Hyperparameters

We have seen a lot of students take a scattershot approach to hyperparameter search, where you try random things and just hope to hit upon a winning combination. We may have oversold the difficulty of hyperparameter optimization in lecture; while randomized search is the best thing to try when you have no other intuition, there's a pretty simple recipe that I usually follow when trying to get things to work in practice.

Your first goal is to make sure the loss is decreasing within the first few iterations. Regularization can get in the way here, so turn it off - this includes L2 and dropout. Learning rate decay is for driving down loss as t goes to infinity, but it will just confuse you at the start so turn it off to start (set it to 1.0). Usually learning rate is the most important hyperparameter, with weight scale becoming important for deeper networks. Thankfully for ReLU nets there is a "correct" value for weight initialization scale - you can find it in the notes and lecture slides.

After setting your weight scale correctly, you need to find a good learning rate. First you'll want to find an upper bound for your learning rate, so keep increasing it until your loss explodes in the first couple iterations. From there, drop the learning rate by factors of 2 or so until you find one that causes loss to go down; you'll know you did this right if you see loss go down (and accuracy above chance) within 100 to 200 iterations.

Once you find this good learning rate, let the model train for an epoch or two. If you see the loss starting to plateau, then try adding learning rate decay to see if you can break through the plateau. If you see overfitting (as evidenced by a large difference between train and val accuracy) then slowly start increasing regularization (L2 weight decay and dropout).

If you see underfitting (no gap between train and val accuracy, loss converging even with weight decay, but still not hitting the accuracy targets) then you might consider increasing your model capacity either by adding extra layers or by adding neurons to your existing layers. After doing this, you'll probably have to start from the top and find a good learning rate, etc.

If you follow these tips you should be able to find hyperparameters that let you beat all the accuracy targets on the assignment within 5 or 10 epochs of training; if you are really careful you can probably beat the targets within 2 epochs of training.