3.1

* The performance increased as the number of layers increased.
* Without hidden layer, MLP is no different from a simple linear classifier;
* Adding a hidden layer with ReLU activation conveys some non-linearity, also increases the depth. Result is more features can be captured.
* Adding another hidden layer has a similar effect.
* In general the more layers there are the more complexity the network can capture.
* Each layer can be seen as an extractor that captures a specific feature in the image (e.g. edge, color, shape)

3.2

* Relu performs slightly better than leaky Relu which in turns performs better than tanh
* Tanh behaves similarly to sigmoid, and can model both positive and negative numbers.
* In contrast Relu is almost linear, except it eliminates everything below 0, which solves the vanishing gradient problem. Also more computationally efficient. Makes sense it performs better than tanh.
* Leaky relu is similar to relu, except it allows for a small gradient for negative inputs. In theory helps network converge faster.
* Several plausible reasons as to why Relu higher accuracy than leaky relu in our case. The training set was large enough, so perhaps number of epochs was insufficient. May need more training to see leaky relu's benefits.
* Leaky relu's main advantage is solving the dying neuron problem encountered in relu (large gradient updating weights such that its neuron ceases activation forever). Another hypothesis is that this problem is not encountered in the training of CIFAR10 dataset.

3.3

* Experimental results: L1 > no reg > L2
* L1 reduces some weights (features) to 0, basically reducing the dimensions by removing useless(not so relevant) features. Results in simpler model, prevents overfitting. Makes sense it has better results.
* L2 is similar to L1, except it reduces the coefficients towards 0 without reaching it. Reduces impact of irrelevant and noisy features, which should improve generalization capabilities. In theory should be better than without reg.
* Discrepancy may be due to insufficient training
* Also L2 is similar to PCA in that it reduces the dimensionality by giving less importance to redundant features. So performance depends on the correlation of the features. CIFAR10 may have low correlation.

3.4

* No convergence whatsoever
* Hypothesis 1: weights initialization expects normalized data. May cause overflow with larger inputs
* Hypothesis 2: unnormalized data may have saturated some relu units in the negative region: not updating weights during training
* Softmax computation may have resulted in overflow due to large input numbers.

3.5

* Convolutional network performs significantly better
* four key ideas behind convolutional layers that take advantage of the properties of natural signals: local connections, shared weights, pooling and the use of many layers.
* Units in a convolutional layer are organized in feature maps, within which each unit is connected to local patches in the feature maps of the previous layer through a set of weights called a filter bank
* role of the convolutional layer is to detect local conjunctions of features from the previous layer, the role of the pooling layer is to merge semantically similar features into one.