## Neural Network Classification with



# **OPyTorch**

## Where can you get help?

- Follow along with the code
- Try it for yourself
- Press SHIFT + CMD + SPACE to read the docstring
- Search for it
- Try again





https://www.github.com/mrdbourke/pytorch-deep-learning/discussions



### "If in doubt, run the code"



	•	œ	02_pytorch_classification.ipynl × +		~
-	$\rightarrow$	СÓ	Colab.research.google.com/driv	ve/1T-s3fFbn1vYvS_OgQGles4Ei0vuEs2ip#scrollTo=RGeZvHsyHC72	역 🏦 🖈 🛈 🔝 🛤 🍖 🗄
PR	0	A File	02_pytorch_classification Edit View Insert Runtime	. <b>ipynb</b> ☆ Tools Help <u>All changes saved</u>	Comment 🗮 Share 🏟 😽
=		+ Cod	ie + lext		Disk Editing
		But i	it's more than enough to get s	started.	
4		We'r	e going to gets hands-on wit	n this setup throughout this notebook.	
>					
٢}	•	1. <b>I</b>	Make classification	<pre>(n_samples=100, *, shuffle=True, noise=None, random_state=None, factor=0.8) -&gt; tuple[Any   list NDarray[f]oating( NBit1@ iadd ll   ndarray, Any</pre>	
		Wall	Luce the make a local as ().	list   ndarray]	aloured data
		wer	TUSE THE <u>make_circles()</u> H	Make a large circle containing a smaller circle in 2d.	coloured dots.
	<b>V</b> Os	0	1 from sklearn.dataset 2 3 # Make 1000 samples 4 n_samples = 1000	A simple toy dataset to visualize clustering and classification algorithms. Read more in the User Guide <sample_generators> .</sample_generators>	↑↓⇔■‡↓∎:
			5 6 # Create circles	Parameters	
			7 X, y = make_circles(	n_samples, noise=0.03, # a little bit of noise to the dots random_state=42) # keep random state so we get the sa	ame values
		Alrig	ght, now let's view the first 5	$\boldsymbol{x}$ and $\boldsymbol{y}$ values.	
=1		[]	<pre>1 print(f"First 5 X fe 2 print(f"\nFirst 5 y</pre>	<pre>atures:\n{X[:5]}") labels:\n{y[:5]}")</pre>	
			First 5 X features: [[ 0.75424625 0.231480	74]	
				✓ 0s completed at 09:38	• ×

O PyTorch documentation – PyTo × +										
$\leftarrow$ $\rightarrow$ C $\triangle$ $ arrow$ pytorch.org/docs/stable/index		€₫≰	₹ 🛈 🔽 M	1 🗯 🚳						
<mark>O</mark> PyTorch	Get Started	Ecosystem	Mobile	Blog	Tutorials	Docs ∽ ●	Reso	urces 🗸	GitHu	
1.10 🔻	Docs > PyTorch	documentation				Edit on	GitHub	Shortcuts		
Q Search Docs	PYTORCH DOCUMENTATION							PyTorch documentation Indices and tables		
Notes [+]	Notes [+]     PyTorch is an optimized tensor library for deep learning using GPUs and CPUs.       Language Bindings [+]     Features described in this documentation are classified by release status:									
Language Bindings [+]										
Python API [-]	<i>Stable</i> : These features will be maintained long-term and there should generally be no major performance limitations or gaps in									
torch	doci	umentation. We als	o expect to mai	ntain backw	vards					
torch.nn	com	patibility (although	breaking chan	ges can happ	pen and notice					
torch.nn.functional	WILL	be given one releas	e ahead of time	.).						
torch.Tensor	Beta	a: These features ar	e tagged as Bet	a because th	he API may					
Tensor Attributes	char	nge based on user f	eedback, becau	se the perfo	ormance needs					
Tensor Views	to in	nprove, or because	coverage acros	s operators	is not yet					
torch.autograd	com	plete. For Beta feat	tures, we are co	ommitting to	seeing the					
torch.cuda	reati	reature through to the Stable classification. We are not, however,								
torch.cuda.amp	com									
torch.backends	torch.backends Prototype: These features are typically not available as part of									
torch.distributed	bina	ry distributions like	e PyPI or Conda	, except som	netimes behind					
torch.distributed.algorithms.join	run-time flags, and are at an early stage for feedback and testing.									

# "What is a classification problem?"

## **Example classification problems**

### "Is this email spam or not spam?"

To: <u>daniel@mrdbourke.com</u> Hey Daniel,

This deep learning course is incredible! I can't wait to use what I've learned!

To: daniel@mrdbourke.com Hay daniel...

