- PyTorch & Colab
  a. Setup
     i. Google Colab
        ● ...easier to set up
     ii. NCC / local
        ● ...a bit more responsive...
        ● visdom
  b. Tensors
     i. Dynamic graph
  c. Devices

- Typical approach
  a. Data loaders
  b. Transforms
  c. Networks
     i. Init
     ii. Forward
  d. Setup optimisation strategy
  e. Training
     i. Sample
     ii. Zero grad
     iii. Forward
     iv. Loss
     v. Backward
     vi. Step
  f. Testing
  g. Plotting

\[ \theta(t+1) = \theta(t) - \eta \nabla_{\theta} L(\theta) \]
**Note:** this lecture is a live programming demonstration. The slides are minimal. The material can be downloaded here:

https://github.com/cwkw/ml-materials
The dataset doesn’t fit in memory...

35,000 * 3 * 256 * 256 * (32 bits) = 27.52 gigabytes *not including the model!*
We run out of memory...

Try and create the tensor

```
In [1]: import torch
In [2]: input = torch.zeros(35000, 3, 256, 256)
```

```
RuntimeError Traceback (most recent call last)
<ipython-input-2-8ca165dcec32> in <module>()
----> 1 input = torch.zeros(35000, 3, 256, 256)

RuntimeError: $ Torch: not enough memory: you tried to allocate 25GB. Buy new RAM!
```

```
In [3]: 
```
We run out of memory...

35,000 * 256 * 256 * 3 * (32 bits) = 27.52 gigabytes not including the model!
The Data Loader

- Split dataset into mini batches each iteration

Deep Function

```
"batch" 16
256 256
```

```
Predictions
16
2
```

```
Labels
35,000
2
2
```
The Data Loader

- Shuffle dataset to get unique batches

Deep Function

```
"shuffle"
```

```
"batch"
```

```
256
```

```
256
```

```
16
```

```
35,000
```

```
16
```

```
2
```

```
2
```
Mini batches in memory

```
(root) chris@chris-lab ~/repos/deep-learning $ master ▶ ipython
Type 'copyright', 'credits' or 'license' for more information
IPython 6.1.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import torch
In [2]: images = torch.zeros(16, 3, 256, 256)
In [3]: output = torch.zeros(16, 2, 1, 1)
In [4]: images.size()
Out[4]: torch.Size([16, 3, 256, 256])
In [5]: output.size()
Out[5]: torch.Size([16, 2, 1, 1])
In [6]:
```
Two simple networks on **Fashion MNIST**

- **32*32 = 1024 pixels**
- **10 output classes**
Lecture Materials

https://github.com/cwkx/ml-materials

Homework:

https://pytorch.org/tutorials/

NCC documentation:

http://ncc.clients.dur.ac.uk/

- You need to request access and read the documentation.

Colab:

http://colab.research.google.com/