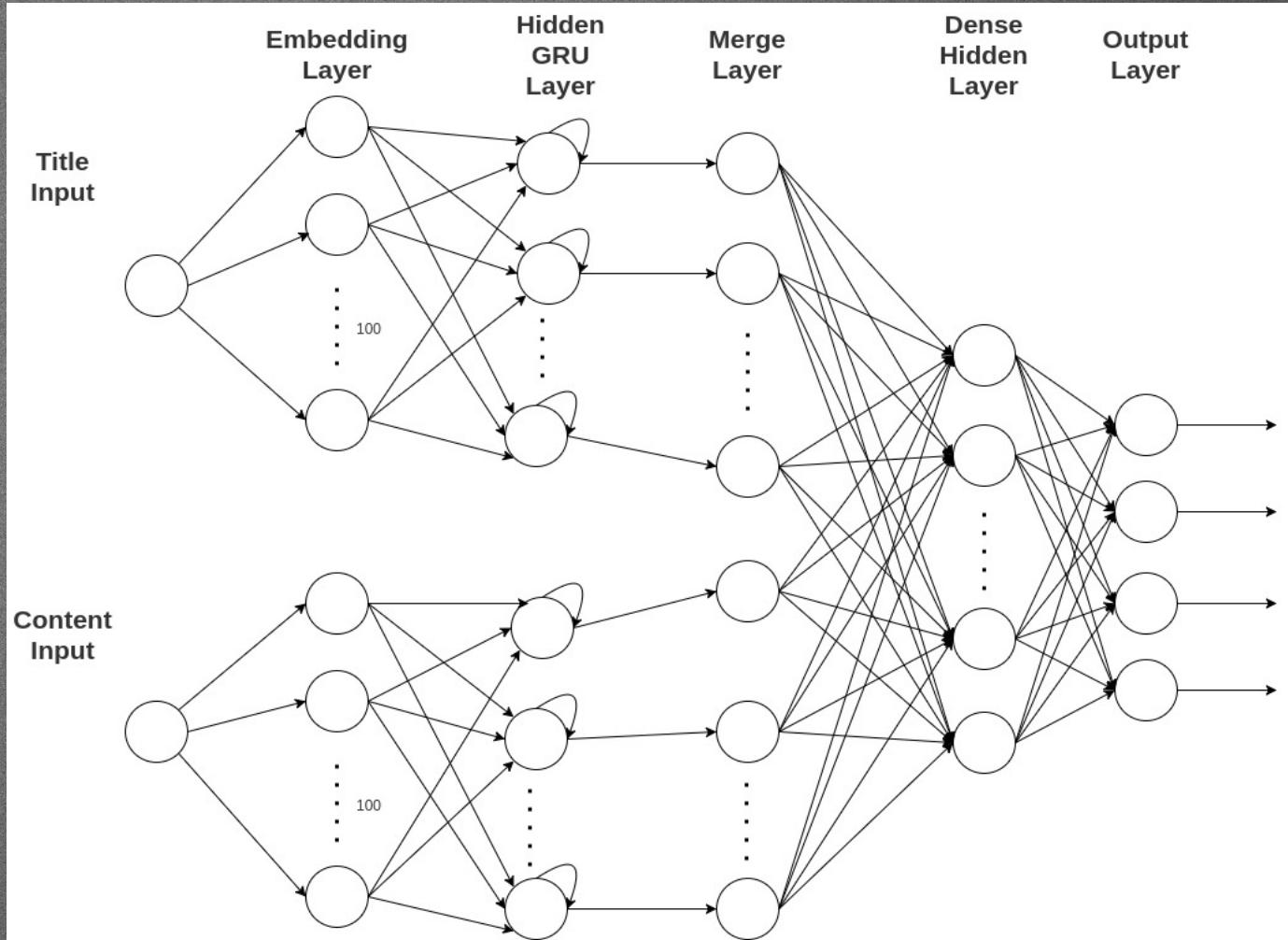
Biểu diễn ví dụ

- Mỗi ví dụ là 1 chuỗi các vector từ.
- Các từ nằm ngoài từ điển được thay thế bằng UNKNOWN_TOKEN. Phần sau câu được chèn các MASK_TOKEN để có cùng độ dài.



Kiến trúc mạng

- Sử dụng 2 lớp GRU độc lập cho tiêu đề và nội dung.
- Gộp lại thành 1 vector thể hiện ngữ nghĩa.
- 1 số lớp mạng liên kết đầy đủ chuyển ngữ nghĩa thành phân loại.



