Artificial Neural Networks and Deep Learning - 25/01/2021 Exam

QUESTION 1: MACHINE LEARNING / DEEP LEARNING

Consider the modern dichotomy between Machine Learning and Deep Learning and answer the following questions.

1

Let's start from the core issue of machine learning, i.e, the overfitting/generalization trade-off. Discuss if and how deep learning (as a general paradigm) differs from (classical) machine learning regarding the overfitting issue.

Hints: start by defining the concepts, then highlight the differences, then connect if/how these differences are related to overfitting/generalization ... (30 Points)

Any program that can improve its performance in a task T with respect to a metric M given experience E can be said to perform "machine learning". Deep learning is a class of algorithms that perform machine learning. In particular, these algorithms are based on deep neural networks. Neural networks in general are structures that perform a series of differentiable vector operations on an input tensor to produce an output tensor.
Overfitting is the divergence of test error from training error, which is mainly considered to be due to the learning algorithm "hallucinating patterns" in the training data rather than capturing actual patterns. This of course affects generalization, since the algorithm will not find those patterns in data that it has not seen.
The propensity to overfit is often correlated to the amount of parameters in a machine learning algorithm, since with many parameters it might be possible fit the training exactly, which would be undesirable since that would mean fitting the noise in the training data too.
Classical ML algorithms usually have less parameters, which makes them less prone to overfit but also more prone to underfit, which would also impair generalisation abilities. Underfitting happens when a model is unable to learn a task, even just on the training data.

2

Does deep learning work only for images and text? Motivate your answer. (20 Points)

No, it can work for different types of data. The core reason behind this is that a deep learning model is not aware of the meaning behind the data that is fed into it. Deep learning models work on tensors, so any data that can be meaningfully represented as a tensor should in principle be viable for deep learning

3

What are the major differences when training feed forward neural networks for regression and for classification? Motivate your answer
(20 Points)

The major differences are in the output of the network, specifically the activation function of the output layer and the loss function. This is because for the inner workings of the network it's been verified that the same general architecture can work well with either task. Of course, the output must encode the information about a class in the case of classification, and a real number in the case of regression.
If the task is classification, the activation of the output layer is sigmoid if the task is binary classification or multi label classification, and softmax if the task is classification over multiple mutually exclusive classes. Loss function is usually cross entropy.
If the task is regression, the activation of the output layer is linear because its codomain must

QUESTION 2: NEURAL NETWORKS TRAINING

Neural networks are powerful non linear approximators used to learn non linear relationships between an input vector and an output vector. The more the neurons, i.e., the more the parameters, the lower the errors they can attain. However overfitting is behind the corner ...

4

List and discuss briefly three (3) techniques used to limit overfitting in neural networks.
(30 Points)

Overfitting is what happens during training when the training error keeps decreasing but the test error plateus or starts increasing.
The simplest technique to tackle it is early stopping, in which we stop training the network when we see that the test error has stopped improving.
L1 and L2 regularization on the other hand consider a prior on the distribution of the weights, and maximize the likelihood of the output given the weights and the prior on what distribution they should follow. In practice, this amounts to a simple modification of the loss function that takes into account how big the weights are.
Dropout is a technique in which we randomly "turn off" some neurons in a layer at training time in the hope of reducing codependency between neurons and increasing generality of learned features. This implicitly creates an ensemble of weaker models.
Batch normalisation is way of performing preprocessing inside the network, by standardizing mean and covariance at a layer.
Other regularization techniques can be structural. In DenseNets, for example, it is believed that

5

What is it and how can you tune the gamma parameter of weight decay? (20 Points)

The gamma parameter in weight decay defines how much the regularization will weight in the loss function. It must not be too high, or it will overcome the gradient and just force down the weights senselessly, nor too low, in which case the regularization will be too weak. As is the case with many hyperparameters, the best wat to tune it is with cross validation. In general it is best to start with small values, and try increasing it by an order of magnitude if it works well, or decrese it by the same if it degrades the performance of the network. Once the appropriate

