

모두의 딥러닝 개정 3판 정오표

쇄	페이지	수정 전	수정 후
1쇄	200	http://www.amstat.org/publications/jse/v19n3/decock/DataDocumentation.txt	http://jse.amstat.org/v19n3/decock/DataDocumentation.txt
2쇄	40	아나콘다에 이미	코랩에 이미
	83	$y_i = ax_i + b$ 를 대입하면	$\hat{y}_i = ax_i + b$ 를 대입하면
	88	<code>ax.scatter3D(x1, x2, y) ;</code>	; 삭제
	107	텐서플로에서 실행하는 다중 선형 회귀	텐서플로에서 실행하는 로지스틱 회귀
	142	주피터 노트북을 통해	코랩을 통해
	150	그림 11-5 'plasma'	'BMI'
	156	그림 12-2 정보 3 첫째줄 '4.0'	'1.4'
	192	학습셋 - '빨간색', 검증셋 - '파란색'	학습셋 - '파란색', 검증셋 - '빨간색'
	193	주피터 노트북에	코랩에
	194	검증셋의 오차(파란색)	학습셋의 오차(파란색)
	194	테스트셋의 오차(빨간색)	검증셋의 오차(빨간색)
	204	<code>plt.show() ;</code>	; 삭제
3쇄	39	model 클래스	models 클래스
	46	<code>model.compile(loss=binary_crossentropy</code>	<code>model.compile(loss='binary_crossentropy'</code>
	66	조금 전 구한 기울기 x 까지	조금 전 구한 기울기 a 까지
	69	최소 제곱근 공식	최소 제곱법 공식
	69	x 의 각 원소와 x 의 평균값들의 차를	x 의 각 원소와 x 의 평균값의 차를
	72	입력 데이터의 종류가 17개나 됩니다	입력 데이터의 종류가 16개나 됩니다
	75	(표 4-3 하단) 1 -5 3 3	-1 5 -3 -3
	76	$1 + (-5) + 3 + 3 = 2$	$-1 + 5 + (-3) + (-3) = -2$
	111	그림 7-2 아달라인	
	112	그림 7-2 로지스틱 회귀	
4쇄	156	속성을 보니 우리가 앞서 다루었던 것과	클래스를 보니 우리가 앞서 다루었던 것과
	189	<code>./data/model/all/</code> 폴더에 모델을 지정해 줍니다	<code>./data/model/all/</code> 폴더에 모델을 저장해 줍니다
	196	<code>model</code> 폴더에 <code>Ch14-4-bestmodel.hdf</code> 라는	<code>model</code> 폴더에 <code>Ch14-4-bestmodel.hdf5</code> 라는
	228	테스트셋의 과적합이 일어나기 전	학습셋의 과적합이 일어나기 전
	353	$\text{가중합}_3 = w_{31}y_{h1} + w_{31}y_{h2} + 1$	$\text{가중합}_3 = w_{31}y_{h1} + w_{32}y_{h2} + 1$
	97	<code>prediction = model.predict([hour])</code>	<code>input_data = tf.constant([[hour]])</code>
	107	#텐서플로 버전업으로 인한 수정	<code>prediction = model.predict(input_data)[0][0]</code>

5쇄	98	prediction = model.predict([[hour, private_class]]) #텐서플로 버전업으로 인한 수정	input_data = tf.constant([[hour, private_class]]) prediction = model.predict(input_data)[0][0]
	189 190 195 196 207 209 223 226 227 339 341에 나오는 모든 .hdf5	#케라스 버전업으로 인한 수정	모두 .keras 로 수정
	227	!pip install Attention	!pip install keras-self-attention
		model.add(Attention())	model.add(SeqSelfAttention(attention_activation='sigmoid'))
	278	from attention import Attention	from keras_self_attention import SeqSelfAttention
		model.add(Dense(1))	model.add(Flatten()) model.add(Dense(1))