

July 11, 2017

Zachary Kirby

Error Module

- $\text{reLU}(\text{initial input array} - \text{predicted image array}) = \text{err1 (E1)}$
- $\text{reLU}(\text{prediction array} - \text{initial image}) = \text{err2 (E2)}$
- `tf.stack(...)`
 - Creates a 5D tensor, `shape=[1, 64, 64, 2, 1]`
 - We must reshape into a usable 4D
 - Squeeze or reshape? - Doesn't matter
- `Output = [1, 64, 64, 2]`
- Output is fed against the U weight matrices

```
with tf.name_scope("ErrMod"):  
    err1      = tf.nn.relu(image - predict, name="E1")  
    err2      = tf.nn.relu(predict - image, name="E2")  
    tensor5D  = tf.stack([err1, err2],axis=3)  
    tensor4D  = tf.squeeze(tensor5D,axis=4,name='PrdErr')  
    #tensor4D = tf.reshape(tensor5D, [1, IM_SZ_LEN, IM_SZ_WID, 2], name="PrdErr")  
    return tensor4D
```

tf.slice()

- `tf.slice(input_, begin, size, name=None)`
 - `Input_` = input tensor to be sliced from
 - `Begin` = starting value of slice
 - `Size` = output tensor shape
 - `Name` = name for TensorBoard
- Only used in TensorBoard debugging
 - Has no affect on training, only used to display a 2 channel image (error stack)
- `tf.slice(err_input, [0,0,0,0], [1, 64, 64, 1])`
 - Takes the first channel of the error stack (E1)
- `tf.slice(err_input, [0,0,0,1], [1, 64, 64, 1])`
 - Takes the second channel of the error stack (E2)

Learning Rates Examined

Learning Rate	Total Steps	Step 0 Loss	Last Step Loss
.1	10	1767.37	988.958
.1	100	1643.27	955.444
.1	1000	1707.64	287.913
.01	10	1684.75	496.2
.01	100	1735.11	318.359
.01	1000	1714.8	219.518
10	10	1709.59	722.292
10	100	1654.2	647.392
10	1000	1768.47	644.19

Low learning rate gives highest loss initially, but scale longer. Steps ~300 become variant.
LOWEST LOSS ACHEIVED: 0.01x10000 -> ~170 around 5000 steps in

What does that look like?



perror_1
~300



perror_2
~300



Lstm_out
~300



Lstm_state
~300



perror_1
1000



perror_2
1000



Lstm_out
1000



Lstm_state
1000

