Land Use and Land Cover segmentation using UNet

Model architecture:

Defined Blocks:

* Convolution block: Made of chains of convolution->convolution->batch normalization. The chain length is 2 in the short version of the model with 61,296,357 parameters, and 3 in the long version of the model with 92,740,773 parameters
* Deconvolution block: Made of chains of deconvolution->deconvolution->batch normalization. Chain length is 2 in the short version of the model, and 3 in the long version
* Downsample: Consists of a single max pooling operation
* Upsample: Consists of a single unpooling operation

Since the network is symmetrical, the outputs of the convolution blocks are fused (concatenated) with the inputs to the corresponding deconvolution blocks, which have the same shapes.

There is only one dropout layer, at the end of the downsampling path. In practice, adding even a little more dropout causes the network to severely underfit.

The output from the last deconvolution block is fed into a bottleneck convolution block which performs 1x1 convolutions with 5 filters, and applies a softmax activation to produce the output from the network. This operation maintains the height and width of the propagating tensor, while reducing the depth to the necessary dimensions.

Also, unlike in the [original UNet paper](https://arxiv.org/abs/1505.04597), this architecture doesn’t require any cropping operations while fusing the intermediary outputs, since the output size of the convolution blocks, and the input size of the deconvolution blocks, have the same shapes.

Results:

The validation set was 20% of the dataset. The rest (80%) was used for training the network for 100 epochs using both Adam and SGD optimization algorithms.

The dropout layer was configured to zero out entire channels, with a rate of 0.2.

An exponentially decaying learning rate was used. The initial value was 0.01. Every 10 epochs, the learning rate would decay by a factor of 0.9. For the SGD optimizer, a Nesterov momentum value of 0.9 was used.

A batch size of 64 was used.

The results are described in the tables below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Configuration | Number of Parameters | Optimization algorithm | Training set accuracy after 100 epochs | Validation set accuracy after 100 epochs |
| Long | 92,740,773 | Adam | 0.7944 | 0.8037 |
| Long | 92,740,773 | SGD | 0.7436 | 0.7470 |
| Short | 61,296,357 | Adam | **0.8139** | **0.8314** |
| Short | 61,296,357 | SGD | 0.8015 | 0.8051 |

The Short configuration, when trained with the Adam optimizer, results in a mIoU of **0.5455** over the validation set. The pixel-wise classification report over the validation set is in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| 1 | 0.70 | 0.51 | 0.59 | 4680386 |
| 2 | 0.81 | 0.81 | 0.81 | 34296074 |
| 3 | 0.79 | 0.83 | 0.81 | 30201961 |
| 4 | 0.00 | 0.00 | 0.00 | 387079 |
| 5 | 0.98 | 0.97 | 0.98 | 18580420 |
| Accuracy | 0.83 | | | 88145920 |
| Macro Avg. | 0.65 | 0.63 | 0.64 | 88145920 |
| Weighted Avg. | 0.83 | 0.83 | 0.83 | 88145920 |

The confusion matrix for the pixel-wise classification over the validation set is in the table below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Predicted | | | | |
| True | 2395269 | 1531856 | 718694 | 0 | 34567 |
| 728794 | 27670630 | 5763501 | 0 | 133149 |
| 263935 | 4627931 | 25112097 | 0 | 197998 |
| 906 | 180588 | 123188 | 0 | 82397 |
| 30706 | 172703 | 261916 | 0 | 18115095 |