C0ongratu1ations! U win \$1139239230

### Not spam

Spam

### **Binary classification**

(one thing or another)

### "What tags should this article have?"



"Is this a photo of sushi, steak or pizza?"



### **Multiclass classification**

(more than one thing or another)

Part of a series on	
Machine learning	
and	
data mining	
Problems	[show]
Supervised learning (classification · regression)	[show]
Clustering	[show]
Dimensionality reduction	[show]
Structured prediction	[show]
Anomaly detection	[show]
Artificial neural network	[show]
Reinforcement learning	[show]

**Multilabel classification** 

Machine learning

epresentation learning

Artificial intelligence

(multiple label options per sample)



## **Binary vs. Multi-class Classification**



### **Binary classification**

(one thing or another)



### **Multiclass classification**

(more than one thing or another)



## What we're going to cover (broadly)

- Architecture of a neural network classification model
- Input shapes and output shapes of a classification model (features and labels)
- Creating custom data to view, fit on and predict on
- Steps in modelling
  - Creating a model, setting a loss function and optimiser, creating a training loop, evaluating a model
- Saving and loading models
- Harnessing the power of non-linearity
- Different classification evaluation methods

(we'll be cooking up lots of code!)

### How:





## **Classification inputs and outputs**





(often already exísts, íf not, you can build one)





### Actual output

••• ,

[[0.97, 0.00, 0.03], [0.81, 0.14, 0.05], $\rightarrow$ [0.03, 0.07, 0.90],

### **Predicted output**

(comes from looking at lots of these)





### Input and output shapes (for an image classification example)



(gets represented as a tensor)

[batch\_size, colour\_channels, width, height]

Shape = [None, 3, 224, 224]or Shape = [32, 3, 224, 224](32 is a very common batch size)



problem you're working on.

## Architecture of a classification

### mode

Hyperparameter	Binary Classification	Multiclass classification
Input layer shape (in_features)	Same as number of features (e.g. 5 for age, sex, height, weight, smoking status in heart disease prediction)	Same as binary classification
Hidden layer(s)	Problem specific, minimum = 1, maximum = unlimited	Same as binary classification
Neurons per hidden layer	Problem specific, generally 10 to 512	Same as binary classification
<b>Output layer shape</b> (out_features)	1 (one class or the other)	1 per class (e.g. 3 for food, person or dog photo)
Hidden layer activation	Usually <u>ReLU</u> (rectified linear unit) but <u>can be many others</u>	Same as binary classification
Output activation	<u>Sigmoid</u> ( <u>torch.sigmoid</u> in PyTorch)	<u>Softmax</u> ( <u>torch.softmax</u> in PyTorch)
Loss function	Binary crossentropy (torch.nn.BCELoss in PyTorch)	Cross entropy ( <u>torch.nn.CrossEntropyLoss</u> in PyTorch)
Optimizer	SGD (stochastic gradient descent), Adam (see torch.optim for more options)	Same as binary classification



$1 \ \text{\# Create a model}$
2 model = nn.Sequential
3 nn.Linear(in_feat
4 nn.Linear(in_feat
5 nn.ReLU(),
6 nn.Linear(in_feat
7)
8
9 # Setup a loss functi
10 loss_fn = nn.BCEWithI
11 optimizer = torch.opt
12
13
14 # Training code
15
16 # Testing code

### (we're going to be building lots of these)



## Let's code.



See more: <u>https://pytorch.org/tutorials/beginner/ptcheat.html</u>



## Improving a mode

```
1 # Create a model
 2 model = nn.Sequential(
       nn.Linear(in_features=3, out_features=100),
       nn.Linear(in features=100, out features=100),
       nn.ReLU(),
       nn.Linear(in features=100, out features=3)
 9 # Setup a loss function and optimizer
10 loss fn = nn.BCEWithLogitsLoss()
11 optimizer = torch.optim.SGD(params=model.parameters(),
12
                               lr=0.001)
13
14 # Training code...
15 epochs = 10
16
17 # Testing code...
```

### Smaller model

### **Common ways to improve a deep model:**

- Adding layers
- Increase the number of hidden units
- Change/add activation functions
- Change the optimization function
- Change the learning rate
- Fitting for longer

(because you can alter each of these, they're <u>hyperparameters</u>)

(from a model's perspective)

```
1 # Create a larger model
 2 model = nn.Sequential(
       nn.Linear(in_features=3, out_features=128),
 3
       nn.ReLU(),
 4
       nn.Linear(in_features=128, out_features=256),
 5
 6
       nn.ReLU(),
       nn.Linear(in_features=256, out_features=128),
 7
       nn.ReLU(),
 8
       nn.Linear(in features=128, out features=3)
 9
10
11
     Setup a loss function and optimizer
12 #
13 loss fn = nn.BCEWithLogitsLoss()
14 optimizer = torch.optim.Adam(params=model.parameters(),
15
                                lr=0.0001)
16
17 # Training code...
18 epochs = 100
19
20 # Testing code...
```

Larger model





## The missing piece: Non-linearity



### Linear data

(possible to model with straight lines)

(i) "What could you draw if you had an unlimited amount of straight (linear) and nonstraight (non-linear) lines?"



