1、P101作者修订时新增下面这段话

**随机拆分**

PyTorch的random\_split()方法是一种简单且成熟的执行**训练-验证拆分**的方法。

到目前为止，一直在使用基于Numpy原始拆分构建的x\_train\_tensor和y\_train\_tensor来构建**训练数据集**。现在，将使用来自Numpy的**完整数据**（和），**首先**构建一个PyTorch数据集，然后才使用random\_split()**拆分**数据。

|  |  |
| --- | --- |
|  | 虽然有一个有趣的理由驱使我选择42这个数字作为随机种子，但我还是会使用其他数字（主要是奇数）作为种子，这只是因为我更喜欢它们:-) |

|  |  |  |
| --- | --- | --- |
|  | 从1.13版开始，PyTorch的random\_split()方法将分数作为拆分的参数（类似于Scikit-Learn的train\_test\_split()）。在下面的示例中，不再需要手动计算n\_train和n\_val。我们可以简单地直接使用比率（分数需要加起来为一）：   |  | | --- | | ratio = .8  eps = 1e-16  train\_data, val\_data = random\_split(  dataset,  [ratio, 1-ratio+eps]  ) |   您可能想知道eps在那里做什么，对吧？事实证明，random\_split()会将每个子集中的元素数量向下舍入，因为精度问题（1-ratio等于0.19999999999999996），这可能会导致一些意想不到的结果（例如，验证集中有19个数据点）。将eps添加到余数可以防止这种情况发生（只要将其添加到表达式的末尾即可）。 |

2、P105作者修订时新增下面这段话

输出

|  |
| --- |
| OrderedDict([('0.weight', tensor([[1.9419]], device='cuda:0')),  ('0.bias', tensor([1.0244], device='cuda:0'))]) |

|  |  |
| --- | --- |
|  | 从1.9版开始，PyTorch提供了一个全新的上下文管理器：torch.inference\_mode()。它还是会禁用梯度计算，但更重要的是，它还会禁用PyTorch的内部视图跟踪，从而可以提供更好的性能。然而，在本书使用的示例中，这些差异可以忽略不计。 |

**绘制损失**

**PyTorch深度学习指南**

**编程基础 卷I**

[1] https://bit.ly/2MQoYRo。

[2] https://bit.ly/37uZgLB。

[3] https://pytorch.org/ecosystem/。

[4] https://bit.ly/2MTN0Lh。

[5] https://bit.ly/2UFHFve。

[6] https://bit.ly/2XXJkyo。

[7] https://openai.com/blog/openai-pytorch/。

[8] https://docs.fast.ai/。

[9] https://course.fast.ai/。

[10] https://course.fast.ai/。

[11] http://pyro.ai/。

[12] https://github.com/horovod/horovod。

[13] https://bit.ly/30CPhm5。

[14] https://bit.ly/3cT1aH2。

[15] <https://colab.research.google.com/notebooks/intro.ipynb>。

[16] <https://mybinder.readthedocs.io/en/latest/>。

[17] <https://www.anaconda.com/products/individual>。

[18] <https://bit.ly/2MVk0CM>。

[19] <https://pytorch.org/>。

[20] <https://developer.nvidia.com/cuda-zone>。

[21] <https://www.tensorflow.org/tensorboard>。

[22] <https://www.graphviz.org/>。

[23] <https://bit.ly/3fIwYA5>。

[24] <https://shorturl.at/eHQR5>。

[25] <https://github.com/szagoruyko/pytorchviz>。

[26] <https://docs.anaconda.com/anaconda/packages/pkg-docs/>。

[27] <https://pypi.org/>。

[28] <https://bit.ly/37onBTt>。

[29] <https://bit.ly/2AAh8J5>。

**第0章**

**可视化梯度下降**

[30] <https://github.com/dvgodoy/PyTorchStepByStep/blob/master/Chapter00.ipynb>。

[31] <https://colab.research.google.com/github/dvgodoy/PyTorchStepByStep/blob/master/Chapter00.ipynb>。

[32] <https://en.wikipedia.org/wiki/Gradient_descent>。

[33] <https://en.wikipedia.org/wiki/Gaussian_noise>。

[34] <https://en.wikipedia.org/wiki/Chain_rule>。

[35] <https://bit.ly/2BxCxTO>。

**第1章**

**一个简单的回归问题**

[36] <https://github.com/dvgodoy/PyTorchStepByStep/blob/master/Chapter01.ipynb>。

[37] <https://colab.research.google.com/github/dvgodoy/PyTorchStepByStep/blob/master/Chapter01.ipynb>。

[38] <https://en.wikipedia.org/wiki/Gaussian_noise>。

[39] <https://bit.ly/2XZXjnk>。

[40] <https://bit.ly/3fjCSHR>。

[41] <https://bit.ly/2Y0lhPn>。

[42] <https://bit.ly/2UDXDWM>。

[43] <https://twitter.com/alecrad>。

[44] <http://cs231n.stanford.edu/>。

[45] <https://realpython.com/python3-object-oriented-programming/>。

[46] <https://realpython.com/python-super/>。

[47] <https://ipython.readthedocs.io/en/stable/interactive/magics.html>。

[48] <https://bit.ly/30GH0vO>。

[49] <https://bit.ly/3g1eQCm>。

**第2章**

**重新思考训练循环**

[50] <https://github.com/dvgodoy/PyTorchStepByStep/blob/master/Chapter02.ipynb>。

[51] <https://colab.research.google.com/github/dvgodoy/PyTorchStepByStep/blob/master/Chapter02.ipynb>。

[52] <https://bit.ly/3jJtJeT>。

[53] <https://bit.ly/37vJVdG>。

[54] <https://bit.ly/39u1tbo>。

[55] <https://bit.ly/39ovRUx>。

[56] <https://www.geeksforgeeks.org/context-manager-in-python/>。

[57] <https://www.tensorflow.org/tensorboard/tensorboard_in_notebooks>。

**第2.1章**

**追求优雅**

[58] <https://github.com/dvgodoy/PyTorchStepByStep/blob/master/Chapter02.1.ipynb>。

[59] <https://colab.research.google.com/github/dvgodoy/PyTorchStepByStep/blob/master/Chapter02.1.ipynb>。

[60] <https://realpython.com/python3-object-oriented-programming/>。

[61] <https://realpython.com/python-super/>。

[62] <https://www.w3schools.com/python/ref_func_setattr.asp>。

[63] <https://pytorch.org/docs/stable/notes/randomness.html>。

[64] <https://github.com/dvgodoy/PyTorchStepByStep/blob/master/stepbystep/v0.py>。

**第3章**

**一个简单的分类问题**

[65] <https://github.com/dvgodoy/PyTorchStepByStep/blob/master/Chapter03.ipynb>。

[66] <https://colab.research.google.com/github/dvgodoy/PyTorchStepByStep/blob/master/Chapter03.ipynb>。

[67] <https://bit.ly/2GlmLO0>。

[68] <http://pyimg.co/kku35>。

[69] <https://bit.ly/34lPAlx>。

[70] <https://bit.ly/30xB9JZ>。

[71] <https://bit.ly/30vF7TE>。

[72] <https://bit.ly/2GCEL6A>。