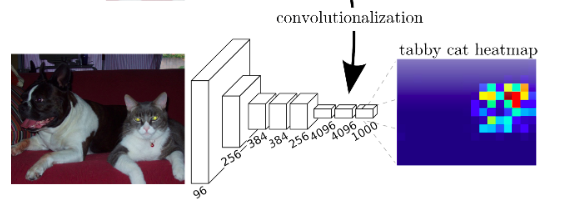
**Minutes**

16/11/22

* Focus of meeting was talking about paper: Fully Convolutional Networks for Semantic Segmentation.
* Discussed supervised pre-training – was necessary with very old models, so they could converge.
* Discussed pre and post-processing. Changes to data like combinations of pixels to a super-pixel.
  + Later discussed potentially making a ‘super amino acid’. This would be an amino acid that was actually 2 or 4 amino acids together. For example, a window of 5 would be able to actually operate on 10 or 20 amino acids instead of just the 5 the window covers.
* Discussed skip connections. Allows earlier layer weights to be updated more. With some networks, if learning has been achieved after so many layers (e.g. 16) we can skip over the rest. This lets networks be very deep, as including skip layers prevents problems (e.g. vanishing gradients) which confuses a network. An example is ResNet.
* Discussed figure 1. Although convolutions here shrink the image dimension, we can also keep the same size. Shrinking is used to reduce computational costs. But it is becoming easier to do computations as we develop more powerful GPUs.
* Discussed dilated convolution and dense CRF inference.
  + Dilated convolution is an advanced technique for finer tuning of a model.
  + Can take steps of ‘X’ distance – strides.
    - Example given is if we want to look at each in the sequence phenylalanine (F), we can position the stride such that it looks only at this position in our amino acid vector.
* Discussed figure 2.
  + 
  + Here each layer is a convolutional layer (FCN). Each channel is a different convolution part. So, there are 4096 different convolution filters in the second last layer. Then 1000 different convolution filters in the last layer. An example of a filter in the last layer is the ‘tabby cat filter/channel’ which produces the heatmap on the right.
  + With different filters – they extract different features. Can think of example: larger channelled layer may have channels with filters ‘roof, window, door’. The next layer with less channels may combine these features into a ‘house’ channel.
  + Tabby cat filter slid over entire image – mapped hotspots to where the tabby cat was found.
* So these convolution kernels will slide over an entire image/sequence – reflecting on aforementioned sliding kernel (which could be 5x5 for example, as seen in week 3 notes of DL).
* Discussed gradient accumulation. As batch size will be 1, would it be possible to have a counter, and ‘if counter mod 32 is 0’ then take the optimiser step. Discussed if this could make the same optimisation step as batching with 32, and zeroing the gradient in each training loop.
  + To do: Look at feasibility if this can work in PyTorch with inputs all different sizes.
  + This can allow stochastic gradient descent.
  + With a batch size of 1 – gradient descent could be slow/unable to reach convergence. - Is difficult to compute this all in memory at times, hence why SGD is used.
* Discussed online learning. This is similar to reinforcement learning, where one input is used at a time and the model learns from this, then progresses to the next input and so on.
* Discussed figure 4. Showing difference of detail in output of skip connection networks.

Goals for this week:

* Apply convolutions on a single protein sequence first. - These convolutions will form the FCN.
* Apply these convolutions to a few sequences in a training loop. Review if this FCN can work.
* Look at gradient accumulation.
  + Articles that have caught my eye initially: <https://kozodoi.me/python/deep%20learning/pytorch/tutorial/2021/02/19/gradient-accumulation.html> , <https://towardsdatascience.com/gradient-accumulation-overcoming-memory-constraints-in-deep-learning-36d411252d01>