Chapter 9: RNN and CNN (80-87)

80. Turning words into numeric IDs

5 titles

	title
0	UPDATE 1-Ousted American Apparel CEO Charney reports 43% stake
1	'Mad Men' premiere draws 2.3 mln, lowest season debut since 2008
2	E-cigarettes CAN help people kick the habit: Study finds they are 60% more ...
3	PRECIOUS-Gold rebounds on bargain hunting after two-day tumble
4	Why so shy? Lea Michele keeps her head down after claims her new boyfriend ...
5	Europe Stocks Rise With Emerging Markets as Bonds Decline
6	"Angelina Jolie: Honorary Damehood Means ""A Great Deal To Me"""
7	The Momentum of Freedom (Passover)
8	Ebersman Departs Facebook on Top After Post-IPO Stock Revival
9	Former Anglo Irish Bank Executives Guilty on 10 Loan Charges

5 titles after converted to ID sequence

	title
0	6 7 8 9 10 11 12 13
1	14 15 16 17 18 19 20 21 22
2	23 24 25 26 27 3 28 29 30 31 32 33
3	34 35 1 36 37 2 38 39
4	40 41 42 43 44 45 4 46 47 2 48 4 49 50
5	51 5 52 53 54 55 56 57 58
6	59 60 61 62 63 64 65 66 67 68
7	3 69 70 71 72
8	73 74 75 1 76 2 77 5 78
9	79 80 81 82 83 84 1 85 86

81. Prediction with an RNN

Conversion titles to onehot vectors

```
types: words which appear more than 2 times are 10159 types
```

Train data and validation data converted to ids

X_train:


```
[
  0 : [
    0 : 10151
    1 : 8393
    2 : 10025
    3 : 9544
```

```
    4 : 10086
    5 : 8811
    6 : 10099
    7 : 9675
  ]
  1 : [
    0 : 9365
    1 : 9847
    2 : 9749
    3 : 8394
    4 : 9876
    5 : 9226
    6 : 9924
    7 : 9829
    8 : 9966
  ]
  2 : [
    0 : 9227
    1 : 10024
    2 : 9988
    3 : 9942
    4 : 8949
    5 : 10156
    6 : 4002
    7 : 10003
    8 : 9891
    9 : 9789
    10 : 10111
    11 : 10125
  ]
]
```


y_train:


```
[
  0 : 0
  1 : 2
  2 : 3
]
```


Data converted to onehot vectors

X_train_onehot `tf.Tensor( [[0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.] ... [1. 0. 0. ... 0. 0. 0.] [1. 0. 0. ... 0. 0. 0.] [1. 0. 0. ... 0. 0. 0.]], shape=(18, 10159), dtype=float32)`

X_valid_onehot `tf.Tensor( [[0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 1. 0.] [0. 0. 0. ... 1. 0. 0.] ... [1. 0. 0. ... 0. 0. 0.] [1. 0. 0. ... 0. 0. 0.] [1. 0. 0. ... 0. 0. 0.]], shape=(18, 10159), dtype=float32)`

Prediction before training

0	1	2	3
0.2384	0.2523	0.2306	0.2787
0.2283	0.2564	0.2327	0.2825
0.2344	0.2774	0.2268	0.2613

```
[
  0 : 0
  1 : 2
  2 : 3
]
```


82. Training with Stochastic Gradient Descent

Prediction after training!

0	1	2	3
0.9987	0.0011	0.0002	0.0001
0.0006	0.0001	0.999	0.0002
0.0061	0.0005	0.0002	0.9933


```
[
  0 : 0
  1 : 2
  2 : 3
]
```

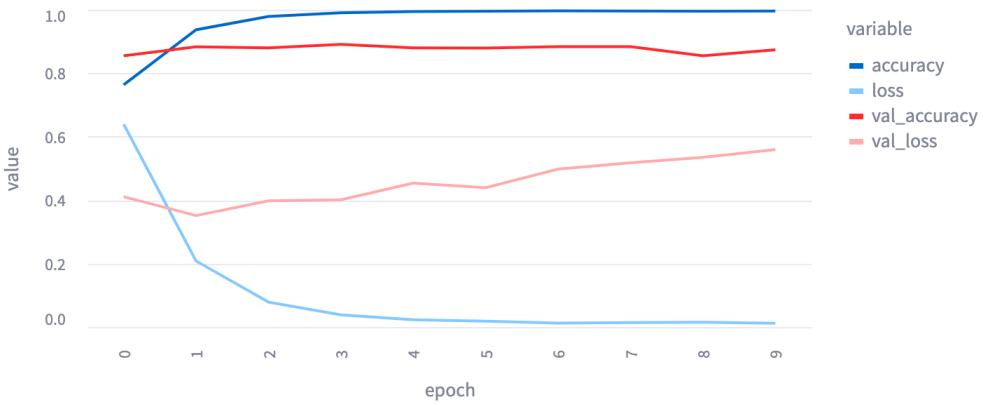

```
training time:  324.2119 seconds!
```

