To-Do's before Tuesday the 1st of June, 19:00.

To be full prepared for Tuesday's online session with a mentor, you should do:

[MUST] Learn all important concepts and run all lines of code in Chapter 4, MNIST Basics

This session digs into the core of what Machine Learning and Deep Learning is and how it is implemented in code. All concepts addressed in this chapter will be called upon in the rest of the book.

It also demystifies some of the concepts and jargon of deep learning: e.g. architecture, weights/bias, forward pass, back-propagation, loss function, gradient, … by showing how they are coded. It is therefore very important to spend time understanding the code itself. And possibly to "rephrase" the code with your own style to ensure you understand it.

Pay special attention to all details. If you do not understand any aspect, ask questions.

[OPTIONAL] You can utilize the video materials from fast.ai course

[MUST] Write down an article with not less than 500 words and publish it on Medium, based on one of the question below:

The questions below cover each of the main aspects in the notebook. In this week's article, focus on understanding better one particular aspect in depth and share your learning (intuition as well as key details and how things are coded) to your fellow students through the article and the PPT.

Choose one topic below and confirm it by adding your name next to the selected topic.
One person per topic.

1. **Computers and Images** [Add a name]
   Overview of how black and white images are represented in a computer. Base your answer mainly on the content of the chapter's notebook, including:
   - PIL
   - Data structures for images (numpy array, tensor rank 2).
     (What is the rank of an array/tensor?)

   Show how to crop images to the top left, top right and/OR center using simple code, by using both PIL and fastai `show_image()`.

   How can a tensor of rank 3 represent a batch of images?

   Generalize how color images can be represented on a computer. What would be the rank of the image tensor. What would be the rank of a batch of color images?

2. **Code on tensors representing images** [Add a name]
   - In fastai we use tensors to represent images. Show some of the code we can use to manipulate images. See section "*NumPy Arrays and PyTorch Tensors*" in the notebook for reference.

   - What does this code do? What is 0?

   ```
   mean3 = stacked_threes.mean(0)
   ```

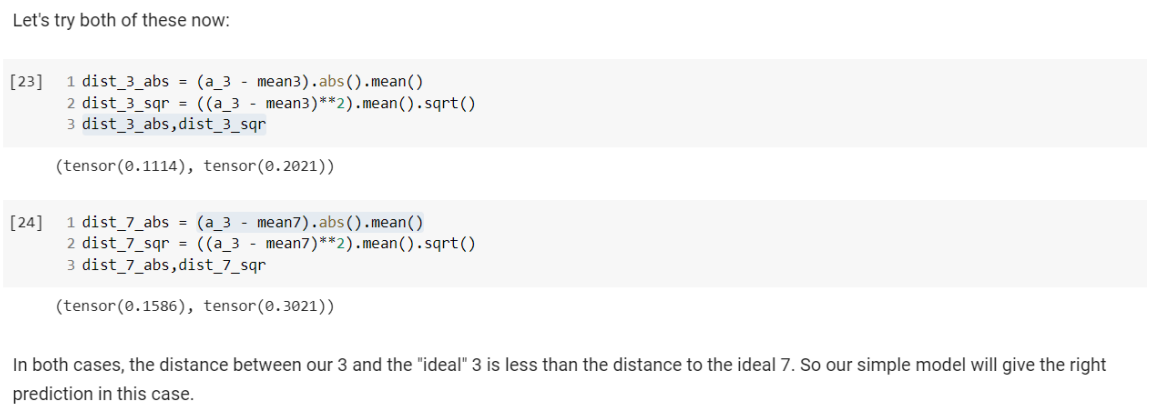
   You can read doc for pytorch.mean or/and test with following code to get a hint

   ```
   for i in range(3):
       display(stacked_threes.mean(dim=i).shape)
   ```

   - What is **broadcasting** with tensors (or numpy arrays)? See section "*Computing Metrics Using Broadcasting*". For more details, you can also consult the following notebook: "04_broadcasting-on-numpy-arrays.ipynb" on the vtecftwy/fastbook repo on github.

3. **Model Baseline: Pixel Similarity** [Add a name]
   Overview of this basic approach:
   - What is the reasoning and steps?
   - Describe what the distance means. Give your thoughts to Jeremy's question:
   "*Let's now pick an arbitrary 3 and measure its distance from our ideal digits.*
   Stop and Think!: How would you calculate how similar a particular image is to each of our ideal digits? Remember to step away from this book and jot down some ideas before you move on! Research shows that recall and understanding improves dramatically when you are engaged with the learning process by solving problems, experimenting, and trying new ideas yourself"

   To experiment, visualize the "distance"e.g. show sample image (im3), stacked 3 image and stacked 7 image, errors/distance vs 3 and 7. To do so you can run the code below (to insert after the cells in the screenshot below). Try to understand what it is doing and certainly reflect on what the output means. Feel free to modify the code as well. Or you can use your own code to visualize the different options.



Let's try both of these now:

```
[23]  1 dist_3_abs = (a_3 - mean3).abs().mean()
      2 dist_3_sqr = ((a_3 - mean3)**2).mean().sqrt()
      3 dist_3_abs,dist_3_sqr

      (tensor(0.1114), tensor(0.2021))

[24]  1 dist_7_abs = (a_3 - mean7).abs().mean()
      2 dist_7_sqr = ((a_3 - mean7)**2).mean().sqrt()
      3 dist_7_abs,dist_7_sqr

      (tensor(0.1586), tensor(0.3021))
```

In both cases, the distance between our 3 and the "ideal" 3 is less than the distance to the ideal 7. So our simple model will give the right prediction in this case.

