dl-assignment-7

November 5, 2024

1 Deep Learning — Assignment 7

Seventh assignment for the 2024 Deep Learning course (NWI-IMC070) of the Radboud University.

Names: Andrew Schroeder and Fynn Gerding

Group: 17

Instructions: * Fill in your names and the name of your group. * Answer the questions and complete the code where necessary. * Keep your answers brief, one or two sentences is usually enough. * Re-run the whole notebook before you submit your work. * Save the notebook as a PDF and submit that in Brightspace together with the `.ipynb` notebook file. * The easiest way to make a PDF of your notebook is via File > Print Preview and then use your browser's print option to print to PDF.

1.1 Objectives

In this assignment you will 1. Think about the design of deep neural networks on a higher level 2. Read papers that use deep learning in an application

This week's assignment does not involve any programming.

1.2 7.1 Predicting the weather (7 points)

Most people don't like to be oustide in the rain. Fortunately, we nowadays have radar sensors, that can show the amount of precipitation over a wide area in real time. Weather services like buienradar use this data to forecast when it will rain. But the models they use are often quite simple: just move the rainclouds according to the current wind patterns.

In this assignment you will investigate whether you you can do better with deep learning.

More concretely: The task that we have in mind is to predict the amount of rain up to 3 hours in advance, for the entire country of the Netherlands. The input will be radar images of the amount of rain *currently* falling. For training, there is historical radar data going back several years. You may assume that the data has been gathered with a sample rate of 5 minutes, and at a resolution of 1500m per pixel. See the KNMI website for more information.

(a) Are there risks of unfair biases in this task? Explain your answer. (1 point)

Given that the data was taken at two locations at Den Helder and at Herwijnen, it is possible that even with linear interpolation and combining the weighted average of these results, that the data is still biased to these locations and don't accurately represent weather patterns accurately, for example in the north in Groningen. It could also be the case that certain weather patterns are underrepresented in the dataset because they don't frequently occur near the radar sights. Also the quality of radar data may be effected by local adverse weather conditions, such as thunderstorms which causes local conditions to distort the data that is meant to represent the entire country.

(b) Suppose we want to use a deep neural network for this task. How would the input data be represented? (1 point)

For now, consider using only the current radar data as input.

In the data, each timestep (every 5 minutes) is represented as a picture, averaged across both sensor stations. Every pixel corresponds to a 1.5km x 1.5km patch of land, where the pixel value is the reflection measurement, so a single channel for each pixel. The total size of a single input will therefore be the height in km divided by 1.5km by the total width in km divided by 1.5km. As there is a measurement every 5 minutes, there will be a series of these images over time.

(c) What are the targets for prediction? How are they represented? (1 point)

The task is to predict the weather in 3h from the current recording t (input). Therefore, we can use the measurement at t + 3h as a true label that we are predicting. The dimensionality would be the same as a single input picture (described above).

(d) If we want to make a prediction for the entire country of the Netherlands, how large should the input be to contain all the relevant information? (1 point)

Ideally, if one is trying to predict the weather for some region, this should be based on current data form that region and some marigin around the region. The weather of one country is not a closed system, but interacts with the weather across the borders. This may be through wind, air pressure or other factors. When looking at the issue of weather prediction from a perspective of time, any historical data that is available may help the decision. Due to the temporal causality in weather, the more recent the data, the more informative.

Rainclouds act differently depending on the terrain. So it can be useful for the model to know which areas of the radar image are above sea, rivers, forests, cities, mountains (as if), etc.

(e) How can you include this information in the input to the model? (1 point)

The additional geographical data could be provided as additional channels of the original weather reccording picture (keeping the same shape consistent). New (constant) features could be the average elevation of some 1.5km x 1.5km patch, if there is water (lakes, rivers,...), or the number of inhabitants, houses, industry, ...

(f) If information on the type of terrain is not available, could it be learned instead? If so, how? (1 point)

If some (implicit) feature is not available in the dataset, but has an influence on the weather, the deep neural network may be able to pick up on these patterns (features) by finding hidden correlations within the existing data.

