Sparse Labels

Understand sparse representation of labels.

Chapter Goals:

- Learn how to apply a sparse softmax cross entropy

A. Sparse representation

As mentioned in the Initialization chapter, the CIFAR-10 labels are sparsely represented. This means that, rather than being one-hot vectors, each label is just the index of its corresponding image class. Most datasets will use sparse representation for their labels, since it saves a ton of space compared to one-hot representation (especially if there are many image classes).

For training the model, we use a sparse version of softmax cross entropy. In TensorFlow, this is provided through the `tf.nn.sparse_softmax_cross_entropy_with_logits` function.

Below we show the full code for setting up and training the model:

```
import tensorflow as tf

class SqueezeNetModel(object):
    # __init__ and other functions omitted

    # Set up and run model training
    def run_model_setup(self, inputs, labels):
      logits = self.model_layers(inputs, is_training)
      self.probs = tf.nn.softmax(logits, name='probs')
      self.predictions = tf.argmax(
          self.probs, axis=-1, name='predictions')
      is_correct = tf.equal(
          tf.cast(self.predictions, tf.int32),
          labels)
      is_correct_float = tf.cast(
          is_correct,
          tf.float32)
      self.accuracy = tf.reduce_mean(
          is_correct_float)
      # calculate cross entropy
      cross_entropy = tf.nn.sparse_softmax_cross_entropy_with_logits(
          labels=labels,
            logits=logits)
        self.loss = tf.reduce_mean(
            cross_entropy)

        adam = tf.train.AdamOptimizer()
        self.train_op = adam.minimize(
            self.loss, global_step=self.global_step)
```

B. Image classification

The code below runs a squeezenet model that has been implemented in the backend. The model was trained on the CIFAR-10 dataset.

It will prompt you to upload your own image, and then print its guess for which of the CIFAR-10 classes your image depicts.

```
run_squeezenet_model()
```