Math for Deep Learning

What You Need to Know to Understand Neural Networks

by Ronald T. Kneusel

errata updated to print 3

<table>
<tr><th>Page</th><th>Error</th><th>Correction</th><th>Print corrected</th></tr>
<tr><td>5</td><td>. . . we can let NumPy choose the data type for us, or we can specify it explictly.</td><td>. . . we can let NumPy choose the data type for us, or we can specify it explicitly.</td><td>Pending</td></tr>
<tr><td>6</td><td><table><tr><th>NumPy Name</th><th>Equivalent C Type</th><th>Range</th></tr><tr><td>float32</td><td>float</td><td>$\pm [1.175 \times 10^{38}, 3.403 \times 10^{38}]$</td></tr><tr><td>uint8</td><td>unsigned char</td><td>$[0, 255 = 2^{2} - 1]$</td></tr></table></td><td><table><tr><th>NumPy Name</th><th>Equivalent C Type</th><th>Range</th></tr><tr><td>float32</td><td>float</td><td>$\pm [1.175 \times 10^{-38}, 3.403 \times 10^{38}]$</td></tr><tr><td>uint8</td><td>unsigned char</td><td>$[0, 255 = 2^{8} - 1]$</td></tr></table></td><td>Print 2</td></tr>
<tr><td>18</td><td>If there's no chance the something will happen, its probability is zero.</td><td>If there's no chance that something will happen, its probability is zero.</td><td>Pending</td></tr>
<tr><td>29</td><td><pre>a = np.random.randint(0,364)
b = np.random.randint(0,364)</pre>The code simulates 100,000 random pairs of people, where the random integer in [0, 364] represents the person's birthday.</td><td><pre>a = np.random.randint(0,365)
b = np.random.randint(0,365)</pre>The code simulates 100,000 random pairs of people, where the random integer in [0, 365] represents the person's birthday.</td><td>Pending</td></tr>
<tr><td>29</td><td><pre>b = np.random.randint(0,364,m)</pre></td><td><pre>b = np.random.randint(0,365,m)</pre></td><td>Pending</td></tr>
<tr><td>39</td><td>. . . which shows us correct way to compare conditional probabilities.</td><td>. . . which shows us the correct way to compare conditional probabilities.</td><td>Pending</td></tr>
<tr><td>39</td><td><pre>s = np.random.randint(0,50,3)</pre></td><td><pre>s = np.random.choice(50,3,replace=False)</pre></td><td>Pending</td></tr>
<tr><td>82</td><td>The top chart in Figure 4-4 shows the box plot for the three sets of exam scores in |exams.npy|.</td><td>The top chart in Figure 4-4 shows the box plot for the three sets of exam scores in exams.npy.</td><td>Pending</td></tr>
</table>

Page	Error	Correction	Print corrected
119	Equation replacement	$$\begin{aligned} \boldsymbol{a} \times \boldsymbol{b} &= \lVert\boldsymbol{a}\rVert \lVert\boldsymbol{b}\rVert \sin(\theta)\hat{\boldsymbol{n}} \\ &= (a_1 b_2 - a_2 b_1, a_2 b_0 - a_0 b_2, a_0 b_1 - a_1 b_0) \end{aligned} \quad (5.6)$$	Print 3
128	Equation replacement	$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} 11 \\ 12 \\ 13 \end{bmatrix} = \begin{bmatrix} 74 \\ 182 \\ 290 \end{bmatrix}$$	Print 3
131	for $n, m \in$ ▯$^+$ (positive integers) and where $\boldsymbol{A}$ is a square matrix.	for $n, m \in \mathbb{Z}^+$ (positive integers) and where $\boldsymbol{A}$ is a square matrix.	Pending
175	But $\boldsymbol{e}$ $\boldsymbol{x}$ $\mathbf{ln}$ $\boldsymbol{a}$ $= a^x$, so we have . . .	But $e^{x \ln a} = a^x$, so we have . . .	Print 3
183	For example, above, we saw that the partial derivative of $f(x, y) = \ldots$	For example, above, we saw that the partial derivative of $f(x, y, \boldsymbol{t}, \boldsymbol{z}) = \ldots$	Print 3
198	Equation replacement	$$\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_{00}}{\partial \boldsymbol{x}} & \frac{\partial f_{01}}{\partial \boldsymbol{x}} & \cdots & \frac{\partial f_{0,m-1}}{\partial \boldsymbol{x}} \\ \frac{\partial f_{10}}{\partial \boldsymbol{x}} & \frac{\partial f_{11}}{\partial \boldsymbol{x}} & \cdots & \frac{\partial f_{1,m-1}}{\partial \boldsymbol{x}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_{n-1,0}}{\partial \boldsymbol{x}} & \frac{\partial f_{n-1,1}}{\partial \boldsymbol{x}} & \cdots & \frac{\partial f_{n-1,m-1}}{\partial \boldsymbol{x}} \end{bmatrix}$$	Print 3
201	Assume f accepts an m-element input and returns an n-element vector output.	Assume $\boldsymbol{f}$ accepts an m-element input and returns an n-element vector output.	Pending
236	Equation replacement	$$f_0 : \begin{bmatrix} 4 & 11 & 8 \\ 9 & 8 & 1 \\ 15 & 0 & 6 \end{bmatrix} + \begin{bmatrix} 10 & 5 & 4 \\ 1 & -2 & -1 \\ -6 & -4 & -3 \end{bmatrix} = \begin{bmatrix} 14 & 16 & 12 \\ 10 & 6 & 0 \\ 9 & -4 & 3 \end{bmatrix} + 1 = \begin{bmatrix} 15 & 17 & 13 \\ 11 & 7 & 1 \\ 10 & -3 & 4 \end{bmatrix}$$	Pending

Page	Error	Correction	Print corrected
257	Equation replacement	$$\begin{aligned}\frac{\partial E}{\partial \boldsymbol{x}} &= \frac{\partial E}{\partial \boldsymbol{y}}\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} \\ &= \left[\frac{\partial E}{\partial y_0}\frac{\partial y_0}{\partial x_0}\ \frac{\partial E}{\partial y_1}\frac{\partial y_1}{\partial x_1}\ \ldots\right]^\top \\ &= \left[\frac{\partial E}{\partial y_0}\sigma'(x_0)\ \frac{\partial E}{\partial y_1}\sigma'(x_1)\ \ldots\right]^\top \\ &= \frac{\partial E}{\partial \boldsymbol{y}} \odot \boldsymbol{\sigma}'(\boldsymbol{x}) \qquad (10.10)\end{aligned}$$	Print 3
261	`self.delta_w += np.dot(self.input.T, output_error)`	`self.delta_w += np.dot(weights_error)`	Print 3
265	. . . so we reshape the training data from (60000,**196**) to (60000,1,196) . . .	. . . so we reshape the training data from (60000,**14,14**) to (60000,1,196) . . .	Pending
286	**As before, we** begin at $x = 0.75$ . . .	**We** begin at $x = 0.75$ . . .	Pending
307	URL update	You can find them here: *https://www.cs.toronto.edu/~hinton/coursera_lectures.html*	Print 2