### Non-linear data

(not possible to model with straight lines)

## The missing piece: Non-linearity







### Linear activation (same as original values)



### **ReLU** activation

(non-línear)

## The machine learning explorer's motto "Visualize, visualize, visualize"









**Predictions** 

## The machine learning practitioner's motto

### "Experiment, experiment, experiment"



(try lots of things and see what tastes good)





## Steps in modelling with PyTorch

1	#	Create a model
2	mc	odel = nn.Sequential(
3		<pre>nn.Linear(in_features=3, out_features=100),</pre>
4		<pre>nn.Linear(in_features=100, out_features=100),</pre>
5		nn.ReLU(),
6		<pre>nn.Linear(in_features=100, out_features=3)</pre>
7	)	
8		
9	#	Setup a loss function and optimizer
10	10	oss_fn = nn.BCEWithLogitsLoss()
11	or	<pre>ptimizer = torch.optim.SGD(params=model.parameters()</pre>
12		lr=0.001)
13		
14	#	Training code
15		
16	#	Testing code

1. Construct or import a pretrained model relevant to your

problem

model's predictions?)

- 2. Prepare the loss function, optimizer and training loop
  - Loss how wrong your model's predictions are compared to
    - the truth labels (you want to minimise this).
  - Optimizer how your model should update its internal
    - patterns to better its predictions.
- 3. Fit the model to the training data so it can discover patterns
  - Epochs how many times the model will go through all of
    - the training examples.
- 4. Evaluate the model on the test data (how reliable are our

(some common)

## **Classification evaluation methods**

**Key: tp** = True Positive, **tn** = True Negative, **fp** = False Positive, **fn** = False Negative

### **Metric Name Metric Forumla**

Accuracy	$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}$
Precision	$\mathbf{Precision} = \frac{tp}{tp + fp}$
Recall	$\mathbf{Recall} = \frac{tp}{tp + fn}$
F1-score	$\mathbf{F1}\operatorname{-\mathbf{score}} = 2 \cdot \frac{\operatorname{precision} \cdot \operatorname{recall}}{\operatorname{precision} + \operatorname{recall}}$

Confusion matrix

NA

### Code

### When to use

torchmetrics.Accuracy() or sklearn.metrics.accuracy\_score() Default metric for classification problems. Not the best for imbalanced classes.

torchmetrics.Precision() or sklearn.metrics.precision score()

Higher precision leads to less false positives.

torchmetrics.Recall() or sklearn.metrics.recall\_score()

Higher recall leads to less false negatives.

torchmetrics.F1Score() or sklearn.metrics.f1\_score() Combination of precision and recall, usually a good overall metric for a classification model.

torchmetrics.ConfusionMatrix()

When comparing predictions to truth labels to see where model gets confused. Can be hard to use with large numbers of classes.





## Anatomy of a confusion matrix



 True positive = model predicts 1 when truth is 1 • True negative = model predicts 0 when truth is 0 • False positive = model predicts 1 when truth is 0 False negative = model predicts 0 when truth is 1



Model learns patterns from here



### **Course materials** (training set)

### Generalization

The ability for a machine learning model to perform well on data it hasn't seen before.

## Three datasets

(possibly the most important concept in machine learning...)



## 1. Initialise with random weights (only at beginning)



[[116, 78, 15], ↓ [117, 43, 96], \_\_\_ [125, 87, 23],

4. Repeat with more examples

••• •

Inputs

Numerical encoding

Learns representation (patterns/features/weights)