The weather also depends on the time of day, in particular on whether the sun is shining. And on the time of year, especially the temperature.

(g) How could the time of day and time of year be given as inputs to the network? (1 point)

The time (dd/mm/yyyy) of some datapoint will likely have a significant influence on the prediction. As days, and seasons are repeating, periodic events, it may be most effective to encode time as a composition of sinus waves with different phases: 24h and 365 days. Like this, the model would be able to infer from the date encoding where in the cycle the current image can be located.

1.3 7.2 Weather prediction models (4 points)

Consider a model that uses *only* the current radar data and the terrain as input, to predict whether it will rain 3 hours from now.

(a) What kind of model would you use for this task? (1 point)

Since no perspective of time would need to be considered, a simple convolutional neural network (CNN) would be a good starting point. With different kernals, the network can learn to detect various spatial patterns including geographic influences on the weather. By stacking convolutional layers, the model can capture low-level features first and later integrate them into higher-level abstractions. Depended on the kind of conclusion that is desired, pooling and fully connected layers may be useful to reduce the dimensionality of the data and come to some clasification / decision.

(b) How would the performance of a model that takes only a single radar image as input compare against the simple baseline model that moves the radar image in the wind direction? (1 point)

The simple baseline does not take into account the temporal nature of the data, nor does it consider the terrain or local geographic features. Similarly the CNN when only provided with the current radar data and terrain as input would struggle to consider the temporal nature of the data, but it may learn more features like local rain patterns, dissipation, etc and improve performance compared to the baseline. However the CNN would be handicapped compared to the baseline if it was not provided the wind direction or speed. Because of this the baseline model might actually outperform the CNN because it is provided with the wind direction, which might offer more predictive power in the short term than a single radar image without wind direction.

To improve the model, it makes sense to take the temporal aspect into account, by including historical data. So radar scans from 5 minutes ago, 10 minutes ago, etc.

(c) Give two ways to include this additional data into the model. (2 points)

Given that we are transitioning to time-series data, it makes sense to train an LSTM or GRU on this data, which would better capture the temporal nature of the data. Additionally we could also use something like a transformer with masking to prevent it from using future images to inform current predictions (it should only use images from the past). In situations where we only care about short term dependencies in the data (last 5, 10, or 20 minutes), we could also encode the different radar scans as separate stacked channels in an image which is fed as an input to the initial CNN. The CNN could then use both spatial data encoded into each channel, and temporal data encoded in different channels.

1.4 7.3 MetNet (4 points)

We are not the first people to think about this problem. The paper MetNet: A Neural Weather Model for Precipitation Forecasting has tried to tackle the task of predicting rain from radar images.

(a) Have a look at the MetNet paper and compare their method to your answers in 7.1 and 7.2. What are they doing the same, what are they doing differently? (3 points)

Input encoding: The MetNet receives a four-dimenstional tensor of size `[t,w,h,c]` that correspond to data from a large patch of the United states with dimensions time, height, width, and number of channels. The time dimension comprises t=6 slizes sampled every 15 minutes over a 90 minutes interval prior to `T_x` where `T_x` is the time at which the model makes a prediction into the future. The input data has a resultion of 1 km^2 and each input patch has width and height of 1024 kilometers, leading to 1024x1024 values. The channels consist of a variety of features scuh as the MRMS radar image, 16 spectral bands of the GOES 16 satellite, latitude and longitude, as well as the hour, day and month of the input. This time data is tiled across the spatial dimensions to keep consistency.

This input is different from ours because our input only used a single input channel (radar image), considered data every 5 minutes instead of every 15 minutes, has a lower resuluation at 1.5km^2 instead of 1km^2, and also our data did not embed time information as a channel in the input. Overall the MetNet Paper encoded much more data in the input than ours did. They also include desired prediction lead time in the input in the paper and we don't. The other difference is that they use data that spans the entire continental US, which is a much larger region than the small Netherlands region.