SAMPLE CODE:

```
fig, axs = plt.subplots(nrows=1, ncols=3, figsize=(10,5))
show_image(mean3, title='Mean 3', ax=axs[0])
show_image(mean7, title='Mean 7', ax=axs[1])
show_image(a_3, title='Image to compare', ax=axs[2])
plt.show()

fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(10,10))
x, y = 0, 0
show_image((a_3 - mean3), title=f'Simple difference to 3', ax=axs[x,y])
x, y = 0, 1
show_image((a_3 - mean7), title=f'Simple difference to 7', ax=axs[x,y])

x, y = 1, 0
show_image((a_3 - mean3).abs(), title=f'Absolute value difference to 3:
{dist_3_abs:1.4f}', ax=axs[x,y])
x, y = 1, 1
show_image((a_3 - mean7).abs(), title=f'Absolute value difference to 7:
{dist_7_abs:1.4f}', ax=axs[x,y])

x, y = 2, 0
show_image((a_3 - mean3).square(), title=f'Squared difference to 3:
{dist_3_sqr:1.4f}', ax=axs[x,y])
x, y = 2, 1
show_image((a_3 - mean7).square(), title=f'Square difference to 7:
{dist_7_sqr:1.4f}', ax=axs[x,y])

plt.show()
```

4. **SGD: The Fundamental Steps of Deep Learning** [Add a name]
   - Summarize the main steps of Machine Learning as described by Arthur Samuel, and how this applies in our case.
   - Describe each of the 7 steps in your words
   - Use the illustration with the quadratic function case.

   Note: This may seem like a simple question, but it is not.
   What is expected is that the student picking this question goes deeper than what is written in the notebook and shares his or her intuition and findings with the others.

5. **SGD: Calculating Gradients and Pytorch** [Add a name]
   Summarize the concept of Gradient and why we calculate it. Use the example of the quadratic function to keep it simple.

   Describe how PyTorch helps with this calculation. Show code experiments that you have run yourself to better understand the concept, and share with your fellow students.

   For more examples, you can refer to the following notebook:
   “04_pytorch_autograd_tutorial.ipynb” on the vtecftwy/fastbook repo on github.
   `

6. **SGD: End to end SGD for Quadratic Function** [Add a name]
   Go through each of the steps (and the code) in section "*An End-to-End SGD Example*" and describe why each step is coded like that. Experiment with the code to be sure to understand. Use visualization or extract intermediate values if it helps explain what is happening. Do not hesitate to change the code to test your own ideas.

   Do not just run the code from the notebook, but reflect on what it means, and imagine how this could be generalized. Describe what some key principles such as back-propagation really mean when seen as a computation action.

7. **SGD: MNIST - Creating a model from scratch** [Add a name]
   This question refers to the notebook section "*The MNIST Loss Function*"

   The section shows how to manually create a dataset, a validation set, a linear layer (matrix) and a function to calculate predictions (what is called "forward pass" in deep learning, as opposed to "backward pass" which is the calculation of gradient).

   1. Reflect on the steps and what they really mean. Describe the mechanics and how it is coded and why.

   ``

   3. Show how minibatch is built and what it does by using the example of the notebook or any other one you played with. Can you think how augmentation would work now that you understand how mini-batch work? What would be needed to make it work?

8. **SGD: MNIST - Putting it all together** [Add a name]
   This notebook section ("*Putting It All Together*") follows the normal steps for creating a deep learning architecture and the functions to estimate predictions for specific inputs (forward pass) as well as the gradient calculation (backward pass) required to adjust the parameters (learning). Reflect on this and its implications. Some ideas to guide your thinking on this topic:
   - Go through the code and show that it is in line with the 7 steps discussed at the start of the notebook. Identify what is done for each step.
   - What do people mean when they refer to the training loop?
   - What is the specific architecture used in this simple model?
   - Where in the code do we define the architecture?

9. **SGD: MNIST - Optimizer** [Add a name]
   Section "*Creating an Optimizer*", shows how we structure the "manual" model we have built above into more general concepts (dataloaders, optimisers, learner) that are classes in PyTorch and fastai (remember, fastai is a package adding functionalities to PyTorch and relying a lot on PyTorch).

   Compare the manual code and the "PyTorch/fastai" code for the same thing in order to highlight what each PyTorch/fastai basic element/primitives/objects do and mean.

   For instance: what is the code in our manual model that correspond to
   ```
   opt = SGD(linear_model.parameters(), lr)
   ```

   Reflect on some of the code in chapter "02_production" and see if you can figure what could be the underlying code (you do not have to get a working code, more like pseudo code)

10. **SGD: MNIST - Nonlinearity and Architecture** [Add a name]
    Based on the section "*Adding a Nonlinearity*". In the section we make a few changes in the model by adding nonlinearity.
    - Explain what is the nonlinearity/activation added and why we add it.
    - What does this change in our model (hint, think of model as an architecture + parameter + forward function + backward function + optimiser)
    - Compare the results of this new model with the previous one we used. Rerun code to have comparable results.

    In the section "*Going Deeper*", we change the architecture to an untrained resnet.
    - Compare the results with our other models. Try to run this resnet with a few more epochs to see improvement and compare to the 40 epochs of our "one layer" model.
    - Reflect on the importance of architecture in deep learning

[MUST] PPT of key concepts from Medium article. **You'll have 5 minutes to present it followed by 5 min Q&A with your classmates and a mentor. Stick to the key points and go straight to the core of the topic. Remember all student had gone through the notebook**

How to publish under the unpackAI publication:

- If you do not have a Medium Account, please register and send us your Medium username as can be seen in the picture.
- Write your own article and add it to the unpackAI Medium publication (https://medium.com/unpackai). How do that is here : https://help.medium.com/hc/en-us/articles/213904978-Add-a-draft-or-post-to-publication