Loss and accuracy

history.history:


```
{
  "loss" : [
    0 : 0.6394266486167908
    1 : 0.20823301374912262
    2 : 0.07831086963415146
    3 : 0.03824350982904434
    4 : 0.022794194519519806
    5 : 0.01818971149623394
    6 : 0.012207314372062683
    7 : 0.013635183684527874
    8 : 0.015166442841291428
    9 : 0.01180181559175253
  ]
  "accuracy" : [
    0 : 0.7635869383811951
    1 : 0.9369378089904785
    2 : 0.9789167642593384
    3 : 0.9904422760009766
    4 : 0.9944714903831482
    5 : 0.9957833290100098
    6 : 0.9966266751289368
    7 : 0.9959707856178284
    8 : 0.9953148365020752
    9 : 0.9959707856178284
  ]
  "val_loss" : [
    0 : 0.41055119037628174
```

```
    1 : 0.35124966502189636
    2 : 0.39772823452949524
    3 : 0.40083709359169006
    4 : 0.4533116817474365
    5 : 0.4389643371105194
    6 : 0.49756357073783875
    7 : 0.5176900625228882
    8 : 0.5348080992698669
    9 : 0.5590986013412476
  ]
  "val_accuracy" : [
    0 : 0.8553223609924316
    1 : 0.883058488368988
    2 : 0.8800599575042725
    3 : 0.8913043737411499
    4 : 0.8800599575042725
    5 : 0.8793103694915771
    6 : 0.8838080763816833
    7 : 0.8838080763816833
    8 : 0.8553223609924316
    9 : 0.8740629553794861
  ]
}
```

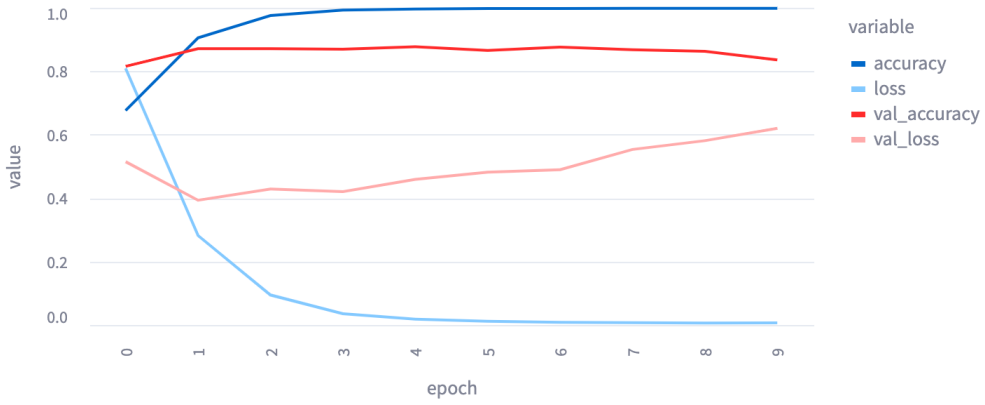



Using sgd as optimizer

0	1	2	3
0.9999	0	0	0
0.0023	0	0.9974	0.0003
0	0.0207	0.0021	0.9771

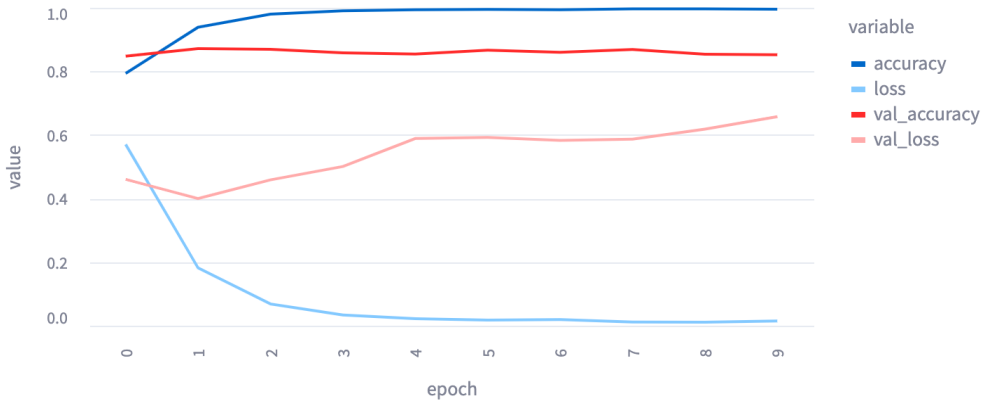

```
[
  0 : 0
  1 : 2
  2 : 3
]
```


```
training time:  282.9564 seconds!
```