Các tập dữ liệu

- Đồ án sử dụng 3 tập dữ liệu:
 - Ag-News: 4 chủ đề; 89,320 ví dụ train; 38,280 ví dụ test.
 - BBC: 5 chủ đề; 1,777 ví dụ train; 447 ví dụ test.
 - Reuters: 86 chủ đề; 7,769 ví dụ train; 3,018 ví dụ test.
- Các tập validation được tách từ tập train tuân theo stratified sampling.

Cài đặt (1)

- Sử dụng Python và Keras - một thư viện Deep Learning mã nguồn mở.
- Keras:
 - Coi mạng neuron là 1 đối tượng chứa các tầng (layer) có thể được lắp ghép tùy ý.
 - Cung cấp các loại neuron phổ biến dưới dạng các lớp, cho phép chỉnh sửa qua kế thừa.
 - Sử dụng nền tảng tính toán biểu tượng TensorFlow/Theano.
 - Hỗ trợ sử dụng GPU để tăng tốc độ tính toán.

Cài đặt (2)

- Mạng neuron của hệ thống được chứa trong lớp Classifier:
 - Lưu trữ toàn bộ các thông tin cần thiết để tiến hành phân loại: từ điển, tên các lớp, mô hình mạng,...
 - Cung cấp các phương thức thực hiện phân loại, đánh giá, huấn luyện,...

Cài đặt (3)

- Các cấu hình của hệ thống được tùy chỉnh trong file cấu hình settings.py
- Hệ thống được chạy nhiều lần với số neuron khác nhau. Kết quả cuối cùng được lấy từ lần huấn luyện có độ chính xác validation cao nhất.

```
LEARNING_RATE = 0.001
N_EPOCH = 200
BATCH_SIZE = 100
VOCABULARY_SIZE = 40000
TITLE_LEN = 30
CONTENT_LEN = 50
TITLE_OUTPUT = 200
CONTENT_OUTPUT = 400
DENSE_NEURONS = [125]
DATASET = 'ag_news'
```

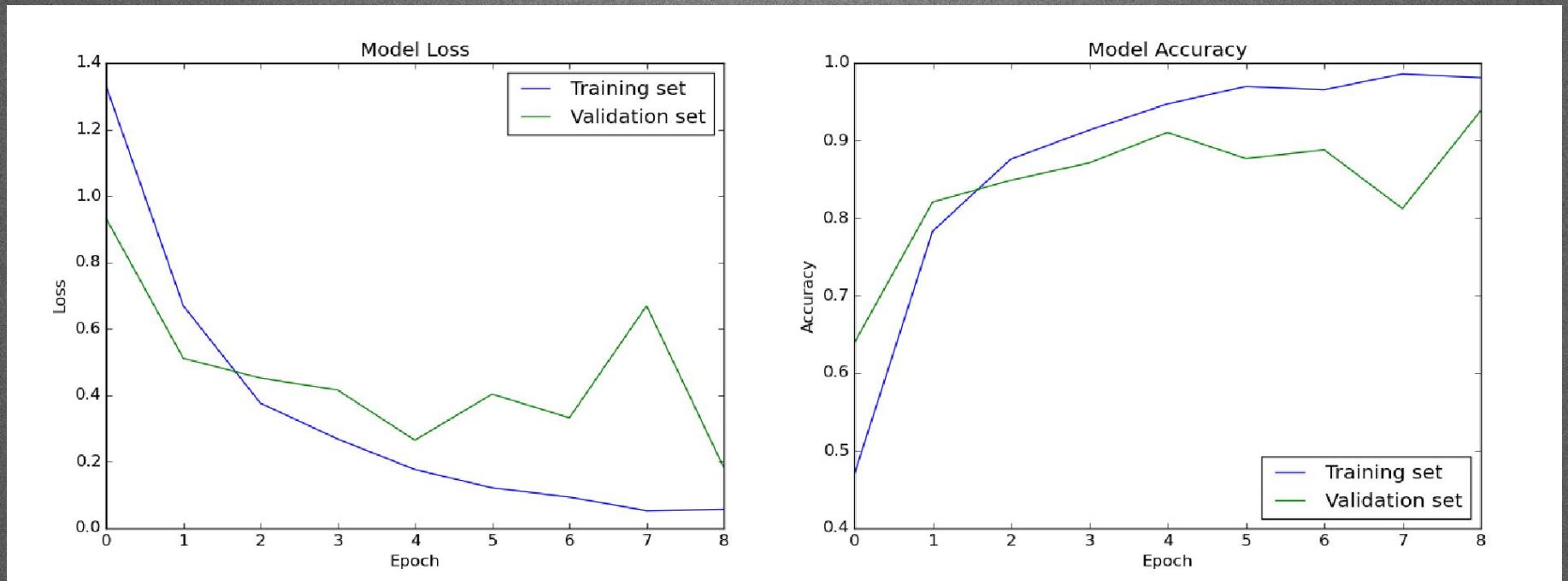
Kết quả (1)

