Extended Exam Revision Question List

Questions with (!) are important questions.

**Topic 1. Introduction to deep learning**

1. (!) Why is deep learning particularly good for supervised learning on unstructured data?
2. (!) What is feature learning? Why is it more effective than using handcrafted features?
3. What are some examples of image tasks using deep learning?
4. What are some examples of text tasks using deep learning?
5. (!) What allows us to plug together different deep learning algorithms like playing Lego?

**Topic 2. The foundations of deep learning**

1. (!) Why do we need activation functions?
2. What are the most popular activation functions?
3. What is the advantage of making models deeper?
4. (!) How do gradients tell us how to change the weights?
5. (!) How does backpropagation work?
6. (!) What do optimisers do?
7. (!) Why do we need loss functions?
8. (!)What does the softmax function do?
9. (!)What does cross-entropy loss do? How does it relate to the softmax function?

**Topic 3. Programming deep learning using PyTorch and TensorFlow**

1. (i) What is the purpose of the PyTorch Dataset class?
2. (!) What functions do you need to implement on a subclass of PyTorch’s Dataset class?
3. (!) What functions do you need to implement on a subclass of PyTorch’s Module class?
4. (!) Given a model and a data loader, how do you write a basic training loop?
5. (!) What do the arguments to nn.Linear change?

**Topic 4. Convolutional neural networks (CNNs)**

1. (!) What is translation invariance?
2. Why is translation invariance good for images?
3. (!) How does a CNN achieve translation invariance?
4. (!) What is a feature map?
5. (!) What is a receptive field?
6. (!) What is downsampling? Why is that useful?
7. How do the outputs of deeper layers differ from the outputs of earlier layers? Why?
8. (!) Why do CNNs use linear / fully connected layers?
9. What is the advantage of BatchNorm?
10. What is the advantage of skip connections?

**Topic 5. Programming CNNs**

1. (!) What PyTorch layer is used for learning a convolution on images?
2. (!) What do the arguments to that layer change?
3. (!) Which PyTorch layers are used to downsample?
4. (!) How do we connect the feature-extracting convolutions with the Linear classifier?
5. (i)Why are images 4D when training a CNN?

**Topic 6. Practical deep learning**

1. What is multi-task learning? When should we use it?
2. What is curriculum learning? When should we use it?
3. How can we modify a neural network to include ancillary data (for image data?)?
4. (!) Why is debugging deep learning code so difficult?
5. (!) How can we identify underfitting and overfitting when training?
6. How can we overcome underfitting/overfitting when training neural networks?
7. How can we identify a class imbalance in our dataset?
8. (!) How can we overcome a class imbalance?
9. (!) You suspect there’s a bug in your code. What steps can you take to identify the bug?

**Topic 7. Dealing with small, labelled datasets**

1. (!) What is transfer learning (Check more detail)?
2. (!) What benefit do we hope to receive from transfer learning?
3. (!) What is unsupervised pre-training?
4. (!)What is the difference between transfer learning and unsupervised pre-training?
5. (!) What benefit do we get from data augmentation?
6. What is early stopping?

**Topic 8. Introduction to deep learning for natural language processing (NLP)**

1. (!) Why do we want to turn text into a vector?
2. (!) What are word vectors/embeddings? How do we get them?
3. Is there a difference between a word vector and a word embedding?
4. (!)What is the advantage of using word embeddings compared to one hot encoding?
5. How does Word2Vec create word vectors?

**Topic 9. Recurrent neural networks**

1. (!) What type of data is an RNN designed to work on?
2. (!) How does a recurrent neuron differ from a standard neuron?
3. (!) What is wrong with plain RNNs?
4. (!) What variant of RNNs is most common?
5. (!) What is autoregression?

**Topic 10. Advanced applications of deep learning to NLP**

1. (!) What is a causal temporal (1D) convolution?
2. What is a dilated convolution?
3. What is the advantage of dilated versus non dilated temporal convolution?

**Topic 11. Practical deep learning for NLP**

1. (!) What is text tokenisation?
2. What is the most common type of tokenisation? Character, word, n-gram?
3. (!) Why does the tokenisation need to be consistent at training and testing time?
4. (!) In practice, do people use transfer learning with LSTM or transformer models or both model types? (i.e. Compare the benefits and disadvantages of using transformer network vs. LSTMs for NLP.)