### 2. Show examples

[[0.092, 0.210, 0.415],[0.778, 0.929, 0.030], [0.019, 0.182, 0.555],



[[0.983, 0.004, 0.013],▶[0.110, 0.889, 0.001], \_\_\_ [0.023, 0.027, 0.985],

Coat, Ankle boot, Shirt, Sandal

3. Update representation outputs (weights & biases)

> Representation outputs







### This course

\*\*\*

				Ś	
Search or jump to	Pull requests Issu	es Marketplace Exploi	re		↓ +• � <del>•</del> •
💂 mrdbourke / pytorch-dee	ep-learning Public		⊙ Unwatch 4 →	양 Fork 11	Starred 76 -
<> Code · Issues 28	1) Pull requests 🛛 🖓 Discussion	ns 🕑 Actions 🖽	Projects 1	Wiki 🕛 Security	🗠 Insights 🛛 …
<mark>ਇ main →</mark> ਇ 3 branches 🤇	<b>&gt;0</b> tags	Go to file Add fi	le • Code •	About	ŝ
mrdbourke update exercises		✓ 4a92ab9 2 days ago	140 commits	Materials for upcomi PyTorch course (wor	ing beginner-friendly k in progress).
.github/workflows	Update make_docs.yml		19 days ago	♂ learnpytorch.io	
docs	update readme and docs homep	age	9 days ago	machine-learning d	eep-learning
extras	add exercises and solutions for (	01	12 days ago	pytorch	
going_modular	update datasetup		2 months ago	Readme	
images	add ztm and pytorch logo		14 days ago	▲ MIT License	
in models	add folder for trained models		4 months ago	<ul> <li>4 watching</li> </ul>	
🗅 .gitignore	update gitignore		23 days ago	양 11 forks	
00_pytorch_fundamentals.i	cleanup annotations, add exercis	ses and solutions links	20 days ago		
01_pytorch_workflow.ipynb	fix typo		2 days ago	Releases	
02_pytorch_classification.ip	update exercises		2 days ago	No releases published Create a new release	
03_pytorch_computer_visio	update 03		3 months ago		

### https://www.github.com/mrdbourke/pytorch-deep-learning

	O PyTorch
<u>PyTorch website</u> & <u>forums</u>	FROM RESE DRODUC
All things PvTorch	< PyTorch

### https://www.github.com/mrdbourke/pytorch-deep-learning/ discussions



### $\bullet \bullet \bullet \quad \Box \lor \prec \quad \lor \quad \bigtriangledown \quad \boxdot \quad \textcircled{0} \quad \textcircled{0} \quad \textcircled{0}$

mrdbourke / pytorch-deep-learning (Public)											
<> Code	⊙ Issue	es 28		îΊ Ρ	ull red	quest	S	Q) D			
+											
+ +											
Announce	ements +										
Welco	me to p	ytoro	ch-c	leep	-lea	rnin	ig D	iscu			
😽 mrdl	bourke +										
Q Search al	l discussio	ons									
Categories				6	, Di	scus	sion	s			
∞ View all						<b>↑</b> 2		6			
≽ Annour	ncements										
💬 Genera	I										
💡 Ideas											
🙏 Q&A											

Search or jump to.

### Resources

### **Course Q&A**



### **Course online book**

••• [] · < > 🛛 🕜 🔘 🗉	earnpytorch.io	5	④ ① +
	Q Searc	sh	Mrdbourke/pytorch_dee     ☆ 76    ぷ 11
Welcome to the Learn PyTorch for De progress)	ep Learning boc	ok (work in	
This will be the homepage for the online book version of the Zero to Mas	stery Learn PyTorch for Deep Le	earning course.	
This course will teach you foundations of deep learning and PyTorch (a	deep learning framework writte	n in Python).	
The course is video based. However, the videos are based on the conten	ts of this online book.		
For full code and resources see the course GitHub.			
Expected release date: Early 2022.			
Get updates: Follow the pytorch-deep-learning repolog or sign up for	or emails.		
		00. F	<sub>Next</sub> PyTorch Fundamentals
Made with Material for MkDocs			<b>y</b> q

### https://learnpytorch.io

	●●●	■ pytorch.org	· 1 + :::		. disc	cuss.pytorch.org	⊕ ≜ + ==
	O PyTorch			<b>O</b> PyTorch		Sign Up	
			Alex -	all categories  Categories Latest Top			
	FROM			Category	Topics	Latest	
Forch website &	RESEARCH	TO		Vision Topics related to either pytorch/vision or vision research related topics kornia	<b>141</b> / month	Model train, val, test workflow verification in PyTorch vision	<b>1</b> 24m
<u>forums</u>	PRODUCTION			nlp Topics related to Natural Language Processing	<b>38</b> / month	Input, output must be on the current device	<b>0</b> 34m
	PyTorch     Comparison     Comparison     PyTorch     Comparison     Comparison     PyTorch     Comparison     Comparison						
	Install >			autograd A category of posts relating to the autograd engine itself.	<b>70</b> / month	Calculate differential of a tensor before loss	<b>1</b> 1h
				mixed-precision	7 / month	The loss is not decreasing	<b>0</b> 1h
	PyTorch 1.10 Release, including CUE	DA Graphs APIs, TorchScript improvements	>	C++ Topics related to the C++ Frontend, C++ API or C++ Extensions	<b>37</b> / month	How to only sample data from a dataloader with target == class	<b>1</b> 1h
things PvTorch				distributed	35 / month		