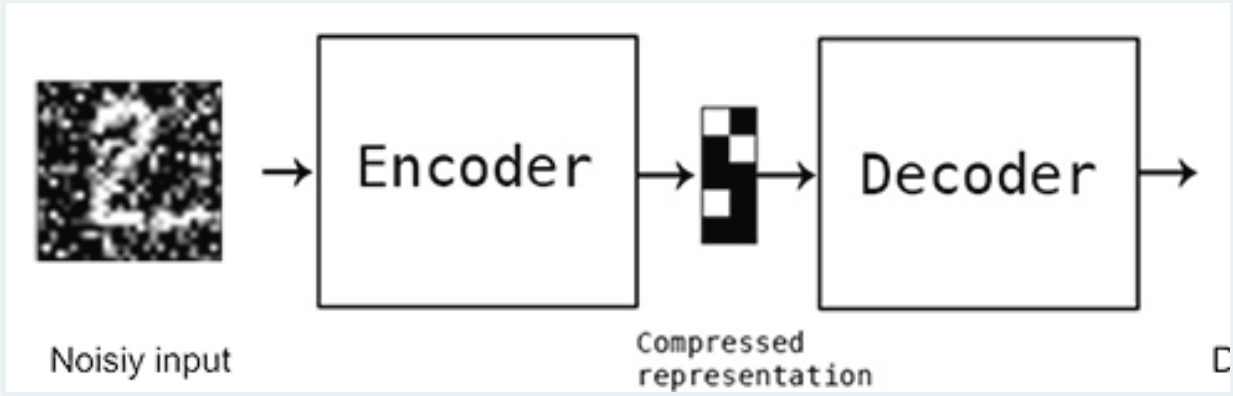
6

How do you initialize the weights of a deep neural network? Why? (20 Points)

In general we use Glorot or Xavier initialization. These techniques take into account the amount of neurons in input/output. The idea is that if we have many inputs, then the variance of the value that enters an output neuron will become higher, and vice versa if we have few inputs. We want this variance to be correct, or the output neurons might saturate or in general be

QUESTION 3: CONVOLUTIONAL NEURAL NETWORKS

In the picture, the general schema of a Denoising Autoencoder is reported with 256x256 greyscale images as input and a desired compressed representation of 8 float numbers.



7

Enumerate the building blocks of the networks you would design to build the Denoising Autoencoder as if you were going to implement it in Keras (no need to write the Keras code) and for each of them tell the number of parameters providing a short description on how you computed them, e.g., 3x5x5=45 and not just 45. (Yes you can use the calculator, but we are more interested in the formula than on the numbers)
(30 Points)

A simple architecture would be:
6 blocks of convolution with 3 by 3 kernels and depth starting at 8 and doubling each block, followed by max pooling
these would have (3x3x1x8+8)+(3x3x8x16+16)+(3x3x16x32+32)+(3x3x32x64+64)+(3x3x64x128+128)+(3x3x128x256) parameters.
At this point we have a 4x4x256 tensor which represents the image, which we can flatten and pass through a first dense layer with 1024 neurons and a second one with 256 neurons, before finally arriving at the compressed representation with a third dense layer with 8 neurons. This part would have (4096*1024+1024)+(1024x256+256)+(256x8+8) parameters.
In the expanding branch of the network I would use the same logic and sizes, but I would

8

Describe the training procedure for the Denoising Autoencoder in the picture in terms of the dataset you need and the loss function you would use.
(20 Points)

I'd use a dataset of clean images, such as the MNIST digit dataset in this case. To generate an (x, y) pair I would take a clean image as y, and then add some noise to get a noisy x. Then I would train the network to output the original image from the noisy one. Since it looks we are dealing with black and white digits, the output would be a 2d tensor of values between 0 and 1 that are strongly polarized to either 0 or 1, so I would use cross entropy on every pixel as a loss function.

Sequential Models

9

What kind of applications can be solved with a Recurrent Neural Network? Make two (2) examples and discuss their characteristics.
(20 Points)

In principle, tasks like translation or voice to text can be tackled with RNNs. They both feature input data that are temporally correlated and sequential, which is the basic feature that points us in the direction of RNNs rather than other models. Of course, simple RNNs might prove too weak to satisfactorily perform these tasks, but they would also reasonably perform better than the even more simple feedforward NNs.
Both the tasks I mentioned are of the many2many type, but this does not necessarily need to be the case to choose RNNs. A many2one task such as sentiment analysis or a language

10

Recurrent Neural Networks can suffer the Vanishing Gradient problem. Discuss what it is due to and how to solve it. Does this problem exist in feed forward neural networks?
(30 Points)

The vanishing gradient problem happens due the repeated multiplication of numbers with absolute value smaller than 1. This in turn happens in the computation of the gradient with the chain rule. This problem can exist in any sufficiently deep network, even feedforward ones, especially if the network uses s shaped activation functions that can saturate. RNNs, however, suffer from this problem more than other architectures because backpropagation through time effectively multiplies the depth of the network by the amount of time steps. A first solution to the problem was the use of ReLU activations which do not saturate, but when that proved to not be enough, the community turned to more complex models such as LSTMs or GRUs. These types

11

The current state of the art in text modeling and prediction is the Transformer. Is it implemented with a Recurrent Neural Network? Does it suffer the Vanishing Gradient problem? Motivate your answer.
(20 Points)

It is not implemented with RNNs since the attention mechanism allows it to elaborate the entire input sequence in parallel. However, it is worth noting that at test time it does perform a form of recurrence on the output since it uses the outputs at the previous steps to produce the next output.
The transformer utilizes residual learning both in the encoder and the decoder, which in conjuction with ReLU activations and attention, make it resilient to vanishing gradient.
Of course, residual learning affects the gradient flow through the network since it allows the gradient at the output of a residual block to directly contribute to the gradient at any layer below it, since each of those layers adds something to the output. There is also the longer path that

NEXT CLICK AND THE EXAM IS OVER !!!!!