

모두의 딥러닝 개정 2판 정오표 (최종 수정 : 2022년 3월 23일)

8쇄	페이지	수정 전	수정 후
	292	상단 코드 내 주석	(삭제)
	121	최대치가 0.3	최대치가 0.25
	99	만약 여기에 입력값이 추가되어 세 개 이상의 입력	만약 여기에 출력값이 추가되어 세 개 이상의 출력
	68	y_i 는 x_i 가 대입되었을 때 직선의 방정식 (여기서는 $p=3x+76$)	\hat{y}_i 는 x_i 가 대입되었을 때 직선의 방정식 (여기서는 $y=3x+76$)
	77	여기서 y_i 은 x_i 를 집어 넣었을 때의 값이므로	여기서 \hat{y}_i 는 x_i 를 집어 넣었을 때의 값이므로

7쇄	페이지	수정 전	수정 후
	331	가중합3 = $w_{31}y_{h1}+w_{31}y_{h2}=1$ (바이어스)	가중합3 = $w_{31}y_{h1}+w_{32}y_{h2}=1$ (바이어스)
	332	$w_{31}y_{h2}$ 과 바이어스 (분모) 가중합	$w_{32}y_{h2}$ 와 바이어스 (분모) 가중합3

6쇄	페이지	수정 전	수정 후
	25	pip install jupyter를 입력해 주피터 노트북을 설치합니다.	conda activate py37을 입력해 py37 작업환경을 열어줍니다. 그리고 pip install jupyter를 입력해 주피터 노트북을 설치합니다.
	72	print("MSE 최종값: " + str(mse_val(predict_result,y)))	print("MSE 최종값: " + str(mse_val(y,predict_result)))
	154	from tensorflow.keras.utils import np_utils	import tensorflow as tf
	193	y_acc=history.history['acc']	y_acc=history.history['accuracy']
	284	generator.add(Conv2D(128, kernel_size=3, ...))	generator.add(Conv2D(64, kernel_size=5, ...))
	284	커널 크기를 3으로 해서 3x3 크기의	커널 크기를 5로 해서 5x5 크기의

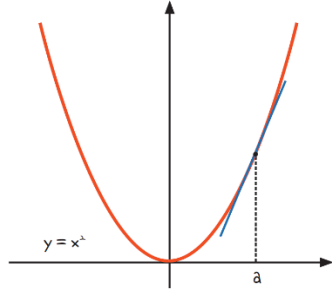
4쇄	페이지	수정 전	수정 후
	38	epochs=30	epochs=100
	327	w_{31} 값은 이미 알고 있으므로	삭제
	336	밑에서 8째줄, 7째줄, 3째줄 $(\delta y_{o1} \cdot y_{o1} + \delta y_{o2} \cdot y_{o2}) y_{h1} (1 - y_{h1}) \cdot x_1$	$(\delta y_{o1} \cdot w_{31} + \delta y_{o2} \cdot w_{41}) y_{h1} (1 - y_{h1}) \cdot x_1$

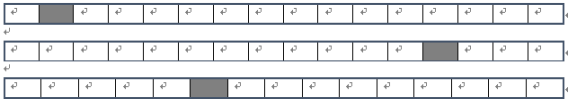
1~3쇄 공통	페이지	수정 전	수정 후
		1. 2020년 8월부로 아나콘다가 업그레이드 되었고, 2020년 3월 부로 구글 코랩이 업그레이드 되었습니다. 2. 현재 1~3쇄를 보고 계시다면, 길벗 자료실에 첨부되어 있는 아래 자료를 따라 설치하셔야 합니다. 1-3쇄만_해당_22-26p_업데이트.pdf 1-3쇄만_해당_350-359p_업데이트.pdf	

3쇄	페이지	수정 전	수정 후
	83	$a1_diff = -(1/\text{len}(x_data)) \sim$ $a2_diff = -(1/\text{len}(x_data)) \sim$	$a1_diff = -(2/\text{len}(x_data)) \sim$ $a2_diff = -(2/\text{len}(x_data)) \sim$

