


Neural Networks in a Nutshell

1-0 Prelude

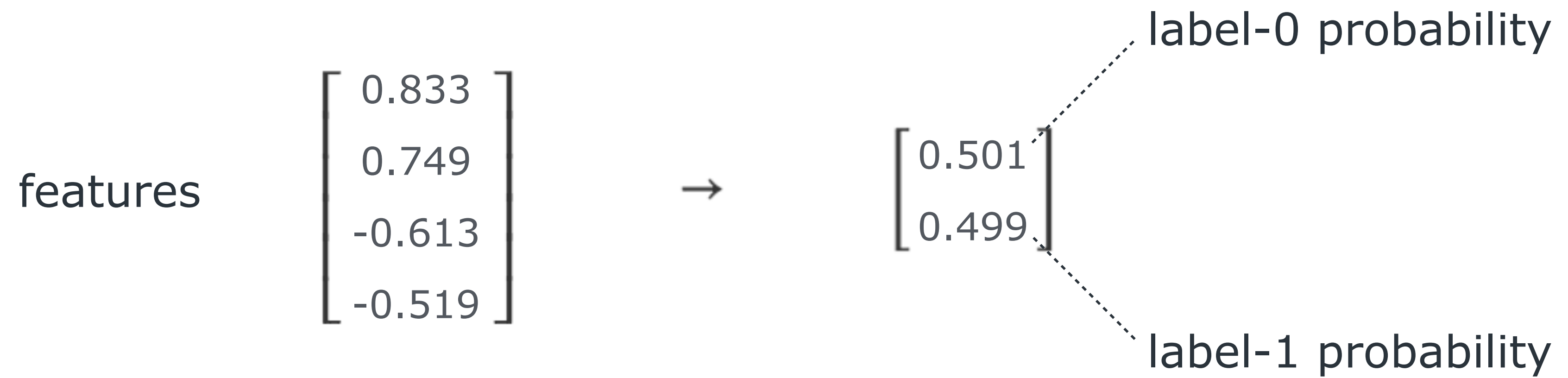
label

features



	Survived	Class	Sex	Age	Fare
0	0	3	1	22	7.2500
1	1	1	0	38	71.2833
2	1	3	0	26	7.9250
3	1	1	0	35	53.1000
4	0	3	1	35	8.0500

1-0 Prelude



1-0 Prelude

$$\begin{array}{c} \mathbf{w} \\ \left[\begin{array}{cccccc} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{array} \right] \end{array} \begin{bmatrix} 0.833 \\ 0.749 \\ -0.613 \\ -0.519 \end{bmatrix} + \begin{array}{c} \mathbf{b} \\ \left[\begin{array}{c} \cdot \\ \cdot \end{array} \right] \end{array} \rightarrow \begin{bmatrix} 0.007 \\ 0.001 \end{bmatrix} \rightarrow \begin{bmatrix} 0.501 \\ 0.499 \end{bmatrix}$$

normalize

1-1 Basic NN

Problem: Random guessing not effective

Solution: Reduce loss in systematic way via SGD

2-2 Regularized NN

Problem: Overfit to training data

Solution: Make network 'work harder' via Dropout

2-3 CNN

Problem: Not using 2D structure of image data

Solution: Take 2D 'snapshots' via CNNs

2-3 CNN - Convolutional layer

0.3	0.2	0.3	0.5	0.3	0.5
0.3	1.0	0.7	0.1	0.4	0.6
0.1	1.0	0.4	0.8	0.8	0.4
0.4	0.6	0.8	0.2	0.3	0.7
0.5	0.2	0.4	0.2	0.5	0.8
0.2	1.0	0.4	0.5	0.5	0.0

.

filter

0.7	1.0	0.2
0.1	0.4	0.7
0.0	0.4	0.1



1.8			

2-3 CNN - Convolutional layer

0.3	0.2	0.3	0.5	0.3	0.5
0.3	1.0	0.7	0.1	0.4	0.6
0.1	1.0	0.4	0.8	0.8	0.4
0.4	0.6	0.8	0.2	0.3	0.7
0.5	0.2	0.4	0.2	0.5	0.8
0.2	1.0	0.4	0.5	0.5	0.0

filter

0.7	1.0	0.2
0.1	0.4	0.7
0.0	0.4	0.1

.