- Kết quả tại lần validation tốt nhất:

	AG-News	BBC	Reuters
Số chủ đề	4	5	86
Số neuron tối ưu	125	250	250
Hàm mục tiêu (train)	0.1702	0.0569	0.2767
Hàm mục tiêu (val)	0.2108	0.1832	0.7872
Hàm mục tiêu (test)	0.2176	0.2324	0.8119
Độ chính xác (train)	0.9400	0.9809	0.9252
Độ chính xác (val)	0.9263	0.9383	0.8156
Độ chính xác (test)	0.9232	0.9284	0.8021
Precision (test)	0.9240	0.9277	0.5649
Recall (test)	0.9232	0.9279	0.2965
Điểm F1 (test)	0.9232	0.9275	0.4494

Kết quả (2)

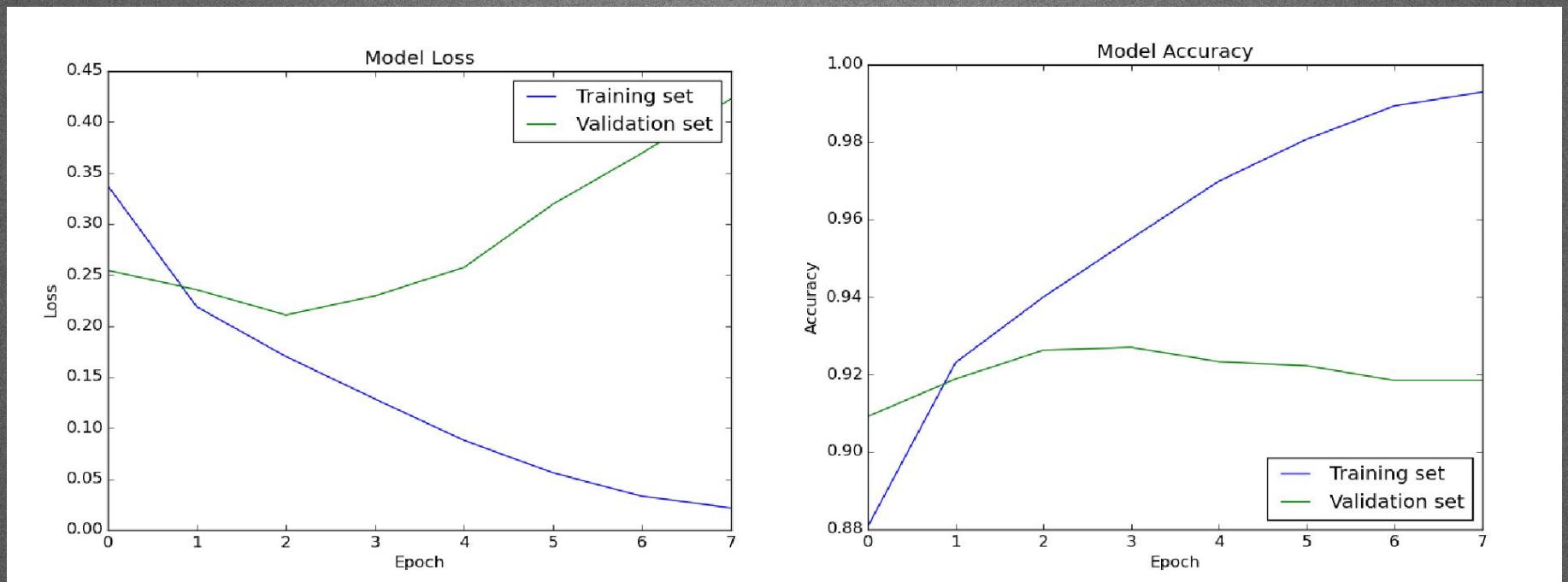
- Đồ thị hàm mục tiêu và độ chính xác trong quá trình huấn luyện:



Dataset BBC

Kết quả (3)