Output encoding: We suggested that the target encoding would be an image that is taken 3 hours after the input image, with the same size. However the paper modifies this and sets the target patch to be much smaller than the input image - the target is a 64 x 64 km patch centered on the input patch which is of size 1024 x 1024. This is to provide the spatial context necessary for accurate predictions. Setting the target patch to this size also allows us to have a maximum lead time of 480 minutes because precipitation and clouds move at an average speed of 1km/minute, so we need a buffer of 480 km around the target area on all sides. This is not something we considered in our model.

Model: We hinted at the idea of using additional channels in the CNN, using an LSTM to capture the time dependency, and using transformers with attention to focus on important details. MetNet has a similar approach where they use a CNN to downsample the time slices sampled every 15 minutes, which are then passed to the temporal encodder which is an LSTM. The Spatial Aggregator then uses 8 axial self attention blocks to increase the repetive field over the input patch.

(b) Can the trained MetNet be used in the Netherlands using the data from KNMI, without retraining? How? Or why not? (1 point)

Note: such a transferred model might not be as good as a retrained model, but the question is whether it is possible at all.

Hmm it would be a stretch to say that we can use MetNet on the netherlands data without retraining. The netherlands data set does not have all the channel information that MetNet uses to make accurate predictions, additionally, the input and output patch sizes are not the same. Additionally, the model has learned patterns relevant to the US which will likely not generalize to the netherlands which has different weather patterns and trends. However, if the Dutch data was cut into pieces that matched the expected input and output dimensions, in theory it would be possible.

1.5 7.4 MetNet - version 2 and 3 (4 points)

You may have noticed that the MetNet paper came out 4 years ago (you can tell the year from the `/YYMM.NNNNN` arxiv url). After that two new models have been released by the same group: * MetNet-2: Skillful Twelve Hour Precipitation Forecasts using Large Context Neural Networks * MetNet-3: Deep Learning for Day Forecasts from Sparse Observations

(a) What method is used by MetNet-2 to get a larger receptive field in the convolutions? (1 point)

From the paper:

"MetNet-2 uses two dimenstional convolutional residual blocks with a sequence of of exponentially increasing dilation factors of size 1, 2, 4, …, 128. Dilation factors increase the receptive field of the convolution by skipping positiosn without increasing the number of parameters."

(b) How many parameters do the original MetNet and MetNet-3 have? Is one model significantly larger than the other? (1 point)

MetNet has 225M parameters while MetNet-3 has 227M parameters, so there is not a significant difference.

(c) Estimate how much CO was emitted for training MetNet-3. (2 points)

Hint: You can find information on the power consumption of google TPUs here (the reported numbers are per chip, not per pod).

Hint 2: Make a reasonable assumption for the CO / kwH (see lectures, or here)

We hvae 512 TPUv3 cores training for 7 days. So we have each TPUv3 core training for 7*24 = 168 hours. Each TPUv3 chip contains two TensorCores. According to this website: https://cloud.google.com/tpu/docs/v3, the mean power is 220 W for a TPUv3 Pod, where a pod contains 1024 chips, and where each chip contains two cores.

So there must be 1024 chips/pod * 2 cores/chip = 2048 cores/pod. So the mean power/core is (220 W/pod)/(2048 cores/pod) = 0.107 W/core.

So total power consumption in watts

$$512 \text{ cores} \times 0.107 \text{ W/core} = 54.784 \text{ W}$$

The total power consumption in kilowatt-hours

$$54.784 \text{ W} \times 168 \text{ hours} = 9203.712 \text{ Wh} = 9.203 \text{ kWh}$$

According to the provided link, the gCO_2/kWh in 2021 in the EU was about 250. Assuming this is similar for the US then we have:

$$250gCO_2/kWh \times 9.203kWh = 2300.75gCO_2$$

1.6 The end

Well done! Please double check the instructions at the top before you submit your results.

This assignment has 19 points. Version 8a85026 / 2024-10-14