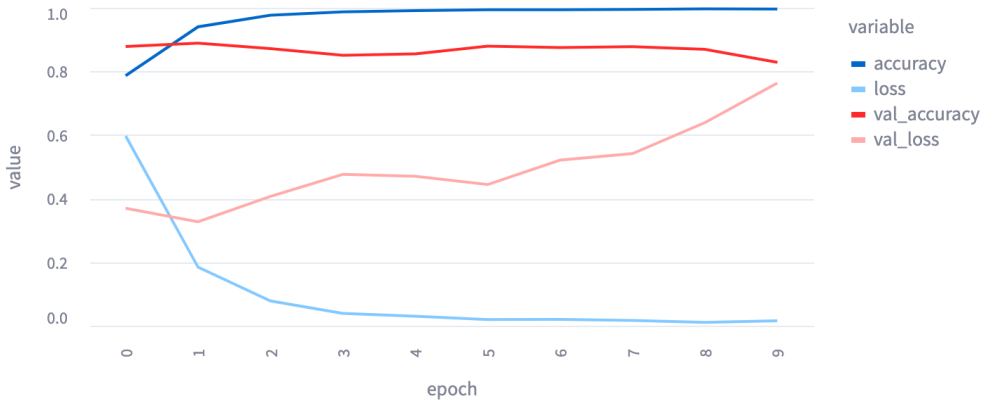


83. Mini-batch Training, GPU Training

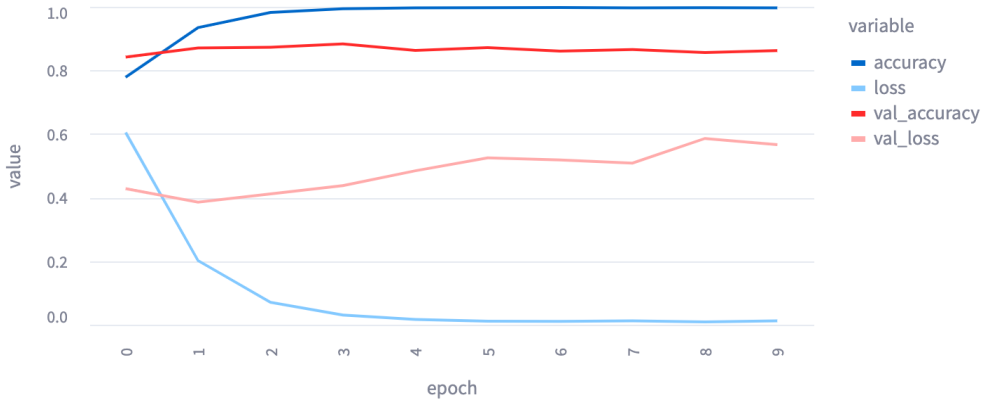
```
batch_size: 4, training time:  855.2132 seconds!
```



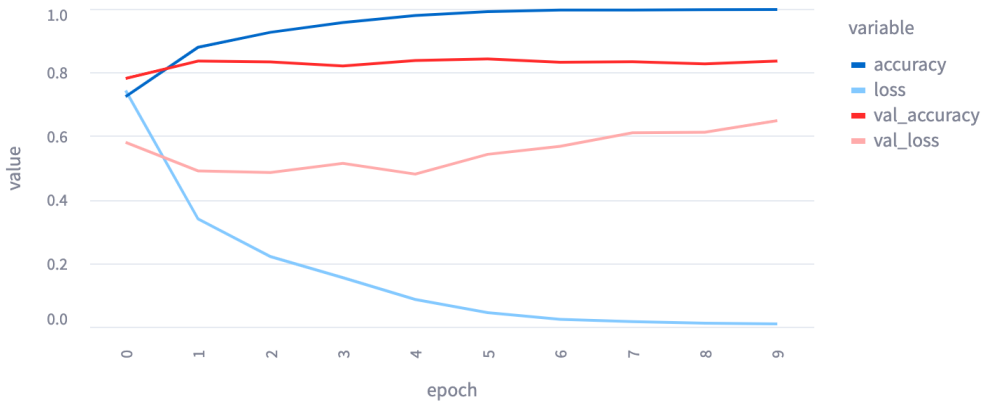

```
batch_size: 8, training time:  538.5973 seconds!
```




```
batch_size: 32, training time:  275.3736 seconds!
```



```
batch_size: 64, training time:  256.6548 seconds!
```