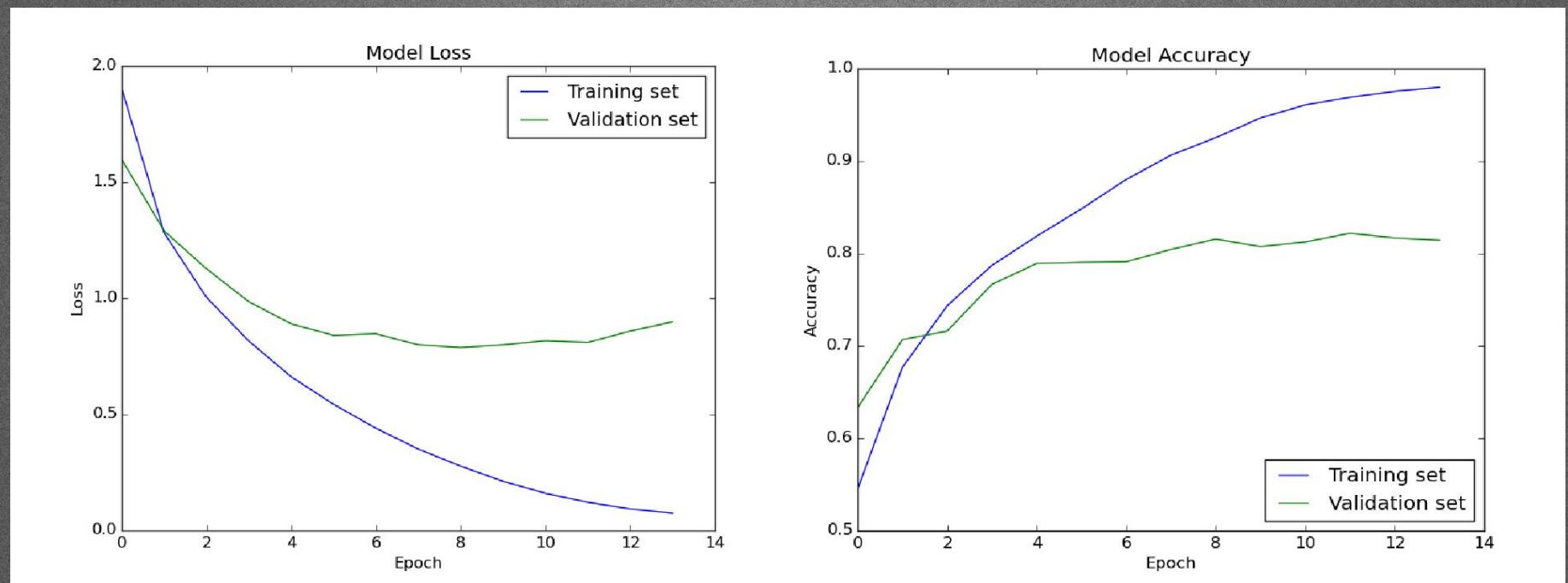
- Đồ thị hàm mục tiêu và độ chính xác trong quá trình huấn luyện:



Dataset Ag-News

Kết quả (4)

- Đồ thị hàm mục tiêu và độ chính xác trong quá trình huấn luyện:



Dataset Reuters

Demo

- Nhóm xây dựng 2 demo để sử dụng các mô hình đã huấn luyện:
 - Demo 1: Giao diện đồ họa cho phép phân loại 1 văn bản với tiêu đề và nội dung. Hiển thị chủ đề được dự đoán và xác suất của các chủ đề trên 1 đồ thị.
 - Demo 2: Tải về các bản tin từ feed RSS của 1 số trang báo và phân loại vào các chủ đề.

Khó khăn, tranh luận

- Lựa chọn giá trị cho vector các từ UNKNOWN_TOKEN và MASK_TOKEN.
- Lựa chọn sử dụng các kĩ thuật giảm overfit.
- Việc sử dụng Theano/TensorFlow phần nào cản trở việc debug và xác thực các kết quả trung gian do quá trình tối ưu hóa.

Kết luận

- Mạng neuron hồi quy có khả năng xử lý ngôn ngữ tự nhiên rất tốt, đặc biệt khi được kết hợp với biểu diễn ngữ nghĩa bằng vector.
- Biểu diễn từ bằng vector lưu giữ được nhiều thông tin hơn về văn bản nhưng đồng thời đưa thêm nhiều vào dữ liệu.
- Hệ thống có thể được mở rộng ra các loại văn bản khác, đặc biệt các văn bản ngắn hoặc trung bình.

Thanks for listening