Prediction

Finally, in the last part of this lab, we will implement a prediction mechanism for our trained neural network. Our final goal is to predict one of 20 object classes for each feature vector. Thus, the output of our prediction function must be a number between 1 and 20 indicating a predicted object class for a given data point. To implement such a prediction function we must do two things:

1. Feed the provided $n \times d$ dimensional feature matrix to our trained network in a feed-forward fashion using our previously implemented `ForwardPass` function. This will produce an $20 \times n$ dimensional output matrix, which will be stored at `nn.a{end}` variable.
2. Every column on the matrix stored in `nn.a{end}` variable depicts 20 object class probabilities for a particular data observation. We will need to select an object class corresponding to the maximum object class probability. This will be our final prediction.

We note that applying this prediction scheme on the testing dataset achieves ~82% accuracy, which is a pretty solid result for a 20-class object classification task.

Your Function

Save   Reset   MATLAB Documentation (https://www.mathworks.com/help/)

```
function preds=PredictNetwork(nn,test_x)
    % produce the predictions of the trained network
    %
    % Input:
    % - nn: a structure storing the parameters of the network
    % - test_x: n x d dimensional feature matrix storing d dimensional feature for n data points
    % Output:
    % - preds: n x 1 matrix that stores the predicted object class indices ranging from values [1...20].

    %1) Perform the forward pass
    nn = ForwardPass(nn,test_x);
    %2) Select object classes cororesponding to maximum probabilities and store them into 'preds' variable
    val_mat = nn.a{end};
    [~, max_idx] = max(val_mat,[],1);

    preds= max_idx';

end
```

Code to call your function

Reset

```
load('X.mat');
load('Y.mat');
X=StandardizeData(X);
[train_x, train_y, test_x, test_y] = splitData(X, Y);
nn=TrainNetwork(train_x,train_y);
preds=PredictNetwork(nn,test_x);
```

▶ Run Function   ?

Previous Assessment: All Tests Passed

Submit   ?

✓ **Does the Predicted Output Matrix Have Correct Dimensions?**

✓ **Does the Function Return Correct Predictions?**