84	$b_diff = -(1/\text{len}(x_data)) \sim$	$b_diff = -(2/\text{len}(x_data)) \sim$
85	lr = 0.05 $a1_diff = -(1/\text{len}(x_data)) \sim$ $a2_diff = -(1/\text{len}(x_data)) \sim$ $b_diff = -(1/\text{len}(x_data)) \sim$	lr = 0.02 $a1_diff = -(2/\text{len}(x_data)) \sim$ $a2_diff = -(2/\text{len}(x_data)) \sim$ $b_diff = -(2/\text{len}(x_data)) \sim$
331	(맨 아래식에서) 가중합3 = $w31y_{h1} + w41y_{h2} = 1$ (바이어스)	가중합3 = $w31y_{h1} + w31y_{h2} = 1$ (바이어스)
332	(7번째 줄) $w41y_{h2}$	$w31y_{h2}$
335	(중간) $(y_{01} - y_{ot})$	$(y_{01} - y_{t1})$
9	파이썬 3.7이상 아나콘다 파이썬버전 3.7 텐서플로 2.0	파이썬 3.7 텐서플로 2.0.0 케라스 2.3

2쇄

페이지	수정 전	수정 후
257, 258	<code>model.fit(padded_x, labels, epochs=20)</code>	<code>model.fit(padded_x, classes, epochs=20)</code>
31,35,38, 163	<code>loss='mean_squared_error'</code>	<code>loss='binary_crossentropy'</code>
43	→그림 2-4	
47	→그림 2-7 $y = ax^2$	$y = a^x$
72	<pre>def mse(y_hat, y): return((y_hat - y)**2).mean() def mse_val(predict_result, y): return mse(np.array(predict_result), np.array(y)) ... print("mse 최종값: " + str(mse_val(predict_result,y)))</pre>	<pre>def mse(y, y_hat): return((y - y_hat)**2).mean() def mse_val(y,predict_result): return mse(np.array(y), np.array(predict_result)) ... print("mse 최종값: " + str(mse_val(y,predict_result)))</pre>
78	$\frac{\partial}{\partial a} MSE(a, b)$	$\frac{\partial}{\partial a} MSE(a, b)$
78	$\frac{\partial}{\partial a} MSE(a, b)$	$\frac{\partial}{\partial b} MSE(a, b)$
78	$\frac{2}{n} (ax_i + b - y_i) [(ax_i + b - y_i)]'$	$\frac{2}{n} \sum (ax_i + b - y_i) [(ax_i + b - y_i)]'$
78, 80	$b_diff = -(1/\text{len}(x_data)) * \text{sum}(y_data - y_pred)$	$b_diff = -(1/\text{len}(x_data)) * \text{sum}(\text{error})$
79	lr = 0.05	lr = 0.03
80	$a_diff = -(1/\text{len}(x_data)) \sim$ $b_diff = -(1/\text{len}(x_data)) \sim$	$a_diff = -(2/\text{len}(x_data)) \sim$ $b_diff = -(2/\text{len}(x_data)) \sim$
235	그림 16-9, Y_hi에서 512개의 노드	128 개의 노드
235	→그림 16-9 두번째 드롭아웃(25%) 상자	드롭아웃(50%)

252	그림 17-1	
253	model.add(Embedding(16,4))	model.add(Embedding(16,4))
312	(10째 줄) model.add(Activation('sigmoid'))	model.add(Activation('softmax'))
329	$(y_{t1} - y_{o1})'$	$(y_{t2} - y_{o2})'$
330	그러면 가중합3을 y01에 대해~~	그러면 y01을 가중합3에 대해~~
331	$f(x) = x^2$	$f(x) = x^a$
355	from goolge.colab	from google.colab

1쇄

페이지	수정 전	수정 후
84, 85	b_new = -(1/len(x1_data)) * sum(y_data - y_pred)	b_diff = -(1/len(x1_data)) * sum(y_data - y_pred)
98	plt.scatter(x, y)	plt.scatter(x_data, y_data)
148	→6번째 줄 np.random.seed(3)	numpy .random.seed(3)
251	→상단 소스 코드 word_size = len(t.word_index)+1	word_size = len(token .word_index)+1