Categorical Data

For the categorical data, I used a neural network with 1 hidden layer. This is because I did not expect the categorical data to be sufficient enough to form a good hypothesis, since any category of items can have any combination of categorical attributes. Thus, the purpose of this neural network would be to complement the predictions made from the other forms of data, and thus it would not make sense to make this neural net too complex.

Noisy Text Description

For the noisy text description, I used an RNN with a stack of 2 GRU cells with a linear and a log softmax layer. I used the RNN because it is ideal for text processing since it can handle variable length data. With the GRU vs a regular RNN, the GRU can control whether it memorizes or forgets certain features of the text sequence, and it is less complex and thus easier to train than a LSTM cell.

Images

For the Images, I used a convolutional neural network. The convolutional neural network offers feature recognition which makes it easier to break down the images into specific features that make it easier for the fully connected layers to classify. I used a stack of 2 smaller 3x3 filters during the convolution as opposed to 1 larger 5x5 filter because of the fewer parameters and to avoid overfitting, and also added dropout to avoid overfitting.

Ensemble Learning

To combine the hypotheses together, I used stacking. I combined the predictions from the simple neural network, the RNN and the CNN into 1 tensor and fed it all into a new neural network with 1 hidden layer. To train this model, I fed in samples from the 3 already trained models and compared the output with the actual category. Because the final prediction is a combination of the 3 predictions made by the simple neural net, the CNN, and the RNN, it made sense to train another neural network to tune exactly how the predictions from the 3 different hypotheses would be combined into a final result.