1.8	1.2		

2-3 CNN - Convolutional layer

0.3	0.2	0.3	0.5	0.3	0.5
0.3	1.0	0.7	0.1	0.4	0.6
0.1	1.0	0.4	0.8	0.8	0.4
0.4	0.6	0.8	0.2	0.3	0.7
0.5	0.2	0.4	0.2	0.5	0.8
0.2	1.0	0.4	0.5	0.5	0.0

.

filter

0.7	1.0	0.2
0.1	0.4	0.7
0.0	0.4	0.1



1.8	1.2	1.5	

2-3 CNN - Convolutional layer

0.3	0.2	0.3	0.5	0.3	0.5
0.3	1.0	0.7	0.1	0.4	0.6
0.1	1.0	0.4	0.8	0.8	0.4
0.4	0.6	0.8	0.2	0.3	0.7
0.5	0.2	0.4	0.2	0.5	0.8
0.2	1.0	0.4	0.5	0.5	0.0

.

filter

0.7	1.0	0.2
0.1	0.4	0.7
0.0	0.4	0.1



1.8	1.2	1.5	1.7
2.3	2.5	1.7	1.4
2.1	1.9	1.7	2.3
1.8	1.7	1.5	1.5

2-3 CNN - Pooling layer

1.8	1.2	1.5	1.7
2.3	2.5	1.7	1.4
2.1	1.9	1.7	2.3
1.8	1.7	1.5	1.5

max
→

2.5	

3-3 RNN

Problem: Not using sequence structure of text data

Solution: Train model sequentially via RNNs

3-3 RNN

"A series of escapades demonstrating ..."

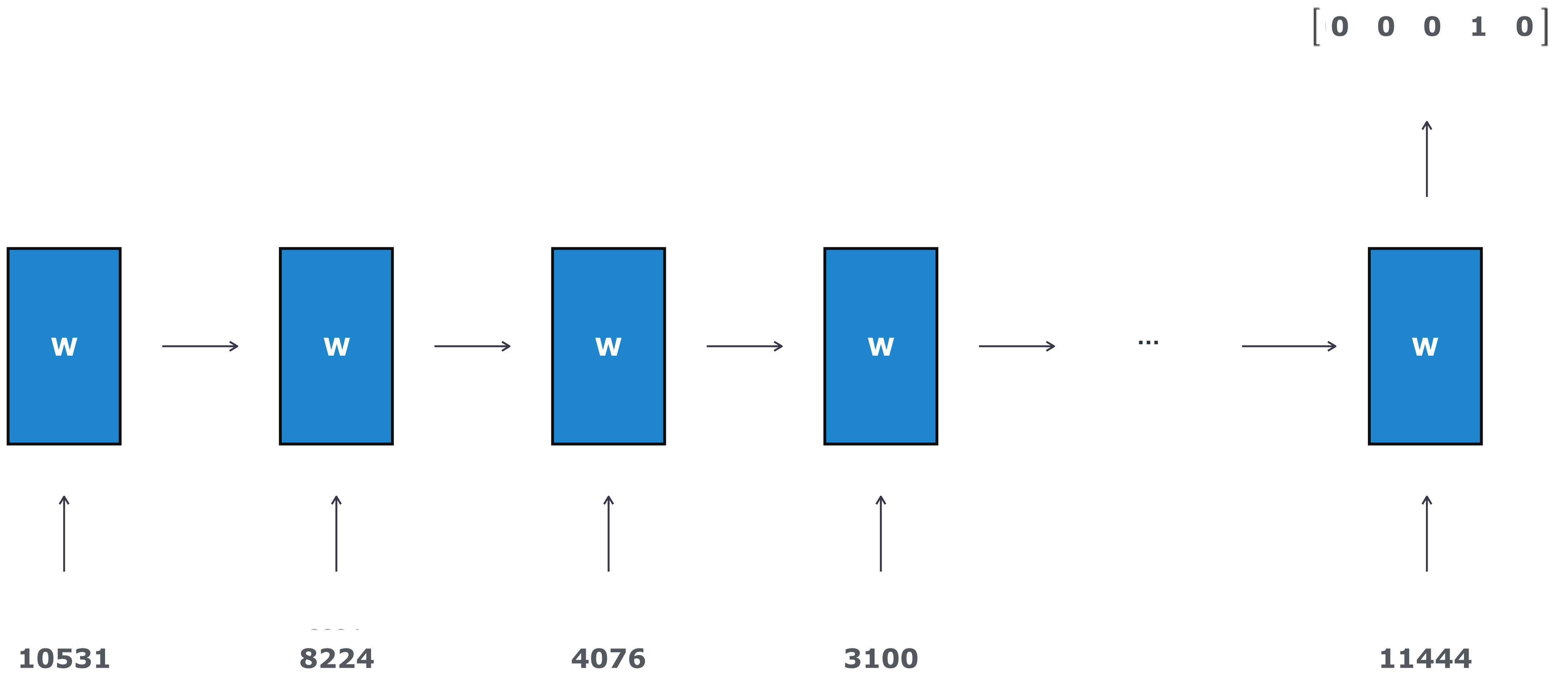
↓ tokenize

[*'series'*, *'of'*, *'escapades'*, *'demonstrating'*, ...]

↓ map to index

[10531, 8224, 4076, 3100, ...]

3-3 RNN



Questions?