BATCH_SIZE*NUM_LABELS

out

$y = \exp(-w^T x + b)$

fc2_weights
512 x NUM_LABELS

all-to-one ??

fc1_weights

IMG_PATCH_SIZE / 4 * IMG_PATCH_SIZE / 4 * 64) x **512**

all-to-all

conv2:
input pool: BATCH * 5x5x3
filter: conv2_weights,
strides=[1, 1, 1, 1],
passing='SAME',
bias = conv2_biases,
pool: relu2 rectifier,
max_pool ksize=[1, 2, 2, 1],
max_pool stride=strides=[1, 2, 2, 1].

conv2_weights
5x5x32x**64**

64 filters per input to conv_weights2

conv1_weights
per block 5x5xCHANNELSx**32**

32 nodes per input to conv_weights1

conv1:
input train_data_node: BATCH * 5x5x3
filter: conv1_weights,
strides=[1, 1, 1, 1],
passing='SAME',
bias = conv1_biases,
pool: relu rectifier,
max_pool ksize=[1, 2, 2, 1],
max_pool stride=strides=[1, 2, 2, 1].

input
data [image index, y, x, depth]

training total: c0+c1 where c0==c1

road

non-road

Data trained with batch stoch. gradient descent:
input: train_data_node: BATCH * 16x16x3 (red dots)
output: train_labels_node: BATCH * NUM_LABELS

BATCH_SIZE x PATCH_SIZE x PATCH_SIZE x CHANNELS

5x5x3 ⟶ patches_vorizontal x patches_vertical x RGB

400 pixels
25 patches

corner cases:
'SAME' (*zero*) padding

400 pixels /25 patches

Image: 400 x 400 pixels
16x16 pixels per patch
(IMG_PATCH_SIZE)
25x25 patches