**Topic 12. Transformer networks for NLP**

1. (!) How does a self-attention layer work?
2. What is position encoding? Why is it necessary in a transformer?
3. (!) How does a transformer use the self-attention and Linear layers?
4. (!) How can we use the Hugging Face transformers library to download a pre-trained tokenizer and network?
5. How can we fine-tune a Hugging Face model on a new task with PyTorch?

**Topic 13. Deep learning for chatbots**

1. What are the two main types of chatbots?
2. How does ChatGPT (instruct GPT) use reinforcement learning to produce more desirable output?

**Topic 14. Introduction to machine learning lifecycle**

1. What are the four steps in a typical machine learning lifecycle?

**Topic 15. Data- versus model-centric machine learning operations (MLOps)**

1. (!) What is the difference between data-centric and model-centric approaches?
2. Why can’t we just collect more data to solve every data issue?
3. (!) Why do we need to retrain models over time?
4. What are some techniques to detect bad or out-of-date data?

**Topic 16. MLOps**

1. (!) We have already tested our models. Why should we monitor model performance in production?
2. (!) What’s wrong with a machine learning engineer making a model and then never touching it again?
3. How does treating ML models as a continually updated service, rather than one-time package, make production better?
4. Why are containers so useful for MLOps?

**Topic 17. Unsupervised learning**

1. (!) Can we use deep learning when we have no labels?
2. (!) What do generative adversarial networks (GANs) do?
3. (!) What are the main pieces of GANs?
4. (!) What does an autoencoder do? Why is it useful?
5. What are some other examples of creating labels from unlabelled data?

**Topic 18. Deep learning for time series analysis**

1. (!) How is time series analysis different to NLP?
2. What techniques are popular in time series analysis?
3. (!) What is backtesting? Why is it necessary for time series analysis?
4. How do we incorporate static data (and dynamic data together or itself) in our time series analysis?

**Topic 19. Deep reinforcement learning**

1. (!) Reinforcement learning is good for what kind of problem?
2. (!) What do the following terms mean? Environment, agent, policy, reward, state, action.
3. (!) What is the long-term reward of a state? How is it different from immediate reward?
4. What is the greedy policy?
5. Why do we iteratively update the long-term expected reward?
6. (!) What is the discount factor used for?
7. What is the optimal policy?
8. (!) In complex scenarios, why do we determine long-term reward for a state-action pair, rather than a state?
9. (!) Why is deep learning good for reinforcement learning in a complex environment?

**Topic 20. Object detection**

1. (!) What is challenging about multi-object detection?
2. How does Fast RCNN work faster than RCNN?
3. How does region of interest pooling work?
4. How does Faster RCNN work faster than Fast RCNN?
5. (!) What does the Region Proposal Network (RPN) do?
6. (!) How do one-stage and two-stage detectors differ?
7. How does CornerNet differ from other detectors?

**Topic 21. Semantic segmentation**

1. (!) What are fully convolutional neural networks?
2. (!) Why are fully convolutional neural networks good for segmentation?
3. How do we both predict for every pixel efficiently, and use a large receptive field at the same time?

**Topic 22. Model interpretation and visualization**

1. (!) Why do we need to interpret model predictions?
2. What do visualizing model predictions on images look like?
3. (!) What are the benefits of class activation maps (CAMs)?

**Topic 23. Introduction to graph neural networks**

1. (!) When should we use graph neural networks?
2. In what way is transductive different to how we have trained models in the past?
3. Can we train a graph neural network with inductive learning?
4. (!) How can we make predictions using sets as input?
5. How do graph convolutions use set-based operations?

**Topic 24. Graph convolutional neural networks**

1. (!) How do dynamic graph convolutions work?
2. (!) How can we invent edges for datasets that don’t already have them?

**Topic 25. Domain adaptation**

1. (!) When is domain adaptation needed?
2. (!) What does it mean for features to be domain agnostic?
3. What are the different types of domain adaptation? Describe an algorithm for each.