84. Add Pretrained Word Embeddings

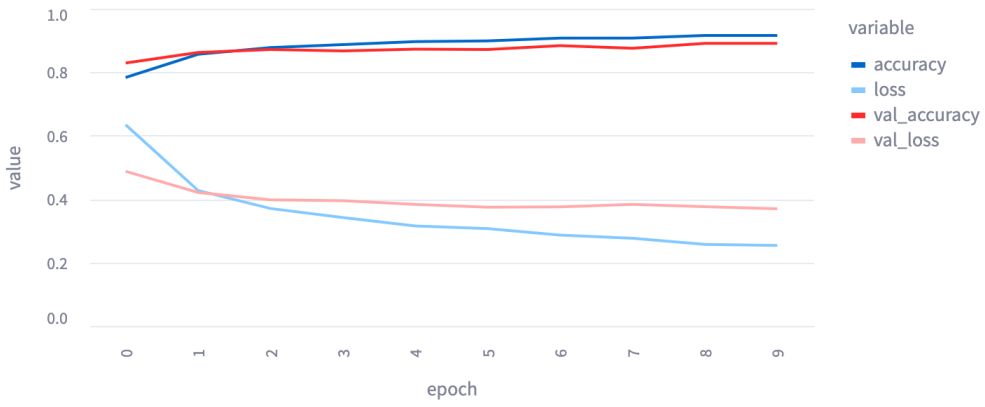
Word embeddings

X_train:

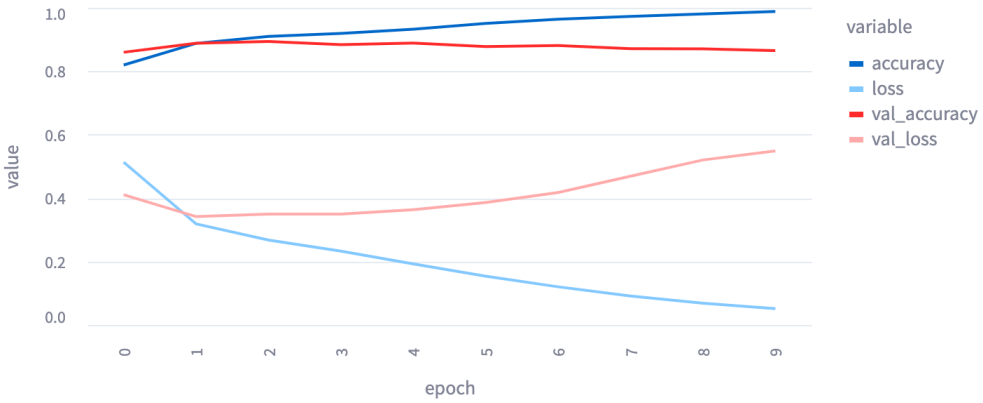
0	1	2	3	4	5	6	7	8	9	10	11
0.0554	-0.0566	0.1504	0.2402	-0.2061	-0.0449	0.1865	-0.3008	0.1631	0.0369	0.0086	-0.184
-0.0479	0.167	0.0781	0.0021	0.2256	-0.2256	-0.21	-0.0374	-0.0186	0.1943	-0.03	-0.177
-0.008	-0.1357	-0.0713	0.1357	-0.0457	0.0767	-0.2793	-0.3223	0.3242	0.2832	0.2314	-0.010
0.1035	0.1279	0.1465	-0.0937	-0.0442	0.0527	-0.1973	-0.0067	-0.1455	0.1621	-0.0203	0.083
0.0305	0.2266	-0.2578	0.2266	-0.1514	-0.2178	0.2734	0.0126	0.2207	-0.2988	-0.4238	0.285
0.0223	-0.5195	0.1465	0.0197	0.2119	-0.0588	-0.2773	-0.2373	0.6563	0.1216	0.0732	-0.121
0.0598	0.1641	-0.1592	0.0615	0.1973	-0.0537	0.0767	-0.0601	0.1455	0.4316	0.0349	-0.08
0.0898	0.0043	-0.3691	-0.0559	0.1709	-0.165	-0.373	-0.4082	0.3008	0.3379	-0.033	-0.140
-0.0064	0.032	0.0129	0.0203	0.0094	-0.0864	-0.2021	-0.1494	0.2715	0.0962	-0.053	-0.241
0	0	0	0	0	0	0	0	0	0	0	0

Training

```
training time:  19.8156 seconds!
```



85. Bi-directional RNN and Multi-layer RNN



86. Convolutional Neural Networks (CNN)

Prediction before training

0	1	2	3
0.2502	0.2459	0.2502	0.2537
0.2505	0.2457	0.2526	0.2511
0.2546	0.2483	0.2468	0.2503


```
[
  0 : 0
  1 : 2
  2 : 3
]
```


87. CNN Learning via Stochastic Gradient Descent

Prediction after training

0	1	2	3
0.9999	0.0001	0	0
0.0094	0	0.9901	0.0005
0.0011	0.0038	0.0049	0.9902


```
[
  0 : 0
  1 : 2
  2 : 3
]
```


```
training time:  206.9427 seconds!
```