* **Batch normalization** is a technique for standardizing inputs into a network, applied to either the activations of a previous layer or direct inputs
* A **residual connection**is a skip-connection in which the residual functions are learned with reference to the layer inputs rather than learning unreferenced functions
* **Hyperparameter optimization** algorithms find the hyperparameter tuple that minimizes a predefined loss function based on independent data
* **Ensemble learning** reduces the variance of predictions and generalization error by combining predictions from multiple neural network models
* **Categorical\_crossentropy** for one-hot encoding and
* **sparse\_categorical\_crossentropy** for integer encoding.

Also note you are using the entire dataset and have not set aside a separate testing set. In other words, if the model were overfitting, you would not see that here.

* You can choose the model with lowest cross-entropy loss on the development set.
* KerasTuner as GridSerachCV
* **L1** regularization penalizes the absolute magnitude of the coefficients. To put it differently, it limits the coefficient size. As a result, L1 can produce sparse models (with few coefficients). Some coefficients can become zero and can be eliminated. Lasso regression uses this technique.
* **L2 regularization**, the square of the magnitude of the coefficients is added to the penalty. As a result, the L2 model does not yield sparse models, and all coefficients are shrunk by the same factor (none are eliminated). Regression and support vector machines use this technique.
* which hyperparameters do you select? Using the KerasTuner, use simple cross-validation rather than k-fold cross-validation.

the one that has the lowest development set cross-entropy loss

* **Overfitting** can be defined as the difference between training and validation data during each epoch of training.